



SPRÁVA PROCESOV

PRÍKAZY PS A KILL



Teoretická časť

Program je súbor inštrukcií pre procesor uložený na nejakom médiu. Po jeho spustení jadro OS zavedie jeho obraz do pamäti a začne vykonávať tieto inštrukcie. Z programu sa tak stane proces. Systém priradí každému procesu jedinečné číslo *PID* (Process Identification Number). Štruktúra procesov je hierarchická, z rodičovských procesov vznikajú procesy potomkov. Príkladom môže byť povedzme príkazový interpret *bash* bežiaci v termináli. Ak v termináli spustíme napr. príkaz **ls**, *bash* vytvorí svoju kópiu a do nej nahrá obraz programu *ls*. Keď sa program *ls* vykoná, ukončí sa a zanikne aj ako proces. Ak v termináli spustíme napr. editor *vim* a necháme ho otvorený, proces *vim* sa neukončí a môžeme si o ňom zobraziť rôzne informácie; okrem iného jeho *PID* aj *PPID* (Parent Process Identification Number - *PID bashu*, z ktorého bol *vim* spustený).

Na zobrazovanie informácií o procesoch sa najčastejšie používa program *ps*. Program *ps* má mnoho prepínačov a navyše podporuje niekoľko typov syntaxe:

- syntax pochádzajúcu z pôvodného *Unixu*
- syntax pochádzajúcu z *BSD* systémov
- syntax pochádzajúcu z projektu *GNU*

Na zobrazenie všetkých procesov tak môžeme použiť štandardný príkaz **ps -e** alebo **ps ax** podľa *BSD* syntaxe. Z toho istého dôvodu nie sú totožné príkazy **ps a** a **ps -a**.

Procesom je možné zasielať signály, nie na všetky však musí proces reagovať. Existujú ale dve výnimky: *SIGSTOP* – pozastavenie procesu a *SIGKILL* – okamžité ukončenie procesu; tieto signály proces ignorovať nemôže.



Pomôcky

Virtuálny stroj s CentOS 7 vytvorený vo VMware vSphere. Úloha je určená pre jedného žiaka.



Úlohy

1. Zapnite virtuálny stroj s CentOS 7, prihláste sa do jeho grafického režimu ako superpoužívateľ *root*.
2. Otvorte si v grafickom režime pseudoterminál, napíšte v ňom príkaz *tty*. Tento terminál bude v ďalšom texte označovaný ako 0. pseudoterminál, lebo v systéme je identifikovaný ako zariadenie */dev/pts/0*
3. V 0. pseudotermináli vytvorte nových používateľov *stolar* a *sekac*. Vytvorte im aj heslá.
4. Otvorte si ďalší pseudoterminál, ktorý bude v texte označovaný ako 1. pseudoterminál, lebo v systéme je identifikovaný ako zariadenie */dev/pts/1* a zmeňte v ňom svoju identitu na používateľa *stolar* príkazom **su - solar**.
5. Stlačte *Ctrl+Alt+F3* a prihláste sa v *tty3* tiež ako *stolar*.
6. Stlačte *Ctrl+Alt+F2* a vráťte sa do grafického režimu, kde je prihlásený používateľ *root*.
7. V 0. pseudotermináli spustíte príkaz *w* a overte, ktorí používatelia a do akého terminálu sú prihlásení.
8. Pozorne si preštudujte manuálovú stránku programu *ps*.
9. V 0. pseudotermináli spustíte príkaz **ps** bez parametrov. Príkaz vypíše iba procesy používateľa, ktorý ho spustil a iba tie, ktoré sú zviazané s terminálom, v ktorom bol spustený.
10. V 0. pseudotermináli spustíte príkaz **ps a**. Príkaz vypíše všetky procesy bežiacie vo všetkých termináloch.
11. V 0. pseudotermináli spustíte príkaz **ps au**. Príkaz vypíše všetky procesy bežiacie vo všetkých termináloch, výpis je podrobnejší ako predchádzajúci, obsahuje okrem iného aj používateľské mená. Z výpisu je vidieť, že používateľ *stolar* je prihlásený v dvoch termináloch a má v nich spustený príkazový interpret *bash*.

Použitie tohto učebného materiálu je určené výhradne pre Duálne vzdelávanie realizované SPŠ elektrotechnickou Košice v spolupráci s Deutsche Telekom IT Solutions Slovakia.

