

Lehký úvod do tajů příkazové řádky



Ve všech operačních systémech je možno pracovat i s příkazovou řádkou. Ne všude se to však stále tolik používá. Avšak v Linuxu je i nadále velmi populární, a je více než pravděpodobné, že se s ní také setkáte ... Příkazová řádka (v Unixech častěji označována jako konzole či terminál anebo shell) je místo pro zadávání příkazů. Příkazy nejsou nic jiného než jednotlivé programy, které zpravidla vykonají svou činnost a ukončí se. Dalším důležitým pojmem je **interpret**. To je program, který interpretuje příkazy = volá ony programy a obvykle umí i další věci. Nejpoužívanějším interpretem v Linuxu je BASH. Sluší se upozornit, že Ubuntu, stejně jako všechny Unixy rozlišuje velká a malá písmena (tj. soubor a SOUBOR není to samé). Termín **prompt** označuje to co vidíme jako `botanicus@ubuntu:~$` či `root@ubuntu:~#` a podobně. Jde o to, že když je prompt volný, můžete zadávat příkazy, když volný není, tak musíte čekat. 😊 Nebo rovněž můžete pracovat s programem - takto se chovají interaktivní programy.

Doplňování

Elementární doplňování bashe



Bash umí doplňovat názvy příkazů. To znamená, že když napíšete první písmena příkazu a zmáčknete tabulátor, tak se příkaz doplní. Pokud však bylo více možností, je třeba tabulátor zmáčknout ještě jednou, vypíše se vám možnosti. Třeba dáte `set` a zmáčknete tabulátor, nic se nestane, zmáčknete jej podruhé a pak to vypadá takto:

```
botanicus@ubuntu:~$ set
set          setlogcons  set_prefix  setvesablank
setkeycodes setmetamode setsid       setxkbmap
setleds     setpci      setterm
```

Teprve když dopíšete ještě `t`, doplní se vám `setterm`. Ale bash umí víc. Umí také doplňovat názvy souborů. Když dáte `cat /home/botanicus/nejaky_libovolny_textovy_soubor`, lze napsat `cat /h` tabulátor `b` tabulátor, `n` tabulátor a vše ostatní se doplní. Pochopitelně za předpokladu, `v /` nemám nic jiného od `h` (jako že to by být nemělo), že neexistuje jiný uživatel od `b` a že ve své domovské složce nemám nic jiného od `n`. Pokud ano, je třeba doplnit alespoň tolik znaků, kolik mají tyto soubory shodné plus jeden navíc - tím se to odliší a bash bude moci název doplnit. Někdy se může stát, že budete mít soubory, které jsou svými prvými znaky shodné, potom je několik znaků rozdílných - třeba soubory `iva` a `iveta`. Dáte `i`, bash vám doplní na `iv`, po dalším stisknutí vám opět dá vybrat.

Bash completion

Ale doplňování bashem není všemocné. Chcete-li třeba nainstalovat „ten program, jak začíná na vim-

g(něco)", tak vám bash už nepomůže. Ale od toho tu je bash-completion.

✘ V Ubuntu je bash-completion zapnuté od počátku, nemusíte ho tedy nijak zapínat.

Aktivovat bash-completion lze editací souboru:

```
gksudo gedit /etc/bash.bashrc
```

Nyní je třeba část „enable bash completion in interactive shells“ odkomentovat (tj. odstranit # na počátku řádku): Před...

```
# enable bash completion in interactive shells
#if [ -f /etc/bash_completion ]; then
#     . /etc/bash_completion
#fi
```

Po...

```
# enable bash completion in interactive shells
if [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
fi
```

A pak už jen stačí uvědomit bash o změně:

```
source /etc/bash_completion
```

(Anebo prostě vypnete a znovu zapnete terminál 😊 či se odlogujete a zalogujete na tty, efekt to má stejný).

Další "doplňovací" tip

✘ Můžete také zkusit jiný interpret, třeba zsh.

Kdo to je ten "root"?



