

Systèmes d'Exploitation – Examen 2013-2014 - CORRECTION

IUT Aix-Marseille / DUT R&T 1^{ière} année

J. Seinturier, Y. Lufimpu-Luviya

Durée : 2h – Le barème des points est donné à titre indicatif.

1. Codage binaire (1,5 pts)

1.1. En prenant comme codage pour l'alphabet le numéro des lettres, coder en binaire sur des mots 8 bits:

« systeme d exploitation » (Le caractère d'espace est codé par le nombre 255). (1 pt)

Table des caractères :

Lettre	Index	Codage
A	1	00000001
D	4	00000100
E	5	00000101
I	9	00001001
L	12	00001100
M	13	00001101
N	14	00001110
O	15	00001111
P	16	00010000
S	19	00010011
T	20	00010100
X	24	00011000
Y	25	00011001
	255	11111111

Texte codé:

00010011 00011001 00010011 00010100 00000101 00001101 00000101

11111111 00000100 11111111

00000101 00011000 00010000 00001100 00001111 00001001 00010100 00000001
00010100 00001001 00001111 00001110

1.2. Quelle est la taille du fichier créé en octet, kilooctet, bits, kilobits ? (0.5 pt)

Le texte compte 22 caractères. 1 caractère étant codé sur 8 bits, la taille du fichier est de 176 bits, soit 0.176 kilobits. La taille du fichier en octet est de 22, soit 0,0214 ko.

2. Descripteurs de fichiers (1,5 pts)

2.1. Décrire en français (type, droits, taille, nom, ...) les objets listés par l'exécution de la commande `ls -l`:

```
-----rwx root root          0 sept. 17 15:21 tmp.log
dr-xr-xr-x jse  www-data 4096 sept. 17 15:21 sav
```

 (1 pt)

Le fichier `tmp.log` créé le 17 septembre à 15h21 appartient à l'utilisateur `root` et au groupe `root`. Il ne contient pas de donnée et est accessible seulement aux autres utilisateurs en lecture, écriture et exécution.

Le répertoire `sav` appartient à l'utilisateur `jse` ainsi qu'au groupe `www-data`. Il a été créé le 17 septembre à 15h21 et possède une taille de 4096 octets (comme tous les répertoires sous linux). Il est accessible en lecture et exécution pour son propriétaire, son groupe et les autres utilisateurs.

2.2. Combien d'utilisateurs le système compte t'il au moins ? (0.5 pt)

Le système à au moins 2 utilisateurs : `root` et `jse`.

3. Générations de systèmes d'exploitation (4,5 pts)

3.1. Décrire brièvement les différentes générations de systèmes d'exploitation. (1 pt)

Il y a eu à l'heure actuelle 5 générations de systèmes d'exploitation :

1. Traitement par lots : Les systèmes de cette génération pouvaient gérer 1 utilisateur réalisant une tâche à la fois ;
2. Multiprogrammation : Ces systèmes pouvaient gérer 1 utilisateur réalisant plusieurs tâches à la fois ;
3. Temps partagé : Premiers systèmes à pouvoir gérer à la fois plusieurs utilisateurs réalisant plusieurs tâches ;
4. Temps réel : Systèmes pouvant réaliser des tâches en temps réel ;
5. Systèmes distribués : Les systèmes de cette génération peuvent gérer des utilisateurs, des tâches et des données réparties sur différentes machines connectées (cloud).

3.2. Que signifient les notions de système multi-utilisateur et système multi-tâche ? (1 pt)

Un système multi-utilisateur est un système pouvant gérer plusieurs utilisateurs en même temps.

Un système multi-tâche est un système pouvant exécuter plusieurs tâches en même temps.

3.3. Quelle est la différence entre un processeur et un processus ? (0.5 pt)

Un processeur est un composant physique capable d'exécuter des instructions. Un processus est un ensemble d'instructions en cours d'exécution. Un processeur permet l'exécution des processus.

3.4. Décrire brièvement ce qu'est un ordonnanceur. (1 pt)

Un ordonnanceur est un module du noyau d'un système d'exploitation. Il sert à répartir la charge du processeur afin d'optimiser l'exécution des processus en parallèle.

3.5. Décrire brièvement un algorithme d'ordonnement de votre choix. (1 pt)

Un algorithme d'ordonnement standard est celui du tourniquet (Round Robin). Donne l'accès à chaque processus de manière équitable au processeur en faisant tourner les processus à chaque exécution.

