T3-Memoria

3.1 Espacio de @ de un proceso

- Espacio de direcciones del **procesador:** conjunto de @ que el procesador puede emitir.
- Espacio de direcciones **lógicas de un proceso:** conjunto de @ lógicas que un proceso puede referenciar.
- Espacio de direcciones **físicas de un proceso:** conjunto de @ físicas asociadas al espacio de direcciones lógicas del proceso.

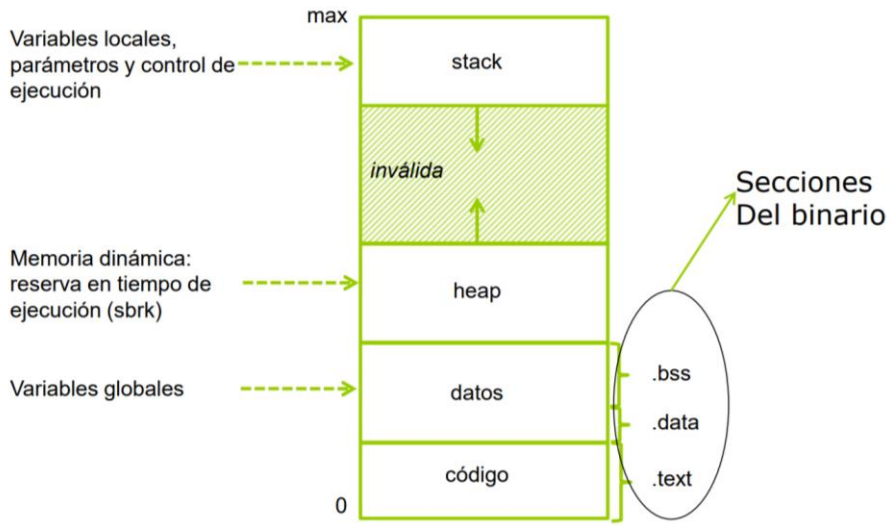
3.2 Sistemas multprogramados

- Se tiene que garantizar **protección de la memoria física**: un proceso no puede acceder a una memoria física que no se le haya asignado.
- **Memory Management Unit (MMU):** traduce y guarda la correspondencia entre direcciones lógicas y direcciones físicas en tiempo de ejecución.

3.3 Traducción

- **Asignar memoria**:
  - Inicialización al asignar una nueva memoria (mutación, execlp)
  - Cambios en el espacio de direcciones: pedir/liberar memoria dinámica.
- **Cambio de contexto**:
  - Proceso que abandona la CPU: almacenar PCB e información para configurar la MMU:
  - Proceso que ocupa la CPU: configurar la MMU.

3.4 Esquema del proceso en memoria



Block Starting Symbol (bss): parte de un ejecutable que tiene variables declaradas sin inicializar.

3.5 Optimizaciones de carga

- **Carga bajo demanda:**
  - Una rutina no se carga hasta que se llama.
  - Acelera el proceso de carga.
  - Cuando se necesita la rutina el SO genera una excepción, la carga en memoria y reinicia la instrucción que ha provocado la excepción.
- **Librerías compartidas:**
  - Disco no contiene código de las librerías dinámicas, sólo un enlace que **se retrasa hasta el momento de ejecución**.
  - Procesos comparten ese espacio de memoria.
  - **Stub:** código de una rutina de enlace que hace de puente a la que realmente contiene el código.

3.6 Reservar/Liberar memoria dinámica: heap

- **Memoria dinámica:** se guarda en el espacio **heap** de memoria lógica.
- **int sbrk(tam_variación_heap):** devuelve la @ del top del heap.
  - > 0: aumenta el heap
  - < 0: reduce el heap
  - = 0: el heap no se modifica
- **int malloc(tam_variación_heap):** devuelve la @del top del heap.
  - Reserva un poco más (dependiendo del SO) de lo que le has pedido en caso de que luego necesites más memoria.

3.7 Problema fragmentación

Asignar una cantidad X en una zona más grande.

