

# Bourne-Again SHell

© 2011 Martin Bruchanov, bruxy@regnet.cz

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

### 1. Interaktivní ovládání

Akce	set -o vi	set -o emacs
vi-command režim (C)	<span>(Esc)</span>	—
Předchozí/následující příkaz v hist.	<span>(j)</span> / <span>(k)</span>	<span>(CTRL+p)</span> / <span>(CTRL+n)</span> <p><span>(PAGEUP)</span> / <span>(PAGEDOWN)</span></p>
Doplnění názvu souboru	<span>(Esc)</span> <span>(Esc)</span>	<span>(TAB)</span>
Výpis všech shod	<span>(Esc)=</span>	<span>(Tab)</span> <span>(Tab)</span>
Horizontální pohyb po řádce	<span>(h)</span> / <span>(l)</span>	<span>(CTRL+b)</span> / <span>(CTRL+f)</span> <span>(←)</span> / <span>(→)</span>
Skoč na začátek/konec řádky	<span>(^)</span> / <span>(\$)</span>	<span>(CTRL+a)</span> / <span>(CTRL+e)</span>
Zpětné/dopředné hledání v historii	<span>(/)</span> / <span>(?)</span>	<span>(CTRL+r)</span> / <span>(CTRL+s)</span>
Vymaže slovo do konce/záчатку	<span>(dw)</span> / <span>(db)</span>	<span>(ESC d)</span> / <span>(ESC h)</span>
Vymaž text od kurzoru do konce/záчатку řádky	<span>(d\$)</span> / <span>(d^)</span>	<span>(CTRL+k)</span> / <span>(CTRL+u)</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 **\$EDIOR** 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

#### 1.2. Nápověda

- type -a příkaz** – informace o příkazu
- help příkaz** – stručná nápověda k příkazu
- man příkaz**, **info příkaz** – podrobná nápověda
- man -k slovo**, **apropos slovo** – prohledá databázi příkazů

### 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
- 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
- \${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>1</sub> h<sub>2</sub> o<sub>3</sub> j<sub>4</sub>
- STRING+=" světe!"** – přidá řetězec
- \${#řetězec}**, **expr length řřetězec** – 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 nejdelší 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**  – Oddělovač polí. Seznam znaků, které slouží k oddělování slov, když shell čte vstup, obvykle mezera, tabulátor  **\$' \t'**  a znak nový řádek  **\$' \n'** .

### 4. Parametry programu

- \$0** ,  **\${0}**  – název skriptu
- \$1 až \$9** ,  **\${1} až \${255}**  – poziční parametry zadané na příkazové řádce
- \$#**  – počet parametrů na příkaz. řádce (argc)
- \$\*, \$@**  – všechny parametry skriptu jako jedna hodnota
- shift**  – přejmenování parametrů, \$2 na \$1 \$3 na \$2, atd.; sniží čítač  **\$#**
- \$\_**  – poslední parametr, předchozího programu

#### 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"

## 5. Řídící výrazy

- (*příkazy*), **\$(příkazy)**, **`příkazy`** – spustí příkazy v subshellu
- test**, [ ] – vyhodnocování podmínek:
  - porovnání čísel – **a -eq b …a = b**, **a -ge b …a ≥ b**, **a -gt b …a > b**, **a -le b …a ≤ b**, **a -lt b …a < b**
  - souborové: **-d soubor** je adresář, **-f soubor** existuje a není to adresář, **-r soubor** existuje a lze ho číst, **-w soubor** existuje a lze do něj psát, **-s soubor** má nenulovou velikost, **-a soubor** existuje
  - logické: **-a and**, **-o or**, **! negace**
- [ [ ] ] – slouží k porovnání řetězců, shoda =, neshoda !=, **-z řetězec**,nulová délka, **-n řetězec** nenulová délka, **<**, **>** lexikální porovnání
- [ *podmínka* ] **&&** [ *podmínka* ]
- true** – vrací návratovou hodnotu 0
- false** – vrací návratovou hodnotu 1
- break** – ukončí prováděný cyklus
- continue** – začne novou iteraci cyklu
- eval parametry** – spustí parametry jako příkaz
- exit hodnota** – ukončení skriptu s návratem
- .** **skript**, **source skript** – načtení a interpretace jiného skriptu
- alias jméno='příkazy'** – chápe *jméno* jako příkaz
- alias jméno** – zruší alias
- if [ *podmínka* ]; then příkazy;**
- elif [ *podmínka* ]; then příkazy;**
- else příkazy; fi**
- for proměnná in argumenty;** do *příkazy*; **done**
  - {a..z}** – rozvine se na a b c … z
  - {i..n..s}** -- sekvence od *i* do *n* s *krokem s*
  - \{a,b,c\}** \ "a" "b" "c"
  - {1,2}{a,b}** -- *expanduje* na 1a 1b 2a 2b
  - seq od krok do -- číselná sekvence**
- for((i=1; i<10; i++)); do příkazy; done**
- while vrací pravda;** do *příkazy*; **done**
- until [ test vrací pravda ];** do *příkaz*; **done**
- case \$prom in hodnota<sub>1</sub>) příkazy ;; hodnota<sub>2</sub>) příkazy ;; \*) implicit. příkazy ;; esac**
- Definice funkce: *function název* () { *příkazy*; }
- return hodnota** – návratová hodnota funkce
- declare -f funkce** – vypíše deklaraci funkce

