

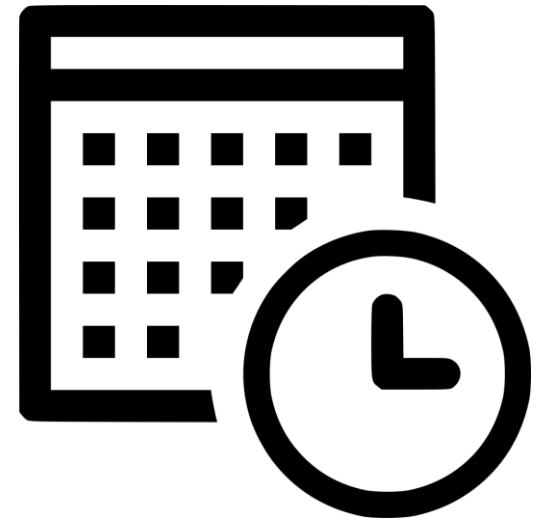
Úvod do operačných systémov

Cvičenie 6



Náplň cvičenia

- Debugger a pomocné výpisy
- Procesy
- Trap
- Cron/Anacron
- Príklad na precvičenie



Debugovanie

- Proces hľadania a riešenia chýb alebo problémov, ktoré bránia správne fungovaniu skriptu
- Tradičné techniky
 - Zvýraznenie syntaxe - zachytenie chýb pri písaní skriptu
 - Pomocné výstupy - príkaz **echo**
- Bash xtrace
 - Prepínač **-x** pri spustení skriptu príkazom **bash**, v shebangu, alebo príkazom **set -x** (deaktivácia **set +x**)
 - Prepínač **-v** zobrazuje riadky skriptu, ako sú čítané
 - Prepínač **-u** zobrazí premenné, ktoré nemajú priradenú hodnotu
- Použitie príkazu **trap** (signály EXIT, DEBUG, prípadne ERR)

Procesy - KDE

Processes — System Monitor

Tools Overview Applications History Processes Add New Page...

Search...

End Process Show: Own Processes Configure columns...

Name	CPU	Memory	Read	Write
plasmashell		297.2 MiB		
plasma-systemmonitor	2.8%	171.9 MiB		
kded5		54.4 MiB		
DiscoverNotifier		35.7 MiB		
kalendarac		34.7 MiB		
kdeconnectd		33.7 MiB		
kwalletd5		31.2 MiB		
ksmserver		30.5 MiB		
kaccess		26.9 MiB		
pulseaudio		22.3 MiB		
Xwayland		20.7 MiB		
ksystemstats		10.2 MiB		
startplasma-wayland		9.3 MiB		
kactivitymanagerd		6.5 MiB		
gmenudbusmenuproxy		6.2 MiB		
xembedsniproxy		5.9 MiB		
kioslave5		5.5 MiB		
gnome-keyring-daemon		5.2 MiB		
systemd		4.7 MiB		
kwin_wayland_wrapper		4.2 MiB		
kioslave5		3.8 MiB		
agent		3.0 MiB		
gsettings-helper		1.6 MiB		
obexd		1.6 MiB		
gcr-ssh-agent		1.3 MiB		
dbus-daemon		1.2 MiB		
VBoxClient		1.0 MiB		

Overview — System Monitor

Tools Overview Applications History Processes Add New Page...

Edit Page

Memory

Used Physical Memory
1.3 GiB
Total Physical Memory
7.8 GiB

Disk

Used Space
9.3 GiB
Total Space
30.4 GiB

CPU

2.5%

Network & System

Networks

Wired connection 1
IPv4 10.0.2.15 fe80::a00:27ff:fe18:ffc9

Network Rates

Wired connection 1
Download 0 B/s Upload 0 B/s

System

Hostname udos
OS Debian GNU/Linux 12 (bookworm)
KDE Plasma Version 5.27.5
KDE Frameworks Version 5.103.0
Qt Version 5.15.8

Applications

Name	CPU	Memory	Read	Write
System Monitor	7.0%	170.2 MiB		3.3 KiB/s
Discover		35.7 MiB		
Calendar Reminders		34.7 MiB		
KDE Connect		33.7 MiB		
Accessibility		26.9 MiB		
vboxclient		4.1 MiB		
Geoclue Demo agent		3.0 MiB		

