

Sistemes Operatius

Unitat 5 - Nel Banqué Torné

Introducció: Necessitat de la gestió de la Memòria

Un sistema operatiu sense gestió de memòria tindria les següent problemàtiques:

- **Impossibilitat de multitàsca:** Dos programes no poden residir simultàniament a la memòria.
- **Comportament erràtic:** L'accés concurrent sobreescrivia dades crítiques.
- **Solució poc viables:** L'ús de swapping constant a memòria secundària seria extremadament lent.

Funcionalitats clau per a la gestió de la memòria:

- Traducció d'adreces.
- Reubicació de dades.
- Protecció de la memòria.
- Compartició de recursos.
- Optimització de l'ús dels recursos de memòria.
- Mecanismes per evitar fragmentació excessiva.

Què és el mapa de memòria?

El mapa de memòria d'un procés és una representació lògica de com la memòria està organitzada per a un programa en execució. Inclou:

- **Regions:** Zones contigües de memòria amb punt inicial i mida definida.
 - **Support:** Contingut inicial de la regió.
 - **Tipus:** Fitxer, anònim.
 - **Protecció:** Lectura, escriptura, execució.
 - **Mida:** Fixa o variable.

Tipus de regions:

- **Codi:** Lectura i execució. Mida fixa. Conté el codi del programa.
- **Dades inicialitzades:** Variables globals i estàtiques inicialitzades. Lectura, escriptura i mida fixa.
- **Dades no inicialitzades:** Lectura i escriptura. Mida fixa.
- **Heap:** Memòria dinàmica. Lectura, escriptura i mida variable.
- **Stack:** Emmatzema crides a funcions. Lectura, escriptura i mida variable.

Observacions:

- Programes amb dades desconegudes durant la compilació necessiten un model de memòria dinàmic.

- Biblioteques dinàmiques afegeixen regions noves en temps d'execució.
 - Alguns sistemes moderns utilitzen memòries no volàtils per a un accés més ràpid i eficient.
-

Espais d'adreces:

- **Espai lògic:** Adreces que contenen codi, dades i pila del procés en temps d'execució.
- **Espai físic:** Adreces reals a la memòria física on es carrega l'espai lògic.
- **Espai CPU:** Adreces generades per la CPU per accedir als segments.

Exemple:

Un programa espera certs continguts en posicions de memòria específiques. La manca de reubicació pot trencar aquestes expectatives. Aquest problema és especialment crític en entorns compartits, com ara servidors amb múltiples usuaris.

Reubicació de memòria:

Permet adaptar les adreces físiques d'un programa en funció de la memòria disponible:

- **Reubicació estàtica en temps de compilació:** Les adreces físiques es generen durant la compilació.
- **Reubicació estàtica en temps de càrrega:** Es realitza una traducció d'adreces lògiques a físiques en temps de càrrega.
- **Reubicació dinàmica en temps d'execució:** Permet moure processos entre zones de memòria. Alenteix l'execució.

En sistemes avançats, la reubicació dinàmica pot ser gestionada per maquinari específic, com ara unitats de traducció d'adreces (MMU).

Mecanismes clau:

Protecció:

- Garanteix que cap procés accedeixi de forma accidental o maliciosa a dades d'un altre procés.
- Es verifica cada referència a la memòria en temps d'execució.
- La protecció és clau en entorns multiusuari per garantir la privadesa i la seguretat.

Compartició:

- Blocs de memòria poden ser compartits entre processos.

- Calen mecanismes de comunicació i sincronització per a un accés segur.
 - Un ús típic és la compartició de biblioteques dinàmiques per reduir l'ús de memòria.
-

Models d'organització de memòria:

Màquina nua:

- Sense sistema de gestió de memòria.
- L'usuari té control complet.
- No hi ha protecció ni eficiència.
- Utilitzat principalment en sistemes molt antics o entorns especialitzats.

Monitor resident:

- Una part de la memòria està dedicada al monitor resident.
- Simplifica la gestió però limita els recursos disponibles pels processos.

Particions múltiples:

- La memòria es divideix en particions, cadascuna per a un procés.
- **Polítiques:** First-fit, best-fit.
- Avantatge: Facilitat d'implementació.
- Desavantatge: Problemes de fragmentació.

Fragmentació:

- **Interna:** Espai no utilitzat dins d'una partició.
- **Externa:** Espai lliure no contigu que no es pot utilitzar eficientment.
- Solucions modernes com la paginació intenten minimitzar aquests problemes.

Paginació:

- Divideix la memòria en pàgines i marcs.
- Permet eliminar fragmentació externa, però pot generar fragmentació interna.
- Mecanismes com la paginació en diversos nivells augmenten l'eficiència.

Segmentació:

- La memòria es divideix en segments que reflecteixen les regions del programa.
- Avantatges: Absència de fragmentació interna i flexibilitat.
- Problemes: Fragmentació externa i necessitat de mètodes de compactació.
- Molts sistemes combinen segmentació i paginació per aprofitar els avantatges d'ambdues tècniques.

Anàlisi i exemples:

- **Super Mario Land 2:** La manca de protecció de memòria permet accés a zones no destinades, provocant glitches.
- **Fragmentació:** Exemple d'una partició de 1000 bytes assignada a un procés que només necessita 800 bytes (fragmentació interna).
- **Entorns empresarials:** La segmentació és habitual en entorns de bases de dades grans, on diferents segments poden contenir índexs, registres i dades temporals.

Conclusió:

La gestió de la memòria és fonamental per garantir l'eficiència i seguretat dels sistemes operatius moderns, mitjançant mecanismes com la paginació, segmentació i reubicació. Un bon disseny de la gestió de memòria pot marcar la diferència en el rendiment i la robustesa del sistema. A més, els avenços tecnològics segueixen aportant noves solucions per optimitzar l'ús de recursos i reduir les limitacions clàssiques.