UECS20CS254 - Operating Systems

**Unit-2 Programming exercises**

|  |  |  |
| --- | --- | --- |
| **NAME : SANMAT SANJAYAKUMAR PAYAGOUDAR** | **SRN : PES1UG20CS385** | **SECTION : G** |

**Q1.** Implement pipe operator in C with the following functionality:

* The parent process reads in a file “input.txt” and redirects the output to the pipe
* The child process must read the contents of the file and perform a word count

**CODE :**

#include<stdio.h>

#include<string.h>

#include<unistd.h>

#include<sys/wait.h>

#include<sys/types.h>

#define BUF\_SIZE 25

#define Read\_END 0

#define WRITE\_END 1

int main()

{

char Read[BUF\_SIZE]="";

int fd[2];

pid\_t pid;

if(pipe(fd) == -1)

{

printf("Pipe Failed\n");

return 1;

}

pid = fork();

if(pid>0)

{

FILE \*fp;

fp = fopen("input.txt","r");

char buff[20],ch;

int i=0;

while((ch=fgetc(fp))!=EOF)

{

buff[i]=ch;

i++;

}

buff[i]=0;

close(fd[Read\_END]);

write(fd[WRITE\_END],buff,strlen(buff)+1);

close(fd[WRITE\_END]);

wait(NULL);

fclose(fp);

}

else if(pid==0)

{

int cnt = 0;

close(fd[WRITE\_END]);

read(fd[Read\_END],Read,BUF\_SIZE);

printf("%s\n",Read);

char \*token = strtok(Read," ");

while(token!=NULL)

{

cnt++;

token = strtok(NULL," ");

}

printf("child process read word count: %d\n",cnt);

close(fd[Read\_END]);

}

else

{

printf("Fork Failed\n");

return 1;

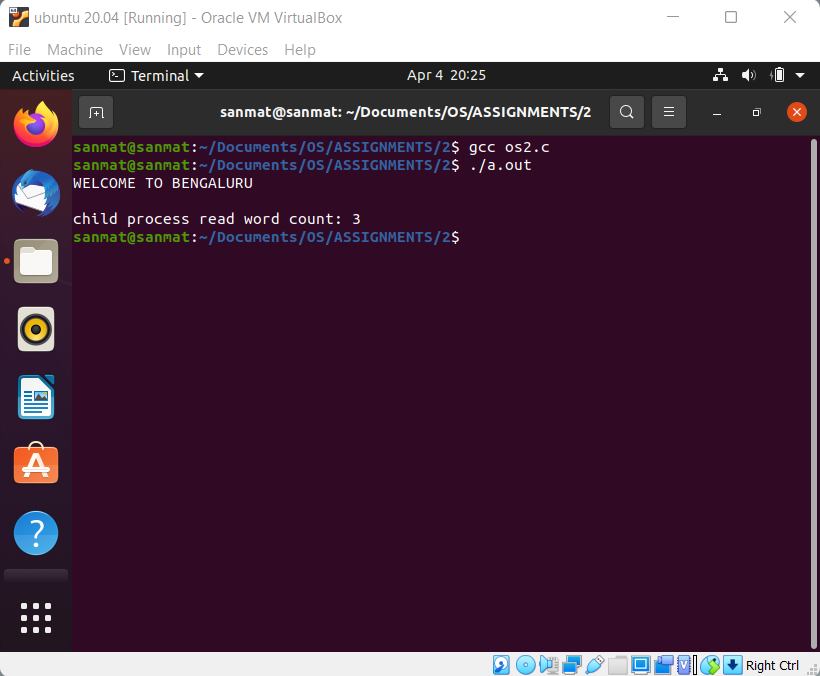
}

}

**INPUT TEXT FILE :**

WELCOME TO BENGALURU

**OUTPUT :**

****