Autor: J. Ploščica
Verzia 3

Predmet: ZIL, 1. ročník
Strana 2 z 4

12. V 1. pseudotermináli spustíte editor *vim*.
13. Stlačíte *Ctrl+Alt+F3* a v *tty3* zmeníte svoju identitu zo *stolar*a na používateľa *sekac* príkazom **su - sekac**. Stlačením *Ctrl+Alt+F2* a vráťte sa do grafického režimu, kde je prihlásený používateľ *root*.
14. V 0. pseudotermináli spustíte príkaz **ps au** a porovnajte ho s výpisom pri predchádzajúcom spustení tohto príkazu. Používateľ *stolar* sa vo výpise nachádza viackrát, *bash* má spustený v dvoch termináloch. V 1. termináli, t. j. v termináli */dev/pts/1* mu z *bashu* vznikol proces potomok *vim*, v termináli *tty3* z *bashu* vznikol proces potomok *su - sekac*, ale vlastníkom tohto procesu je používateľ *root*, lebo program *su* má nastavený *SUID bit*. Okrem toho vo výpise pribudol používateľ *sekac*, ktorý má spustený *bash*.
15. Predchádzajúci príkaz v 0. pseudotermináli zmeníte na **ps afu**. Vo výpise je aj strom používateľských procesov v termináloch, z ktorého je vidieť mechanizmus vzniku procesov ešte lepšie ako v predchádzajúcom prípade.
16. V 0. pseudotermináli si vypíšete všetky bežiace procesy, nielen tie, ktoré bežia v termináloch. Najskôr príkazom **ps ax** a potom príkazom **ps aux**, ktorý vypíše aj používateľské meno.
17. V 0. pseudotermináli spustíte príkaz **ps ax** ktorý používa *BSD* syntax a potom štandardný príkaz **ps -e**. Porovnajte ich výstup.
18. V 0. pseudotermináli spustíte postupne nasledujúce príkazy, výstup z nich je „rúrou“ poslaný programu *more*, aby bolo možné sledovať a porovnať, ktoré stĺpce sa zobrazujú:
ps -e | more
ps -ef | more
ps -el | more
19. V 0. pseudotermináli spustíte príkaz **ps aux | grep stolar** a potom príkaz **ps -ef | grep sekac** a prezrite si zoznam procesov súvisiacich s týmito používateľmi.
20. V 0. pseudotermináli si zobrazte všetky procesy používateľa *stolar* príkazom **ps -u stolar**. Analogicky si zobrazte procesy používateľov *sekac* a *root*.
21. V 0. pseudotermináli si zobrazte podrobnejší výpis všetkých procesov používateľa *stolar* pomocou príkazu **ps -afu stolar**. Vo výpise sa zobrazujú aj stĺpce s *PID* a *PPID*, všimnite si v nich, že v termináli *tty3* z *bashu*, ktorý tam mal

- pôvodne spustený používateľ *stolar* vznikol proces *su* patriaci *rootovi* a z neho ďalší *bash*, ktorý patrí používateľovi *sekac*.
22. Pomocou prepínača **-o** je možné v programe *ps* zvoliť, ktoré stĺpce a v akom poradí sa vypíšu. V 0. pseudotermináli si zobrazte procesy používateľa *stolar* s voliteľnými stĺpcami príkazom **ps -u stolar -o user,uid,pid,ppid,s,cmd**. Pre zobrazenie procesov používateľa *sekac* použite príkaz **ps -u sekac -o user,tty,pid,ppid,time,cmd**.
 23. V 0. pseudotermináli spustíte príkaz **ps -u stolar -o user,pid,s,cmd**. Všimnite si, že všetky procesy tohto používateľa sú v stave *S* (sleep) pretože čakajú na používateľský vstup. Zašlite procesu *vim* signál na pozastavenie príkazom **kill -SIGSTOP pid_procesu_vim**. Znovu použite príkaz **ps -u stolar -o user,pid,s,cmd**, všimnite si, že proces *vim* je v stave *T* (traced).
 24. V 1. pseudotermináli si overte, že editor *vim* bol naozaj pozastavený.
 25. V 0. pseudotermináli spustíte príkaz **ps -u stolar -o user,pid,tty,cmd**. Z výstupu identifikujte *PID* programu *bash*, ktorý má používateľ *stolar* spustený v termináli */dev/pts/1*. Potom reláciu tohto používateľa násilne ukončíte príkazom **kill -SIGKILL pid_zisteneho_procesu**. V 1. pseudotermináli si overte, že relácia používateľa *stolar* v ňom bola ukončená a je v ňom prihlásený pôvodný používateľ *root*.
 26. Zavrite 1. pseudoterminál. V 0. pseudotermináli sa pokúste odstrániť zo systému používateľský účet *sekac* príkazom **userdel -r sekac**. Nepodariť sa vám to, lebo tento používateľ je prihlásený v *tty3*. Zobrazí sa aj systémové hlásenie obsahujúce *PID* procesu používateľa *sekac*. Ukončíte tento proces pomocou programu *kill* a potom odstráňte používateľský účet *sekac*. Analogicky ukončíte procesy používateľa *stolar* a odstráňte jeho používateľský účet.