Note Projet SEC

Valentin Moutteau

rendu intermédiaire

Question 2 :

Par exemple, en utilisant le programme sleep.c qui lance lui même la primitive sleep, et qui en plus envoie un message à la fin, on observe que le message de fin d'attente arrive après l'invite de commande.

Avancé du projet

Globalement, le code fournit répond aux questions 1 à 6 du projet, avec quelques problèmes restant comme par exemple le fait qui l'on attend la fin d'un processus en avant plan et qu'un processus en arrière plan se finit avant, l'invite de commande va se remontrer. Deux modules de plus sont utilisés : liste_jobs qui implémente un structure de liste chaînées pour la gestion des processus courant, et cmd_internes qui vérifie et éxécute, dans le cas échéant les commandes internes à ce minishell. Pour l'instant, le programme ne gère pas la vérification du bon fonctionnement des appels aux primitives.

Choix de conception

Pour les commandes internes : gérées par un module, qui, si une commande interne est éxécutée renvoie des variables qui disent si l'on doit créer un fils, ou attendre un processus en avant-plan.

Pour la gestion des processus courant, on utilise un handler qui s'éxécute à la réception d'un SIGCHLD, qui, à l'aide de la primitive waitpid et de certaines macros, permet de mettre à jour l'état des processus courant. On utilise de plus des listes chaînées, qui ajoute toutes les commandes non internes. Pour l'instant, le programme ne supprime pas les processus terminés, un tri est fait au moment de l'affichage. De plus, dans l'affichage des processus courant, on utilise une valeur de 0 correspond à un processus actif, 1 à un processus

Questions 7 et 8

Pour ces questions, j'ai pensé à associé un handler aux signaux SIGINT et SIGSTOP qui n'interomperaient pas le minishell, mais qui doivent interrompre, ou terminer le processus en avant plan. Pour cela, je pense qu'il faudrait rajouter un champ est_en_avant_plan au type job, afin de pouvoir, lors de la réception d'un des deux signaux, de parcourir la liste des processus courant, et de l'interrompre, ou le stopper avec la primitive kill.