Procesy

- Keď spustíme príkaz, aplikáciu či program, vznikne proces so svojim PID (jedinečný identifikátor procesu, `echo $$`)
- Každý proces, ktorý beží má svojho predka
 - proces s ID 1 (*init*) a prípadne s ID 2 (*kthreadd*) majú predka s ID 0 (*swapper* alebo *sched*)
- Statický prehľad o procesoch - príkaz `ps`
 - Prepínačom `--forest` sa zobrazí hierarchia medzi procesmi grafickou formou (`_`)
 - Prepínačom `-ef` alebo argumentom `aux` je možné vidieť všetky procesy systému
 - UID – používateľ, ktorý spustil proces
 - PID – identifikačné číslo procesora
 - PPID – identifikačné číslo predka
 - C – využitie procesora, celočíselná hodnota v %
 - STIME – začiatok behu procesu
 - TTY – terminál, na ktorom beží proces
 - TIME – kumulatívny čas behu na CPU
 - CMD – príkaz so všetkými argumentmi ako reťazec
 - ...
 - Prepínačom `-u N` sa zobrazia procesy používateľa N
 - Prepínačom `-o` je možné vypísať špecifikované stĺpce

Procesy

- Dynamický prehľad o procesoch - príkaz **top**
 - Ukončenie pomocou q, pomoc pomocou h, zmena prekresľovania na vlastnú hodnotu pomocou d, stlačením K je možné poslať procesu s daným PID daný signál, stlačením c sa zobrazí absolútna cesta k procesu, stlačením M sa usporiada výpis podľa vyťaženia pamäte, stlačením N sa usporiada výpis podľa PID, pomocou r je možné zmeniť „nice“ hodnotu procesu s daným PID, pomocou z je možné zmeniť výstupnú farbu na zobrazenie bežiacich procesoch
 - Výpis obsahuje tieto náležitosti:
 - PID – identifikačné číslo procesu
 - USER – používateľ, ktorý spustil proces
 - PR – priorita procesu pre plánovač
 - NI – tzv. „nice“ hodnota procesu, pričom menšia hodnota predstavuje väčšiu prioritu ($PR = 20 + NI$)
 - VIRT – veľkosť virtuálnej pamäte, ktorú používa proces (všetok kód, dáta a zdieľané knižnice + stránky)
 - RES – veľkosť fyzickej pamäte, ktorú využíva proces
 - SHR – veľkosť zdieľanej pamäte, ktorú používa proces (pamäť, ktorá môže byť zdieľaná medzi procesmi)
 - S – stav procesu (D - neprerušiteľný spánok, R - bežiaci, S - spiaci, T - zastavený, Z – zombie, čo znamená, že proces už skončil, ale je stále evidovaný)
 - %CPU – percentuálne využitie CPU
 - %MEM – percentuálne použitie pamäte
 - TIME+ – celkový čas používania CPU
 - COMMAND – názov príkazu, ktorý odštartoval proces

Posielanie signálov procesom

- Poslanie špecifického signálu procesu s daným PID - príkaz `kill`
 - Výpis zoznamu všetkých signálov (`-l`), pričom dôležité sú signály `SIGKILL(9)` na okamžité ukončenie procesu, `SIGTERM(15)` - jemnejšie ukončenie, kde sa čaká na skončenie procesu, `SIGINT(2)` je prerušenie (`CTRL + C`)
 - Odoslanie signálu `N` príslušnému procesu (`-N`, `N` môže byť číslo, ale aj označenie)
 - Ak miesto PID procesu sa zadá `-1`, potom sa daný signál pošle všetkým procesom, ktoré reagujú na signály od používateľa
- Poslanie špecifického signálu procesom s daným názvom, prípadne podľa času behu - príkaz `killall`
 - Len presný názov (`-e`), ignorovanie case sensitivity (`-I`), vyžiadanie povolenia na poslanie signálu (`-i`), len pre špecifického používateľa (`-u`), výpis spätnej väzby (`-v`), mladší alebo starší ako daný časový údaj (`-o`, `-y`)
- Poslanie špecifického signálu procesom s daným názvom a inými atribútmi - príkaz `pkill`
 - Len novšie alebo staršie nájdené procesy (`-n`, `-o`), len tie, ktoré presne zodpovedajú predpisu (`-x`), pošle sa špecifický signál procesom (`-signal`), len nájdené procesy používateľa (`-u`)

