

# ZOS cvičení 06

L. Pešička, 2014

# Úkol – napište skript, který ...

- ▶ **-a f1**  
změní práva souboru f1 na rwx r-x r—
- ▶ **-b f1 f2 f3**  
vytvoří symbolický link na f1 s názvem f2  
a hardlink na f1 s názvem f3
- ▶ **-c f1**  
vypíše, zda je f1 blokové nebo znakové  
zařízení, případně že ani jedno z toho
- ▶ **-d f1**  
soubor f1 obsahující (jméno, příjmení, telefon)  
nahradí obsahem (příjmení, jméno, telefon)

# Rozdělení na 1.test

- ▶ Test bude na **30 minut**
- ▶ JIS kartu s sebou, na stole
- ▶ 1.skupina v čase x, 2. skupina x+60 min
  - Mějte prosím strpení při průtahu
- ▶ Výsledek ANO/NE, nadpoloviční počet bodů
- ▶ K dispozici
  - manuálové stránky **man**
  - Vytisknutý seznam základních příkazů (dokument z portálu) – dostanete, nenoste si nic 😊

# Doplnění a rozšíření příkazů

ls -F	Označí ve výpise soubory dle typů např: <a href="#">adresar/</a> <a href="#">text.txt</a> <a href="#">a.out*</a>
ls -R	Vypíše rekurzivně obsah podadresářů
du -h	Využití diskového prostoru (kolik každý soubor zabírá)
df -h	Využití diskového prostoru filesystemů (kapacitu jednotlivých filesystemů, kolik je volno)
groups	Vypíše skupiny, kterých je uživatel členem
uptime	Jak dlouho systém běží

# Vzdálené kopírování – scp

```
scp ceny.txt pesicka@proteus.fav.zcu.cz:~/
```

lokální soubor  
**ceny.txt**

vzdálený stroj:  
**proteus.fav.zcu.cz**  
  
uživatel na  
vzdáleném stroji:  
**pesicka**

domovský adresář  
uživatele, tedy ~

zkopíruje soubor na vzdálený stroj

# Ukázka: vzdálené vypnutí PC

```
#!/bin/bash
```

```
for i in 201 202 203
```

```
do
```

```
    echo "prave vypinam pc s ip: 10.1.2.$i "
```

```
    ssh 10.1.2.$i /sbin/halt
```

```
done
```



# Ukázka: vzdálené kopírování

```
#!/bin/bash
```

```
for i in 201 202 203 204  
do
```

```
  echo "prave instaluji iptables na pc s ip: 10.1.2.$i"
```

```
  scp /root/iptables.acm 10.1.2.$i:/root/iptables.acm
```

```
  ssh 10.1.2.$i iptables -P OUTPUT ACCEPT
```

```
  ssh 10.1.2.$i iptables -F
```

```
  ssh 10.1.2.$i iptables-restore < /root/iptables.acm
```

```
done
```

příkaz scp slouží ke kopírování souborů/adresářů mezi lokálním a vzdáleným strojem

# Vybrané proměnné prostředí

Vypíšeme příkazem set

\$HOME domovský adresář uživatele

\$PATH prohledávací cesta  
zde se hledají příkazy, odděleno :

\$USER jméno uživatele

\$UID číslo uživatele

\$PS1 primární prompt příkazové řádky

\$PS2 čeká na dokončení příkazu

...





# Informace o procesech

```
eryx2> ps -u pesicka -o pid,ppid,s,cmd
```

Procesy  
uživatelé  
pesicka

Položky PID, PID rodiče, stav,  
příkazová řádka

PID	PPID	S	CMD
21796	21791	S	sshd: pesicka@pts/6
21797	21796	S	-tcsh
21840	21797	S	/bin/bash
21843	21840	R	ps -u pesicka -o ...

# Odpočítávání příkazem for

```
#!/bin/bash
```

```
#odpocitavaci skript pomoci cyklu for
```

```
for (( c=10; c>=0; c-- ))
```

```
do
```

```
    echo "odpocitavam $c"
```

```
    sleep 1
```

```
done
```



# Porovnání řetězců

```
#!/bin/bash
```

```
#test přihlaseneho uzivatele
```

```
echo "Zadej svoje prihlasovaci jmeno"  
read JMENO
```

```
if test "$JMENO" = "$USER"
```

```
then
```

```
    echo "Zadal jsi tvoje prihlasovaci jmeno"
```

```
else
```

```
    echo "Zadaj jsi jine jmeno, nez pod kterym jsi  
    prihlasen"
```

```
fi
```



# Interaktivní skript – select

```
#!/bin/bash
```

```
select i in pivo vino ; do
    echo Odpovedel jsi: $i
    if [ -n "$i" ] ; then
        break
    fi
done
echo $REPLY
```

Zobrazí nabídku:

- 1) pivo
- 2) vino

Čeká na vstup  
uživatele, dokud  
nezvolí jednu z  
možností

\$i .. pivo nebo vino  
\$REPLY .. 1 nebo 2

# Přesměrování souboru celému cyklu

```
#!/bin/bash
```

```
while read VSTUP  
do  
    echo "$VSTUP"  
done < /etc/passwd
```

Bude číst postupně jednotlivé řádky ze souboru

# Ukázka změny koncovky souborů z .kuk na .puk

```
#!/bin/bash
```

```
for F in *.kuk  
do  
    mv $F ${F%.*}.puk  
done
```

```
date +"V Plzni dne %d. %m. %Y v %H : %M hod"
```

