

TD 3 SR02 : Gestion des entrées sorties et fichiers

TD Exercice sur une séance de 2h

Exercice 1. (Programmation d'un robot)

Un robot dispose de 2 roues. La circonférence d'une roue est de 10cm. Le contrôleur du robot dispose de deux registres :

- RE : ce registre d'état est mis à 1 quand le robot est prêt.
- RC : quand ce registre de commande est mis à 1, les roues font un tour complet.

Question

Ecrire les procédures nécessaires pour faire avancer le robot de 1m dans deux cas :

- Le processeur ne dispose pas d'un mécanisme d'interruption.
- Le processeur dispose d'un mécanisme d'interruption : à chaque fois que les roues font un tour complet, le contrôleur des roues émet un signal d'interruption au processeur.

On suppose que l'ordinateur exécute plusieurs processus. Lorsqu'un processus désire faire avancer le robot, il passe par un SVC (Appel Système).

N.B. : le système dispose entre autres des primitives suivantes :

- Sauver_contexte(...)
- Restaurer_contexte(...)
- Charger_mep(...) /* charger le mot d'état du processeur */

HYPOTHESES

- On suppose que (RC,RE) sont remis automatiquement à (0,1) à la fin de l'opération d'ES.
- On dispose de primitives enfiler(), defiler(), supprimer(), vide() sur les files d'attente et garantissant l'accès en exclusion mutuelle
- Programme utilisateur

.....

Syscall_rouler(10) ;

.....

Fin.

Exercice 2. (Ordonnancement des requêtes de disques)

Un disque comporte 200 pistes numérotées de 0 à 199. La dernière requête traitée concernait la piste 112 et une requête pour la piste 138 est en cours. La liste des nouvelles requêtes dans l'ordre d'arrivée est la suivante :

91, 165, 67, 158, 43, 132, 28, 106, 84

- Calculer le déplacement total de la tête pour les algorithmes suivants :

a) FIFO b) PCTR c) SCAN d) LOOK

- Pourquoi est-il dangereux d'autoriser le traitement immédiat de nouvelles requêtes ? Quels sont les algorithmes qui ne sont pas sujets aux famines ?

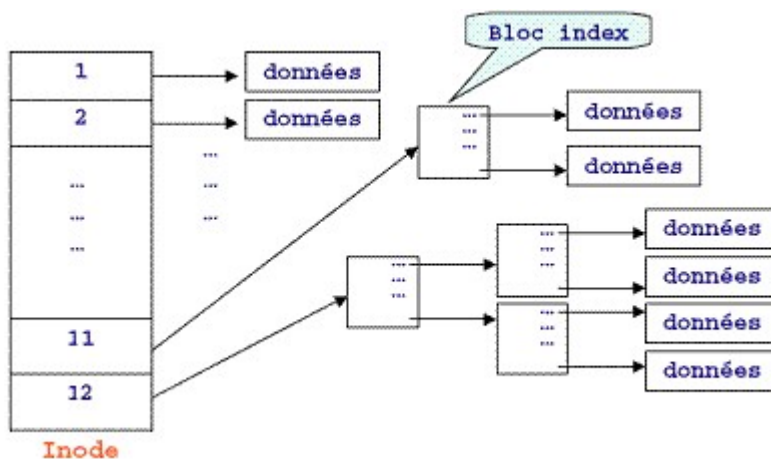
Exercice 3. (Techniques d'accès aux fichiers)

Un fichier occupe 100 blocs. Supposons que le bloc de contrôle de fichier (et le bloc d'index, dans le cas d'une allocation indexée) est déjà en mémoire. Dans le cas de l'allocation contiguë, supposez qu'il n'y a pas de place pour grandir au début, mais qu'il y a de la place pour grandir à la fin. Supposons que les informations de bloc à ajouter sont stockées en mémoire.

- Calculez le nombre d'opérations d'E/S de disque requises pour ajouter un bloc au début du fichier et enlever un bloc à la fin du fichier pour les méthodes d'allocation suivantes : contiguë, chaînée, chaînage indexé.

Exercice 4. (I-node d'UNIX)

On suppose qu'un système UNIX utilise la structure d'i-node illustrée à la figure suivante :



Allocation indexée avec i-nœuds

Chaque bloc d'index peut contenir 256 adresse de blocs.

- Combien d'accès sont-ils nécessaires pour accéder au 583ème bloc d'un fichier qui en compte 896 ?