Avancée du projet au 28 avril (avant le rendu intermédiaire du 29)

Questions traitées :
Les questions 1 à 7 sont selon moi traitées et fonctionnelles.
La question 8 est traitée partiellement uniquement car non fonctionnelle (voir note).

Choix de conceptions :
- Pour effectuer une commande externe (i.e une commande qui n'est pas cd / lj / sj / bg / fg)
on fork pour créer un nouveau processus qui exécute la commande externe.

- Pour effectuer une commande interne (i.e une commande qui est cd / lj / sj / bg / fg)
on utilise pour cd la primitive chdir pour changer le répertoire courant.
on utilise pour lj / sj / bg / fg le module job.h qui permet à travers une liste chainée de processus
de conserver une trace des processus en cours d'exécution avec leur état, on peut alors modifier cette liste
selon le besoin des différentes commandes internes effectuées.

- Pour gérer l'execution des commandes en avant-plan / arrière-plan
dans le cas d'un processus en avant plan on force l'attente de la fin de l'exécution de la commande avec la fonction
waitForegroundedProcess qui attend que le processus en avant plan se termine, dans le cas d'un processus en arrière plan on laisse
le contrôle au signal handler handleSIGCHLD qui se charge de gérer les changements d'état des processus en arrière plan.

- Pour gérer ^C et ^Z
On utilise le signal handler handleSIGINT qui se charge de gérer les interruptions de l'utilisateur.
On utilise le signal handler handleSIGTSTP qui se charge de gérer les interruptions de l'utilisateur.
On utilise donc un masquage de ces deux signaux pour le processus principal pour que le shell ne se ferme pas lorsque
l'utilisateur effectue une de ces deux actions. On considère également que le processus fils ne peut pas démasquer les deux
signaux ou modifier les traitants.

note : il y a un problème du à la conception du shell lors de l'utilisation de ^Z, l'état du processus est modifié correctement
mais la main n'est pas redonnée correctement au processus principal. Il faudra corriger cela.

- Pour la lecture des commandes
On utilise le module fourni readcmd qui permet de lire les commandes entrées par l'utilisateur.
Pour implanter des fonctionnalités comme l'auto-complétion avec tab ou la proposition des commandes précédentes avec la flèche du haut
il faudrait trouver une autre solution.

/** Non implantés **/
- Pour gérer les redirections d'entrée et de sortie

- Pour gérer les pipes