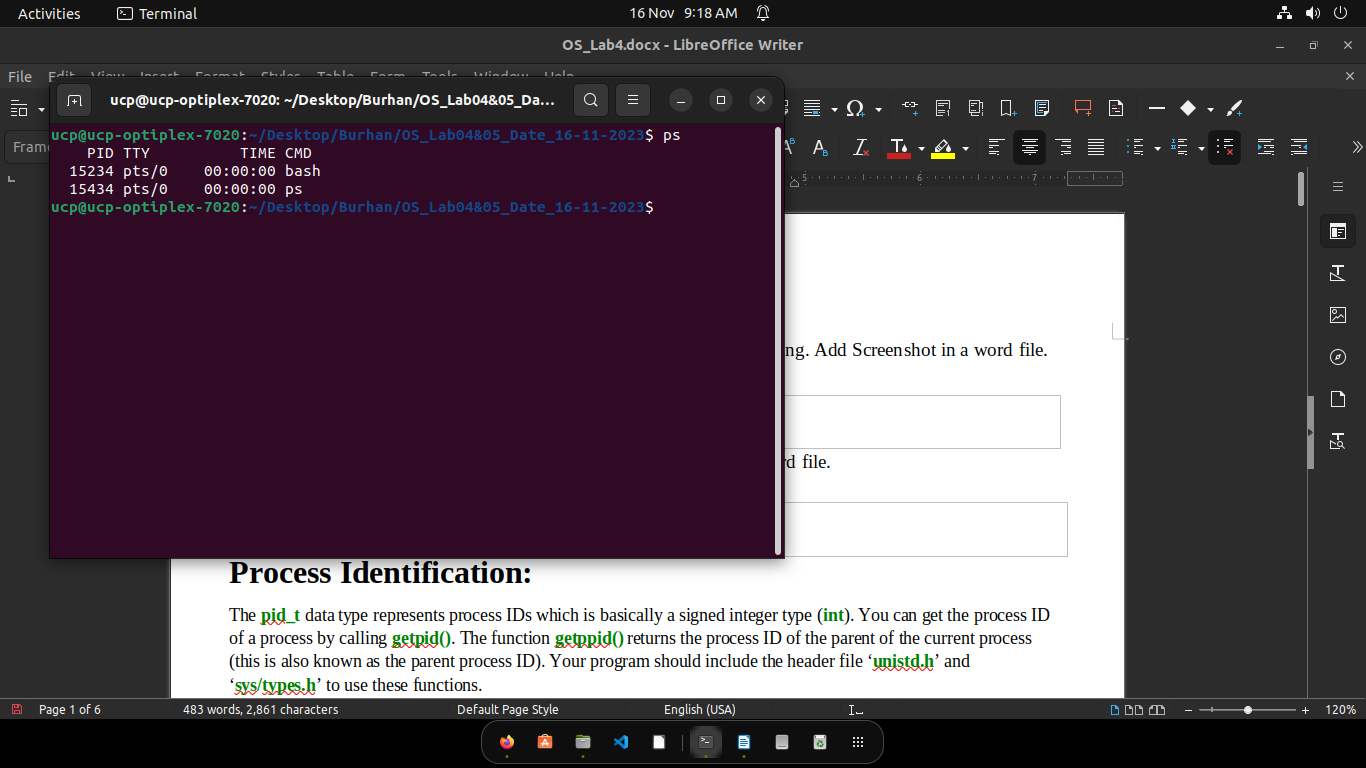
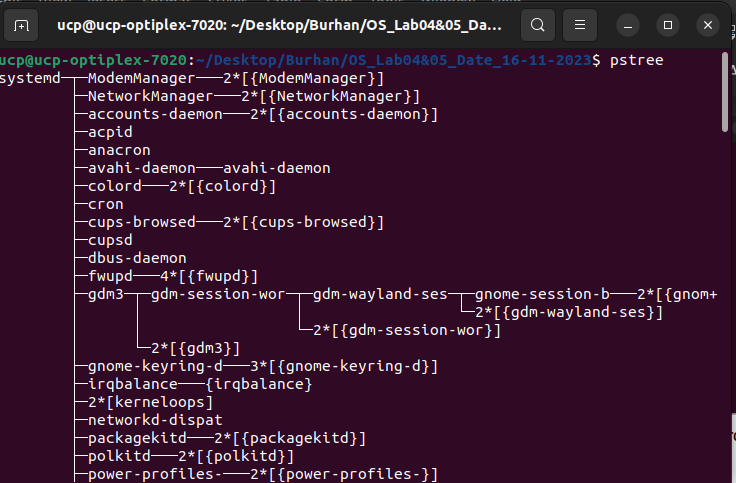
# Lab 4 Tasks

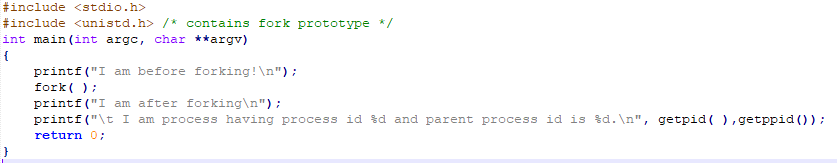
1. Display information about your processes that are currently running. Add Screenshot in a word file.



