

# Rapport intermédiaire Projet Mini-Shell

## Joël Roman Ky 1SN-M

**Question 1 :** Implémentation de la boucle principale avec la procédure **lire\_commande()** qui affiche le nom du répertoire principale et récupère le répertoire courant

**Question 2-3 :** Utilisation de la commande **waitpid** pour mettre en attente le processus père jusqu'à la fin de l'exécution du fils.

**Question 4 :** Ajout des commandes internes **cd** à l'aide d'une procédure void en supposant que le paramètre `cmd->seq[0][0]` est égale à **cd** en distinguant deux cas :

-si `cmd->seq[0][1] = NULL` : le répertoire courant n'est pas **HOME**. On récupère le chemin de Home en utilisant `getenv("Home")` et on se déplace en utilisant `chdir`.

-si `cmd->seq[0][1] != NULL` : on utilise `chdir` pour remplacer le répertoire courant par celui donné en paramètre.

La commande interne **exit** se fait avec la procédure prédéfinie.

**Question 5 :**

Question pas bien comprise

-Si `cmd -> backgrounded = NULL` : Dans ce cas on attend le fils jusqu'à qu'il finit l'exécution de la commande en utilisant `waitpid` sinon on appelle pas `waitpid`.

**Question 6 :**

Implémentation des commandes externes **bg, fg, stop, jobs** traitées dans la boucle principal avec un **if...else if ..else**.

Les commandes **bg, fg et stop** consistent à retrouver le pid du processus et arreter le processus en cours et mettre à jour la liste des processus après.

La commande **jobs** consiste à afficher la liste des processus ce qui est déjà fait par la procedure dans le module **Processus.h**