

Linux – zjednodušený učební text pro výuku robotiky



Přihlášení

Z Windows se hlásíme za pomoci programu PuTTY, kam zadáme adresu student.prf.jcu.cz. Doporučuji nastavit Encoding na UTF-8, ostatní není nutné měnit. Doporučuji nastavit si záložku.

Z Linuxu (a Mac OS X) se připojíme pomocí programu ssh. Nejlépe takto:

```
ssh uzivatel@student.prf.jcu.cz
```

Přenos souborů

Z Windows pro přenos souborů na a z serveru student použijeme program WinSCP.

Z Linuxu (a Mac OS X) provedem přenos pomocí scp

Z serveru student: `scp uzivatel@student.prf.jcu.cz:/cesta/soubor .`

Např: `scp novakj00@student.prf.jcu.cz:uloha1.txt .`

Na server student: `scp soubor uzivatel@student.prf.jcu.cz:`

Např: `scp test novakj00@student.prf.jcu.cz:`

Prvky ochrany uživatelů v UNIXu

Pro přihlášení k UNIXu je nutný uživatelský účet, který obsahuje následující informace:

- Jméno účtu a heslo
- Skupina a číslo uživatele
- Identifikace uživatele (jméno, příjmení...)
- Domácí adresář (/home/jmeno)
- Interpret příkazů shell (bash)
- Další informace (datum a místo posledního přihlášení)

Každý uživatelský účet je reprezentován jednoznačným identifikačním číslem (**UID**) a patří do

nejméně jedné skupiny uživatelů (reprezentované jednoznačným identifikačním číslem **GID**).

Prvky ochrany se určují zvlášť pro:

- majitele souboru (u - user)
- skupinu uživatelů (g - group)
- ostatní, svět (o - other)
- všechny (a - all)

Přidělují se práva pro

- soubor
 - čtení (r)
 - zápis (w)
 - provedení (spuštění - x)
- adresář
 - výpis obsahu (r)
 - přejmenování (smazání) (w)
 - vstup do adresáře (x)

Existuje jeden speciální uživatel zvaný superuživatel (root), který má právo na cokoliv.

Tyto prvky lze rozšířit pomocí rozšíření ACL (Access Control List).

Struktura adresářů

UNIX má jednotnou strukturu adresářů, která je popsána v normách POSIX a POSIX-2. Aby byl systém považován za UNIX musí této normě vyhovovat.

Typický strom adresářů Linuxu vypadá následujícím způsobem:

- **/boot** – Adresář obsahující jádro OS a zavaděč systému (obvykle GRUB)
- **/bin** – Základní systémové programy a příkazy
- **/dev** – Adresář speciálních souborů pro obsluhu periférií
- **/etc** – Adresář konfiguračních souborů systému
- **/lib** – Adresář knihoven
- **/mnt, /media** – Pomocné adresáře pro připojování dočasných systémů souborů (/mnt pro ruční připojení, /media pro automatické)
- **/tmp** – Veřejný adresář pro pomocné a dočasné soubory (temp)
- **/home** – Adresář s domovskými adresáři uživatelů
- **/usr** – Adresáře se soubory, které typicky z kapacitních důvodů nejsou v kořenovém

adresáři. V současné době již často tvoří adresář `/bin` link na `/usr/bin`, `/lib` na `/usr/lib` atd.

