



Vyšší odborná škola obalové techniky
a střední škola, Štětí

Digitální učební materiály

Operační systémy - Linux

Ivan Pomykacz



esf evropský
sociální
fond v ČR



EVROPSKÁ UNIE



MINISTERSTVO ŠKOLSTVÍ,
MLÁDEŽE A TĚLOVÝCHOVY



OP Vzdělávání
pro konkurenceschopnost

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Licence



Digitální učební materiály, jejímž autorem je Ivan Pomykacz, podléhají licenci [Creative Commons: Uvedte autora - Nevyužívejte dílo komerčně - Zachovejte licenci 3.0 Unported](https://creativecommons.org/licenses/by-nc-sa/3.0/).

Vytvořeno na základě tohoto díla: <http://dumy.odbornaskola.cz/pomykacz>

Práva nad rámec této licence jsou popsána zde: <http://dumy.odbornaskola.cz/pomykacz>.

Disclaimer

Tento PDF dokument byl strojově vygenerován z HTML stránek

<http://dumy.odbornaskola.cz/pomykacz/>.

Je tedy možné, že sazba textu může obsahovat chyby. Jde převážně o vizuální a typografické chyby, které mohou narušit plynulou čitelnost textu. V některých případech může být ohrožena i funkčnost některých komponent (jako vnitřní odkazy).

Vzhledem k tomu, že vypracované materiály nebyly nikdy určeny pro výstupní formát PDF, autor se zříkává jakékoli odpovědnosti za nalezené chyby. Nesnažte se proto v této souvislosti autora kontaktovat.

Programové vybavení

Textový procesor

Obsah

- Co je to Linux?
- Linuxové distribuce
- Stažení a instalace
- Příkazový shell
- Základní příkazy
- Nastavení sítě
- Správa uživatelských účtů
- Práva k souborům
- Správa úložiště
- Správa balíčků
- Diskové kvóty
- Pokročilý příkazový shell
- Správa procesů
- Plánování úloh
- Logování
- Archivace dat
- Skriptování
- Linux v příkazech
- Grafické prostředí
- Virtualizace

Co je to Linux?

Název školy	Vyšší odborná škola obalové techniky a Střední škola, Štětí, příspěvková organizace		
Adresa školky	Kostelní 134, 411 08 Štětí		
IČ	46773509		
Název operačního programu	OP Vzdělávání pro konkurenceschopnost		
Registrační číslo	CZ.1.07/1.5.00/34.1006		
Označení vzdělávacího materiálu	VY_32_INOVACE_21_PSS_401		
Název tématické oblasti (sady)	Operační systémy		
Název materiálu	Co je to Linux?		
Anotace	Materiál obsahuje prezentační a výkladovou část o Linuxu. Představuje pestrý svět Linuxu formou odvozených děl. Předkládá rozpor otevřeného přístupu (svobodných licencí) a proprietárního řešení. Dotýká se tedy pojmů licencování, GNU, GPL, FSF, FOSS.		
Autor	Ivan Pomykacz	Jazyk	český
Očekávaný výstup	Povědomí o alternativních operačních systémech založených na jiných filosofických principech, jako open source a free software.		
Klíčová slova	Linux, GNU, distribuce, jádro, FSF, FOSS, GPL		
Druh výukového zdroje	Výklad	Věková skupina žáků	17+
Typ interakce	aktivita	Ročník	3.
Speciální vzdělávací potřeby	žádné		
Zhotoveno, (datum/období)	10.09.2013	Celková velikost	

Obsah

- [Prezentace](#)
- [GNU Linux](#)
- [Licence GNU/GPL](#)
- [Vývoj jádra Linuxu](#)

Prezentace

online verze: <http://prezi.com/>

offline verze: [co-je-to-linux.pdf](#)



GNU Linux

Správné označení je GNU Linux, alespoň pro systém, jak jej známe dnes. U zrodu byli dva pánové: Richard Stallman a Linus Torvalds.

Oba dva nepřímo na sobě začali vyvíjet svůj vlastní operační systém. Jednou z hlavních motivací pro ně bylo vyvinout Unix-like operační systém. Unixy byly (jsou) uznávané OS, ale též velice drahé.

Na začátku zrodu GNU Linux to vypadalo asi tak, že Linus Torvalds měl jádro (kernel) operačního systému, ale neměl aplikace, které s jádrem tvoří operační systém. Richard Stallman měl zase aplikace, ale neměl hotové jádro. Spojením pak vzniklo označení GNU Linux.

Licence GNU/GPL

Cílem R. Stallmana nebyl ovšem jen nějaký operační systém, ale **svobodný** operační systém. Tzn. takový, který lze šířit, zkoumat, upravovat dle libosti. Aby to bylo v našem světě možné, resp. aby bylo možné zaručit tyto svobody, bylo nezbytné sepsat licenci, která z právního hlediska tyto věci zaručí. Výstupem je pak licence **GNU/GPL** pod kterou je GNU Linux šířen společně s ohromným množstvím software.

Vývoj jádra Linuxu

Na vývoji linuxového jádra se dnes podílí velké množství vývojářů z různého prostředí, tj. korporátního nebo nezávislého. Tzn. jsou zde společnosti jako RedHat, Intel, Novell nebo IBM, které přispívají do vývoje jádra, ale z celkového počtu přispěvatelů tu je 19% nezávislých vývojářů, kteří přispívají psaním kódu.

Linuxové distribuce

Název školy	Vyšší odborná škola obalové techniky a Střední škola, Štětí, příspěvková organizace		
Adresa školky	Kostelní 134, 411 08 Štětí		
IČ	46773509		
Název operačního programu	OP Vzdělávání pro konkurenceschopnost		
Registrační číslo	CZ.1.07/1.5.00/34.1006		
Označení vzdělávacího materiálu	VY_32_INOVACE_21_PSS_402		
Název tematické oblasti (sady)	Operační systémy		
Název materiálu	Linuxové distribuce		
Anotace	Představení linuxových distribucí jako operačních systémů s vlastním vývojem a odlišnostmi. Rozdělení zaměření distribucí na desktop a server a tedy i cílovou skupinu uživatelů. Výhody/nevýhody této diverzity pro uživatele.		
Autor	Ivan Pomykacz	Jazyk	český
Očekávaný výstup	Orientace v linuxovém světě distribucí. Volí distribuci pro konkrétní nasazení v dané situaci.		
Klíčová slova	Linux, distribuce, desktop, server		
Druh výukového zdroje	Výklad	Věková skupina žáků	17+
Typ interakce	aktivita	Ročník	3.
Speciální vzdělávací potřeby	žádné		
Zhotoveno, (datum/období)	10.09.2013	Celková velikost	

Obsah

- Co je to distribuce
 - Odlišnosti
- Distribuce pro desktop
- Distribuce pro server
- Specializované distribuce

Co je to distribuce

Samotný Linux je pouze jádro OS. Jádro je "pouze" mezičlánek mezi hardware a software v počítači. Tzn., má nastrosti komunikaci s HW, spravuje paměť, spouští programy, přistupuje k diskům atd.

To, co se nabalí kolem jádra jsou aplikace, které činí operační systém použitelným. Aplikace jsou na jádře závislé (odtud platformní závislost). Není možné vzít jen tak aplikaci z jedno operačního systému, např. MS Windows a spustit ji na linuxu.

Kolekce takových aplikací - mezi nimiž je např. i desktopové prostředí, pak tvoří distribuci Linuxu.

Odlišnosti

Linuxové distribuce se mohou lišit v mnoha ohledech. Tím, na první pohled, nezjevnějším je vzhled distribuce, tj. použitý příkazový shell nebo rovnou grafická nadstavba (Gnome, KDE, XFCE, LXQT apod.). Dále je to vybraná kolekce software, která je zpravidla předinstalovaná - tzn. jedna distribuce má primární předinstalováno LibreOffice a jiná zase KOffice. Většinou jde o drobnosti: přehrávač hudby, správce souborů, webový prohlížeč atd.

Zpravidla platí, že pokud je nějaký software v jedné distribuci, pak ho najdete i v jiné. Jenom prostě není nainstalovaný ve výchozím stavu.

Další detaily jsou potom až "pod kapotou". Distribuce bývají zaměřené na desktopy nebo servery. Distribuce mají různý vývojový cyklus (rychlost vydávání nových verzí). Používají různé verze jader (s čímž souvisí mimo jiné podpora nejnovějšího HW). Distribuce se předhánají v rychlosti startování OS. Některé distribuce mají odlišný přístup k bezpečnosti (stupeň paranoii). Jiné distribuce preferují čistě svobodný SW (GNU/GPL a jiné, příbuzné licence), což se pak může projevit i tím, že máte problém přehrát třeba mp3 soubor bez nutnosti instalace sw třetích stran.

Je toho opravdu hodně a je z čeho vybírat. Někdo v tuto chvíli jásá, jiný utíká se zděšením zpět k MS Windows - ty jsou jenom jedny a pořád stejné.

Distribuce pro desktop

To jsou takové distribuce, které jsou optimalizovány především na běžného uživatele. Myslím, že je třeba zmínit, že Linux se mezi běžné uživatele pustil teprve nedávno - doménou Linuxu jsou především servery. Z toho vyplývá, že Linux nemusí, resp. ne vždy nabízí komfort jiných operačních systémů, které pro změnu mažou běžnému uživateli med kolem huby. Naproti tomu, většina linuxových distribucí je šířená zdarma (tj. bez zpoplatněné licence).

- Linux Mint
- Ubuntu

- Debian
- Fedora
- openSUSE
- Elementary OS
- Mageia
- Gentoo
- Arch
- Slackware

Distribuce pro server

Zaměření na server zpravidla znamená jiný typ software, který je instalovaný ve výchozím stavu a jinak přednastavené parametry. Nicméně desktopové a serverové distribuce lze i zaměnit - jde tu spíše jen o pohodlnost pro uživatele, resp. správce sítě.

- RHEL
- SLES
- CentOS
- Debian
- Gentoo
- Slackware

Specializované distribuce

Tyto distribuce použijete při různých příležitostech. Jde např. o různé servisní distribuce, pomocí kterých můžete opravit jiný rozbitý systém. Nebo jsou to distribuce pro geeky, které baví stavět si svůj systém. Nebo prostě jinak cílené distribuce "na jedno" použití (nikoli jednorázové).

- Gentoo
- Arch
- SystemRescueCD
- Tiny Core
- Puppy Linux
- Slax
- SteamOS

Stažení a instalace

Název školy	Vyšší odborná škola obalové techniky a Střední škola, Štětí, příspěvková organizace		
Adresa školky	Kostelní 134, 411 08 Štětí		
IČ	46773509		
Název operačního programu	OP Vzdělávání pro konkurenceschopnost		
Registrační číslo	CZ.1.07/1.5.00/34.1006		
Označení vzdělávacího materiálu	VY_32_INOVACE_21_PSS_403		
Název tematické oblasti (sady)	Operační systémy		
Název materiálu	Stažení a instalace		
Anotace	Popisuje možnosti stažení a instalace konkrétní linuxové distribuce Debian Linuxu. Od stažení instalačního obrazu, přes průběh instalace až po přihlášení do systému.		
Autor	Ivan Pomykacz	Jazyk	český
Očekávaný výstup	Nainstaluje na Linuxu založený operační systém.		
Klíčová slova	Linux, Debian, instalace, stažení, root		
Druh výukového zdroje	Výklad	Věková skupina žáků	17+
Typ interakce	aktivita	Ročník	3.
Speciální vzdělávací potřeby	žádné		
Zhotoveno, (datum/období)	17.09.2013	Celková velikost	

Obsah

- Způsob instalace
- Stažení
 - Architektura
 - ISO pro Debian GNU/Linux
- Instalace
 - Název počítače
 - Uživatel root
 - Rozdělení disku
 - Síťové zrcadlo
 - Úlohy
 - Grub

Způsob instalace

Před samotnou instalací systému je třeba rozhodnout o způsobu instalace. Zpravidla víme, kam budeme instalovat, takže zbývá zvolit z čeho.

Většina linuxových distribucí nabízí následující postup:

1. Stáhnout obraz ISO (někdy i v tzv. live verzi).
2. Obraz vypálit na CD/DVD nebo nahrát na flash disk.
3. Spustit počítač v režimu, kdy je umožněno tzv. nabootovat z jiného média (CD/DVD nebo flash disk), než z pevného disku (HDD, SSD).
4. Po spuštění instalátoru pokračovat dle jeho instrukcí.

Kroky 3-4 lze pak opakovat na dalších počítačích se stejným médiem (flash diskem). Při vyšší počtu (i třeba souběžných) instalací je ale na pováženu, zda-li nezvolit právě jiný způsob instalace, zejména kvůli časové úspoře.

Pokud zamýšlíte instalovat do virtuálního stroje, pak samozřejmě krok 2 vynecháváte zcela. ISO obrazy lze totiž použít tak, jak jsou a přímo je např. připojit do virtuálního počítače.

Stažení

Každá linuxová distribuce má na svých webových stránkách (www.debian.org) odkaz, který vás nasměruje na stažení ISO obrazu.

Pokud máte stabilní a rychlý internet, nebojte se využít možností jako **síťové instalace**. Pokud ovšem z nějakého důvodu potřebujete tzv. offline instalaci, pak jednoduše stáhněte plnohodnotný obraz (CD nebo DVD). Čím větší obraz, tím více balíčků (software) budete mít k dispozici v offline podobě.

Pokud ovšem plánujete svojí distribuci udržovat aktualizovanou, pak se bez připojení do sítě neobejdete. Jsou sice možnosti, jak si vytvořit lokální zrcadlo repozitáře s balíčky, ale to si necháme do kapitoly o správě software v Debian GNU/Linuxu.

Architektura

V závislosti na zvolené distribuci vás průvodce před jejím stažením vyzve k volbě architektury













procesoru. Pokud to neudělá, tak jste pravděpodobně stáhli verzi pro 64bit procesory (amd64).

Ve výjimečných případech vás může zajímat architektura pro 32bit procesory (i386) nebo i jiné architektury, např. armel, powerpc nebo sparc (tj. "neintelovské" procesory).

ISO pro Debian GNU/Linux

Co tedy stáhnout? V [tomto výpise](#) vás bude zajímat CD-1 s nejvyšším číslem v názvu, hned za slovem "debian". Podle obrázku níže, je to: `debian-7.4.0-amd64-CD-1.iso`.

Index of /debian-cd/current/amd64/iso-cd

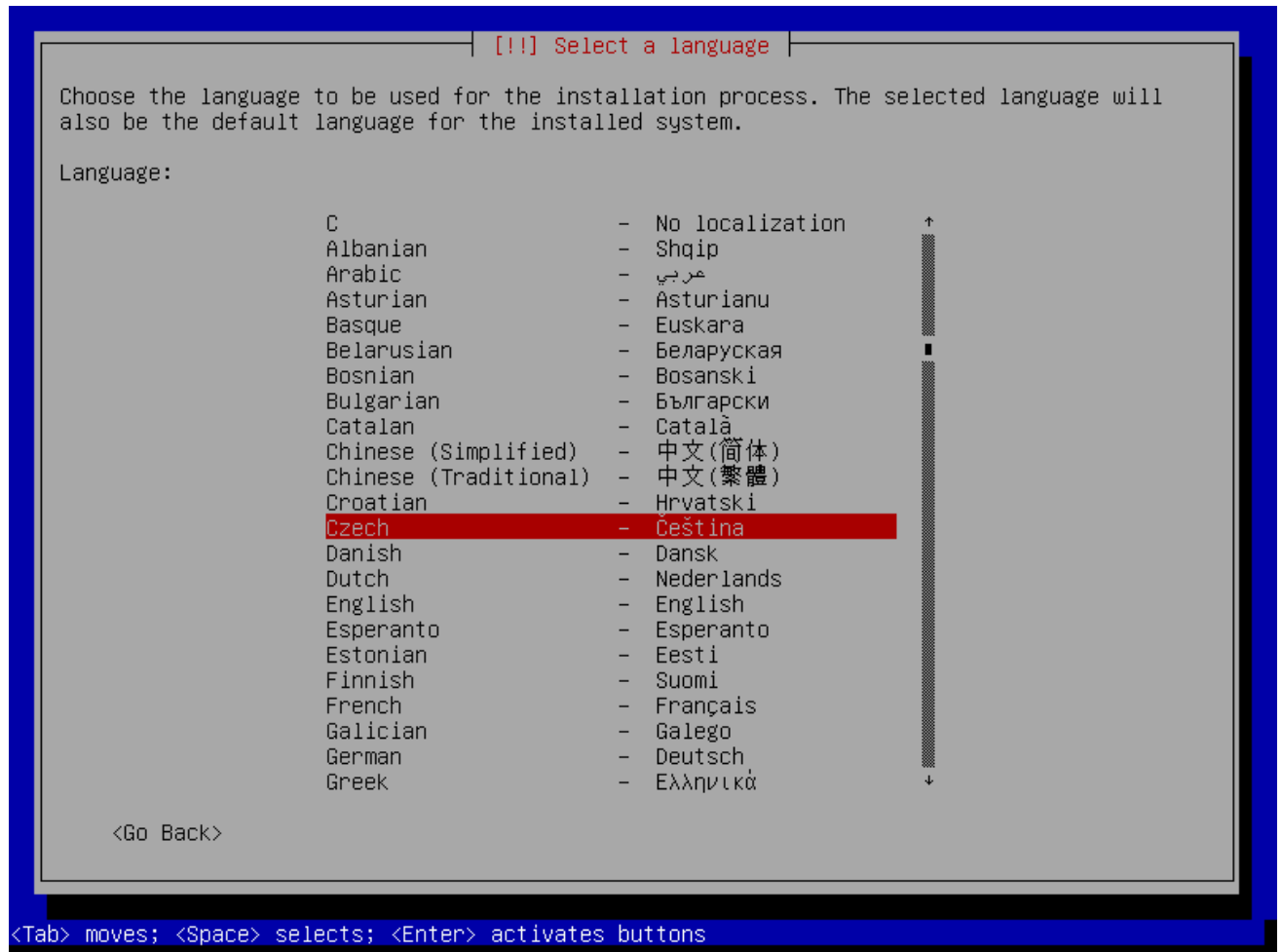
Name	Last modified	Size
 Parent Directory		-
 MD5SUMS	2014-02-09 19:25	5.0K
 MD5SUMS.sign	2014-02-09 19:31	836
 SHA1SUMS	2014-02-09 19:25	5.7K
 SHA1SUMS.sign	2014-02-09 19:31	836
 SHA256SUMS	2014-02-09 19:25	7.6K
 SHA256SUMS.sign	2014-02-09 19:31	836
 SHA512SUMS	2014-02-09 19:25	13K
 SHA512SUMS.sign	2014-02-09 19:31	836
 debian-7.4.0-amd64-CD-1.iso	2014-02-08 15:30	623M
 debian-7.4.0-amd64-CD-2.iso	2014-02-08 15:30	633M
 debian-7.4.0-amd64-CD-3.iso	2014-02-08 15:30	648M

Instalace

Následující instalace je provedena na verzi Debian GNU/Linux 7.4.0 (Wheezy).



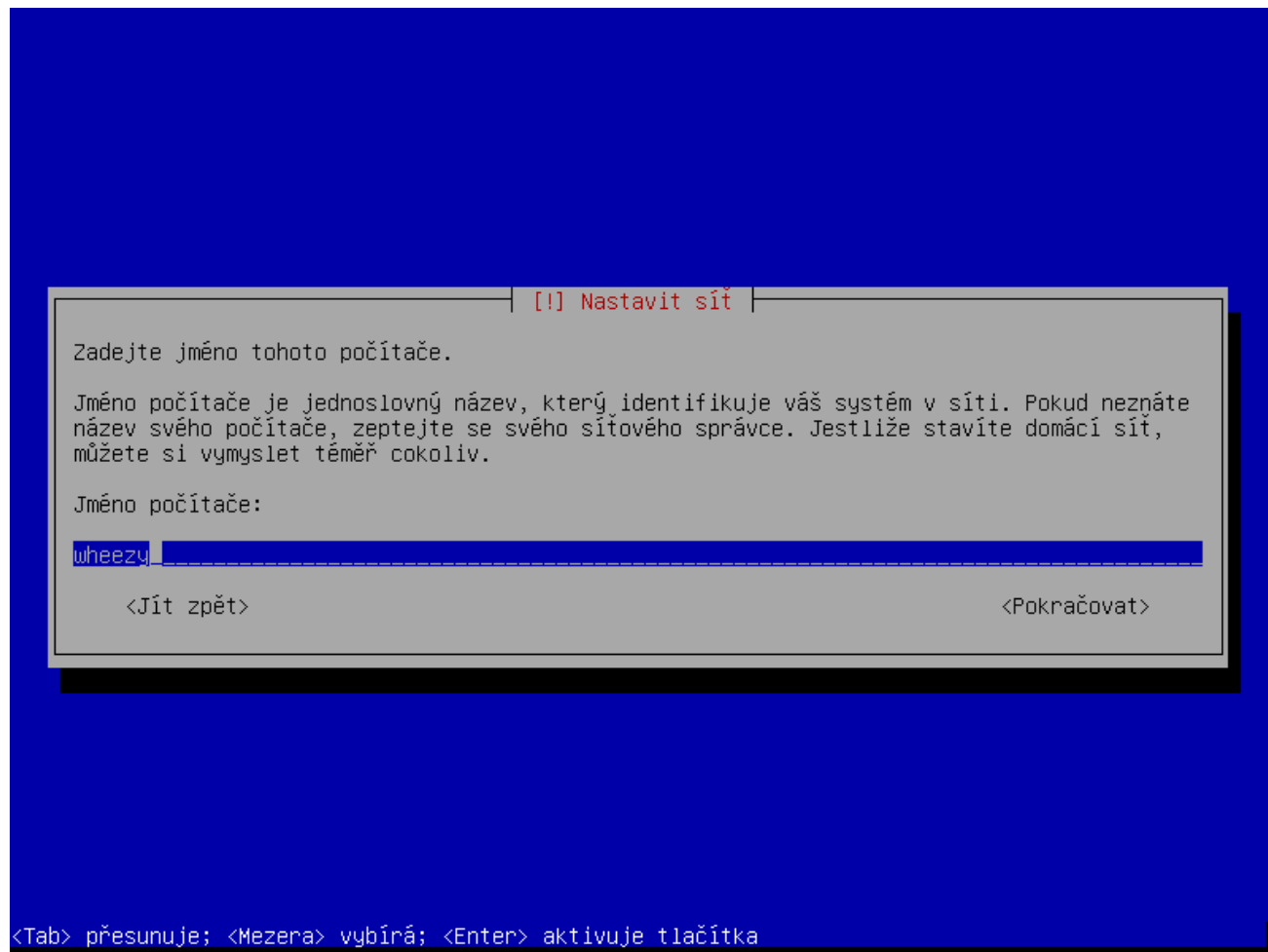
Po úvodní obrazovce a spuštění instalátoru se zobrazí volba jazykového nastavení (včetně klávesnice).



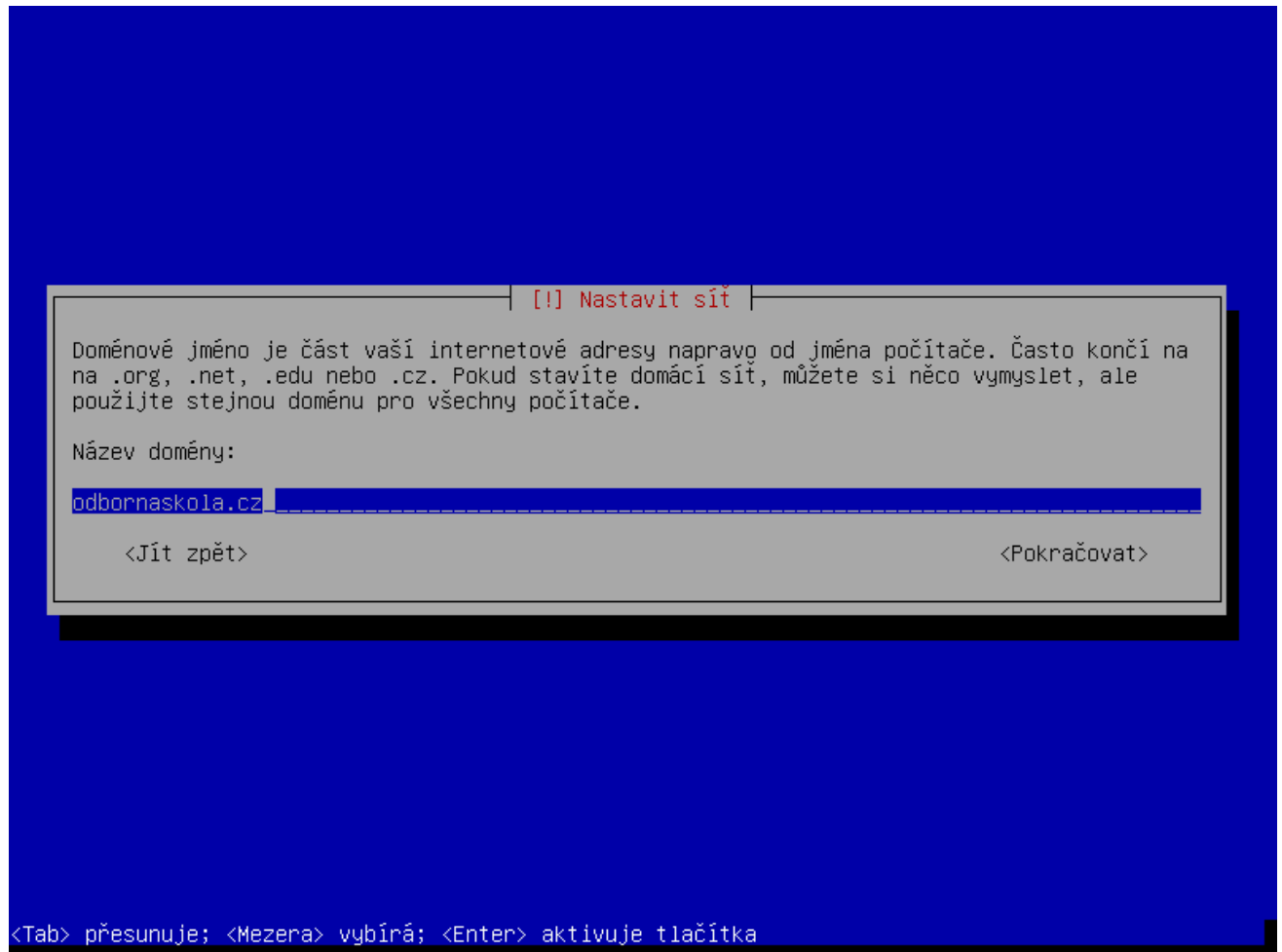
Název počítače

Následuje zadání názvu počítače, tj. název, který se bude objevovat v síti a v terminále po přihlášení. Většinou nejzásadnější část celé instalace, zbytek už je pak hračka.

Uživatel root

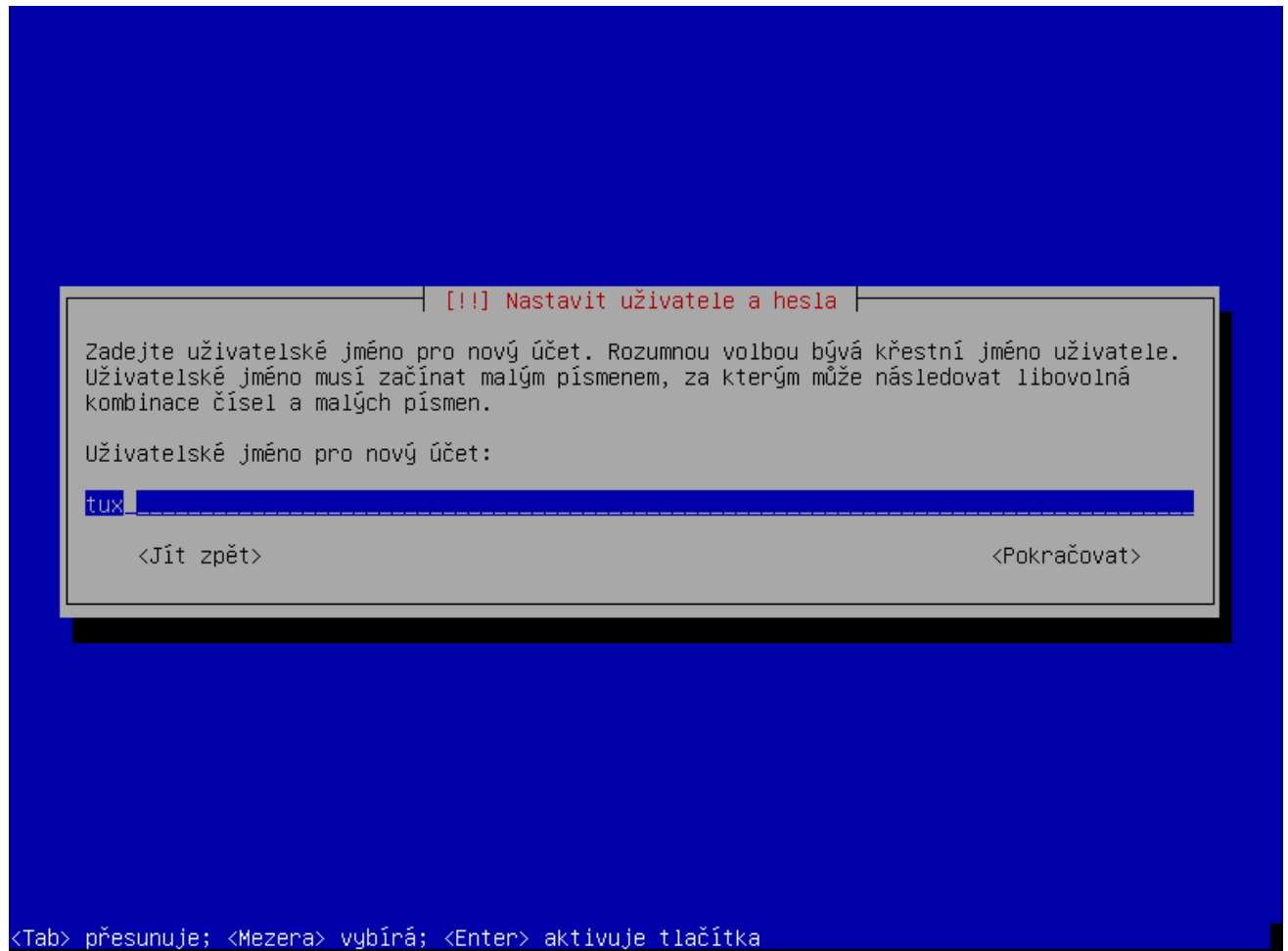
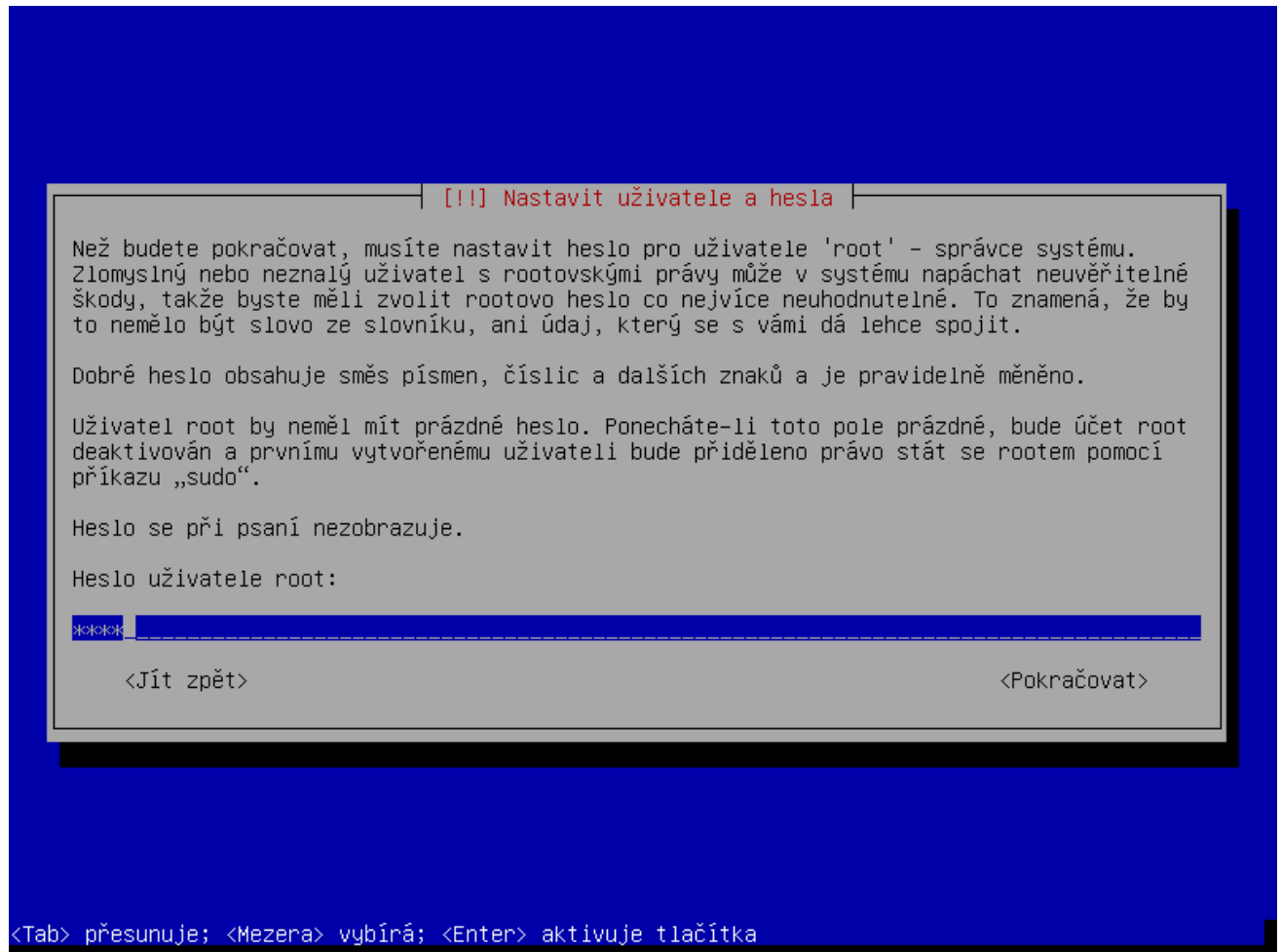


Při konfiguraci sítě můžete být dotázáni na název doménového jména. Nebudeme nyní nějak zvlášť řešit, protože nastavením sítě se budeme ještě detailněji zabírat v dalších cvičeních.



