

Projet de Programmation Système

Création d'un mini-Shell sous Ubuntu



Réalisé par : Grich Yahya

Filière : Biostatistique Demographie et Big Data

Encadré par : Mr Anouar Bouchal

Date : 19 mai 2025

Introduction

Dans le cadre du cours de programmation système sous Unix, ce projet consiste à réaliser un mini-Shell fonctionnant sous Ubuntu. Ce mini-Shell permet à l'utilisateur d'exécuter des commandes Unix classiques depuis un programme écrit en langage C.

Le programme exploite les appels système essentiels à la gestion des processus : `fork()`, `execvp()`, et `waitpid()`. Il propose un prompt personnalisé invitant l'utilisateur à saisir des commandes, et continue à fonctionner jusqu'à ce que la commande `exit` soit entrée, ce qui provoque la fermeture du shell.

Objectifs du mini-Shell

- Afficher un prompt personnalisé mini-shell>
- Lire la commande saisie par l'utilisateur
- Analyser et découper la commande en arguments
- Exécuter la commande dans un processus fils
- Attendre la fin du processus fils avant de reprendre
- Quitter le mini-shell sur la commande `exit`

Environnement de travail

- **Système d'exploitation** : Ubuntu 22.04 (machine virtuelle VMware)
- **Langage de programmation** : C
- **Compilateur** : gcc
- **Éditeur de texte** : nano
- **Terminal** : Bash sous Linux

Développement du mini-Shell

Le fonctionnement du mini-shell repose sur une boucle infinie qui réalise les étapes suivantes :

1. Affichage du prompt mini-shell>
2. Lecture de la ligne de commande saisie par l'utilisateur via `getline()`
3. Vérification si la commande est `exit` : dans ce cas, le programme se termine
4. Découpage de la chaîne de caractères en tokens (arguments) avec `strtok()`
5. Création d'un processus fils avec `fork()`
6. Dans le fils, exécution de la commande grâce à `execvp()`
7. Dans le père, attente de la terminaison du fils avec `waitpid()`
8. Retour au début de la boucle pour saisir une nouvelle commande

Cette architecture garantit que chaque commande est exécutée dans un processus séparé, ce qui permet de conserver l'intégrité du shell principal et d'attendre la fin de chaque tâche avant d'accepter la suivante.

Annexes

Installation des outils nécessaires

Avant la compilation, les outils suivants ont été installés :

```
yahya@yahya-VMware-Virtual-Platform:~$ sudo apt install build-essential  
[sudo] Mot de passe de yahya : █
```

Compilation

Pour compiler le programme :

```
yahya@yahya-VMware-Virtual-Platform:~$ gcc minishell.c -o minishell
```

Exécution

Pour lancer le mini-shell :

```
yahya@yahya-VMware-Virtual-Platform:~$ ./minishell
```

Résultats et exemples d'utilisation

Voici quelques exemples de commandes exécutées dans le mini-shell :

```
yahya@yahya-VMware-Virtual-Platform:~$ ./minishell  
mini-shell> ls  
backup      ISTA      myshell    resultat   test      TST  
builtins.c  minishell myshell.c  SE         test2.bz2 Vidéos  
Bureau      minishell.c Prof       snap       test3  
Documents   Modèles   Public     TDI        test4  
Images      Musique    Rempli.txt Téléchargements TRI  
mini-shell> pwd  
/home/yahya  
mini-shell> echo Bonjour Insea  
Bonjour Insea  
mini-shell> exit  
Bye!
```

