

TVY-3

Technické vybavení pro 3. ročník

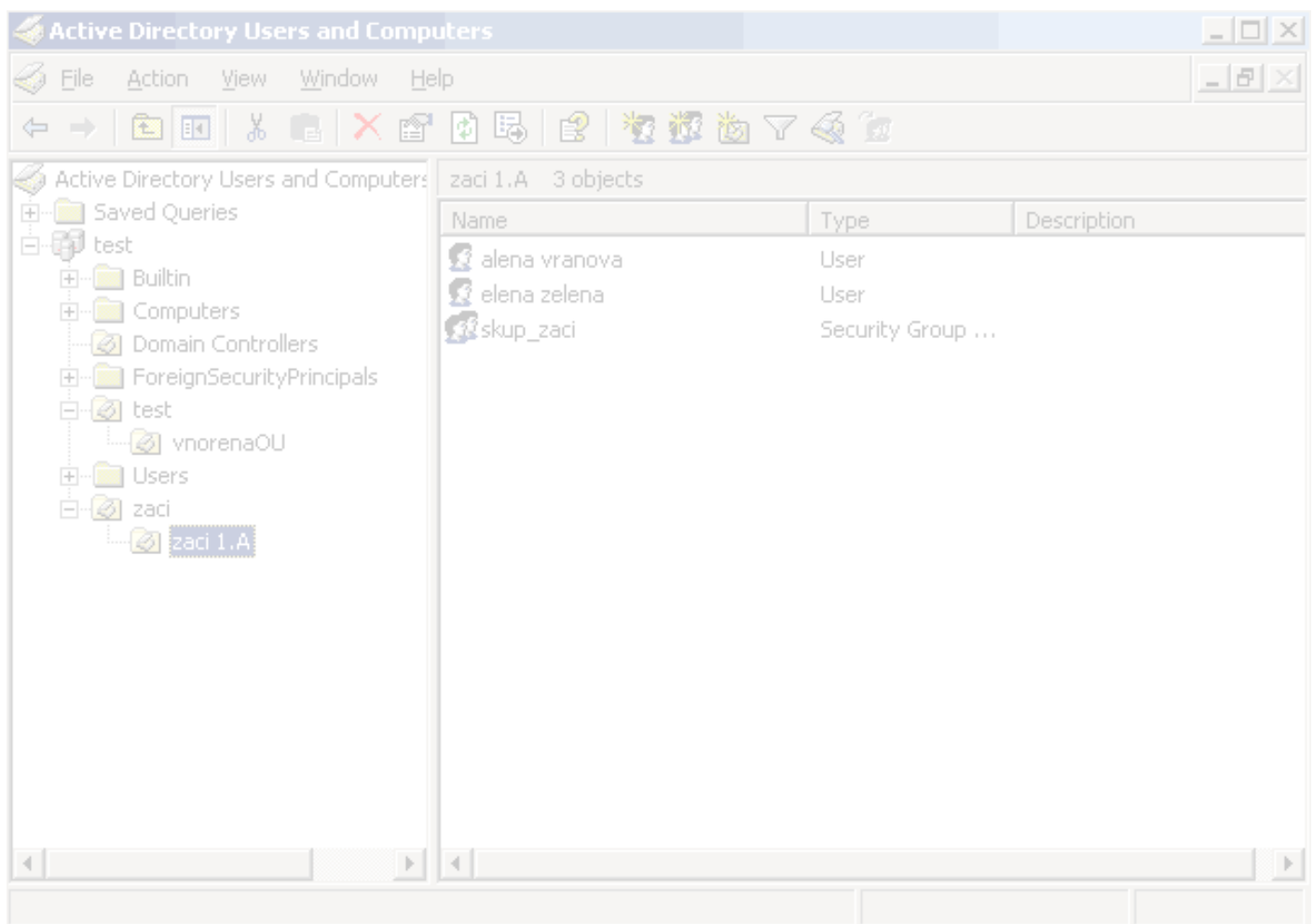


Autor: Mgr. Ivona Horáková
Smíchovská střední průmyslová škola
Vydáno v roce 2006

Učební text vznikl na základě podpory z Evropského sociálního fondu, státního rozpočtu České republiky a rozpočtu hlavního města Prahy.

TVY-3

Technické vybavení pro 3. ročník



Autor: Mgr. Ivona Horáková
Lektor: Ladislav Spurný

2006

Obsah:

1.	OS Linux – základní pojmy, historie (1. – 2. hodina)	9
2.	Struktura souborů (3. – 4. hodina)	18
3.	Příkazy (man, info, mount, umount, cd, ls, cat, tac, touch, head, tail, more, less) (5. – 6. hodina).....	24
4.	Příkazy su, whoami, pwd, clear, echo, grep, who, standardní vstupy a výstupy, chybové výstupy (7.- 8. hodina).....	39
5.	Základní příkazy – opakování (9. – 10. hodina)	53
6.	Mc, editor vi, hard link, soft link (11. – 12. hodina)	56
7.	Příkazy tee, locate, find, shutdown, cal, bc, mv, cp, mkdir, rm, rmdir (13. - 14. hodina).....	64
8.	Procesy (15. – 16. hodina).....	74
9.	Správa uživatelů (17. – 18. hodina)	87
10.	Oprávnění u souborů, adresářů (19. – 20. hodina)	95
11.	Uživatelé, oprávnění – opakování, další příkazy (21. - 22. hodina).....	102
12.	Umask, kvóty pro uživatele a skupinu (23. - 24. hodina).....	106
13.	Sledování systému (25. – 26. hodina)	112
14.	SSH (27. – 28. hodina)	125
15.	Samba (29. – 30. hodina).....	131
16.	Instalace OS Linux a nastavení sítě (31. - 32. hodina)	141
17.	Základní pojmy ze sítě (33. – 34. hodina)	177
18.	Třídy IP adres, privátní adresy, unicast, multicast, broadcast (35. - 36. hodina).....	187
19.	Podsítě (37. – 38. hodina)	193
20.	Topologie, síťová zařízení (39. – 40. hodina)	199
21.	Výroba UTP kabelů (41. – 42. hodina)	206
22.	Instalace zásuvek, patchpanely (43. – 44. hodina)	216
23.	Podsítě, konstrukce kabelů, zásuvek – prohlubování znalostí (45. - 46. hodina).....	221
24.	MS Windows Server 2003 (47. – 48. hodina).....	227
25.	Úvod do správy (49. – 50. hodina).....	236
26.	Pojmenování, uživatelské účty (51. – 52. hodina)	243
27.	Účty počítačů, změny (53. – 54. hodina).....	248
28.	Skupiny (55. – 56. hodina)	254
29.	Přístup ke zdrojům (57. – 58. hodina)	262
30.	RIS – Remote Installation Services (59. - 60. hodina).....	273
31.	Opakování (61. – 62. hodina).....	284
32.	Opakování (63. – 64. hodina).....	285
33.	Rejstřík pojmů	286

Úvod

V předmětu Technické vybavení pro 3. ročník se seznámíte se základní správou počítačů na platformě Linux GNU Debian (základní příkazy, správa účtů, nastavení, Samba, oprávnění, kvóty, SSH) a Windows Server2003 a se základními informacemi z oblasti sítí a s jejich stavbou.

Windows Server 2003 a další pokusný klientský systém Windows XP doporučuji nainstalovat do MS Virtual PC, který je od července 2006 volně ke stažení z webu Microsoftu. Oběma virtuálním strojům nastavte síťové nastavení „Local Only“, což jim umožní vzájemnou komunikaci.

1. OS Linux – základní pojmy, historie (1. – 2. hodina)

1.1 Historie

Pro zopakování zde uveďme stručný přehled vývoje Linuxu.

Kořeny Linuxu vycházejí z UNIXU.

V roce 1991 začal finský student Linus Benedict Torvalds vyvíjet základní jádro tohoto OS. Tento svůj produkt poskytl veřejnosti. V počátcích sám netušil, jak moc se tento OS rozšíří a jak široká programátorská veřejnost se do vývoje zapojí.

Svůj zdrojový kód jádra poskytl s **GPL – General Public License**, což umožňovalo komukoliv číst a měnit tento zdrojový kód. Součástí GPL je požadavek opětovného poskytnutí změněného zdrojového kódu veřejnosti.

V současnosti se na vývoji podílí mnoho lidí, na vývoj a implementace změn do jádra linuxu stále dohlíží Linus Torvalds.

Funkce jádra zahrnují kontrolu vstupů a výstupů, kontrolu a komunikaci se zařízeními, správu procesů a souborů.

1.2 K jakým účelům se dá Linux využít

Systém Linux se dá využít k nejrůznějším účelům.

Jako například

- pracovní stanice
- souborový server
- tiskový server
- webový server
- dns server
- poštovní server
- IP směrovač, firewall
- databázový server

1.3 Doporučené HW požadavky

Obecně má Linux nízké požadavky na HW. Záleží na tom, co na daném počítači nebo serveru budeme provozovat, kolik požadavků od různých uživatelů nebo aplikací bude muset počítač zpracovávat.

Pokud během instalace vybereme pouze nezbytně nutné balíky (protože každá běžící aplikace nebo služba zabírá systémové prostředky), snížíme tím nároky na HW.

Pro běžné funkce by mělo stačit následující:

- Pentium III nebo AMD 750 MHz
- 512 MB RAM
- HDD 4 GB
- síťové rozhraní (min. 100 Mb/s)

Velký diskový prostor potřebujeme u souborových serverů, databázových serverů. Rychlý disk budeme potřebovat především v případě, kdy počítač používá hodně uživatelů, často se zapisuje a čte z disku.

Stejně tak i pro paměť platí, že pokud budeme počítač zatěžovat náročnými operacemi (časté databázové dotazy, webový server, velmi vytížený poštovní server), je větší kapacita paměti výhodou.

Konkrétní HW požadavky jsou závislé na požadovém účelu a předpokládané zátěži.

Dlouhodobé zatížení serveru by nemělo přesahovat hodnotu 1. Zatížení si můžeme zobrazit například příkazem **top**.

```
linux:/etc# top
top - 15:56:32 up 4:19, 2 users, load average: 0.00, 0.02, 0.05
Tasks: total, running, sleeping, stopped, zombie
Cpu(s): us, sy, ni, id, wa, hi,
Mem: total, used, free, buffers
Swap: total, used, free, cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2446	root	15	0	25204	8032	13m	S	0.4	4.4	0:51.35	XFree86
3341	student	15	0	27560	14m	24m	S	0.2	8.1	0:19.93	kdeinit
1	root	16	0	1504	504	1352	S	0.0	0.3	0:00.82	init
2	root	34	19	0	0	0	S	0.0	0.0	0:00.00	ksoftirqd/0
3	root	5	-10	0	0	0	S	0.0	0.0	0:02.52	events/0
4	root	6	-10	0	0	0	S	0.0	0.0	0:00.00	khelper
24	root	5	-10	0	0	0	S	0.0	0.0	0:00.13	kblockd/0
48	root	15	0	0	0	0	S	0.0	0.0	0:00.01	pdflush
49	root	15	0	0	0	0	S	0.0	0.0	0:00.10	pdflush
51	root	5	-10	0	0	0	S	0.0	0.0	0:00.00	aio/0
50	root	15	0	0	0	0	S	0.0	0.0	0:00.32	kswapd0
187	root	25	0	0	0	0	S	0.0	0.0	0:00.00	kseriod
290	root	15	0	0	0	0	S	0.0	0.0	0:02.50	kjournald

Tři hodnoty u „load average“ znamenají zátěž během poslední minuty, 5 minut a 15 minut.

1.4 Charakteristika OS

Existují základní dvě verze – **stabilní** a **vývojářská**. Stabilní poznáte podle toho, že za poslední tečkou je sudé číslo (2.6), vývojářskou verzi poznáte podle lichého čísla za tečkou v označení verze.

Dále existuje celá řada distribucí linuxu, jako např. **Mandriva, Fedora, SUSE, Debian, Slackware ...**

My tu budeme používat distribuci **Debian GNU/Linux**. Během instalace je možno vybrat si z více než 15 tisíc balíčků předkompilovaného

softwaru (v roce 2005), řešení závislostí za nás udělá instalační nástroj **aptitude**. Stejně tak výborně řeší odstraňování, aktualizace apod.

Protože velká část základních nástrojů, které spolu s jádrem vytvářejí kompletní operační systém, pochází z projektu **GNU** (<http://www.gnu.org/home.cs.html>), nazýváme tento systém **GNU/Linux**.

Pro získání Debianu můžete navštívit www.debian.cz.

OS Linux je multiuživatelský a multitaskový.

1.5 Základní pojmy

Multiuživatelský znamená, že umožňuje více uživatelům používat OS současně.

Multitasking (více procesů) znamená umožnění spuštění různých procesů různými uživateli současně.

Díky multiuživatelskému a multitaskovému prostředí se zdá, že se mohou vykonávat různé procesy současně. Toto by bylo možné jen s více procesory. S jedním procesorem je to jen zdánlivé a procesy se střídají.

V UNIXU se zavedl tzv. **preemptive multitasking**. Znamená to, že každému procesu je přidělen určitý maximální čas, během kterého může pracovat. Po uplynutí tohoto času předá operační systém procesorový čas jinému procesu.

Operační systém koordinuje přístup k různým zdrojům (např. k HDD, disketě, pásce apod.).

Problém může nastat v případě, že více procesů si vyhrazuje právo pracovat s určitými zdroji.

Např. proces P1 chce pracovat se zdroji Z1 a Z2 a proces P2 se zdroji Z2 a Z1. Každý z procesů si vyhradí nejdříve právo přístupu na jeden ze dvou zdrojů (P1 na Z1, P2 na Z2). Ve chvíli, kdy chtějí přistupovat na druhý zdroj – zjistí – že daný zdroj je již využíván.

Nastává čekání, procesy se dostávají do mrtvého bodu (**dead lock**).

Řešit tento problém pomáhá tzv. **multithreading** (thread = vlákno).

V případě zmíněného problému proces předem informuje OS, které zdroje chce zamknout a teprve potom začne zamykat.

Multithreading – v rámci jednoho procesu se využívá více vláken (posloupnost instrukcí, které se vykonávají), např. v rámci jednoho procesu se zobrazuje správce souborů, zobrazují se informace o

souborech v nápovědných bublinách u ikon, současně s tím se stahuje nějaký soubor.

V Linuxu neplatí **binární kompatibilita** (program zkompileovaný pro jednu verzi nemusí běžet na jiné verzi OS, jsou zde jiné verze knihoven).

Knihovna je soubor, který obsahuje podprogramy, metody, procedury, které jsou využívány nějakým programem.

Shell – příkazový interpreter. Lze vnořovat jeden do druhého, platnost proměnných uvnitř se nepřenáší do vnějších úrovní, platí jen lokálně. Každý má své vlastní proměnné prostředí.

1.6 Zavaděč operačního systému

Po startu počítače se pomocí zavaděče operačního systému spustí operační systém. Na počítači je možné mít více operačních systémů a pomocí nabídky zavaděče zvolit, který z nich spustit.

LILO

Potřebuje vědět, na kterých sektorech je jádro uloženo. Nezná strukturu file systému. Po změně jádra se v příkazovém řádku spustí příkaz **lilo** pro aktualizaci zavaděče.

GRUB (GRand Unified Bootloader)

Po změně jádra nemusíme na rozdíl od zavaděče LILO aktualizovat zavaděč. Je výhodný, používáme-li více jader.

1.7 Souborový systém (tradiční, žurnálový, síťový)

Tradiční souborové systémy

ext2 – běžný souborový systém Linuxu

minix – první souborový systém podporovaný Linuxem

MS-DOS/VFAT – souborový systém pro DOS a Window9x. VFAT je 32 bitová verze 16 bitového souborového systému.

Žurnálové souborové systémy

ext3 – žurnálová verze souborového systému ext2

ReiserFS – zahrnuto v Linuxovém jádru z roku 2001. Dobře zachází s malými soubory nebo velkým počtem souborů. Užívá mnoho technik pro úsporu místa a zlepšení výkonu (vytváření INODE na vyžádání místo v době vzniku souboru, vytváření vyváženého stromu souborů).

NTFS – Windows souborový systém. Z Linuxu je podporováno jen čtení (zápis je ve vývoji)

Sítové souborové systémy

NFS – Network File System. Navržen pro sdílení souborů a adresářů v síti. Potřebuje konfiguraci NFS serveru, který soubory poskytuje a NFS klientů, kteří soubory používají, jako by byly přímo na jejich počítačích.

SMB – Server Message Block. Dovoluje Linuxu připojovat sdílené složky z Windows.

NCP – Netware Core Protocol. Umožňuje pracovat se sdílenými prostředky z jiného operačního systému (Novell).

Rozdíl mezi **tradičním** a **žurnálovým** systémem je v postupu ukládání dat.

Tradiční systém nejprve uloží data a poté metadata.

Žurnálový systém nejprve zapíše do žurnálu, co bude ukládat. Poté uloží data, metadata a po úspěšném zápisu smaže záznam ze žurnálu.

Výhoda žurnálového systému ukládání se projeví např. v případě pádu systému. Po nastartování systému stačí zkontrolovat záznamy v žurnálu, nemusí se procházet a kontrolovat všechny soubory. Záznam v žurnálu znamená, že bylo v plánu nějaká data uložit nebo že pád nastal v průběhu ukládání dat nebo metadat. Pokud by vše proběhlo, smazal by se záznam ze žurnálu.

1.8 Označování disků, logických jednotek, zařízení

/dev/tty1 ... 1. virtuální konzole
/dev/tty2 ... 2. virtuální konzole
atd.

/dev/ttyS0 ... sériový port COM1
/dev/ttyS1 ... sériový port COM2
/dev/lp0 ... první paralelní port

/dev/fd0 ... disketa A:
/dev/fd1 ... disketa B:

/dev/hda ... první IDE kanál master, většinou IDE hard disk
/dev/hdb ... první IDE kanál slave
/dev/hdc ... druhý IDE kanál master, většinou CD ROM
/dev/hdd ... druhý IDE kanál slave

/dev/hda1 ... první partition na hda
atd.

/dev/sda ... první SCSI disk
/dev/sda1 ... první partition na první SCSI disku

USB klíč se přiřadí na první volný sd (když není žádný SCSI disk, přiřadí se na sda, připojíme jej jako sda1)

/dev/scd0 ... první SCSI CD ROM

1.9 Rozdělení disku (fdisk)

fdisk

vypíše základní možnosti zápisu fdisk

Usage: fdisk [-l] [-b SSZ] [-u] device

E.g.: fdisk /dev/hda (for the first IDE disk)

or: fdisk /dev/sdc (for the third SCSI disk)

or: fdisk /dev/eda (for the first PS/2 ESDI drive)

or: fdisk /dev/rd/c0d0 or: fdisk /dev/ida/c0d0 (for RAID devices)

fdisk -l /dev/hda

zobrazí informace o rozdělení a souborovém systému disku a jeho logických jednotek.

Disk /dev/hda: 10.0 GB, 10056130560 bytes

255 heads, 63 sectors/track, 1222 cylinders

Units = cylinders of 16065 * 512 = 8225280 bytes

Device	Boot	Start	End	Blocks	Id	System
/dev/hda1	*	1	791	6353676	7	HPFS/NTFS
/dev/hda2		792	1156	2931862+	83	Linux
/dev/hda3		1157	1222	530145	82	Linux swap / Solaris

Upravování rozdělení disku hda

fdisk /dev/hda

Command (m for help): m **po napsání písmene m se zobrazí nápověda**

Command action

a toggle a bootable flag

b edit bsd disklabel

c toggle the dos compatibility flag

d delete a partition ... smazání partition

l list known partition types ... výpis možných typů soub. syst.

m print this menu

n add a new partition ... vytvoření nové partition

o create a new empty DOS partition table

p print the partition table ... výpis aktuálního rozdělení

q quit without saving changes ... odchod bez uložení

s create a new empty Sun disklabel

t change a partition's system id ... změna typu souborového systému

u change display/entry units

v verify the partition table

w write table to disk and exit

x extra functionality (experts only)

Command (m for help): n ... **přidávání nové partition**

Command action

e extended

p primary partition (1-4)

p

Selected partition 4

No free sectors available

Command (m for help): **t ... změna typu soub. syst.**

Partition number (1-4): 1

Hex code (type L to list codes): **L ... výpis možností soub. syst.**

```

192.168.0.2 - PuTTY
0 Empty 1e Hidden W95 FAT1 80 Old Minix be Solaris boot
1 FAT12 24 NEC DOS 81 Minix / old Lin bf Solaris
2 XENIX root 39 Plan 9 82 Linux swap / So c1 DRDOS/sec (FAT-
3 XENIX usr 3c PartitionMagic 83 Linux c4 DRDOS/sec (FAT-
4 FAT16 <32M 40 Venix 80286 84 OS/2 hidden C: c6 DRDOS/sec (FAT-
5 Extended 41 PPC PReP Boot 85 Linux extended c7 Syrinx
6 FAT16 42 SFS 86 NTFS volume set da Non-FS data
7 HPFS/NTFS 4d QNX4.x 87 NTFS volume set db CP/M / CTOS / .
8 AIX 4e QNX4.x 2nd part 88 Linux plaintext de Dell Utility
9 AIX bootable 4f QNX4.x 3rd part 8e Linux LVM df BootIt
a OS/2 Boot Manag 50 OnTrack DM 93 Amoeba e1 DOS access
b W95 FAT32 51 OnTrack DM6 Aux 94 Amoeba BBT e3 DOS R/O
c W95 FAT32 (LBA) 52 CP/M 9f BSD/OS e4 SpeedStor
e W95 FAT16 (LBA) 53 OnTrack DM6 Aux a0 IBM Thinkpad hi eb BeOS fs
f W95 Ext'd (LBA) 54 OnTrackDM6 a5 FreeBSD ee EFI GPT
10 OPUS 55 EZ-Drive a6 OpenBSD ef EFI (FAT-12/16/
11 Hidden FAT12 56 Golden Bow a7 NeXTSTEP f0 Linux/PA-RISC b
12 Compaq diagnost 5c Priam Edisk a8 Darwin UFS f1 SpeedStor
14 Hidden FAT16 <3 61 SpeedStor a9 NetBSD f4 SpeedStor
16 Hidden FAT16 63 GNU HURD or Sys ab Darwin boot f2 DOS secondary
17 Hidden HPFS/NTF 64 Novell Netware b7 BSDI fs fd Linux raid auto
18 AST SmartSleep 65 Novell Netware b8 BSDI swap fe LANstep
1b Hidden W95 FAT3 70 DiskSecure Mult bb Boot Wizard hid ff BBT
1c Hidden W95 FAT3 75 PC/IX
    
```

vybereme možnost 7 pro NTFS, c pro FAT32, 82 pro swapovací oddíl linuxu, 83 pro linux (ext2, ext3, reiser).

Hex code (type L to list codes): 7

Command (m for help): **q ... pro ukončení bez uložení, chceme-li změny zapsat, napíšeme w**

```

192.168.0.2 - PuTTY
Device Boot Start End Blocks Id System
/dev/hda1 * 1 791 6353676 7 HPFS/NTFS
/dev/hda2 792 1156 2931862+ 83 Linux
/dev/hda3 1157 1222 530145 82 Linux swap / Solaris

Command (m for help): d
Partition number (1-4): █
    
```

pro smazání oddílu napíšeme **d** a následně číslo oddílu.

1.10 Formátování

Po rozdělení disku je potřeba formátováním připravit oddíly na zápis dat.

Formátování na ext2

```
mkfs -t ext2 /dev/hda5  
mkfs.ext2 /dev/hda5  
mke2fs /dev/hda5
```

Formátování na ext3

```
mke2fs -j /dev/hda5
```

Formátování oddílu swap

```
mkswap /dev/hda6
```

1.11 Cvičení

- Co je to GPL?
- Uvedte některé z distribucí Linuxu.
- Co znamená termín multiuživatelský OS, multitasking.
- Co je to dead lock?
- Co je to knihovna (v OS Linux)?
- Co je to shell?
- Jaké typy zavaděčů OS znáte?
- Rozdělte souborové systémy a u každého uveďte typické zástupce.
- Jakým způsobem se označují zařízení v Linuxu?
- Jakým způsobem můžeme změnit rozdělení pevného disku?
- Jakým příkazem vypíšeme rozdělení pevného disku?
- Jakým příkazem zformátujeme logickou jednotku na zvolený souborový systém?

 *Domácí úkol*

Zopakujte si základní pojmy a poznatky probrané v této kapitole (historie, GPL, distribuce Linuxu, charakteristika OS Linux, problémy s dead lock, pojmy knihovna, shell, zavaděč, typy souborových systémů, označení zařízení, rozdělení a formátování disku).

 *Shrnutí*

- ✓ zopakovali jste si historii vývoje OS Linux
- ✓ seznámili jste se s pojmem GPL
- ✓ umíte uvést některé z běžných distribucí Linuxu
- ✓ umíte vysvětlit pojmy multitasking, multithreading, víceuživatelský OS
- ✓ víte, jakým způsobem řeší OS problém dead lock
- ✓ umíte vysvětlit pojmy knihovna, shell, zavaděč
- ✓ znáte rozdělení souborových systémů a u každého umíte uvést typické zástupce
- ✓ víte, jak Linux označuje disky, logické jednotky a další zařízení
- ✓ víte, jakým příkazem rozdělit a zformátovat disk

2. Struktura souborů (3. – 4. hodina)

2.1 Základní adresáře v /

☞ Poznámka

Popis hierarchie souborů v systému si můžete vypsat pomocí stránky manuálu příkazem **man hier**.

☞ Poznámka

Mezi konzolami se přepínáme klávesovými zkratkami ALT+F1 (až F7), z grafické konzoly na textovou se přepínáme stejně s navíc stisknutou klávesou CTRL.

☞ Poznámka

Výpis obsahu hlavního adresáře / vypíšete příkazem **ls /** nebo **ls -al /**.
V prvním případě se vypíše jen seznam souborů (adresář je speciálním případem souboru), ve druhém případě jde o výpis všech (a) souborů a to v podrobném formátu (l).

```
linux:~# ls -al /
drwxr-xr-x  21 root root  4096 2005-10-31 17:31 .
drwxr-xr-x  21 root root  4096 2005-10-31 17:31 ..
drwxr-xr-x   2 root root  4096 2005-10-25 13:35 bin
drwxr-xr-x   3 root root  4096 2005-10-07 13:14 boot
lrwxrwxrwx   1 root root    11 2005-10-07 13:06 cdrom -> media/cdrom
drwxr-xr-x  11 root root 24576 2005-11-06 11:38 dev
drwxr-xr-x  94 root root  4096 2005-11-06 11:38 etc
drwxrwsr-x  21 root staff 4096 2005-10-19 18:05 home
drwxr-xr-x   9 root root  4096 2005-10-20 20:49 lib
drwxr-xr-x   2 root root 49152 2005-10-07 13:05 lost+found
drwxr-xr-x   6 root root  4096 2005-10-22 15:25 media
drwxr-xr-x   3 root root  4096 2005-10-18 17:17 mnt
drwxr-xr-x   2 root root  4096 2005-10-07 13:08 opt
dr-xr-xr-x 150 root root    0 2005-11-06 12:37 proc
drwxr-xr-x  13 root root  4096 2005-10-22 18:07 root
drwxr-xr-x   2 root root  4096 2005-10-18 17:43 sbin
drwxr-xr-x   2 root root  4096 2005-10-07 13:08 srv
drwxrwxrwt  11 root root  4096 2005-11-06 11:58 tmp
drwxr-xr-x  12 root root  4096 2005-10-07 16:05 usr
drwxr-xr-x  14 root root  4096 2005-10-08 12:30 var
```

2.2 Adresář /

Je to výchozí adresář celého adresářového stromu, tzv. kořen stromu – root.

Výpis obsahu adresáře „/“ uděláte příkazem **ls /**.

2.3 Adresář bin

Obsahuje spustitelné soubory, příkazy (pro práci se soubory, konfiguraci a analýzy systému), které jsou potřeba v jednoruživatelském režimu (obdoba nouzového režimu z windows) a pro obnovu a opravu systému.

Je přístupný i běžným uživatelům.

Výpis obsahu adresáře bin uděláte příkazem **ls /bin**.

2.4 Adresář boot

Obsahuje soubory zavaděče (LILO nebo GRUB), soubory potřebné ke spuštění (kromě textových konfiguračních souborů).

Dále jsou zde informace o umístění jádra na disku. Bývá zde uloženo i jádro.

Výpis obsahu adresáře boot uděláte příkazem **ls /boot**.

2.5 Adresář dev

Každá HW komponenta, která existuje na systému (HDD, logické jednotky HDD, CD, myš, tiskárna...), je reprezentována některým souborem v tomto adresáři.

Jsou zde soubory, které jsou odkazy na jednotlivá zařízení.

Výpis obsahu adresáře dev uděláte příkazem **ls /dev**.

Poznámka

Zařízení **null** můžeme s trochou nadsázky označit za „černou díru“.

Jestliže do tohoto zařízení přesměrujeme standardní výstup nebo chybový výstup, který by se normálně zobrazil na obrazovce, tato data jsou smazána.

2.6 Adresář etc

Obsahuje konfigurační soubory, většinou v textové podobě. Běžní uživatelé mohou většinou tyto soubory číst, ale nemůžou je upravovat.

Neměly by zde být umístěny žádné spustitelné soubory. Nicméně podadresáře obsahují mnoho shellových skriptů.

Příklady některých konfiguračních souborů:

/etc/inittab ... konfigurační soubor pro spouštěcí proces, obsahuje informace např. o tom, jaké run levely odpovídají jakým číslům, co se stane při stisku CTRL+ALT+DEL atd.

Výpis obsahu souboru uděláte příkazem **cat /etc/inittab**.

V adresáři /etc/init.d najdete spoustu konfiguračních souborů pro různé služby.

/etc/fstab ... konfigurační soubor obsahující seznam automaticky připojovaných zařízení během startování systému

/etc/passwd ... databáze uživatelů (jméno, příslušnost k primární skupině atd., neobsahuje hesla), mohou ji číst i běžní uživatelé

/etc/shadow ... kryptovaná hesla uživatelů, soubor nemohou číst běžní uživatelé

/etc/group ... databáze uživatelských skupin

/etc/motd ... uvítací text, který se objeví uživateli po přihlášení (message of the day)

2.7 Adresář home

Jsou zde umístěny domovské adresáře uživatelů. Do svého domovského adresáře se z jakéhokoliv umístění v adresářové struktuře dostaneme příkazem `cd ~`.

2.8 Adresář lib

Jsou zde umístěny knihovny používané programy, které jsou umístěné v adresářích `/bin` a `/sbin`.

2.9 Adresář lost+found

Linux sem ukládá fragmenty souborů. Jsou to ztracené clustery, o kterých operační systém ví, že obsahují data některých souborů, ale kvůli poškozené struktuře je nedokáže k nějakému souboru přiřadit.

2.10 Adresář media

Obsahuje adresáře, do kterých se připojují vyměnitelná média (CD, DVD, disketa)

2.11 Adresář mnt

Je určen pro připojování např. jiného disku, síťového disku apod.

2.12 Adresář opt

Nainstalované programy mohou ukládat své statické soubory do adresáře `/opt`. Vytvoří si zde adresář svého jména a do něj své soubory ukládá.

2.13 Adresář proc

Soubory zde umístěné se vztahují k běžícím procesům a jádru.

2.14 Adresář root

Domovský adresář systémového administrátora `root`. Měl by být umístěn na stejné logické jednotce jako `/`. Předejde se tím problémům s přihlášením.

2.15 Adresář sbin (system binaries)

Jsou zde umístěny systémové binární soubory, měl by být přístupný pro změny pouze administrátorům, ostatním uživatelům jen pro čtení.

2.16 Adresář srv

Jsou zde umístěny podadresáře pro různé služby (Apache, Ftp)

2.17 Adresář tmp

Obsahuje dočasné soubory.

2.18 Adresář usr

V tomto adresáři by měla být umístěna data přístupná všem uživatelům pro čtení.

Obsahuje programy aplikací, další knihovny, sdílené adresáře obsahující dokumentaci.

V adresáři **/usr/bin** jsou umístěné binární soubory aplikací, v adresáři **/usr/share/doc** je umístěna dokumentace programů.

2.19 Adresář var

Jsou zde ukládány soubory obsahující systémové protokoly, bezpečnostní protokoly aj.

V podadresáři **/var/log** najdeme spoustu logovacích souborů, např. **dmesg**, který obsahuje hlášení vzniklá během spouštění.

Souboru **/var/log/messages** obsahuje hlášení z různých programů.

2.20 Cvičení

- Jakým příkazem vypíšete informace o systému souborů v systému?
- Jakým příkazem vypíšete obsah adresáře?
- Jak se označuje kořenový adresář celého systému, přípojný bod pro všechny soubory?
- Co obsahuje adresář bin?
- Co obsahuje adresář boot?
- Co obsahuje adresář dev?
- Co obsahuje adresář etc?
- V jakém souboru je uložen seznam lokálních uživatelů?
- V jakém souboru je uložen seznam hesel uživatelů?
- V jakém souboru je uložen seznam skupin uživatelů?
- Co obsahuje adresář home?
- Kde je většinou umístěn domovský adresář hlavního administrátora root?
- Co obsahuje adresář lib?
- Co obsahuje adresář lost+found?
- Kam se připojují vyměnitelná média?
- K čemu slouží adresář mnt?
- K čemu slouží adresář opt?
- Co obsahuje adresář proc?
- Co obsahuje adresář sbin?
- Co obsahuje adresář srv?
- Co obsahuje adresář tmp?
- Co obsahuje adresář usr?
- V jakém souboru najdeme výpis hlášení z doby startu systému?

 *Domácí úkol*

Zopakujte si, jaká je základní adresářová struktura v Linuxu, jaké typy souborů obsahují adresáře v kořenovém adresáři /.

 *Shrnutí*

- ✓ Seznámili jste se se základní strukturou souborů v Linuxu.
- ✓ Umíte popsat, jaké základní typy souborů obsahují adresáře bin, boot, dev, etc, home, lib, lost+found, media, mnt, opt, proc, root, sbin, srv, tmp, usr, var.
- ✓ Víte, jakým příkazem vypsát obsah adresáře.
- ✓ Víte, jakým příkazem vypsát obsah souboru.
- ✓ Víte, jakým příkazem se přesunete do svého domovského adresáře.

3. Příkazy (man, info, mount, umount, cd, ls, cat, tac, touch, head, tail, more, less) (5. – 6. hodina)

3.1 Příkaz man a info

Výpis nápovědy ke zvolenému příkazu lze většinou získat příkazy

man příkaz

nebo

info příkaz

3.1.1 Příklad

man ls

info ls

3.2 Příkaz mount

Zadáme-li příkaz **mount** bez parametrů, zobrazí se připojená zařízení. Je to výpis záznamů v souboru /etc/mtab.

3.2.1 Příklad

```
linux:~# mount
/dev/hda2 on / type ext3 (rw,errors=remount-ro,usrquota,grpquota)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /dev/shm type tmpfs (rw)
usbfs on /proc/bus/usb type usbfs (rw)
/dev/hda1 on /mnt/win type ntfs (rw)
```

Příkazem **mount -l -t typ_file_systému** zobrazíme všechna připojená zařízení zvoleného souborového systému.

3.2.2 Příklad

mount -l -t ext3

```
linux:~# mount -l -t ext3
/dev/hda2 on / type ext3 (rw,errors=remount-ro,usrquota,grpquota) [//]
```

mount -l -t ntfs

```
linux:~# mount -l -t ntfs
/dev/hda1 on /mnt/win type ntfs (rw)
```

Připojení zařízení uděláme příkazem **mount -t název_file_systému zařízení_které_připojujeme_místo_kam_připojit**

Název souborového systému zvolíme podle typu zařízení, které připojujeme.

Např. FAT, NTFS, EXT2, EXT3, REISER, iso9660 (pro CD a některá DVD), UDF (pro DVD).

Některé typy souborových systémů dokáže OS rozeznat sám a parametr **-t** nemusíme uvádět.

3.2.3 Příklad

```
linux:/mnt# mount /dev/hda1 /mnt/win
```

Další volby pro připojení uvádíme za přepínač **-o**.

3.2.4 Příklad

```
mount -t iso9660 -o ro /dev/cdrom /media/cdrom
```

Za přepínačem **-o** následují volby **ro**, což znamená „read only“ – pro čtení. Připojujeme zařízení CD do adresáře **/media/cdrom**.

3.2.5 Příklad

Připojení flash disku.

Vytvoříme adresář flash v adresáři **/media**, pokud ještě není vytvořen.

```
mkdir /media/flash
```

Připojíme disk

```
mount /dev/sda1 /media/flash
```

Pokud nemáme zařízení **sda** resp. **sda1** obsazené, pak se flash disk většinou připojí právě do **sda1**. Pokud zařízení **sda1** je obsazené např. SATA diskem, pak se flash disk připojí na **sdb1**.

☞ Poznámka

Nevytahujte připojený flash disk bez toho, že byste jej předem odpojili (příkaz **umount**), může to způsobit poškození dat na flash disku. Příkazem **mount** si zkontrolujte, zda flash disk není stále připojen.

Pokud za přepínač **-o** napíšem **rw**, pak je systém připojován pro „read – write“ – zápis i čtení. Tato volba je přednastavená, pokud neuvedeme jinak.

☞ Poznámka

Souborové systémy může normálně připojovat jen administrátor. Pokud je ale v souboru **/etc/fstab** uvedena volba „user“, pak může souborové systémy připojovat i běžný uživatel.

3.2.6 Příklad

V souboru `/etc/fstab` je uveden řádek

```
/dev/cdrom /media/cd iso9660 ro,user,noauto
```

Jákýkoliv uživatel může připojit souborový systém iso9660 nalezený na CD ROM s použitím příkazu

```
mount /dev/cdrom
```

nebo

```
mount /media/cd
```

Výpis souboru `fstab` uděláte příkazem

```
cat /etc/fstab
```

Poznámka

Pro další podrobnosti o příkazu `mount` se podívejte do manuálové stránky příkazem `man mount`.

3.3 Příkaz `umount`

Tento příkaz používáme pro odpojení připojeného souborového systému.

Souborový systém, který chceme odpojit, nesmí být ve chvíli odpojování používán. Pro odpojení stačí napsat zkrácený zápis – `umount místo_připojení`.

3.3.1 Příklad

Je-li příkazem `mount` připojen CD ROM do adresáře `/media/cdrom`, odpojíme jej příkazem `umount /media/cdrom`.

3.4 Příkaz `cd`

Tento příkaz slouží pro pohyb v adresářové struktuře.

```
cd /etc ... přesun do adresáře /etc.
```

Pokud bychom zapomněli na lomítko před `etc`, znamenalo by to relativní cestu vůči aktuálnímu umístění a adresář `etc` by se hledal v adresáři, ve kterém právě teď jsme.

Pokud bychom byli přímo v rootu `/`, ve kterém se adresář `etc` nachází, přepnuli bychom se do adresáře `etc` bez problémů, jinak by se vyhlásila chyba.

```
cd / ... přesun do rootu / z jakékoliv pozice.
```

```
cd ~ ... přesun do domovského adresáře aktuálně přihlášeného uživatele, stejně funguje i samotný příkaz cd bez parametrů.
```

```
cd .. ... přesun o jednu úroveň nahoru.
```

Pro zjištění celé cesty k aktuálnímu adresáři slouží příkaz **pwd** (print name of current working directory).

```
linux:~# pwd
/root
```

3.5 Příkaz ls

Příkaz **ls** slouží pro výpis obsahu adresáře.

Pokud nezadáme žádné další parametry, získáme stručný výpis, ve kterém se nezobrazí skryté soubory.

```
linux:/home/l# ls
a a.txt zahr.jpg
```

S použitím parametru **-R** získáme výpis seznamu souborů i v podadresářích (rekurzivní způsob výpisu).

```
linux:/home/l# ls -aR
.:
. a.txt .bash_profile Krava.txt novy_soubor2.txt vysledek.txt
.. Babika.txt .bashrc MaMa.dat soubor.txt zahr.jpg
a .bash_history b.dat mama.txt soubor2.txt 2004-11-24

./a:
. .. a.txt
```

Příkazem **ls -a** získáme výpis všech souborů v adresáři včetně skrytých souborů (začínají tečkou).

```
linux:/home/l# ls -a
. .. a a.txt .bash_history .bash_profile .bashrc zahr.jpg
```

Příkazem **ls -l** získáme podrobný výpis obsahu adresáře, zobrazí se atributy souborů. Skryté soubory se nezobrazí.

Ve výpisu vidíme (zleva) typ souboru (d – znamená adresář), oprávnění pro vlastníka, skupinu a ostatní (např. rwxrw-r--), počet odkazů na soubor, vlastníka, skupinu, velikost, datum poslední změny, název souboru.

```
linux:/home/l# ls -l
celkem 28
drwx----- 2 1 users 4096 2005-10-17 20:30 a
-rw-r--r-- 1 1 users 0 2005-10-18 20:27 a.txt
-rwx----- 1 1 users 22028 2005-10-18 20:24 zahr.jpg
```

Parametry lze seskupovat.

Příkazem `ls -al` zobrazíme podrobný výpis adresáře včetně skrytých souborů.

```
linux:/home/l# ls -al
celkem 48
drwx-----  3 1    users  4096 2005-10-18 20:27 .
drwxrwsr-x 21 root staff  4096 2005-10-19 18:05 ..
drwx-----  2 1    users  4096 2005-10-17 20:30 a
-rw-r--r--  1 1    users    0 2005-10-18 20:27 a.txt
-rw-----  1 1    users   31 2005-11-06 12:02 .bash_history
-rwx-----  1 1    users  567 2005-10-16 22:42 .bash_profile
-rwx-----  1 1    users 1834 2005-10-16 22:42 .bashrc
-rwx-----  1 1    users 22028 2005-10-18 20:24 zahr.jpg
```

Příkazem `ls -alh` zobrazíme podrobný výpis adresáře a velikosti souborů budou uvedeny v lidsky čitelné podobě (h – human readable).

```
linux:/home/l# ls -alh
celkem 48K
drwx-----  3 1    users 4,0K 2005-10-18 20:27 .
drwxrwsr-x 21 root staff 4,0K 2005-10-19 18:05 ..
drwx-----  2 1    users 4,0K 2005-10-17 20:30 a
-rw-r--r--  1 1    users  0 2005-10-18 20:27 a.txt
-rw-----  1 1    users  31 2005-11-06 12:02 .bash_history
-rwx-----  1 1    users 567 2005-10-16 22:42 .bash_profile
-rwx-----  1 1    users 1,8K 2005-10-16 22:42 .bashrc
-rwx-----  1 1    users 22K 2005-10-18 20:24 zahr.jpg
linux:/home/l# ls -al
celkem 48
drwx-----  3 1    users  4096 2005-10-18 20:27 .
drwxrwsr-x 21 root staff  4096 2005-10-19 18:05 ..
drwx-----  2 1    users  4096 2005-10-17 20:30 a
-rw-r--r--  1 1    users    0 2005-10-18 20:27 a.txt
-rw-----  1 1    users   31 2005-11-06 12:02 .bash_history
-rwx-----  1 1    users  567 2005-10-16 22:42 .bash_profile
-rwx-----  1 1    users 1834 2005-10-16 22:42 .bashrc
-rwx-----  1 1    users 22028 2005-10-18 20:24 zahr.jpg
```

Příkazem `ls -alS` zobrazíme podrobný výpis adresáře seřazený podle velikosti sestupně.

```
linux:/home/l# ls -alS
celkem 48
-rwx-----  1 1    users 22028 2005-10-18 20:24 zahr.jpg
drwx-----  3 1    users  4096 2005-10-18 20:27 .
drwxrwsr-x 21 root staff  4096 2005-10-19 18:05 ..
drwx-----  2 1    users  4096 2005-10-17 20:30 a
-rwx-----  1 1    users  1834 2005-10-16 22:42 .bashrc
-rwx-----  1 1    users   567 2005-10-16 22:42 .bash_profile
-rw-----  1 1    users    31 2005-11-06 12:02 .bash_history
-rw-r--r--  1 1    users     0 2005-10-18 20:27 a.txt
```

Příkazem `ls -alSr` zobrazíme podrobný výpis adresáře seřazený podle velikosti vzestupně, protože parametr `r` mění účinek řazení parametrem `S` na opačný (r – reverse).

```
linux:/home/l# ls -alSr
celkem 48
-rw-r--r--  1 1    users     0 2005-10-18 20:27 a.txt
-rw-----  1 1    users    31 2005-11-06 12:02 .bash_history
-rwx-----  1 1    users   567 2005-10-16 22:42 .bash_profile
-rwx-----  1 1    users  1834 2005-10-16 22:42 .bashrc
drwx-----  2 1    users  4096 2005-10-17 20:30 a
drwxrwsr-x 21 root staff  4096 2005-10-19 18:05 ..
drwx-----  3 1    users  4096 2005-10-18 20:27 .
-rwx-----  1 1    users 22028 2005-10-18 20:24 zahr.jpg
```

Pomocí parametru `t` můžeme řadit podle data poslední změny, nejnovější soubory budou ve výpisu uvedeny nejvýš.

```
linux:/home/l# ls -alt
celkem 48
-rw-----  1 1    users    31 2005-11-06 12:02 .bash_history
drwxrwsr-x 21 root staff  4096 2005-10-19 18:05 ..
drwx-----  3 1    users  4096 2005-10-18 20:27 .
-rw-r--r--  1 1    users     0 2005-10-18 20:27 a.txt
-rwx-----  1 1    users 22028 2005-10-18 20:24 zahr.jpg
drwx-----  2 1    users  4096 2005-10-17 20:30 a
-rwx-----  1 1    users   567 2005-10-16 22:42 .bash_profile
-rwx-----  1 1    users  1834 2005-10-16 22:42 .bashrc
```

Pomocí parametru `u` lze seřadit soubory podle data posledního přístupu.

Pokud při výpisu chceme vypsát jen soubory, které začínají zvoleným řetězcem znaků, libovolné znaky nahradíme zástupným symbolem `*`.

Na obrázku je výpis nejprve celého adresáře a pak jen takových souborů, které začínají řetězcem soub.

```
linux:/home/l# ls -al
celkem 48
drwx-----  3 1    users  4096 2005-11-12 15:06 .
drwxrwsr-x  21 root staff  4096 2005-10-19 18:05 ..
drwx-----  2 1    users  4096 2005-10-17 20:30 a
-rw-r--r--   1 1    users    0 2005-10-18 20:27 a.txt
-rw-----   1 1    users   31 2005-11-06 12:02 .bash_history
-rwx-----   1 1    users  567 2005-10-16 22:42 .bash_profile
-rwx-----   1 1    users 1834 2005-10-16 22:42 .bashrc
-rw-r--r--   1 root root    0 2005-11-12 15:06 novy_soubor2.txt
-rw-r--r--   1 root root    0 2005-11-12 15:02 soubor.txt
-rw-r--r--   1 root root    0 2005-11-12 15:06 soubor2.txt
-rwx-----   1 1    users 22028 2005-10-18 20:24 zahr.jpg
linux:/home/l# ls -al soub*
-rw-r--r--   1 root root 0 2005-11-12 15:02 soubor.txt
-rw-r--r--   1 root root 0 2005-11-12 15:06 soubor2.txt
```

Pokud dáme do výpisu `*soub*`, pak se vypíše všechny soubory, které v názvu obsahují řetězec soub.

```
linux:/home/l# ls -al *soub*
-rw-r--r--   1 root root 0 2005-11-12 15:06 novy_soubor2.txt
-rw-r--r--   1 root root 0 2005-11-12 15:02 soubor.txt
-rw-r--r--   1 root root 0 2005-11-12 15:06 soubor2.txt
```

Příkazem `ls -ald [as]*` vypíšeme všechny soubory v podrobném formátu, které začínají na písmeno **a** nebo **s**. Parametr **d** je použit proto, aby se nevypisoval obsah podadresářů.

```
linux:/home/l# ls -ald [as]*
drwx-----  2 1    users  4096 2005-10-17 20:30 a
-rw-r--r--   1 1    users    0 2005-10-18 20:27 a.txt
-rw-r--r--   1 root root  49 2005-11-12 15:24 soubor.txt
-rw-r--r--   1 root root  18 2005-11-12 15:50 soubor2.txt
```

Pokud chceme vypsát seznam souborů, které začínají písmeny např. **n** až **v**, použijeme pro definování rozsahu pomlčku.

`ls -ald [n-v]*`

```
linux:/home/l# ls -ald [n-v]*
-rw-r--r--   1 root root 0 2005-11-12 15:06 novy_soubor2.txt
-rw-r--r--   1 root root 49 2005-11-12 15:24 soubor.txt
-rw-r--r--   1 root root 18 2005-11-12 15:50 soubor2.txt
-rw-r--r--   1 root root 67 2005-11-12 15:54 vysledek.txt
```

Pokud chceme vypsat soubory začínající různými rozsahy písmen, např. a až n, B až L, lze použít pro výpis příkaz

ls -ald [a-nB-L]*

Na obrázku je vidět v prvním výpisu seznam všech souborů v adresáři a na druhém výpisu seznam souborů vyhovujících zadanému kritériu.

```
linux:/home/l# ls -al
celkem 60
drwx-----  3 1    users  4096 2005-11-12 16:11 .
drwxrwsr-x  21 root staff  4096 2005-10-19 18:05 ..
drwx-----  2 1    users  4096 2005-10-17 20:30 a
-rw-r--r--   1 1    users    0 2005-10-18 20:27 a.txt
-rw-r--r--   1 root root    0 2005-11-12 16:10 Babika.txt
-rw-----   1 1    users   31 2005-11-06 12:02 .bash_history
-rwx-----   1 1    users  567 2005-10-16 22:42 .bash_profile
-rwx-----   1 1    users 1834 2005-10-16 22:42 .bashrc
-rw-r--r--   1 root root    0 2005-11-12 16:10 b.txt
-rw-r--r--   1 root root    0 2005-11-12 16:10 Krava.txt
-rw-r--r--   1 root root    0 2005-11-12 16:11 MaMa.txt
-rw-r--r--   1 root root    0 2005-11-12 16:11 mama.txt
-rw-r--r--   1 root root    0 2005-11-12 15:06 novy_soubor2.txt
-rw-r--r--   1 root root   49 2005-11-12 15:24 soubor.txt
-rw-r--r--   1 root root   18 2005-11-12 15:50 soubor2.txt
-rw-r--r--   1 root root   67 2005-11-12 15:54 vysledek.txt
-rwx-----   1 1    users 22028 2005-10-18 20:24 zahr.jpg
linux:/home/l# ls -ald [a-nB-L]*
drwx-----  2 1    users  4096 2005-10-17 20:30 a
-rw-r--r--   1 1    users    0 2005-10-18 20:27 a.txt
-rw-r--r--   1 root root    0 2005-11-12 16:10 Babika.txt
-rw-r--r--   1 root root    0 2005-11-12 16:10 b.txt
-rw-r--r--   1 root root    0 2005-11-12 16:10 Krava.txt
-rw-r--r--   1 root root    0 2005-11-12 16:11 MaMa.txt
-rw-r--r--   1 root root    0 2005-11-12 16:11 mama.txt
-rw-r--r--   1 root root    0 2005-11-12 15:06 novy_soubor2.txt
```

Pro výpis seznamu takových souborů, které například končí zvolenými příponami, např. txt, dat, použijeme zápis pomocí složených závorek.

ls -ald *.{txt,dat}

```
linux:/home/l# ls -ald *.{txt,dat}
-rw-r--r--   1 1    users    0 2005-10-18 20:27 a.txt
-rw-r--r--   1 root root    0 2005-11-12 16:10 Babika.txt
-rw-r--r--   1 root root    0 2005-11-12 16:10 b.dat
-rw-r--r--   1 root root    0 2005-11-12 16:10 Krava.txt
-rw-r--r--   1 root root    0 2005-11-12 16:11 MaMa.dat
-rw-r--r--   1 root root    0 2005-11-12 16:11 mama.txt
-rw-r--r--   1 root root    0 2005-11-12 15:06 novy_soubor2.txt
-rw-r--r--   1 root root   49 2005-11-12 15:24 soubor.txt
-rw-r--r--   1 root root   18 2005-11-12 15:50 soubor2.txt
-rw-r--r--   1 root root   67 2005-11-12 15:54 vysledek.txt
```

Poznámka

Výpis určitého druhu souborů pomocí symbolů [] {} apod. je vyhledávání pomocí tzv. regulárních výrazů.

3.6 Příkaz cat

Příkaz **cat** vypíše obsah souboru na standardní výstup (obrazovku). Výpis lze přesměrovat například do souboru.

Na následujícím obrázku je uveden výpis obsahu souboru soubor.txt, který obsahuje 7 řádků textu.

```
linux:/home/l# cat soubor.txt
radek1
radek2
radek3
radek4
radek5
radek6
radek7
```

Lze vypsat současně obsah více souborů. V příkazu cat je oddělíme mezerou.

```
linux:/home/l# cat soubor.txt soubor2.txt
radek1
radek2
radek3
radek4
radek5
radek6
radek7
Text v souboru 2
```

Poznámka

Pokud chceme výstup na obrazovku přesměrovat např. do souboru, napíšeme za příkaz cat **>** a název nového souboru, do kterého se má výstup přesměrovat.

```
linux:/home/l# cat soubor.txt soubor2.txt > vysledek.txt
linux:/home/l# cat vysledek.txt
radek1
radek2
radek3
radek4
radek5
radek6
radek7
Text v souboru 2
```


3.7 Příkaz tac

Tento příkaz slouží pro výpis obsahu souboru v opačném pořadí řádků.

```
linux:/home/l# cat soubor.txt
radek1
radek2
radek3
radek4
radek5
radek6
radek7
linux:/home/l# tac soubor.txt
radek7
radek6
radek5
radek4
radek3
radek2
radek1
```

3.8 Příkaz touch

Příkaz touch slouží k vytvoření souboru nebo k přenastavení doby posledního přístupu k souboru.

Příkazem **touch soubor** vytvoříme soubor s nulovou velikostí.

Čas poslední modifikace lze u souboru změnit na aktuální čas příkazem

touch -m soubor

Čas poslední modifikace všech souborů lze změnit na aktuální čas příkazem

touch -m *

U zvoleného souboru (nebo souborů) lze změnit čas poslední modifikace na zvolené datum a čas. Nové datum a čas se zadává ve formátu YYYYMMDDHHmm (rok, měsíc, den, hodina, minuta).

Na následujícím obrázku je vidět použití příkazu touch s modifikací času u souboru zahr.jpg.

```
linux:/home/l# ls -al zahr.jpg
-rwx----- 1 l users 22028 2004-11-24 17:25 zahr.jpg
linux:/home/l# touch -t 200005301135 zahr.jpg
linux:/home/l# ls -al zahr.jpg
-rwx----- 1 l users 22028 2000-05-30 11:35 zahr.jpg
```

Do souboru lze vložit textový řetězec např. příkazem **echo**.
echo „textový řetězec“ > soubor1.txt

Přidat textový řetězec na konec souboru lze udělat podobně.
echo „textový řetězec“ >> soubor1.txt

3.9 Příkaz head

Příkazem head lze vypsát začátek souboru, přednastaveno je 10 řádků. Můžeme zvolit, kolik řádků chceme vypsát.

head vysledek.txt ... vypíše prvních 10 řádků souboru vysledek.txt

head -n3 vysledek.txt ... vypíše první 3 řádky

```
linux:/home/1# head vysledek.txt
radek1
radek2
radek3
radek4
radek5
radek6
radek7
Text v souboru 2
radek1
radek2
linux:/home/1# head -n3 vysledek.txt
radek1
radek2
radek3
```

3.10 Příkaz tail

Příkazem tail lze vypsát konec souboru, přednastaveno je 10 řádků. Můžeme využít například u logovacích souborů, do nichž se nové změny ukládají na konec.

tail vysledek.txt ... vypíše posledních 10 řádků souboru vysledek.txt

tail -n3 vysledek.txt ... vypíše poslední 3 řádky

```
linux:/home/1# tail vysledek.txt
radek5
radek6
radek7
radek1
radek2
radek3
radek4
radek5
radek6
radek7
linux:/home/1# tail -n3 vysledek.txt
radek5
radek6
radek7
```

Zobrazování posledních 10 řádků v reálném čase
`tail -f /var/log/messages`
(výpis přerušíme např. CTRL+C)

3.11 Příkaz more

Slouží pro stránkované zobrazení obsahu souboru.

```
more vysledek.txt
```

Posun na další stránku – mezerník, posun na další řádek – klávesa enter.

3.12 Příkaz less

Slouží pro stránkované zobrazení obsahu souboru, podobně jako u příkazu more, navíc se lze posouvat ve směru vzad i vpřed (`PageUp` nebo `u`, `PageDown` nebo `d`), vyhledávat (za / napíšeme hledaný řetězec a potvrdíme klávesou `enter`, posun na další vyhledané slovo – stiskneme klávesu `n` – pro hledání ve směru dolů, `N` – pro hledání ve směru nahoru).

```
less vysledek.txt
```

Posun na další stránku – mezerník, posun na další řádek – klávesa enter, posun na 1. řádek – klávesa `g`, posun na poslední řádek – klávesa `G`.

```
less -N vysledek.txt ... u řádků se zobrazují jejich čísla
```

Další možnosti jsou popsány v manuálu příkazu (man less).

Poznámka

Do prohlížečů more a less je možno přeměrovat standardní výstup pomocí tzv. pipe ... |.

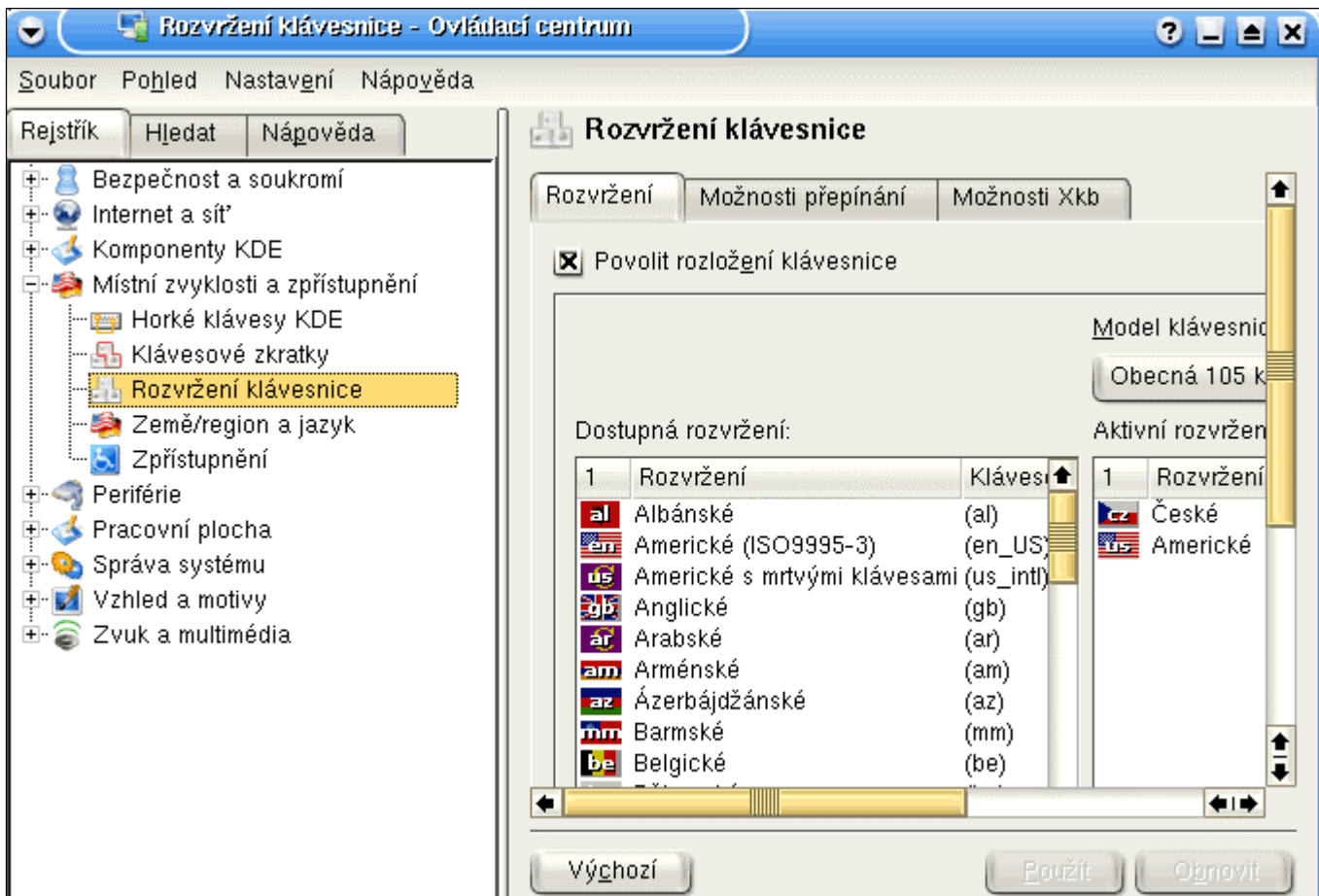
```
cat vysledek.txt | less  
cat vysledek.txt | more
```

☞ Přepínání klávesnice

V grafickém režimu myší kliknutím na ikonu jazyka.



Pokud se kliknutím rozvržení klávesnice nezmění, je potřeba zřejmě v ovládacím centru zvolit, která rozvržení klávesnice se budou používat.



Další možnost je klávesová zkratka, která je přednastavena na CTRL+ALT+K. Změnit se dá v Ovládacím centru (Místní zvyklosti – Klávesové zkratky).

V textovém režimu přidržetím pravého ALT se dočasně změní rozvržení klávesnice, trvale pomocí Pause/Break.

☞ Přepínání mezi konzolemi, grafický a textový režim

Z textového režimu se do jiných konzolí přepneme stisknutím kombinace kláves ALT+F1 (až F7).

Z grafického režimu se do textových konzolí přepneme stisknutím kombinace kláves CTRL+ALT+F1 (až F6).

☞ Výpis příkazů začínajících zvolenými znaky

Napišeme začátek příkazu a stiskneme dvakrát tabulátor.

3.13 Cvičení

- Ve svém domovském adresáři vytvořte soubory: **a1.txt**, **a2.txt**, **b1.txt**, **B1.txt**, **c1.log**, **D1.log**.
- Vypište si obsah svého domovského adresáře ve zkrácené formě.
- Vypište si obsah svého domovského adresáře v podrobné formě, včetně skrytých souborů.
- Vypište pouze soubory, které začínají na písmeno **a**.
- Vypište soubory, které začínají na písmeno **a, b, D**.
- Vypište soubory, které začínají písmeny od **b – m** a od **B do M**.
- Přesuňte se do rootu /.
- Vypište si podrobný seznam všech souborů.
- Přesuňte se do svého domovského adresáře.
- Vypište si cestu k aktuálnímu adresáři.
- Do souboru **a1.txt** napište text „Jméno uživatele“ a do souboru **a2.txt** napište text „Jan Zelený“.
- Vypište obsah souboru **a1.txt** a **a2.txt** na obrazovku.
- Obsah obou souborů přesměrujte do souboru **c1.log**.
- Vypište obsah souboru **c1.log** na obrazovku.
- Vypište seznam všech souborů v podrobné podobě tak, aby byly velikosti souborů uvedeny s jednotkami K, M atp.
- Vypište seznam všech souborů v podrobné podobě tak, aby byly seřazeny podle velikosti od nejmenšího k největšímu.
- Vypište seznam všech souborů v podrobné podobě tak, aby byly seřazeny podle data poslední změny, nejnovější soubor nahoře.
- Vytvořte prázdný soubor **a1.dat**, **a2.dat**.
- Změňte datum poslední modifikace na 15.3.2002 14:25 u souborů **a1.dat** a **a2.dat**. Zkontrolujte výpisem.
- Vypište v podrobném formátu všechny soubory s příponou **dat** a **log**.
- Obsah souboru **a1.txt** pomocí **cat** zkopírujte do souboru **a1.dat** a **a2.txt** do **a2.dat**.
- Obsah souboru **c1.log** vypište v obráceném pořadí řádků.
- Obsah souboru **a1.dat** a **a2.dat** přidejte na konec souboru **c1.log** a výsledný obsah souboru vypište na obrazovku.
- Příkazem **head** vypište první dva řádky souboru **c1.log**.
- Vypište poslední dva řádky souboru **c1.log**.

 *Domácí úkol*

Zopakujte si práci s příkazy **mount, umount, man, info, cd, ls, cat, tac, touch, head, tail, more, less**.

Pomocí manuálových stránek prostudujte možnosti těchto příkazů.

 Shrnutí

- ✓ umíte připojit souborový systém
- ✓ umíte odpojit souborový systém
- ✓ umíte využívat nápovědu pomocí man a info
- ✓ umíte se pohybovat v adresářové struktuře
- ✓ umíte vypsát obsah adresáře a znáte základní přepínače příkazu ls
- ✓ umíte vypsát obsah souboru na obrazovku a tento výstup umíte přesměrovat do souboru
- ✓ umíte vypsát řádky souboru v opačném pořadí
- ✓ umíte vytvořit soubor a změnit datum a čas poslední modifikace
- ✓ umíte vypsát zadaný počet řádků od začátku i od konce souboru
- ✓ umíte obsah souboru zobrazovat v prohlížeči more a less
- ✓ v prohlížeči less umíte vyhledávat a používat klávesové zkratky pro pohyb v něm

4. Příkazy su, whoami, pwd, clear, echo, grep, who, standardní vstupy a výstupy, chybové výstupy

(7.- 8. hodina)

4.1 Příkaz su, sux, su -

Příkaz su se používá pro změnu uživatele.

Např. příkazem su student se přihlásíme na účet uživatele student.

Pokud vynecháme jméno účtu, ke kterému se chceme přihlásit, přihlásíme se jako superuživatel - root.

Pokud za su napíšeme pomlčku, pak se přihlásíme k účtu uživatele a všechny jeho proměnné i prostředí budou nastaveny podle jeho nastavení.

Po zadání jména bude následovat dotaz na heslo uživatele.

Př. su - ... přihlášení na účet roota, bude následovat výzva pro zadání hesla.

4.2 Příkaz whoami

Tento příkaz vypíše aktuálně přihlášeného uživatele.

4.3 Příkaz pwd

Tento příkaz vypíše jméno aktuálního adresáře (celou cestu k tomuto adresáři).

4.4 Příkaz clear

Tímto příkazem vymažeme obsah obrazovky.

4.5 Příkaz du

Použitím tohoto příkazu lze zobrazit odhad využitého místa na disku (nejedná se o velikost souboru). Lze zobrazit zabrané místo na disku pro jednotlivé soubory.

du ... vypíše zabrané místo pro jednotlivé adresáře v aktuálním adresáři.

du -a ... vypíše zabrané místo nejen pro adresáře, ale i soubory.

Příklad příkazu du -a. Zobrazí zabrané místo jednotlivými soubory (v kB).

```

iv:~# du -a
4      ./profile
4      ./bashrc
4      ./dbootstrap_settings
4      ./install-report.template
0      ./aptitude/config
4      ./aptitude
4      ./mc/ini
4      ./mc/history
4      ./mc/cedit
4      ./mc/Tree
4      ./mc/filepos
24     ./mc
4      ./ssh/known_hosts
8      ./ssh
4      ./gstreamer-0.8/registry.xml
8      ./gstreamer-0.8
4      ./gnome
4      ./gconf
16     ./gconfd/saved_state
20     ./gconfd
4      ./bash_history
24     ./vypisbaliku.txt
4      ./soubor1.txt
124    .
    
```

Příklad příkazu ls. Zobrazí velikosti souborů.

```

iv:~# ls -al
celkem 84
drwxr-xr-x  9 root root  4096 2005-11-16 16:12 .
drwxr-xr-x 21 root root  4096 2005-10-07 18:29 ..
drwx-----  2 root root  4096 2005-11-15 13:43 .aptitude
-rw-----  1 root root  2602 2005-11-15 14:29 .bash_history
-rw-r--r--  1 root root   412 2004-12-15 23:53 .bashrc
-rw-r--r--  1 root root   181 2005-10-07 13:15 dbootstrap_sett:
drwx-----  2 root root  4096 2005-10-07 18:01 .gconf
drwx-----  2 root root  4096 2005-10-07 18:02 .gconfd
drwx-----  2 root root  4096 2005-10-07 18:00 .gnome
drwxr-xr-x  2 root root  4096 2005-10-07 18:02 .gstreamer-0.8
-rw-r--r--  1 root root  1336 2005-10-07 13:15 install-report.t
drwxr-xr-x  3 root root  4096 2005-11-15 10:34 .mc
-rw-r--r--  1 root root   110 2004-11-10 17:10 .profile
-rw-r--r--  1 root root    14 2005-11-16 16:12 soubor1.txt
drwx-----  2 root root  4096 2005-10-11 16:36 .ssh
-rw-r--r--  1 root root 21619 2005-10-11 14:04 vypisbaliku.txt
    
```

du -c -h /home

Tímto příkazem spočítáme velikost zabraného prostoru pro jednotlivé adresáře (i podadresáře) v adresáři home. Přepínač **h** znamená "human readable" - lidsky čitelná podoba, u velikostí zabraného prostoru jsou zobrazeny jednotky K, M atd.

du -s -h /home

Tímto příkazem spočítáme celkovou velikost zabraného prostoru v adresáři /home.

du -s -h /home/*

Tímto příkazem spočítáme velikost zabraného prostoru všech hlavních podadresářů v adresáři /home.

```
linux:/home/l# du -c -h
8. OK      ./a
8. OK      ./adresar
68K        .
68K        total
linux:/home/l# du -s -h
68K        .
linux:/home/l# du -s
68         .
linux:/home/l# du -s -h ./*
0          ./2004-11-24
0          ./Babika.txt
0          ./Krava.txt
0          ./MaMa.dat
8. OK      ./a
0          ./a.txt
8. OK      ./adresar
0          ./b.dat
0          ./mama.txt
0          ./novy_soubor2.txt
4. OK      ./soubor.txt
4. OK      ./soubor2.txt
4. OK      ./vysledek.txt
24K        ./zahr.jpg
linux:/home/l# du -s -h ./* | sort -n
0          ./2004-11-24
0          ./Babika.txt
0          ./Krava.txt
0          ./MaMa.dat
0          ./a.txt
0          ./b.dat
0          ./mama.txt
0          ./novy_soubor2.txt
4. OK      ./soubor.txt
4. OK      ./soubor2.txt
4. OK      ./vysledek.txt
8. OK      ./a
8. OK      ./adresar
24K        ./zahr.jpg
linux:/home/l#
```

du -s /home/* | sort -n

Tímto příkazem spočítáme velikost zabraného prostoru všech podadresářů v adresáři /home a seřadíme je podle velikosti vzestupně.

du -s /home/* | sort -nr

Tímto příkazem spočítáme velikost zabraného prostoru všech podadresářů v adresáři /home a seřadíme je podle velikosti sestupně.

☞ Poznámka

Tento příkaz se může hodit, pokud chceme rychle zjistit, který uživatel zabírá nejvíc místa v adresáři home.

Pokud nás zajímá například jen prvních pět uživatelů, kteří zabírají nejvíc prostoru v adresáři home, pak předchozí výpis přesměrujeme do příkazu tail.

du -s /home/* | sort -nr | tail -n5

```
linux:/home/l# du -s -h ./* | sort -n | tail -n5
4.0K      ./soubor2.txt
4.0K      ./vysledek.txt
8.0K      ./a
8.0K      ./adresar
24K       ./zahr.jpg
```

4.6 Příkaz df

Tímto příkazem lze zobrazit informace o využití prostoru na připojených souborových systémech.

df -h

Tento příkaz vypíše informace o využití prostoru všech připojených souborových systémů.

df -h /dev/hda2

Tento příkaz vypíše informace o využití prostoru na zařízení hda2.

```
linux:/home/l# df
Filesystem          1K-blocks      Used Available Use% Mounted on
/dev/hda2            2885812    1686568    1052652   62% /
tmpfs                91892         0         91892    0% /dev/shm
//192.168.0.1/linux  80033792    36874240    43159552   47% /media/sit
linux:/home/l# df -h
Filesystem          Size  Used Avail Use% Mounted on
/dev/hda2           2.8G  1.7G  1.1G   62% /
tmpfs               90M   0     90M   0% /dev/shm
//192.168.0.1/linux  77G   36G   42G   47% /media/sit
```

df -l

Tento příkaz vypíše informace o využití prostoru na všech lokálních souborových systémech.

df -i

Tento příkaz vypíše informace o využití inodů.

Inode je datová struktura v souborovém systému, která v sobě udržuje základní informace o souboru, adresáři a dalších systémových objektech.

Podle standardu POSIX jsou to alespoň následující informace.

- ❑ délka souboru v bytech
- ❑ identifikace zařízení obsahující daný soubor
- ❑ user ID vlastníka souboru
- ❑ group ID souboru
- ❑ číslo inodu unikátní na daném souborovém systému
- ❑ souborový mód určující, zda uživatel bude moci soubor číst, spouštět nebo měnit

- časové známky (atime – kdy bylo k souboru přistupováno, ctime – kdy byl inode sám změněn, mtime – kdy byl obsah naposled modifikován)
- číslo určující, kolik hardlinků na daný inode odkazuje

```
linux:/home/l# df -i
Filesystem          Inodes    IUsed    IFree    IUse%    Mount
/dev/hda2           366528    99976    266552    28%      /
tmpfs                22973     1         22972    1%       /dev/
//192.168.0.1/linux 0          0         0         -        /medi
linux:/home/l# df -l
Filesystem          1K-blocks    Used    Available    Use%
/dev/hda2         2885812     1688892    1050328    62%
tmpfs              91892        0         91892      0%
linux:/home/l# df -lh
Filesystem          Size    Used    Avail    Use%    Mounted on
/dev/hda2           2.8G    1.7G    1.1G     62%    /
tmpfs               90M     0       90M     0%     /dev/shm
linux:/home/l# df -h /dev/hda2
Filesystem          Size    Used    Avail    Use%    Mounted on
/dev/hda2           2.8G    1.7G    1.1G     62%    /
linux:/home/l# █
```

4.7 Příkaz echo

Slouží k výpisu na standardní výstup – obrazovku. Výstup lze přesměřovat do souboru.

Pro výpis obsahu proměnné se za příkaz echo napíše symbol dolaru a název proměnné.

echo \$USER

Příkazy i proměnné jsou case sensitive, na velikosti písmen záleží.

Pro výpis seznamu proměnných prostředí můžete využít příkaz **set**.

echo "ahoj"

echo ahoj

Oba příkazy vypíší na obrazovku slovo ahoj.

Pokud chceme nějaký text přesměřovat do souboru, použijeme symbol ">" a název souboru.

echo "text posílaný do souboru" > soubor.txt

```

linux:/home/l# echo $USER
root
linux:/home/l# set
BASH=/bin/bash
BASH_VERSINFO=( [0]="2" [1]="05b" [2]
gnu")
BASH_VERSION='2.05b.0(1)-release'
COLUMNS=82
DIRSTACK=()
EUID=0
GROUPS=()
HISTFILE=/root/.bash_history
HISTFILESIZE=500
HISTSIZ=500
HOME=/root
HOSTNAME=linux
HOSTTYPE=i386
HZ=100
IFS=$' \t\n'
LINES=48
LOGNAME=root
MACHTYPE=i386-pc-linux-gnu
MAIL=/var/mail/root
MAILCHECK=60
OLDPWD=/home/l/adresar
OPTERR=1
OPTIND=1
OSTYPE=linux-gnu
PATH=/usr/local/sbin:/usr/local/bin
PIPESTATUS=( [0]="0")
PPID=6926
PS1='\h:\w\$ '
PS2='>'
PS4='+ '
PWD=/home/l
SHELL=/bin/bash
SHELLOPTS=braceexpand:emacs:hashall
r
SHLVL=1
TERM=xterm
UID=0
USER=root
_=root
    
```

4.8 Příkaz grep

Tento příkaz slouží pro vyhledávání zadaného řetězce v zadaném umístění. Vypíše řádky, které hledaný řetězec obsahují.

grep "hledaný řetězec" prohledávaný_adresář

grep "neco a" *

Tento příkaz prohledá všechny soubory v aktuálním adresáři, bude se hledat řetězec "neco a".

Výsledkem je výpis řádků, ve kterých se našla shoda s hledaným řetězcem, u každého řádku bude napsáno jméno souboru, ve kterém se řetězec našel.

```
linux:/home/l# grep -r "neco a" *
a/soubor2.txt:a proto neco a dalsi
a.txt:a proto bylo neco a jeste
a.txt:neco atd
mama.txt:neco avice
```

grep -r "neco a" *

Tento příkaz funguje podobně jako předchozí. Bude ale hledat rekurzivně i v podadresářích aktuálního adresáře.

Pro spočítání řádků v zadaném souboru, které obsahují hledaný řetězec zadáme příkaz **grep -c "neco" a.txt**

Příkaz grep -c "neco" *

vypíše počty řádků, ve kterých se nachází hledaný řetězec, u všech souborů v aktuálním adresáři.

