

# Bourne-Again SHell a Linux CLI

© 2013 Martin Bruchanov, bruxy@regnet.cz

Nastavení interpretru: `#!/bin/bash`
Komentáře: `#` toto je komentář
Interaktivní ovládání

Akce	set -o vi	set -o emacs
vi-command režim (C)	<span><span>ESC</span></span>	—
Předchozí/následující příkaz v hist.	<span><span>↑</span></span> / <span><span>↓</span></span>	<span><span>CTRL+↑</span></span> / <span><span>CTRL+↓</span></span> / <span><span>PAGEUP</span></span> / <span><span>PAGEDOWN</span></span>
Doplnění názvu souboru	<span><span>ESC</span></span> <span><span>ESC</span></span>	<span><span>TAB</span></span>
Výpis všech shod	<span><span>ESC</span></span> <span><span>G</span></span>	<span><span>TAB</span></span> <span><span>TAB</span></span>
Horizontální pohyb po řádce	<span><span>←</span></span> / <span><span>→</span></span>	<span><span>CTRL+←</span></span> / <span><span>CTRL+→</span></span> / <span><span>←</span></span> / <span><span>→</span></span>
Skoč na začátek/konec řádky	<span><span>←</span></span> / <span><span>\$</span></span>	<span><span>CTRL+a</span></span> / <span><span>CTRL+e</span></span>
Zpětné/dopředné hledání v historii	<span><span>↑</span></span> / <span><span>↓</span></span>	<span><span>CTRL+↑</span></span> / <span><span>CTRL+S</span></span>
Vymaže slovo do konce/záчатку	<span><span>dw</span></span> / <span><span>db</span></span>	<span><span>ESC d</span></span> / <span><span>ESC b</span></span>
Vymaž text od kurzoru do konce/záчатку řádky	<span><span>\$D</span></span> / <span><span>←</span></span>	<span><span>CTRL+R</span></span> / <span><span>CTRL+U</span></span>

#### 1.1. Historie příkazů

- `history`, `fc -l` – zobrazí číslovanou historii příkazů shellu
- `!n` – spustí příkaz číslo *n*
- `!p` – spustí poslední příkaz začínající na *p*
- `!!` – provede poslední zadaný příkaz
- `!!:n` – expanduje na *n*-tý parametr předchozího příkazu
- `!$` – expanduje na poslední parametr předchozího příkazu
- `fc` – spustí zadaný `$EDITOR` s posledním příkazem
- `fc -e vim z k` – otevře editor `vim` s příkazy od *z* do *k*
- `~old~new` – nahrazení *old* za *new* v posledním příkazu
- `příkaz `!!`` – použije výstup z předchozího příkazu jako vstup

#### 1.2. Náповěda

- `type -a příkaz` – informace o příkazu
- `help příkaz` – stručná nápověda k příkazu
- `man příkaz`, `info příkaz` – podrobná nápověda
- `man -k klíč`, `apropos klíč`, `whatis příkaz` – hledá příkaz

### 2. Ladění

Spustit skript: `bash volba skript jeho parametry`

- `bash -x` – příkazy před provedením vypíše
- `bash -u` – použití nedefinované proměnné vrátí chybu
- `bash -v` – vypisuje načtené řádky skriptu před provedením
- `bash -n` – zabrání vykonání příkazů

