MiniShell

MiniShell est un interpréteur de commandes minimaliste permettant d'exécuter des commandes système et de gérer des commandes internes.

# 1. Compilation et installation

**Compiler le projet**

Utilisez le fichier **Makefile** pour compiler MiniShell, cela permet de générer l’executable my\_shell. Faisable en utiliant la commande :

*make*

# 2. Lancer MiniShell

**Mode interactif**

Pour lancer MiniShell en mode interactif :

*./my\_shell*

Dans ce mode, MiniShell affiche un prompt (my\_shell>) et attend que l'utilisateur entre des commandes.

# 3. Utilisation des commandes

**Commandes internes**

MiniShell prend en charge les commandes internes suivantes :

| **Commande** | **Description** | **Exemple** |
| --- | --- | --- |
| cd <dir> | Change le répertoire de travail. | cd /home/user |
| pwd | Affiche le répertoire de travail actuel. | pwd |
| exit | Quitte MiniShell. | exit |
| history | Affiche l'historique des commandes exécutées. | history |

**Commandes multiples**

MiniShell supporte l'exécution de plusieurs commandes sur une seule ligne grâce à des opérateurs :

| **Opérateur** | **Description** | **Exemple** |
| --- | --- | --- |
| ; | Exécute plusieurs commandes séquentiellement. | pwd; ls -l; echo "Done" |
| && | Exécute la commande suivante si la précédente réussit. | mkdir test && cd test |

# 4. Mode batch

MiniShell permet également d'exécuter une commande directement depuis la ligne de commande (sans entrer dans le mode interactif).

*./my\_shell -c "commande"*

**Exemples** :

* Exécuter une commande simple :
* *./my\_shell -c "ls -l"*
* Exécuter plusieurs commandes :
* *./my\_shell -c "pwd; whoami"*

# 5. Historique des commandes

MiniShell enregistre toutes les commandes exécutées dans un fichier **history.txt** et permet de les afficher directement dans le terminal avec la commande *history*.

Exemple :

*my\_shell> history*

*1: pwd*

*2: ls -l*

*3: echo "Hello, Shell!"*

# 6. Nettoyage et recompilation

Pour nettoyer les fichiers générés par la compilation :

*make clean*

Cette commande supprime :

* Les fichiers objets (.o).
* L’exécutable (my\_shell).

Pour recompiler après nettoyage :

*make*

# 7. Structure du projet

Voici la structure des fichiers dans MiniShell :

├── main.c # Point d'entrée principal

├── functions.h # Prototypes des fonctions

├── functions.c # Implémentations des fonctions

├── typedef.h # Définition des structures

├── Makefile # Fichier pour compiler le projet

├── README.md # Documentation du projet

├── history.txt # Historique des commandes

**Exemple complet d'utilisation**

1. **Compilez le projet** :
2. make
3. **Lancez MiniShell en mode interactif** :
4. ./my\_shell
5. **Exécutez quelques commandes** :
6. my\_shell> pwd
7. my\_shell> ls -l
8. my\_shell> cd /home
9. my\_shell> history
10. **Quittez MiniShell** :
11. my\_shell> exit
12. **Exécutez une commande en mode batch** :
13. ./my\_shell -c "pwd; ls -l"
14. **Nettoyez les fichiers compilés** :
15. make clean