Année scolaire 1997-1998 **2ème annnée** Promotion 1999

Contrôle de connaisance.

Sans document.
Répondre directement dans les boîtes.

Question 1) Qu'est ce qu'un micro-noyau et en quoi le noyau UNIX n'est pas un un micro-noyau ?

Question 2) C'est quoi le contexte matériel d'un processus. Quelle information contient-il typiquement ?

Question 3) Pourquoi l'appel d'une fonction interface d'un noyau (Unix ou micro-noyau) n'est pas un appel normal de fonctions. Qu'est qui doit être fait au moment de l'appel d'une fonction interface de noyau?

28/02/01

Question 4)

On se place dans le cas du noyau Unix, et non d'un micro-noyau. En Unix, chaque processus possède deux structures de données différentes contenant des informations sur le processus, information gérée par le noyau Unix. Dans la structure, dite u, on trouve les informations dont le noyau a besoin que lorsque le processus est actif (le noyau exécute alors un service pour le compte d'un processus). Dans la structure de donnée proc, on trouve des informations que le noyau a besoin en permanence sur le processus (par exemple lorsqu'il traite une interruption matérielle. Donnez des exemples d'informations qui doivent se trouver dans l'une ou l'autre des structures pour les fonctionnalités suivante d'un noyau Unix :

	Structure u	Structure proc
gestion des fichiers		
gestion de la mémoire, mémoire virtuelle		
gestion des entrées-sorties et drivers.		
scheduling (ordonnancement des processus)		

Question 5) Réalisation d'une fonction de création de nb processus.

Soit la fonction void parallele (int nb, void * table_de_fonctions []) fournie par un noyau.

Elle permet de lancer l'exécution en parallèle (et sans notions de priorité entre les différentes fonctions) les nb fonctions passées en paramètre dans table_fonction. Ces fonctions sont sans paramètre et sans valeur retournées. Parallèle se termine quand toutes les fonctions ont terminé leur exécution. C'est comme si la fonctionnalité wait d'Unix était incluse dans la fonctionnalité du fork.

On vous demander de réaliser la fonction parallèle dans le cadre d'un noyau monoprocesseur.

A vous de définir les fonctions de base dont vous avez besoin. Il suffit d'en donner une description en français. A vous de définir les informations que le noyau a besoin sur chaque processus, et les données globales (listes de processus). Exemple de fonction de base :

void Sauvegarder_ctxt (ctxt *adr). /* Copie le contexte matériel "courant" vers l'adresse adr"/

void masquer-les-interruptions ( );

Données globales :

Données pour chaque processus

Fonctions de bases :

Réalisation en français de la fonction parallèle:

Question 6) Donnez les avantages et inconvénients respectifs des liens symboliques et des liens durs.

Question 7) Le protocole NFS (Network File System) est dit sans état (côté serveur). Il existe des systèmes (exemple, RFS - Remote File system) qui sont au contraire avec état côté serveur. Dans le cas de RFS, un circuit virtuel est établi à l'opération de montage mount. Quels avantages ou inconvénients, vous voyez à l'une ou l'autre méthode?

Questions 8) Indiquez si les assertions suivantes sont vraies ou fausses :

Un driver en mode caractère permet à processus de ne lire or d'écrire que caractère, par caractère.	
Un driver en mode caractère peut être associé à un disque.	
Un lien dur permet d'un répertoire sur un disque A d'accéder à un fichier sur un disque différent (B)	

Question 9) Quel est le rôle de la commande système kill sur Unix (Dire que cela sert à tuer n'est pas suffisant)?

Question 10) Quel est le rôle de la fonction mknod?

Question 11) Pourquoi les systèmes UNIX modernes ont introduit en plus des processus (création par fork) la notion de threads?