



8. cvičení

Práce s procesy a úlohami, příkaz test





- Práce s procesy
 - ps, ptree, pgrep, time
 - kill, trap
 - nice, stty, truss (strace)
 - vmstat, top
- Práce s úlohami
 - nohup, &, ^Z, fg, bg, jobs
- test
 - čísla
 - řetězce
 - soubory



- Výpis podrobností o procesech uživatele

```
passwd $USER &
```

```
ps -f -u $USER      (effective)
```

```
ps -f -U $USER      (real)
```

```
ps -ef | grep "^ *$USER "
```

- Formátovaný výpis některých podrobností

```
ps -f -U $USER -o user=EFFECTIVE -o ruser=REAL \
-o pid,pcpu,comm
```

- Zobrazení stromu procesů (dědičných vztahů)
k aktuálnímu procesu

```
ptree $$
```



- Zjištění PID procesů, jejichž jméno obsahuje bash

```
pgrep -u $USER bash
```

```
ps -u $USER | grep bash | awk '{print $1}'
```
- Zjištění PID a jména procesů, které jsou spuštěny přímo z aktuálního procesu

```
sleep 100 & sleep 100 & sleep 100 &
```

```
pgrep -l -P $$
```
- Zjištění doby běhu programu nebo úlohy

```
time du
```

```
time sleep 5
```

```
time { ls; sleep 5; }
```



- Nastavení reakce na signál

```
trap date INT
```

```
^C
```

```
trap 'rm /tmp/data.$$; exit' 2 3 15
```

```
trap -p
```

```
trap '' quit ; trap - int ; trap -p
```

- Vypis signálů

```
kill -l ; kill -l 2 ; kill -l quit
```

- Zaslání signálu procesu

```
touch /tmp/data.$$ ; ls /tmp/data.*
```

```
kill -3 $$ ; kill $$
```



- Změna priority procesu

```
time echo "for (i=0;i<1000;i++) e(i)" \  
| bc -l >/dev/null
```

```
time echo "for (i=0;i<1000;i++) e(i)" \  
| nice -n 19 bc -l >/dev/null
```

- Přenastavení speciálních znaků (^C → ^T)

```
stty -a | grep '\^'
```

```
stty intr ^t
```

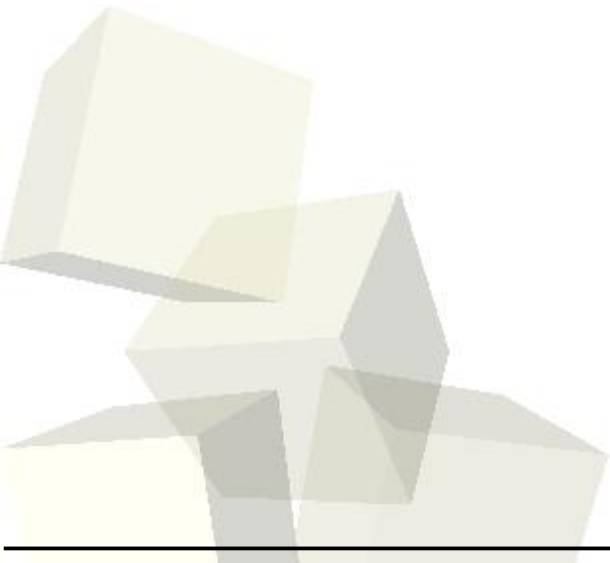
- Informace o průběhu procesu

```
truss date 2>&1 1>/dev/null | grep ^open
```

```
truss -t open date 1>/dev/null
```



- `vmstat 1 10`
- `vmstat -s`
- `prstat`
- `top`





- Spuštění úlohy na pozadí / popředí

```
cmd=/home/courses/Y36U0S/common/cv08/proces.sh
```

```
$cmd & sleep 3 ; $cmd
```

- Pozastavení úlohy na popředí

```
$cmd
```

```
^Z
```

- Seznam spuštěných úloh

```
sleep 10 & sleep 15 & sleep 20
```

```
^Z
```

```
jobs
```




- Přesun úlohy na popředí / pozadí

```
sleep 100 & $cmd &
```

```
jobs ; fg
```

```
^Z
```

```
jobs ; bg %1
```

- Přerušování úlohy na popředí

```
ls -lR /      ^C
```

- Spuštění úlohy na pozadí běžící i po skončení rodiče

```
nohup find /usr -name '*zip*' 2>/dev/null &
```

```
less nohup.out
```



Návratový kód

- Úspěch

```
grep ^root: /etc/passwd
echo $?
```
- Neúspěch

```
grep ^rooooot: /etc/passwd
echo $?
```
- Špatné použití

```
grep ^root: /etc/foo
echo $?
```
- Špatná přístupová práva

```
~/.bash_history
echo $?
```
- Program nenalezen

```
winzip
echo $?
```
- Ukončení signálem

```
ls -lR /          ^C
echo $?
```



SYNTAXE: **test** *výraz*
 [*výraz*]

x=5

```
test $x -eq 5 ; echo $?
```

```
test $x -gt 5 ; echo $?
```

```
[ $x -ge 5 ] ; echo $?
```

```
grep ^root: /etc/passwd >/dev/null
```

```
[ $? -eq 0 ] ; echo $?
```

```
[$x -ge 5] ; [ $x -ne a ] ; [ $x -lt $y ]
```



test – řetězce, log. operátory

```
[ -z "$PATH" ] ; echo $?
```

```
[ -n "$PATH" ] ; echo $?
```

```
max=`prstat 1 1 | awk 'NR==2 { print $2 }'`
```

```
[ "$max" == $USER ] ; echo $?
```

```
load=`prstat 1 1 | sed -n '$s/.* \(.*\)$/\1/;$s/\././p'`
```

```
[ "$max" != root -a "$load" -gt 50 ] ; echo $?
```

```
[ ! \ ( $PWD == $HOME -o $PWD == /tmp \ ) ] ; echo $?
```



```
[ -e /usr/bin/acroread5 ] ; echo $?
```

```
[ -f /usr/bin/acroread5 ] ; echo $?
```

```
[ -L /usr/bin/acroread5 ] ; echo $?
```

```
[ -x /usr/bin/acroread5 ] ; echo $?
```

```
[ -s /usr/bin/acroread5 ] ; echo $?
```

```
[ /usr/xpg4/bin/grep -ef /usr/xpg4/bin/egrep ] ; echo $?
```



Rozšířený test – [[]] +

```
[ ! \ ( $PWD == $HOME -o $PWD == /tmp \ ) ]
```

```
[[ ! ( $PWD == $HOME || $PWD == /tmp ) ]]
```

```
[ text_x \> text_a ]
```

```
[[ text_x > text_a ]]
```

```
[[ $SHELL == *bash ]]
```

```
line=`head -n 1 /etc/passwd`
```

```
[[ "$line" =~ ^root: ]]
```



Příprava na příští cvičení

- Vytvořte skript, který vytvoří výpis podobný příkazu `prstat 1 1` (velikost okna terminálu zjistěte pomocí `stty`, zátěž systému pomocí `uptime`).
- Zvolte si kritéria pro výběr procesů. Spočítejte procesy podle zvolených kritérií.
- Vytvořte skript, který přečte jméno souboru (včetně cesty) ze souboru `~/.filename`. Pro uvedené jméno otestujte
 - existenci cesty (adresáře) a její přístupnost,
 - existenci souboru v zadané cestě,
 - zda se jedná o obyčejný soubor,
 - zda uživatel má právo čtení a spuštění,
 - zda soubor není prázdný.