#### 5.1. Vstup pro interaktivní programy

./program << EOF            ./program <<-'EOF'

Vstup1                        Vstup1

Vstup2                        Vstup2

EOF                            EOF

### 6. Přesměrování

- > soubor** – přesměrování, vytvoří nebo přepíše soubor
- >> soubor** – připojí na konec souboru
- příkaz < soubor** – vstup příkazu ze souboru
- tee soubor** – čte stdin, zapisuje do souboru a do stdout
- příkaz 2> soubor** – přesměruje chybové hlášení do souboru
- 2>** – přesměruje pouze chybová hlášení
- 2>&1** – přesměruje chybová hlášení do std. výstupu
- exec 3<>/dev/tcp/addr/port** – deskriptor pro čtení a zápis po síti
- příkaz 2>&1 /dev/null** – potlačí všechny výpisy
- n> n>> n>&m** – operace přesměrování pro deskriptory *n*, *m*
- mkfifo název** – vytvoří pojmenovanou rouru, soubor do kterého jeden proces zapisuje, další čte.
- příkaz<sub>1</sub> | příkaz<sub>2</sub>** – roura, propojování příkazů
- read parametry** – čte řádek vstupu a rozloží ho na parametry

#### 6.1. Zpracování vstupu po řádcích

**cat** soubor.txt | (while read L; do echo "\$L"; done)

### 7. Vyhodnocování matematických výrazů

- let výraz**, **expr výraz**, **\$((výraz))**, **\$((výraz1, výraz2))**,
- Číselné soustavy: *základ#číslo*; šestnáctkově 0xABC, osmičkově 0253, dvojkově 2#10101011
- Operátory: **i++**, **++i**, **i--**, **--i**, **+**, **-**; **\*\*** umocňování, **\***, **/**, **%** zbytek po dělení;

logické: **!** neg., **&&** and, **||** or; binární: **~**, **&**, **|**; **<<**, **>>** posuvy; přiřazení: **\* =**  **/= %= += -= <>= &= ^= |= >>= <<=**; porovnávání: **<** **<=** **>** **>=**

- factor n** – rozloží *n* na prvočísla
- V plovoucí řadové čárce: **echo "scale=10; 22/7" | bc**

### 8. Výstup na obrazovku

- echo "text"** – vypíše text, **echo \*** vypíše soubory v daném adresáři
- echo -e "text"** – interpretuje escape-sequence (**\t** tab., **\a** píp., **\f** nová stránka, **\n** nový řádek), **-n**, **\c** potlačí **\n**, **\xHH** *hexa byte*, **\nnn** *oct. byte*

- stty -- mění a vypisuje nastavení terminálové linky**
- tty -- vypíše jméno terminálu připojeného na std. výstup**
- printf formát hodnoty -- formátovaný výpis**
- printf -v proměnná form. hod. -- form. výpis do proměnné**
  - % [příznaky][šířka][.přesnost][modifikátor]formát**
  - Formát: **%u,%d,%i** desítkově; **%E,%f** float, **%x,%X** hexa; **%o** osmičkově, **%s** řetězec, **%%** znak **%**
  - Šířka: *n* tiskne alespoň *n* znaků, mezery zprava, *0n* tiskne alespoň *n* znaků, *nu-* ly zleva, **\*** počet znaků udává předchozí argument
  - Přesnost: min. počet cifer, počet cifer za des. tečkou počet tištěných znaků, **\*** počet znaků udává předchozí argument
  - Příznak: - zarovná doleva, zprava mezery, + tiskne č. se znaménkem +/-

- tput** akce **-- provedení akce závislé na typu terminálu**
- reset, tput sgr0, tset -- resetuje terminál, zruší atributy**
- clear, tput clear -- vymaže obsah obrazovky**