Signál	Štandard	Akcia	Popis
SIGABRT	P1990	ukončenie + informácia core(5)	Ukončovací signál z funkcie abort(3)
SIGALRM	P1990	ukončenie	Signál časovača z funkcie alarm(2)
SIGBUS	P2001	ukončenie + informácia core(5)	Chyba zbernice (zlý prístup k pamäti)
SIGCHLD, SIGCLD	P1990	ignorovanie	Detický proces bol pozastavený alebo skončený
SIGCONT	P1990	pokračovanie	Pokračovanie vo vykonávaní, ak bol proces pozastavený
SIGEMT	-	ukončenie	Emulovanie signálu trap
SIGFPE	P1990	ukončenie + informácia core(5)	Výnimka „Floating-point“
SIGHUP	P1990	ukončenie	Zavesenie („Hangup“) zistené na riadiacom termináli alebo ukončenie riadiaceho procesu
SIGILL	P1990	ukončenie + informácia core(5)	Neplatná inštrukcia
SIGINT	P1990	ukončenie	Prerušenie zadané z klávesnice
SIGIO	-	ukončenie	I/O operácie sú k dispozícii (4.2BSD)
SIGIOT	-	ukončenie + informácia core(5)	IOT trap signál. Synonymum pre SIGABRT
SIGKILL	P1990	ukončenie	Signal „kill“ na násilné ukončenie procesu
SIGLOST	-	ukončenie	Strata zámku na súbor (nepoužívané)
SIGPIPE	P1990	ukončenie	Pokazená rúra: zápis do rúry bez žiadnych čitateľov; viac info vid' pipe(7)
SIGPOLL	P2001	ukončenie	„Pollable“ udalosť (System V). Synonymum pre SIGIO
SIGPROF	P2001	ukončenie	Časovač profilovania uplynul
SIGPWR, SIGINFO	-	ukončenie	Zlyhanie napájania (System V)
SIGQUIT	P1990	ukončenie + informácia core(5)	Ukončenie zadané z klávesnice
SIGSEGV	P1990	ukončenie + informácia core(5)	Neplatný odkaz na pamäť
SIGSTKFLT	-	ukončenie	Zlyhanie zásobníka na koprocessore (nepoužívané)
SIGSTOP	P1990	zastavenie	Zastavenie procesu
SIGTSTP	P1990	zastavenie	Zastavenie procesu zadané na termináli
SIGSYS, SIGUNUSED	P2001	ukončenie + informácia core(5)	Zlé systémové volanie (System V SVr4); viac info vid' seccomp(2)
SIGTERM	P1990	ukončenie	Ukončenie procesu
SIGTRAP	P2001	ukončenie + informácia core(5)	„Trace/breakpoint trap“
SIGTTIN	P1990	zastavenie	Vstup z terminálu pre proces na pozadí
SIGTTOU	P1990	zastavenie	Výstup na terminál pre proces na pozadí
SIGURG	P2001	ignorovanie	Naliehavý stav na sockete (4.2BSD)
SIGUSR1	P1990	ukončenie	Používateľom definovaný signál 1
SIGUSR2	P1990	ukončenie	Používateľom definovaný signál 2
SIGVTALRM	P2001	ukončenie	Alarm virtuálnych hodín (4.2BSD)
SIGXCPU	P2001	ukončenie + informácia core(5)	Prekročený čas používania CPU (4.2BSD); viac info vid' setrlimit(2)
SIGXFSZ	P2001	ukončenie + informácia core(5)	Prekročený limit veľkosti súboru (4.2BSD); viac info vid' setrlimit(2)
SIGWINCH	-	ignorovanie	Signál zmeny veľkosti okna (4.3BSD, Sun)

Procesy a úlohy

- Pre spustenie procesu na pozadí je potrebné pridať na konci znak &
 - `sleep 5 &`
- Pre pozastavenie procesu - CTRL + Z
- Pre výpis aktuálnych úloh - príkaz `jobs`
 - Úloha môže byť jeden proces, ale môže mať aj viac podprocesov či detských procesov (pipeline)
 - Výpis aj PID (`-l`), len úlohy, čo zmenili stav (`-n`), len PID (`-p`), len bežiace alebo zastavené úlohy (`-r`, `-s`)
- Posun procesu z pozadia do popredia - `fg %1`
- Posun procesu z popredia do pozadia - `bg %s1`

