Procesy

23. 2. 2012

proces

- proces = bežiaci program
- úvodný proces = init
- prvý proces
- štandardne beží stále
- v GRUBe parameter **kernel**
  - init=/bin/bash
- možno zmeniť cestu k prvému skriptu
  - ako hacknúť systém

- v PS: stavy procesu v druhom stĺpci
- proces, ktorý osirie, je adoptovaný init-om

- v OS bežia procesy
- v procesoch bežia vlákna
- vznik procesu
  - vyhradí sa pamäť
    - pre samotný program
    - heap (halda)
    - stack (zásobník)
    - kapacita až po veľkosť operačnej pamäte + swap
  - dostané PID
  - tabuľka resourcov
    - sockety, rúry, súbory...

- vlákno zdieľa časti pamäte procesu, v ktorom beží
- vlákno je identifikované TID
  - thread ID
- v ps
  - l indikátor vlákna
  - vlákna zdieľajú TID s procesmi
- vlákno bez procesu neexistuje

- preemptivita: procesor má právo odobrať procesu procesorový čas
- nepreemptívne
  - vs kooperativita: proces sa sám vzdá
  - v jadre existujú hooky, keď úloha môže odovzdať kontrolu jadru: čakanie na vstup, ...
- dobrovoľné nepreemptívne
  - v jadre existujú dodatočné hooky, keď úloha môže odovzdať kontrolu jadru
  - jadro môže indikovať úlohe, že sa môže vzdať cpu, ak chce
- preemptívna

Plánovače

- štandardný plánovač
  - **nice**: priorita úlohy -20 po +19
  - čím menší nice, tým viac procesorového času má proces
  - default: 0
- úrovne priorít
  - čím menší nice (= vyššia priorita), tým viac cpu time
  - milý proces dostáva viac cpu time

- dávkový plánovač
  - lepší pre výpočty

- FIFO (front)
  - obdoba nepreemptívneho módu
  - pridelí sa mu úloha
  - čaká sa, kým sa úloha nevzdá cpu time
  - vhodné pre úlohy v reálnom čase:
    - VoIP, spracovanie obrazu, kryptovanie
- mFIFO (modified FIFO) a.k.a. round robin
  - dokola prideľuje cputime len úlohám, ktoré sú definované ako prioritné
  - úlohu možno označiť ako prioritné
  - priorita od 0-99

- štandardne štandarndý plánovač
- ostatné obvykle len pri embedded zariadeniach a pod.

Personalities

- po zavedením procesu do pamäte
- pred odovzdaním riadenia
- možno nastaviť procesu časovac

Proces

- PCB = process control block
- PID, pamäť, handler, heap, stack, tabuľka resourcov
- UID = reálne user ID
- EID = efektívne user ID
- ENV = premenné prostredia
- base pointer, instruction pointer (EBP, ESP, EIP)
- kontrola konzistencie programu
- sanity checks: kontrola povolených inštrukcií

- na základe binárnej hlavčiyk sa vytvorí ECB
- formát ELF
  - principiálne podobný hlavičke EXE
  - začiatok a koniec prog. kódu
  - offsety dátových štruktúr
  - ...
- COM v DOSe nemal hlavičku

Koniec procesu

- volanie exit()
- možno zaregistrovať final handler
  - on exit
  - on end
- ak program končí exit(), zavolá sa final handler
- vieme upratovať globálne premenné, vyriešiť upratovanie potomkov, upozorniť deti, že končíme a pod.
- _exit()

- potomok vznikne fork()om
- zdieľa zdroje s rodičom

Spúšťanie špecifických procesov

- prípony netreba
  - na rozdiel od Windowsu
- nástroj **file** vie zistiť typ súboru
  - nahliadne do hlavičky
- alternatívne shebang line určí interpreter
  - #!/bin/bash
- linux dokáže spustiť hocičo, ak pre to existuje riadiaci program

- jadrový modul **binfmt_misc**
- **/proc/fs/binfmt_misc**
- možno zaregistrovať súbor s magickým reťazcom
- ak sa nájde CAFEBABE (Java), vie sa rovno zavolať Java
- je to záležitosť OS
  - Windows: záleží na príponách

Spúšťacie domény

- execution domains
- natívne možno spúšťať kódy iných POSIX systémov
  - napr. FreeBSD program na Linuxe
  - musia sedieť architektúry procesora
- transformujú systémové volania
  - jeden odovzdáva parametre sys. volaní na stacku
  - iný v registroch