### 9. Správa procesů

- příkaz &* – spustí *příkaz* na pozadí
- prog<sub>1</sub> && prog<sub>2</sub>** – spustí prog<sub>2</sub>, pokud prog<sub>1</sub> skončí úspěšně
- prog<sub>1</sub> || prog<sub>2</sub>** – spustí prog<sub>2</sub>, pokud prog<sub>1</sub> skončí chybně
- (CTRL+Z)** – zastaví proces (SIGSTOP)
- bg/fg** – spustí zastavený proces na pozadí/popředí
- jobs** – vypíše procesy běžící na pozadí
- exec příkaz** – nahradí běžící proces za *příkaz*
- wait** – shell čeká na skončení příkazů puštěných na pozadí
- top** – sledování procesů na CPU
- ps -xau** – výpis všech procesů a uživatelů, **ps -xaf, pstree** hierarchicky
- pgrep proces** – vypíše PID podle jména procesu
- nice -n p příkaz** – priorita *p* od **-20** (max.) do 19 (min.)
- renice -n p -p pid** – změni prioritu běžícího procesum
- kill -s k n** – pošle signál *k* procesu id. *n*, 0, 1 SIGHUP; 2 SIGINT (CTRL+C); 3 SIGQUIT; 9 SIGKILL; 15 SIGTERM; 24 SIGSTOP
- trap 'příkaz' signály** – unalias *jméno* – zruší alias
- killall jméno** – pošle signál procesům podle jména
- nohup příkaz &** – příkaz bude pokračovat po odhlášení uživatele
- time příkaz** – vypíše čas běhu
- times** – info o spotřebě uživ. a sys. času v daném shellu
- watch -n s příkaz** – každých *s* sekund pustí příkaz

### 10. Čas a plánování procesů

- date** – vypíše datum, **date --date=@unix\_čas**
- date +%Y%m%d %H:%M:%S %Z** – formát data 20090923 13:39:02 CEST
- cal** – zobrazí kalendář
- crontab -e** – editace crontabu, **-l** vylistování, údaje *min hour date month day command, \* \* \* \* \** příkaz každou minutu, *1 \* \* \* \** příkaz 1. min. hodiny
- at, batch, atq, atrm** – umisťuje do fronty, zobrazuje či maže úlohy určené pro pozdější provedení

### 11. Souborové operace

Nahrazení v jménech souborů: ? znak; \* žádný nebo více znaků; [množina] jeden nebo více daných znaků, rozsah [0-9] [a-z], [A-Z]; [!množina], [^množina] žádný ze znaků.

- ls** – vylistuje adresář, **ls -la**, **vdir** všechny soubory a info
- tree** – zobrazí stromovou strukturu adresářů
- file soubor** – identifikuje soubor podle jeho magického čísla
- lsattr, chattr** – vypsání a nastavení atributů ext2,3
- umask** – definuje masku příst. práv u nového soubnoru
- pwd (-P)** – logická (fyzická) cesta do aktuálního adresáře
- cd adresář** – změni adresář, **cd skočí do \$HOME**, **cd -** do \$OLDPWD
- dirs** – zobrazí zásobník adresářů
- pushd adresář** – uloží *adresář* do zásobníku
- popd** – změni aktuální adresář za ten na vrcholu zásobníku
- cp zdroj cíl** – kopíruje soubor
- ln -s cíl linka** – vytvoří symbolický link
- mkdir, rmdir** – vytvoří, smaže adresář
- rm soubor**, **rm -r -f adresář** – mazání
- touch soubor** – vytvoří soubor, existujícímu nastaví akt. čas
- du -h** – zobrazí velikost podadresářů
- stat soubor** – info o souboru, **stat --format=%s** velikost
- basename jméno přípona** – vypíše bez cesty nebo přípony
- dirname /cesta/soubor** – vypíše pouze cestu

### 12. Práce s obsahem souboru

- cat** – slučuje soubory a vypisuje je na standardní výstup
- cat > soubor** – vytvoří soubor, ukončení pomocí (CTRL+d)
- tac** – jako cat, ale řádkově pozpátku
- more, less** – výpis po stránkách, skrolovatelně
- od, hexdump -C** – výpis souboru oktalově, výpis hexdumpu
- wc** – zjistí počet řádků **-l**, písmen **-n**, bytů **-c**, slov **-w**
- head/tail** – vypíše začátek/konec souboru, **tailf, tail -f** čeká na nové
- split, csplit** – rozdělí na části podle velikosti, obsahu

- sort** – řazení, **-n** – číselné, **-r** reverzní, **-f** ignoruje velikost písmen
- uniq** – opakující se řádky vypíše pouze jednou
- sed -e 'skript'** – provede řádkovou editaci, skript **y/ABC/abc/** nahradí znaky A, B, C za a, b, c; **s/regexp/náhrada/**
- tr a b** – náhrada znaku *a* za *b*
- tr '[a-z]' '[A-Z]'** < soubor.txt – změni malá na velká
- awk '/výraz/ akce '** *soubor* – zpracuje řádky obsahující výraz
- cut -d oddělovač -f pole** – vypíše daný sloupec
- cmp soubor1 soubor2** – porovná soubory a vypíše první rozdíl
- diff, sdiff, vimdiff** – porovnání souborů
- dd if=vst. of=výst. bs=k count=n** – přečte *n* bloků velikosti *k*
- strings** – vypisuje tisknutelné řetězce souboru
- paste soubor<sub>1</sub> soubor<sub>2</sub>** – sloučí soubory po řádcích
- rev** – otočí každý řádek souboru

