

B4B33PSY - Seminar 2: Bash scripts, more constructions

Bc. Štěpán Pressl

Skripty, argumenty

Vytvoříme si soubor, ve kterém bude napsaný skript.

Nezapomeňte nastavit execute práva (`chmod u+x <skript>`).

- `$#`: počet vstupních argumentů
- `$1`, `$2`, .. `$9`, pro více argumentů je nutný použít `shift` (z `$2` se stane `$1`, atp., `$#` se dekrementuje)
- v `$0` je uložené jméno spuštěného skriptu
- je dobré specifikovat defaultní interpreter pomocí `#!/usr/bin/env bash` (pro Python třeba `/usr/bin/env python3`)
- všechny argumenty lze referencovat pomocí `$@` či `$*`

Více příkazů najednou

Příkazy lze sloučit do skupiny pomocí `{ ... }` nebo `(...)`.

- První skupina spouští skupinu příkazu v momentálním shellu
- Druhá skupina spustí subshell a v něm spustí skupinu příkazů
- Pojdme debatovat o problémech!

Spouštění skriptu vs source, viditelnost proměnné

- Když spustíte skript pomocí `/bin/bash` či `./`, vytvoří se nový proces
- `source` vykoná příkaz v momentální instanci shellu
- pokud chcete předat proměnné prostředí do dalších procesů, použijte `export`

Cykly v bashi: for, while, IFS

- viz `example1.sh` (for cyklus), `example2.sh` (while cyklus s argumenty)
- cyklus `for` iteruje přes argumenty v `in`, oddělené “separační proměnnou `$IFS`”, popř. přes numerický range (což jsou mezerou oddělená čísla, viz `seq`)
- cyklus `while` vykonává blok skriptu, jestliže je splněna podmínka
- lze použít `continue`, `break`, `atp.`

File globbing, wildcards

- tzv. zástupné znaky, zaměříme se na * a ?
- *: udělá match s nula či více jakýmkoli znaky
- ?: udělá match s jakýmkoli právě jedním znakem
- použití ve for cyklu (viz `example3.sh`)

Cvičení (tisknutí souborů splňující podmínku na velikost)

Napište skript, který bude brát přesně dva parametry. První parametr bude obsahovat příponu souboru. Druhý parametr bude obsahovat hraniční hodnotu velikosti souboru v bytech. Skript v adresáři . projede všechny soubory s danou příponou a jestliže je jejich velikost menší než hranice, tak je vypíše na standardní výstup. Jestliže nejsou zadány argumenty, skončete chybou.

Nápověda: pro zjištění velikosti souboru použijte:

```
size=$(du -b $file | cut -f 1)
```


Absolutní vs. relativní cesty

- `basename <file>`: získá jméno souboru (odstraní cesty k souboru), nezajímá ho, zda-li soubor neexistuje
- `dirname <file>`: opak - získá adresář souboru, kde se nachází

Uvažujme skript `skript.sh` a ve stejném adr. `soubor.txt`:

```
if [ -e "soubor.txt" ]; then
    echo "Soubor.txt nalezen."
else
    echo "Soubor.txt prostě zmizel."
fi
```

Skript pustíme z . jako ./skript.sh a poté z .. jako skriptdir/
skript.sh. Kde je problém?

Řešení problému cesty

- viz `example4.sh` a `example4-fixed.sh`
- Problémem je místo, odkad' spouštíme skript.
- Potřebujeme vztahovat místo souboru k absolutní cestě skriptu
- Řešením je vlézt do cesty skriptu a tam vytisknout jeho cestu

Manipulace s proměnnou

- Otevřte si `https://psy.pages.fel.cvut.cz/labs/02-bash/bash/` :-)

Pole

- Otevřte si `https://psy.pages.fel.cvut.cz/labs/02-bash/bash/` :-)

Example5

Uvažujte, že máte na vstupu n čísel. Skript bude brát $n + 1$ argumentů, první argument bude hraniční hodnota, bude následovat n čísel. Pokud bude počet argumentů nula, skript automaticky skončí - vypíše ERROR a vrátí 1. Jinak vytvoří pole obsahující všechna čísla, která jsou větší nebo rovno hraniční hodnotě, vytiskne je a vrátí 0. Pokud velikost pole je nula, skript vytiskne NO ELEMENTS a vrátí 2.