

2 minutes – Système d'exploitation

Gestion des processus

- 1- Expliquez pourquoi `errno` est problématique dans un système multi-threadé ?
 - `Errno` est défini comme une variable globale externe
 - Son utilisation entraîne des résultats nondéterministes : plusieurs threads peuvent avoir des erreurs et mettre à jour `errno`
- 2- Solution apportée par les UNIX actuels ?
 - Le symbole `errno` est défini dans le fichier d'en-tête `<errno.h>`. Chaque thread associé à un processus donné ne peut pas influencer la valeur d'`errno` d'autres threads.

Ordonnancement

Enoncé : Une liste de processus, avec leur durée de cycle et leur niveau de priorité. Faire un diagramme de Gantt de l'exécution des processus suivant le système FCFS, SJF, avec priorité, et tourniquet (RR).

FCFS: First Come First Serve. Algo sans réquisition, file d'attente FIFO pour les processus prêts.

SJF : Le plus court d'abord. Algo sans réquisition, en cas d'égalité -> FCFS, temps moyen d'attente minimal, difficulté de calculer la longueur des cycles

Ordonnancement avec priorité : Généralisation du SJF, avec ou sans réquisition, priorité interne (consommation des ressources) et priorité externe (utilisateur) pris en compte. Problème : blocage infini (« famine »), solution : technique du vieillissement (augmentation de la priorité des processus en attente)

Tourniquet/RR : principe FCFS avec réquisition sur une base de quantum de temps, nécessite une horloge. Réquisition pour les cycles plus longs que le quantum, commutation passive (FCFS) pour les cycles plus courts

Synchronisation

Exclusion mutuelle : Un seul processus à la fois peut exécuter sa section critique

Algo TaS (Test-and-Set) ressemble à un passage de relais.

Gestion de la mémoire

Pagination

Une mémoire virtuelle est formée de zones de même taille, appelées *pages*.

Une adresse virtuelle est donc un couple (numéro de page, déplacement dans la page). La taille des pages est une puissance de deux, de façon à déterminer sans calcul le déplacement (10 bits de poids faible de l'adresse virtuelle pour des pages de 1 024 mots), et le numéro de page (les autres bits).

Défaut de page : Requête d'une page inexistante et donc création d'une nouvelle page

Fragmentation

Fragmentation externe : espace suffisant pour l'allocation d'un nouveau processus, mais non contiguë

Fragmentation interne : allocation volontaire de zones inoccupées pour diminuer le travail de gestion de la mémoire

Dans un système de pagination, on a de la fragmentation interne : une page entière est allouée à un processus, alors que seuls quelques octets sont occupés.

Mémoire virtuelle

Principe du LRU : Il consiste à choisir comme victime le cadre qui n'a pas été référencé depuis le plus longtemps

Algo à pile : Algo utilisant une structure de donnée LIFO (Last In First Out)

Allocation contigüe

First-Fit : premier trou suffisant. Rapide.

Next-Fit : idem, mais recherche à partir de l'emplacement précédent. Un peu moins bon.

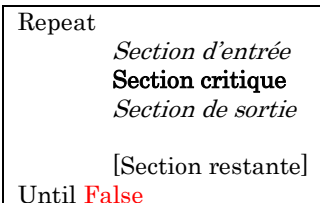
Best-Fit : trou le plus petit possible. Moins performant.

Worst-Fit : trou le plus grand. Bof.

Quick-Fit : maintient de listes par tailles fréquentes. Rapide pour la recherche, lent pour la désallocation

Ordonnancement des disques

- Algorithme d'ordonnancement



FCFS : servir la requête arrivée en premier. Implémentation simple. Intrinsèquement juste. Risque de déplacement « sauvage » des têtes.

SSTF : servir la requête la plus proche de la position actuelle. Basé sur l'écart entre pistes. Analogue au SJF. Risque de famine. Non optimal (contrairement au SJF).

SCAN : balayage avant / arrière perpétuel des pistes. Service des requêtes en fonction de la position des têtes. Aussi appelé « ascenseur » ou « chasse-neige ».

C-SCAN : SCAN circulaire. En fin de disque, retour au début sans traiter aucune requête.

[C-]LOOK : [C-] SCAN avec retour quand plus aucune requête n'existe dans la direction courante. Le plus utilisé.

- Quelques définitions

Temps de positionnement : déplacement des têtes sur la piste (le cylindre) considéré(e).

Temps de latence : passage du bloc sous la tête par rotation des plateaux.

Temps de transfert : temps effectif de transfert des données vers (ou depuis) la mémoire

Vecteur binaire : (Mac OS) représentation de chaque bloc dans l'ordre par un bit d'occupation.

Instructions matérielles facilitant la manipulation (i386, 68030). Efficace à condition que le vecteur en entier soit maintenu en mémoire vive.

Liste chaînée : un pointeur sur le premier bloc libre. Chaque bloc libre pointe sur le suivant. Inefficace en cas de besoin de plus d'un bloc.

Groupement : stockage des adresses de blocs libres dans autant de blocs libres que nécessaire. Efficace grâce au principe de lecture par bloc.

Compactage : en général, plusieurs blocs libres se suivent. Stockage d'une adresse de bloc et du nombre de blocs libres contigus. Liste plus courte que dans les cas précédents

- Type d'allocation

Allocation contigüe

Principe : Fichiers stockés par blocs contigus sur le disque - Temps de positionnement des têtes minimal - Entrée de répertoire : adresse du premier bloc et longueur (en nombre de blocs) - Accès direct et séquentiel faciles à implémenter : il suffit de mémoriser l'adresse du premier bloc - Gestion de l'espace libre : fragmentation externe, compactage etc.

Problème majeur : fichier de taille variable - Trop d'espace : fragmentation interne - Pas assez d'espace : déplacement (coûteux) du fichier. Pas toujours possible

Allocation chaînée

Principe : Fichier = chaîne non contiguë de blocs disque ; Chaque bloc se termine par un pointeur sur le bloc suivant ; Une entrée de répertoire contient un pointeur sur le premier bloc

Avantages : Pas de fragmentation externe - Pas de limite de taille

Inconvénients : Accès direct inefficace ; Fiabilité : perte de pointeur critique. Solutions : listes doublement chaînées, reproduction du nom de fichier et numéro de bloc dans chaque bloc etc. Coûteux dans tous les cas.

Allocation indexée

Principe : Chaque fichier possède un bloc d'index (1^{er} bloc) - Une entrée de répertoire pointe sur le bloc d'index - La i^{ème} entrée du bloc d'index pointe sur le i^{ème} bloc de données du fichier

Avantages : Implémentation efficace de l'accès direct - Table des i-nodes de taille proportionnelle au nombre de fichier

Inconvénients : Fragmentation interne plus grande qu'avec l'allocation chaînée - Problème de la taille des index