Manipulace s  
hodnotou proměnné

Ořízne pravou část po  
tečku

# Poznámka

- ▶ další možnosti použití závorek v bashi:

<http://stackoverflow.com/questions/2188199/how-to-use-double-or-single-bracket-parentheses-curly-braces>

# Subshell

- ▶ vykonání operací v subshellu bez ovlivnění prostředí aktuálního shellu – ( příkazy )
- ▶ `eryx2> pwd`
- ▶ `/users/p/pesicka/home`
- ▶ `eryx2> ( cd /tmp ; pwd )`
- ▶ `/tmp`
- ▶ `eryx2> pwd`
- ▶ `/users/p/pesicka/home`



# Třídění pomocí sort

- ▶ `sort -gr -k 2 ceny.txt`
  - `-g` .. Porovná data jako čísla
  - `-r` .. Třídí sestupně
  - `-k 2` .. Dle druhého sloupce

Setřídí např.:

snídaně 50

oběd 150

večeře 70

# Porovnání obsahu souborů

- ▶ `cmp ceny.txt ceny2.txt`
- ▶ `comm ceny.txt ceny2.txt`
- ▶ `diff ceny.txt ceny2.txt`

Pro aplikaci záplat se používá **patch**

Patch umí na základě souborů vytvořených diffem aplikovat změny do daného souboru

# Použití patche

soubor ceny.txt

jablka 100  
hrusky 50

soubor ceny2.txt

jablka 80  
hrusky 50

```
diff ceny.txt ceny2.txt > mujpatch  
patch ceny.txt mujpatch
```

co bude nyní obsahovat soubor ceny.txt?  
(apt-get install patch)

# Rozdělování souborů

- ▶ **split -l 3 ceny.txt ceny-**
  - Rozdělí soubor ceny.txt po 3 řádkách (volba -l 3)
  - Jednotlivé soubory budou ceny-aa, ceny-ab, ...
  - Další možností dělení je -b počet bytů
- ▶ **cat ceny-a\* > cenyznovu.txt**

Další možností, jak manipulovat s bloky dat, je využití příkazu dd, např.

```
dd if=/dev/zero of=pokus.txt bs=1k count=1
```

# Proudový editor sed

- ▶ **sed /<sup>^</sup>#/d s1.txt**
  - Odstraní řádky, které začínají #
  - Mezi /... / je regulární výraz
  - <sup>^</sup> značí nový řádek
  - instrukce d – smaže řádek
- ▶ **sed '1,2d' s1.txt**
  - Smaže řádky 1–2

Vědět o něm,  
že existuje

Více viz  
cvičení  
KIV/FJP

Literatura:

<http://melkor.dnp.fmph.uniba.sk/~zenis/prirucky/sed.html>

# ukázka sedu

# nahradi slovo obed slovem blaf

```
cat jidla.txt | sed -e s/obed/blaf/
```

# na konec kazdeho radku prida tecku

```
cat jidla.txt | sed s/$/./
```

# 100 brambor zmeni na brambor 100

```
sed -r -e "s/([0-9]+)[[:space:]]+([[:alpha:]]+)/\2 \1/g"
```

# Editor emacs

- ▶ Autorem je Richard Stallman
- ▶ Spuštění: **emacs pokus.txt**
- ▶ Ukončení: Ctrl+x, Ctrl+c
  - Potvrdit uložení souboru: y
- ▶ viz: <http://www.abclinuxu.cz/clanky/navody/emacs-jak-zacit>

# Spouštění a zastavování služeb

- ▶ `/etc/init.d/apache` start
- ▶ `/etc/init.d/apache` stop
- ▶ `/etc/init.d/apache` restart

Start, zastavení a restart služby

V adresáři `/etc/init.d` najdeme skripty, které umožní spustit či zastavit danou službu

Někdy je lepší použít akci restart, pokud bychom např. při zastavení služby ztratili přístup ke stroji a už bychom nebyli schopni provést start



# Samostatná práce

1. Napište skript, který spojí dva soubory, které budou zadány jako parametry a před každým souborem vypíše jeho jméno  
*př.: soubor1.txt obsah soubor2.txt obsah*
2. Modifikujte skript z bodu jedna, aby čísloval řádky v souboru:
  - a) Každý soubor čísluje zvlášť
  - b) čísluje řádky kontinuálně
3. Modifikujte skript z bodu 1, pro libovolný počet souborů
4. Modifikujte 3, aby vypisoval nejdříve "Ahoj, já jsem skript  
*jméno a tady je obsah souborů :*" ...
5. Dalším krokem k 4. bude vypsání počtu parametrů...

# Samostatná práce – pokračování

6. Ošetřete vstup, aby nebylo možno zadat víc jak 10 souborů
7. Popřípadě, že bude víc jak 10 souborů, vypište jedenáctý, který je první navíc...
8. Zkuste spustit skript na pozadí, co se stane, když mu zadáte / nezadáte argumenty ?
9. Upravte ho tak, aby soubory vypsal desetkrát po sobě s 5-ti vteřinovým intervalem
10. Zkuste si ve skriptu vypsat speciální znaky - *ahoj, já jsem skript "jmeno" a budu vypisovat soubory - můj první parametr \$1 je "jméno souboru" a poslední \$číslo je "jméno souboru"*

# Na co se podívat

- ▶ Courseware KIV/ZOS – Cvičení
  - řešení skriptu cv05
  - řešení skriptu cv06b
  - příklady k opakování (!!)
  - včetně řešení
  - jednoduchy skript