



Vyšší odborná škola obalové techniky  
a střední škola, Štětí

# Digitální učební materiály

Operační systémy - Linux

Ivan Pomykacz



**esf** evropský  
sociální  
fond v ČR



EVROPSKÁ UNIE



MINISTERSTVO ŠKOLSTVÍ,  
MLÁDEŽE A TĚLOVÝCHOVY



OP Vzdělávání  
pro konkurenceschopnost

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

# Licence



Digitální učební materiály, jejímž autorem je Ivan Pomykacz, podléhají licenci [Creative Commons: Uvedte autora - Nevyužívejte dílo komerčně - Zachovejte licenci 3.0 Unported](https://creativecommons.org/licenses/by-nc-sa/3.0/).

Vytvořeno na základě tohoto díla: <http://dumy.odbornaskola.cz/pomykacz>

Práva nad rámec této licence jsou popsána zde: <http://dumy.odbornaskola.cz/pomykacz>.

# Disclaimer

Tento PDF dokument byl strojově vygenerován z HTML stránek

<http://dumy.odbornaskola.cz/pomykacz/>.

Je tedy možné, že sazba textu může obsahovat chyby. Jde převážně o vizuální a typografické chyby, které mohou narušit plynulou čitelnost textu. V některých případech může být ohrožena i funkčnost některých komponent (jako vnitřní odkazy).

Vzhledem k tomu, že vypracované materiály nebyly nikdy určeny pro výstupní formát PDF, autor se zříkává jakékoli odpovědnosti za nalezené chyby. Nesnažte se proto v této souvislosti autora kontaktovat.

# **Operační systémy**

**Linux**

# Obsah

- Správa procesů

# Správa procesů

<b>Název školy</b>	Vyšší odborná škola obalové techniky a Střední škola, Štětí, příspěvková organizace		
<b>Adresa školky</b>	Kostelní 134, 411 08 Štětí		
<b>IČ</b>	46773509		
<b>Název operačního programu</b>	OP Vzdělávání pro konkurenceschopnost		
<b>Registrační číslo</b>	CZ.1.07/1.5.00/34.1006		
<b>Označení vzdělávacího materiálu</b>	VY_32_INOVACE_21_PSS_413		
<b>Název tématické oblasti (sady)</b>	Operační systémy		
<b>Název materiálu</b>	Správa procesů		
<b>Anotace</b>	Materiál popisuje rozdíl mezi aplikacemi a službami v systému. Demonstruje aktivaci a deaktivaci démonů. Nakonec ukazuje možnosti sledování procesů a jejich vytížení CPU.		
<b>Autor</b>	Ivan Pomykacz	<b>Jazyk</b>	český
<b>Očekávaný výstup</b>	Aktivuje a deaktivuje služby/démony. Získá informace o běžících procesech. Ukončuje proces nebo službu. Rozumí přenášení uživatelských práv na proces/službu.		
<b>Klíčová slova</b>	proces, služba daemon, PID, monitorování		
<b>Druh výukového zdroje</b>	Výklad	<b>Věková skupina žáků</b>	17+
<b>Typ interakce</b>	aktivita	<b>Ročník</b>	3.
<b>Speciální vzdělávací potřeby</b>	žádné		
<b>Zhotoveno, (datum/období)</b>	15.4.2014	<b>Celková velikost</b>	

## Obsah

- Spouštění procesů a služeb (démonů)
  - Programy vs služby
  - Programy na pozadí
  - Služby (aka démoni)
    - Aktivace/Deaktivace služeb
- Monitorování procesů
  - uptime
  - ps
  - top, htop

## Spouštění procesů a služeb (démonů)

To jak se spouští běžné programy už víme. Zpravidla se nacházejí v cestách uvedených v proměnné \$PATH. Nebo je spouštíme zadáním relativní nebo absolutní cesty.

Spuštěný program má práva uživatele, který jej spustil (většinou). V některých případech tomu tak není, např. u programu `passwd` (pro změnu hesla). Výjimka u `passwd` je proto, že program provádí

změny v souboru `/etc/shadow`, a do tohoto souboru má práva zápisu pouze root.

