

CSI3531 – Systèmes d'exploitation

Tutorat 1 - Hiver 2016 - Solutionnaire

1. Quels sont les trois rôles principaux du système d'exploitation?
 - **Abstraction du matériel :** Pour offrir un environnement dans lequel un utilisateur d'ordinateur puisse exécuter des programmes dans divers matériel facilement et de façon efficace.
 - **Allocation de ressources :** Pour faire l'allocation des diverses ressources de l'ordinateur selon le besoin des programmes. L'allocation devrait être juste et efficace le plus possible.
 - **Programme de contrôle :** Un programme de contrôle a deux fonctions principales : (1) superviser l'exécution des programmes d'utilisateurs pour prévenir des erreurs et la mauvaise utilisation de l'ordinateur; et (2) la gestion de l'opération et le contrôle des appareils E/S.
2. Considérez les définitions diverses du *système d'exploitation*. Considérez la question d'inclure dans le système d'exploitation des applications telles que fureteurs WEB ou programmes de courriel. Débattre le pour et le contre de cette question avec justification.

Pour: Les applications telles que fureteurs WEB et outils de courriel jouent un rôle de plus en plus important lors de l'utilisation du PC moderne. Pour répondre à ces besoins, ces applications pourraient être incorporées dans le système d'exploitation. Ainsi, on pourrait atteindre une meilleure performance et une meilleure intégration avec le reste du système. En plus, ces applications importantes pour avoir le même aspect et convivialité (look and feel) du logiciel de système du SE.

Contre: Le rôle fondamental du système d'exploitation est de gérer les ressources du système tel que l'UCT, la mémoire, les appareils d'E/S, etc. En plus, son rôle est de rouler des applications logicielles telles que les fureteurs WEB et applications courriels. En incorporant de telles applications dans le SE, on l'alourdit avec ces fonctions additionnelles. Ce fardeau additionnel peut donner un SE qui accomplit moins bien sa tâche de gestion des ressources du système. En plus, on accroît la grandeur du SE, et ainsi accroît la possibilité que le SE plante et les bris de sécurité.
3. Comment la distinction entre le mode noyau et le mode utilisateur sert à la protection (sécurité) rudimentaire du système?

La distinction entre le mode noyau et le mode utilisateur offre une forme de protection rudimentaire de la façon suivante. Certaines instructions ne peuvent être exécutées que lorsque l'UCT est en mode noyau. Aussi, les appareils ne peuvent être accédés par un programme que lorsqu'il exécute en mode noyau (i.e. à l'intérieur d'un appel système). Le contrôle d'activation et de désactivation des interruptions se fait seulement que lorsque l'UCT est en mode noyau. Donc, l'UCT est assez limité quand il exécute en mode utilisateur, et ainsi protège les ressources critiques durant l'exécution du code utilisateur.
4. Lesquelles des instructions devront être privilégiées?
 - a. **Mise à jour de minuterie.**
 - b. Lire l'horloge.
 - c. **Effacer la mémoire.**
 - d. Exécuter une interruption logicielle (ceci est une instruction de l'UCT).
 - e. **Désactiver les interruptions.**
 - f. **Modifier les entrées dans un tableau des états d'appareils.**
 - g. Commuter entre le mode usager et le mode noyau.
 - h. **Accéder un appareil E/S.**
5. Une minuterie peut servir pour déterminer le temps courant. Donnez une courte description comment accomplir cette tâche.

Un programme peut utiliser l'approche suivante pour déterminer le temps courant avec les interruptions de minuterie. Le programme configure la minuterie afin de générer une interruption après un certain temps spécifique et ensuite être suspendu (s'endormir). Lorsqu'activé par l'interruption, il peut mettre à jour une variable dont sa valeur reflète le nombre d'interruption reçu à date. Ces étapes sont répétées continuellement, i.e., configurer l'interruption et mettre à jour la variable lorsque l'interruption est produite. La variable peut ainsi ternir compte du temps, particulièrement s'il est initialisé à une valeur significative. Les SEs UNIX utilise une telle approche en comptant le nombre de secondes depuis le 1 jan, 1970 avec un compteur de 32 bits (débordement aura lieu le 19 jan, 2038).

6. Quelle est le but des appels systèmes?

Les appels systèmes permettent aux processus utilisateurs de faire des demandes de services au système d'exploitation. Notez que les appels systèmes sont réalisés avec des interruptions logicielles. Prenez conscience que le SE fonctionne à base d'interruptions et qu'il peut recevoir une interruption du matériel ET du logiciel (i.e. programmes utilisateurs) comme demande de compléter une action quelconque.

7. Quelles sont cinq activités principales du système d'exploitation lors de la gestion de processus?

- a. La création et terminaison des processus utilisateurs et système.
- b. La suspension et reprise des processus.
- c. La provision de mécanismes pour la synchronisation de processus.
- d. La provision de mécanismes pour la communication entre processus.
- e. La provision de mécanismes pour composer avec l'impasse.

8. Quelles sont trois activités principales du système d'exploitation lors de la gestion de mémoire?

- a. Tenir compte de quelles parties de la mémoire sont utilisées et par quels processus.
- b. Décider quels processus sont chargés en mémoire lorsque de l'espace de mémoire devient disponible.
- c. L'allocation et la libération de l'espace mémoire selon le besoin.

9. Quelles sont trois activités principales du système d'exploitation lors de la gestion de la mémoire secondaire (secondary storage memory)?