Následuje část vytvoření hesla pro uživatele "root". Jak instalátor popisuje, jde o správce systému. Uživatel root má mezi uživateli systému výjimečné postavení, může totiž provádět cokoli - má na všechno práva.

Volba silného hesla je proto naprosto zásadní. Vizte text instalátoru.

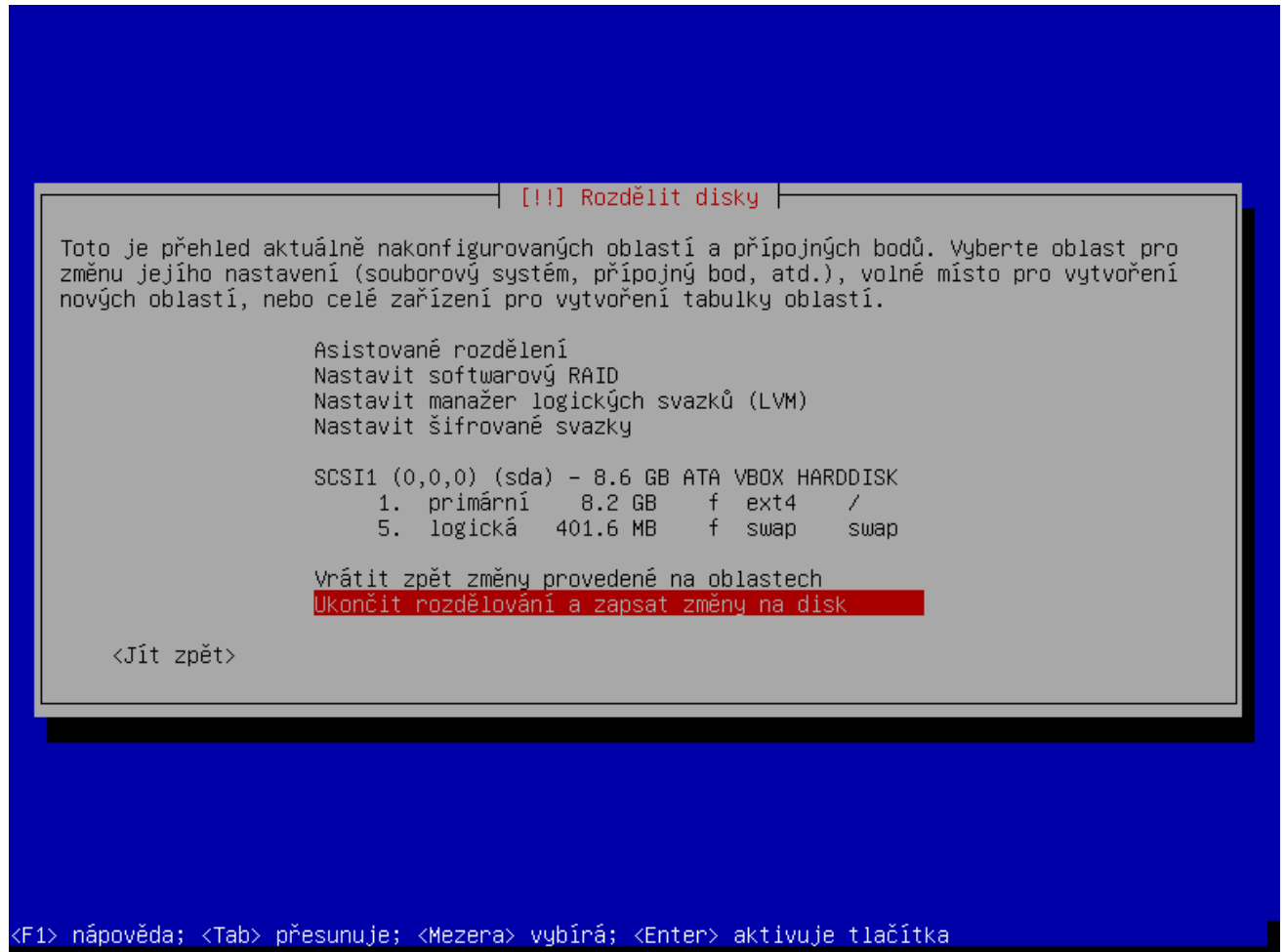


Rozdělení disku

Instalátor umožňuje poměrně detailně spravovat během instalace diskové oblasti.

Problematika diskových oblastí bude rezebírána v dalších cvičeních. Zvolme tedy asistované rozdělení, kde použijeme jediný disk, který je v počítači.

Výsledek bude vypadat nějak takto:



```
[!!!] Rozdělit disky

Toto je přehled aktuálně nakonfigurovaných oblastí a přípojných bodů. Vyberte oblast pro
změnu jejího nastavení (souborový systém, přípojný bod, atd.), volné místo pro vytvoření
nových oblastí, nebo celé zařízení pro vytvoření tabulky oblastí.

Asistované rozdělení
Nastavit softwarový RAID
Nastavit manažer logických svazků (LVM)
Nastavit šifrované svazky

SCSI1 (0,0,0) (sda) - 8.6 GB ATA VBOX HARDDISK
  1. primární   8.2 GB   f  ext4   /
  5. logická   401.6 MB  f  swap   swap

Vrátit zpět změny provedené na oblastech
Ukončit rozdělování a zapsat změny na disk

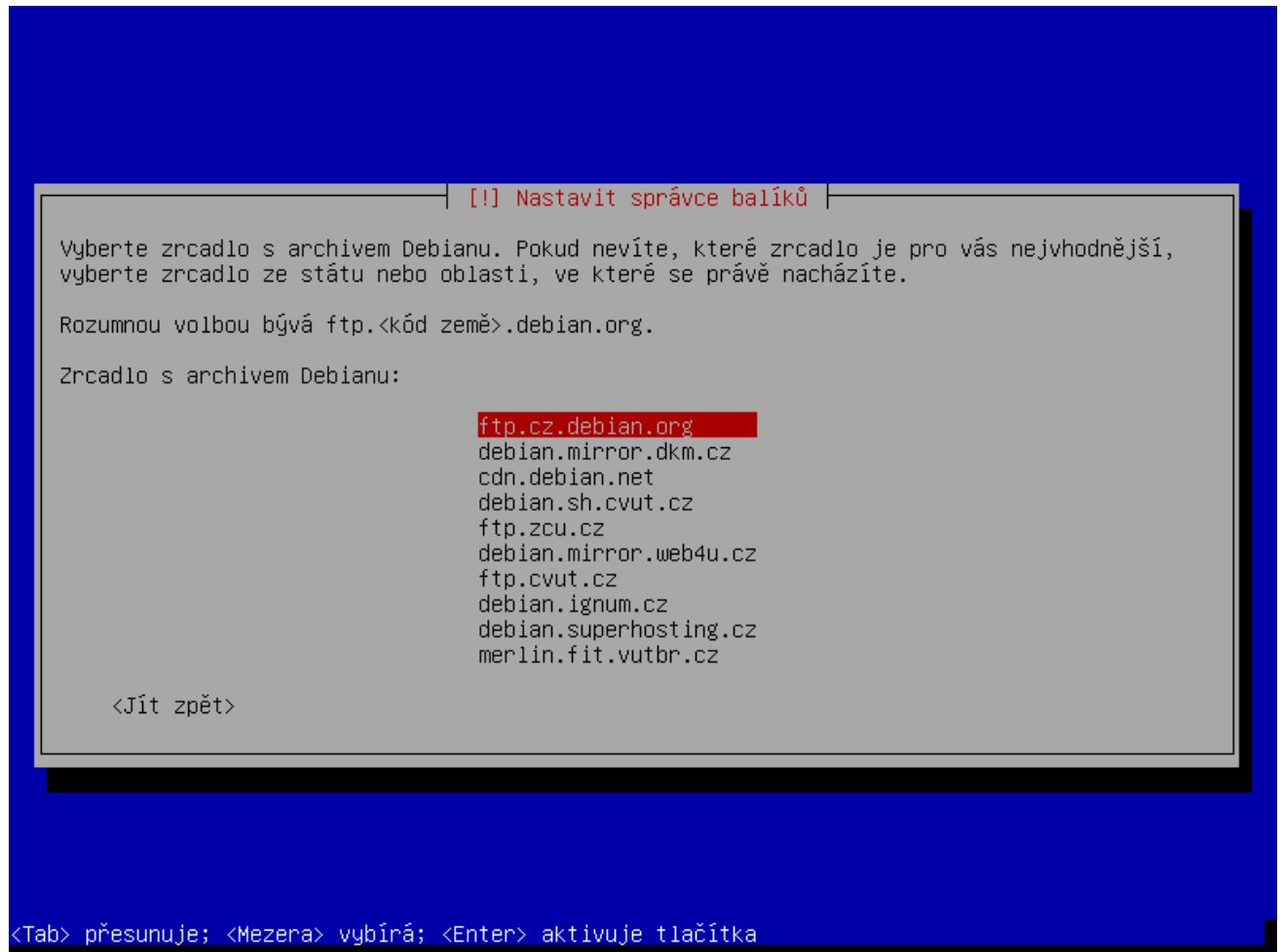
<Jít zpět>

<F1> nápověda; <Tab> přesunuje; <Mezera> vybírá; <Enter> aktivuje tlačítka
```

Síťové zrcadlo

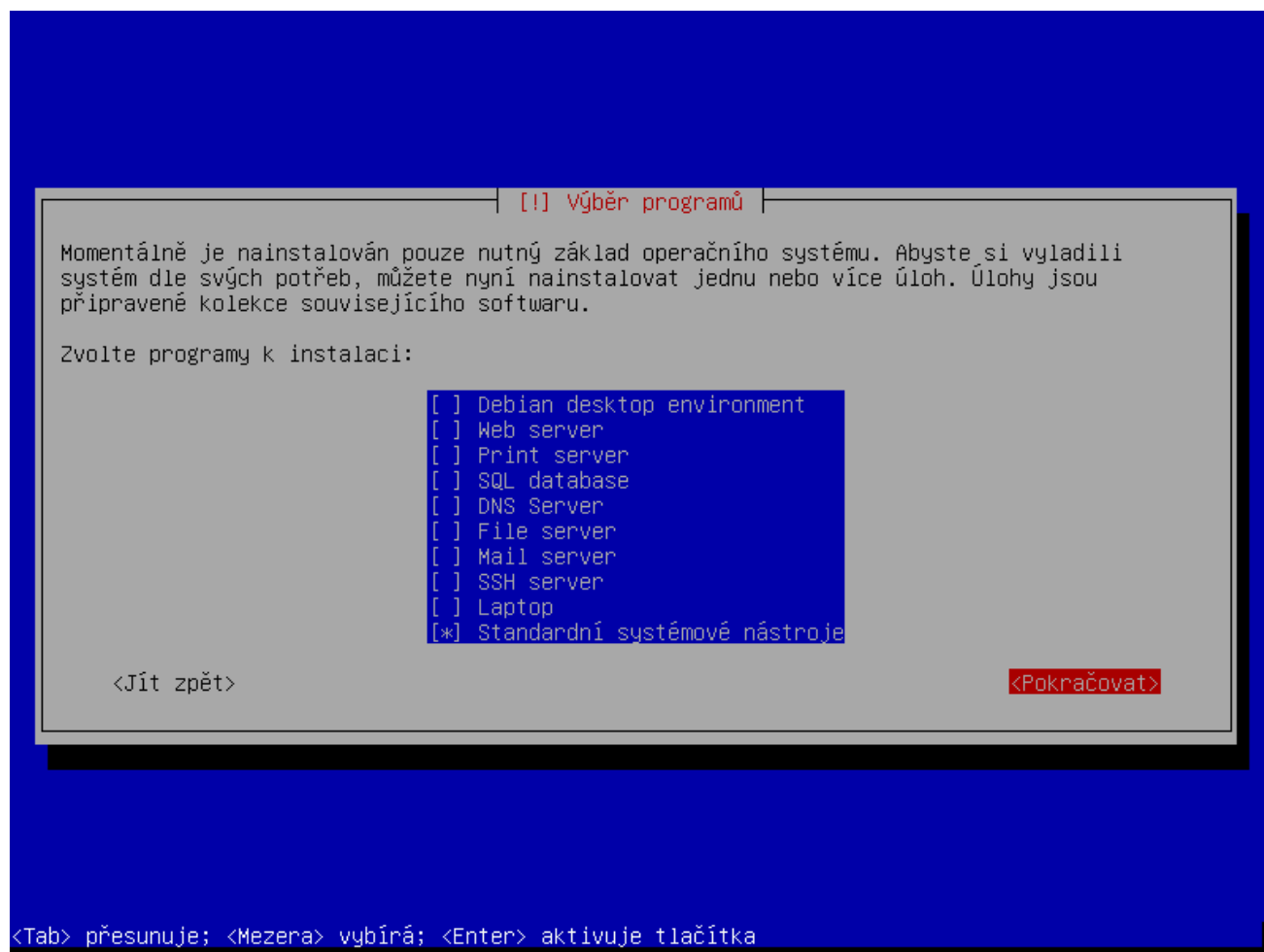
Pokud neinstalujeme offline verzi, pak rozhodně musíme nastavit síťové zrcadlo (repozitář). Jde vlastně o úložiště balíčků (software), ze kterého se bude instalovat veškerý software a aktualizace. (V systému může být repozitářů i několik)

Zpravidla volíme takové zrcadlo, které je nám nejbližší (geograficky). Jednak kvůli vytížení linek, a tím i kvůli rychlosti. Proč stahovat balíčky z Číny, když můžu ze sousedního Německa nebo přímo z Čech?



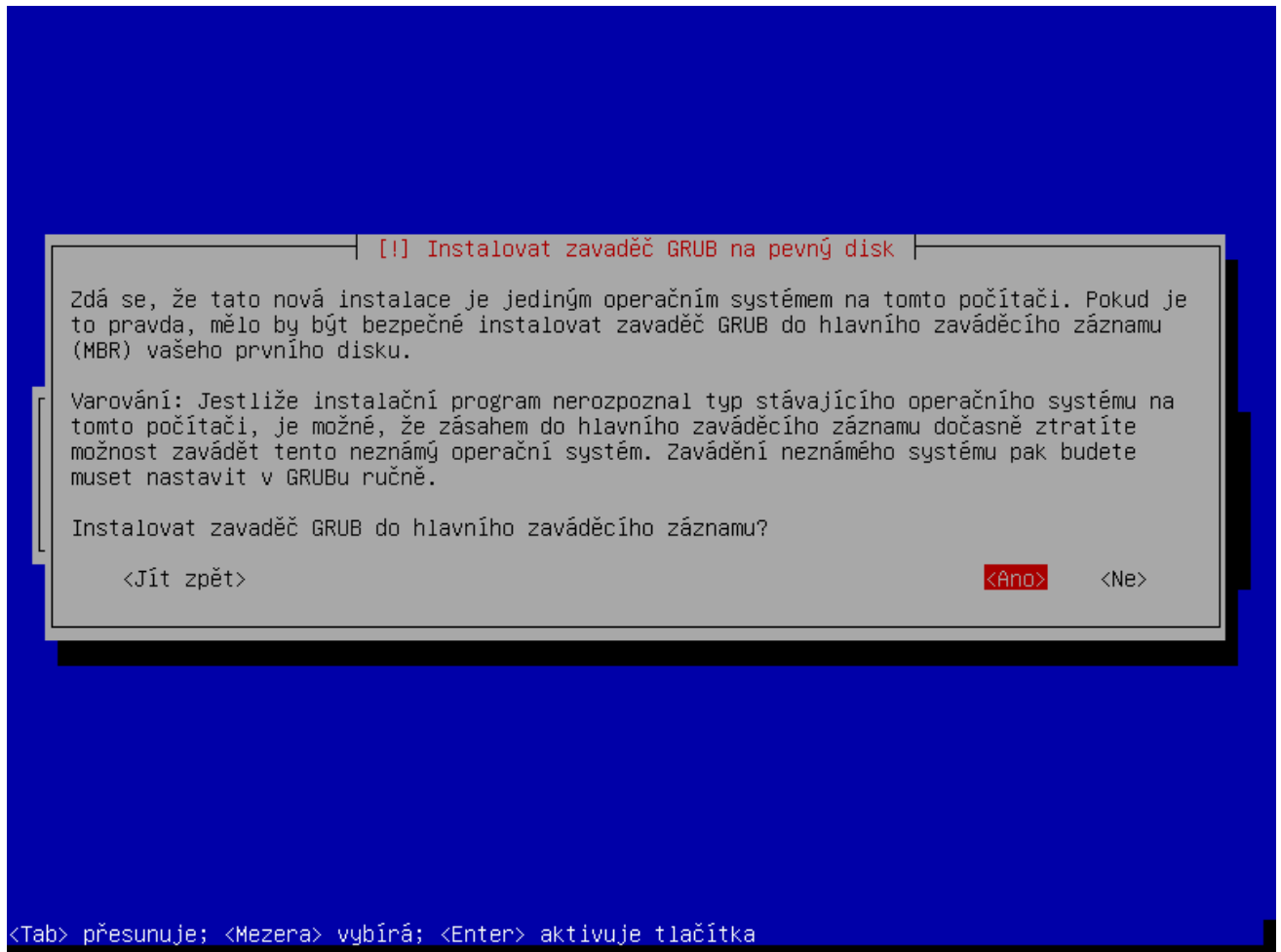
Úlohy

Instalátor umožňuje dopředu nainstalovat balíčky týkající se různých úloh, který může náš systém zastávat. Prozatím pomocí mezerníku odznačíme vše až na "Standardní systémové nástroje". Uvedené úlohy lze samozřejmě doinstalovat i potom.



Grub

Pro začátečníky je značně matoucí poslední instalační otázka typu ano/ne.



Odpověď je zde "ano" Bez zavaděče by systém nemohl tzv. nabootovat. Zavaděč (zde GRUB) je program, který spouští BIOS ihned po provedení všech POST testů z pevného disku (nebo jiného média).

Příkazový shell

Název školy	Vyšší odborná škola obalové techniky a Střední škola, Štětí, příspěvková organizace		
Adresa školky	Kostelní 134, 411 08 Štětí		
IČ	46773509		
Název operačního programu	OP Vzdělávání pro konkurenceschopnost		
Registrační číslo	CZ.1.07/1.5.00/34.1006		
Označení vzdělávacího materiálu	VY_32_INOVACE_21_PSS_404		
Název tématické oblasti (sady)	Operační systémy		
Název materiálu	Příkazový shell		
Anotace	Úvod do pracovního prostředí příkazového shellu v systému Linux. Přihlášení, orientace v příkazové řádce, zadávání příkazů a využívání některých funkcí příkazového shellu (historie, doplňování příkazů).		
Autor	Ivan Pomykacz	Jazyk	český
Očekávaný výstup	Přihlásí se, odhlásí se do/ze systému. Používá více uživatelských relací. Zadává příkazy. Využívá funkce shellu - automatické doplňování příkazů a historii zadaných příkazů. Vypne počítač.		
Klíčová slova	přihlášení, příkazový shell, zadávání příkazů, historie příkazů		
Druh výukového zdroje	Výklad	Věková skupina žáků	17+
Typ interakce	aktivita	Ročník	3.
Speciální vzdělávací potřeby	žádné		
Zhotoveno, (datum/období)	24.09.2013	Celková velikost	

Obsah

- Login
 - Více uživatelů
 - Přihlášení
- Zadávání příkazů
 - Klávesa Tab
- Ukončení práce
 - Odhlášení
 - Vypnutí systému
- Historie příkazů

Login

```
Debian GNU/Linux 7 wheezy tty1
wheezy login: _
```

Uvítací obrazovka nevypadá nijak vábně. Kolik času ovšem trávíte na přihlašovací obrazovce? Nejvíc asi ti, co zapomenou přihlašovací jméno nebo heslo ještě před skončením instalace.

První věc, kterou je potřeba si uvědomit, že linux bývá odborníky považován nikoli za to jak vypadá, ale jak funguje. I když to, jak váš linuxový systém bude vypadat, záleží hlavně na vás.



Více uživatelů

Pomocí kláves **alt+F1** až **alt+F6** se lze přepínat mezi tzv. virtuálními terminály (tty1-tty6). Každý z terminálů lze použít pro novou relaci jiného nebo i téhož uživatele.

Např. v jednom terminálu jste jako běžný uživatel *tux* a v druhém terminálu jako *root*.

Přihlášení

Pozor na velká a malá písmena. V linuxových systémech hrají velkou roli, a to nejen u jmen uživatelů, ale i názvů souborů.

Při psaní hesla se vám nebudou zobrazovat hvězdičky nebo puntíky jako v jiných systémech. Není to chyba, ale bezpečnostní prvek. Počet hvězdiček odhalí počet znaků ve vašem hesle, což značně usnadní útok hrubou silou.

```
Debian GNU/Linux 7 wheezy tty1
wheezy login: tux
Password:
Linux wheezy 3.2.0-4-486 #1 Debian 3.2.54-2 i686

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.
tux@wheezy:~$ _
```

Po přihlášení se spustí tzv. příkazový shell (bash).

```
tux@wheezy:~$
```

Shell zobrazuje běžnému uživateli jeho přihlašovací jméno, zde *tux*. Za oddělovačem `@` následuje název počítače, zde *wheezy*. Za dalším oddělovačem `:` se nachází pracovní adresář, tj. adresář, kde se uživatel aktuálně nachází. Momentálně je zde zástupný symbol `~` pro tzv. domovský adresář. Tilda, jak se nazývá symbol `~` se dá s výhodou použít pro označení domovského adresáře libovolného uživatele, musíte být ovšem jako tento uživatel přihlášení. Nakonec symbol `$` ukončuje informační část shellu.

Vše lze pochopitelně přenastavit, přeuspořádat, případně zcela zrušit.

Zadávání příkazů

Do příkazového shellu lze zadávat příkazy, které mohou představovat názvy spustitelných programů a nebo přímo programátorské konstrukce (proměnné, podmínky, cykly), ale pro tyto účely je možná vhodnější použít skript.

Napíšete-li text `clear` a potvrdíte klávesou Enter, vykoná se příkaz shellu, který jakoby "vyčistí" obrazovku.

Nyní, napíšete-li do shellu textu `date`, potvrdíte klávesou Enter, vypíše se aktuální datum a čas. Tentokrát jste ovšem spustili program (jež se nachází v `\bin\date`). Neřešme nyní, co je příkaz shellu, a co program. Obojí vykonává definovanou akci a to je vše.

```
tux@wheezy:~$ date
Po dub 21 20:24:49 CEST 2014
tux@wheezy:~$ _
```

Klávesa Tab

Psaní příkazů nesmírně usnadní klávesa **Tab**. V podstatě jde o automatické doplnění (dopsání) rozepsaných příkazů.

Napišete-li písmeno **d** a stisknete **Tab**, nic se nestane. Stisknete-li **Tab** podruhé, vypíší se všechny příkazy začínající na písmeno **d**.

```
tux@wheezy:~$ d
dash                debconf-communicate  discover-config
date                debconf-copydb       disown
db5.1_archive       debconf-escape       dmesg
db5.1_deadlock      debconf-set-selections dnsdomainname
db5.1_dump          debconf-show         do
db5.1_hotbackup     declare              domainname
db5.1_checkpoint    delpart              done
db5.1_load          dequote              dotlockfile
db5.1_log_verify    df                    dpkg
db5.1_printlog      dh_bash-completion  dpkg-deb
db5.1_recover       dh_dkms              dpkg-divert
db5.1_replicate     dh_installx catalogs dpkg-maintscript-helper
db5.1_stat          dh_pysupport         dpkg-query
db5.1_upgrade       dh_python2           dpkg-split
db5.1_verify        diff                  dpkg-statoverride
dc                  diff3                 dpkg-trigger
dd                  dig                    dprofpp
ddate               dir                    du
deallocvt           dircolors             dumpkeys
debconf             dirname
debconf-apt-progress dirs
tux@wheezy:~$ d_
```

Dopíšeme-li písmeno **a** a stiskneme opět 2x Tab, vypíší se už jen dvě možnosti (dash a date).

Pokud byste napsali nejprve **dat** a poté stiskli **1x** Tab, doplní se zbytek příkazu, protože již existuje právě jedna možnost. Ano, u příkazu, jako je **date** nemá moc smysl používat klávesu Tab pro

doplnění posledního písmenka. Jsou tu ovšem jiné, delší názvy příkazů a nebo názvy souborů/adresářů. A tam už se klávesa Tab osvědčí mnohem víc.

Vyzkoušejte klávesu **Tab** na:

- pw
- cl
- upt
- sle (použití např. `sleep 3s`)
- whe (použití např. `whereis date`)
- whoa
- apti

Ukončení práce

Odhlášení

Příkazem `logout` nebo `exit` ukončíte práci v shellu. Do souboru `~/.bash_history` se uloží veškeré příkazy, které jste do shellu zadali. Nakonec se spustí znovu přihlašovací obrazovka.

Vypnutí systému

Vypnutí systému lze docílit několika způsoby.

- Klávesová kombinace `Ctrl+Alt+Del` vyvolá automaticky restart počítače. (Ve virtuálním stroji bude nejspíš tato kombinace namapována na nějakou klávesou zkratku, aby nedošlo ke konfliktu z hostitelským strojem. VirtualBox používá klávesu HostKey+Del)
- Fyzickým tlačítkem pro zapnutí/vypnutí počítače. Tedy alespoň v případě, že je v BIOSu počítače nastavená událost Soft-Off. Jde vlastně o událost, kterou je schopen zachytit operační systém a ten se začne automaticky ukončovat. V některých případech to může celkem násilně odstříhnout přihlášené uživatele. Pokud na počítači nepracujete sami, tak rozhodně toto není doporučený způsob jeho vypnutí.
- Příkazem `halt` nebo `shutdown`. K tomu je ovšem zapotřebí být přihlášen jako uživatel `root`.

Historie příkazů

Kurzurovými klávesami *nahoru* a *dolu* lze procházet historii spuštěných příkazů.

Historie příkazů se uchovává v souboru `~/.bash_history`. Pozor, použité příkazy z aktuálního sezení se uloží až po té, co se uživatel odhlásí.

Pokud nechcete, aby se příkaz zapsal do historie, napište před něj mezeru.

Základní příkazy

Název školy	Vyšší odborná škola obalové techniky a Střední škola, Štětí, příspěvková organizace		
Adresa školky	Kostelní 134, 411 08 Štětí		
IČ	46773509		
Název operačního programu	OP Vzdělávání pro konkurenceschopnost		
Registrační číslo	CZ.1.07/1.5.00/34.1006		
Označení vzdělávacího materiálu	VY_32_INOVACE_21_PSS_405		
Název tematické oblasti (sady)	Operační systémy		
Název materiálu	Základní příkazy		
Anotace	Představení základní příkazů pro práci s manuálovými stránkami a pohybu v adresářové struktuře. Popis automatického doplňování názvů souborů/adresářů. Výpis obsahu adresářů a práce se skrytými soubory.		
Autor	Ivan Pomykacz	Jazyk	český
Očekávaný výstup	Orientuje se v manuálových stránkách programů. Pohybuje se v adresářové struktuře příkazem cd. Rozlišuje relativní a absolutní adresy. Používá funkce shellu, jako automatické doplnění názvu souboru/adresáře. Vypisuje obsah adresáře včetně skrytých souborů.		
Klíčová slova	manuálové stránky, man, cd, ls, skrytý soubor, výpis adresáře		
Druh výukového zdroje	Výklad	Věková skupina žáků	17+
Typ interakce	aktivita	Ročník	3.
Speciální vzdělávací potřeby	žádné		
Zhotoveno, (datum/období)	1.10.2013	Celková velikost	

Obsah

- Manuálové stránky
 - NAME
 - SYNOPSIS
 - DESCRIPTION
 - OPTIONS
 - USAGE
- Pohyb v adresářové struktuře
 - Aktuální pozice
 - Změna adresáře
 - Automatické doplňování
- Obsah adresářů
 - Skryté soubory
 - Speciální adresáře `.` a `..`
 - Výpis obsahu adresáře
 - Detailní výpis
 - Kombinovaný výpis

Manuálové stránky

Jako vůbec první příkaz je vhodné představit příkaz `man`.

```
tux@wheezy:~$ man
Kterou manuálovou stránku si přejete?
tux@wheezy:~$
```

Příkaz `man` slouží pro přístup a prohlížení manuálových stránek nainstalovaných programů a příkazů. Uvedeme-li název programu (zpravidla je názvem programu slovo, kterým se program spouští), zobrazí se manuálová stránka právě tohoto programu (existuje-li).

```
tux@wheezy:~$ man man
```

Uvedený příkaz není chybně napsaný dvakrát. Jde o výpis manuálové stránky k programu `man`.

```
MAN(1)                                Manual pager utils
MAN(1)

NAME
    man - an interface to the on-line reference manuals

SYNOPSIS
    man [-C file] [-d] [-D] [--warnings[=warnings]] [-R encoding] [-L
locale] [-m system[,...]]
        [-M path] [-S list] [-e extension] [-i|-I] [--regex|--wildcard] [-
-names-only] [-a] [-u]
        [--no-subpages] [-P pager] [-r prompt] [-7] [-E encoding] [---
o-hyphenation] [--no-justifica-
```

```

tion] [-p string] [-t] [-T[device]] [-H[browser]] [-X[dpi]] [-Z]
[[section] page ...] ...
man -k [apropos options] regexp ...
man -K [-w|-W] [-S list] [-i|-I] [--regex] [section] term ...
man -f [whatis options] page ...
man -l [-C file] [-d] [-D] [--warnings[=warnings]] [-R encoding] [-L
locale] [-P pager] [-r
prompt] [-7] [-E encoding] [-p string] [-t] [-T[device]] [-H[browser]]
[-X[dpi]] [-Z] file ...
man -w|-W [-C file] [-d] [-D] page ...
man -c [-C file] [-d] [-D] page ...
man [-hV]

```

DESCRIPTION

man is the system's manual pager. Each page argument given to man is normally the name of a program, utility or function. The manual page associated with each of these arguments is then found and displayed. A section, if provided, will direct man to look only in that section of the manual. The default action is to search in all of the available sections, following a pre-defined order and to show only the first page found, even if page exists in several sections.

The table below shows the section numbers of the manual followed by the types of pages they contain.

```

1 Executable programs or shell commands
2 System calls (functions provided by the kernel)
3 Library calls (functions within program libraries)
4 Special files (usually found in /dev)
5 File formats and conventions eg /etc/passwd
6 Games
7 Miscellaneous (including macro packages and conventions), e.g.
man(7), groff(7)
8 System administration commands (usually only for root)
9 Kernel routines [Non standard]

```

A manual page consists of several sections.

Kurzorové klávesy umožní pohyb nahoru/dolů (případně klávesy mezerník nebo Page Up / Page Down. Klávesa **q** ukončí prohlížení manuálových stránek.

Klávesou **/** lze vyvolat vyhledávání ve stránce. Funguje tak, že stisknete klávesu **/** a začnete psát hledaný řetězec. Stiskem klávesy **Enter** se vyhledá jeho nejbližší výskyt (Ctrl+C akci zrušíte). Vyvoláte-li **/** podruhé, a stisknete ihned klávesu **Enter**, zopakuje se hledání posledního řetězce.

Manuálová stránka se skládá z několika částí / sekcí.

NAME

Obsahuje pouze název a krátký popis programu.

SYNOPSIS

Tato sekce shrnuje možné varianty a přepínače pro spuštění programu. Zpravidla platí tato notace:

tučný text .. napište přesně jak uvedeno
text kurzívou .. dosadte vhodný argument
[-abc] .. žádný nebo jakýkoli argument je volitelný
-a|-b .. je třeba zvolit právě jednu ze dvou voleb
argument .. argument je možné zopakovat
[výraz] .. celý výraz v závorkách lze zopakovat

DESCRIPTION

Obsahuje podrobný popis programu.

OPTIONS

V této sekci je uveden výpis a popis všech voleb (parametrů) a argumentů, které lze použít při spuštění programu.

USAGE

Na konci manuálových stránek bývá (nemusí) ukázka toho, jak lze program používat.

Pohyb v adresářové struktuře

Aktuální pozice

Naše aktuální pozice je už vypsána v levé, informační, části shellu.

```
tux@wheezy:~$
```

Tilda `~` je symbol pro domovský adresář právě přihlášeného uživatele. Tzn. použijete-li někdy symbol `~`, pak za vás shell dosadí absolutní cestu k domovskému adresáři uživatele, pod kterým jste právě teď přihlášení.

Existuje příkaz `pwd`, který vypíše (vrátí) aktuální pozici.

```
tux@wheezy:~$ pwd  
/home/tux  
tux@wheezy:~$
```

Změna adresáře

Příkazem `cd` (nejde o program v pravém smyslu slova) lze měnit adresář, ve kterém se právě nacházíme.

```
tux@wheezy:~$ cd ..
tux@wheezy:/home$
```

Parametr `..` znamená přesun o úroveň výše. Tzn. že pokud jsme byli v adresáři `/home/tux`, pak příkaz `cd ..` nás přesunul do adresáře `/home`. Což je i vidět, protože shell změnil náš aktuální adresář z `~` na `/home`.

Návrat zpět do domovského adresáře můžeme provést např. takto:

```
tux@wheezy:/home$ cd tux
tux@wheezy:~$
```

Všimněte si tildy, jsme opět ve svém domovském adresáři.

Pohyb mezi adresáři, který jsme právě teď učinili byl tzv. **relativní**. Pohybovali jsme se totiž relativně k naší pozici.