Root (čili superuživatel) je správce systému. Má absolutní moc nad systémem, narozdíl od obyčejného uživatele, který může poškodit tak maximálně své soubory, nikoliv však systémové. Jistě je vám jasné, jak užitečné je toto oddělení, i význam toho např. proti možnosti šíření virů (které zatím pro Linux v zásadě neexistují). Avšak tu a tam je třeba vykonávat i tu administraci, a tedy se na roota přihlásit. Dlužno poznamenat, že v Ubuntu je root defaultně (ve výchozím nastavení) zakázán. Neznamená to, že nelze provádět administraci 😊 pouze se dělá jinak - přes příkaz sudo. Chcete-li třeba zeditovat /etc/fstab, tak dáte:

```
gedit /etc/fstab
```

Nicméně gedit vás změny nenechá uložit, neb nemáte oprávnění. Musíte jej otevřít s rootovskými oprávněními:

```
sudo gedit /etc/fstab
```

Chcete-li získat terminál roota, abyste nemuseli před každý příkaz psát sudo, pak takto:

```
sudo -i
```



Další důležité informace se dozvíte na stránce [Root sudo](#), kterou velice doporučujeme k pečlivému prostudování.

Základní příkazy

ls Vypis složky. Buď aktuální či té, kterou uvedeme v cestě (ls /home/botanicus vypíše obsah mého domovského adresáře). Lze mu předat parametry, třeba ls -l vypíše podrobnější informace a ls -a vypíše i skryté soubory.

```
root@ubuntu:/home/botanicus/gaim-2.0.0beta2/src# ls
account.c          gtkconvwin.h      log.c
account.h          gtkdebug.c        log.h
accountopt.c      gtkdebug.h        Makefile.am
(...)
```

man a **info** Manuálové stránky. Info je podrobnější. Jako parametr se udává program, o němž se chcete něco dozvědět. Zkuste třeba

```
man man
```

cd Vejde do zadané složky. Buď lze použít absolutní cestu - tedy cestu od kořene

```
cd /home/uživatel
```

nebo relativní - ta nezačíná /, a udává cestu od vaší aktuální pozice (jsem-li v /home/uživatel a chci-li se dostat do /home/uživatel data, stačí dát

```
cd data
```

. označuje aktuální složku

.. označuje nadřazenou složku.

cp Slouží ke kopírování souborů. Chceme-li kopírovat složku, pak je třeba použít parametr -R. Parametr -i zabraňuje možnosti přepsání souboru - zeptá se vás, zda soubor chcete či nechcete

přepsat.

```
mv /home/botanicus/co /data/kam
```

mv Slouží k přesouvání souborů. Nerozlišuje mezi souborem a složkou jako cp. Opět zná parametr -i.

rm Maže soubory. Parametr -R dovolí mazat adresáře.

```
rm -R /home/botanicus/slozka
```

mkdir a **rmdir** Tvoří (mkdir) či maže (rmdir) prázdnou složku. (Pozn.: Na mazání neprázdné složky slouží rm -R)

```
mkdir /home/botanicus/nova_slozka
```

apt-cache search *balik* Hledá zadaný balík.

apt-cache show *balik* Zobrazí informace o zadaném balíku.

apt-get install *balik* Instaluje zadaný balík. Více na [Pokročilá správa softwaru](#)

apt-get remove *balik* Odstraní zadaný balík. Více na [Pokročilá správa softwaru](#)

aptitude Je vylepšené apt. Bez parametrů se spustí interaktivně. Více na [Pokročilá správa softwaru](#)

history Vypíše historii provedených příkazů. Jakýkoli příkaz z historie je možné zopakovat takto (n=pořadové číslo příkazu):

```
user@ubuntu:~$ !n
```

grep Hledá text a vypisuje řádku, v níž je obsažen.

```
botanicus@ubuntu:~$ grep -e řetězec wiki
''*'' - nahrazuje jakýkoliv počet jakýchkoliv znaků - od žádného znaku po libovolný řetězec.
```

cat Vypíše obsah textového souboru.

```
root@ubuntu:/home/botanicus/gaim-2.0.0beta2/src# cat stun.c
/**
 * @file stun.c STUN (RFC3489) Implementation
 * @ingroup core
 *
 * gaim
 *
 * STUN implementation inspired by jstun [http://jstun.javawi.de/]
 * (...)
```

more less Vylepšený cat - vypíše soubor po částech, když chceme „popojet“ níže, dáme mezerník - o stránku či enter - o řádek.