- **`/usr/local/`** – Lokálně instalované programy, knihovny atd.
- **`/usr/sbin`, `/sbin`** – Systémové programy určené zpravidla superuživateli
- **`/var`** – Adresář pracovních a administrativních souborů systému, logů, webových stránek (obvykle `/var/www`), databází atd.
- **`/opt`** – adresář s dalšími programy (obvykle komerčními)

Základní příkazy

Nápověda

- **`man`** nápověda k příkazům (zkus `man man`)
- **`apropos` *vyraz*** – kde lze najít nápovědu k pojmu *vyraz*

Ovládání příkazu `man`

`PgUp`, `b` – stránka vzhůru

`PgDn`, mezerník – stránka dolů

šipky nahoru a dolů – pohyb o řádek

`g`, `G` – skok na konec a začátek souboru

`/` – vyhledávání dopředu

`n` – následující výskyt

`N` – předchozí výskyt

`&` – zobraz pouze řádky se zadaným výrazem

`?` – vyhledej směrem vpřed

`h` – nápověda

`q` – konec

Příkazy pro práci s adresáři

- **`cd`** změna adresáře
 - `cd adresar1`
 - `cd ..` - přechod do nadadresáře
 - `cd` (anebo `cd ~`) – skok do domácího adresáře
 - Příklad: `cd ../adresar2` – přechod do adresáře, který se nachází v nadadresáři
- **`mkdir`** vytvoření adresáře
- **`rmdir`** smazání adresáře
- **`pwd`** výpis aktuálního adresáře

Příkazy pro práci se soubory

- **touch soubor** – vytvoření prázdného souboru se jménem soubor. Pokud soubor již existuje, nastaví se mu čas posledního přístupu na nyní.
- **ls** výpis souborů a adresářů
 - `ls -l` dlouhý výpis souborů
 - `ls -a` výpis i skrytých souborů a adresářů
 - lze kombinovat `ls -la` (anebo `ls -al`, `ls -a -l`, `ls -l -a`)
- **cp** kopírování souborů
 - zápis `cp odkud kam`
- **mv** přesun souborů
 - zápis `mv odkud kam`
- **rm** smazání souborů, pozor *mazání je nevratné*.
 - `rm -rf adresar` smaže adresář včetně obsahu
- **which** umístění spustitelného souboru
- **cat** prohlédnutí obsahu textového souboru
- **less** prohlédnutí obsahu textového souboru se stránkováním, ovládání stejné jako u programu **man**.

Práva k souborům a adresářům

- **chmod** změna práv souboru
 - **chmod práva soubor** – práva tři osmičková čísla – každé je součet práv 1(spouštění) + 2(zápis) + 4(čtení), čísla jsou vlastník, skupina, ostatní
 - Příklad: `chmod 755 program`
 - **chmod komu(+/-)práva soubor**
 - komu – u (vlastník), g (skupina), o (ostatní), a (všichni)
 - práva – r (čtení), w (zápis), x (spuštění – soubor, procházení – adresář)
 - Příklad: `chmod u+w soubor`, `chmod a+x adresar`

Linky

Vytvoření odkazu

```
ln [-s] zdroj link
```

-s vytvořit symbolický odkaz

zdroj – zdrojový soubor

link – název linku

Výpis velikostí podadresářů

- `du` výpis velikosti adresářů
 - `du -h --max-depth=1` – dobře čitelný výpis velikostí podadresářů

Psaní skriptů

Příkaz `echo` – výpis textu

```
echo Text
```

```
echo "Ahoj svete"
```

Struktura skriptu

- První řádek musí být něco jako `#!/bin/bash`
- Musí začít dvojicí znaků `#!` a pak následuje cesta k programu pomocí kterého se skript spouští.
- Ve skriptu lze použít libovolné příkazy.
- Bývá zvykem použít příponu `*.sh`, ale není to povinné.
- Spouští se:
 - buď příkazem `sh skript.sh`
 - nebo se nastaví práva po spuštění `chmod a+x skript.sh` a pak se spustí `./skript.sh`
 - pokud máme ve svém domovském adresáři adresář `bin` a spustitelný skript umístíme do něj lze jej pouštět i přímo `skript.sh`

Vyhledávání

grep

- Vyhledávání v souboru `grep řetězec soubor`

locate

Příkaz vyhledá všechny soubory na disku obsahující dané slovo. Lze užít hvězdičkovou konvenci. Využívá databázi, která se automaticky občerstvuje v noci. Lze jí obnovit i manuálně kdykoliv příkazem `updatedb`, který však vyžaduje práva roota.