### 3. Proměnné a pole

- `jméno=10` – přiřazení hodnoty proměnné `$PROM`, `${PROM}`
- `export PROM=10`, `typedef -x` – nastaví jako proměnou prostředí
- `D=${date}`; `D=`date`` – proměnná obsahuje výstup příkazu `date`
- `env`, `printenv` – vypíše proměnné prostředí
- `set` – vypíše prom. prostředí, umožňuje nastavení příznaků `shopt`
- `unset název` – odstraní proměnnou nebo funkci
- `typeset`, `declare` – nastavení typu proměnné.
- `readonly proměnná` – nastaví pouze pro čtení
- `local proměnná` – nastaví lokální proměnou uvnitř funkce
- `${!var}`, `eval `$$var` – nepřímá reference
- `${parametr-slovo}` – jestliže *parametr* má hodnotu, pak se použije jeho hodnota, jinak se použije hodnota *slovo*.
- `${parametr=slovo}` – jestliže *parametr* nemá přiřazenou hodnotu přiřadí se mu *slovo*. Nefunguje na \$1, \$2, atd.
- `${parametr:-slovo}` – funguje na \$1, \$2, atd.
- `${parametr?slovo}` – jestliže *parametr* má hodnotu, použije se *ta*; v opačném případě se zobrazí *slovo* a shell se ukončí.
- `${parametr+slovo}` – jestliže *parametr* má hodnotu, použije se *slovo*, jinak prázdný řetězec.
- `pole=(a b c)`; `echo ${pole[1]}` – vypíše ‚b‘
- `pole+=(d e f)` – přidá nové prvky/pole na konec
- `${pole[*]}`, `${pole[@]}` – všechny prvky pole
- `${#pole[*]}`, `${#pole[@]}` – počet prvků pole
- `declare -A hash` – vytvoří asociativní pole (od verze 4.0)
- `hash=([klíč1]=hodnota ["jiný klíč2"]="jiná hodnota")`
- `${hash["jiný klíč2"]}`, `${hash[jiný klíč2]}` – přístup
- `${hash[@]}`, `${hash[*]}` – všechny prvky
- `${!hash[@]}`, `${!hash[*]}` – všechny klíče

#### 3.1. Řetězce

- `STRING="Ahoj"` – indexování: `A`<sub>0</sub> `h`<sub>1</sub> `o`<sub>2</sub> `j`<sub>3</sub>
- `STRING+= " světe!"` – přidá řetězec
- `${#řetězec}`, `expr length $řetězec` – délka řetězce
- `${řetěz:pozice}` – extrahuje podřetězec od pozice
- `${řetěz:pozice:délka}` – extrahuje podřetěz délky od pozice
- `${řetěz/podřetěz/náhrada}` – nahradí první výskyt podřetězce
- `${řetěz//podřetěz/náhrada}` – nahradí všechny výskyty
- `${řetěz/%podřetěz/náhrada}` – nahradí první výskyt od konce
- `${řetěz#podřetěz}` – vymaže nejkratší podřetězec
- `${řetěz##podřetěz}` – vymaže nejdělsí podřetězec

#### 3.2. Standardní proměnné

- `-`, `$HOME` – domovský adresář uživatele
- `$PS1`, `$PS2` – primární, sekundární prompt
- `$PWD`, `--` / `$OLDPWD`, `--` – aktuální / předchozí adresář
- `$RANDOM` – gen. náhodných čísel od 0 do 32 767
- `$?` – návratová hodnota předchozího programu
- `$$` – id. číslo právě běžícího procesu
- `$!` – id. číslo posledního příkazu spuštěného na pozadí
- `$PPID` – id. číslo rodičovského procesu
- `$-` – zobrazí nastavené příznaky
- `$LINENO` – aktuální číslo řádku v prováděném skriptu

- `$PATH` – seznam cest se spustitelnými příkazy
- `$IFS` – Vnitřní oddělovač polí. Znaky sloužící k oddělování slov, když shell čte vstup, obvykle mezera, tabulátor `$'\t'` a nový řádek `$'\n'`.

#### 4. Parametry programu

- `$0`, `${0}` – název skriptu
- `$1` až `$9`, `${1}` až `${255}` – poziční parametry na příkazové řádce
- `$#` – počet parametrů na příkaz. řádce (`argc`)
- `$*` – rozvine všechny parametry, `"$*" = "$1 $2 $3..."`
- `$@` – rozvine všechny parametry, `"$@" = "$1" "$2" "$3"..."`
- `_` – poslední parametr předchozího programu
- `shift` – přejmenování parametrů, `$2` na `$1`, `$3` na `$2`, atd.; sniží čítač  `$#`
- `xargs prog` – načítá vstup a dává ho jako parametry *prog*

