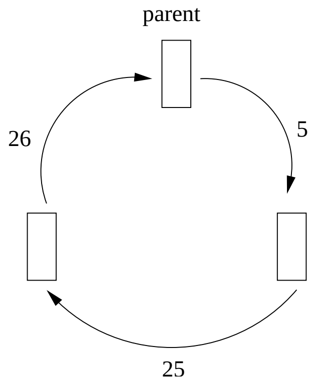
Lab #8: Pipes

Overview

The purpose of this lab is to introduce the use of pipes.

Part 1: `square_plus1`

Write a program, named `square_plus1`, that reads integers from the keyboard until EOF. This program must create two child processes and connect the children and parent in a ring using pipes. For example,



Once properly connected, the parent will read an integer from the keyboard and send that value to the child via a pipe. The first child will square the received value and forward the result to the next child. The second child will add one to the received value and return it to the parent. Finally, the parent will print the new value to the screen and read another integer.

When EOF is reached, the parent must close its pipes and wait for its children to properly exit.

Part 2: `pipeit`

Write a program that takes, as command-line arguments, two other program `and` their command-line arguments; the two programs, with their command-line arguments, are to be separated by a `;`[1]. Your program should spawn two processes to run the specified programs with a pipe linking the `stdout` of the first to the `stdin` of the second.

For instance, the following would send the output of `ls -l` to `more`.

```
pipeit ls -l \; more
```

Note, your program should support execution of any programs found in your path.

Demonstration

Demonstrate your working programs to the instructor.

[1] Type \; in the shell to prevent the shell from processing the ;.