```
linux:/home/l# grep -c "neco" a.txt
2
linux:/home/l# grep -c "neco" *
2004-11-24:0
Babika.txt:0
Krava.txt:0
MaMa.dat:0
a:0
a.txt:2
adresar:0
b.dat:0
mama.txt:1
novy_soubor2.txt:0
soubor:0
soubor.txt:0
soubor2.txt:0
vysledek.txt:0
zahr.jpg:0
```

Příkaz grep může získat své parametry i pomocí **přesměrování** ze standardního výstupu.

Například při výpisu dlouhého logovacího souboru nás zajímají jen řádky obsahující určitý řetězec.

cat /var/log/messages | grep e100

```
linux:~# cat /var/log/messages | grep e100
Nov 14 18:12:07 linux kernel: e100: eth0: e100_watch
link up, 100Mbps, full-duplex
Nov 19 08:57:24 linux kernel: e100: Intel(R) PRO/100
ork Driver, 3.0.18
Nov 19 08:57:24 linux kernel: e100: Copyright(c) 199
4 Intel Corporation
Nov 19 08:57:24 linux kernel: e100: eth0: e100_probe
r Oxf4020000, irq 11, MAC addr 00:DO:59:7F:21:7B
Nov 19 08:57:24 linux kernel: e100: eth0: e100_watch
link up, 100Mbps, full-duplex
Nov 19 13:50:10 linux kernel: e100: eth0: e100_watch
link down
Nov 19 13:50:12 linux kernel: e100: eth0: e100_watch
link up, 100Mbps, full-duplex
```

Přepínač **"i"** zajistí, že se bude ignorovat rozdílnost malých a velkých písmen. Pokud například hledáme řetězec **"necO"** a v souborech se vyskytuje jen s malými písmeny, pak by příkaz grep nenašel žádnou shodu. S přepínačem **"i"** bude rozdílnost malých a velkých písmen ignorovat.

grep -ir "necO" *

Přepínač **"r"** je zde pro rekurzivní vyhledávání i v podadresářích.

Pro vyhledávání lze použít i **regulární výrazy** (setkali jsme se s nimi již u příkazu ls).

Jestliže chceme vyhledat řádky, které obsahují hledaný výraz na **začátku** řádku, použijeme pro hledání tzv. metaznak (sám se nebude hledat) **^**.

Pro hledání výrazu na **konci** řádku použijeme metaznak **\$**.

Je lépe složitější hledaný výraz uzavřít do **apostrofů**.

grep '^neco' *

vyhledá výraz **"neco"** na začátku řádků ve všech souborech v aktuálním adresáři.

grep 'neco\$' *

vyhledá výraz **"neco"** na konci řádků.

grep '^neco\$' *

vyhledá výraz **"neco"**, který je současně na začátku i na konci řádku. Tzn. hledá se řádek, který obsahuje pouze hledaný výraz.

Jestliže chceme vyhledat v souboru slova, která obsahují velké písmeno, dáme do vyhledávání na místo písmene seznam [A-Z].

[A-Z] znamená jakékoliv velké písmeno.

[a-z] znamená jakékoliv malé písmeno.

Tečka . znamená jeden libovolný znak.

Hvězdička * znamená libovolný počet libovolných znaků (určených v zápisu předchozím znakem).

Seznam určitých znaků se uzavírá mezi **hranaté** závorky, např. **[abc]** znamená vyhledat na dané pozici a nebo b nebo c.

[A-Za-z] je jakékoliv malé nebo velké písmeno.

[0-9] znamená jakoukoliv číslici od 0 do 9.

Pokročilejší příkaz vycházející z příkazu grep je **egrep**. Když chceme používat složitější regulární výrazy, budeme někdy muset použít příkaz egrep.

Pokud chceme vyhledat výskyt například deseti velkých písmen, napíšeme za symbol seznamu do složených závorek číslo znamenající počet výskytů - **[A-Z]{10}**.

Pokud chceme vyhledat slova, která obsahují například pět až deset velkých písmen, napíšeme za symbol seznamu do složených závorek dolní a horní mez oddělené čárkou - **[A-Z]{5,10}**.

Pokud nám na počtu opakování vzoru nezáleží, napíšeme za seznam v hranatých závorkách hvězdičku - **[A-Z]***.

Pokud při vyhledávání chceme **vyločit nějaký znak**, pak se **do hranatých závorek** před znak napíše **^**.

egrep '[^N]eco' a.txt

Ze souboru a.txt nevypíše řádky, ve kterých se nachází „Neco“. Pokud ale v řádku existuje také řetězec „neco“ nebo „deco“ apod., pak se řádek vypíše.

Pokud chceme vypsát řádky, ve kterých se nevyskytuje velké písmeno „U“, tak to zapíšeme následovně:

egrep '^[^U]*\$' a.txt

Říkáme tím, že nechceme řádky, ve kterých se vyskytuje velké písmeno U na libovolné pozici od prvního do posledního znaku v libovolném počtu opakování.

První stříška znamená začátek řádku, druhá stříška uvnitř seznamu v hranatých závorkách znamená vyloučení následujících hodnot.

egrep '^[^UT]*\$' a.txt

Tímto příkazem navíc od předchozího příkladu říkáme, že nechceme zobrazit řádky obsahující velké písmeno U nebo velké písmeno T.

Pokud chceme **vyhledat** nějaký **metaznak**, například tečku, stříšku, dolar, pak před něj ve vyhledávací formuli napíšeme **zpětné lomítko**.

Chceme-li například vyhledat číslo **1.2**, pak musíme před tečku napsat ****.

egrep '1\.2' a.txt

Pokud chceme vyhledat jakékoliv desetinné číslo, vyhledáme jej podle vzorce

```
egrep '[0-9]\.[0-9]' a.txt
```

Příkaz **egrep '*' a.txt** vyhledá řádky obsahující hvězdičku.

4.9 Příkaz who

Zobrazí, kdo je aktuálně přihlášen.

```
who
```

```
who -a
```

zobrazí informace o přihlášených uživateli na všech konzolách.

```
who -b
```

zobrazí čas posledního startování systému.

Informace o přihlášených uživateli můžete zobrazit také příkazy **finger, users, w**.

4.10 Standardní vstupy a výstupy

Za standardní vstup se považuje zadávání dat z klávesnice.

Za standardní výstup se považuje zobrazení dat na obrazovce.

Přesměrování standardního výstupu do souboru uděláme napsáním znaku **>** a jméno souboru.

Například rekurzivní výpis obsahu adresáře uložíme do souboru vypis.txt.

```
ls -aR > vypis.txt
```

Standardní výstup můžete **předat jako parametry** nějakému dalšímu příkazu.

Za program1 napíšeme znak **|** (pipe, roura) a pak následuje program2, který výstup z programu1 zpracuje.

Například příkazem **cat** vypíšeme obsah souboru a tento výpis předáme příkazu **grep** jako parametr a necháme vypsát jen ty řádky, které obsahují zvolený řetězec.

```
cat a.txt | grep -i 'neco'
```

Dalším příkladem může být výpis systémových hlášení, která probíhají při startu počítače.

Zobrazení těchto hlášení uděláte příkazem **dmesg**.

Příkaz má dlouhý výstup, proto jej předáme příkazu **less** (nebo **more**) nebo zobrazíme pomocí příkazu **grep** jen ty řádky, které obsahují jen slovo, které nás zajímá.

dmesg | less
dmesg | more
dmesg | grep CPU
dmesg | grep -i acpi

Každý běžící program má i svůj **standardní chybový výstup**, který se zobrazuje na obrazovce.

Přesměrování standardního chybového výstupu do souboru provedeme napsáním "**2>**" za příkaz, jehož standardní chybový výstup chceme přesměrovat.

program 2> soubor.txt

Například **cat -nesmysl soubor.txt** vyvolá chybový výstup, který příkazem **cat -nesmysl soubor.txt 2> vystup.txt** přesměrujeme do souboru vystup.txt.

```
linux:/home/l# cat -nesmysl soubor.txt
cat: invalid option -- m
Try `cat --help' for more information.
linux:/home/l# cat -nesmysl soubor.txt 2> vystup.txt
linux:/home/l# cat vystup.txt
cat: invalid option -- m
Try `cat --help' for more information.
```

Spojení chybového výstupu se standardním výstupem se udělá pomocí "**2>&1**".

Například příkazem **ls** chceme vypsát obsah adresářů "a" a "b", ale adresář "b" neexistuje, proto se na obrazovku vypíše chybové hlášení kvůli neexistujícímu adresáři "b" a dále se vypíše obsah adresáře "a".

```
linux:/home# ls l nesmysl
ls: nesmysl: No such file or directory
l:
2004-11-24  MaMa.dat   adresar   novy_soubor2.
Babika.txt  a             b.dat     soubor
Krava.txt   a.txt        mama.txt  soubor.txt
```

Pokud výpis "**ls a b**" přesměrujeme do souboru **vypis.txt** příkazem **ls a b > vypis.txt**, pak se do souboru vypis.txt uloží jen výpis existujícího adresáře "a" a chybové hlášení o neexistenci adresáře "b" se zobrazí jen na obrazovce.

Pokud jej chceme uložit spolu se standardním výstupem do souboru vypis.txt, pak to uděláme příkazem

ls a b > vystup.txt 2>&1

Tím říkáme, že chybový výstup (který je označen číslem 2) se připojí ke standardnímu výstupu (který je označen číslem 1) a pak se uloží do souboru vypis.txt.

Poznámka

Standardní vstup je označen číslem 0, standardní výstup číslem 1 a chybový výstup číslem 2.

Připojení na konec souboru

Pro připojení použijeme dvě znaménka **>>**.

Například **cat a.txt >> vystup.txt** připojí obsah souboru a.txt na konec souboru vystup.txt.

4.11 Cvičení

- ❑ Spustíte terminálový program "Konzole".
- ❑ Zjistěte, jako jaký uživatel jste přihlášen ke konzoli.
- ❑ Přepněte se na textovou konzoli (CTRL+ALT+F1) a přihlašte se jako libovolný uživatel.
- ❑ Přepněte se zpět na grafickou konzoli a zjistěte, kdo a na jaké konzoli je přihlášen.
- ❑ Zjistěte, v jakém adresáři se nacházíte.
- ❑ Zjistěte množství zabraného prostoru ve vašem domovském adresáři.
- ❑ Zjistěte, jaké místo zabírají všechny hlavní adresáře ve vašem domovském adresáři.
- ❑ Seřadte hlavní podadresáře ve vašem domovském adresáři podle velikosti od nejmenšího k největšímu.
- ❑ Vypište první tři největší adresáře ve vašem domovském adresáři.
- ❑ Zobrazte informace o využití prostoru na aktuálním souborovém systému.
- ❑ Zobrazte informace o využití prostoru na všech lokálních souborových systémech.
- ❑ Zobrazte informace o využití inodů na aktuálním souborovém systému.
- ❑ Vytvořte soubor soubor.txt a vložte do něj příkazem echo text „aktuální uživatel“ a jméno tohoto uživatele.
- ❑ Do souboru soubor.txt vložte další řádky
řádek 1 nějaký text
řádek 2 další TEXT
ŘÁDEK 3 Txt abc
řádek 4 CDA
- ❑ Příkazem grep vyhledejte v souboru soubor.txt řádky, které obsahují text "text".

- Vyhledejte řádky, které na prvním místě písmeno "ř" a poslední je buď "t" nebo "T".
`grep '^ř.*[tT]$' soubor.txt`
- Vyhledejte řádky, které obsahují číslo 1 nebo 3.
- Vyhledejte řádky, které neobsahují velká písmena.
`grep '^[^A-Z]*$' soubor.txt`
- Vyhledejte řádky, které neobsahují písmeno "x" ani "X".
`grep '^[xX]*$' soubor.txt`
- Vyhledejte řádky, které obsahují slovo "řádek", na velikosti písmen nezáleží.
- Vyhledejte řádky, které mají libovolné první písmeno, pak následuje "á" a řádek končí písmenem A nebo T.
`grep '^a.*[AT]$' soubor.txt`
- Zobrazte čas posledního startování systému.
- Vytvořte soubor vystup.txt.
- Vypište obsah adresářů /media a /neznamy a celý výpis (standardní výstup i chybový výstup) uložte do souboru vypis.txt.
- Vypište řádky systémových hlášení ze startu systému, které obsahují informace o "ip" bez ohledu na velká – malá písmena.

 *Domácí úkol*

Zopakujte si práci s příkazy

- su
- whoami, who, finger, w, users
- clear, echo
- grep

Dále si zopakujte, jak pracovat se vstupy a výstupy, jak se dají přesměřovat apod.

 Shrnutí

- ✓ umíte změnit identitu uživatele
- ✓ víte, jak zjistit, jako jaký uživatel jste aktuálně na konzoli přihlášen
- ✓ víte, jak zjistit absolutní cestu aktuálního adresáře
- ✓ víte, jak vyčistit obrazovku
- ✓ víte, k čemu se používají příkazy w, who, whoami, users, finger
- ✓ víte, jak použít příkaz echo
- ✓ umíte příkazem grep filtrovat zobrazení řádků souboru
- ✓ znáte základní práci s regulárními výrazy
- ✓ víte, jak se označují standardní vstupy, výstupy a chybové výstupy
- ✓ víte, jak přesměřovat standardní výstup i chybový výstup do souboru
- ✓ víte, jak spojit standardní výstup se standardním chybovým výstupem do jednoho souboru
- ✓ víte, jak předat výsledek jednoho příkazu jako parametr jinému příkazu

5. Základní příkazy – opakování (9. – 10. hodina)

5.1 Cvičení

- Přepněte se na textovou konzolu a přihlašte se pod běžným uživatelským účtem (např. **student**).
- Zjistěte absolutní **cestu** k aktuálnímu adresáři.
- Příkazem **mkdir pokus** vytvořte ve svém domovském adresáři podadresář jménem pokus.
- Přepněte se do tohoto adresáře.
- Zde vytvořte deset souborů (**soubor1 až soubor10**).
- Zobrazte stručně **obsah** adresáře **/var**.
- Tento výpis přesměrujte do souboru **soubor1**.
- Zobrazte **podrobně obsah** adresáře **/var** (i skryté soubory), včetně vnořených podsložek.
- Tento výpis **přesměrujte** do souboru **soubor2**.
- **Chybová** hlášení, která se objevují při snaze vypsát obsah vnořených podadresářů adresáře **/var**, ke kterým nemá běžný uživatel přístup, přesměrujte do souboru **soubor3**.
- Podrobný výpis adresáře **/var** (včetně vnořených podadresářů, včetně skrytých souborů) spolu s chybovými hlášeními **spojte** a výpis přesměrujte do souboru **soubor4**.
- Vypište obsahy souborů **soubor1 – soubor4**.
- Vypište jen ty řádky těchto souborů, které **obsahují** slovo „odmítnut“ (deny).
- Nechejte **spočítat** počet řádků, které obsahují slovo „odmítnut“.
- Nechejte **spočítat** počet řádků, které obsahují slovo „X11“.
- Výpis řádků, které obsahují slovo **X11**, **přesměrujte** do souboru **soubor5**.
- Ze souboru **soubor3** vypište jen ty řádky, které **obsahují** na **začátku** text „ls: /var/spool“.
- Ze souboru soubor4 vypište jen ty řádky, které **končí** na „.gz“.
- **Spočítejte** tyto řádky (příkazem).
- Vypište ze souboru **soubor4** takové řádky, ve kterých je řetězec „rwxrwxrwx“.
- **Spočítejte** tyto řádky.
- Do souboru **soubor6** vložte text „Plne opraveni pro vsechny“.
- Výpis řádků, ve kterých je řetězec „rwxrwxrwx“ (ze souboru soubor4) přesměrujte a **přidejte** na konec do souboru **soubor6**.
- **Spočítejte** počet těchto řádků a číslo přesměrujte na **konec** souboru **soubor6**.
- **Vypište** obsah souboru **soubor6** a zkontrolujte.
- Ze souboru **soubor4** vypište řádky, které obsahují buď „Arial“ nebo „arial“.

- Tento výpis přesměrujte do souboru **soubor7**.
- Na **konec** souboru přidejte **číslo** vyjadřující počet **řádků** (příkazem).
- Zobrazte obsah souboru **soubor7**.
- Vypište řádky ze souboru **soubor4**, které obsahují alespoň **5 číslic** po sobě.
- Výpis přesměrujte do souboru **soubor8**.
- Na konec souboru připojte číslo odpovídající **počtu řádků** (vhodným příkazem).
- Ze souboru **soubor4** vypište řádky, které **neobsahují** žádné číslice.
- Do souboru **soubor9** vložte na **začátek** text „**Radky bez cislic**“.
- Výpis řádků ze souboru **soubor4**, které **neobsahují** žádnou **číslici**, **přesměrujte** a **připojte** na konec do souboru **soubor9**.
- Určete **počet** těchto **řádků** a číslo vložte na **konec** souboru **soubor9**.
- Zobrazte si obsah celého souboru **soubor9** pro kontrolu.
- Slučte obsahy souborů **soubor1 – soubor9** do souboru **soubor10**.
- Zobrazte si podrobně výpis adresáře **pokus**.
- **Seřad'te** výpis souborů **vzestupně** podle **velikosti**.
- **Seřad'te** výpis souborů **sestupně** podle **velikosti**.
- **Seřad'te** výpis souborů od **nejnovějších** k nejstarším.
- **Seřad'te** výpis souborů od **nejstarších** k nejnovějším.
- **Seřad'te** výpis souborů **vzestupně** podle **velikosti** se zobrazením **jednotek**.
- **Seřad'te** výpis souborů **sestupně** podle velikosti se zobrazením jednotek.
- **Seřad'te** výpis souborů od **nejnovějších** k nejstarším se zobrazením **jednotek**.
- **Seřad'te** výpis souborů od **nejstarších** k nejnovějším se zobrazením **jednotek**.
- Vypište, kolik **místa zabírá** celý adresář **pokus**.
- Vypište, kolik místa zabírají **jednotlivé** adresáře v adrsáři **pokus**.
- **Seřad'te** podadresáře podle **místa**, které zabírají na disku – **vzestupně** a **sestupně**.
- Přepněte se do svého domovského adresáře a příkazem **rm -r pokus** **smažte** adresář **pokus** i s obsahem.

 *Domácí úkol*

Zopakujte si látku probranou v předchozích kapitolách a projděte si všechna praktická cvičení.

 *Shrnutí*

- ✓ V této kapitole jste si především zopakovali práci se soubory, výpisy obsahu adresáře, výpis velikosti zabraného místa na disku, přesměrování standardního výstupu a chybového výstupu, filtrování výpisu příkazem grep a další příkazy.

6. Mc, editor vi, hard link, soft link (11. – 12. hodina)

6.1 Midnight commander (mc)

Pro spuštění dvoupanelového prohlížeče souborů - Midnight Commander – lze použít příkaz **mc**.

Funguje zde používání funkčních kláves (F3, F4, ...), jejichž stručný popis je vidět ve spodní části MC.

Např. klávesou F4 editujete vybraný soubor, klávesou F5 kopírujete vybraný soubor do místa určeného ve druhém panelu atd.

Popis funkčních kláves je vidět ve spodní části obrázku.

Tip: S nalezenými soubory můžete pracovat. Stiskněte tlačítko Panelizace.

```
student@linux:~$
```

1 Nápověda 2 Menu Už 3 Prohlíž 4 Editace 5 Kopírov 6 Přesun 7 Nový adr 8 Smazat 9 Hl Menu 10 Konec

F1 – nápověda

F2 – uživatelské menu (lze zde zvolit například kompresi souboru, adresáře)

F3 – prohlížení souboru

F4 – editování souboru

F5 – kopírování souboru (případně připojení k existujícímu souboru)

F6 – přesun nebo přejmenování souboru (přesun – do jiného adresáře, přejmenování – při přesunu do stejného adresáře pod jiným jménem)

F7 – vytvoření nového adresáře

F8 – odstranění souboru

F9 – přesun do hlavní nabídky (lze zde například změnit režim zobrazení v levém a pravém panelu, upravit nastavení, vykonávat různé příkazy – například vytváření odkazů – linků atd.)

F10 – ukončení MC

6.1.1 Cvičení

- Spustíte MC.
- V obou panelech MC si najdete svůj domovský adresář.
- Vytvořte zde adresář „pokus“.
- Vhodným příkazem vytvořte soubor „soubor1“.
- Do souboru vložte pár řádků textu a uložte.
- Pomocí kopírování vytvořte v adresáři „pokus“ soubor „soubor2“, který je kopií souboru „soubor1“.
- Přejmenujte soubor „soubor2“ na „soubor3“.
Tip: V obou panelech si zobrazte obsah adresáře „pokus“ a použijte klávesu F6.
- Přesuňte soubor „soubor3“ o úroveň nahoru – do domovského adresáře.
- Pomocí kopírování F5 připojte obsah souboru „soubor3“ k obsahu souboru „soubor1“.
- Prohlédněte si obsah souboru „soubor1“.
- Odstraňte soubory „soubor1“ a „soubor3“.
- Odstraňte adresář „pokus“.

6.1.2 Typy souborů

- **normální soubor**
- je to soubor dat určených jedním jménem.
Patří sem textové soubory, spustitelné programy, grafické soubory atd.
- **adresář**
- každý adresář obsahuje základní dva záznamy: „.“ – tento záznam ukazuje přímo na daný adresář a „..“ – odkazuje o jednu úroveň nahoru – do nadřazené struktury v souborové hierarchii.
- **soubor zařízení**
- každé zařízení v systému Linux je reprezentováno souborem daného zařízení. Pokud chce nějaký program komunikovat s vybraným zařízením, dělá to přes odpovídající soubor zařízení.
- **odkaz (link)**
- je to soubor odkazující na jiný soubor. Změny provedené na originálu souboru se promítnou automaticky do odkazů na tento soubor.
- **soket (socket)**
- je speciální soubor, přes který si dva lokálně běžící procesy vyměňují data.
- **pipe (roura)**
- speciální soubor používaný pro výměnu dat mezi procesy, ale na rozdíl od soketu je výměna jednosměrná.

6.2 Editor vi

Editor vi je jednoduchý textový editor běžně přítomný v Linuxu.

Spustí se příkazem **vi**.

Je možno přepínat se mezi třemi základními módy.

Základní módy jsou:

- příkazový
- řádkový příkazový
- vkládací

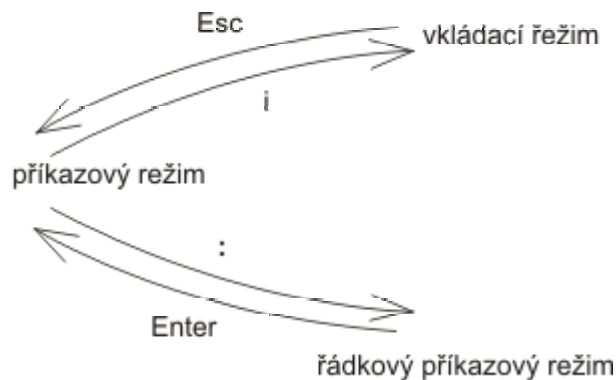
Po spuštění editoru se nacházíme v **příkazovém** módu. Proto vše co napíšeme je považováno za příkazy.

Do **vkládacího** režimu se přepneme klávesou **i** nebo klávesou **Insert**.

Zpět do **příkazového** režimu se vrátíme klávesou **Esc**.

Třetím módem je **řádkový příkazový režim**. Dostaneme se do něj z příkazového režimu stisknutím klávesy **:**. Po stisknutí dvojtečky se kurzor přesune za poslední řádek a čeká se na příkaz.

Po zadání příkazu se stiskne klávesa **Enter** a přesuneme se zpět do **příkazového** režimu.



Některé základní příkazy v příkazovém režimu

- i ... přepnutí do vkládacího režimu
- Delete nebo x ... smaže znak, na kterém zrovna stojí
- dd ... smaže řádek, na kterém stojí a vloží jej do schránky
- D ... od aktuální pozice kurzoru smaže všechny znaky až do konce řádku
- yy ... kopíruje aktuální řádek do schránky
- p ... vloží obsah schránky za pozici kurzoru
- P ... vloží obsah schránky před pozici kurzoru
- ZZ ... uloží aktuální soubor a ukončí editor vi
- u ... vrátí zpět poslední akci
- / ... vyhledávání – za lomenu napíšeme řetězec, který hledáme a dáme Enter. Hledá se od pozice kurzoru vpřed.

- n ... zobrazí další výskyt hledaného řetězce ve směru vpřed.
- ? ... vyhledávání – podobně jako /, ale v opačném směru
- N ... podobně jako n zobrazí výskyt hledaného řetězce, ale v opačném směru
- Pokud chceme příkaz použít pro více jednotek, napíšeme číslo reprezentující počet před začátek příkazu. Např. 4dd smaže 4 řádky, 5yy kopíruje 5 řádků do schránky, 5x smaže 5 znaků atd.

Některé základní příkazy v řádkovém příkazovém režimu

- :w ... uloží aktuální dokument
- :q ... ukončí editor vi, pokud nebyly udělány žádné změny v souboru
- :q! ... ukončí editor vi bez uložení
- :wq ... uloží soubor a ukončí editor vi
- :w jméno_souboru ... uloží dokument do zvoleného souboru, ale pokračujeme v editování původního dokumentu.

6.2.1 Cvičení

Ve svém domovském adresáři vytvořte a editujte editorem vi soubor pokus.

Napište do něj deset řádků textu, vyzkoušejte příkazy dd, yy, 5dd, 4yy (apod.), p, P, x, D, vyhledávání pomocí /, ?, n, N.

Uložte soubor a ukončete vi.

Pak do něj editorem vi vstupte znovu, provedte nějaké změny, uložte pod názvem pokus2 a ukončete vi.

Prohlédněte si oba soubory.

Nakonec oba soubory smažte (rm).

6.3 Odkaz na soubor (link)

Link je odkaz na nějaký soubor. Jména se mohou lišit. Na jeden soubor může existovat více odkazů s různými jmény.

Link se vytváří příkazem ln.

Existují dva typy linků:

- **hard link**
- **symbolic link** (symlink)

6.4 Vytváření linků

6.4.1 Hard link

Hard link se vytvoří příkazem **ln puvodni_soubor novy_soubor**.

Hard link ukazuje na inode již existujícího souboru. K danému souboru lze přistupovat pomocí obou jmen – původního jména i jména hard linku.

Nelze určit, který ze souborů existoval dřív, který je originálem.

Hard link je možné použít pouze tehdy, když soubor i link leží na stejném souborovém systému, protože čísla inodů jsou unikátní v rámci stejného souborového systému.

Výpis souborů se zobrazením jejich inodů je možno pomocí příkazu **ls -i**.

6.4.2 Příklad

Vytvoříme adresář pokus, v něm soubor original. Zobrazíme si číslo jeho inodu.

Vytvoříme hard link na soubor original pod jménem kopie. Zobrazíme si čísla jejich inodů – budou shodná.

```
student@linux:~$ mkdir pokus
student@linux:~$ cd pokus/
student@linux:~/pokus$ touch original
student@linux:~/pokus$ ls -i
117122 original
student@linux:~/pokus$ ls -il
celkem 0
117122 -rw-r--r-- 1 student users 0 2005-12-03 09:29 original
student@linux:~/pokus$ ln original kopie
student@linux:~/pokus$ ls -il
celkem 0
117122 -rw-r--r-- 2 student users 0 2005-12-03 09:29 kopie
117122 -rw-r--r-- 2 student users 0 2005-12-03 09:29 original
student@linux:~/pokus$
```

Ve výpisu **ls -il** je vidět, že na soubor s inodem 117122 ukazuje jak kopie, tak original. Číslo „2“ znamenají, že na inode 117122 existují dva odkazy.

6.4.3 Symbolic link (symlink)

Symbolický link se vytvoří příkazem **ln -s puvodni novy_soubor**. Odkazuje na jméno souboru.

Symbolic link má vlastní inode, proto je možno původní soubor a odkaz na něj rozlišit.

Symbolic link a soubor, na který odkazuje, mohou být na různých souborových systémech.

Nevýhodou může být, že symbolic link může odkazovat na soubor, který již nemusí odkazovat.

6.4.4 Příklad

Vytvoříme v adresáři pokus soubor original.

Vytvoříme symbolický odkaz na soubor original pod jménem odkaz.

Zobrazíme si dlouhý výpis adresáře pokus se zobrazením inodů.

Je vidět, že symbolický odkaz a původní soubor mají odlišné inody. Navíc u symbolického odkazu je ve výpisu vidět písmeno „l“

v oprávněních, což znamená symbolický odkaz. Navíc je vidět, na jaký soubor odkaz odkazuje.

Vytvoříme dále hard link na soubor original pod jménem pevný_odkaz.

Zobrazíme dlouhý výpis včetně inodů souborů. Pevný odkaz i originál souboru mají stejný inode a číslo 2 znamená počet hard linků na daný inode.

```
student@linux:~/pokus$ touch original
student@linux:~/pokus$ ln -s original odkaz
student@linux:~/pokus$ ls -il
celkem 0
117124 lrwxrwxrwx  1 student users 8 2005-12-03 09:44 odkaz -> original
117122 -rw-r--r--  1 student users 0 2005-12-03 09:43 original
student@linux:~/pokus$ ln original pevný_odkaz
student@linux:~/pokus$ ls -il
celkem 0
117124 lrwxrwxrwx  1 student users 8 2005-12-03 09:44 odkaz -> original
117122 -rw-r--r--  2 student users 0 2005-12-03 09:43 original
117122 -rw-r--r--  2 student users 0 2005-12-03 09:43 pevný_odkaz
```

Pokud smažeme soubor original, pak bude symbolický odkaz odkazovat na neexistující soubor.

6.5 Inode, vyhledání všech odkazů na jeden inode

ls -IRi | grep číslo_inodu

R ... rekurzivní hledání i v podadresářích

nebo rekurzivní výpis adresáře přesměrujeme do souboru a v něm pak číslo inodu vyhledáme (příkazem less zobrazíme obsah souboru a vyhledáváme pomocí /^číslo_inodu ... stříška za lomítkem říká, že číslo má být na začátku řádku).

```
ls -IRi > vypis  
less vypis  
/^117122
```

6.6 Cvičení

- Ve svém domovské adresáři vytvořte adresář „pokus1“ a „pokus2“.
- V adresáři „pokus1“ vytvořte soubor „soubor1“.
- V editoru vi do něj vložte deset řádků textu a uložte.
- V adresáři „pokus2“ vytvořte hard link a symbolic link na „soubor1“ pod názvy „soubor1_hl“ a „soubor1_sl“.
- Zobrazte výpis adresářů „pokus1“ a „pokus2“ v dlouhé formě včetně inodů.
- Ve svém domovském adresáři vytvořte pevný a symbolický odkaz na soubor „soubor1_hl“ pod názvy „soubor1_hl_hl“ a „soubor1_hl_sl“.
- Vypište si rekurzivně obsah svého domovského adresáře v podrobné formě včetně inodů.
- Příkazem grep vypište jen ty řádky, které obsahují číslo inodu souboru „soubor1“.
- Vypište si rekurzivně obsah svého domovského adresáře v podrobné formě včetně inodů a výpis přesměrujte do souboru „vypis“. V tomto souboru vyhledejte řádky obsahující řetězec „soubor1“
- Zobrazte obsah souborů „soubor1_hl_hl“ a „soubor1_hl_sl“.
- Smažte soubor „soubor1_hl“.
- Zobrazte obsah souborů „soubor1_hl_hl“ a „soubor1_hl_sl“.
- Smažte všechny nově vytvořené soubory a adresáře.

 *Domácí úkol*

- ❑ Procvičte si práci s Midnight Commanderem.
- ❑ Jaké typy souborů najdeme v linuxovém file systému?
- ❑ Procvičte si práci s editorem vi (včetně klávesových zkratk příkazů). V jakých režimech v editoru vi pracujeme?
- ❑ Vytvořte si ve svém domovském adresáři soubor a vytvořte na něj několik hard linků a symlinků z různých míst file systému.
- ❑ Pomocí rekurzivního výpisu adresářů zobrazte všechny soubory odkazující na původní soubor a na jeho inode.
- ❑ Vytvořené soubory smažte.
- ❑ Jaký je rozdíl mezi hard linkem a symlinkem?

 *Shrnutí*

- ✓ Seznámili jste se s Midnight Commanderem.
- ✓ Seznámili jste se s editorem vi a umíte v něm přecházet mezi jeho základními třemi režimy, umíte používat zkratkové klávesy pro práci v tomto editoru.
- ✓ Znáte rozdíl mezi symbolickým a pevným odkazem na soubor.
- ✓ Zopakovali jste si, co znamená inode.
- ✓ Umíte vyhledat soubory se stejným inodem.

7. Příkazy tee, locate, find, shutdown, cal, bc, mv, cp, mkdir, rm, rmdir (13. - 14. hodina)

7.1 Příkaz tee

Příkaz **tee** čte ze standardního vstupu a zapisuje na standardní výstup a do souborů.

7.1.1 Příklad

- **tee soubor1**
- **tee soubor1 soubor2**
- **tee -a soubor1**

Ve všech třech případech se po zadání příkazu objeví kurzor na novém řádku a čeká na vstup z klávesnice, obecně ze standardního vstupu.

To, co napíšeme, se zapíše v prvním případě do souboru **soubor1**, v druhém případě do souborů **soubor1** a **soubor2**, v třetím případě se pomocí parametru **-a** přidá na konec souboru **soubor1**.

Kromě zápisu do souboru se vše také vypisuje na standardní výstup, tedy na obrazovku.

Zadávání vstupů ukončíme klávesovou zkratkou **CTRL+D**.

7.1.2 Příklad

ls -al | tee soubor1

Tímto příkazem zobrazíme obsah aktuálního adresáře a navíc tento výstup předáme příkazu tee, který jej bude považovat za svůj standardní vstup a tudíž výpis zapíše do souboru **soubor1**.

7.1.3 Příklad

ls -al | tee soubor1 soubor2

Tímto příkazem zobrazíme obsah aktuálního adresáře a navíc tento výstup předáme příkazu tee, který jej bude považovat za svůj standardní vstup a tudíž výpis zapíše do souborů **soubor1** a **soubor2**.

7.2 Příkaz locate

Příkaz pro rychlé vyhledávání. Vyhledává rychleji než find, protože hledá v indexované databázi všech souborů jménem locatedb, která se pravidelně obnovuje.

locate název_souboru

S parametrem **-i** vyhledáváme soubor bez ohledu na velikost písmen.

locate -i název_souboru

7.2.1 Příklad

locate dmsg

```
student@linux:~/tvy$ locate dmsg
/usr/share/gettext/intl/loadmsgcat.c
```

Vyhledá umístění souborů, které mají v názvu dmsg.

Poznámka

Pokud chceme zaktualizovat databázi **updatedb**, která obsahuje informace o umístění souborů, zadáme jako root příkaz **updatedb**.

7.2.2 Cvičení

Zaktualizujte databázi **locatedb** a příkazem **locate** vyhlejte umístění souboru **locatedb**.

7.3 Příkaz find

Příkaz pro vyhledávání. Automaticky funguje rekurzivně, což znamená, že zahájí hledání ve specifikovaném adresáři a pokračuje do podadresářové struktury.

find výchozí_adresář -name vzor_který_hledáme

7.3.1 Příklad

```
find ~ -name „*tvy*“
```

Vyhledá v domovském adresáři soubory, které obsahují v názvu řetězec **tv**.

Jestliže uzavřeme vyhledávaný řetězec do uvozovek, je hledání spolehlivější.

Jestliže chceme, aby příkaz **find** při vyhledávání ignoroval malá a velká písmena, použijeme přepínač **-iname**.

```
find ~ -iname „*tvy*“
```

Pro vyhledávání můžeme použít **regulární výrazy**.

7.3.2 Příklad

find ~ -name „[a-mz]*“

Tímto příkazem vyhledáme v domovském adresáři všechny soubory, které začínají na **a-m** nebo **z**.

Lze vyhledat soubory, které byly změněny během posledních x dní pomocí volby **-mtime** nebo například minut pomocí volby **-mmin**.

7.3.3 Příklad

find ~ -mtime -1

V domovském adresáři vyhledá soubory změněné během posledního jednoho dne.

7.3.4 Příklad

find ~ -mmin -10

V domovském adresáři vyhledá soubory změněné během posledních 10 minut.

7.4 Příkaz shutdown

Z příkazové řádky můžeme počítač vypnout nebo restartovat.

shutdown -r now

Tímto příkazem restartujeme počítač okamžitě.

shutdown -h now

Tímto příkazem vypneme počítač okamžitě.

shutdown -r 10

Tímto příkazem restartujeme počítač za 10 minut.

7.5 Příkaz cal

Příkaz pro zobrazení kalendáře.

cal 5 2006

zobrazí kalendář pro květen 2006.

Zobrazení začíná nedělí.

Pokud chceme získat zobrazení začínající dnem pondělí, použijeme přepínač **-m**.

cal -m 5 2006

zobrazí kalendář pro květen 2006 a první v pořadí je pondělí.

```
student@linux:~$ cal 5 2006
    Květen 2006
Ne  Po  Út  St  Čt  Pá  So
   7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31

student@linux:~$ cal -m 5 2006
    Květen 2006
Po  Út  St  Čt  Pá  So  Ne
   1  2  3  4  5  6  7
   8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31
```

Pokud chceme zobrazit měsíc současný, předchozí a následující, použijeme přepínač **-3**.

cal -3

A když tento výpis chceme začínat od pondělí, přidáme k přepínači **-3** přepínač **-m**.

cal -3m

```
student@linux:~$ cal -m3
    Listopad 2005      Prosinec 2005      Leden 2006
Po  Út  St  Čt  Pá  So  Ne  Po  Út  St  Čt  Pá  So  Ne  Po  Út  St  Čt  Pá  So  Ne
   1  2  3  4  5  6   5  6  7  8  9 10 11   2  3  4  5  6  7  8
   7  8  9 10 11 12 13   12 13 14 15 16 17 18   9 10 11 12 13 14 15
 14 15 16 17 18 19 20   19 20 21 22 23 24 25   16 17 18 19 20 21 22
 21 22 23 24 25 26 27   26 27 28 29 30 31   23 24 25 26 27 28 29
 28 29 30                30 31   30 31
```

Pro zobrazení kaledáře na celý rok použijeme přepínač **-y**.

cal -y

Pro zobrazení kaledáře na určitý rok, například 2006, napíšeme za přepínač **-y číslo roku**.

cal -y 2006

```
student@linux:~$ cal -y
                2005
    Leden          Únor          Březen
Ne Po Út St Čt Pá So Ne Po Út St Čt Pá So Ne Po Út St Čt Pá So
  2  3  4  5  6  7  8   6  7  8  9 10 11 12   6  7  8  9 10 11 12
  9 10 11 12 13 14 15  13 14 15 16 17 18 19  13 14 15 16 17 18 19
 16 17 18 19 20 21 22  20 21 22 23 24 25 26  20 21 22 23 24 25 26
 23 24 25 26 27 28 29  27 28                27 28 29 30 31
 30 31

    Duben          Květen         Červen
Ne Po Út St Čt Pá So Ne Po Út St Čt Pá So Ne Po Út St Čt Pá So
  3  4  5  6  7  8  9   1  2  3  4  5  6  7   5  6  7  8  9 10 11
 10 11 12 13 14 15 16  15 16 17 18 19 20 21  12 13 14 15 16 17 18
 17 18 19 20 21 22 23  22 23 24 25 26 27 28  19 20 21 22 23 24 25
 24 25 26 27 28 29 30  29 30 31  26 27 28 29 30

    Červenec       Srpen         Září
Ne Po Út St Čt Pá So Ne Po Út St Čt Pá So Ne Po Út St Čt Pá So
  3  4  5  6  7  8  9   7  8  9 10 11 12 13   4  5  6  7  8  9 10
 10 11 12 13 14 15 16  14 15 16 17 18 19 20  11 12 13 14 15 16 17
 17 18 19 20 21 22 23  21 22 23 24 25 26 27  18 19 20 21 22 23 24
 24 25 26 27 28 29 30  28 29 30 31  25 26 27 28 29 30
 31

    Říjen          Listopad       Prosinec
Ne Po Út St Čt Pá So Ne Po Út St Čt Pá So Ne Po Út St Čt Pá So
  2  3  4  5  6  7  8   6  7  8  9 10 11 12   4  5  6  7  8  9 10
  9 10 11 12 13 14 15  13 14 15 16 17 18 19  11 12 13 14 15 16 17
 16 17 18 19 20 21 22  20 21 22 23 24 25 26  18 19 20 21 22 23 24
 23 24 25 26 27 28 29  27 28 29 30  25 26 27 28 29 30 31
 30 31
```

```
student@linux:~$ cal -y 2006
                2006
    Leden          Únor          Březen
Ne Po Út St Čt Pá So Ne Po Út St Čt Pá So Ne Po Út St Čt Pá So
  1  2  3  4  5  6  7   5  6  7  8  9 10 11   5  6  7  8  9 10 11
  8  9 10 11 12 13 14  12 13 14 15 16 17 18  12 13 14 15 16 17 18
 15 16 17 18 19 20 21  19 20 21 22 23 24 25  19 20 21 22 23 24 25
 22 23 24 25 26 27 28  26 27 28                26 27 28 29 30 31
 29 30 31

    Duben          Květen         Červen
Ne Po Út St Čt Pá So Ne Po Út St Čt Pá So Ne Po Út St Čt Pá So
  2  3  4  5  6  7  8   7  8  9 10 11 12 13   4  5  6  7  8  9 10
  9 10 11 12 13 14 15  14 15 16 17 18 19 20  11 12 13 14 15 16 17
 16 17 18 19 20 21 22  21 22 23 24 25 26 27  18 19 20 21 22 23 24
 23 24 25 26 27 28 29  28 29 30 31  25 26 27 28 29 30
 30

    Červenec       Srpen         Září
Ne Po Út St Čt Pá So Ne Po Út St Čt Pá So Ne Po Út St Čt Pá So
  2  3  4  5  6  7  8   6  7  8  9 10 11 12   3  4  5  6  7  8  9
  9 10 11 12 13 14 15  13 14 15 16 17 18 19  10 11 12 13 14 15 16
 16 17 18 19 20 21 22  20 21 22 23 24 25 26  17 18 19 20 21 22 23
 23 24 25 26 27 28 29  27 28 29 30 31  24 25 26 27 28 29 30
 30 31

    Říjen          Listopad       Prosinec
Ne Po Út St Čt Pá So Ne Po Út St Čt Pá So Ne Po Út St Čt Pá So
  1  2  3  4  5  6  7   5  6  7  8  9 10 11   3  4  5  6  7  8  9
  8  9 10 11 12 13 14  12 13 14 15 16 17 18  10 11 12 13 14 15 16
 15 16 17 18 19 20 21  19 20 21 22 23 24 25  17 18 19 20 21 22 23
 22 23 24 25 26 27 28  26 27 28 29 30  24 25 26 27 28 29 30
 29 30 31
```

7.6 Příkaz bc

Příkaz pro zobrazení jednoduchého kalkulátoru.
 Sčítá, odčítá a násobí i desetinná čísla, dělí celočíselně.

```
student@linux:~$ bc
bc 1.06
Copyright 1991-1994, 1997, 1998, 2000 Free Software
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
2+6.1
8.1
3.1/2
1
```

Ukončení počítání – CTRL+D.

7.7 Příkaz mv

Tento příkaz slouží pro přesun nebo přejmenování souboru.

mv zdroj cíl

7.7.1 Příklad

V domovském adresáři vytvoříme adresář „test“ a v něm vytvoříme soubor „soubor“.

```
mkdir test  
cd test  
touch soubor
```

Přejmenujeme soubor „soubor“ na „soubor1“.

```
mv soubor soubor1
```

Přesuneme soubor „soubor1“ o úroveň výš do domovského adresáře.

```
mv soubor1 ..
```

Přesuneme se do domovského adresáře a přesuneme soubor1 do adresáře test pod názvem soubor.

```
cd ..  
mv soubor1 test/soubor
```

Poznámka

Pokud někam přesouváme soubor a v daném umístění již soubor pod stejným názvem existuje, pak se může stát, že se existující soubor přepíše bez ptaní.

Interaktivitu dotazů zajistíme přepínačem **-i**, pak se nás systém zeptá, zda má existující soubor přepsat.

```
mv -i soubor2 test/soubor2
```

Pokud je systém nastaven tak, že se automaticky dotazuje, zda přepsat existující soubor a chceme na všechny takové dotazy odpovědět „yes“, pak tuto otázku potlačíme přepínačem **-f** (force) a soubory se přepíší.

```
mv -f soubor2 test/soubor2
```

7.8 Příkaz cp

Tento příkaz slouží pro kopírování souborů.

cp zdrojový_soubor cíl

7.8.1 Příklad

V domovském adresáři máme adresář „test“ a „test2“.

V adresáři „test“ je soubor „soubor1“. Zkopírujeme ho do adresáře „test2“.

cp test/soubor1 test2

Poznámka

Během kopírování můžeme zadat jméno novému souboru.

7.8.2 Příklad

V domovském adresáři máme adresář „test“ a „test2“.

V adresáři „test“ je soubor „soubor2“. Zkopírujeme ho do adresáře „test2“ pod názvem „zaloha“.

cp test/soubor2 test2/zaloha

Rekurzivní kopírování celé adresářové struktury

Toto zajistíme pomocí přepínače **-r**.

7.8.3 Příklad

V domovském adresáři máme adresář „test“ a „test2“.

Adresář „test“ obsahuje podadresáře „a1“ a „a2“ se soubory.

Adresář „test2“ je prázdný.

Obsah adresáře „test“ překopírujeme do adresáře „test2“.

cp test/* test2

Poznámka

Pokud někam kopírujeme soubor a v daném umístění již soubor pod stejným názvem existuje, pak se může stát, že se existující soubor přepíše bez ptaní, to závisí na nastavení systému.

Interaktivitu dotazů zajistíme přepínačem **-i**, pak se nás systém zeptá, zda má existující soubor přepsat.

cp -i test/* test2

Pokud je systém nastaven tak, že se automaticky dotazuje, zda přepsat existující soubor a chceme na všechny takové dotazy odpovědět „yes“, pak tuto otázku potlačíme přepínačem **-f** (force) a soubory se přepíší.

cp -f test/* test2

7.9 Příkaz mkdir

Příkaz pro vytvoření adresáře.

mkdir název_adresáře

7.9.1 Příklad

mkdir test2

vytvoří adresář test2 v aktuálním adresáři – relativní adresování.

mkdir /home/student/test2

vytvoří adresář test2 v domovském adresáři uživatele student – absolutní adresování.

☞ Vytvoření celé podadresářové struktury

Pokud chceme vytvořit **celou podadresářovou strukturu**, například test/prvni/druhy/treti/ctvrty, můžeme toto zajistit přepínačem **-p**.

mkdir -p test/prvni/druhy/treti/ctvrty

7.10 Příkaz rm

Příkaz pro smazání souboru.

rm název_souboru

☞ Poznámka

Pokud mažeme soubor, záleží na nastavení systému, zda se nás bude ptát, zda soubor smazat či ne.

Interaktivitu dotazů zajistíme přepínačem **-i**, pak se nás systém zeptá, zda má existující soubor smazat.

rm -i test/soubor

Pokud je systém nastaven tak, že se automaticky dotazuje, zda smazat soubor a chceme na všechny takové dotazy odpovědět „yes“, pak tuto otázku potlačíme přepínačem **-f** (force) a soubory se smažou bez ptaní.

rm -f test/soubor

Celou adresářovou strukturu včetně všech souborů a podadresářů smažeme pomocí přepínače **-R**.

rm -R test

Tímto příkazem smažeme adresář test včetně všech jeho souborů a podadresářů.

rm -R test/*

Tímto příkazem smažeme obsah adresáře test, sám adresář test zůstane.

7.11 Příkaz rmdir

Tento příkaz slouží pro smazání prázdného adresáře.

rmdir adresář

7.12 Cvičení

- ❑ Ve svém domovském adresáři vytvořte podadresář "cv1".
- ❑ V adresáři "cv1" vytvořte soubor "zivotopis.txt", do kterého vložíte svůj krátký životopis. Pro editování souboru využijte editor vi.
- ❑ V adresáři "cv1" vytvořte podadresářovou strukturu data/2004/leden až data/2004/prosinec, data/2005/leden až data/2005/prosinec, osobni/2004 a osobni/2005.
- ❑ Přesuňte soubor "zivotopis.txt" do adresáře "osobni/2005".
- ❑ Zkopírujte soubor "zivotopis.txt" do adresáře "osobni/2004" pod názvem "zivotopis04.txt"
- ❑ Upravte obsah souboru "zivotopis04.txt" tak, aby na začátku byl text "Životopis z roku 2004".
- ❑ Přesuňte se do domovského adresáře a vypište rekurzivně obsah adresáře "cv1".
- ❑ Tento výpis přesměrujte do souboru "vypis.txt" v adresáři "cv1/data2005/květen".
- ❑ Vypište rekurzivně obsah adresáře "cv1/data".
- ❑ Tento výpis přesměrujte do souboru "vypis_data.txt" v adresáři "cv1/".
- ❑ Přejmenujte tento soubor na soubor "data.txt".
- ❑ Přesuňte jej do adresáře „osobni/2004“.
- ❑ Smažte rekurzivně obsah adresáře „cv1“.
- ❑ Smažte prázdný adresář „cv1“.

Domácí úkol

- ❑ Ve svém domovském adresáři vytvořte podadresář "DU".
- ❑ V adresáři "DU" vytvořte soubor "ukol.txt", do kterého vložte výpis kurzovního lístku. Pro editování souboru využijte editor vi.
- ❑ V adresáři "DU" vytvořte podadresářovou strukturu prvni/druhy/treti.
- ❑ Zkopírujte soubor "ukol.txt" do všech adresářů prvni, druhy, treti.
- ❑ Smažte soubor "ukol.txt" z adresáře "DU".
- ❑ Vypište příkazem dmesg úvodní hlášení systému a tato hlášení přesměrujte do souboru „dmesg.txt“, který bude umístěn v adresáři „DU“.
- ❑ Příkazem locate najděte všechny soubory obsahující slovo „dmesg“.
- ❑ Zaktualizujte jako root databázi, ze které čerpá příkaz locate.
- ❑ Příkazem locate najděte všechny soubory obsahující slovo „dmesg“.
- ❑ Smažte rekurzivně obsah adresáře „DU“.
- ❑ Smažte prázdný adresář „DU“.

Shrnutí

- ✓ Umíte pracovat s příkazy pro vyhledávání locate, find, případně z dřívějších lekcí s příkazy grep, ls.
- ✓ Umíte vytvářet a mazat soubory a adresáře.
- ✓ Umíte kopírovat, přesouvat a přejmenovávat soubory.
- ✓ Umíte zobrazit kalendář, pracovat s jednoduchým kalkulátorem bc a příkazem vypnout a restartovat počítač.

8. Procesy (15. – 16. hodina)

8.1 Spuštění procesu v popředí, na pozadí, pozastavení procesu, ukončení procesu

V prostředí, kde funguje multitasking, musí procesor přidělovat svůj čas ve zlomcích sekundy jednotlivým procesům jednomu po druhém.

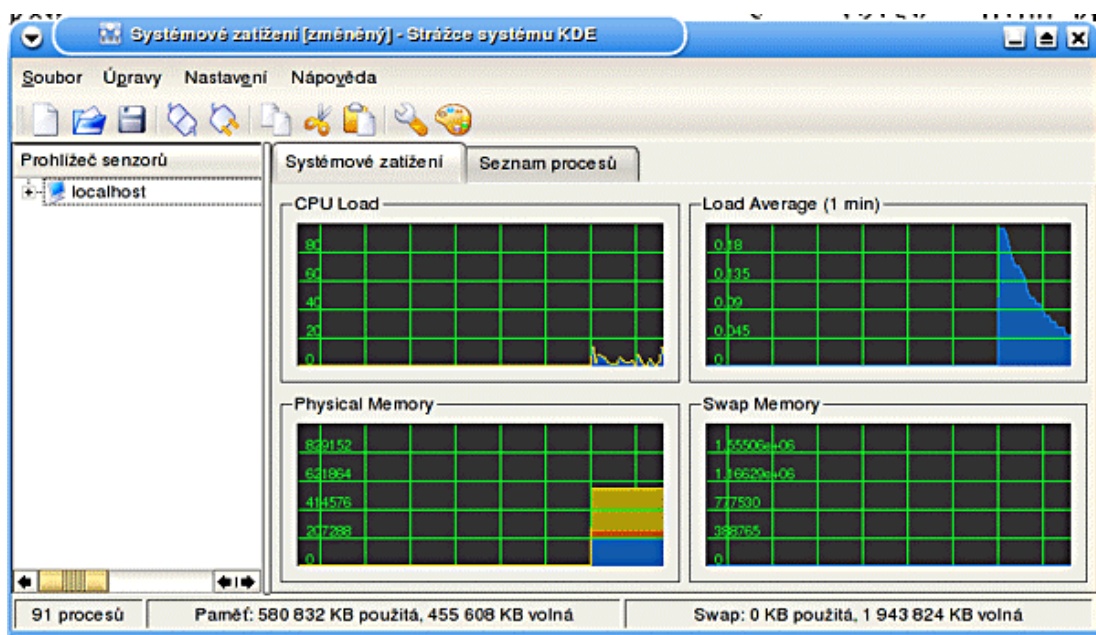
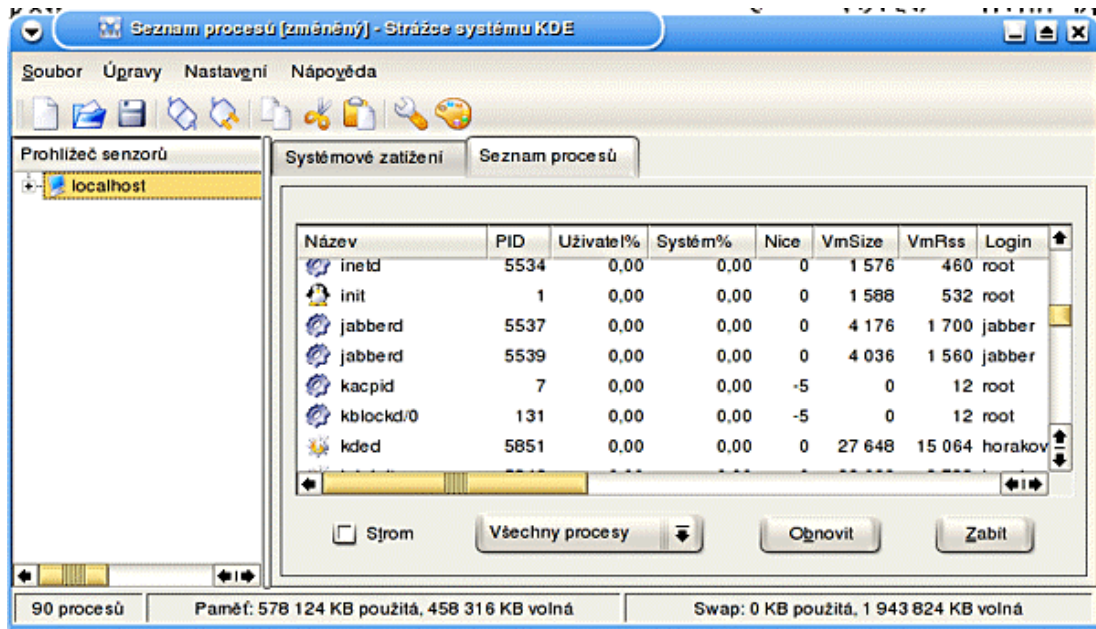
To poněkud zpomalí vykonávání procesu, ale na druhou stranu nemusí jeden proces čekat na kompletní dokončení jiného procesu.

Tak může současně běžet několik procesů bez příliši ztelného přerušování.

Příkazem **jobs** je možné zobrazit všechny procesy, které byly spuštěny z daného shellu a běží na pozadí.

```
sh-2.05b$ jobs
[1]-  Running          xcalc &
[2]+  Running          xclock &
sh-2.05b$
```

V grafickém prostředí je možné zobrazit běžící procesy pomocí nástroje **Kontrola systému pro prostředí KDE** (nabídka KDE – Systém – Monitor – Kontrola systému pro prostředí KDE).



8.1.1 Instalace balíčku

Pokud se tento nástroj nezobrazuje, nainstalujte si balíček **ksysguard**.

V **/etc/apt** je soubor **sources.list**, ve které je seznam zdrojů pro instalování. Znak # na začátku řádku znamená komentář. Pokud nějaký zdroj zakomentujeme, nebude se využívat.

Příklad tohoto souboru je níže:

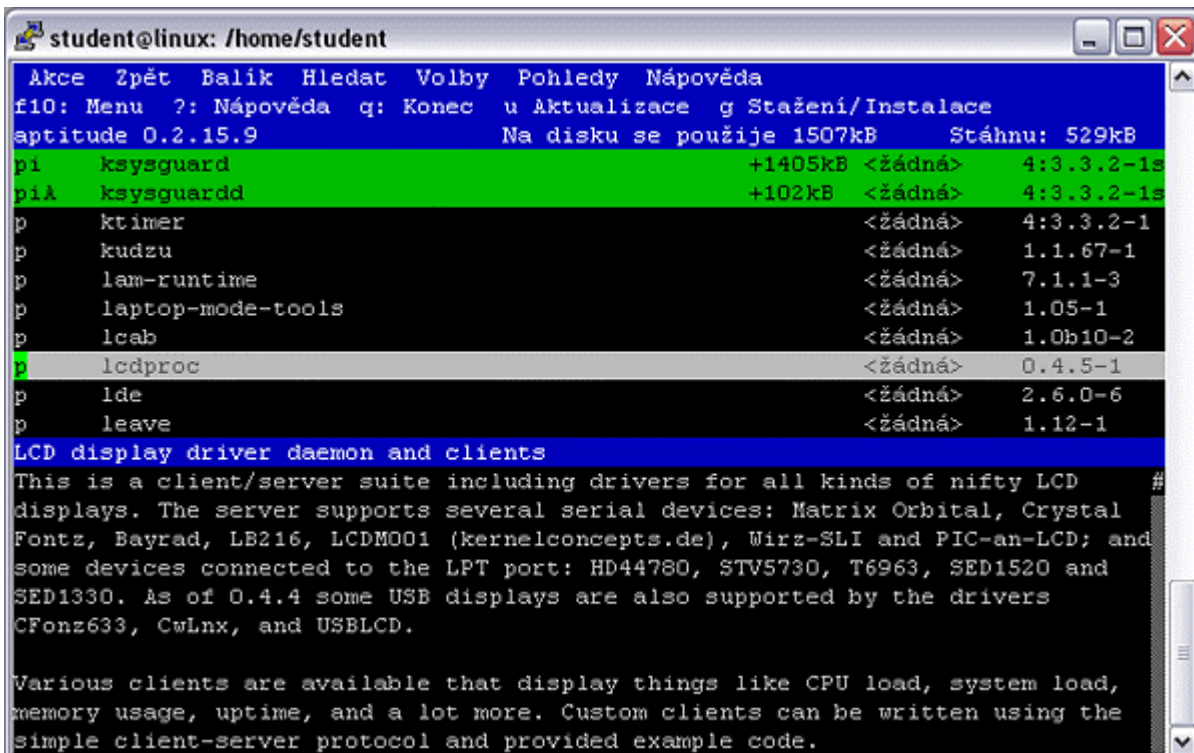
```
deb cdrom:[Debian GNU/Linux 3.1 r0a _Sarge_ - Official i386 Binary-2 (20050607)]/
unstable contrib main
deb cdrom:[Debian GNU/Linux 3.1 r0a _Sarge_ - Official i386 Binary-1 (20050607)]/
unstable contrib main
deb http://ftp.cz.debian.org/debian/ stable main non-free contrib
deb-src http://ftp.cz.debian.org/debian/ stable main non-free contrib
deb http://ftp.sk.debian.org/debian-volatile stable/volatile main
deb http://www.axis.cz/linux/debian stable axis
deb http://security.debian.org/ stable/updates main contrib non-free
deb http://debian.sh.cvut.cz/debian/ stable main non-free contrib
deb-src http://debian.sh.cvut.cz/debian/ stable main non-free contrib
```

V tomto seznamu vidíme, že zdrojem jsou 2 DVD s instalací Debian a dále se čerpá z několika ftp a http zdrojů.

Jako root spustíme **aptitude**.

Vyhledáme balíček. Napíšeme / a název hledaného balíčku ... **/ksysguard**. Pro potvrzení vyhledání stiskneme Enter.

Stojíme-li na správném řádku, napíšeme plus +. Tím se balíček označí k instalaci.

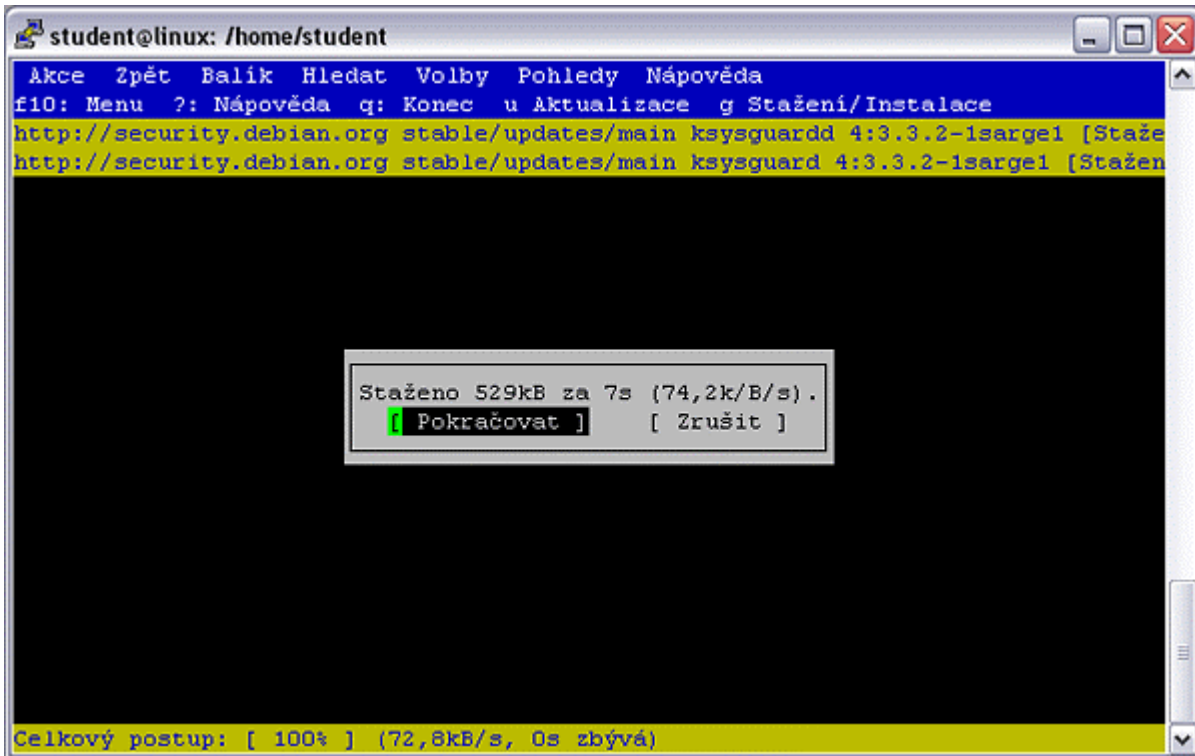


```
student@linux: /home/student
Akce Zpět Balík Hledat Volby Pohledy Nápověda
f10: Menu ?: Nápověda q: Konec u Aktualizace g Stažení/Instalace
aptitude 0.2.15.9 Na disku se použije 1507kB Stáhnou: 529kB
pi ksysguard +1405kB <žádná> 4:3.3.2-1s
pi ksysguardd +102kB <žádná> 4:3.3.2-1s
p ktimer <žádná> 4:3.3.2-1
p kudzu <žádná> 1.1.67-1
p lam-runtime <žádná> 7.1.1-3
p laptop-mode-tools <žádná> 1.05-1
p lcab <žádná> 1.0b10-2
p lcdproc <žádná> 0.4.5-1
p lde <žádná> 2.6.0-6
p leave <žádná> 1.12-1
LCD display driver daemon and clients
This is a client/server suite including drivers for all kinds of nifty LCD
displays. The server supports several serial devices: Matrix Orbital, Crystal
Fontz, Bayrad, LB216, LCDMO01 (kernelconcepts.de), Wirz-SLI and PIC-an-LCD; and
some devices connected to the LPT port: HD44780, STV5730, T6963, SED1520 and
SED1330. As of 0.4.4 some USB displays are also supported by the drivers
CFonz633, CwLnx, and USBLCD.

Various clients are available that display things like CPU load, system load,
memory usage, uptime, and a lot more. Custom clients can be written using the
simple client-server protocol and provided example code.
```

Automaticky se vyberou i balíčky, které souvisejí s instalací vybraného balíčku.

Pak stiskneme **g**, zobrazí se informace o tom, co se bude instalovat, kolik dat se stáhne a podobně. Instalaci spustíme dalším stisknutím **g**.



```

student@linux: /home/student
Akce Zpět Balík Hledat Volby Pohledy Nápověda
f10: Menu ?: Nápověda q: Konec u Aktualizace g Stážení/Instalace
http://security.debian.org stable/updates/main ksysguardd 4:3.3.2-1sarge1 [Stáže
http://security.debian.org stable/updates/main ksysguard 4:3.3.2-1sarge1 [Stážen

Stáženo 529kB za 7s (74,2kB/s).
[ Pokračovat ] [ Zrušit ]

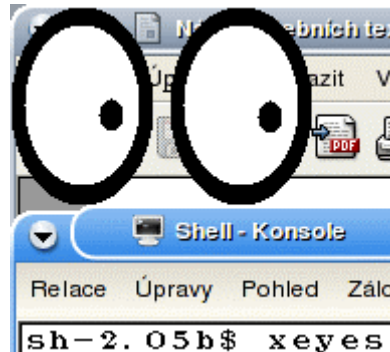
Celkový postup: [ 100% ] (72,8kB/s, 0s zbývá)
    
```

Pro ukončení aptitude stiskneme písmeno **q**.

8.1.2 Spouštění procesů na pozadí a v popředí

Příkazy v shellu mohou být spuštěny na pozadí nebo v popředí. Příkazy spuštěné v popředí přijímají vysílané signály přímo.

Např. chceme-li spustit aplikaci xeyes v popředí, spustíme ji příkazem **xeyes**



Přerušeni procesu provedeme klávesovou zkratkou **CTRL+Z**. Proces není ukončen, je pouze stopnut.

Pro zobrazení procesů z příkazového řádku lze použít příkaz **jobs**.

Pro opětovné spuštění tohoto procesu na pozadí použijeme příkaz **bg**

případně **bg číslo_job_ID**, pokud by bylo přerušeno více procesů a museli bychom specifikovat číslo procesu – job ID.

Pokud chceme spustit daný proces na pozadí přímo, použijeme v zápisu příkazu na konci ampersand &.

xeyes &

```
sh-2.05b$ xeyes
[1]+  Stopped (SIGTSTP)          xeyes
sh-2.05b$ jobs
[1]+  Stopped (SIGTSTP)          xeyes
sh-2.05b$ bg
[1]+  xeyes &
sh-2.05b$ jobs
[1]+  Running                    xeyes &
sh-2.05b$ jobs
[1]+  Done                        xeyes
sh-2.05b$ xeyes &
[1] 6269
sh-2.05b$
```

Protože byl příkaz spuštěn na pozadí, jsme vráceni do příkazového řádku shellu a můžeme spouštět další příkazy.

Ukončit běžící aplikaci můžeme pomocí pravého tlačítka myši na položce aplikace na hlavním panelu.

Každému procesu spuštěnému z shellu je přiřazeno job ID. Výpis provedeme příkazem **jobs**.

Proces běžící na pozadí můžeme přenést do popředí příkazem fg následovaným svým číslem job ID.

fg 1

```
sh-2.05b$ xeyes &
[1] 6275
sh-2.05b$ fg
xeyes
```

8.2 Výpis procesů

8.2.1 Příkaz ps

Pro zobrazení běžících procesů lze použít příkaz **ps** (process status). Tímto příkazem se zobrazí běžící procesy na daném terminálu.

```
sh-2.05b$ ps
  PID TTY          TIME CMD
  5929 pts/1        00:00:00 sh
  6282 pts/1        00:00:00 ps
sh-2.05b$ █
```

Pro zobrazení procesů na všech terminálech přidáme parametr **x**.

ps x

```
sh-2.05b$ ps x
  PID TTY          STAT      TIME COMMAND
  5558 ?        Ss         0:00 /usr/sbin/famd -T O
  5788 ?        S          0:00 /bin/sh /usr/bin/startl
  5816 ?        Ss         0:00 /usr/bin/ssh-agent /usr
  5843 ?        Ss         0:00 kdeinit Running...
  5846 ?        S          0:00 dcopserver [kdeinit] de
  5848 ?        S          0:00 klauncher [kdeinit] kla
  5851 ?        S          0:00 kded [kdeinit] kded
  5858 ?        R          0:00 /usr/bin/artsd -F 10 -s
  5861 ?        S          0:00 kxkb [kdeinit] kxkb
  5863 ?        S          0:00 knotify [kdeinit] knot:
  5868 ?        S          0:00 kwrapper ksmserver
  5870 ?        S          0:00 ksmserver [kdeinit] ksm
  5871 ?        S          0:00 kwin [kdeinit] kwin -se
  5875 ?        S          0:00 khotkeys [kdeinit] khot
```

Ke každému procesu je přiřazeno unikátní systémové číslo **PID** – process identification number. PID rovno **1** patří "otci" všech procesů – procesu **init**. Je to první spuštěný proces systému a spouští další procesy, které mohou spouštět další procesy.

Parametry příkazu ps:

- x ... seznam procesů nezávisle na terminálu
- a ... seznam procesů nezávisle na uživateli
- l ... zobrazí podrobnější seznam i s detaily o procesech
- u ... seznam řadí podle uživatele
- f ... seznam řadí hierarchicky do stromu

Výpis ps může obsahovat sloupce

- **UID** ... číslo uživatele
- **PID** ... číslo procesu
- **PPID** ... číslo rodičovského procesu
- **TTY** ... číslo terminálu
- **PRI** ... priorita – nižší číslo znamená, že procesor tomuto procesu přiřadí více času
- **NI** ... nice – ovlivňuje dynamicky přidělenou prioritu, proces lze spustit se zvolenou prioritou
- **SIZE** ... celkové využití paměti procesem
- **STAT** ... status procesu (R – runnable – proces může být spuštěn, S – sleeping – proces čeká na nějakou externí událost, např. příchod dat, D – nepřerušitelné čekání – proces nemůže být v danou chvíli ukončen, T – proces je pozastaven, Z – zombie – proces se již ukončil, ale jeho návratová hodnota zatím nebyla předána, systém neví o jeho ukončení)
- **TIME** ... využitý čas počítače
- **COMMAND** ... název příkazu

☞ Poznámka

Pokud chceme z výpisu procesů vyfiltrovat jen ty řádky, které obsahují např. slovu "cup", použijeme přesměrování do příkazu grep.

ps ax | grep „cup“

```
sh-2.05b$ ps x
  PID TTY          STAT       TIME COMMAND
 5558 ?           Ss          0:00 /usr/sbin/famd -T
 5788 ?           S           0:00 /bin/sh /usr/bin/s
 5816 ?           Ss          0:00 /usr/bin/ssh-agent
 5843 ?           Ss          0:00 kdeinit Running...
 5846 ?           S           0:00 dcopserver [kdeini
 5848 ?           S           0:00 klauncher [kdeinit
 5851 ?           S           0:00 kded [kdeinit] kde
 5858 ?           S           0:00 /usr/bin/artsd -F
 5861 ?           S           0:00 kxkb [kdeinit] kxk
 5863 ?           S           0:00 knotify [kdeinit]
 5868 ?           S           0:00 kwrapper ksmserver
```

```
sh-2.05b$ ps a
  PID TTY          STAT       TIME COMMAND
 5599 tty1        Ss          0:00 -bash
 5601 tty2        Ss+         0:00 /sbin/getty 38400 tty2
 5602 tty3        Ss+         0:00 /sbin/getty 38400 tty3
 5603 tty4        Ss+         0:00 /sbin/getty 38400 tty4
 5604 tty5        Ss+         0:00 /sbin/getty 38400 tty5
 5605 tty6        Ss+         0:00 /sbin/getty 38400 tty6
 5929 pts/1      Rs          0:00 /bin/sh
 6301 tty1        S+          0:00 bc
 6312 pts/1      R+          0:00 ps a
```



```
sh-2.05b$ ps l
F  UID      PID  PPID  PRI   NI   VSZ  RSS  WCHAN  STAT
O  1001    5929  5928   15    0   3148 1720  wait   Ss
O  1001    6316  5929   15    0   2380  684  -      R+
```

```
sh-2.05b$ ps u
USER      PID  %CPU  %MEM    VSZ   RSS  TTY      STAT
horakova  5929  0.0   0.1   3148  1720 pts/1    Ss
horakova  6317  0.0   0.0   2568   844 pts/1    R+
```

```
sh-2.05b$ ps f
  PID  TTY      STAT    TIME  COMMAND
  5929 pts/1    Rs      0:00  /bin/sh
  6319 pts/1    S       0:00  \_ xcalc
  6320 pts/1    S       0:00  \_ xclock
  6321 pts/1    R+     0:00  \_ ps f
```

```
sh-2.05b$ ps au
USER      PID  %CPU  %MEM    VSZ   RSS  TTY      STAT
student   5599  0.0   0.1   3076  1644 tty1     Ss
root      5601  0.0   0.0   1580   488 tty2     Ss+
root      5602  0.0   0.0   1580   484 tty3     Ss+
root      5603  0.0   0.0   1580   484 tty4     Ss+
root      5604  0.0   0.0   1584   492 tty5     Ss+
root      5605  0.0   0.0   1584   492 tty6     Ss+
horakova  5929  0.0   0.1   3148  1720 pts/1    Ss
student   6301  0.0   0.0   2340   956 tty1     S+
horakova  6322  0.0   0.0   2568   848 pts/1    R+
[1]-  Done                    xcalc
[2]+  Done                    xclock
```

```
sh-2.05b$ ps aux
USER      PID  %CPU  %MEM    VSZ   RSS  TTY      STAT
root      1  0.0   0.0   1588   520  ?       S
root      2  0.0   0.0    0      0  ?       S
root      3  0.1   0.0    0      0  ?       S
root      4  0.0   0.0    0      0  ?       S
root      5  0.0   0.0    0      0  ?       S
root      7  0.0   0.0    0      0  ?       S
root     131  0.0   0.0    0      0  ?       S
root     157  0.0   0.0    0      0  ?       S
root     158  0.0   0.0    0      0  ?       S
root     160  0.0   0.0    0      0  ?       S
root     159  0.0   0.0    0      0  ?       S
root     745  0.0   0.0    0      0  ?       S
root     866  0.0   0.0    0      0  ?       S
root     872  0.0   0.0    0      0  ?       S
```

8.2.2 Příkaz pstree

Pro zobrazení stromové struktury, kdy vidíme, který proces spouští jiné, můžeme použít příkaz **pstree**.

Jestliže chceme ukončit celou sadu procesů v některé větvi stromu, stačí ukočit "otce" těchto procesů, ze kterého vycházejí tyto procesy.

Pro zobrazení PID procesu použijeme parametr **-p**, pro zobrazení UID použijeme parametr **-u**.

pstree -pu

```
sh-2.05b$ pstree
init--apache2---5*[apache2]
      |
      |--atd
      |--bash---bc
      |--clamd
      |--cron
      |--dbus-daemon-1
      |--dcopserver
      |--events/0
      |--exim4
      |--famd
      |--freshclam
      |--gconfd-2
      |--5*[getty]
      |--gpm
      |--hald
      |--inetd
      |--jabberd---jabberd
      |--kded
      |--kdeinit--artsd
                  |--2*[kio_file]
```

```
sh-2.05b$ pstree -pu
init(1)--apache2(5575)--apache2(5579, www-data)
          |             |--apache2(5580, www-data)
          |             |--apache2(5581, www-data)
          |             |--apache2(5582, www-data)
          |             |--apache2(5583, www-data)
          |
          |--atd(5567, daemon)
          |--bash(5599, student)---bc(6301)
          |--clamd(5353, clamav)
          |--cron(5570)
          |--dbus-daemon-1(5397, messagebus)
          |--dcopserver(5846, horakova)
          |--events/0(3)
          |--exim4(5494, Debian-exim)
          |--famd(5558, horakova)
          |--freshclam(5394, clamav)
          |--gconfd-2(5922, horakova)
```

8.2.3 Příkaz top

Pro zobrazení využití strojového času jednotlivými běžícími procesy můžeme využít příkaz `top`.

Výpis se aktualizuje každých 5 sekund.

Výpis je řazen podle využití strojového času.

Ukončení výpisu uděláme stisknutím klávesy **q**.

```
top - 14:21:13 up 29 min, 2 users, load average:
Tasks: 88 total, 1 running, 87 sleeping, 0 s
Cpu(s): 1.7% us, 0.3% sy, 0.0% ni, 98.0% id, 0
Mem: 1036440k total, 452372k used, 584068k f
Swap: 1943824k total, 0k used, 1943824k f
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM
5759	root	5	-10	42616	22m	1980	S	1.0	2.2
5871	horakova	15	0	26092	13m	11m	S	0.3	1.4
5928	horakova	15	0	29192	15m	12m	S	0.3	1.6
6109	horakova	16	0	26884	14m	11m	S	0.3	1.4
1	root	16	0	1588	520	452	S	0.0	0.1
2	root	34	19	0	0	0	S	0.0	0.0
3	root	10	-5	0	0	0	S	0.0	0.0
4	root	10	-5	0	0	0	S	0.0	0.0
5	root	10	-5	0	0	0	S	0.0	0.0
7	root	20	-5	0	0	0	S	0.0	0.0

8.3 Spuštění procesu se zvolenou prioritou

Proces jde spustit se zvolenou prioritou příkazem nice.

