

Posix Shell

Languages Allowed: C/C++

Task : Develop a working POSIX compatible shell with a subset of feature support of your default shell. Apart from basic working, required features to be implemented are listed below.

Pre-requisites

- Multiprocess approach using **fork** syscall.
- **Pipes** for execution chaining and I/O redirection.
- **Signals** and **handlers** (minimal usage).
- **TRIE** data structure.
- Environment variables and export.
- Understanding of `".bashrc"` and `".profile"`.
- Following functions will be of help - `fork`, `exec`, `execvp`, `wait`, `waitpid`, `kill`, `dup`, `pipe`, `strncmp`, `strlen`, `malloc`, `free`, `getcwd`, `chdir`, `open`, `close`, `readline`, `gets`, `fgets`, `getchar`, `signal`
- Files permissions and ownership in *nix or derived systems.
- **alarm()** posix function.
- Prompt and it's meaning for each field, E.g. `user@ubuntu:~$`

DONT 's

1. No menu driven programs.
2. Do not use `system()` function.
3. You do not have to make any commands yourself. See `/bin` directory.
4. No support for shell scripting is needed.
5. Space handling for filenames is not mandatory, but if it helps while calling functions then do it.
6. This is not a design competition so nothing fancy is needed; for prompt string and format of displaying output from your process, keep it clean. Functionality is important.
7. **Avoid naming anything as test or something which clashes with internally existing tools or keywords.**

Syntax

1. IO redirection with `'>>'` and `'>'` will be done for one source and one destination only.
2. For `'|'`, a minimum 3 programs must be in pipe. E.g. `cat file1.txt | sort | uniq > save.txt`. Do not hard-code for 3 programs.
3. Prompt must change for distinguishing users(`$`)@hostname and root(`#`)@hostname and also while searching in extended features.

Implementation Requirements

Shell Internals

1. Your program must be an input driven independent executable, no menu options are to be provided. It can take n number of strings in the format of a command following with its parameters itself.
2. For interactive processes it needs to be a waiting process.
3. For background command execution, '&' can be passed as last token of current command. You need to maintain a list for background processes and can be made into a foreground process by using 'fg' command. You can implement your own data structure or logic for it.
4. You must maintain a configuration file which your program reads on startup and sets the environment accordingly. Like your own ".bashrc".
5. Support for these initialization variable are to be provided - **PATH, HOME, USER, HOSTNAME, PS1**.
6. Association of ~ with **HOME** variable.
7. Look of **prompt via PS1** is to be handled.
8. Alias support for commands. E.g. **alias ll='ls -l'**.
9. Handle support for - **\$\$, \$? Similar to shell**.

Extended Features (NOT OPTIONAL)

1. Recording as script command, where a user can initiate recording of all input and output to a file, this happens in the background. It will be invoked as **record start** and **record stop**.
2. History as buffer - **searchable via TRIE**. This is a list of all commands executed ever. This can have a cap limit which can be configured via ".myrc" file in **HISTSIZE variable**. It will be invoked as **history**.
3. TRIE data structure for tab completion support is needed. List of directories to be traversed on startup which are mentioned in **PATH** variable for populating all command names into the structure.
4. Understanding of environment variables and **export**.
5. A reminder(text based) or alarm(just notification) feature in shell is to be implemented. Maintain a file and in memory structure with **time::data** pair to interrupt and show alarm/reminders or on startup show the missed ones.
6. Set default applications via .myrc file. This requires extension matching/mapping to your preferred application. Handle its I/O and parameter passing on executing.
 - a. E.g. **MEDIA=['vlc', 'vlc path', '.mp4']** in config file.
So on running - open same.mp4 , it must invoke vlc.
!! Mac users be careful for collision of open keyword in your shell.