

Questions d'examen

Session 2014-15

Chapitre 1

Préalables

Questions

1. Une question théorique
2. Un exercice pratique complet (.tar - make - SOURCES/make)

Ressources pour le question pratique

- Que le P.C. local (Mandriva ou Opensuse)
- Pas de liens avec linux2 ni Linux4
- Pas de notes
- Juste bic et crayon
- Les fichiers .tar donnés au labo pendant l'année

Modalités

- Examen s'étend sur maximum 3h - groupes de max 6 élèves un horaire de passage sera affiché.
- 10' - mise en place : L'étudiant tire une carte au hasard. Cette carte correspond à une feuille d'examen contenant les questions. L'étudiant reçoit cette feuille, le numéro de son P.C., 1 feuille de papier ministre et des feuilles de brouillon.
- maximum 40' : question théorie avec machine éteinte. Je ramasse la feuille papier ministre avec les réponses. Ce qui n'est pas sur la feuille de réponse est considéré non connu.
- 50+10' : interview de 10 minutes avec l'étudiant pour vérifier éventuellement la compréhension de la question 1. En parallèle avec la question pratique sur machine. Au moment déterminé par l'horaire, je récupère les fichiers SujetGroupeNom.tar qui doivent se trouver dans le répertoire /home/user0/Desktop du P.C. (les sujets étant FS, Process, IPC et Memoire)
- Interview de 10 minutes pour la partie pratique, les autres étudiants préparent leur Interview en attendant.
 - Normalisé et respect de l'énoncé
 - Texte structuré avec commentaires valables apparaît avec make
 - Démo fonctionnelle logique et complète apparaît avec cd SOURCES ;make.
 - Sources, script, makefiles corrects.
 - + Vérifier la compréhension

Avant de commencer

- Un jeu de cartes
- Un jeu de questions
- 24 feuilles papier ministre
- Les feuilles de présence et l'horaire de passage
- Le carnet de présence
- Les PC avec les .tar du labo sur le bureau
- Une feuille de cotes de présence
- Le PC 17 affiche xclock.

Respect des contraintes

- Q1 40', Q2 50'
- Q1 - Une question théorique sur un des chapitres
- Q2 - Une question "technique" sur un des chapitres

1.1 Liste des questions donnée aux étudiants

1.1.1 F.S.

- FSa : Décrivez la structure d'une partition formatée en F.A.T. Détaillez comment le S.E. retrouve un fichier, ajoute des données à ce fichier, efface ce fichier.
- FSb : Décrivez la structure d'une partition formatée en F.A.T. Détaillez les situations d'incohérence et montrer comment le S.E. récupère cette situation.
- FSc : Décrivez la structure d'une partition formatée en ext2. Détaillez comment le S.E. retrouve un fichier, ajoute des données à ce fichier, efface ce fichier.
- FSD : Décrivez la structure d'une partition formatée en ext2. Détaillez les situations d'incohérence et montrer comment le S.E. récupère cette situation.
- FSe : Décrivez la structure d'une partition formatée en ext2. Détaillez le contenu d'un S.B. et l'utilité des champs qui s'y trouvent
- FSf : Décrivez la structure d'une partition formatée en ext2. Détaillez le contenu d'un inode et l'utilité des champs qui s'y trouvent
- FSg : Décrivez la structure d'une partition formatée en ext2. Détaillez les appels système qui permettent de l'utiliser (open, read, write, close, dir, dup, lseek, stat, ...) et comment le S.E. implémente ces appels système (handle, TDFO).
- FSh : Décrivez la structure d'une partition formatée en ext2. Détaillez comment le S.E. mémorise les liens à l'aide d'exemple. (hard, soft)
- FSi : Décrivez la structure d'une partition formatée en ext2. Détaillez la notion de fichier creux à l'aide d'un exemple (création, taille, occupation du disque)
- FSj : Décrivez la structure d'une partition formatée en ext et une autre en ext2. Détaillez les avantages de ext2 et l'implémentation de ces avantages. Comment ext2 a-t-il évolué ensuite?
- FSk : Détaillez comment le S.E. permet de découper un disque en plusieurs partitions. (primaire, logique, étendue) Quelles sont les limites de cette technique? Comment évolue-t-elle? Quels sont les outils utilisés pour gérer ces partitions (mkfs, fdisk, mount, umount, df, /dev, ...)