Priorita je číslo v rozmezí od -20 do 19, přičemž -20 je nejvyšší priorita a 19 nejnižší.

Záporné priority může přidělovat jen root.

nice -n 2 xclock &

spustí aplikaci xclock na pozadí s prioritou 2.

Běžný uživatel může spustit proces s prioritou jen nižší, než je jeho standardní priorita daného procesu.

8.4 Změna priority

Příkazem renice můžeme změnit prioritu spuštěného procesu.

Běžný uživatel může prioritu změnit jen na menší, než je aktuální priorita (je-li priorita 5, může nastavit jen 6, 7, ...).

Root může měnit prioritu libovolně.

Kontrolu můžete provést například v nástroji **Kontrola systému pro prostředí KDE**.

renice výše_priority -p číslo_procesu
renice 3 -p 6341

```
sh-2.05b$ nice -n 10 xeyes &
[1] 6434
sh-2.05b$ jobs
[1]+  Running                  nice -n 10 xeyes &
sh-2.05b$ renice 11 -p 6434
6434: old priority 10, new priority 11
```

8.5 Příkaz kill, killall

Příkaz kill vyžaduje parametr PID – číslo procesu, který má ukončit (PID zjistíme např. příkazem top nebo ps).

kill PID

Příkaz killall vyžaduje jako svůj parametr název procesu, který má ukončit.

killall název

Pokud ani jedním způsobem nelze proces ukončit, je možné použít násilné ukončení s využitím parametru **-9** nebo **-SIGKILL**.

kill -9 1522

killall -9 xcalc

Tento příkaz používejte jen vyjímečně, protože tím můžeme operačnímu systému způsobit jisté problémy s daty v bufferech apod.

8.6 Run levely – úrovně běhu

Úrovně běhu znamenají určité režimy chodu systému.

V souboru **etc/inittab** je definováno, jak jsou číslovány run levely.

- ❑ 0 ... halt - vypnutí
- ❑ 1 ... single user – jedno – uživatelské prostředí určené pro údržbu systému
- ❑ 2 – 5 ... multi user – mnohoživatelské prostředí
 - 2 – bez podpory sítě
 - 3 – většinou textové prostředí, vhodné pro servery
 - 4 – nemusí být definována
 - 5 – grafické prostředí
- ❑ 6 ... reboot – restart systému

Číslování run levelů se může lišit, úrovně běhu lze definovat.

Do daného run levelu se přepneme příkazem **init číslo_levelu**, např. `init 3`.

8.7 Zombie procesy

Procesy, které se již ukončily, ale nepředaly svou návratovou hodnotu procesu, který jej spustil.

8.8 Cvičení

- ❑ Spustíte grafický nástroj Kontrola systému pro prostředí KDE. Pokud není nainstalován, nainstalujte jej.
- ❑ Zjistěte, který proces má nejnižší a nejvyšší PID.
- ❑ Zobrazte sloupec PPID a zjistěte, kolik a jaké procesy spustil proces s číslem 1.
- ❑ Určete procesy s nejvyšší a nejnižší prioritou.
- ❑ Určete, které procesy zabírají nejvíce systémových prostředků.
- ❑ Zobrazte sloupec Stav a zjistěte, který proces je v danou chvíli „běžící“.
- ❑ Spustíte z konzole hodiny příkazem `xclock &` a pak v nástroji Kontrola systému pro prostředí KDE proces najdete, označte a stiskněte tlačítko Zabít - proces ukončete.

8.9 Cvičení

- ❑ Spustte na pozadí jednoduchý kalkulátor.
- ❑ Spustte na pozadí digitální hodiny, jejichž čas se bude obnovovat každou sekundu. Vyhledejte v manuálu pro příkaz xclock parametry pro zobrazení ve formě digitálních hodin.
- ❑ Vypište procesy.
- ❑ Určete čísla procesů PID spuštěných procesů.
- ❑ Napište příkaz pro ukončení kalkulátoru.
- ❑ Hodiny přeneste do popředí.
- ❑ Stopněte je a pak je přeneste do pozadí.
- ❑ Změňte prioritu hodin na 15.
- ❑ Zkontrolujte v grafickém správci, že změna proběhla.
- ❑ Zkuste prioritu nastavit na původní hodnotu – 0.
- ❑ Přepněte se na textovou konzolu na virtuální terminál TTY1 a spusťte příkazem bc kalkulátor. Přepněte se zpět do grafické konzoly a zkontrolujte běžící procesy na všech konzolách.
- ❑ Vypište graficky závislosti procesů.

 *Domácí úkol*

- Nainstalujte nástroj **Kontrola systému pro prostředí KDE**.
- Zobrazte běžící procesy jak pomocí příkazů tak pomocí tohoto nástroje.
- Vyzkoušejte spouštění procesu na pozadí, v popředí a změnu jejich stavu – stopnutí, přenesení do popředí nebo na pozadí atd.
- Zobrazte graficky závislosti spouštění procesů.
- Na jiné konzole spusťte nějaký proces a v grafické konzole zobrazte běžící procesy na všech konzolách.
- Určete, který proces zabírá nejvíc systémových prostředků.
- Spusťte proces se zvolenou prioritou.
- Změňte prioritu procesu.
- Ukončete proces.

 *Shrnutí*

- ✓ Víte, co jsou procesy a jak s nimi nakládá procesor v multitaskingovém prostředí.
- ✓ Umíte vypsat běžící procesy.
- ✓ Znáte příkazy jobs, top, ps i s jejich potřebnými parametry.
- ✓ Umíte spustit proces se zvolenou prioritou a změnit prioritu běžícímu procesu.
- ✓ Umíte nainstalovat potřebný balíček.
- ✓ Umíte se orientovat v nástroji **Kontrola systému pro prostředí KDE**.

9. Správa uživatelů (17. – 18. hodina)

9.1 Schémata uživatelů (privátní, veřejné)

Když vytváříme uživatele, je zařazen do své primární skupiny podle jednoho ze dvou základních schémat.

Privátní schéma

Uživateli je přiřazena primární skupina stejného jména (např. uživateli host by byla přiřazena primární skupina host), kterou může spravovat.

Veřejné schéma

Uživatel je přiřazen do primární skupiny, která je veřejnou skupinou, např. je přiřazen do skupiny „users“. Protože do této skupiny jsou přiřazováni všichni noví uživatelé, může ji spravovat jen systémový administrátor.

9.2 Založení uživatele

Z příkazového řádku lze založit uživatele příkazem

useradd jméno

Tím se založí uživatel do souboru **/etc/passwd** a **/etc/shadow**.

Tímto příkazem se sice vytvoří uživatel, ale nemá svůj domovský adresář a nemá heslo.

V souboru **/etc/passwd** jsou uloženi uživatelé a v souboru **/etc/shadow** jsou uloženi i se svými kryptovanými hesly.

Soubor **/etc/passwd** je určen pro čtení komukoliv, soubor **/etc/shadow** může číst jen root. Je to z toho důvodu, že kdyby byl soubor s kryptovanými hesly přístupný komukoliv, mohl by záškodník použít program, který se hrubou silou bude snažit rozluštit hesla.

9.2.1 Příklad

Provedte výpis souboru **/etc/passwd** a pokuste se i o výpis souboru **/etc/shadow**.

9.2.2 Struktura **/etc/shadow**

student:lsaf!jKS255sd:11525:0:99999:7:0:12345:

Vysvětlivky (zleva doprava)

jméno uživatele : zakryptované heslo : datum poslední změny : po kolika dnech může být heslo změněno : počet dnů před vypršením platnosti, kdy si uživatel může změnit heslo : kolik dní před vypršením platnosti hesla má být uživatel upozorněn : kolik dní je ještě heslo

použitelné, ačkoli jeho platnost již vypršela : datum, kdy je účet zamknut :

Pokud je v políčku pro heslo hvězdička * nebo vykřičník !, uživatel se nemůže přihlásit k systému, existuje jen kvůli využití nějakými programy nebo je jeho účet zamknut.

Pokud je políčko prázdné, může se uživatel přihlásit k systému bez hesla.

9.2.3 Struktura /etc/passwd

V **/etc/passwd** jsou informace o uživatelských účtech ve formátu:
host:x:1002:100:ucet pro hosta:/home/host:

Vysvětlivky parametrů:

název účtu : x znamená, že heslo existuje v /etc/shadow : UID
 uživatele : GID primární skupiny uživatele : komentář : domovský adresář
 :

9.2.4 Důležité parametry příkazu useradd

- **-m** ... **useradd -m jméno** ...automaticky se vytvoří domovský adresář v adresáři home. Do domovského adresáře se zkopírují soubory ze vzorového adresáře /etc/skel, který slouží jako šablona pro nově vznikající domovské adresáře účtů.
- **-c** ... během vytváření nového účtu je možné zadat komentář k danému účtu.
- **-g** ... je možno definovat primární skupinu účtu buď ve formě čísla GID nebo jména skupiny.
- **-p** ... je možno zadat heslo nově zakládanému účtu, heslo se ale musí zadávat již kryptovaně.
 Např. useradd -m -p „jsdfiklu58ESA“ student
- **-e** ... je možno zadat datum vypršení platnosti účtu.
 Např. useradd -m -e 2006-05-25 student

Pro rychlé vytvoření uživatele i s heslem se často používá kombinace příkazů useradd a passwd.

**useradd -m -c „studentský účet“ student
 passwd student**

9.3 Změna hesla uživatele

Uživatel si může změnit své heslo, root může měnit hesla všem.

passwd jméno

Pokud uživatel zadá jen příkaz **passwd**, bude měnit heslo sám sobě.

Parametry příkazu passwd

- **-l ... passwd -l student ...** účet bude zamknut
- **-u ... passwd -u student ...** účet bude odemknut
- **-S ... passwd -S student ...** výpis statutu uživatele
ve výpisu je jméno uživatele následované statutem – L – zamknut, NP – bez heslo, P – platné heslo, pak následuje datum poslední změny hesla, minimální doba platnosti, maximální doba platnosti, kolik dní před vypršením platnosti bude uživatel varován a poslední parametr je počet dní po vypršení platnosti, kdy ještě heslo bude akceptováno.
- **-i ... passwd -i číslo účet ...** nastaví počet dnů, po kterých se účet deaktivuje po vypršení platnosti hesla.
- **-n ... passwd -n číslo účet ...** nastaví minimální počet dnů před vypršením platnosti, během kterých si může uživatel změnit heslo.
- **-w ... passwd -w číslo účet ...** počet dnů před vypršením platnosti hesla, během kterých bude uživatel upozorněn systémem.
- **-x ... passwd -x číslo účet ...** nastaví maximální počet dnů, kdy je heslo platné

9.3.1 Příklad

passwd -x 100 -w 10 student

V tomto případě bude heslo platné po dobu 100 dnů a 10 dnů před vypršením platnosti bude uživatel upozorňován.

9.4 Změna parametrů uživatele

Příkazem `usermod` lze měnit některé parametry uživatelského účtu, jako například číslo UID, domovský adresář, primární skupinu uživatele.

usermod -d /novy_adresar/student -m student

změní domovský adresář uživatele student

usermod -u 105 student

změní UID uživatele student

usermod -g 1024 student

změní primární skupinu uživatele student, která musí existovat

usermod -G skupina,skupina2,skupina3 student

nastaví uživateli student další skupiny, jichž je členem. Pokud by uživatel byl členem skupiny, která není v seznamu uvedena, bude z ní odebrán.

9.5 Mazání uživatele

Příkazem `userdel` smažeme uživatelský účet.

userdel student

...smaže účet `student` – ze souborů `/etc/passwd`, `/etc/shadow`, `/etc/group`, ale domovský adresář nesmaže.

Pro to, aby se spolu s účtem smazal i domovský adresář, je nutné přidat parametr **-r**.

userdel -r student

9.6 Hesla uživatelů

Pokud chcete mít systém zabezpečen před útoky, je vhodné poučit uživatele o tom, jaká hesla by si měli, resp. neměli volit.

Nejsou vhodná hesla, která jdou snadno uhádnout. Např. heslo je podmnožinou uživatelského jména nebo v hesle se odráží některé známé informace o uživateli, jako jména jeho dětí, příbuzných, oblíbených písní, herců apod.

Vhodná jsou hesla, která obsahují alespoň 7 znaků, obsahují čísla i písmena, velká i malá, přičemž velké písmeno nemusí být jen na prvním místě a heslo je například složeninou nějakých zkratk z několika slov.

Dešifrovací programy často znají i zdánlivě bezpečné zaměnění nuly za O, jedničky za I, trojky za E apod.

9.7 Příklad

V `/etc/shadow` jsou záznamy ve formátu:

```
host:$1$gLiJCKz9$nNDsOVzS60QDIjXeNi2Pm0:13132:0:99999:7:::
```

V tomto příkladu byl uživatel **host** založen s **komentářem** a **domovským adresářem**.

Po příkazu **passwd -l host**, kdy účet zamkneme, je v `/etc/shadow` změna:

```
host:!!$1$gLiJCKz9$nNDsOVzS60QDIjXeNi2Pm0:13132:0:99999:7:::
```

... u hesla je vykřičník, což znamená, že účet je zamknut a nepůjde se na něj přihlásit.

Pro zobrazení statutu účtu použijeme příkaz **passwd -S host**, zobrazí **L**, což znamená Locked.

```
host L 12/15/2005 0 99999 7 -1
```

Po odemknutí příkazem **passwd -u host** vykřičník z `/etc/shadow` zmizí a účet je opět funkční.

Zobrazíme status účtu `host`:

host P 12/15/2005 0 9999 7 -1

P znamená, že heslo je platné.

Pokud v **/etc/shadow** zakryptované heslo uživatele host **smažeme**, bude po zobrazení statutu uživatele zobrazení **NP** – no password.

host NP 12/15/2005 0 9999 7 -1

Po zadání příkazu **passwd -i 5 host** se změní počet dnů, které uplynou po vypršení hesla a účet bude deaktivován.

host NP 12/15/2005 0 9999 7 5

v **/etc/shadow** bude změna – **host::13132:0:99999:7:5::**

Po zadání příkazu **passwd -n 14 host** bude v **/etc/shadow** změna:

host::13132:14:99999:7:5::

je to počet dnů před vypršením platnosti, kdy si uživatel může změnit heslo.

Pro zadání počtu dnů před vypršením účtu, během kterých má být uživatel upozorněn na nutnost změny zadáme příkaz:

passwd -w 8 host

V **/etc/shadow** se projeví změny:

host::13132:14:99999:8:5::

Po zadání příkazu **passwd -x 1000 host** bude změněna doba platnosti hesla. V **/etc/shadow** se projeví změna:

host::13132:14:1000:8:5::

Poznámka

Čísla, která znamenají data, jsou ve formátu počtu dnů od 1.1.1970.

Příkazem **passwd -S host** zobrazíme výsledný status uživatele host:

host NP 12/15/2005 14 1000 8 5

Vysvětlivky: název účtu je host, bez hesla, změna účtu proběhla 15.prosince 2005, 14 dní před vypršením platnosti hesla si ho může uživatel změnit, heslo je platné po dobu 1000 dnů, 8 dní před vypršením platnosti hesla bude uživatel upozorněn systémem na nutnost změny, po 5-ti dnech po vypršení hesla bude účet deaktivován.

Změna domovského adresáře

usermod -d /home/host2 -m host

Tímto příkazem změníme domovský adresář uživatele host na **/home/host2**. Adresář předem nevytváříme. Přenesou se do něj všechny soubory z původního domovského adresáře a původní adresář zmizí. V podstatě dojde k přejmenování domovského adresáře.

Změny se projeví ve výpisu z `/etc/passwd`.

usermod -u 1003 host

Změna UID uživatele host. Ve výpisu z `/etc/passwd` najdeme změnu:

host:x:1003:100:ucet pro hosta:/home/host2:

UID musí být unikátní, proto nemůžeme přiřadit uživateli UID, které je již použito u jiného účtu.

usermod -g 42 host

Tímto příkazem změníme primární skupinu uživatele na skupinu s GID 42.

Skupina musí existovat.

Poznámka

Výpis skupin provedeme příkazem **cat /etc/group**. Jsou zde záznamy ve formátu:

cdrom:x:24:student,hal

Vysvětlivky: název skupiny : políčko pro heslo : číslo GID skupiny : uživatelé, kteří jsou členy této skupiny a není to jejich primární skupina

Uživatel může být členem ještě dalších skupin, nejen své primární skupiny. Přiřazení do skupiny nebo skupin provedeme příkazem

usermod -G 102,103,scanner host

Tímto příkazem přiřadíme uživatele host do skupin s GID 102 a 103 a do skupiny se jménem scanner.

9.8 Založení skupiny

groupadd sk1

vytvoří skupinu sk1, bude jí přiřazeno GID alespoň 100 takové, které je větší než GID všech ostatních skupin. GID musí být unikátní. GID menší než 100 jsou určeny pro systémové účty.

groupadd -g 1008 sk2

vytvoří skupinu sk2 s číslem GID 1008, pokud ještě toto číslo není použito.

Poznámka

Výpis skupin provedeme příkazem **cat /etc/group**. Jsou zde záznamy ve formátu:

cdrom:x:24:student,hal

Vysvětlivky: název skupiny : políčko pro heslo : číslo GID skupiny : uživatelé, kteří jsou členy této skupiny a není to jejich primární skupina

9.9 Mazání skupiny

groupdel sk1

Smazání skupiny lze provést jen pokud není primární skupinou nějakého účtu. Skupina musí existovat.

9.10 Cvičení

- ❑ Založte uživatele pokus s komentářem „pokusný účet“ a domovským adresářem /home/pokus.
- ❑ Zobrazte si výpis /etc/passwd a /etc/shadow.
- ❑ Účtu nastavte heslo pokus.
- ❑ Zobrazte si výpis /etc/passwd a /etc/shadow a porovnejte s předchozím výpisem.
- ❑ Vypište status účtu.
- ❑ Zamkněte účet a vypište status.
- ❑ Odemkněte účet.
- ❑ Nastavte, aby 20 dnů před vypršením platnosti hesla si jej uživatel mohl změnit, heslo bylo platné po dobu 365 dnů, 10 dnů před vypršením platnosti hesla byl uživatel upozorněn na nutnost změny, po 2 dnech po vypršení platnosti hesla byl účet deaktivován.
- ❑ Změňte domovský adresář na pokus2.
- ❑ Zobrazte výpis z /etc/passwd a zkontrolujte.
- ❑ Provedte změnu zpět na domovský adresář /home/pokus a zkontrolujte ve výpisu z /etc/passwd.
- ❑ Vytvořte skupinu pomocna_skupina.
- ❑ Změňte primární skupinu uživatele pokus na pomocna_skupina.
- ❑ Uživateli pokus přiřadte další dvě libovolné skupiny.
- ❑ Zkuste smazat skupinu pomocna_skupina.
- ❑ Smažte uživatele pokus i s jeho domovským adresářem.
- ❑ Smažte vytvořenou skupinu pomocna_skupina.

 *Domácí úkol*

Zopakujte si zakládání uživatele, skupiny, mazání uživatele i skupiny, nastavování parametrů, změnu parametrů a naučte se, co které parametry ve výpisech znamenají.

 *Shrnutí*

- ✓ umíte založit uživatele a znáte parametry zakládání
- ✓ umíte založit skupinu a znáte parametry zakládání
- ✓ umíte změnit parametry uživatele a víte, co které parametry ve výpisech znamenají
- ✓ umíte smazat uživatele a znáte parametry mazání
- ✓ umíte smazat skupinu a znáte podmínky smazání

10. Oprávnění u souborů, adresářů (19. – 20. hodina)

10.1 Výpis oprávnění u souborů, typy souborů

Pro výpis obsahu adresáře včetně skrytých souborů můžeme použít příkaz

ls -al

```
student@linux:~/pokus$ ls -al
celkem 12
drwxr-xr-x   3 student users 4096 2005-12-18 17:43 .
drwx-----  44 student users 4096 2005-12-18 17:41 ..
drwxr-xr-x   2 student users 4096 2005-12-18 17:43 adresar
lrwxrwxrwx   1 student users   8 2005-12-03 09:44 odkaz -> original
-rw-r--r--   1 student users   0 2005-12-03 09:43 pevny_odkaz
-rwxrwxr--   1 student users   0 2005-12-18 17:42 soubor
```

Ve výpisu je poznat, který soubor je adresářem – má na začátku **d** (jako directory), který je symbolickým odkazem (má na začátku **l** – link, a ve výpisu je vidět, na který soubor odkazuje) a dále jsou zde vidět oprávnění ke čtení (**r**), zápisu (**w**), spouštění (**x**).

Oprávnění **x** u adresáře znamená možnost procházet adresářem, u souboru jde o možnost spustit soubor.

Oprávnění **rw** jsou definované pro **vlastníka**, **skupinu** a **ostatní** – tzv. svět.

10.1.1 Příklad

Například u souboru „soubor“, jehož výpis vypadá následovně:

```
-rwxrwxr-- 1 student users 0 2005-12-18 17:42 soubor
```

je vidět, že vlastníkem je uživatel student a má práva rwx, skupina je users a má práva rwx, všichni ostatní mají právo pouze ke čtení.

Datum modifikace souboru je 18.12.2005 v 17:42.

Skryté soubory začínají tečkou a ve výpisu **ls -l** nejsou vidět.

10.1.2 Oprávnění

Oprávnění r – čtení – u souboru umožňuje jeho čtení, u adresáře jde o právo čtení obsahu adresáře, ale musí být v kombinaci s právem x – procházet adresář.

Oprávnění w – zápisu – u souboru umožňuje měnit obsah souboru, avšak neumožňuje smazání vlastního souboru. Pro smazání souboru je nutné mít právo w – zápisu do adresáře, ve kterém se daný soubor nachází. Oprávnění w u adresáře umožňuje vytváření souborů a podadresářů a jejich mazání.

Oprávnění x – spouštění – u souboru znamená, že soubor je označen za spustitelný. Spustitelnost souboru se v Linuxu neřídí žádnou koncovkou, jako například .exe, .com nebo .bat – jak tomu je u Windows, ale právě tímto oprávněním. Pokud je soubor binární nebo obsahuje nějaký skript, spuštěním tohoto souboru se skript vykoná.

U adresáře znamená oprávnění x možnost procházet daným adresářem.

☞ Poznámka

Změnu vlastnictví souboru může provést jen root a změnu oprávnění u souboru může provést buď jeho vlastník nebo root.

Nastavení oprávnění pro skupinu znamená nastavení oprávnění pro všechny členy dané skupiny.

10.2 Změna vlastníka souboru

Změnu vlastnictví souboru může provést jen uživatel root.

chown uživatel soubor

Například **chown novy /home/student/pokus/soubor**

změní vlastníka souboru soubor v adresáři pokus na uživatele „novy“.

```
student@linux:~/pokus$ ls -al
celkem 12
drwxr-xr-x  3 student users 4096 2005-12-18 17:51 .
drwx----- 44 student users 4096 2005-12-18 17:41 ..
drwxr-xr-x  2 student users 4096 2005-12-18 17:43 adresar
lrwxrwxrwx  1 student users   8 2005-12-03 09:44 odkaz -> original
-rw-r--r--  1 student users   0 2005-12-03 09:43 pevny_odkaz
-rw-r--r--  1 student users   0 2005-12-18 17:51 .skryty
-rwxrwxr--  1 student users   0 2005-12-18 17:42 soubor
student@linux:~/pokus$ su -
Password:
linux:~# chown novy /home/student/pokus/soubor
linux:~# ls -al /home/student/pokus/soubor
-rwxrwxr--  1 novy users 0 Dec 18 17:42 /home/student/pokus/soubor
```

Změna vlastníka celého adresáře a jeho obsahu se provede příkazem

chown -R uživatel adresář

Například **chown -R novy /home/student/pokus/** nastaví jako vlastníka uživatele „novy“ pro celý adresář pokus **rekurzivně** – tzn. i pro celý obsah tohoto adresáře.

```
linux:~# chown -R novy /home/student/pokus/
linux:~# ls -al /home/student/pokus/
total 12
drwxr-xr-x  3 novy users 4096 Dec 18 17:51 .
drwx----- 44 student users 4096 Dec 18 17:41 ..
-rw-r--r--  1 novy users   0 Dec 18 17:51 .skryty
drwxr-xr-x  2 novy users 4096 Dec 18 17:43 adresar
lrwxrwxrwx  1 novy users   8 Dec  3 09:44 odkaz -
-rw-r--r--  1 novy users   0 Dec  3 09:43 pevny_o
-rwxrwxr--  1 novy users   0 Dec 18 17:42 soubor
```

10.2.1 Změna vlastníka i skupiny

Pomocí příkazu **chown** lze změnit jak vlastníka, tak i skupinu.

chown vlastník.skupina soubor

Pro změnu pouze skupiny se vynechá vlastník, napíše se tečka a název skupiny.

chown .skupina soubor

10.3 Změna skupiny specifikované u souboru

Každý soubor může mít právě jednoho vlastníka a právě jednu skupinu.

Změnu skupiny pro daný soubor může provést jen root.

Změna skupiny specifikované u souboru se provede příkazem

chgrp skupina soubor

10.3.1 Příklad

chgrp skupina2 /home/student/pokus/soubor

změníme skupinu pro soubor „soubor“ v adresáři pokus na „skupina2“.

```
linux:~# chgrp skupina2 /home/student/pokus/soubor
linux:~# ls -al /home/student/pokus/
total 12
drwxr-xr-x   3 student users    4096 Dec 18 17:51 .
drwx----- 44 student users    4096 Dec 18 17:41 ..
-rw-r--r--   1 student users      0 Dec 18 17:51 .skryt
drwxr-xr-x   2 student users    4096 Dec 18 17:43 adresa
lrwxrwxrwx   1 student users      8 Dec  3 09:44 odkaz
-rw-r--r--   1 student users      0 Dec  3 09:43 pevny_
-rwxrwxr--   1 student skupina2  0 Dec 18 17:42 soubor
```

10.4 Změna oprávnění pro vlastníka, skupinu a ostatní

Pro změnu oprávnění slouží příkaz **chmod**.

chmod oprávnění soubor

Oprávnění se zadává buď číselně nebo pomocí písmen.

10.4.1 Číselné vyjádření oprávnění:

- Čtení – r – hodnota 4
- Zápis – w – hodnota 2
- Spouštění – x – hodnota 1

Oprávnění se nastavují pro vlastníka, skupinu a svět. Například oprávnění **rwxrw-r--** bude číselně vyjádřeno následovně:

pro vlastníka s oprávněním **rw**x sečteme hodnoty **4+2+1=7**, pro skupinu s oprávněním **rw-** sečteme hodnoty **4+2+0=6**, pro ostatní je nastaveno oprávnění **r--**, což odpovídá číslu **4+0+0=4**.

Proto nastavíme oprávnění k souboru ve tvaru **764**, kde první číslo je oprávnění pro vlastníka, druhé číslo je oprávnění pro skupinu, třetí číslo je oprávnění pro ostatní.

Například **chmod 764 soubor** nastaví výše zmiňovaná oprávnění souboru jménem „soubor“.

```
-rwxrw-r-- 1 student skupina2 0 2005-12-18 17:42 soubor
```

10.4.2 Příklad

chmod 777 soubor

nastaví plné oprávnění vlastníkovi, skupině i ostatním.

10.4.3 Příklad

chmod 754 soubor

nastaví plné oprávnění vlastníkovi, čtení a spouštění skupině a čtení ostatním.

10.4.4 Příklad

chmod 740 soubor

nastaví plné oprávnění vlastníkovi, čtení skupině a žádné oprávnění ostatním.

10.4.5 Přiřazení oprávnění písmeny

Přidání oprávnění

chmod u+w ...přidá vlastníkovi právo zápisu, ostatní oprávnění pro vlastníka nechá beze změny.

chmod g+x ... přidá skupině právo spouštět, ostatní oprávnění pro skupinu nechá beze změny.

chmod o+r ... přidá ostatním právo čtení, ostatní oprávnění pro ostatní nechá beze změny.

Odebrání oprávnění

chmod u-w ...odebere vlastníkovi právo zápisu, ostatní oprávnění pro vlastníka nechá beze změny.

chmod g+x ... odebere skupině právo spouštět, ostatní oprávnění pro skupinu nechá beze změny.

chmod o+r ... odebere ostatním právo čtení, ostatní oprávnění pro ostatní nechá beze změny.

Poznámka

Přidávání a odebrání oprávnění lze kombinovat, oddělí se čárkou bez mezery.

10.4.6 Příklad

Před použitím příkazu bylo oprávnění nastaveno na:
-rwx--x--- 1 student skupina2 0 2005-12-18 17:42 soubor

po příkazu **chmod g+r,g-x,u-r,o+w soubor** se oprávnění nastaví na
--wxr---w- 1 student skupina2 0 2005-12-18 17:42 soubor

10.4.7 Pevné nastavení oprávnění

Lze specifikovat pomocí rovnítka =.

chmod u=rwx,g=rw,o=r soubor

-rwxrw-r-- 1 student skupina2 0 2005-12-18 17:42 soubor

chmod u=rwx,g=rx soubor

-rwxr-xr-- 1 student skupina2 0 2005-12-18 17:42 soubor

Oprávnění pro vlastníka a skupinu se nastaví na hodnoty rwx a rx, oprávnění pro ostatní zůstane beze změny.

10.5 Speciální bity (sticky bit, SGID, SUID)

10.5.1 Sticky bit

Pro **soubory** se nepoužívá.

U **adresáře** se používá v tom smyslu, že uživatel může smazat jen soubory, jejichž je vlastníkem nebo je vlastníkem adresáře, ve kterém se soubor nachází nebo je uživatel root. Využívá se například u adresáře **/tmp** – vytvořený soubor může smazat jen ten, kdo ho vytvořil nebo root.

Ve výpisu se zobrazuje jako písmeno **t** nebo **T** na pozici **x** u oprávnění pro **ostatní**.

Má hodnotu **1** a přiřazuje se příkazem **chmod**, kdy před oprávnění pro vlastníka, skupinu a ostatní napíšeme **jedničku**.

chmod 1775 adresar

drwxrwxr-t 2 student users 4096 2005-12-18 17:43 adresar

10.5.2 SGID

Nastaven na **soubor**: jestliže je soubor – program spuštěn, je GID jeho procesu nastaveno na skupinu tohoto souboru.

Nastaven na **adresář**: soubory vytvořené v tomto adresáři patří do skupiny, které patří daný adresář a ne primární skupině uživatele, který soubor vytváří. Podadresáře vytvořené v tomto adresáři dědí tento SGID bit.

Má hodnotu **2**.

Ve výpisu se zobrazuje jako **S** nebo **s** na pozici **x** pro **skupinu**.

Přiřazuje se příkazem **chmod**, kdy před oprávnění pro vlastníka, skupinu a ostatní napíšeme číslo **2**.

chmod 2777 adresar

drwxrwsrwx 2 student users 4096 2005-12-18 17:43 adresar

10.5.3 SUID bit

Nastaven na **soubor**: po spuštění tohoto souboru – programu se nastaví UID procesu na vlastníka tohoto souboru.

Na **adresář** se nenastavuje.

Má hodnotu **4**.

Ve výpisu se zobrazuje jako **S** nebo **s** na pozici **x** v oprávněních **vlastníka**.

Přiřazuje se příkazem **chmod**, kdy před oprávnění pro vlastníka, skupinu a ostatní napíšeme číslo **4**.

chmod 4777 soubor

Poznámka

Zrušení platnosti speciálních bitů provedeme opětovným nastavením oprávnění pomocí příkazu **chmod** bez čísel reprezentujících speciální bity.

Např. **chmod 777 soubor**

10.6 Cvičení

- Jako běžný uživatel vytvořte ve svém domovském adresáři adresář **test**.
- Zkontrolujte, jak se nastavila oprávnění, kdo je vlastníkem, jaká je skupina adresáře.
- Přihlaste se jako **root**.
- Vytvořte skupinu **sk1**.
- Změňte skupinu adresáře **test** na **sk1**.
- Odhlaste se a jako původní uživatel v adresáři **test** vytvořte soubor **soubor1**.
- Zkontrolujte, jak jsou nastavena práva, kdo je vlastníkem a jaká je skupina souboru **soubor1**.
- Nastavte **SGID** adresáři **test**.
- V adresáři **test** vytvořte další soubor **soubor2**.
- Zkontrolujte, jak jsou nastavena práva, kdo je vlastníkem a jaká je skupina souboru **soubor2**.
- Odstraňte **SGID** z adresáře **test** (**chmod** bez čísla **2** před oprávněními pro uživatele, skupinu a ostatní).
- V adresáři **test** vytvořte soubor **soubor3**.
- Zkontrolujte, jak jsou nastavena práva, kdo je vlastníkem a jaká je skupina souboru **soubor3**.
- Smažte adresář **test**.
- Smažte skupinu **sk1** – jako **root**.

 *Domácí úkol*

- Vytvořte ve svém domovském adresáři podadresář pokus.
- Vypište oprávnění u tohoto adresáře, zjistěte vlastníka a skupinu.
- Nastavte u adresáře pokus oprávnění na `rxw-----`.
- Přidejte oprávnění skupině na `rw-`.
- Přidejte oprávnění ostatním na `r--`.
- Jako root vytvořte skupinu pokus.
- Vytvořte jako root v adresáři pokus soubor test.
- Odhlaste se a jako původní uživatel vytvořte v adresáři pokus soubor test2.
- Vypište podrobně obsah adresáře pokus a zjistěte vlastníky, skupiny a oprávnění u souborů.
- Nastavte SGID adresáře pokus.
- Jako root změňte skupinu u adresáře pokus na skupinu pokus.
- Vytvořte jako root v adresáři pokus soubor test3.
- Odhlaste se a jako původní uživatel vytvořte v adresáři pokus soubor test4.
- Vypište podrobně obsah adresáře pokus a zjistěte vlastníky, skupiny a oprávnění u všech souborů.
- Udělejte závěr ohledně SGID.
- Smažte adresář pokus a skupinu pokus.

 Shrnutí

- ✓ Umíte vypsát oprávnění, vlastníka a skupinu u souboru.
- ✓ Umíte změnit vlastníka, skupinu a oprávnění souboru.
- ✓ Víte, co které oprávnění znamená.
- ✓ Znáte účel speciálních bitů – sticky bit, SGID bit, SUID bit.

11. Uživatelé, oprávnění – opakování, další příkazy

(21. - 22. hodina)

11.1 Výpis přihlášených uživatelů

Pro výpis přihlášených uživatelů k systému na všech konzolách slouží příkazy **finger, who, w, users**. Zobrazují víceméně stejné informace (viz obrázek).

```
linux:~# users
root student
linux:~# w
 17:21:01 up 4:30, 2 users, load average: 0.22, 0.08,
USER      TTY      FROM          LOGIN@      IDLE        JCPU
root      tty1    -             17:20      7.00s      0.03s
student   :0      -             12:51     ?xdm?     11:11
linux:~# who
root      tty1          Dec 18 17:20
student   :0           Dec 18 12:51
linux:~# finger
Login      Name      Tty      Idle      Login Time  Office
root      root      *tty1    -         Dec 18 17:20
student   student   *:0      -         Dec 18 12:51
```

V našem příkladu se na konzoli 1 přihlásil uživatel root, v grafické konzoli 7 se v konzolovém programu přihlásil uživatel student.

11.2 Výpis skupin, ve kterých je uživatel členem

Příkaz **groups** vypíše skupiny, ve kterých je přihlášený uživatel členem.

groups

Pro zjištění členství ve skupinách jiného uživatele použijeme stejný příkaz následovaný jménem uživatele.

groups student

```
linux:~# groups
root
linux:~# groups student
student : users dialout cdrom floppy audio video plugdev
linux:~# █
```

11.3 Zjištění ID, GID a ostatních skupin uživatele

Příkazem **id** zjistíme informace o ID uživatele, GID jeho primární skupiny a členství v dalších skupinách.

id

Pro zobrazení těchto informací o jiném uživateli použijeme stejný příkaz následovaný jménem uživatele.

id student

```
linux:~# id
uid=0(root) gid=0(root) groups=0(root)
linux:~# id student
uid=1000(student) gid=100(users) groups=100(users), 20(dialout), 24(cdrom), 25(floppy), 29(audio), 44(video), 46(plugdev)
```

11.4 Výpis všech uživatelů v systému

Příkazem **cut** můžeme ze souboru zobrazit zvolené sloupce, v tomto případě se bude hodit zobrazit první sloupec ze souboru **/etc/passwd**.

cut -f1 -d: /etc/passwd

případně seřazeně

cut -f1 -d: /etc/passwd | sort

f1 znamená první sloupec (fields), **d** je oddělovač (delimiter) – v našem případě je v **/etc/passwd** oddělovačem dvojtečka, proto je za **d** uvedena dvojtečka.

11.5 Cvičení

- ❑ Přihlaste se jako root.
- ❑ Vytvořte uživatele host1 s domovským adresářem /home/host1.
- ❑ Přiřadte účtu heslo host1.
- ❑ Zkontrolujte zápis parametrů v souborech /etc/passwd a /etc/shadow.
- ❑ Jak je vyjádřeno datum poslední změny účtu?
- ❑ Kolik dní před vypršením platnosti hesla bude uživatel vyzván ke změně hesla?
- ❑ Kolik dní před vypršením platnosti hesla si jej může uživatel změnit?
- ❑ Kolik dní je ještě heslo použitelné, ačkoli jeho platnost již vypršela?
- ❑ Do jaké primární skupiny účet host1 patří?
- ❑ Do jakých dalších skupin tento účet patří?
- ❑ K účtu přidejte komentář „zkušební účet“.
- ❑ Vytvořte skupinu sk_host.
- ❑ Změňte primární skupinu účtu host1 na sk_host.

- ❑ Vytvořte skupiny sk_1, sk_2, sk_3.
- ❑ Přidejte účet host1 do těchto nových skupin.
- ❑ Přihlaste se na účet host1.
- ❑ Vytvořte v jeho domovském adresáři adresář test.
- ❑ V adresáři test vytvořte soubor test.txt.
- ❑ Vypište oprávnění u adresáře test a souboru test.txt.
- ❑ Změňte oprávnění u souboru test.txt na rwxr-xr--.
- ❑ Odhlaste se a jako root změňte domovský adresář uživatele host1 na /home/host.
- ❑ Zkontrolujte výpisem obsah tohoto adresáře.
- ❑ Zamkněte účet host1.
- ❑ Zkuste se na něj přihlásit.
- ❑ Odemkněte účet.
- ❑ Změňte skupinu u souboru test.txt na „users“.
- ❑ Nastavte pro účet host1 platnost hesla po dobu 300 dnů, 20 dnů před vypršením platnosti hesla bude uživatel upozorňován na nutnost změny.
- ❑ Vypište status účtu host1.
- ❑ Změňte UID uživatele host1 na nejbližší vyšší volné UID.
- ❑ Vypište status účtu host1.
- ❑ Nastavte SGID bit na adresář test.
- ❑ Jako uživatel host1 v tomto adresáři vytvořte soubor test2.txt.
- ❑ Výpisem zkontrolujte nastavení oprávnění, vlastníka a skupinu.
- ❑ Vytvořte v adresáři test podadresář test2.
- ❑ Výpisem zkontrolujte nastavení oprávnění, vlastníka a skupinu.
- ❑ Změňte primární skupinu uživatele host1 na users.
- ❑ Změňte rekurzivně vlastnictví adresáře test na účet host1 a skupinu users.
- ❑ Jako uživatel test změňte oprávnění u souboru test2.txt tak, aby vlastník mohl číst, měnit i spouštět soubor, skupina mohla číst, ostatní nemohli nic.
- ❑ Dále přidejte skupině oprávnění zápisu.
- ❑ Uberte vlastníkovu právo spouštět.
- ❑ Povolte ostatním číst tento soubor.
- ❑ K čemu slouží sticky bit, SGID bit, SUID bit?
- ❑ Jak se nastavují? Jak se odebírají?
- ❑ Odstraňte účet host1 i s domovským adresářem.
- ❑ Odstraňte skupiny sk_host, sk_1, sk_2 a sk_3.

 *Domácí úkol*

- Zopakujte si vytváření účtů a skupin.
- Zopakujte si změny parametrů u uživatelských účtů.
- Zopakujte si, jak smazat účet a skupinu.
- Zopakujte si nastavování oprávnění souborům, změnu vlastníka, skupiny.
- K čemu slouží speciální bity u oprávnění, jak se nastavují a jak se ruší?
- Jak se zjistí přihlášení uživatelé?
- K čemu slouží příkazy `id` a `groups`?
- Jak vypíšeme první sloupec ze souboru `/etc/passwd`?

 *Shrnutí*

- ✓ Zopakovali jste si vytváření uživatelských účtů a skupin.
- ✓ Umíte změnit parametry uživatelských účtů.
- ✓ Víte, jak a kdy mazat účty a skupiny.
- ✓ Umíte vysvětlit systém oprávnění u souborů.
- ✓ Umíte vysvětlit význam speciálních bitů u oprávnění souborů.
- ✓ Umíte změnit oprávnění u souborů pro vlastníka, skupinu a svět.
- ✓ Umíte používat příkazy `id`, `groups`, `cut`.

12. Umask, kvóty pro uživatele a skupinu (23. - 24. hodina)

12.1 Umask

Pokud není nastaveno jinak, vytváří se soubory a adresáře s výchozími oprávněními, např. soubory mají nejvyšším oprávnění **666** – rw-rw-rw- (u souboru není přednastaveno automatické přidělování oprávnění spouštět, z bezpečnostních důvodů, toto oprávnění musíme přidělit úmyslně), adresáře mají oprávnění **777** – rwxrwxrwx.

Aby tomu tak nebylo a nemuseli jsme měnit dodatečně oprávnění u souborů a adresářů, lze pomocí umask změnit toto výchozí nastavení.

umask odebírané oprávnění

Například **umask 011** – odebere od přednastaveného oprávnění oprávnění 011 (právo x pro skupinu a svět), takže pokud bylo výchozí oprávnění adresáře 777, bude nové oprávnění 766, takže skupina a svět nebude mít právo x – procházet adresářem. U souboru by se odebralo právo spouštět pro skupinu a svět.

Týká se to nově vznikajících souborů a adresářů. Ty budou vznikat s právy odpovídajícími nejvyššímu původnímu nastavení bez oprávnění definovaného pomocí umask. Toto platí po dobu přihlášení uživatele.

V **/etc/profile** je definována hodnota umask, která se používá, pokud neřekneme příkazem umask jinak.

Pokud je v tomto souboru nastaveno **umask 022**, pak nově vznikající soubory a adresáře **nebudou** mít povoleno právo zápisu pro skupinu a svět, navíc soubory se budou vytvářet bez práva x – spouštět, což vychází z výchozího nastavení nejvyššího oprávnění pro soubory 666 (kde chybí právo spouštět pro kohokoliv).

12.1.1 Příklad

Při výchozím nastavení oprávnění 777 u adresářů a 666 u souborů, má nastavení **umask 013** za následek u adresáře odebrání práva **x** – procházet adresářem - pro skupinu (skupině zůstane $rw=4+2$) a práva **w** a **x** ($2+1=3$) pro ostatní (zůstane $r=4$), vlastníkově se tady nic neodebírá, tedy výsledné oprávnění bude 764 pro adresář.

Pro soubor bude výsledek aplikování **umask 013** ten, že skupině by se teoreticky odebíralo právo **x** (1) – spouštět soubor – ale to při výchozím nastavení $6=4+2=r+w$ stejně neměla, takže pro skupinu to nebude znamenat žádnou změnu (zůstane jí právo $rw=4+2$). Pro svět se bude odebírat právo $3=2+1=w+x$ – zápis a spouštění, ale vliv bude mít jen odebírané právo zápisu, protože spouštění nebylo povoleno (světu zůstane právo $r=4$).

Výsledné oprávnění u souboru s výchozím nastavení oprávnění 666 bude po aplikaci umask 013 – **664**.

☞ Poznámka

Výpis aktuálního nastavení umask provedeme příkazem **umask**. Pokud chceme nastavit oprávnění na 777 u adresářů a 666 u souborů, zadáme příkaz **umask 000**. Pak nově vznikající soubory a adresáře budou mít tato oprávnění (777 a 666).

12.1.2 Příklad

Výpis adresáře po nastavení umask na 000.
Nechali jsme vzniknout soubor a adresář (viz výpis).

```
drwxrwxrwx 2 horakova users 4096 2005-12-20 16:49 adr
-rw-rw-rw- 1 horakova users 0 2005-12-20 16:48 soub
```

Nastavení umask platí během přihlášení. Pokud změníme umask, tak budeme během daného přihlášení vytvářet adresáře a soubory s oprávněními sníženými o hodnotu definovanou příkazem umask.

Po odhlášení a novém přihlášení bude opět platit původní hodnota (/etc/profile).

12.2 Nastavení kvóty pro účet

Uživatelům lze nastavit omezení zabraného prostoru na souborovém systému, dále lze nastavit maximální počet souborů, které může vytvořit. Totéž platí i pro skupinu.

Kroky, které je nutné udělat, aby bylo možno nastavit kvóty.

- ❑ připravit souborový systém
- ❑ inicializovat systém kvót
- ❑ nastavit kvóty uživatelům a skupinám
- ❑ spustit službu zabezpečující hlídání kvót

12.2.1 Příprava filesystemu

Kvóty lze nastavovat na systémech typu ext2, ext3, reiserfs.

Postup:

- Do /etc/fstab napíšeme do sloupce, kde jsou optiony - volby **usrquota** a **grpquota** - aby šly nastavovat kvóty pro uživatele i skupiny.
- Například:
/dev/sda6 / ext3 defaults,errors=remount-ro,usrquota,grpquota 0 1
- rebootovat server nebo jej znovu připojit příkazem **mount -o remount**

12.2.2 Inicializace systému kvót

- nainstalujeme balíček **quota**
- inicializovat quota systém příkazem **quotacheck** tento příkaz prohledá systém na slovo quota a zjištěné hodnoty uloží do souboru **aquota.user** a **aquota.group** (vytvoří se v rootu /).
- Výpis – příkazem **quotacheck -avug** ... všechny (a) připojené systémy jsou prohledány a kontrolovány na datové bloky a inody (kontrola velikosti a počtu souborů) jednotlivých uživatelů (u) a skupin (g), detailní výpis (v).
- zobrazení nastavení kvóty pro daného uživatele:
quota -vu host
- **quotaon -av** zapnutí kvót na připojených file systémech
- **repquota -av**výpis quota reportu

12.2.3 Nastavení kvóty uživateli

Kvóta se nastavuje na počet zabraných bloků nebo na počet souborů (inodů).

Pro zjištění velikosti bloku můžeme použít příkaz **dumpe2fs /dev/sdaX**, ve kterém vyhledáme informaci typu

Block size: 4096.

Název souborového systému se může podle skutečnosti lišit (sda1, sda6, hda1 apod.)

Pokud je velikost bloku **4096 B** a chceme uživateli povolit maximální zabraný prostor **10 MB**, musíme zjistit odpovídající počet bloků (10 MB = 10485760B, 10485760B / 4096B = 2560 bloků).

- Pro nastavení kvóty uživateli test zadáme příkaz **edquota -u test**
Zobrazí se textový editor, ve kterém nastavíme soft limit a hard limit (soft limit může být krátkodobě překročen).
Čísla jsou počet bloků.

Poznámka

Když chceme v našem případě nastavit soft limit na 10MB = 2560 bloků, a hard limit na například 3000 bloků, napíšeme tyto hodnoty do souboru a uložíme.

Stejně tak můžeme nastavit soft limit a hard limit pro počet inodů (počet souborů). Ostatní údaje needitujeme.

12.2.4 Příklad

Nastavíme uživateli host soft limit 17 bloků a hard limit 20 bloků.

Ve výpisu **repquota -av** vidíme, kolik bloků uživatel aktuálně zabírá a jak jsou nastaveny limity.

```
User      used  soft  hard grace  used  soft  hard grace
host     --   16   17   20         4    0    0
```

Aktuálně zabírá **16** bloků.

Jako uživatel **host** vytvořme v jeho domovském adresáři soubor **yes.txt**, do kterého funkcí **yes** plňme písmena **y**.

yes > yes.txt

Jakmile převýšíme **hard limit** 20 bloků, zapisování se přeruší.

Ve výpisu **repquota -av** vidíme změnu:

```
User      used  soft  hard grace  used  soft  hard grace
host     +-   20   17   20  7days    5    0    0
```

Dokud nesnížíme počet zabraných bloků pod **hard limit**, nebudeme moci jako uživatel **host** nic vytvořit.

Hodnota **grace** znamená počet dnů, během kterých můžeme překračovat **soft limit**.

12.2.5 Cvičení

Vyzkoušejte nastavit **soft limit** a **hard limit** na počet inodů u uživatele **host**, vypněte limity zabraného místa – limit na počet bloků. Zkuste jako daný uživatel vytvořit ve své domovské složce další soubory a překročit limit počtu souborů.

Jako **root** zjistěte pomocí **repquota** aktuální stav.

12.3 Kopírování kvóty z účtu na účet

Nastavení kvóty pro uživatele se dá kopírovat na další vybrané účty - edquota s přepínačem p .

edquota -p zdrojový_účet -u další_účet další_účet2 ...

12.3.1 Příklad

Na účtu host jsou nastaveny kvóty, na účtu host2 zatím ne.

Ve výpisu repquota vidíme následující:

User		used	soft	hard	grace	used	soft	hard	grace
host	--	16	17	20		4	0	0	
host2	--	12	0	0		3	0	0	

Příkazem **edquota -p host -u host2** zkopírujeme nastavení kvót z účtu host na účet host2.

User		used	soft	hard	grace	used	soft	hard	grace
host	--	16	17	20		4	0	0	
host2	--	12	17	20		3	0	0	

12.4 Kvóta pro skupinu

Obdobně jako pro uživatele jde nastavit kvóta na celou skupinu, například **edquota -g users**.

Když bude mít skupina jako celek zaplněno, tak už nikdo ze skupiny nic nevytvoří, neuloží, i když by jako jednotlivec ještě měl dost místa.

Výpis kvót i se skupinami ... **repquota -avg**.

12.4.1 Cvičení

Vytvořte skupinu test_gr a uživatele host1, host2, host3.

Nastavte uživatelům tuto skupinu jako jejich primární skupinu.

Nastavte kvóty pro skupinu (bloky i inody).

Jako jednotliví uživatelé této skupiny vytvářejte adresáře a soubory v jejich domovských adresářích. Vyzkoušejte překročení limitů a zjistěte chování systému v případě překročení.

Vypište si příkazem repquota aktuální stav.

Odstraňte nastavení limitů.

Smažte skupinu i nové uživatele.

 *Domácí úkol*

Procvičte si práci s nastavováním umask.
Vyzkoušejte nastavení kvót na systému pro uživatele a skupinu.

 Shrnutí

- ✓ Umíte vysvětlit účel nastavení umask.
- ✓ Víte, kde je uložena výchozí hodnota umask.
- ✓ Umíte na souborovém systému zapnout podporu kvót.
- ✓ Umíte doinstalovat potřebné balíčky.
- ✓ Umíte zobrazit stav kvót na systému.
- ✓ Umíte přiřadit uživateli kvótu na počet souborů a velikost zabraného místa.
- ✓ Umíte kopírovat nastavení kvót z jednoho účtu na další.
- ✓ Umíte přiřadit kvóty pro celou skupinu.
- ✓ Umíte vysvětlit pojmy soft limit, hard limit, grace.

13. Sledování systému (25. – 26. hodina)

13.1 HW informace

13.1.1 Hwinfo

Doinstalujte v případě potřeby balíček **hwinfo**.

☞ Postup instalace

Jako root spustit **aptitude**. Pomocí lomítka následovaného hledaným balíčkem /hwinfo vyhledat balíček. Balíček zvolíme k instalaci klávesou +. Souhlas s instalováním provedeme dvojitým stisknutím klávesy g (gg).

Postupujeme dle zobrazovaných informací. Aptitude ukončíme stisknutím klávesy q.

Pak lze spustit příkaz **hwinfo**.

System se zanalyzuje a výsledné informace se zobrazí na obrazovce. Dlouhý výpis můžeme přesměrovat do souboru.

hwinfo > /home/student/vypis_hw.txt

Pro výpis informací jen o některém zařízení můžeme specifikovat požadavek, například **hwinfo --mouse** , **hwinfo --network interface** apod.

13.1.2 Výpis zařízení

Výpis zařízení použitých na systému Linux – vypíšeme obsah souboru **/proc/devices**.

cat /proc/devices | less

13.1.3 Výpis informací o procesoru

Informace o procesoru – vypíšeme obsah souboru **cpuinfo** v adresáři /proc

cat /proc/cpuinfo

13.1.4 Výpis informací o I/O portech

Informace o I/O portech – vypíšeme obsah souboru **ioports** v adresáři /proc

cat /proc/ioports

13.1.5 Výpis informací o IRQ přerušeních

Informace o IRQ přerušeních – vypíšeme obsah souboru **interrupts** v adresáři /proc

cat /proc/interrupts

13.1.6 Výpis informací o DMA kanálech

Informace o DMA kanálech – vypíšeme obsah souboru **dma** v adresáři /proc

cat /proc/dma

13.1.7 Výpis informací o PCI zařízeních

Informace o PCI zařízeních – vypíšeme obsah souboru **devices** v adresáři /proc/bus/pci/

cat /proc/bus/pci/devices

nebo zadáním příkazu **lspci**

13.1.8 Výpis SCSI informací

Tyto informace lze najít v souborech v adresáři /proc/scsi, souhrnné informace jsou v souboru /proc/scsi/scsi.

cat /proc/scsi/scsi

13.1.9 Výpis USB zařízení

USB zařízení vypíšeme příkazem **lsusb** nebo lze zobrazit detailní informace výpisem souboru **devices** a **drivers** v adresáři /proc/bus/usb.

cat /proc/bus/usb/devices

cat /proc/bus/usb/drivers

Soubor drivers neexistuje, pokud nejsou na počítači žádné USB ovladače.

13.1.10 Výpis informací o geometrii disku

Informace o geometrii disku (počet hlav, cylindrů apod.) můžeme vypsat příkazem

fdisk -l /dev/hda

(pokud se jedná o hda, jinak např. **fdisk -l /dev/sda**)

Disk /dev/hda: 10.0 GB, 10056130560 bytes
 255 heads, 63 sectors/track, 1222 cylinders
 Units = cylinders of 16065 * 512 = 8225280 bytes

Device	Boot	Start	End	Blocks	Id	System
/dev/hda1	*	1	791	6353676	7	HPFS/NTFS
/dev/hda2		792	1156	2931862+	83	Linux
/dev/hda3		1157	1222	530145	82	Linux swap / Solaris

13.1.11 Informace o nastavení hard disku

Informace o nastavení parametrů HDD vypíšeme příkazem

hdparm /dev/hda

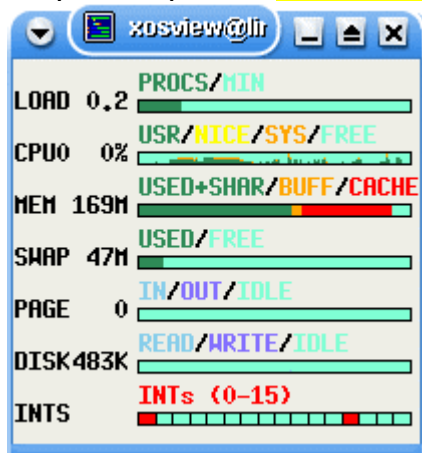
```
linux:/proc# hdparm /dev/hda
/dev/hda:
multcount    = 0 (off)
IO_support   = 1 (32-bit)
unmaskirq    = 1 (on)
using_dma    = 1 (on)
keepsettings = 0 (off)
readonly     = 0 (off)
readahead    = 256 (on)
geometry     = 19485/16/63, sectors = 19640880, start = 0
```

Pokud příkaz není přístupný, bude jej třeba nainstalovat (instalujte balíček **hdparm**).

13.2 Grafické nástroje

13.2.1 XOsview

Podle potřeby nainstalujte balíček xosview a následně z příkazového řádku spusťte příkaz **xosview &**.



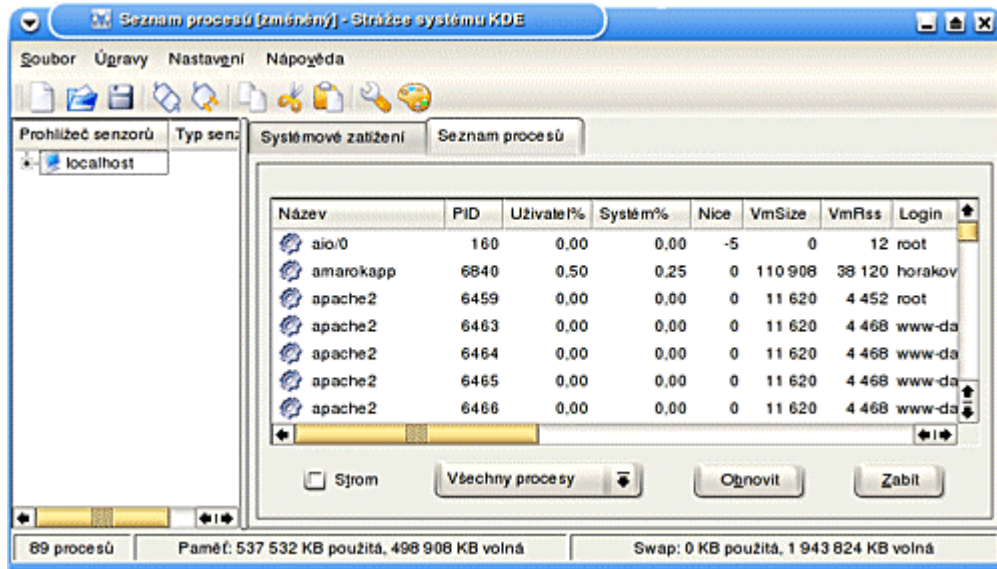
Zobrazí se informace o využití prostředků počítače.

13.2.2 Strážce systému

Další možností je **Strážce systému KDE**, kde si opět můžeme zobrazit běžící procesy a systémové zatížení.

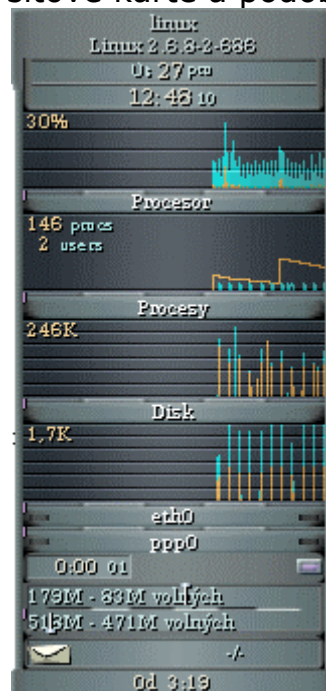
Najdeme jej v menu – **hlavní nabídka KDE – Systém – Kontrola systému pro prostředí KDE**.

Lze zde přidat nové senzory (vytvoří se další karta – Soubor – Nový – na kterou z levého seznamu přetáhneme myší položku, kterou chceme sledovat).



13.2.3 Gkrellm

Gkrellm – příkaz pro sledování systému a zobrazování informací o procesoru, procesech, disku, síťové kartě a podobně.



13.3 Logovací soubory

Příkazem **dmesg** vypíšeme hlášení probíhající při startu systému.

V adresáři **/var/log** jsou mimo jiné uloženy soubory monitorující systém (lastlog, faillog, messages, mail.err, syslog, boot).

13.4 Příkazy uname, uptime, netstat, top, ps

13.4.1 Příkaz uname

Příkaz `uname` vypisuje systémové informace.

uname

bez parametrů vypíše jméno jádra

uname -a

vypíše všechny informace (jméno jádra, síťové jméno, verzi jádra, hardwarové jméno, operační systém)

```
linux:/proc/bus/usb/001# uname -a
```

```
Linux linux 2.6.8-2-686 #1 Thu May 19 17:53:30 JST 2005 i686 GNU/Linux
```

13.4.2 Příkaz `uptime`

Příkaz **uptime** vypíše aktuální čas, jak dlouho systém běží, kolik je aktuálně přihlášeno uživatelů a průměrnou zátěž systému během posledních 1, 5 a 15 minut.

```
linux:/proc/bus/usb/001# uptime
```

```
10:53:01 up 1:23, 2 users, load average: 0.07, 0.11, 0.09
```

13.4.3 Příkaz `netstat`

Příkaz **netstat** vypíše síťová spojení, směrovací tabulku, statistiky síťového rozhraní, členství v multicastových skupinách.

Bez parametrů vypíše všechna **otevřená spojení**.

```
linux:/proc/bus/usb/001# netstat
```

Active Internet connections (w/o servers)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	localhost.localdo:32770	localhost.localdo:32768	ESTABLISHED
tcp	0	0	localhost.localdo:32768	localhost.localdo:32770	ESTABLISHED
tcp	0	0	192.168.0.2:33720	37.113.broadband.i:smtp	TIME_WAIT

Výpis směrovací tabulky:

```
linux:/proc/bus/usb/001# netstat -r
```

Kernel IP routing table

Destination	Gateway	Genmask	Flags	MSS	Window	irtt	Iface
192.168.0.0	*	255.255.255.0	U	0 0	0		eth0
default	192.168.0.1	0.0.0.0	UG	0 0	0		eth0

Výpis všech síťových rozhraní

```
linux:/proc/bus/usb/001# netstat -i
```

Kernel Interface table

Iface	MTU	Met	RX-OK	RX-ERR	RX-DRP	RX-OVR	TX-OK	TX-ERR	TX-DRP	TX-OVR	Flg
eth0	1500	0	12247	0	0	0	16100	0	0	0	BMRU
lo	16436	0	127	0	0	0	127	0	0	0	LRU

Výpis souhrnných statistik pro každý protokol

```
linux:/proc/bus/usb/001# netstat -s
```

Ip:

```
12619 total packets received
0 forwarded
0 incoming packets discarded
12619 incoming packets delivered
16423 requests sent out
```

Icmp:

```

3 ICMP messages received
0 input ICMP message failed.
ICMP input histogram:
  echo requests: 3
3 ICMP messages sent
0 ICMP messages failed
ICMP output histogram:
  echo replies: 3
    
```

Tcp:

```

1002 active connections openings
19 passive connection openings
0 failed connection attempts
4 connection resets received
4 connections established
11452 segments received
15305 segments send out
33 segments retransmited
0 bad segments received.
0 resets sent
    
```

Udp:

```

1164 packets received
0 packets to unknown port received.
0 packet receive errors
1115 packets sent
    
```

13.4.4 Příklad top

Příkazem **top** vypíšeme běžící procesy, průměrné zatížení systému během poslední 1, 5 a 15 minut, zatížení procesoru, paměti, využití odkládacího prostoru apod.

```
linux:~# top
```

```

top - 11:06:46 up 1:37, 2 users, load average: 0.17, 0.12, 0.09
Tasks: 90 total, 3 running, 87 sleeping, 0 stopped, 0 zombie
Cpu(s): 10.3% us, 1.0% sy, 0.0% ni, 84.8% id, 4.0% wa, 0.0% hi, 0.0% si
Mem: 183784k total, 179880k used, 3904k free, 7468k buffers
Swap: 530136k total, 44760k used, 485376k free, 78188k cached
    
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2452	root	15	0	27572	13m	13m	S	5.3	7.6	3:21.82	XFree86
4089	student	16	0	27464	16m	23m	S	4.0	9.1	2:59.35	ksysguard
4090	student	16	0	2160	1012	1868	S	1.3	0.6	0:40.23	ksysguardd
2126	root	16	0	1648	708	1472	S	0.3	0.4	0:00.87	syslogd
3365	student	15	0	28028	13m	24m	S	0.3	7.4	0:20.82	kdeinit
3436	student	16	0	14620	1788	5824	S	0.3	1.0	0:00.96	sshd
6938	root	16	0	2064	1056	1852	R	0.3	0.6	0:00.33	top
1	root	16	0	1504	492	1352	S	0.0	0.3	0:00.63	init

13.4.5 Příklad ps a pstree

Pro výpis procesů můžeme využít příkazy **ps** a **pstree**.
Podrobný popis v kap. 8.

ps aux, pstree

13.5 Aptitude (instalování, odstraňování, update, výpis nainstalovaných balíčků, načtení zdrojů)

13.5.1 Výpis zdrojů

V souboru **/etc/apt/sources.list** je seznam všech zdrojů, ze kterých se bude čerpat při instalaci balíčků.

Výpis souboru může vypadat například takto:

```
deb cdrom:[Debian GNU/Linux 3.1 r0a _Sarge_ - Official i386 Binary-2 (20050607)]/
unstable contrib main
deb cdrom:[Debian GNU/Linux 3.1 r0a _Sarge_ - Official i386 Binary-1 (20050607)]/
unstable contrib main
deb http://ftp.cz.debian.org/debian/ stable main non-free contrib
deb-src http://ftp.cz.debian.org/debian/ stable main non-free contrib
deb http://ftp.sk.debian.org/debian-volatile stable/volatile main
deb http://www.axis.cz/linux/debian stable axis
deb http://security.debian.org/ stable/updates main contrib non-free
deb http://debian.sh.cvut.cz/debian/ stable main non-free contrib
deb-src http://debian.sh.cvut.cz/debian/ stable main non-free contrib
```

Pokud se v seznamu vyskytují položky s # na začátku, jsou to pouze komentáře a proto se neakceptují.

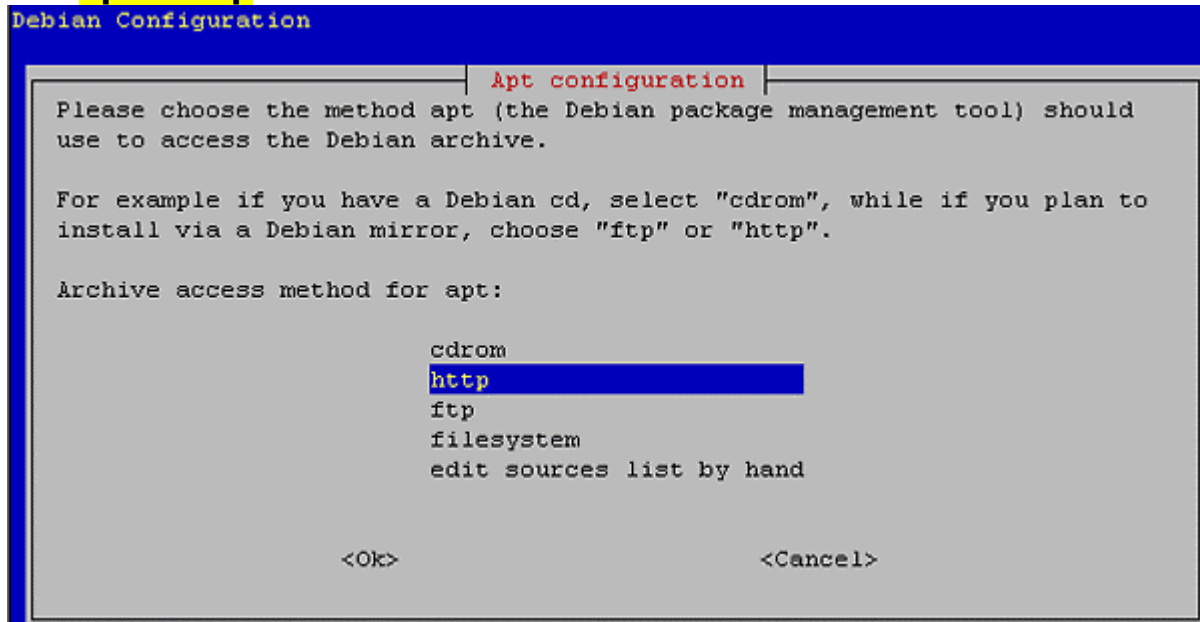
```
#deb cdrom:[Debian GNU/Linux 3.1 r0a _Sarge_ - Official i386 Binary-13 (20050607)]/
unstable contrib main
#deb cdrom:[Debian GNU/Linux 3.1 r0a _Sarge_ - Official i386 Binary-12 (20050607)]/
unstable contrib main
```

Tímto způsobem můžeme také soubor **sources.list** ručně upravit. Aby se některé zdroje dočasně ignorovaly, zakomentujeme je pomocí #.

13.5.2 Načtení zdrojů

Pokud chceme zdroje zaktualizovat, spustíme příkaz **apt-setup** pro konfiguraci Apt.

apt-setup



Po zadání příkazu **apt-setup** se zobrazí dialog, ve kterém můžeme vybrat metodu, pomocí které budeme přistupovat ke zdrojům.

Máme-li například k dispozici CD nebo DVD s instalací Debianu, vybereme položku **cdrom** a necháme program načíst seznam všech balíčků, které jsou k dispozici na daném médiu.

Posléze můžeme doplnit další zdroje, které jsou například k dispozici přes **http** nebo **ftp**.

13.5.3 Výpis nainstalovaných balíčků

dpkg -l **dpkg --get-selections**

Výpis balíčků a jejich stavů se nevejde obvykle na obrazovku, proto jej přesměrujeme do souboru

dpkg --get-selections > vypis

Tento soubor s výpisem balíčků lze přenést na jiný počítač a výběr balíčků na tomto počítači provést pomocí tohoto souboru příkazem

dpkg --set-selections < vypis

Tím se balíčky nenainstalují ani neodinstalují, jen se označí a instalaci provedeme například pomocí **aptitude**.

13.5.4 Instalace balíčku pomocí aptitude

Jako root spustíme **aptitude**.

Vyhledáme balíček. Napíšeme **/** a název hledaného balíčku – například **/ksysguard**. Pro potvrzení vyhledání stiskneme **Enter**.

Pokud se nevyhledal správný balíček, stiskneme písmeno **n** (next) a tím se přesuneme na další řádek, který odpovídá hledanému balíčku.

Chceme-li, aby se vyhledal balíček, který začíná určitými písmeny, napíšeme před hledaný řetězec znak **^**.

Chceme-li, aby se vyhledal balíček, který končí určitými písmeny, napíšeme za hledaný řetězec znak **\$**.

Chceme-li, aby se vyhledal balíček, jehož název je přesně roven určitému řetězci, napíšeme před hledaný řetězec znak **^** a na konec znak **\$**.

Stojíme-li na správném řádku, napíšeme plus **+**. Tím se balíček označí k instalaci.

```

student@linux: /home/student
Akce Zpět Balík Hledat Volby Pohledy Nápověda
f10: Menu ? : Nápověda q: Konec u Aktualizace g Stažení/Instalace
aptitude 0.2.15.9 Na disku se použije 1507kB Stáhnou: 529kB
p1 ksysguard +1405kB <žádná> 4:3.3.2-1s
p1^ ksysguardd +102kB <žádná> 4:3.3.2-1s
p ktimer <žádná> 4:3.3.2-1
p kudzu <žádná> 1.1.67-1
p lam-runtime <žádná> 7.1.1-3
p laptop-mode-tools <žádná> 1.05-1
p lcab <žádná> 1.0b10-2
p lcdproc <žádná> 0.4.5-1
p lde <žádná> 2.6.0-6
p leave <žádná> 1.12-1
LCD display driver daemon and clients
This is a client/server suite including drivers for all kinds of nifty LCD
displays. The server supports several serial devices: Matrix Orbital, Crystal
Fontz, Bayrad, LB216, LCDM001 (kernelconcepts.de), Wirz-SLI and PIC-an-LCD; and
some devices connected to the LPT port: HD44780, STV5730, T6963, SED1520 and
SED1330. As of 0.4.4 some USB displays are also supported by the drivers
CFonz633, CwLnx, and USBLCD.

Various clients are available that display things like CPU load, system load,
memory usage, uptime, and a lot more. Custom clients can be written using the
simple client-server protocol and provided example code.
    
```

Automaticky se vyberou i balíčky, které souvisejí s instalací vybraného balíčku.

Pak stiskneme **g**, zobrazí se informace o tom, co se bude instalovat, kolik dat se stáhne a podobně. Instalaci spustíme dalším stisknutím **g**.

13.5.5 Aktualizace

Spustíme **aptitude**.

Pro aktualizaci seznamu všech dostupných balíčků stiskneme písmeno **u** (update). Ze zadaných zdrojů (v sources.list) se zjistí dostupné aktualizace.

Stisknutím **U** nastavíme všem aktualizovatelným balíčkům příznak, že se mají zaktualizovat a po stisknutí **g** se vypíší balíčky, které se budou aktualizovat včetně velikosti stahovaných dat. Po dalším stisknutí písmene **g** se aktualizace balíčků spustí.

```

student@linux: /home/student
Akce Zpět Balík Hledat Volby Pohledy Nápověda
f10: Menu ?: Nápověda q: Konec u Aktualizace g Stažení/Instalace
http://security.debian.org stable/updates/main ksysguardd 4:3.3.2-1sarge1 [Staže
http://security.debian.org stable/updates/main ksysguard 4:3.3.2-1sarge1 [Staže

```

Staženo 529kB za 7s (74,2kB/s).

[**g** Pokračovat] [Zrušit]

```

Celkový postup: [ 100% ] (72,8kB/s, 0s zbývá)
    
```

Pro ukončení aptitude stiskneme písmeno **q**.

13.5.6 Odinstalování balíčku

Spustíme aptitude, vyhledáme balíček a stiskneme klávesu minus **-**.

Pak stiskneme klávesu **g** – zobrazí se balíčky, které budou odstraněny (díky řešení závislostí se odinstalují všechny balíčky související s odinstalovávaným balíčkem).

Dalším stisknutím klávesy **g** se proces spustí.

Odstranit balíček včetně všech jeho **konfiguračních** souborů lze také pomocí klávesy podtržítka **_**. Další postup je stejný jako v předchozím případě.

13.6 Příkazy lastlog, last, faillog

13.6.1 Lastlog

Příkaz **lastlog** vypíše všechny účty a datum a čas jejich posledního přihlášení.

Tento příkaz formátuje a vypisuje obsah souboru /var/log/lastlog.

```
student@linux:~$ lastlog
```

```
Debian-exim                **Nikdy nebyl přihlášen**
student                    pts/1      192.168.0.1  Út pro 27 13:37:30 +0100 2005
sshd                      **Nikdy nebyl přihlášen**
saned                     **Nikdy nebyl přihlášen**
clamav                    **Nikdy nebyl přihlášen**
jabber                     **Nikdy nebyl přihlášen**
testik                     :0        Ne říj 16 18:08:36 +0200 2005
```

Zajímá-li nás konkrétní účet, zobrazíme si informace o jeho přihlášení příkazem

lastlog -u účet

```
student@linux:~$ lastlog -u student
```

```
Uživatel      Port      Z      Naposledy
student       pts/1     192.168.0.1  Út pro 27 13:37:30 +0100 2005
```

13.6.2 Last

Příkaz **last** vypíše seznam naposledy přihlášených uživatelů.

```
student@linux:~$ last
```

```
student pts/1      192.168.0.1    Tue Dec 27 13:37    still logged in
student :0          Tue Dec 27 13:30    still logged in
reboot  system boot  2.6.8-2-686    Tue Dec 27 13:30    (00:30)
student pts/1      192.168.0.1    Tue Dec 27 09:32 - down (03:54)
student :0          Tue Dec 27 09:30 - down (03:57)
reboot  system boot  2.6.8-2-686    Tue Dec 27 09:29    (03:57)
root    tty1          Sun Dec 18 17:20 - down (03:16)
student :0          Sun Dec 18 12:51 - 20:36 (07:45)
reboot  system boot  2.6.8-2-686    Sun Dec 18 12:50    (07:46)
student tty1          Tue Dec 13 20:01 - down (00:27)
student pts/1      192.168.0.1    Tue Dec 13 19:17 - 20:22 (01:05)
student :0          Tue Dec 13 19:14 - 20:28 (01:14)
```

13.6.3 Faillog

Příkaz **faillog** vypíše neúspěšná přihlášení k systému.