Trap

- Zabudovaný príkaz, ktorý sa používa na vykonanie príkazu, keď shell prijme definovaný signál
- Úloha uvoľnenia zdrojov pri nečakanom ukončení skriptu sa dá ľahko vykonať pomocou tohto príkazu
- Signály je možné zobrazíť príkazom `kill -l`, `trap -l` alebo `man 7 signal`
- Pre znefunkčnenie signálu (napr. SIGTERM) - `trap "" SIGTERM`
- Pre obnovenie signálu je možné použiť `trap - SIGTERM`
- Je možné zachytiť viacero signálov na ten istý príkaz

Trap

```
#!/bin/bash
let pocet_koncov=0
let sucet=0
ukonci_program() {
    if [ $pocet_koncov -le 0 ]; then
        echo -e "\nKoniec este nenastal!"
        let pocet_koncov++
    else
        echo -e "\nTak teda koncim, sucet bol $sucet"
        exit 0
    fi
}
trap "ukonci_program" SIGINT
let cislo=0
while true; do
    let cislo=1+RANDOM%100
    echo "Generujem cislo z rozsahu <1, 100> a to cislo $cislo"
    let sucet+=cislo
    sleep 1
done
exit 0
```

Ukončenie OS/PC

- Príkaz `shutdown`
 - Naplánuje čas vypnutia OS (`now`, `hh:mm`), pričom je možné dodať aj informáciu, ktorá sa zobrazí pre všetkých aktuálne prihlásených používateľov
 - Môže sa použiť na zastavenie (`-H`), vypnutie (`-P`) alebo reštart (`-r`) OS, ktoré je možné aj prerušiť (`-c`)
- Príkazy `halt`, `reboot` a `poweroff`
 - Je ich možné použiť na zastavenie, vypnutie alebo reštartovanie OS
 - Všetky tri príkazy majú rovnaké prepínače
 - Je možné zmeniť význam príkazu pomocou prepínačov `--halt`, `--poweroff(-p)` a `-reboot`
 - Pre potreby vyvolania bezodkladného príkazu je možné použiť prepínač `-f` a pri jeho zdvojení sa obchádza aj manažér systému

Cron

- Daemon (`man daemon`), ktorý beží v Linuxových strojoch (ak nie je spustený, tak `crond` ako správca)
- Umožňuje používateľom spúšťať skripty alebo iba príkazy či skupinu príkazov automaticky a pravidelne v stanovenom čase
- Nie každý môže používať cron - správa cez `/etc/cron.allow` a `/etc/cron.deny` (ak neexistujú, tak ich treba vytvoriť podľa potreby)
- Crontab (cron table) - textový súbor, ktorý špecifikuje časový harmonogram úloh
 - `crontab`
 - Pre editáciu `-e`, pre prehľad `-l`, pre odstránenie `-r`, ...
 - Pre nastavenie konkrétneho editora súborov
 - `EDITOR=nano crontab -e`
 - Používateľské súbory - `/var/spool/cron/crontabs/pouzivatel` (pre Debian, `/var/spool/cron/pouzivatel` pre RedHat)
 - root súbory - `/etc/crontab` a `/etc/cron.d/` (prípadne `/etc/cron.{hourly,daily,weekly,monthly}/`)

Cron - stanovenie času

- Každá úloha by mala mať stanovený čas (periódu), kedy sa bude vykonávať (pomôcka [tu](#)), ktorý predstavuje prvých 5 hodnôt, nasleduje používateľ, ktorý danú úlohu vykonáva (systémové úlohy) a nakoniec príkaz na vykonanie
- Pokiaľ je použitý znak *, tak to treba chápať ako akúkoľvek hodnotu (vždy)
- Pokiaľ je potrebné uviesť viac hodnôt, oddeľovač je znak , (napr. 1,3,5)
- Pre rozsah je možné použiť znak - (napr. 5-8)
- Pre opakovanie s určitou periódou je možné použiť znak / (napr. */4 pri hodinách je ekvivalentné s 0,4,8,12,16,20)

```
# Example of job definition:
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .----- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# | | | | |
# * * * * * user-name command to be executed
```

