List of contents

INTRODUCTION

The main goal of this homework is to make two processes communicate with each other via a file using Signals. We have the Chat.txt file which is empty file and our processes have the permission to read and write in this file.
Signals are the mechanism for handling asynchronous events. So they synchronous events. And one by one synchronization operation is done by using SIGUSR1 and SIGUSR2 signals.
I have used Linux Operating system, and the program is written in C programming language. The file Chat.txt is located in *tmp* directory which means that two different Linux user can use it.

LIBRARIES

signal.h : This function sets a function to handle signals it is a signal handler. This function causes signal sig to be generated
string.h : This library has several common functions for dealing with strings stored in arrays of characters. For example returning number of characters, comparing two strings and so on .
stdio.h : It stand for standard input and output. It contains many standard library functions for file input and output, such as printf and scanf .
stdlib.h : It includes functions involving memory allocation ,process control, conversions and so on.
unistd.**h :**Is the name of the header file that provides access to the POSIX operating system API. Not everything is defined in there but some definitions are done by references to the GNU C standard library headers (like stddef.h) which provide the type size_t and many more.

EVALUATION PROCEDURE

Firstly I defined a function called signal catcher which we use it to send the SIGUSR1 signal. Inside the main I control the signal if it occurs any error.
Now I opened the while loop and iterates infinitely , when it enters, the first thing it does is stopping in pause. After this we try to find the others process's id and send the signal to wake it up and start chatting.
I defined I variable flag of second process in bool type and at first is false. When the flag is false it opens the file and the file is empty it writes the id. Now when the process 2 comes it reads the file , the first line it contains the id of the first process, second process it stores the id which was inside the file. Process 2 now is available to send signal by id. After finding each others ids now is left only to take input from user and write it inside the file. When the other process comes reads what is inside and shows it in screen which has to be read from user. So they signal each other in synchronization.

USER MANUAL

First we have to open three terminals , two of them are used to run the program (make two processes) , y the commands ./pr1 and ./pr2 .In the third terminal we using it to give the signals SIGUSR1 to the first process and SIGUSR2 to the second process with kill command. After we signal the processes we can start chatting between the processes.
We can open one more terminal and we use this one to check the Chat.txt file which message is written last (this is additional).

CONCLUSION

So the challenge in this project was that the two processes has to know each others id but by using the file the problem is solved. To make two processes communicate with each other we can use many methods as lock file , locking file , pipes and so on which may be easier.