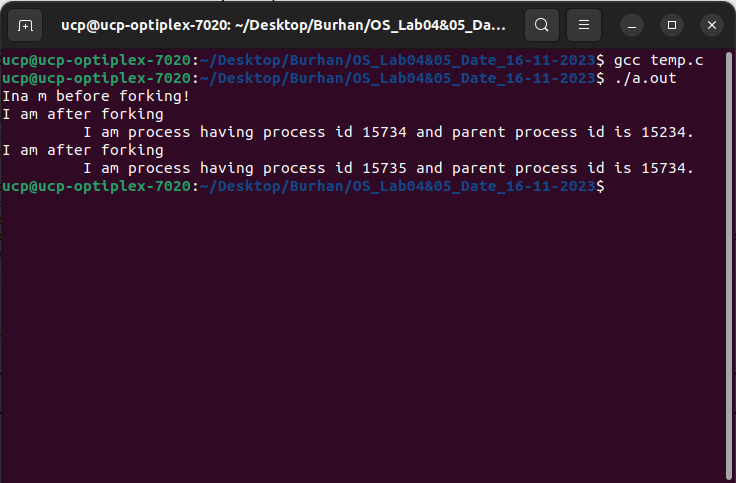
1. Display tree structure of your processes. Add Screenshot in a word file.



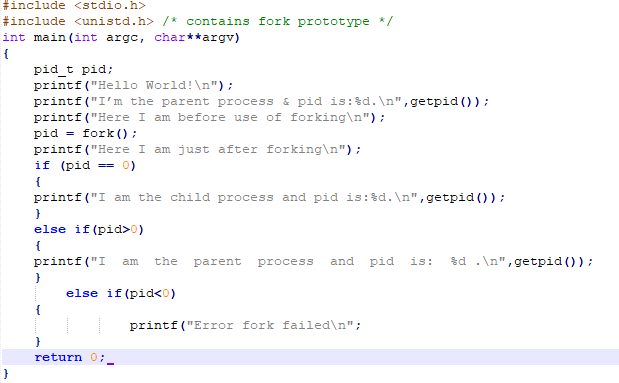
# Example Task 1: (Add output in word file)

****

Output:



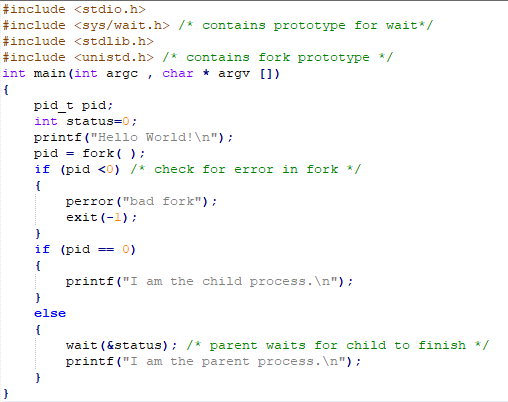
# Example Task 2: (Add output in word file)

****

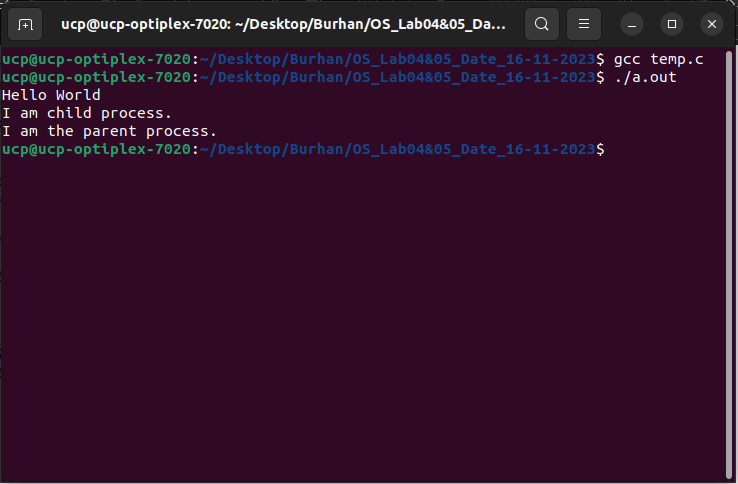
Output:

# 

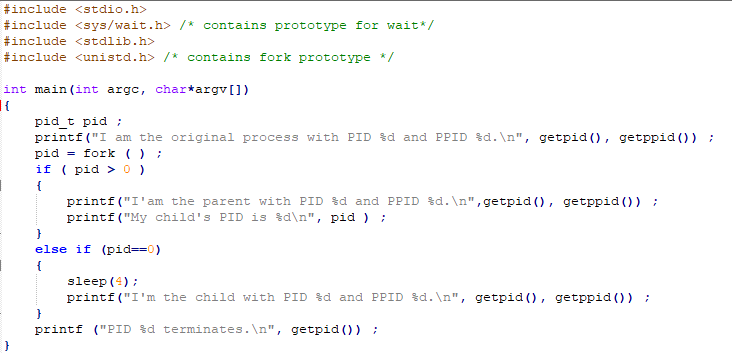
# Example Task 3: (Add output in word file)

****

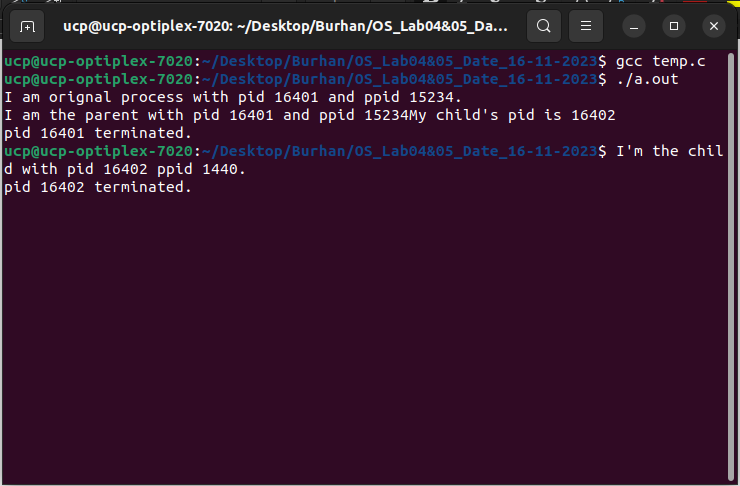
Output:



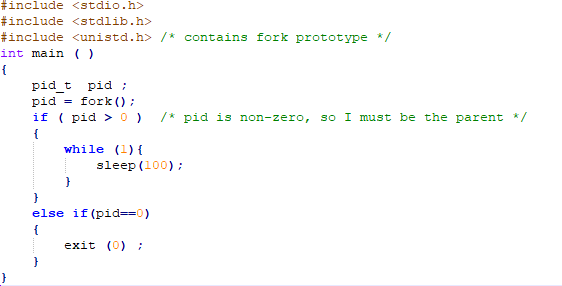
# Example Task 4: (Add output in word file)

****

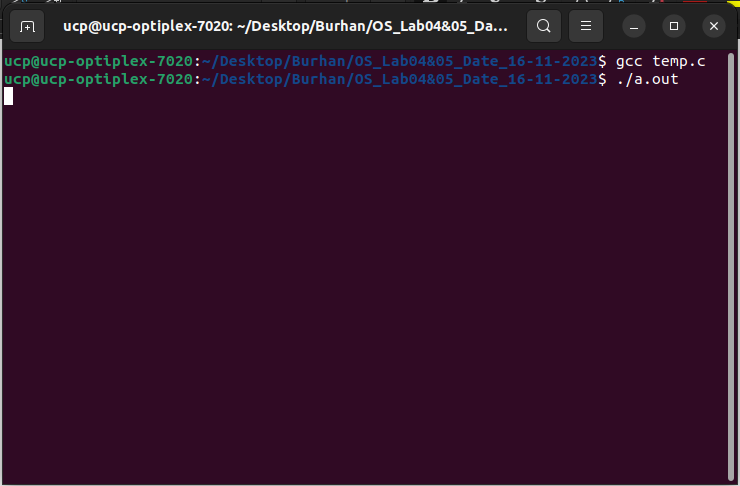
Output:



# Example Task 5: (Add output in word file)

****

Output:



# Lab 5 Tasks

# Sample Code 1

#include <unistd.h>

#include <stdlib.h>

#include <stdio.h>

#include <string.h>

int main()