Opakem relativního pohybu je **absolutní**. Abychom mohli změnit adresář pomocí absolutního pohybu, musíme zadat absolutní cestu k adresáři, kam se chceme dostat.

```
tux@wheezy:~$ cd /home
tux@wheezy:/home$
```

V adresářové struktuře systému Linux začíná vše u kořene (aka rootu) `/`. Chceme-li tedy zadávat cestu absolutně, musíme začít vždy od kořene.

Pokud bychom chtěli zpět do svého domovského adresáře, ale pomocí absolutní cesty, jsou dvě možnosti:

```
tux@wheezy:/home$ cd /home/tux/
tux@wheezy:~$
```

nebo

```
tux@wheezy:/home$ cd ~
tux@wheezy:~$
```

S domovským adresářem existuje ještě jeden trik, zadáte-li pouze `cd`, ať jste, kde jste, tak vrátíte do svého dom. adresáře.

Automatické doplňování

Vzpomínáte na klávesu `Tab`, ta která doplňuje názvy příkazů? Lze ji s výhodou použít i při

doplňování názvu adresářů/souborů.

Platí stejná pravidla jako před tím, tj. existuje-li ze zadaných písmen už jen jedna možnost, název se doplní. Pokud existuje více variant, je třeba stisknout **Tab** ještě jednou a tyto varianty se zobrazí. Pokud ovšem zadávaný adresář/soubor neexistuje, pak se při opakovaném stisku klávesy **Tab** nic nestane.

Srovnajte, jak rychle vám zabere napsat příkaz níže s a bez použití klávesy **Tab**.

```
tux@wheezy:~$ cd /usr/local/share/xml/misc/
tux@wheezy:/usr/local/share/xml/misc$
```

Časem zjistíte (budete si pamatovat), kdy je třeba zadat kolik písmenek, aby vám klávesa **Tab** doplnila název na jedno stisknutí.

Obsah adresářů

Příkazem **ls** lze nahlížet do adresářů a zjistit tak, jaké soubory a podadresáře se v nich nachází, kdy a kým byly vytvořeny, jaká nesou oprávnění atp.

Podíváme-li se do manuálové stránky **man ls** zjistíme, že

```
LS(1)                                User Commands
LS(1)

NAME
    ls - list directory contents

SYNOPSIS
    ls [OPTION]... [FILE]...

DESCRIPTION
    List information about the FILES (the current directory by default).
    Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.
```

Příkaz **ls** lze spustit bez parametru, protože všechny parametry jsou volitelné (jsou v hranatých závorkách).

Spuštěním příkazu **ls** v našem domovském adresáři se jakoby nic nestane.

```
tux@wheezy:~$ ls
tux@wheezy:~$
```

Je to proto, že zde "nic" není. Respektive nic není na první pohled vidět. Základní tvar programu **ls** nezobrazuje totiž skryté soubory/adresáře.

Skryté soubory

Program `ls` umí ovšem zobrazit i takové soubory. Podíváme-li se ovšem do manuálových stránek `man ls`, nic ohledně "skrytých" souborů/adresářů se nedočteme.

V linuxu totiž nejsou skryté soubory v právním smyslu slova. Je ovšem možné vytvořit soubory/adresáře, které mají na začátku názvu tečku. Např. `.profile`. Takto pojmenované soubory/adresáře se pak v běžném výpisu nezobrazují, jsou tedy jakoby skryté.

Zadáme-li následující příkaz, zobrazí se nám všechny položky, tedy i ty, které začínají na `.`.

```
tux@wheezy:~$ ls -a
.  ..  .bash_history  .bash_logout  .bashrc  .lessht  .profile
tux@wheezy:~$
```

Speciální adresáře `.` a `..`

Ve výpisu se na prvních dvou pozicích objevují zvláštní položky: `.` a `..`. Jde o speciální adresáře nebo spíš odkazy, kde `.` je odkaz na aktuální adresář a `..` je odkaz na adresář o úroveň výše.

Vzpomeňte na `cd ..`, to je přesně ta situace, kde využijete `..`. Je samozřejmě možné napsat příkaz ve stylu `cd ../..`.

Výpis obsahu adresáře

Chceme-li vypsát obsah jiného adresáře než toho, ve kterém se nacházíme, stačí uvést na konec příkazu `ls` cestu (ať relativní nebo absolutní) k tomuto adresáři.

```
tux@wheezy:~$ ls /home/
tux
tux@wheezy:~$ ls /
bin  dev  home      lib          media  opt   root  sbin   srv  tmp  var
boot etc  initrd.img lost+found  mnt    proc  run   selinux sys  usr
vmlinuz
tux@wheezy:~$
```

Detailní výpis

Pro detailnější pohled je zde přepínač (parametr) `-l`

```
tux@wheezy:~$ ls -l /home/
celkem 4
drwxr-xr-x 2 tux tux 4096 kvě 17 10:08 tux
tux@wheezy:~$
```

V adresáři `/home` se nachází pouze jeden podadresář se jménem `tux`.

Prozatím neřešme, co všechno lze z výpisu vyčíst. Snad jen, že je-li první písmenko ve výpisu položek

d, pak se jedná o adresář. Je-li na první pozici pomlčka, pak jde o obyčejný soubor (jsou totiž i soubory neobyčejné).

```
tux@wheezy:~$ ls -l /
celkem 76
drwxr-xr-x  2 root root  4096 dub 21 18:55 bin
drwxr-xr-x  3 root root  4096 dub 21 18:56 boot
drwxr-xr-x 13 root root 3020 kvě 17 09:12 dev
drwxr-xr-x 75 root root  4096 kvě 17 10:50 etc
drwxr-xr-x  3 root root  4096 dub 21 18:57 home
lrwxrwxrwx  1 root root    28 dub 21 18:48 initrd.img -> /boot/initrd.img-
3.2.0-4-486
drwxr-xr-x 14 root root  4096 dub 21 18:51 lib
drwx----- 2 root root 16384 dub 21 18:47 lost+found
drwxr-xr-x  3 root root  4096 dub 21 18:47 media
drwxr-xr-x  2 root root  4096 úno  3 10:24 mnt
drwxr-xr-x  2 root root  4096 dub 21 18:47 opt
dr-xr-xr-x 70 root root    0 kvě 17 09:12 proc
drwx----- 3 root root  4096 kvě 17 09:19 root
drwxr-xr-x 12 root root   580 kvě 17 09:17 run
drwxr-xr-x  2 root root  4096 dub 21 18:59 sbin
drwxr-xr-x  2 root root  4096 čen 10 2012 selinux
drwxr-xr-x  2 root root  4096 dub 21 18:47 srv
drwxr-xr-x 12 root root    0 kvě 17 09:12 sys
drwxrwxrwt  4 root root  4096 kvě 17 10:51 tmp
drwxr-xr-x 10 root root  4096 dub 21 18:47 usr
drwxr-xr-x 11 root root  4096 dub 21 18:47 var
lrwxrwxrwx  1 root root    24 dub 21 18:48 vmlinuz -> boot/vmlinuz-3.2.0-
4-486
tux@wheezy:~$
```

Kombinovaný výpis

Parametry při spouštění příkazu je možné i kombinovat (zejména u programu **ls**). Můžeme si tak vynutit detailní výpis, včetně skrytých souborů

```
tux@wheezy:~$ ls -l -a
celkem 28
drwxr-xr-x 2 tux  tux  4096 kvě 17 10:08 .
drwxr-xr-x 3 root root 4096 dub 21 18:57 ..
-rw----- 1 tux  tux   111 dub 21 21:38 .bash_history
-rw-r--r-- 1 tux  tux   220 dub 21 18:57 .bash_logout
-rw-r--r-- 1 tux  tux 3392 dub 21 18:57 .bashrc
-rw----- 1 tux  tux    46 kvě 17 10:08 .lesshst
-rw-r--r-- 1 tux  tux   675 dub 21 18:57 .profile
tux@wheezy:~$
```

Jaký adresář jsme to právě vypsalí?

Nastavení sítě

Název školy	Vyšší odborná škola obalové techniky a Střední škola, Štětí, příspěvková organizace		
Adresa školky	Kostelní 134, 411 08 Štětí		
IČ	46773509		
Název operačního programu	OP Vzdělávání pro konkurenceschopnost		
Registrační číslo	CZ.1.07/1.5.00/34.1006		
Označení vzdělávacího materiálu	VY_32_INOVACE_21_PSS_406		
Název tematické oblasti (sady)	Operační systémy		
Název materiálu	Nastavení sítě		
Anotace	Text popisuje konfiguraci síťových rozhraní v systému Debian GNU Linux. Popisuje základní typy rozhraní jako loopback a klasické rozhraní pro technologii ethernet. Pomocí příkazů popisuje, jak lze zjistit aktuální stav konfigurace a jak lze konfiguraci měnit i trvale.		
Autor	Ivan Pomykacz	Jazyk	český
Očekávaný výstup	Zjistí, jakou IP adresu má systém přidělenou. Změní IP adresu rozhraní. Konfiguruje rozhraní pro spolupráci s DHCP serverem. Zvládá základní diagnostiku nástrojem ping.		
Klíčová slova	sítě, ethernet, loopback, ping, MAC, ICMP, IP adresa		
Druh výukového zdroje	Výklad	Věková skupina žáků	17+
Typ interakce	aktivita	Ročník	3.
Speciální vzdělávací potřeby	žádné		
Zhotoveno, (datum/období)	8.10.2013	Celková velikost	

Obsah

- Aktuální stav
 - loopback
 - eth
 - MAC adresa
- Přidělení IP
 - Aktivace rozhraní
 - Deaktivace rozhraní
- Diagnostika
 - Chybné stavy
 - Network is unreachable
 - Destination Host Unreachable
- Uložení nastavení
 - Aktivace konfigurace
 - Možné problémy

Smyslem GNU Linuxu je být připojen do sítě. Je tak i vidět z nástrojů pro konfiguraci a diagnostiku připojení. Následující text je opět zaměřen na distribuci Debian Wheezy. Zmiňuji to především proto, že nastavení sítě se v některých distribucích značně rozchází.

Nicméně, všechny distribuce disponují nástrojem `ip`.

Aktuální stav

V jakém stavu se nachází vaše síťové rozhraní lze zjistit z výpisu příkazu `ip addr` (nebo zkráceně `ip a`):

```
root@wheezy:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen 1000
    link/ether 08:00:27:64:08:c4 brd ff:ff:ff:ff:ff:ff
```

Z výpisu je patrné, že jsou v systému dvě rozhraní: `lo` a `eth0`.

loopback

Neboli zpětná smyčka je speciální virtuální rozhraní, tzn. fyzicky neexistuje. Vzniklo jako potřeba pro aplikace typu client/server, kde obě části takové aplikace (jak client, tak server) běží na tom samém počítači.

V Linuxu ovšem existují ještě tzv. sockety, speciální soubory, přes které mohou aplikace typu client/server rovněž komunikovat, ale toto teď neřešme.

Co *loopback* nabízí je tedy síťové rozhraní, ke kterému má přístup pouze operační systém (a aplikace v něm běžící). I když má *loopback* svou IP adresu (IPv4 127.0.0.1, IPv6 ::1), je tato adresa

dosažitelná pouze z daného počítače. Každý počítač má totiž svůj vlastní loopback, ovšem s naprosto stejnou IP adresou.

Spuštěné aplikace typu server pak mohou naslouchat na daném rozhraní a čekat na příchozí spojení, následně spojení navázat a poté komunikovat třeba pomocí TCP/IP.

Rozhraní *loopback* je zpravidla nastaveno defaultně, tzn. je k dispozici ihned po instalaci, takže není třeba nic řešit. Toto již funguje.

eth

Rozhraní označená jako *ethX*, kde *X* je číslo větší nebo rovno 0, je už reprezentace fyzického rozhraní (konkrétně na bázi technologie ethernet). V našem případě je v systému dostupné jediné ethernetové rozhraní *eth0*. Protože bylo detekováno jako první a první v pořadí, nese číslo 0.

Toto rozhraní bude pro nás nejzajímavější, protože toto rozhraní nám umožní přístup do sítě nebo naopak přístup do našeho systému ze sítě.

Aktuální stav je ten, že rozhraní není aktivní. Ve výpisu lze vidět příznak *state DOWN*.

MAC adresa

Z výpisu `ip addr` lze vyčíst i MAC adresu rozhraní. Jde o hexadecimální číslo zapsané po oktetech oddělených dvojtečkou, zde:

```
link/ether: 08:00:27:64:08:c4
```

Přidělení IP

Aby počítač mohl komunikovat na lokální síti, stačí mu pouze IP adresa a příslušná maska. Tady ovšem pozor, jedno rozhraní může mít současně více IP adres. Možná bude vhodné, před nastavování rozhraní jej jednoduše vyčistit od všech adres příkazem

```
root@wheezy:~# ip addr flush dev eth0
```

Nyní přidělme rozhraní *eth0* adresu `192.168.12.32` s maskou 24 bitů (tedy 255.255.255.0).

```
root@wheezy:~# ip addr add 192.168.12.32/24 dev eth0
```

Přidělenou adresu můžeme ověřit opětovným zadáním `ip addr`.

```
root@wheezy:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
```



```
2: eth0: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast state DOWN qlen 1000
   link/ether 08:00:27:64:08:c4 brd ff:ff:ff:ff:ff:ff
   inet 192.168.12.32/24 scope global eth0
```

Aktivace rozhraní

Rozhraní má sice už svou IP adresu, ale z výpisu `ip addr` je patrné, že stále není aktivní. To lze ovšem změnit příkazem:

```
root@wheezy:~# ip link set up dev eth0
root@wheezy:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
   inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
   qlen 1000
   link/ether 08:00:27:64:08:c4 brd ff:ff:ff:ff:ff:ff
   inet 192.168.12.32/24 scope global eth0
   inet6 fe80::a00:27ff:fe64:8c4/64 scope link
       valid_lft forever preferred_lft forever
```

Při aktivaci rozhraní se mezi adresy přidala ještě navíc IPv6 lokální linková adresa `fe80::a00:27ff:fe64:8c4/64`.

Deaktivace rozhraní

Stejný příkaz jako u aktivace, jen slovíčko *up* zaměníme za *down*:

```
root@wheezy:~# ip link set down dev eth0
```

Diagnostika

Jak ověřit, že jsme připojeni do lokální sítě? K tomu lze využít nástroj `ping`. Můžeme provést ping z počítače na ten samý počítač.

```
root@wheezy:~# ping -c 3 192.168.12.32
PING 192.168.12.32 (192.168.12.32) 56(84) bytes of data:
64 bytes from 192.168.12.32: icmp_req=1 ttl=64 time=0.016 ms
64 bytes from 192.168.12.32: icmp_req=2 ttl=64 time=0.020 ms
64 bytes from 192.168.12.32: icmp_req=3 ttl=64 time=0.019 ms

--- 192.168.12.32 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.016/0.018/0.020/0.004 ms
```

Parametr `-c 3` vyšle pouze 3 ICMP žádosti. Bez omezení parametrem `-c` bychom pingovali ještě dnes (v takovém případě lze přerušit vykonávání ping klávesami `Ctrl+C`).

Chybné stavy

V podstatě bychom mohli docílit několika chybných stavů v závislosti na tom, kde jsme udělali chybu mi nebo zda-li je chyba v síti.

Network is unreachable

- Rozhraní není aktivní - aktivujte jej příkazem `ip link set up dev eth0`.
- Rozhraní eth0 nemá přidělenou žádnou adresu - přidejte ji `ip addr add 192.168.12.32/24 dev eth0`.
- Rozhraní eth0 je na jiné síti, nebo mi provádíme ping do jiné sítě - zkontrolujte první 3 oktety IP adresy.

Destination Host Unreachable

Tuto hlášku uvidíte zejména tehdy, budete-li provádět `ping` na vzdálený počítač nebo ze vzdáleného počítače. Tak nebo tak, počítač prostě není dosažitelný a nebo příchozí/odchozí žádosti blokuje firewall.

Pro testovací účely jsem naschvál provedl ping na jiný počítač (232), ale na stejné síti, protože předpokládám, že takovou adresu nikdo mít nebude.

```
root@wheezy:~# ping -c 3 192.168.12.232
PING 192.168.12.33 (192.168.12.33) 56(84) bytes of data.
From 192.168.12.32 icmp_seq=1 Destination Host Unreachable
From 192.168.12.32 icmp_seq=2 Destination Host Unreachable
From 192.168.12.32 icmp_seq=3 Destination Host Unreachable

--- 192.168.12.232 ping statistics ---
3 packets transmitted, 0 received, +3 errors, 100% packet loss, time 1999ms
pipe 3
```

Uložení nastavení

Přidělená IP adresa příkazem `ip` zůstane přidělená až do restartu počítače. Jakmile bychom tedy provedly vypnutí a zapnutí počítače, naše nastavení by zmizelo.

Abychom tomu zabránili, musíme někde uložit informaci o tom, jakou adresu si přejeme, aby naše rozhraní mělo přidělenou. Pro tyto účely lze využít konfigurační soubor `/etc/network/interfaces`. Soubor editujeme obyčejným textovým editorem, např. `nano`.

```
root@wheezy:~# nano /etc/network/interfaces
```

Obsah souboru je poněkud strohý (záleží, jak byla provedena instalace).

```
# 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
```

V mém případě je v souboru zmínka pouze o rozhraní *loopback*. Volba *auto* aktivuje rozhraní při startu (resp. během spouštění skriptu pro aktivaci síťových rozhraní).

Za volbou *auto* obvykle následuje blok, který deklaruje parametry dalšího rozhraní. Pro naše potřeby dopíšme následující řádky:

```
auto eth0
iface eth0 inet static
    address 192.168.12.32
    netmask 255.255.255.0
```

Nyní soubor uložíme a ukončíme editor `nano`.

Aktivace konfigurace

Pozn.: Pokud jste prováděli konfiguraci pomocí příkazu `ip`, je třeba ji zrušit (`ip addr flush dev eth0`).

Před aktivací nové konfigurace je nejprve potřeba deaktivovat tu stávající. Pro tyto účely se využívá služba `networking`.

```
root@wheezy:~# service networking stop
[ ok ] Deconfiguring network interfaces...done.
```

Následně lze provést aktivaci:

```
root@wheezy:~# service networking start
[ ok ] Configuring network interfaces...done.
```

Můžeme ověřit:

```
root@wheezy:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
   qlen 1000
    link/ether 08:00:27:64:08:c4 brd ff:ff:ff:ff:ff:ff
    inet 192.168.12.32/24 brd 192.168.12.255 scope global eth0
```

Kdyby jste nyní provedli restart počítače, tak se nastaví síťového rozhraní automaticky obnoví.

Možné problémy

Při aktivaci služby se vypíše hláška: *Failed to bring up eth0*.

```
root@wheezy:~# service networking start
[....] Configuring network interfaces...Missing required variable: address
Missing required configuration variables for interface eth0/inet.
Failed to bring up eth0.
done.
```

Z výpisu lze dále vyčíst něco ohledně chybějící proměnné *address*. Buď tato proměnná opravdu chybí a nebo je zkomolený její název (typická chyba je ve spellingu, kde se píše dvě "d" a dvě "s").

Po spuštění příkazu `service networking start` se zdá být vše OK, ale při výpisu `ip addr` adresa u *eth0* stále chybí. Nezapomněli jste dopsat do `/etc/network/interfaces` řádek `auto eth0`?

Správa uživatelských účtů

Název školy	Vyšší odborná škola obalové techniky a Střední škola, Štětí, příspěvková organizace		
Adresa školky	Kostelní 134, 411 08 Štětí		
IČ	46773509		
Název operačního programu	OP Vzdělávání pro konkurenceschopnost		
Registrační číslo	CZ.1.07/1.5.00/34.1006		
Označení vzdělávacího materiálu	VY_32_INOVACE_21_PSS_407		
Název tematické oblasti (sady)	Operační systémy		
Název materiálu	Správa uživatelských účtů		
Anotace	Text je zaměřen na popis a správu lokálních uživatelských účtů v systému Debian GNU Linux. Popisuje atributy uživatelů jako UID, GID, domovský adresář nebo příkazový shell. Popisuje koncept skupin (primární a dodatečné). Dále ukazuje vnitřní strukturu databáze uživatelských účtů. Praktická část je zaměřena na vytváření uživatelů a přepínání mezi účty v rámci jednoho sezení.		
Autor	Ivan Pomykacz	Jazyk	český
Očekávaný výstup	Vytvoří uživatelský účet. Rozumí uspořádání databáze uživatelských účtů. Nastaví heslo uživateli. Přihlásí se na uživatele v rámci jednoho sezení.		
Klíčová slova	uživatelé, uživatelský účet, heslo, skupiny, domovský adresář, shell		
Druh výukového zdroje	Výklad	Věková skupina žáků	17+
Typ interakce	aktivita	Ročník	3.
Speciální vzdělávací potřeby	žádné		
Zhotoveno, (datum/období)	22.10.2013	Celková velikost	

Obsah

- Lokální uživatelské účty
 - Skupiny
 - Lokální databáze
 - /etc/passwd
 - /etc/shadow
- Založení nového uživatele
 - Skupiny, domovský adresář, shell, ...
 - Nastavení hesla
 - Přepnutí se na uživatele
- Odstranění uživatele

V Debianu lze provozovat tzv. lokální účty (což bude náš případ), ale i jiné typy účtů. Např. takové, které jsou svázány s nějakou centrální autentizační autoritou (např. Active Directory)

Lokální uživatelské účty

Běžný uživatel má své přihlašovací jméno (login) a heslo, kterými se autentizuje. A každý "běžný" uživatel má svůj tzv. domovský adresář, kam si může ukládat svá data.

Co už na první pohled vidět není, je to, že každý uživatel má svůj číselný identifikátor **UID** a je členem tzv. primární skupiny **GID**. Příkazem `id` lze tyto informace zobrazit.

```
root@wheezy:~# id
uid=0(root) gid=0(root) skupiny=0(root)
```

V případě uživatele root jsou hodnoty UID a GID rovny 0. Pokud byste některému uživateli přidělili stejné UID, pak se i z něho stane super uživatel (jen bude mít jiné přihlašovací jméno).

Příkazem `id` se můžeme podívat i na jiného uživatele tím, že zadáme jeho *login* jako parametr:

```
root@wheezy:~# id tux
uid=1000(tux) gid=1000(tux)
skupiny=1000(tux),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev)
)
```

Skupiny

Jak už bylo řečeno, každý účet v systému je členem alespoň jedné, tzv. primární skupiny. Ve výpisu příkazu `id tux` si lze všimnout, že tento uživatel je navíc členem ještě dalších skupin. Členstvím ve skupinách lze dosáhnout zpravidla oprávnění k přístupu souborům nebo i HW zařízením.

Spoutěná aplikace má totiž taková práva, jako ten, kdo jí spouští.

Lokální databáze

Existují tři soubory, ve kterých jsou uloženy informace o lokálních účtech a skupinách. Jde o:

- `/etc/passwd` .. Obsahuje informace o uživatelských účtech (s výjimkou hesla).
- `/etc/shadow` .. Obsahuje hesla uživatelů v zašifrované podobě.
- `/etc/group` .. Obsahuje definované skupiny a členství uživatelů ve skupinách (avšak ne těch primárních)

Všechny tři soubory jsou obyčejné textové soubory, tudíž je lze editovat přímo (a tím vytvořit, upravit nebo smazat uživatele). To se ovšem nedoporučuje. Existuje celá řada nástrojů, grafické i konzolové, které tu jsou pro tyto účely.

Podívat se do nich ale můžeme.

`/etc/passwd`

```
root@wheezy:~# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
Debian-exim:x:101:104::/var/spool/exim4:/bin/false
statd:x:102:65534::/var/lib/nfs:/bin/false
tux:x:1000:1000:Tux,,,:/home/tux:/bin/bash
```

Není to příliš patrné, ale soubor je vlastně taková tabulka. Každý účet je popsán jedním řádkem v souboru. Na každém řádku jsou hodnoty oddělené dvojtečkou. Vezmeme-li to zleva, pak sloupce nesou informace o:

- přihlašovací jméno;
- heslo zastoupené znakem x, hesla se již do tohoto souboru neukládají;
- UID
- GID, název skupiny lze vyčíst ze souboru `/etc/group`;
- informace o uživateli, jako celé jméno, adresa, atp.;
- cesta k domovskému adresáři - neznamená to ovšem, že tento adresář existuje;
- příkazový shell - běžní uživatelé mají zpravidla nastavený `/bin/bash`.

/etc/shadow

```
root@wheezy:~# cat /etc/shadow
root:Wn0HjfK$qob/ecdPlDEeZPrTkVaUCiny1mKNcU0L5gNEIvCgnz9fzIZMI6X9RnsHTPRM/sgb
rgVWpJvQXY.gv4Bq1c41H/:16181:0:99999:7:::
daemon*:16181:0:99999:7:::
bin*:16181:0:99999:7:::
sys*:16181:0:99999:7:::
sync*:16181:0:99999:7:::
games*:16181:0:99999:7:::
man*:16181:0:99999:7:::
lp*:16181:0:99999:7:::
mail*:16181:0:99999:7:::
news*:16181:0:99999:7:::
uucp*:16181:0:99999:7:::
proxy*:16181:0:99999:7:::
www-data*:16181:0:99999:7:::
backup*:16181:0:99999:7:::
list*:16181:0:99999:7:::
irc*:16181:0:99999:7:::
gnats*:16181:0:99999:7:::
nobody*:16181:0:99999:7:::
libuuid!:16181:0:99999:7:::
Debian-exim!:16181:0:99999:7:::
statd*:16181:0:99999:7:::
tux:GvPuCr$ilXbBAk3Mtr.sZ7PJEhTW8xgFr0hJ.2ze0pVW0KXuEdAPMg9G1MstB0vosUPWL3CUj
jB3jQSJDE36160Pzg.S.:16181:0:99999:7:::
```

Na každém řádku je opět jeden uživatel a jeho heslo, nebo také ne, pokud jej nemá. V takovém případě je na pozici hesla **!**.

Další parametry souvisí např. s různými daty, jako kdy expiruje heslo, kdy bylo naposledy změněno, lhůta pro vypršení hesla, atp. (vizte [man shadow](#)).

Založení nového uživatele

Nový účet lze založit příkazem [useradd](#). Pokud jej spustíte jen tak, vypíše vám nápovědu (dokonce v češtině).

Z manuálové stránky lze vyčíst, že nejjzákladnější použití spočívá v zadání přihlašovacího jména jako parametr programu [useradd](#), tedy:

```
root@wheezy:~# useradd franta
```

Uvedeným příkazem došlo skutečně k vytvoření uživatele s přihlašovacím jménem *franta*. Můžeme se přesvědčit např. příkazem [id franta](#)

```
root@wheezy:~# id franta
```



```
uid=1001(franta) gid=1001(franta) skupiny=1001(franta)
```

Všimněte si, že se automaticky vytvořila skupina se stejným názvem jako je přihlašovací jméno a shodou okolností i stejným GID jako UID. Podíváme-li se na konec souboru `/etc/passwd`, zjistíme, jak se nastavily ostatní parametry uživatele.

```
root@wheezy:~# tail -n 1 /etc/passwd
franta:x:1001:1001:~/home/franta:/bin/sh
```

Uživatel má nastavený příkazový shell `/bin/sh` a domovský adresář v umístění `/home/franta`. Pokud bychom se ale podívali do adresáře `/home`, zjistíme, že adresář se nevytvořil.

```
root@wheezy:~# ls /home/
tux
```

V souboru `/etc/shadow` navíc zjistíme, že uživatel *franta* nemá heslo.

```
root@wheezy:~# tail -n 1 /etc/shadow
franta:!:16213:0:99999:7:::
```

Skupiny, domovský adresář, shell, ...

Založme nyní další účet, ale tentokrát si zvolíme vybrané parametry podle sebe. Chceme, aby nový uživatel:

- patřil do primární skupiny *studenti*;
- měl vytvořený domovský adresář;
- měl nastavený příkazový shell na `/bin/bash`;

Potom by mohl příkaz na vytvoření nového uživatele *pepa* vypadat:

```
root@wheezy:~# useradd -g studenti -m -s /bin/bash pepa
useradd: skupina „studenti“ neexistuje
```

Pokud na vás vyskočí hláška, že skupna *studenti* neexistuje, pak je ji jednoduše založíme a příkaz zopakujeme:

```
root@wheezy:~# groupadd studenti
root@wheezy:~# useradd -g studenti -m -s /bin/bash pepa
```

Nyní se můžeme přesvědčit, že

```
root@wheezy:~# ls /home/
pepa tux
```

domovský adresář uživatele *pepa* již existuje.

Nastavení hesla

Uživatel stále nemá heslo. Pozor, není možné použít přepínač `-p`, protože tento parametr přebírá heslo v již zašifrované podobě.

Heslo přidělíme uživateli např. příkazem `passwd`.

```
root@wheezy:~# passwd pepa
Zadejte nové UNIX heslo:
Opakujte nové UNIX heslo:
passwd: heslo bylo úspěšně změněno
```

Pokud spustíte `passwd` bez parametru, pak měníte heslo sobě!

Přepnutí se na uživatele

V konzoli je možné pomocí příkazu `su` se přepnout na libovolného uživatele. Jste-li navíc přihlášení jako `root`, pak dokonce nemusíte znát ani heslo uživatele (nebo tento uživatel nemusí mít heslo ani vytvořeno).

```
root@wheezy:~# su pepa
pepa@wheezy:/root$ cd
pepa@wheezy:~$ pwd
/home/pepa
```

Nejprve jsem se přepnul na uživatele `pepa`, potom jsem příkazem `cd` přešel do svého (pepova) domovského adresáře, čehož je důkazem výstup příkazu `pwd`.

Příkazem `exit` se můžu vrátit zpět k účtu, pod kterým jsem byl přihlášen před tím, než jsem zadal příkaz `su`.

```
pepa@wheezy:~$ exit
exit
```

Uvádím to zde proto, že je možné opakovaně zadávat `su`, a tím se jakoby zanořovat do pomyslné hloubky. Pokud zapomenete na to, že jste zanořeni, může se vám stát že některé příkazy začnou hlásit podivné chybové hlášky (např. při odpojení disku: *Device or resource is busy*).

Pokud by se chtěl `pepa` povýšit na uživatele `root`, stačí zadat pouze `su`. Nyní zde záměrně uvádím poněkud nesmyslné zanořování příkazem `su` a následné vnořování.

```
root@wheezy:~# su pepa
pepa@wheezy:/root$ su
Heslo:
root@wheezy:~# su pepa
pepa@wheezy:/root$ su
Heslo:
root@wheezy:~# su
```

```
root@wheezy:~# su
root@wheezy:~# su pepa
pepa@wheezy:/root$ exit
exit
root@wheezy:~# exit
exit
root@wheezy:~# exit
exit
root@wheezy:~# exit
exit
pepa@wheezy:/root$ exit
exit
root@wheezy:~# exit
exit
pepa@wheezy:/root$ exit
exit
```

A ještě pro ukázkou, jak to vypadá s uživatelem *franta*, který má jiný příkazový shell a nemá domovský adresář.

```
root@wheezy:~# su franta
$ pwd
/root
$ cd
sh: 2: cd: can't cd to /home/franta
$ exit
```

Odstranění uživatele

Pro odstranění účtu ze systému se používá příkaz `userdel`. Jeho použití je intuitivní

```
root@wheezy:~# userdel franta
```

Pokud má uživatel domovský adresář a pokud jej chceme odstranit společně s uživatelským účtem, použijeme přepínač `-r`.

```
root@wheezy:~# userdel -r pepa
userdel: pepa mail spool (/var/mail/pepa) not found
```

Chybovou hlášku ohledně chybějícího adresáře `/var/mail/pepa` můžeme tiše přejít. Žádný takový adresář jsme totiž nevytvářeli, ale příkaz `userdel -r` se jej pokouší i přesto smazat.

Práva k souborům

Název školy	Vyšší odborná škola obalové techniky a Střední škola, Štětí, příspěvková organizace		
Adresa školky	Kostelní 134, 411 08 Štětí		
IČ	46773509		
Název operačního programu	OP Vzdělávání pro konkurenceschopnost		
Registrační číslo	CZ.1.07/1.5.00/34.1006		
Označení vzdělávacího materiálu	VY_32_INOVACE_21_PSS_408		
Název tematické oblasti (sady)	Operační systémy		
Název materiálu	Práva k souborům		
Anotace	Představení systému přístupových práv v Linuxových systémech. Popis funkcionality UGO – RWX. Praktická část je zaměřena na nastavování práv k souborům a adresářům s akcentem na mírné odlišnosti mezi souborem a adresářem.		
Autor	Ivan Pomykacz	Jazyk	český
Očekávaný výstup	Rozumí problematice přístupových práv. Odlišuje jednotlivé složky UGO. Interpretuje rozdíl RWX u souborů a adresářů. Aplikuje nový mód na soubor dle zadání/situace za účelem odepření nebo nabytí přístupu ke zdroji.		
Klíčová slova	UGO, vlastník, skupina, ostatní, RWX, čtení, zápis, spuštění, mód souboru		
Druh výukového zdroje	Výklad	Věková skupina žáků	17+
Typ interakce	aktivita	Ročník	3.
Speciální vzdělávací potřeby	žádné		
Zhotoveno, (datum/období)	26.11.2013	Celková velikost	

Obsah

- UGO
- RWX
 - Soubory
 - Adresáře
- Příklad se souborem
 - Mód souboru
 - Vlastnictví souboru
- Příklad s adresářem
- Změna módu

V Debianu je implementována technika přístupových práv k souborům dle standardu POSIX. Takže všechny POSIXové systémy mají stejný způsob řízení přístupových práv, tzn. všechny Linuxy, Unixy, Androidi a třeba i MAC OS (jež je založený na UNIXu).

UGO

Každý soubor nese informaci o vlastníkovi (**U**ser) a skupině (**G**roup), který jej vytvořil. Takže když uživatel vytvoří soubor, automaticky bude vlastníkem a jako skupina se u souboru nastaví primární skupina tohoto uživatele. To o skupině nemusí platit vždy, vizte `setgid`.

