



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

PSK3-7

Název školy:	Vyšší odborná škola a Střední průmyslová škola, Božetěchova 3
Autor:	Ing. Marek Nožka
Anotace:	Správa sezení a plánované spouštění procesů
Vzdělávací oblast:	Informační a komunikační technologie
Předmět:	Počítačové sítě a komunikační technika (PSK)
Tematická oblast:	Operační systém Linux/Unix
Výsledky vzdělávání:	Žák naplánuje spouštění programu na určený čas, ovládá program screen
Klíčová slova:	Linux, Unix, shell, at, cron, screen
Druh učebního materiálu:	Online vzdělávací materiál
Typ vzdělávání:	Střední vzdělávání, 4. ročník, technické lyceum
Ověřeno:	VOŠ a SPŠE Olomouc; Třída: 4L
Zdroj:	Vlastní poznámky, Vilém Vychodil: Linux Příručka českého uživatele

Plánované spouštění programů a správce sezení

Skupiny procesů

Každý proces je vytvořen jako *potomek svého rodiče*. Procesy jsou na sobě závislé a tvoří skupiny. Například program `man` běží v shellu a zároveň používá program `less`. V následujícím příkladu vidíme ve sloupci `PPID` PID rodičovského procesu a ve sloupci `PGRP` PID vedoucího skupiny.

Jestli-že bude z jakéhokoliv důvodu rodičovský proces ukončen, budou s ním ukončeni i všichni jeho potomci.

```
$ ps -u marek --forest -o tty, pid, ppid, pgrp, comm
TT      PID  PPID  PGRP  COMMAND
tty2    27603 2941 27603  zsh
tty2    27737 27603 27737  \_ ps
tty1    27562 2940 27562  zsh
```

```
tty1 27577 27562 27577 \_ man
tty1 27588 27577 27577 \_ less
```

Pokud požadujeme, aby byl proces nezávislý na svém rodiči a aby byl sám vedoucím skupiny spustíme ho na pozadí (pomocí metaznaku &) nebo použijeme program `nohup`.

```
$ zalohujDATA &
$ exit
.....
..... na jiném terminálu
$ ps -u marek --forest -o tty,pi d,ppi d,pgrp,comm
TT      PID  PPID  PGRP COMMAND
tty2    27603 2941 27603 zsh
tty2    27737 27603 27737 \_ ps
tty1    27562 2940 27562 zsh
tty1    27577 27562 27577 \_ man
tty1    27588 27577 27577 \_ less
?       27941      1 27941 zalohujDATA
```

Nyní se může uživatel odhlásit a program `zalohujDATA` poběží dál.

Atd

V operačním systému Unix lze spuštění procesu naplánovat. K jednorázovému naplánování spuštění procesů slouží systemová služba atd. Uživatelský obslužný program se jmenuje `at`

Syntaxe programu `at` může obecně vypadat takto

```
at ČAS DATUM + PŘÍRŮSTEK
```

ČAS lze specifikovat jako

dvojtečkou oddělené hodiny a minuty ve 24-hodinovém formátu nebo `mi dni ght, noon, teati me (4pm)`.

DATUM lze specifikovat jako

den v týdnu zapsaný pomocí anglických zkratk (Sun, Mon, Tue, Wed, Thu, Fri, Sat)

DEN.MESIC

DEN.MESIC.ROK

pokud datum není uvedeno program kalkuluje s aktuálním datem

PŘÍRŮSTEK

se zapisuje jako počet jednotek: `mi nutes, hours, weeks, months, years`

`at 9:45 26.3.2056` 26. března 2056

`at tomorrow + 1 hours` zítra o hodinu později

`now + 2 weeks` za dva týdny

`13:10 + 20 minutes` ve 13:30

Například:

```
$ at now + 20 minute
```


^{-e} proměnně prostředí EDITOR

```
$ echo $EDITOR
vim
$ EDITOR=mcedit
$ echo $EDITOR
mcedit
$ crontab -e
```

Sezení

Existují speciální programy, které jsou označovány jako *terminal multiplexer*. Jejich zástupcem může být oblíbený program GNU Screen.

- <http://www.abclinuxu.cz/clanky/tipy/gnu-screen>

Umožňuje uživatelům přistupovat k několika terminálovým sezením uvnitř jediného okna terminálu či vzdáleného terminálového sezení. To je užitečné pro práci s více programy z příkazové řádky a pro oddělení programů od unixového shellu, který program spustil.

Dá se říci, že se jedná o textovou verzi správce oken nebo o umístění virtuálních terminálů do libovolného přihlášeného sezení. Je to obal, který umožňuje běh několika textových programů zároveň a dovoluje uživateli efektivně používat programy v jediném rozhraní.

Program také umožňuje nechat běžet několik interaktivních programů a přitom nebýt přihlášen. Nebo programy spustit, odhlásit se, potom se opět přihlásit a pokračovat v rozdělané práci.

Program spustíme příkazem

```
$ screen
```

Po úvodní obrazovce se spustí uživatelský shell a program `screen` není na první pohled vidět. Všechny klávesové zkratky a začínají kombinací `Ctrl + A` a za ním následuje stisk další klávesy (nebo kombinace).

Ctrl+A ?

Vypíše nápovědu.

Ctrl+A C

Vytvoří nové okno. V každém okně může běžet jiný program

Ctrl+A N Ctrl+A P

Cykluje mezi otevřenými okny

Ctrl+A Ctrl+A

Cykluje mezi dvěma naposledy otevřenými okny

Ctrl+A "

Vyvolá (interaktivní) seznam oken

Ctrl+A Shift+A

Přejmenuje aktuální okno

Ctrl+A 0 či Ctrl+A 1 či Ctrl+A 2 či ...

Přepne na okno číslo 0 či 1 či 2 atd.

Ctrl+A Shift+S

Rozdělí okno horizontálně na dva regiony.

Ctrl+A I

Rozdělí okno vertikálně na dva regiony.

Ctrl+A X

Zruší aktuální region.

Ctrl+A TAB

Přepne na následující region.

Ctrl+A I či Ctrl+A Ctrl+I či Ctrl+A ESC

zapne "copy/scrollback" mód. V tomto módu se lze pohybovat pomocí šipek, kláves jhkl, PageUp, PageDown atd. Stiskem mezerníku můžeme označovat text. Označení textu se ukončí opět mezerníkem. Text je uložen ve schránce. Tento mód lze ukončit i bez mezerníku pomocí ESC.

Ctrl+A J

Vloží text ze schránky.

Ctrl+A d

Odpojí okno. Screen stále běží, ale uživatel se od něj odpojil. Nyní se může uživatel odhlásit; programy spuštěné ve screenu poběží dál. Uživatel se může například přesunout k jinému počítači. Po opětovném přihlášení (například pomocí SSH) se může uživatel znovu připojit příkazem `screen -r`.

```
$ screen -r
```

Screen není jediným zástupcem svého druhu. Já mám například rád program tmux: <http://www.root.cz/clanky/okna-v-terminalu-pomoci-tmux/>.