chroot Mění kořenový adresář. Nemá smysl se o něm rozepisovat podrobně, zmiňuji jej, protože v různých krizových situacích se hodí (zachraňování poškozeného systému z liveCD atp.).

```
botanicus@ubuntu:~$ chroot /mnt/slackware/
```

mount a **umount** Připojuje (mount) či odpojuje (umount) partition, které jsou zapsány v /etc/fstab. Může používat pouze root (či přes sudo). Tedy lze to zpřístupnit i obyčejným uživatelům, když do /etc/fstab do čtvrtého sloupečku přepíšeme user.

```
mount /dev/hda2
```

Pokud partition není v /etc/fstab, je nutné zadat *mountpoint*.

```
mount /dev/hda2 /mnt/data
```

eject Odmountuje a vysune CD mechaniku. Parametr -t (tedy eject -t), případně -T (takže eject -T) zasune CD mechaniku zpět; ale tento parametr není podporován všemi CD mechanikami. Máte-li mechanik více, na konkrétní ukážete například takto:

```
eject /dev/sr0
```

whereis Vypisuje soubory programu (kam je nainstalován, konfigurační a manuálové stránky).

```
botanicus@ubuntu:~$ whereis bash
bash: /bin/bash /etc/bash.bashrc /usr/share/man/man1/bash.1.gz
```

which Vypíše kde je nainstalován daný program.

```
botanicus@ubuntu:~$ which bash
/bin/bash
```

df Vypisuje velikost a obsazení jednotlivých partition.

```
botanicus@ubuntu:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/hda1       14G   6,8G   7,0G   50% /
varrun          244M   112K   244M    1% /var/run
varlock         244M    4,0K   244M    1% /var/lock
udev            244M   132K   244M    1% /dev
devshm          244M     0    244M    0% /dev/shm
lr              244M   19M   226M    8%
/lib/modules/2.6.15-21-686/volatile
/dev/hda2       36G   24G   11G   69% /data
```

du Vypisuje velikost daného souboru.

```
botanicus@ubuntu:~$ du wiki
12      wiki
```

top Monitoruje procesy. Doporučuji nainstalovat htop, je mnohem lepší.

ps Vypisuje procesy a informace o nich. Používejte jej třeba s parametry aux.

```
botanicus@ubuntu:~$ ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.1  1564   528 ?        S    07:08   0:01 init [2]
root         2  0.0  0.0      0     0 ?        S    07:09   0:00
[migration/0]
root         3  0.0  0.0      0     0 ?        SN   07:09   0:00
[ksoftirqd/0]
root         4  0.0  0.0      0     0 ?        S    07:09   0:00
[watchdog/0]
root         5  0.0  0.0      0     0 ?        S<   07:09   0:00 [events/0]
(...)
```

Žolíkové znaky

* - nahrazuje jakýkoliv počet jakýchkoliv znaků - od žádného znaku po libovolný řetězec.

```
root@ubuntu# apt-get install gaim*
```

Takto nainstalujete cokoliv co začíná na „gaim“. Lze předpokládat, že to bude gaim a všemožná rozšíření k němu (ovšem neručím za to, že tam nebude žádný konflikt mezi balíky, to je zase jiná věc).

? Nahrazuje právě jeden libovolný znak.

~ Označuje domovskou složku (/home/uživatel). Použiji-li třeba cd ~, pak budu v /home/botanicus. Použije-li totéž Pepa, dostane-jej to do /home/Pepa. Pokud bude Pepa chtít do mé domovské složky, může to udělat přes **cd ~botanicus**. Pozn.: Rozhodně to není to samé, co regulární výrazy!

