

C2110 Operační systém UNIX a základy programování

9. lekce

Linux aneb co se nevezlo do úvodu

Petr Kulhánek, Jakub Štěpán

kulhanek@chemi.muni.cz

Národní centrum pro výzkum biomolekul, Přírodovědecká fakulta
Masarykova univerzita, Kamenice 5, CZ-62500 Brno

Obsah

- **Procesy**
- **Souborový systém**
- **Vzdálený přenos souborů**
- **Kompresa**
- **Archivy**

Procesy

Procesy na popředí a pozadí

Spouštění aplikací na popředí

```
$ gimp
```

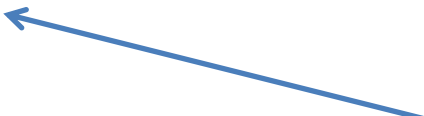
procesy běžící **na popředí blokují terminál**, protože používají jeho standardní vstup a výstup

Spouštění aplikací na pozadí

```
$ gimp &
```

procesy běžící **na pozadí neblokují terminál**

na konec (za argumenty a přesměrování) příkazu uvedeme ampersand



Terminál (užitečné klávesové zkratky):

- Ctrl+Z** pozastaví běh procesu, další osud procesu lze kontrolovat pomocí příkazů:
- jobs** vypíše procesy, které shell spravuje
- bg** přesune proces do pozadí
- fg** přesune proces do popředí
- disown** zruší vazbu procesu na shell (proces není ukončen při ukončení shellu)

Příklady

```
$ kwite          # spustí aplikaci kwite na popředí
^Z              # pozastaví aplikaci (Ctrl+Z)
[1]+  Stopped                kwite
$ jobs # vypíše aplikace na pozadí nebo pozastavené aplikace
[1]+  Stopped                kwite
$ bg %1        # pozastavenou aplikaci 1 spustí na pozadí
[1]+  kwite &
$ jobs
[1]+  Running                kwite &
$ fg %1        # aplikaci 1 běžící na pozadí přesune do popředí
kwite          # terminál je zablokovaný, nutno použít (Ctrl+Z)
```

Specifikace úlohy:

%n nebo %name (někdy stačí i samotné číslo úlohy)



číslo nebo jméno z tabulky úloh (**jobs**)

Signály a procesy

Terminál (užitečné klávesové zkratky):

Ctrl+C běžícímu procesu zašle signál SIGINT (Interrupt), proces je ve většině případů násilně ukončen

Příkaz kill:

```
$ kill [-signal] PID
```

číslo procesu, kterému se má signál zaslat (lze zjistit příkazem **ps**, **top**, **pstree**)

specifikace signálu: -N (číslo signálu), -NAME (jméno signálu), -SIGNAME (SIG+jméno signálu)

Užitečné signály:

TERM	15	žádost o ukončení (proces na signál může reagovat)
INT	2	žádost o přerušení (ekvivalent Ctrl+C) (proces na signál může reagovat)
KILL	9	ukončení (proces nemůže signál ignorovat, je násilně ukončen)
STOP		pozastaví proces (proces nemůže signál ignorovat) (ekvivalent Ctrl+Z)
CONT		obnoví běh pozastaveného procesu (proces nemůže signál ignorovat)

Příklady

```
$ ps -u kulhanek
```

```
PID TTY          TIME CMD
...
5440 pts/8      00:00:00 bash
5562 pts/8      00:00:00 kwrite
5566 pts/8      00:00:00 ps
```

```
$ kill 5562 # ukončí proces kwrite
```

```
$ kwrite # spustí aplikaci kwrite na popředí
[1]+  Stopped                  kwrite
```

```
jiný terminál
$ ps -u kulhanek
PID TTY          TIME CMD
...
8401 pts/1      00:00:00 kwrite

$ kill -STOP 8401 # pozastaví kwrite
```