Poslední písmenko ve zkratce UGO jsou tzv. ostatní (**O**thers). To doslova znamená všichni ostatní uživatelé v systému.

RWX

Další trojicí písmenek lze definovat, jaká konkrétní oprávnění má mít k souboru vlastník nebo skupina nebo ostatní. Trošku posunutý význam RWX mají oproti souborům adresáře, i když adresář je vlastně jen soubor s příznakem, že je adresářem.

Uvádím proto význam RWX nejprve pro soubory.

Soubory

- **R** .. Read (čtení)
- **W** .. Write (zápis)
- **X** .. eXecute (spuštění)

Jak ste si nejspíš právě odvodili, právo *Read* znamená číst obsah souboru, právo *Write* umožňuje měnit obsah souboru a poslední právo *Execute* dovolí soubor spustit (má samozřejmě význam u souborů, které jsou určeny pro spuštění).

Adresáře

Mají stejná oprávnění v podobě UGO-RWX jako soubory, jen význam je přenesený s ohledem na to, že jde o adresář, který slouží jako kontejner pro soubory.

- **R** .. čtení obsahu adresáře (informace o souborech)
- **W** .. vytváření nových souborů (a tedy i adresářů)

- **X** .. procházení (vstup do adresáře)

Příklad se souborem

Vytvořme soubor *pokus.txt* příkazem `touch`.

```
tux@wheezy:~$ touch pokus.txt
```

Nyní příkazem `ls -l` si vypíšeme obsah adresáře, kde se objeví i právě vytvořený soubor. Parametr `-l` se postará o podrobný výpis, ze kterého budeme moci vyčíst vlastníka souboru, skupinu a práva.

```
tux@wheezy:~$ ls -l
celkem 0
-rw-r--r-- 1 tux tux 0 pro 24 10:40 pokus.txt
```

Nenechte se zmást informací v úvodu: *celkem 0*, nejde o počet souborů, ale počet alokovaných bloků u všech souborů ve výpisu. Náš soubor má nulovou velikost, proto 0 (vizte `ls -l -s`).

Co je pro nás ale momentálně důležitější, je řádek popisující soubor *pokus.txt*. Začněme zleva.

Mód souboru

První znak, ve výpisu pomlčka (z celé sekvence `-rw-r--r--`), znamená, že jde o regulérní soubor. Na této pozici se může objevit např. `l` - symbolický odkaz, `c` - znakové zařízení, `b` - blokové zařízení nebo `d` - adresář.

Po prvním znaku následuje sekvence tří trojic (můžete je spočítat) znaků, kde se objevuje (zpravidla) jedno z písmenek `RWX` nebo zase pomlčka. V našem případě to je `rw-r--r--`. Když bychom sekvenci rozepsali po trojicích, dostaneme `rw-`, `r--` a `r--`. A znamená to, že první uvedená trojice `rw-` platí pro vlastníka (**U**ser), druhá `r--` pro skupinu (**G**roup) a třetí `r--` pro ostatní (**O**thers).

Pokud bych volně přeložil význam `rw-`, pak vlastník může soubor *pokus.txt* číst (zjistit, co je v něm) a upravovat (měnit jeho obsah). Nemůže jej však spustit (na třetí pozici je totiž `-`), protože chybí právo `x`.

Skupina a ostatní mají stejná práva. Mohou pouze číst obsah souboru (a tím jej třeba i zkopírovat).

Vlastnictví souboru

Z výpisu `ls -l` lze dále vyčíst, kdo je vlastníkem souboru a jakou má přidělenou skupinu. Jak již bylo popsáno výše, za běžných okolností se u vytvářených souborů nastavuje vlastník a skupina uživatele, který soubor vytvořil.

```
-rw-r--r-- 1 tux tux 0 pro 24 10:43 pokus.txt
```

Nenechte se zmást, že se výpisu objevuje dvakrát *tux*. V prvním případě jde o vlastníka a v druhém s skupinu. Připomeňme si *id*, ze kterého lze vyčíst jméno vlastníka (uživatele) a primární skupinu.

```
tux@wheezy:~$ id
uid=1000(tux) gid=1000(tux)
skupiny=1000(tux),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev)
)
```

Příklad s adresářem

Podívejme se, jak vypadá obsah kořenového adresáře.

```
tux@wheezy:~$ ls -l /
celkem 76
drwxr-xr-x  2 root root  4096 dub 21 18:55 bin
drwxr-xr-x  3 root root  4096 dub 21 18:56 boot
drwxr-xr-x 13 root root 3040 kvě 24 07:37 dev
drwxr-xr-x 76 root root  4096 kvě 23 13:27 etc
drwxr-xr-x  3 root root  4096 kvě 23 13:27 home
lrwxrwxrwx  1 root root    28 dub 21 18:48 initrd.img -> /boot/initrd.img-3.2.0-4-486
drwxr-xr-x 14 root root  4096 dub 21 18:51 lib
drwx----- 2 root root 16384 dub 21 18:47 lost+found
drwxr-xr-x  3 root root  4096 dub 21 18:47 media
drwxr-xr-x  2 root root  4096 úno  3 10:24 mnt
drwxr-xr-x  2 root root  4096 dub 21 18:47 opt
dr-xr-xr-x 72 root root    0 kvě 24 07:37 proc
drwx----- 3 root root  4096 kvě 23 10:34 root
drwxr-xr-x 12 root root   560 kvě 24 07:37 run
drwxr-xr-x  2 root root  4096 dub 21 18:59 sbin
drwxr-xr-x  2 root root  4096 čen 10 2012 selinux
drwxr-xr-x  2 root root  4096 dub 21 18:47 srv
drwxr-xr-x 12 root root    0 kvě 24 07:37 sys
drwxrwxrwt  4 root root  4096 kvě 24 12:15 tmp
drwxr-xr-x 10 root root  4096 dub 21 18:47 usr
drwxr-xr-x 11 root root  4096 dub 21 18:47 var
lrwxrwxrwx  1 root root    24 dub 21 18:48 vmlinuz -> boot/vmlinuz-3.2.0-4-486
```

Nyní si můžete všimnout, že většina položek ve výpisu jsou adresáře (písmenko "d"). Až na dvě výjimky, což jsou případy symlinků (symbolický odkaz).

Za všimnutí rovněž stojí i to, že všechny adresáře mají vždy kombinaci práv `r-x`, někdy `rwX`. V případě adresářů by použití samotného `r` nebo `X` nemělo moc význam. Uvažte, jak by to vypadalo, kdybychom mohli číst obsah adresáře, ale nemohli jej procházet?

Vytvořme adresář a podívejme se, jaká se mu nastaví práva.

```
tux@wheezy:~$ ls -l
celkem 4
drwxr-xr-x 2 tux tux 4096 pro 24 12:29 hokus
-rw-r--r-- 1 tux tux    0 pro 24 10:43 pokus.txt
```

Téměř stejné jako u souboru, jen **x** je zde navíc.

Změna módu

Přidání nebo odebrání některého z práv (rwx) vlastníkovi, skupině nebo ostatním lze provést příkazem **chmod**. Příkaz **chmod** má poněkud barvitou syntaxi. Různé věci lze nastavovat různě složitě. V ukázce níže odebereme všechna přidělená práva *ostatním* u adresáře *hokus*.

```
tux@wheezy:~$ chmod o=-r-x hokus/
tux@wheezy:~$ ls -l
celkem 4
drwxr-x--- 2 tux tux 4096 pro 24 12:29 hokus
-rw-r--r-- 1 tux tux    0 pro 24 10:43 pokus.txt
```

Na pozici *ostatních* jsou nyní tři pomlčky **---**. Obdobně bychom mohli odebrat práva i skupině.

```
tux@wheezy:~$ chmod g=-r-x hokus/
tux@wheezy:~$ ls -l
celkem 4
drwx----- 2 tux tux 4096 kvě 24 12:29 hokus
-rw-r--r-- 1 tux tux    0 kvě 24 10:43 pokus.txt
```

Kdybychom naopak chtěli vše vrátit zpátky:

```
tux@wheezy:~$ chmod g=+r+x,o=+r+x hokus/
tux@wheezy:~$ ls -l
celkem 4
drwxr-xr-x 2 tux tux 4096 kvě 24 12:29 hokus
-rw-r--r-- 1 tux tux    0 kvě 24 10:43 pokus.txt
```


Správa úložšť

Název školy	Vyšší odborná škola obalové techniky a Střední škola, Štětí, příspěvková organizace		
Adresa školky	Kostelní 134, 411 08 Štětí		
IČ	46773509		
Název operačního programu	OP Vzdělávání pro konkurenceschopnost		
Registrační číslo	CZ.1.07/1.5.00/34.1006		
Označení vzdělávacího materiálu	VY_32_INOVACE_21_PSS_409		
Název tematické oblasti (sady)	Operační systémy		
Název materiálu	Správa úložšť		
Anotace	<p>Popis práce s disky a diskovými oblastmi v systému Debian GNU Linux. Text představuje tzv. bloková zařízení, virtuální soubory, pomocí kterých se přistupuje fyzickým zařízením. Praktická část je zaměřena na vytvoření tabulky oblastí, ve které pak lze vytvářet primární nebo logické oblasti. Následně popisuje souborové systémy a způsob, jakým lze FS vytvořit v připravené oblasti. Nakonec je představen příkaz mount pro připojení oblasti do adresářové struktury.</p>		
Autor	Ivan Pomykacz	Jazyk	český
Očekávaný výstup	Orientuje se ve virtuálních zařízeních pevných disků. Vytvoří logickou oblast na pevném disku. Na oblasti vytvoří linuxový souborový systém a oblast připojí do adresářové struktury.		
Klíčová slova	MBR, partition table, mount point, blokové zařízení		
Druh výukového zdroje	Výklad	Věková skupina žáků	17+
Typ interakce	aktivita	Ročník	3.
Speciální vzdělávací potřeby	žádné		
Zhotoveno, (datum/období)	10.12.2013	Celková velikost	

Obsah

- Zařízení
- Nový disk
 - Správa oblastí
 - Systém souborů
 - Vytvoření FS
 - Připojení oblasti
 - Odpojení oblasti

Pro potřeby tohoto tutoriálu jsem přidal do virtuálního stroje další pevný disk, připojený k SATA řadiči na port 1. Tento text neuvažuje použití GPT (GUID Partition Table), která je nezbytná pro disky větší jak 2TB.

Zařízení

V systémech na bázi Linuxu existuje speciální adresář `/dev`, který obsahuje virtuální soubory zařízení. Připojíme-li např. flash disk (přes USB) nebo pevný disk (přes SATA), objeví se zde nový soubor reprezentující toto zařízení.

Vypíšeme-li alespoň část adresáře (všechny soubory začínající na "sd"), uvidíme několik souborů typu "b" (blokové zařízení).

```
root@wheezy:~# ls -l /dev/sd*
brw-rw---T 1 root disk 8,  0 kvě 24 14:11 /dev/sda
brw-rw---T 1 root disk 8,  1 kvě 24 14:03 /dev/sda1
brw-rw---T 1 root disk 8,  2 kvě 24 14:03 /dev/sda2
brw-rw---T 1 root disk 8,  5 kvě 24 14:03 /dev/sda5
brw-rw---T 1 root disk 8, 16 kvě 24 14:03 /dev/sdb
```

Ve výpisu se objevují celkem dva pevné disky (mimočodem tímto nelze rozeznat např. flash disk od HDD nebo SSD). Jejich označení je `/dev/sda` a `/dev/sdb`. (Později se podíváme do adresáře `/dev/disk/`, kde lze získat podrobnější informace)

Začátek názvu souboru "sd" je v tomto případě neměnný. Mění se třetí písmenko a přidává číslice. Zařízení `sda` je v tomto případě disk, který byl detekován jako první (SATA port 0). Druhý v pořadí je `sdb` (SATA port 1).

Čísla (`sda1`, `sda2` a `sda5`) nesou označení logických oblastí (partitions), které jsou na disku vytvořené. Konkrétně jde o primární oblast (1) a rozšířenou oblast (2), ve které se nachází jedna logická (5). Na disku je použité rozdělení dle DOS-MBR.

Ponechme nyní disk `/dev/sda`, protože na něm je nainstalovaný náš systém. Zaměříme se na `/dev/sdb`, který je prázdný.

Nový disk

Chceme-li začít používat nový disk musíme napřed učinit několik kroků.

1. Vytvořit na disku oblast (`cfdisk`).

2. Vytvořit v oblasti souborový systém (`mkfs`).
3. Připojit disk do adresářové struktury (`mount`).

Správa oblastí

Příkazem `fdisk /dev/sdb` lze editovat tabulku oblastí na zařízení `sdb`.

```
root@wheezy:~# fdisk /dev/sdb
```

```
fdisk (util-linux 2.20.1)
```

```
Disk: /dev/sdb
```

```
Velikost 8 589 934 592 bajtů, 8 589 MB
```

```
Hlav: 255   Sektorů na stopu: 63   Cylindrů: 1 044
```

Název [Popis]	Příznaky Velik. (MB)	Typ oddílu	Typ SS
-			

		Pri/Log	Volný prostor
8589,94	*		

```
[Nápověda]  [ Nový  ]  [ Tisk  ]  [ Konec  ]  [ Jednotky ]  [
Uložit  ]
```

Vytvoří nový oddíl ve volném prostoru

Pomocí kurzorových kláves vlevo a vpravo se pohybujete v dolní nabídce. Klávesami nahoru a dolů měníte oblast/volný prostor, který chcete spravovat. Nyní ovšem není z čeho vybírat.

Vytvořme novou *primární* oblast s celkovou velikostí disku. Příznak zaveditelnosti můžeme ignorovat, neboť na tomto disku nebude operační systém, a navíc zavaděč GRUB jej nepoužívá.

```
fdisk (util-linux 2.20.1)
```

```
Disk: /dev/sdb
```

Velikost 8 589 934 592 bajtů, 8 589 MB
 Hlav: 255 Sektorů na stopu: 63 Cylindrů: 1 044

Název [Popis]	Příznaky Velik. (MB)	Typ oddílu	Typ SS
-			
sdb1 8589,94	*	Primární	Linux

[Zavedit.] [Smazat] [Nápověda] [Zvětšit] [Tisk] [Konec]
 [Druh] [Jednotky] [Uložit]

Přepne příznak zaveditelnosti aktuálnímu oddílu

V seznamu se objevila nová položka **sdb1**. Nyní je důležité nezapomenout uložit změny a poté můžeme program ukončit. Objeví se varovná hláška programu:

Na disku byly provedeny změny.

VAROVÁNÍ: Pokud jste vytvořili či změnili nějaké DOS 6.x oddíly, přečtěte si prosím návod k programu cfdisk, abyste získali další informace.

root@wheezy:~#

S MS-DOS MBR můžou být trochu komplikace, zejména při změnách velikostí oblastí. V manuálové stránce je pak uveden příkaz, kterým lze případné problémy odstranit (doslova). Nicméně je dobré brát na zřetel, že některé změny se nemusí projevit okamžitě po jejich uložení. V některých případech se nevyhneme i restartu systému.

V našem případě by ovšem příkaz `ls -l /dev/sdb*` ukáže další zařízení, které je spojeno s právě vytvořenou oblastí.

```
root@wheezy:~# ls -l /dev/sd*
brw-rw---T 1 root disk 8, 16 kvě 24 15:42 /dev/sdb
brw-rw---T 1 root disk 8, 17 kvě 24 15:15 /dev/sdb1
```

Pokud by se tak nestalo, restartuje počítač.

System souborů

Následuje vytvoření souborového systému, který nám umožní na oblasti (disku) vytvářet soubory a adresáře, jinými slovy umožní nám ukládat data.

Linux podporuje celou řadu souborových systémů. Ty se liší svou vyspělostí a funkcemi. Určitý souborový systém můžeme volit z různých důvodů - je optimalizován pro SSD, podporuje žurnálování, je optimalizován na práci s tisíci drobných souborů, podporuje snapshoty, dynamické zvětšování oblasti za chodu, atd.

Pro Linux existuje tzv. *nativní* souborový systém. Asi jako pro MS Windows je nativním NTFS, tak pro Linux to je "ext", konkrétně ve verzi 4, tedy "ext4".

Podpora souborového systému bývá většinou otázka jádra. Např. MS Windows oficiálně nepodporují ext4 (prostě není možné MS Windows nainstalovat na tento souborový systém). Linux zase nepodporuje NTFS (nezjišťoval jsem, zda-li je možné provést instalaci na tento FS). Nicméně, jak v MS Windows, tak v Linuxu je možné připojit jak ext4 (i když třeba jen pro čtení, v případě MS Windows), tak NTFS (i pro zápis).

Vytvoření FS

Příkazem `mkfs` je tedy možné vytvořit (někdy se též používá označení formátovat oblast) souborový systém.

```
root@wheezy:~# mkfs -t ext4 /dev/sdb1
mke2fs 1.42.5 (29-Jul-2012)
Jmenovka systému souborů=
Typ OS: Linux
Velikost bloku=4096 (log=2)
Velikost fragmentu=4096 (log=2)
Krok=0 bloků, Šířka pásu=0 bloků
524288 iuzlů, 2097144 bloků
104857 bloků (5.00 %) rezervováno pro superuživatele
První blok dat=0
Maximum bloků v systému souborů=2147483648
64 skupin bloků
32768 bloků ve skupině, 32768 fragmentů ve skupině
8192 iuzlů ve skupině
Zálohy superbloku uloženy v blocích:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632
```

```
Alokují se tabulky skupin: hotovo
Zapisuji tabulky iuzlů: hotovo
Vytváří se žurnál (32768 bloků): hotovo
Zapisuji superbloky a účtovací informace systému souborů: hotovo
```

Tady pozor! Program se neptá: *A opravdu chcete tento disk naformátovat?*. Ostatně jako u většiny příkazů v Linuxu, pokud si dáte tu práci a příkaz napíšete do příkazové řádky a potvrdíte klávesou

Enter, pak asi víte co děláte nebo se to od vás alespoň očekává.

Přepínačem `-t` se nastavuje požadovaný souborový systém. V případě "ext" byste mohli použít verze "ext2" nebo "ext3", ale asi k tomu nebudete mít důvod.

Připojení oblasti

Nakonec musíme připravenou oblast připojit do adresářové struktury. Je to vcelku na nás, ale ... Opět musíme vědět, co děláme.

Zařízení má smysl připojit do prázdného adresáře, protože jinak se obsah tohoto adresáře překryje daty z právě připojeného zařízení. Překryje neznámá že nenávratně zmizí. Zmizí jen do doby, než zařízení zase odpojíme.

Vytvořme podadresář `/mnt/data` a připojme do něj zařízení `/dev/sdb1` příkazem `mount`.

```
root@wheezy:~# mkdir /mnt/data
root@wheezy:~# mount -t ext4 /dev/sdb1 /mnt/data/
```

Odpojení oblasti

Oblast se automaticky odpojí při vypnutí počítače. Při zapnutí se totiž automaticky nepřipojí, ledaže bychom provedli zásah do souboru `/etc/fstab`. Ruční odpojení provedeme příkazem `umount`.

```
root@wheezy:~# umount /mnt/data
```

Správa balíčků

Název školy	Vyšší odborná škola obalové techniky a Střední škola, Štětí, příspěvková organizace		
Adresa školky	Kostelní 134, 411 08 Štětí		
IČ	46773509		
Název operačního programu	OP Vzdělávání pro konkurenceschopnost		
Registrační číslo	CZ.1.07/1.5.00/34.1006		
Označení vzdělávacího materiálu	VY_32_INOVACE_21_PSS_410		
Název tématické oblasti (sady)	Operační systémy		
Název materiálu	Správa balíčků		
Anotace	Materiál vysvětluje pojmy a koncepci repozitářů a balíčků. Dále pak popisuje správu software, totiž instalaci/odinstalaci balíčků a přidávání dalších repozitářů jako zdrojů pro balíčky.		
Autor	Ivan Pomykacz	Jazyk	český
Očekávaný výstup	Nastaví vzdálený repozitář. Instaluje/Odinstaluje balíček. Rozumí závislostem mezi balíčky a proč je tyto závislosti vhodné dodržovat.		
Klíčová slova	balíček, repozitář, instalace, odinstalace		
Druh výukového zdroje	Výklad	Věková skupina žáků	17+
Typ interakce	aktivita	Ročník	3.
Speciální vzdělávací potřeby	žádné		
Zhotoveno, (datum/období)	21.01.2014	Celková velikost	

Obsah

- Repozitáře
 - Balíček
 - Vyhledávání balíčků
- Správce balíčků
 - Instalace
 - Odinstalace
 - Aktualizace seznamu balíčků
 - Aktualizace software
 - Povýšení verze celého systému
- Správa zdrojů

Repozitáře

V systému Linux se software spravuje jiným způsobem než např. v MS Windows. Princip spočívá v tom, že veškerý software je umístěn v tzv. repozitáři. Repozitář může být jakékoli úložiště, např. CD/DVD nebo server síti Internet. Například právě nainstalovaný Debian GNU Linux (Wheezy) obsahuje na instalačním CD část repozitáře s vybranými balíčky, nutnými pro instalaci základního systému.

Umístění repozitářů lze samozřejmě mezi sebou kombinovat. Část repozitáře je tedy na CD a zbytek v Internetu. Repozitář není navíc jen jeden, ale můžeme jich mít nastaveno hned několik.

Balíček

Balíčky jsou soubory se speciální koncovkou deb. Je to vlastně obyčejný komprimovaný archiv, který se rozbalí do předem daných adresářů. Může obsahovat poinstalační skripty, které se postarají o dodatečnou konfiguraci dle aktuálního nastavení systému.

Pokud chceme nainstalovat nějaký program musíme znát název balíčku, a potom stačí tento balíček nainstalovat přes správce balíčků. Například program Midnight Commander má instalační balíček `mc`. Lze jej nainstalovat (jako root) příkazem:

```
root@wheezy:~# apt-get install mc
Čtu seznamy balíčků... Hotovo
Vytváří se strom závislostí
Čtu stavové informace... Hotovo
Následující extra balíky budou instalovány:
  libglib2.0-0 libglib2.0-data mc-data shared-mime-info unzip
Navrhované balíky:
  zip arj xpdf pdf-viewer dbview odt2txt gv catdvi djvulibre-bin imagemagick
  python-boto python-tz
Následující NOVÉ balíky budou nainstalovány:
  libglib2.0-0 libglib2.0-data mc mc-data shared-mime-info unzip
0 aktualizováno, 6 nově instalováno, 0 k odstranění a 0 neaktualizováno.
Potřebuji stáhnout 5 785 kB archivů.
Po této operaci bude na disku použito dalších 22,8 MB.
Chcete pokračovat [Y/n]?
```


Výpis program `apt-get` je poněkud upovídáný. Později se dozvíme, že existuje něco jako závislosti mezi balíčky.

Balíčky nemusí nutně obsahovat jen programy, ale mohou to být například tapety na plochu nebo sdílené součásti mezi programy například knihovna SDL. Nebo to mohou být jazykové balíčky pro dané programy (např. `iceweasel-l10n-cs`).

Vyhledávání balíčků

Pokud neznáme název balíčku, můžeme prohledávat databázi balíčku pomocí klíčových slov. K tomu slouží příkaz `apt-cache`:

```
root@wheezy:~# apt-cache search webový prohlížeč
ajaxterm - Webový terminál napsaný v Pythonu
bluefish - pokročilý Gtk+ HTML editor
chimera2 - Webový prohlížeč pro X
doc-central - Webový prohlížeč dokumentace
dpkg-www - Webový prohlížeč balíčků Debianu
elinks - Pokročilý textový prohlížeč WWW stránek
epiphany-browser - Intuitivní webový prohlížeč pro GNOME
epiphany-browser-data - Datové soubory pro hlavní webový prohlížeč prostředí GNOME
epiphany-browser-debug - Ladicí symboly pro hlavní webový prohlížeč prostředí GNOME
ezmlm-browse - Webový prohlížeč archivů ezmlm-idx
feh - Prohlížeč obrázků používající imlib2
httrack - Zkopíruje kompletní weby do Vašeho počítače (prohlížení offline)
jsmath - Rovnice TeXu v dokumentech HTML
junior-internet - Debian Jr. - internetové nástroje
links2 - Webový prohlížeč běžící v grafickém i textovém módu
midori - Rychlý, nenáročný grafický prohlížeč
...
```

Větší šanci ovšem budete mít, použijete-li anglická slova, neboť ne všechny texty jsou přeložené do češtiny.

Správce balíčků

Každá distribuce používá zpravidla jeden balíčkovací systém. Distribuce odvozené od Debianu (např. Ubuntu) používají nástroje APT. Spustíte-li samotný příkaz `apt-get`, vypíše se vám nápověda, kde je vše potřebné.

Instalace

Instalaci provedeme příkazem `apt-get install`, za kterým následuje jeden nebo více balíčků, které chceme nainstalovat. Není tedy nutné při instalaci 5 balíčků spouštět příkaz pětikrát.

Odstalace

Lze provést příkazy `apt-get remove` nebo `apt-get purge`. V druhém případě se odstraní i konfigurační soubory (jak uvádí nápověda)

Aktualizace seznamu balíčků

Debian si stahuje seznamy všech balíčků pro offline použití. Např. když jsme hledali v databázi balíčků příkazem `apt-cache`.

Pokud ovšem přidáme do seznamu zdrojů další repozitáře (nebo naopak odebereme), je nezbytné okamžitě provést `apt-get update`. Ten se postará o stažení a sestavení aktuální cache balíčků.

Aktualizace software

Pokud chceme provést aktualizaci nainstalovaných balíčků, použijeme příkaz `apt-get upgrade` (v návaznosti na `apt-get update`). APT pak stáhne aktuální verze balíčků z repozitáře a nainstaluje.

Povýšení verze celého systému

Předchozí verze Debianu (v době psaní) byla verze 6 (kódové jméno Squeeze). Nyní je aktuální verze 7 (Wheezy). Mezi vydáním verze 6 a 7 uplynuly 3 roky. Za takovou dobu se toho ve vývoji distribuce vždy hodně změní (většinou k lepšímu).

Změny mají ovšem negativní vliv na stávající konfiguraci systému (ve smyslu nekompatibility). Vždy hodně záleží, co všechno a jak jste v systému nastavovali. Pokud na systému provozuje minimum věcí a víte, že upgrade na tuto verzi neovlivní budoucí fungování, např. databáze, která běží na serveru, pak asi nebude problém.

Linux je ovšem "zrádný" v tom, že umožňuje věci tvořit na míru. S tím ovšem nepočítají hlavní vývojáři distribuce a je proto možné, že upgradem rozbijí vaše nastavení a systém pak nebude fungovat jako před upgradem.

Zkráceně se tedy nedoporučuje provádět povyšování systému. Je třeba nejprve nastudovat všechny změny, které se odehrály za poslední 3 roky, aby nedošlo k nějaké kolizi. V produkčním prostředí si totiž nemůžeme dovolit žádné výpadky serveru.

Správa zdrojů

Ne všechny programy musím nutně být v jednom repozitáři. Zpravidla to funguje tak že distribuce má svůj hlavní repozitář kde se nachází většina používaných balíčků. Tyto balíčky mají zpravidla podobnou licenci která umožňuje tento způsob jejich distribuce. pokud balíčky mají jinou licenci pak se nachází v jiné větvi repozitáře. main free non-free Může se stát že balíček se neobjeví v žádné z větví repozitáře. Může to tak být z různých důvodů buď si to nepřejí vývojáři distribuce nebo autor balíčku nebo to neumožňuje license. Výhoda repozitářů je ta že se software jednoduše udržuje. To znamená pokud někdo vydá aktualizaci tato aktualizace se umístí do repozitáře a všichni si jí mohou stáhnout.

V závislosti na distribuci se liší konfigurace repozitáře. Debian GNU/Linux používá jako správce balíčků: aptitude. Jeho konfigurační soubory se nacházejí v adresáři `/etc/apt`. Tam se nachází soubor

`sources.list`, kde jsou zapsány všechny repozitáře, ze kterých se mohou balíčky stahovat. Nový řádek v souboru `sources.list` vypadá obecně takto: `deb :// ...`. Nechtě máme repozitář např. na `www` serveru `dusin` v adresáři `debian`. Tento lokální repozitář obsahuje několik balíčků z větve `main` a `contrib`. Obě tyto větve chceme využít. Použitý repozitář musí odpovídat instalované verzi Debian Linuxu, což je v našem případě verze Lenny. Výsledný řádek by pak mohl vypadat: `deb http://dusin/debian lenny main contrib`. Poslední tři klíčová slova v řádku tedy označují verzi GNU/Linuxu Debian, tj. `lenny` a dále dvě podskupiny repozitáře: hlavní větev, tj. `main` a větev přispěvatelů (`contrib`). Debian se takto snaží rozlišit balíčky od různých dodavatelů, případně různých licencí, např. `free` nebo `nonfree`. Při změně v konfiguraci je potřeba provést update repozitářů, během kterého si `aptitude` stáhne aktuální seznam dostupných balíčků. `apt-get update`

Diskové kvóty

Název školy	Vyšší odborná škola obalové techniky a Střední škola, Štětí, příspěvková organizace		
Adresa školky	Kostelní 134, 411 08 Štětí		
IČ	46773509		
Název operačního programu	OP Vzdělávání pro konkurenceschopnost		
Registrační číslo	CZ.1.07/1.5.00/34.1006		
Označení vzdělávacího materiálu	VY_32_INOVACE_21_PSS_411		
Název tematické oblasti (sady)	Operační systémy		
Název materiálu	Diskové kvóty		
Anotace	Text popisuje princip diskových kvót dvojího typu: blokových a souborových kvót. Seznamuje s principem užití a následně prakticky demonstruje instalaci a aktivaci kvót na konkrétním diskové oblasti.		
Autor	Ivan Pomykacz	Jazyk	český
Očekávaný výstup	Aktivuje diskové kvóty na zvolené diskové oblasti. Nastaví kvóty uživateli/skupině. Simuluje dosažení/překročení kvóty.		
Klíčová slova	kvóty, fstab, volby připojení, kvótové limity		
Druh výukového zdroje	Výklad	Věková skupina žáků	17+
Typ interakce	aktivita	Ročník	3.
Speciální vzdělávací potřeby	žádné		
Zhotoveno, (datum/období)	4.3.2014	Celková velikost	

Obsah

- Podpora diskových kvót
 - Podpůrný balíček
 - Volby připojení
 - Kvótové soubory
 - Aktivace kvót
- Přidělení kvót
 - Blokovaná a souborová kvóta
 - Dosažení kvóty
- Přehled kvót

Stále více obsahu digitálního obsahu je častěji ukládáno skrze služby běžících na serverech někde v síti. Místo k ukládání dat však není neomezené a vzhledem k tomu, že služby využívá současně více uživatelů, je mnohdy žádoucí zaručit, aby se dostalo na každého. Navíc lze koncept kvót velice jednoduše zpoplatnit - připlatíte a dostanete více místa.

Pro tento tutoriál budeme potřebovat nový disk (zde `/dev/sdb`). Na disku vytvoříme jednu oblast se souborovým systémem ext4. Zatím oblast nebudeme mountovat..

```
root@wheezy:~# cfdisk /dev/sdb
root@wheezy:~# mkfs.ext4 /dev/sdb1
```

Podpora diskových kvót

Podpůrný balíček

Podpora diskových kvót není k dispozici v základní instalaci. Stačí však doinstalovat balíček `quota`.

```
root@wheezy:~# apt-get install quota
```

Volby připojení

Nyní můžeme připojit disk příkazem `mount`, ale navíc přidáme volby `usrquota,grpquota` prepínačem `-o`. V podstatě tím oznámíme, že chceme sledovat uživatelské a skupinové kvóty.

Princip je ten, že systém pak sleduje a eviduje zabrané místo uživatelů nebo skupin v reálném čase. Udržuje si doslova databázi těchto údajů, protože výpočet zabraného místa na vyžádání je operace trvající dlouho. Stejně tak, když dojde k porušení konzistence této databáze a bude třeba ji znova vygenerovat.

```
root@wheezy:~# mount -t ext4 -o usrquota,grpquota /dev/sdb1 /srv/
```

Pozor, toto připojení není trvalé, po restartu se zařízení znovu automaticky nepřipojí. Zde je vhodnější použít `/etc/fstab`.

Kvótové soubory

Následuje vytvoření kvótových souborů, kde jsou uloženy informace o tom, kolik který uživatel/skupina má obsazeno místa/souborů. V našem případě to bude rychlý proces, protože právě připojený disk je prázdný.

```
root@wheezy:~# quotacheck -c -u -g /srv/
root@wheezy:~# ls /srv/
aquota.group  aquota.user  lost+found
```

Přepínač `-c` zajistí nové vytvoření souborů `aquota.group` a `aquota.user`. Přepínače `-u` a `-g` se používají pro uživatelské kvóty, resp. skupinové.

Aktivace kvót

Poslední akcí je aktivace kvót příkazem `quotaon`.

```
root@wheezy:~# quotaon /srv/
```

Přidělení kvót

Kvóty se přidělují příkazem `edquota`. Přepínačem `-u`, resp. `-g` se upřesňuje nastavení pro uživatele nebo skupinu. Po spuštění příkazu se otevře editor (nano), ve kterém lze přidělovat kvóty.

```
root@wheezy:~# edquota -u tux
Disk quotas for user tux (uid 1000):
  Filesystem            blocks          soft          hard          inodes
soft    hard
  /dev/sdb1              0                0            0            0
0        0
```

Kdybychom měli více disků připojených s podporou kvót, byly by zde uvedeny všechny.

V tabulce jsou sloupce pro nastavení limitů pro tzv. **blokovou** a **souborovou** kvótu. Ve sloupcích *blocks* a *inodes* jsou aktuální hodnoty. Tato čísla se needitují.

Do sloupce *soft* a *hard* se již vkládá požadovaná hodnota. Existuje možnost nastavení tzv. *soft* limitu, který je dovolen uživateli překročit. Hodnotu ve sloupci *hard* už překročit nelze. Pokud bude uživatel s takovou kvótou zrovna v polovině nějaké souborové operace (např. kopírování), pak tato operace jednoduše selže z důvodu nedostatku místa.