```
Uživatel  Chyb      Maximum  Poslední
root      0          0        St zář 29 11:42:56 +0200 2004 on tty1
```

Tento příkaz formátuje a vypisuje obsah souboru /var/log/faillog.

13.7 Cvičení

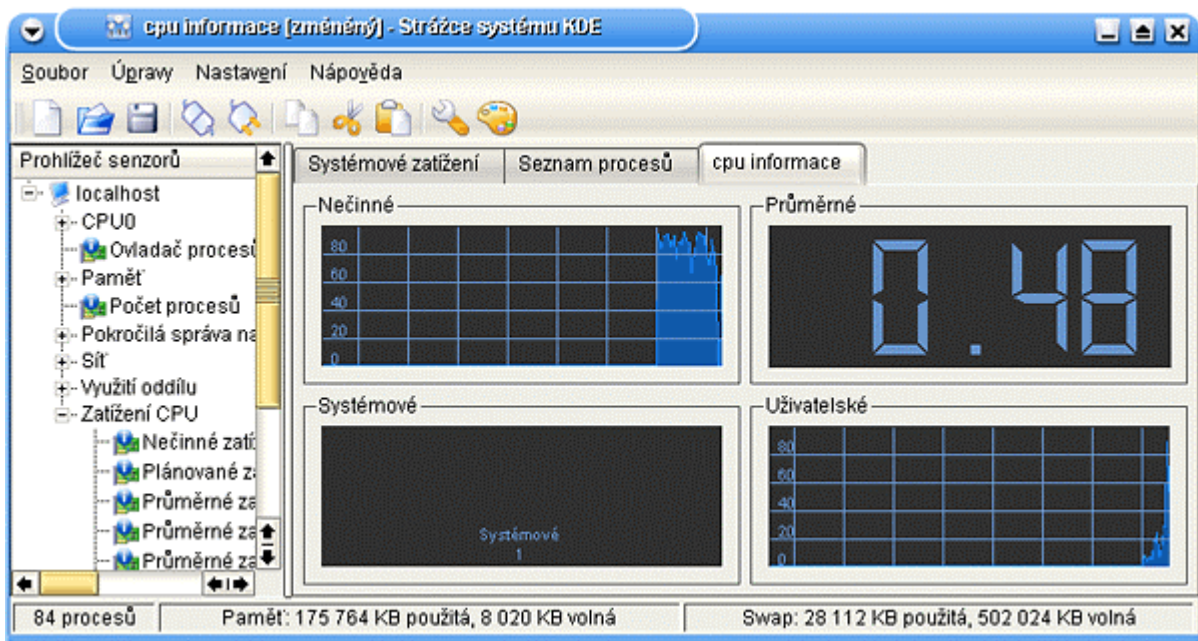
Spusťte **Strážce systému**.

Vytvořte novou záložku, která bude mít 2 řádky a 2 sloupce, nazvěte ji „**Informace o procesoru**“.

V levé části z rozbalovacího seznamu localhost vyberte „**Zatížení procesoru**“.

Po rozbalení přetáhněte myši do oblastí senzorů například „**Nečinné zatížení**“, „**Průměrné zatížení (5 min)**“, „**Uživatelské zatížení**“ a „**Systémové zatížení**“.

Průměrnému zatížení nastavte zobrazování číselné – multimetr, ostatním nastavte zobrazování ve formě buď sloupcového grafu nebo vykreslovače signálu.



13.8 Cvičení

Nainstalujte **xball** a **xpenguins**.

V nastavení pracovní plochy v záložce Chování nastavte Povolit programy v okně plochy.

Spusťte program xpenguins příkazem **xpenguins &**.



Spusťte program xball příkazem **xball &**.

Zjistěte čísla procesů a procesy xpenguins a xball ukončete.

 *Domácí úkol*

- Nainstalujte si balíček hwinfo a vypište si hw informace o svém počítači. Tyto informace přesměrujte do souboru vypis. Při výpisu souboru vyfiltrujte jen ty řádky, které obsahují řetězec eth.
- Vypište informace o procesoru.
- Vypište informace o scsi zařízeních.
- Vypište informace o usb zařízeních.
- Vypište informace o rozdělení a geometrii disku.
- Určete, kolik cylindrů, hlav a sektorů na stopu má váš HDD.
- Nainstalujte balíčky xosview, ksysguard a grellm. Použijte tyto grafické nástroje pro zobrazení běžících procesů v systému a pro zobrazení zatížení systému.
- Vypište systémové informace (OS, hw jméno, verze jádra, síťové jméno apod.)
- Vypište, jak dlouho systém běží, kolik je k němu přihlášeno uživatelů a jaká je systémová zátěž.
- Vypište otevřená spojení, směrovací tabulku, síťová rozhraní a statistiky pro jednotlivé protokoly.
- Nainstalujte xpenguins a spusťte.
- Zaktualizujte seznam dostupných balíčků a tyto balíčky zaktualizujte.
- Odinstalujte xpenguins.
- Použijte příkazy last, lastlog a faillog pro výpis informací o přihlašování uživatelů.
- Vypište seznam nainstalovaných balíčků.

 Shrnutí

- ✓ Umíte použít program aptitude pro instalaci, odinstalování a aktualizaci balíčků. Nainstalované balíčky umíte vypsat.
- ✓ Umíte zobrazit informace o HW.
- ✓ Umíte zobrazit systémové informace.
- ✓ Umíte vypsat běžící procesy.
- ✓ Umíte používat grafické nástroje pro zobrazení systémových informací.
- ✓ Umíte vypsat směrovací tabulku a ostatní informace ohledně síťového připojení.

14. SSH (27. – 28. hodina)

14.1 Vzdálený přístup SSH

SSH client je program pro vzdálený přístup k jinému počítači. Umožňuje vzdálené spuštění příkazů.

Zajišťuje šifrovanou bezpečnou komunikaci mezi dvěma počítači přes nezabezpečenou síť.

Ke vzdálenému počítači, na němž běží ssh (zkontrolujte výpis balíčku, měl by obsahovat ssh) se připojíme např. příkazem

```
ssh uživatel@jméno_počítače
```

kde uživatel je existující povolený účet na vzdáleném počítači a jméno_počítače je buď IP adresa počítače nebo jméno počítače, ke kterému existuje DNS překlad.

14.1.1 Příklad

Přihlásíme se vzdáleně k počítači s IP adresou 192.168.0.2 jako uživatel student.

```
linux:~# ssh student@192.168.0.2
```

```
The authenticity of host '192.168.0.2 (192.168.0.2)' can't be established.
```

```
RSA key fingerprint is c9:a0:c3:f4:8f:14:f4:4a:07:b6:cf:28:39:47:fc:81.
```

```
Are you sure you want to continue connecting (yes/no)? yes
```

```
Warning: Permanently added '192.168.0.2' (RSA) to the list of known hosts.
```

```
Password:
```

```
Linux linux 2.6.8-2-686 #1 Tue Aug 16 13:22:48 UTC 2005 i686 GNU/Linux
```

```
The programs included with the Debian GNU/Linux system are free software;
```

```
the exact distribution terms for each program are described in the
```

```
individual files in /usr/share/doc/*/copyright.
```

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
```

```
permitted by applicable law.
```

```
Last login: Tue Dec 27 19:04:32 2005 from 192.168.0.1
```

Během přihlašování se zobrazí varování, zda RSA key fingerprint daného čísla odpovídá. Toto číslo vrací SSH server a měli bychom si ho hned po nainstalování a přihlášení po bezpečné lince zapsat a při dalším přihlášení přes nezabezpečenou linku porovnat.

Pokud by čísla neodpovídala, mohlo by to znamenat, že máme v cestě prostředníka, který se snaží vydávat za cílový počítač – tzv. **man-in-the-middle**.

Po potvrzení fingerprintu (napíšeme na dotaz – yes) se do souboru **known_hosts** v domovském adresáři **/domovský_adresář/.ssh/known_hosts** přidá tento vzdálený počítač mezi známé počítače, jimž důvěřujeme. Příště již na dotaz, zda fingerprint odpovídá, již nebudeme muset odpovídat, porovnání fingerprintu se udělá skrytě pomocí souboru known_hosts.

Bude následovat jen zadání hesla k účtu, ke kterému se hlásíme.

14.2 Kopírování pomocí scp

Můžeme kopírovat soubory na vzdálený nebo ze vzdáleného počítače příkazem scp.

scp odkud/soubor kam

například

scp student@192.168.0.2:soubor .

nebo naopak

scp ./soubor student@192.168.0.2:

Vzdálený počítač je identifikován IP adresou (nebo přeložitelným jménem), za níž je **dvojtečka**, která znamená domovský adresář účtu, ke kterému se přihlašujeme.

Tečka označuje aktuální adresář.

14.3 Příklad

- Přihlaste se vzdáleně pomocí ssh k jinému počítači jako např. student.
- V domovském uživatele student vytvořte adresář pokus.
- V tomto adresáři vytvořte soubor test a vložte do něj pár řádků textu.
- Odhlaste se.
- Ve svém domovském adresáři vytvořte adresář pokus_local.
- Pomocí scp soubor test ze vzdáleného počítače zkopírujte do svého domovského adresáře do podadresáře pokus_local.
- V adresáři pokus_local vytvořte soubor test2 a pomocí scp jej zkopírujte na vzdálený počítač do adresáře test.
- Přihlaste se na vzdálený počítač a výpisem zkontrolujte výsledek.
- Nakonec smažte adresáře pokus i pokus_local.

14.3.1 Řešení

- student@linux:~\$ ssh student@192.168.0.2
Password:
Last login: Wed Dec 28 09:56:50 2005 from 192.168.0.1
student@linux:~\$
- student@linux:~\$ mkdir pokus
- student@linux:~\$ touch test
student@linux:~\$ echo "trochu textu" > pokus/test
student@linux:~\$ cat pokus/test
trochu textu
- student@linux:~\$ logout

- ```

e) student@linux:~$ pwd
/home/student
student@linux:~$ mkdir pokus_local
f) student@linux:~$ scp student@192.168.0.2:pokus/test pokus_local/
Password:
test 100% 13 0.0KB/s 00:00
student@linux:~$ ls pokus_local/
test
g) student@linux:~$ touch pokus_local/test2
student@linux:~$ ls pokus_local/
test test2
student@linux:~$ scp pokus_local/test2 student@192.168.0.2:pokus
Password:
test2 100% 0 0.0KB/s 00:00
student@linux:~$
h) student@linux:~$ ssh student@192.168.0.2
Password:
Last login: Wed Dec 28 09:57:38 2005 from 192.168.0.2
student@linux:~$ ls pokus
test test2
i) student@linux:~$ rm pokus/*
student@linux:~$ rmdir pokus
student@linux:~$ logout
Connection to 192.168.0.2 closed.
student@linux:~$ rm pokus_local/*
student@linux:~$ rmdir pokus_local/

```

## 14.4 Veřejný a privátní klíč

Pokud máme zabezpečený počítač a často se z něj hlásíme pomocí ssh k jinému vzdálenému počítači, můžeme pro autorizaci použít veřejný a privátní klíč. Poté bude autorizace probíhat skrytě a nebudeme muset zadávat heslo.

### 14.4.1 Jak probíhá přihlášení pomocí privátního a veřejného klíče (zjednodušeně)?

Na lokálním počítači je privátní klíč.

Na vzdáleném je veřejný klíč.

Lokální počítač inicializuje spojení.

Vzdálený počítač si „vymyslí“ např. nějaké číslo a to zašifruje svým veřejným klíčem a pošle lokálnímu počítači.

Lokální počítač tento řetězec dešifruje svým privátní klíčem a zašifruje např. MD5 a pošle vzdálenému počítači.

Vzdálený počítač si své původní číslo také zašifruje MD5 a porovná s řetězcem, který dostal od lokálního počítače.

Pokud se řetězce rovnají, přihlášení je ověřeno a začne probíhat následná komunikace.

### 14.4.2 Postup vygenerování a použití klíčů

- a) Přihlásíme se na lokální počítač, například jako uživatel **student**.

- b) Dáme vygenerovat klíče příkazem **ssh-keygen** (vygenerují se dva klíče – veřejný – id\_rsa.pub a privátní – id\_rsa).  
 student@linux:~\$ **ssh-keygen -t rsa**  
 Generating public/private rsa key pair.  
 Enter file in which to save the key (/home/student/.ssh/id\_rsa):  
 Enter **passphrase** (empty for no passphrase):  
 Enter same passphrase again:  
 Your identification has been saved in /home/student/.ssh/id\_rsa.  
 Your public key has been saved in /home/student/.ssh/id\_rsa.pub.  
 The key fingerprint is:  
 df:8f:fe:6a:56:aa:3d:fc:97:ac:d4:d5:e9:6e:b8:8c student@linux
- c) Připojíme se na vzdálený počítač pomocí ssh  
 student@linux:~\$ **ssh student@192.168.0.2**  
 Password:
- d) V domovském adresáři vytvoříme adresář .ssh.  
 student@linux:~\$ **mkdir .ssh**
- e) Změníme .ssh práva na 700 (plné oprávnění jen pro vlastníka)  
 student@linux:~\$ **chmod 700 .ssh/**
- f) V .ssh vytvoříme soubor authorized\_keys  
 student@linux:~\$ **cd .ssh/**  
 student@linux:~/ssh\$ **touch authorized\_keys**
- g) Změníme práva pro soubor authorized\_keys na 600  
 student@linux:~/ssh\$ **chmod 600 authorized\_keys**
- h) Odhlásíme se  
 student@linux:~/ssh\$ **logout**
- i) Jako uživatel student na lokálním počítači odešleme veřejný klíč id\_rsa.pub do adresáře .ssh na vzdálený počítač  
 student@linux:~\$ **scp ~/.ssh/id\_rsa.pub student@192.168.0.2:~/.ssh**  
 Password:  
 id\_rsa.pub 100% 223 0.2KB/s 00:00
- j) Přihlásíme se ssh na vzdálený počítač  
 student@linux:~\$ **ssh student@192.168.0.2**  
 Password:
- k) Přidáme veřejný klíč z id\_rsa.pub do autorizovaných klíčů  
 student@linux:~\$ **cat .ssh/id\_rsa.pub >> .ssh/authorized\_keys**
- l) Smažeme veřejný klíč  
 student@linux:~\$ **rm .ssh/id\_rsa.pub**
- m) Odhlásíme se a zkusíme se znovu přihlásit ssh. Nemělo by se po nás chtít přihlašovací heslo.  
 student@linux:~\$ **logout**  
 Connection to 192.168.0.2 closed.
- n) student@linux:~\$ **ssh student@192.168.0.2**  
 Last login: Wed Dec 28 10:19:55 2005 from 192.168.0.2  
 student@linux:~\$

#### 14.4.3 Poznámka

Ve chvíli, kdy se generovaly klíče, se mohla zadat tzv. **passphrase**. Pokud bychom ji zadali, pak bychom ji při ssh přístupu ke vzdálenému počítači museli zadávat. Výhodou by bylo, že by se při přihlašování neodesílalo heslo ke vzdálenému účtu v žádné podobě – ani šifrované – ale odesílala by se passphrase.



Toto heslo – passphrase – si může pamatovat ssh-agent. Pak jej nebudeme muset zadávat.

**Postup:**

- ❑ **ssh-agent bash**
- ❑ **ssh-add**
- ❑ na výzvu zadáme heslo – passphrase

## 14.5 Cvičení

- ❑ Vytvořte nové účty, například test a pokus.
- ❑ V příkazovém řádku se přihlaste jako uživatel test.
- ❑ Vyzkoušejte ssh spojení na počítač spolužáka (nebo můžete i na svůj vlastní) a zkuste se přihlásit na účet pokus.
- ❑ V domovském adresáři uživatele pokus na vzdáleném počítači vytvořte adresář adr1 a v něm soubor s1, do něhož vložte nějaký text.
- ❑ Pomocí scp zkopírujte soubor s1 do domovského adresáře uživatele test.
- ❑ Jako uživatel test na lokálním počítači upravte soubor s1 (přidejte do něj další text) a pomocí scp jej zkopírujte do adresáře adr1 (přepište původní soubor).
- ❑ Výpisem zkontrolujte, zda se přepsání uskutečnilo.
- ❑ Jako uživatel test na lokálním počítači vytvořte pár klíčů. Výpisem zkontrolujte jejich obsah.
- ❑ Zajistěte, aby se při ssh přístupu nemuselo zadávat heslo a pokud při vytváření klíčů použijete neprázdnou passphrase, nechejte ji pamatovat ssh-agentovi.
- ❑ Ověřte a na závěr uveďte vše do výchozího stavu (smazat klíče i nové účty).

 *Domácí úkol*

Vyzkoušejte ssh přístup na vzdálený počítač (můžete takto přistupovat i ke svému lokálnímu počítači).

Vyzkoušejte vzdálené kopírování pomocí scp (v obou směrech, z domovské adresáře i podadresářů).

Vyzkoušejte přihlašování s pomocí klíčů.

Vyzkoušejte práci s ssh-agentem.

Zopakujte si vytváření účtů, skupin, nastavování oprávnění.

 *Shrnutí*

- ✓ Umíte se přihlásit pomocí ssh na vzdálený počítač.
- ✓ Umíte kopírovat pomocí scp.
- ✓ Umíte vytvořit veřejný a privátní klíč pro ssh přístup.
- ✓ Umíte použít ssh-agenta pro pamatování si passphrase.
- ✓ Umíte použít klíče pro přihlašování přes ssh.

## 15. Samba (29. – 30. hodina)

### 15.1 Instalace samby

Nainstalujeme balíčky **smbfs**, **smbclient** a **samba**.

### 15.2 Konfigurace

V souboru **/etc/samba/smb.conf** jsou definovány parametry konfigurace samby.

Řádky začínající středníkem; nebo mřížkou # jsou komentáře.

Po modifikaci souboru by se měl spustit příkaz testparm, kterým ověříme, že v souboru nemáme syntaktické chyby.

#### 15.2.1 Úprava konfiguračního souboru smb.conf

V **Global settings** odkomentujeme nebo napíšeme název pracovní skupiny nebo domény.

**workgroup = DOMA**

V části **Authentication** nastavíme parametr **security=user** (je-li nutno, odkomentujeme řádek smazáním #).

Další možnosti nastavení security (určí se tím, jak bude prováděno ověřování uživatelů):

- share – ověřování jako u Windows95
- user – ověřování vůči uživatelům v Linuxu
- server – ověřování vůči jinému serveru
- domain – pro případ, že by byl linuxový počítač součástí domény
- ads- pro případ, že by byl linuxový počítač součástí domény s Active Directory

Pro správné **kódování** přidáme odstavec

##### **Dalsi nastaveni** #####

**dos charset=cp852**

**unix charset=iso8859-2**

V části **Homes** nastavíme:

**valid users = %S**

tím zajistíme, že se uživatel bude moci přihlásit jen ke svému domovskému adresáři

**writable = yes**

tím zajistíme, že uživatel bude moci i zapisovat.

## 15.3 Heslo do samby

Protože Windows a Linux používají jiný systém kódování, je nutno nastavit uživatelům, kteří chtějí přes sambu k linuxu přistupovat, také heslo do samby.

To provedeme příkazem **smbpasswd**.

**smbpasswd -a účet**

**smbpasswd -a student**

## 15.4 Restart a reload samby

**/etc/init.d/samba restart**

...zastavení a spuštění služby

**/etc/init.d/samba reload**

...pouze znovu načtení konfiguračních souborů bez restartu služby

**/etc/init.d/samba force-reload**

...vynucené načtení konfiguračních souborů bez restartu služby

**/etc/init.d/samba start**

...spuštění služby

**/etc/init.d/samba stop**

...stopnutí služby

## 15.5 Výpisy

### 15.5.1 Testparm

**testparm -v > vypis\_parametru**

Tímto příkazem vypíšeme nastavení parametrů a přesměrujeme do souboru vypis\_parametru.

#### Příklad části výpisu:

```
Global parameters
[global]
```

```
dos charset = cp852
unix charset = iso8859-2
display charset = LOCALE
workgroup = DOMA
realm =
netbios name = LINUX
netbios aliases =
netbios scope =
server string = %h server (Samba %v)
interfaces =
bind interfaces only = No
security = USER
auth methods =
encrypt passwords = Yes
update encrypted = No
```

```
client schannel = Auto
server schannel = Auto
allow trusted domains = Yes
hosts equiv =
min password length = 5
map to guest = Never
null passwords = No
obey pam restrictions = Yes
password server = *
smb passwd file =
/etc/samba/smbpasswd
private dir = /etc/samba
passdb backend = tdbsam, guest
algorithmic rid base = 1000
root directory =
guest account = nobody
enable privileges = No
```

```

pam password change = No
passwd program = /usr/bin/passwd %u [homes]
passwd chat = comment = Home Directories
Enter\snew\sUNIX\spassword: %n\n valid users = %S
Retype\snew\sUNIX\spassword: %n\n . read only = No
passwd chat debug = No create mask = 0700
passwd chat timeout = 2 directory mask = 0700
check password script = browseable = No
username map =
password level = 0 [printers]
username level = 0 comment = All Printers
unix password sync = No path = /tmp
restrict anonymous = 0 create mask = 0700
lanman auth = Yes printable = Yes
ntlm auth = Yes browseable = No
client NTLMv2 auth = No
client lanman auth = Yes
client plaintext auth = Yes [print$]
 comment = Printer Drivers
 path = /var/lib/samba/printers

```

### 15.5.2 Smbstatus

**Smbstatus** – výpis kdo je kam připojen a jaké soubory jsou zamčeny.

**Př.**

```

student@linux:~$ smbstatus
Samba version 3.0.14a-Debian
PID Username Group Machine

 9300 student users klient98-69 (192.168.0.1)

Service pid machine Connected at

student 9300 klient98-69 Wed Dec 28 12:36:57 2005

Locked files:
Pid DenyMode Access R/W Oplock Name

9300 DENY_NONE 0x20089 RDONLY EXCLUSIVE+BATCH
/home/student/heslo.txt Wed Dec 28 12:38:50 2005

```

### 15.6 Zrušení konekce z windows

V příkazovém řádku Windows napíšeme příkaz

**net use**

kterým zjistím všechny aktuální spojení z Windows na nějaký vzdálený počítač.

```

C:\Documents and Settings\I> net use
Nová připojení budou zapamatována.
Stav Místní Vzdálené Síťové

OK \\192.168.0.2\STUDENT Síť Microsoft Windows
Příkaz byl úspěšně dokončen.

```

Konekci zrušíme příkazem `net use název_konekce /d`, například v předchozím příkladu

```
C:\Documents and Settings\I> net use \\192.168.0.2\STUDENT /d
```

Otevřené soubory nebo nedokončená prohlédávání adresářů čekají na vyřízení u připojení \\192.168.0.2\STUDENT.

Opravdu chcete v odpojení pokračovat a vynutit jejich zavření? (A/N) [N]: a  
 \\192.168.0.2\STUDENT odstraněno úspěšně.

## 15.7 Připojení sdíleného adresáře

### 15.7.1 Připojení pomocí mount s právy root

```
mount -t smbfs -o username=jméno_co_připojit kam
```

```
mount -t smbfs -o username=linux //192.168.0.1/linux /media/sit
```

Username je jméno, které nás opravňuje na vzdáleném počítači k připojení specifikované složky. V dalším kroku bude vyžadováno heslo.

Je možné specifikovat další parametry, jako například kódování, heslo, uid uživatele, který bude moci do připojené složky přistupovat, fmask a dmask – přednastavená oprávnění pro vznikající soubory a adresáře.

```
mount //192.168.0.1/linux /media/sit -t smbfs -o
```

```
oiocharset=iso8859-2,codepage=cp852,username=linux,guest,uid=student,fmask=644,dmask=755
```

Předchozí příklad je napsát v jednom řádku, za `-o` je mezera, mezi jednotlivými parametry za `-o` nejsou žádné mezery, parametry jsou odděleny čárkami.

Pořadí je většinou libovolné, `-t smbfs` můžeme napsat hned za `mount`, stejně tak parametry `-o ...` můžeme napsat před název připojované složky.

V předchozím příkladu se připojuje složka **linux z Windows** do linuxového adresáře **/media/sit**. Kvůli rozdílnému kódování Windows vs. Linux je dobré uvést **kódování** (oiocharset a codepage).

Parametr **guest** znamená, že budeme ke složce linux na vzdáleném počítači přistupovat bez hesla.

**UID** je nastaveno na student, proto bude moci i uživatel student vytvářet v připojeném adresáři (/media/sit) soubory a složky s přednastavenými právy 755 a 644.

Vytvoříme-li jako student v adresáři soubor a.txt a adresář a, bude výpis vypadat takto:

```
drwxr-xr-x 1 student root 4096 2005-12-28 12:48 a
-rw-r--r-- 1 student root 835 2005-12-28 12:50 a.txt
```

### 15.7.2 Připojení s právy běžného uživatele

Jako běžný uživatel nebude mít nejspíš oprávnění použít příkaz mount k připojení vzdáleného zdroje.

Lze pro tento účel použít příkaz smbmount, přičemž připojovat můžeme jen do složky, ke které máme jako běžný uživatel přístup.

**smbmount co\_připojit kam -o username=jméno**

**smbmount //192.168.0.1/linux /media/sit -o username=linux**

Stejně jako v předchozím příkladu připojení pomocí mount lze specifikovat další parametry za -o (např. password, dmask, fmask, guest, ro – pro čtení, rw – pro čtení i zápis, uid, codepage, iocharset apod.)

### 15.7.3 Připojení pomocí fstab

Abychom nemuseli pokaždé, kdy potřebujeme připojit vzdálený prostředek, vypisovat dlouhý příkaz pro připojení, můžeme toto nadefinovat do souboru /etc/fstab.

#### Příklad souboru fstab:

```
/etc/fstab: static file system information.
#
<file system> <mount point> <type> <options> <dump> <pass>
proc /proc proc defaults 0 0
/dev/hda2 / ext3 defaults,errors=remount-ro,usrquota,grpquota 0 1
/dev/hda3 none swap sw 0 0
/dev/hdc /media/cdrom0 iso9660 ro,user,noauto 0 0
/dev/fd0 /media/floppy0 auto rw,user,noauto 0 0
//192.168.0.1/linux /media/sit smbfs noauto,user,iocharset=iso8859-2,codepage=cp852,username=l,guest,fmask=777,dmask=777 0 0
```

Poslední řádek je do souboru fstab přidán.

Definuje se v něm, že připojujeme adresář linux na počítači s IP adresou 192.168.0.1 do adresáře /media/sit na lokálním počítači.

Typ připojeného adresáře je smbfs (nezáleží na tom, jaký se skutečně souborový systém na vzdáleném prostředku, protože jej připojujeme prostřednictvím samby).

V options je definováno: noauto – při startu OS se nesnaží tento vzdálený prostředek automaticky připojit – připojí jej, až o to uživatel požádá příkazem mount.

Pokud chce uživatel připojit vzdálený prostředek, který se automaticky nepřipojuje, stačí definovat v příkazu mount přípojný bod a zbytek se dohledá v fstab.

**mount /media/sit**

## 15.8 Výpis sdílených složek

Potřebujeme-li vědět, jaké prostředky pod jakými jmény jsou na určitém počítači sdíleny, použijeme následující příkaz:

**smbclient -L IP\_adresa**

### 15.8.1 Příklad

**smbclient -L 192.168.0.1**

Domain=[KLIENT98-69] OS=[Windows 5.1] Server=[Windows 2000 LAN Manager]

| Sharename    | Type    | Comment                                |
|--------------|---------|----------------------------------------|
| -----        | ----    | -----                                  |
| fotak        | Disk    |                                        |
| HPD          | Printer | HPD                                    |
| IPC\$        | IPC     | Vzdálený IPC                           |
| D\$          | Disk    | Výchozí sdílená položka                |
| print\$      | Disk    | Ovladače tiskárny                      |
| Tiskárna     | Printer | Tiskem na tomto zařízení odešlete fax. |
| cisco        | Disk    |                                        |
| F\$          | Disk    | Výchozí sdílená položka                |
| ADMIN\$      | Disk    | Vzdálený správce                       |
| linux        | Disk    |                                        |
| C\$          | Disk    | Výchozí sdílená položka                |
| backup_sys   | Disk    |                                        |
| SdílenéDokum | Disk    |                                        |
| web_server   | Disk    |                                        |

## 15.9 Konekce z Windows na Linux

Za předpokladu, že máme na Linuxu spuštěnou a správně nakonfigurovanou sambu a hesla do samby, můžeme zkusit z windowsovského stroje najít například pomocí IP adresy tento linuxový stroj.

Je-li k dispozici účet bb, vyhledáme z Windows **192.168.0.2/bb**.

Pokud jsou parametry v smb.conf nastaveny tak, jak bylo popsáno v části o konfiguraci, měli byste být schopni získat přístup do domovských adresářů uživatelů, samozřejmě po ověření jména a hesla.

Pokud jste se již k linuxovému počítači jednou úspěšně přihlásili a chcete se přihlásit na jiný účet, budete muset pravděpodobně předchozí konekci zrušit (**net use**).



## 15.10 Výpis smb.conf

```
Sample configuration file for the Samba suite for Debian GNU/Linux.
This is the main Samba configuration file. You should read the smb.conf(5) manual page in order to understand the
#options listed here. Samba has a huge number of configurable options most of which are not shown in this example
Any line which starts with a ; (semi-colon) or a # (hash) is a comment and is ignored. In this example we will use a # for
#commentary and a ; for parts of the config file that you may wish to enable
#
NOTE: Whenever you modify this file you should run the command "testparm" to check that you have not many any basic
#syntactic errors.
```

```
#===== Global Settings =====
```

```
[global]
```

```
Browsing/Identification
Change this to the workgroup/NT-domain name your Samba server will part of
workgroup = DOMA
server string is the equivalent of the NT Description field
server string = %h server (Samba %v)
Windows Internet Name Serving Support Section:
WINS Support - Tells the NMBD component of Samba to enable its WINS Server
; wins support = no
WINS Server - Tells the NMBD components of Samba to be a WINS Client
Note: Samba can be either a WINS Server, or a WINS Client, but NOT both
; wins server = w.x.y.z
This will prevent nmbd to search for NetBIOS names through DNS.
dns proxy = no
What naming service and in what order should we use to resolve host names
to IP addresses
; name resolve order = lmhosts host wins bcast
Debugging/Accounting
This tells Samba to use a separate log file for each machine
that connects
log file = /var/log/samba/log.%m
Put a capping on the size of the log files (in Kb).
max log size = 1000
If you want Samba to only log through syslog then set the following
parameter to 'yes'.
; syslog only = no
We want Samba to log a minimum amount of information to syslog. Everything
should go to /var/log/samba/log.{smbd,nmbd} instead. If you want to log
through syslog you should set the following parameter to something higher.
syslog = 0
Do something sensible when Samba crashes: mail the admin a backtrace
panic action = /usr/share/samba/panic-action %d
```

```
Authentication
```

```
"security = user" is always a good idea. This will require a Unix account
in this server for every user accessing the server. See
/usr/share/doc/samba-doc/html/docs/ServerType.html in the samba-doc
package for details.
security = user
You may wish to use password encryption. See the section on
'encrypt passwords' in the smb.conf(5) manpage before enabling.
encrypt passwords = true
If you are using encrypted passwords, Samba will need to know what
password database type you are using.
passdb backend = tdbsam guest
obey pam restrictions = yes
; guest account = nobody
invalid users = root
map to guest = never
This boolean parameter controls whether Samba attempts to sync the Unix
password with the SMB password when the encrypted SMB password in the
passdb is changed.
; unix password sync = no
```

```
For Unix password sync to work on a Debian GNU/Linux system, the following
parameters must be set (thanks to Augustin Luton <aluton@hybrigenics.fr> for
sending the correct chat script for the passwd program in Debian Potato).
passwd program = /usr/bin/passwd %u
passwd chat = *Enter\snew\sUNIX\spassword:* %n\n *Retye\snew\sUNIX\spassword:* %n\n .

This boolean controls whether PAM will be used for password changes
when requested by an SMB client instead of the program listed in
'passwd program'. The default is 'no'.
; pam password change = no

Printing

If you want to automatically load your printer list rather
than setting them up individually then you'll need this
; load printers = yes

lpr(ng) printing. You may wish to override the location of the
printcap file
; printing = bsd
; printcap name = /etc/printcap
CUPS printing. See also the cupsaddsmb(8) manpage in the
cupsys-client package.
; printing = cups
; printcap name = cups
When using [print$], root is implicitly a 'printer admin', but you can
also give this right to other users to add drivers and set printer
properties
; printer admin = @ntadmin

File sharing

Name mangling options
; preserve case = yes
; short preserve case = yes

Dalsi nastaveni
dos charset=cp852
unix charset=iso8859-2

Misc

Using the following line enables you to customise your configuration on a per machine basis. The %m gets replaced
#with the netbios name of the machine that is connecting
; include = /home/samba/etc/smb.conf.%m

Most people will find that this option gives better performance.
See smb.conf(5) and /usr/share/doc/samba-doc/htmldocs/speed.html for details
You may want to add the following on a Linux system:
SO_RCVBUF=8192 SO_SNDBUF=8192
socket options = TCP_NODELAY

The following parameter is useful only if you have the linpopup package installed. The samba maintainer and the linpopup
#maintainer are working to ease installation and configuration of linpopup and samba.
; message command = /bin/sh -c '/usr/bin/linpopup "%f" "%m" %s; rm %s' &

Domain Master specifies Samba to be the Domain Master Browser. If this machine will be configured as a BDC (a
#secondary logon server), you must set this to 'no'; otherwise, the default behavior is recommended.
; domain master = auto

Some defaults for winbind (make sure you're not using the ranges for something else.)
; idmap uid = 10000-20000
; idmap gid = 10000-20000
; template shell = /bin/bash

===== Share Definitions =====

[homes]
comment = Home Directories
```

```
browseable = no
valid users = %S

By default, the home directories are exported read-only. Change next parameter to 'yes' if you want to be able to write to them.
writable = yes

File creation mask is set to 0700 for security reasons. If you want to create files with group=rw permissions, set next
parameter to 0775.
create mask = 0700

Directory creation mask is set to 0700 for security reasons. If you want to
create dirs. with group=rw permissions, set next parameter to 0775.
directory mask = 0700

Un-comment the following and create the netlogon directory for Domain Logons
(you need to configure Samba to act as a domain controller too.)
;[netlogon]
; comment = Network Logon Service
; path = /home/samba/netlogon
; guest ok = yes
; writable = no
; share modes = no

[printers]
comment = All Printers
browseable = no
path = /tmp
printable = yes
public = no
writable = no
create mode = 0700

Windows clients look for this share name as a source of downloadable
printer drivers
[print$]
comment = Printer Drivers
path = /var/lib/samba/printers
browseable = yes
read only = yes
guest ok = no
Uncomment to allow remote administration of Windows print drivers.
Replace 'ntadmin' with the name of the group your admin users are
members of.
; write list = root, @ntadmin

A sample share for sharing your CD-ROM with others.
;[cdrom]
; comment = Samba server's CD-ROM
; writable = no
; locking = no
; path = /cdrom
; public = yes

The next two parameters show how to auto-mount a CD-ROM when the
cdrom share is accessed. For this to work /etc/fstab must contain
an entry like this:
#
/dev/scd0 /cdrom iso9660 defaults,noauto,ro,user 0 0
#
The CD-ROM gets unmounted automatically after the connection to the
#
If you don't want to use auto-mounting/unmounting make sure the CD
is mounted on /cdrom
#
; preexec = /bin/mount /cdrom
; postexec = /bin/umount /cdrom
```

 *Domácí úkol*

- Nainstalujte sambu.
- Nakonfigurujte ji.
- Vytvořte testovací účet, například test.
- V jeho domovském adresáři vytvořte soubor test.txt.
- Z Windows vyhledejte adresář test na Linuxu a editujte soubor test.txt.
- Vytvořte zde nějaký další adresář a v něm nový soubor.
- Zkuste konekci opačnou – z Linuxu do Windows.
- Ve Windows nasdílejte adresář.
- Zkuste jej z Linuxu připojit a po připojení do něj zapisovat.
- Vyzkoušejte neautomatické připojení této složky z Windows pomocí definice připojení v fstab.
- Vyzkoušejte příkaz smbmount a mount.
- Zkuste se z Windows přihlásit do jiného domovského adresáře a v souvislosti s tím zrušte předchozí konekci.
- Otevřete nějaký soubor a pomocí smbstatus ověřte, zda je tento soubor zamknut.

 *Shrnutí*

- ✓ Umíte nainstalovat potřebné balíčky pro spuštění samby.
- ✓ Umíte udělat základní potřebné změny v konfiguračním souboru, aby fungovalo spojení z jiného počítače na počítač se sambou.
- ✓ Umíte připojit z Linuxu sdílenou složku z jiného počítače.
- ✓ Umíte v fstab nadefinovat neautomatické připojení sdílené složky z jiného počítače.
- ✓ Umíte jako běžný uživatel připojit sdílenou složku pomocí smbmount.
- ✓ Umíte zjistit, jaké konekce se právě uskutečňují.

## 16. Instalace OS Linux a nastavení sítě (31. - 32. hodina)

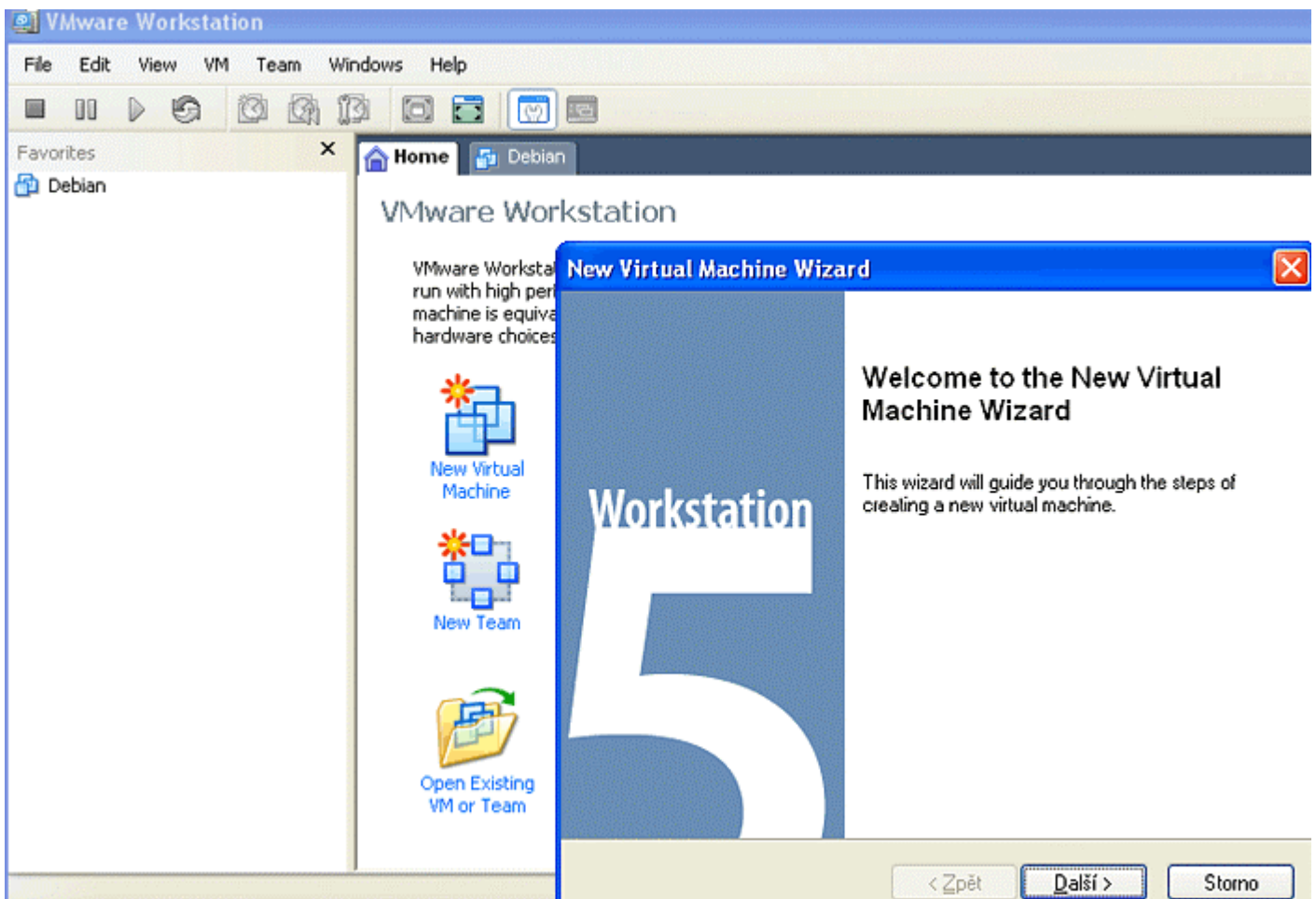
V této hodině provedeme instalaci OS Linux. Nejprve nainstalujeme základní systém, pak systém naučíme znát zdroje pro instalace dalších programů. Následně si vybereme, jaký typ souboru balíčků budeme instalovat. Je možno vybrat si např. ze stolního počítače, poštovního serveru atd. nebo je možno vybrat balíčky ručně.

V této hodině zvolíme instalaci přednastavené sady balíčků pro stolní počítač.

### 16.1 Instalace OS

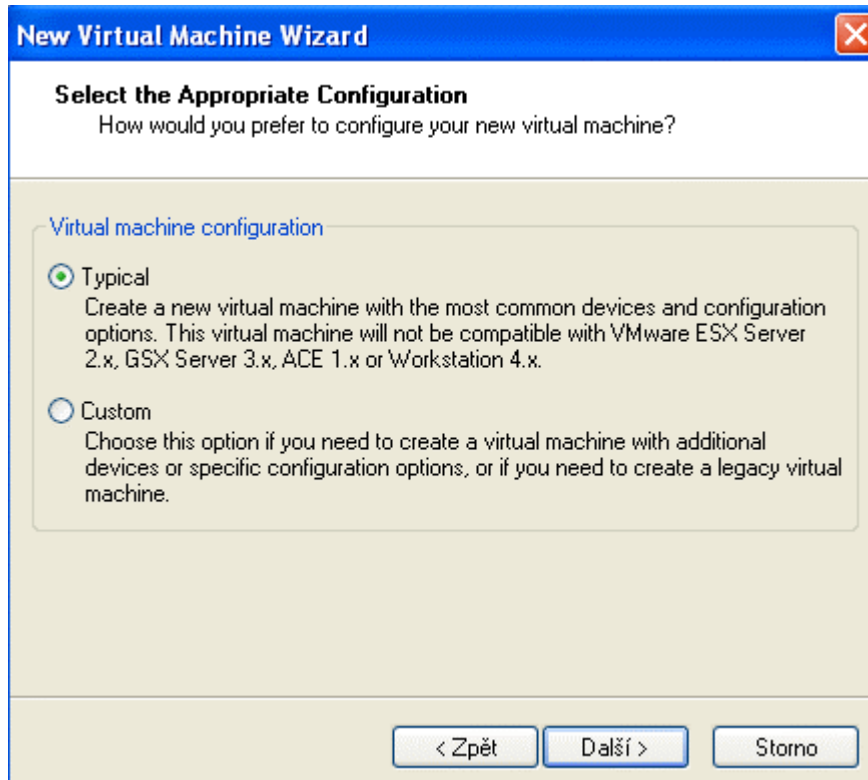
Na počítačích je nainstalován program **VMware Workstation**, který umožňuje nainstalovat souběžně další operační systémy, které mohou i běžet souběžně.

Program spustíme a klikneme na ikonu „**New Virtual Machine**“.

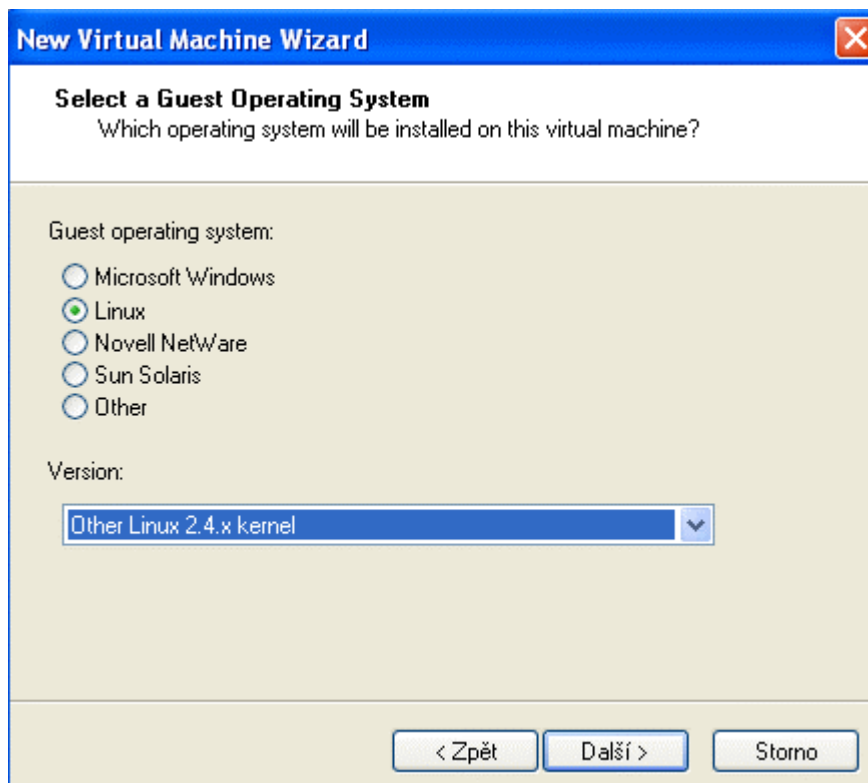


Klikneme na tlačítko **Další**.

Vybereme první volbu – „**Typical**“.



Klikneme na tlačítko **Další**. V následujícím dialogovém okně vybereme „**Linux**“ a verzi „**Other Linux 2.4.x kernel**“.



Po kliknutí na tlačítko **Další** se zobrazí další dialogové okno. Zadáme název nového virtuálního stroje a umístění. Každý virtuální stroj je nainstalován ve zvoleném adresáři a je možné jej zazálohovat.

**New Virtual Machine Wizard** [Close]

**Name the Virtual Machine**  
What name would you like to use for this virtual machine?

Virtual machine name  
Linux Debian

Location  
C:\Documents and Settings\student\Dokumenty\My Virtual Mac [Browse...]

< Zpět    Další >    Storno

Klikneme na tlačítko **Další**. Dále vybereme první možnost – „**Use bridged networking**“.

**New Virtual Machine Wizard** [Close]

**Network Type**  
What type of network do you want to add?

Network connection

- Use bridged networking  
Give the guest operating system direct access to an external Ethernet network. The guest must have its own IP address on the external network.
- Use network address translation (NAT)  
Give the guest operating system access to the host computer's dial-up or external Ethernet network connection using the host's IP address.
- Use host-only networking  
Connect the guest operating system to a private virtual network on the host computer.
- Do not use a network connection

< Zpět    Další >    Storno

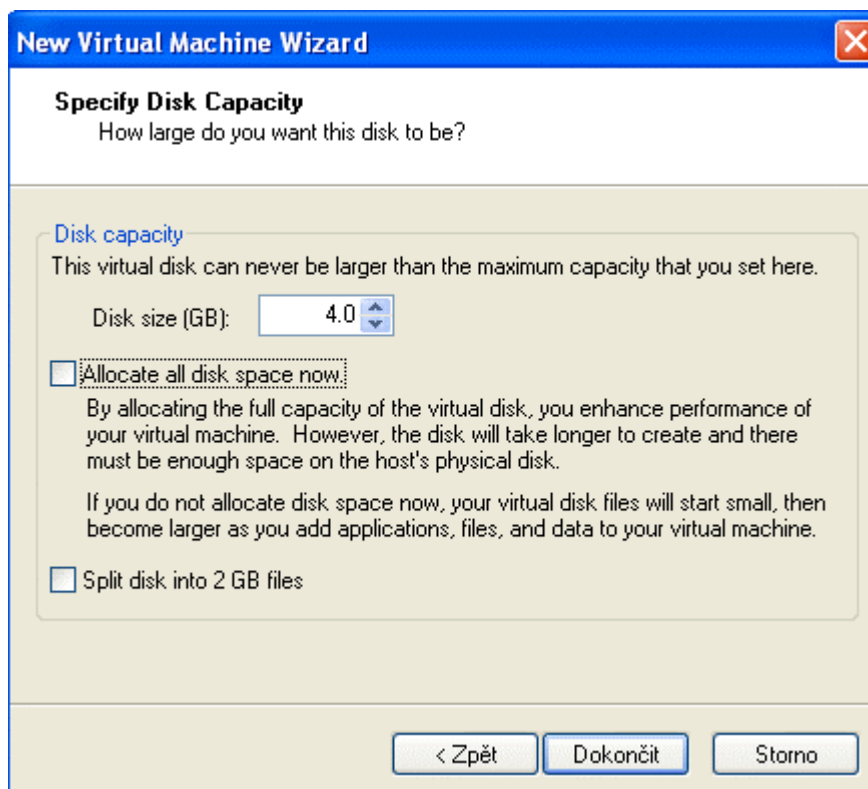
„**Bridged networking**“ umožňuje tomuto nově instalovanému OS přímý přístup do externí sítě. Tento nový OS bude mít svou vlastní IP adresu.

Druhá možnost – „**Network address translation – NAT**“ – umožňuje virtuálnímu stroji přistupovat k externí síti pomocí IP adresy rodičovského operačního systému, ve kterém je VMWare nainstalován. Tento virtuální stroj se skrývá za IP adresu rodičovského systému.

Třetí možnost – „**host-only networking**“ – umožňuje virtuálnímu stroji spojit se s privátní virtuální sítí rodičovského počítače.

Při volbě poslední možnosti vytvoříme virtuální stroj bez síťového spojení.

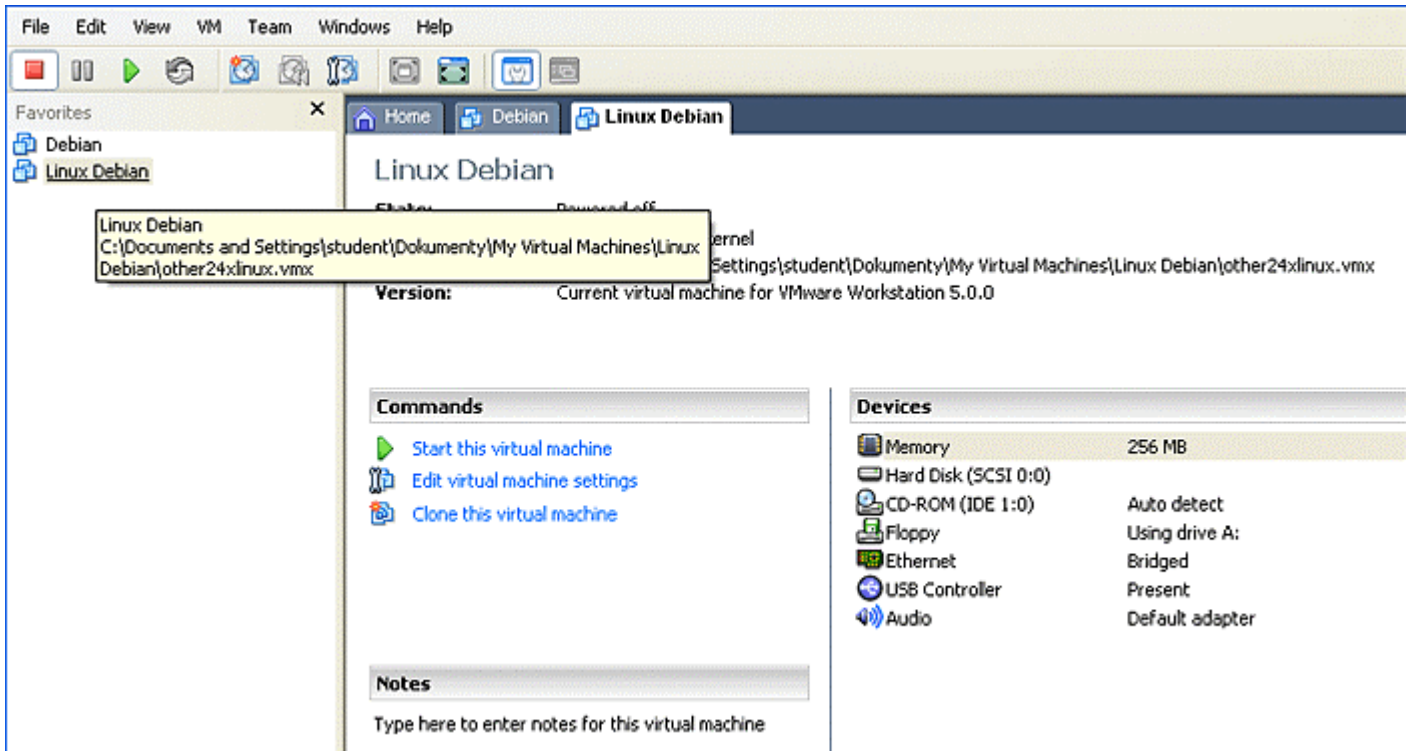
Po kliknutí na tlačítko **Další** se zobrazí následující dialogové okno, kde je možno zvolit velikost diskového prostoru, který bude virtuálnímu stroji přidělen.



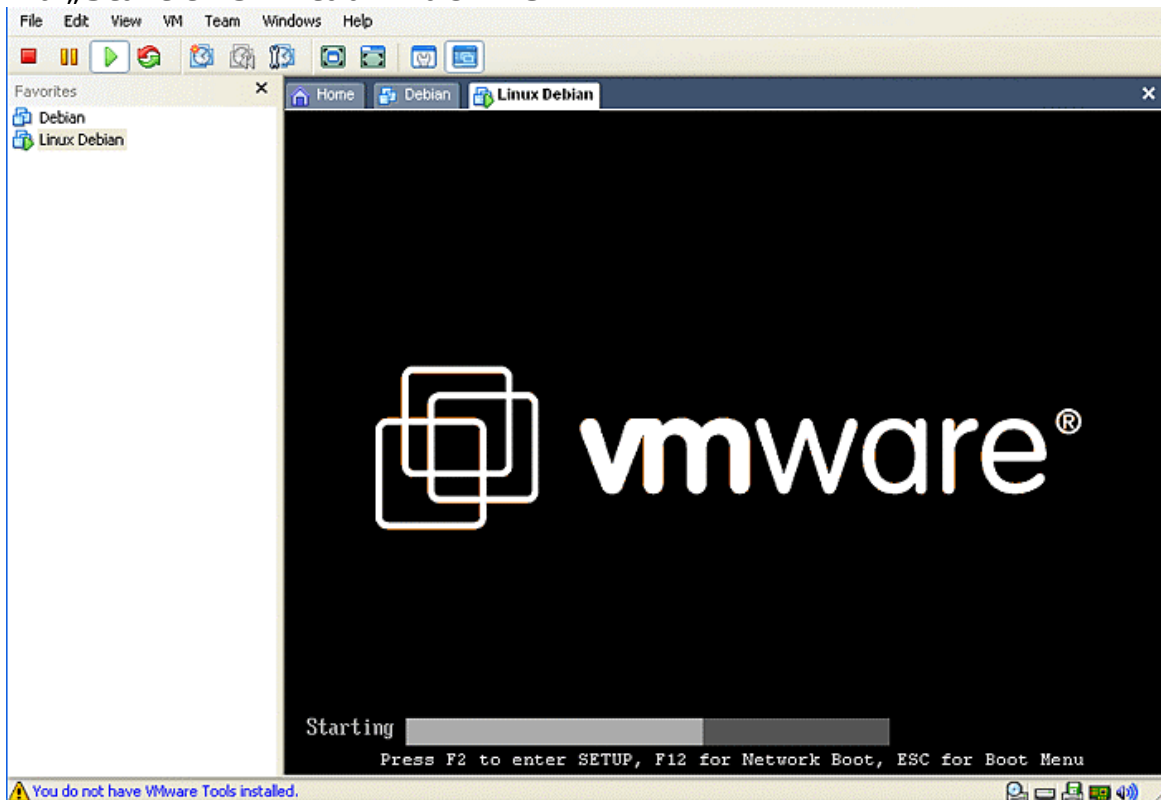
Klikneme na tlačítko **Dokončit**.



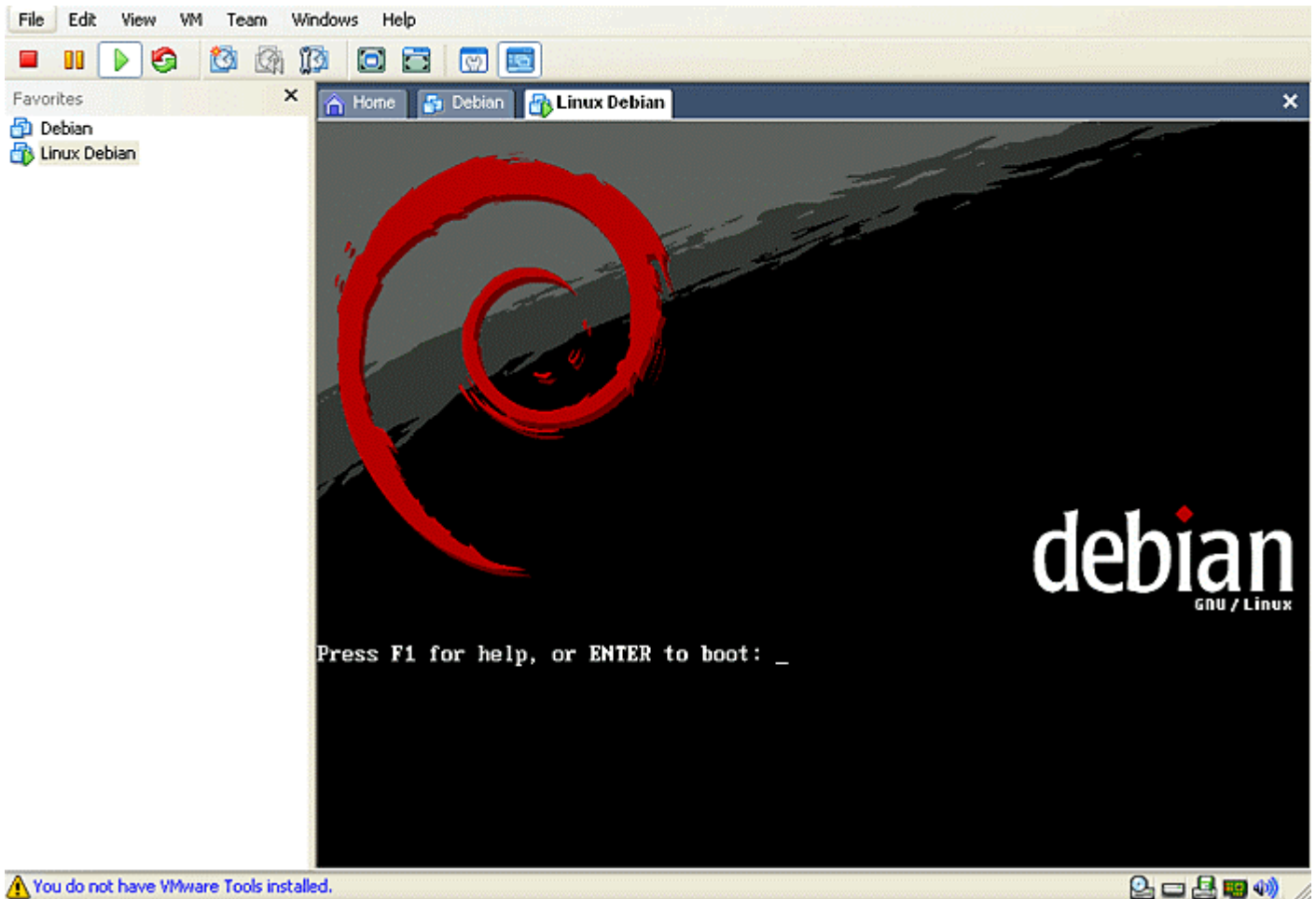
Předchozím postupem jsme vytvořili nový virtuální stroj se jménem **Linux Debian**, do kterého nainstalujeme zvolený operační systém.



Do DVD mechaniky vložíme první instalační DVD Debianu a klikneme na „**Start this virtual machine**“.



Ukazatel myši umístíme do prostoru určenému pro virtuální stroj a stiskneme **Enter**, čímž spustíme bootování z DVD.



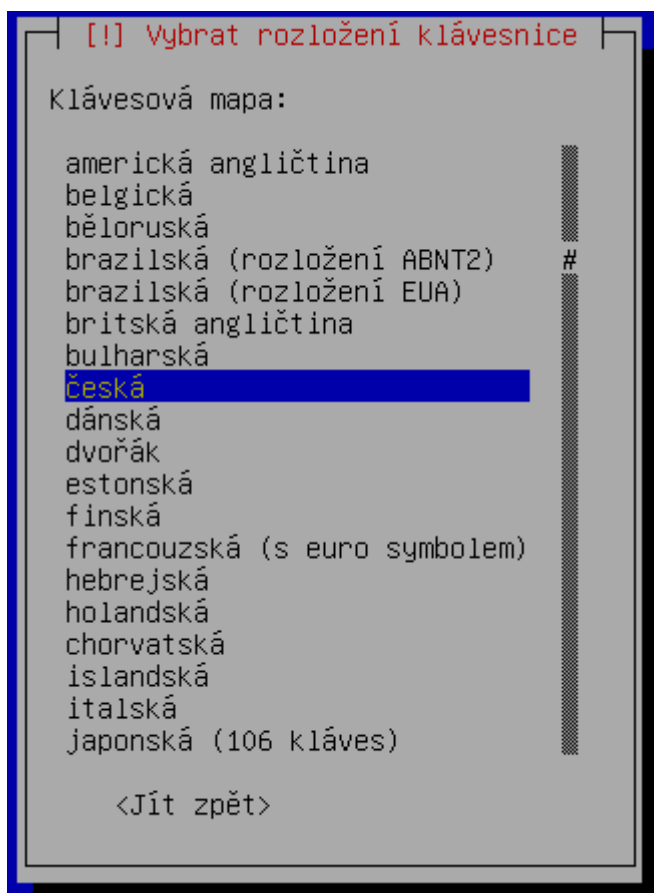
### Poznámka

Pokud chceme opustit virtuální stroj a vrátit se do rodičovského systému, stiskneme kombinaci kláves **CTRL+ ALT**.

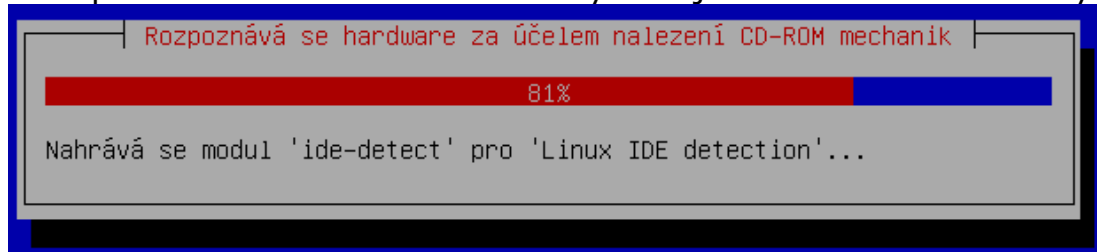
Po nabořování z DVD se zobrazí nabídka pro výběr jazyka. Vybereme češtinu.



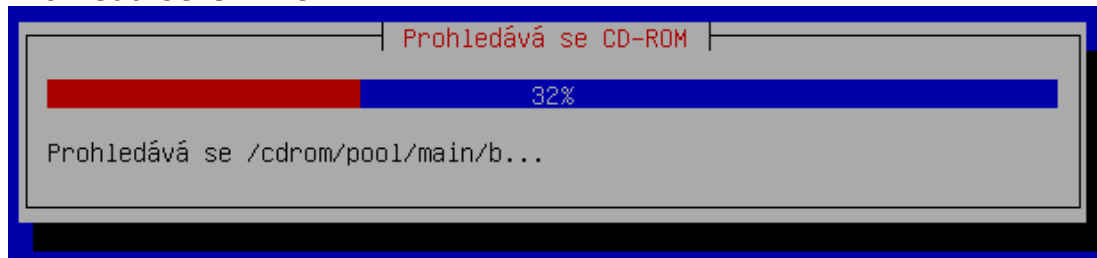
Dále vybereme rozložení klávesnice. Zvolíme volbu „česká“.



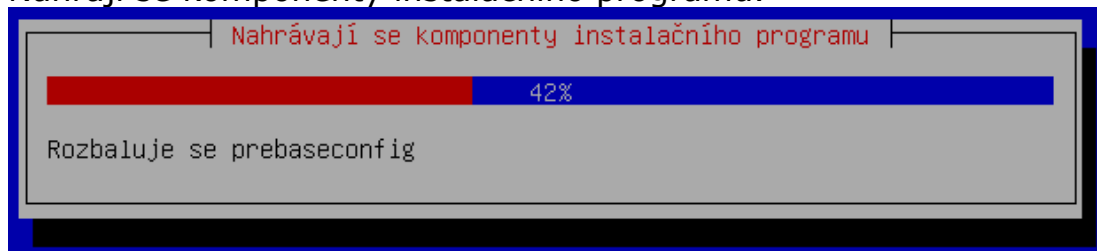
Dále proběhne detekce hardwaru. Vyhledají se CD-ROM mechaniky.



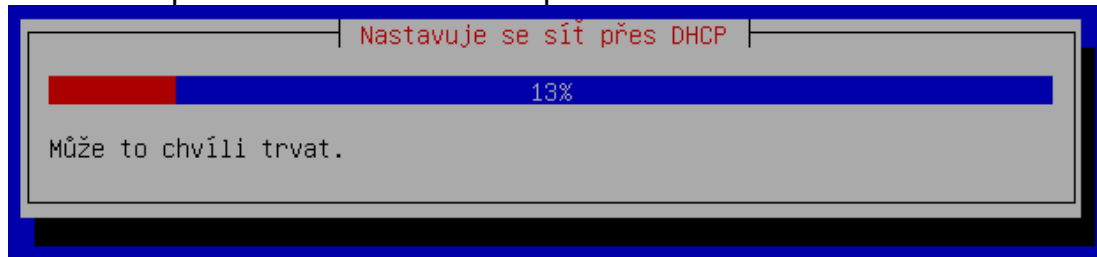
Prohledá se CD-ROM.



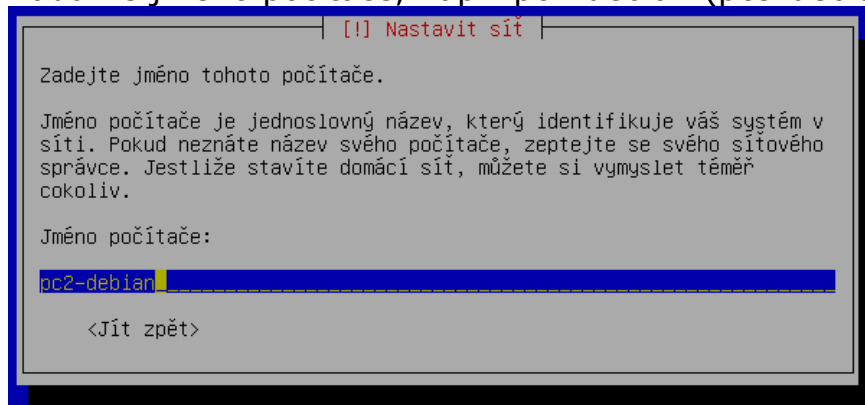
Nahrávají se komponenty instalačního programu.



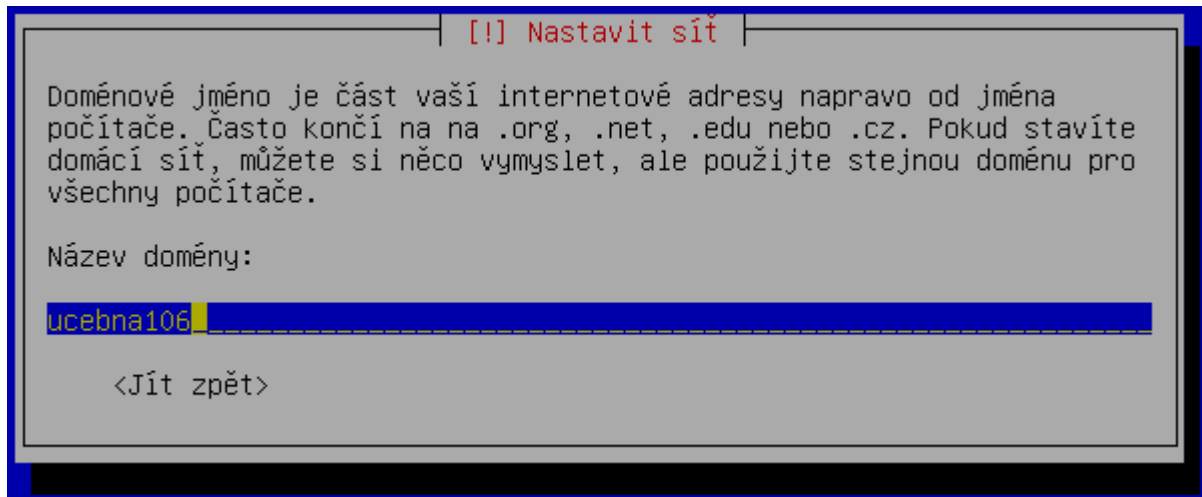
Proběhne pokus o nastavení sítě přes DHCP.



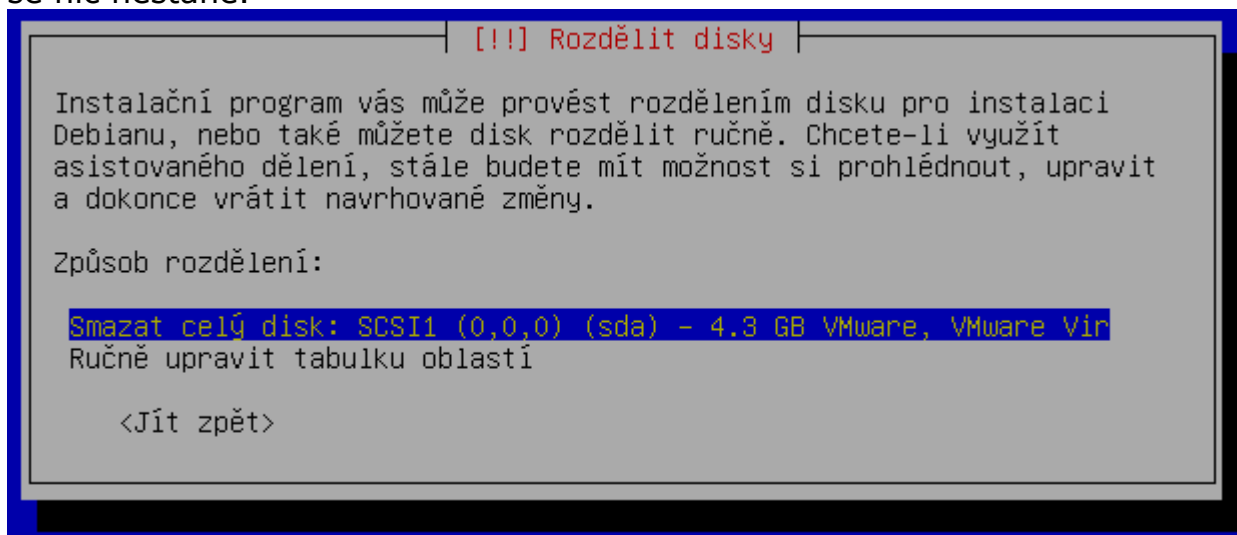
Zadáme jméno počítače, např. pc2-debian (pc3-debian atd.).



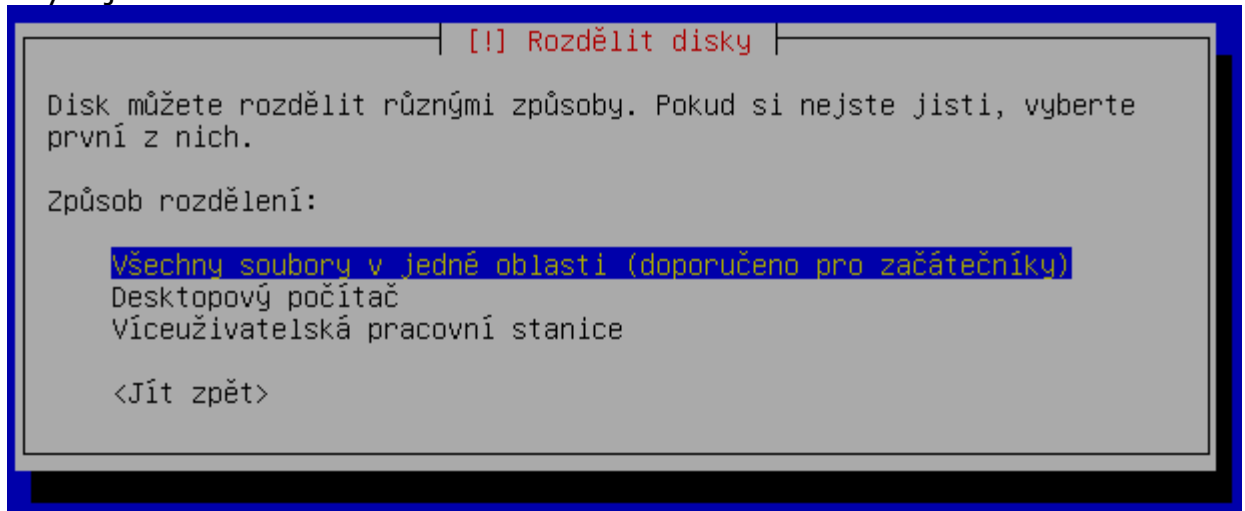
Zadáme název domény – ucebna106.



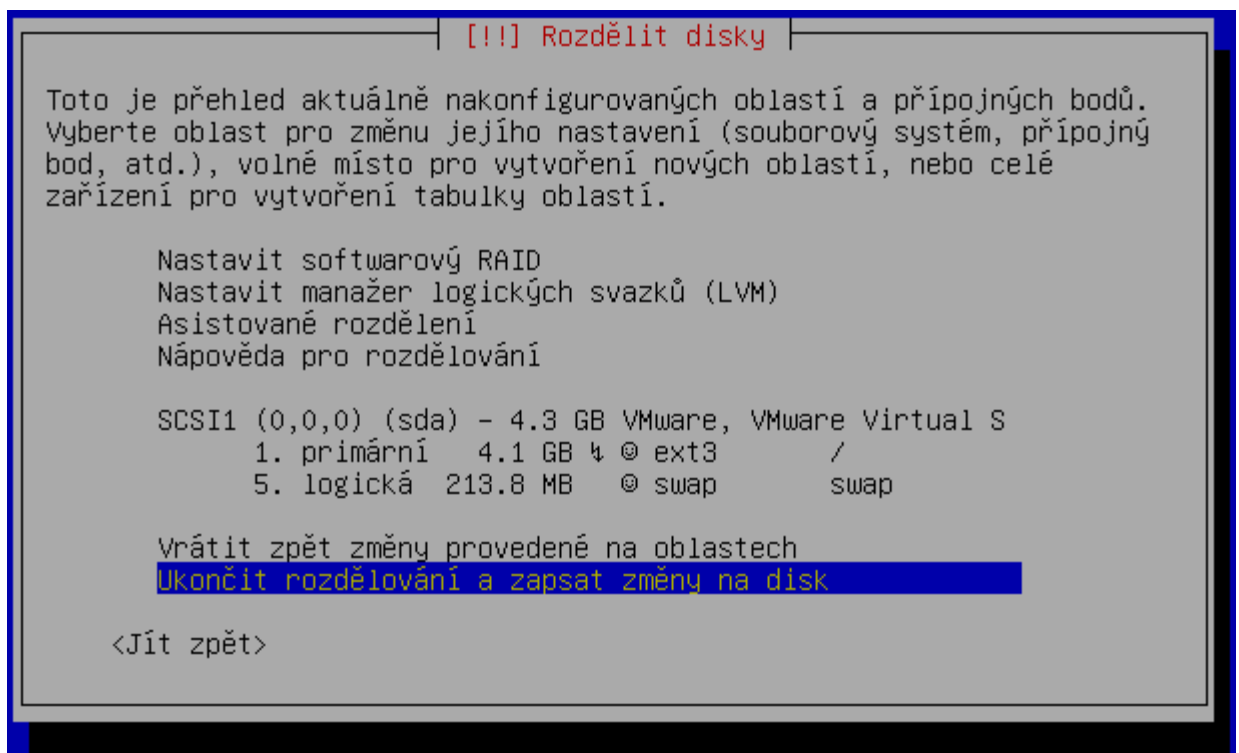
Zvolíme první možnost – Smazat celý disk. Touto volbou se smaže přidělený virtuální prostor na disku, s rodičovským operačním systémem se nic nestane.



Vybereme způsob rozdělení disku. Zvolme první možnost – Všechny soubory v jedné oblasti.

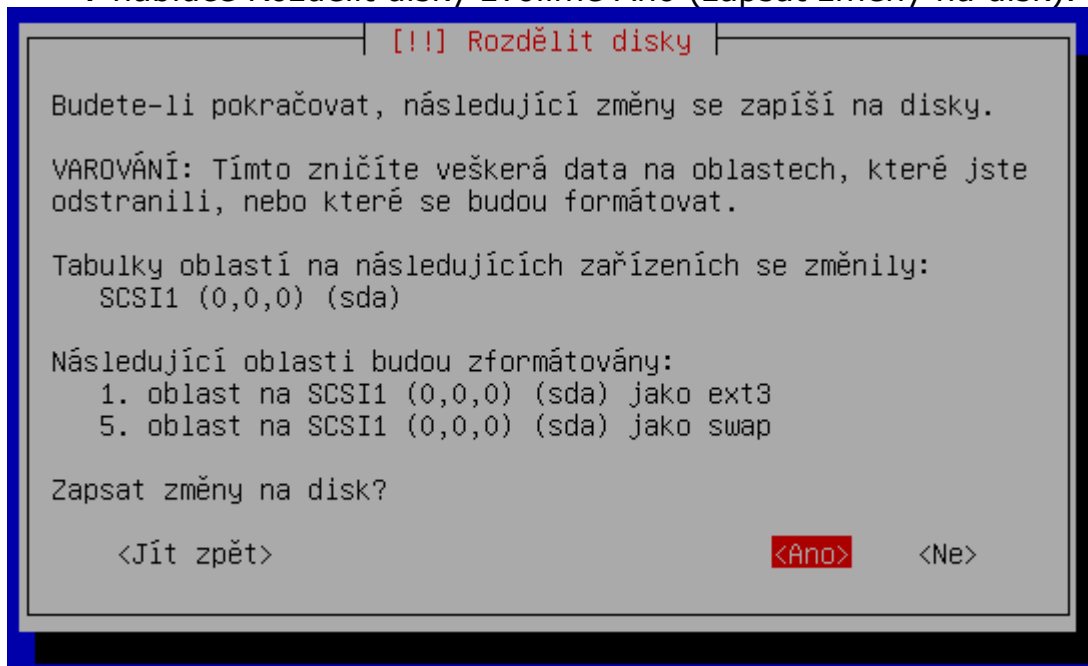


Zobrazí se přehled rozdělení disku. Vytvořil se primární oddíl o velikosti cca 4 GB se souborovým systémem ext3 a odkládací prostor o velikosti cca 200 MB – swap.

