

## TD5 : Gestion des processus

### Exercice 1 :

1. Lancer la commande `sleep 9999` en arrière-plan
2. Quels sont le PID et le numéro de travail (job) du processus `sleep` précédent ?
3. Suspendre le processus `sleep` précédent et vérifier son état.
4. Relancer maintenant le processus `sleep` et vérifier son état.
5. Tuer le processus `sleep`.

### Exercice 2 :

1. Lancer deux terminaux (`tty1` et `tty2`)
2. Créer le script suivant et rendre le exécutable. Nommer le script `l.sh`:

```
#!/bin/bash
alphabet="a b c d e f g h i j k l m n o p q r s t u v w x y z"
while true
do
  for carac in $alphabet
  do
    echo $carac
    sleep 1
  done
done
```

Executer la commande suivante: `./script.sh>>./log &`

Analyser le script en expliquant chaque partie.

3. Dans le terminal `tty2`, utiliser la commande (`tail -f`) pour afficher en temps réel le contenu du fichier `./log`
4. Dans le `tty1`, interrompre le processus en cours d'exécution en utilisant `Ctrl+Z`.
5. Quel est le numéro de travail associé au processus interrompu ?
6. Vérifier dans `tty2` que le remplissage du fichier `log` est arrêté.
7. Dans le `tty1`, lancer à nouveau le processus arrêté mais en arrière plan.
8. Vérifier que le travail déjà arrêté est en cours d'exécution. Vérifier dans `tty2`, l'état de fichier `./log`

9. Dans tty1, créer et lancer le script nombre.sh suivant avec la commande

```
./nombre.sh>>./log &

#!/bin/bash
c=0
while true
do
    c=`expr $c + 1`
    echo $c
    sleep 1
done
```

10. Dans tty2, vérifier que les trois processus sont en cours d'exécution et consulter le fichier ./log
11. Dans tty1, utiliser ensuite kill pour envoyer un SIGSTOP au numéro de travail du premier processus exécuté pour le suspendre. Vérifier avec la commande jobs
12. Relancer le travail du premier processus à l'aide de kill et le signal SIGCONT.
13. Utiliser SIGTERM pour terminer les travaux [2] et [3]. Vérifier avec jobs qu'ils sont terminés.
14. Vérifier si tous les processus ont été terminés, sinon terminer ceux qui restent.