Příklady:

```
locate acroread
```

```
locate *.doc
```

find

Příkaz s mnoha parametry. Ukázky na příkladě, všechny parametry si prohlédněte v manuálových stránkách. Příklady:

```
find /etc/default
```

```
find /usr -name acroread
```

```
find /etc -name *.conf
```

```
find /tmp -size +200k
```

```
find /usr -perm 755 (má právo na spuštění)
```

```
find /var -not -perm 111 (nemá právo spuštění)
```

```
find /tmp -not -perm 020 -size +20k (není právo skupiny na zápis (první nula je nepovinná))
```

```
find ~ -name *.bak -exec rm {} \;
```

Najde v domovském adresáři všechny soubory s příponou bak a smaže je. Symbol `{}` je zástupný symbol pro vyhledané soubory a adresáře. Příkaz za `exec` musí být buď celý v apostrofech nebo ukončený dvojicí znaků `\;`

Textové editory v příkazovém řádku

McEdit

Textový editor programu Midnight Commander. Lze volat i nezávisle.

```
mcedit soubor
```

Vim

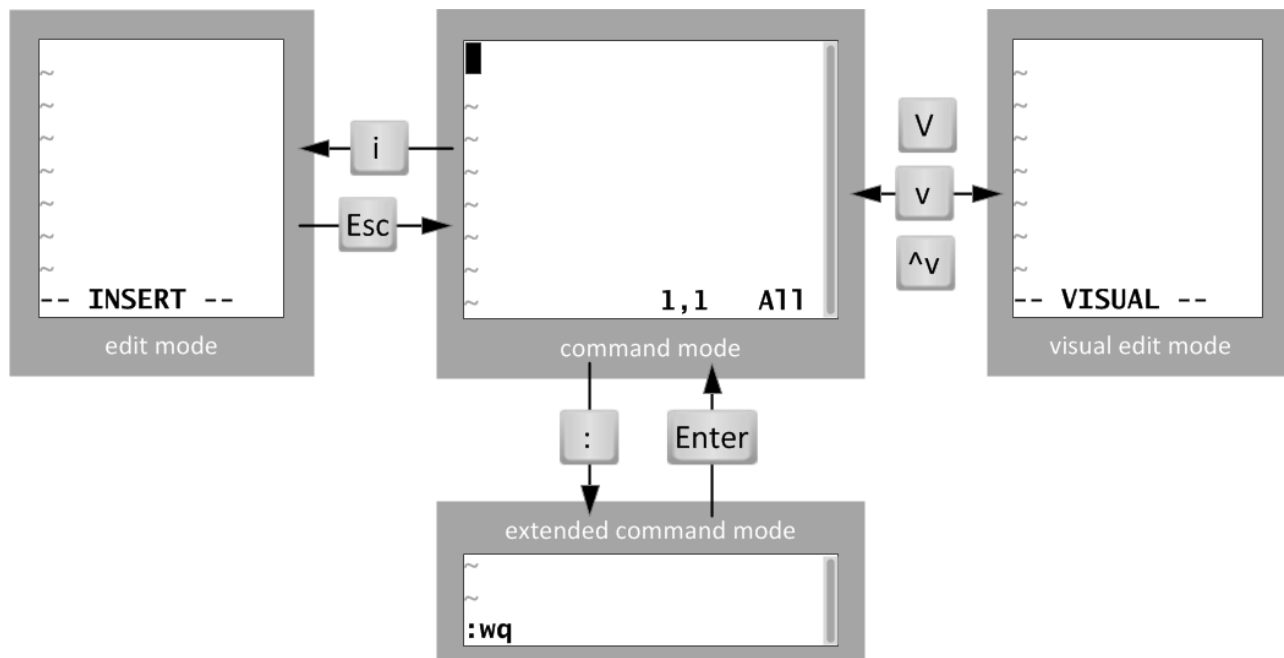
Motto: „Víš jaký je nejlepší editor?“ „Vim.“

Textový editor, který existuje snad na všech operačních systémech. V Linuxu by měl být všude součástí základní instalace. Někdy je jeho odlehčená verze **vi**. Ovládání pro začátečníka velmi obtížné. Vyplatí se minimálně si pamatovat sekvenci kláves pro ukončení bez uložení:

„Esc : q !“.