Cron a anacron

Cron

- Je to daemon, ideálny pre servery
- Umožňuje vykonávať úlohy v definovaných časoch až na úroveň minút
- Pokiaľ je stroj vypnutý, tak sa naplánovaná úloha nevykoná
- Môžu ho používať používatelia systému a aj root

Anacron

- Nie je daemon, ideálny pre PC
- Naplánovanie úloh je možné až na úroveň dní
- Formát: `period delay(min) job-identifier command`
 - `@daily 10 example.daily /bin/bash /home/sometnig/bin/backup.sh`
- Pokiaľ je stroj vypnutý, tak sa naplánovaná úloha vykoná pri najbližšom štarte stroja
- Používa ho iba root, pokiaľ nie je povolený prístup pre ostatných používateľov cez konfiguračné súbory

Príklady na precvičenie

- Vytvorte skript, ktorý bude načítavať slová z klávesnice, až kým sa nezadá slovo „koniec“. Skript bude pracovať v štyroch režimoch (používateľ sa prepne medzi režimami pomocou číslíc 1-4).
 - V prvom režime sa budú z načítaných slov získavať prvé tri znaky a tieto znaky sa budú určitým spôsobom evidovať (case senzitivitu nie je potrebné riešiť), pričom pri skončení skriptu sa vypíše početnosť jednotlivých trojíc znakov (napr. zadá sa slovo „Ahoj“ a uloží sa „Aho“ a na konci sa vypíše, koľkokrát sa vyskytoval reťazec „Aho“)
 - V druhom režime sa na základe každého slova vytvorí súbor, ktorého názov sa bude zhodovať s poslednými tromi znakmi (nie je potrebné riešiť, či taký súbor existuje). Pri testovaní skriptu odporúčam vytvorenie testovacieho adresára
 - V treťom režime sa slová budú ukladať do skrytého súboru, pričom pri skončení skriptu sa vypíše top 3 zoznam najdlhších slov a skrytý súbor sa vymaže
 - Vo štvrtom režime sa na základe každého slova vymaže súbor, ktorého názov sa bude zhodovať s poslednými tromi znakmi (je potrebné riešiť, či taký súbor existuje)

Príklady na precvičenie

- Vytvorte skript, v ktorom hodíte 10-strannou kockou x -krát ($49 < x < 101$ je hodnota zadaná ako argument skriptu, pričom ak nie je zadaná alebo nie je platná, tak skript sa hneď skončí)
- Následne sa vypíše top 3 zoznam hodnôt, ktoré padli najčastejšie (napr. pre 50 hodov s počtami padnutí [1,5,2,3,9,8,7,5,6,4] hodnôt 1-10 je top 3 list 5;6;7)
- Pozor, pokiaľ bude viac hodnôt s rovnakým počtom padnutí, tak treba vypísať všetky hodnoty (napr. pre 50 hodov s počtami padnutí [2,5,2,2,8,8,8,5,6,4] hodnôt 1-10 je top 3 list 5,6,7;9;2,8)
- V prípade, že by nemohol vzniknúť top 3 zoznam, tak sa vypíše na chýbajúcom mieste nikto (napr. pre 50 hodov s počtami padnutí [6,6,6,6,6,4,4,4,4,4] hodnôt 1-10 je top 3 list 1,2,3,4,5;6,7,8,9,10;nikto)

Príklady na precvičenie

- Skúste si v rámci niektorého vami vytvoreného skriptu spustiť `bash xtrace` a skúste otestovať všetky prepínače uvedené v materiáloch (x, v, u)
- Vytvorte skript, v ktorom budete donekonečna hádzať kockou so sekundovou pauzou. Po stlačení CTRL + C sa vám vypíše aktuálny počet padnutí čísla 6 a po opätovnom stlačení CTRL + C sa skript končí, pričom sa predtým vypíše celkový počet hodov kockou
- Spustíte príkaz `sleep 50` na pozadí. Zistíte jeho PID a následne sa ho pokúste zastaviť príslušným signálom pomocou vami zvoleného príkazu
- Zistíte, aké úlohy máte definované v cron-e vo vašom OS
- Vytvorte si používateľskú cron tabuľku a v nej stanovte vykonávanie príkazu `ps -f -u $USER >> $HOME/state`, každých 5 minút