Bloková a souborová kvóta

Uživatelům lze přidělovat *blokové* kvóty, což je vlastně místo, které mohou uživatelé obsadit. (Vzpomínáte na blokové zařízení - pevný disk?) Hodnota je v KB. Tzn., že pro limit 10MB je třeba zadat 10240.

Druhým typem kvóty je kvóta *souborová*. Ta se vztahuje na počet souborů/adresářů, který může

uživatel vytvořit. Možná vám připadá zbytečné tento limit nastavovat. Nicméně u souborového systému ext4 je omezen počet souborů, který lze vytvořit. A pokud by nějaký splašený uživatel začal vytvářet prázdné soubory, mohl by tak zabrat všechny i-nody, aniž by zabral jakékoli místo. Jenomže bez volných i-nodů nelze vytvářet soubory - pro nikoho.

Dosažení kvóty

Přidělme uživateli *tux* následující limity na počet souborů:

```
Disk quotas for user tux (uid 1000):
  Filesystem          blocks      soft      hard      inodes
soft      hard
  /dev/sdb1           0           0         0         0
3          5
```

Vytvořme adresář `/srv/test` a nastavme mu nějaká práva, aby do něj mohl *tux* zapisovat.

```
root@wheezy:~# mkdir /srv/test
root@wheezy:~# chmod 777 /srv/test/
```

Nyní, jakou uživatel *tux* vytvořme několik prázdných souborů příkazem `touch`.

```
root@wheezy:~# su tux
tux@wheezy:/root$ cd /srv/test/
tux@wheezy:/srv/test$ touch pokus1
tux@wheezy:/srv/test$ touch pokus2
tux@wheezy:/srv/test$ touch pokus3
tux@wheezy:/srv/test$ touch pokus4
sdb1: warning, user file quota exceeded.
tux@wheezy:/srv/test$ touch pokus5
tux@wheezy:/srv/test$ touch pokus6
sdb1: write failed, user file limit reached.
touch: nelze se dotknout (provést příkaz „touch“) „pokus6“: Překročena
disková kvóta
```

Přehled kvót

Příkazem `repquota` lze monitorovat stavy kvót u uživatelů nebo skupin.

```
root@wheezy:~# repquota -u /srv/
*** Report for user quotas on device /dev/sdb1
Block grace time: 7days; Inode grace time: 7days
      Block limits                File limits
User      used  soft  hard  grace  used  soft  hard  grace
-----
root      --    24    0     0           3     0     0
tux       +-     0     0     0           5     3     5 6days
```


Pokročilý příkazový shell

Název školy	Vyšší odborná škola obalové techniky a Střední škola, Štětí, příspěvková organizace		
Adresa školky	Kostelní 134, 411 08 Štětí		
IČ	46773509		
Název operačního programu	OP Vzdělávání pro konkurenceschopnost		
Registrační číslo	CZ.1.07/1.5.00/34.1006		
Označení vzdělávacího materiálu	VY_32_INOVACE_21_PSS_412		
Název tematické oblasti (sady)	Operační systémy		
Název materiálu	Pokročilý příkazový shell		
Anotace	Text seznamuje některé pokročilejší vlastnosti příkazového shellu. Popisuje co se odehrává v pozadí zadávání příkazů do shellu, totiž kde se vyhledávají spustitelné soubory. Popisuje proměnné prostředí a jak lze měnit především proměnnou PATH. Dále ukazuje práci s aliasy, spouštění programů na pozadí, podmíněné spuštění a přesměrování std. Výstupu.		
Autor	Ivan Pomykacz	Jazyk	český
Očekávaný výstup	Získá informace z proměnných prostředí. Upravuje proměnnou PATH. Vytváří aliasy. Spouští program na pozadí. Používá přesměrování výstupu do souboru nebo jiných programů.		
Klíčová slova	PATH, proměnné prostředí, alias, program na pozadí, přesměrování výstupu		
Druh výukového zdroje	Výklad	Věková skupina žáků	17+
Typ interakce	aktivita	Ročník	3.
Speciální vzdělávací potřeby	žádné		
Zhotoveno, (datum/období)	1.4.2014	Celková velikost	

Obsah

- Jak je to s aplikacemi?
 - Práva na spuštění
- Proměnné prostředí
 - PATH
 - Mimo PATH
 - Nastavení PATH
 - Trvalé nastavení PATH
- Aliasy
- Program na pozadí
- Podmíněné spuštění příkazu
- Přesměrování výstupu
 - Přesměrování chybového výstupu

Jak je to s aplikacemi?

Aplikace - program se zpravidla skládá z několika částí:

- spustitelný binární kód (pokud nejde o skript - python, bash, perl, ruby, ...);
- sdílené knihovny a jiná data;
- globální konfigurace (společná pro všechny uživatele - tzv. výchozí/defaults);
- lokální konfigurace (individuální pro uživatele).

Například program *sleep* má svoji binární podobu v adresáři `/usr/bin` (lze zjistit příkazem `whereis`). Dále má své manuálové stránky v adresáři `/usr/share/man`.

Pokud tedy nainstalujeme nějakou aplikaci, tak se zpravidla rozkopíruje do různých adresářů v systému. Každý adresář má ovšem svůj význam.

Zatím co adresáře `/bin` a `/sbin` obsahují základní programy a systémové nástroje, tak adresář `/usr/bin` (`/usr/sbin`) obsahuje převážně uživatelské a správcovské aplikace. Ale pouze spustitelné soubory. Ostatní součásti aplikací jsou v dalších adresářích, jako `/usr/share`, `/usr/lib` nebo `/etc`.

Práva na spuštění

Chceme-li aplikaci spustit, musí mít soubor, kterým se spouští nastavená práva na spuštění (+x). Je vcelku jedno, zda-li jde o binární soubor ne skript. Spuštění skriptu je možné však obejít spuštěním interpretu, který následně vykoná předložený skript.

Zmíněné obcházení spouštění skriptů neberte jako bezpečnostní nedostatek. Platí totiž to, že uživatel musí mít práva nejen na spuštění skriptu, ale rovněž na všechny příkazy ve skriptu obsažené. Tzn., že pokud bude ve skriptu příkaz *mkfs*, uživatel musí mít práva i k tomuto příkazu zvlášť.

Proměnné prostředí

Kdykoli se přihlásí uživatel do systému, jsou v rámci jeho relace dostupné tzv. proměnné prostředí. Tyto proměnné obsahují zpravidla informace o relaci, jako kdo je přihlášen, kde se nachází (v jakém adresáři), kde má domovský adresář, atp.

Můžete si vypsat tyto proměnné příkazem `printenv`.

PATH

Proměnná `$PATH` obsahuje výčet cest, kde se nachází spustitelné soubory. Lze ji vypsat příkazem `echo $PATH`.

```
# echo $PATH
```

```
/usr/local/bin:/usr/bin:/bin:/usr/local/sbin:/usr/sbin:/home/mist/.local/bin:/home/mist/bin
```

Kdykoli tedy napíšete název programu do příkazového řádku, tak shell okamžitě začne prohledávat adresáře v proměnné `$PATH`, a pokud příkaz najde, tak se jej pokusí spustit.

Mimo PATH

Pokud se spouštěný program nenachází v cestách uvedených v proměnné `PATH`, pak nezbyvá, než zadat cestu ke spustitelnému souboru.

- **absolutní cesta**

```
# /usb/bin/sleep
```

- **relativní cesta**

```
# ./aplikace
```

(příklad nebude fungovat) U relativní cesty je nezbytné, pokud se spouštěná aplikace nachází v témže adresáři jako uživatel, zadat nejprve odkaz na aktuální adresář (`.`).

Nastavení PATH

Proměnnou lze nastavit příkazem `export`. Jeho spuštění ovšem ovlivní pouze relaci toho uživatele, kde byl příkaz spuštěn. To mj. znamená, že pokud relaci ukončíte a znovu otevřete, vše se vrátí do původního stavu

Ne zrovna ideální příklad:

```
# export PATH=/moje/biny
```

Všimněte si, že při exportu proměnné (její deklaraci) se nepoužívá znak `$`.

Uvedený příklad vám ovšem tak nějak znefunkční vaší relaci. To proto, že jste si z proměnné `$PATH` odstranili (prostě přepsali) ostatní cesty. Zkuste `echo $PATH` a uvidíte.

Pokud chcete relaci oživit, stačí ji ukončit a znovu spustit (otevřít). Korektnější varianta příkazu pro změnu, resp. přidání nové cesty by byla:

```
# export PATH=$PATH:/moje/biny
```

V tomto případě se do proměnné `$PATH` uloží i původní obsah. Dvojtečka `:` slouží jako oddělovač cest.

Trvalé nastavení PATH

Je poměrně nepraktické po každém přihlášení nebo otevření relace znovu zadávat ty samé příkazy pro export proměnných. Existuje proto soubor (skript), který se postará o tyto rutinní akce. Nazývá se `.bashrc`.

Uvedený příkaz pro export jednoduše napíšete na konec tohoto souboru a při dalším otevření relace se tyto změny automaticky aplikují.

Aliases

V některých případech admin/uživatel uvítá, aby si mohl některé příkazy pojmenovat třeba jinak. Nebo, a to je častější případ, aby se spuštěný příkaz automaticky spustil s konkrétními parametry.

Např. program `ls` pro výpis obsahu adresáře. Poměrně často se používá varianta `ls -l`. S aliasem je možné zápis zkrátit na jiný název, např. `ll`

```
alias ll="ls -l"
```

Z uvedeného příkladu je zřejmé, že nový alias se jmenuje `ll`. Při spuštění tohoto aliasu se ale vykoná příkaz v uvozovkách, tj. `ls -l`.

Aliases se zpravidla definují do souboru `.bashrc`.

Program na pozadí

V příkazové řádce je možné příkazy spouštět přímo na pozadí. Za spouštěným příkazem se připojí znak `&` a program se následně spustí automaticky na pozadí.

Podmíněné spuštění příkazu

Podmíněné spuštění programu lze docílit použitím operátoru `&&`. Je-li návratový kód předchozího příkazu `true`, vykoná se příkaz následující.

Přesměrování výstupu

Přesměrování výstupu pomocí `>` a `>>` umožňuje veškerý výstup programu přesměrovat do souboru nebo i jiného programu.

Přesměrování chybového výstupu

Přesměrování chybového výstupu `2>` znamená, že veškeré chybové hlášky lze stejně jako v případě přesměrování `std. výstupu` do souboru nebo jinam.

Správa procesů

Název školy	Vyšší odborná škola obalové techniky a Střední škola, Štětí, příspěvková organizace		
Adresa školky	Kostelní 134, 411 08 Štětí		
IČ	46773509		
Název operačního programu	OP Vzdělávání pro konkurenceschopnost		
Registrační číslo	CZ.1.07/1.5.00/34.1006		
Označení vzdělávacího materiálu	VY_32_INOVACE_21_PSS_413		
Název tematické oblasti (sady)	Operační systémy		
Název materiálu	Správa procesů		
Anotace	Materiál popisuje rozdíl mezi aplikacemi a službami v systému. Demonstruje aktivaci a deaktivaci démonů. Nakonec ukazuje možnosti sledování procesů a jejich vytížení CPU.		
Autor	Ivan Pomykacz	Jazyk	český
Očekávaný výstup	Aktivuje a deaktivuje služby/démony. Získá informace o běžících procesech. Ukončuje proces nebo službu. Rozumí přenášení uživatelských práv na proces/službu.		
Klíčová slova	proces, služba daemon, PID, monitorování		
Druh výukového zdroje	Výklad	Věková skupina žáků	17+
Typ interakce	aktivita	Ročník	3.
Speciální vzdělávací potřeby	žádné		
Zhotoveno, (datum/období)	15.4.2014	Celková velikost	

Obsah

- Spouštění procesů a služeb (démonů)
 - Programy vs služby
 - Programy na pozadí
 - Služby (aka démoni)
 - Aktivace/Deaktivace služeb
- Monitorování procesů
 - uptime
 - ps
 - top, htop

Spouštění procesů a služeb (démonů)

To jak se spouští běžné programy už víme. Zpravidla se nacházejí v cestách uvedených v proměnné \$PATH. Nebo je spouštíme zadáním relativní nebo absolutní cesty.

Spuštěný program má práva uživatele, který jej spustil (většinou). V některých případech tomu tak není, např. u programu `passwd` (pro změnu hesla). Výjimka u `passwd` je proto, že program provádí změny v souboru `/etc/shadow`, a do tohoto souboru má práva zápisu pouze root.

Služby neboli démoni (daemons) jsou speciální procesy, které běží v systému na pozadí. Spouští se zpravidla při startu systému procesem `init` nebo jiným (`systemd`, dle distribuce linuxu). Služba může běžet s právy uživatele `root`, ale z bezpečnostních důvodů se spouští pod tzv. neprivilegovaným uživatelem. Totiž takovým uživatelem, který nemá v systému zbytečně vysoká oprávnění.

Spouštění a ukončování služeb lze ručně ovládat příkazem `service`.

Programy vs služby

Běžné programy od služeb (démonů) se liší především tím, že disponují nějakou formou GUI, případně TUI a nebo jen CLI. Služba zpravidla jen zprostředkovává, nabízí (servíruje, odtud server) nějakou ... službu, kterou pak využívají konkrétní aplikace. Např. službu na servírování webových stránek zajišťuje `apache`. K prohlížení webových stránek je však potřeba webový prohlížeč (např. Mozilla Firefox).

Zatímco programy se spustí, udělají co je třeba, a následně se ukončí (uživatel je ukončí), tak služby jsou navrženy tak, aby běžely neustále. Nikdo vám ovšem nebrání spustit Firefox a nechat jej běžet několik dní nebo i déle.

Programy na pozadí

Pokud pracujeme v GUI, pak toto nemá moc smysl řešit. Grafické uživatelské prostředí je navrženo tak, abychom mohli spouštět více aplikací, tyto pak třeba minimalizovat a zase zpětně je vyvolat podle potřeby.

Pokud jsme v konzoli, tak je situace trochu jiná. Spustíme-li např. stahování souboru příkazem `wget` (`curl`), pak se příkazová řádka odmlčí do doby, než se stahování dokončí, a teprve potom budeme moci zadávat další příkazy.

Ano, je možné se přihlásit v další konzoli (**Alt+F2**), a potom v další a další, dokud nám nedojdou. Nebo můžeme program spustit přímo na pozadí.

Ukažme si to na příkladu s programem **sleep**. Při spuštění program s parametrem **10s** dojde k odmlčení programu na 10 vteřin a následně skončí.

```
tux@wheezy:~$ sleep 10s
```

Pokud bychom chtěli ihned po spuštění programu pokračovat psaním dalších příkazů, musíme spustit program **sleep** na pozadí. To lze provést pomocí **&**, který zapíšeme na konec příkazu:

```
tux@wheezy:~$ sleep 10s &
[1] 29115
tux@wheezy:~$
```

Program se spustí na pozadí a jakmile skončí, automaticky se ukončí. To zjistíme mj. tím, že stiskneme klávesu Enter a shell nás informuje, že aplikace na pozadí skončila.

```
tux@wheezy:~$
[1]+  Dokončena          sleep 10s
tux@wheezy:~$
```

Služby (aka démoni)

Služba je vlastně též program, který se umí většinou sám démonizovat, tj. spustit na pozadí. Nicméně tyto programy nespouštíme přímo, ale pomocí nástroje **service**, který spustí připravený skript pro nastartování požadované služby.

Např. služba pro konfiguraci sítě. Je to sice trochu atypický příklad, ale jde o princip.

Následující příkaz provede zastavení (zneplatnění) konfigurace sítě, tj. připojení k síti přestane fungovat. Všimněte si, že pouze root může zastavovat nebo spouštět služby.

```
root@wheezy:~# service networking stop
```

Opětovné spuštění/aktivace se provede příkazem:

```
root@wheezy:~# service networking start
```

Pokud má služba svůj init skript, pak je možné ji tímto způsobem spouštět nebo zastavovat.

Aktivace/Deaktivace služeb

Samotnou kapitolkou je pak aktivace, resp. deaktivace služby, aneb jak zařídit, aby se služba spouštěla při startu systému nebo naopak nespouštěla.

Za tímto účelem je tu pomocný skript **update-rc.d**.

```
root@wheezy:~# update-rc.d isc-dhcp-server disable
```

Obdobně, jako je deaktivace se provádí aktivace klíčovým slovem `enable`.

```
root@wheezy:~# update-rc.d isc-dhcp-server enable
```

Monitorování procesů

Pokud nás zajímá, co se se systémem zrovna děje - alespoň z hlediska procesů - pak se nám budou hodit nástroje jako: `uptime`, `ps`, `top` (`htop`).

Jde o jednoúčelové utility pro sledování procesů a jejich vytěžování systémových zdrojů (zejména operační paměť a procesor).

uptime

Příkaz vypíše jak dlouho systém běží bez restartu a průměrnou zátěž systému.

```
root@wheezy:~# uptime
12:50:53 up 286 days, 12:02, 1 user, load average: 0.08, 0.02, 0.01
root@wheezy:~#
```

ps

Podrobnější informace nabídne nástroj `ps`. I když v základu (bez prepínačů při spuštění) toho zas až tak moc nevypíše.

```
root@wheezy:~# ps
  PID TTY          TIME CMD
 3159 pts/0        00:00:00 bash
 3724 pts/0        00:00:00 ps
root@wheezy:~#
```

Z výpisu lze vyčíst PID - identifikátor procesu (ten se nám bude hodit až budeme procesy ukončovat příkazem `kill`). TTY je označení virtuálního terminálu a TIME je využitý čas na CPU.

Všimněte si, že ve výpisu je pouze `bash` a `ps`. `bash` je příkazový shell, ze kterého jsme spustili příkaz `ps` a jelikož i `ps` v době zjišťování běžících procesů běžel, je uveden rovněž ve výpisu.

Podrobnější informace lze zobrazit prepínačem `-f`. Tady pozor, příkaz `ps` má poněkud starší historii a jeho použití se liší v používání prepínačů, které zdědil od různých systémů (zejména BSD a UNIX). Např. příkaz `ps aux` je podobný jako `ps -ely`. Co z toho vyplývá je, že v různých tutoriálech se objevují různé příklady téhož.

```
root@wheezy:~# ps -f
UID          PID  PPID  C STIME TTY          TIME CMD
root         6356  5847  0 12:56 pts/2        00:00:00 /bin/bash
```



```
root      6722  6356  0 13:29 pts/2    00:00:00 ps -f
root@wheezy:~#
```

Výše už je alespoň vidět, kdo proces vlastník (UID), identifikátor rodičovského procesu (PPID), využití CPU (C), čas spuštění procesu (STIME) a celou cestu k programu (CMD).

Ve výpisu je ovšem stále poněkud málo procesů. Na průměrném systému je něco kolem stovky aktivních procesů. Výpis výše obsahuje pouze procesy v dané terminálové relaci. Vyzkoušejte příkazy:

```
root@wheezy:~# ps -e
root@wheezy:~# ps -ef
```

top, htop

```
root@wheezy:~# top
top - 13:40:50 up 5:01, 4 users, load average: 0.06, 0.20, 0.21
Tasks: 194 total, 4 running, 190 sleeping, 0 stopped, 0 zombie
Cpu(s): 3.2%us, 1.1%sy, 0.0%ni, 94.6%id, 1.0%wa, 0.0%hi, 0.0%si,
0.0%st
Mem: 2052200k total, 1915736k used, 136464k free, 93480k buffers
Swap: 2928636k total, 2580k used, 2926056k free, 619804k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
5414	tux	20	0	729m	243m	44m	S	6	12.1	1:05.39	firefox
1768	root	20	0	28572	4020	3188	R	2	0.2	0:02.30	upowerd
2546	root	35	15	6652	4132	956	R	2	0.2	0:13.91	preload
1	root	20	0	3740	2128	1352	S	0	0.1	0:00.90	init
2	root	20	0	0	0	0	S	0	0.0	0:00.00	kthreadd
3	root	20	0	0	0	0	S	0	0.0	0:01.08	ksoftirqd/0
5	root	20	0	0	0	0	S	0	0.0	0:04.86	kworker/u:0
6	root	RT	0	0	0	0	S	0	0.0	0:00.09	migration/0
7	root	RT	0	0	0	0	S	0	0.0	0:00.14	watchdog/0
8	root	RT	0	0	0	0	S	0	0.0	0:00.05	migration/1
10	root	20	0	0	0	0	S	0	0.0	0:00.85	ksoftirqd/1
12	root	RT	0	0	0	0	S	0	0.0	0:00.12	watchdog/1
13	root	0	-20	0	0	0	S	0	0.0	0:00.00	cpuset

Plánování úloh

Název školy	Vyšší odborná škola obalové techniky a Střední škola, Štětí, příspěvková organizace		
Adresa školky	Kostelní 134, 411 08 Štětí		
IČ	46773509		
Název operačního programu	OP Vzdělávání pro konkurenceschopnost		
Registrační číslo	CZ.1.07/1.5.00/34.1006		
Označení vzdělávacího materiálu	VY_32_INOVACE_21_PSS_414		
Název tématické oblasti (sady)	Operační systémy		
Název materiálu	Plánování úloh		
Anotace	Popisuje možnosti plánování spuštění úloh v systému pomocí démona Crond. Ukazuje způsob, jakým se úlohy plánují. Předkládá příklady plánování v atypických časech, cyklické plánování nebo jednorázové.		
Autor	Ivan Pomykacz	Jazyk	český
Očekávaný výstup	Naplánuje spuštění úlohy v daný čas nebo cyklu.		
Klíčová slova	plánování úloh, cron, crontab		
Druh výukového zdroje	Výklad	Věková skupina žáků	17+
Typ interakce	aktivita	Ročník	3.
Speciální vzdělávací potřeby	žádné		
Zhotoveno, (datum/období)	29.4.2014	Celková velikost	

Obsah

- Cron
- crontab
 - Editace crontabu
 - Způsoby zápisu

Čas od času se hodí, aby se provedla nějaká naplánovaná akce. Např. pročištění dočasných adresářů, ze kterých se odstraní soubory starší než ... Nebo spouštění pravidelných záloh vždy po práci. A co třeba automatizovaná kontrola/analýza log souborů za posledních 24 hodin.

Výhoda plánovače úloh spočívá v tom, že úloha nemusí běžet neustále v paměti a hlídat čas, kdy má provést požadovanou akci. Naopak, je tu plánovač, který dané úlohy spouští dle plánu.

Cron

V linuxových systémech je jedním z takových plánovačů **cron**. Je to démon/slужba, která běží na pozadí a hlídá si tzv. crontab soubory, což jsou vlastně jen tabulky jednotlivých uživatelů, kde je napsáno, co se má kdy spustit.

V systému jsou definované tzv. globální crontaby v adresáři **/etc**. Jmenovitě jde o

- cron.hourly
- cron.daily
- cron.weekly
- cron.monthly

Pokud do tohoto adresáře nahrajete skript, cron jej bude automaticky spouštět (každou hodinu, denně, týdně nebo měsíčně).

Podíváme-li se do těchto adresářů, tak např. **/etc/cron.daily** není prázdný.

```
tux@wheezy:~$ ls /etc/cron.daily/  
apt aptitude bsdmainutils dpkg exim4-base logrotate man-db mlocate  
passwd
```

Jsou zde skripty pro **apt** (správce balíčků) nebo **logrotate**, který má na starosti rotaci log souborů.

crontab

Plány spouštění se ukládají do crontabů, což jsou jednoduše tabulky, kde na každém řádku je definována samostatná úloha a její načasování.

Příklad plánu spuštění skriptu **uklid.sh** každou středu, ve dvě hodiny ráno.

```
0 2 * * 3 uklid.sh
```

Pokud se podíváte na klíč zápisu (níže), hned pochopíte o čem jsou čísla 0, 2 a 3. Hvězdička má speciální význam a znamená každou hodnotu z dané množiny. Pokud je daná množina dny v měsíci,

tak to budou všechny dny (tedy každý den) v daném měsíci.

```
# * * * * * příkaz
# | | | | |
# | | | | |
# | | | | |
# | | | | | _____ den v týdnu (0 - 7) (0 až 6 jsou neděle až sobota, nebo lze
použít 1 až 7 pro pondělí až neděle (neděle má tedy čísla 0, 7).
# | | | | | _____ měsíc (1 - 12)
# | | | | | _____ den v měsíci (1 - 31)
# | | | | | _____ hodina (0 - 23)
# | | | | | _____ minuta (0 - 59)
```

Editace crontabu

Každý uživatel může příkazem `crontab -l` vypsat svůj vlastní crontab.

```
tux@wheezy:~$ crontab -l
no crontab for tux
```

Pokud žádný crontab nemá, zobrazí hláška, kterou vidíte výše.

Nový crontab se založí příkazem `crontab -e`. Jak záhy uvidíte, spustí se předdefinovaný editor (pravděpodobně `nano`), ve kterém již bude otevřený dočasný soubor (v mém případě: `/tmp/crontab.TjuJUp/crontab`). Jakmile provedete potřebné změny a soubor uložíte (a ukončíte `nano`), crontab vás informuje, že zavádí nový crontab.

```
tux@wheezy:~$ crontab -e
no crontab for tux - using an empty one
crontab: installing new crontab
```

Pokud nyní zadáme výpis crontabu, dostaneme:

```
tux@wheezy:~$ crontab -l
0 2 * * 3 uklid.sh
```

Společně ještě se zakomentovaným textem, který byl z ukázky vyňat.

Způsoby zápisu

- každou druhou hodinu .. `*/2`, zápis pomocí `/` dělí interval na požadovaný počet.
- od pondělí do pátku .. `1-5`, použitý symbol `-` lze použít pro spojitý rozsah.
- v pondělí, středu a pátek .. `1,3,5`, symbolem `,` lze vyjmenovat požadované hodnoty (pozor na mezery, ty se nepíší).

Uvedené způsoby zápisu lze samozřejmě kombinovat a použít i na jiné hodnoty než zmíněné.

Logování

Název školy	Vyšší odborná škola obalové techniky a Střední škola, Štětí, příspěvková organizace		
Adresa školky	Kostelní 134, 411 08 Štětí		
IČ	46773509		
Název operačního programu	OP Vzdělávání pro konkurenceschopnost		
Registrační číslo	CZ.1.07/1.5.00/34.1006		
Označení vzdělávacího materiálu	VY_32_INOVACE_21_PSS_415		
Název tématické oblasti (sady)	Operační systémy		
Název materiálu	Logování		
Anotace	Text seznamuje se způsobem záznamu událostí v systému do tzv. logů. Popisuje přístup k logům, jejich prohlížení a interpretace. Představuje některé možnosti pro analýzu log souborů. Objasňuje automatickou archivaci a rotaci logů.		
Autor	Ivan Pomykacz	Jazyk	český
Očekávaný výstup	Lokalizuje systémový log nebo log pro odpovídající službu v systému. Prohlíží a analyzuje log soubor. Používá nástroje pro automatickou analýzu log souborů.		
Klíčová slova	log soubor, rotace logů, analýza logů		
Druh výukového zdroje	Výklad	Věková skupina žáků	17+
Typ interakce	aktivita	Ročník	3.
Speciální vzdělávací potřeby	žádné		
Zhotoveno, (datum/období)	6.5.2014	Celková velikost	

Obsah

- [Syslog](#)
 - [Informace z logu](#)
 - [Analýza logů](#)
 - [Rotace logů](#)

Každý systémový administrátor ví, že operační systém by měl logovat nejlépe všechno, a zároveň je potřeba tyto logy kontrolovat pro chyby, varování a podezřelé aktivity.

V Debianu je předinstalovaný logovací daemon *rsyslogd*. Je to služba, která ukládá informace, které jí posílají jiné služby, ať lokální nebo vzdálené. To mj. znamená, že v síti může být centrální počítač pro pouze logovací účely.

V našem případě se budeme zabývat pouze lokálním logováním.

Syslog

V adresáři `/var/log` se ukládají všechny logy.

```
root@wheezy:~# ls -l /var/log/
celkem 1840
-rw-r--r-- 1 root      root    17566 dub 21 18:55 alternatives.log
drwxr-xr-x 2 root      root     4096 dub 21 18:48 apt
-rw-r--r-- 1 root      root     4111 dub 21 18:51 aptitude
-rw-r----- 1 root      adm    24279 kvě 25 09:17 auth.log
-rw-rw---- 1 root      utmp         0 dub 21 18:47 btmp
-rw-r----- 1 root      adm    11234 kvě 25 08:12 daemon.log
-rw-r----- 1 root      adm  106472 kvě 25 08:12 debug
-rw-r----- 1 root      adm    25442 kvě 25 08:12 dmesg
-rw-r----- 1 root      adm    25461 kvě 24 14:03 dmesg.0
-rw-r----- 1 root      adm     8187 kvě 24 07:37 dmesg.1.gz
-rw-r----- 1 root      adm     8124 kvě 23 09:09 dmesg.2.gz
-rw-r----- 1 root      adm     8035 kvě 23 08:20 dmesg.3.gz
-rw-r----- 1 root      adm     8180 kvě 23 08:14 dmesg.4.gz
-rw-r--r-- 1 root      root  245497 kvě 24 18:20 dpkg.log
drwxr-s--- 2 Debian-exim adm     4096 dub 21 18:59 exim4
-rw-r--r-- 1 root      root    24072 kvě 24 18:20 faillog
drwxr-xr-x 2 root      root     4096 dub 21 18:48 fsck
drwxr-xr-x 3 root      root     4096 dub 21 18:59 installer
-rw-r----- 1 root      adm   412939 kvě 25 08:12 kern.log
-rw-rw-r-- 1 root      utmp  292876 kvě 25 08:48 lastlog
-rw-r----- 1 root      adm         0 dub 21 18:59 lpr.log
-rw-r----- 1 root      adm         0 dub 21 18:59 mail.err
-rw-r----- 1 root      adm         0 dub 21 18:59 mail.info
-rw-r----- 1 root      adm         0 dub 21 18:59 mail.log
-rw-r----- 1 root      adm         0 dub 21 18:59 mail.warn
-rw-r----- 1 root      adm   306646 kvě 25 08:12 messages
drwxr-xr-x 2 root      root     4096 dub 21 18:59 news
-rw-r----- 1 root      adm   434886 kvě 25 09:17 syslog
```

```
-rw-r----- 1 root      adm      1714 kvě 25 08:12 user.log
-rw-rw-r-- 1 root      utmp    140544 kvě 25 08:48 wtmp
```

Názvy souborů jsou definovány v konfiguračním souboru `/etc/rsyslogd.conf`. V systému momentálně nejsou nainstalovány služby jako apache, postfix nebo mysql. "Jediné" co nyní můžeme sledovat jsou výpisy jádra (`dmesg`), lokální pošta (`mail`), autentizace uživatelů (`auth.log`, včetně ssh) a obecný systémový log (`syslog`).

Podívejme se do takového logu, např. syslog.

```
root@wheezy:~# tail /var/log/syslog
May 25 08:12:00 wheezy acpid: 1 rule loaded
May 25 08:12:00 wheezy acpid: waiting for events: event logging is off
May 25 08:12:00 wheezy /usr/sbin/cron[2108]: (CRON) INFO (pidfile fd = 3)
May 25 08:12:00 wheezy /usr/sbin/cron[2109]: (CRON) STARTUP (fork ok)
May 25 08:12:00 wheezy /usr/sbin/cron[2109]: (CRON) INFO (Running @reboot jobs)
May 25 08:12:01 wheezy /usr/sbin/gpm[2409]: *** info [daemon/startup.c(131)]:
May 25 08:12:01 wheezy /usr/sbin/gpm[2409]: Started gpm successfully. Entered daemon mode.
May 25 08:12:10 wheezy kernel: [ 15.664189] eth0: no IPv6 routers present
May 25 08:17:01 wheezy /USR/SBIN/CRON[2527]: (root) CMD ( cd / && run-parts --report /etc/cron.hourly)
May 25 09:17:01 wheezy /USR/SBIN/CRON[2588]: (root) CMD ( cd / && run-parts --report /etc/cron.hourly)
```

Příkazem `tail` se vypíše posledních 10 řádků ze souboru. Logy jsou ve většině případů obyčejné textové soubory, kde na každém řádku je (většinou) nová zpráva. Na konci souboru jsou ty nejnovější záznamy (proto `tail`).

Pokud byste chtěli sledovat soubor po nějakou delší dobu, můžete s výhodou využít parametr `-f` u příkazu `tail`. Po spuštění příkazu `tail -f /var/log/syslog` program neskončí, ale čeká na změny v souboru, a potom je okamžitě vypíše. Klávesami `Ctrl+C` ukončíte činnost programu.

Informace z logu

U každého záznamu je časový údaj, za nímž následuje název počítače (zde wheezy). Následuje název démona, kterého se událost týká. Ve výpisu výše to jsou `acpid` a `cron`. `kernel` není přímo démon, ale v syslogu se objeví i výpisy z jádra.

Analýza logů

Není možné ručně procházet všechny logy, na všech systémech. Pro tyto potřeby byly vytvořeny různé analyzátoři, které parsují logy a sestavují reporty, které pak zobrazí správci, např. v e-mailu. Můžete najít analyzátoři logů pro konkrétní služby nebo obecné, které pomocí pluginů rozšiřují své možnosti.

Jedním z obecných analyzátorů je např. `logwatch`. Provádí analýzu vybraných logů a následně zasílá reporty na e-mail správce.

```

root@wheezy:~# apt-get install logwatch
root@wheezy:~# mkdir /var/cache/logwatch
root@wheezy:~# cp /usr/share/logwatch/default.conf/logwatch.conf
/etc/logwatch/conf/
root@wheezy:~# nano /etc/logwatch/conf/logwatch.conf

```

V příkazech výše jsem trochu předběhl a vytvořil adresář `/var/cache/logwatch`, který se uvádí v konfiguračním souboru `/etc/logwatch/conf/logwatch.conf`. Konfigurační soubor byl zkopírován z výchozí konfigurace. Kde jsem zjistil, že to tak musím udělat? Z manuálových stránek `man logwatch`. V konfiguračním souboru nás může zajímat:

```

Output = mail
Format = text
MailTo = root
MailFrom = Logwatch
Range = yesterday
Detail = Low

```

Až na *Output* jsem vše ponechal beze změny.

Po instalaci přibyl v adresáři `/etc/cron.daily` soubor `00logwatch`.

