Utilizatori și procese

Utilizarea Sistemelor de Operare

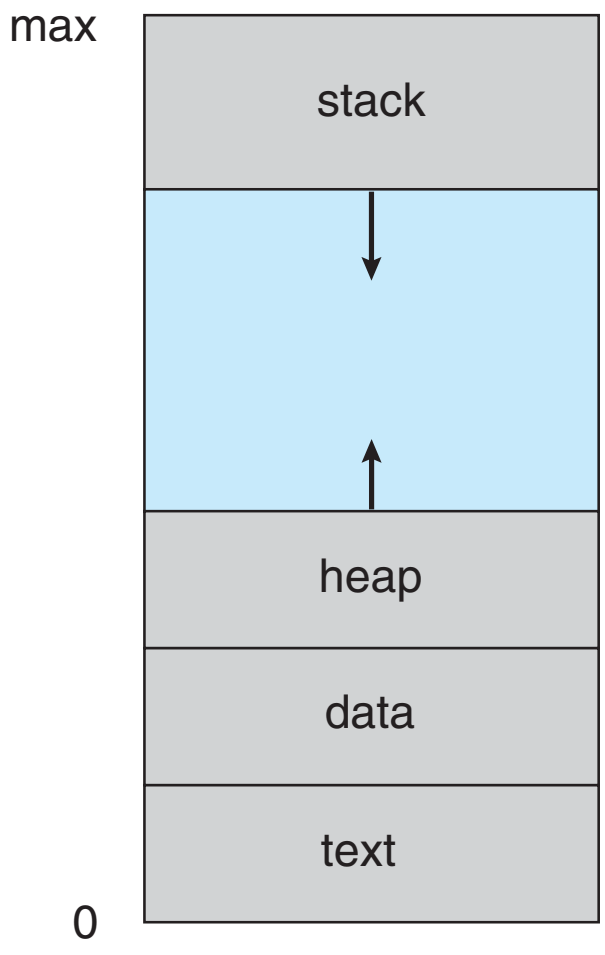
Paul Irofti

Universitatea din București
Facultatea de Matematică și Informatică
Department de Informatică
Email: paul.irofti@fmi.unibuc.ro

Proces

- un proces reprezintă un program în execuție
- el este însoțit de date suplimentare față de codul asociat programului
  - program counter
  - regiștri
  - date globale
  - stiva – include parametrii funcției curente, adresa de întoarcere, variabile locale etc.
- un program devine proces când este încărcat în memorie

Structura unui proces



http://codex.cs.yale.edu/avi/os-book/

Utilizator

- un actor în cadrul sistemului de operare
- nu neapărat un actor uman
- folosit în scopuri de organizare și separare a datelor și proceselor
- un utilizator poate avea mai multe procese în execuție
- un sistem de operare poate avea mai mulți utilizatori activi concomitent
- **probleme**: securitate, intimitate, acces (echitabil) la resurse

Securitate – Autentificare

- parolă
- cheie criptografică
- cu ajutorul mai multor factori

Autentificare – parolă

Pereche `user:pass`

- fiecare utilizator are asociată o parolă de acces
- cel mai vechi și des întâlnit mecanism de autentificare
- parolele trebuie să fie schimbate des (6 luni)
- evitați parole scurte și simple:
  - un singur cuvânt (atac de dicționar)
  - data nașterii sau alte date personale (găsite în conturi și acte publice)
  - parole care conțin doar litere sau doar cifre (atac brute-force)
- comandă în Unix: `passwd(1)`
- fișiere: `/etc/passwd`, `/etc/master.passwd`
- parola este salvată criptografic (ex. folosind o funcție de hashing)
- folosită în URI: `https://user:pass@mail.fmi.unibuc.ro/`

Autentificare – cheie criptografică

Criptografie asimetrică

- autentificare cu ajutorul a două chei: una publică și una privată
- cheia publică poate fi distribuită tuturor (ex. homepage)
- cheia publică este preluată de terți și instalată pe sistemele la care vor să vă dea acces
- cheia privată este folosită împreună cu cheia publică pentru autentificare și acces
- cheia privată poate fi însoțită de o parolă (2FA)
- cheia privată trebuie păstrată în siguranță
- dacă cheia privată a fost compromisă, trebuie creată o nouă pereche de chei iar cea publică trebuie înlocuită pe toate sistemele
- comandă în Unix: `ssh-keygen(1)`
- fișiere: `~/.ssh/id_{dsa,ecdsa,ed25519,rsa}[.pub]`

Autentificare – 2FA

Two Factor Authentication (2FA)

- autentificare folosind doi factori (ex. ATM: card și PIN)
- pe calculator se folosește de regulă parola și un număr (PIN)
- PIN-ul poate fi trimis după autentificarea cu parolă prin SMS
- PIN-ul poate fi un număr pseudo-aleator generat de un token
- token-ul poate fi o aplicație de telefon (ex. Authy) sau un dispozitiv separat (folosit în general de bănci)
- pe laptop și telefoane mobile se pot folosi senzori biometrici (ex. amprenta) pentru al doilea factor

Intimitate – Date

- separarea accesului la fișiere și directoare între utilizatori
- drepturi de acces primare: citire, scriere, execuție
- cine aribtrează separarea?
- o soluție: fiecare director și fișier aparțin unui singur utilizator
- rezultat: lucrul în echipă și distribuirea datelor devine dificil
- îmbunătățire: grupuri de utilizatori care au acces

