

Sémaphores

main.c

```
int main() {
    return 0;
}
```

Interblocage de processus

Conditions nécessaires (simultanées) d'interblocage :

- Exclusion mutuelle
- Occupation et attente : Processus occupant au moins une ressource et qui attends d'acquérir des ressources supplémentaires détenues par d'autres processus
- Pas de réquisition : Les ressources déjà détenues ne peuvent être retirées de force à un processus
- Attente circulaire

Si chaque type de ressource possède exactement un seul exemplaire alors : il y a situation d'interblocage SSI le graphe d'allocation possède un circuit

Cas des ressources possédant plusieurs exemplaires : Si le graphe d'allocation est sans circuit alors un processus n'est pas dans une situation d'interblocage (réciproque fausse)

=> Interblocage s'il existe un sous-ensemble D (ensemble des sommets formant un circuit) non vide de processus tel que, pour tout P_i appartenant à D, l'inégalité : $DEMANDE[i] \leq N - \text{somme}(\text{ALLOCATION})$ avec $N[i]$ nombre max de la ressource i est fausse

Evitement des interblocages => Algorithme du banquier (évalue le risque d'interblocage pouvant être provoqué par une demande de ressource) :

1. Vérifier la cohérence de la requête : $DEMANDE \leq BESOIN$
2. Vérifier la disponibilité des ressources : $DEMANDE \leq DISPONIBILITE$
3. Accepter temporairement la demande et vérifier l'état du système :
 $DISPONIBLE = BESOIN[i] - DEMANDE$
 $ALLOCATION[i] = \text{old.ALLOCATION}[i] + DEMANDE$
 $BESOIN[i] = \text{old.BESOIN}[i] + DEMANDE$
4. Appliquer l'algorithme de détermination d'un état sain : $TRAVAIL = DISPONIBLE$
 Pour tous les P_i , vérifier que $BESOIN[i] \leq TRAVAIL$, si oui $TRAVAIL += ALLOCATION[i]$, si non => système malsain

Détection ds interblocages et reprise : Graphe d'attente si toutes les ressources possèdent une seule instance (on élimine les noeuds de type ressource)

Interblocage SSI le graphe d'attente contient un circuit

Ordonnancement de processus

Critères d'ordonnancement :

- Rendement d'utilisation du CPU : pourcentage de temps pendant lequel le CP est actif => Plus élevé, mieux c'est
- Utilisation globale des ressources : assurer une occupation maximale des ressources de la machine et minimiser le temps d'attente pour l'allocation d'une ressource à un processeur
- Equité : capacité de l'ordonnanceur (module du système d'exploitation qui attribue le contrôle du CPU à tour de rôle aux différents processus en compétition) à allouer le CPU d'une façon équitable à tous les processus de même priorité (éviter la famine)
- Temps de rotation : durée moyenne nécessaire pour qu'un processus termine son execution
- Temps d'attente : durée moyenne qu'un processus passe à attendre le CPU
- Temps de réponse : temps moyen qui s'écoule entre le moment où un utilisateur soumet une requête et celui où il commence à recevoir les réponses
 $TMT = (\text{somme}(\text{date de fin d'exécution} - \text{date d'arrivée du processus})) / \text{nombre de processus}$

Algorithmes d'ordonnancement :

- **FCFS** (First Come First Served) : Les tâches sont ordonnancées dans l'ordre où elles sont venues
- **SJF** (Shortest Job First) : Le CPU est attribué au processus dont le temps d'exécution estimé est minimal
- **SRT** (Shortest Time Remaining) : Le CPU est attribué au processus qui a le plus petit temps d'execution restant (réévaluation à chaque quantum)
- **RR** (Tourniquet) : Le controle du CPu est attribué à chaque processus pendant un quantum

Algorithmes avec priorité : CPU attribué au processus de plus haute priorité

Threads

2 types de retour possible (p. 132)

Gestion de la mémoire

Partitions multiples contiguës :

- Contiguës fixes :
 - files d'attente séparées
 - files d'attente communes
- Contiguës dynamique :
 - First fit (selon les adresses croissantes)
 - Best fit (selon les tailles croissantes)
 - Worst fit (selon les tailles décroissantes)
- Contiguë siamoise

Algorithme de remplacement de pages :

- Algorithme optimal : Lors d'un défaut de page, choisit comme victime une page qui ne fera l'objet d'aucune référence ultérieure ou qui fera l'objet de la référence la plus tardive
- FIFO : Choisit comme victime la page la plus anciennement chargée
- LRU : Choisit comme victime la page ayant fait l'objet de la référence la plus ancienne