The Shell and System Calls

Created By

Alex McCulloch, Edgar Sanchez, Luis Chaparro, and Russell Price

Design Overview

A shell implementation that can be run in interactive or batch mode.

Complete Specification

To compile the program automatically, execute:

```
$ make
```

To compile the program manually, execute:

```
$ gcc -std=c99 -Wall -o shell main.c
```

To run the program, execute:

```
$ ./shell [batchFile]
```

- **batchFile**: an optional argument (indicated by square brackets as above). If present, the shell will read each line of the batchFile for commands to be executed and will exit when it reaches a `quit` command, the end of the file, or if you enter '`Ctrl-D`'. If not present, the shell will run in interactive mode by printing a prompt to the user at stdout and reading the command stdin.

For example, if you run the program as:

```
$ ./shell /home/mat0299/csce3600/batchFile
```

then the shell will read commands from `/home/mat0299/csce3600/batchFile` until it sees the `quit` command, it reaches the end of file, or you type '`Ctrl-D`'/'`Ctrl-C`'.

Alternatively, if you run the program as:

```
$ ./shell
```

then the shell will display an interactive prompt:

```
$ ./shell
```

```
-----[ INTERACTIVE MODE ]-----
--> Type "help" to view commands

prompt>
```

This interactive shell will accept commands and display output until you `quit` the program.

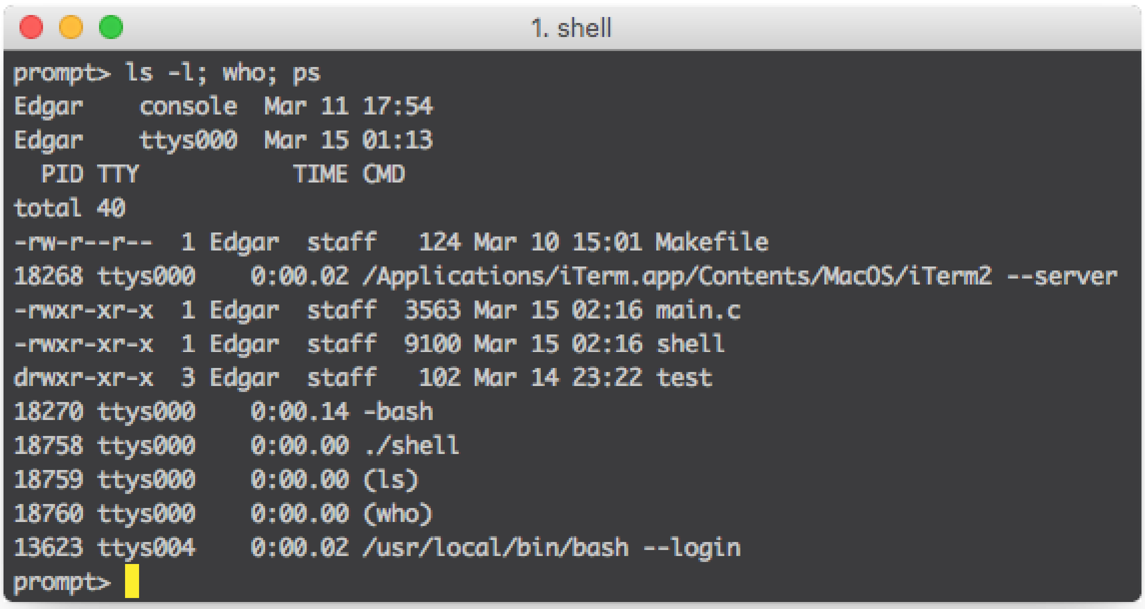
To view a list of internal commands, type `help`:

```
$ prompt> help
/--------[ HELP: LIST OF INTERNAL COMMANDS ]-------\
| history       - prints list of commands entered  |
| prompt        - sets custom prompt string        |
| customize     - sets customized shell options    |
| path          - sets PATH directory              |
| cd            - change current directory         |
| quit OR exit  - exits shell program              |
\--------------------------------------------------/
```

You may execute multiple commands in interactive mode or in batch files by separating them with a "`;`". Such as:

```
prompt> ls -l; who; ps
```

which would output something similar to this:



```
prompt> ls -l; who; ps
Edgar    console  Mar 11 17:54
Edgar    ttys000  Mar 15 01:13
  PID TTY           TIME CMD
total 40
-rw-r--r--  1 Edgar  staff   124 Mar 10 15:01 Makefile
18268 ttys000    0:00.02 /Applications/iTerm.app/Contents/MacOS/iTerm2 --server
-rwxr-xr-x  1 Edgar  staff  3563 Mar 15 02:16 main.c
-rwxr-xr-x  1 Edgar  staff  9100 Mar 15 02:16 shell
drwxr-xr-x  3 Edgar  staff   102 Mar 14 23:22 test
18270 ttys000    0:00.14 -bash
18758 ttys000    0:00.00 ./shell
18759 ttys000    0:00.00 (ls)
18760 ttys000    0:00.00 (who)
13623 ttys004    0:00.02 /usr/local/bin/bash --login
prompt>
```

Commands are run simultaneously, which means that some of the output may be intermixed. This is by design.

To exit the shell, simply type `quit`. This will cause the shell to stop processing any more commands and will exit.

To remove executable and object files, execute:

```
$ make clean
```

Directory Structure

Directory	Description
/docs	Program documentation.
/src	Source(.c), header(.h), and make files.
/src/test	Testing/debugging files.
README.md	Basic documentation and information.

Known Bugs or Problems

View current issues.