Rapport Synthétique : Justification des Choix Techniques du Mini Shell

1. Objectif et Organisation

Le mini-shell a été conçu pour exécuter des commandes interactives, gérer les commandes intégrées (cd, pwd), les processus d'arrière-plan (&), les pipelines (|), et la redirection de sortie (>). L'organisation du code est modulaire pour séparer les responsabilités : la lecture (recuperer_cmd.c), le parsing (parser_cmd.c), et l'exécution (exec_cmd.c).

2. Justification des Choix de Fonctions Système

Fonction Système	Contexte d'Utilisation	Justification Technique
fgets	Récupération de l'entrée utilisateur.	Choisi à la place de scanf et fscanf (interdits par le sujet) et pour sa robustesse, évitant les débordements de buffer contrairement à gets (interdit également).
strtok_r vs strtok	Découpage de la ligne de commande.	L'utilisation de strtok_r (version réentrante) est indispensable car le parsing nécessite un **double niveau de découpage** : 1) séparation par opérateurs 2) séparation par les espaces
fork / execvp / waitpid	Exécution des commandes externes.	L'utilisation de ce trio est le mécanisme fondamental pour lancer un processus externe. fork crée le processus enfant, execvp le remplace par la commande demandée et waitpid permet au processus parent d'attendre la fin de l'exécution pour le mode **foreground**.
chdir	Commande intégrée cd.	**Nécessité :** Le changement de répertoire (cd) est une **commande interne** (*built-in*) qui doit affecter le répertoire de travail du shell parent. chdir est la seule fonction qui modifie l'état du processus appelant, rendant son exécution via fork/execvp inappropriée.

pipe / dup2 / close	Pipelines et redirections.	Ces fonctions sont la brique de base pour le contrôle de flux. pipe crée le canal de communication, et dup2 est utilisé dans le processus fils pour rediriger stdout vers l'entrée du pipe suivant (ou le fichier de sortie, pour la redirection >).
freopen (pour >)	Redirection de sortie vers un fichier.	Utilisation simple et efficace pour le cas simple (commande > fichier). Exécuté dans le processus enfant (post-fork), il modifie le flux stdout du processus fils sans affecter le shell parent.
waitpid avec WNOHANG	Suivi des jobs en arrière-plan.	Permet de vérifier l'état des processus enfants lancés en arrière-plan (&) sans bloquer le shell interactif. La vérification régulière est une solution simple choisie à la place des signaux (SIGCHLD), qui sont plus complexes à gérer.

3. Justification de la Représentation des Données

Structure command pour l'Exécution

Le choix d'une structure interne avec un tableau d'arguments (char *arg[]) et le nom de la commande (char *name) est directement justifié par la signature de la fonction d'exécution : **execvp(name, arg)**. Cette structure garantit que les données du parsing sont prêtes à être transmises directement au noyau, simplifiant ainsi la phase d'exécution.

Liste Chaînée pour les Commandes Multiples

L'utilisation d'une **liste chaînée** pour enchaîner les structures command permet de gérer facilement :

- **Opérateurs Séquentiels** : Chaque nœud stocke le séparateur (&, |, >, ;) qui le *suit*, permettant à l'orchestrateur d'exec_cmd.c de parcourir la liste et d'exécuter l'action appropriée pour chaque maillon.
- **Flexibilité** : La structure peut s'étendre naturellement pour gérer les chaînes de commandes (;) et les chaînes de tuyaux (|).