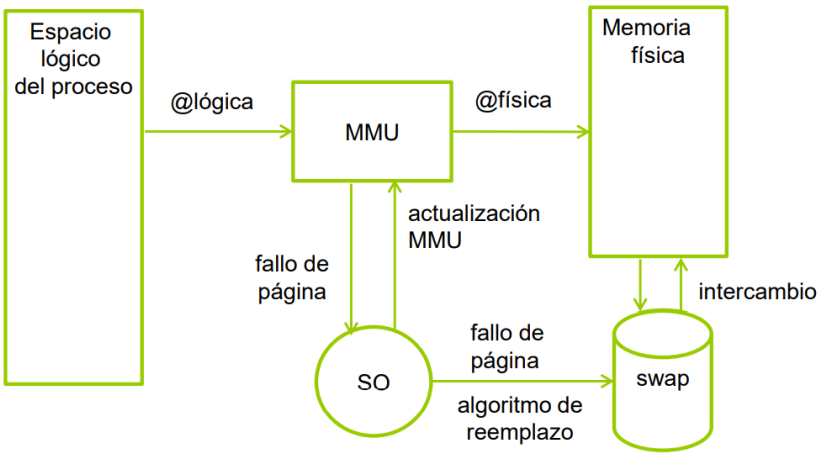
- **Fragmentación interna:** memoria asignada a un proceso aunque no la necesita.
- **Fragmentación externa:** memoria libre y no asignada pero no se puede asignar por no estar contigua.

3.8 Asignación de memoria

- Tipos de asignación:
  - **Asignación contigua:** poco flexible y dificulta aplicar optimizaciones.
  - **Asignación no contigua:** aumenta flexibilidad, granularidad y complejidad del SO y de la MMU.
- Basado en:
  - **Paginación:** espacio de @lógicas dividido en particiones de tamaño fijo: **páginas**.
    - Memoria física dividida en particiones del mismo tamaño: **marcos.**
    - **Tabla de páginas:** una tabla por proceso para mantener información a nivel de página.
  - **Segmentación:** espacio de @ lógicas dividido en particiones de tamaño variable: **segmentos.**
    - Selecciona la cantidad de memoria necesaria para el segmento y el resto continúa en la lista de particiones libres.

3.9 Optimizaciones

- **Copy on Write (COW):** retrasar el momento de la copia de código, datos, etc. Mientras sólo se acceda en modo lectura.
    - Se puede hacer a nivel de página o segmento.
    - En un fork por ejemplo, sólo se reservará memoria cuando alguna de las variables compartidas entre padre e hijo cambie de valor.
- **Memoria virtual:** un proceso realmente sólo necesita memoria física para la instrucción actual y los datos que esa instrucción referencia. "sacar" cosas bajo demanda.
- **Swapping:**
    - Si el proceso que está en CPU necesita más memoria, podemos expulsar temporalmente de memoria uno de los procesos cargados (**swap out**).
    - **Swap area:** zona de disco para guardar los procesos que han sido expulsados de memoria principal.
    - **Backing storage:** guarda espacio lógico de los procesos a la espera de ocupar la CPU.



- **Thrashing (sobrepaginación):** invierte más tiempo en el intercambio de memoria que avanzando su ejecución.
- **Prefetch:** minimizar número de fallos de página.

T4-Entrada/Sortida

4.1 Principis de disseny

- **Independència dels dispositius**
- Possibilitat de **redireccionament**

4.2 Nivells

- **Dispositiu (nivell) virtual:** correspondència entre **nom simbòlic (de fitxer)** i aplicació d'usuari. **Nombre enter**
- **Dispositiu (nivell) lògic:** nom de fitxer. Es tenen en compte permisos.
- **Dispositiu (nivell) físic:** inicialitza el dispositiu, retorna resultats. S'identifica:
  - Tipus: **bloc/character**
  - **Major:** tipus de dispositiu
  - **Minor:** quina instància de major és.

4.3 Exemples de dispositius

- **Pipe:** buffer temporal amb estructura FIFO.
  - **Pipes sense nom:** comunicar processos amb parentesc.
  - **Pipes amb nom:** permet connectar qualsevol procés que tingui permís per accedir al dispositiu.
- **Socket:** igual que pipe però per processos en diferents computadores.

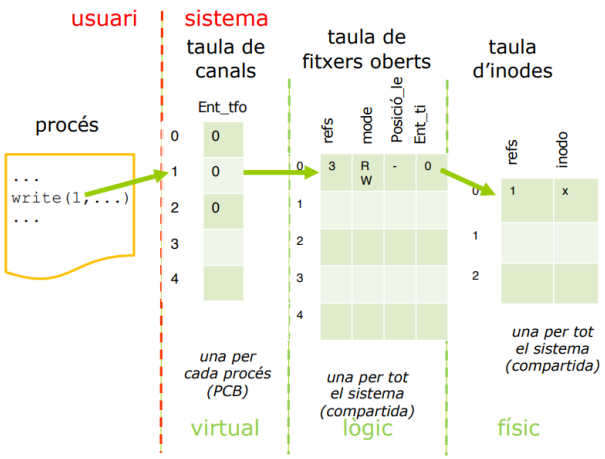
4.4 Tipus de fitxers

- **C:** dispositiu de caràcters
- **B:** dispositiu de blocs
- **P:** pipe (no cal posar major/minor)

