National University of Singapore
School of Computing

CS2106 **Laboratory 8** Semester 1 11/12

Deadline

13 November, 2011, Sunday, 10:00pm.

Platform

This exercise should be done on SunFire.

If you want to use your own Linux machines to work on the assignment, make sure you have the GNU Readline library installed.

If you want to use Linux machines to work on the assignment, make sure to test it on SunFire before submission (the OS and system calls have slightly different behaviour).

Submission

Your working directory should be named `lab08-A000000X` where `A000000X` is your matriculation number. Create a tar ball named `lab08-A000000X.tar` containing your version of the source code by:

```
tar cvf lab08-A000000X.tar lab08-A000000X/*
```

and submit the tar file into the IVLE workbin named Lab 8.

Failure to follow the instructions above would lead to a 3 mark penalty.

Marking Scheme

This lab exercise is worth 10 marks. 3 marks will be deducted if your tar file or directory are not named properly. For late submission, we deduct 0.1 marks for every minute late after the deadline.

Tips

`man` and Google are your best friends. Use them effectively to learn about the functions and system calls used in this lab exercise.

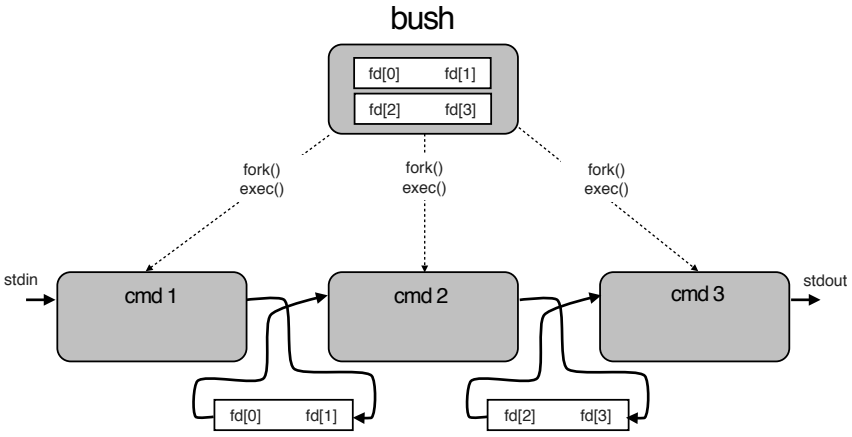

Figure 1: Pipes

In this lab exercise, you will extend the `bush` shell, which you have developed in Lab 4, to support pipes. To simplify the assignments, however, several features from `bush` have been removed. Specifically, background execution is no longer supported, and the "no child left behind" policy is no longer needed.

Pipes work as follows. Users can type in a chain of commands, separated by the `|` character. For example,

```
ps -eaf | grep user1 | wc -l
```

The commands are then executed in sequence per normal, but instead of reading from stdin and writing to stdout, the output of one command is redirected as the input to the next command. In the example above, the output of `ps` is redirected into the input of `grep`; the output of `grep` is redirected into the input of `wc`.

There are several ways such a sequence of pipes can be constructed. In the given skeleton code, we adopt the interative approach, where the parent process (`bush`) setup the pipes, and `fork` multiple times, once for each command issued. A child process, through the inherited pipes from the parent, communicate with its siblings.

Figure 1 illustrates the relationship between the parent, the children, and the pipes.

Your task in this lab is the following: using a combination of `pipe`, and `dup2`, and `close` system calls, setup a chain of pipes among the child processes, so that output from one command is redirected to the input of the next command through pipes created using the `pipe` system call.

Your code should be able to work for a chain longer than two commands. It should handle errors due to execvp() properly.

A code skeleton has been given to you. You may download it from the URL `http://www.comp.nus.edu.sg/~cs2106/lab08-A000000X.tar`. `bush2.c` is the only file you need to edit. Do NOT edit `parse.c` or create any other additional files.

THE END