Srujan Vepuri

Svepuri@albany.edu

Using c programming language to make two-way communication with process

Project 1

Duplex pipes

Contents

[System documentation 2](#_Toc114783929)

[A high-level data flow diagram for the system 2](#_Toc114783930)

[A list or routines and their brief description 2](#_Toc114783931)

[Implementation details 2](#_Toc114783932)

[Test documentation 4](#_Toc114783933)

[How you tested your program 4](#_Toc114783934)

[My test sets 4](#_Toc114783935)

[User documentation 7](#_Toc114783936)

[How to run your program 7](#_Toc114783937)

[Describe any parameters (if any) 7](#_Toc114783938)

# System documentation

## A high-level data flow diagram for the system

Diagram

Description automatically generated

## A list or routines and their brief description

1. Frist thing data come from editSource.txt
2. Second its child read the data than write to pipe 1
3. Third the parent reads from pipe 1 than make and write to “theCount.txt” and pipe2
4. Then the child read from pipe 2 and display out to screen and what ever in the file “theCount.txt”

## Implementation details

* First make 2 pipes and initial an int value (later we use for wait ()) and another called pid for fork if any errors we return 1 not move forward
* Then we will set pid = fork ();
* If current pid ==0 than we are in child process
  + We read all the data from editSoruce.txt
  + Then we write to pipe 1
  + We print out pid using getpid () and parent pid using getppid()
  + Now we wait till our parent write back something to our pipe 2
  + Once pipe 2 has something than we will read from pipe2[0] than display to the screen using printf()
* If current pid > 0 than we are in parent process
  + We wait till we can read something for pipe 1 using pipe1[0]
  + Once we have data from pipe 1 than we go head and calculate number of charters, lines, words
  + For charters we simple count how many bytes we read
  + For words we count how many time our char value == ‘ ’
  + For number of lines, we count how many time our char value == ‘\n’
  + Once we have number of charters, words, and lines than we will create theCount.txt
  + Then write to theCount.txt the number of charters, number of words, and number of lines
  + Then we will write the same information back to pipe 2
* After parent process is done it will wait for the child finish up before ending the programing.

# Test documentation

## How you tested your program

I broke down my test into Fork, pipes, inputfile.txt, outputfile.txt

1. Fork:
   1. I used prints statements to test the parent and child, I also test to make sure I do not have zombie child. I add wait (), so my parent will wait for the child to finish.
2. Pipes:
   1. I test pipe1 with writing all the content of editSource.txt into my pipe1[1] than read it content using pipe1[0].
   2. I also tested my pipe2 using but in other process just writing random data and reading it off in other process.
   3. Then later also make sure that child uses pipe1 to write and pipe2 to read.
   4. For parent it uses pipe1 to read and pipe2 to write.
3. Inputfile.txt
   1. I test my inputfile.txt by adding more paragraphs and more lines.
4. Outputfile.txt
   1. I test my outputfile.txt by deleting Outputflie.txt form our previous runs. I also add random data to the file to make sure it doesn't append data. It should overwrite data and only have the latest run data.

## My test sets

1. Source code

represents the part of

process that

contains the programming

language itself. You may

use a text editor to

write your source code file.  
A screenshot of a computer

Description automatically generated with medium confidence

1. I wanna take you somewhere so you know I care But it's so cold and I don't know where I brought you daffodils on a pretty string   
     
   But they won't flower like they did last spring And I wanna kiss you, make you feel alright I'm just so tired to share my nights I wanna cry and I wanna love But all my tears have been used up  
      
   On another love, another love All my tears have been used up On another love, another love All my tears have been used up On another love, another love All my tears have been used up, up  
     
   And if somebody hurts you, I wanna fight But my hands been broken one too many times So I'll use my voice, I'll be so fu\*\*\*ng rude Words they always win, but I know I'll lose  
   A screenshot of a computer

   Description automatically generated with medium confidence

# User documentation

## How to run your program

1. To run my program, first have “editSource.txt” file or change the read () in child process. In that text file please had some words, paragraphs, etc...
2. Then have Linux system, then install GCC.
3. Once install please open Linux system and type in command prompt “gcc newProducerConsumer.c -o p”
4. Than run “./p”
5. When successfully run you should see theCount.txt in current directory.

## Describe any parameters (if any)

None, u have pass through the command line

But you are expected to have editsoruce.txt in current directory.