CS242
System Programming Lab No. - 5

Objective

In this lab you will explore the workings of a command-line interface (CLI) and build a simple shell in stages.

Exercise

The CLI will be a simple implementation of the commonly used command shells such as sh, bash, csh, and tcsh.

The assignment is split-up in *tasks* to implement separate features of our shell program.

Consult the manual pages to understand the following system calls and library functions:

- `int chdir(const char *path)`
- `int execvp(const char *file, char *const argv[])`
- `void exit(int status)`
- `pid_t fork(void)`
- `char *getcwd(char *buf, size_t size)`
- `char getenv(const char *name)`
- `void perror(const char *string)`
- `int setenv(const char *name, const char *value, int overwrite)`
- `sig_t signal(int sig, sig_t func)`
- `pid_t wait(int *status)`
- `pid_t waitpid(pid_t wpid, int *status, int options)`

Task 1 - Adding Built-in Commands

Create a new file `shell.c` . This will be our main shell program that will execute commands from a command line. Our shell should interpret the following built-in commands:

- `cd`: changes the current working directory
- `pwd`: prints the current working directory
- `echo`: prints a message and the values of environment variables
- `exit`: terminates the shell
- `env`: prints the current values of the environment variables
- `setenv`: sets an environment variable

For example:

```
1> echo hello world
hello world
2> cd test
3> pwd
/home/faculy/engelen/test
3> setenv greeting hello
4> echo $greeting $OSTYPE
hello linux
5> env
...
greeting=hello
...
```

```
6> exit
```

A tutorial on how to write a shell is given in the following url:
http://stephen-brennan.com/2015/01/16/write-a-shell-in-c/

Deliverables

Create a pdf document containing the output of your program. Next create an archive file by the name <roll number>.tar that contains the pdf document and the C program.
Mail the document to cs242@iitp.ac.in with subject "Lab <no>".

---000---