1.1.2 Processus

- PRa : Un processus effectue un `fork()`. Quel est le rôle de la table des interruptions, de la table des processus, de l'ordonnanceur, du système d'exploitation dans ce cas ? Quelle est la conséquence pour une variable `x` définie avant le `fork()` ? Que vaut `&x` ?
- PRb : Un processus effectue un `fork()`. Quel est le rôle de la table des interruptions, de la table des processus, de l'ordonnanceur, du système d'exploitation dans ce cas ? Quelle est la conséquence pour une lecture dans un fichier qui a été ouvert avant le `fork()` ?
- PRc : Un processus effectue un `fork()` dans une boucle. Quel est le rôle de la table des interruptions, de la table des processus, de l'ordonnanceur, du système d'exploitation dans ce cas ? Que se passe-t-il si la boucle est un `for(i=0;i<3;i++)` ?
- PRd : `wait()` ; `wait4()` : Quelle est l'utilité ? Quels sont les arguments ? Quelle est la valeur de retour ? Définir un zombie, son utilité, les problèmes qu'il génère, comment en créer, comment les détruire.
- PRE : `execve()` ; `execl` ; `execv` ; `execvp` : Quelle est l'utilité ? Quels sont les arguments ? Quelle est la valeur de retour ? Quel est le rôle de la table des interruptions, de la table des processus, de l'ordonnanceur, du système d'exploitation dans ce cas ?

1.1.3 I.P.C.

- IPCa : Expliquez le mécanisme du producteur-consommateur. Détaillez-en le principe, le code, les appels système liés.
- IPCb : Expliquez la réalisation d'une section critique via "variable partagée", "blocage des interruptions" et via "sémaphores de Dijkstra". Détaillez les appels système Down et Up. Comparez ces trois approches.
- IPCc : Expliquez la réalisation d'une section critique via "BTS", "alternance" et via "sémaphores de Dijkstra". Détaillez les appels système Down et Up. Comparez ces trois approches.
- IPCd : semget(), semctl(), semop() : Quelle est l'utilité? Quels sont les arguments? Quelle est la valeur de retour? Établissez le lien entre ces appels système et ceux vus en théorie (up() et down()).
- IPCe : `ls | wc -l`. Comment le S.E. parvient à exécuter cette ligne de commande? Expliquez en détail le mécanisme sous-jacent. De façon très détaillée, expliquez comment il parvient à synchroniser wc et ls afin que wc ait toujours des données à lire.
- IPCf : Quels appels système permettent de gérer les signaux? Détaillez-en les paramètres et le fonctionnement. Quelles sont les limites et les défauts de ces signaux? Quel est le rôle de la table des interruptions, de la table des processus, de l'ordonnanceur dans ces cas?
- IPCg : socket();bind();listen();accept();connect() : Quelle est l'utilité? Quels sont les arguments? Quelle est la valeur de retour? Quelle est l'utilité de la fonction htons()? Comment faire parvenir un message à un processus qui s'exécute sur un autre ordinateur?
- IPCh : socket();bind();listen();accept();connect() : Quelle est l'utilité? Quels sont les arguments? Quelle est la valeur de retour? Comment écrire une application client/serveur où plusieurs clients peuvent être connectés au serveur en même temps?
- IPCi : Utilisation de l'appel système pipe et situations d'interblocage. Expliquez en vous basant sur des exemples de code comment une telle situation peut être obtenue. Détaillez et faites le lien avec les appels système Up et Down.

1.1.4 Mémoire

- MEMa : Décrivez en détail le principe et l'utilité de la segmentation. Comment est-il mis en oeuvre lors du fonctionnement en mode réel et protégé du processeur ? Comment la notion de 'ring' est elle exploitée avec cette segmentation en mode protégé ?
- MEMb : Décrivez en détail le principe et l'utilité de la segmentation. Comment est-il mis en oeuvre lors du fonctionnement en mode réel et protégé du processeur ? Donnez un exemple d'instruction qui peut provoquer une erreur de segmentation.
- MEMc : Décrivez en détail le principe et l'utilité de la segmentation. Comment est-il mis en oeuvre lors du fonctionnement en mode réel et protégé du processeur ? Comment ne pas perdre trop de temps en raison des plus nombreux accès à la mémoire ?
- MEMd : Décrivez en détail le principe et l'utilité de la pagination. Détaillez comment le processeur 32 bits a implémenté cette pagination.
- MEMe : Décrivez en détail le principe et l'utilité de la pagination. Quel est le rôle des interruptions dans cette pagination ?
- MEMf : Décrivez en détail le principe et l'utilité de la pagination. Comment ne pas perdre trop de temps en raison des plus nombreux accès à la mémoire ?
- MEMg : Décrivez en détail le principe et l'utilité de la pagination. Comment ne pas perdre trop de place en mémoire en raison de la présence de la table des pages ?
- MEMh : Décrivez en détail le principe et l'utilité de la pagination. Que faire si la mémoire vive de l'ordinateur est saturée ?