# Travaux Dirigés Système n°1

**Cours d'Informatique de Deuxième Année**

*— Licence 2ème Année, Deuxième Semestre MIAS —*

## Shell

Tout ce TD sera **obligatoirement** effectué en mode console. Pour passer d'un mode à l'autre vous utiliserez les combinaisons de touches suivantes :

- Mode console : *CTRL+ALT+F1*.
- Mode graphique : *CTRL+ALT+F7*.

**▸ Exercice 1. (Récupération d'informations sur votre système)**
*À l'aide de la commande cat, déterminez les informations suivantes :*

1. *Fréquence de votre processeur.*
2. *Quantité de mémoire vive.*
3. *Les versions de votre noyau et du compilateur gcc.*

**▸ Exercice 2. (Navigation dans les répertoires)**
*À l'aide de la commande ls déterminer :*

1. *Le contenu de votre répertoire courant.*
2. *Réaliser l'affichage ligne par ligne.*
3. *Affichez également les fichiers cachés*
4. *Affichez uniquement les fichiers se terminant en ".c"*

*Maintenant que fait la commande suivante :*

```
ls -l | grep -v '^d'
```

**▸ Exercice 3. (Statistiques sur un fichier)**
*À l'aide de la commande wc déterminer :*

1. *Le nombre de lignes de votre projet du semestre précédent.*
2. *La longueur de la plus longue ligne.*
3. *Le nombre de mots (uniquement).*

*À l'aide de la commande file déterminer :*

1. *Le type du fichier (texte, image, etc.).*
2. *Le format des données dans le cas de fichiers multimédias.*
3. *Le système d'encodage dans le cas de fichiers de texte.*



**▸ Exercice 4. (Chaînage de commandes)**
*Il est possible grâce à un tube (symbôle |) de chaîner plusieurs commandes. Déterminer avec les commandes ls et wc le nombre d'étudiants appartenant à la même filière que vous.*

**▸ Exercice 5. (Variables)**
*Tapez tout d'abord les commandes suivantes :*

1. ```
   VAR="ls -l | grep -v '^d'"
   echo $VAR
   ```
2. ```
   VAR=`ls -l | grep -v '^d'`
   echo $VAR
   ```
3. ```
   VAR='ls -l | grep -v '^d''
   echo $VAR
   ```
4. ```
   VAR=$(ls -l | grep -v '^d')
   echo $VAR
   ```

*Et expliquez le résultat de ces différentes commandes.*

**▸ Exercice 6. (Affichage des paramètres d'un script)**
*Ecrivez un script qui affiche :*

1. *Le premier argument passé en entrée*
2. *Tous les arguments passés en entrée.*

**▸ Exercice 7. (Tables de multiplication)** *Ecrivez un script qui permet d'afficher les tables de multiplication. Par exemple, lorsque vous lancerez votre script avec les paramètres 5 et 10, votre script affichera le résultat suivant :*

```
0 x 5 = 0
1 x 5 = 1
2 x 5 = 10
3 x 5 = 15
4 x 5 = 20
5 x 5 = 25
6 x 5 = 30
7 x 5 = 35
8 x 5 = 40
9 x 5 = 45
10 x 5 = 50
```

*Indice : Servez vous de la commande seq et de la commande interne for.*

▸ **Exercice 8. (Lecture sur l'interface d'entrée)** *Ecrivez un script qui permet de saisir un nom et un prénom. Pour ce faire vous utiliserez la commande read. Le script doit ensuite afficher ceux ci dans une phrase complète.*

▸ **Exercice 9. (Exercice conditionnel)** *Ecrivez un script qui affiche si vous etes :*

1. *Dans votre répertoire d'accueil.*
2. *Dans le répertoire /etc.*
3. *Dans le répertoire /bin.*
4. *etc.*

*Les forêts d'ifs ne sont pas toujours très lisibles surtout lorsqu'il existe plusieurs conditions très similaires sous des formes à peine différentes ! Pallier à ce problème de lisibilité en utilisant une boucle case.*

▸ **Exercice 10. (While ou until)** *Ecrivez un script permettant de tester si un mot de passe saisie par l'utilisateur correspond au mot de passe spécifié « en dur » dans votre script. Utilisez tout d'abord la commande while puis la commande pendante until.*

*Quelle différence existe-t-il entre ces deux commandes ?*

▸ **Exercice 11. (Taille d'un répertoire)**
*Ecrivez un script shell* `size-rep.sh` *qui permet de connaître la taille, i.e. la somme de la taille de tous ses fichiers, d'un répertoire passé en paramètre.*

**Bonus :** *inclure la taille de tous les sous-répertoires dans ce calcul.*

▸ **Exercice 12. (QCM)** *Ecrivez un script permettant de répondre à un formulaire par des choix multiples.*

*Malheureusement, de nombreux utilisateurs vont faire des fautes de frappes... Pour éviter de devoir réaliser de nombreux tests bien souvent illisibles, utilisez la commande select.*

▸ **Exercice 13. (Un petit jeu)** *Ecrivez un script qui permet de dire à l'utilisateur s'il est proche (brulant), éloigné (gelé), etc. d'un nombre mystère.*

▸ **Exercice 14. (La commande find)**
*Utilisez la commande* `find` *pour retrouver tous les fichiers de nom* `a.out`*, tous les*



*fichiers dont le nom se termine par* `~` *et tous ceux dont le nom commence par* `#`*.*
*Ecrivez un script* `clean.sh` *qui supprime tous les fichiers trouvées par la commande* `find`*. **Attention :** on n'utilisera pas le symbole* `|` *pour effectuer cette action, mais uniquement les paramètres de la commande* `find`*.*

▸ **Exercice 15. (Variables d'environnements)**

1. *Affichez toutes les variables d'environnements.*
2. *Affichez toutes les variables d'environnements dont les valeurs associées contiennent le mot '*`fr`*'*
3. *Ecrivez un script qui demande à l'utilisateur sa nationalité (on se limitera à anglais et français dans les résultats) et qui affiche le résultat entré par l'utilisateur.*
4. *Modifier les variables d'environnements obtenues précédement en fonction du résultat entré par l'utilisateur.*

▸ **Exercice 16. (Bonus : Le bashrc)** *A la racine de votre compte, ouvrez le fichier .bashrc. Ajoutez y des raccourcis qui permettent d'executer les commandes* `size-rep.sh` *et* `clean.sh` *quelque soit votre répertoire courant, en tapant simplement les commandes* `size-rep` *et clean*

▸ **Exercice 17. (Bonus : Session X)** *A partir du mode console, ouvrez une nouvelle session X. Pour se faire tapez la commande* `{X :3\&}`*. Vous pouvez maintenant lancez un gestionnaire de fenêtre ou un terminal dans cette nouvelle session.*

▸ **Exercice 18. kill_by_name.sh**
*A l'aide de la commande ps qui affiche la liste des processus, récupérez à l'aide de la commande grep et cut l'identifiant de tous les processus vous appartenant et dont le nom est passé en argument et terminez les à l'aide de la commande kill."*

▸ **Exercice 19. Compression de projets**
*Cette commande vours servira pour vos projet. On vous demande d'écrire une commande qui à l'aide de find et de zip (et eventuellement grep) n'archive dans le fichier zip que vous envoyer à votre professeur uniquement les fichiers strictement necessaire à la compilation, au test et à la notation de votre projet.*
*Bonus : faire la même chose avec la commande tar et gzip.*