USER DOCUMENTATION
Mini UNIX Shell (AP_SHELL)

Assignment: Lab 1
Language: C (GCC Compiler)

Team: Abhinav Gupta (2025MCS2089)
Pratik Chaudhari (2025MCS2096)

---

1. Introduction

AP_SHELL is our UNIX-style command-line interpreter designed to provide an interface for interacting with the operating system. It supports standard UNIX features, such as I/O redirection, pipelining, and background execution, and also supports features like command history and logical operator(&&).

2. Installation and Compilation

Prerequisites

- A Unix-based operating system.
- GCC (GNU Compiler Collection).
- Make utility.

Compilation Steps

This project includes a Makefile for automated compilation. To compile the shell, go to the source directory in your terminal and run the command:

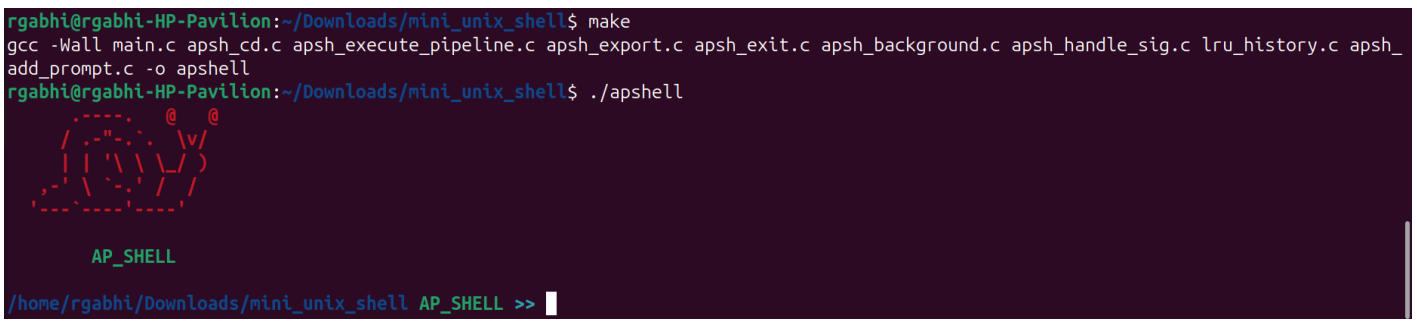
> Bash: make

3. Getting Started

Starting the Shell

Once the code is compiled, run:

