Operativni sistemi

Milena Vujošević-Janičić

www.matf.bg.ac.rs/~milena

Arhitektura i operativni sistemi
Beograd, 17. decembar, 2013.

Procesi

Procesi

- Proces je program u izvršenju.
- Linux omogućava istovremeno izvršavanje više procesa.
- Ovakvi procesi se izvršavaju u odvojenim virtualnim adresnim prostorima.
- Komunikacija između procesa je moguća samo pomoću bezbednih mehanizama koje omogućava i kontroliše kernel.

Procesi

- Procesi u toku izvršavanja koriste razne resurse sistema (procesor, memorija, fajlovi, fizički uređaji).
- Linux mora da upravlja ovim resursima u cilju pravednog izvršavanja svih procesa.
- Najvažniji resurs je procesor i prilikom upravljanja procesima bitno je maksimizovati njegovu iskorišćenost.
- Procesi se identifikuju identifikacionim brojevima (PID).

Procesi

- Informacije o procesima se čuvaju u nizu `task_struct` struktura (tzv. task vektor, tabela procesa) koja sadrži sledeće informacije:
  - Stanje
    - Izvršno — proces se izvršava ili je spreman za izvršavanje
    - Čekanje — proces čeka neki događaj kao što je signal sistema ili hardverski resurs.
    - Zaustavljen — proces može da se vrati u izvršno stanje
    - Zombi — proces koji je završio izvršavanje, ali su njegovi podaci i dalje u task vektoru.

Procesi

- Informacije o rasporedu (politika, prioritet, preostalo vreme...)
- Identifikatori (uid, gid, effektivni uid i gid...)
- Informacije o mehanizmima međuprocesne komunikacije
- Pokazivači na roditeljski proces, decu i procese sa istim roditeljem.
- Informacije o vremenu izvršavanja i tajmerima.
- Informacije o otvorenim fajlovima.
- Informacije o virtualnoj memoriji.
- Sadržaj registara procesora (processor context).

Init

- Po pokretanju sistema kreira se jedan proces — init čiji je PID 1 i koji vrši inicijalizaciju sistema npr. vezivanje root fajlsistema i pokretanje osnovnih procesa iz /etc/inittab (Ubuntu — /etc/init).
- init je krajnji roditeljski proces svih procesa na sistemu.

Procesi

- Hijerarhija: roditelj, dete, siroče, zombi
- Roditelj — proces koji kreira novi proces, novi proces je njegovo dete
- Ukoliko roditelj završi sa radom pre deteta, ili nasilno bude prekinut, dete postaje siroče i usvaja ga proces Init
- Kada dete završi sa radom, očekuje se da njegov roditelj pročita exit status deteta, ukoliko roditelj to ne uradi, dete postaje zombi
- Zombi ne drži memoriju, već samo PID i unos u tabeli procesa
- Postojanje zombija u sistemu ukazuje na nekakvu grešku u roditelju (ukoliko postoje duži vremenski period)
- Zombi se ne može ubiti, može se ubiti samo roditelj, čime zombi postaje siroče i njegov status razrešava init proces

Kreiranje procesa

- Novi procesi se u Linux-u kreira na sledeći način:
  - Postojeći proces pravi svoju kopiju koja dobija svoj PID, adresni prostor... Ovo se radi pomoću fork sistemskog poziva.
  - Kopija procesa nastavlja izvršavanje, može da pokrene pomoću exec sistemskog poziva novi program, koji dobija njen adresni prostor.
  - Novi program nasleđuje okruženje, standardni ulaz, izlaz i izlaz za greške, kao i prioritet izvršavanja koji je imao polazni proces.
  - Na ovaj način se pokreću svi procesi osim init-a.

Primer pokretanja programa

- Prvi proces je init. Nakon inicijalizacije sistema, init kreira svoju kopiju pomoću fork, a zatim kopija pomoću exec pokreće agetty program.
- agetty otvara tty port, traži login ime i kada korisnik unese svoje ime pokreće login program pomoću exec (da bi se tražila lozinka).
- nakon uspešnog logovanja, login pokreće bash pomoću exec.
- bash pravi kopiju pomoću fork, a ona pomoću exec pokreće program (npr. ls).

Vrste procesa

- Demoni
- Interaktivni procesi

Demonski

- Demonski procesi su serverski procesi koji se izvršavaju u pozadini.
- Kod Windows sistema, ovakvi procesi se nazivaju servisi
- Oni se obično pokreću u vreme podizanja sistema i čekaju dok nekom procesu ne zatreba njihova usluga.
- init, syslogd, sendmail, crond, agetty, inetd, named, httpd...

Interaktivni procesi

- Interaktivni procesi se pokreću i kontrolišu iz terminala.
- Interaktivni proces može biti izvršavan vezan za terminal kada samo on može da prima ulaz sa terminala ili u pozadini kada je terminal slobodan za pokretanje drugih programa.
- Procesi se mogu poslati u pozadinu ili se iz nje vratiti na terminal.