{

int data\_processed = 0;

int file\_pipes[2];

const char some\_data[] = "123";

char buffer[BUFSIZ + 1];

memset(buffer, '\0', sizeof(buffer));

if (pipe(file\_pipes) == 0)

{

data\_processed = write(file\_pipes[1], some\_data, strlen(some\_data));

printf("Wrote %d bytes\n", data\_processed);

data\_processed = read(file\_pipes[0], buffer, BUFSIZ);

printf("Read %d bytes: %s\n", data\_processed, buffer);

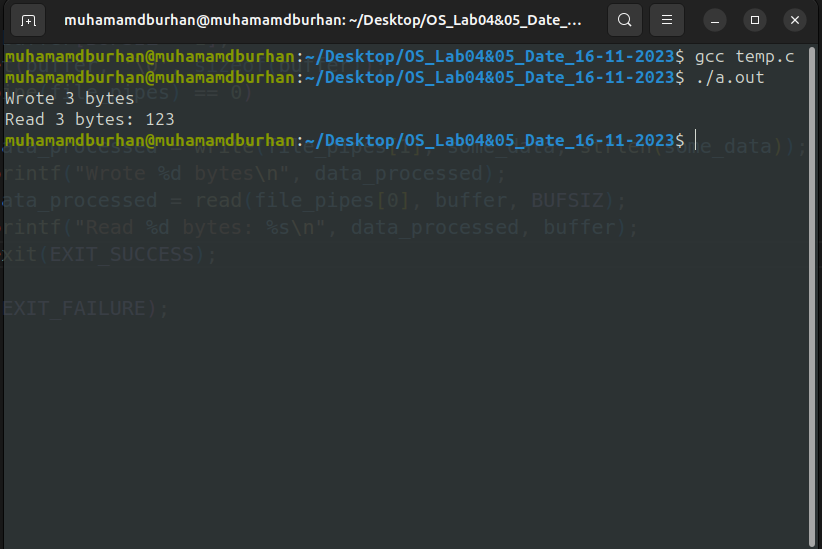
exit(EXIT\_SUCCESS);

}

exit(EXIT\_FAILURE);

}

**Output**



# Sample Code 2

# #include <unistd.h>

# #include <stdlib.h>

# #include <stdio.h>

# #include <string.h>

# #define READ 0

# #define WRITE 1

# int main()

# {

# char \*phrase = "This is OS lab time";

# int fd[2], bytesread;

# char message[100];

# pipe(fd);

# if (fork() == 0)

# /\* child, writer \*/

# {

# close(fd[READ]);

# /\* close unused end \*/ write(fd[WRITE], phrase, strlen(phrase) + 1);

# close(fd[WRITE]);

# /\* close used end \*/

# }

# else

# /\* parent, reader \*/

# {

# close(fd[WRITE]);

# /\* close unused end \*/

# bytesread = read(fd[READ], message, 100);

# printf("Read %d bytes : %s\n", bytesread, message);

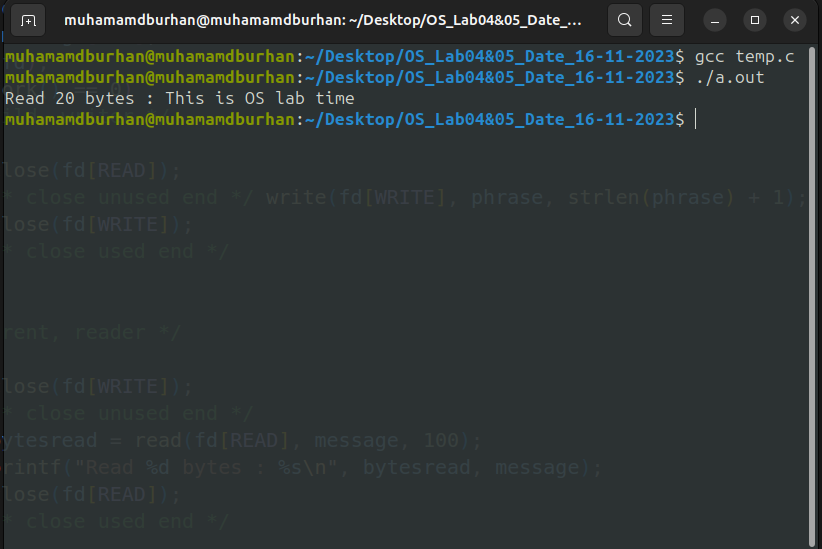
# close(fd[READ]);

# /\* close used end \*/

# }

# }

**Output**

****