> Bash: ./apshell



4. Features and Usage Guide

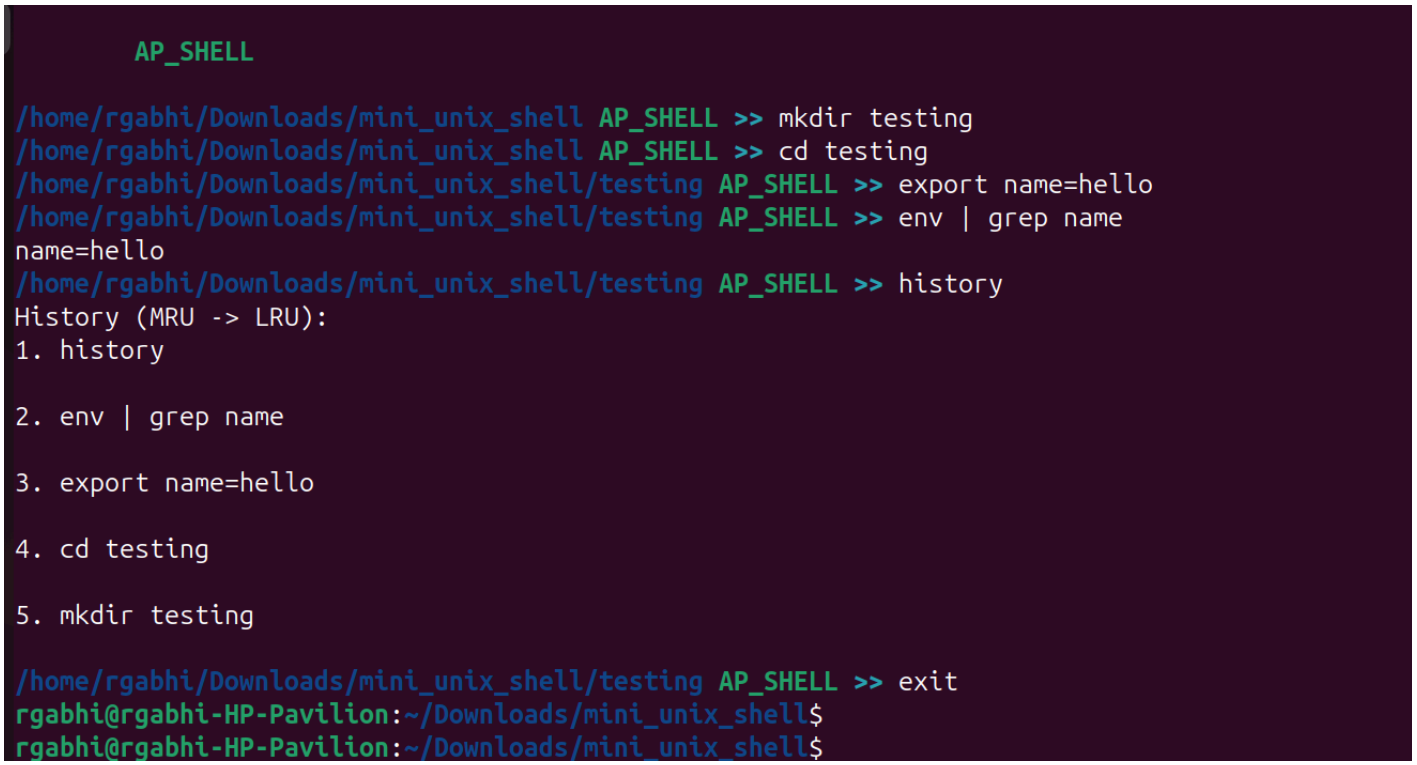
4.1. Basic Command Execution

You can run standard commands like **ls**, **grep**, and **sleep** just like you would in a normal terminal. AP_SHELL automatically finds them for you using your system's PATH, so you don't have to type out the full file location.

4.2. Built-in Commands

AP_SHELL includes internal commands that execute directly within the shell process rather than creating a new child process.

- **exit :**
  - **Description:** Terminates the current shell session and closes the apshell program.
  - **Usage:** Type exit.
  - **Behavior:** The loop checks the variable `status` at the start of every cycle. As long as `status` is 1 (true), it keeps running.
- **cd <directory> :**
  - **Description:** Changes the current working directory.
  - **Usage:** cd /path/to/directory or cd .. to go up one level.
  - **Error Handling:** Displays an error if the directory does not exist or if no argument is provided.
- **export KEY=VALUE :**
  - **Description:** Sets environment variables that persist for the session and are inherited by child processes.
  - **Usage:** export MY_VAR=hello
- **history :**
  - **Description:** Displays a list of the most recently executed commands (up to 100).
  - **Usage:** history



4.3. I/O Redirection

You can redirect the standard input and output of commands using < and >.

- **Output Redirection (>):** Redirects the output of a command to a file. If the file exists, it is overwritten; otherwise, it is created.
    - **Usage:** ls -l > output.txt
- **Input Redirection (<):** Give the content of a file as input to a command.
    - **Usage:** tr a-z A-Z < input.txt

```
       AP_SHELL

/home/rgabhi/Downloads/mini_unix_shell AP_SHELL >>
/home/rgabhi/Downloads/mini_unix_shell AP_SHELL >>
/home/rgabhi/Downloads/mini_unix_shell AP_SHELL >> tr a-z A-Z < input.txt
ECHO HELLO/home/rgabhi/Downloads/mini_unix_shell AP_SHELL >> ls -l > output.txt
/home/rgabhi/Downloads/mini_unix_shell AP_SHELL >> cat output.txt
total 80
-rw-rw-r-- 1 rgabhi rgabhi   318 Dec  9 00:22 apsh_add_prompt.c
-rw-rw-r-- 1 rgabhi rgabhi   486 Dec  8 21:55 apsh_background.c
-rw-rw-r-- 1 rgabhi rgabhi   341 Dec  8 18:11 apsh_cd.c
-rwxrwxr-x 1 rgabhi rgabhi 22424 Dec  9 04:18 apshell
-rw-rw-r-- 1 rgabhi rgabhi  1209 Dec  9 00:22 apsh_execute_pipeline.c
-rw-rw-r-- 1 rgabhi rgabhi    91 Dec  8 17:43 apsh_exit.c
-rw-rw-r-- 1 rgabhi rgabhi  1300 Dec  8 21:55 apsh_export.c
-rw-rw-r-- 1 rgabhi rgabhi   557 Dec  9 00:22 apsh_handle_sig.c
-rw-rw-r-- 1 rgabhi rgabhi  1080 Dec  9 00:22 apsh_module.h
```

4.4. Pipelining (|)

AP_SHELL supports single-stage pipelining, allowing the output of one command to serve immediately as the input to another.

- ***Usage*:** cmd1 | cmd2
- ***Example*:** ls | wc -l
- ***Mechanism*:** The shell creates a pipe, forks two child processes, and connects the stdout of the first to the stdin of the second.

```
/home/rgabhi/Downloads/mini_unix_shell AP_SHELL >> ls | wc -l
15
/home/rgabhi/Downloads/mini_unix_shell AP_SHELL >> ls | grep .c
apsh_add_prompt.c
apsh_background.c
apsh_cd.c
apsh_execute_pipeline.c
apsh_exit.c
apsh_export.c
apsh_handle_sig.c
lru_history.c
main.c
/home/rgabhi/Downloads/mini_unix_shell AP_SHELL >>
```

5. Signal Handling

The shell handles standard interrupts:

- **Ctrl-C (SIGINT):** Pressing Ctrl-C does not terminate the shell. Instead, it interrupts the currently running foreground process (if any) and reprints the prompt on a new line, ensuring the shell remains active.

6. Technical Specifications

- **History size:** Currently history cache is implemented to store 100 commands in a session.
- **Pipeline depth:** The current version of shell supports input with single pipeline only. (one pipe between two commands).
- **Input buffer:** The tokenization logic handles standard whitespace delimiters (like spaces, tabs, newlines).
- **Memory management:** The shell uses dynamic memory allocation for tokenizing input and managing the history-linked list to ensure efficient memory usage.

7. Troubleshooting

- **"apsh: allocation error"**: Occurs if the system runs out of memory during command parsing.
- **"apsh: command not found"**: The command is not a built-in and could not be found in the system PATH.
- **"no output/input file given"**: Ensure you provide a filename immediately after > or <.