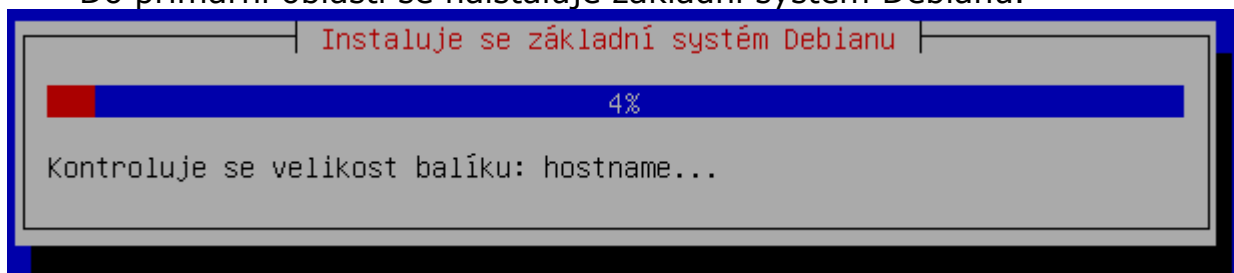


Zvolíme položku Ukončit rozdělování a zapsat změny na disk.

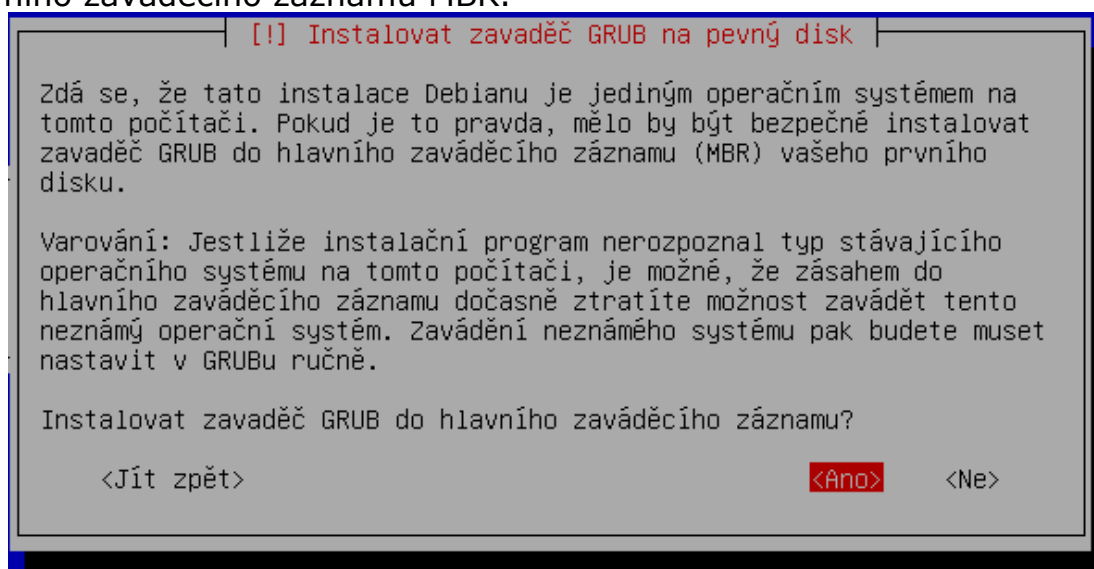
V nabídce Rozdělit disky zvolíme Ano (zapsat změny na disk).



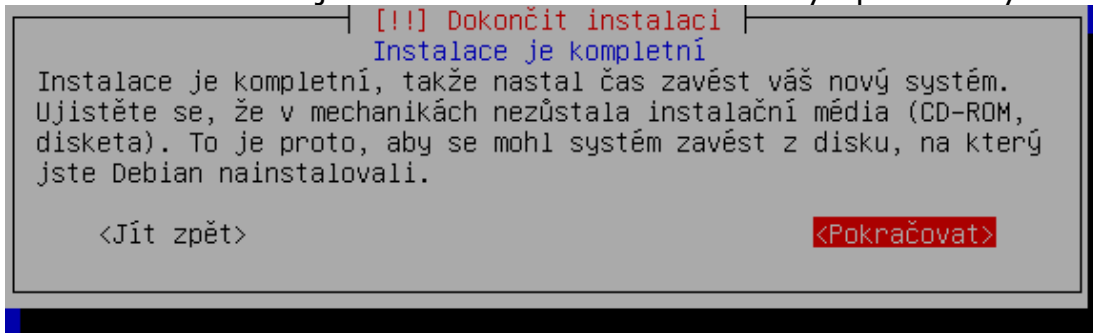
Do primární oblasti se nainstaluje základní systém Debianu.



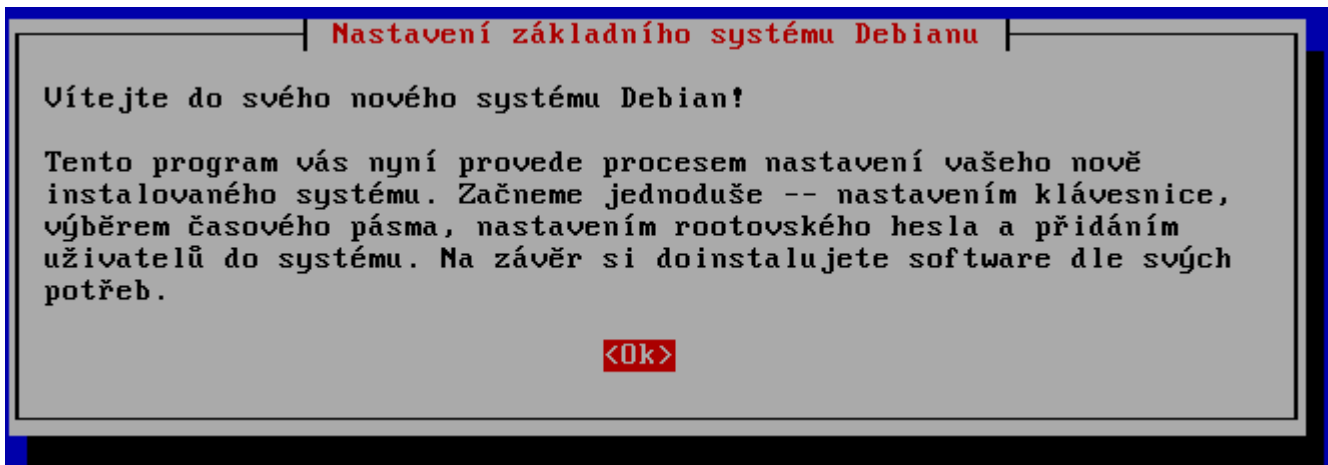
V další nabídce zvolíme, že chceme instalovat zavaděč GRUB do hlavního zaváděcího záznamu MBR.



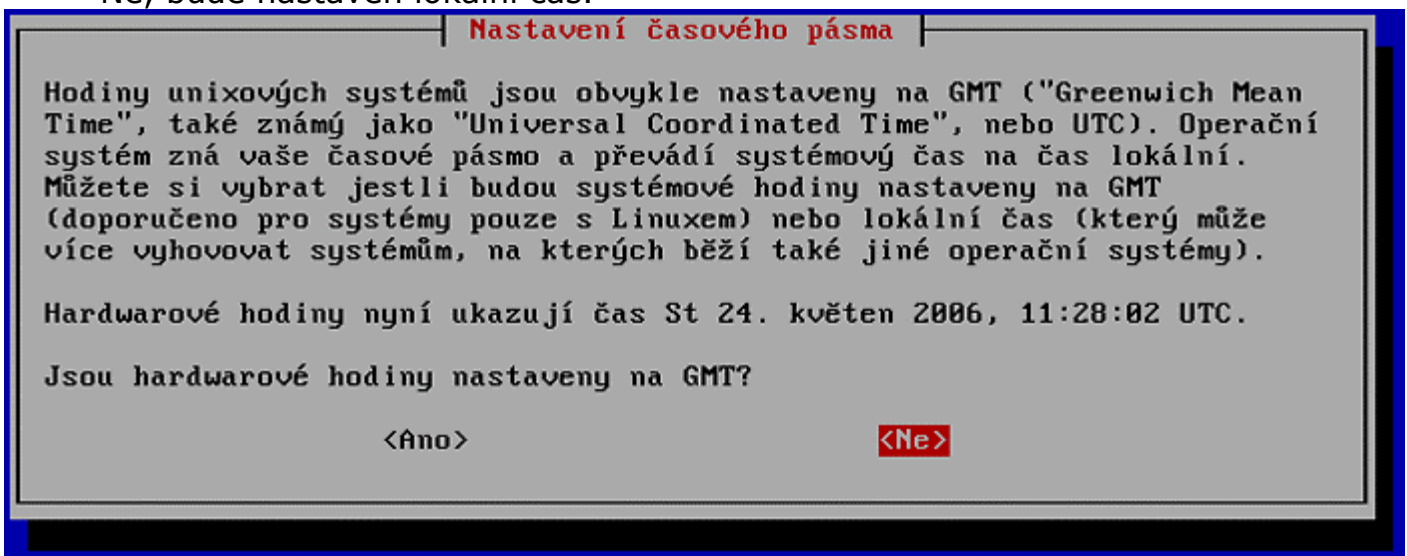
Z mechaniky vyjmeme instalační DVD a instalaci dokončíme volbou **Pokračovat**. Následně již naběhne nově nainstalovaný operační systém.



Průvodce nás provede základním nastavením Debianu. Nastavíme **klávesnici, časové pásmo, rootovské heslo, přidáme další uživatele do systému**.

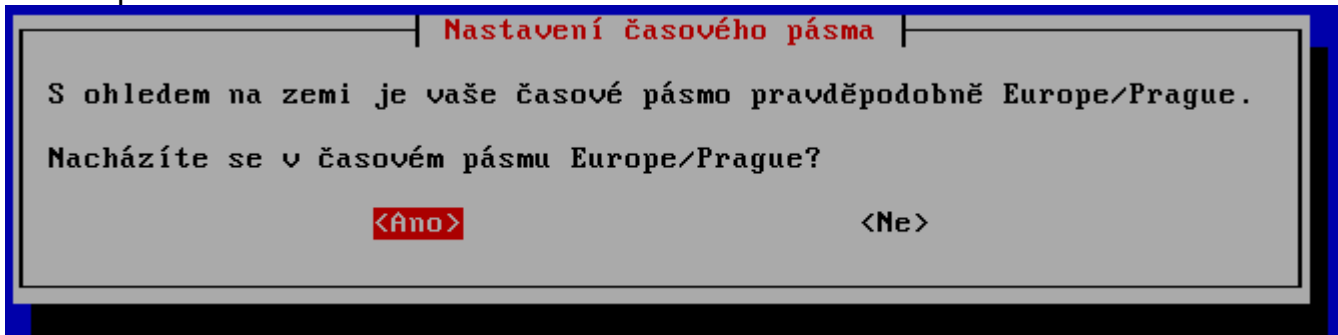


Na otázku „Jsou hardwarové hodiny nastaveny na GMT“ odpovězte **Ne**, bude nastaven lokální čas.

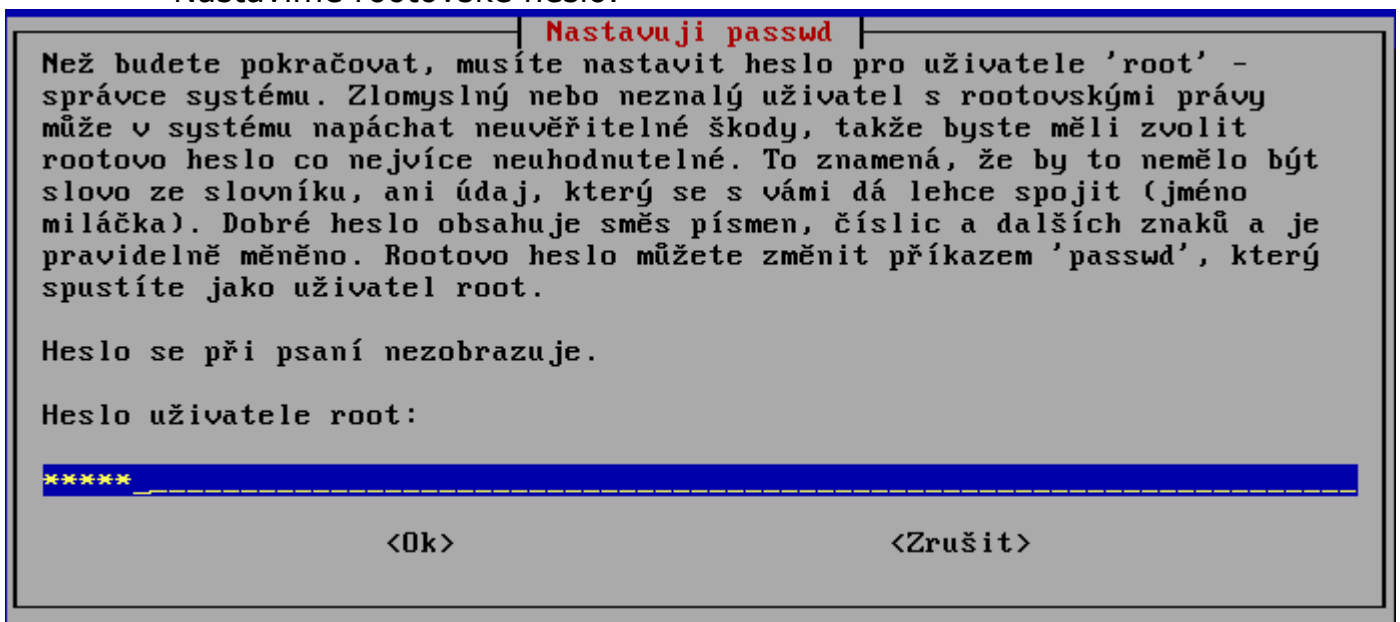




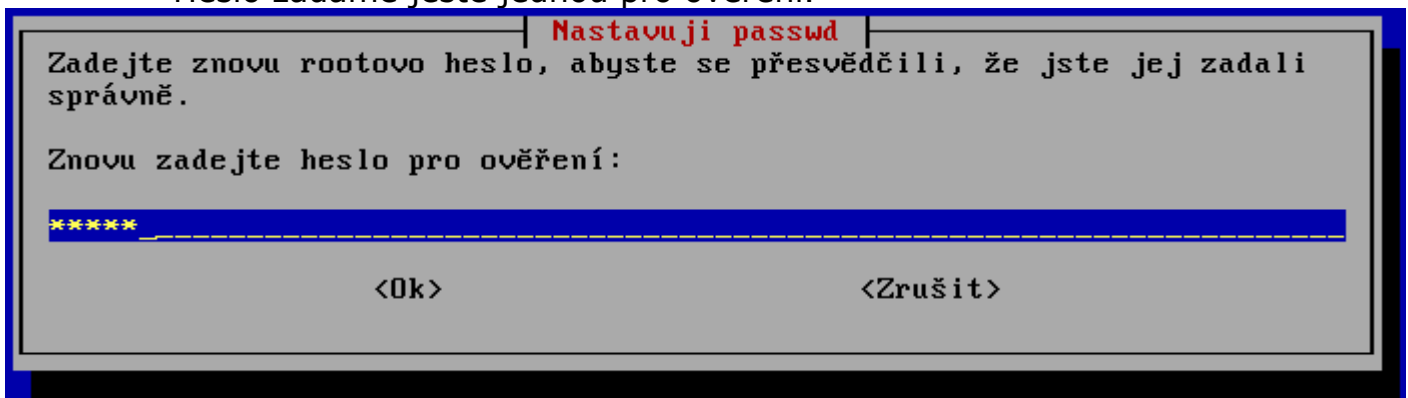
Na otázku „Nacházíte se v časovém pásmu Europe/Prague?“ odpovíme Ano.



Nastavíme rootovské heslo.



Heslo zadáme ještě jednou pro ověření.



Dále vytvoříme nového uživatele, který slouží k běžné neadministrátorské práci.

**Nastavuji passwd**

Nyní bude vytvořen uživatelský účet, který slouží k běžné (neadministrátorské) práci.

Zadejte celé jméno nového uživatele:

student

<Ok> <Zrušit>

Zadáme heslo pro tohoto nového uživatele.

**Nastavuji passwd**

Dobré heslo obsahuje směs písmen, číslic a dalších znaků a je pravidelně měněno.

Zadejte heslo pro nového uživatele:

\*\*\*\*\*

<Ok> <Zrušit>

Zadáme heslo ještě jednou pro ověření.

**Nastavuji passwd**

Zadejte prosím stejné heslo, abyste si ověřili, že jste jej zadali správně.

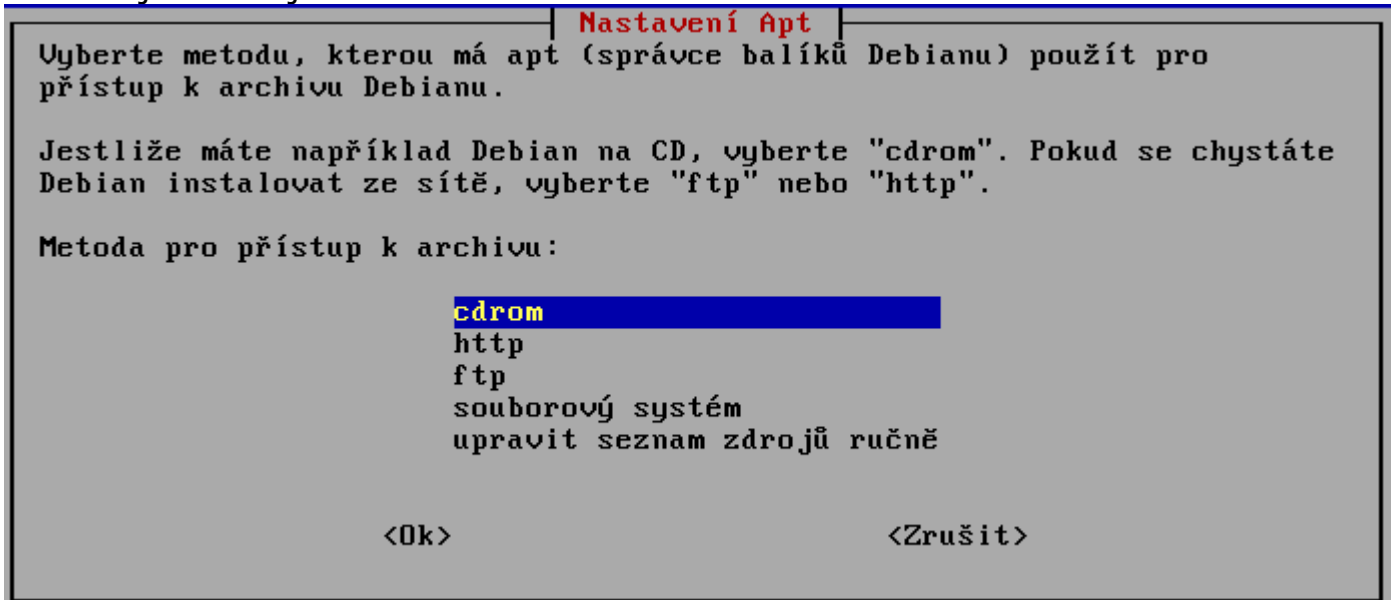
Znovu zadejte heslo pro ověření:

\*\*\*\*\*

<Ok> <Zrušit>

V nastavení Apt vybereme metodu, kterou má Apt – správce balíků Debianu – použít pro přístup k archívu.

Vybereme **cdrom** a případně následně i **ftp** a **http**, pokud máme nějaké zdroje na síti.



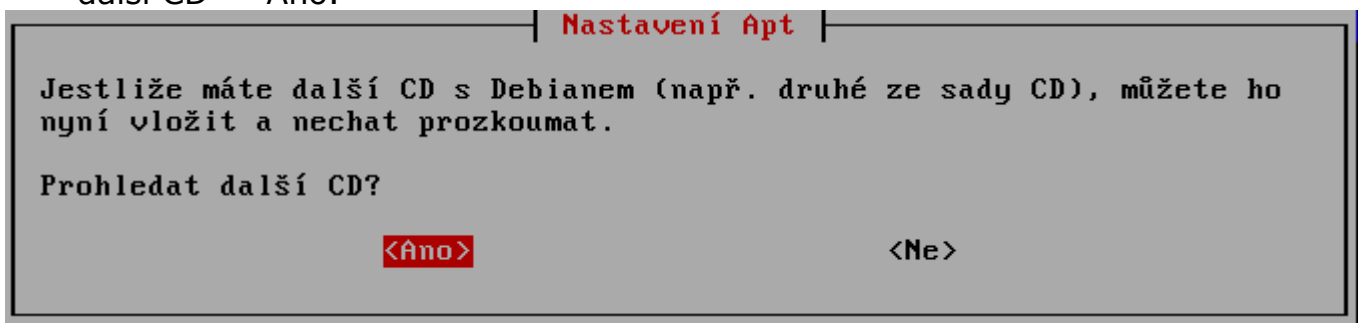
### ☞ Poznámka

Pokud chceme v budoucnu zdroje zaktualizovat, spustíme příkaz **apt-setup** pro konfiguraci Apt.

Prozkoumá se první instalační DVD.

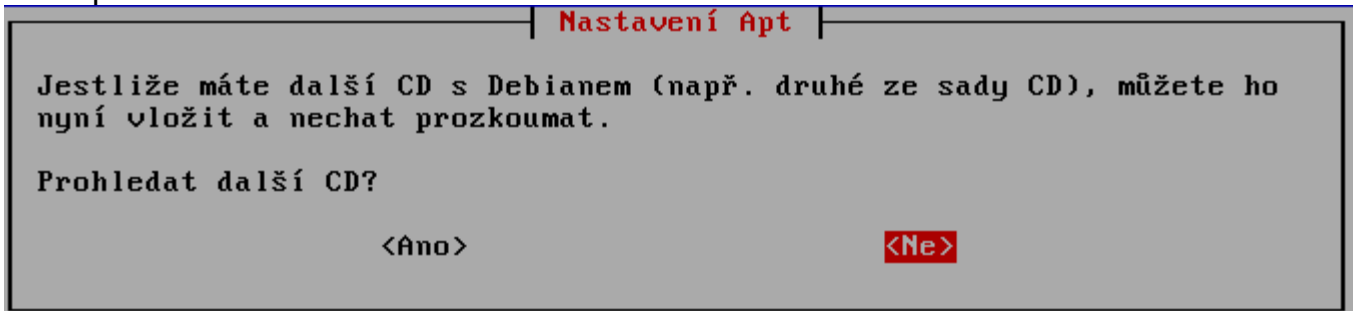
```
Zkoumá se CD, bude to chvíli trvat.
Using CD-ROM mount point /cdrom/
Identifying.. [ca516dca38db4d7175e5e976215935ff-2]
Scanning Disc for index files.. _
```

Do mechaniky vložíme druhé instalační DVD a zvolíme „Prohledat další CD“ – Ano.



```
Zkoumá se CD, bude to chvíli trvat.
Using CD-ROM mount point /cdrom/
Identifying.. [292bfddab46e89a674ff592cc76ff195-2]
Scanning Disc for index files.. _
```

Další DVD zkoumat nebudeme, proto na otázku „Prohledat další CD“ odpovíme Ne.



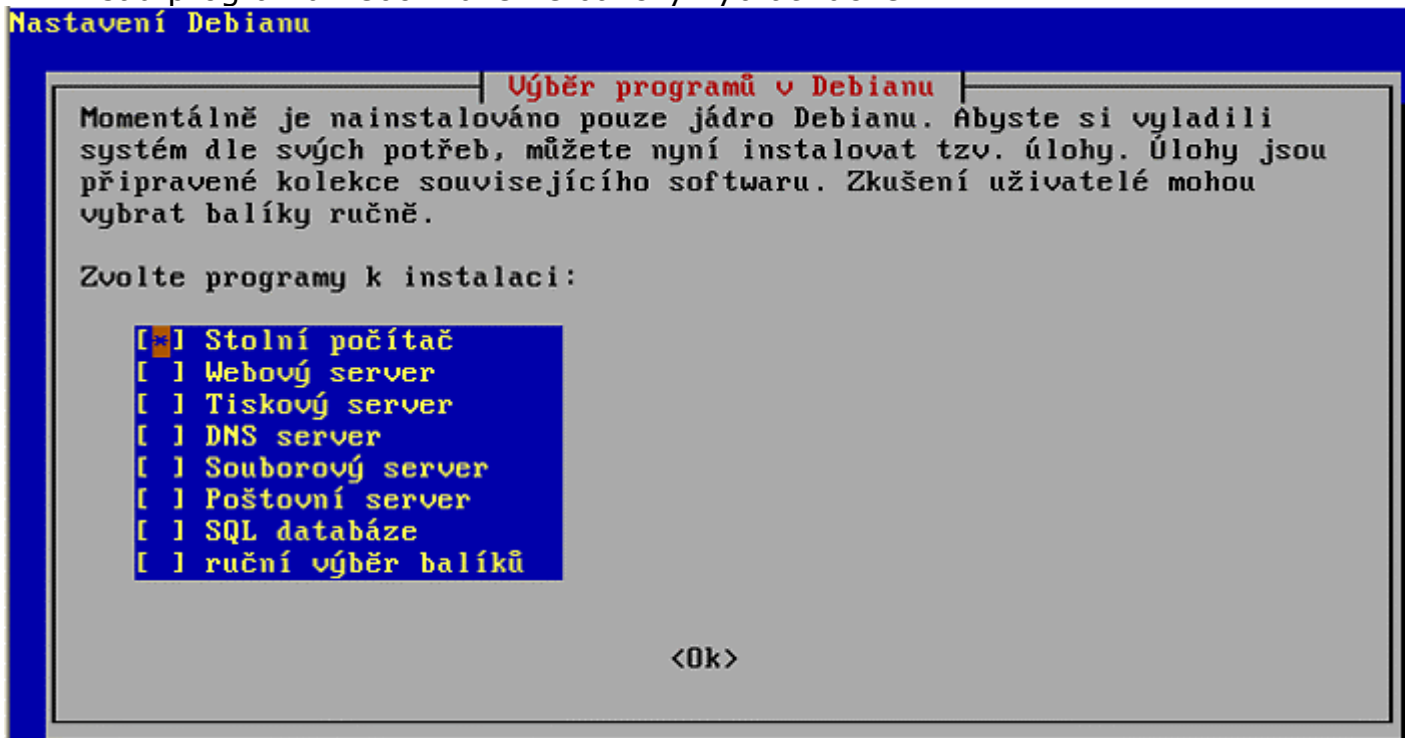
```

Čtu seznamy balíků... Hotovo
Vytvářím strom závislostí... Hotovo
Následující NOVÉ balíky budou nainstalovány:
 mdetect
0 aktualizováno, 1 nově instalováno, 0 k odstranění a 13 neaktualizováno.
Potřebuji stáhnout 0B/15,8kB archívů.
Po rozbalení bude na disku použito dalších 77,8kB.
0%[Pracujilcdrom: open failed.
Úměna média: Vložte disk nazvaný
'Debian GNU/Linux 3.1 r0a _Sarge_ - Official i386 Binary-1 (20050607)'
do mechaniky '/cdrom/' a stiskněte enter

```

System si načte seznam balíčků, vytvoří strom závislostí a bude chtít nainstalovat mdetect. Vložíme první instalační DVD a stiskneme Enter.

V tuto chvíli máme nainstalováno pouze jádro Debianu. Pro doinstalování dalších programů si vybereme buď některou z nabízených sad programů nebo můžeme balíčky vybrat ručně.



Vybereme sadu **Stolní počítač**.

V následujících několika desítkách minut bude instalován potřebný software. Programy budou také stahovány ze sítě.

```

gocr gok gsfontr-x11 gstreamer0.8-plugins gthumb gtk-engines-industrial
gtk-engines-smooth gtk-smooth-themes imlib1 jackd kdm kompare ktalkd
libatk1.0-data libcompress-zlib-perl libft-perl libglib2.0-data
libgnome2-perl libhtml-format-perl libio-socket-ssl-perl libmikmod2
libpaper-utils libpt-plugins-alsa libpt-plugins-avc libpt-plugins-dc
libpt-plugins-v4l librsvg2-bin libsasl2-modules libwmf-bin lilo-config
lisa lzop ncompress pcsd perl-doc perl-suid perlmagick psfontmgr
pstoedit python2.3-cjkcodecs python2.3-iconvcodec
python2.3-japanese-codecs python2.3-korean-codecs qca-tls rdesktop rpm
sketch spamassassin tetex-bin ttf-kochi-gothic ttf-kochi-mincho unzip
vorbis-tools wbritish wdg-html-reference x-ttcidfont-conf xfonts-konsole
xli xloadimage zip zoo
11 balíků aktualizováno, 731 nově instalováno, 0 k odstranění a 2 neaktualizováno.
Potřebuji stáhnout 182MB/514MB archivů. Po rozbalení bude použito 1525MB.
Zapisuji rozšířené stavové informace... Hotovo
Mám:1 http://security.debian.org stable/updates/main bsutils 1:2.12p-4sarge1 [65,6kB]
Mám:2 http://security.debian.org stable/updates/main mount 2.12p-4sarge1 [139kB]
Mám:3 http://security.debian.org stable/updates/main perl-base 5.8.4-8sarge3 [752kB]
Mám:4 http://security.debian.org stable/updates/main tar 1.14-2.1 [499kB]
Mám:5 http://security.debian.org stable/updates/main util-linux 2.12p-4sarge1 [378kB]
0% [5 util-linux 102486/378kB 27%] 96,7kB/s 31m2s_

```

Po stažení a nainstalování programů budeme moci nastavit parametry. Nastavíme **xserver-xfree86**. Vybereme ovladač X serveru – Vesa.

```

Nastavuji xserver-xfree86
Aby grafické rozhraní X Window System pracovalo správně, je potřeba vybrat ovladač grafické karty.

Ovladače se obvykle nazývají podle výrobce čipové sady, rodiny čipových sad, nebo podle konkrétního modelu.

Vyberte ovladač X serveru.

siliconmotion ↑
sis █
tdfx █
tga █
trident #
tseng █
vesa █ ↓

<Ok>

```

Povolíme počítači rozpoznat monitor.

**Nastavuji xserver-xfree86**

Mnoho monitorů (včetně LCD) a grafických karet podporují komunikační protokol, kterým může počítač zjistit technické parametry monitoru. Pokud váš monitor i grafická karta hovoří stejným dialektem tohoto protokolu, budou následující otázky předzodpovězeny.

Jestliže máte grafickou kartu NVidia, raději tuto možnost zamítněte, protože zmíněné karty implementují DDC protokol tak bídně, že při pokusu jej použít, můžete dokonce zablokovat systém.

Pokud automatické rozpoznání monitoru selže, budete požádáni o zadání informací ručně.

Pokusit se o rozpoznání monitoru?

**<Ano>**

<Ne>

Na otázku, zda je monitor LCD zařízení, odpovíme Ano.

**Nastavuji xserver-xfree86**

Pokud je váš monitor typu LCD (což je případ naprosté většiny notebooků), měli byste odpovědět kladně.

Uživatelé klasických monitorů (CRT) by měli odpovědět záporně.

Je váš monitor LCD zařízením?

**<Ano>**

<Ne>

**Nastavuji xserver-xfree86**

Aby grafické rozhraní X Window System pracovalo správně, musí znát některé vlastnosti připojeného monitoru.

Pro volbu "simple" vám stačí znát pouze fyzickou velikost monitoru. Tím se nastaví standardní hodnoty pro typický CRT monitor této velikosti, které ale nemusí vyhovovat vysoce kvalitním monitorům. (Tato volba je nedostupná pro uživatele s LCD, protože tyto obrazovky se konfiguruji pro konkrétní rozlišení.)

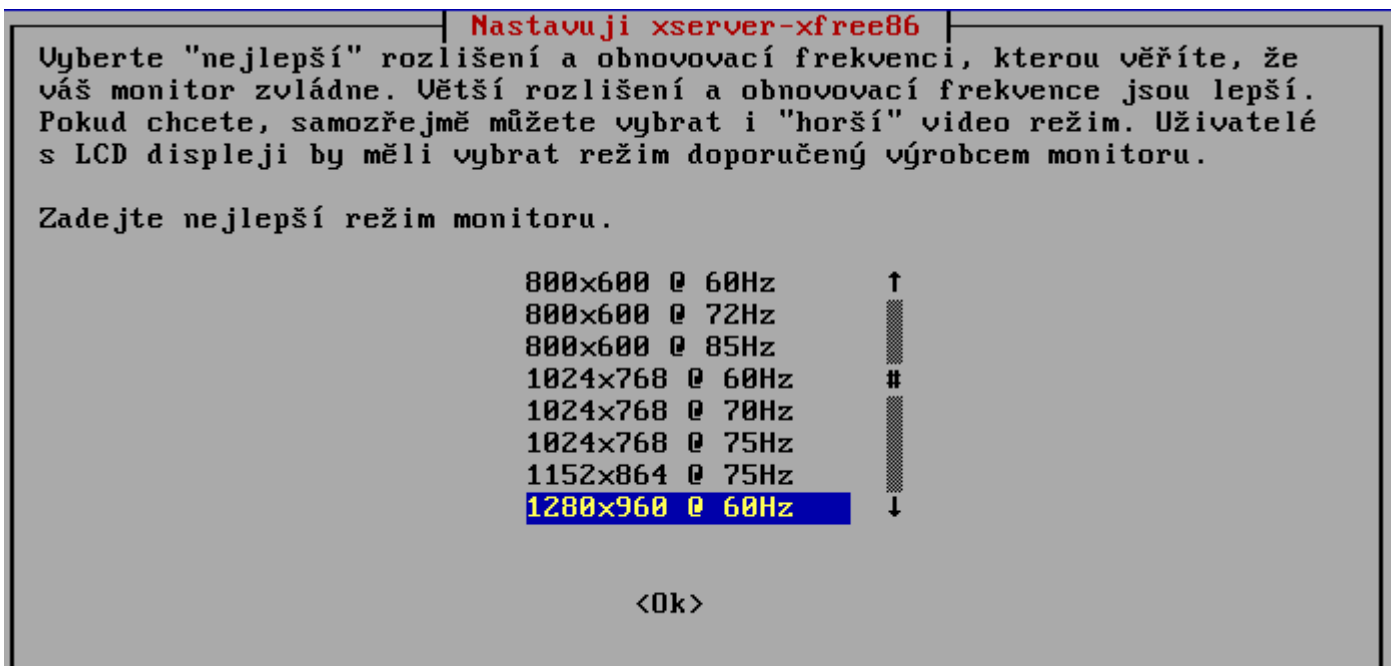
Možnost "medium" vám nabídne seznam rozlišení a obnovovacích frekvencí, např. "800x600 @ 85Hz". Měli byste si vybrat nejlepší režim, který budete používat a který zvládne váš monitor.

Volba "advanced" vás nechá přímo zadat rozsah vertikální a horizontální frekvence monitoru.

**<Ok>**

Ve způsobu výběru vlastností monitoru vybereme Medium. Budeme mít na výběr seznam rozlišení a obnovovacích frekvencí.

V možnosti Advanced bychom mohli přímo zadat rozsah vertikální a horizontální frekvence monitoru.



Po nastavení parametrů dojde k nainstalování dalších balíčků.

```

Instalují balík libscrollkeeper0.
Rozbalují libscrollkeeper0 (z .../libscrollkeeper0_0.3.14-10_i386.deb) ...
Instalují balík sgml-base.
Rozbalují sgml-base (z .../sgml-base_1.26_all.deb) ...
Instalují balík xml-core.
Rozbalují xml-core (z .../xml-core/xml-core_0.09_all.deb) ...
Instalují balík sgml-data.
Rozbalují sgml-data (z .../sgml-data_2.0.3_all.deb) ...
Instalují balík docbook-xml.
Rozbalují docbook-xml (z .../docbook-xml_4.3-1.1_all.deb) ...
Instalují balík scrollkeeper.
Rozbalují scrollkeeper (z .../scrollkeeper_0.3.14-10_i386.deb) ...
Instalují balík bug-buddy.
Rozbalují bug-buddy (z .../bug-buddy_2.8.0-3_i386.deb) ...
Instalují balík bzip2.

```

### ☞ Poznámka

Pokud bychom chtěli překonfigurovat xserver, zadáme příkaz **dpkg-reconfigure xserver-xfree86**.

V nastavení pošty zvolme „nyní nic nenastavovat“.

Nastavení Eximu v4 (exim-config)

Uberte typ nastavení, který nejlépe odpovídá vašim požadavkům.

Počítače s dynamickými IP adresami by měly být nastaveny tak, že veškerou odchozí poštu posílají jinému, "chytrému", počítači (smarthost), který se postará o její doručení. Můžete si vybrat, zda na takovém systému chcete poštu přijímat, nebo ji zakázat (s výjimkou uživatele root a postmaster).

Základní model poštovního nastavení:

- internetový počítač; pošta je přijímána a doručována přímo SMTP
- zasílání pošty přes chytrý počítač; příjem přes SMTP nebo fetchmail
- zasílání pošty přes chytrý počítač; žádná lokální pošta
- pouze lokální pošta; počítač není na síti
- ručně převést nastavení z Eximu verze 3
- nyní nic nenastavovat**

Na nejasně položenou otázku „Opravdu nechcete ...“ odpovězte Ano, při volbě Ne se vrátíte do předchozího okna.

Nastavení Eximu v4 (exim-config)

Dokud poštovní systém nenastavíte, bude porouchaný a nemůžete jej využívat. Nastavení samozřejmě můžete provést později ručně nebo příkazem "dpkg-reconfigure exim4-config" (obojí pod uživatelem root).

Opravdu nechcete nastavit poštovní systém?

Nastavení základního systému Debianu

Děkujeme, že jste si vybrali Debian!

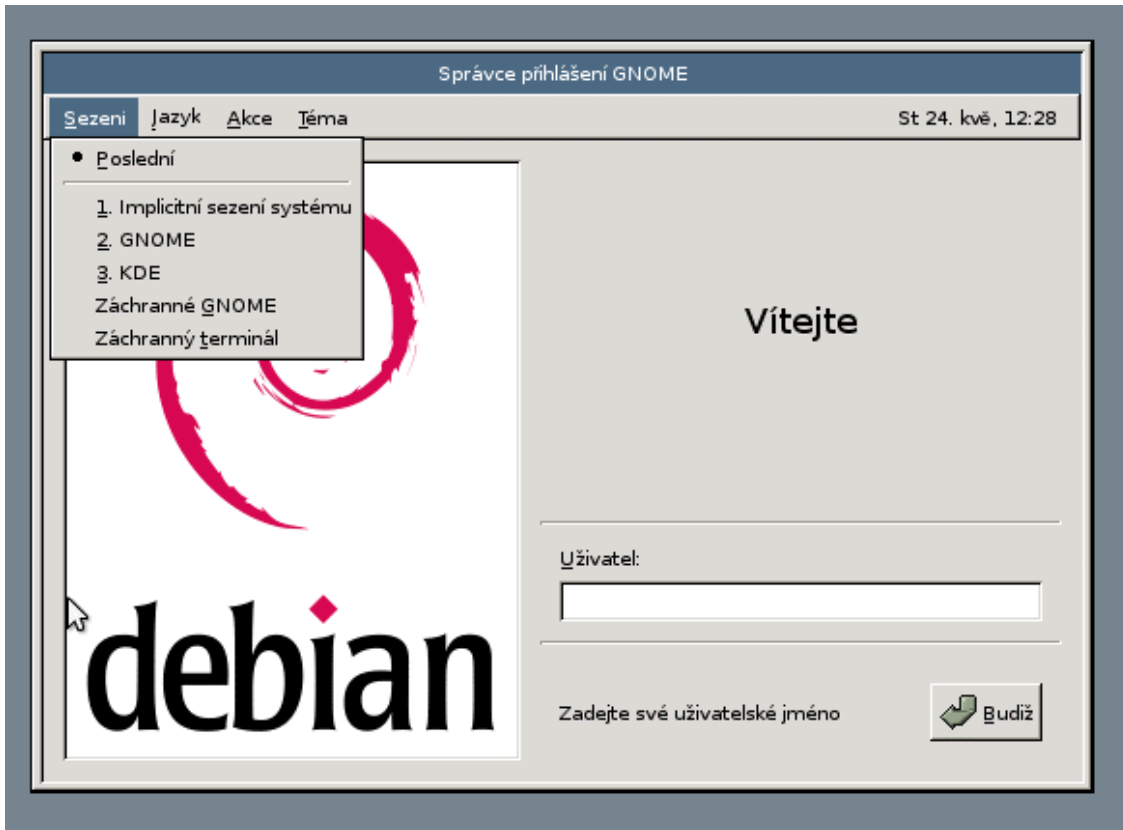
Nastavení Debianu je u konce. Nyní se můžete přihlásit na výzvě login:

Jestliže budete někdy chtít změnit volby, které jste zde nastavili, jednoduše spusťte program base-config.

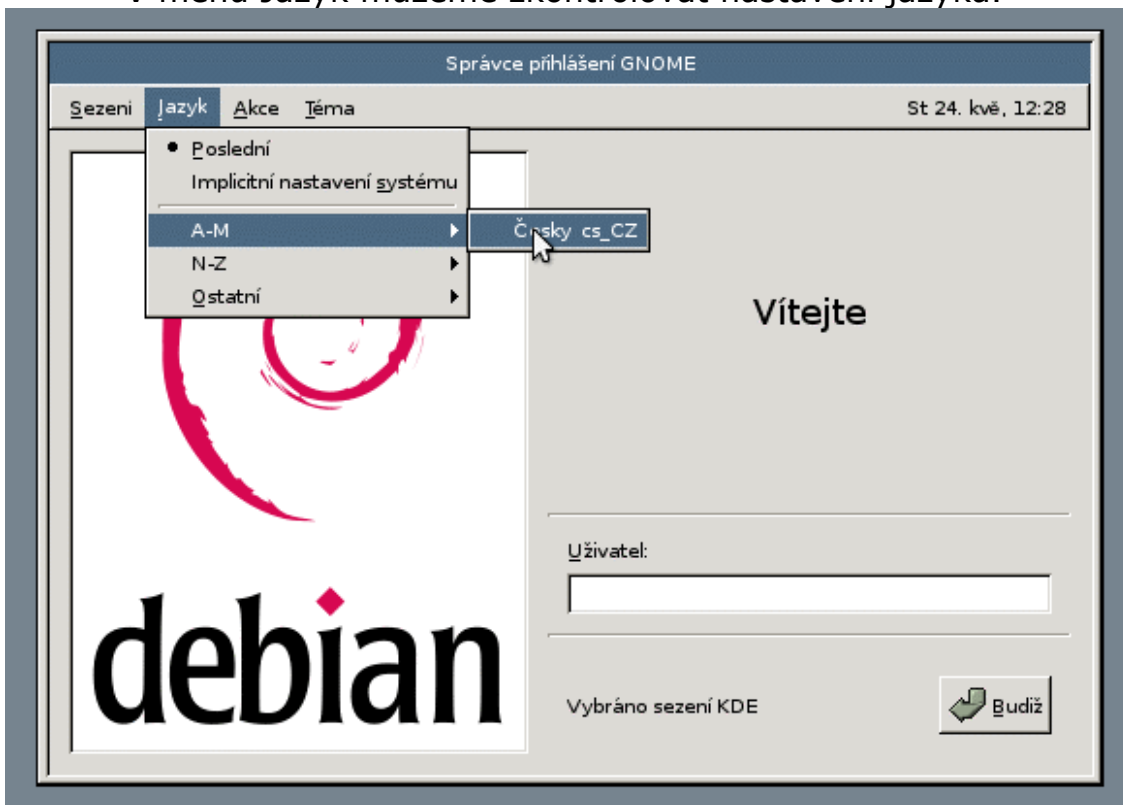
<Ok>



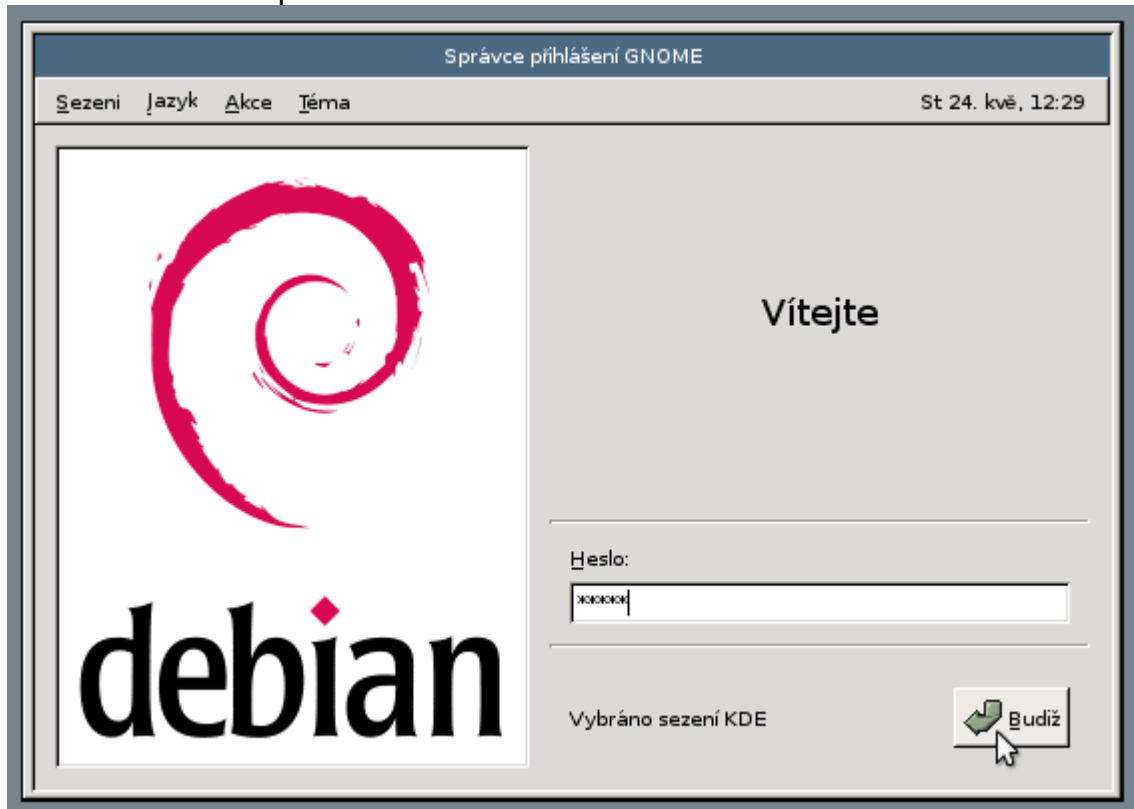
Po naběhnutí grafické konzoly můžeme vybrat v menu Sezení položku KDE.



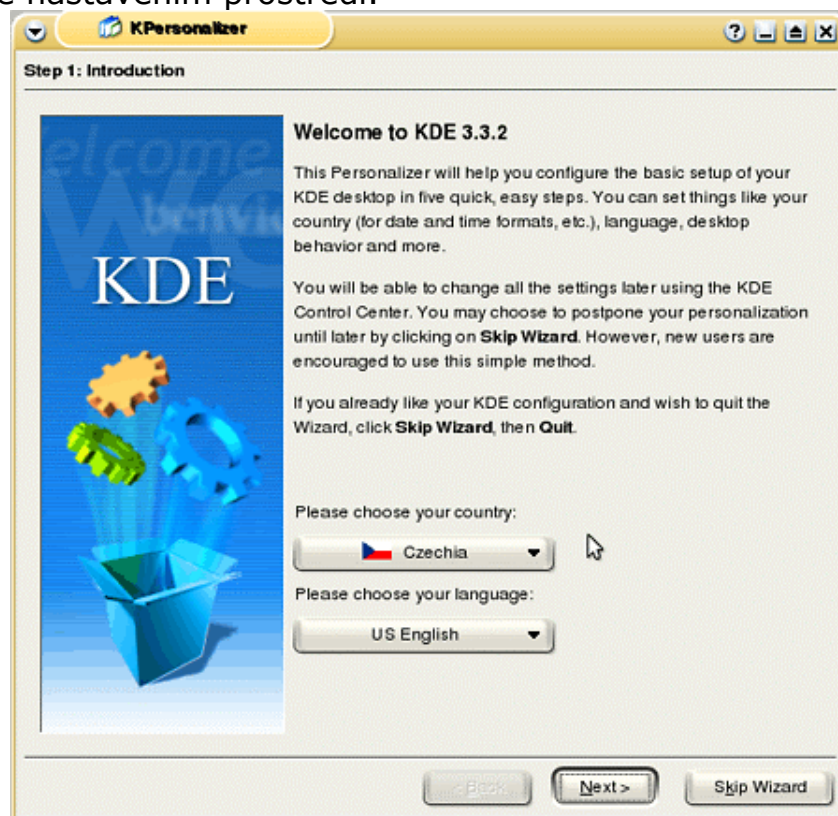
V menu Jazyk můžeme zkontrolovat nastavení jazyka.

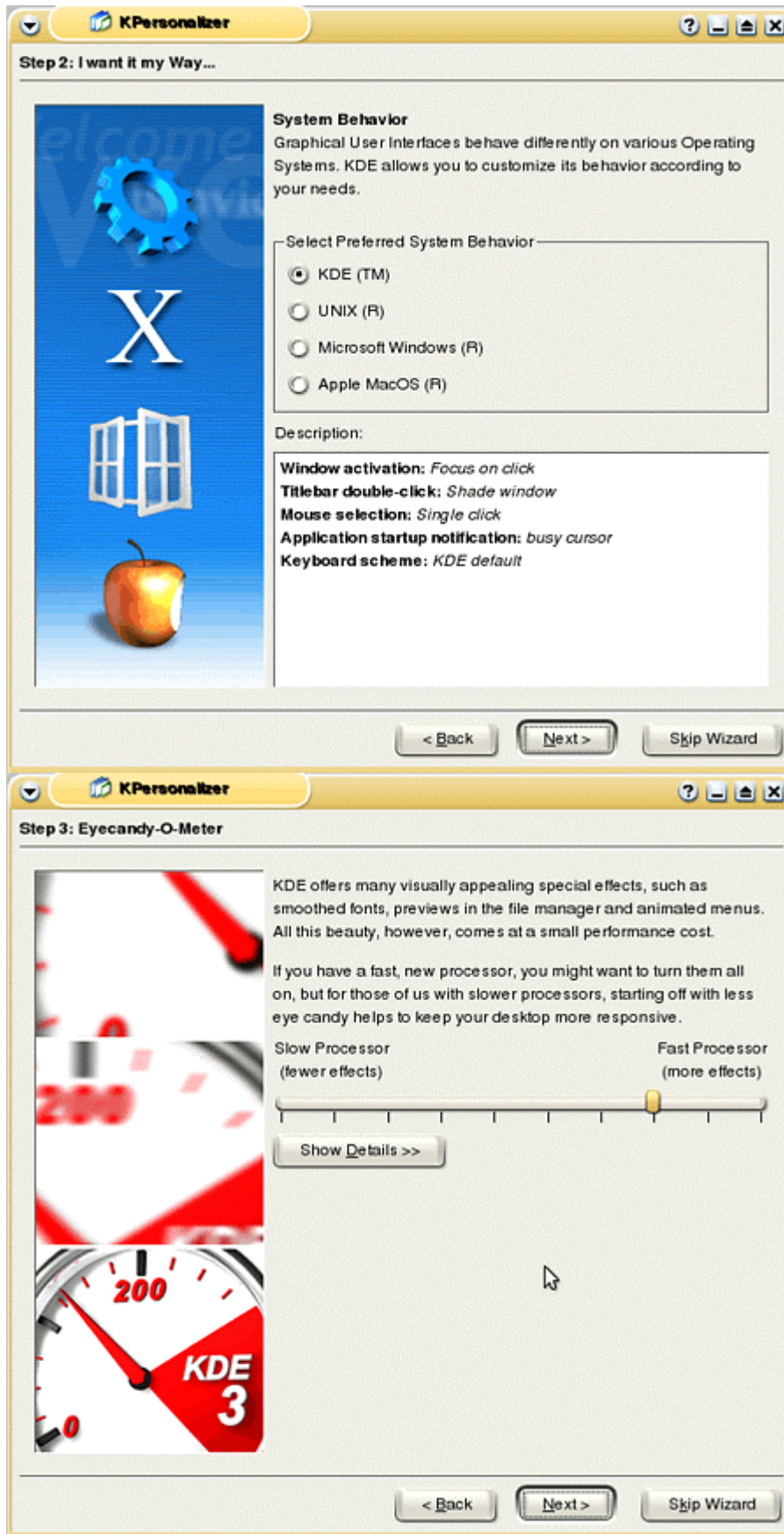


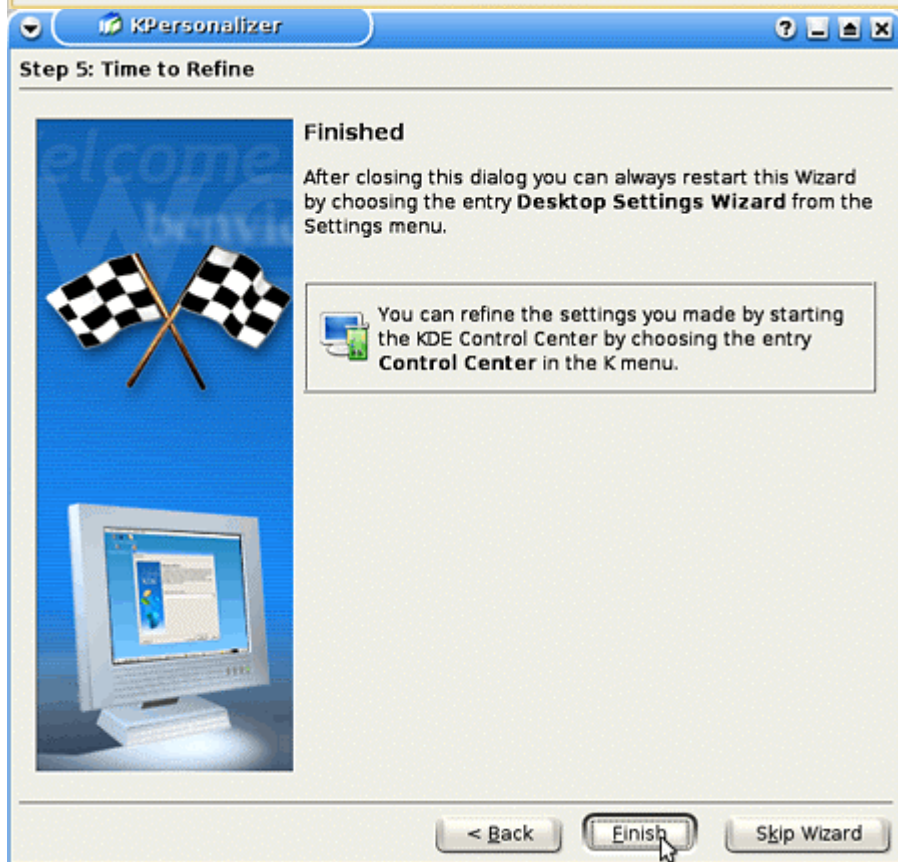
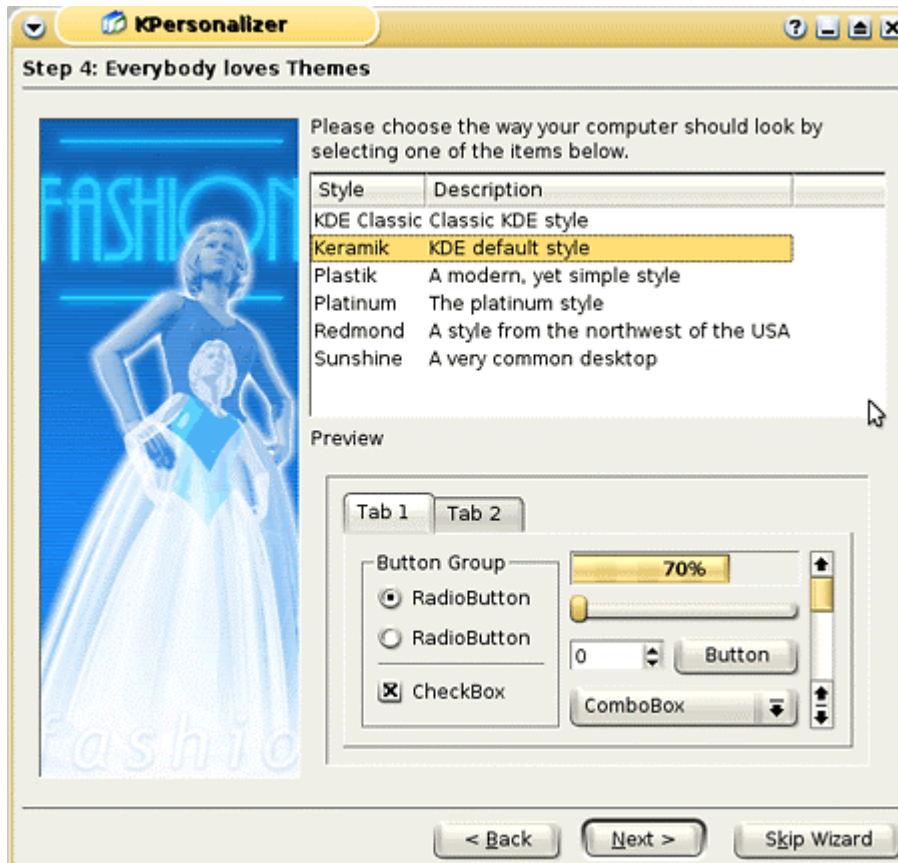
Zadáme heslo uživatele student, kterého jsme si založili pro neadministrátorskou práci.



Projdeme nastavením prostředí.







## 16.2 Konfigurace sítě

Spustíme konzolu.

Příkazem „su -“ se přehlásíme za uživatele root.

Zobrazíme si konfiguraci síťových rozhraní příkazem **ifconfig**.

```
student@pc2-debian:~$ su -
Password:
pc2-debian:~# ifconfig
eth0 Link encap:Ethernet HWaddr 00:0C:29:DF:55:A4
 inet addr:172.27.150.108 Bcast:172.27.150.255 Mask:255.255.255.0
 UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
 RX packets:130411 errors:0 dropped:0 overruns:0 frame:0
 TX packets:67906 errors:0 dropped:0 overruns:0 carrier:0
 collisions:0 txqueuelen:1000
 RX bytes:190630956 (181.7 MiB) TX bytes:4509275 (4.3 MiB)
 Interrupt:18 Base address:0x1080

lo Link encap:Local Loopback
 inet addr:127.0.0.1 Mask:255.0.0.0
 UP LOOPBACK RUNNING MTU:16436 Metric:1
 RX packets:92 errors:0 dropped:0 overruns:0 frame:0
 TX packets:92 errors:0 dropped:0 overruns:0 carrier:0
 collisions:0 txqueuelen:0
 RX bytes:6040 (5.8 KiB) TX bytes:6040 (5.8 KiB)
```

IP adresa byla přidělena dynamicky pomocí **DHCP**. Pokud ji chceme dočasně přenastavit, můžeme použít příkaz **ifconfig** s parametry.

**ifconfig** *název\_rozhraní* *IP\_adresa* **netmask** *maska* **broadcast** *IP\_broadcastu* **up/down**

Následujícím příkazem přenastavím pro síťové rozhraní **eth0** IP adresu na **172.27.106.202**, masku na **255.255.255.0** a rozhraní zapneme pomocí slovíčka **up**.

```
pc2-debian:~# ifconfig eth0 172.27.106.202 netmask 255.255.255.0 up
```

Funčnost předchozího příkazu ověříme výpisem **ifconfig**.

Vyzkoušejte, zda funguje **ping** na rodičovský počítač. Pokud jsou obě IP adresy adresami ve stejné síti, měl by ping fungovat (pokud odpovídání na ping nebrání například nastavení firewallu).

```

eth0 Link encap:Ethernet Hwaddr 00:0C:29:DF:55:A4
 inet addr:172.27.106.202 Bcast:172.27.255.255 Mask:255.255.255.0
 UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
 RX packets:130580 errors:0 dropped:0 overruns:0 frame:0
 TX packets:67913 errors:0 dropped:0 overruns:0 carrier:0
 collisions:0 txqueuelen:1000
 RX bytes:190649492 (181.8 MiB) TX bytes:4509905 (4.3 MiB)
 Interrupt:18 Base address:0x1080

lo Link encap:Local Loopback
 inet addr:127.0.0.1 Mask:255.0.0.0
 UP LOOPBACK RUNNING MTU:16436 Metric:1
 RX packets:92 errors:0 dropped:0 overruns:0 frame:0
 TX packets:92 errors:0 dropped:0 overruns:0 carrier:0
 collisions:0 txqueuelen:0
 RX bytes:6040 (5.8 KiB) TX bytes:6040 (5.8 KiB)

pc2-debian:~# ping 172.27.106.102
PING 172.27.106.102 (172.27.106.102) 56(84) bytes of data.
64 bytes from 172.27.106.102: icmp_seq=1 ttl=128 time=0.182 ms
64 bytes from 172.27.106.102: icmp_seq=2 ttl=128 time=0.185 ms
64 bytes from 172.27.106.102: icmp_seq=3 ttl=128 time=2.85 ms
64 bytes from 172.27.106.102: icmp_seq=4 ttl=128 time=0.143 ms

```

Vyzkoušejte také opačný **ping**, z rodičovského systému na virtuální stroj s nainstalovaným linuxem.

```

C:\WINDOWS\system32\cmd.exe - ping 172.27.106.202
Microsoft Windows XP [Verze 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\student>ping 172.27.106.202

Příkaz PING na 172.27.106.202 s délkou 32 bajtů:

Odpověď od 172.27.106.202: bajty=32 čas < 1ms TTL=64
Odpověď od 172.27.106.202: bajty=32 čas < 1ms TTL=64
-
64 bytes from 172.27.106.102: icmp_seq=12 ttl=128 time=0.106 ms
64 bytes from 172.27.106.102: icmp_seq=13 ttl=128 time=0.130 ms
64 bytes from 172.27.106.102: icmp_seq=14 ttl=128 time=0.160 ms
64 bytes from 172.27.106.102: icmp_seq=15 ttl=128 time=0.141 ms
64 bytes from 172.27.106.102: icmp_seq=16 ttl=128 time=0.146 ms
64 bytes from 172.27.106.102: icmp_seq=17 ttl=128 time=4.69 ms

```

Restartujte OS a následně si zobrazte příkazem **ifconfig**, zda si OS nepamatuje nastavení rozhraní, které jste předtím příkazem **ifconfig** nastavili.

Zopakujte si instalování vybraného balíčku pomocí **aptitude**. Nainstalujte **midnight commander**.

```
pc2-debian:~# aptitude
```

```

Actions Undo Package Search Options Views Help
f10: Menu ? : Help q: Quit u: Update g: Download/Install/Remove Pkgs
aptitude 0.2.15.9 Will use 5328kB of disk space DL Size: 2155kB
[?] mc +5328kB <none> 1:4.6.0-4.
p mconfig <none> 0.20-3.1
p mcrypt <none> 2.6.4-3
p mdbtools <none> 0.5.99.0.6
p mdns-scan <none> 0.4-1
p memtester <none> 2.93.1-2
p mencal <none> 2.3-4
p mergelog <none> 4.5.1-7
p metar <none> 20050103.1
p mifluz <none> 0.24.0-9
midnight commander - a powerful file manager
GNU Midnight Commander is a text-mode full-screen file manager. It uses a two panel
interface and a subshell for command execution. It includes an internal editor with
syntax highlighting and an internal viewer with support for binary files. Also
included is Virtual Filesystem (VFS), that allows files on remote systems (e.g. FTP,
SSH, SMB servers) and files inside archives to be manipulated like real files.

```

Spustte **mc**.

```

pc2-debian:~# aptitude
Selecting previously deselected package mc.
(Reading database ... 87784 files and directories currently installed.)
Unpacking mc (from .../mc_4.6.0-4.6.1-pre3-3_i386.deb) ...
Setting up mc (4.6.0-4.6.1-pre3-3) ...

Press return to continue.

pc2-debian:~# mc

```

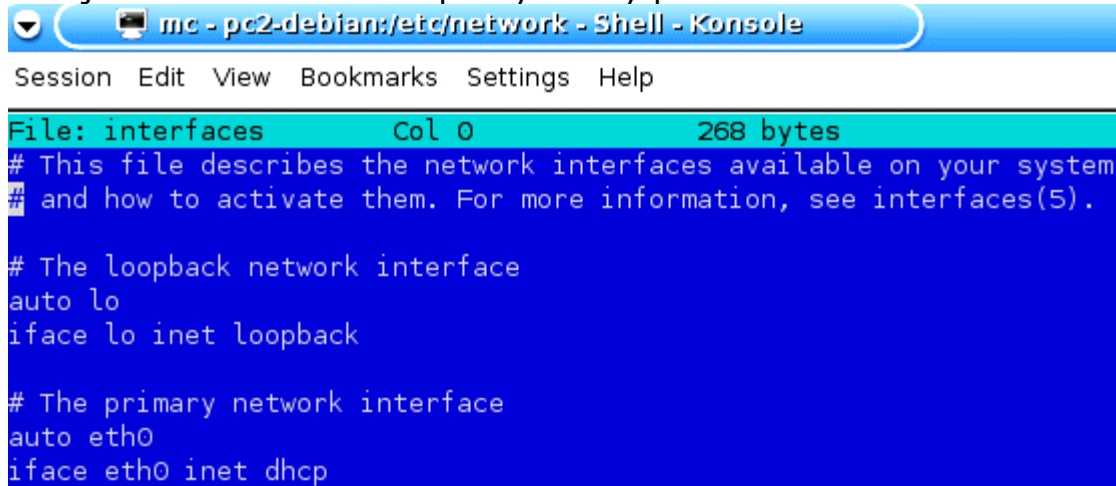
Najděte soubor **interfaces** v adresáři **/etc/network/** a zobrazte si jeho obsah.

```

< /etc/network v>
 Name | Size | MTime
|-----|-----|-----|
| ../ | UP- -DIR|
| /if-down.d | 4096 | May 2 2005|
| /if-post-down.d | 4096 | May 2 2005|
| /if-pre-up.d | 4096 | May 2 2005|
| /if-up.d | 4096 | May 2 2005|
| /run | 4096 | May 24 13:27|
| interfaces | 268 | May 24 13:23|
| options | 45 | May 24 13:25|

```

V tomto souboru jsou zapsány informace o síťových rozhraních. V našem případě je zde interní **loopback** rozhraní a síťové rozhraní **eth0**, kterému jsou síťové informace poskytovány pomocí DHCP.



```

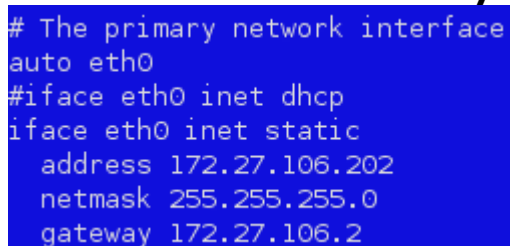
File: interfaces Col 0 268 bytes
This file describes the network interfaces available on your system
and how to activate them. For more information, see interfaces(5).

The loopback network interface
auto lo
iface lo inet loopback

The primary network interface
auto eth0
iface eth0 inet dhcp

```

Zakomentujte řádek s **dhcp** a vytvořte další řádky, které budou nastavovat rozhraní eth0 **staticky**.



```

The primary network interface
auto eth0
#iface eth0 inet dhcp
iface eth0 inet static
 address 172.27.106.202
 netmask 255.255.255.0
 gateway 172.27.106.2

```

Po uložení souboru si vypište informace o síťových rozhraních pomocí **ifconfig**. Protože nedošlo k restartu služby nebo celkovému restartu systému, nejsou změny provedené v konfiguračním souboru ještě akceptovány. Používají se staré informace získané z dhcp serveru.

