

Systeme d'exploitation

Examen – 26 mai 2014

2h

*Le barème est donné à titre indicatif. Les différentes parties sont totalement indépendantes.
L'examen est volontairement trop long, passez à une autre partie si vous êtes bloqués.*

I. Questions d'échauffement

11 points

Les sous-questions suivantes sont indépendantes.

- a) Quelle est la différence entre l'administrateur (root) et le noyau ? Quels privilèges ont-ils ? Qui met en place ces privilèges ? Et si on rajoute une couche de virtualisation ? 3 pts
- b) Que pensez-vous de la phrase suivante ?
« L'ordonnanceur est un processus qui se réveille régulièrement et change les processus en cours d'exécution sur les différents processeurs. » 2 pts
- c) Dans quel cas (événement et attributs de la page) le système doit-il remplir une page de zéros pour éviter une fuite d'information entre deux processus ? Vaut-il mieux faire le remplissage le plus tôt ou le plus tard possible ? 3 pts
- d) Comment implémenter un logiciel « preload » qui fait en sorte que les programmes les plus couramment utilisés soient chargés en mémoire (mais pas lancés) au démarrage de la machine (sans provoquer de swap gênant) ? 3 pts

II. Table de pages

9 points

On dispose d'une architecture 64bits avec 5 niveaux de table de pages matériels et des pages de 2ko (2^{11} octets). La table est constituée de tableaux de la taille d'une page. Ces tableaux peuvent être remplis avec des pointeurs vers les sous-niveaux (pointeurs de taille 4 octets) ou avec des PTE (de taille 8 octets).

Pour les questions suivantes, on pourra arrondir les calculs.

Rappel: $1\text{ k} \approx 10^3 \approx 2^{10}$ $1\text{ M} \approx 10^6 \approx 2^{20}$ $1\text{ G} \approx 10^9 \approx 2^{30}$ $1\text{ T} \approx 10^{12} \approx 2^{40}$ $1\text{ P} \approx 10^{15} \approx 2^{50}$

- a) Rappelez rapidement ce qui est stocké dans chaque niveau de la table de pages. Pourquoi utilise-t-on en général des tableaux de la taille d'une page ? 1 pt
- b) Combien de pointeurs ou PTE peut-on mettre par page ? En déduire le nombre de bits utilisés pour chaque niveau de la table de pages. Combien de bits sont nécessaires pour décrire un décalage dans une page ? 2 pts
- c) En déduire la taille maximale des adresses virtuelles manipulées sur cette architecture. Comment relier ce résultat au fait que l'architecture est présumée « 64bits » ? Que fait-on des bits restants ? 2 pts
- d) Quel espace mémoire est nécessaire pour stocker la table des pages d'un processus qui utilise l'intégralité de son espace d'adressage ? Et s'il n'utilise que 100 octets ? Et s'il utilise 64Mo contigus ? Comparez avec le cas d'une table linéaire (non-hiérarchique à 1 seul niveau). 4 pts

III. Pagination à la demande

11 points

On considère une architecture où la taille des pages est 4ko. La mémoire physique sera considérée comme infinie.

a) On lance un programme qui alloue 400ko (mapping public anonyme) puis les remplit de zéros. Expliquez pourquoi le lancement de notre programme provoque environ 100 défauts de page. Expliquez ensuite pourquoi ces défauts de page disparaissent si le programme alloue la mémoire mais ne l'utilise pas. 2 pts

b) On remarque ensuite que le lancement du processus, même sans allouer de la mémoire ou la modifier, provoque tout de même quelques défauts de page. A quoi sont-ils dûs ? Que peut-on observer si on lance plusieurs fois le processus d'affilée peu après le démarrage de la machine ? 2 pts

c) On modifie maintenant notre programme pour créer 2 fils par *fork* après avoir alloué les 400ko de mémoire et les avoir remplis de zéros. Les 2 fils remplissent alors les 400ko avec d'autres valeurs. Quand les 2 fils ont terminé, le père remet la mémoire à zéro. Qu'observe-t-on en terme de défauts de page ? 3 pts

d) Le mapping est maintenant privé et non plus public. Que devient votre réponse à la question précédente ? Expliquez ce qui se passe et l'intérêt du modèle. Explicitez les différents états des pages (protection matérielle, compteur de références, ...) et les défauts de pages observés par les trois processus au fur et à mesure de leur exécution. 4 pts

IV. Entrées-sorties logiques ou physiques ou les deux

9 points

a) Donnez un exemple d'entrée-sortie logique au niveau de l'application qui n'implique aucune entrée-sortie physique ? 1 pt

b) Donnez un exemple d'entrée-sortie physique qui n'est pas demandée par une entrée-sortie logique dans une application ? 1 pt

On s'intéresse maintenant plus particulièrement à l'émission de paquets réseau.

c) Combien d'entrées-sorties logiques vont correspondre à combien d'entrées-sorties physiques ? De quel(s) paramètre(s) cela dépend-il ? 2 pts

d) Quel(s) avantage(s) et inconvénient(s) y a-t-il à faire une copie intermédiaire dans le noyau pendant une émission ? 2 pts

e) Que se passerait-il si la zone mémoire où se trouvent les données à envoyer dans l'application se faisait swapper pendant l'émission ? Comment le système peut-il éviter ce problème ? On distinguera les cas où une copie intermédiaire est utilisée ou non. 3 pts