# Lab 6: IPC Using Files and Pipes

# Motivation

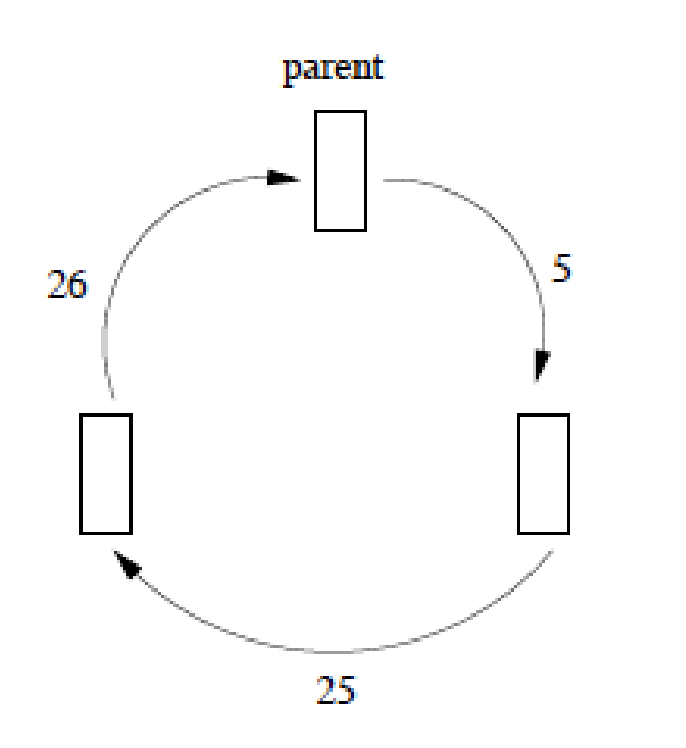
Modern programs communicate with other programs. It is rare to find a program that interacts only with the user or only with files. For better or worse, many programs that we use on a regular basis will communicate with other computers (servers) or with other processes (mobile apps commonly share services). This lab requires that you explore a bit of that communication.

### Task 1: Pipes

Write a program, named uppercase that reads strings from the keyboard. This program must save the input into a file and then using a child process should convert every lowercase letter of the string to uppercase. Then the parent process should print the string to the screen.

### Task 2: Pipes

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 (as shown in the image below).



Once properly connected, the parent will read an integer from the keyboard and send that value to one child via a pipe. That child (the first) will square the received value and forward the result to the other 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.

### Submission

Submit your source code with a makefile that creates the executable for task 1 and task 2 on CANVAS. Make sure that the makefile has a rule to clean all the executables and files.