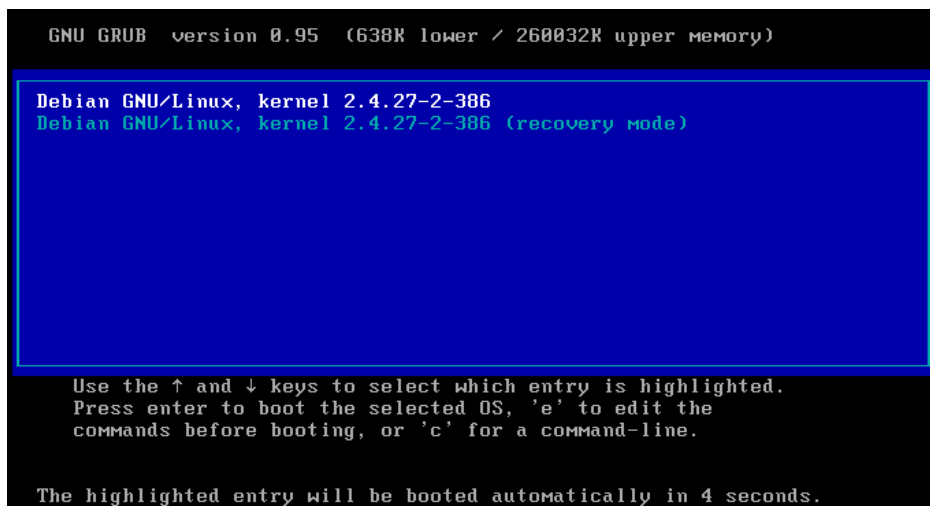


```
pc2-debian:~# ifconfig
eth0 Link encap:Ethernet HWaddr 00:0C:29:DF:55:A4
 inet addr:172.27.106.202 Bcast:172.27.255.255 Mask:255.255.255.0
 UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
 RX packets:130888 errors:0 dropped:0 overruns:0 frame:0
 TX packets:67934 errors:0 dropped:0 overruns:0 carrier:0
 collisions:0 txqueuelen:1000
 RX bytes:190687146 (181.8 MiB) TX bytes:4511867 (4.3 MiB)
 Interrupt:18 Base address:0x1080

lo Link encap:Local Loopback
 inet addr:127.0.0.1 Mask:255.0.0.0
 UP LOOPBACK RUNNING MTU:16436 Metric:1
 RX packets:92 errors:0 dropped:0 overruns:0 frame:0
 TX packets:92 errors:0 dropped:0 overruns:0 carrier:0
 collisions:0 txqueuelen:0
 RX bytes:6040 (5.8 KiB) TX bytes:6040 (5.8 KiB)

pc2-debian:~# shutdown -r now
```

System zrestartujte.



```
GNU GRUB version 0.95 (638K lower / 260032K upper memory)

Debian GNU/Linux, kernel 2.4.27-2-386
Debian GNU/Linux, kernel 2.4.27-2-386 (recovery mode)

Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the
commands before booting, or 'c' for a command-line.

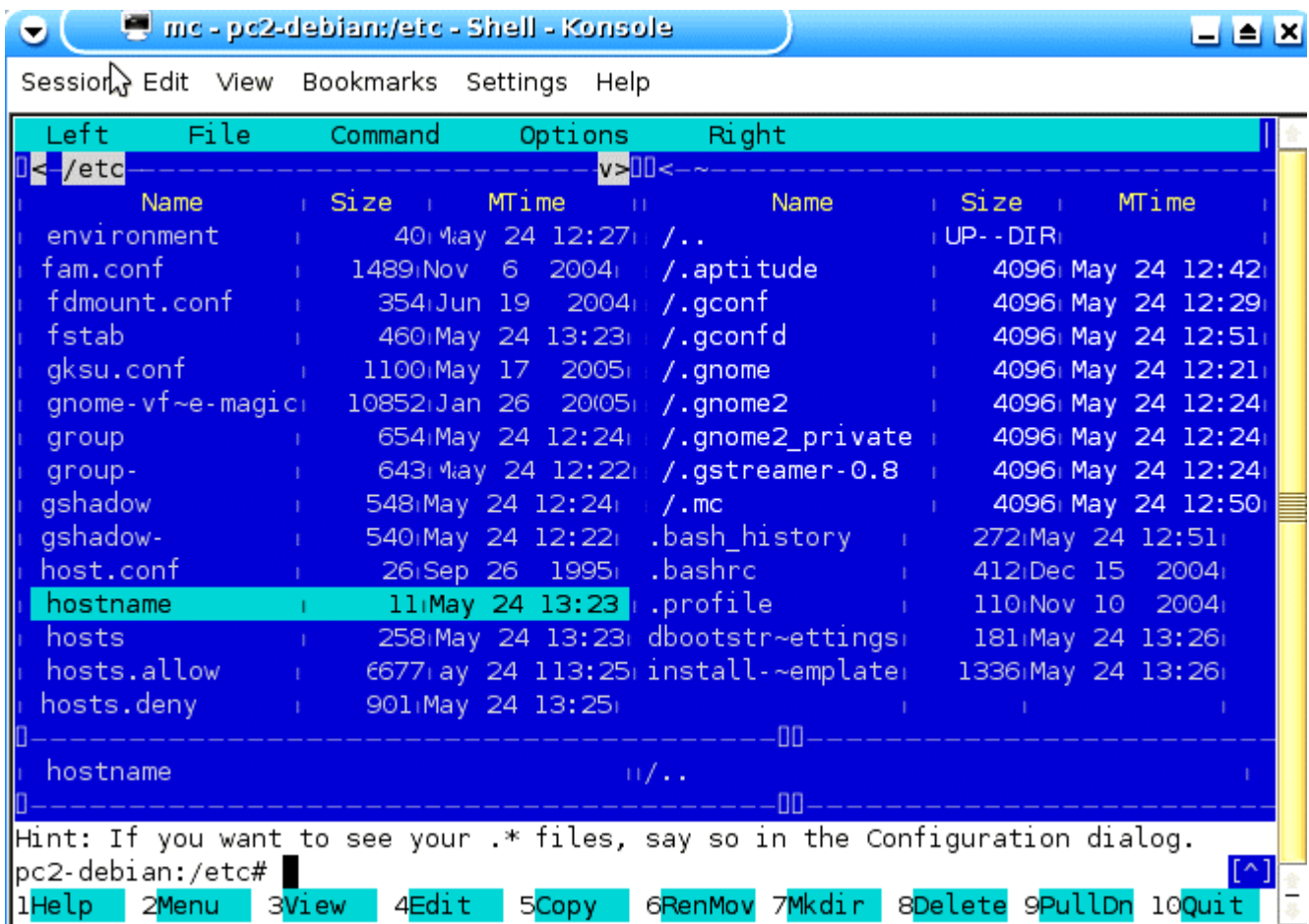
The highlighted entry will be booted automatically in 4 seconds.
```

Po restartu si vypíšte příkazem **ifconfig**, zda se nastavení zapsané v konfiguračním souboru aplikovalo. Do konfiguračního souboru ještě připište adresu **broadcastu** (**broadcast 172.27.106.255**).

```
pc2-debian:~# ifconfig
eth0 Link encap:Ethernet HWaddr 00:0C:29:DF:55:A4
 inet addr:172.27.106.202 Bcast:172.27.106.255 Mask:255.255.255.0
 UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
 RX packets:29 errors:0 dropped:0 overruns:0 frame:0
 TX packets:14 errors:0 dropped:0 overruns:0 carrier:0
 collisions:0 txqueuelen:1000
 RX bytes:2939 (2.8 KiB) TX bytes:917 (917.0 b)
 Interrupt:18 Base address:0x1080

lo Link encap:Local Loopback
 inet addr:127.0.0.1 Mask:255.0.0.0
 UP LOOPBACK RUNNING MTU:16436 Metric:1
 RX packets:72 errors:0 dropped:0 overruns:0 frame:0
 TX packets:72 errors:0 dropped:0 overruns:0 carrier:0
 collisions:0 txqueuelen:0
 RX bytes:4740 (4.6 KiB) TX bytes:4740 (4.6 KiB)
```

V adresáři **/etc/** najdete soubor **hostname**, ve kterém je definováno jméno tohoto virtuálního počítače.



```

mc - pc2-debian:/etc - Shell - Konsole
Session Edit View Bookmarks Settings Help
hostname [----] 10 L:[1+ 0 1/ 2]
pc2-debian

```

Najděte soubor **resolv.conf** v adresáři **/etc/**. Jsou v něm definovány nameservery.

```

mc - pc2-debian:/etc - Shell - Konsole
Session Edit View Bookmarks Settings Help
Left File Command Options Right
|< /etc |>
-----v-----
Name | Size | MTime | Name | Size | MTime
passwd | 1128 | May 24 12:24 | /.. | UP- | DIR
passwd- | 1128 | May 24 12:24 | /.aptitude | 4096 | May 24 12:42
pmount.allow | 104 | May 18 2005 | /.gconf | 4096 | May 24 12:29
printcap | 643 | May 24 12:21 | /.gconfd | 4096 | May 24 12:51
profile | 501 | Nov 10 2004 | /.gnome | 4096 | May 24 12:21
protocols | 2478 | Mar 21 2005 | /.gnome2 | 4096 | May 24 12:24
reportbug.conf | 2555 | Dec 7 2004 | /.gnome2_private | 4096 | May 24 12:24
resolv.conf | 40 | May 24 13:27 | /.gstreamer-0.8 | 4096 | May 24 12:24
*rmt | 268 | Aug 3 2004 | /.mc | 4096 | May 24 12:55
rpc | 887 | Mar 21 2005 | .bash_history | 2 | 272 | May 24 12:51
scrollke~er.conf | 23 | Apr 17 2005 | .bashrc | 4 | 412 | Dec 15 2004
securetty | 994 | May 18 2005 | .profile | 1 | 110 | Nov 10 2004
services | 17571 | Mar 21 2005 | dbootstr~ettings | 1 | 181 | May 24 13:26
shadow | 766 | May 24 12:24 | install~emplate | 1 | 1336 | May 24 13:26
shadow- | 741 | May 24 12:22 | | | | |
|-----|-----|
| resolv.conf | | | | | |
|-----|-----|
Hint: If you want to see your .* files, say so in the Configuration dialog.
pc2-debian:/etc#
1Help 2Menu 3View 4Edit 5Copy 6RenMov 7Mkdir 8Delete 9PullDn 10Quit

```

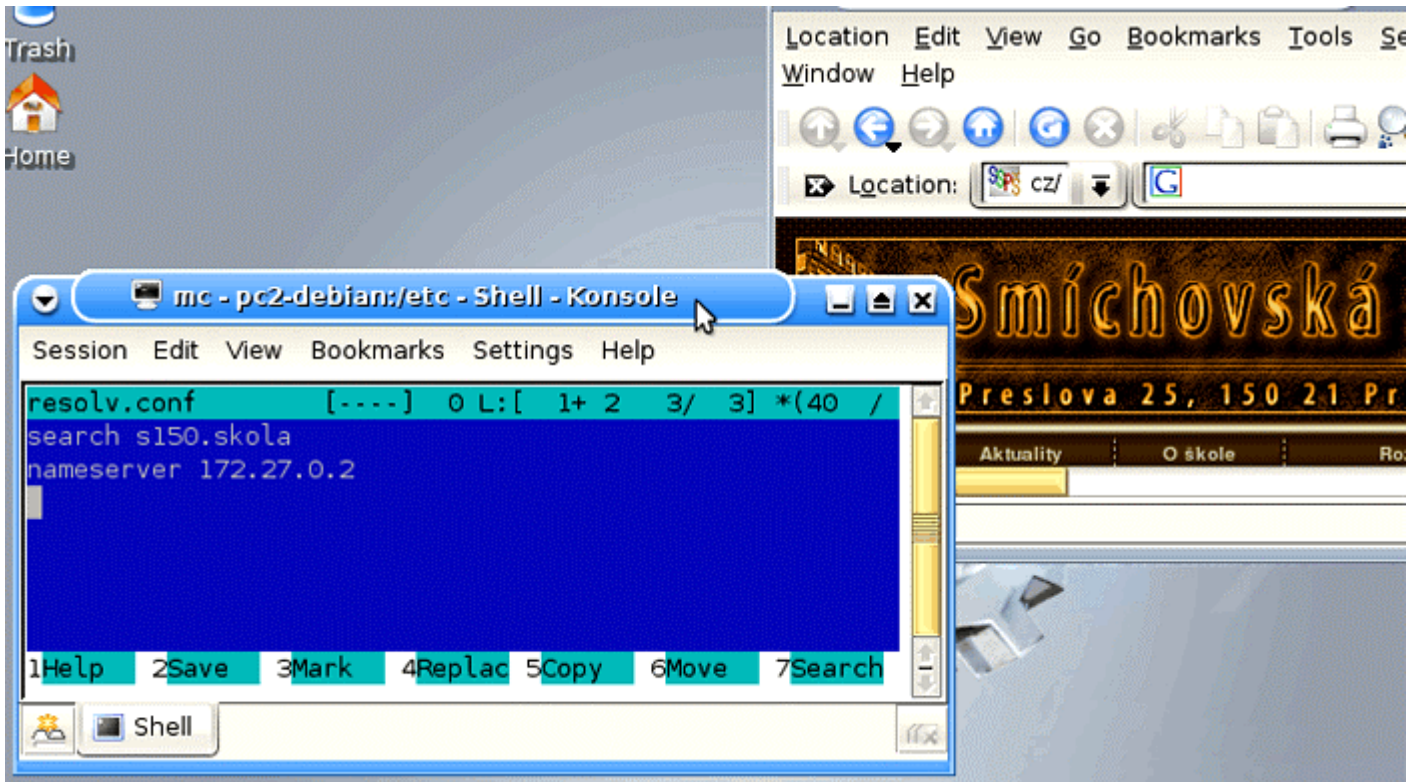
```

mc - pc2-debian:/etc - Shell - Konsole
Session Edit View Bookmarks Settings Help
resolv.conf [----] 0 L:[1+ 2 3]
search s150.skola
nameserver 172.27.0.2

```

Vyzkoušejte pomocí například internetového prohlížeče, že fungují DNS překlady. Zkuste zadat například www adresu školního webu. Pokud se stránka zobrazí, dá se z pohledu OSI modelu říci, že funguje všech 7

vrstev tohoto modelu. Funkčnost DNS překladačů můžeme vyzkoušet také např. pomocí **ping** nebo **tracert**, tím ověříme funkčnost OSI modelu do prvních tří vrstev.



```

pc2-debian:~# tracert www.ssps.cz
tracert to www.ssps.cz (81.95.96.94), 30 hops max, 38 byte packets
 1 172.27.106.2 (172.27.106.2) 0.295 ms 0.246 ms 0.194 ms
 2 10.0.0.138 (10.0.0.138) 0.543 ms 0.483 ms 0.426 ms
 3 80.188.61.118 (80.188.61.118) 4.615 ms 4.611 ms 5.013 ms
 4 33.189.broadband3.iol.cz (85.70.189.33) 5.330 ms 5.687 ms 5.627 ms
 5 80.188.33.245 (80.188.33.245) 5.441 ms 18.733 ms 5.077 ms
 6 194.228.21.32 (194.228.21.32) 23.950 ms 4.439 ms 38.054 ms
 7 nix2-ge.active24.cz (194.50.100.235) 10.708 ms 31.406 ms 11.292 ms
 8 uvirt7.active24.cz (81.95.96.94) 32.508 ms 10.122 ms 36.338 ms
pc2-debian:~# █

```

## 16.3 Ethereal

Nainstalujte si balíček **ethereal**. Jedná se o zachytávač paketů. Lze si zobrazit kompletní strukturu a obsah balíčku.

```

mc - pc2-debian/etc - Shell - Konsole
Session Edit View Bookmarks Settings Help
Actions Undo Package Search Options Views Help
f10: Menu ? : Help q: Quit u: Update g: Download/Install/Remove Pkgs
aptitude 0.2.15.9 Will use 53.4MB of disk space DL Size: 14.9MB
pi ethereal +1237kB <none> 0.10.10-2s
piA ethereal-common +17.8MB <none> 0.10.10-2s
p etherwake <none> 1.08-1
p ethstats <none> 1.0-2
p ethstatus <none> 0.4.2-2
p ettercap <none> 1:0.7.1-1s
p ettercap-common <none> 1:0.7.1-1s
p ettercap-gtk <none> 1:0.7.1-1s
p eudc <none> 1.28b-12
p ez-ipupdate <none> 3.0.11b8-8
p farpd <none> 0.2-7
network traffic analyzer
Ethereal is a network traffic analyzer, or "sniffer", for Unix and Unix-like operating systems.
A sniffer is a tool used to capture packets off the wire. Ethereal decodes numerous protocols
(too many to list).
This package provides ethereal (the GTK+ version)

```

```

cdrom://[Debian GNU/Linux 3.1 r0a _Sarge_ - Official i386 Binary-1 (20050607)] unsta
cdrom://[Debian GNU/Linux 3.1 r0a _Sarge_ - Official i386 Binary-1 (20050607)] unsta
cdrom://[Debian GNU/Linux 3.1 r0a _Sarge_ - Official i386 Binary-1 (20050607)] unsta
cdrom://[Debian GNU/Linux 3.1 r0a _Sarge_ - Official i386 Binary-1 (20050607)] unsta
cdrom://[Debian GNU/Linux 3.1 r0a _Sarge_ - Official i386 Binary-1 (20050607)] unsta
ethereal-common [1365kB/4496kB]
Total Progress: [17%] (52.1kB/s, 2m0s remaining)

```

Pro instalaci budeme potřebovat druhé instalační DVD.

```

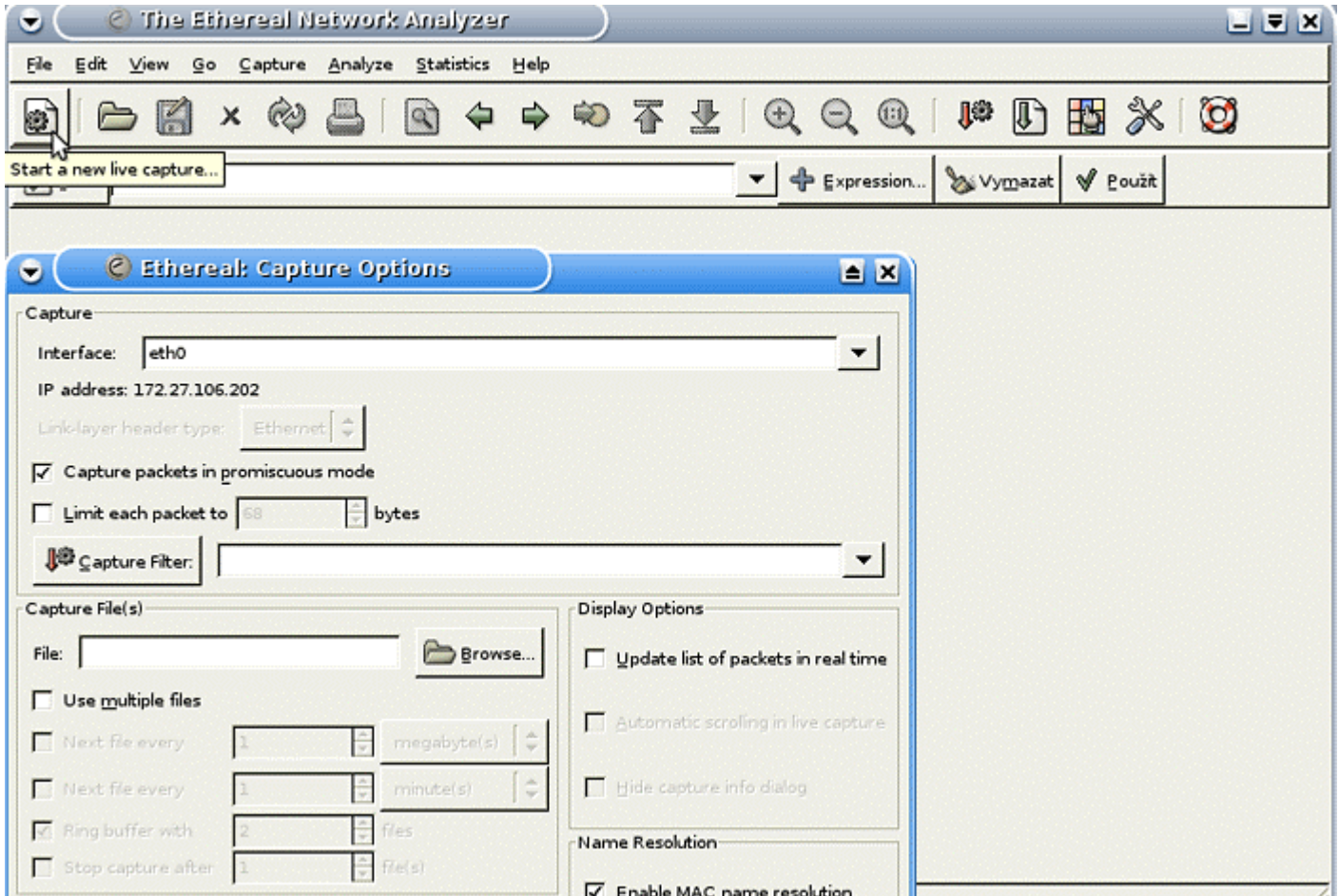
Please insert the following disc into the drive "/cdrom/":
Debian GNU/Linux 3.1 r0a _Sarge_ - Official i386 Binary-2 (20050607)
[ok]

```

### Poznámka

**Ethereal** najdeme po nainstalování v základní nabídce KDE v položce **Internet**.

Spustíte **Ethereal** a kliknete na „**Start a new live capture**“. Spustíte zachytávání. Vytvořte během zachytávání nějakou síťovou aktivitu, například příkazem **ping** otestujte spojení s rodičovským systémem. Poté zachytávání stopněte a prozkoumejte zachycené pakety.



The screenshot shows the Ethereal interface with a terminal window titled 'mc - pc2-debian/etc - Shell - Konsole'. The terminal displays four ICMP echo requests from 172.27.106.102 to 172.27.106.102. The main window shows a 'Captured Packets' graph with the following data:

| Protocol | Count | % of total |
|----------|-------|------------|
| Total    | 26    |            |
| ICMP     | 14    | 53.8%      |
| Other    | 2     | 7.7%       |

The screenshot shows the Ethereal interface with a list of captured packets. The selected packet (No. 23) is an ICMP Echo (ping) request from 172.27.106.202 to 172.27.106.102. The detailed view shows the Ethernet II header and the ICMP payload.

| No. | Time      | Source            | Destination    | Protocol | Info                                        |
|-----|-----------|-------------------|----------------|----------|---------------------------------------------|
| 1   | 0.000000  | 02:01:00:00:00:00 | Broadcast      | 0x886f   | HS NLB heartbeat                            |
| 12  | 10.035924 | 02:01:00:00:00:00 | Broadcast      | 0x886f   | HS NLB heartbeat                            |
| 41  | 19.991156 | 02:01:00:00:00:00 | Broadcast      | 0x886f   | HS NLB heartbeat                            |
| 8   | 9.744460  | 172.27.106.202    | Broadcast      | ARP      | Who has 172.27.106.102? Tell 172.27.106.202 |
| 9   | 9.744671  | 172.27.106.102    | 172.27.106.202 | ARP      | 172.27.106.102 is at 00:13:d3:cd:27:78      |
| 10  | 9.744751  | 172.27.106.202    | 172.27.106.102 | ICMP     | Echo (ping) request                         |
| 11  | 9.744829  | 172.27.106.102    | 172.27.106.202 | ICMP     | Echo (ping) reply                           |
| 21  | 10.757175 | 172.27.106.202    | 172.27.106.102 | ICMP     | Echo (ping) request                         |
| 22  | 10.757305 | 172.27.106.102    | 172.27.106.202 | ICMP     | Echo (ping) reply                           |
| 23  | 11.779893 | 172.27.106.202    | 172.27.106.102 | ICMP     | Echo (ping) request                         |
| 24  | 11.780062 | 172.27.106.102    | 172.27.106.202 | ICMP     | Echo (ping) reply                           |
| 25  | 12.788413 | 172.27.106.202    | 172.27.106.102 | ICMP     | Echo (ping) request                         |
| 26  | 12.788577 | 172.27.106.102    | 172.27.106.202 | ICMP     | Echo (ping) reply                           |
| 27  | 13.797865 | 172.27.106.202    | 172.27.106.102 | ICMP     | Echo (ping) request                         |
| 28  | 13.798025 | 172.27.106.102    | 172.27.106.202 | ICMP     | Echo (ping) reply                           |
| 29  | 14.807537 | 172.27.106.202    | 172.27.106.102 | ICMP     | Echo (ping) request                         |
| 30  | 14.807678 | 172.27.106.102    | 172.27.106.202 | ICMP     | Echo (ping) reply                           |
| 31  | 15.816619 | 172.27.106.202    | 172.27.106.102 | ICMP     | Echo (ping) request                         |

Frame 23 (98 bytes on wire, 98 bytes captured)  
 Ethernet II, Src: 00:0c:29:df:55:a4, Dst: 00:13:d3:cd:27:78

```

0000 00 13 d3 cd 27 78 00 0c 29 df 55 a4 08 00 45 00 'x...'U...E.
0010 00 54 00 02 40 00 40 01 0d 40 ac 1b 6a ca ac 1b .T.@.@.@.j...
0020 6a 66 08 00 da df 96 06 00 03 e7 3e 74 44 3f 90 jf.....>tD?.
0030 01 00 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15
0040 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 !#$%

```

 *Domácí úkol*

Zkuste si stáhnout třicetidenní verzi VMware Workstation a zkuste v tomto programu nainstalovat OS Linux.

Vyzkoušejte konfiguraci síťového rozhraní, jména počítače i adresy nameserverů.

Vyzkoušejte ping a traceroute.

Vyzkoušejte zachytávání paketů pomocí Ethereal a vyhledejte nějaké konkrétní informace v zachycených paketech.

 *Shrnutí*

- ✓ Seznámili jste se s vytvářením virtuálního stroje v programu VMware Workstation.
- ✓ Vyzkoušeli jste si instalaci OS Linux Debian, typ stolní počítač.
- ✓ Umíte pomocí aptitude instalovat zvolené balíčky.
- ✓ Umíte použít příkaz ifconfig pro zobrazení informací i konfiguraci síťového rozhraní.
- ✓ Víte, kde trvale přenastavit konfiguraci síťového rozhraní.
- ✓ Víte, kde je nastaveno jméno počítače.
- ✓ Víte, kde jsou nastaveny adresy nameserverů.
- ✓ Umíte používat ping a traceroute.
- ✓ Umíte nainstalovat a použít zachytávač paketů Ethereal.



## 17. Základní pojmy ze sítí (33. – 34. hodina)

### 17.1 MAC adresa

Síťová karta počítače má svou identifikaci, MAC adresu. Je přiřazena výrobcem. Nazývá se také fyzickou adresou. Pracuje v druhé vrstvě OSI modelu (data link).

Je to **48-bitové** číslo. Vyjadřuje se v hexadecimální soustavě jako 6 skupin čísel, každé číslo je tvořeno dvěma hexadecimálními číslicemi, např. FF-AA-B2-CA-12-DF.

Na Windows XP lze vypsat MAC adresu např. příkazem **ipconfig /all** zadaném v příkazovém řádku.

### 17.2 IP adresa

Proto, aby mohly dva systémy spolu komunikovat, musí být schopny se vzájemně identifikovat.

Pro identifikaci slouží IP adresa, což je logická adresa počítače. Pracuje na vrstvě 3 (síťová vrstva) OSI modelu.

**IP adresa verze 4** je **32-bitové** číslo. Skládá se z části síťové (network part) a části určené pro koncové zařízení (host part). Všechny počítače, které jsou součástí jedné sítě, mají stejnou síťovou část.

Příklad IP adresy: 192.168.12.1.

Model IP adres verze 4 umožňuje vytvořit cca 4 miliardy IP adres ( $2^{32}$ ).

**IP adresa verze 6** vznikla z důvodu nedostatku adresního prostoru, který poskytuje IP adresa verze 4. Množství počítačů na světě během svého vývoje obrovsky vzrostl a v modelu verze 4 dochází volné IP adresy.

Řeší se to např. adresováním počítačů pomocí privátních IP adres a jejich překladem na menší počet veřejných adres (PAT – port address translation, NAT – network address translation).

Další možností, jak nedostatek volných IP adres řešit, je přejít na nový model – verzi 6.

IP adresa verze 6 je **128 bitové** číslo. Je rozděleno po 16 bitech do 8 skupin, vyjadřuje se hexadecimálně.

Příklad IP adresy:

AB21:CD22:5419:0024:00B0:AABB:FF22:1111.

Model IP adres verze 6 umožňuje vytvořit cca  $3,4 \times 10^{38}$  IP adres ( $2^{128}$ ).

Na Windows XP lze vypsat informace o IP adrese a dalších nastaveních např. příkazem **ipconfig /all** zadaném v příkazovém řádku.

```

C:\>ipconfig /all

Konfigurace protokolu IP systému Windows

 Název hostitele : ivona
 Primární přípona DNS. :
 Typ uzlu : hybridní
 Povoleno směrování IP : Ne
 WINS Proxy povoleno : Ne
 Prohledávací seznam přípon DNS. . : domena408.skola

Adaptér sítě Ethernet Připojení k místní síti:

 Přípona DNS podle připojení . . . : domena408.skola
 Popis : Realtek RTL8169/8110 Family Gigabit
Ethernet NIC
 Fyzická Adresa. : 00-11-09-C4-CB-A1
 Protokol DHCP povolen : Ano
 Automatická konfigurace povolena . : Ano
 Adresa IP : 172.27.50.70
 Maska podsítě : 255.255.255.0
 Úchozí brána : 172.27.50.2
 Server DHCP : 172.27.0.2
 Servery DNS : 172.27.0.2
 Primární server WINS. : 172.27.0.2
 Zapůjčeno : 14. března 2006 9:45:44
 Zapůjčka vyprší : 14. března 2006 21:45:44

```

### 17.3 Síťová maska

Síťová maska (někdy též maska podsítě) je 32 bitové číslo, které slouží k rozdělení IP adresy na část síťovou a část určenou pro koncové zařízení.

V binárním zápisu je tvořena spojitou řadou jedniček následovaných nulami.

Např. 11111111.11111111.11111111.11000000, dekadicky 255.255.255.192 určuje, že prvních 26 bitů v IP adrese je část síťová (musí být stejná pro všechny v počítače v jednom LAN segmentu) a posledních 6 bitů rozlišuje jednotlivá síťová zařízení.

### 17.4 LAN

**LAN** – local area network – lokální síť. Je to vysokorychlostní datová síť nacházející se na relativně malém geografickém území, cca do několika tisíc metrů. Propojuje počítače a jiná síťová zařízení v rámci jedné budovy nebo v rámci určitého geograficky omezeného území. Standardy LAN definují, jak má vypadat kabeláž a signalizace na LAN. Rozšířené LAN technologie jsou Ethernet, FDDI a Token Ring.

## 17.5 MAN

**MAN** – metropolitan area network – městská síť. Síť v rozsahu města. Zahrnuje území větší než LAN a menší než WAN.

## 17.6 WAN

**WAN** – wide area network – rozlehlá síť. Je to datová komunikační síť zahrnující rozsáhlé geografické území. Často pro přenos komunikace používá již existující linky.

Příklady WAN – Frame Relay, X.25, ISDN, DSL.

## 17.7 SAN

**SAN** – storage area network – ukládací síť. Specializovaná, výkonná, vysokorychlostní síť určená pro přenos velkých objemů dat mezi servery a úložišti dat.

## 17.8 DNS, DHCP

**DNS** – Domain Name System

System pro překlad IP adres na své slovní ekvivalenty. Využíváte jej například při prohlížení webových stránek, kdy píšete slovní názvy stránek namísto jejich IP adres.

**DHCP** – Dynamic Host Configuration Protocol

Umožňuje klientům na síti získat svou konfiguraci z DHCP serveru. Server přiděluje IP adresu, masku, adresu brány, může přidělit informace o DNS serverech, doménové jméno apod.

## 17.9 Adresa síť

Jak bylo v části o síťové masce řečeno, maska určuje, jaká část IP adresy je část síťová a jaká je část koncového zařízení.

Všechna zařízení v jednom LAN segmentu mají stejnou síťovou část IP adresy. V host part se liší.

### 17.9.1 Příklad

**IP adresa: 192.168.2.200**

**maska: 255.255.255.192**

binárně:

**11000000.10101000.00000010.11001000** – IP adresa

**11111111.11111111.11111111.11000000** – maska

**11000000.10101000.00000010.11000000** – IP adresa síť

IP adresu sítě získáme z IP adresy a masky tak, že napíšeme (binárně) bity z IP adresy na pozicích, kde jsou v masce jedničky a zbytek doplníme nulami.

V našem příkladě je IP adresa sítě dekadicky **192.168.2.192**.

### 17.9.2 Otázka

Jsou následující dvě IP adresy ve stejné síti?

192.168.2.193

192.168.2.190

maska obou je 255.255.255.192

binárně:

11000000.10101000.00000010.11000001

11000000.10101000.00000010.10111110

Žlutě je označena síťová část. Liší se. Z toho důvodu nejsou obě adresy ve stejné síti.

První IP adresa patří do sítě 192.168.2.192 a druhá patří do sítě 192.168.2.128.

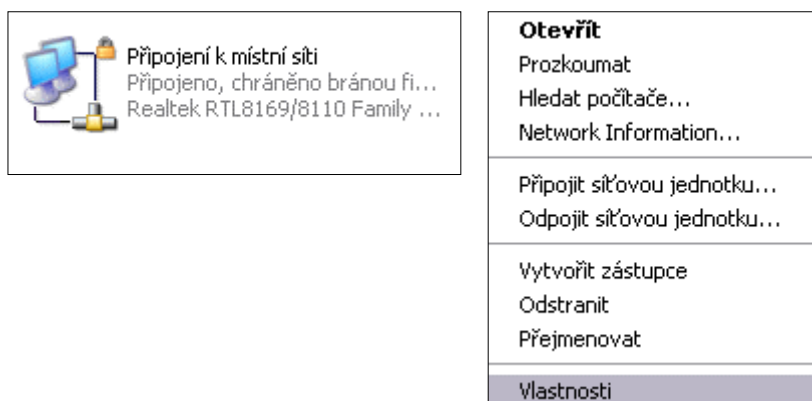
Adresy sítě získáme tak, že žlutě označenou síťovou část ponecháme a zbytek vyplníme nulami.

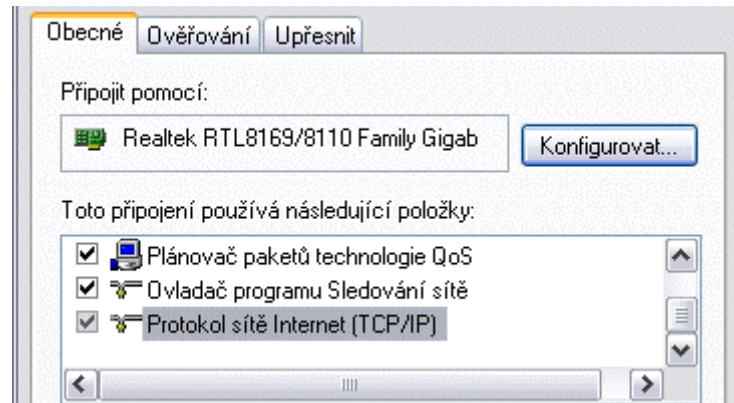
## 17.10 Nastavení IP adresy, masky, brány ve Windows

Pro nastavení parametrů potřebujeme vyvolat dialogové okno Vlastnosti daného síťového připojení.

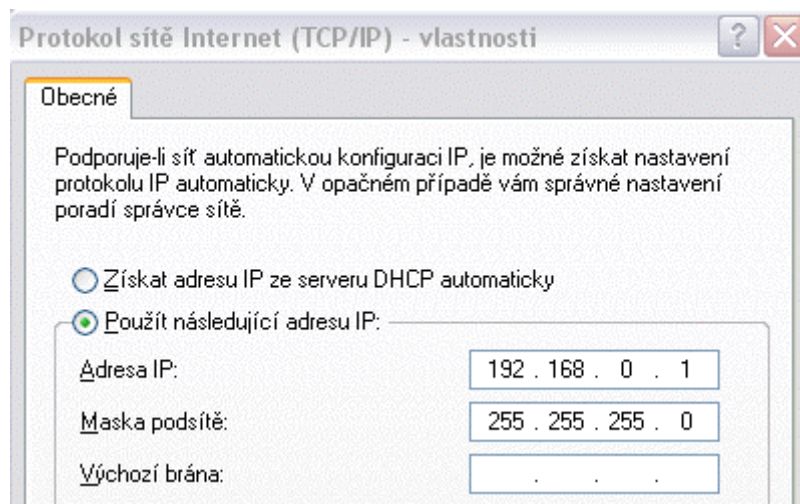
Jedna z možností je:

Nabídka Start – Nastavení – Síťová připojení, pravé tlačítko myši – Vlastnosti.





Vybereme položku „**Protokol sítě Internet (TCP/IP)**“ a klikneme na tlačítko **Vlastnosti**.

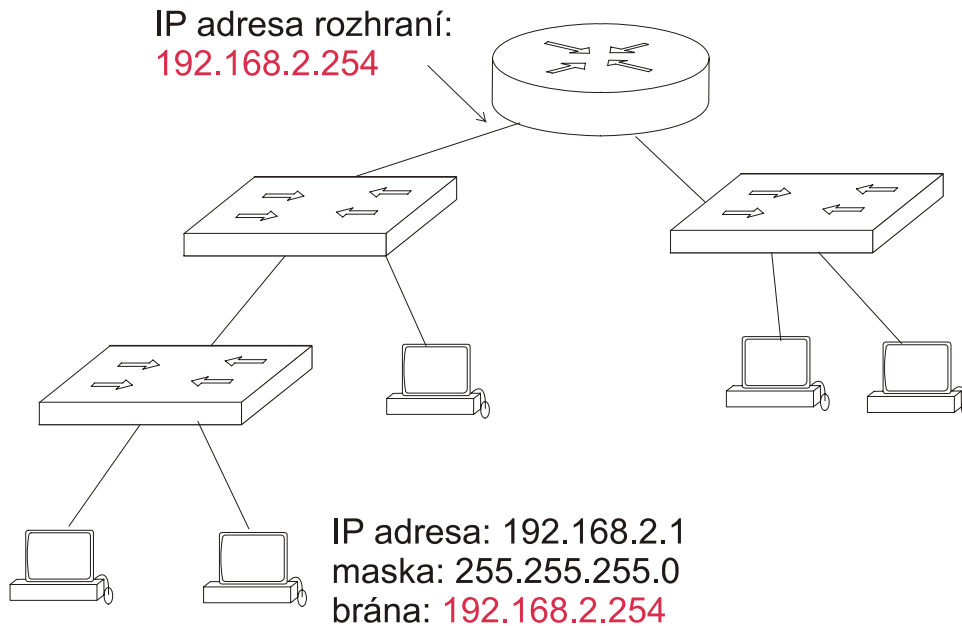


V tomto dialogovém okně máme možnost zvolit, zda bude IP adresa přiřazována dynamicky **DHCP** serverem nebo bude přiřazena **staticky**.

V druhém případě musíme zvolit IP adresu, masku podsítě a bránu.

IP adresu musíme zvolit vhodně tak, aby byla součástí sítě, ve které se počítač nachází.

**Brána** je nejbližší rozhraní routeru, který náš lokální segment sítě připojuje do jiné části sítě.



## 17.11 Ping, tracert

### 17.11.1 Ping

Pro ověření dosažitelnosti daného síťového zařízení je možné použít příkaz **ping**. Ne vždy ale musí zařízení na ping odpovídat, což ale neznamená, že nutně neexistuje.

```

C:\WINDOWS\system32\cmd.exe

C:\>ping www.cisco.com

Příkaz PING na www.cisco.com [198.133.219.25] s délkou 32 bajtů:

Upršel časový limit žádosti.
Odpověď od 198.133.219.25: bajty=32 čas=166ms TTL=107
Upršel časový limit žádosti.
Odpověď od 198.133.219.25: bajty=32 čas=165ms TTL=107

Statistika ping pro 198.133.219.25:
Pakety: Odeslané = 4, Přijaté = 2, Ztracené = 2 (ztráta 50%),
Přibližná doba do přijetí odezvy v milisekundách:
 Minimum = 165ms, Maximum = 166ms, Průměr = 165ms

```

V ukázce byl proveden ping na [www.cisco.com](http://www.cisco.com). Je vidět, že na dva ze čtyř odeslaných paketů cílový počítač neodpověděl v časovém limitu, proto je ztráta 50%.

Dále se ve statistice uvádí počet odeslaných paketů, počet přijatých paketů, maximální, minimální a průměrná doba odezvy.

Pomocí **přepínače -n** je možné testovat spojení více odeslanými pakety.

```
C:\>ping www.seznam.cz -n 10

Příkaz PING na www.seznam.cz [194.228.32.3] s délkou 32 bajtů:

Odpověď od 194.228.32.3: bajty=32 čas=12ms TTL=56
Odpověď od 194.228.32.3: bajty=32 čas=8ms TTL=56
Odpověď od 194.228.32.3: bajty=32 čas=5ms TTL=56
Odpověď od 194.228.32.3: bajty=32 čas=6ms TTL=56
Odpověď od 194.228.32.3: bajty=32 čas=15ms TTL=56
Odpověď od 194.228.32.3: bajty=32 čas=7ms TTL=56
Odpověď od 194.228.32.3: bajty=32 čas=6ms TTL=56
Odpověď od 194.228.32.3: bajty=32 čas=12ms TTL=56
Odpověď od 194.228.32.3: bajty=32 čas=22ms TTL=56
Odpověď od 194.228.32.3: bajty=32 čas=9ms TTL=56

Statistika ping pro 194.228.32.3:
Pakety: Odeslané = 10, Přijaté = 10, Ztracené = 0 (ztráta 0%),
Přibližná doba do přijetí odezvy v milisekundách:
 Minimum = 5ms, Maximum = 22ms, Průměr = 10ms
```

Pomocí **přepínače -t** je možné opakovaně odesílat určenému cíli žádost o ozvěnu, až do ukončení.

Ukončení se provede klávesovou zkratkou **CTRL+C**, přerušení s výpisem statistik a pokračování odesílání paketů se provede klávesovou zkratkou **CTRL+Break**.

```
C:\>ping www.seznam.cz -t

Příkaz PING na www.seznam.cz [194.228.32.3] s délkou 32 bajtů:

Odpověď od 194.228.32.3: bajty=32 čas=6ms TTL=56
Odpověď od 194.228.32.3: bajty=32 čas=6ms TTL=56
Odpověď od 194.228.32.3: bajty=32 čas=6ms TTL=56

Statistika ping pro 194.228.32.3:
Pakety: Odeslané = 3, Přijaté = 3, Ztracené = 0 (ztráta 0%),
Přibližná doba do přijetí odezvy v milisekundách:
 Minimum = 6ms, Maximum = 6ms, Průměr = 6ms
Ctrl+Break
Odpověď od 194.228.32.3: bajty=32 čas=15ms TTL=56
Odpověď od 194.228.32.3: bajty=32 čas=17ms TTL=56
Odpověď od 194.228.32.3: bajty=32 čas=5ms TTL=56
Odpověď od 194.228.32.3: bajty=32 čas=14ms TTL=56
Odpověď od 194.228.32.3: bajty=32 čas=6ms TTL=56
Odpověď od 194.228.32.3: bajty=32 čas=16ms TTL=56
Odpověď od 194.228.32.3: bajty=32 čas=6ms TTL=56

Statistika ping pro 194.228.32.3:
Pakety: Odeslané = 10, Přijaté = 10, Ztracené = 0 (ztráta 0%),
Přibližná doba do přijetí odezvy v milisekundách:
 Minimum = 5ms, Maximum = 17ms, Průměr = 9ms
Control-C
^C
```

### 17.11.2 Tracert

Pomocí příkazu **tracert** je možno zjistit, jakou cestou putuje paket ke zvolenému cíli a odhalit zdroje možných problémů se spojením.

```
C:\>tracert www.seznam.cz

Úypis trasy k www.seznam.cz [194.228.32.3]
s nejuýše 30 směřováními:

 1 1 ms 1 ms 1 ms gw-kolovraty.sxg.cz [62.77.98.1]
 2 4 ms 4 ms 4 ms gw.sxg.cz [62.77.92.1]
 3 5 ms 3 ms 4 ms klient98-193.sxg.cz [62.77.98.193]
 4 5 ms 5 ms 7 ms 62.84.143.13
 5 5 ms 5 ms 5 ms 213.29.73.77
 6 15 ms 11 ms 5 ms 194.50.100.160
 7 17 ms 6 ms 6 ms 194.228.37.219
 8 15 ms 6 ms 6 ms 194.228.32.244
 9 15 ms 6 ms 6 ms 194.228.32.3

Trasování bylo dokončeno.
```

Jsou zde vidět doby jednotlivých odezev všech uzlů, kterým paket cestuje. Lze tu odhalit, který z bodů je přetížen provozem, případně na kterém místě je spojení přerušeno.

```
C:\>tracert www.cisco.com

Úypis trasy k www.cisco.com [198.133.219.25]
s nejuýše 30 směřováními:

 1 19 ms 1 ms 1 ms gw-kolovraty.sxg.cz [62.77.98.1]
 2 6 ms 3 ms 3 ms gw.sxg.cz [62.77.92.1]
 3 3 ms 4 ms 4 ms klient98-193.sxg.cz [62.77.98.193]
 4 5 ms 6 ms 5 ms 62.84.143.13
 5 24 ms 5 ms 5 ms 213.29.73.77
 6 5 ms 5 ms 20 ms 195.39.48.85
 7 22 ms 13 ms 17 ms 195.39.208.113
 8 12 ms 12 ms 27 ms 217.147.111.113
 9 12 ms 12 ms 12 ms 217.147.96.41
10 21 ms 22 ms 21 ms 213.206.129.65
11 93 ms 93 ms 94 ms 144.232.20.45
12 93 ms 93 ms 93 ms 144.232.7.110
13 95 ms 95 ms 95 ms 144.232.20.97
14 95 ms 95 ms 95 ms 144.232.16.93
15 164 ms 163 ms 164 ms 144.232.20.163
16 164 ms 164 ms 164 ms 144.232.4.234
17 165 ms 165 ms 166 ms 144.232.20.99
18 166 ms 165 ms 165 ms 144.232.3.138
19 166 ms 168 ms 166 ms 144.228.44.14
20 217 ms 228 ms 221 ms 128.107.239.89
21 166 ms 166 ms 166 ms 128.107.224.73
22 * * * Vypršel časový limit žádosti.
23 * * * Vypršel časový limit žádosti.
24 * * * Vypršel časový limit žádosti.
25 * * * Vypršel časový limit žádosti.
26 * * * Vypršel časový limit žádosti.
27 * * * Vypršel časový limit žádosti.
28 * * * Vypršel časový limit žádosti.
29 * * * Vypršel časový limit žádosti.
30 * * * Vypršel časový limit žádosti.

Trasování bylo dokončeno.
```



### 17.12 Cvičení

Určete MAC adresu svého počítače.

Určete IP adresu svého počítače, bránu, masku, název počítače, DNS server.

Z informací zjistěte síťovou adresu svého LAN segmentu.

Co je LAN, WAN, MAN, SAN?

### 17.13 Cvičení

Určete adresu sítě z následujících informací:

IP adresa: 172.17.175.35

maska: 255.255.248.0

### 17.14 Cvičení

Jsou následující adresy součástí jedné sítě? Napište jejich příslušné síťové adresy.

48.211.2.35

48.223.4.54

48.86.211.25

48.103.154.21

maska: 255.240.0.0

### 17.15 Cvičení

Zjistěte minimální, maximální a průměrnou dobu odezvy a procentuelní ztrátu příkazu ping na [www.centrum.cz](http://www.centrum.cz) při odeslání 10 požadavků na ozvěnu.

Určete odpovídající IP adresu této domény.

Kolika uzly projde paket při cestě do tohoto cíle?

 *Domácí úkol*

Vyzkoušejte si nastavení IP adresy, masky a brány svého počítače.  
Zjistěte MAC adresu své síťové karty.  
Určete síťovou adresu odpovídající vaší IP adrese a masce.  
Zjistěte, jaké DNS servery používáte.  
Zjistěte název svého počítače.  
Zopakujte si použití příkazu ping a tracert.

 *Shrnutí*

- ✓ Umíte vysvětlit, co je MAC adresa a umíte ji zjistit.
- ✓ Víte, k čemu slouží IP adresa, jaké jsou její typy, jaký má formát a jak se nastavuje.
- ✓ Víte, jakou funkci má maska podsítě.
- ✓ Umíte z IP adresy a masky podsítě zjistit adresu sítě.
- ✓ Víte, co je výchozí brána a jak ji nastavíte.
- ✓ Umíte zjistit, jaké DNS servery váš počítač používá.
- ✓ Umíte zjistit, jaké jméno má váš počítač nastaveno.
- ✓ Umíte používat příkazy ping a tracert.

## 18. Třídy IP adres, privátní adresy, unicast, multicast, broadcast (35. - 36. hodina)

### 18.1 Třídy IP adres

IP adresy jsou rozděleny do skupin zvaných třídy.

#### 18.1.1 Třída A

Tato třída je vhodná pro sítě, které obsahují extrémně mnoho koncových zařízení, každá taková síť má adresní prostor až pro cca 16 miliónů IP adres.

IP adresa se skládá z části síťové, která je tu zastoupena 8 bity, zbývajících 24 bitů je určeno pro identifikaci koncových zařízení. Pro zvolenou síťovou adresu je tedy k dispozici  $2^{24}$  permutací jedniček a nul, což pokrývá zmíněných cca 16 miliónů IP adres.

V binárním zápisu začíná IP adresa třídy A vždy nulou.

**0xxx xxxx**.hhhh hhhh.hhhh hhhh.hhhh hhhh

Žlutě je označena část identifikující síť. Část určená pro identifikaci koncového zařízení je zde vyznačena písmeny „h“ (host part).

Na prvním místě je vždy nula. Zbývajících 7 bitů je možno doplnit jedničkami a nulami. Nejnižší varianta je tedy **0000 0000**, nejvyšší **0111 1111**. Dekadicky **0 až 127**.

Nula ani 127 se pro adresování sítí nepoužívá. Adresa začínající 127 je adresa **loopbacku**, který slouží pro testování funkčnosti TCP/IP konfigurace počítače (ping 127.0.0.1).

Tzn. **IP adresy třídy A začínají čísly 1 až 126**.

**Maska podsítě** pro třídu A je **255.0.0.0**.

#### 18.1.2 Třída B

Tato třída je vhodná pro střední až velké sítě, každá taková síť má adresní prostor pro cca 65000 IP adres.

IP adresa se skládá z části síťové, která je tu zastoupena 16 bity, zbývajících 16 bitů je určeno pro identifikaci koncových zařízení. Pro zvolenou síťovou adresu je tedy k dispozici  $2^{16}$  permutací jedniček a nul, což pokrývá zmíněných cca 65000 IP adres.

V binárním zápisu začíná IP adresa třídy B vždy jedničkou a nulou.

**10xx xxxx**.xxxx xxxx.hhhh hhhh.hhhh hhhh

Žlutě je označena část identifikující síť. Část určená pro identifikaci koncového zařízení je zde vyznačena písmeny „h“ (host part).

Na prvním místě je vždy jednička a nula. Zbývajících 6 bitů prvního bytu je možno doplnit jedničkami a nulami. Nejnižší varianta je tedy **1000 0000**, nejvyšší **1011 1111**.

Dekadicky **128 až 191**.

Tzn. **IP adresy třídy B začínají čísly 128 až 191**.

**Maska podsítě** pro třídu B je **255.255.0.0**.

### 18.1.3 Třída C

Tato třída je vhodná pro menší sítě, každá taková síť má adresní prostor pro 254 IP adres.

IP adresa se skládá z části síťové, která je tu zastoupena 24 bity, zbývajících 8 bitů je určeno pro identifikaci koncových zařízení. Pro zvolenou síťovou adresu je tedy k dispozici  $2^8$  permutací jedniček a nul, což pokrývá zmíněných 254 IP adres ( $2^8$  je sice 256, ale dvě z těchto adres se nepoužívají pro koncová zařízení, což bude objasněno později).

V binárním zápisu začíná IP adresa třídy C vždy jedničkou, jedničkou a nulou.

**110x xxxx.xxxx xxxx.xxxx xxxx**.hhhh hhhh

Žlutě je označena část identifikující síť. Část určená pro identifikaci koncového zařízení je zde vyznačena písmeny „h“ (host part).

Na prvním místě je vždy jednička, jednička a nula. Zbývajících 5 bitů prvního bytu je možno doplnit jedničkami a nulami. Nejnižší varianta je tedy **1100 0000**, nejvyšší **1101 1111**.

Dekadicky **192 až 223**.

Tzn. **IP adresy třídy C začínají čísly 192 až 223**.

**Maska podsítě** pro třídu C je **255.255.255.0**.

### 18.1.4 Třída D

Tato třída je určena pro multicast, což je vysílání předem definované skupině IP adres. Proto může jednotlivé síťové zařízení vysílat najednou data určité skupině koncových zařízení.

V binárním zápisu začíná IP adresa třídy D vždy jedničkou, jedničkou, jedničkou a nulou.

**1110** xxxx.xxxx xxxx.xxxx xxxx.xxxx xxxx

Na prvním místě je vždy jednička, jednička, jednička a nula. Zbývajících 4 bity prvního bytu je možno doplnit jedničkami a nulami. Nejnižší varianta je tedy **1110 0000**, nejvyšší **1110 1111**.

Dekadicky **224 až 239**.

Tzn. **IP adresy třídy D začínají čísly 224 až 239**.

### 18.1.5 Třída E

Tato třída je určena pro výzkumné účely IETF (Internet Engineering Task Force), nepoužívá se na internetu.

V binárním zápisu začíná IP adresa třídy E vždy čtyřmi jedničkami.

**1111** xxxx.xxxx xxxx.xxxx xxxx.xxxx xxxx

Na prvním místě jsou vždy jedničky. Zbývající 4 bity prvního bytu je možno doplnit jedničkami a nulami. Nejnižší varianta je tedy **1111 0000**, nejvyšší **1111 1111**.

Dekadicky **240 až 255**.

Tzn. **IP adresy třídy E začínají čísla 240 až 255**.

## 18.2 Privátní IP adresy

Veřejné IP adresy jsou a musí být unikátní, jinak by nebylo možno činit rozhodnutí, kam směřovat paket určený pro určitou cílovou IP adresu.

Narozdíl od toho existuje skupina IP adres nazývaných privátní IP adresy a ty mohou existovat v libovolném množství kopií. Podmínkou je, že tyto privátní adresy jsou „schovány“ za určitou veřejnou adresou, na kterou se překládají (pomocí NAT a PAT).

Například firma má 200 počítačů a ty jsou adresovány privátními IP adresami. Při přístupu na internet se překládají na jednu veřejnou IP adresu. Veřejná IP adresa musí být v celém internetu unikátní, ale privátní IP adresy se mohou kdekoli jinde opakovat, samozřejmě musí být opět schovány za jinou veřejnou adresou. V rámci jedné privátní sítě se IP adresy nesmí duplikovat.

Pro třídu A je privátní adresa sítě jen jedna: **10.0.0.0**, tzn. adresní rozsah této sítě je **10.0.0.0 – 10.255.255.255**.

Pro třídu B je privátních adres sítí 16: **172.16.0.0 – 172.31.0.0**. Každá z těchto sítí má adresní rozsah **x.x.0.0 – x.x.255.255**.

Pro třídu C je privátních adres sítí 256: **192.168.0.0 – 192.168.255.0**.

Každá z těchto sítí má adresní rozsah **x.x.x.0 – x.x.x.255**.

Pakety adresované privátními IP adresami se v internetu nesměřují, routery je zahazují.

## 18.3 Unicast, multicast, broadcast

**Unicast** znamená vysílání pro jedno koncové zařízení. Cílová IP adresa je adresou cílového zařízení.

**Multicast** je vysílání předem definované skupině IP adres. Proto může jednotlivé síťové zařízení vysílat najednou data určité skupině koncových zařízení.

**Broadcast** je vysílání určené všem počítačům v dané síti. Adresa broadcastu (na 3. vrstvě OSI modelu) pro danou síť se skládá z části síťové (ta je stejná pro všechny počítače v dané síti) a části určené pro koncové zařízení, která je v tomto případě vyplněna **samými jedničkami** (binárně).

### 18.3.1 Příklad

Počítač má IP adresu **192.168.1.25**, masku podsítě **255.255.255.0**. Broadcast v dané síti je **192.168.1.255**.

*Proč?*

Host part je zde reprezentována posledním bytem (to poznáme z masky), který binárně vyplníme samými jedničkami (1111 1111), což je dekadicky **255**. Síťová část (192.168.1) zůstává nezměněna.

### 18.3.2 Příklad

Počítač má IP adresu **172.16.2.3**, masku podsítě **255.255.0.0**. Broadcast v dané síti je **172.16.255.255**.

*Proč?*

Host part je zde reprezentována posledními dvěma byty (poznáme z masky), které při vytváření broadcastu vyplníme samými jedničkami (11111111.11111111), což je dekadicky **255.255**. Síťová část (172.16) zůstává nezměněna.

### 18.3.3 Příklad

Počítač má IP adresu **52.15.193.55** a masku **255.255.240.0**.

*Jaká je adresa broadcastu?*

Zde se nejedná o třídní masku, protože ta by byla pro třídu A (to je poznat z prvního čísla v IP adrese – 52) 255.0.0.0. Masku je zde delší, jedná se o podsít' původní třídy A.

Protože to není tak zřejmé, jako u třídních masek (v předchozích dvou příkladech), rozepíšeme to binárně.

Binárně je maska 255.255.240.0 rovna

**11111111.11111111.11110000.00000000**.

Pro vytvoření broadcastu je potřeba posledních **12** bitů v původní IP adrese vyplnit samými jedničkami.

Původní IP adresa 52.15.193.55 je binárně rovna

**00110100.00001111.11000001.00110111**

a pokud posledních 12 bitů vyplníme jedničkami, bude vypadat

**00110100.00001111.11001111.11111111**

Převědeme zpět do dekadické soustavy a získáme adresu broadcastu

**52.15.207.255**.

#### 18.3.4 Cvičení

Určete adresu broadcastu pro IP adresu a masku podsítě:

- a) 15.25.2.5, 255.0.0.0
- b) 130.15.1.14, 255.255.0.0
- c) 200.1.20.4, 255.255.255.0
- d) 20.40.29.100, 255.240.0.0
- e) 172.27.163.5, 255.255.224.0
- f) 192.168.3.168, 255.255.255.248

#### 18.3.5 Cvičení

Použijte adresy z předchozího cvičení a určete adresy sítí.

 *Domácí úkol*

Určete z IP adresy a masky adresu sítě a broadcastu a uveďte, zda se jedná o veřejnou nebo privátní adresu.

- a) 10.2.25.1, 255.0.0.0
- b) 10.12.28.132, 255.255.252.0
- c) 45.2.85.129, 255.255.255.128
- d) 128.24.38.96, 255.255.0.0
- e) 158.56.140.38, 255.255.255.224
- f) 191.45.200.2, 255.255.224.0

 *Shrnutí*

- ✓ Znáte rozdělení IP adres do tříd a umíte uvést podrobnosti.
- ✓ Znáte privátní rozsahy IP adres pro jednotlivé třídy.
- ✓ Víte, co je unicast, multicast a broadcast.
- ✓ Z IP adresy a masky podsítě umíte zjistit adresu sítě a broadcastu.



## 19. Podsítě (37. – 38. hodina)

### 19.1 Podsítě

Tím, že prodloužíme masku podsítě, stane se z třídní masky netřídní a z původní sítě se stane podsít.

Podsítě se vytváří mimo jiné proto, že pokud je vytváříme systematicky, lze je agregovat (sumarizovat) do jedné adresy a informace o této adrese posílat dále.

Pokud si půjčíme z host part například 3 bity, můžeme na nich vytvořit celkem 8 různých permutací jedniček a nul (000, 001, 010, 011, 100, 101, 110, 111), přičemž samé nuly a samé jedničky nebudeme používat (kvůli možné záměně s adresou sítě a broadcastu).

#### 19.1.1 Příklad

Jak by vypadaly IP adresy všech podsítí vytvořených ze sítě 172.27.0.0 vypůjčením 3 bitů z host part?

Nová maska bude mít namísto původních 16 jedniček o 3 jedničky víc:

**11111111.11111111.111**00000.00000000  
 dekadicky: 255.255.224.0

Půjčíme si z host part 3 bity, vytvoříme na nich celkem 8 různých permutací jedniček a nul (000, 001, 010, 011, 100, 101, 110, 111), přičemž samé nuly a samé jedničky nebudeme používat (kvůli možné záměně s adresou sítě a broadcastu).

**172.27.0.0** je binárně:  
 10101100.00011011.**000**00000.00000000

Na vypůjčené bity (vyznačeny žlutě) dosadíme možné permutace jedniček a nul:

1. podsít:  
10101100.00011011.**001**00000.00000000
2. podsít:  
10101100.00011011.**010**00000.00000000
3. podsít:  
10101100.00011011.**011**00000.00000000
4. podsít:  
10101100.00011011.**100**00000.00000000
5. podsít:  
10101100.00011011.**101**00000.00000000
6. podsít:  
10101100.00011011.**110**00000.00000000

Vyjádřeno dekadicky:

1. podsít: 172.27.32.0
2. podsít: 172.27.64.0
3. podsít: 172.27.96.0
4. podsít: 172.27.128.0
5. podsít: 172.27.160.0
6. podsít: 172.27.192.0

#### 19.1.2 Cvičení

Vypište adresy všech podsítí vzniklých ze sítě 192.168.1.0 vypůjčením 2 bitů z host part.

Vypište adresy všech podsítí vzniklých ze sítě 192.168.1.0 vypůjčením 3 bitů z host part.

### 19.1.3 Příklad

Je dána IP adresa 151.29.7.5.

1. Třída IP adresy: B
2. Defaultní síťová část: první dva oktety – „bajty“ (151.29)  
Defaultní host part: poslední dva oktety – „bajty“ (7.5)
3. Třídní maska: 255.255.0.0
4. Adresa sítě (pro třídní masku): **151.29.0.0**
5. Adresa broadcastu (pro třídní masku): 151.29.255.255
6. **Maska podsítě** je dána 255.255.254.0. Protože se jedná o IP adresu třídy B (její třídní maska by byla 255.255.0.0), je zřejmé, že nová maska vznikla z původní rozšířením o 7 bitů na úkor host part.  
O 7 bitů se tedy rozšířila síťová část.  
Jak již víme z dřívějších kapitol, **adresu sítě** (nebo podsítě) získáme ze znalosti IP adresy zařízení a masky tak, že zachováme bity z IP adresy na těch pozicích, kde jsou v masce jedničky a zbytek bitů se vynuluje. Jedná se o logický součin – **And**.

maska: 1111 1111.1111 1111.1111 1110.0000 0000  
 IP ad.: 1001 0111.0001 1101.0000 0111.0000 0101  
 síť.ad.(And): 1001 0111.0001 1101.0000 0110.0000 0000  
 adresa podsítě dekadicky: **151.29.6.0**  
 broadcast: 1001 0111.0001 1101.0000 0111.11111111  
 broadcast podsítě dekadicky: **151.29.7.255**

Závěr: IP adresa 151.29.7.5 s maskou 255.255.254.0 je součástí sítě 151.29.6.0.

#### 19.1.4 Cvičení

Z následujících IP adres určete:

- a) třídu
- b) adresu sítě
- c) adresu broadcastu v síti
- d) třídní masku
- e) s novou maskou podsítě 255.255.255.224 určete adresu
  - o podsítě
  - o broadcastu v podsíti

- 1) 52.0.0.1
- 2) 176.100.2.0
- 3) 26.100.15.254
- 4) 157.182.255.250
- 5) 192.205.132.132
- 6) 126.183.131.255
- 7) 223.146.233.129
- 8) 179.188.138.255

#### 19.1.5 Příklad

Jak vypadá 1., 2. a poslední IP adresa v síti 172.27.15.0 s maskou 255.255.255.128?

*Pozn.: 172.27.15.0 je jednou z podsítí sítě 172.27.0.0, která vznikla z původní sítě půjčením 9 bitů z host part.*

Rozepišme binárně IP adresu sítě a masku a vyznačme prostor, který zbývá pro adresaci koncových zařízení.

**255.255.255.128**

11111111. 11111111. 11111111. 10000000

**172.27.15.0**

10101100. 00011011. 00001111. 00000000

Na posledních 7 bitech (označeny oranžově) můžeme libovolně měnit jedničky a nuly.

Nejmenší číslo je tvořeno samými nulami (**0000000**), ale to by bylo shodné s adresou **podsítě**, proto takovou adresu nepovažujeme za použitelnou adresu pro koncové zařízení.

Byla by to adresa 172.27.15.0.

Největší číslo by bylo tvořeno sedmi jedničkami (**1111111**), ale to by se shodovalo s adresou **broadcastu**, proto takovou adresu nepovažujeme za použitelnou adresu pro koncové zařízení.

Byla by to adresa 172.27.15.127.

10101100. 00011011. 00001111. 01111111

**První** použitelná IP adresa bude mít v posledních sedmi bitech 0000001. Tj. bude vypadat:

10101100. 00011011. 00001111. 00000001

dekadicky: 172.27.15.1

**Druhá** použitelná IP adresa bude mít v posledních sedmi bitech 0000010. Tj. bude vypadat:

10101100. 00011011. 00001111. 00000010

dekadicky: 172.27.15.2

**Poslední** použitelná IP adresa bude mít v posledních sedmi bitech 1111110. Tj. bude vypadat:

10101100. 00011011. 00001111. 01111110

dekadicky: 172.27.15.126

#### 19.1.6 Cvičení

S maskou podsítě 255.255.255.224 určete adresu

- podsítě
- broadcastu v podsíti
- 1., 2., 3. a poslední použitelnou IP adresu v dané podsíti.

- 1) 53.0.10.5
- 2) 172.100.12.0
- 3) 26.120.17.254
- 4) 117.192.255.250
- 5) 192.215.112.112
- 6) 128.123.131.255
- 7) 223.136.153.129
- 8) 169.128.158.255

 *Domácí úkol*

Je dána IP adres 201.158.25.35 s maskou podsítě 255.255.255.240.  
Určete adresu podsítě, broadcastu v podsíti, první, druhou, třetí, čtvrtou a poslední použitelnou IP adresu v této podsíti.

 *Shrnutí*

- ✓ Z IP adresy umíte určit její třídu a třídní masku.
- ✓ Umíte určit adresu sítě a broadcastu v dané síti.
- ✓ Umíte vytvořit podsít a určit broadcast v této podsíti.
- ✓ Umíte určit zvolenou použitelnou IP adresu v dané podsíti.

## 20. Topologie, síťová zařízení (39. – 40. hodina)

### 20.1 Topologie sítí

Rozlišujeme dva typy topologií

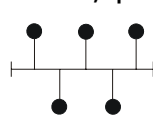
- **fyzické** – podle typu zapojení
- **logické** – podle přístupu zařízení k vysílání dat

#### Logické topologie

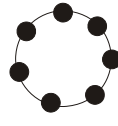
- **broadcast** – žádné zařízení nemá přednost pro vysílání. Pokud chce zařízení vysílat, poslouchá, zda není na médiu provoz a pokud ne, začne vysílat. Může nastat kolize s vysíláním jiného zařízení. Příkladem této logické topologie je Ethernet.
- **token passing** – zařízení si předávají elektronický poukaz pro vysílání. Pokud zařízení nemá nic k vysílání, předá poukaz dále. Příkladem této logické topologie je Token Ring a FDDI (Fiber Distributed Data Interface).

#### Fyzické topologie

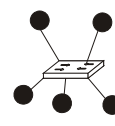
- **sběrnicová (bus)** – zařízení jsou připojena k jednomu kabelu zakončenému na obou koncích
- **kruh (ring)** – zařízení jsou spojena do kruhu jeden k druhému
- **hvězda (star)** – zařízení se připojují k centrálnímu bodu
- **rozšířená hvězda (extended star)** – jednotlivé hvězdy se spojují dohromady např. pomocí hubů nebo switchů
- **hierarchická topologie** – podobná rozšířené hvězdě, ale oproti spojování hubů a switchů dohromady, systém je propojen do počítače, který kontroluje provoz v dané topologii
- **mesh** – každé zařízení je spojeno s každým jiným. Úplná mesh topologie vyžaduje velké množství spojovacích linek, což se ve skutečnosti nedělá, používá se částečná mesh topologie.



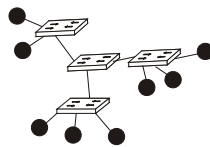
sběrnice



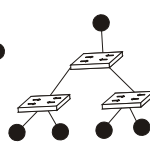
kruh



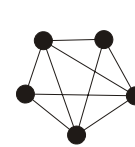
hvězda



rozšířená  
hvězda



hierarchická  
topologie



mesh

### 20.2 OSI model

Pro zopakování uvedme vrstvy OSI modelu (Open System Interconnection):

7. **aplikační** (application) – aplikace, např. e-mail, ftp, www
6. **prezentační** (presentation) – kódování, komprese, převod dat do standardních síťových formátů
5. **relační** (session) – uskutečňuje, udržuje a ukončuje relaci mezi aplikacemi
4. **transportní** (transport) – patří sem protokoly TCP a UDP, datová jednotka je zde segment, do datagramu se přidává informace o zdrojovém a cílovém portu, který určuje, pro kterou aplikaci jsou data určena
3. **síťová** (network) – zajišťuje výběr cesty přes síť, datová jednotka je zde paket, pracuje se zde se zdrojovou a cílovou IP adresou
2. **spojová** (data-link) – řídí tok dat na přenosovém médiu, datová jednotka je zde rámec, přidává se zde informace o zdrojové a cílové MAC adrese
1. **fyzická** (physical) – zajišťuje fyzické propojení a binární přenos, datová jednotka je zde bit. Problémy související s touto vrstvou se týkají kabelů, konektorů, signalizace na médium.

Během přípravy dat na vysílání probíhá přibalování dalších informací. Během zabalování probíhají data od aplikační vrstvy směrem k nižším vrstvám.

V transportní vrstvě se přidají informace o portech, které slouží pro identifikaci aplikace, pro kterou jsou data určena, v síťové vrstvě se přidá zdrojová a cílová IP adresa, ve spojové vrstvě se přidají informace o MAC adresách a kontrolním součtu, fyzická vrstva zakóduje rámec ze spojové vrstvy do binární podoby, 1 a 0 a pošle je na přenosové médium.

Síťová zařízení, přes která data tečou, se s těmito daty vypořádají podle toho, na které vrstvě OSI modelu pracují.

### 20.3 Kolizní a broadcast doména

**Kolizní doména** je část sítě, ve které může docházet ke kolizím. Kolize vzniká tehdy, když na sdílené přenosové médium začnou vysílat alespoň dva počítače v jednu chvíli.

Například počítače připojené na jeden hub jsou všechny součástí jedné kolizní domény, ale počítače připojené na switch jsou v oddělených kolizních doménách (viz níže – síťová zařízení).

**Broadcast doména** je celá síť, tj. tvoří ji všechny počítače se stejnou síťovou adresou. Broadcast je vysílání, které se posílá všem počítačům v síti.

Šíření broadcastu blokuje router, proto router odděluje broadcast domény (viz níže – síťová zařízení).



## 20.4 Síťová zařízení

### 20.4.1 Repeater

**Repeater** je síťové zařízení, které přijme signál na jednom portu, zregeneruje jej a pošle ven druhým portem. Pracuje na fyzické vrstvě OSI modelu, proto regenerace signálu probíhá na bitové úrovni.

Každé síťové zařízení, tedy i hub, přispívá ke spoždění signálu, což má za následek více pozdních kolizí.

**Kolize** nastává tehdy, když více než jeden počítač v danou chvíli začne vysílat data na sdílené médium.

### 20.4.2 Hub

**Hub** je označován jako multiportní repeater. Repeater má většinou jen dva porty, hub většinou 4 až 24 portů. Hub se využívá k vytvoření hvězdicovité topologie.



Hub je síťové zařízení, které pracuje na první vrstvě OSI modelu. Nedělá žádná rozhodnutí o tom, kam datagramy posílat.

Pošle jej všemi porty s výjimkou toho, odkud datagram přišel.

Rozlišuje tři typy hubů:

- ❑ **pasivní** – pro svůj provoz nepotřebuje elektrický proud, signál neregeneruje ani nezesiluje, je využíván jen pro sdílení přenosového média.
- ❑ **aktivní** – potřebuje elektrický proud, signál zesiluje a regeneruje.
- ❑ **inteligentní** – funguje jako aktivní hub, navíc umí provádět různé diagnostiky stavů.

Protože hub posílá data všemi porty s výjimkou příchozího, zvětšuje se jeho použitím v síti kolizní doména, což je prostor, ve kterém může docházet ke kolizím.

**Kolize** nastává tehdy, když více než jeden počítač v danou chvíli začne vysílat data na sdílené médium.

### 20.4.3 Bridge

**Bridge** je síťové zařízení, které dělá rozhodnutí, kam poslat rámeček, podle cílové MAC adresy. Rozhodnutí probíhají na softwarové úrovni, což je pomalejší než u switchu, kde tato rozhodnutí probíhají na hardwarové úrovni.

Bridge se používá pro rozdělení sítě na segmenty.  
Rozděluje síť na různé kolizní domény.

#### 20.4.4 Switch

**Switch** je síťové zařízení, které dělá rozhodnutí, kam poslat datagram, podle cílové MAC adresy.



Po spuštění je switch po určitou dobu ve stavu, kdy neví, na jakém portu má připojený jaký počítač, resp. nemá udělanou vazbu mezi portem a MAC adresou připojeného zařízení.

Pokud v tuto chvíli přijde datagram a switch neví přesně, kam jej poslat, pošle jej všemi porty s výjimkou toho, odkud datagram přišel.

Z probíhajícího provozu se switch učí, na kterém portu je připojené jaké zařízení. Přiřazuje si porty ke zdrojovým MAC adresám, které jsou uvedeny v datagramech, které přes něj probíhají.

Zjištěné informace si ukládá do **CAM** (Content Adressable Memory) tabulky.

Pokud ve chvíli, kdy má switch informace o vazbách port – MAC adresa, přijde datagram pro určitou cílovou MAC adresu, podívá se switch do své CAM tabulky a pošle datagram jen tím portem, se kterým je svázána cílová MAC adresa.

Rozděluje síť na různé **kolizní domény**.

Na každém portu switche je jedna kolizní doména. Znamená to, že pokud jeden počítač pošle data jinému počítači, pak se data posílají jen cílovému počítači a ostatním počítačům se tato data neposílají. Zvyšuje to přenosovou kapacitu linky pro všechny počítače, může probíhat paralelní komunikace mezi různými počítači v jednu chvíli, aniž by docházelo ke kolizím (jak je tomu např. u hubu).

**Switch neoděluje broadcast domény.** To znamená, že pokud počítač v síti pošle broadcast, pak se tento broadcast dostane ke všem počítačům v dané síti (ke všem se stejnou síťovou adresou).

Přepínání switche probíhá na hardwarové úrovni, proto je rychlejší než přepínání bridge, které probíhá na softwarové úrovni.

Na rozdíl od bridge má switch i další specifické schopnosti, například některé switche umí STP – Spanning Tree Protocol.

**Spanning Tree Protocol – STP** – slouží k tomu, aby na síti LAN nevznikaly okruhy (i když jsou síťová zařízení zapojena do okruhu, například kvůli záložním linkám, které se aktivují v případě vypnutí původní linky), kterými by rámec cestoval stále kolem dokola a zahlcovala by se tak síť.

Switche mohou podporovat **virtuální síť – VLAN**. Některé porty mohou patřit do jedné VLAN, jiné porty do jiné VLAN atp.

Pokud chtějí počítače v různých VLAN spolu komunikovat, musí k tomu použít router, který provoz mezi různými VLAN přesměruje. To může přispívat ke zvýšení bezpečnosti na síti a zvýšení přenosové kapacity na jednotlivých segmentech sítě.

### 20.4.5 Router

**Router** je síťové zařízení, které pracuje na třetí vrstvě OSI modelu, na síťové vrstvě.



Pro svá rozhodnutí, kam poslat paket, používají informaci o cílové IP adrese obsažené v datagramu.

Router odděluje sítě, na dvou portech routeru nemůže být stejná síť, protože pak by router nevěděl, na který port směřovat příchozí paket.

Router si vede směrovací tabulku, ve které má informace, jakým portem se do cílových sítí dostane. Tyto informace se do směrovací tabulky konfiguruje buď staticky (administrátorem) nebo se je router učí dynamicky od ostatních routerů pomocí určitých směrovacích protokolů.

Protože router odděluje sítě, **odděluje i broadcast domény**, tím spíše kolizní domény.

Broadcast je vázaný na síť a kolize může vzniknou na sdíleném médiu (např. s použitím hubů).

### 20.5 Zpoždění v síti

Při šíření signálu v síti dochází ke zpoždění. Je to doba mezi tím, kdy rámec začne opouštět zdrojové zařízení a chvílí, kdy jeho první část dorazí k cílovému zařízení.

K tomuto zpoždění přispívají různé faktory:

- přenosové médium – omezené maximální možnou rychlostí šíření signálu
- zpoždění způsobená síťovými zařízeními, které musí dělat rozhodnutí, kam poslat rámec. Zařízení jako switch a router musí rozbalit příchozí rámec a identifikovat pro ně podstatné informace o cílové adrese. Router přispívá větším zpožděním než switch, protože musí rozbalit příchozí data až do vrstvy 3.

 *Domácí úkol*

Zopakujte si topologie sítí, vrstvy OSI modelu, datové jednotky pracující v jednotlivých vrstvách OSI modelu, síťová zařízení a jejich použití, rozdíl mezi broadcast a kolizní doménou a příčiny zpoždění v síti.

 *Shrnutí*

- ✓ Umíte jmenovat logické topologie v síti.
- ✓ Umíte jmenovat fyzické topologie v síti.
- ✓ Znáte OSI model, jeho vrstvy, datové jednotky, zařízení pracující ve vrstvách 1 – 3.
- ✓ Znáte síťová zařízení, jejich použití a vlastnosti.
- ✓ Znáte příčiny zpoždění v síti a důsledky.
- ✓ Víte, co je kolizní a broadcast doména.

## 21. Výroba UTP kabelů (41. – 42. hodina)

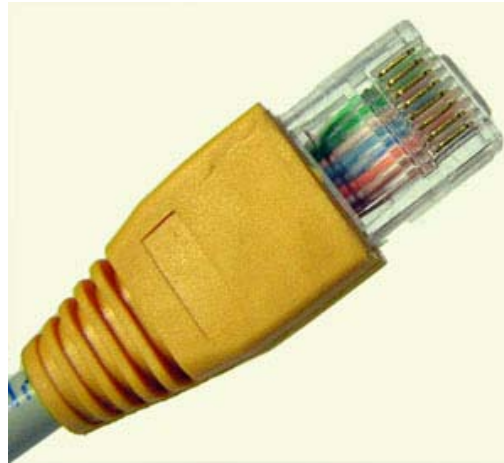
### 21.1 Koncovka typu A a B

UTP kabel obsahuje 4 páry drátů odlišených barvami. V páru je vždy jeden plně barevný a jeden polobarevný.

Barvy:

- zelená
- oranžová
- modrá
- hnědá

Pořadí drátů popisujeme v pohledu ze spodní části koncovky a jakobychom ji drželi za kabel.



#### 21.1.1 Koncovka typu A – pořadí drátů

- polozelená
- zelená
- polooranžová
- modrá
- polomodrá
- oranžová
- polohnědá
- hnědá

#### 21.1.2 Koncovka typu B – pořadí drátů

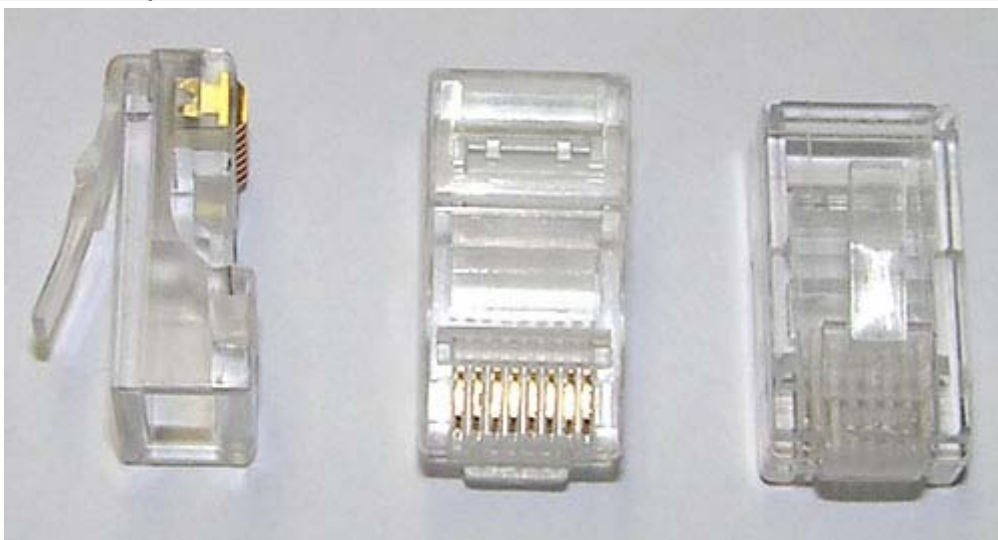
- polooranžová
- oranžová
- polozelená
- modrá
- polomodrá
- zelená
- polohnědá
- hnědá

## 21.2 Pomůcky pro výrobu

### 21.2.1 Krimpovací kleště



### 21.2.2 Koncovky RJ-45



### 21.2.3 UTP kabel

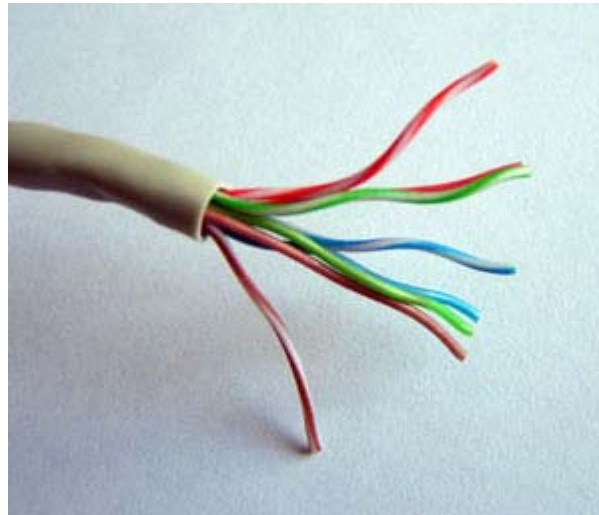
UTP kabel se skládá ze čtyř párů drátů barevně odlišených. Barvy jsou popsány u popisu koncovek.

Dráty v párech jsou smotány. Smotání má příznivý vliv na přenos signálu. Proto rozmotáváme dráty při výrobě koncovky jen v části koncovky.

Pokud párem drátů vede signál, pak jedním drátem je signál vysílán a druhým přijímán. U sítí 10 Mb/s nebo 100 Mb/s je signál veden zeleným a oranžovým párem. Ostatní dva páry jsou nevyužité.

U sítí 1 Gb/s jsou využity všechny 4 páry a požadavky na kvalitu kabelu se zvyšují.

UTP kabel je kategorizován do kategorií kvality. Pro přenos 1 Gb/s je vhodný kabel kategorie 5e a vyšší, pro přenos 100 Mb/s stačí kategorie 5.



Délka jednoho segmentu kabelu mezi zařízeními by neměla přesahovat 100 m, po delších vzdálenostech by již docházelo k nepříjemným zeslabením signálu a mohlo by docházet i pozdním kolizím. Segmenty je možné propojovat aktivními prvky, které signál zregenerují a pošlou dál.

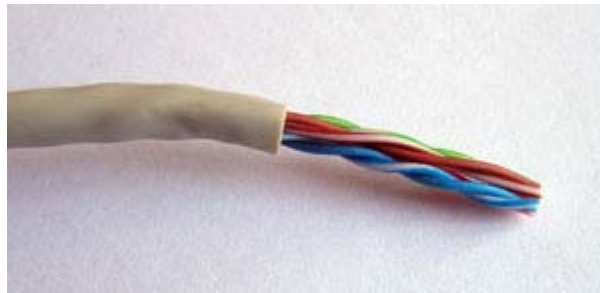


### 21.3 Postup výroby koncovky

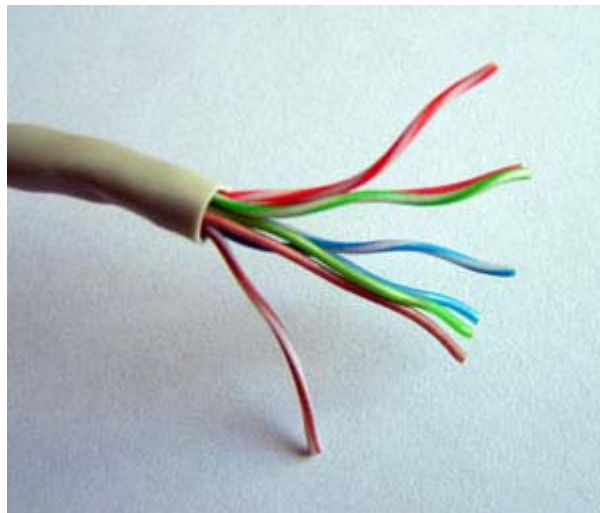
Pomocí ostří na krimpovacích kleštích nařízneme plastový obal kabelu. Méně je zde více. Naříznete-li kabel příliš, proříznete i barevné plastové obaly jednotlivých drátů. Ty se pak mohou v naříznutí zlomit nebo to může mít nepříznivý vliv na přenos signálu.



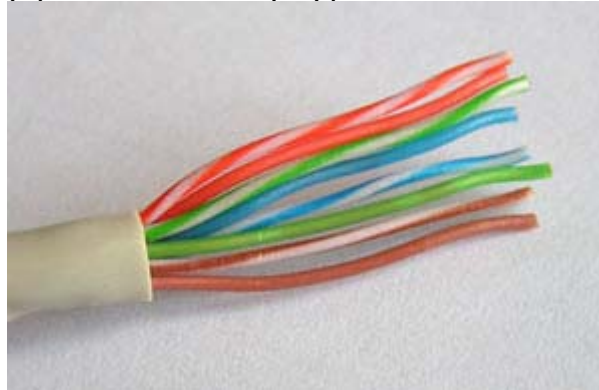
Plastový obal stáhneme.



