

INTRODUCTION À UNIX  
Feuille n° 2

---

## 1 Gestion de processus

1. Copier le fichier `commande_longue` dans votre home directory.
2. Exécuter cette commande. Suspendez l'exécution, puis reprenez la, cette fois en arrière plan. Ramenez la au premier plan et tuez la avec un `Ctrl C`.
3. Relancez la même commande sans qu'elle se fasse tuer par votre déconnection.
4. Reconnectez vous et tuez la par un `kill`.

## 2 Flux et redirections d'entrées/sorties

1. Lister `/bin` et compter le nombre de commandes.
2. Afficher les lignes 15 à 20 de `zola1.txt`.
3. Supprimer les lignes vides des fichiers `zola*.txt` en mettant le résultat dans un seul fichier `germinal_sansligneblanche.txt`. À l'aide de la commande `cut`, en considérant le signe `:` comme un délimiteur de champs, on éliminera de chaque ligne le nom du fichier d'où elle provient.
4. Recopier `MPP.tgz` dans votre home directory. En extraire le répertoire MPP. Créer un fichier `s1np` avec les lignes contenant `s1np` se trouvant dans tous les fichiers du répertoire. Éliminer de chaque ligne tout le début (jusqu'à la valeur numérique de `s1np` et créer un second fichier `valeurs_s1np`. Enfin trier par ordre croissant ce fichier de valeurs numériques (on créera un troisième fichier `valeurs_s1np_triees`.
5. Créer un fichier `liste_fic` contenant la liste de tous les fichiers (uniquement les fichiers, pas les répertoires) de votre compte avec leurs caractéristiques (taille, droits ...)

6. Quels sont les dix plus gros fichiers de `/usr/bin` ?

### 3 Shell scripts

1. Éditer (avec `xemacs`) un fichier `mon_script`, contenant le texte suivant :

```
#!/bin/bash
# ligne de commentaire
echo "Le nombre d'arguments de ce script est $#"
```

```
echo "Les arguments sont $*"
echo "Le premier argument est $1"
echo "Entrez un nombre au clavier"
read nombre
echo "Le nombre que vous avez entre est $nombre"
```

Exécutez-le.

2. **Redirections**

- (a) Écrire un script qui à chaque exécution rajoute dans le fichier `'journal'` les informations suivantes :
  - heure d'exécution du programme
  - nom de l'utilisateur
  - charge de la machine

Après deux exécutions, le fichier `'journal'` pourrait avoir l'aspect suivant :

```
----
mer sep 10 11:30:36 CEST 2003
sylvain
0.02
----
mer sep 10 11:31:07 CEST 2003
sylvain
0.05
----
```

3. **Un peu plus compliqué**

Écrire un script qui imprime le chemin vers le répertoire racine d'un utilisateur. Le nom de l'utilisateur étant pris en argument. (indication : les informations sur les utilisateurs peuvent être obtenues en tapant la commande `yycat passwd` )

#### 4. Tests

- (a) Écrire un script qui teste l'existence d'un fichier.
- (b) Écrire une commande 'cp' sécurisée, qui refuse d'écraser un fichier existant
- (c) Modifier le programme précédent pour qu'il demande une confirmation avant d'écraser le fichier destination

#### 5. Boucles

- (a) Écrire un script qui rajoute .c a la fin de tous les fichiers d'un répertoire
- (b) Écrire un script qui renomme tous les fichiers portant le suffixe .f en fichiers portant le suffixe .c.
- (c) Écrire un script qui détruit toutes les instances d'un programme donné (par exemple toutes les instances de Mozilla en train de s'exécuter).
- (d) Écrire un script lsdirc qui liste seulement les répertoires.