Michael Smith

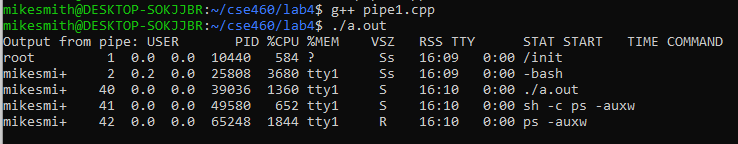
Lab 4

Total Points: 20 Points

1 February 2018

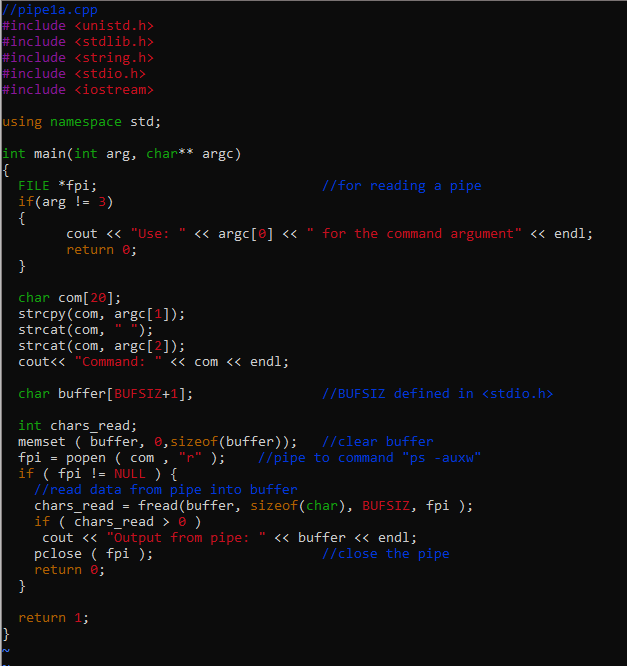
2. Process Pipes

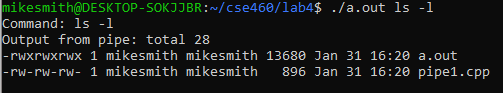
Pipe1.cpp output



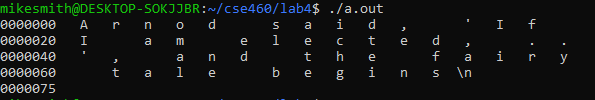
This program creates a "file" variable to read from the popen which creates a pipe, forks and then creates a shell. Then runs the command ps –auxw with the modifier –r which makes the pipe read the output from the shell. The program reads the shell output and stores them in a character array and prints it out on the terminal, which is what we see in the screen capture.

Pipe1a.cpp and output





Pipe2.cpp output



The Program sends the text out of the pipe when it runs the od command with the –c modifier. Printing out the statement shown above.

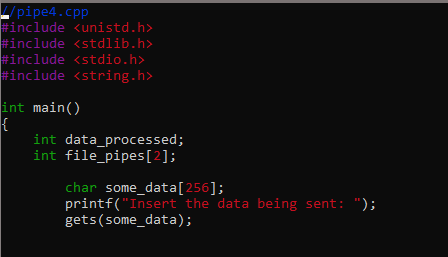
3. Pipe3.cpp Output



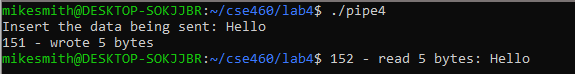
This code will display the number of bytes sent to the pipe and then displays the number of bytes and the data sent. When the pipe is created with the fd argument, it makes the fd handle the read end and the write end for the pipe. The write will display the first line in the output, while the read will output the second line of the output.

4. Parent and Child Processes

Had to modify the code in pipe4 to read out to the terminal to prompt user to send data, then store the data in the array instead of "123." Here is the modified lines

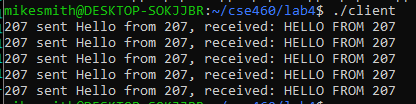


Output

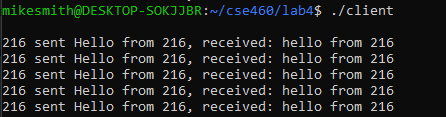


5. Special Pipes

Code test before modification

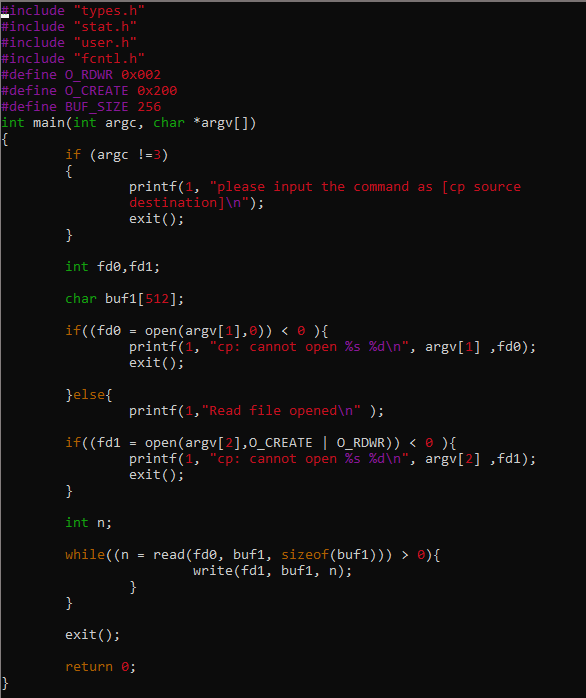


Modifying the code in server.cpp to make the output into lower case instead of upper only requires the change of the line 49 code. Change the code from "\*tmp\_char\_ptr = toupper(\*tmp\_char\_ptr);" to "\*tmp\_char\_ptr = tolower(\*tmp\_char\_ptr);". Then the output becomes



6. XV6 Coding

Here is the code for the cp



Here is the output for the cp test