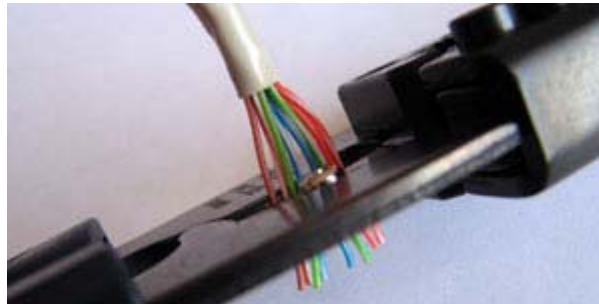
Jednotlivé dráty rozmotáme.



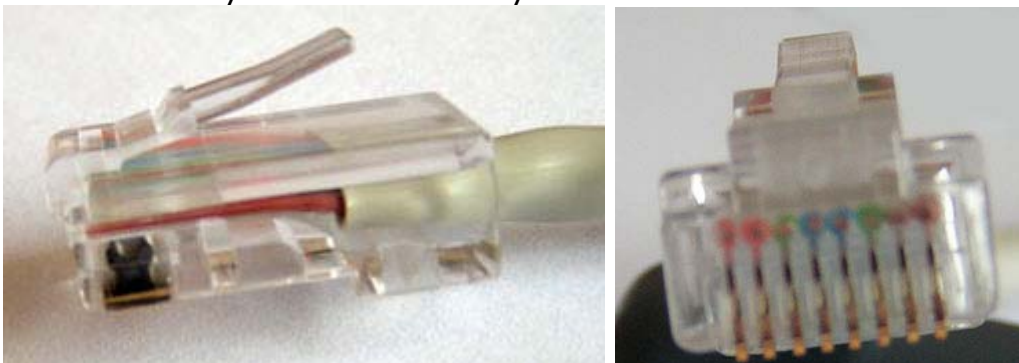
Jednotlivé dráty narovnáme a srovnáme podle typu koncovky. Na obrázku jsou dráty srovnány podle koncovky typu B.



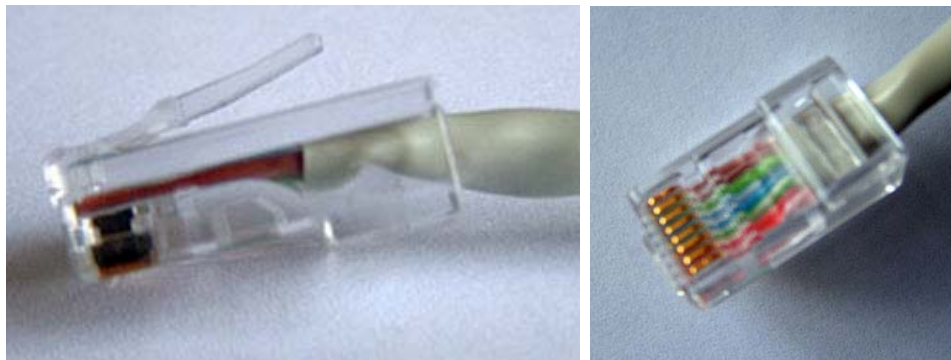
Dráty krimpovacími kleštěmi ucvakneme v přiměřené vzdálenosti (zůstává cca 1,5 cm).



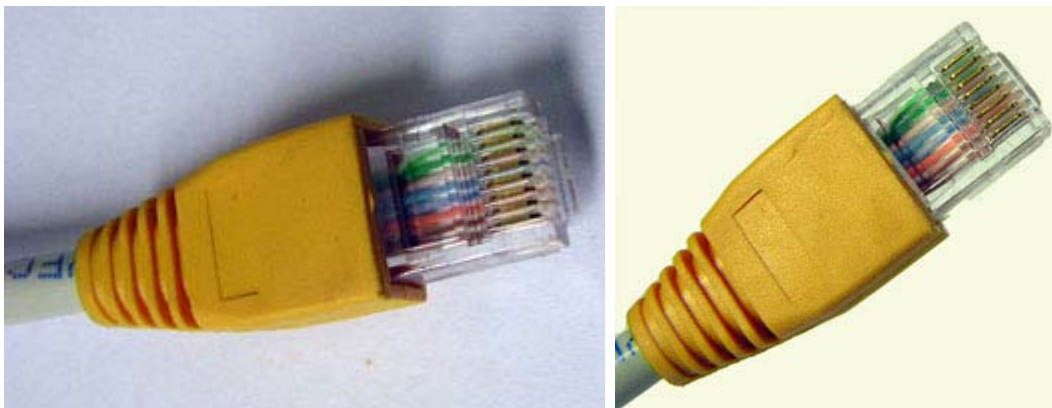
Dráty zasuneme do koncovky RJ-45. Musíme je zastrčit až do konce, aby se nožíky na konci koncovky po secvaknutí zařizly do drátů. Pokud ponecháme délku drátů příliš malou, nožíky se nebudou moci do ničeho zařiznout, pokud bude délka příliš velká, nebude vnější plastový obal po secvaknutí koncovky koncovkou zachycen.



Koncovku zastrčíme do krimpovacích kleští a secvakneme. Nožíky se zařiznou do drátů a plastový zobáček přichytí vnější obal kabelu.



Na kabel je možné ještě před zastrčením do koncovky navléci krytku.



#### 21.4 Přímý kabel

Přímý kabel je takový, který má na obou koncích stejný typ koncovky, např. oba konce mají koncovky typu A nebo oba mají koncovky typu B.

#### 21.5 Křížený kabel

Křížený kabel je takový, který má na svých koncích rozdílný typ koncovek, např. jeden konec má koncovku typu A a druhý konec má koncovku typu B.

## 21.6 Rollover – konzolový kabel

Konzolový kabel má na jednom konci koncovku typu A nebo B a na druhém konci zrcadlový obraz zvolené koncovky.

### 21.6.1 Př.

První koncovka má pořadí barev drátů:  
polozelený, zelený, polooranžový, modrý, polomodrý, oranžový, polohnědý, hnědý.

Druhá koncovka má opačné pořadí barev:  
hnědý, polohnědý, oranžový, polomodrý, modrý, polooranžový, zelený, polozelený.

## 21.7 Použití kabelů

Pro spojení zařízení podobného typu se používá křížený kabel, pro spojení zařízení rozdílného typu se používá přímý kabel.

### 21.7.1 Př.

**Přímý kabel** se používá pro spojení následujících zařízení:

PC (nebo server) – switch

PC (nebo server) – hub

router – switch

router – hub

PC (nebo server) – bridge

router – bridge

**Křížený kabel** se používá pro spojení následujících zařízení:

PC (nebo server) – PC (nebo server)

PC (nebo server) – router

switch – switch

switch – hub

hub – hub

router – router (přes ethernetové zásuvky)

**Konzolový (rollover) kabel** se používá pro základní konfigurování routerů, switchů přes konzolový port.

Na jedné straně se kabel zapojí do sériového portu počítače (pokud jsme si kabel vyrobili sami, má na obou koncích zástrčku RJ-45 a pro zastrčení do sériového portu budeme potřebovat transceiver) a na druhé straně se zapojí do konzolového portu routeru nebo switche.

## 21.8 Měření kabelů

Pro měření kabelů existuje celá řada více či méně inteligentních měřáků. Od jejich přesnosti, citlivosti a různých dovedností se také odvíjí jejich cena.

Cena se pohybuje řádově od jednotek do stovek tisíců.

Na obrázku vidíte měřák z levnější řady. Dokáže změřit funkčnost přímého a kříženého UTP kabelu. Skládá se ze dvou oddělitelných částí, do kterých se zapojují konce kabelu. Proto lze proměřit např. i zalištovaný kabel, který nemá své dva konce u sebe.



## 21.9 Napojení kabelů

Pokud potřebujeme prodloužit kabel, můžeme použít spojku. Je to pasivní prvek, který neregeneruje signál.



## 21.10 Cvičení

- Vytvořte přímý UTP kabel.
- Vytvořte křížený UTP kabel.
- Vytvořte rollover kabel.

Pomocí měřáku ověřte funkčnost kabelů.

Vizuálně zkontrolujte, že jsou koncovky nainstalovány správně, že jsou dráty vsunuty až do konce koncovek, vzájemně jsou rovnoběžné a vnější obal kabelu je zachycen plastovým zobáčkem koncovky.

---

 *Domácí úkol*

Zopakujte si barevná schémata koncovek typu A a B a jejich instalaci.  
Zopakujte si výrobu přímého, kříženého, rolloveru kabelu a jejich použití.

---

 *Shrnutí*

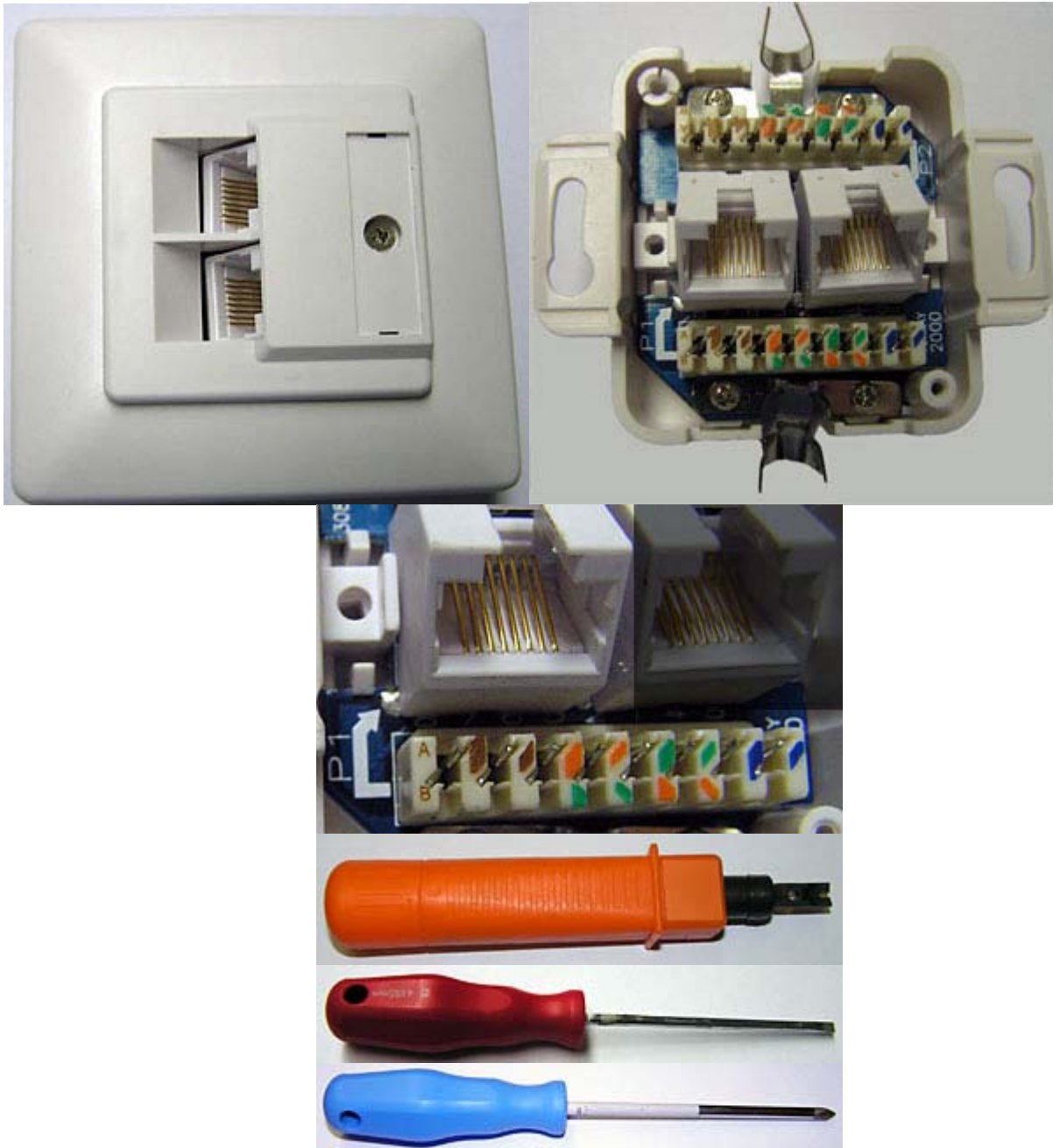
- ✓ Znáte typy koncovek RJ-45.
- ✓ Umíte správně vytvořit přímý, křížený a rollover kabel.
- ✓ Víte, k čemu se UTP kabely používají a jaký typ musíme zvolit pro spojení určitých zařízení.

## 22. Instalace zásuvek, patchpanely (43. – 44. hodina)

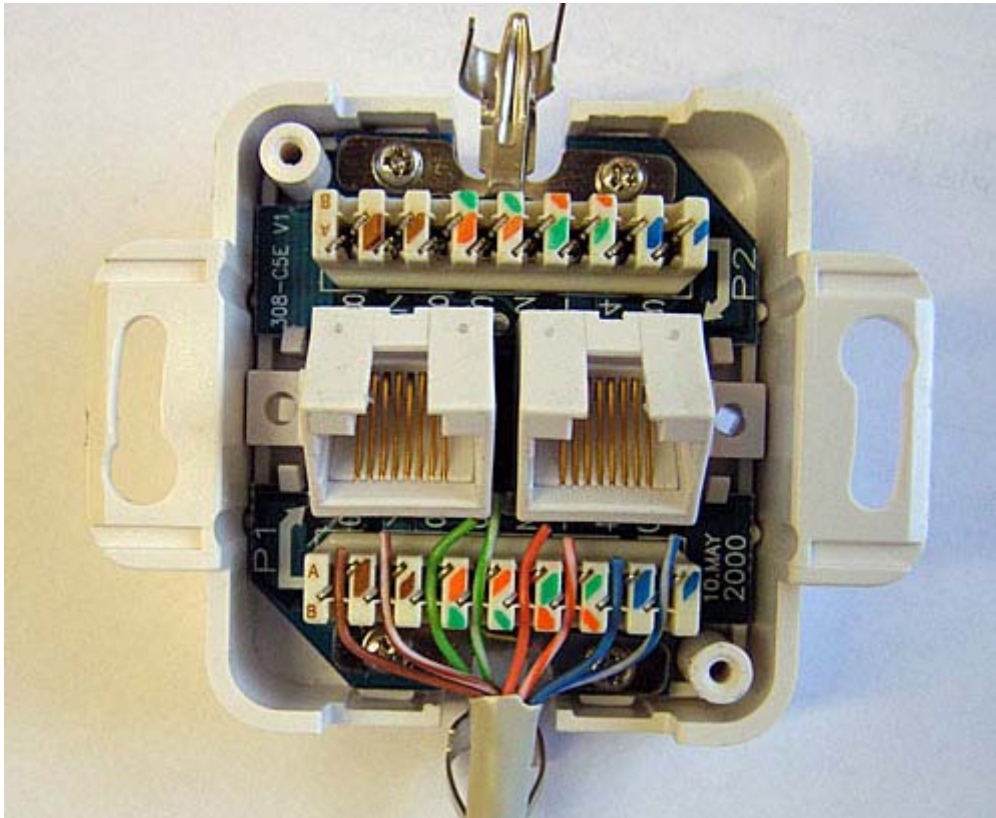
### 22.1 Zásuvka RJ-45

Pro instalování zásuvek potřebujeme kromě zásuvek a UTP kabelu šroubovák a narážečku.

Šroubovák pro odstranění šroubků a narážečku pro naražení drátů UTP kabelu do míst k tomu určených v zásuvce.







Na obrázku jsou dráty UTP kabelu nachystány k naražení. V tomto případě je vyráběna levá zásuvka (je to poznat ze šipky vedoucí od barevně vyznačených míst k levé zásuvce) a zásuvka bude typu B (to je poznat z toho, že umístění drátů se řídí podle spodní poloviny barevného schématu a to je označeno písmenem B).

Pokud by zásuvka nebyla takto zřetelně popsána, museli bychom například pomocí měření odporu zjistit, kam který drát vede.

Podle barevného schématu vyznačeného na zásuvce nachystáme jednotlivé dráty UTP kabelu do samořezných kontaktů zásuvky a narážkou je do kontaktů zarazíme.

## 22.2 Patch panel

Patch panel je sada RJ-45 zásuvek umístěných v jednom panelu. Na obrázku je 24 – portový patch panel, jedná se o patch panel montovatelný do rozvaděčové skříně.

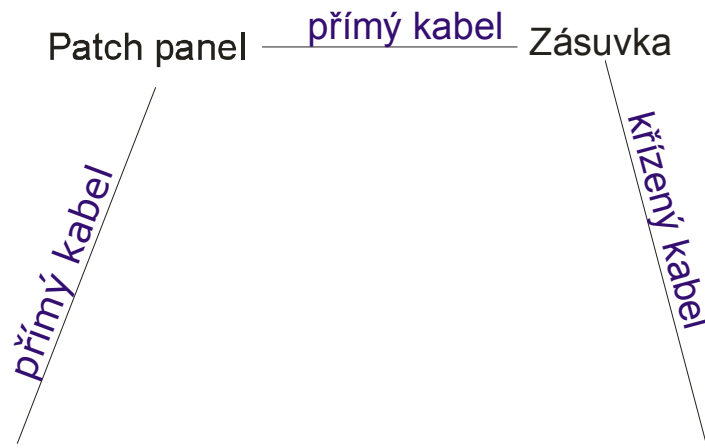


Instalace patch panelu se velmi podobá instalaci zásuvky RJ-45. Na obrázku je vidět, že jednotlivé samořezné kontakty pro naražení drátů UTP kabelu jsou označeny barevně i čísly. Čísla odpovídají rozložení drátů v koncovkách typu A nebo B.

V našem případě čísla 5, 4, 1, 2, 3, 6, 7, 8 odpovídají u zásuvky typu A drátům: polomodrý, modrý, polozelený, zelený, polooranžový, oranžový, polohnědý, hnědý a u zásuvky typu B drátům: polomodrý, modrý, polooranžový, oranžový, polozelený, zelený, polohnědý, hnědý.

### 22.3 Cvičení - komplexní konstrukce

- 1) Vytvořte přímý kabel.
- 2) Vytvořte křížený kabel.
- 3) Propojte přímým kabelem patch panel a zásuvku RJ-45.
- 4) Do patch panelu zapojte váš přímý kabel a do zásuvky váš křížený kabel.
- 5) Proměřte funkčnost instalace.



 *Domácí úkol*

- ❑ Zopakujte si barevné schéma RJ-45 koncovek typu A a B.
- ❑ Zopakujte si vytváření přímého, kříženého a rolloveru kabelu.
- ❑ Zopakujte si vytváření RJ-45 zásuvek.
- ❑ Zopakujte si instalaci zásuvky v patch panelu.

 *Shrnutí*

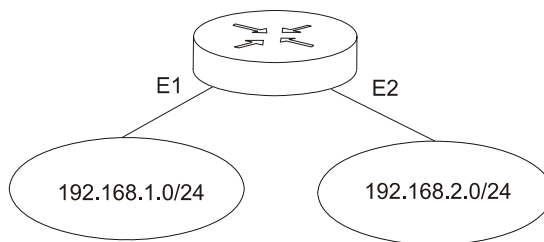
- ✓ Umíte nainstalovat zásuvku RJ-45.
- ✓ Umíte nainstalovat zásuvku v patch panelu.
- ✓ Umíte vytvořit přímý, křížený a rollover kabel a umíte ověřit jeho funkčnost.

## 23. Podsítě, konstrukce kabelů, zásuvek – prohlubování znalostí (45. - 46. hodina)

### 23.1 Cvičení

- 1) Technik přidává do existující sítě switch a počítač. Před připojení počítače ke switchi zjistí, že kabel má udělány koncovky tak, že jedna je zrcadlovým obrazem druhé. Jaký je to kabel? Je vhodný pro připojení PC ke switchi?
  - a. křížený
  - b. přímý
  - c. rollover
  - d. sériový
- 2) Kolik broadcast domén je v síti složené z 15 počítačů, 2 serverů, 4 switchů, 2 síťových tiskáren?
  - a. 1
  - b. 23
  - c. 21
  - d. 4

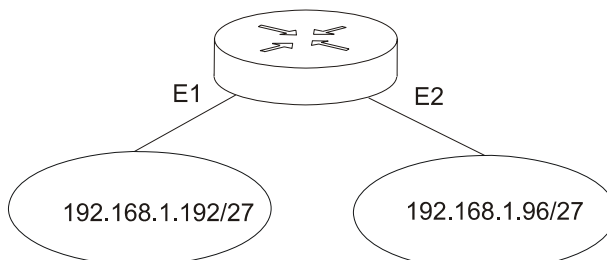
3)



Jaké IP adresy můžeme přiřadit rozhraním E1 a E2 na routeru?

- a. E1: 192.168.2.1  
E2: 192.168.1.1
- b. E1: 192.168.2.1  
E2: 192.168.2.2
- c. E1: 192.168.1.1  
E2: 192.168.1.2
- d. E1: 192.168.1.1  
E2: 192.168.2.1

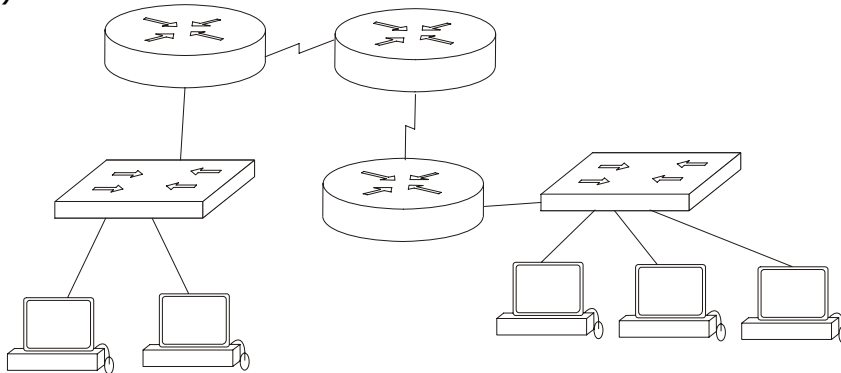
4)



Jaké IP adresy můžeme přiřadit rozhraním E1 a E2 na routeru?

- a. E1: 192.168.1.224  
E2: 192.168.1.97
  - b. E1: 192.168.1.193  
E2: 192.168.1.97
  - c. E1: 192.168.1.224  
E2: 192.168.1.255
  - d. E1: 192.168.1.220  
E2: 192.168.1.68
- 5) Ke které podsíti patří IP adresa 172.27.90.1/19?
- a. 172.27.90.0
  - b. 172.27.0.0
  - c. 172.27.64.0
  - d. 172.27.96.0
- 6) Jaké tvrzení ohledně operace ping je pravdivé?
- a. ping otestuje funkčnost spojení na úrovni nevyšší datové vrstvy OSI modelu
  - b. ping otestuje funkčnost spojení na úrovni nejvyšší síťové vrstvy OSI modelu
  - c. ping se používá ve spojení s fyzickou adresou počítače
  - d. ping na adresu 172.0.0.1 se označuje za loopback test
- 7) Jaký kabel je vhodný pro připojení počítače ke switchi?
- a. přímý
  - b. křížený
  - c. rollover
  - d. seriový
- 8) Jaký kabel je vhodný pro připojení počítače ke switchi pro konzolové spojení?
- a. přímý
  - b. křížený
  - c. rollover
  - d. seriový
- 9) Vytvoříte-li UTP kabel kategorie 5 s koncovkami RJ-45, jedna podle standardu T568A a druhá podle standardu T568B, je to kabel
- a. přímý
  - b. křížený
  - c. rollover
  - d. sériový
- 10) Co hledá switch ve svém CAM tabulce, aby mohl přepnout rámec na správně rozhraní?
- a. IP adresu cíle
  - b. MAC adresu cíle
  - c. IP adresu zdroje
  - d. MAC adresu zdroje

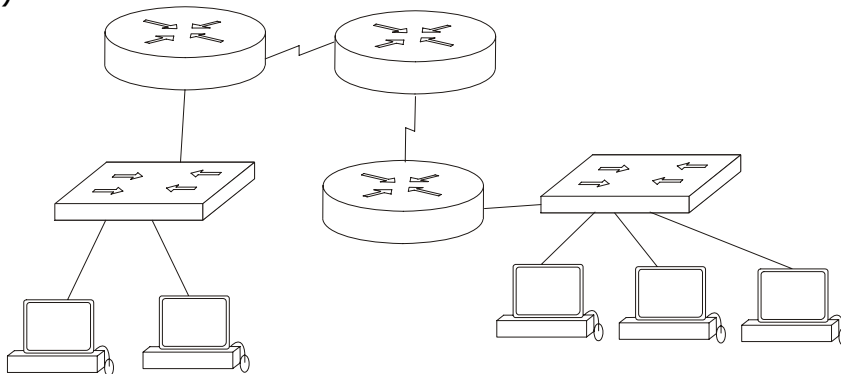
11)



Kolik kolizních domén je na obrázku?

- a. 7
- b. 8
- c. 9
- d. 1

12)



Kolik broadcast domén je na obrázku?

- a. 1
- b. 3
- c. 4
- d. 6

13) Co je to latence v síti?

- a. ztráta síly signálu při cestě od zdroje k cíli
- b. nežádoucí rušení v síti
- c. doba, kterou potřebuje paket při své cestě od zdroje k cíli
- d. dělení se o šířku pásma mezi počítači

14) Z IP adresy třídy B chceme udělat podsítě. Jaká bude optimální maska podsítě, chceme-li, aby měla adresní prostor pro 300 počítačů?

- a. /9
- b. /23
- c. /16
- d. /25

- 15) Máme-li podsít 172.27.192.0 /19, jaký je rozsah použitelných adres?
- 172.27.193.0 – 172.27.193.255
  - 172.27.192.0 – 172.27.192.255
  - 172.27.192.1 – 172.27.223.254
  - 172.27.192.0 – 172.27.223.255
- 16) K jaké síti patří IP adresa 172.27.217.1 s maskou 255.255.224.0?
- 172.27.0.0
  - 172.27.192.0
  - 172.27.192.1
  - 172.27.255.0
- 17) Jaká maska odpovídá zápisu /26?
- 255.255.255.192
  - 255.255.255.0
  - 255.255.0.0
  - 255.255.192.0
- 18) Jaká je IP adresa broadcastu v síti 172.16.168.0 /21?
- 172.16.175.255
  - 172.16.168.255
  - 172.16.175.0
  - 172.16.0.255
- 19) Kolik použitelných IP adres bude mít síť 192.168.10.192 /28?
- 10
  - 12
  - 14
  - 16
- 20) Jaká bude adresa broadcastu v síti 192.168.10.192 /28?
- 192.168.10.193
  - 192.168.10.207
  - 192.168.255.255
  - 192.168.10.255
- 21) Jaká bude adresa sítě pro IP adresu 192.168.10.68 /28?
- 192.168.10.69
  - 192.168.10.65
  - 192.168.10.64
  - 192.168.10.0
- 22) IP adresa třídy B začíná dekadicky číslem
- 127
  - 128
  - 192
  - 224



- 23) Rozhraní na routeru má přiřazenou IP adresu 172.16.10.10 a masku 255.255.248.0. Jakou IP adresu můžeme přiřadit počítači v této síti?
- a. 172.16.8.255
  - b. 172.16.16.1
  - c. 172.16.4.25
  - d. 172.16.2.10
- 24) Vytvořte přímý kabel.
- 25) Vytvořte křížený kabel.
- 26) Vytvořte síťovou zásuvku.
- 27) Vytvořte zásuvku v patch panelu.

 **Domácí úkol**

Zopakujte si vrstvy OSI modelu, pojmy ze sítí, konstrukci UTP kabelu s koncovkami RJ-45, broadcast a kolizní domény, síťová zařízení a jejich úloha v síti, výpočty z IP adresy a masky, podsítě.

 **Shrnutí**

- ✓ zopakovali jste si výpočty podsítí
- ✓ umíte specifikovat třídy IP adres
- ✓ víte, k čemu slouží maska
- ✓ umíte určit adresu broadcastu a sítě
- ✓ chápete pojmy kolizní a broadcast doména
- ✓ znáte základní síťové prvky
- ✓ umíte vyrobit koncovky RJ-45
- ✓ umíte vyrobit síťovou zásuvku a patchpanel

 **Řešení otázek**

1c - ne, 2a, 3d, 4b, 5c, 6b, 7a, 8c, 9b, 10b, 11c, 12c, 13c, 14b, 15c, 16b, 17a, 18a, 19c, 20b, 21c, 22b, 23a

## 24. MS Windows Server 2003 (47. – 48. hodina)

### 24.1 Verze MS Windows Serveru 2003

- Standard Edition
- Enterprise Edition
- Datacenter Edition
- Web Edition
- Small Business Server Edition

#### 24.1.1 Popis verzí

##### **Web Edition**

Koncipováno speciálně pro využití jako web server. Počítače s touto verzí operačního systému mohou být členy domény s Active Directory, ale nemohou být řadičem domény, nelze na nich spustit Active Directory. Poměrně drahá edice, ale není třeba řešit otázku licencování připojení (CAL).

##### **Small Business Server Edition**

Určeno pro malé společnosti do 75 počítačových stanic s jedním serverem. Obsahuje nástroje pro e-mail, sdílení dokumentů, zvýšení zabezpečení pro přístup k internetu, úložiště dat, tisk a faxování, vzdálenou správu jednoho serveru.

Lze na něm provozovat web server, souborový server, řadič domény.

##### **Standard Edition**

Určeno pro využití menšími společnostmi. Spolehlivý síťový operační systém. Vhodný, pokud nevyžadujeme zvýšenou podporu hardwaru a clustrování, což nabízí verze Enterprise.

Lze na něm provozovat web server, souborový server, řadič domény.

##### **Enterprise Edition**

Určeno pro střední až velké společnosti. Nabízí stejné nástroje jako Standard Edition, navíc nabízí zvýšenou dostupnost, rozšiřitelnost, spolehlivost a vyšší výkon.

Vhodné pro velmi výkonné servery. Doporučeno pro servery, na kterých běží aplikace pro sítě, databáze, elektronický obchod.

Umožňuje clustrování (například 2 servery se navenek jeví jako 1).

##### **Datacenter Edition**

Určeno pro velké společnosti.

Vhodné pro aplikace, které vyžadují vysokou dostupnost a možnost velké rozšiřitelnosti. Podpora více procesorů a velké paměti.

## 24.2 Licence

Pro legální provoz sítě sestávající ze serverů a stanic musíme vlastnit licence jak pro provoz serverů a stanic, tak i pro spojení mezi serverem a stanicí.

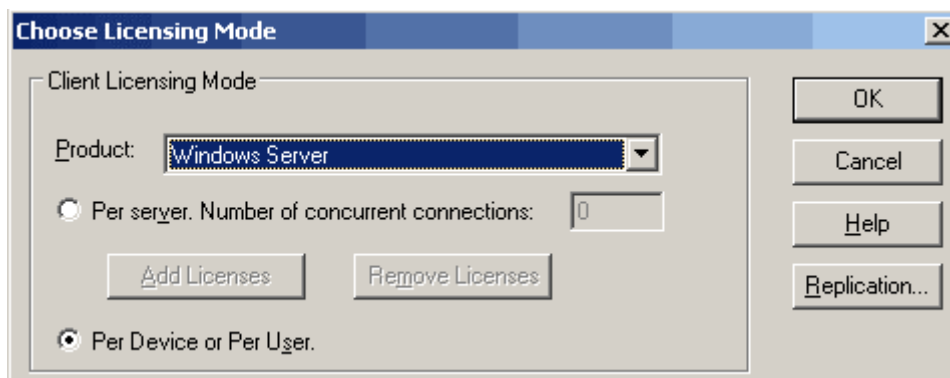
**CAL** (Client access license) – licence spojení

### 2 typy

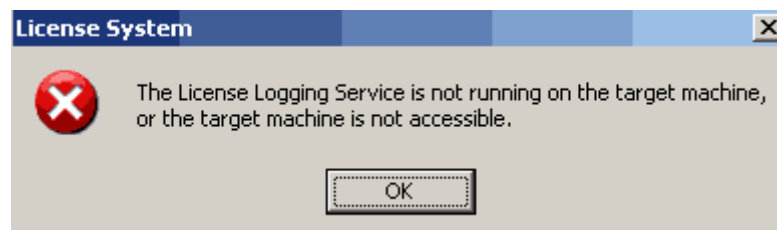
- **Per server** – každá konekce na server vyžaduje licenci.
- **Per device or Per User** – každý klient potřebuje licenci
  - licence spojení vázána na **uživatele** – vhodná, pokud je menší počet uživatelských účtů než stanic, na které se hlásí
  - licence spojení vázána na **zařízení** – vhodná, pokud se na stanice hlásí větší počet uživatelů

### 24.2.1 Jak zjistit licenční režim

Start – Control Panel – Licensing



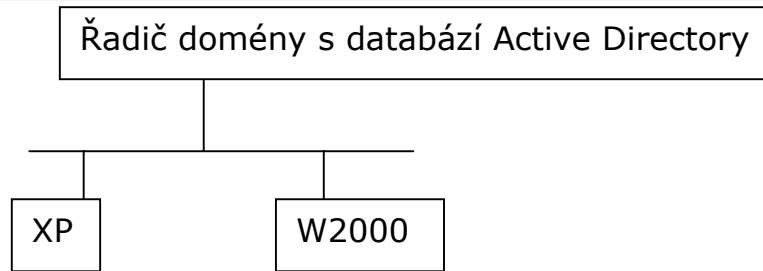
Pokud chceme změnit typ licencování na „**Per server**“ a přidat licence, objeví se chybové hlášení, které říká, že musíme ve službách povolit „**Licence Logging**“.



## 24.3 Doména, pracovní skupina

Při instalaci počítače můžeme zvolit, zda jej vkládáme do existující domény nebo do pracovní skupiny. Toto nastavení lze změnit i dodatečně.

### 24.3.1 Doména



Na řadiči domény je databáze Active Directory, ve které jsou definovány například doménové účty uživatelů, počítačů, organizační jednotky.

#### 2 typy účtů:

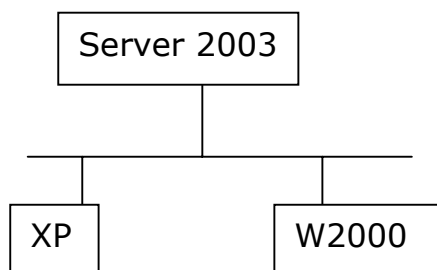
- lokální
- doménové

Když se uživatel hlásí ze stanice pod svým uživatelským účtem k doméně, probíhá jeho ověření vůči databázi Active Directory, která je umístěna na řadiči domény, kterým může být Windows Server 2003.

Jestliže se uživatel hlásí lokálně k počítači, probíhá jeho ověření proti lokální **SAM (Security Account Manager)** databázi, ve které jsou uložena přihlašovací jména a hesla uživatelů v zašifrované formě.

Na řadič domény se nelze hlásit lokálně. Ověření probíhá vůči Active Directory. Pokud je Active Directory porouchaná, je problém se přihlásit.

### 24.3.2 Pracovní skupina



Pokud jsou počítače připojeny do pracovní skupiny (**Workgroup**), pak jsou vůči sobě ve vztahu **Peer – to – peer**, tzn. jsou rovnocenné.

Ověřování při přihlášení probíhá lokálně vůči SAM databázi. Pokud chce uživatel přistupovat ke zdrojům umístěným na jiném počítači v pracovní skupině, musí na něm mít vytvořen svůj účet. Není zde centralizovaná správa účtů jako v doméně s Active Directory.

---

## ☞ Poznámka

---

Je-li účet nazván stejně lokálně i v doméně, jsou to jiné účty.

## 24.4 Instalace

Pokud instalujeme první server ve vznikající doméně, nelze jej rovnou přiřadit do domény, proto jej během instalace přiřadíme do pracovní skupiny (Workgroup).

Instalace domény je dodatečná záležitost.

### 24.4.1 Product key

Product key lze dodatečně zjistit i změnit například pomocí utility **keyfinder.exe**, kterou lze najít na webu <http://www.magicaljellybean.com/>.

Funguje na operačních systémech Windows 95, 98, ME, NT4, 2000, XP, Server 2003.

### 24.4.2 Administrátorské heslo při instalaci

Během instalace serveru 2003 ještě nefungují bezpečnostní politiky pro vytváření hesel, proto lze zadat administrátorovi i prázdné heslo.

Heslo, které je v souladu s bezpečnostní politikou, obsahuje aspoň 7 znaků, obsahuje nejméně 3 ze 4 sad (malá písmena, velká písmena, číslice, speciální znaky).

Toto heslo je třeba při některých záchranných akcích, například při poruše Active Directory a následném obnovování.

### 24.4.3 Nastavení sítě

Během instalace máme možnost dvojího nastavení sítě:

- ❑ typické – v síti musí běžet DHCP
- ❑ vlastní (custom) – síťové parametry zadáváme ručně

---

## ☞ Poznámka

---

Síťové parametry můžeme vypsát příkazem

**ipconfig /all**

Pokud máme síťové parametry nastavovány pomocí DHCP, můžeme je uvolnit příkazem

**ipconfig /release**

O nové nastavení zažádáme příkazem

**ipconfig /renew**

#### 24.4.4 APIPA

**APIPA – Automatic Private IP addressing** – umožňuje DHCP klientovi samostatně si nakonfigurovat IP adresu a masku podsítě, pokud není dostupný server DHCP.

IP adresa je z rozsahu 169.254.0.1 – 169.254.255.254.

Maska podsítě je třídy B – 255.255.0.0.

Klient používá tuto adresu do té doby, než je DHCP server opět dostupný.

Služba APIPA pravidelně kontroluje přítomnost DHCP serveru – každých 5 minut. Pokud zjistí, že DHCP server je opět dostupný, zastaví svůj běh a síťové parametry jsou přenastaveny dynamicky nastavenými parametry z DHCP serveru.

#### 24.4.5 Gateway

**Gateway – brána** – IP adresa rozhraní v síti, na které je směrována komunikace směřující mimo lokální segment podsítě.

Pro komunikaci dvou počítačů na jednom lokálním segmentu podsítě není brána potřebná.

Příkazem **route print** lze vypsat brány. Ve výpisu je vidět řádek podobný tomuto

| Cíl v síti | Síťová maska | Brána      | Rozhraní    | Metrika |
|------------|--------------|------------|-------------|---------|
| 0.0.0.0    | 0.0.0.0      | 62.77.98.1 | 62.77.98.10 | 20      |

Znamená to, že pokud je komunikace směrována na neznámý cíl (0.0.0.0 0.0.0.0), bude z rozhraní 62.77.98.10 směrována na bránu 62.77.98.1.

#### 24.4.6 DNS

**DNS – Domain Name System** – zajišťuje překlad mezi IP adresami a jejich jmennými ekvivalenty (a naopak).

Správná konfigurace DNS je klíčová. Pokud je špatně nakonfigurováno, nemusí fungovat například přihlašování do domény.

V síťových nastaveních, kde se nastavuje DNS, je možnost zadat **primární DNS** a **sekundární DNS**.

Primární DNS by měl být lokální server. Sekundární DNS přijde na řadu, až když *nereaguje* primární DNS. Pokud primární DNS překlad nezná (ale funguje), sekundární DNS se ke slovu nedostane.

Přihlašování k doméně probíhá tak, že uživatel zadá své jméno a heslo a vybere doménu, ke které se chce přihlašovat. V síti musí být DNS

server, který přeloží název domény (řadiče) na IP adresu, tu vrátí stanici, na které se přihlašuje uživatel a pak stanice na tuto IP adresu pošle přihlašovací údaje.

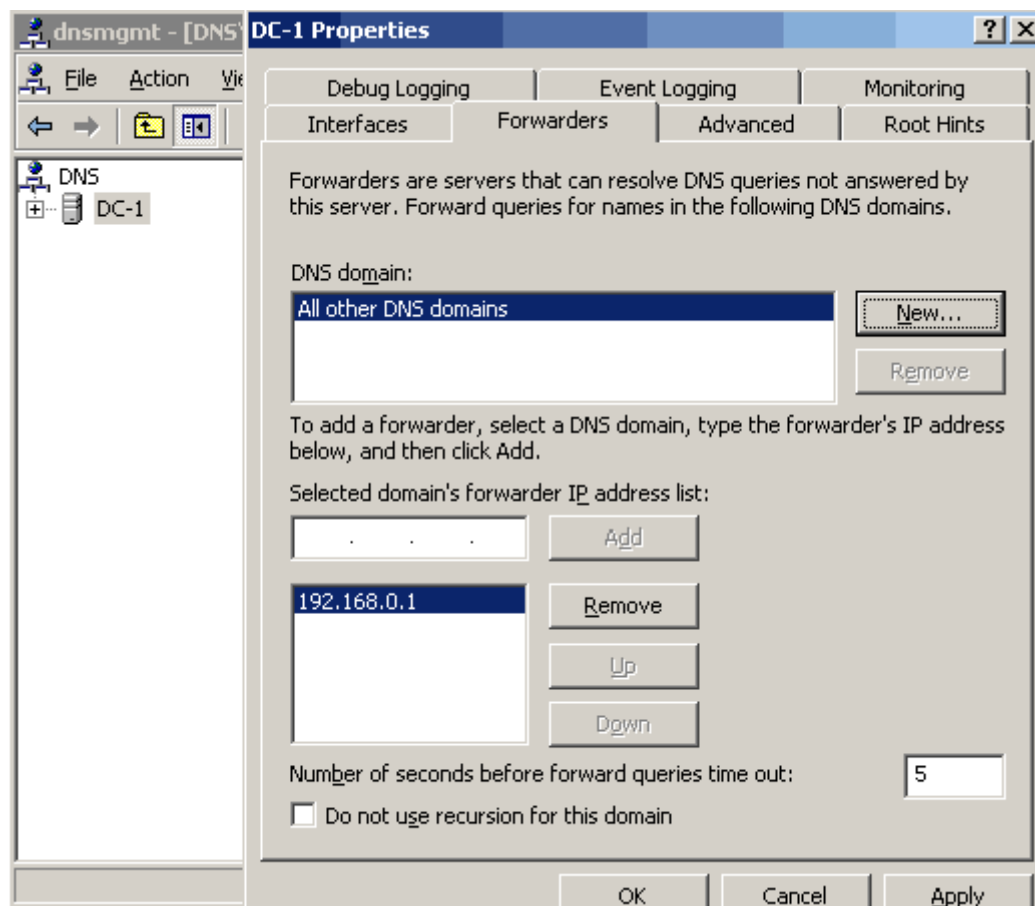
Pokud by byl jako primární DNS server uveden nějaký veřejný DNS server na internetu, který o naší lokální doméně neví, nebude nám ji moci přeložit na správnou IP adresu.

Proto je vhodné v síťovém nastavení počítače uvádět jen lokální DNS server jako primární a sekundární vůbec nezadávat.

Na lokálním DNS serveru nastavit DNS forwarder, který bude zajišťovat, že pokud nebude znát nějaký požadovaný překlad, předá jej dál jinému DNS serveru.

V případě potřeby doinstalujte potřebnou roli serveru.

#### Control Panel – Administrative Tools – DNS – Properties – Forwarders.



Pokud na počítači v doméně nezadáme ani jeden DNS server, bude se počítač při přihlašování do domény ptát broadcastem, zda v síti existuje řadič daného jména. Pokud daný řadič bude běžet, tak počítači odpoví.



Nevýhoda tohoto nastavení je, že přihlašování trvá dlouho.

V mezipaměti počítače se pamatují negativní odpovědi DNS serveru 5 minut. Chceme-li mezipaměť vyčistit okamžitě, můžeme použít příkaz **ipconfig /flushdns**.

#### Poznámka

- ❑ Co dát do nastavení DNS přímo na DNS serveru? Jeho vlastní adresu.
- ❑ Na řadiči domény nastavit do DNS také adresu DNS serveru.
- ❑ Je-li řadič současně i DNS serverem, nastavit jako DNS server opět sám sebe.

#### Poznámky

Stanice XP si pamatuje posledních 10 přihlášení (pokud to není stanoveno politikami jinak). Proto pokud jsme se v minulosti někdy úspěšně přihlásili do domény, tak i když je DNS špatně nakonfigurováno, je možné se přihlásit.

Každý doménový uživatel může vložit do domény až 10 počítačů (pokud je na daném počítači administrátorem a nástroj pro přidání do domény má k dispozici).

Vyjmout z domény může pouze doménový administrátor.

Pokud je v doméně více řadičů, replikují si účty mezi sebou.

## 24.5 Cvičení

- ❑ Víte, co je CAL?
- ❑ Víte, jak zjistit licenční režim na serveru?
- ❑ Umíte změnit licenční režim a zvýšit počet licencí pro připojení na server?
- ❑ Znáte rozdíl mezi doménou a pracovní skupinou?
- ❑ Víte, kde jsou ukládána hesla lokálně?
- ❑ Pokud vytvoříme v doméně účet student a na počítači v SAM databázi účet student, jsou to stejné účty?
- ❑ Zjistěte product key.
- ❑ Jaká hesla by se měla volit?
- ❑ Co je DHCP?
- ❑ Jak zjistit IP adresu, kterou počítači přidělil DHCP server?
- ❑ Pokud má počítač nastaveno přidělování IP adresy DHCP serverem, jakou adresu počítač získá, pokud DHCP server není dostupný?
- ❑ Jak vyčistit přidělenou IP adresu?

- ❑ Co je gateway?
- ❑ Potřebují počítače s adresami 192.168.1.1 a 192.168.1.2 s maskou 255.255.255.0 pro svou komunikaci bránu?
- ❑ Potřebují počítače s adresami 192.168.2.1 a 192.168.1.2 s maskou 255.255.255.0 pro svou komunikaci bránu?
- ❑ Potřebují počítače s adresami 192.168.1.7 a 192.168.1.3 s maskou 255.255.255.252 pro svou komunikaci bránu? Jsou ve stejné podsíti? V jaké?
- ❑ Jaké je správné nastavení primárního a sekundárního DNS serveru?

 *Domácí úkol*

Zopakujte si pojmy jako DNS, DHCP, IP, maska podsítě, síťová adresa, broadcast.

Zopakujte si výpočty (IP adresa, maska podsítě, adresa podsítě, broadcast v dané podsíti).

 *Shrnutí*

- ✓ Víte, co je CAL. Znáte základní typy licencování připojení počítače k serveru.
- ✓ Umíte zjistit a změnit product key.
- ✓ Víte, jak jsou v doméně a v pracovní skupině spravovány účty.
- ✓ Znáte princip přidělování síťových parametrů pomocí DHCP.
- ✓ Víte, co je APIPA.
- ✓ Umíte používat příkaz ipconfig.
- ✓ Víte, co je DNS a víte, jak nastavit primární a případně sekundární DNS server.
- ✓ Víte, k čemu slouží brána.

## 25. Úvod do správy (49. – 50. hodina)

### 25.1 Pojmy

#### 25.1.1 Servery

##### Typy serverů

- ❑ **řadič domény**
- ❑ **členský server** – member server – je v doméně, ale není řadičem domény, může mít funkci souborového serveru (file server) nebo tiskového serveru (print server)
- ❑ **samostatný server** – standalone server – není součástí domény, je v pracovní skupině (workgroup), může sloužit například jako firewall nebo webserver. Volí se jako samostatný z důvodu možného ohrožení, aby v případě napadení nebyla hned ohrožena celá doména.

##### Rozdělení podle rolí, které plní

- ❑ DNS server
- ❑ DHCP server
- ❑ WINS server

Jakou roli plní – zjistíme pomocí **Start – Manage your server**.

#### 25.1.2 Adresářová služba – Directory Service

**Adresářová služba** je hierarchická struktura objektů (uživatelé, počítače, tiskárny, sdílené složky).

**Active Directory** je adresářová služba ve Windows Serveru 2003.

Má další rozšíření oproti adresářové službě obecně:

- ❑ integrace DNS
- ❑ rozšiřitelnost – může růst stejně jako organizace, ve které se používá
- ❑ centralizovaná správa
- ❑ možnost delegovat práva jiným subjektům

#### 25.1.3 Doména

**Doména** – je logické uspořádání objektů v síti.

V doméně musí být nejméně jeden řadič.

Doména musí mít jméno – 2 názvy – DNS název a NetBios název.

*DNS název* vypadá například skola.local.

*NetBios název* je část před tečkou – skola. Tento typ názvu používaly spíše starší operační systémy a některé aplikace.

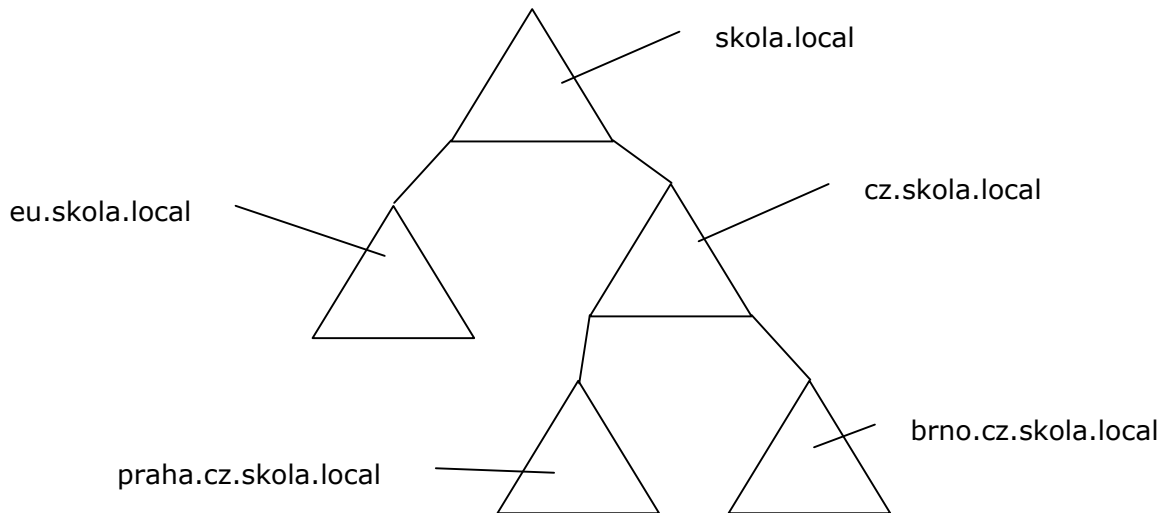
V diagramech se značí jako trojúhelník.



*Doporučené názvy: něco.něco, takové, aby neexistovaly v internetu. To proto, aby doména firmy nebyla z venku dostupná.*

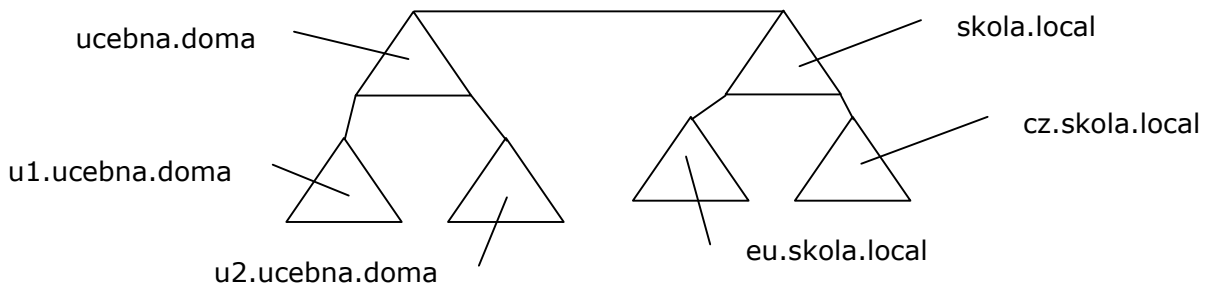
#### 25.1.4 Doménový strom – tree

**Doménový strom** se skládá z více domén, v každé doméně je řadič, mezi doménami existuje hierarchické uspořádání a spojitě DNS pojmenování.



#### 25.1.5 Les – forest

**Les** vzniká spojením více domén nebo doménových stromů, které mají společný globální katalog, konfiguraci a schéma.



#### Poznámka

V krajním případě mohou pojmy splynout.  
les = strom = doména


### 25.1.6 Organizační jednotka

**Organizační jednotka** je typ objektu, ve kterém může správce seskupovat zvolené objekty domény.

Organizační jednotka může obsahovat uživatele, skupiny, tiskárny, počítače nebo jiné organizační jednotky.

Organizační jednotky lze ručně vytvářet.

Důvod pro organizování objektů v organizačních jednotkách může být delegování administrátorských oprávnění nebo dědění oprávnění a skupinových politik.

V okně **Active Directory Users and Computers** jsou organizační jednotky značeny jako žlutá složka se symbolem knihy. 

V případě potřeby doinstalujte potřebnou roli serveru.

### 25.1.7 Kontejner

Kontejner vzniká při instalaci – například **Users, Computers**.

V okně **Active Directory Users and Computers** jsou kontejnery značeny jako žlutá složka. 

### 25.1.8 Site

**Site** – tento termín má v češtině spoustu výrazů, například stanoviště, umístění, pracoviště a mnohé jiné.

**Site** je fyzické uspořádání počítačů. Po instalaci je doména v jednom situ. Sity a domény nemají žádnou vzájemnou fyzickou vazbu. Jeden site může obsahovat více domén a naopak doména může být rozdělena do více sitů.

Například firma může mít svůj jeden fyzický segment v Praze a druhý v Brně, v každém městě jeden site.

## 25.2 Přihlašování

- **lokální** – uživatel je ověřen vůči lokální SAM databázi, dostane access token – vstupenku k počítači
- **doménové** – uživatelské údaje se zašifrují a pošlou řadiči, na kterém proběhne ověření. Uživatel dostane access token – vstupenku pro přístup ke zdrojům v doméně (přístupné pro daného uživatele).

Při přihlašování je možno zvolit, zda se budeme logovat k lokálnímu počítači nebo k doméně.

Někteří uživatelé si toto pletou, proto je možno zajistit, aby se vždy logovali k doméně.

Přihlašovací jméno se zadá pomocí **UPN** (User Principal Name):  
**jméno@název-domény**

### 25.3 Secondary logon, run as

Je vhodné používat pro běžnou práci druhý účet, který nemá administrátorská oprávnění.

Například na stanici XP (se SP1 a vyšším) lze doinstalovat administrátorský balíček – AdminPak, který obsahuje nástroje pro administraci.

Po přihlášení na svůj druhý uživatelský účet lze vybraný nástroj spustit jako jiný uživatel, například administrátor.

Pravým tlačítkem klikneme na vybraný nástroj a zvolíme **Run As**. Někdy se stane, že nabídka **Run As** není vidět – většinou pomůže kombinace **SHIFT + pravé tlačítko**.

#### 25.3.1 Kdo je přihlášen?

Zjistit, kdo je zrovna přihlášen, lze více způsoby:

- ❑ po kliknutí na **Start** se objeví jméno v nabídce
- ❑ **CTRL+ALT+DEL** – ve správci úloh
- ❑ řádkový příkaz – **whoami** (pokud není na XP, stačí jej ze serveru nakopírovat)

### 25.4 Nástroje pro správu

Administrativní nástroje umožňují síťovému administrátorovi přidávat, hledat a měnit počítače a síťová nastavení a objekty v Active Directory.

Tyto nástroje lze instalovat na Windows Server2003 a Windows XP Professional.

Najdeme je v nabídce **Start – Administrative Tools**. Jsou to soubory s příponou .msc (dsa.msc, eventvwr.msc, ...).

**Mezi nejznámější nástroje patří:**

- ❑ Active Directory Users and Computers
- ❑ Active Directory Sites and Services
- ❑ Active Directory Domains and Trusts
- ❑ Computer Management
- ❑ DNS
- ❑ Remote desktops

Na XP lze doinstalovat sadu nástrojů pro správu **adminpak.msi**.

Z XP se můžeme vzdáleně připojit na server **\\server\c\$** na jeho disk C (jako administrátor)

**Adminpak.msi** najdeme ve složce **windows\system32**.

Po nainstalování najdeme další nástroje v **Ovládacích panelech – Nástroje pro správu**.

**Computer management** – pravé tlačítko – můžeme se vzdáleně připojit k jinému počítači a spravovat jej (dívat se do logů, spravovat disky apod.). Záleží na nastavení firewallu vzdáleného počítače.

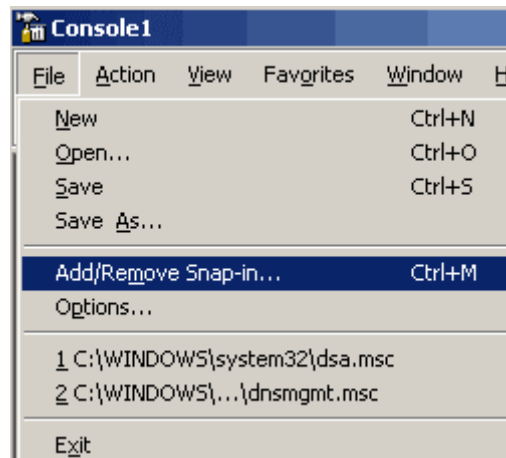
## 25.5 MMC

**MMC – Microsoft Management Console**, spouští se příkazem **mmc.exe**.

Do konzoly se dají přidat často používané nástroje – anglicky **snap-in**, například **Computer management** nejen pro lokální server, ale i pro jiné počítače. Podoba konzoly se dá uložit a její zástupce umístit na plochu.

Přidání snap-in: **File – Add/Remove Snap-in – Add** – vybrat položku, kterou často používáme.

Vyzkoušejte!



## 25.6 Cvičení

- ❑ Popište, co je to řadič domény.
- ❑ K čemu může sloužit member server?
- ❑ K čemu může sloužit standalone server?
- ❑ Jaké role může server plnit?
- ❑ Co je to Active Directory? Jakými disponuje rozšířeními oproti obecné adresářové službě?



- ❑ Co je to doména?
- ❑ Co je to doménový strom?
- ❑ Co je to forest?
- ❑ Co je to site?
- ❑ Co je to OU a kontejner?
- ❑ Jaké znáte druhy přihlašování?
- ❑ Jak zajistit přihlášení do domény, aniž by se vybírala myší?
- ❑ Proč je vhodné vytvářet druhý účet s uživatelskými oprávněními pro administrátora?
- ❑ Jak se dá spustit zvolený program pod jiným uživatelem?
- ❑ Jaké způsoby zjištění, kdo je zrovna přihlášen, znáte?
- ❑ Jak lze z XP používat administrátorské nástroje pro správu, které nejsou na XP standardně nainstalovány.
- ❑ Jak se do MMC přidá zvolený snap-in? Jak se podoba konzoly uloží a zástupce pošle na plochu?

 *Domácí úkol*

Zopakujte si pojmy jako server, řadič, typy serverů, funkce serverů, doména, strom, les, site, OU, kontejner, Active Directory.

Vyzkoušejte spouštění programů pod jiným uživatelem – *run as*.  
Vyzkoušejte instalaci administrátorských nástrojů pro správu.

Vyzkoušejte nastavení MMC.

 *Shrnutí*

- ✓ Víte, co je to řadič domény, member server, standalone server, víte, jaké funkce může server plnit.
- ✓ Umíte vysvětlit pojmy doména, doménový strom, les, organizační jednotka, kontejner, site.
- ✓ Víte, co je Active Directory.
- ✓ Umíte používat Run As.
- ✓ Umíte zjistit jméno přihlášeného uživatele.
- ✓ Umíte používat administrátorské nástroje pro správu i na běžné stanici XP.
- ✓ Umíte spustit MMC a provést úpravy.

## 26. Pojmenování, uživatelské účty (51. – 52. hodina)

### 26.1 Pojmenování

Každý objekt v Active Directory lze pojmenovat a jménem vyjádřit jeho umístění v hierarchii adresářové struktury.

Active Directory pracuje se třemi typy pojmenování objektů:

- LDAP relativní jméno
- LDAP plné jméno
- kanonické jméno

#### 26.1.1 Příklad

**LDAP relativní jméno:**

OU=moje\_jednotka

**LDAP plné jméno:**

OU=moje\_jednotka, DC=ssps, DC=cz

**Kanonické jméno:**

ssps.cz/moje\_jednotka

**LDAP** – Lightweight Directory Access Protocol – protokol, který se používá pro adresářové služby.

OU = organisation unit

DC = domain component

CN = common name

OU=vnořenáOU, OU=hlavníOU, DC=ssps, DC=local

Pokud by byla někde v názvu mezera, uzavře se celá definice do uvozovek.

„OU=vnořená OU, OU=hlavní OU, DC=ssps, DC=local“

#### 26.1.2 Jak se popíše kontejner „Computers“ LDAPem?

CN=Computers, DC=ssps, DC=local

#### 26.1.3 Jak se popíše počítač PC1 v kontejneru „Computers“ LDAPem?

CN=PC1, CN=Computers, DC=ssps, DC=local

## 26.2 Cvičení

V doméně vytvořte organizační jednotku „test“ a v ní vnořenou organizační jednotku „vnorenaOU“.



### Poznámka

Organizační jednotky lze myší přesouvat.

## 26.3 Řádkový příkaz pro vytváření objektu

Příkazem **dsadd** lze vytvářet uživatele, počítač, organizační jednotku v doméně v Active Directory.

### 26.3.1 Příklad

**dsadd computer „cn=PC1,dc=ssps,dc=local“**  
vytvoří účet počítače v doméně ssps.local

### 26.3.2 Příklad

**dsadd ou „ou=test,dc=ssps,dc=local“**  
vytvoří organizační jednotku „test“ v doméně ssps.local

## 26.4 Uživatelské účty

V doméně existují některé přednastavené účty, které nelze smazat – **administrator, guest**.

### 26.4.1 Vytvoření nového uživatelského účtu

V nástroji **Active Directory Users and Computers** klikneme myší na organizační jednotku, ve které chceme uživatele založit. Pravým tlačítkem myši zvolíme **New – User**.

Uživatelský účet je možno založit řádkovým příkazem **dsadd**.

### 26.4.2 Příklad

**dsadd user „cn=ivona,ou=testOU,dc=ssps,dc=local“**

Tímto příkazem se vytvoří účet ivona v organizační jednotce **testOU**, ale účet bude zakázán, protože byl vytvořen bez hesla. Pokud na konec příkazu přidáme „ **-pwd heslo**“, bude vytvořen účet i s heslem.

Heslo musí splňovat podmínku 3 ze 4,  $\geq 7$  (aspoň 7 znaků z minimálně tří sad – velká, malá písmena, číslice, speciální znaky).

## 26.5 Cvičení

V doméně vytvořte organizační jednotku „zaci“, v této OU vytvořte OU „zaci 1.A“ – „zaci 4.S“.

V OU se jménem vaší třídy vytvořte uživatelské účty třídy.

## 26.6 Hromadné vytváření uživatelských účtů

Jedna z možností, jak na nově vznikající uživatelské účty aplikovat nastavení některého existujícího uživatele (např. členství ve skupinách, cesta k domovskému adresáři, cesta k osobnímu profilu apod.), je kopírování.

Na vzorovém uživatelském účtu vyvoláme pravým tlačítkem doplňkovou nabídku a zvolíme **Copy** (kopírovat) a dále postupujeme vytvořením dalšího uživatele.

## 26.7 Vlastnosti uživatelských účtů

Ve vlastnostech účtu na záložce **Account** lze nastavit možnost logovat se jen v určitém čase – **Logon Hours** a na vybrané počítači – **Log On To**.

Přednastaveno je logování na všechny počítače s výjimkou řadiče bez omezení času.

### **Poznámka**

U XP se někdy stává, že i po zakázání uživatelského účtu se uživatel ještě jednou přihlásí.

## 26.8 Cvičení

Na zvoleném účtu nastavte možnost logovat jen v čase od 8 - 17 v pracovních dnech.

## 26.9 Uživatelské profily

Uživatelské profily jsou dvojího druhu:

- **lokální** – jsou uloženy na disku stanice, ke které se uživatel přihlásil. Na každém počítači jsou jiné.
- **cestovní** (roaming profile) – je vhodný v případě, že uživatel často mění stanoviště. Profil je uložen na serveru ve sdílené složce.

### 26.9.1 Vytvoření cestovního profilu

Na serveru vytvoříme sdílenou složku nazvanou například **Profily**. Nasdílíme ji s oprávněními – **Full Control – Everyone**.

Do profilu uživatele (ve vlastnostech uživatelského účtu) na kartě **Profile** napíšeme cestu k jeho cestovnímu profilu:

**\\jméno\_serveru\Profily\%USERNAME%**

### Poznámka

Aby se administrátor dostal do cestovního profilu bez problémů (problémy se zabezpečením omezeným na vlastníka), je toto nutno předem nastavit například v politikách (bude ukázáno později).

Při prvním přihlášení uživatele s nově vytvořeným cestovním profilem na stanici se zjistí, že uživatel má mít cestovní profil, ale zatím žádný na serveru neexistuje. Proto se na serveru vytvoří složka profilu pro daného uživatele – prázdná.

Lokálně na stanici se vytvoří lokální kopie cestovního profilu. Při odhlášení uživatele od stanice se změny přesunou na server do složky cestovního profilu.

Při dalším přihlášení na stanici se ze serveru přenesou nové změny v profilu na stanici do lokální kopie cestovního profilu.

### Upozornění

Je třeba dát pozor na souběžné přihlášení jednoho uživatele na více stanic. Při odhlášení z první stanice se přenesou změny na server. Při odhlášení z druhé stanice se přenesou změny z této stanice na server a může dojít k přemazání souborů z první stanice.

## 26.10 Domovský adresář

Podobně jako profily můžeme centralizovat i domovské adresáře uživatelů. Může to být vhodné zejména z hlediska zálohování a kontroly.

Na serveru vytvoříme složku nazvanou například **Home**, kterou nasdílíme s plnými oprávněními pro všechny (full control – everyone).

Ve vlastnostech účtu na kartě **Profile** doplníme do kolonky domovský adresář

**\\jméno\_serveru\Home\%USERNAME%**

## 26.11 Cvičení

Nastavte vybranému uživateli cestovní profil a domovský adresář na server a vyzkoušejte.

 *Domácí úkol*

- Zopakujte si typy pojmenování v Active Directory.  
Vyzkoušejte vytváření OU a uživatelských účtů (pomocí dialogového okna i pomocí řádkového příkazu).  
Vyzkoušejte nastavení cestovního profilu na serveru.  
Přesměrujte domovský adresář uživatele na server.  
Vyzkoušejte kopírování uživatelských účtů.

 *Shrnutí*

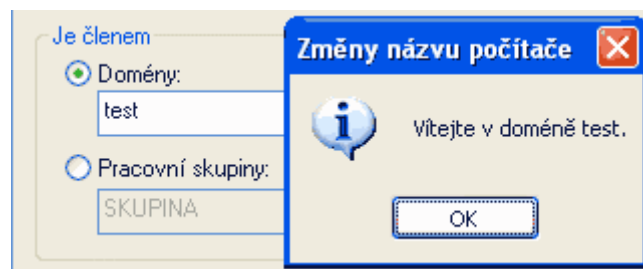
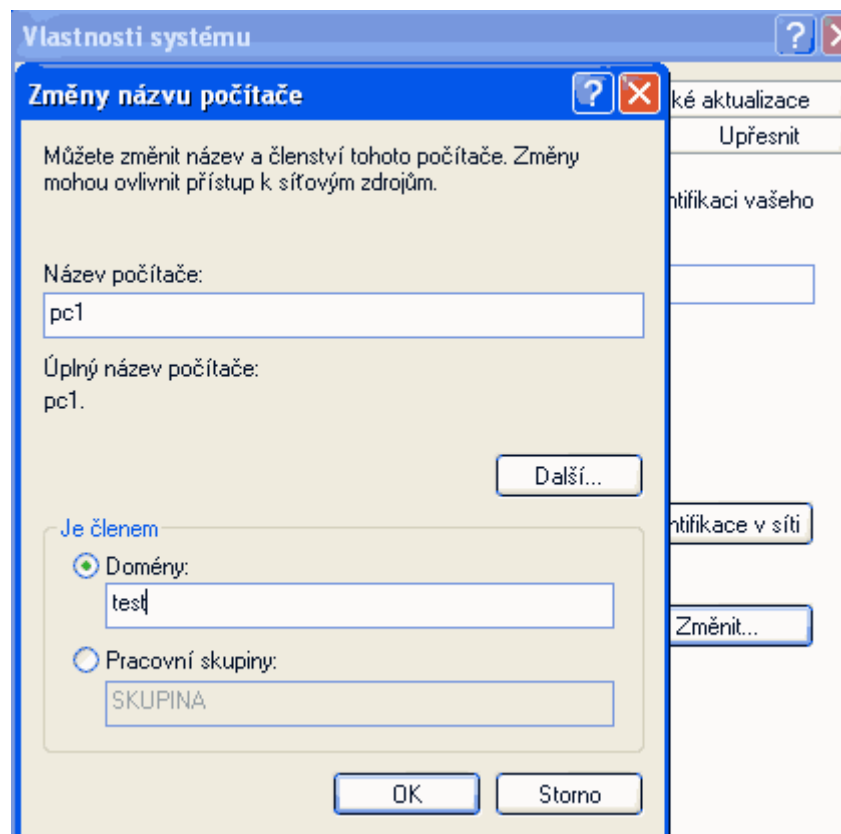
- ✓ Víte, s jakými typy pojmenování Active Directory pracuje.
- ✓ Znáte zkratky, které se při LDAP pojmenování používají.
- ✓ Umíte vytvořit OU pomocí dialogového okna i pomocí řádkového příkazu.
- ✓ Umíte vytvořit uživatelský účet pomocí dialogového okna i pomocí řádkového příkazu.
- ✓ Umíte ve vlastnostech uživatelského účtu nastavit parametry určující například možnou dobu přihlášení ke zvoleným počítačům, členství ve skupinách nebo nastavení cestovního profilu a přesměrování domovského adresáře.
- ✓ Víte, jak použít kopírování účtu.
- ✓ Víte, jaký je rozdíl mezi lokálním a cestovním profilem.

## 27. Účty počítačů, změny (53. – 54. hodina)

Každý počítač s operačním systémem MS Window NT, Windows 2000, Windows XP Professional nebo Windows Server 2003 má po připojení do domény v Active Directory svůj účet.

Účet v Active Directory můžeme vytvořit ručně stejně jako účet uživatele, ale to neznamená, že po tomto procesu bude počítač v doméně. Tímto způsobem můžeme přednastavit některé parametry účtu počítače, například uživatele, který může počítač vložit do domény nebo umístění ve zvolené organizační jednotce.

Musíme se k počítači přihlásit jako lokální administrátor, kliknout na **Tento počítač** pravým tlačítkem, zvolit **Vlastnosti** a na kartě **Název počítače** kliknout na tlačítko **Změnit**.





Zvolíme volbu – **je členem domény** a vepíšeme název domény. Po kliknutí na OK bude vyžadováno přihlášení doménového administrátora nebo pověřeného uživatele z Active Directory.

Každý počítač v doméně komunikuje s řadičem domény přes zabezpečený kanál, přes který se řadiči identifikuje a probíhá autentikace.

Každých 30 dní se řadič domény domluví s počítači v doméně na hesle. Pokud je počítač dlouho vypnutý, může dojít k tomu, že se s ním řadič nespojí a počítač nebude informován o novém hesle pro přístup k řadiči. To má za následek vyjmutí počítače z domény.

## 27.1 Cvičení

Založte v kontejneru **Computers** účet počítače. Přihlaste tento počítač do domény.

### Poznámka

Aby se do domény daly připojit počítače s operačními systémy W9x, Me, NT, musí se na ně nainstalovat „Directory Services Client“.

## 27.2 Změna názvu účtu

Pokud některý uživatel potřebuje změnit název svého účtu (může nastat například při změně jména), není vhodné účet smazat a vytvořit nový, protože nový účet bude mít jiný identifikátor, jiný domovský adresář, profil, členství ve skupinách atd.

Proto lze účet přejmenovat.

V dialogovém okně **Active Directory Users and Computers** klikneme pravým tlačítkem na název účtu, který chceme změnit a zvolíme **Rename** (prejmenovat). Napíšeme nové uživatelské jméno a potvrdíme.

Přejmenovaný účet bude mít stejná oprávnění, bezpečnostní nastavení, vlastnosti a práva, jaká byla spojena s původním jménem účtu.

## 27.3 Zakázání – povolení účtu

V dialogovém okně **Active Directory Users and Computers** můžeme pomocí pravého tlačítka myši vyvolat doplňkovou nabídku k vybranému účtu a v této nabídce zvolit **Enable Account / Disable Account** (Povolit účet / Zakázat účet).

Po zakázání účtu bude blokováno přihlášení k tomuto účtu.

### Poznámka

U XP se někdy stává, že i po zakázání uživatelského účtu se uživatel ještě jednou přihlásí.

Důvod pro zakázání účtu může být například dlouhá absence uživatele nebo bezpečnostní důvody. Nebo například vytváříme účty, které budeme chtít používat v budoucnu a zatím je necháme zakázané.

## 27.4 Cvičení

Změňte název účtu.  
Zakažte účet a vyzkoušejte přihlášení.  
Povolte účet a vyzkoušejte přihlášení.

## 27.5 Změna a resetování hesla

Pokud není uživateli zakázáno měnit si své heslo, může si je změnit sám (např. po stisknutí **CTRL+ALT+DEL**).

Pokud uživatel zapomene své heslo, může mu jej administrátor resetovat. K resetování není třeba znát staré heslo, zadá se pouze dvakrát nové heslo.

K resetování lokálních hesel jsou autorizováni lokální administrátoři.

K resetování hesel doménových účtů jsou autorizováni doménoví administrátoři (domain administrators), podnikoví administrátoři (enterprise administrators), účtový specialista (account operator) a jiní uživatelé a skupiny, kterým bylo právo resetovat hesla delegováno.

### **Pozor**

Poté, co je resetováno heslo lokálního uživatele na počítači, mohou být některé šifrované informace nedostupné. Je to proto, že šifrování je založeno také na hesle uživatele.

Příkladem mohou být šifrované e-maily pomocí veřejného klíče, internetová hesla uložená na počítači, soubory, které si uživatel zašifroval.

## 27.6 Resetování účtu počítače

Počítače se autentizují k řadiči domény. Každých 30 dní se s řadičem domlouvají na hesle (které nemůžeme ovlivnit). Když nastane situace, kdy stanice nemá aktuální heslo (například byla dlouho vypnutá nebo bylo nutno ji obnovit ze zálohy, která obsahuje staré heslo), je problém přihlásit stanici do domény, protože jí řadič nedůvěřuje.

### **Co udělat, když se nedaří přihlásit do domény?**

- Jako lokální administrátor se podívat na nastavení TCP/IP na stanici, zejména zkontrolovat nastavení DNS.

Vyzkoušet **ping** ze stanice na řadič. Zkusit přeložit název domény (příkaz „**nslookup název\_domény**“ by měl vrátit IP adresu řadiče).

- Pokud je nastavení TCP/IP v pořádku, zkusit **reset účtu** počítače (**Active Directory Users and Computers** – pravé tlačítko na zvoleném počítači – **Reset Account**). Toto bohužel ne vždy funguje.
- Další možnost je **vyjmout počítač z domény**, z **Active Directory Users and Computers** smazat účet počítače, pokud tam zůstal, a znovu počítač do domény **vložit**.
- Pokud proces stále není úspěšný, je potřeba hlubší analýzy (kontrola **DNS** serveru, kontrola, jak probíhá **replikace** mezi řadiči, kontrola **nastavení práv** v Active Directory, kontrola nastavení **politik**).

## 27.7 Vyhledávání v Active Directory

Někdy je v **Active Directory** nutno vyhledávat, protože může obsahovat tisíce objektů. Každý objekt v **Active Directory** má nastavené své vlastnosti (například uživatelský účet může mít zadány hodnoty jako jméno, příjmení, jméno firmy atd., účet počítače může mít nastaveny vlastnosti jako umístění počítače, operační systém, jméno apod.).

Vyhledávat můžeme dotazem na jednotlivé vlastnosti objektů nebo na různé kombinace vlastností.

V **Active Directory Users and Computers** zvolíme v nabídce **Action** volbu **Find**. Otevře se okno **Find Users, Contacts, and Groups**. V nabídce **Find** vybereme typ objektu, který chceme vyhledávat (například Users, Contacts, and Groups nebo Computers, Printers apod.).

Dále vybereme **umístění**, kde budeme chtít objekt vyhledat (například celou Active Directory, řadiče, konkrétní kontejner).

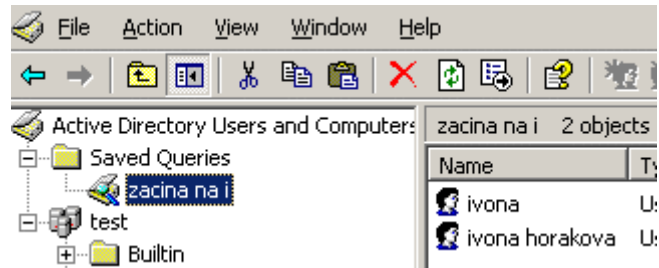
Do políčka **Name** napíšeme vyhledávané jméno objektu.

Pro složitější vyhledávání můžeme použít záložku **Advanced**. Vybereme **sloupec**, který budeme prohledávat, zvolíme **operátor** (začíná, končí, je přesně rovno, obsahuje ...) a zadáme **hodnotu**, kterou budeme ve zvoleném sloupci vyhledávat.

Nabízené sloupce se liší podle typu objektu, který vyhledáváme.

### Poznámka

V **Active Directory Users and Computers** lze často opakované vyhledávací dotazy uložit. Jejich seznam najdeme v **Saved Queries**. Vyzkoušejte.



## 27.8 Cvičení

Vyhledejte všechny počítače v celé **Active Directory**, které začínají na **PC**.