Příklady

```
$ time kwrite
```

```
real    0m3.188s
```

```
user    0m0.349s
```

```
sys     0m0.044s
```

skutečná doba běhu procesu

spotřebovaný strojový čas procesem

spotřebovaný strojový čas procesem v systémových volání

```
$ time sleep 10
```

```
real    0m10.002s
```

```
user    0m0.000s
```

```
sys     0m0.002s
```

```
$ time sleep 0.01
```

```
real    0m0.012s
```

```
user    0m0.000s
```

```
sys     0m0.002s
```

spuštění procesu má určitou režii

Přehled příkazů

top	průběžně zobrazuje procesy setříděné podle zátěže procesoru (ukončení klávesou q)	
ps	vypíše procesy běžící v daném terminálu nebo podle zadaných specifikací (<code>ps -u user_name</code>)	
pstree	vypíše procesy (stromový výpis)	
type	vypíše cestu k standardní aplikaci/příkazu	
kill	zašle signál procesu, lze použít k ukončení problematických programů	
time	vypíše délku běhu procesu	
ssh	spustí příkaz na vzdáleném počítači	
jobs	vypíše procesy na pozadí	
fg	převede proces do popředí	
bg	převede proces do pozadí	
sleep	spustí proces, který čeká po zadanou dobu	
disown	odpojí proces od terminálu	pro pokročilé
nohup	spustí proces bez interakce s terminálem (C2115)	
wait	čeká na dokončení procesů na pozadí (C2115)	

Souborový systém

Kvóty

Pro vaše domovské adresáře jsou nastaveny kvóty na využití diskového prostoru na diskovém oddílu **wolf.wolf.inet:/home/**. Aktuální stav zaplnění a nastavení kvót lze zjistit příkazem **quota**:

```
[kulhanek@wolf ~]$ quota -vs
Disk quotas for user kulhanek (uid 18773):
  Filesystem  blocks      quota  limit  grace  files      quota  limit
wolf.wolf.inet:/home/
                1550M    1954M   2051M          20453         0         0
```

Aktuální využití

Kvóta, kterou lze dočasně překročit.

Tvrдый limit, který již nelze překročit.

Překročení kvóty může vést k **nefunkčnímu přihlášení** pomocí grafického rozhraní. V tomto případě se přihlaste v textovém terminálu (např: Ctrl+Alt+F1) a přesuňte soubory na jiný diskový oddíl (např. dočasně do adresáře /scratch/username nebo smažte nepotřebné soubory).

Diskové zařízení

Přehled o využití souborových systémů, o diskových zařízeních a jejich přípojných bodech poskytuje příkaz **df**.

Filesystem	Type	Size	Used	Avail	Use%	Mounted on
/dev/mapper/server1-root	ext4	15G	8.4G	5.5G	61%	/
none	tmpfs	4.0K	0	4.0K	0%	/sys/fs/cgroup
udev	devtmpfs	3.9G	4.0K	3.9G	1%	/dev
tmpfs	tmpfs	793M	888K	792M	1%	/run
none	tmpfs	5.0M	0	5.0M	0%	/run/lock
none	tmpfs	3.9G	952K	3.9G	1%	/run/shm
none	tmpfs	100M	36K	100M	1%	/run/user
/dev/mapper/server1-vbox	ext4	64G	52G	9.5G	85%	/win
/dev/mapper/server1-scratch	ext4	598G	2.8G	565G	1%	/scratch
wolf.wolf.inet:/software/ncbr	nfs4	197G	156G	33G	83%	/software/ncbr
wolf.wolf.inet:/home/	nfs4	493G	371G	98G	80%	/home

zařízení

typ souborového systému

přípojný bod

Typy souborových systémů:

ext3, ext4 third / fourth extended filesystem (nativní souborový systém linuxu)

nfs3, nfs4 network filesystem

vfat Virtual File Allocation Table (souborový systém používaný MS Windows)

ntfs New technology File System (vyvinul Microsoft pro svoje operační systémy)

!!! nejsou case-sensitive!!! – pozor při kopírování souborů lišících se velikostí znaků

USB disk

USB disky se **připojují** automaticky v grafickém prostředí do svazku **/media/username**.

```
[kulhanek@wolf01 ~]$ df -Th
```

Filesystem	Type	Size	Used	Avail	Use%	Mounted on
.....						
wolf.wolf.inet:/home	nfs4	280G	164G	102G	62%	/home
/dev/sdg1	vfat	962M	841M	122M	88%	/media/kulhanek/B19A-1CA2

Disk lze **odpojit** v grafickém prostředí nebo příkazem **umount**. Argumentem příkazu je přípojný bod zařízení.

```
[kulhanek@wolf01 ~]$ umount /media/kulhanek/B19A-1CA2
```

