

# Devoir à la maison de système d'exploitation

## Première année

Département informatique - IUT Villetaneuse  
Vendredi 24 mars 2006 au lundi 27 mars 2006

Remarques : le barème est indicatif.

### 1- Commandes unix, questions liés au cours – 5 pts

	<i>Commandes</i>	<i>Questions / Affirmations</i>	<i>Réponse / Justification</i>	<i>Points attribués</i>
1.1	<pre>\$ a=MOU.htmlj \$ if [ `echo \$a   grep "[:alpha:][:punct:]" ` ]; then echo "YES"; else echo "NO"; fi</pre>	Le programme affiche YES ?	Justifications	1.5
1.2		Il est possible de récupérer les sources écrites en C de la commande grep et de les recompiler sous Windows .	Vrai ou Faux	1/2
1.3		L'utilitaire bc comporte un langage de programmation permettant de faire de l'arithmétique.	Vrai ou Faux	1/2
1.4		La commande info tr permet d'obtenir de l'aide sur tr. C'est un complément de man tr ?	Vrai ou Faux	1/2
1.5		La commande tr, comme sed, permet de faire de la recherche et de la substitution ?	Vrai ou Faux	1/2
1.6	<pre>\$ ps aux   grep -E "1{2}"</pre>	Affichage des informations sur les processus dont les programmes associés comportent exactement 2 l.	Justifications	1.5

### 2- Développement d'un programme Bash - 5 pts

On vous demande d'écrire un programme Bash de nom `exo.sh` qui prend 3 paramètres que nous appelons dans l'énoncé `x`, `y` et `fic` et qui doivent être fournis dans cet ordre. `x` et `y` sont deux entiers, `fic` un nom de fichier texte. Le script affichera `x` lignes du fichier `fic` prises toutes les `y` lignes à partir de la première ligne. On vérifiera dans le script que les paramètres sont corrects ( $y \geq x > 0$  et que `fic` existe).

*Hint* : voir les commandes `head` et `tail`.

### 3- Analyse et développement d'un script – 10 pts

3.1 Expliquez le code suivant. a) comment s'utilise t-il b) que fait-il ?

```
#!/bin/bash
# Source: M. Divay
case $# in
  0)NB=30;;
  1)NB=$1;;
  *)echo "usage: $0 [nombre-de-jours]"; exit 1;;
esac

Reppou=$HOME/.poubelle
if [ ! -d $Reppou ]; then
  echo "$Reppou n'existe pas"; exit 1;
fi

find $Reppou -mtime +$NB -exec /bin/rm -i -r {} \; 2>/dev/null

if [ `ls $Reppou | wc -l` -eq 0 ]; then
  rmdir $Reppou
fi
# Options de find :
# -mmin n
# Fichier dont les données ont été modifiées il y a n minutes.
#
# -mtime n
# Fichier dont les données ont été modifiées il y a n*24 heures.
#
# Les arguments numériques peuvent être indiqués comme suit :
# +n supérieur à n,
# -n inférieur à n,
# n égal à n.
```

3.2. Modifiez le code pour traiter successivement les cas suivants : a) seuls les fichiers dont le nom contient un certain motif donné par \$2 et vérifiant les propriétés du code ci-dessus doivent être détruits b) les fichiers contenant le motif \$2 ne doivent pas être détruits mais déplacés dans \$HOME/.poubelleCOPY c) les fichiers contenant le motif \$2 ne doivent pas être détruits mais déplacés dans \$HOME/.poubelleCOPY en passant les noms en majuscules.

**Notes :** a) le motif \$2 n'est pas une expression régulière mais peut contenir des classes comme dans "[[:digit:]]\*.txt" si bien que l'on peut utiliser l'option -name de find b) pensez à bien expliquer ce que vous faites et en particulier à mentionner les vérifications d'usage.