4.5 Inodes

Estructura de dades que conté tota la informació relativa a un fitxer:

- Tamany
- Tipus
- Permisos
- Propietari i grup
- Data de modificació, creació...
- Nombre d'enllaços a l'inode
- Índex de les dades



4.6 Operacions d'E/S

Crida a sistema	Descripció
open	Donat un nom de fitxer i un mode d'accés, retorna el nombre de canal per poder accedir
read	**Llegeix N bytes d'un dispositiu** (identificat amb un nombre de canal) i ho guarda a memoria
write	Llegeix N bytes de memoria i els **escriu al dispositiu** indicat (amb un nombre de canal)
close	Allibera el canal indicat i el deixa lliure per ser reutilitzat.
dup/dup2	Duplica un canal. El nou canal fa referencia a la mateixa entrada de la TFO que el canal duplicat.
pipe	Crea un dispositiu tipus pipe llesta per ser utilitzada per comunicar processos
lseek	Mou la posició actual de L/E (per fitxers de dades)

Les crides open, read i write poden bloquejar el procés

- **Fd = open(nom, access_mode [,permission_flags]);**
  - access_mode:
    - O_RDONLY
    - O_WRONLY
    - O_RDWR
  - Fd: dispositiu virtual
  - permission_flags:
    - O_CREAT: crea el fitxer si no està creat, amb | i després els permisos (0664 és el més comú).
- **n = read(fd, buffer, count);**
  - n: nombre de bytes (caràcters) llegits.
  - Count: llegeix tants bytes con count digui.
- **n = write(fd, buffer, count);**
  - n: nombre de bytes escrits.
  - Count: nombre de bytes a escriure.
- **Newfd = dup(fd);**
  - Ocupa el primer canal lliure i hi copia el dispositiu fd. Newfd és l'identificació del nou canal.
- **Newfd = dup2(fd, newfd);**
  - Igual que dup però ara en el canal newfd hi haurà el mateix que a oldfd.
- **Close(fd);**
- **Pipe(fd_vector);**
  - Crea una pipe sense nom.
- **Mknod(name, mode, dev) + open;**
  - Crea una pipe amb nom
  - Name: nom de la pipe
  - Mode: S_FIFO|0666
- **Nova_pos = lseek(fd, despl, relatiu_a);**
  - Despl: desplaçament des de relatiu a
  - Relatiu_a:

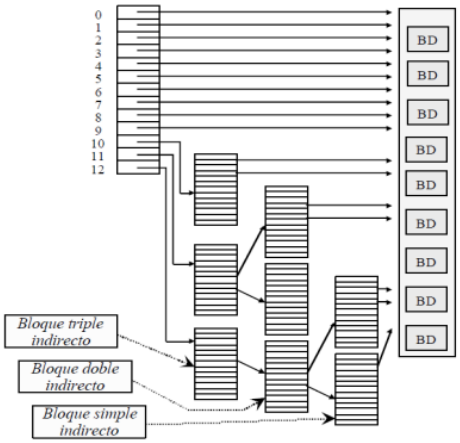
- SEEK_SET: principi del fitxer.
- SEEK_CUR: lloc al que estàs actualment.
- SEEK_END: final del fitxer.

4.7 Pipes

- Lectura:
  - Si hi ha dades, agafa les que necessita (o les que hi hagi)
  - Si no hi ha dades, es queda bloquejat fins que n'hi hagi alguna
  - Si la pipe està buida i no hi ha cap possible escriptor (tots els canals d'escriptura de la pipe tancats), el procés lector rep EOF
- Escriptura
  - Si hi ha espai a la pipe, escriu les que té (o les que pugui)
  - Si la pipe està plena, el procés es bloqueja
  - Si no hi ha cap possible lector, rep SIGPIPE.

4.8 Sistema de fitxers: gestió de l'espai ocupat

- Índexos a blocs de dades (1/4 Kb)
  - 10 índexos directes
    - (10 blocs = 10/40Kb)
  - 1 índex indirecte
    - (256/1024 blocs = 256Kb/4Mb)
  - 1 índex indirecte doble
    - (65K/1M blocs = 65Mb/4 Gb)
  - 1 índex triple indirecte
    - (16M/1G blocs = 16Gb/4 Tb)



4.9 Metadades

- **Superbloc:** bloc de partició que conté les metadades del sistema de fitxers.