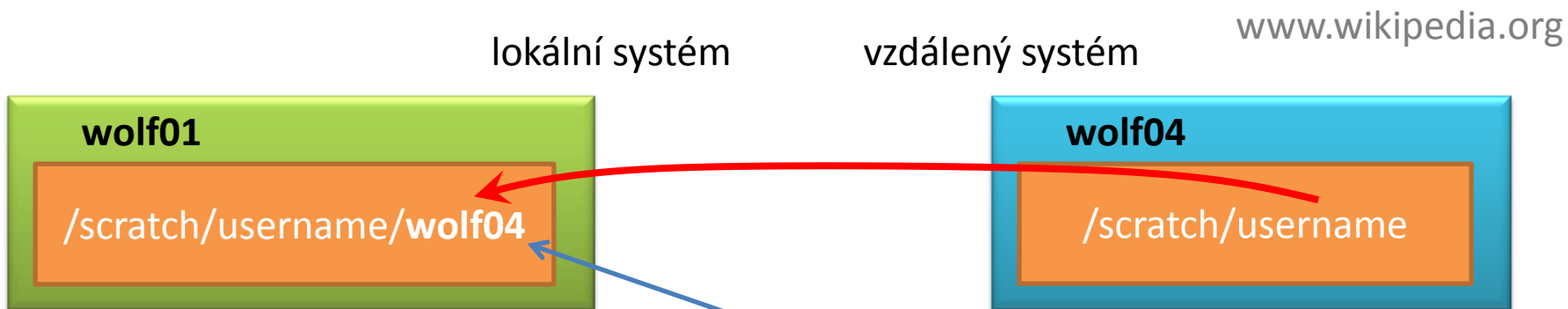
Disk lze odpojit pouze pokud není využíván (nesmí být otevřen žádný soubor, žádný proces nesmí mít jako pracovní adresář nastaven (pod)adresář z přípojného bodu včetně přípojného bodu). Přehled procesů využívajících daný adresář (přípojný bod) lze získat příkazem **lsdf** (nebo **fuser**).

```
[kulhanek@wolf01 ~]$ lsdf /media/kulhanek/B19A-1CA2/
```

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE/OFF	NODE	NAME
bash	31521	kulhanek	cwd	DIR	8,97	4096	518	/media/kulhanek/B19A-1CA2/GoslarFinal
bash	31893	kulhanek	cwd	DIR	8,97	4096	518	/media/ kulhanek/ B19A-1CA2/GoslarFinal
vi	32011	kulhanek	cwd	DIR	8,97	4096	518	/media/ kulhanek/ B19A-1CA2/GoslarFinal

sshfs

SSHFS neboli SSH File System slouží **k připojení vzdáleného souborového systému do lokálního adresářového stromu** prostřednictvím šifrovaného spojení SSH. Na serveru není nutné kromě SSH nic dalšího konfigurovat. Na klientovi je třeba nainstalovat balíček sshfs.



Připojení:

```
$ mkdir /scratch/username/wolf04
$ sshfs wolf04:/scratch/username /scratch/username/wolf04
```

vzdálený souborový systém

vytvoření přípojného bodu (měl by být prázdný adresář)

lokální přípojný bod

Odpojení:

```
$ fusermount -u /scratch/username/wolf04
```

Cvičení

lokální systém

1. Běžte do adresáře /scratch/username
2. Vytvořte adresář "remote"
3. Do adresáře připojte vzdálený souborový systém /scratch/username
4. Zkontrolujte připojení příkazy df a mount
5. Běžte do adresáře "remote"
6. Vytvořte v něm soubor pokus.txt

7. Vypište obsah adresáře, jakou velikost má soubor pokus.txt?
8. Odpojte vzdálený systém

vzdálený systém

1. Běžte do adresáře /scratch/username
2. Vypište jeho obsah
3. Smažte soubor pokus.txt

**jako vzdálený systém použijte
sousední stanici
použijte dva terminály**

Přehled příkazů

Souborový systém:

cd	změní aktuální pracovní adresář
pwd	vypíše cestu k aktuálnímu pracovnímu adresáři
ls	vypíše obsah adresáře

mkdir	vytvoří adresář
rmdir	smaže adresář (musí být prázdný)
cp	zkopíruje soubor či adresář
mv	přesune soubor či adresář
rm	odstraní soubor či adresář
find	vyhledává soubory či adresáře
scp	vzdálené kopírování souborů či adresářů

základní operace

id	vypíše zařazení uživatele do skupin
getent	vypisuje informace o uživateli, skupinách uživatelů a další informace
chmod	změní přístupová práva k souboru či adresáři
chown	změní vlastníka souboru či adresáře
chgrp	změní přístupovou skupinu souboru či adresáře
umask	výchozí přístupová práva pro nově vytvářené soubory či adresáře