```

root@wheezy:~# ls /etc/cron.daily/
apt aptitude bsdmainutils dpkg exim4-base logrotate man-db mlocate
passwd quota 00logwatch

```

Nebudeme čekat do zítřka a spustíme `logwatch` už teď, abychom se mohli podívat na výstup.

```

root@wheezy:~# /etc/cron.daily/00logwatch

```

Nyní byl odeslán e-mail uživateli *root*, resp. *tux*. Ve výchozím stavu se e-maily pro uživatele *root* přesměrovávají uživateli vytvořenému při instalaci systému. Více v souboru `/etc/aliases`. Přihlásíme-li se jako *tux*, uvidíme v shellu zprávu.

```

You have mail.
Last login: Mon Apr 21 19:52:27 2014
tux@wheezy:~$

```

Programem `mutt` si můžeme došlou zprávu přečíst.

```

tux@wheezy:~$ mutt

```

```

q:Konec d:Smazat u:Obnovit s:Uložit m:Psát r:Odepsat g:Skupině
?:Nápověda
  1 N   May 25 logwatch@wheezy ( 101) Logwatch for wheezy (Linux)

```



```
---Mutt: /var/mail/tux [Msgs:1 New:1 3,5K]---(threads/da-
e)------(all)---
```

Pro zobrazení zprávy stiskneme Enter.

```
Date: Sun, 25 May 2014 10:20:05 +0200
From: logwatch@wheezy.odbornaskola.cz
To: root@wheezy.odbornaskola.cz
Subject: Logwatch for wheezy (Linux)
```

```
##### Logwatch 7.4.0 (05/02/12) #####
Processing Initiated: Sun May 25 10:20:04 2014
Date Range Processed: yesterday
                      ( 2014-May-24 )
                      Period is day.
Detail Level of Output: 0
Type of Output/Format: mail / text
Logfiles for Host: wheezy
```

```
#####
```

```
----- dpkg status changes Begin -----
```

Installed:

```
dbus:i386 1.6.8-1+deb7u1
gdisk:i386 0.8.5-1
libdbus-1-3:i386 1.6.8-1+deb7u1
libc48:i386 4.8.1.1-12+deb7u1
libnl-3-200:i386 3.2.7-4
libnl-genl-3-200:i386 3.2.7-4
libsystemd-login0:i386 44-11+deb7u4
quota:i386 4.00-4+deb7u1
```

```
----- dpkg status changes End -----
```

```
----- Kernel Begin -----
```

WARNING: Kernel Errors Present

```
EXT3-fs (sdb1): error: couldn't mount ...: 1 Time(s)
EXT4-fs (sda1): re-mounted. Opts: errors=remount-ro ...: 2 Time(s)
Error: Driver 'pcspkr' ...: 2 Time(s)
```

```
----- Kernel End -----
```

Rotace logů

Syslog (rsyslogd) se automaticky stará o tzv. rotaci logů. Soubory časem nakynou a zabírají místo. Objemné soubory se pomaleji parsují, a navíc obsahují i stará data. Syslog provádí pravidelnou tzv.

rotaci log souborů, kdy vezme aktuální log soubor a zkomprimuje jej pomocí gzip. Výsledek je hezky vidět na logu `dmesg`.

```
root@wheezy:~# ls -l /var/log/dmesg*
-rw-r----- 1 root adm 25442 kvě 25 08:12 /var/log/dmesg
-rw-r----- 1 root adm 25461 kvě 24 14:03 /var/log/dmesg.0
-rw-r----- 1 root adm 8187 kvě 24 07:37 /var/log/dmesg.1.gz
-rw-r----- 1 root adm 8124 kvě 23 09:09 /var/log/dmesg.2.gz
-rw-r----- 1 root adm 8035 kvě 23 08:20 /var/log/dmesg.3.gz
-rw-r----- 1 root adm 8180 kvě 23 08:14 /var/log/dmesg.4.gz
```

Nejaktuálnější log je v souboru `dmesg` a ten nejstarší v souboru `dmesg.4.gz`. Pro prohlížení souboru `.gz` lze využít program `zcat`, který transparentně provádí dekompresi.

```
root@wheezy:~# zcat /var/log/dmesg.4.gz | head
[ 0.000000] Initializing cgroup subsys cpuset
[ 0.000000] Initializing cgroup subsys cpu
[ 0.000000] Linux version 3.2.0-4-486 (debian-kernel@lists.debian.org)
(gcc version 4.6.3 (Debian 4.6.3-14) ) #1 Debian 3.2.54-2
[ 0.000000] BIOS-provided physical RAM map:
[ 0.000000] BIOS-e820: 0000000000000000 - 000000000009fc00 (usable)
[ 0.000000] BIOS-e820: 000000000009fc00 - 00000000000a0000 (reserved)
[ 0.000000] BIOS-e820: 00000000000f0000 - 0000000000100000 (reserved)
[ 0.000000] BIOS-e820: 0000000000100000 - 000000001fff0000 (usable)
[ 0.000000] BIOS-e820: 000000001fff0000 - 0000000020000000 (ACPI data)
[ 0.000000] BIOS-e820: 00000000fffc0000 - 0000000100000000 (reserved)
```

Archivace dat

Název školy	Vyšší odborná škola obalové techniky a Střední škola, Štětí, příspěvková organizace		
Adresa školky	Kostelní 134, 411 08 Štětí		
IČ	46773509		
Název operačního programu	OP Vzdělávání pro konkurenceschopnost		
Registrační číslo	CZ.1.07/1.5.00/34.1006		
Označení vzdělávacího materiálu	VY_32_INOVACE_21_PSS_416		
Název tématické oblasti (sady)	Operační systémy		
Název materiálu	Archivace dat		
Anotace	Seznamuje s archivací a bezztrátovou kompresí souborů pro účely zálohování dat. Na příkladech ukazuje, jak se vytváří archiv, jak se komprimuje a samozřejmě předkládá i obrácený proces, tj. dekomprese a rozbalení archivu.		
Autor	Ivan Pomykacz	Jazyk	český
Očekávaný výstup	Vytvoří archiv z několika souborů nebo adresářů. Komprimuje data. Rozbalí archiv a dekomprimuje data. Používá nástroje tar a gzip/gunzip.		
Klíčová slova	archivace, komprese, dekomprese, gzip, tar		
Druh výukového zdroje	Výklad	Věková skupina žáků	17+
Typ interakce	aktivita	Ročník	3.
Speciální vzdělávací potřeby	žádné		
Zhotoveno, (datum/období)	12.5.2014	Celková velikost	

Obsah

- tar
 - Tvorba archivu
 - Rozbalení archivu
- Komprese a dekomprese
 - gzip
 - gunzip
- tar a gzip

Pod archivací souborů si můžeme představit proces, kdy se jeden nebo spíše několik souborů naskládá do jednoho. Výsledný archiv se pak ještě komprimuje, kvůli úspoře místa.

tar

Tvorba archivu

Program `tar` má poměrně rozsáhlou manuálovou stránku, což poněkud znesnadňuje jeho rychlé zvládnutí. Začneme přepínačem `-c`, který indikuje vytvoření nového archivu. Následuje `-f` pro pojmenování výsledného archivu (včetně případné cesty).

```
tux@wheezy:~$ tar -c -f zaloha.tar /etc/passwd /etc/group
tar: Odstraňuje se úvodní „/„ z názvů prvků
tux@wheezy:~$ ls
zaloha.tar
```

Příkazem `ls` se pak můžeme přesvědčit o novém souboru `zaloha.tar`. Pokud vás zarazí, co se myslí tím, že se odstraňuje úvodní `/`, tak to dělá přesně, to co říká. Podívejme se do archivu:

```
tux@wheezy:~$ tar -t -f zaloha.tar
etc/passwd
etc/group
```

Přepínačem `-t` se pouze vypíše obsah archivu. Všimněte si, že u obou souboru chybí lomítko na začátku. Je tak z jediného důvodu a sice, že při rozbalování archivu by hrozilo nechtěné přepsání původních souborů.

Archivace celého adresáře je stejná jako v případě souborů.

```
tux@wheezy:~$ tar -c -f etc.tar /etc/
tar: Odstraňuje se úvodní „/„ z názvů prvků
tar: /etc/at.deny: Funkce open selhala: Operace zamítnuta
tar: /etc/exim4/passwd.client: Funkce open selhala: Operace zamítnuta
tar: /etc/gshadow-: Funkce open selhala: Operace zamítnuta
tar: /etc/group-: Funkce open selhala: Operace zamítnuta
...
tar: Končí se chybovým kódem, protože byly zaznamenány chyby
```

Uvedený příklad skončil chybou, protože běžný uživatel nemá práva na čtení všech souborů v /etc. Archiv je ovšem i tak vytvořen, ovšem některé soubory chybí.

Rozbalení archivu

Rozbalení se provádí za účasti parametrů `-x` a cesty k archivu `-f`.

```
tux@wheezy:~$ tar -x -f zaloha.tar
tux@wheezy:~$ ls
etc  etc.tar  zaloha.tar
```

V našem domovském adresáři přibyl nový adresář `etc`. Pozor, neplést s `/etc`, protože tento je v `/home/tux/etc` a nachází se v něm pouze dva soubory:

```
tux@wheezy:~$ ls etc
group  passwd
```

Důležité u rozbalování archivuje, že se rozbalí do umístění, kde se nacházíte. Tzn., pokud bychom chtěli archiv rozbalit jinde, musíme se nejprve příkazem `cd` do toho umístění dostat.

```
tux@wheezy:~$ mkdir Temp
tux@wheezy:~$ cd Temp/
tux@wheezy:~/Temp$ tar -x -f /home/tux/zaloha.tar
tux@wheezy:~/Temp$ ls
etc
```

V tom případě se změní ovšem cesta k souboru `-f`.

Kompresce a dekomprese

Kompresí lze ušetřit hodně místa v závislosti na tom, jaká data komprimujeme. Pokud budeme do archivu umísťovat jpeg obrázky nebo snad mp4 videa, výsledný efekt bude nulový. A to především proto, že obrázky i videa jsou již komprimované, a to ztrátovou kompresí.

Kompresce, kterou budeme provádět my, je bezztrátová. Tzn., že naprosto stejná data, která jsme zkomprimovali, obdržíme i při dekompresi.

Existuje několik kompresních algoritmů lišící se různými parametry. Jsou to např. kompresní poměr, rychlost komprese a dekomprese (která nemusí být nutně symetrická).

gzip

Jedním z rozšířených programů je gzip (GNU Zip). Použijeme jej na již vytvořený archiv `zaloha.tar`.

```
tux@wheezy:~$ gzip zaloha.tar
tux@wheezy:~$ ls
etc  etc.tar  Temp  zaloha.tar.gz
```

Po ukončení program `gzip` zmizí soubor `zaloaha.tar` a na místo něj tu uvidíme `zaloaha.tar.gz`.

Program `gzip` neumí komprimovat více souborů najednou, proto se soubory umísťují nejprve do `tar`.

gunzip

Přesně opačný efekt má program `gunzip`, který provede dekompresi archivu a podobně jako `gzip`, odstraní původní komprimovaný soubor a zůstane po něm jen ten dekomprimovaný.

```
tux@wheezy:~$ gunzip zaloaha.tar.gz
tux@wheezy:~$ ls
etc etc.tar Temp zaloaha.tar
```

tar a gzip

Program `tar` umí pomocí dalšího parametru `-z` vytvořit rovnou komprimovaný archiv.

```
tux@wheezy:~$ tar -z -c -f zaloaha.tar.gz /etc/passwd /etc/group
tar: Odstraňuje se úvodní „/„ z názvů prvků
```

bylo je potřeba upravit název souboru `-f`, aby bylo zřejmé, že jde již o `gzip` soubor.

Obdobně je na tom i dekomprese a rozbalzení archivu.

```
tux@wheezy:~$ tar -z -x -f zaloaha.tar.gz
```

Skriptování

Název školy	Vyšší odborná škola obalové techniky a Střední škola, Štětí, příspěvková organizace		
Adresa školky	Kostelní 134, 411 08 Štětí		
IČ	46773509		
Název operačního programu	OP Vzdělávání pro konkurenceschopnost		
Registrační číslo	CZ.1.07/1.5.00/34.1006		
Označení vzdělávacího materiálu	VY_32_INOVACE_21_PSS_417		
Název tématické oblasti (sady)	Operační systémy		
Název materiálu	Skriptování		
Anotace	Předkládá základy skriptování v shellu BASH. Ukazuje na jednoduchém příkladu způsob psaní skriptů, používání proměnných a výstupu z jiných programů.		
Autor	Ivan Pomykacz	Jazyk	český
Očekávaný výstup	Vytvoří spustitelný skript. Používá proměnné a výstupy jiných programů.		
Klíčová slova	skript, bash, proměnná		
Druh výukového zdroje	Výklad	Věková skupina žáků	17+
Typ interakce	aktivita	Ročník	3.
Speciální vzdělávací potřeby	žádné		
Zhotoveno, (datum/období)	13.5.2014	Celková velikost	

Obsah

- První skript
 - Spuštění skriptu
 - Použití interpretu
 - Práva na spuštění
 - Proměnná PATH
- Proměnné
 - Výstup programu
- Parametry při spuštění

Na Linuxu založené systémy jsou známé mimo jiné i tím, jak jsou svázány se skriptovacími jazyky (jako bash, perl nebo python). Např. pomocí bashe jsou napsány init skripty, pomocí kterých se startuje operační systém.

První skript

Vytvořme soubor `skript.sh`. Na první řádek se standardně píše speciální uvozovací řetězec, který říká, jaký interpret se má použít pro vykonání skriptu.

```
#!/bin/bash

echo "Skript běží ..."
sleep 2s
echo "Skript skončil."
```

Příkaz `echo` vypíše text v uvozovkách (i když ty nejsou vždy nutné). Program `sleep` pozastaví běh skriptu na dvě vřetiny.

Spuštění skriptu

Existuje několik možností jak skript spustit. Vytvořený soubor nemá práva na spuštění, a i kdyby je měl, nepůjde spustit jenom tak, že napíšeme do shellu název skriptu (jako u ostatních programů).

Použití interpretu

Tento způsob spočívá v tom, že prostě spustíme interpret bashe, který se postará o vykonání skriptu.

```
tux@wheezy:~$ bash skript.sh
Skript běží ...
Skript skončil.
```

Práva na spuštění

Skriptu jednoduše přidáme právo x. Následně skript spustíme tím, že k němu zadáme cestu (relativní nebo absolutní).


```
tux@wheezy:~$ chmod +x skript.sh
tux@wheezy:~$ ls -l skript.sh
-rwxr-xr-x 1 tux tux 68 kvě 25 21:02 skript.sh
tux@wheezy:~$ ./skript.sh
Skript běží ...
Skript skončil.
```

Proměnná PATH

Nejpoužitelnější je však umístění skriptu na místo, které je definováno v proměnné `PATH`. Případně toto místo do proměnné `PATH` přidáme.

Vytvořme adresář `~/bin` a přesuňme do něj vytvořený skript, který již má nastavená práva na spuštění.

```
tux@wheezy:~$ mkdir bin
tux@wheezy:~$ mv skript.sh bin/
```

Výhoda tohoto řešení je, že Debian počítá dopředu s tím, že si uživatel vytvoří u sebe v adresáři podadresář `~/bin`. Při dalším přihlášení bude tento adresář automaticky přidán do proměnné `PATH`.

Jakmile se tedy odhlasíte a znovu přihlásíte, můžete spustit skript pouhým zapsáním jeho názvu.

```
tux@wheezy:~$ skript.sh
Skript běží ...
Skript skončil.
```

Proměnné

BASH je skriptovací jazyk, který umožňuje deklarovat proměnné, testovat výrazy, cyklení kódu i deklarace funkcí. Proměnné se deklarují zapsáním názvu a symbolem rovná se. Hodnota se nemusí zadávat, proměnná bude prostě prázdná.

Deklarujme proměnnou `CAS` s hodnotou "5s". Proměnnou pak dosadíme do prvního příkazu `echo` a do příkazu `sleep` (na místo původních 2s).

```
#!/bin/bash

CAS=5s

echo Skript poběží po dobu $CAS ...
sleep $CAS
echo Skript skončil.
```

Výstup programu

Do proměnné lze uložit výstup jiného programu a ten pak použít dál. Takový příkaz se musí uzavřít do speciální strof.

```
#!/bin/bash
```

```
CAS=5s
```

```
DATUM=`date +%d.%m.%Y`
```

```
echo Skript byl spuštěn dne $DATUM  
echo Skript poběží po dobu $CAS ...  
sleep $CAS  
echo Skript skončil.
```

V tomto případě ovšem proměnnou DATUM vůbec nepotřebujeme. Výraz ``date +%d.%m.%Y`` bychom mohli napsat přímo do příkazu `echo`.

```
tux@wheezy:~$ skript.sh  
Skript byl spuštěn dne 25.02.2014  
Skript poběží po dobu 5s ...  
Skript skončil.
```

Parametry při spuštění

Při spuštění lze skriptu předat parametry, se kterými lze následně ve skriptu dál operovat.

Jsou to automaticky dostupné proměnné `$0`, `$1`, `$2`, .. `$9`. Přičemž proměnná `$0` obsahuje název spuštěného skriptu.

Linux v příkazech

Název školy	Vyšší odborná škola obalové techniky a Střední škola, Štětí, příspěvková organizace		
Adresa školky	Kostelní 134, 411 08 Štětí		
IČ	46773509		
Název operačního programu	OP Vzdělávání pro konkurenceschopnost		
Registrační číslo	CZ.1.07/1.5.00/34.1006		
Označení vzdělávacího materiálu	VY_32_INOVACE_21_PSS_418		
Název tématické oblasti (sady)	Operační systémy		
Název materiálu	Linux v příkazech		
Anotace	Materiál představuje souhrn příkazů/programů, včetně krátkého popisu a ukázky v prostředí GNU Linuxu. Obsahuje i konfigurační soubory a nejpoužívanější direktivy.		
Autor	Ivan Pomykacz	Jazyk	český
Očekávaný výstup	Chápe syntaxi příkazů. Vyhledává v referenčních příručkách.		
Klíčová slova	příkazy, syntaxe, použití příkazů		
Druh výukového zdroje	Výklad	Věková skupina žáků	17+
Typ interakce	aktivita	Ročník	3.
Speciální vzdělávací potřeby	žádné		
Zhotoveno, (datum/období)	11.2.2014	Celková velikost	

Obsah

- Jak text používat
- Přehled příkazů
 - Základní příkazy
 - cd
 - cp
 - ls
 - man
 - mkdir
 - mv
 - nano
 - rm
 - rmdir
 - service
 - tar
 - touch
 - unzip
 - Správa uživatelských účtů
 - /etc/group
 - /etc/passwd
 - /etc/shadow
 - groupadd
 - groupdel
 - id
 - passwd
 - su
 - useradd
 - userdel
 - usermod
 - Přístupová práva
 - chgrp
 - chmod
 - chown
 - Nastavení sítě
 - /etc/hosts
 - /etc/network/interfaces
 - /etc/resolv.conf
 - dhclient
 - ip
 - route
 - Správa disků
 - /etc/fstab
 - blkid
 - cfdisk
 - df
 - edquota
 - mkfs
 - mount
 - quotacheck
 - quotaon

- repquota
- umount
- Správa balíčků
 - /etc/apt/sources.list
 - apt-cache
 - apt-get
 - aptitude
 - dpkg

Jak text používat

- Text normálním řezem je třeba opsat (uváděný v syntaxi).
- *Text kurzívou* se nahrazuje požadovanou hodnotou.
- Hranaté závorky `[]` ohraničují možnosti, které jsou volitelné.
- Složené závorky `{}` ohraničují možnosti, ze kterých si musíte vybrat.
- Svislice `|` odděluje jednotlivé možnosti, tzn. je třeba vybrat jednu z uvedených.

Přehled příkazů

Základní příkazy

cd

Změna pracovního adresáře.

syntaxe:

```
cd [cesta]
```

použití:

```
cd /etc
```

cp

Zkopíruje soubor nebo adresář. Při použití přepínače `-p` se zachová vlastnictví souborů (může pouze root). Při použití přepínače `-r` zkopíruje adresář rekurzivně.

syntaxe:

```
cp [-r] [-p] zdroj cíl
```

použití:

```
cp /etc/fstab /etc/fstab.bak
cp -r /etc/ /mnt/backup/
```

ls

Vypíše obsah adresáře. Přepínačem `-l` se zobrazí podrobný výpis. Přepínač `-a` zahrne do výpisu i skryté soubory.

syntaxe:

```
ls [-l] [-a] [cesta]
```

použití:

```
ls
ls -l /etc/
```

man

Vyvolá manuálovou stránku pro zadaný příkaz.

syntaxe:

```
man příkaz
```

použití:

```
man man
```

mkdir

Vytvoří adresář dle zadané cesty. Přepínač lze použít, pokud je třeba vytvořit i nadřazené adresáře.

syntaxe:

```
mkdir [-p] /cesta/k/adresáři
```

použití:

```
mkdir /srv/www
```

```
mkdir -p /srv/data/public
```

mv

Přesune nebo přejmenuje soubor/adresář.

syntaxe:

```
mv zdroj cíl
```

použití:

```
mv /etc/naztaveni.ini /etc/nastaveni.ini  
mv /home/user/zalohy/ /mnt/backups/
```

nano

Textový editor. Jako parametr lze uvést cestu k souboru, který se má editovat. Pokud soubor neexistuje, tak se při uložení vytvoří.

syntaxe:

```
nano [/cesta/k/souboru]
```

použití:

```
nano /etc/fstab  
nano /etc/nonexistent
```

rm

Smazání souboru nebo adresáře (rekurzivně).

syntaxe:

```
rm [-r] [-f] /cesta/k/souboru/nebo/adresari
```

použití:

```
rm /etc/fstab.bak
```

```
rm -r /etc/  
rm -r -f /home/user/temp/
```

rmdir

Smaže prázdný adresář.

syntaxe:

```
rmdir /cesta/k/adresari
```

použití:

```
rmdir /srv/empty/
```

service

Příkaz spustí/zastaví/restartuje systémovou službu (démona).

syntaxe:

```
service služba {start|stop|restart}
```

použití:

```
service gpm start  
service gpm stop  
service gpm restart
```

tar

Příkaz lze použít mj. k rozbalení (přepínač -x) archivu (přepínač -f). Pokud je na archivu použita komprese gzip (obvyčejně koncovka archiv.tar.gz), pak je možné použít přepínač -z, který se postará o dekompresi. Archiv se rozbalí do aktuálního adresáře. Přepínačem -c se naopak archiv vytváří.

syntaxe:

```
tar [-z] [-x] -f /cesta/k/souboru.tar  
tar [-z] [-c] -f /cesta/k/souboru.tar /cesta/k/adresari
```

použití:

```
tar -x -f /mnt/backups/zaloha.tar
tar -z -x -f /mnt/backups/zaloha.tar.gz
tar -c -f /mnt/backups/etc.tar /etc/
tar -z -c -f /mnt/backups/etc.tar.gz /etc/
```

touch

Aktualizuje časovou značku u souboru, pokud tento existuje. Vytvoří prázdný soubor, pokud neexistuje.

syntaxe:

```
touch /cesta/k/souboru
```

použití:

```
touch /tmp/dummy.txt
```

unzip

Rozbalí archiv do aktuálního adresáře.

syntaxe:

```
unzip soubor
```

použití:

```
unzip archiv.zip
```

Správa uživatelských účtů

/etc/group

Soubor udržuje seznam skupin a členství v sekundárních skupinách. Soubor se needituje přímo. Změny v něm lze dosáhnout příkazy `usermod`, `groupadd`.

syntaxe:

```
skupina:x:GID:členové
```

použití:

```
nano /etc/group
```

/etc/passwd

Soubor s uživatelskými účty. Soubor se většinou needituje přímo. Změny v něm lze provádět pomocí příkazu `useradd`, resp. `usermod`.

syntaxe:

```
login:x:UID:GID:info:home:shell
```

použití:

```
nano /etc/passwd
```

/etc/shadow

Soubor obsahuje zašifrovaná hesla uživatelů. Tento soubor se needituje přímo. Změny v něm lze provést pomocí příkazu `passwd`.

syntaxe:

```
login:x:heslo:platnost
```

použití:

```
nano /etc/passwd
```

groupadd

Přidá do systému uživatelskou skupinu.

syntaxe:

```
groupadd skupina
```

použití:

`groupadd zamestnanci`

groupdel

Odstraní ze systému uživatelskou skupinu.

syntaxe:

`groupdel skupina`

použití:

`groupdel zamestnanci`

id

Vypíše informace o aktuálně přihlášeném uživateli (při použití bez parametru) nebo o vybraném uživateli.

syntaxe:

`id [login]`

použití:

`id`
`id root`

passwd

Změní heslo aktuálně přihlášenému uživateli nebo zadanému přihlašovacímu jménu.

syntaxe:

`passwd [login]`

použití:

`passwd`
`passwd franta`

su

Povýšení na superuživatele (root) a nebo na uživatele dle přihlašovacího jména. Uživatel root se může přepnout na lib. uživatele bez znalosti jeho hesla.

syntaxe:

```
su [login]
```

použití:

```
su  
su franta
```

useradd

Založí nový účet do systému. Ve výchozím stavu se nevytváří domovský adresář a není-li zadáno vytvoří se pro uživatele primární skupina pod stejným názvem jako přihlašovací jméno uživatele (*login*). Dalším výchozím nastavením je příkazový shell, který je nastaven na `/bin/sh`, nicméně častěji se běžným uživatelům nastavuje `/bin/bash`, což lze ovlivnit v souboru `/etc/default/useradd`. Přepínač `-g` nastavuje primární skupinu. Přepínač `-G` nastavuje další, dodatečné skupiny. Přepínač `-m` zajistí vytvoření domovského adresáře.

syntaxe:

```
useradd [-m] [-g skupina] [-G skupina[,skupina,...]] [-s shell] login
```

použití:

```
useradd franta  
useradd -m franta  
useradd -m -g zamestnanci -s /bin/bash franta
```

userdel

Odstraní uživatelský účet. Při použití `-r` odstraní i domovský adresář.

syntaxe:

```
userdel [-r] login
```

použití:

```
userdel franta
userdel -f franta
```

usermod

Změní nastavení uživatele. Přepínače mají stejný význam jako u příkazu usermod. Není zde přepínač -m. Přepínač -a zajistí, že v kombinaci s -G nedojde k přepsání stávajících skupin, ale k přidání.

syntaxe:

```
usermod [-g skupina] [-a] [-G skupina[,skupina,...]] [-s shell] login
```

použití:

```
usermod -s /bin/bash franta
usermod -G zalohy franta # nahradí členství ve všech dodatečných skupinách
skupinou zalohy.
usermod -a -G zalohy franta # doplní ke stávajícím členství další skupinu
zalohy.
```

Přístupová práva**chgrp**

Nastaví skupinu na souboru nebo adresáři.

syntaxe:

```
chgrp skupina /cesta/k/souboru
```

použití:

```
chgrp zamestnanci /srv/private
```

chmod

Nastavuje mód - přístupová práva k souboru dvojím způsobem. Symbolicky (rwx) nebo absolutně (osmičkový zápis).

syntaxe:

```
chmod [u=[[+|-]r[+|-]w[+|-]x]],[g=...],[o=...] /cesta/k/souboru
chmod [000-777] /cesta/k/souboru
```

použití:

```
chmod u+r+w+x,g+r-w+x,o=-r-w-x /srv/data/private
chmod 750 /srv/data/private
# výsledek rwxr-x---
```

chown

Nastaví vlastníka, případně i skupinu na adresář nebo soubor.

syntaxe:

```
chown login[:skupina] /cesta/k/souboru
```

použití:

```
chown franta /srv/private
chown franta:zamestnanci /srv/private
```

Nastavení sítě**/etc/hosts**

Mapování jmen na IP adresy. FQDN je plně kvalifikované doménové jméno (existuje-li, jinak se použije rovnou alias).

syntaxe:

```
IPadresa FQDN Alias1 Alias2 AliasN
```

použití:

```
127.0.0.2 frantovo.localhost frantovo
```

/etc/network/interfaces

Konfigurační soubor pro síťová rozhraní. Direktiva "auto" aktivuje rozhraní automaticky při startu. Volby *inet* nebo *inet6* konfiguruje rozhraní pro IPv4, resp. IPV6. Je-li na síti DHCP server, lze využít volbu *dhcp*. Statickou konfiguraci lze vynutit volbou *static*, pak je ovšem nezbytné vyplnit direktivy: *address*, *netmask* a *gateway* (je-li na síti brána). Aktivace konfigurace se provádí vypnutím a spuštěním služby *networking*.

syntaxe:

```
auto rozhraní [, rozhraní[,...]]
iface rozhraní {inet|inet6} {static|dhcp|loopback}
    [address adresa]
    [netmask maska]
    [gateway brána]
```

použití:

```
auto eth0
iface eth0 inet dhcp
```

/etc/resolv.conf

Konfigurace DNS resolveru. Soubor se edituje přímo. V případě konfigurace z DHCP serveru se může soubor automaticky přepsat.

syntaxe:

```
nameserver adresa
domain doména
search doména
```

použití:

```
nano /etc/resolv.conf
```

dhclient

Aktivuje a nakonfiguruje rozhraní dle DHCP serveru. Pokud je tento na síti.

syntaxe:

```
dhclient rozhraní
```

použití:

```
dhclient eth0
```

ip

Zobrazí nebo nastaví konfiguraci síťových rozhraní. Neaktivní rozhraní je třeba aktivovat pomocí `ip link ...`. Přidělení IP adresy lze pomocí `ip addr ...`

syntaxe:

```
ip {link|addr|route}
ip link set {up|down} dev rozhraní
ip addr {add|del} adresa/prefix dev rozhraní
ip addr flush dev rozhraní
ip route
```

použití:

```
ip link # přehled síťových rozhraní a fyzických adres
ip addr # přehled síťových rozhraní a přidělených IP adres
ip route # směrovací tabulka
```

```
ip link set up dev eth0
ip addr add 192.168.1.2/24 dev eth0
ip addr flush dev eth0
```

route

Výpis a konfigurace směrovací tabulky. Přepínač `-n` nebude provádět překlad adres na jména (může být pak rychlejší výpis). Cíl je adresa sítě, do které se dostaneme přes adresu brány (gw). Pro výchozí bránu lze použít klíčové slovo `default`.

syntaxe:

```
route [-n]
route add cíl gw adresa
route del cíl gw adresa
```

použití:

```
route add default gw 10.0.2.2
route add 172.26.34.0 gw 10.0.2.2
```

```
route del 172.26.34.0 gw 10.0.2.2
```

Správa disků

/etc/fstab

Tabulka připojovaných souborových systémů při startu OS.

syntaxe:

```
file system ... zařízení  
mount point ... přípojný-bod  
type ... typfs, typ souborového systému  
options ... volby připojení, obvykle „defaults“  
dump ... dump diskové oblasti, hodnoty: 0-1  
pass ... kontrola diskové oblasti, hodnoty: 0-2
```

použití:

```
nano /etc/fstab
```

blkid

Zobrazí UUID ke všem zařízením, kde je souborový systém.

syntaxe:

```
blkid
```

použití:

```
blkid
```

dfdisk

Spravuje diskové oddíly na daném zařízení.

syntaxe:

```
dfdisk zařízení
```

použití:

```
dfdisk /dev/sda
```

df

Zobrazí využití místa na souborových připojených systémech. Přepínač -h ukáže ve výpise srozumitelnější velikosti.

syntaxe:

```
df [-h]
```

použití:

```
df
```

edquota

Přidělení/nastavení kvót pro uživatele (-u) nebo skupinu (-g).

syntaxe:

```
edquota [-u | -g] {uživatel | skupina}
```

použití:

```
edquota -u franta
```

mkfs

Vytvoří souborový systém na daném zařízení. Přepínačem -t lze nastavit typ souborového systému (ext2, ext4, vfat, ...)

syntaxe:

```
mkfs [-t typfs] zařízení
```

použití:

```
mkfs -t ext4 /dev/sda1
```

mount

Zobrazí seznam všech připojených souborových systémů nebo připojí zařízení se souborým systémem do adresářové struktury. Přípojný bod je tedy adresář někde v adresářové stuktuře, např. /mnt/disk. Má-li zařízení záznam v /etc/fstab, pak je možné použít třetí variantu příkazu mount.

syntaxe:

```
mount  
mount [-t typfs] zařízení přípojný-bod  
mount přípojný-bod
```

použití:

```
mount -t ext4 /dev/sda1 /mnt/disk
```

quotacheck

Vytvoření (přepínač -c) nebo kontrola kvótových souborů pro uživatele (-u) nebo skupiny (-g). Použití příkazu je podmíněno nainstalovaným balíčkem quota.

syntaxe:

```
quotacheck [-c] [-g] [-u] přípojný-bod
```

použití:

```
quotacheck -c -u -g /mnt/kvoty
```

quotaon

Zapne kvóty na daném zařízení (přípojném bodě). Opakem příkazu je quotaoff.

syntaxe:

```
quotaon přípojný-bod
```

použití:

quotaon /mnt/kvoty

repquota

Zobrazí souhrn stavu obsazeného místa a přidělených kvót pro uživatele (-u) nebo skupiny (-g). Není-li uvedeno, je výchozí volba -u.

syntaxe:

repquota [-u | -g] *přípojný-bod*

použití:

repquota -u /mnt/kvoty

umount

Odpojí souborový systém.

syntaxe:

umount *zařízení*
umount *přípojný-bod*

použití:

umount /mnt/disk

Správa balíčků

/etc/apt/sources.list

Schéma je zpravidla typ aplikačního protokolu, běžně http. Verze je kódové jméno pro verzi používaného systému (např. Debian 6 má kódové jméno squeeze a Debian 7 má kódové jméno wheezy). Větev je označení skupiny balíčků, např. pod společnou licenci (jsou to názvy adresářů na dané URL adrese v adresáři pool, např. main, free, nonfree).

syntaxe:

deb *schema://adresa verze větev [větev [...]]*

použití:

```
deb http://ftp.cz.debian.org/debian/ wheezy main
```

apt-cache

Příkaz pracuje s balíčkovací keší. Je možné vyhledávat balíčky (search) a nebo získat detailnější informace (show).

syntaxe:

```
apt-cache {show|search} balíček
```

použití:

```
apt-cache search editor
```

apt-get

Nástroj pro aktualizaci cache balíčků (update) nebo na instalaci a odstranění balíčků.

syntaxe:

```
apt-get update  
apt-get [install|remove|purge] balíček [balíček...]
```

použití:

```
apt-get install apm
```

aptitude

TUI/CLI nástroj pro správu balíčků

syntaxe:

```
aptitude
```

použití:

```
aptitude  
aptitude install gpm
```

dpkg

Nízkoúrovňový nástroj pro instalaci balíčků, které nejsou v repozitáři.

syntaxe:

```
dpkg -i /cesta/k/balíčku
```

použití:

```
dpkg -i razor.deb
```

Grafické prostředí

Název školy	Vyšší odborná škola obalové techniky a Střední škola, Štětí, příspěvková organizace		
Adresa školky	Kostelní 134, 411 08 Štětí		
IČ	46773509		
Název operačního programu	OP Vzdělávání pro konkurenceschopnost		
Registrační číslo	CZ.1.07/1.5.00/34.1006		
Označení vzdělávacího materiálu	VY_32_INOVACE_21_PSS_419		
Název tematické oblasti (sady)	Operační systémy		
Název materiálu	Grafické prostředí		
Anotace	Text představuje přehled grafických prostředí ve světě Linuxu. V textu je popsán rozdíl mezi desktopovým prostředím a správcem oken. V praktické části popisuje instalaci vybraného prostředí XFCE a předkládá ukázky některých programů, které do určité míry suplují činnost příkazů v shellu.		
Autor	Ivan Pomykacz	Jazyk	český
Očekávaný výstup	Instaluje grafické prostředí. Používá grafické aplikace jako terminál, textový editor, správce diskových oblastí, nastavení sítě, atp.		
Klíčová slova	GUI, DE, WM, grafické prostředí, desktop, správce oken		
Druh výukového zdroje	Výklad	Věková skupina žáků	17+
Typ interakce	aktivita	Ročník	3.
Speciální vzdělávací potřeby	žádné		
Zhotoveno, (datum/období)	25.3.2014	Celková velikost	

Obsah

- Instalace
 - Po instalaci
- Nástroje pro správu
 - Terminál
 - Textový editor
 - Správce souborů
 - Nastavení sítě
 - Správa diskových oblastí
 - Správa balíčků
 - Správce procesů
 - Správa archivů

Grafické prostředí v Linuxu existuje. Tou strohou a možná až sklesle znějící větou nechci budít nějaký můj negativní dojem z grafického prostředí v Linuxu. Ale popořadě.

Linux je především nástroj. Na každý nástroj je třeba naučit se hrát. Některé nástroje hrají sami hned, jak si je pořídíte (tím myslím MS Windows nebo Mac OS). Stačí se naučit pár akordů. Pokud ovšem budete chtít, aby váš nástroj vydával jiný zvuk, pravděpodobně neuspějete (ať už z licenčních důvodů nebo jiných).

Naučit se hrát na Linux je poněkud obtížnější. Nabízí možná až moc variací zvuků, které může vydávat. Uživatelé začátečníci mají problém se v této směsici zorientovat.

Ale dost už analogií. V Linuxu existuje několik grafických prostředí. Samotný termín je možná i nepřesný, možná by bylo lepší jej formulovat jako *desktopová grafická prostředí*. Jak desktop (DE - Desktop Environment) lze označit několik součástí, které jej utváří. Jsou to především: grafický subsystém (X Window System), správce oken (WM - Window Manager), grafické prostředí (GUI - Graphical User Interface) a nakonec aplikace, o které nám jde především.

Desktopová prostředí v Linuxu (podle abecedy):

- Cinnamon
- Elive
- Gnome
- KDE
- LXDE
- LXQt
- MATE
- Pantheon (Elementary OS)
- XFCE
- Unity (Ubuntu)

Pokud je nějaké prostředí svázáno se svou distribucí, je tato uvedena v závorce. Uvedený seznam jsou plnohodnotné DE. Běžný *linuxák* si ovšem vystačí i s jen samotným WM:

- Blackbox
- Enlightenment
- Fluxbox
- FVWM
- i3

- IceWM
- Window Maker

Až na pár výjimek, můžete ve své distribuci používat téměř jakékoli DE nebo WM (v závislosti na HW).

Instalace

Pokud bychom chtěli do naší stávající instalace nainstalovat plnohodnotné desktopové prostředí (tj. se všemi obvyklými aplikacemi), pak můžeme nainstalovat balíček `task-xfce-desktop` pro DE XFCE.

My si vystačíme pouze s balíčkem `xfce4`.

