Cues i Planificació

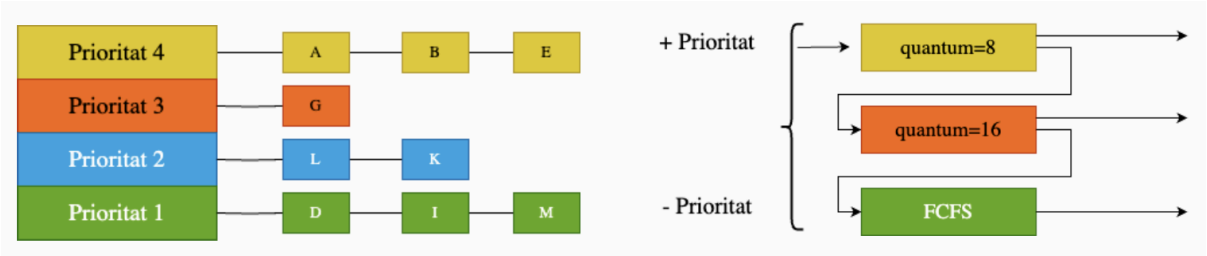
Unitat 3.1 - Nel Banqué Torné

La planificació de processos és una funció crítica del sistema operatiu que gestiona l'assignació de la CPU als processos actius. Diferents estratègies i algoritmes asseguren un ús eficient dels recursos, una execució justa i una resposta ràpida per a tasques interactives.

---

1. Cues Multinivell

Les cues multinivell són una tècnica de planificació en què els processos es classifiquen en diverses cues, cadascuna amb polítiques i prioritats diferents.



Busquem processos que, el menys prioritaris promocionin als de dalt i els més prioritaris degradin els d'abaix.

1.1. Cues Multinivell sense Retroalimentació

- **Característiques**:
  - Els processos es col·loquen en cues basant-se en característiques fixes, com ara el temps d'execució, la intensitat d'ús de la CPU o les necessitats d'E/S.
  - Cadascuna de les cues té una **política de planificació específica**:
    - **Round Robin** per a processos interactius.
    - **First-Come-First-Served** per a tasques de fons.
- **Estratègies de Prioritat**:
  - **Prioritat Absoluta**:
    - Els processos de les cues amb més prioritat són completats abans d'assignar la CPU a cues inferiors.
    - Ex.: En un sistema de cua triple, la cua A obtindrà tots els recursos abans que s'executin tasques de la cua B.
  - **Assignació Temporal**:
    - Cada cua rep un percentatge determinat del temps de CPU.
    - Ex.: Un sistema pot assignar el 50% del temps a tasques interactives, un 30% a tasques semiprioritàries i un 20% a tasques de fons.

- **Exemples Pràctics**:
    - **Aplicació en Sistemes Bancaris**:
        - Peticions urgents (transferències en temps real) s'assignen a la cua de màxima prioritat.
        - Processos menys crítics, com generar extractes, van a cues de baixa prioritat.
- **Limitacions**:
    - La manca de retroalimentació fa que els processos no puguin adaptar-se a canvis dinàmics, afectant-ne l'eficiència en entorns canviants.

1.2. Cues Multinivell amb Retroalimentació

- **Característiques Dinàmiques**:
    - Els processos es mouen entre cues en funció del seu comportament.
    - **Promoció**:
        - Si un procés consumeix pocs recursos o compleix els seus objectius dins del quantum assignat, pot ser promocionat a una cua de major prioritat.
    - **Degradació**:
        - Els processos que excedeixen el quantum de temps o que són molt intensius en CPU descendeixen a cues amb menys prioritat.
- **Avantatges**:
    - Redueix la latència per a tasques interactives.
    - Optimitza l'ús de recursos evitant que processos llargs bloquegin el sistema.
- **Exemples**:
    - **Sistemes Multitasca**:
        - Els navegadors web mantenen la seva prioritat alta gràcies al seu ús interactiu.
        - Tasques de còpia de seguretat o compressió de fitxers descendeixen progressivament en prioritat.

---

2. Estratègies de Planificació Avançades

2.1. Planificació Justa

- **Objectiu**:
    - Garantir una distribució equitativa dels recursos entre processos o usuaris.
- **Algoritmes Associats**:
    - **Fair Sharing**: Proporcionalitat estricta segons el nombre de processos.
    - **Max-Min Fairness**: Redistribució dels recursos no utilitzats per alguns processos cap a d'altres.
    - **Fair Queueing**: Assigna recursos basant-se en temps de finalització virtual calculat.

- **Exemple Pràctic**:
  - **Infraestructura Cloud**:
    - Els recursos de CPU i memòria es distribueixen entre clients segons les seves necessitats i contractes.

2.2. Planificació Garantida

- **Descripció**:
  - Cada procés rep una quota mínima garantida de temps de CPU (1/N del temps disponible si hi ha N processos actius).
  - Selecciona processos basant-se en el ratio de temps utilitzat respecte al temps esperat.
- **Exemple**:
  - En un sistema amb 5 processos actius, cada procés rep, com a mínim, el 20% del temps total disponible.

N processos en execució $\Rightarrow$ cada procés rep almenys $1/N$ del temps de CPU.

Calcula el ratio de temps consumit respecte al temps des de la seva creació.

- $\frac{1}{2} \Rightarrow$ 50% del temps de CPU.
- $\frac{2}{1} \Rightarrow$ 200% del temps de CPU.

Seleccionem sempre el procés amb el ratio més baix per garantir que tots els processos rebin una part equitativa del temps de CPU.

2.3. Planificació amb Algorisme Loteria

- **Funcionament**:
  - Cada procés rep una quantitat de "bitllets" segons la seva prioritat.
  - Un sorteig determina quin procés s'executa.
  - Els processos amb més bitllets tenen més probabilitats de ser seleccionats, però els de menor prioritat també tenen opcions.
- **Avantatges**:
  - Introducció de variabilitat i equitat probabilística.
  - Els processos poden transferir bitllets entre si per cooperar.
- **Exemple**:
  - En una xarxa de videojocs en línia, les tasques que gestionen les accions dels jugadors poden tenir més bitllets per garantir una resposta ràpida.
- **És just?**
  - No, qui tingui més bitllets té més possibilitats de guanyar, no tots tenen la mateixa probabilitat.

---

3. Planificació de Processos en Linux

3.1. Gestió de Prioritats

- **Prioritats del Sistema**:
  - 0–99: Tasques de temps real (temps crític).
  - 100–139: Tasques normals.
- **Ajust de Prioritats**:
  - Les prioritats es modifiquen amb la comanda `nice` o `renice`, que permet ajustar el comportament d'un procés.
  - Els processos interactius guanyen crèdits si romanen inactius durant períodes llargs.

3.2. Tasques de Temps Real

- **SCHED_FIFO**: Assigna la CPU a un procés fins que finalitza o cedeix la seva execució.
- **SCHED_RR**: Implementa un esquema Round Robin per processos de temps real amb igual prioritat.
- **Exemple**:
  - En un sistema de control industrial, una tasca que monitoritza la temperatura d'un forn té prioritat màxima i s'assigna a SCHED_FIFO.