- a. La gestion de l'espace libre.
- b. L'allocation de mémoire secondaire.
- c. L'ordonnancement de disque.

10. Quel est le but de l'interpréteur de commande (CLI)?

Il lit des commandes de l'utilisateur ou d'un fichier de commandes et les exécute soit directement ou en lançant un autre processus pour exécuter un programme séparé.

11. Quels appels systèmes sont exécutés par l'interpréteur de commande ou shell pour démarrer un nouveau processus?

Dans le système UNIX, l'appel système *fork* suivi de l'appel *exec* doit être fait pour lancer un nouveau processus. L'appel *fork* clone le processus qui fait l'appel, tandis que l'appel *exec* remplace le programme dans le processus avec un nouveau programme. Ceci est un concept difficile – ce qui faut tenir compte c'est que *fork* est exécuté par le processus original (parent), tandis que *exec* est exécuté par le processus lancé (enfant). Nous étudierons ces appels cette semaine.

12. Quelle est le but des programmes systèmes?

Les programmes systèmes peuvent être vus comme des logiciels d'appels systèmes. Il offre des fonctions de base aux utilisateurs pour éviter de les obliger d'écrire des programmes pour résoudre des problèmes communs. Les programmes systèmes communs comprennent des utilitaires pour manipuler des fichiers et répertoires, des interpréteurs de commandes pour exécuter des programmes (y compris des utilitaires systèmes), des utilitaires pour surveiller la base d'utilisateur et le réseau (e.g. UNIX *who*, *rwho*, montre quels utilisateurs ont une session ouverte dans le système local et les systèmes distants), la gestion de processus, etc.

13. Quelle est l'avantage principal de concevoir un système avec des couches? Quelles sont les désavantages de cette approche?

Comme dans tous les cas de conception modulaire, la conception d'un système d'exploitation avec une approche modulaire comprend plusieurs avantages. Le système est plus facile à déboguer et à modifier puisque les changements affectent seulement des parties limitées au lieu d'affecter toutes sections du SE. Les données sont entretenues seulement où nécessaire et accessibles à partir de régions bien définies et limitées, ce qui veut dire que les bugs affectant des données sont limités à un module ou couche spécifique.

14. Pour chacune des cinq services suivants offerts par un système d'exploitation, expliquez comment chacun d'eux est pratique pour l'utilisateur. Expliquez dans quel cas il serait impossible d'offrir ces services avec des programmes usagers.
- **L'exécution de programme.** Le système d'exploitation charge le contenu (ou parties du contenu) d'un fichier exécutable en mémoire primaire et lance son exécution. Il alloue les ressources au programme durant son exécution (e.g. UCT). Il n'est pas possible d'avoir confiance en un programme utilisateur pour faire une bonne allocation de ressources tel que l'UCT.
 - **Les opérations E/S.** La communication avec les disques, bandes magnétiques, lignes de communication série, et autres appareils doit se faire à un très bas niveau. Le programme utilisateur ne fait que spécifier l'appareil (en principe comme un fichier) et les opérations à faire avec l'appareil; et le SE convertit ces demandes en instructions spécifiques au contrôleur de l'appareil. Il n'est pas possible de faire confiance à un programme utilisateur d'accéder les appareils auxquels ils ont droit, et aux moments qu'ils sont libres.
 - **La manipulation du système de fichier.** Il existe beaucoup de détails pour la création de fichiers, suppression de fichiers, allocation d'espace pour les fichiers, et la définition des noms de fichiers. Le disque est organisé en blocs de données qui doivent être gérés. La protection doit assurer l'accès approprié des fichiers. Il est impossible de faire confiance aux programmes utilisateurs d'adhérer au règles de protections; de faire l'allocation de seulement de blocs libres à la création et mise à jour des fichiers; et de libérer les blocs lors de la suppression de fichier.
 - **Les communications.** L'échange de messages entre systèmes (et processus) nécessite que les messages soient transformés en paquets de données qui sont envoyés à un contrôleur de communication, transmis via un support de transmission, et réassemblés dans le système distant. La mise en ordre et la correction de paquets doit aussi se faire. Encore, un programme utilisateur ne pourrait pas faire un bon accès du contrôleur de l'appareil de communication, ou pourrait prendre des paquets destinés à d'autres processus.
 - **La détection d'erreur.** La détection d'erreurs se fait au niveau matériel et au niveau logique. Au niveau matériel, tout transfert de données doit être inspecté pour assurer que les données n'ont pas été corrompues durant le transit. Toute donnée sur support doit être vérifiée afin d'assurer qu'elle n'a pas changé depuis son inscription sur le support. Au niveau logique, le support doit être vérifié pour la détérioration de données; par exemple, assurer que le nombre de blocs alloués et de blocs libres de la mémoire secondaire correspond au nombre total de blocs de l'appareil. Les erreurs dans ces cas sont souvent indépendantes des processus spécifiques (par exemple, la corruption de données sur un disque), et donc un programme global (le SE) doit composer avec ce type d'erreur. Aussi, en ayant le SE composer avec ce type de problème, il n'est pas nécessaire d'inclure dans des programmes utilisateurs du code pour composer avec toutes les erreurs possibles dans le système.

En plus des raisons données ci-dessus contre avoir les programmes utilisateurs composer directement avec ces services, imaginez le travail à créer même un programme simple si le programmeur devrait composer avec toutes les questions discutées ci-dessus.