Rendu intermédiaire

Romain Peyremorte

Questions complétements traitées

Les questions 1 à 8 ont été selon moi complétement traitées.

Conception

La compréhension de la fonction readcmd() fut un premier point de blocage, ainsi que comment gérer plusieurs commandes entrées dans le terminale. Pour ce dernier point, une boucle *while* a été choisie avec l'évolution d'un entier *i* qui correspond à la commande lu, et qui évolue jusqu'à ce qu'il y ait plus de commande.

Un autre point de blocage est la lecture des commandes pour différencier celles qui ont un traitement particulière (*exit*,*cd*, etc). Il a été décidé d'utiliser le comparateur de chaîne *string*. Ce comparateur empêcha l'utilisation efficace du *switch case* qui ne marche pas sur les *strings*, donc le choix d'utiliser un enchaînement d'*else if* s'est imposé, mais ne me semble pas propre, surtout si d'autres commandes particulières doivent être traitées.

Pour la partie mémorisation des commandes en cours de fonctionnement, j'ai opté pour une liste chaînée pour ne pas à avoir de problèmes de taille (comme avec un tableau). Pour cela, il a fallu implémenter des fonctions propres à la gestion de liste chaînée, mais aussi des fonctions propres à cette liste : retrouver un pid à partir d'un id ou inversement par exemple.

Une difficulté rencontrée est la gestion du signal *SIGCHLD* où cela fut nouveau et où il fallait gérer la suppression de la liste des commandes finies. Cela a permis de commencer à découvrir le *waitpid* dans un nouveau cas.

La question de l'interruption de la commande exécutée en premier plan fut aussi compliquée car il a fallu reprendre la gestion d'ajout ou non de commande à la liste, ainsi que la suppression ou non. De plus, le *wait* n'était plus suffisant, j'ai dû le remplacer dans ce cas par un *waitpid*.

La question de l'arrêt de la commande à l'aide du *crtl-C* ne fut pas tant compliqué, seulement la partie de gestion pour ne pas arrêter le processus père principal.

Une difficulté rencontrée mais non solutionnée est la gestion d'une suite de commande avec un ou plusieurs en arrière-plan: par exemple la suite de commande « ls &|pwd &|lj » entraine une erreur et « ls &|pwd|lj » entraine l'exécution en arrière-plan de « ls » et de « pwd » alors que que « ls » à *&*. Ce problème repose sur la détection du *&* dans *readcmd()* et le retour de cette dernière dans une seule « variable » commune pour toutes les commandes.