

24 Procesy v Linuxu, 2. část

Obsah hodiny



Obsahem hodiny je popis procesů z pohledu OS Linux: komunikace mezi procesy, spouštění procesů na pozadí, plánování úloh.

Cíl hodiny



Po této hodině budete schopni:

- popsat komunikace mezi procesy pomocí signálů
- zrušit proces zasláním signálu
- charakterizovat procesy na popředí a na pozadí
- orientovat se v možnostech plánování procesů

Klíčová slova



Signály, Kill, Démon, Přepínání procesů, Plánování úloh, At, Cron

24.1 Komunikace mezi procesy, rušení procesů

Pro komunikaci mezi procesy v Linuxu je k dispozici řada mechanismů: roury, sockety, sdílená paměť, zasílání zpráv a signálů.

Komunikace mezi procesy v Linuxu je založena především na signálech.

Signál je systémem generovaná událost. Systém umožňuje zaslat každému procesu nějaký signál. Tento signál pak upraví chování procesu a tím tedy v konečném důsledku i chování samotného programu.

Signál je vlastně bit v tabulce signálů, na který plánovač procesů reaguje zavoláním obsluhy signálu v daném procesu. Proces tak může zareagovat na nějakou událost prakticky okamžitě, přičemž se přeruší právě prováděná činnost procesu. Po obslužení signálu se program vrací na místo přerušení a pokračuje tam, kde přestal.

Jakákoliv výjimečná situace může být hlášena pomocí signálu. Existuje řada signálů (je definováno 28 základních), jsou popsány v manuálových stránkách (*man signal*). Signál je označen jménem a číslem

Za základní signály lze považovat:

–	SIGTERM	15	žádost o ukončení procesu
–	SIGSEGV	11	porušení segmentace paměti
–	SIGABRT	6	ukončení z důvodu I/O operace
–	SIGHUP	1	odříznutí od terminálu (hangup)
–	SIGKILL	9	okamžité ukončení "vražda"
–	SIGQUIT	3	konec s uložením obrazu paměti
–	SIGILL	4	neplatná instrukce
–	SIGCONT	24	navrácení procesu ze stavu pozastavení
–	SIGSTOP	23	pozastavení procesu (opak SIGCONT)
–	SIGINT	2	ukončení procesu na popředí

V případě, že je signál přenechán k vyřízení jádru, dochází zpravidla k ukončení procesu, někdy jádro ukládá obsah procesu do souboru *core*. Tento soubor lze číst speciálními programy – *debuggery*. (Lze využít při ladění programu, při hledání chyby)

Libovolný signál lze vyvolat příkazem *kill*. Jako parametr se zadává číslo procesu a jméno nebo číslo zasílaného signálu.

Např., když se rozhodneme některý z procesů ukončit, zadáme jednoduše příkaz *kill* následovaný číslem procesu (PID). Tím vlastně posíláme procesu signál SIGTERM.

kill 1703 nebo *kill -9 1234* nebo *kill SIGTERM 1234*

Pokud chceme ukončit proces běžící na popředí, pošleme mu signál SIGINT, což vlastně provádíme i stiskem kláves [Ctrl+C].

Jiný příkaz, příkaz *killall*, zasílá signál všem procesům, které vykonávají daný příkaz, procesy se identifikují jménem:

killall -9 cat Jako parametr se zadává místo PID příkaz.

V KDE stačí stisknout CTRL+ALT+ESC. Kurzor se změní na lebku a tou kliknout na okno (např. Opey). Proces, kterému dané okýnko patří, bude okamžitě zabít.

24.2 Procesy běžící na pozadí

Linuxu podobně jako ve Windows běží řada služeb, jsou to procesy, které nejsou vázány na terminál a běží na pozadí. Označují se jako *démoni*.

Na pozadí mohou běžet i uživatelské procesy svázané s terminálem a to tak, že je uživatel takto spustí nebo proces běžící na terminálu (proces na popředí) přepne na pozadí a uvolní si tak terminál pro další úlohy. Pozor takto lze přepínat pouze procesy neinteraktivní. Nelze na pozadí spouštět procesy, které vyžadují vstupy nebo naopak posílají výstupy na terminál. Takové procesy sice lze přesunout na pozadí, ale při pokusu o odeslání výstupu nebo při očekávání vstupu dojde k jejich pozastavení.