přístupová práva

Přehled příkazů

Souborový systém (pokračování):

pokročilé funkce

quota	vypíše informace o nastavených kvótách pro přípojně body
du	vypíše velikost adresáře nebo souborů
stat	vypíše podrobné informace o souboru či adresáři
df	vypíše informace o připojených diskových oddílech
lsuf	vypisuje procesy, které mají otevřeny soubory/adresáře na přípojném bodu (adresáři)
sshfs	připojí vzdálený souborový systém do lokálního adresářového stromu pomocí ssh protokolu
mount	připojí zařízení do lokálního adresářového stromu, popř. vypíše seznam připojených zařízení (připojení datových úložišť z Metacentra – C2115)

Vzdálený přenos souborů

- scp
- WinSCP
- wget (domací úkol)

Vzdálené kopírování

Ke vzdálenému kopírování slouží příkaz **scp**.

Syntaxe:

[] - možno vynechat

```
$ scp [-r] zdroj cil
```

Zdroj a cíl může být soubor nebo adresář. V případě kopírování adresářů je nutno použít volbu **-r** (recursive).

Vzdálený cíl nebo host se identifikuje názvem stroje odděleného od jména souboru či adresáře dvojtečkou.

```
[user@]hostname : [cesta/] soubor
```

Příklady použití:

```
$ scp pokus.txt wolf01.wolf.inet:/scratch/kulhanek
```

```
$ scp wolf01.wolf.inet:/scratch/kulhanek/pokus.txt .
```

Cvičení

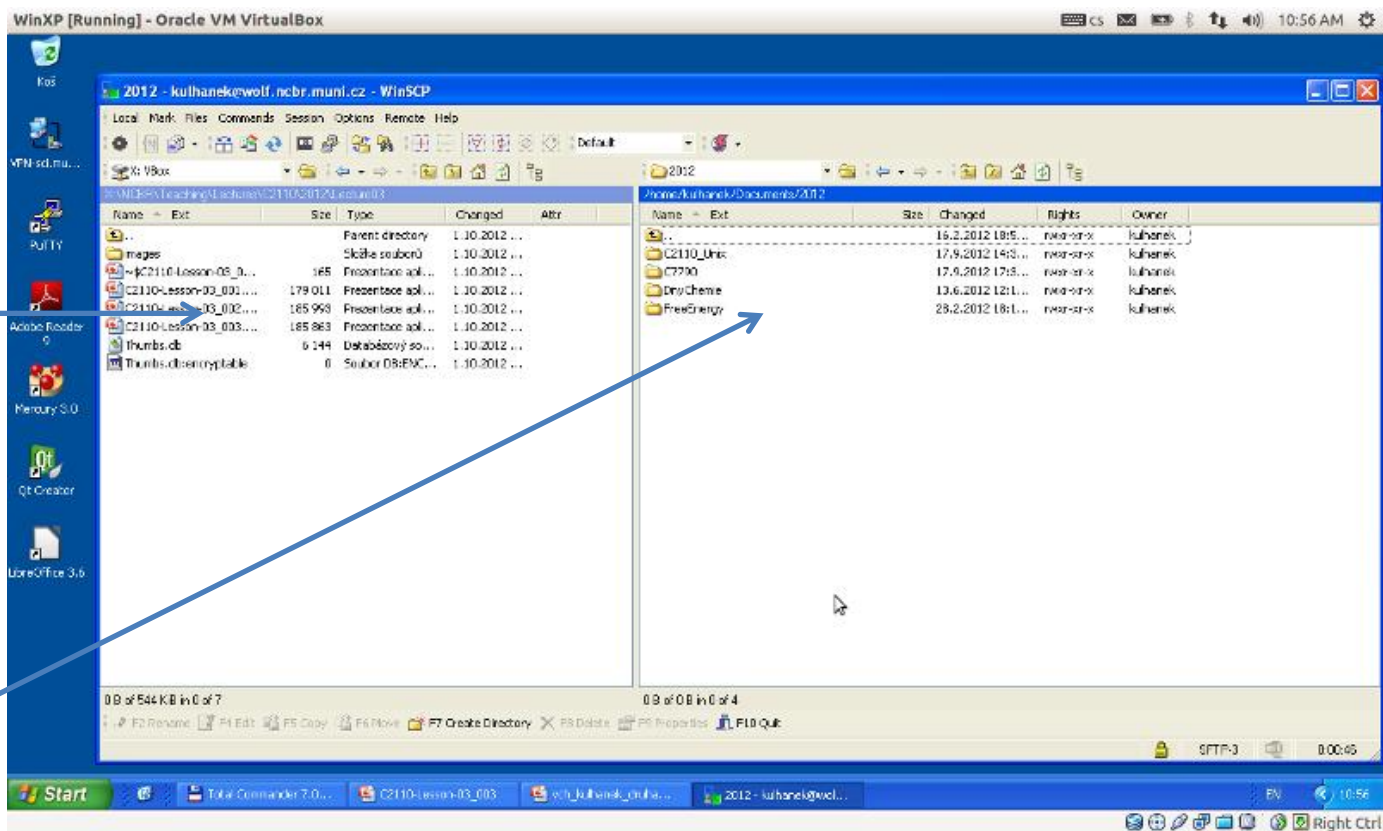
1. Vytvořte kopii adresáře ~/Documents do adresáře /scratch/username
2. Překopírujte obsah adresáře /scratch/username/Documents na vzdálený stroj dle vašeho výběru do adresáře /scratch/username/wolfXX, kde wolfXX je číslo zdrojového stroje. Ke kopírování použijte příkaz scp
3. Smažte adresáře /scratch/username/Documents a /scratch/username/wolfXX

