

Projet système, Polytech ET3

Introduction

Dans ce projet, nous vous proposons de coder un shell en C, avec quelques fonctionnalités basiques du shell. Vous pourrez vous aider de la documentation suivante: https://www.gnu.org/software/libc/manual/html_node/Implementing-a-Shell.html#Implementing-a-Shell. Ce projet devra être réalisé en C, le code devra compiler sans erreurs ni warnings. Votre code doit être écrit en respectant les bases de la bonne programmation (modularité, clarté, commentaires...).

1 Organisation

- Projet à faire par groupe de 3 ou 4.
- Les séances de TP projet sont l'occasion de poser des questions et de se faire aider par vos enseignants. On vous conseille de travailler le projet en dehors de ces séances pour mieux avancer.
- On conseille aux membres du groupe de bien s'organiser pour répartir au mieux la charge de travail.
- Les soutenances se feront en dernière séance de TP projet par les membres du groupe. Mais l'évaluation est individuelle si l'on considère qu'un étudiant a relativement mieux travaillé que les autres. On vous donnera plus de détails prochainement.
- Vous devez fournir par ailleurs un petit rapport descriptif de votre travail juste avant la soutenance.
- Vous devrez rendre le code de votre shell dans une archive zippée. Veillez à ce que la racine de votre archive soit un répertoire avec un nom permettant de vous différencier des autres élèves.

2 Premier Shell

Dans cette première partie vous allez mettre en place les fonctionnalités principales de votre shell.

Ces fonctionnalités sont :

- Lancer un programme dans un nouveau processus depuis une commande (par exemple `./Discord` pour lancer Discord si vous êtes dans le répertoire où vous avez installé Discord).
- La commande `cd` vous permettant de changer de répertoire courant.

Vous pouvez vous aider de la documentation et du code donné dans l'introduction.

3 Gestion des entrées/sorties

Dans cette section vous allez devoir gérer les redirections de vos entrées et sorties.

- Votre shell devra pouvoir rediriger l'entrée du processus grâce à `< filename`.
- Il devra rediriger la sortie du processus grâce à `> filename`.
- Il devra être possible d'utiliser les deux conjointement pour rediriger l'entrée et la sortie.

Vous pourrez vous aider de l'appel système `dup`.

L'étape suivante est de mettre en place les pipes. Vous devrez pouvoir enchaîner les pipes. N'oubliez pas qu'un pipe est entre deux processus, vous devrez donc pouvoir lancer deux ou plus processus depuis la même commande.

4 Mettre les processus en background

Dans cette section vous devrez gérer la mise en foreground ou background des processus. Vous pourrez vous aider de la documentation pour comprendre en détails les implications de la mise en avant ou en arrière d'un processus. Pour résumer:

- Un processus en foreground prend le contrôle du terminal et doit le rendre quand il se termine.
- Un processus en background ne doit pas bloquer le terminal.

5 Gestion de fichiers et de répertoires

Implémentez les commandes `cp` (avec l'option `-r`) et `mkdir`. `cp` permet de copier/coller des fichiers et `mkdir` permet de créer un répertoire.