

## Contrôle continu - 2/11/2021

*Durée 1h30. Documents autorisés : une feuille A4 R/V notes de CM et de TP.*

**Nom et prénom :**

**Groupe :**

**Exercice 1** Que font les commandes suivantes ?

1. `find /bin -perm -110 -depth 1 -name 'p?'`

2. `ls / | head -5`

3. `ls exo[0-9]-[1-5]? .sh`

4. `sort -r < texte1.txt > texte2.txt`

**NB** : l'option `-perm -mode` de la commande `find` sélectionne les fichiers ayant au minimum les droits spécifiés par `mode`.

**Exercice 2** Donner le résultat de la commande :

```
for x in {5..10}; do let s+=x; done; echo $s;
```

**Exercice 3** La commande `sleep` permet de marquer un temps d'arrêt dans un script bash (`sleep - suspend execution for an interval of time`). Considérons le script `narco.sh` suivant :

```
#!/bin/bash
for x in $*; do
  ( sleep $x; echo -n $x) &
done
```

Quel est le résultat de `narco.sh 3 1 4 1 5` ?

**Exercice 4** La commande `date` affiche la date courante au format suivant :

```
Lun 18 oct 2021 21:59:17 CEST
```

Écrire une commande bash en une ligne qui affiche seulement l'heure et les minutes comme suit :

```
21:59
```

**Exercice 5** Rappelons que la commande `ls -l` produit un affichage comme suit pour chaque fichier ou répertoire :

```
-rw-rw-r-- 1 toto staff 239 16 oct 2020 exo1-tp3.sh
-rwx--x--x 1 toto staff 222 27 sep 2020 exo1.sh
-rw-r--r-- 1 toto staff 327 16 oct 2020 exo2.sh
-rwx--x--x 1 toto staff 197 15 déc 2020 exo3-1.sh
```

Écrire une commande bash en une ligne qui compte le nombre de fichiers script shell accessibles en lecture pour les utilisateurs du même groupe que le propriétaire.

**Exercice 6** Soit un fichier dont le nom est dans une variable `fnom`. Écrire un script shell qui crée un fichier dont le nom est le contenu de la variable `fnom` auquel on a ajouté le suffixe `2` (par exemple, si le fichier s'appelle `foo`, le fichier qu'on doit créer s'appellera `foo2`) et tel que le contenu de ce fichier créé est le même que celui du fichier d'origine sans aucune ligne vide.

**Exercice 7** Soit la commande qui déplace tous les fichiers dont le nom finit par `".mov"` dans un répertoire fils déjà existant appelé `films`. Écrire un script shell qui réalise cette commande en adoptant les deux solutions suivantes :

1. utiliser tout simplement l'instruction `for fichier in *.mov`

2. utiliser la commande `find`, sans descente récursive (avec une profondeur maximale de 1).

**NB** : avec l'option `-name`, la commande `find` permet de chercher des noms de fichiers spécifiques.

À partir du répertoire courant, la commande `find` préfixe par `'./'` les noms de fichiers. On rappelle que la longueur d'une chaîne `s` est obtenue par `${#s}` et que `${s:position}` désigne la chaîne extraite de `s` à partir de l'index `position`.

**Exercice 8** Un enseignant de TD dispose d'un fichier `elevés.txt` contenant les prénoms et noms de l'ensemble des étudiants du groupe de TD. Le fichier est de la forme suivante :

```
nom1:prenom1
nom2:prenom2
nom3:prenom3
```

1. Écrire un script shell qui crée un fichier `emails.txt` dans lequel chaque ligne contient l'adresse mail d'un étudiant du fichier `elevés.txt`, sous la forme `prenom.nom@etu.univ-orleans.fr`.

2. En utilisant le fichier `emails.txt`, écrire *une commande bash* en une ligne de commande pour afficher la liste des étudiants sous le format

```
prenom1 nom1  
prenom2 nom2  
prenom3 nom3
```

**Exercice 9** On possède un fichier `IP.csv` qui contient les informations sur l'inscription pédagogique, où chaque ligne est du format suivant :

```
nom:prenom:inscIF01:inscIF02:inscIF03:inscIF04:inscIF05
```

où pour les cinq modules, `inscIF0i` est `oui` ou `non` pour l'inscription dans le module.  
Écrire un script shell qui compte le nombre d'inscriptions pour le module IF03.