použijte dva terminály

WinSCP

WinSCP <http://winscp.net/eng/docs/lang:cs>

Program pro přenos souborů mezi MS Windows a počítači podporující SFTP či SCP protokoly (převážně unixového a linuxového typu).



lokální stroj

vzdálený stroj

Textové soubory MS Win ↔ Linux

Textové soubory vytvořené pod MS Windows a Linuxem **nejsou** zcela **kompatibilní**, protože každý operační systém používá jiné kódování **konce řádku**.

Linux: \n (line feed 0x0A)

MS Windows: \r+\n (carriage return 0x0D, line feed 0x0A)

Ke konverzi souborů lze použít programy **d2u** a **u2d** (na klastru WOLF).

1) Aktivace modulu cats

```
$ module add cats
```

2) Konverze MS Windows => Linux

```
$ d2u soubor.com
```

3) Konverze Linux => MS Windows

```
$ u2d soubor.log
```

Doplňující informace:
<http://en.wikipedia.org/wiki/Newline>

Cvičení

1. Z adresáře /home/kulhanek/Downloads/ si překopírujte do vašeho domovského adresáře soubor 1SS9.pdb.
2. Spustíte virtuální stroj s MS Windows XP (/win/win).
3. Spustíte aplikaci WinSCP.
4. Do virtuálního stroje stáhněte soubor 1SS9.pdb z vašeho domovského adresáře. Soubor otevřete v programu Poznámkový blok (Notepad). Zobrazí se obsah souboru správně?
5. Opravte kódování konců řádků v souboru 1SS9.pdb pomocí příkazu u2d (modul cats) a soubor znovu otevřete ve virtuálním stroji v programu Poznámkový blok. Zobrazí se obsah souboru správně?

Komprese

- **Bezeztrátová**
- **Ztrátová**

Kompresa

Kompresa (komprimace) je postup, kterým dochází k snížení velikosti dat (souborů). Docílí se toho vyhledáváním **redundantních** nebo **nepodstatných** informací v datech, které jsou pak ukládány efektivněji. Podle typu kompresního algoritmu, lze kompresi dat rozdělit do dvou základních kategorií:

- **ztrátová komprese** – dochází k nevratné ztrátě některých nepodstatných informací, což je většinou tolerováno při kompresi obrazových či zvukových dat
- **bezeztrátová komprese** – nedochází k žádné ztrátě původní informace, komprimovaná data lze obnovit do původního stavu, míra komprese je několikanásobně nižší než u ztrátové komprese

Obnova komprimovaných dat se nazývá **dekomprese**.

Kompresní poměr udává kvalitu komprese. Udává se jako poměr velikosti původních dat (v bytech) vůči velikosti komprimovaných dat.

Ztrátová komprese

Programy pro ztrátovou kompresi a dekompresi:

- **mplayer**
- **mencoder**
- **convert** (Image Magick)
- a další ...

