

InstallFest 2009

BASH - WORKSHOP

Radim “worwan” Roška

Silicon Hill

March 24, 2009

0.1 Agenda

- Efektivita
- základy
- Kouzla s promennými(vypučenno ze silkovy prednasky)
- řetězení příkazů
- if, test
- cyklus for,while
- matematika

MOTTO: Kdykoliv se ptejte;)

1.1 Efektivita

- nejpouzivanejsi klavesa = tab..
- sipka nahoru - vypisuje prikazy, které byly pouzity (historie prikazu)
- ctrl + r - hledani v historii prikazu (opakovany stisk = next)
- alt + . - doplni posledni argument (opakovani doplňuje predchozi atd.)
- ctrl + e - na konec prikazu
- ctrl + a - na zacatek prikazu
- ctrl + y - vlozi posledne smazane slovo
- ctrl + t - prohodi 2 pismena pred kurzorem
- ctrl + k - smaze vse za kurzorem
- ctrl + u - smaze vse pred kurzorem
- ctrl + leva/prava sipka - jak jste zvykli :)

Doporucuji zvednou promennou HISTSIZE, ctrl + r se pak stava ultimatni nastroj.

```
(echo 'export HISTSIZE=5000' >> ~/.bashrc)
```

1.2 základy

- projdeme si příkazy na taháku..ptejte se ;)
- proměnné - money money money..(“\$proměnná”).
- regularni vyrazy - bash type
- řetězce - různé druhy uvozovek (' “ ‘)

1.3 Kouzla s promennými

- Proměnná `a=soubor_1.jpg`
- `${a%.jpg}` – vypíše `soubor_1`
- `${a/1/jedna}` – vypíše `soubor_jedna.jpg`
- `${a#soubor}` – vypíše `_1.jpg`
- `${a#bor}` – to samé
- `${#a}` – 12 (délka hodnoty)
- `${a:3:4}` – `bor_` (substring)
- `${a:-default_file}` – vypíše `soubor_1.jpg`, ale když proměnná `a` není nastavená, vypíše `default_file`.

1.4 řetězení příkazů

kolony - příkaz | příkaz | příkaz... (výstup jednoho na vstup druhého)
Úkol - seřaďte obsah adresáře / (kromě souboru, adresářů začínajících na i)
podle abecedy sestupně: (napověda: může se hodit parametr -d u příkazu
ls)

1.4 řetězení příkazů

kolony - příkaz | příkaz | příkaz... (výstup jednoho na vstup druhého)

Úkol - seřaďte obsah adresáře / (kromě souboru, adresářů začínajících na i) podle abecedy sestupně: (napověda: může se hodit parametr -d u příkazu ls)

- `ls -d /[~i]* | sort -r`

1.5 if, test

- `if false;then echo a; else echo b ;fi`
- příkaz `test` (man `test`), testuje něco a vrácí `true/false`, `test` lze zapsat také jako příkaz `[co]`
- vrácenou hodnotu zjistíme přečtením proměnné `$?`
`true = 0`
`false = 1`

example: `test -f /etc/passwd; echo $?`

1.6 cyklus for,while

```
for promenna in z_ceho_vybirame; do prikaz $promenna ;done  
while boolean; do prikaz; done
```

Úkol - simulujte ls pomocí cyklu for:

1.6 cyklus for,while

```
for promenna in z_ceho_vybirame; do prikaz $promenna ;done while  
boolean; do prikaz; done
```

Úkol - simulujte ls pomocí cyklu for:

- `for neco in *; do echo $neco; done`

matematika

- echo $\$(3 \text{ operace } 2)$
- modulo: echo $\$(3 \% 2) \Rightarrow 1$
- mocnina: echo $\$(3 ** 2) \Rightarrow 9$
- VAR=10
 - echo $\$(VAR ** 3)$

užitečné nástroje - sed

velmi zjednodusene:

```
sed '[specifikace_radku] [co s nim]'
```

regularní výrazy...

```
● cat /etc/apache2/apache2.conf | sed '
/^ *#/d
/^$/d
'
```

```
cat /etc/passwd | sed 's/root/neroot/g'
```

```
cat /etc/passwd | sed '/root/a o radek vys je root'
```

```
cat /etc/passwd | sed '/root/ w /tmp/root_line'
```

```
cat /etc/passwd | sed '
> s/bin/unbin/
> 5q
> '
```

užitečné nástroje - find

find argumenty jsou postupně podmínky, které hledaný soubor musí splňovat
find / -name jmeno -type f -readable
example?

Úkol: Jak zobrazit strukturu adresaru po 3. uroven: (man find)

užitečné nástroje - find

find argumenty jsou postupně podmínky, které hledaný soubor musí splňovat
find / -name jmeno -type f -readable
example?

Úkol: Jak zobrazit strukturu adresaru po 3. uroven: (man find)

- `find /xx/yyy/zzz -maxdepth 3 -type d`

jednoduchy skript

Skript by mel mit 1. řádek: `#!/bin/bash`
hello world script

- `#!/bin/bash`

```
echo Hello world!
```

slozitejší skript

Napište skript, který projde všechny soubory v homu, mladší než /etc/passwd, ověřte že je můžeme spouštět - právo x - (help:if a test)

další?:)

Zaver

- Díky za pozornost..