Design Document – Mini Bash Shell

1. Purpose and Requirements

The purpose of this project is to implement a mini shell in C that simulates basic Bash functionality. The shell allows the user to type commands, execute internal commands like exit and cd, and run external commands from $HOME or /bin. It interacts directly with the kernel via system calls, without using higher-level abstractions like system () .

Implemented requirements:

- Prompt display (bash-mini$)
- Input reading and tokenization
- Internal commands: exit and cd
- External commands with process management (fork(), execv())
- Error handling for unknown commands and execution failures

2. System Calls

- **write() –** prints the prompt
- **getline() –** reads input from the user
- **strtok() –** splits input into tokens (argv[])
- **strcmp() –** detects internal commands
- **chdir() –** changes directory for cd
- **getenv() –** retrieves $HOME
- **access() –** checks if a file is executable
- **fork() –** creates a child process
- **execv() –** runs external commands
- **waitpid() –** waits for child completion
- **perror() / fprintf()–** reports errors

using minimal system calls per command for efficient execution.

3. **Flow Logic**
    - Start infinite loop.
    - Print prompt.
    - Read input using getline()
    - Remove trailing newline. Skip empty input.
    - Parse input into tokens (argv[]) using strtok().
    - Check internal commands:
        - exit -> exit shell
        - cd -> change directory using chdir()
    - For external commands:
        - Fork child process (fork())
        - Check $HOME/command, then /bin/command. If executable, run with execv(). If not found, print [command]: Unknown Command.
        - Parent: wait for child completion (waitpid()), print exit code.
    - Repeat loop.

4. **Efficiency**
    - Only essential system calls are used per command: 1 fork(), up to 2 access(), 1 execv(), 1 waitpid()
    - No unnecessary copying or threads
    - Internal commands and empty input skip extra calls

5. **Error Handeling**
    - Unknown commands: Unknown Command
    - Execution errors: printed via perror()
    - Missing cd argument: prints cd: missing argumentt
    - Fork failures: prints fork failed
    - Input EOF: exits cleanly