

Systèmes Avancés EXAM 2009/2010

1) - Différences entre distribution linux et noyau Linux :

Le noyau est le coeur d'une distribution qui comprend entre autres les utilitaires utilisateurs et les commandes d'administration

Une distribution est un ensemble de logiciels nécessaires pour une fonctionnalité complète et du noyau. Elle peut être configurée pour un type d'utilisation particulier (bureautique, serveur, développement). Une distribution est donc un ensemble cohérent de logiciels (la plupart libre) assemblés autour du noyau (Linux).

Aperté : fonctions d'un noyau Linux : ordonnanceur (ordre, chargement et exécution des processus), gestionnaire de mémoire (allocation de mémoire aux différents processus), appels systèmes, gestion du matériel (gestions des différents pilotes).

Certains noyaux offrent des fonctionnalités telles que : gestion des systèmes de fichiers, supports réseaux, etc...

2) Voir TP3 Systèmes Avancés ou TP7 Info Embarquée

3) **ABI : Application Binary Interface.** Une ABI décrit une interface bas niveau entre les applications et le système d'exploitation , entre une application et une bibliothèque, ou bien entre différentes parties d'une application. Ex : l'ABI d'AMD

API : Application Programming Interface. Du point de vue technique, une API est un ensemble de fonctions, procédures ou classes mises à disposition par une bibliothèque logicielle, un système d'exploitation ou un service. La connaissance des API est indispensable à l'interopérabilité entre les composants logiciels. Exemple : IPC, libc, SAX (Simple API for XML), DOM (Document Object Model), JDBC.

Une ABI diffère d'une API puisque une API définit une interface entre du code source et une bibliothèque, de façon à assurer que le code source compilera sur tout système supportant cette API. Une ABI doit permettre la portabilité des applications binaires entre S.E. fournissant la même API sur des plateformes matérielles compatibles (même ISA utilisateur => même famille de processeur), puis exercice 1.7 poly Introduction (poly 1), pages 53 et suivantes du poly 1

4) Un module Linux est un morceau de code permettant d'ajouter des fonctionnalités au noyau : pilotes de périphériques matériels, protocoles réseaux, etc... Il peut être chargé dynamiquement sans avoir besoin de recompiler le noyau (commande *insmod* ou *modprobe*) ou de redémarrer le système.

Qui peut manipuler les modules? : le super-utilisateur (*root*)

Comment se construit un module Linux? : à revoir!!

5) structure de données utilisée pour représenter un processus : PCB (Process Control Block) : bloc de contrôle du processus.

Une thread est représentée dans le PCB d'un processus, l'est-t-il tout seul??

Comment différencie-t-on une thread d'un processus? : deux processus sont typiquement indépendants et peuvent interagir uniquement à travers une API fournie par le système telle que IPC. D'un autre côté les *threads* partagent une information sur l'état du processus, des zones de mémoires ainsi que d'autres ressources. Puisqu'il n'y a pas de changement de mémoire virtuelle, la commutation de contexte (*context switch*) entre deux *threads* est moins coûteuse que la commutation de contexte entre deux processus

Fonction bibliothèque : `pthread_create(thread, attribut, routine, argument de la routine)`
`clone()`

6) Ordonnement des processus.

Classes d'ordonnement supportées par un noyau linux : page 35 poly 2 (Linux_Process)
Nice permet d'exécuter un processus avec une priorité d'ordonnement modifiée.

7) Rôle MMU : L'utilisation la plus courante et connue d'une MMU est la protection de plages mémoire. Un programme donné ne doit pas pouvoir accéder (en lecture et/ou écriture) à la mémoire utilisée par un autre programme, voire par le [système d'exploitation](#) lui même.

D'une manière simple, chaque programme exécuté par le système d'exploitation se voit attribuer une zone mémoire protégée, dans laquelle aucun autre programme ne peut écrire; translation d'adresses logiques en adresses linéaires par l'unité de segmentation; traduction d'adresses linéaires en adresses physiques par l'unité de pagination, arbitrage de bus, etc...

La MMU est un composant matériel.

Slide 21, poly 3, Linux_Mémoire

8)