#### 4.1. Čtení parametrů z příkazové řádky

```
while getopts "a:b" opt; do case $opt in
a) echo a = $OPTARG ;;
b) echo b ;;
\?) echo "Neznámý parametr" ;;
esac; done
shift ${#OPTARG - 1}); echo "Poslední: $1"
```

esac; done

```
shift ${#OPTARG - 1}); echo "Poslední: $1"
```

while getopts "a:b" opt; do case \$opt in

```
a) echo a = $OPTARG ;;
```

```
b) echo b ;;
```

```
\?) echo "Neznámý parametr" ;;
```

```
esac; done
```

```
shift ${#OPTARG - 1}); echo "Poslední: $1"
```

while getopts "a:b" opt; do case \$opt in

```
a) echo a = $OPTARG ;;
```

```
b) echo b ;;
```

```
\?) echo "Neznámý parametr"span> ;;
```

```
esac; done
```

while getopts "a:b" opt; do case \$opt in

```
a) echo a = $OPTARG ;;
```

```
b) echo b ;;
```

```
\?) echo "Neznámý parametr"span> ;;
```

```
esac; done
```

while getopts "a:b" opt; do case \$opt in

```
a) echo a = $OPTARG ;;
```

```
b) echo b ;;
```

```
\?) echo "Neznámý parametr"span> ;;
```

```
esac; done
```

while getopts "a:b" opt; do case \$opt in

```
a) echo a = $OPTARG ;;
```

```
b) echo b ;;
```

```
\?) echo "Neznámý parametr"span> ;;
```

```
esac; done
```

while getopts "a:b" opt; do case \$opt in

```
a) echo a = $OPTARG ;;
```

```
b) echo b ;;
```

```
\?) echo "Neznámý parametr"span> ;;
```

```
esac; done
```

while getopts "a:b" opt; do case \$opt in

```
a) echo a = $OPTARG ;;
```

```
b) echo b ;;
```

```
\?) echo "Neznámý parametr"span> ;;
```

```
esac; done
```

while getopts "a:b" opt; do case \$opt in

```
a) echo a = $OPTARG ;;
```

```
b) echo b ;;
```

```
\?) echo "Neznámý parametr"span> ;;
```

```
esac; done
```

while getopts "a:b" opt; do case \$opt in

```
a) echo a = $OPTARG ;;
```

```
b) echo b ;;
```

```
\?) echo "Neznámý parametr"span> ;;
```

```
esac; done
```

while getopts "a:b" opt; do case \$opt in

```
a) echo a = $OPTARG ;;
```

```
b) echo b ;;
```

```
\?) echo "Neznámý parametr"span> ;;
```

```
esac; done
```

while getopts "a:b" opt; do case \$opt in

```
a) echo a = $OPTARG ;;
```

```
b) echo b ;;
```

```
\?) echo "Neznámý parametr"span> ;;
```

```
esac; done
```

while getopts "a:b" opt; do case \$opt in

```
a) echo a = $OPTARG ;;
```

```
b) echo b ;;
```

```
\?) echo "Neznámý parametr"span> ;;
```

```
esac; done
```

while getopts "a:b" opt; do case \$opt in

```
a) echo a = $OPTARG ;;
```

```
b) echo b ;;
```

```
\?) echo "Neznámý parametr"span> ;;
```

```
esac; done
```

while getopts "a:b" opt; do case \$opt in

```
a) echo a = $OPTARG ;;
```

```
b) echo b ;;
```

```
\?) echo "Neznámý parametr"span> ;;
```

```
esac; done
```

while getopts "a:b" opt; do case \$opt in

```
a) echo a = $OPTARG ;;
```

```
b) echo b ;;
```

```
\?) echo "Neznámý parametr"span> ;;
```

```
esac; done
```

while getopts "a:b" opt; do case \$opt in

```
a) echo a = $OPTARG ;;
```

```
b) echo b ;;
```

```
\?) echo "Neznámý parametr"span> ;;
```

```
esac; done
```

while getopts "a:b" opt; do case \$opt in

```
a) echo a = $OPTARG ;;
```

```
b) echo b ;;
```

```
\?) echo "Neznámý parametr"span> ;;
```

```
esac; done
```