Aliases

Možnost 1

Příkaz alias umožňuje použití některého příkazu vždy s určitými parametry. Kupříkladu příkaz df vypisuje velikost jednotlivých partition. Jenže v KB, což není moc přehledné. Přidáme-li mu parametr -h, použije MB a GB. Chcete-li, aby df vždycky použilo tento parametr, použijte alias.

```
alias df='df -h'
```

Když teď dáte df, je to jak byste použili df -h. No jo, ale když se odhlásíte, tak budou změny ztraceny. Proto je třeba přidat alias do ~/.bashrc

```
botanicus@ubuntu:~$ cat >> .bashrc
```

```
alias df='df -h'  
Ctrl+D
```

Pozn.: jsou tam dvě », pokud dáte jedno, máte problém ... 😊 nebo takto:

```
botanicus@ubuntu:~$ gedit .bashrc
```

... a tam to prostě dopište 😊

Možnost 2

Použijeme svůj oblíbený textový editor k editaci souboru **.bashrc**, kde vyhledáme

```
# Alias definitions.  
# You may want to put all your additions into a separate file like  
# ~/.bash_aliases, instead of adding them here directly.  
# See /usr/share/doc/bash-doc/examples in the bash-doc package.  
  
#if [ -f ~/.bash_aliases ]; then  
#    . ~/.bash_aliases  
#fi
```

Odkomentujeme řádky s if:

```
if [ -f ~/.bash_aliases ]; then  
    . ~/.bash_aliases  
fi
```

Nyní použijeme textový editor k vytvoření souboru **.bash_aliases**, kam můžete vepsat Vaše aliasy, které zůstanou i po logoutu. Výhodou je přehlednost oproti souboru **.bashrc**. Soubor tedy může vypadat např. takto:

```
alias vypni='shutdown -h'  
alias mysqlprace='mysql -h nejakaip -u user -p'
```

Práci dokončíme reloadem bashe.

```
$ bash
```

Lehký úvod do skriptů

Skript je seznam příkazů pro interpret. Například skriptík pro stahování nejprve vytočí připojení (máte-li modem), poté stáhne poštu a případně další věci, co mu uložíme, a pak zase spojení ukončí. Příkazy se mohou buď provádět nezávisle na sobě (stáhnutí pošty a k tomu třeba aktualizace) anebo mohou mít vnitřní hierarchii - vytoč spojení a pokud jej vytočíš, tak teprve stahuj, pokud se nepovede vytočit, stahování se neuskuteční. Klasické bashovské skripty se sestávají z jednotlivých příkazů - tedy


jednotlivých programů.

Takto třeba může vypadat úplně jednoduchý skriptík pro zálohování souborů ve složce Soubkzal:

```
# tohle zalohuje :-)  
cd /home/botanicus/data/Zalohy/Soubkzal  
tar cjf /home/botanicus/data/Zalohy/"Bi0etS0C-`date`.tbz"  
/home/botanicus/data/Zalohy/Soubkzal/*  
echo 'Zaloha uspesne provedena :-)'  
sleep 3  
exit
```

První řádek skriptu určuje interpret. Skripty pro shell začínají zpravidla `#!/bin/sh` či `#!/bin/bash`, ovšem klidně můžeme mít třeba `#!/bin/zsh`. Rozdíl je v tom, jaký shell použijete - jednoduché skripty vám budou fungovat jak v `sh`, tak v `bash`, ale pozor, není to totéž. Budete-li psát skripty třeba pro Perl či Python, tak zase `#!/usr/bin/perl` a `/usr/bin/python`. Ale když se váš skript dostane třeba do FreeBSD, které nemá `bash` v `/bin`, ale v `/usr/bin`, co potom? Lze to obejít - nejelegantnější cestou je říci systému, ať si zjistí, kde ten interpret vůbec je. Na to lze použít toto: `#!/usr/bin/which` interpret (třeba `#!/usr/bin/which perl`). Následující řádek je komentář - komentáře se píše za `#`. Za tímto znakem budou všechny znaky až do konce řádku ignorovány. Toho se využívá také při „zakomentování“ - tedy když chcete něco ve skriptu pro teď zakázat, ale nechcete to smazat, tak před to jen napište mřížku. Další řádky už jsou jen jednotlivé příkazy: vejdi, sbal, pojmenuj dle data, vytiskni na standardní výstup, že se to povedlo, vydrž tři sekundy a pak se ukonči.

From: <https://wiki.ubuntu.cz/> - **Ubuntu Česká republika**

Permanent link: https://wiki.ubuntu.cz/lehk%C3%BD_%C3%BAvod_do_taj%C5%AF_p%C5%99%C3%ADkazov%C3%A9_%C5%99%C3%A1dky 

Last update: **2019/02/25 18:20**