Grupuri

- mai mulți utilizatori pot aparține unui grup
- un utilizator poate aparține mai multor grupuri
- drepturi de acces la nivel de grup
- exemplu: grupul `staff` pentru cei ce administrează sistemul
- comandă în Unix: `groupadd(1)`, `usermod(1)#-G`
- exmeplu: `groupadd gr151 && usermod alex -G gr151`
- fișiere: `/etc/group`, `/etc/passwd`

Acces la date – Unix

Fișierele și directoare au trei tipuri de acces pentru trei categori

- acces: citire (`r`), scriere (`w`), executare (`x`)
- reprezentați prin biți (ex. `000`→`---`, `110`→`rw-`, `101`→`r-x`)
- categorii: utilizator (`u`), grup (`g`), restul (`o`)
- `ls(1)#-l` codifică `drwxrwxrwx`
  - prima literă spune dacă obiectul este director sau nu
  - urmată de grupuri de acces pentru utilizator, grup și restul
- directoarele trebuie să aibă `x` ca să poată fi parcurse
- `-r--r--r--` – un fișier cu drepturi de citire pentru toată lumea
- `dr-xr-xr--` – un director în care pot intra doar utilizatorul și cei din grup
- ce se întâmplă dacă vrem să dăm acces la mai mult de un grup?
- comenzi: `chown(1)`, `chmod(1)`

Intimitate – Procese

- separarea accesului la procese între utilizatori
- programele sunt executate de utilizatori $\rightarrow$ procese
- procesele sunt deținute de utilizatorul care le-a pornit
- procesele au acces la resursele sistemului în funcție de privilegiile utilizatorului
- un proces fiu creat de alt proces moștenește drepturile părintelui
- Unix: un proces poate avea dreptul de a-și schimba utilizatorul și grupul de care aparține: `setuid` și `setgid`
- procesele unui utilizator nu pot fi oprite, pornite sau manipulate în alt mod de alți utilizatori
- excepție: administratorii sistemului
- comandă în Unix: `ps(1)`, `top(1)`, `kill(1)`

Resurse

- un proces poate influența execuția altora prin modul în care folosește resurse
- pe un procesor poate fi activ un singur proces la un moment dat
- ordinea în care intră pe procesor un proces este dictată de algoritmul de scheduling din sistemul de operare (numit și `scheduler`)
- `scheduler`-ul ordonează execuția în funcție de prioritatea (ponderea) asignată proceselor
- Unix: prioritatea proceselor obișnuite este între -20 (cea mai mare) și 20 (cea mai mică)
- comandă în Unix: `nice(1)`, `renice(8)`

Adiministrare și comenzi

- cod de administrator: `root` în Unix, `administrator` în Windows
- utilizator cu drepturi comune pentru activitățile de zi cu zi
- apel la contul de administrare doar când este cazul
  - instalare de software
  - actualizarea sistem de operare
  - modificare fișiere de configurare etc.
- comenzi Unix: `su(1)`, `sudo(1)`

su(1)

- su(1) – substitute user
- implicit execută un shell drept root (#)
- cere parola înainte de a schimba utilizatorul
- la ieșire revine la shell-ul utilizatorului original
- su alex – pornește un shell drept utilizatorul alex
- su -c cmd – execută cmd fără a intra în shell
- su alex -c cmd – execută cmd drept utilizatorul alex

sudo(8)

`sudo` – substitute user and do

- funcționalitate similară cu `su#-c`
- bazat pe un set de reguli ce includ
  - ce utilizatori pot apela comanda (implicit cei din grupul `wheel`)
  - au nevoie de parolă pentru execuție?
  - implicit e nevoie de parola utilizatorului curent
  - restricționarea la un set specific de comenzi ce sunt permise (ex. `alex` execută doar `ls(1)` ca `root`)
- set de reguli scris în `/etc/sudoers`
- editat cu `visudo(8)` – `vi(1)` specializat pe format `sudo(8)` ce verifică erori înainte de a modifica configurația
- ce face comanda: `sudo su`?

Administrare Unix – comenzi uzuale

- `useradd(8)` – adăugare utilizator (`adduser(8)` – versiunea interactivă)
- `userdel(8)` – ștergere utilizator
- `usermod(8)` – modificare date utilizator (`$HOME`, grupuri, `$SHELL`)
- `groupadd(8)` – adăugare grup
- `groupdel(8)` – ștergere grup
- `groupmod(8)` – modificare grup (schimbare nume, id)
- `vipw(8)` – editor `vi(1)` specializat pentru fișierul `passwd`
- `chpass(8)` – schimbare date utilizator (nume, telefon, când să schimbe parola)
- `id(1)`, `finger(1)`, `who(1)`, `w(1)`, `last(1)` – informații despre utilizatori și activitatea lor recentă

Administrare Unix – nomenclatură

- `pid` – proccess identification
- `ppid` – parent proccess identification
- `tid` – thread identification
- `uid` – user identification
- `gid` – group identification
- `res` – resident memory (cât ocupă în memoria principală)
- `pri` – prioritatea procesului
- `state` – starea procesului (gata de execuție, în așteptare, zombie, în execuție etc.)
- `init` – primul proces în sistem (`pid`$= 1$)
- `wchan` – numele funcției în care procesul e suspendat așteptând evenimente sau date noi