Vyhledejte uživatelské účty, které končí na **a**.

### Domácí úkol

Zkuste vytvořit účet počítače v **Active Directory**. Vyzkoušejte vložení počítače do domény. Vyzkoušejte odstranění počítače z domény. Projděte si vlastnosti účtu počítače.

Vytvořte účet uživatele. Nastavte mu cestovní profil do sdílené složky na serveru. Přejmenujte účet uživatele. Zkontrolujte, kam směřuje jeho profil.

Vyzkoušejte zakázání účtu. Vyzkoušejte přihlášení po zakázání účtu. Povolte účet.

Vyzkoušejte změnu hesla. Vyzkoušejte resetování hesla.

Zopakujte si vyhledávání objektů v Active Directory.

### Shrnutí

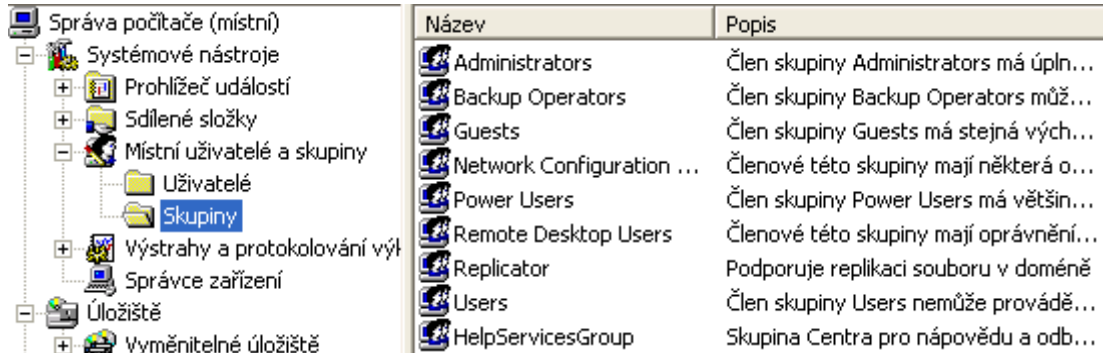
- ✓ Umíte přidat účet počítače do Active Directory.
- ✓ Umíte zakázat a povolit účty.
- ✓ Umíte přejmenovat účet.
- ✓ Umíte odstranit počítač z domény.
- ✓ Umíte změnit a resetovat heslo účtu.
- ✓ Umíte v Active Directory vyhledávat.

## 28. Skupiny (55. – 56. hodina)

Skupina je soubor uživatelských účtů. Skupiny se používají k efektivnímu nastavování oprávnění přístupu ke zdrojům.

Skupiny lze vnořovat do jiných skupin.

Zobrazit skupiny můžeme na serveru v Active Directory Users and Computers, na XP ve **Správě počítače** – pravým tlačítkem myši klikneme na **Tento počítač** – zvolíme **Spravovat**.



| Název                     | Popis                                  |
|---------------------------|----------------------------------------|
| Administrators            | Člen skupiny Administrators má úpln... |
| Backup Operators          | Člen skupiny Backup Operators můž...   |
| Guests                    | Člen skupiny Guests má stejná vých...  |
| Network Configuration ... | Členové této skupiny mají některá o... |
| Power Users               | Člen skupiny Power Users má většín...  |
| Remote Desktop Users      | Členové této skupiny mají oprávnění... |
| Replicator                | Podporuje replikaci souboru v doméně   |
| Users                     | Člen skupiny Users nemůže provádě...   |
| HelpServicesGroup         | Skupina Centra pro nápovědu a odb...   |

### 28.1 Rozdělení skupin

- **lokální** – uložena v SAM databázi lokálně na počítači. Některé lokální skupiny:
  - **Administrators** - člen skupiny Administrators má úplný a neomezený přístup k příslušnému počítači.
  - **Backup Operators** - člen skupiny Backup Operators může pro účely zálohování a obnovování dat překonat zabezpečující omezení. Může zálohovat, obnovovat, nemá právo číst, ale pokud si zálohu obnoví na počítači, kde má vyšší práva, získá k záloze plný přístup.
  - **Power Users** - člen skupiny Power Users má většinu oprávnění pro správu počítače s některými omezeními, takže kromě certifikovaných aplikací může spouštět i starší aplikace. Oproti běžnému uživateli může například nasdílet složku, instalovat.
  - **Network Configuration Operators** - členové této skupiny mají některá oprávnění správce ke správě konfigurace síťových funkcí.
  - **Users** - člen skupiny Users nemůže provádět nechtěné ani úmyslné změny systému, může tedy spouštět certifikované aplikace, nemůže však spouštět většinu starších aplikací.
- **doménová** – uložena v Active Directory
  - **Rozdělení podle typu**
    - **security** – lze jí přidělovat práva (například tisk na tiskárně ...)

- **distribution** – nelze jí přidělovat práva jako security skupině, tento typ se využívá pro nějaké aplikace, například pro e-mail ...
- **Rozdělení podle působnosti**
  - **domain local**
  - **global**
  - **universal**

Charakteristika skupin v Active Directory závisí na doménové funkční vrstvě. Jde v podstatě o to, jaké operační systémy se v dané doméně na řadičích vyskytují.

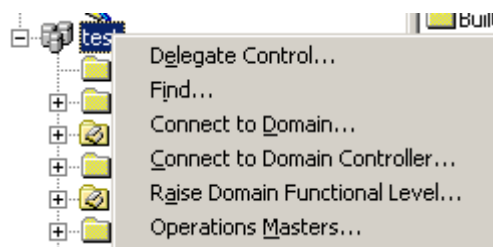
**Existují 4 úrovně doménových funkčních vrstev:**

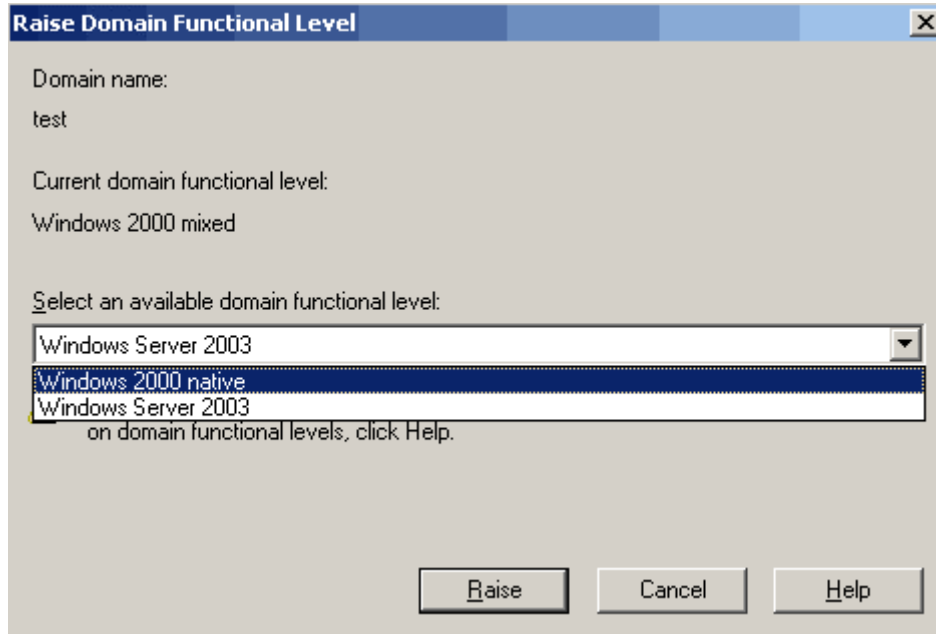
- **Windows 2000 mixed** – je výchozí po instalaci, mohou v ní být řadiče Windows NT Server 4.0, Windows 2000, Windows Server 2003. Je nejnižší úrovní. Lze ji povýšit na:
- **Windows 2000 native** – obsahuje řadiče Windows 2000 a Windows Server 2003. Lze ji povýšit na:
- **Windows Server 2003** – obsahuje jen řadiče Windows Server 2003.
- **Windows Server 2003 Interim** – speciální případ, kdy se v doméně vyskytují řadiče Windows NT Server 4.0 a Windows Server 2003.

**Souvislost se skupinami:**

- **Windows 2000 mixed** – podporuje skupiny typu global a domain local, nepodporuje univerzální skupiny
- **Windows 2000 native** – podporuje skupiny typu global, domain local a universal
- **Windows Server 2003** – podporuje skupiny typu global, domain local a universal
- **Windows Server 2003 Interim** – podporuje skupiny typu global a domain local, nepodporuje univerzální skupiny

Zjištění, v jaké úrovni domény jsme, provedeme kliknutím pravého tlačítka myši na název domény. Zde můžeme najít možnost zvýšit úroveň – **Raise Domain Functional Level**. Snižovat úroveň domény můžeme pouze ze zálohy.

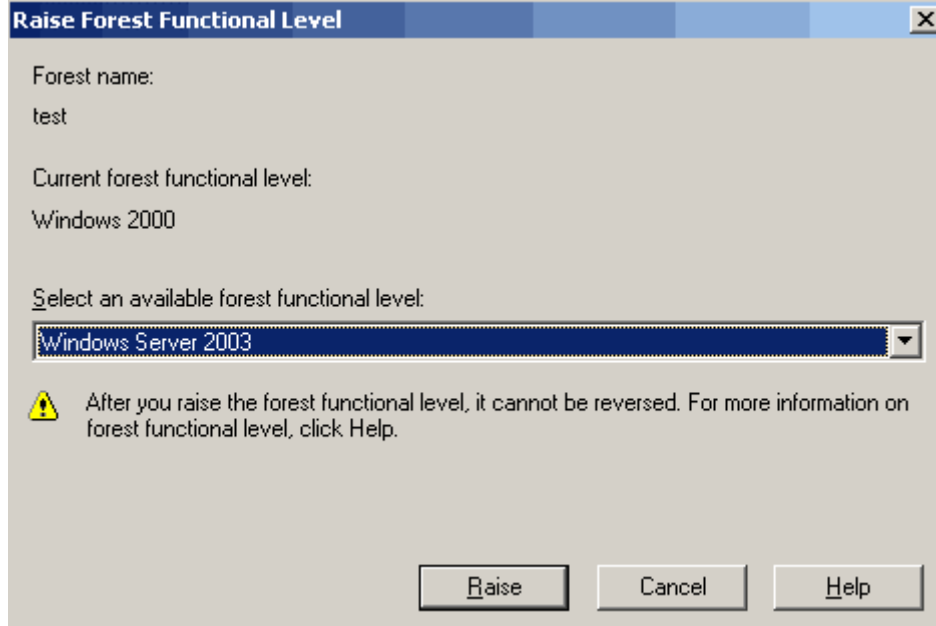




### Poznámka

Celý forest lze povýšit na Server 2003, pokud jsou všechny domény uvnitř typu Server 2003. Pak jdou domény také přejmenovávat.

**Administrative Tools – Active Directory Domains and Trusts – Rais Forest Functional Level** – zobrazí úroveň lesa, může zde být možnost zvýšit úroveň, dovoluje-li to struktura lesa.



## 28.2 Globální skupina

Globální skupina je buď typu **security** nebo **distribution** a může obsahovat uživatele, skupiny a počítače, které jsou ze **stejně** domény jako globální skupina.

Globální security skupinu je možno použít k přiřazování uživatelských práv, delegování pravomocí v objektech Active Directory, k přiřazování



oprávnění přístupu ke zdrojům v jakékoli doméně v daném forestu nebo v jiných doménách v jiném forestu, ke kterému existuje vztah důvěry – trust.

Globální skupina je viditelná v rámci své domény a všech domén, se kterými je navázán trust (vztah důvěry), což zahrnuje také všechny domény v rámci lesa (forestu).

### 28.3 Univerzální skupina

Univerzální skupina je buď typu **security** nebo **distribution** a může obsahovat uživatele, skupiny a počítače, které jsou z jakékoli domény v lese (forestu) dané skupiny.

Univerzální skupinu nelze vytvořit v doméně typu Windows 2000 mixed nebo Windows 2003 Interim.

Univerzální skupina je viditelná ze všech domén lesa a všech domén, se kterými je navázán trust.

### 28.4 Doménová lokální skupina

Doménová lokální skupina je skupina typu **security** nebo **distribution**, která může obsahovat jiné doménové lokální skupiny ze své vlastní domény. Dále může obsahovat univerzální skupiny, globální skupiny a účty z jakékoli domény v lese nebo z jakékoli domény, se kterou je navázán vztah důvěry.

Lze ji použít na přiřazení uživatelských práv a oprávnění přístupu ke zdrojům jen v rámci stejné domény, ve které je skupina vytvořena.

Doménová lokální skupina je viditelná jen ve své vlastní doméně.

Doménovou lokální skupinu můžeme použít, chceme-li nastavit oprávnění přístupu ke zdrojům umístěným ve stejné doméně jako skupina. Globální skupiny, které potřebují přístup k těmto zdrojům, můžeme umístit do této doménové lokální skupiny.

#### **Poznámka**

V prostředí pracovní skupiny může lokální skupina (ne doménová) obsahovat jen lokální uživatele z počítače, na kterém je vytvořena. Pokud je počítač členem domény, může lokální skupina obsahovat doménové účty, globální a univerzální skupiny (ne doménové lokální) z domény, do které počítač patří a z domén, se kterými je uzavřen vztah důvěry.

Naopak to nefunguje. Lokální skupina nemůže být členem žádné jiné skupiny.

#### **Poznámka**

Pokud máme prostředí s jednou doménou, je vhodné zvolit skupiny typu **global**, splňují vše, co je třeba.

## 28.5 Vytvoření skupiny

Lokální skupinu (na XP) můžeme vytvořit v dialogovém okně **Správa počítače**.

V **Active Directory Users and Computers** můžeme vytvořit skupinu v jakékoliv doméně, se kterou je navázán vztah důvěry, máme-li k tomu oprávnění.

Pomocí řádkového příkazu:

```
dsadd group „cn=žáci 3A,ou=žáci,dc=ssps,dc=local“ –samid žáci3A –secgrp yes –scope g
```

**samid** – specifikuje unikátní SAM (Security Accounts Manager) jméno  
**secgrp** – následuje-li „yes“, bude skupina typu security, pokud následuje „no“, bude typu distribution

**scope** – působnost – následuje-li „g“, bude skupina působnosti globální, „l“ znamená domain local, „u“ znamená universal

Nápovědu k tomuto příkazu vypíšeme pomocí **dsadd group /?**.

## 28.6 Cvičení

Vytvořte globální skupinu **žáci** v Active Directory Users and Computers.

Pomocí řádkového příkazu vytvořte v doméně organizační jednotku „**test**“ a v ní pomocí řádkového příkazu vytvořte globální skupinu „**skupina 1**“ se samid „skupina1“, typu security.

## 28.7 Změna působnosti skupiny

Při vytváření nové skupiny je přednastaveno, že je typu **security** a působnosti **global** bez ohledu na aktuální doménovou funkční vrstvu. V doménách s doménovou funkční vrstvou Windows 2000 native nebo Windows Server 2003 je možno měnit působnost skupiny od domain **local** – **global** – **universal**.

Některé přesuny nejdou přímo, je nutno například při změně z **global** na **domain local** nejprve udělat změnu z **global** na **universal** a pak z **universal** na **domain local**.

**Global – Universal – Domain Local – Universal – Global.**

**Změna z Global na Universal:** povoleno pouze, pokud skupina není členem jiné globální skupiny.

**Změna z Domain Local na Universal:** povoleno, pokud skupina není členem skupina jiné Domain Local skupiny.

**Změna z Universal na Global:** povoleno pouze, když skupina neobsahuje jinou univerzální skupinu nebo globální skupinu z jiné domény.

**Změna z Universal na Domain Local:** povolena bez omezení.

## 28.8 Přednastavené skupiny v Active Directory

Přednastavené skupiny najdeme v kontejnerech **Builtin** a **Users**.

Je jich mnoho. Uvedme některé z nich:

- ❑ **Domain Users** – každý uživatel se automaticky stává členem této skupiny stejně jako lokální skupiny Users.
- ❑ **Domain Computers** – členem se stává každý počítač přiřazený do domény
- ❑ **Domain Controllers** – členem se stává každý řadič domény
- ❑ **Domain Admins** – správce domény, má právo vytvářet objekty v doméně (organizační jednotky, uživatele, ...), ale nemůže například vytvořit podřízenou doménu v lese.
- ❑ **Enterprise Admin** – má na starosti celý forest (les), může vytvářet další domény.
- ❑ **DNS Admin** – správce DNS serveru
- ❑ **DHCP Admin** – správce DHCP serveru

## 28.9 Členství ve skupinách

Ve vlastnostech uživatelského účtu můžeme zjistit přímé členství ve skupinách. Skupina ale může být členem jiné skupiny.

Pro zjištění, v jakých skupinách se jako uživatel nacházím, mohu použít příkaz **whoami /groups**.

Příkaz **dsget** můžeme použít pro zjištění atributů objektu (uživatel, počítač, OU) v Active Directory.

**dsget user „cn=Ivona,dc=ssps,dc=local“**

**dsget user „cn=Ivona,dc=ssps,dc=local“ -memberof**

Členství v **systemových skupinách** (např. Everyone – jejímž členem je každý, Authenticated Users – když se uživatel úspěšně přihlásí, do odhlášení) nelze ovlivnit, členem se uživatel stane nějakou akcí.

## 28.10 Cvičení

Skupinu **test** vložte do skupiny **žáci**. Do skupiny žáci vložte vybraného žáka. Vypište členství zvoleného žáka ve skupinách.

Prozkoumejte nápovědu k příkazu **dsget user /?**.

## 28.11 Strategie AGDLP

### A G DL P

A – accounts (účty)

G – globální skupina

DL – doménová lokální skupina

P – permissions (oprávnění)

Při vytváření strategie oprávnění přístupu ke zdrojům je vhodné postupovat s rozmyslem, aby nedocházelo k nejasným vnořením členství jedné skupiny v jiných a následného zmatení oprávnění.

Tento model spočívá v tom, že uživatelé jsou umístěni do globálních skupin, globální skupiny jsou umístěny do lokálních doménových skupin a doménovým skupinám jsou přiřazena oprávnění pro přístup ke zdrojům.

| A                                                            | G                                        | DL                                                              | P                                                                          |
|--------------------------------------------------------------|------------------------------------------|-----------------------------------------------------------------|----------------------------------------------------------------------------|
| účty                                                         | globální skupiny                         | skupiny podle síťových zdrojů                                   | oprávnění pro skupiny DL                                                   |
| Novák →<br>Nováková →<br>Zelený →<br>Horáková →<br>Severyn → | Vedení<br>Účtárna<br>Učitelé<br>Studenti | Tisk uč.106<br>Tisk kancelář<br>Dokumenty-čtení<br>Dokumenty-RW | na konkrétních zařízeních, zdrojích, nastavíme oprávnění pro různé skupiny |
| DL                                                           |                                          |                                                                 |                                                                            |

Na vybraném zdroji, například složce **Soubory** nastavíme NTFS oprávnění (zabezpečení) pro skupinu **Dokumenty-čtení** na pouhé **čtení**, skupině **Dokumenty-RW** nastavíme oprávnění **čtení i zápisu**.

Do skupiny **Dokumenty-RW** zařadíme všechny skupiny uživatelů, kterým budeme chtít povolit čtení i zápis. V našem příkladu je členem skupiny Dokument-RW skupina **Vedení**, proto všichni uživatelé skupiny Vedení budou moci **upravovat i číst** soubory ve složce **Soubory**.

Pokud přijde do firmy nový zaměstnanec, stačí jej zařadit do patřičné globální skupiny (nebo skupin) a získá přístup ke všem zdrojům, ke kterým má daná skupina přístup.

 *Domácí úkol*

Zopakujte si, co je to skupina, jak lze skupinu vytvořit, jak skupiny rozdělujeme.

Připomeňte si, jaká oprávnění mají některé základní lokální a doménové skupiny.

Zopakujte si význam působnosti skupin.

Jaké existují doménové funkční vrstvy? Jak zjistíme, v jaké doménové funkční vrstvě se nacházíme?

Jak se změní působnost skupiny?

Jak zjistíme členství ve skupinách?

Co vyjadřuje strategie AGDLP?

 *Shrnutí*

- ✓ Víte, co je skupina v Active Directory.
- ✓ Umíte vytvořit skupinu jak pomocí dialogového okna Active Directory Users and Computers, tak i řádkovým příkazem.
- ✓ Umíte vložit uživatele do skupiny a zjistit členství uživatele ve skupinách.
- ✓ Dokážete vysvětlit, jaká oprávnění mají základní lokální a doménové skupiny.
- ✓ Víte, co znamená u skupiny termín distribution a security.
- ✓ Víte, co u skupiny znamená typ global, domain local, universal.
- ✓ Víte, jaké existují doménové funkční vrstvy a umíte zjistit, v jaké se nachází váš řadič.
- ✓ Umíte popsat strategii přístupu ke zdrojům, kterou byste měli zvážit při spravování domény.

## 29. Přístup ke zdrojům (57. – 58. hodina)

Mechanismus rozhodování, zda systém dovolí uživateli přístup ke zdroji, sestává ze dvou fází:

- ❑ **autentikace** – ověření uživatele
- ❑ **autorizace** – rozhodnutí, jak bude požadavek vyřízen

Oprávnění a právo není totéž.

### 29.1 Práva – rights

Jedná se o práva k systému.

Například právo přihlásit se k počítači, právo ke změně systémového času. Práva se nastavují v politikách.

### 29.2 Oprávnění - permission

Oprávnění – permission – nastavuje se na zdrojích (složky, tiskárny, soubory).

Definují typ přístupu, který bude přidělen uživateli, skupině nebo počítači k danému objektu.

Například na souboru můžeme nastavit, kdo bude moci soubor číst, měnit, kdo bude mít plnou kontrolu apod.

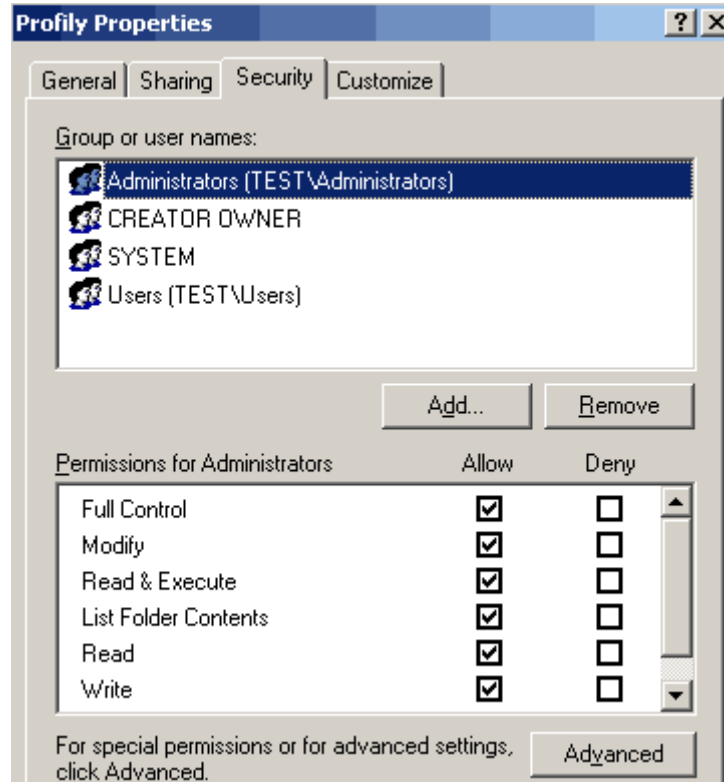
Oprávnění se skládá ze dvou částí:

- ❑ **oprávnění sdíleného adresáře** – uplatňuje se při vzdáleném přístupu přes síť.
- ❑ **NTFS oprávnění (zabezpečení)** – uplatňuje se vždy, ať už při přístupu přes síť nebo přímém přístupu k lokálnímu zdroji.

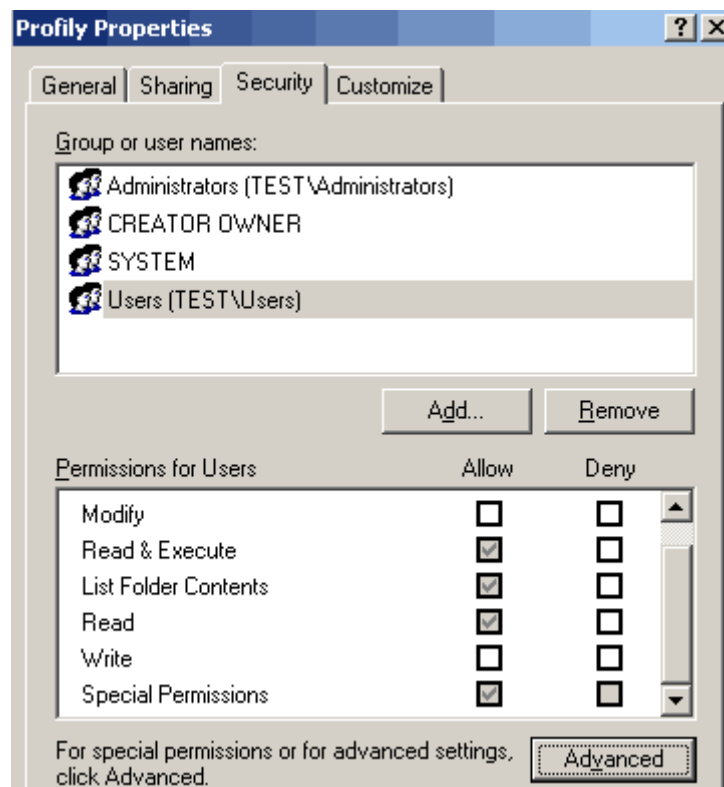
#### 29.2.1 Standardní a speciální NTFS oprávnění

Ve vlastnostech objektu na kartě **Security** najdeme seznam uživatelů a skupin, kterým je nastavováno oprávnění.

Po kliknutí na některého uživatele v seznamu uvidíme ve spodní části, která oprávnění jsou povolena a která zakázána.



Po kliknutí na **Advanced** můžeme blíže specifikovat oprávnění. V tomto případě bude u uživatele, který má změněné speciální oprávnění, zaškrtnuté políčko **Special permission**.



### 29.3 Cvičení

Vytvořte pokusnou složku **test**. Zjistěte, kteří uživatelé nebo skupiny jsou mezi autentikovanými uživateli a jak jsou nastavena defaultní oprávnění.

### 29.4 SID

**SID – Security Identifier** – unikátní číslo identifikující účet. Systém bere uživatele podle SIDu, ne podle jeho jména.

Pokud se nastaví na nějakém objektu zabezpečení – NTFS permission – pro určitého uživatele, a pak dojde ke smazání účtu tohoto uživatele a následně vytvoříme účet stejného jména, tak to bude jiný účet s jiným SID, i když jméno bude stejné. Původní oprávnění smazaného uživatele se nového uživatele nebudou týkat, protože má jiný SID.

### 29.5 DACL

**DACL – Discretionary Access Control List** – seznam účtů, kterým je povolen nějaký druh přístupu k danému zdroji.

NTFS vede pro každý soubor a složku na NTFS logické jednotce seznam všech uživatelských účtů, skupin a počítačů, kterým je povolen přístup k daným souborům a složkám.

Struktura DACL:

| SID                 | speciální oprávnění | zda bylo oprávnění přiděleno přímo nebo bylo zděděno | Příznak, zda je dané oprávnění povoleno nebo blokováno |
|---------------------|---------------------|------------------------------------------------------|--------------------------------------------------------|
| S-123<br>(horakova) | R                   | přímo přiděleno                                      | Allow                                                  |
| S-234<br>(student)  | FC                  | zděděno                                              | Deny                                                   |
| S-788<br>(Everyone) | R,W                 | zděděno                                              | Allow                                                  |

Jak probíhá rozhodnutí, zda bude daný uživatel mít přístup ke zdroji? Systém vybere řádky z DACL, ve kterých najde SID uživatele a řádky skupin, ve kterých je uživatel členem. Provede výpočet a podle výsledku rozhodne.

Například uživatel má přímý zákaz a zděděné povolení – výsledek: zákaz.

Všechny řádky mají stejnou váhu – ať pochází od skupin nebo patří uživateli.



### 29.5.1 Příklad

Ve vlastnostech souboru na kartě **Security** najdeme pro zvoleného uživatele u oprávnění **Read** prázdná obě políčka – Allow i Deny.

**Read**       Allow       Deny

Výsledek: oprávnění Read nebude povoleno.

Stejný výsledek platí pro všechny uživatele, kteří tam vůbec nejsou uvedeni.

### 29.5.2 Příklad

**student**      **Read**       Allow       Deny

**skupina (student je v ní členem)**

**Read**       Allow       Deny

Výsledek: všichni budou mít právo číst – skupina i student.

### 29.5.3 Příklad

**student**      **Read**       Allow       Deny

**skupina (student je v ní členem)**

**Read**       Allow       Deny

Výsledek: Skupina bude mít právo číst, student ne.

### 29.5.4 Příklad

**student**      **Read**       Allow       Deny

**skupina (student je v ní členem)**

**Read**       Allow       Deny

Výsledek: všichni budou mít zakázáno číst – skupina i student.

## 29.6 Dědění oprávnění

To, že je oprávnění **zdeděno** poznáme tak, že políčko u oprávnění je zašedlé.

### 29.6.1 Příklad

**student**      **Read**       Allow       Deny

Přímý zákaz je silnější než zděděné povolení, výsledek je zákaz čtení.

### 29.6.2 Příklad

**student**      **Read**       Allow       Deny

Přímé povolení je silnější než zděděný zákaz, výsledek je povolení čtení.

## 29.7 Jak systém postupuje?

- 1) Hledá **přímé Deny** (patřící uživateli nebo skupině, ve které je členem) – pokud najde, nastává okamžitý zákaz.
- 2) Když neobjeví přímé Deny, hledá **přímé Allow** (patřící uživateli nebo skupině, ve které je členem). Najde-li, nastává povolení.
- 3) Když nenajde přímé Deny ani přímé Allow, hledá **zděděný Deny**. Najde-li, dojde k zákazu oprávnění.
- 4) Když nenajde přímé Deny ani přímé Allow ani zděděné Deny, hledá **zděděné Allow**. Najde-li, dojde k povolení.
- 5) Nenajde-li nic, odepře oprávnění.

Ve skutečnosti systém neprochází DACL opakovaně. Předem si jej seřadí (nejprve přímé zákazy, pak přímá povolení, zděděné zákazy a zděděná povolení).

## 29.8 Oprávnění při kopírování a přesunu

Při kopírování souboru z jedné složky do druhé převezme soubor oprávnění z cílové složky.

Vytvoříme-li soubor ve složce, přebírá oprávnění z této složky.

Při přesunu souboru ze složky do složky musíme rozlišit dva stavy: přesouváme mezi logickými jednotkami nebo ne.

Při přesunu souboru mezi různými logickými jednotkami je to stejné jako u kopírování – soubor přebírá oprávnění z cílové složky.

### **Pozor**

Při přesunu v rámci jedné logické jednotky si soubor zachovává své původní nastavení oprávnění, ale jen dočasně – do doby, než se zasáhne do DACL – pak dojde k přepočítání oprávnění (stanou se z toho cílová oprávnění).

Je na to potřeba dát pozor, na první pohled to nemusí vůbec být vidět.

## 29.9 Cvičení

Vytvořte dvě složky, jedné nastavte pro zvoleného uživatele oprávnění **Read** a druhé složce nastavete oprávnění pro tohoto uživatele na **Write**.

V první složce vytvořte soubor a zkontrolujte, jak se nastavilo jeho oprávnění.

**Přesuňte** soubor do druhé složky a zkontrolujte jeho oprávnění. Mělo by být zachováno oprávnění z první složky.

Přidejte na kartě Security dalšího uživatele a nastavte mu libovolné oprávnění (zásah do DACL).

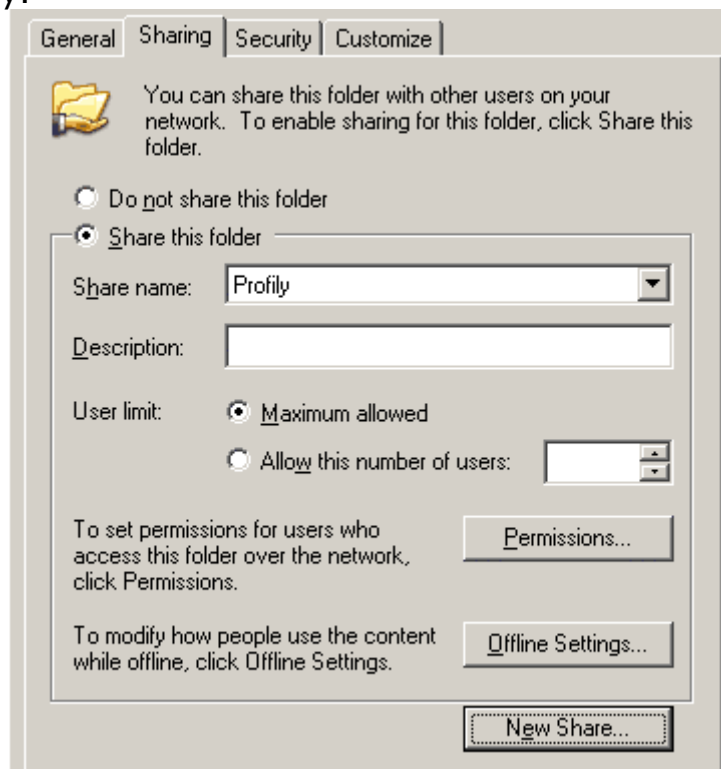
Co se stalo s oprávněním přenášeným z první složky?

### Poznámka

Pokud se na v nastavení oprávnění pro nějaký objekt nacházejí SIDy nějakých uživatelů a nezobrazují-li se jejich jména, znamená to buď smazání uživatele z Active Directory nebo nemožnost spojit se s řadičem domény, který by poskytl překlad SIDu na jméno. To je třeba zvážit před ukvapeným smazáním takového uživatele ze seznamu oprávnění.

## 29.10 Sdílené složky

Ve vlastnostech složky můžeme na kartě **Sharing** (sdílení) zapnout sdílení této složky.



Je možno zvolit **jméno**, pod kterým bude složka na síti vidět. Pokud zvolíme jméno končící dolarem, např. **share\$**, bude tato složka skrytá.

Ze vzdáleného počítače se do složky dostaneme příkazem

**\\jméno\_serveru\share\$**

Zobrazit složky, které máme sdílené, můžeme příkazem **net share**.

```
C:\Documents and Settings\Administrator>net share
```

| Share name | Resource                              | Remark             |
|------------|---------------------------------------|--------------------|
| C\$        | C:\                                   | Default share      |
| ADMIN\$    | C:\WINDOWS                            | Remote Admin       |
| IPC\$      |                                       | Remote IPC         |
| Home       | C:\Home                               |                    |
| NETLOGON   | C:\WINDOWS\SYSVOL\sysvol\test\SCRIPTS | Logon server share |
| Profily    | C:\Profily                            |                    |
| SYSVOL     | C:\WINDOWS\SYSVOL\sysvol              | Logon server share |

The command completed successfully.

Ve správě počítače – **Computer Management** (pravým tlačítkem klikneme na **My Computer** a zvolíme **Manage**) – můžeme najít, kam sdílené **složky fyzicky** vedou.

| Share Name | Folder Path                           | Type   |
|------------|---------------------------------------|--------|
| ADMIN\$    | C:\WINDOWS                            | Window |
| C\$        | C:\                                   | Window |
| Home       | C:\Home                               | Window |
| IPC\$      |                                       | Window |
| NETLOGON   | C:\WINDOWS\SYSVOL\sysvol\test\SCRIPTS | Window |
| Profily    | C:\Profily                            | Window |
| SYSVOL     | C:\WINDOWS\SYSVOL\sysvol              | Window |

### 29.10.1 Mapování složky příkazem

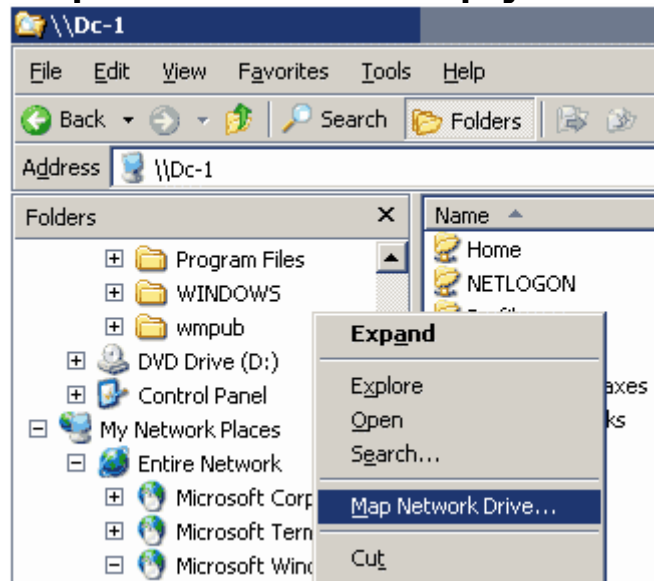
#### **net use písmeno složka**

například

**net use X: \\server\share\$**

### 29.10.2 Mapování složky přes průzkumníka

V průzkumníku vyhledáme složku v síti, klepneme na ni pravým tlačítkem a zvolíme **Map Network Drive – Připojit síťovou jednotku**.



### 29.10.3 Problémy s pravdivým zobrazením míst v síti

V síti je jeden počítač „vládce“ – **Master Browser** – má v sobě databázi informací, co je v síti zapnuto a co je na počítačích sdíleno.

Když si dáme zobrazit místa v síti, počítač kontaktuje master browser, ten jej odkáže na záložní – backup browser a ten teprve počítači odpoví.

Na začátku při startu počítačů probíhají volby, kdo bude master browser a kdo backup browser.

Když se posléze zapne další počítač, vyvolá opětovné volby.

Vše nějakou dobu trvá, proto v danou chvíli nemusí být informace o místech v síti pravdivé.

Pokud chceme počítači nařídit, aby nevyvolával volby a aby nebyl master browser, nastavíme mu v registrech (příkaz **regedit**) parametry **IsDomainMaster** a **MaintainServerList** na hodnotu **false**.

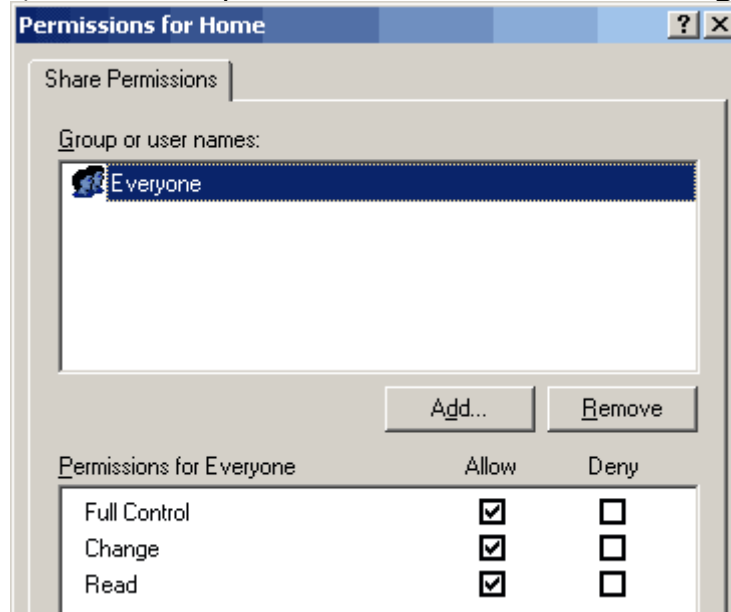
Procházení sítě bude fungovat.

Další možnost je přímo napsat **UNC cestu** sdílené složky na síti, do které se chceme připojit

**\\server\složka**

## 29.11 Sdílená oprávnění

Ve vlastnostech složky je kromě karty Security, kde se nastavují NTFS permissions, také karta pro nastavení sdílení – **Sharing**.



Jsou tři typy sdíleného oprávnění:

- **Read** – číst – přednastaveno pro skupinu Everyone – povoluje číst soubory, povoluje vidět názvy souborů a podadresářů, povoluje spouštět programy.
- **Change** – měnit – zahrnuje v sobě vše, co obsahuje oprávnění Read, povoluje vytvářet soubory a podadresáře, povoluje měnit data v souborech, povoluje mazat podadresáře a soubory.
- **Full Control** – plná kontrola, zahrnuje v soubě vše, co obsahují oprávnění Read a Change, i možnost měnit NTFS oprávnění

U **sdíleného oprávnění nefunguje dědění** jako u NTFS permissions.

Pokud je soubor umístěn v adresáři, který má sdílená oprávnění nastavena na Change, tento adresář je umístěn v adresáři, na kterém jsou nastavena sdílená oprávnění na Read, pak bude výsledné oprávnění pro soubor záležet na tom, přes který sdílený adresář budeme k souboru přes síť přistupovat.

Budeme-li k souboru přistupovat přes adresář s oprávněním Read, budeme moci soubor jen číst.

Budeme-li k souboru přistupovat přes adresář s oprávněním Change, budeme moci soubor i měnit.

## 29.12 Jak spolu souvisí sdílené oprávnění a NTFS permission – zabezpečení

Mezi oprávněními nastavenými na kartě sdílení a na kartě zabezpečení je vztah **průniku**.

Pro zjednodušení práce je vhodné nastavit **sdílená oprávnění** pro skupinu **Everyone** na **Full Control** a na kartě **Security** – zabezpečení – nastavit **NTFS permissions** podle potřeby.

Po průniku s plnými oprávněními z karty Sdílení bude výsledkem oprávnění nastavené na kartě Security.

### **Pozor**

Pokud necháme ve sdíleném oprávnění prázdko, pak bude průnik s NTFS oprávněními prázdko, jakýkoliv přístup bude odepřen!

## 29.13 Cvičení

Vytvořte složku „trida3A“, která bude sdílená v síti. Ve složce budou existovat podsložky jednotlivých studentů třídy. Tyto složky nesdílejte.

Nastavte sdílená oprávnění a NTFS permissions tak, aby každý student mohl ukládat jen do své podsložky a společnou složkou třídy mohl jen procházet.

 *Domácí úkol*

- Zopakujte si rozdíl mezi oprávněními a právy.
- Jak se nastavují NTFS permissions a sdílená oprávnění?
- Co je to DACL, jakou má přibližnou strukturu? Jak se vyhodnocuje oprávnění uživatele pro přístup k danému zdroji?
- Jaký má vliv dědění NTFS permissions na výsledné nastavení oprávnění pro přístup k danému zdroji? Dá se dědění oprávnění zrušit?
- Jaký je výsledek oprávnění po kopírování a přesunu objektu?
- Co je to UNC cesta?
- Jak lze příkazem přimapovat sdílenou složku na síti?
- Jak spolupracují sdílená oprávnění a NTFS oprávnění?

 *Shrnutí*

- ✓ Víte, jak souboru nastavit NTFS oprávnění.
- ✓ Umíte nasdílet složku, umíte ji nasdílet i pod jiným jménem, případně z ní udělat skrytou složku v síti.
- ✓ Umíte vyhodnotit výsledek nastavení oprávnění.
- ✓ Víte, co je DACL a jak se vyhodnocuje.
- ✓ Umíte připojit sdílenou složku na síti.
- ✓ Víte, jak spolupracují sdílená oprávnění a NTFS permissions.
- ✓ Víte, jak dopadnou nastavená oprávnění při kopírování a přesunu objektu.



## 30. RIS – Remote Installation Services (59. - 60. hodina)

### 30.1 Popis

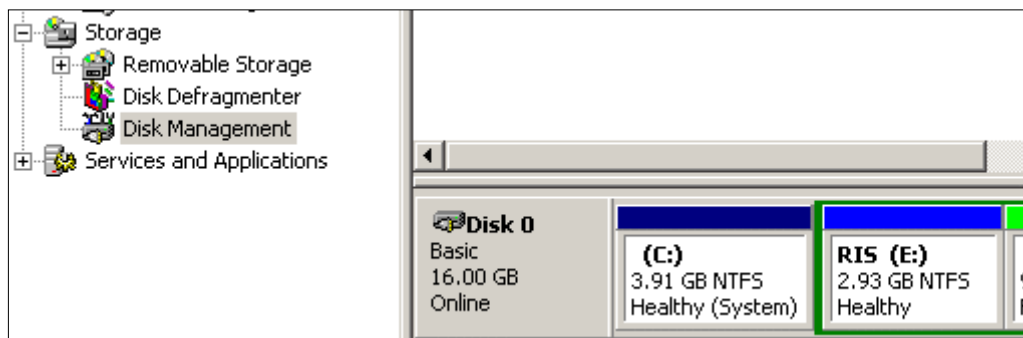
RIS slouží pro automatické vzdálené instalace počítačů v síti.

Pomocí **CD-based** imagů umístěných na RIS serveru můžeme instalovat různé varianty operačních systémů na klientské stanice.

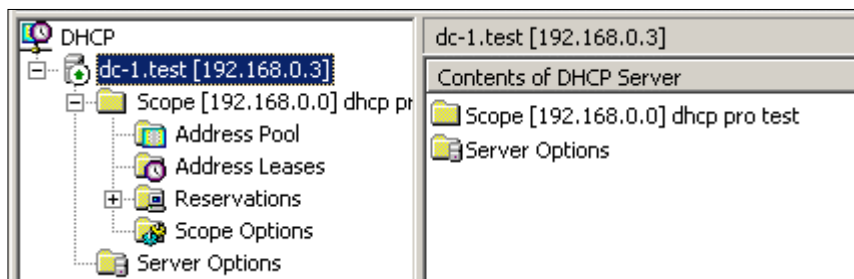
Pomocí **RIPrep** imagů můžeme instalovat operační systém i s aplikacemi.

### 30.2 Požadavky

- Tato služba běží na serveru 2000 nebo serveru 2003.
- Na serveru je nutné mít logickou jednotku se souborovým systémem NTFS, která není ani systémová ani bootovací (RIS používá službu „single instant storage“, která se nesnáší s operačním systémem).



- V síti musí běžet DNS, DHCP a Active Directory. V případě potřeby přidejte roli DHCP serveru. Po konfiguraci DHCP serveru je nutno ještě tento autorizovat v Active Directory (**Administrative Tools – DHCP – Action – Authorize**).



## 30.3 Postup

### 30.3.1 Instalace RIS serveru.

RIS server se přidá jako jiné programy, tj. spustíme **Add/Remove Programs**, zvolíme Add/Remove Windows Components a vyhledáme **RIS – Remote Installation Services**.

### 30.3.2 Konfigurace RIS

Po restartu spustíme průvodce **risetup.exe** (z nabídky Start zvolíme **Run – Spustit** a zadáme příkaz **risetup**, nebo lze tento nástroj najít v nabídce **Administrative Tools**).

Postupujeme podle průvodce.

Určíme NTFS jednotku a adresář, do kterého budeme instalovat podpůrné soubory a základní CD-base image.

Určíme, jakým způsobem bude RIS server odpovídat klientům. Jsou na výběr možnosti:

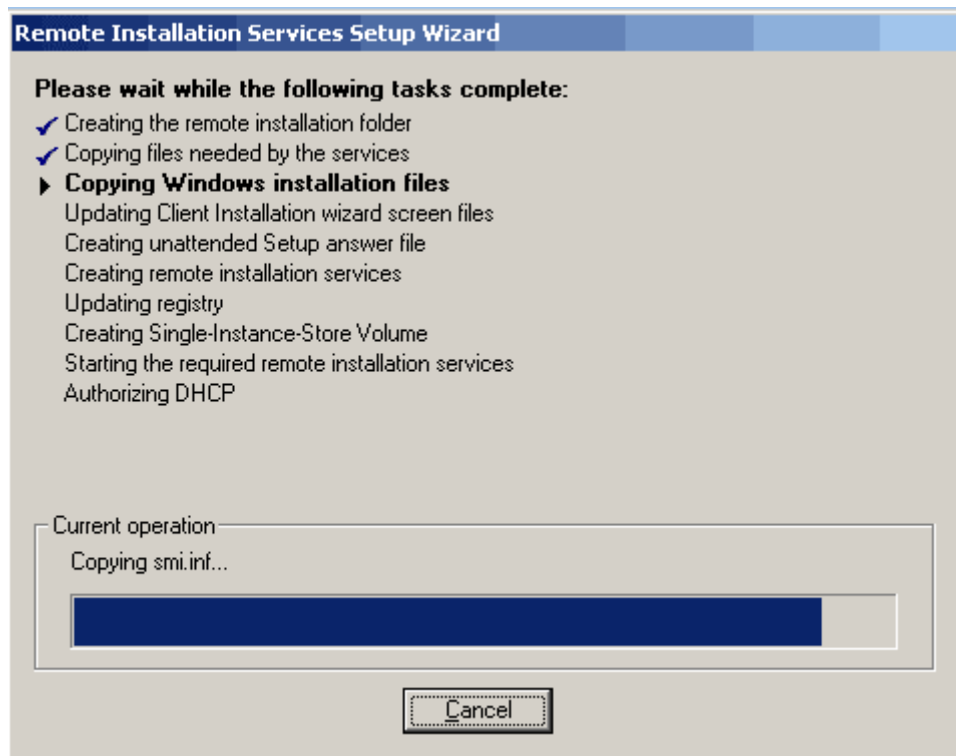
- Neodpovídat na žádné požadavky klientům (je to přednastaveno)
- Odpovídat na všechny požadavky klientů
- Odpovídat na požadavky jen známých klientů (počítače, které budou vybrány) – zaškrtnutím obou políček „Respond to client computers requesting service“ a „Do not respond to unknown client computers“

Dále musíme specifikovat umístění zdrojového instalačního CD (s Windows XP nebo Serverem 2003), ze kterého se vytvoří základní **CD-based image**. Určíme adresář, kam se CD-based image vytvoří.

Specifikujeme popis a vysvětlující text, který se bude zobrazovat uživatelům, když budou po přihlášení k RIS serveru vybírat image ze seznamu v instalačním průvodci.

Po ukončení průvodce budou mimo jiné udělány následující úkoly:

- ❑ vytvoření adresářové struktury RIS
- ❑ vytvoření souboru odpovědí, které dohromady s CD-based imagem vytváří instalační image
- ❑ vytvoření služby RIS
- ❑ úprava registrů
- ❑ vytvoření jednotky „single instance storage“
- ❑ spuštění potřebných služeb na RIS serveru
- ❑ autorizace RIS serveru v Active Directory

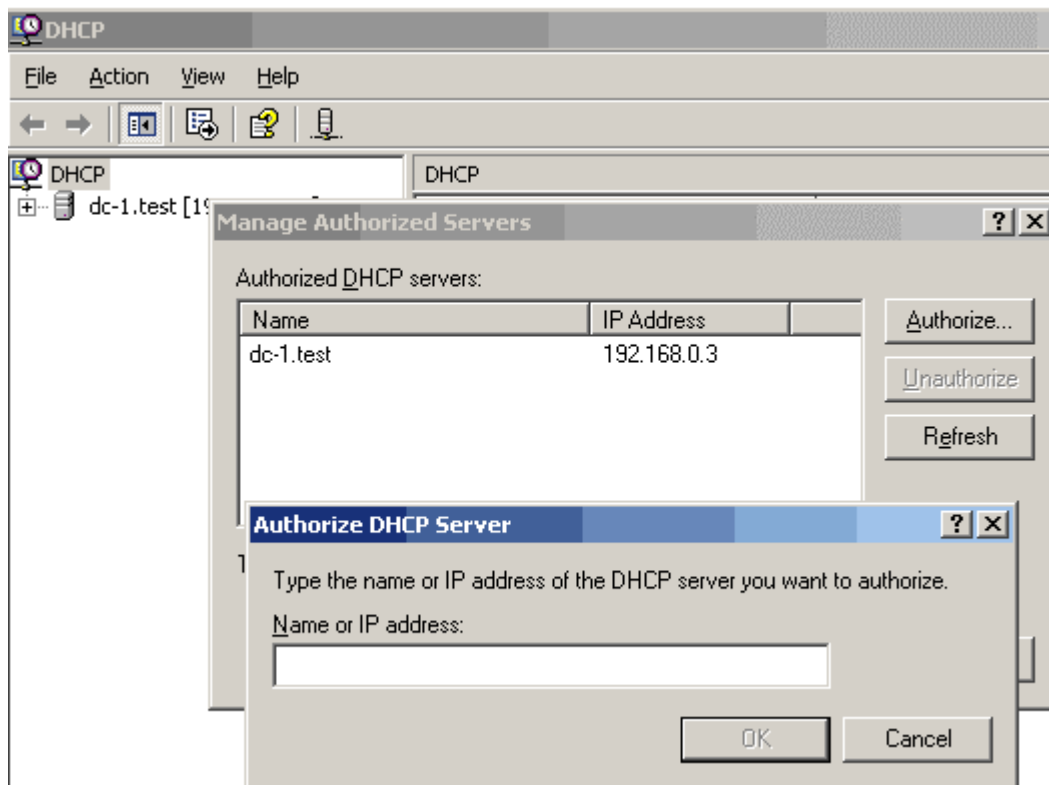


## Autorizace RIS serveru v Active Directory

Předtím, než může RIS server začít odpovídat na požadavky klientů, musí být server autorizován v Active Directory. Průvodce RIS by měl tuto autorizaci provést. Autorizace RIS serveru v Active Directory se dělá proto, aby se zabránilo někomu jinému v instalaci svého RIS serveru, který by šířil neautorizované image na klientské počítače.

Pro autorizaci RIS serveru je potřeba udělat následující:

- Z administrativních nástrojů – **Administrative tools** – spustit **DHCP**.
- V otevřeném okně klikneme pravým tlačítkem na **DHCP** a zvolíme „**Manage Authorized Servers**“.
- V okně „Manage Authorized Servers“ klikneme na „**Authorize**“ a vepíšeme IP adresu nebo jméno RIS serveru, který chceme autorizovat.



### 30.3.3 Nastavení instalačních voleb pro klienty

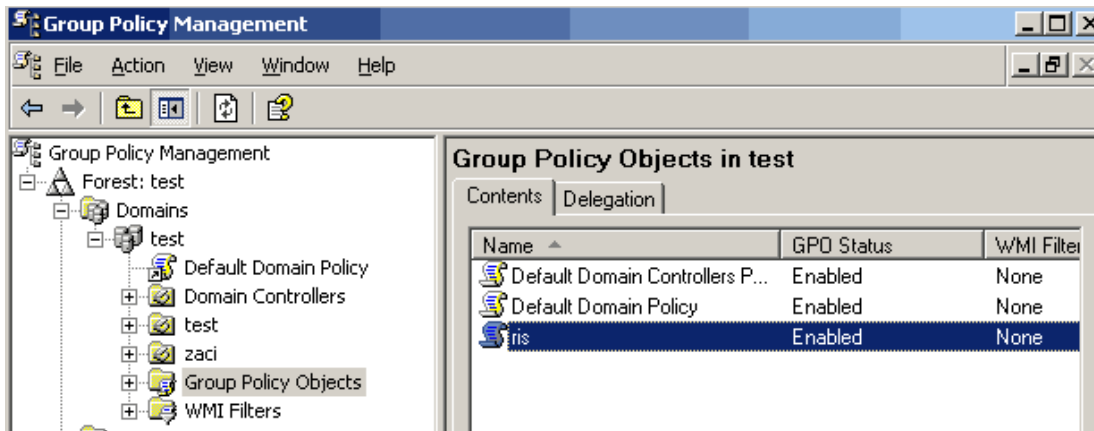
Pomocí **skupinových politik** můžeme rychle nastavit instalační volby pro všechny nebo vybrané počítače.

Pokud ještě není nainstalována **MS Group Policy Management Console**, stáhneme ji z webu Microsoftu a nainstalujeme.

Po nainstalování se použít příkazem **gpmc.msc**.

V okně skupinových politik klepneme na doménu, v rozbalené stromové struktuře zvolíme **Group Policy Objects** – klepneme na tuto

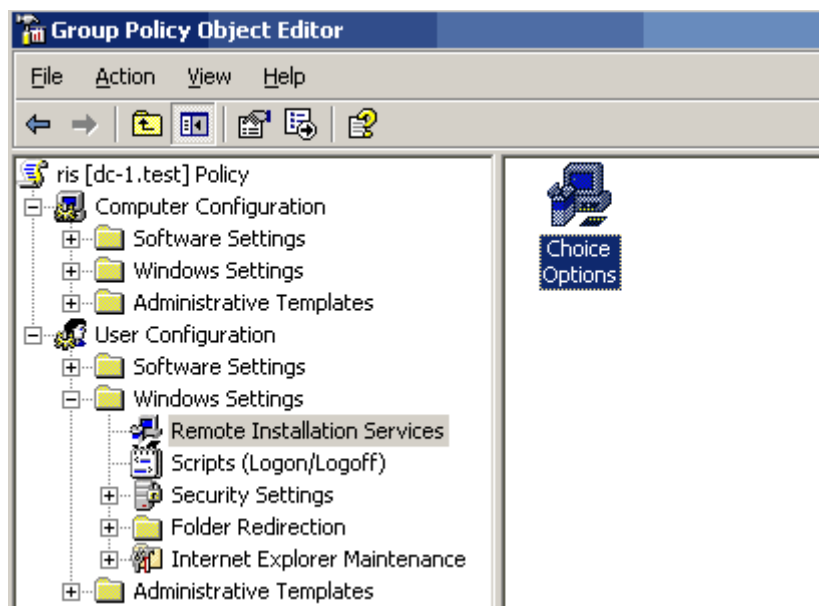
položku pravým tlačítkem a zvolíme **New**. Vytvoříme novou politiku (zde nazvaná **ris**).



Klepeme pravým tlačítkem myši na název politiky a zvolíme **Edit**. Otevře se editor politik.

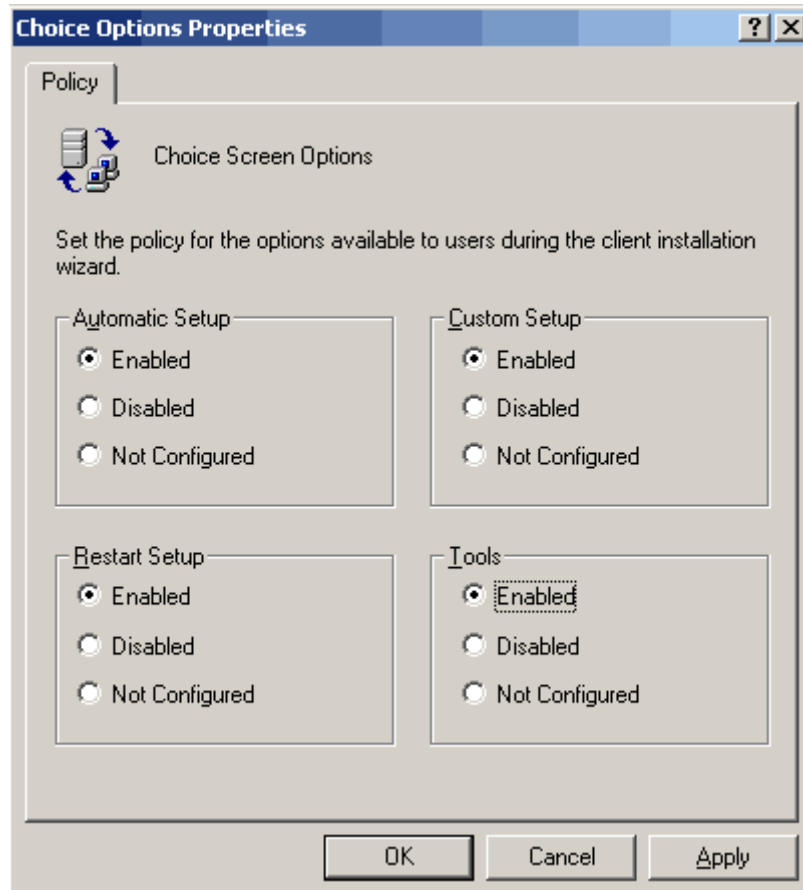
V konzole **Group Policy Object Editor** rozbalíme ve stromové struktuře **User configuration – Windows Settings – Remote installation Services**.

V pravé části detailů klikneme dvakrát na **Choice Options**.

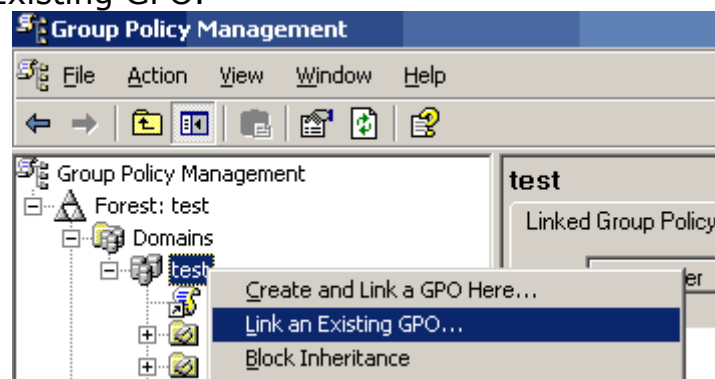


V okně vlastností zaškrtneme **Enabled** u všech čtyř kategorií – Automatic Setup, Custom Setup, Restart Setup, Tools.

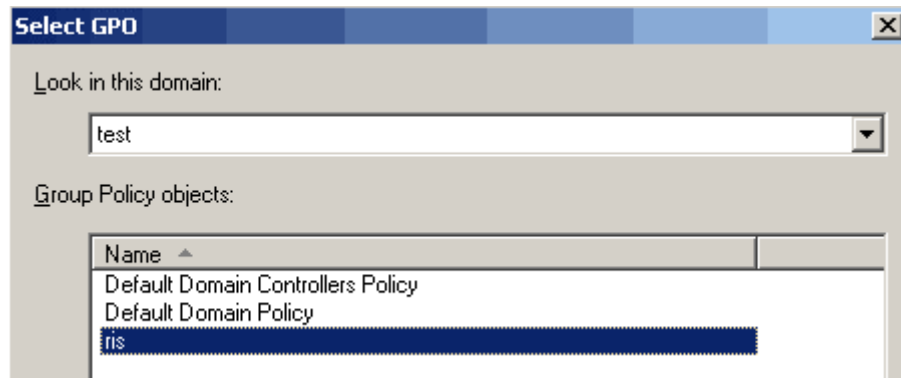
Tím zajistíme, že tyto volby budou dostupně během klientské instalace.



V konzole Group Policy Management klepneme pravým tlačítkem myši na OU nebo celou doménu (podle toho, na co chceme politiku aplikovat) a zvolíme Link an Existing GPO.

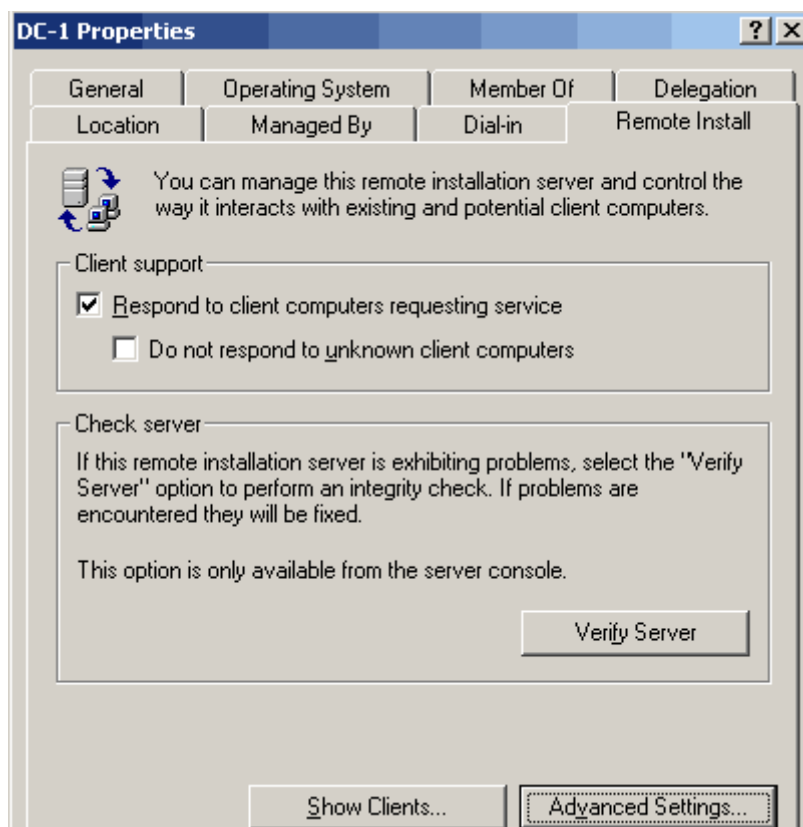


Ze seznamu politik vybereme tu, kterou chceme aplikovat, v našem případě politiku s názvem ris.



### 30.3.4 Nastavení vlastností účtu v Active Directory

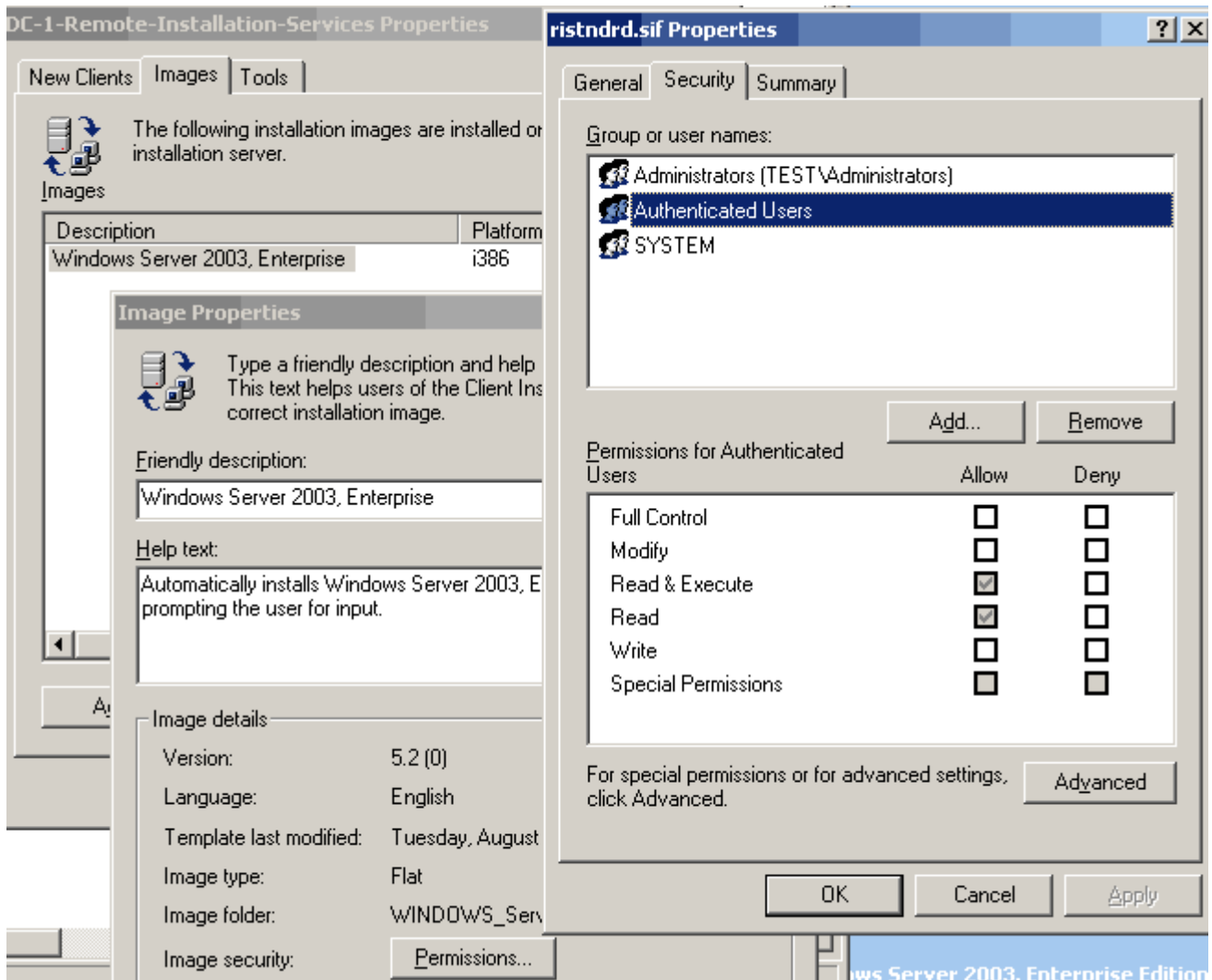
V okně **Active Directory Users and Computers** vybereme řadiče (**Domain Controllers**), pravým tlačítkem klepneme na řadič, na kterém je nainstalován **RIS server** a vybereme **Properties** – vlastnosti. Na kartě „**Remote install**“ zaškrtneme volbu, aby odpovídal klientům.



Klikneme na **Advanced Settings** a zvolíme kartu **Images**.

Na kartě **Images** klikneme na instalaci a ve vlastnostech klikneme na „**permissions**“. Na kartě **Security** zvolíme **Add** – přidat a zvolíme uživatele, kterým dovolíme přístup ke zvolenému image.

Přednastavená oprávnění jsou nastavena tak, aby uživatel mohl použít soubor odpovědí k nainstalování image (jsou nastavena na **Read a Read & Execute**).



### 30.3.5 Vytvoření startovací diskety

Pokud klientské počítače nedokáží bootovat ze sítě (PXE proces), je možné vytvořit startovací disketu, která bude simulovat PXE proces.

Z nabídky **Start** na XP zvolíme **Run** – spustit a napíšeme příkaz **rbfg.exe**. Tento příkaz není na XP přítomen, proto jej spustíme vzdáleně ze serveru.

Ze stanice:

**\\jméno\_RIS\_serveru\reminst\admin\i386\rbfg.exe**



Zobrazí se dialogové okno, do kterého napíšeme administrátorské jméno a heslo pro přístup na server.

Následně se zobrazí dialogové okno pro vytvoření diskety.



### 30.3.6 Instalace image na klientský počítač

Po spuštění počítače zvolíme klávesou F12 bootování ze sítě nebo pomocí startovací diskety tento proces budeme imitovat.

Poté, co se klientský počítač spojí s RIS serverem, bude uživatel vyzván, aby znovu stiskl F12, čímž se spustí instalační průvodce.

Uživatel se zalogueje do domény a dále se zobrazí instalační volby.

**Automatic setup** nechá uživatele vybrat image k instalaci. Pokud je k dispozici jen jeden image, vybere se automaticky.

**Custom setup** umožní uživateli například zadat jméno počítače.

**Restart a Previous Setup Attempt** – vhodné pro případ, že předchozí instalace operačního systému havarovala.

## 30.4 Vytvoření RIPrep imagů

Jedná se o vytvoření image obsahujícího kromě operačního systému také aplikace.

Nejprve je potřeba nainstalovat jeden zdrojový počítač včetně aplikací a konfigurací. Je vhodné vše otestovat, protože po vytvoření image nejsou změny možné, je možné pouze image vytvořit celý znovu.

Dále je vhodné zastavit všechny nepotřebné programy a služby před vytvářením image.

Není vhodné přenášet a klonovat některé unikátní identifikátory, jako například SID (Security Identifier – číslo počítače vzniklé při instalaci), jméno počítače.

K odstranění těchto unikátních identifikátorů slouží například utilita **NewSID**, kterou lze stáhnout z webu <http://www.sysinternals.com/Utilities/NewSid.html>

K vytvoření RIPrep image slouží příkaz **riprep.exe** spuštěný z nabídky **Start – Run** (spustit) na zdrojovém počítači. Tento příkaz není na XP přítomen, proto jej spustíme vzdáleně ze serveru.

**\\jmeno\_RIS\_serveru\reminst\admin\i386\riprep.exe**

Zadáme potřebné parametry (jméno RIS serveru, umístění složky pro image, popis image) a postupujeme podle výzev průvodce.

 *Domácí úkol*

Projděte si celý proces vzdálené instalace operačního systému pomocí CD-based imagů a RIPrep imagů.

Podle možností vyzkoušejte prakticky.

 *Shrnutí*

- ✓ Seznámili jste se s možností vzdálené instalace pomocí RIS.
- ✓ Víte, jaké možnosti nabízí práce s CD-based imagem a s RIPrep imagem.
- ✓ Umíte nainstalovat RIS server.
- ✓ Víte, jak bootovat ze sítě a umíte vytvořit startovací disketu pro případ, že bootování ze sítě není automaticky podporováno.
- ✓ Umíte vytvořit RIPrep image a CD-based image.
- ✓ Víte, jak z těchto imagů nainstalovat vzdáleně počítač.

## 31. Opakování (61. – 62. hodina)

## 32. Opakování (63. – 64. hodina)

## 33. Rejstřík pojmů

---

/

/ · 18

---

### A

AGDLP · 260  
APIPA · 231  
aptitude · 11, 76, 77, 112, 120, 121, 124,  
167, 176  
apt-setup · 119

---

### B

bc · 68  
bin · 18  
boot · 19  
Bridge · 201  
broadcast · 199  
Broadcast · 190  
Broadcast doména · 200  
bus · 199

---

### C

cal · 66  
CAL · 228  
cat · 32  
cd · 26  
clear · 39  
cp · 70  
cut · 103

---

### D

DACL · 264  
dev · 19  
df · 42  
DHCP · 179  
distribution · 255  
dmesg · 115  
DNS · 179  
Doména · 229  
dpkg --get-selections · 119  
dpkg -l · 119  
dpkg --set-selections · 119  
dsadd · 244  
dsget · 259  
du · 39

---

### E

egrep · 47  
echo · 43  
etc · 19  
ethereal · 173  
ext3 · 12  
extended star · 199

---

### F

Faillog · 122  
fdisk · 14  
fg · 79  
find · 65  
finger · 102  
Formátování · 16

---

### G

Gateway · 231  
Gkrellm · 115  
GPL · 9  
grep · 44  
groupadd · 92  
groupdel · 92  
groups · 102  
grpquota · 107  
GRUB · 12

---

### H

hard link · 59  
hdparm · 113  
head · 34  
hierarchická topologie · 199  
home · 20  
hostname · 170  
Hub · 201  
Hwinfo · 112

---

### Ch

chgrp · 97  
chmod · 97  
chown · 96

---

### I

id · 103  
ifconfig · 165  
info · 24

Inode · 62  
Instalace balíčku · 76  
interfaces · 167  
IP adresa · 177  
ipconfig · 230

---

## **J**

jobs · 74

---

## **K**

kill · 83  
killall · 83  
klávesnice · 36  
Kolizní doména · 200  
Kontejner · 238  
Kontrola systému · 75  
Krimpovací kleště · 207

---

## **L**

LAN · 178  
Last · 122  
Lastlog · 122  
LDAP · 243  
Les · 237  
less · 35  
lib · 20  
LILO · 12  
locate · 64  
lost+found · 20  
ls · 27, 95

---

## **M**

MAC adresa · 177  
man · 24  
MAN · 179  
maska · 178  
Mc · 56  
media · 20  
mesh · 199  
mkdir · 71  
MMC · 240  
mnt · 20  
more · 35  
mount · 24, 134  
Multicast · 189  
Multitasking · 11  
Multithreading · 11  
Multiuživatelský · 11  
mv · 69

---

## **N**

NCP · 13  
net use · 133  
netstat · 116  
NFS · 13  
nice · 83  
nslookup · 251  
NTFS · 12

---

## **O**

Oprávnění · 262  
opt · 20  
Organizační jednotka · 238  
OSI model · 199

---

## **P**

passwd · 88  
Patch panel · 218  
ping · 166  
Ping · 182  
Pracovní skupina · 229  
Práva · 262  
Privátní IP adresy · 189  
privátní klíč · 127  
proc · 20  
ps · 79, 117  
pstree · 81, 117  
pwd · 39

---

## **Q**

quota · 108

---

## **R**

ReiserFS · 12  
renice · 83  
Repeater · 201  
resolv.conf · 171  
ring · 199  
RIS · 273  
RJ-45 · 207  
rm · 71  
rmdir · 72  
root · 20  
route print · 231  
Router · 204  
run as · 239

---

## **S**

SAM · 229  
samba · 131

---

---

SAN · 179  
sbin · 21  
scp · 126  
security · 254  
Server 2003 · 227  
SGID · 99  
Shell · 12  
shutdown · 66  
SID · 264  
Site · 238  
SMB · 13  
smbclient · 131, 136  
smbfs · 131  
smbmount · 135  
smbpasswd · 132  
Smbstatus · 133  
sort · 103  
srv · 21  
SSH · 125  
star · 199  
Sticky · 99  
STP · 203  
strom · 237  
su · 39  
SUID · 100  
Switch · 202  
symlink · 60

---

## **T**

tac · 33  
tail · 34  
tee · 64  
testparm · 132  
tmp · 21  
token passing · 199  
top · 82, 117  
touch · 33  
traceroute · 172  
tracert · 182  
Třídy IP adres · 187

---

## **U**

umask · 106  
umount · 26  
uname · 116  
Unicast · 189  
uptime · 116  
useradd · 87  
userdel · 90  
usermod · 89  
users · 102  
usr · 21  
usrquota · 107  
UTP · 208

---

## **V**

var · 21  
veřejný klíč · 127  
vi · 58  
VLAN · 203  
vstupy · 48  
výstupy · 48

---

## **W**

w · 102  
WAN · 179  
who · 48, 102  
whoami · 39

---

## **X**

xeyes · viz  
xosview · 114

---

## **Z**

Zásuvka RJ-45 · 216  
Zombie · 84