

## **Documentation Projet Minishell**

### **I. Introduction**

Le projet Minishell est un interpréteur de commandes Unix basique, développé en langage C et qui offre une interface en ligne de commande permettant d'exécuter des commandes internes, des commandes externes, ainsi que des opérations de redirection et de composition de commandes.

### **II. Structure du Projet**

Le projet est organisé en plusieurs fichiers pour une meilleure modularité et lisibilité du code.

Les principaux fichiers sont :

- **main.c** : Contient la boucle principale du shell, la lecture des commandes et l'exécution des commandes.
- **parser.h / parser.c** : Gère le parsing des lignes de commande utilisateur, la séparation des tokens, et la substitution des variables d'environnement.
- **builtin.h / builtin.c** : Implémente les commandes internes du shell, telles que cd, exit, et export.
- **cmd.h / cmd.c** : Gère la création, l'exécution, et la gestion des structures de commandes (cmd\_t).
- **Makefile** : Fichier permettant de compiler le projet.

### **III. Fonctionnalités Prises en Charge**

#### **1. Commandes Internes**

Le shell prend en charge plusieurs commandes internes, dont :

- cd : Changement de répertoire courant.
- exit : Quitte le shell.
- export : Ajout ou modification de variables d'environnement.
- echo: Afficher du text à l'écran.
- unset : Suppression de variable d'environnement.

#### **2. Commandes Externes**

Le shell peut exécuter des commandes externes fournies par le système d'exploitation.

#### **3. Redirections et Opérateurs**

Le projet prend en charge divers opérateurs et redirections, tels que :

- > : Redirection de la sortie standard vers un fichier.

- **2>** : Redirection de la sortie d'erreur vers un fichier.
- **<** : Redirection de l'entrée standard depuis un fichier.
- **>>** : Ajout de la sortie standard à un fichier existant.
- **2>>** : Ajout de la sortie d'erreur à un fichier existant.
- **>&2**: Rédigé vers la sortie d'erreur.
- **2>&1**: Inverser sortie standard et erreur.
- **|** : Composition de commandes via un tube.

#### **4. Variables d'Environnement**

Le shell peut gérer les variables d'environnement via la commande 'export' et 'unset'.

#### **IV. Utilisation**

1. Compilation : Utiliser le Makefile pour compiler le projet (**make**).
2. Exécution : Lancer le shell avec la commande **“./minishell”** suivante.
3. Utilisation : Saisir des commandes dans l'invite du shell.

#### **V. Conclusion**

Minishell offre une base fonctionnelle pour comprendre et étendre un interpréteur de commandes Unix. Vous pouvez enrichir les fonctionnalités et optimiser le code en fonction des besoins spécifiques du projet.