

Bash III

Eliška Jégrová

2. 11. 2025

Obsah

Proměnné

Cyklus for

Proměnné prostředí

Skripty

Kombinace příkazů

Programy, aliasy

Programy, funkce a aliasy

Proměnné

- ▶ Příklad přiřazení

```
$ jmeno=minotaur.dat
```

- ▶ Obsah proměnné

```
$ echo $jmeno
```

- ▶ Proměnnou lze použít v příkazu, např.:

```
$ head -n2 $jmeno | tail -n1
```

- ▶ Hodnota proměnné můžeme přepsat

```
$ jmeno=unicorn.dat  
$ head -n2 $jmeno | tail -n1
```

Cyklus for

- ▶ Syntaxe:

```
for prvek in seznam
do
    prikaz
done
```

- ▶ Příklad s proměnnou:

```
for jmeno in minotaur.dat unicorn.dat
do
    head -n1 $jmeno
done
```

- ▶ Příklad - vypsání souborů:

```
for soubor in *.dat; do
    echo $soubor
done
```

Cyklus for – vnořené cykly

- Příklad: Vytvořit složky z názvů molekul a určitých teplot

```
for molekula in cubane ethane methane
do
    for teplota in 25 30 37 40
    do
        echo mkdir $molekula-$teplota
    done
done
```

Proměnné prostředí

- ▶ Proměnné bashu - jsou dostupné jen pro samotný Bash
- ▶ Proměnné prostředí - dostupné i pro běžící programy
- ▶ Příkaz `env` vypíše všechny proměnné prostředí:

```
$ env | head  
$ env | grep ^LANG
```

Exportování

- ▶ způsob, jak předat proměnnou programům

```
$ export jmeno=basilisk.dat  
$ export jmeno
```

- ▶ Příklad

Bez exportu:

```
jmeno=minotaur.dat  
env | grep jmeno  
echo $jmeno  
minotaur.dat
```

S exportem:

```
export jmeno  
env | grep jmeno  
jmeno=minotaur.dat  
echo $jmeno  
minotaur.dat
```

Složení názvů proměnných, Výzva

- ▶ Pokud chceš mít text za proměnnou (např. „ovoce“ v „jablko“), použij složené závorky:

```
$ majitel=Petr  
$ echo ${majitel}ovo jablko
```

- ▶ Bash má vlastní proměnné používané pro prompt: PS1, PS2.
 - ▶ Příklad přenastavení výzvy:

```
$ PS1=' [0_0] '
```

Skripty

- ▶ Skripty jsou jednoduché programy – často spojují existující příkazy.
- ▶ Ukázka obsahu klasifikace:

```
for x in *.dat
do
    head -n2 $x | tail -n1
done
```

Spustitelný skript

- ▶ Skript se spouští pomocí Bashe:

```
$ bash klasifikace
```

- ▶ Aby šel spustit jako příkaz, je třeba:

1. Přidat **shebang** na první řádek:

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

2. Nastavit spustitelný příznak:

```
$ chmod +x klasifikace
```

3. Spustit:

```
$ ./klasifikace
```

Argumenty skriptu

```
$ ./klasifikuj minotaur.dat
```

- ▶ Argumenty z příkazové řádky se předávají pomocí speciálních proměnných
 - ▶ "\$0" – jméno skriptu
 - ▶ "\$1" – první argument
 - ▶ "\$2", "\$3" – další argumenty
 - ▶ "\$@" – všechny argumenty
- ▶ Použití ve skriptu:

```
#!/bin/bash  
head -n2 "$1" | tail -n1
```

Proměnná PATH

- ▶ Systém hledá spustitelné programy v adresářích uvedených v proměnné PATH.

```
$ echo $PATH  
/usr/local/bin:/usr/bin:/bin:/usr/local/sbin
```

- ▶ Skript přesuneme do složky v PATH, např. \$HOME/bin
- ▶ Spustíme skript jen názvem:

```
$ klasifikace *.dat
```

- ▶ K trvalé změně PATH je potřeba upravit ji v souboru .bashrc

Další kombinace příkazů

- ▶ Příkazy lze řetězit pomocí `;` – provedou se postupně:

```
$ echo "Seznam souboru:"; ls
```

- ▶ Operátor `&&` – druhý příkaz se provede jen pokud první uspěl:

```
$ mkdir test && echo "Složka vytvořena."
```

- ▶ Operátor `||` – druhý příkaz se provede jen pokud první selhal:

```
$ cat neexistujici.txt || echo "Soubor nebyl nalezen"
```

Jak Bash hledá příkazy

- ▶ Když do Bashi zadáš jméno příkazu (např. `ls`), Bash hledá v několika druzích příkazů:
 - ▶ exteníbinární programy v adresářích z `$PATH`
 - ▶ zabudované příkazy (built-in) – např. `cd`, `export`
 - ▶ aliasy – zkratky pro jiné příkazy
 - ▶ funkce shellu – skripty přímo definované v Bashi
- ▶ Příkaz `type` ukáže, k jakému typu příkaz patří:

```
type cd
type ls
type myalias
```

Alias – jednoduchá zkratka

- ▶ Alias je zkratka, která nahrazuje jeden příkaz jiným (v rámci shellu).

- ▶ Definice aliasu:

```
alias ll='ls -l --color=auto '  
alias gs='git status '
```

- ▶ Použití: když napíšeš `ll /home`, Bash místo toho spustí `ls -l -color=auto /home`.

- ▶ Zrušení aliasu:

```
unalias gs
```

- ▶ Poznámka: alias funguje jen na začátku příkazu – nepřepíše části za středníkem či v řetězci příkazů.

Funkce v Bashi

- ▶ Funkce je více „robustní“ zkratka – může obsahovat více příkazů, logiku, podmínky.
- ▶ Definice:

```
myskupina() {  
    echo "Spoustim funkci s argumenty: $@"  
    ls -l "$1"  
}
```

- ▶ Volání:

```
myskupina /etc
```

- ▶ Argumenty uvnitř: \$1, \$2, ..., počet v \$#, všechny v \$@.
- ▶ Užitečný příklad – kombinace mkdir + cd:

```
mkcd() {  
    mkdir -p "$1" && cd "$1"  
}
```

Zapamatování aliasů & funkcí (např. v .bashrc)

- ▶ Alias nebo funkci definujeme ručně, ale zmizí po ukončení Bashi.
- ▶ Řešení: vložit je do souboru `/.bashrc`, který se načte při startu shellu:

```
alias ll='ls -l --color=auto'

mkcd() {
    mkdir -p "$1" && cd "$1"
}
```

- ▶ Po úpravě spustíme:

```
source ~/.bashrc
```

- ▶ Nebo otevřít nový terminál — definice se načtou.