Převod obrázku ve formátu PNG (Portable Network Graphics) do JPEG (Joint Photographic Experts Group):

```
$ convert input.png -quality number output.jpeg
```

↑
využívá bezeztrátové
komprese

↑
míra kvality výsledného obrázku
od 1 (nejhorší kvalita s největší
kompresí) po 100 (nejlepší kvalita
s nejhorší kompresí)

↑
využívá ztrátovou
kompresi

Cvičení

1. Z adresáře `/home/kulhanek/Data/Komprese` zkopírujte obrázek **test.png** do vašeho domovského adresáře.
2. Jakou velikost má soubor obrázku v bytech?
3. Proved'te **ztrátovou kompresi** obrázku do formátu **jpeg**. Ke kompresi použijte kvalitu **10, 50 a 90**. Výsledné obrázky ukládejte každý zvlášť.
4. Porovnejte **vizuální kvalitu** komprimovaných obrázků (příkaz `display`).
5. Jaký je **kompresní poměr** pro kvalitu 10 a 90?

Bezeztrátová komprese

Programy pro **bezeztrátovou** kompresi a dekompresi:

- **gzip/gunzip**
- **bzip2/bunzip2**
- **zip/unzip**
- a další ...

Komprese textového souboru:

```
$ gzip soubor.txt
```

```
$ bzip2 soubor.txt
```

výsledný soubor se bude jmenovat
soubor.txt.gz

výsledný soubor se bude jmenovat
soubor.txt.bz2

Dekomprese komprimovaných dat:

```
$ gunzip soubor.txt.gz
```

```
$ bunzip2 soubor.txt.bz2
```

Kompresi či dekompresi lze uskutečnit tak, že výsledek je posílán **do standardního výstupu** (originální soubor pak zůstává nezměněn), např.:

```
$ bunzip2 --stdout soubor.txt.bz2 | wc
```

Cvičení

1. Z adresáře **/home/kulhanek/Data/Komprese** zkopírujte textový soubor **bu6_f.log** do vašeho domovského adresáře.
2. Jakou velikost má soubor obrázku v bytech?
3. Proveďte **bezeztrátovou kompresi** souboru pomocí programů **gzip** a **bzip2**. Který z programů dosahuje **vyššího kompresního poměru**?
4. Který z programů komprimuje soubor **rychleji**?

Archívy

- Typy
- Vytváření a rozbalování archivů

Archívy - tar

tar (zkratka z **anglického tape archiver neboli páskový archivovač**) je souhrnný název jednak pro souborový formát sloužící k uložení mnoha jednotlivých souborů, jednak pro jednoúčelové programy, které s tímto formátem pracují. Formát samotný vznikl v počátcích Unixu a až později byl standardizován v rámci normy POSIX. Původně pomáhal při archivaci souborů na páskových jednotkách, ale později se jeho užívání rozšířilo a dnes je používán zkrátka tam, kde je vhodné pro účely distribuce či archivace sloučit **více souborů do jednoho tak, aby se zachovaly informace o adresářové struktuře, přístupových právech a dalších attributech**, které běžně souborový systém obsahuje.

www.wikipedia.org

Rozbalení archivu:

```
$ tar xvf archiv.tar
```

Vytvoření archivu:

```
$ tar cvf archiv.tar adresar/
```

```
$ cd adresar
```

```
$ tar cvf /cesta/k/archiv.tar *
```

pokud jméno archivu obsahuje zakončení .gz nebo .bz2 tak je archiv automaticky dekomprimován nebo komprimován

Cvičení

1. Jaký význam mají volby **cvf** příkazu tar?
2. Jaký význam mají volby **xvf** příkazu tar?
3. Vytvořte archiv ze souborů uložených v adresáři: **/home/kulhanek/Data/Archive**
4. Jakou velikost má soubor obsahující archiv?
5. Proveďte kompresi archivu. Jaký je **kompresní poměr**?
6. Archiv rozbalte do adresáře **/scratch/vas_login/archiv**

Domácí úkoly



Stahování souborů z webu

Ke stahování souborů z webu lze využít příkaz **wget**. Vzdálené stroje musí soubory poskytovat pomocí protokolu **ftp**, **http** nebo **https**.

```
$ wget [-O output_name] url
```

velké O



url (uniform resource locator)
www identifikátor souboru



Příklady použití:

```
$ wget http://www.rcsb.org/pdb/files/1SS9.pdb
```

PDB kód struktury

