

A Tiny Unix Shell (tsh) – Part V

Introduction

This is the fifth and final piece of the project on process control and signaling. In this assignment you will extend your shell to implement job control. This assignment must be completed *individually*.

Job Control

Unix shells support the notion of *job control*, which allows users to move jobs back and forth between background and foreground, and to change the process state (running, stopped, or terminated) of the processes in a job.

Unix shells also provide various built-in commands that support job control. For example:

1. `jobs`: List the running and stopped background jobs.
2. `bg <job>`: Change a stopped background job to a running background job.
3. `fg <job>`: Change a stopped or running background job to a running in the foreground job.
4. `kill <job>`: Terminate a job.

What to do

Expand your `builin_cmd` routine to support three additional built-in commands:

1. The `bg <job>` command restarts `<job>` by sending it a `SIGCONT` signal, and then runs it in the background. The `<job>` argument can be either a PID, or a JID preceded by `%`:

```
tsh> bg 17774    /* Argument is PID */
tsh> bg %2      /* Argument is JID */
```

The functions `atoi` and `isdigit` will come in handy when parsing the second argument of this command. Use the Unix manual pages to learn about these functions.

2. The `fg <job>` command restarts `<job>` by sending it a `SIGCONT` signal, and then runs it in the foreground. The `<job>` argument can be either a PID or a JID. The shell must wait for as long as a FG process exists, before printing a new prompt for the user.
3. The `kill <job>` command terminates `<job>` by sending it a `SIGKILL` signal. The `<job>` argument can be either a PID or a JID.

Make sure that a job with the indicated PID or JID exists, before sending it a signal. Feel free to define new functions, as you think necessary. Also make sure that you clear your account of all running or stopped jobs. Use the piped command

```
ps -ef | grep youraccountname
```

to find out the PID of your jobs, then terminate them with the `kill` shell command.

Evaluation

Your score will be computed out of a maximum of 50 points based on the following distribution:

40 Correctness.

10 Style points. We expect you to have good comments and to check for errors.

Hand In Instructions

Hand in a printed copy of the following:

1. Source code of `eval.c`.
2. A sample output of your code.
3. A short README file explaining any bugs and deviations from the assignment specifications.

Make sure you have included your name and your Unix login name in the README file. Leave your source code in your Unix account. Good luck!