Komanda ps

- ps — lista trenutne procese `ps [-options]`
- ps --help
- ps -e ili ps -A — prikazuje PID, TTY, TIME i CMD svih procesa na sistemu
- ps -f — full: prikazuje dodatne informacije o svim procesima koji su pokrenuti iz tekućeg shell konteksta ili Terminal prozora. U dodatne informacije spadaju: ID korisnika koji je pokrenuo komandu koja je inicirala proces (UID), identifikator proces roditelja (PPID), prioritet procesa (C) i vreme kada je proces počeo sa izvršenjem (STIME).
- ps -u UID — prikazuje PID, TTY, TIME i CMD svih procesa koje je inicirao korisnik čiji je UID naveden kao parametar.
- opcija --forest

Komanda top

- top — dinamički pogled na sistem, slično kao system monitor, samo tekstualni interfejs

Vreme

- Komanda time — Određivanje vremena potrebnog za izvršenje procesa
- Komanda time na ekranu prikazuje tri vremena: realno (real), sistemsko (system) i korisničko (user).
- Realno vreme obuhvata interval od zadavanja komande do potpunog izvršenja i povratka komandnog prompta, uključujući i vreme čekanja na ulaz, izlaz i ostale događaje.
- Korisničko vreme je količina procesorskog vremena utrošena na samo izvršenje procesa.
- Sistemsko vreme je vreme koje je kernel utrošio na opsluživanje procesa.

Signali

- Signali predstavljaju način međuprocesne komunikacije.
- Služe da obaveste procese o nekim događajima u sistemu.
- Procesi mogu na neki način da odgovore na signale ili da ih ignorišu.
- Odgovor na signal može da definiše proces, ukoliko ga ne definiše izvršava se podrazumevana akcija
- Jedini signali koji se ne mogu ignorisati i predefinisati su SIGKILL i SIGSTOP.
- Zavisno od implementacije, u svakom UNIX sistemu je definisano 30 do 40 signala, od kojih je svaki predstavljen imenom i brojem — pogledati signal.h (locirati ga sa locate, pogledati default action za signale). Tu su i real-time signali.

Slanje signala

- Komanda stty štampa ili menja karakteristike terminala
- stty -a — štampa karakteristike terminala
- Ctrl-C — SIGINT
- Ctrl-Z — SIGSTOP
- Ctrl-\ — SIGQUIT
- kill šalje odgovarajući signal procesu koji se zadaje koristeći PID, opcija -l štampa imena signala i odgovarajuće brojeve, info kill
- killall — šalje signal procesu a kao argument prima ime procesa, a ne PID kao kill, po default-u šalje SIGTERM signal
- info killall

Slanje signala

Koji signal treba poslati procesu?

- Ne treba započeti sa KILL, već prvo probati sa TERM
- Ukoliko uništenje ne uspe, nastaviti signalom INT
- Ukoliko uništenje ne uspe, nastaviti signalom HUP
- Ukoliko uništenje ne uspe, nastaviti signalom QUIT, ovaj signal daje sliku procesa
- Ukoliko je proces i dalje živ, uništava se signalom KILL (kill -s 9 PID)

Slanje signala

- Signal HUP — hang-up se koristi prilikom odjavljivanja sa sitema — ovaj signal se salje svim procesima koji su pokrenuti u pozadini i njime se zaustavlja rad tih procesa kada se korisnik odjavi sa sistema
- Ukoliko zelimo da neki proces nastavi izvrsavanje i nakon odjavljivanja korisnika sa sistema, koristi se nohup komanda:
  nohup komanda argumenti
  Na taj način pokrenuta komanda postaje imuna na neke signale i nastavlja sa izvršavanjem i nakon odjavljivanja sa sistema

Komande za rad sa procesima

- jobs (-lnprs) — lista procese koji se obavljaju u pozadini (-l lista id)
- fg — prebacuje pozadinski posao u terminal — argument je broj pozadinskog posla (moze se procitati sa jobs), ukoliko je jedinstveno ime posla onda moze sa imenom, ukoliko je samo jedan pozadinski posao može i bez argumenta
  bg — prebacuje posao u pozadinu
  & — prilikom pokretanja pokreće u pozadini

Vežbanje

- Izlistati aktivne procese komandom ps
- Izlistati aktivne procese komandom ps sa opcijom -A
- Pokrenuti proces evince sa argumentom aos13.pdf u pozadini
- Izlistati sve pozadinske procese (komanda jobs)
- Prebaciti pozadinski proces evince u terminal
- Poslati procesu SIGSTOP signal
- Izlistati sve pozadinske procese i obratiti paznju na status procesa
- Prebaciti proces ponovo u pozadinu
- Izlistati pozadinske procese i obratiti paznju na status procesa
- Pratiti stanje sistema koristeći komandu top

Pitanja

- Šta je proces?
- Koja su četiri osnovna stanja procesa?
- Koji se proces prvi pokreće prilikom pokretanja operativnog sistema?
- Koji sistemski poziv služi za kreiranje novog procesa?
- Šta su demonski procesi?
- Šta su signali?
- Nabroj imena i brojeve tri signala.
- Koji signali se ne mogu ignorisati?
- Kako se šalje SIGSTOP?
- Kako se šalje SIGINT?