4. Commandes de base (2,5 pts)

Dans cet exercice vous devez écrire en réponse aux questions les commandes ou suites de commandes à taper pour exécuter l'action demandée.

4.1. Afficher la liste détaillée de tous les fichiers et répertoires contenus dans /etc. (0.5 pt)

```
ls -al /etc
```

4.2. Afficher le contenu du fichier /home/book.txt. (0.5 pt)

```
cat /home/book.txt
```

4.3. Aller dans le répertoire de l'utilisateur courant. (0.5 pt)

```
cd ~
```

4.4. Afficher la liste de tous les processus tournant sur la machine. (0.5 pt)

```
ps aux ou ps -aux
```

4.5. Afficher le répertoire courant de la console. (0.5 pt)

```
pwd
```

5. Grep (4 pts)

Dans cet exercice vous devez écrire en réponse aux questions les commandes ou suites de commandes à taper pour exécuter l'action demandée.

5.1. Lister les lignes du fichier texte.txt qui contiennent la chaîne abc ou la chaîne abcabc. (0.5 pt)

```
grep texte.txt -e '\(abc\)\{1,2\}'
```

5.2. Lister les lignes de texte_01.txt qui contiennent des chaînes de caractères composées d'un d suivi de trois caractères quelconques puis d'un f. (0.5 pt)

```
grep texte_01.txt -e 'd...f'
```

5.3. Lister les commandes du répertoire /usr/bin dont le nom débute par un x, termine par un v et contient au moins une lettre minuscule. (1 pt)

```
ls /usr/bin | grep '^x.*[a-z]\+.*v$'
```

5.4. Compter le nombre de lignes vides du fichier texte_01.txt. (1 pt)

```
grep texte_01.txt -e '^$' -c
```

5.5. Afficher toutes les adresses internet du fichier exam.html qui sont contenues entre des guillemets ("). Une adresse internet est définie par http:// puis une suite de caractères n'étant pas des espaces. (1 pt)

```
grep exam.html -e 'http://[^\ ]*' (Attention à l'espace après le ^)
```

6. Find (4 pts)

Dans cet exercice vous devez écrire en réponse aux questions les commandes ou suites de commandes à taper pour exécuter l'action demandée.

6.1. Rechercher dans le répertoire /mydir et ses sous-répertoires les fichiers terminant par .avi et contenant 2 chiffre dans leur nom. (0.5 pt)

```
find /mydir -type f -name '*[0-9][0-9]*.avi'
```

6.2. Rechercher dans le répertoire /mydir et ses sous-répertoires les fichiers contenant repA ou rep2 dans leur chemin. (0.5 pt)

```
find /mydir -type f -path '*rep[A2]*'
```

6.3. Rechercher dans le répertoire /mydir les répertoires dont le nom termine par une lettre ou les fichiers dont le nom termine par un nombre différent de 3. (1 pt)

```
find /mydir -type d -name '[a-zA-Z]' -o -type f -not -name '*3'
```

6.4. Rechercher dans le répertoire /mydir et ses sous-répertoires les fichiers dont le nom commence par la lettre A mais ne finissent pas par .tmp. (1 pt)

```
find /mydir -type f -name 'A*' -not -name '*.tmp'
```

6.5. Afficher les droits des fichiers contenus dans le répertoire /mydir et ses sous-répertoires et dont les noms commencent par f et finissent par un t ou un 4. (1 pt)

```
find /mydir -type f -name 'f*[t4]' -exec ls -l {} \;
```

7. Avancé (2 pts)

Dans cet exercice vous devez écrire en réponse aux questions les commandes ou suites de commandes à taper pour exécuter l'action demandée.

7.1. Afficher le nom des commandes en exécution sur la machine triées de celles qui consomment le moins de mémoire à celle qui en consomme le plus. (1 pt)

```
ps aux | tr -s ' ' | cut -d ' ' -f 6 | sort -n (le numéro de colonne du cut peut être quelconque)
```

7.2. Afficher la liste des utilisateurs exécutant des processus sur votre machine (chaque utilisateur ne doit apparaître qu'une fois). (1 pt)

```
ps aux | tr -s ' ' | cut -d ' ' -f 1 | sort | uniq
```