Služby neboli démoni (daemons) jsou speciální procesy, které běží v systému na pozadí. Spouští se zpravidla při startu systému procesem `init` nebo jiným (`systemd`, dle distribuce linuxu). Služba může běžet s právy uživatele `root`, ale z bezpečnostních důvodů se spouští pod tzv. neprivilegovaným uživatelem. Totiž takovým uživatelem, který nemá v systému zbytečně vysoká oprávnění.

Spouštění a ukončování služeb lze ručně ovládat příkazem `service`.

## Programy vs služby

Běžné programy od služeb (démonů) se liší především tím, že disponují nějakou formou GUI, případně TUI a nebo jen CLI. Služba zpravidla jen zprostředkovává, nabízí (servíruje, odtud server) nějakou ... službu, kterou pak využívají konkrétní aplikace. Např. službu na servírování webových stránek zajišťuje `apache`. K prohlížení webových stránek je však potřeba webový prohlížeč (např. Mozilla Firefox).

Zatímco programy se spustí, udělají co je třeba, a následně se ukončí (uživatel je ukončí), tak služby jsou navrženy tak, aby běžely neustále. Nikdo vám ovšem nebrání spustit Firefox a nechat jej běžet několik dní nebo i déle.

## Programy na pozadí

Pokud pracujeme v GUI, pak toto nemá moc smysl řešit. Grafické uživatelské prostředí je navrženo tak, abychom mohli spouštět více aplikací, tyto pak třeba minimalizovat a zase zpětně je vyvolat podle potřeby.

Pokud jsme v konzoli, tak je situace trochu jiná. Spustíme-li např. stahování souboru příkazem `wget` (`curl`), pak se příkazová řádka odmlčí do doby, než se stahování dokončí, a teprve potom budeme moci zadávat další příkazy.

Ano, je možné se přihlásit v další konzoli (`Alt+F2`), a potom v další a další, dokud nám nedojdou. Nebo můžeme program spustit přímo na pozadí.

Ukažme si to na příkladu s programem `sleep`. Při spuštění programu s parametrem `10s` dojde k odmlčení programu na 10 vteřin a následně skončí.

```
tux@wheezy:~$ sleep 10s
```

Pokud bychom chtěli ihned po spuštění programu pokračovat psaním dalších příkazů, musíme spustit program `sleep` na pozadí. To lze provést pomocí `&`, který zapíšeme na konec příkazu:

```
tux@wheezy:~$ sleep 10s &  
[1] 29115  
tux@wheezy:~$
```

Program se spustí na pozadí a jakmile skončí, automaticky se ukončí. To zjistíme mj. tím, že stiskneme klávesu `Enter` a shell nás informuje, že aplikace na pozadí skončila.

```
tux@wheezy:~$  
[1]+  Dokončena          sleep 10s  
tux@wheezy:~$
```

## Služby (aka démoni)

Služba je vlastně též program, který se umí většinou sám démonizovat, tj. spustit na pozadí. Nicméně tyto programy nespouštíme přímo, ale pomocí nástroje `service`, který spustí připravený skript pro nastartování požadované služby.

Např. služba pro konfiguraci sítě. Je to sice trochu atypický příklad, ale jde o princip.

Následující příkaz provede zastavení (zneplatnění) konfigurace sítě, tj. připojení k síti přestane fungovat. Všimněte si, že pouze root může zastavovat nebo spouštět služby.

```
root@wheezy:~# service networking stop
```

Opětovné spuštění/aktivace se provede příkazem:

```
root@wheezy:~# service networking start
```

Pokud má služba svůj init skript, pak je možné ji tímto způsobem spouštět nebo zastavovat.

## Aktivace/Deaktivace služeb

Samotnou kapitolkou je pak aktivace, resp. deaktivace služby, aneb jak zařídit, aby se služba spouštěla při startu systému nebo naopak nespouštěla.

Za tímto účelem je tu pomocný skript `update-rc.d`.

```
root@wheezy:~# update-rc.d isc-dhcp-server disable
```

Obdobně, jako je deaktivace se provádí aktivace klíčovým slovem `enable`.

```
root@wheezy:~# update-rc.d isc-dhcp-server enable
```

## Monitorování procesů

Pokud nás zajímá, co se se systémem zrovna děje - alespoň z hlediska procesů - pak se nám budou hodit nástroje jako: `uptime`, `ps`, `top` (`htop`).

