Assignment 1 – Code Explanation & Documentation

This C console program will allow the user to create n number of child

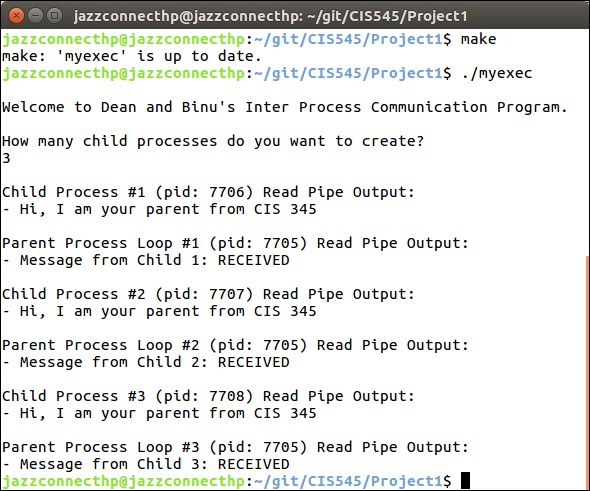
processes and exchange messages between parent process and child processes.

The source code of this program is in the file named "main.c". It comes with an accompanying Makefile in the project folder.

To compile and execute the program just type in the “*make*” command at the Terminal. Then run the executable binary by typing in “.*/myexec*”

If the Makefile is not present, you can compile the executable via gcc in Terminal with the following command:

*gcc -o myexec main.c -I. -lm*

**

The program will prompt the user to enter the number of child processes.

The parent-child process will have two Pipes to interact:

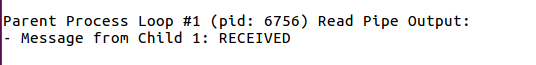
* one for the parent to send messages to child ,and
* one for the child to send message to its parent process.

Each child process reads and prints the message sent by the parent process:



The parent process also reads and prints the messages it receives from each

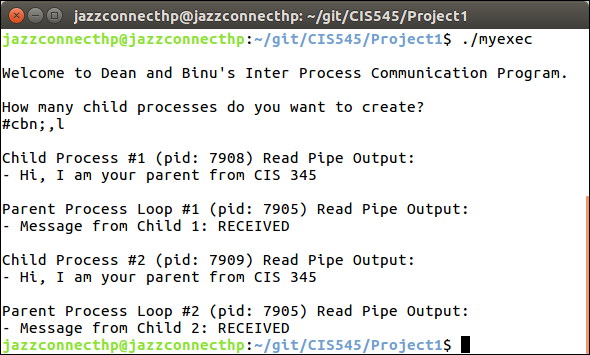
of it child process as below:



We have used dynamic memory allocation (calloc) to initailize and allocate memory for the file descriptors for the pipes.

*Int \*\*fd = calloc((size\_t) 2\*numChild, sizeof(int \*));*

We have used ‘*numchild*’ variable stores the number of child processes that the user enters and it also controls the “for” loop used in the program. If the user enters anything other than numbers the program will create 2 child processes as that has been set as the default.



String funtionalities have been used to build the message sent by the child process. Pasted below is the extract of the code that concatenates the strings into one message and stores in “*message2*” and then writes to the pipe.

*//How many digits in the integer i*

*int idigits = floor(log10 (abs(i+1))) + 1;*

….

*buffsize = strlen(childmsg1) + strlen(childmsg2) + idigits + 1;*

*char message2[buffsize];*

*char numi[idigits + 1];*

*sprintf(numi, "%d", (i+1));*

*strcpy(message2, childmsg1);*

*strcat(message2, numi);*

*strcat(message2, childmsg2);*

*//Child Process will write to Pipe2 and send child's message to Parent*

*if (write(fd[2\*i+1][1], message2, buffsize) == -1)*

*~~~~~~~~~~~~~~*