```
root@wheezy:~# apt-get install xfce4
```

Nyní můžeme spustit (jako běžný uživatel) grafické prostředí příkazem `startx`

Po instalaci

Moc aplikací na nás v grafickém prostředí ale čekat nebude, neboť jsme udělali opravdu minimalistickou instalaci. Doinstalujme si proto nějaký základní software: prohlížeč pdf souborů (mupdf), webový prohlížeč (iceweasel), textový editor (mousepad).

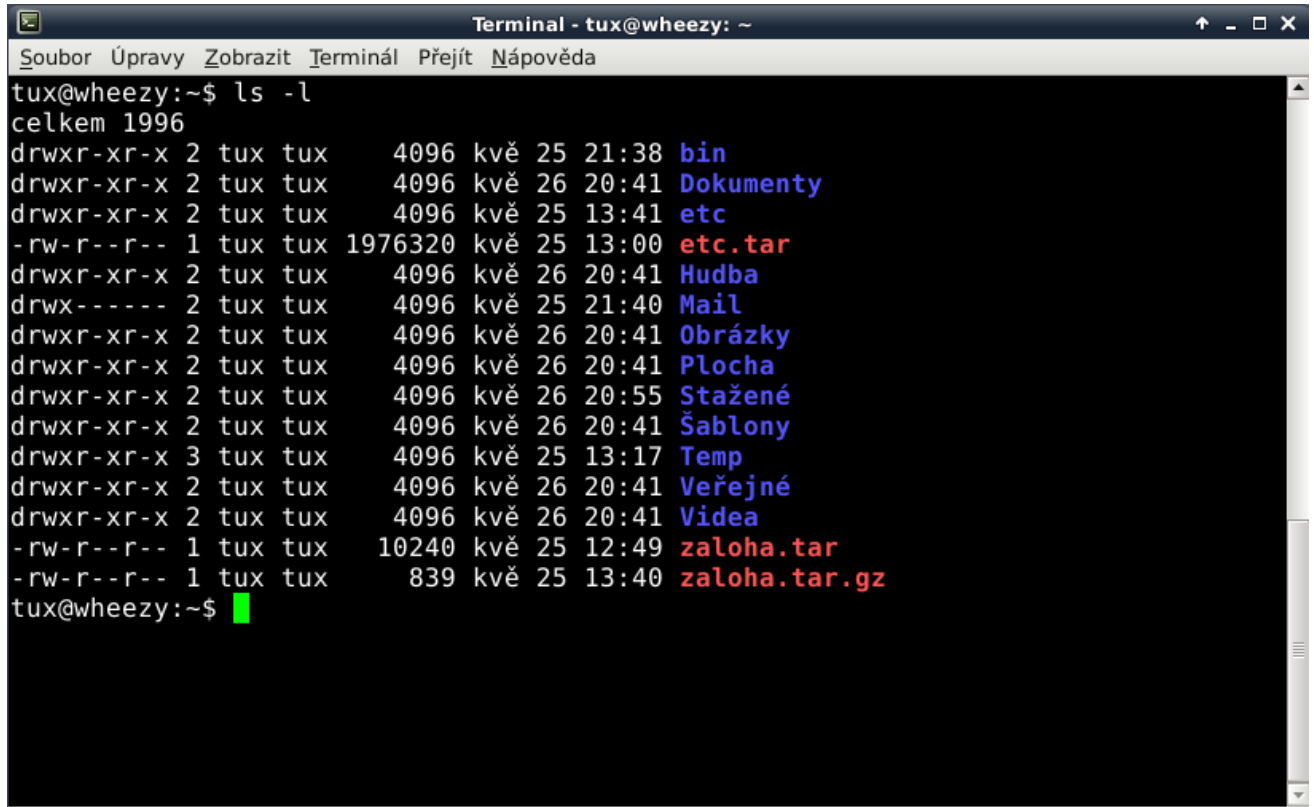
```
root@wheezy:~# apt-get install mupdf iceweasel iceweasel-ll0n-cs mousepad
```

Nástroje pro správu

Terminál

V grafickém prostředí je k dispozici samozřejmě i terminál, každé DE má svojí vlastní implementaci.

```
root@wheezy:~# apt-get install xfce4-terminal
```

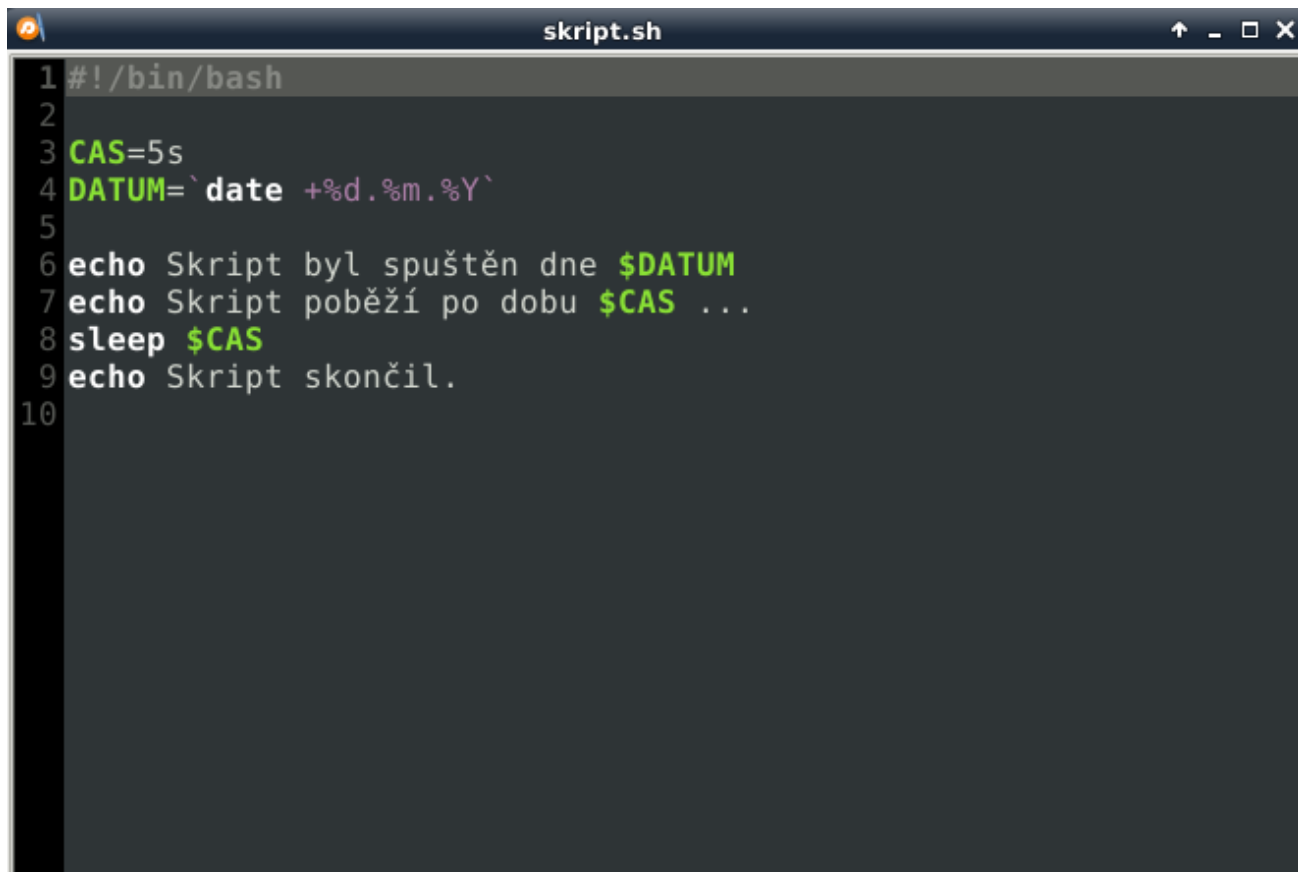


```
Terminal - tux@wheezy: ~
Soubor Úpravy Zobrazit Terminál Přejít Nápověda
tux@wheezy:~$ ls -l
celkem 1996
drwxr-xr-x 2 tux tux 4096 kvě 25 21:38 bin
drwxr-xr-x 2 tux tux 4096 kvě 26 20:41 Dokumenty
drwxr-xr-x 2 tux tux 4096 kvě 25 13:41 etc
-rw-r--r-- 1 tux tux 1976320 kvě 25 13:00 etc.tar
drwxr-xr-x 2 tux tux 4096 kvě 26 20:41 Hudba
drwx----- 2 tux tux 4096 kvě 25 21:40 Mail
drwxr-xr-x 2 tux tux 4096 kvě 26 20:41 Obrázky
drwxr-xr-x 2 tux tux 4096 kvě 26 20:41 Plocha
drwxr-xr-x 2 tux tux 4096 kvě 26 20:55 Stažené
drwxr-xr-x 2 tux tux 4096 kvě 26 20:41 Šablony
drwxr-xr-x 3 tux tux 4096 kvě 25 13:17 Temp
drwxr-xr-x 2 tux tux 4096 kvě 26 20:41 Veřejné
drwxr-xr-x 2 tux tux 4096 kvě 26 20:41 Video
-rw-r--r-- 1 tux tux 10240 kvě 25 12:49 zaloha.tar
-rw-r--r-- 1 tux tux 839 kvě 25 13:40 zaloha.tar.gz
tux@wheezy:~$
```

Textový editor

Textový editor, který mimo jiné zvýrazňuje syntaxi.

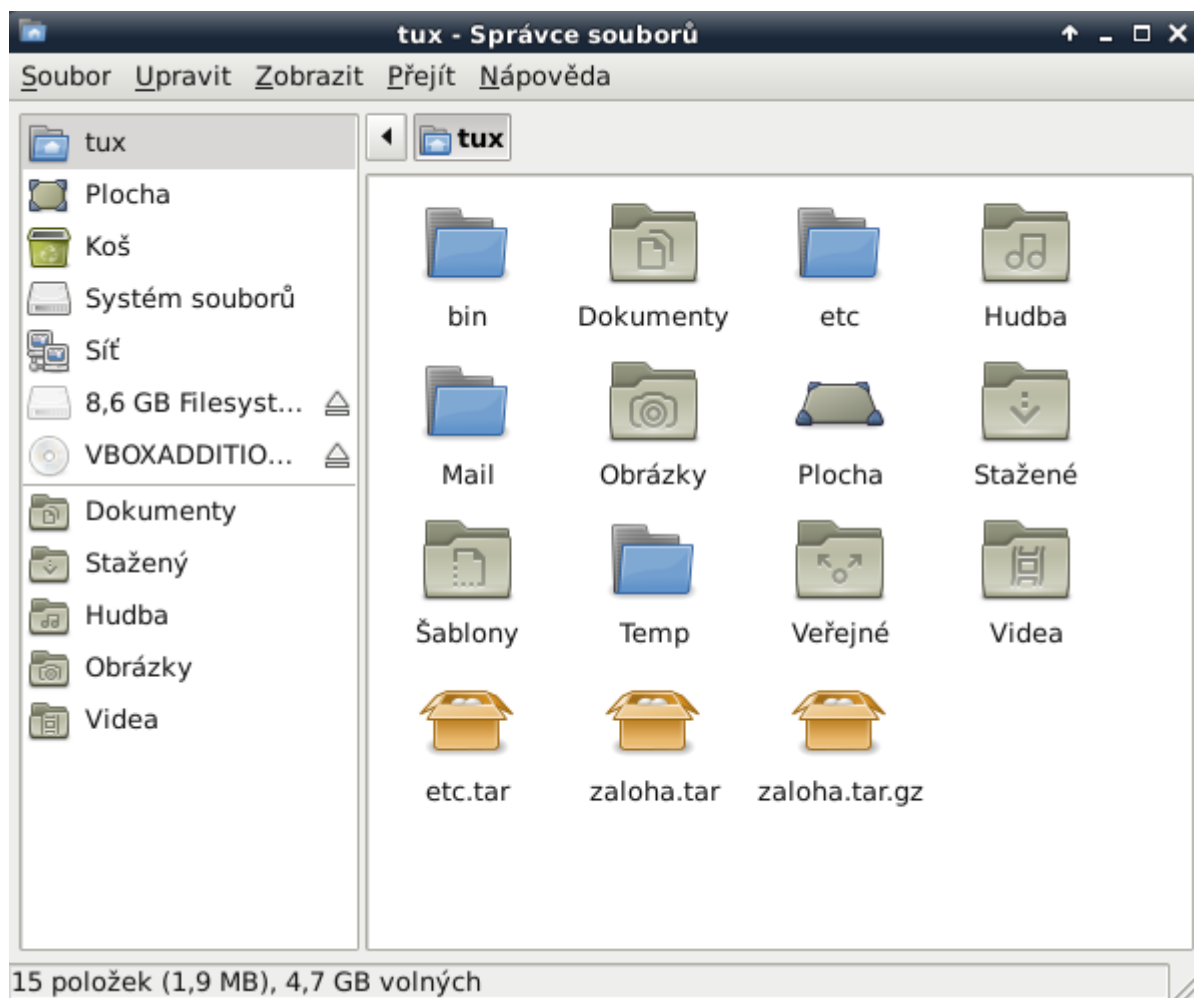
```
root@wheezy:~# apt-get install scribes
```



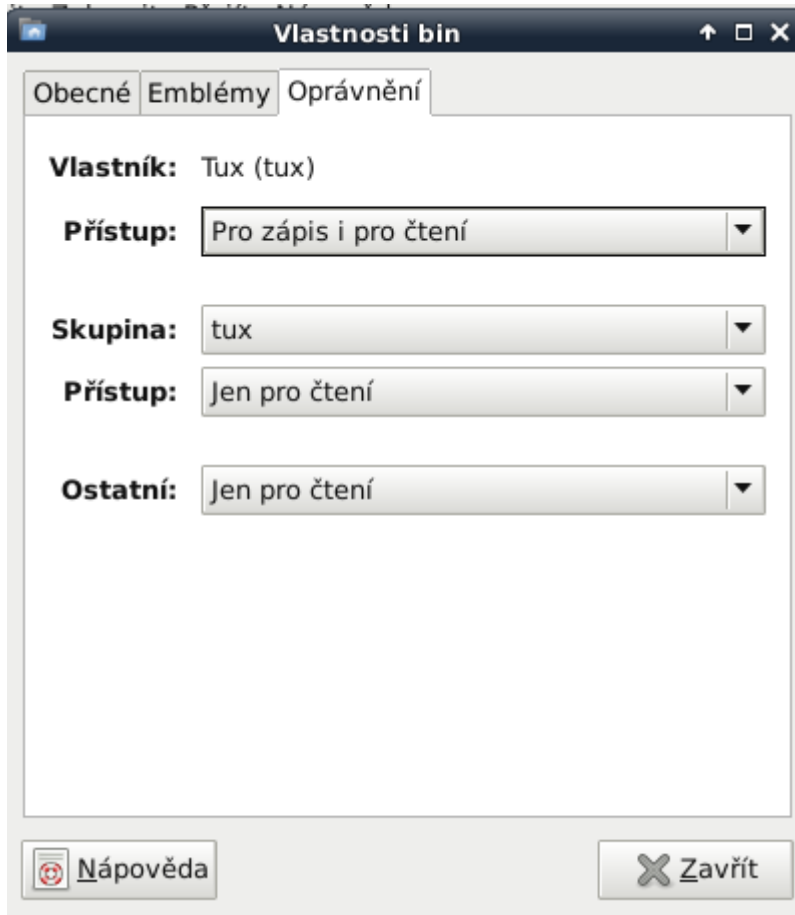
```
1 #!/bin/bash
2
3 CAS=5s
4 DATUM=`date +%d.%m.%Y`
5
6 echo Skript byl spuštěn dne $DATUM
7 echo Skript poběží po dobu $CAS ...
8 sleep $CAS
9 echo Skript skončil.
10
```

Správce souborů

Již nainstalovaný **thunar** má podobné chování jako např. *Průzkumník* z MS Windows.



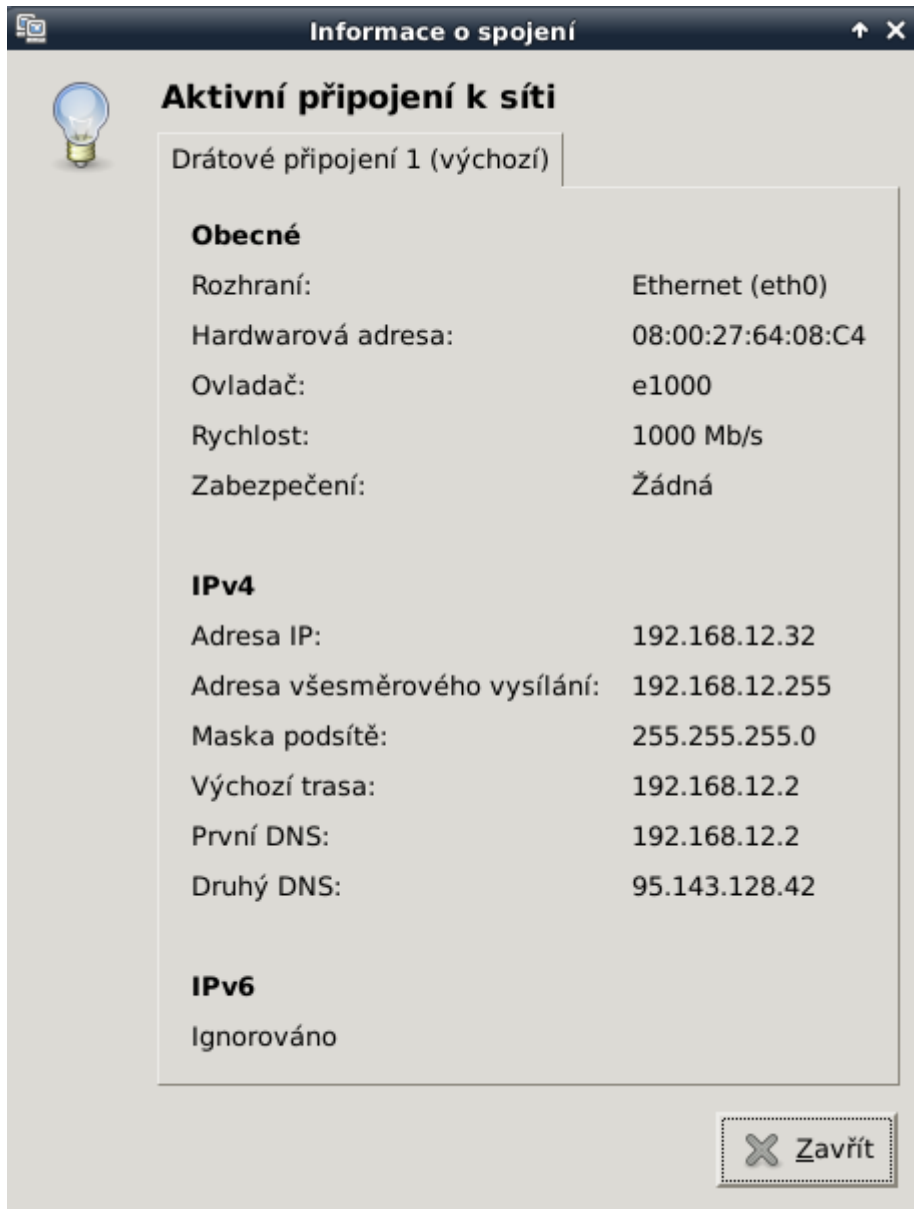
Součástí správce souborů je i nastavování přístupových práv.



Nastavení sítě

XFCE nemá svůj vlastní nástroj pro nastavení sítě, ale lze použít např. ten pro Gnome. Tady pozor! Network-manager si nerozumí s `/etc/network/interfaces` (bylo by dobré změny v tomto souboru zrušit).

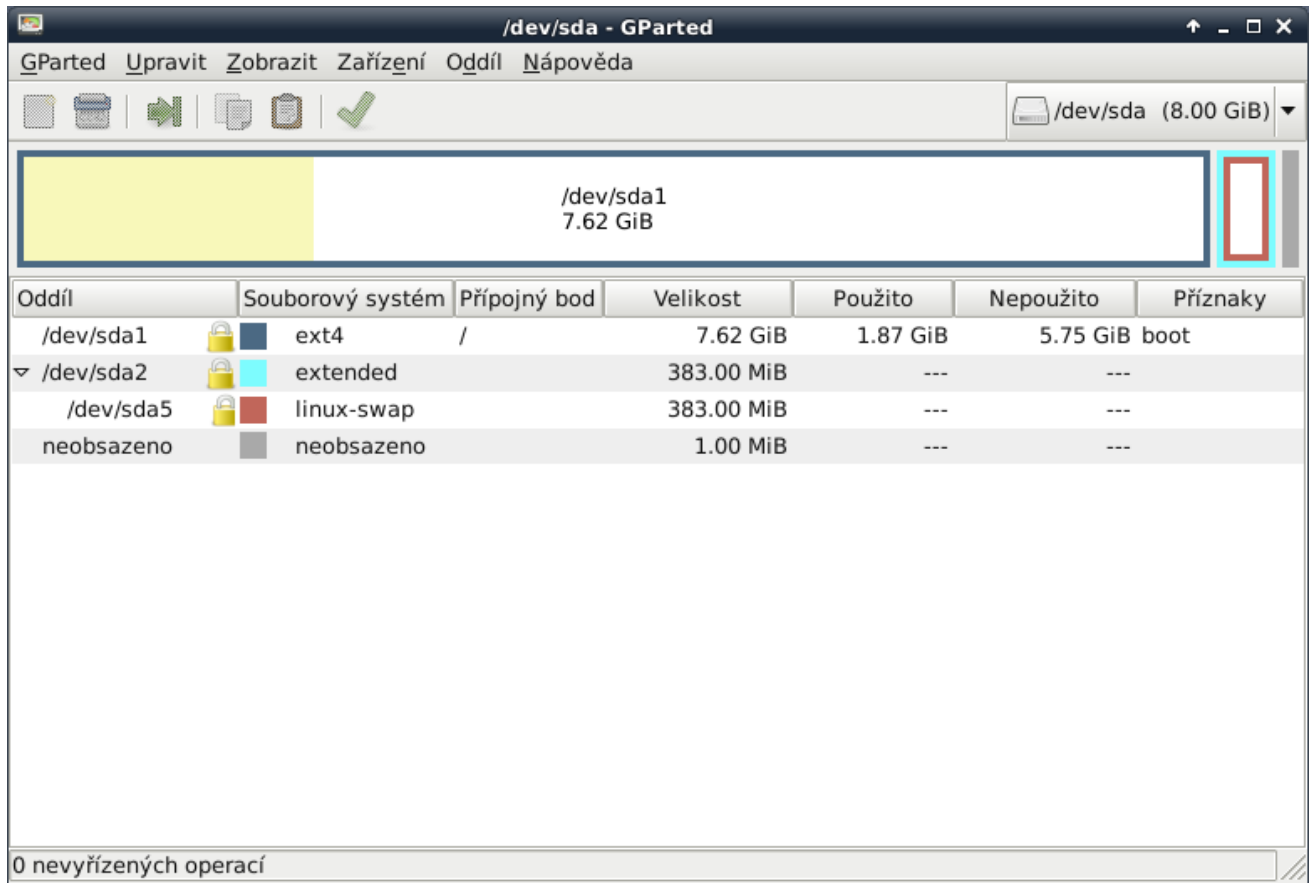
```
root@wheezy:~# apt-get install network-manager-gnome
```



Správa diskových oblastí

Pro správu diskových oblastí lze doinstalovat populární gparted.

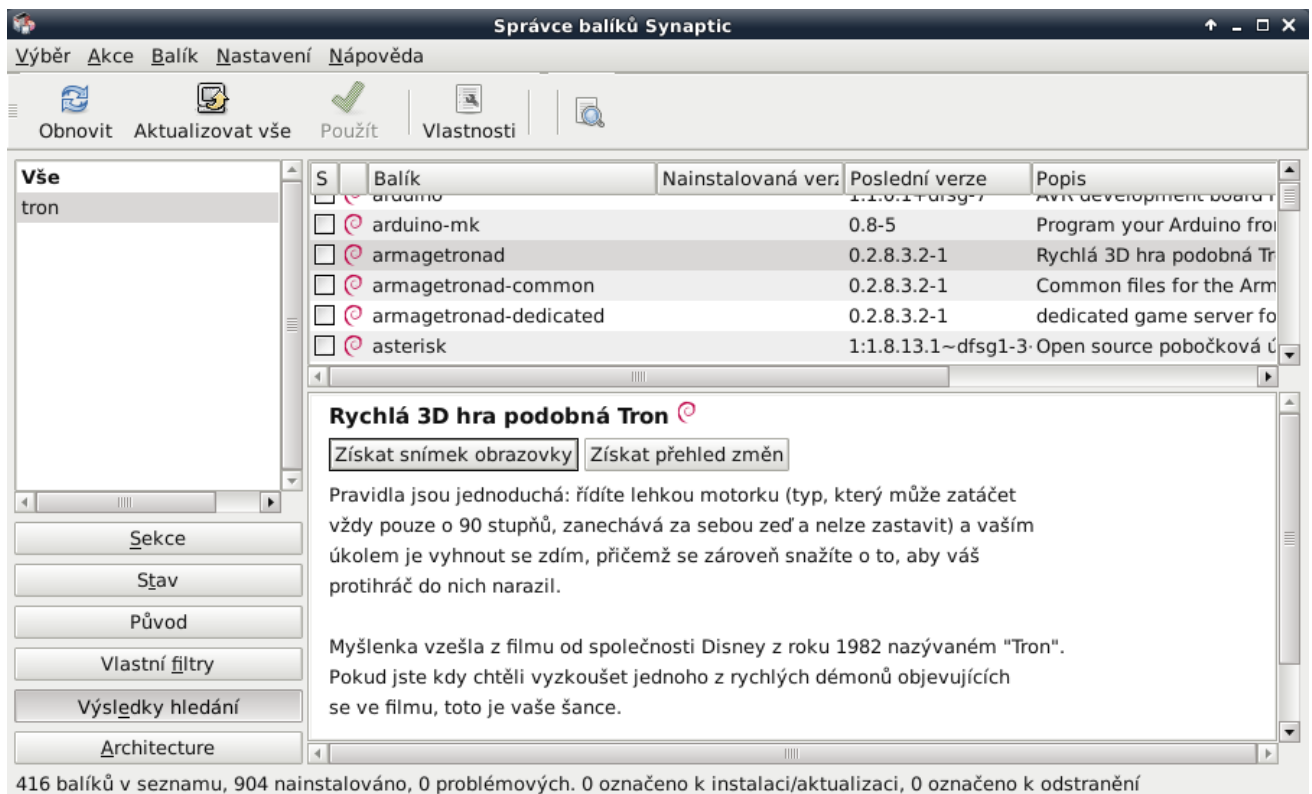
```
root@wheezy:~# apt-get install gparted
```



Správa balíčků

Balíčky a zdroje repozitářů lze spravovat přes **synaptic**.

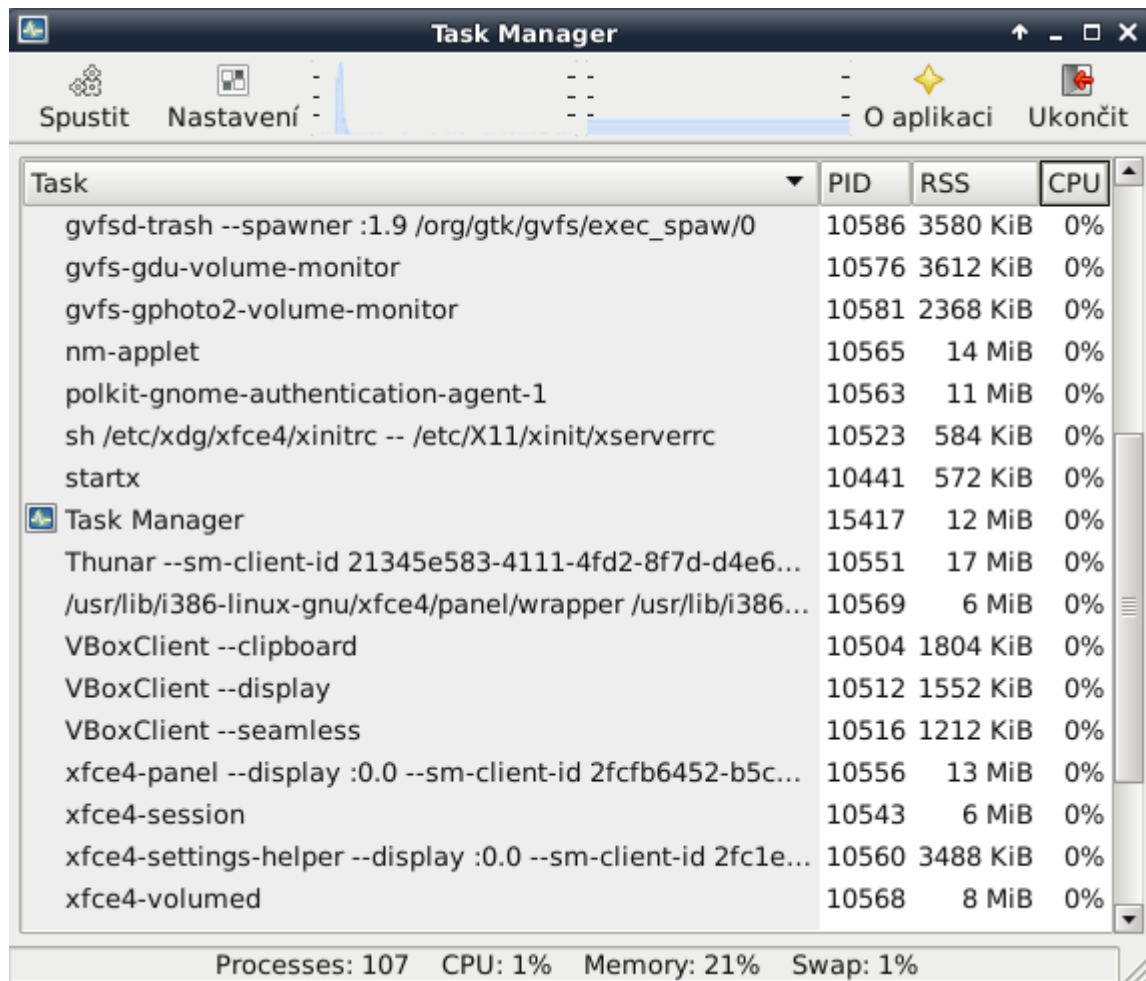
```
root@wheezy:~# apt-get install synaptic
```




Správce procesů

Jednoduchý monitor běžících procesů.

