



Vyšší odborná škola obalové techniky
a střední škola, Štětí

Digitální učební materiály

Operační systémy - Linux

Ivan Pomykacz



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.

Operační systémy

Linux

Obsah

- Základní příkazy

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 tématické 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 pravém 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 vypsat 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 spuštění 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?