Editor má tři režimy:

- příkazový – editor ovládáme pomocí klávesových zkratk
- vkládací – pořizujeme text
- příkazový řádek – píšeme příkazy (např. `syntax on`)
- vizuální – výběr textu



Klávesy pro přechod mezi jednotlivými režimy jsou rovněž v následujícím obrázku. Z příkazového do vkládacího lze obvykle přejít i klávesou Insert (má-li jí váš počítač).

Vyplatí se ještě pamatovat si klávesovou zkratku `Esc : wq` (anebo `Esc : x`) – ulož aktuální stav a skonči.

[Nejlepší český manuál je zde](#)

[Základní manuál vim \(root.cz\)](#)

[Editor VIM prakticky](#)

Nano

Velice jednoduchý editor, který také obvykle bývá součástí většiny distribucí. Návod je třeba [zde](#).

Uložení a ukončení práce klávesou F2.

Spouštění příkazů

Přesměrování vstupu a výstupu

- `Příkaz > abc` – výstup příkazu jde do souboru `abc`. Pokud soubor neexistuje je vytvořen. Pokud existuje je přepsán.
- `Příkaz >> abc` – výstup příkazu jde do souboru `abc`. Pokud soubor neexistuje je vytvořen. Pokud existuje je výstup přidán na jeho konec.
- `Příkaz < abc` – vstupem příkazu je text zapsaný v souboru `abc`.
- `Příkaz1 | Příkaz 2` – výstup Příkazu1 je použit jako vstup Příkazu2 (tzv. Roura).

Posloupnost příkazů

- `Příkaz1 ; Příkaz2` – Příkaz2 se provede po skončení Příkazu1.
- `Příkaz1 && Příkaz2` – Příkaz2 se provede jen tehdy, pokud Příkaz1 skončí úspěšně.
- `Příkaz1 || Příkaz2` – Příkaz2 se provede jen tehdy, pokud Příkaz1 skončí neúspěšně.

Práce s textovými soubory, archivace

file soubor

- informace o souboru

head [přepínače] soubor

- výpis prvních deset řádků souboru
- `-5` – prvních pět řádků
- `-c6` – prvních šest bytů souboru

tail [přepínače] soubor

- posledních deset řádků souboru
- přepínače jako u `head`
- `-f` – soubor zůstane otevřený a zobrazují se nové řádky (hodí se např. u logů)

Práce s procesy v UNIXu

ps

výpis procesů

ps axu – přepínače (lze použít i syntaxi **ps -axu**) – na pořadí přepínačů nezáleží

- x - rozšířený výpis
- a - všechny procesy
- u - všech uživatelů

utíká-li výpis, lze použít `ps axu | less`

chceme-li konkrétní proces, použijeme `grep`, např.: `ps axu | grep mc`

výpis procesů uživatele: `ps -lu uživatel`

top

Program na výpis běžících procesů seřazený podle toho jak aktuálně využívají systémové zdroje počítače.

Ukončuje se klávesou q.

htop

Ještě lepší grafický manažér procesů. Ovládání podobné programu mc, pomocí kláves F1 až F10. Procesy je možné řadit dle různých kritérií, vyhledávat, měnit prioritu i ukončovat (podle různých signálů).

kill

ukončení procesu

`kill číslo procesu`

nefunguje-li ukončení, lze použít ve tvaru `kill -9 číslo procesu`