

Examen de Programmation système – Systèmes d'exploitation

Enseignant (s) : ...Mickaël Hoerd.....

Année(s) d'études : ...2.

Nom, Prénom :

Date et heure : ...5/12/2014.....8h15...

Locaux : ...Batiment B.....

Durée de l'examen : 90 min ; pas de sortie durant les 45 premières minutes

Nombre de feuilles : ...3 + 1 (Annexe)...

Modalités : tous les documents sont autorisés, vos réponses sont à fournir directement sur ce document.

Spécifications particulières : Examen sur 25 points. Les parties A et B sont à réponses courtes. Les parties C et D sont à réponses qui nécessitent plus de temps. **N'oubliez pas d'y indiquer votre nom et prénom(s) et évaluez soigneusement le temps consacré à chaque question : les parties A et B ne devraient pas vous prendre plus de 20 min.**

A. Structures de données d'un système d'exploitoin (3 points)

- A.1 Expliquer pourquoi l'abstraction d'un fichier de type UNIX permet [3 points] d'accéder aussi bien à des ressources logiques (comme les fichiers stockés par ex) qu'à des ressources physiques (comme une souris ou de la mémoire par ex) en utilisant les mêmes appels systèmes `read()` et `write()`

B. Shell (9 points)

En annexe AN.1, on considère la copie d'écran suivante prise lors d'une session shell exécutée dans un terminal. Chaque ligne de la copie d'écran a été préfixée par son numéro.

- B.1 Quel est le numéro d'identifiant d'utilisateur qui a exécuté cette session ? [1 points]
- B.2 Expliquer le plus précisément possible chaque partie composant la ligne 1 et 2 [1 points]
- B.3 Que représente la ligne 4 ? Expliquer le plus précisément possible [1 point] comment elle a été produite ? A partir de la ligne 5, déduire le chemin d'accès absolu des fichiers listés en lignes 7 à 11.
- B.4 Donner en français les droits d'accès d'utilisateur, de groupe et de tous [0.5 points] les autres du fichier `python2.7`
- B.5 `pacome` pourra-t-il exécuter le fichier `python3.2` ? Justifier [1 point] soigneusement votre réponse.
- B.6 Qui peut exécuter le fichier `udev` ? [0.5 points]
- B.7 L'administrateur de la machine aimerait permettre à tous les utilisateurs [1 point] du groupe `staff` d'exécuter `python_read`. Donner la ligne de commande Unix exacte qui permet d'effectuer cette action. On suppose que l'administrateur se trouve dans le même chemin d'accès que `pacome`.

B.8 Tout de suite après avoir exécuté `ls -l ..`, pacome exécute la ligne de commande suivante : [1 point]
`pacome@examen:/usr/local/lib$ cd ../; echo p* p*`

Qu'affichera précisément cette commande sur le terminal ?

B.9 Le fichier `python_read` est un exécutable qui lit l'entrée standard et affiche sur la sortie standard le résultat du traitement de ce qu'il a reçu sur l'entrée standard. Donner une commande shell qui permet d'envoyer `data.txt` dans l'entrée standard de `python_read` et en même temps de rediriger la sortie de `python_read` dans un fichier `results.txt` situé dans `/tmp` [1 point]

B.10 Plutôt que d'envoyer `data.txt` dans `python_read`, on aimerait envoyer la sortie standard du programme `python2.7` dans `python_read`. Donner une commande du shell qui permet d'effectuer cette action. [1 point]

C. Shell-scripts (8 points)

Soit le script mystérieux suivant en annexe AN.2

C.1 Quel est l'effet de l'option `-n` sur la commande `echo` ? [0.5 points]

C.2 Donner le résultat sur la sortie standard de l'exécution du script avec comme paramètres les valeurs 1 et 2 puis 4 et 5. Vous pouvez vous aider de la question C.5 [3 points]

- C.3 Comment faire pour transformer ce script en fonction shell dont le premier argument est \$1 ? [0.5 points]
- C.4 D'après la question C.2 à quoi servent les entiers x , y dans l'algorithme du script ? [3 points]
- C.5 En fait ce script affiche un triangle rectangle sur le terminal. Modifier le script pour qu'il inverse le sens d'affichage du triangle de haut en bas. [1 points]

D. Entrée/Sorties système en langage C (5 points)

Cette partie n'a qu'une question : D.1

- D.1 La commande Unix `wc` l'entrée standard jusqu'à la fin de fichier, puis affiche sur la sortie standard le nombre de mot, de ligne et de caractères lu. Écrire la commande `wc` en C à l'aide des appels système vus en cours.

AN.1

```
1 pacome@examen:/usr/local/lib$ id
2 uid=1244(pacome) gid=2000(staff) groupes=2000(staff)
3 pacome@examen:/usr/local/lib$ pwd
4 /usr/local/lib
5 pacome@examen:/usr/local/lib$ ls -l ..
6 total 16
7 -r-xr-xrwx 4 root staff 4096 nov 24 17:23 python2.7
8 -rw-rwxr-- 3 root staff 4096 oct 17 2012 python3.2
9 -rwxr-xr-x 4 root root 4096 oct 26 08:37 python_read
10-rw-r--r-x 3 root root 4096 sep 6 14:11 udev
11-rw-r----- 1 pacome syslog 55 nov 12 08:06 data.txt
```

AN.2

```
#!/bin/bash
```

```
for ((x=0;x<$1;x++))
do
    for((y=x;y>=0;y--))
    do
        echo -n "#"
    done
    echo " "
done
```