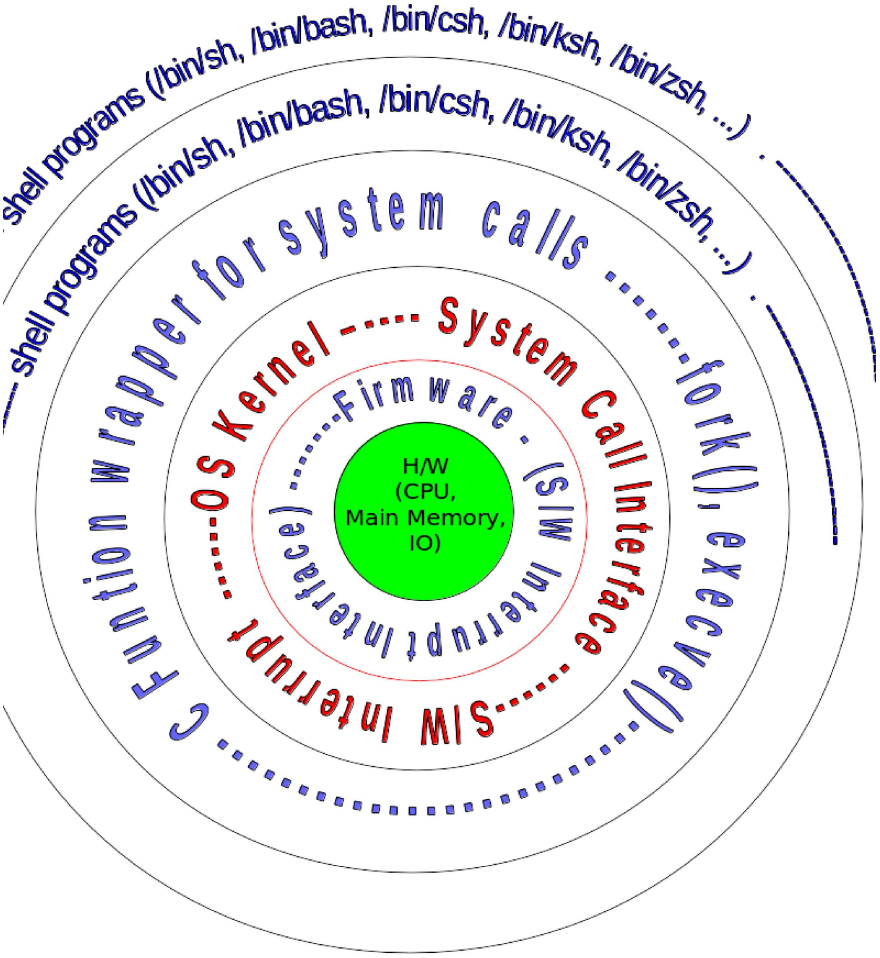
Operating Systems Laboratory (2024-25)

Implementation of a primitive shell

Opened: Tuesday, 8 April 2025, 9:55 AM
Due: Tuesday, 8 April 2025, 12:40 PM

Mark as done

A ***shell*** is an interface program between the **Operating System** and a **User**. That is, a shell takes commands from the user and executes those commands through invocation of appropriate system calls as depicted in the following figure.



You are familiar with Bourne Shell (/bin/sh) or Bourne Again Shell (/bin/bash). You may refer to the manual (guide) on bash (available with this assignment) to have an idea about the language (shell script) in which user may issue commands to it. The present assignment is to construct a trivial shell (say, ***mysh***).

In general, the commands a shell can execute are of 2 types - **internal commands** and **external commands**. Commands like ***cd***, ***pwd***, etc., are internal commands. Please note that the shell does not need any external executable file for the internal commands. Whereas, external commands (eg., **ls**, **vi**, ...) are essentially executable files and are managed by the shell by first creating a child process (***fork()***), and then making the child process execute (execve()) the executable file.

The trivial shell should support internal commands like ***cd, pwd, clear, exit***, etc. It should support any external command with or without command line arguments. For example, myshell should be able to execute both "***ls***" as well as "***ls -lt***" as external commands. Multiple commands can be can be connected by "***;***" (semicolon, sequential execution), "**|**" (pipe), "***&&***" (logical and), "**||**" (logical or). That is "***command1 ; commnd2***" or "***command1 && commnd2***" or "***command1 || commnd2***" are all valid commands for "***mysh***".

You may refer to the guides (associated herewith) for the semantics (meaning) of these operators.

The algorithm for "**mysh**" should be quite straight forward.

do

show the shell prompt

read a line in the string variable cmd

parse cmd into subcommands /* in "command1 ; command2" command1 and command2 are subcommands */

parse the subcommands for command line arguments

if a subcommand is internal then do what is necessary for it. Else if there is an executable for the command then fork() and let the child process execute the executable
while (true);

Please note that "***mysh***" can be executed interactively, that is, upon execution it accepts commands from the ***stdin***. "***mysh***" can also be executed in batch mode (for example, ***mysh file.sh***) when **mysh** will read the commands from the file "***file.sh***" and execute them.

To enhance your trivial shell (**mysh**), you may (optional) attempt to support the following features.

- IO redirection (Eg., ***./a.out > report.txt*** saves the output (***stdout***) of ***a.out*** to the file ***report.txt***)
- support for environment variable like "**PATH**".
- autocompletion of commands
- navigating through the old commands executed by mysh (so that the user need not retype them while wishes to execute them again.

abs-guide.pdf	8 April 2025, 9:13 AM
Bash-Beginners-Guide.pdf	8 April 2025, 9:13 AM
bash.pdf	8 April 2025, 9:13 AM

Submission status

Attempt number	This is attempt 1.
Submission status	No attempt
Grading status	Not graded
Time remaining	2 hours 38 mins
Last modified	-
Submission comments	Comments (0)

Add submission

You have not made a submission yet.

◄ Assignment: Continuation of previous class's assignment on filesystem (myfsv2)

Jump to...

Final submission for earlier assignment titled "Assignment on filesystem" ►

OSLabJan2025

Data retention summary
Get the mobile app