SO – 4

Comunicació Interprocés

* Comuniació Interprocés: els processos han de comunicar-se, enviar-se informació entre sí
  + Tècniques:
    - Canonades
    - Arxius
    - Xarxa
    - Senyals
  + POSIX (Portable Operating System Interface): mantenir la compatibilitat entre SO
    - Un fitxer a POSIX pot fer referència a qualsevol sistema de comunicació
    - Tots els fitxers oberts en un SO tipus POSIX tenen un descriptor de fitxer com un sencer no negatiu.
    - Aquest descriptor és un índex a un vector del procés que conté la informació dels fitxers oberts
  + Crides a SO per fer la comunicació interprocés
    - Obrir una comunicació (Retornen el descriptor de fitxer)
      * Open (obrir fitxer)
      * Pipe
      * Mkfifo (canonada amb nom)
      * Socket (crear connexió via xarxa)
    - Write i Read, Close
  + Canonades
    - Desavantatge: només permeten la comunicació interprocés si els processos tenen una relació pare-fill
    - Canonades amb nom:
      * Al obrir una canonada amb nom, se li associa un nom de fitxer de disc.
      * Qualsevol procés pot obrir aquest fitxer per lectura/escriptura
  + Arxius
    - Avantatges:
      * El disc es un dispositiu on hi poden accedir tots els processos
      * Un procés es pot situar a la posició del fitxer que vulgui
      * No cal utilitzar canonades amb nom
    - Desavantatges:
      * Manipular arxius és més complicat que manipular directament la memòria
      * Assegurar que el fitxer conté dades consistents en cas de múltiples processos hi puguin llegir/escriure
    - Funcions:
      * Crides a SO
        + Open: obre un fitxer a disc. Retorna el seu descriptor
        + Iseek: establir la posició actual dins del fitxer
        + L’SO utilitza un buffer intern per augmentar l’eficiència
      * Llibreries: fopen, fread, fwrite
        + Utilitzen estructura FILE: un buffer propi per emmagatzemar dades
        + Fprint o Fwrite escriuren al buffer propil. Quan es ple, es crida write que les escriu al buffer intern
        + Fscanf o fread llegeixen les dades del buffer propi, si no estan disponibles, s’utiltiza read per omplir el buffer de FILE.
        + Minimitza ús de write i read.
        + No són adequades per la comunicació interprocés

En el cas que hi hagi múltiples processos, cada un d’ells té el seu propi buffer, i no es podrà veure per l’altre procés

Crides a sistema amb bloc petit: molt temps (passar de mode nucli a mode usuari)

Crides a sistema amb bloc gran: poc temps

Funcions de llibreria amb bloc petit: poc temps

Funcions de llibreria amb bloc gran: molt temps

Amb crides a la llibreria el temps d’usuari és més llarg pel codi addicional de les crides a les funcions

Amb crides a llibreria el temps de sistema és independent de la mida del bloc

* + Xarxa (Sockets)
    - Comunicació interprocés entre diferents ordinadors
    - 8 funcions bàsiques: socket, bind, listen, accept, connect, read, write i close
    - Un procés A pot escoltar les peticions entrants enganxan-se a un port
    - Un altre procés B pot enviar-li un missatge...:
      * Obrint un canal de comunicació amb la mateixa IP i el port
  + Senyals: interrupcions de programari. Gestionen notificacions asíncrones
    - El SO pot enviar una notificació asíncrona a un procés (Upcall) i un procés pot enviar una notificació asíncrona a un altre procés
    - Serveix
      * Quan es produeix una excepció, el SO pot cridar a la funció del SO que l’ha provocat.
      * En Sistemes d’Alimentació Ininterrompuda (SAI) el SO pot cridar a una funció de cada procés que s’està executant.
    - Que es pot fer quan es produeix un senyal:
      * Es captura: el procés indica al SO quina funció cal cridar
      * Acció per defecte: exit – dump core
      * Ignorar el senyal: SIGKILL i SIGSTOP no es poden ignorar