Příkazy pro přesouvání procesů na pozadí a zpět:

<i>program &</i>	spuštění úlohy (jobu) na pozadí
<i>jobs</i>	výpis úloh běžících na pozadí (číslo jobu/úlohy)
<i>fg %číslo úlohy</i>	přesun úlohy na popředí (
<i>Ctrl-Z</i>	pozastavení procesu (SIGSTOP)
<i>bg %číslo úlohy</i>	přesun úlohy na pozadí (SIGCONT)

24.3 Plánování spuštění úloh, aplikací

Standardní součástí OS jsou i nástroje pro naplánování spuštění úlohy v určitém čase (hodina, minuta, den v týdnu, den v měsíci) a to buďto jednorázově příkazem *at* nebo opakovaně příkazem *crontab* (služba *cron*).

Jednorázové plánování procesů

O jednorázové plánování uživatelských procesů se stará systémový proces *atd*. Pokud tento proces na serveru běží, prohledává frontu procesů definovaných příkazem *at* a v určeném čase je spouští.

at <čas> <datum> + <přírůstek>

Po spuštění programu se správně zadaným časovým údajem se objeví výzva k zadání příkazů (*at* má vlastní prompt: *at>...*), které se zadaném čase budou spouštět. Po vložení příkazů se vstup z klávesnice musí ukončit stiskem *Ctrl-D*.

Další příkazy *at*:

<i>atq</i>	fronta naplánovaných procesů
<i>atrm</i>	vymazání procesu z fronty

Periodické plánování procesů

Periodické plánování procesů zajišťuje systémový proces daemon *cron*. Informace pro jeho činnost lze zadávat a prohlížet pomocí programu *crontab*.

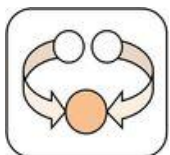
<i>crontab -e</i>	otevření a editace tabulky periodicky spouštěných
<i>crontab -l</i>	prohlížení tabulky periodicky spouštěných procesů

Formát tabulky je přesně dán. Každý řádek obsahuje pět časových údajů oddělených mezerou a příkaz pro spuštění procesu:

<minuty> hodiny> <den v měsíci> <měsíc> <den v týdnu> <příkaz>

Každý časový údaj musí být vyplněn, pokud se konkrétní hodnota nezadává, nahrazuje se hvězdičkou. Hvězdička má pak význam vždy (každou minutu, každou hodinu, každý den v měsíci, každý měsíc). Hodnoty *den v měsíci* a *den v týdnu* jsou ve vztahu nebo (buďto zadáme jedno nebo druhé).

Shrnutí kapitoly



Komunikace mezi procesy v Linuxu je založena především na signálech.

Signál je systémem generovaná událost. Systém umožňuje zaslat každému procesu nějaký signál. Tento signál pak upraví chování procesu a tím i chování samotného programu.

Signál je vlastně bit v tabulce signálů, na který plánovač procesů reaguje zavoláním obsluhy signálu v daném procesu.

Linuxu podobně jako ve Windows běží řada služeb, jsou to procesy, které nejsou vázány na terminál a běží na pozadí. Označují se jako démoni. Na pozadí mohou běžet i uživatelské procesy svázané s terminálem a to tak, že je uživatel takto spustí.

Standardní součástí OS jsou i nástroje pro naplánování spuštění úlohy v určitém čase (hodina, minuta, den v týdnu, den v měsíci) a to buďto jednorázově (at) nebo opakovaně (cron, crontab).

Kontrolní otázky a úkoly

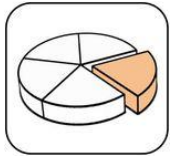


- 1) Jak funguje komunikace na základě signálů?
- 2) Co jsou to signály?
- 3) Jak funguje rušení procesů?
- 4) Co je to proces označovaný jako démon?
- 5) Jak lze plánovat spouštění úloh?

Otázky k zamyšlení



- 1) Jak funguje přepínání procesů na pozadí?



Použitá literatura a jiné zdroje:

- [1] VYCHODIL, Vilém. Operační systém Linux: příručka českého uživatele. 1. vyd. Brno: Computer Press, 2003, 260 s. ISBN 80-722-6333-1.
- [2] MACEK, Petr. Procesy v Linuxu aneb jeden admin vládne všem. *Root.cz* [online]. 11. 1. 2011 [cit. 2011-12-07]. Dostupné z: <http://www.root.cz/clanky/procesy-v-linuxu-aneb-jeden-admin-vladne-vsem/>.
- [3] PODHOLA, Martin. Signály a procesy 2. *LinuxExpres* [online]. 14. prosinec 2005 [cit. 2011-12-07]. Dostupné z: <http://www.linuxexpres.cz/praxe/signal-y-a-proces>