### 13. Vyhledávání

- whereis, which** – hledá systémové soubory
- grep** – **-i** ignoruje velikost písmen, **-n** vypíše číslo řádku s nálezem, **-v** vypisuje všechno mimo vzoru, **-E** rozšířený regexp
- locate jméno** – hledá soubor podle jména
- find cesta -name 'soubor\*'** – hledá všechny *soubor\**
- find cesta -exec grep text -H {} \;** – hledá soubor s *textem*

### 14. Uživatelé a práva

- whoami, who am i** – zjistí, kdo jsem
- w, who, users, finger** – vypíše připojené uživatele
- last / lastb** – historie úspěšných / neúspěšných přihlášení
- logout, (CTRL+d)** – odhlášení z shellu
- su jméno** – změni uživatele
- sudo** – spustí soubor pod daným uživatelem
- id jméno, groups jméno** – vypíše identitu, skupiny uživatele
- useradd, userdel, usermod** – vytvoří, smaže, upraví uživatele
- groupadd, groupdel, groupmod** – vytvoří, smaže, upraví skupinu
- passwd** – změň heslo
- pwck** – kontrola integrity */etc/passwd*
- chown vlastník:skupina soubor** – změna vlastníka, **-R** rekurze
- chgrp skupina soubor** – změni skupinu souboru
- chmod práva soubor** – změna práv, oktalově vlastník, skupina, ostatní; 444=**-r--r--r--**, 700=**-rwx-----**, 550=**-r-xr-x---**

### 15. Systémové nástroje

- uname -a** – název a verze operačního systému
- uptime** – uplynulý čas od spuštění systému
- fuser** – zjistí běžící procesy které se souborem právě pracují
- lsdf** – vypisuje seznam otevřených souborů
- sync** – vyprázdní buffery souborového systému
- chroot adresář příkaz** – příkaz bere adr. jako kořen
- strace,ltrace program** – zobrazí použitá systémová/knihovni volání
- ldd binárka** – vypíše knihovni závislosti

#### 15.1. Diskové oddíly

- df** – ukáže volné místo na připojených discích
- mount** – ukáže připojené oddíly
- mount -o remount -r -n /** – změna připojení / pouze pro čtení
- mount -o remount -w -n /** – změna připojení / pouze pro zápis
- mount -t iso9660 cdrom.iso /mnt/dir -o loop** – připoj image
- mount -t cifs \\\\server\ftp /mnt/adr -o user=a,passwd=b**
- umount oddíl** – odpojí oddíl
- fdisk -l** – vypíše disky a oddíly na nich
- blkid** – vypíše atributy blokových zařízení
- sync** – vyprázdni buffery souborového systému

#### 15.2. Systémové prostředky

- ulimit -l** – vypíše omezení sys. prostředků
- free, vmstat** – zobrazí využití fyzické, virt. paměti
- lspci, lsusb** – výpis PCI, USB zařízení
- dmesg** – výpis hlášek jádra
- sysctl** – nastavení parametrů jádra Linuxu za běhu
- dmidecode** – vypsání informací o BIOSu

### 16. Síť

- hostname** – vypíše jméno počítače
- ping host** – pošle ICMP ECHO\_REQUEST
- dhclient eth0** – nastaví dynamicickou konfiguraci eth0
- host, nslookup host/adr** – dotaz na záznam v DNS
- dig** – zjišťování informací z DNS
- whois doména** – zjistí vlastníka domény
- ethtool eth0** – měni HW parametry síťové karty eth0
- ifconfig** – vypíše síťová zařízení, nastaví adresry, masky
- ifconfig eth0 add 10.0.0.1 netmask 255.255.255.0**
- ifconfig eth0 hw ether 01:02:03:04:05:06** – změni MAC adresu
- route add default gw 10.0.0.138** – nastavení brány
- route -n, netstat -rn** – výpis routovací tabulky
- netstat -tlnp** – ukáže programy poslouchající na portech
- arp** – zobrazení ARP tabulky
- iptables -L** – zobrazí pravidla firewallu
- tcpdump -i eth0 'tcp port 80'** – zobrazí komunikaci s HTTP
- tcpdump -i eth0 'not port ssh'** – všechna komunikace mimo SSH
- ssh user@hostname příkaz** – spustí vzdáleně příkaz
- mail -s subject ad**