```
root@wheezy:~# apt-get install xfce4-taskmanager
```



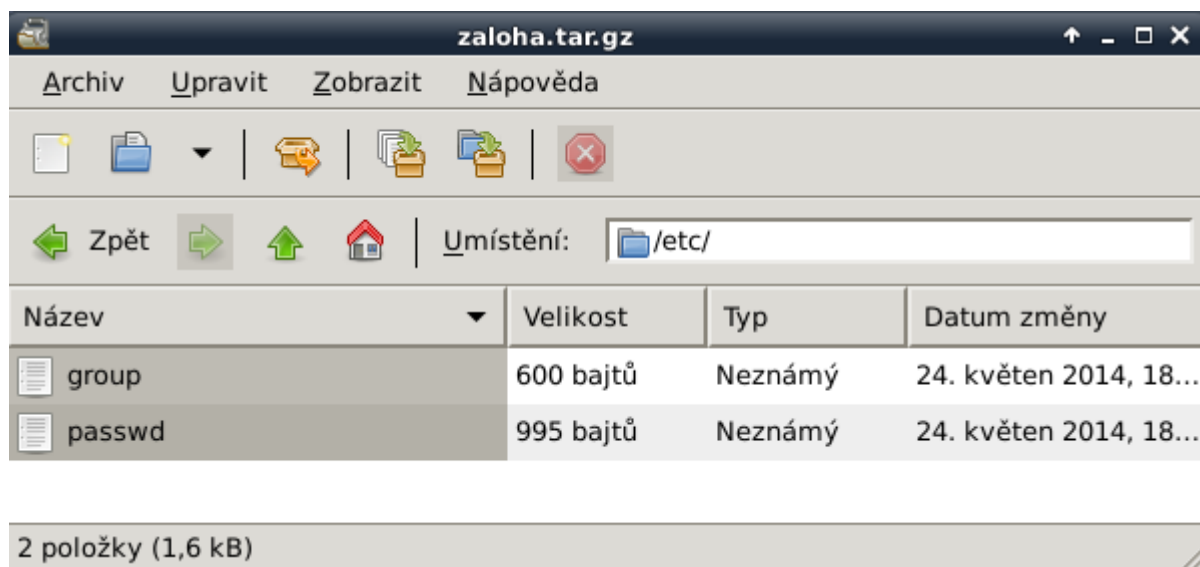
Task	PID	RSS	CPU
gvfsd-trash --spawner :1.9 /org/gtk/gvfs/exec_spaw/0	10586	3580 KiB	0%
gvfs-gdu-volume-monitor	10576	3612 KiB	0%
gvfs-gphoto2-volume-monitor	10581	2368 KiB	0%
nm-applet	10565	14 MiB	0%
polkit-gnome-authentication-agent-1	10563	11 MiB	0%
sh /etc/xdg/xfce4/xinitrc -- /etc/X11/xinit/xserverrc	10523	584 KiB	0%
startx	10441	572 KiB	0%
 Task Manager	15417	12 MiB	0%
Thunar --sm-client-id 21345e583-4111-4fd2-8f7d-d4e6...	10551	17 MiB	0%
/usr/lib/i386-linux-gnu/xfce4/panel/wrapper /usr/lib/i386...	10569	6 MiB	0%
VBoxClient --clipboard	10504	1804 KiB	0%
VBoxClient --display	10512	1552 KiB	0%
VBoxClient --seamless	10516	1212 KiB	0%
xfce4-panel --display :0.0 --sm-client-id 2fcfb6452-b5c...	10556	13 MiB	0%
xfce4-session	10543	6 MiB	0%
xfce4-settings-helper --display :0.0 --sm-client-id 2fc1e...	10560	3488 KiB	0%
xfce4-volumed	10568	8 MiB	0%

Processes: 107 CPU: 1% Memory: 21% Swap: 1%

Správa archivů

Poněkud závislostmi překypělý `file-roller` lze používat k rozbalování i vytváření archivů.

```
root@wheezy:~# apt-get install file-roller
```

Virtualizace

Název školy	Vyšší odborná škola obalové techniky a Střední škola, Štětí, příspěvková organizace		
Adresa školky	Kostelní 134, 411 08 Štětí		
IČ	46773509		
Název operačního programu	OP Vzdělávání pro konkurenceschopnost		
Registrační číslo	CZ.1.07/1.5.00/34.1006		
Označení vzdělávacího materiálu	VY_32_INOVACE_21_PSS_420		
Název tématické oblasti (sady)	Operační systémy		
Název materiálu	Virtualizace		
Anotace	Text předkládá jednu z možností virtualizace počítače. Jeho součástí je tutoriál na tvorbu virtuálního stroje, do kterého lze následně instalovat operační systém.		
Autor	Ivan Pomykacz	Jazyk	český
Očekávaný výstup	Vytváří virtuální stroj s danými parametry pro hostovaný operační systém (CPU, RAM, HDD, NIC).		
Klíčová slova	VM, virtuální stroj, VirtualBox		
Druh výukového zdroje	Výklad	Věková skupina žáků	17+
Typ interakce	aktivita	Ročník	3.
Speciální vzdělávací potřeby	žádné		
Zhotoveno, (datum/období)	16.09.2013	Celková velikost	

Obsah

- [Jak virtualizovat](#)
 - [HW nároky na virtualizaci](#)
 - [Virtualizační software](#)
 - [Kde získat VirtualBox](#)
- [Nový virtuální stroj](#)

Jak virtualizovat

V tomto textu se budeme zabývat pouze tzv. plnou virtualizací. Nemá teď smysl řešit, co je to ta plná virtualizace, každopádně by bylo vhodné si ujasnit už na začátku, že jsou různé druhy virtualizací.

U tohoto typu virtualizace je výhodou, že žádným způsobem neohrozí váš stávající systém. Jediné, co budeme muset obětovat, je pár GB (2-16 GB) místa na disku v závislosti na tom, jaký systém budeme virtualizovat. Co budeme potřebovat:

- funkční počítač (rozuměj počítač, kde už nějaký OS běží);
- virtualizační software (qemu, virtualbox, vmware, ms virtual machine);
- instalační médium operačního systému, který chceme virtualizovat.

HW nároky na virtualizaci

U plné virtualizace nám postačí následující minimální požadavky na hardware:

- Procesor kompatibilní s x86 architekturou (jakýkoliv AMD nebo Intel procesor);
- 1 GB RAM je doporučené minimum (i když virtualizovat lze i s 512 MB);
- Místo na disku - odvíjí se od virtualizovaného OS - jak již bylo řečeno v úvodu: 0 až řádově GB volného místa.

Virtualizační software

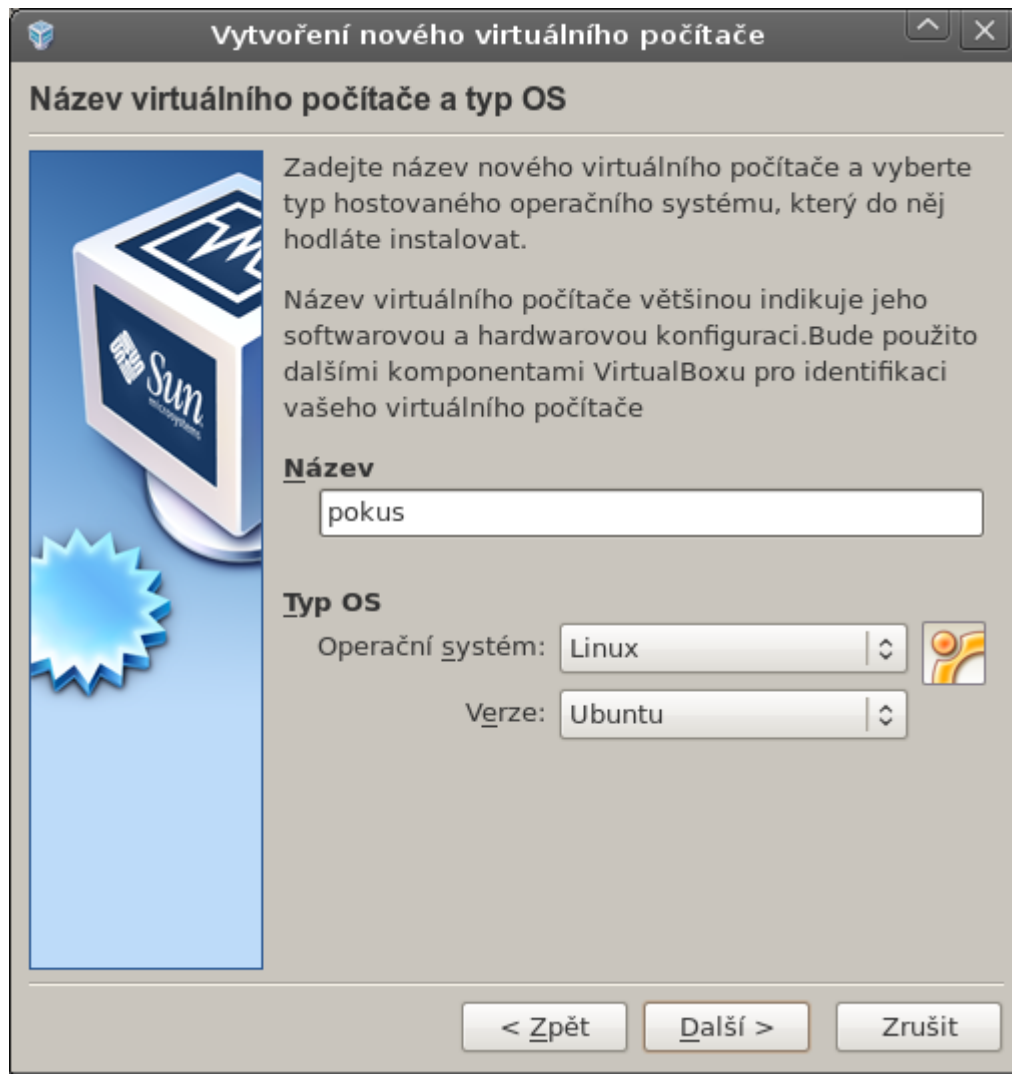
Existuje několik programů pro plnou virtualizaci, např. qemu, virtualbox, vmware nebo ms virtual machine. Ve škole je předinstalován software od firmy Oracle (dříve Sun) - VirtualBox (www.virtualbox.org).

Kde získat VirtualBox

Na stránkách virtualbox.org je sekce Downloads, odkud je možné stáhnout poslední verzi VirtualBoxu pro váš operační systém - tzn. systém, kam budete VirtualBox instalovat. (Pokud pracuju s MS Windows, stáhnu si verzi pro MS Windows - nikoli však ~~mám Ms Windows a chei virtualizovat Linux, proto si stáhnu verzi pro Linux.~~)

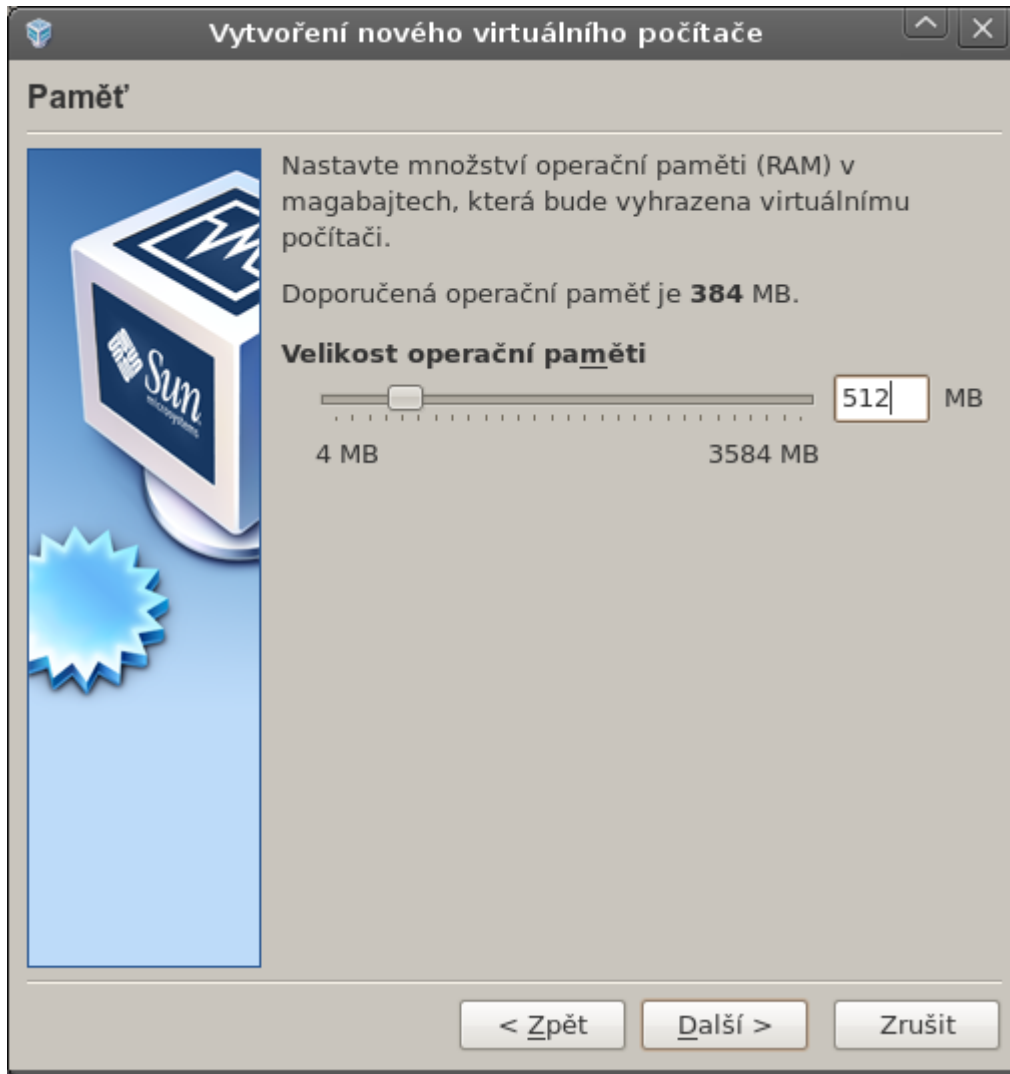
Nový virtuální stroj

VirtualBox má, ostatně i jako ostatní programy stejného typu, velice jednoduché uživatelské rozhraní (GUI). Virtuální stroj lze však vytvořit i bez něho, ale pro názornost GUI využijeme. Pro vytvoření nového virtuálního stroje se po kliknutí na příslušné tlačítko zobrazí průvodce novým virtuálním strojem.



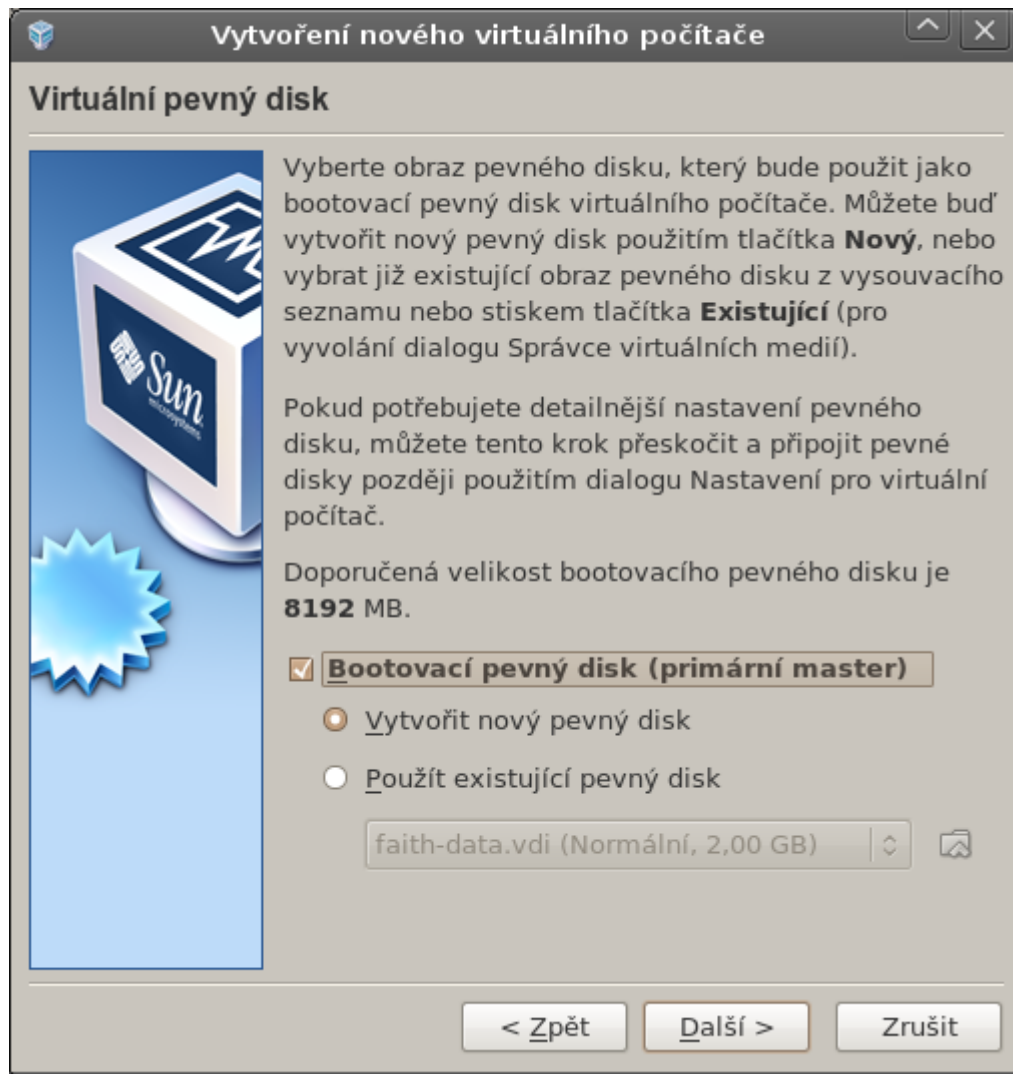
Nemá smysl popisovat všechny volby, následuje jen chronologický výběr těch důležitých.

Přiřazení operační paměti - mějte na paměti, že virtuální stroj ubírá pro virtuální RAM paměť z vaší fyzické RAM, takže zbytečně hodnotu RAM nenavysuňte, aby to váš hostitelský systém ustál. 256 MB je doporučené minimum, 512 MB postačí na všechno, co kdy ve škole budeme dělat.

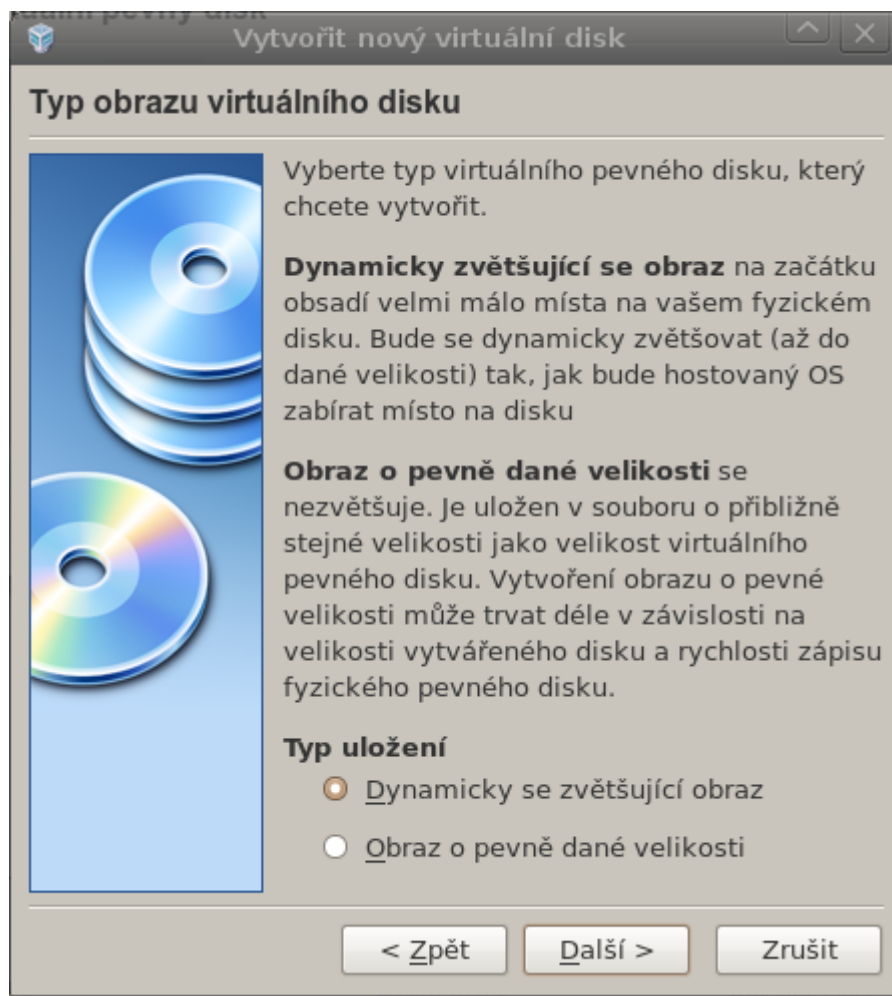


Pokud budete chtít systém nainstalovat, musíte do virtuálního stroje připojit virtuální pevný disk. Tento virtuální disk je ve skutečnosti jeden (nebo více) souborů na vašem pevném disku. Tyto soubory mají příponu *.vdi. To ovšem víte vy, virtualizovaný systém si myslí, že pracuje s normálním pevným diskem.

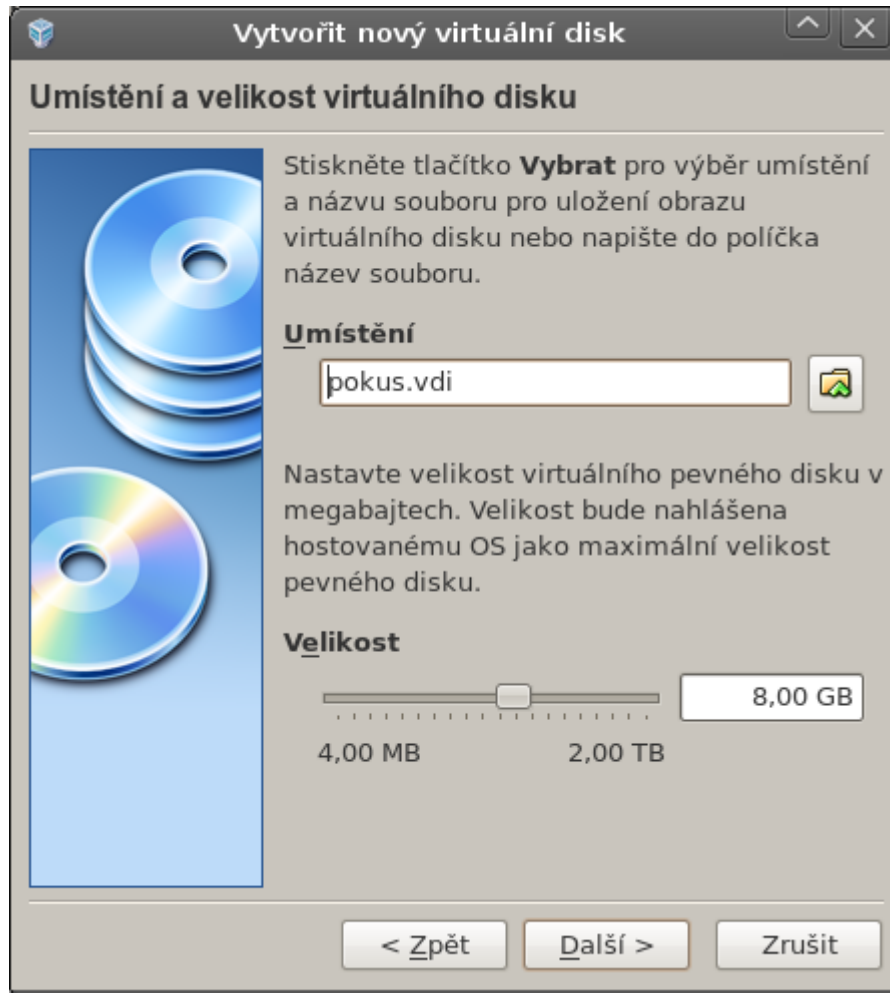
Pokud z tohoto disku budete zavádět operační systém, měl by být označen jako bootovací.



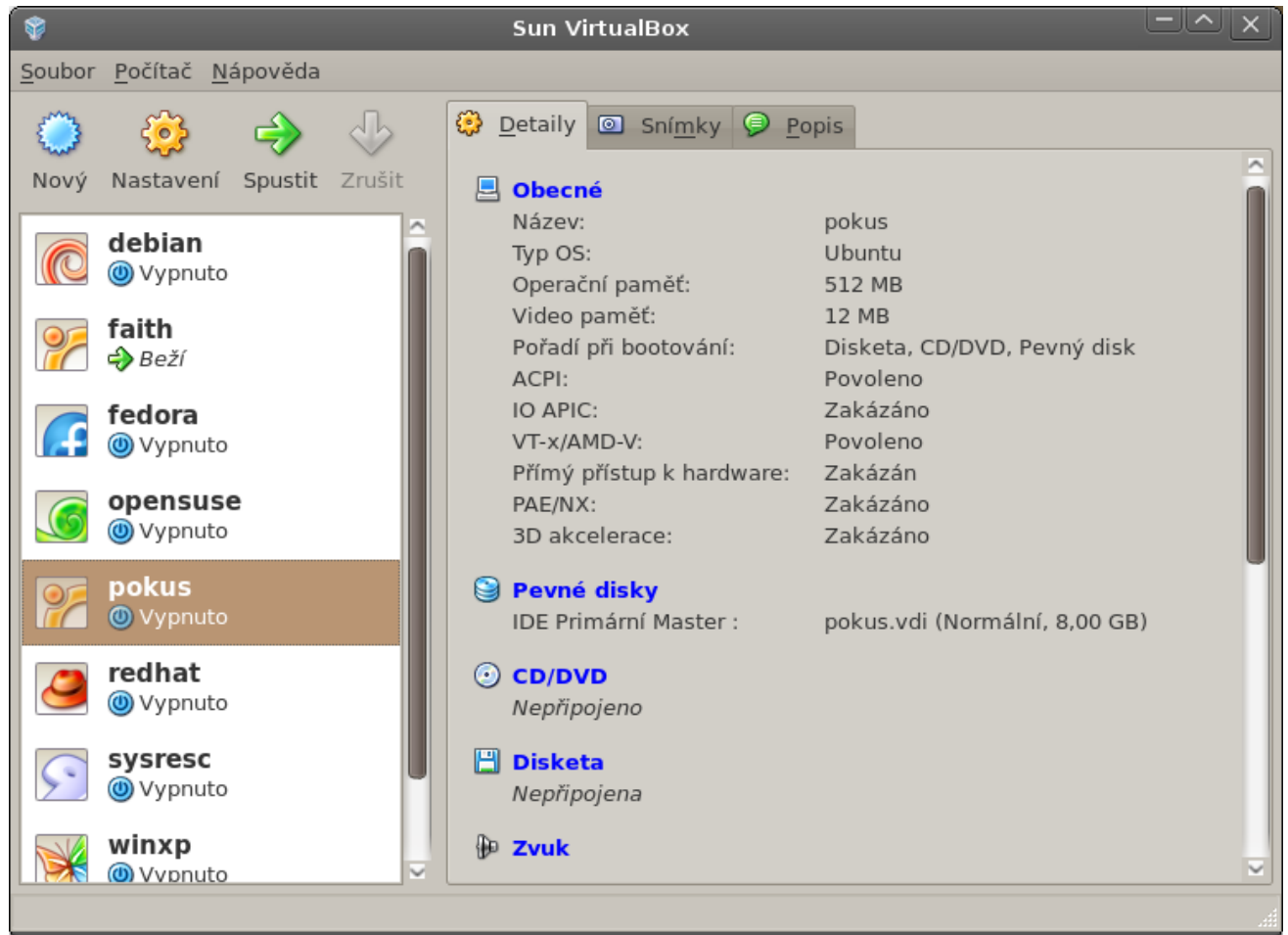
Pro testovací účely vám postačí dynamicky se zvětšující obraz.



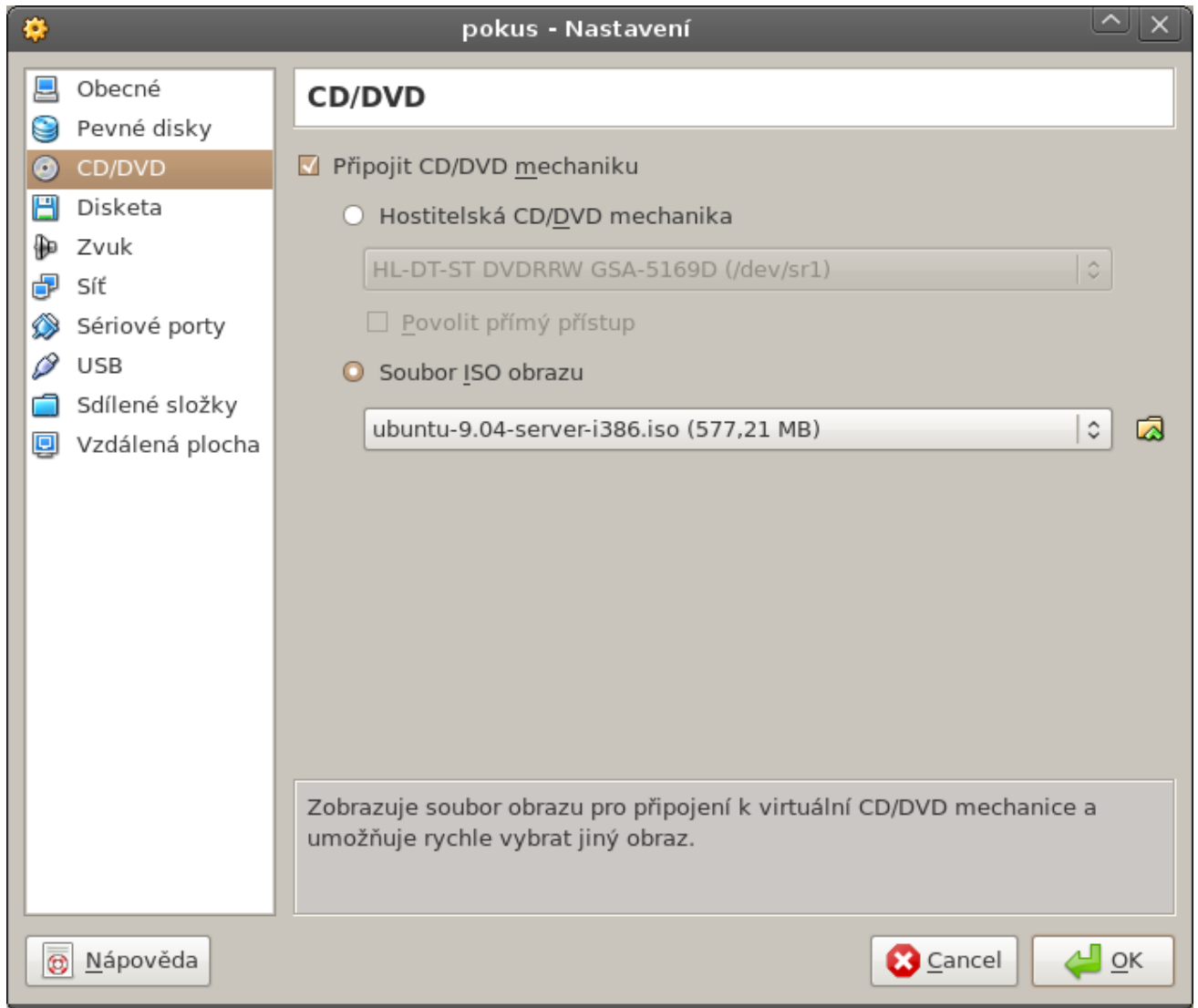
Pokud chcete, můžete změnit i umístění, kde se má soubor s virtuálním diskem nacházet. S velikostí to opět nepřehánějte. Pro naše potřeby nebudeme potřebovat disk s více jak 4 GB.



Po skončení průvodce zbývá jen připojit médium CD nebo ISO obraz s instalací linuxu. To provedete např. výběrem virtuálního stroje (u mě je to "pokus") a kliknutím na "Nastavení" v horní liště.



Ve výběrovém dialogu jsem vybral soubor ISO obrazu, protože jej mám uložený na disku (není třeba jej tedy vypalovat).



Pak už stačí virtuální stroj spustit a ...

