# OS Linux - Tvorba skriptů. Práce s textovými soubory, filtrace, Regulární výrazy.

Obsah obrázku text, snímek obrazovky, logo, Písmo

Popis byl vytvořen automaticky

## Tvorba scriptu

Bash script je série příkazů napsaná v souboru. Ty jsou následně prováděny basham. Program běží řádek po řádce. Využíváme jej často když si chceme ulehčit práci s nějakou opakující se operací.

Bash je přímo v komandlajně.

* **#!/bin/bash:** it used bash as a interpreter (you can use all commands that you are using in bash)
* **#!/bin/sh:**It is used to execute the file using sh, which is a Bourne shell, or a compatible shell
* **#!/bin/csh:**It is used to execute the file using csh, the C shell, or a compatible shell.
* **#!/usr/bin/perl -T:**It is used to execute using Perl with the option for taint checks.
* **#!/usr/bin/php:**It is used to execute the file using the PHP command-line interpreter.
* **#!/usr/bin/python -O:**It is used to execute using Python with optimizations to code.
* **#!/usr/bin/ruby:**It is used to execute using Ruby.

Na začátek scriptu je potřeba definovat interpretera daného kódu. Nejčastěji pokud budeme psát script. #! … se nazývá **shabang** a používá se úplně na začátku skriptu.

### Proměné

* **#!/bin/bash**
* # A simple variable example
* greeting=Hello
* name=Tux
* echo $greeting $name

### Input

* **#!/bin/bash**
* echo "Enter a numner"
* read a
* echo "Enter a numner"
* read b
* var=$((a+b))
* echo $var

### Condition

Syntax:

if [[ condition -a condition ]]

then

statement

elif [[ condition -o condition]]; then

statement

else

do this by default

fi

* read x
* read y
* if [ $x -gt $y ]
* then
* echo X is greater than Y
* elif [ $x -lt $y ]
* then
* echo X is less than Y
* elif [ $x -eq $y ]
* then
* echo X is equal to Y
* fi

### Loops

* for i in {1..5}
* do
* echo $i
* done
* i=1
* while [[ $i -le 10 ]] ; do
* echo "$i"
* (( i += 1 ))
* done

## Práce s textovými soubory

Mkdir rmdir pro práci se složkami

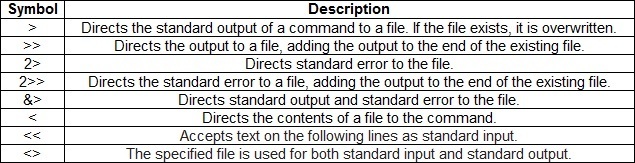
### Vytvoření

* **touch [filename]**
* **nano [filename]**
* **cat > [filename]**
* **vi [filename]**

### Ostatní

* Smazání … **rm [filename]**
* Kopírování … **cp [source file/dir] [destination file/dir]**
* Přesun … **mv [source file/dir] [destination file/dir]**
* Najít … **find [path to search] –name [file/dir]**
* Zobrazení obsahu:
  + **cat** **[filename] … all the contetn of the file**
  + **head [filename] … first 10 lines**
  + **tail [filename] … last 10 lines**

### Přesměrování



* (0) stdin (standard input) devie is the keyboard … <
* (1) stdout (standard output) device is the screen … >
* (2) stderr (standard error)

1. Overwrite

* “>” – standard input
* “<” – standard output

2. Appends

* “>>” – standard output
* “<<” – standard output

3. Merge

* “p>&q” – merge output from stream p with stream q
* “p>&q” – merge input from stream p with stream q

Redirecting STDERR **ls -l /root/ 2>ls-error.log**

Redirecting STDOUT/STDERR **ls -l /root/ >ls-error.log 2>&1**

Redirecting STDIN/STDOUT to a file **sort <domains.list >sort.output**

Redirecting STDIN/STDOUT to different files **sudo ls -l /root/ 2 > ls-error.log 1 > ls.log**

## Pipe

Pajpa redirektuje výstup jednoho příkazu do vstupu jiného. Pajpy můžeme řetězit do nekonečna.

* Grep … patern a expresion vyhledávájící komand
* Sort … seřadí řádky
  + -d … dictionary order sort
  + -n … arithmetic/numeric order
  + -r … reverse order
  + -k … n Sorts on field n
  + -b … ignore leading blanks
  + -m … Merge (not sort)
* Unique … pokud se prvky opakují vybere je jen jednou
  + -c – -count : It tells how many times a line was repeated by displaying a number as a prefix with the line.
  + -u – -unique : It allows you to print only unique lines.
  + -i – -ignore case : By default, comparisons done are case sensitive but with this option case insensitive comparisons can be made.
* Tr … maže nebo přepisuje chraktery v souboru
  + -d … Deletes each character from standard input that is contained in the string specified by String1.
  + tr '{}' '()' < textfile > newfile
* More … zobrazí content s pageing control
* Less … zobrazeni kde je možné scrolovat
* Nl … line numbering
* Wc … number of lines, word count, byte and characters count

cat /etc/group | egrep -i "[0-9]+:.+" | sort | nl

cat /etc/group | nl | egrep -i "[0-9]+:.+" | sort -d -r

## Grep

grep – i (ignore case sensitive) -E ‘regex’ [file]

egrep -i ‘regex’ [file]

egrep == grep -E (extended)

Můžeme spustit process na pozadí za pomocí ampersantu … gedit &

