

Série d'exercices 1 : Concepts généraux

- 1.1 Quelles sont les fonctions principales d'un système d'exploitation? Les interpréteurs de commandes et les compilateurs font-ils parties du système d'exploitation?
- 1.2 Dans le système Linux, les appels système sont généralement effectués à partir de quelle couche: programme utilisateur, commande shell, procédure de la bibliothèque standard?
- 1.3 Quelle est l'utilité de l'accès direct à la mémoire (en anglais Direct Memory Access DMA)? En supposant qu'une unité centrale de traitement serait capable de transferts à la même vitesse que l'unité DMA, serait-il néanmoins utile et souhaitable que le système d'exploitation utilise l'unité DMA?
- 1.4 Les appels systèmes sont-ils exécutés en mode superviseur (noyau) ou en mode utilisateur?
- 1.5 Il existe maintenant différents mécanismes pour interagir avec un ordinateur, qui présentent chacun une charge différente sur le système. Estimez la quantité de mémoire nécessaire pour mémoriser l'état d'un terminal alphanumérique (24 lignes de 80 caractères). Comment cela se compare-t-il à la mémoire requise pour l'état d'un terminal graphique (carte graphique, écran HD 1920x1080, clavier, souris)?
- 1.6 Comment sont organisés les fichiers dans le système Linux? Un utilisateur peut-il accéder à un fichier d'un autre utilisateur? Si oui, comment?
- 1.7 Pour lancer en parallèle plusieurs traitements d'une même application, vous avez le choix entre les appels système `fork()` et `pthread_create()`. Laquelle des deux possibilités choisir? Pourquoi?
- 1.8 Citez quatre événements qui provoquent un changement de contexte (invocation de l'ordonnanceur) d'un processus en cours d'exécution dans le système Linux.
- 1.9 Lesquelles des opérations suivantes devraient-elles être permises seulement en mode système: a) désactiver les interruptions, b) lire l'horloge, c) changer l'heure de l'horloge, d) changer la table de pages?
- 1.10 Quel est le rôle de l'ordonnanceur? Comment peut-il favoriser les processus interactifs?

- 1.11 Un système avec mémoire virtuelle contient de la mémoire cache, de 2ns de temps d'accès, de la mémoire vive, de 10ns de temps de pénalité d'accès (i.e. 10ns qui s'ajoutent aux 2ns d'essai d'accès en cache), et un disque, de 10ms de temps de pénalité d'accès. En moyenne, 95% des accès peuvent se faire en mémoire cache et, pour les accès qui ne peuvent être servis en cache, 99% d'entre eux peuvent se faire en mémoire vive. Si l'accès n'est pas en mémoire, alors le système d'exploitation doit faire un accès disque. Quel est le temps d'accès moyen à la mémoire dans cette situation?
- 1.12 Dans quelles conditions le partage de données pose-t-il des problèmes dans un programme ayant plusieurs fils d'exécution?
- 1.13 Expliquez pourquoi les processeurs ont deux modes de fonctionnement (noyau et utilisateur).
- 1.14 Lors d'un appel système pour la lecture, le processus est normalement bloqué jusqu'à ce que la donnée soit disponible. Qu'en est-il pour les écritures, le processus est-il nécessairement bloqué jusqu'à ce que l'accès disque soit complété?
- 1.15 Pour le programmeur, est-il utile de savoir si une fonction de la librairie standard résulte ou non en un appel système?
- 1.16 Quelle est la différence entre une interruption logicielle (aka trap, déroutement) et une interruption matérielle?