Jde o jednoúčelové utility pro sledování procesů a jejich vytěžování systémových zdrojů (zejména operační paměť a procesor).

### uptime

Příkaz vypíše jak dlouho systém běží bez restartu a průměrnou zátěž systému.



```
root@wheezy:~# uptime
12:50:53 up 286 days, 12:02, 1 user, load average: 0.08, 0.02, 0.01
root@wheezy:~#
```

## ps

Podrobnější informace nabídne nástroj `ps`. I když v základu (bez přepínačů při spuštění) toho zas až tak moc nevypíše.

```
root@wheezy:~# ps
  PID TTY          TIME CMD
 3159 pts/0    00:00:00 bash
 3724 pts/0    00:00:00 ps
root@wheezy:~#
```

Z výpisu lze vyčíst PID - identifikátor procesu (ten se nám bude hodit až budeme procesy ukončovat příkazem `kill`). TTY je označení virtuálního terminálu a TIME je využitý čas na CPU.

Všimněte si, že ve výpisu je pouze `bash` a `ps`. `bash` je příkazový shell, ze kterého jsme spustili příkaz `ps` a jelikož i `ps` v době zjišťování běžících procesů běžel, je uveden rovněž ve výpisu.

Podrobnější informace lze zobrazit přepínačem `-f`. Tady pozor, příkaz `ps` má poněkud starší historii a jeho použití se liší v používání přepínačů, které zdědil od různých systémů (zejména BSD a UNIX). Např. příkaz `ps axu` je podobný jako `ps -ely`. Co z toho vyplývá je, že v různých tutoriálech se objevují různé příklady téhož.

```
root@wheezy:~# ps -f
UID          PID  PPID  C  STIME TTY          TIME CMD
root         6356  5847  0 12:56 pts/2    00:00:00 /bin/bash
root         6722  6356  0 13:29 pts/2    00:00:00 ps -f
root@wheezy:~#
```

Výše už je alespoň vidět, kdo proces vlastník (UID), identifikátor rodičovského procesu (PPID), využití CPU (C), čas spuštění procesu (STIME) a celou cestu k programu (CMD).

Ve výpisu je ovšem stále poněkud málo procesů. Na průměrném systému je něco kolem stovky aktivních procesů. Výpis výše obsahuje pouze procesy v dané terminálové relaci. Vyzkoušejte příkazy:

```
root@wheezy:~# ps -e
root@wheezy:~# ps -ef
```

## top, htop

```
root@wheezy:~# top
top - 13:40:50 up 5:01, 4 users, load average: 0.06, 0.20, 0.21
Tasks: 194 total, 4 running, 190 sleeping, 0 stopped, 0 zombie
```

Cpu(s): 3.2%us, 1.1%sy, 0.0%ni, 94.6%id, 1.0%wa, 0.0%hi, 0.0%si, 0.0%st

Mem: 2052200k total, 1915736k used, 136464k free, 93480k buffers

Swap: 2928636k total, 2580k used, 2926056k free, 619804k cached

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
5414	tux	20	0	729m	243m	44m	S	6	12.1	1:05.39	firefox
1768	root	20	0	28572	4020	3188	R	2	0.2	0:02.30	upowerd
2546	root	35	15	6652	4132	956	R	2	0.2	0:13.91	preload
1	root	20	0	3740	2128	1352	S	0	0.1	0:00.90	init
2	root	20	0	0	0	0	S	0	0.0	0:00.00	kthreadd
3	root	20	0	0	0	0	S	0	0.0	0:01.08	ksoftirqd/0
5	root	20	0	0	0	0	S	0	0.0	0:04.86	kworker/u:0
6	root	RT	0	0	0	0	S	0	0.0	0:00.09	migration/0
7	root	RT	0	0	0	0	S	0	0.0	0:00.14	watchdog/0
8	root	RT	0	0	0	0	S	0	0.0	0:00.05	migration/1
10	root	20	0	0	0	0	S	0	0.0	0:00.85	ksoftirqd/1
12	root	RT	0	0	0	0	S	0	0.0	0:00.12	watchdog/1
13	root	0	-20	0	0	0	S	0	0.0	0:00.00	cpuset