while getopts "a:b" opt; do case \$opt in

```
a) echo a = $OPTARG ;;
```

```
b) echo b ;;
```

```
\?) echo "Neznámý parametr"span> ;;
```

```
esac; done
```

while getopts "a:b" opt; do case \$opt in

```
a) echo a = $OPTARG ;;
```

```
b) echo b ;;
```

```
\?) echo "Neznámý parametr"span> ;;
```

```
esac; done
```

while getopts "a:b" opt; do case \$opt in

```
a) echo a = $OPTARG ;;
```

```
b) echo b ;;
```

```
\?) echo "Neznámý parametr"span> ;;
```

```
esac; done
```

while getopts "a:b" opt; do case \$opt in

```
a) echo a = $OPTARG ;;
```

```
b) echo b ;;
```

```
\?) echo "Neznámý parametr"span> ;;
```

```
esac; done
```

while getopts "a:b" opt; do case \$opt in

```
a) echo a = $OPTARG ;;
```

```
b) echo b ;;
```

```
\?) echo "Neznámý parametr"span> ;;
```

```
esac; done
```

while getopts "a:b" opt; do case \$opt in

```
a) echo a = $OPTARG ;;
```

```
b) echo b ;;
```

```
\?) echo "Neznámý parametr"span> ;;
```

```
esac; done
```

while getopts "a:b" opt; do case \$opt in

```
a) echo a = $OPTARG ;;
```

```
b) echo b ;;
```

```
\?) echo "Neznámý parametr"span> ;;
```

```
esac; done
```

while getopts "a:b" opt; do case \$opt in

```
a) echo a = $OPTARG ;;
```

```
b) echo b ;;
```

```
\?) echo "Neznámý parametr"span> ;;
```

```
esac; done
```

while getopts "a:b" opt; do case \$opt in

```
a) echo a = $OPTARG ;;
```

```
b) echo b ;;
```

```
\?) echo "Neznámý parametr"span> ;;
```

```
esac; done
```

while getopts "a:b" opt; do case \$opt in

```
a) echo a = $OPTARG ;;
```

```
b) echo b ;;
```

```
\?) echo "Neznámý parametr"span> ;;
```

```
esac; done
```

while getopts "a:b" opt; do case \$opt in

```
a) echo a = $OPTARG ;;
```

```
b) echo b ;;
```

```
\?) echo "Neznámý parametr"span> ;;
```

```
esac; done
```

while getopts "a:b" opt; do case \$opt in

```
a) echo a = $OPTARG ;;
```

```
b) echo b ;;
```

```
\?) echo "Neznámý parametr"span> ;;
```

```
esac; done
```

while getopts "a:b" opt; do case \$opt in

```
a) echo a = $OPTARG ;;
```

```
b) echo b ;;
```

```
\?) echo "Neznámý parametr"span> ;;
```

```
esac; done
```

while getopts "a:b" opt; do case \$opt in

```
a) echo a = $OPTARG ;;
```

```
b) echo b ;;
```

```
\?) echo "Neznámý parametr"span> ;;
```

```
esac; done
```

while getopts "a:b" opt; do case \$opt in

```
a) echo a = $OPTARG ;;
```

```
b) echo b ;;
```

```
\?) echo "Neznámý parametr"span> ;;
```

```
esac; done
```

while getopts "a:b" opt; do case \$opt in

```
a) echo a = $OPTARG ;;
```

```
b) echo b ;;
```

```
\?) echo "Neznámý parametr"span> ;;
```

```
esac; done
```

while getopts "a:b" opt; do case \$opt in

```
a) echo a = $OPTARG ;;
```

```
b) echo b ;;
```

```
\?) echo "Neznámý parametr"span> ;;
```

```
esac; done
```

while getopts "a:b" opt; do case \$opt in

```
a) echo a = $OPTARG ;;
```

```
b) echo b ;;
```

```
\?) echo "Neznámý parametr"span> ;;
```

```
esac; done
```

while