**FINAL PROJECT REPORT**

**Fall 2020**

**System Programming**

Submitted by:

**Husna Noor** (**18PWCSE1730)**

Class Section: **C**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Student Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Submitted to:

**Engr. Madiha Sher**

March 18, 2021

**Department of Computer Systems Engineering**

**University of Engineering and Technology, Peshawar**

Project:

Audio Modification

By: Javeria Rahman

Husna Noor

Introduction:

The input for our program are two Audio Files of any format where the output of program is combination of the input files.

The combined music file can be played stopped and paused the sound.

Libraries used:

* #include <stdlib.h>
* #include <unistd.h>
* #include <stdio.h>
* #include <errno.h>
* #include <fcntl.h>
* #include <sys/types.h>
* #include <string.h>
* #include <sys/wait.h>

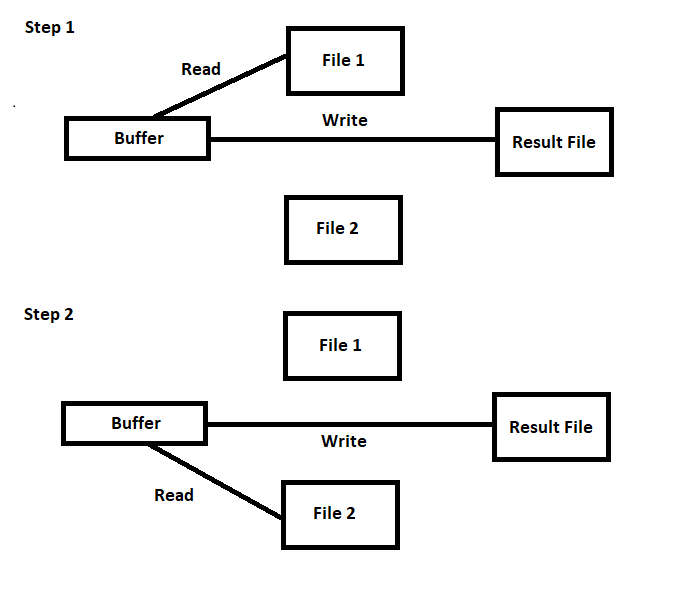
Language:

C language used

Algorithm for code:

* 3 inputs are taken from user from command line
* File1, file2 and result file are created of integer type
* For file1:
  + The first argument is given read write permission is granted.
* For file2:
  + The second argument is given read write permission is granted.
* For result file:
  + The third argument is given read write permission is granted
  + a new file is also created
* Data is read from file1 and written on resultant file.
* Data from file2 is read and written at the end of resultant file.
* A new combined file is created.
* If we want to stop it at one point, we can also stop it
* If we want to pause it, we can also pause it
* If we want to play it so we can also play it.

Block Diagram:



Code:  *(new version) \**

#include <stdlib.h>

#include <unistd.h>

#include <stdio.h>

#include <errno.h>

#include <fcntl.h>

#include <sys/stat.h>

#include <sys/types.h>

#include <string.h>

#include <sys/wait.h>

#include <signal.h>

#define Size 1000000

int main(int argc, char \*\*argv)

{

// Checking for command line argument

// it must be equals to 4 1(program file name) 2 3 4(mp3,mp3,output)

if(argc != 4){

printf("\nError: %s required argument [File1.mp3] [File2.mp3] [output.mp3]\n\n",argv[0]);

exit(EXIT\_FAILURE);

}

int File1,File2,Result\_File;

char Buff[Size];

int r,w;

unlink(argv[3]); // Delete out file if Exist

// Opening File 1

File1 = open(argv[1] , O\_RDONLY);

if(File1 < 0)

{

perror("Error in File 1 Opening: ");

exit(EXIT\_FAILURE);

}

// Opening File 2

File2 = open(argv[2] ,O\_RDONLY);

if(File2 < 0)

{

perror("Error in File 2 Opening: ");

exit(EXIT\_FAILURE);

}

// Result File

Result\_File = open(argv[3] ,O\_WRONLY | O\_CREAT,S\_IRWXU | S\_IRWXG);

if(Result\_File < 0)

{

perror("Error in Result File: ");

}

// Read from File 1

while((r = read(File1,Buff,Size))>0){

if(r < 0)

{

perror("Error in Reading from File 1: ");

}

w = write(Result\_File,Buff,r);

if(w < 0)

{

perror("Error in Writing from File 1: ");

}

printf("File 1 Total Bytes: %d\n",r);

if(r == 0)

{

break;

}

}

// Clear Buffer

memset(Buff,0,Size);

// Write new contents at the of Result file

lseek(Result\_File,0,SEEK\_END);

close(Result\_File); //close the output file

// OPEN result file and append data

Result\_File = open(argv[3] ,O\_WRONLY | O\_APPEND,S\_IRWXU | S\_IRWXG);

// Reading from File 2

while((r = read(File2,Buff,Size))>0)

{

if(r < 0)

{

perror("Error in Reading from File 1: ");

}

w = write(Result\_File,Buff,r);

if(w < 0)

{

perror("Error in Writing from File 1: ");

}

printf("File 2 Total Bytes: %d\n",r);

if(r == 0)

{

break;

}

}

close(Result\_File);

close(File1);

close(File2);

printf("\nWant to play music?[y/n]\n");

char choice;

scanf("%s",&choice);

while (choice!='n' || choice!='N' || choice!='q' || choice!='Q'){

printf("\n\t<<<<<<<<<<\*\*\*\*\*>>>>>>>>>>>\n");

printf("\ta. Play %s\n",argv[1]);

printf("\tb. Play %s\n",argv[2]);

printf("\tc. Play %s\n",argv[3]);

printf("\tq. Quit\n");

printf("\n\t<<<<<<<<<<\*\*\*\*\*>>>>>>>>>>>\n");

printf("Enter Choice : ");

scanf("%s",&choice);

if(choice=='a' || choice=='A'){

int x=fork();

if(x==0)

execlp("play","play",argv[1],NULL);

else{

printf("\n\n\t<<<<<<<<<<Enter ' s ' to stop Music : >>>>>>>>>>\n\n");

scanf("%s",&choice);

if(choice=='s' || choice=='S'){

kill(x,SIGKILL);

}

wait(NULL);

}

}

if(choice=='b' || choice=='B'){

int x=fork();

if(x==0)

execlp("play","play",argv[2],NULL);

else{

printf("\n\n\t<<<<<<<<<<Enter ' s ' to stop Music : >>>>>>>>>>\n\n");

scanf("%s",&choice);

if(choice=='s' || choice=='S'){

kill(x,SIGKILL);

}

wait(NULL);

}

}

if(choice=='c' || choice=='C'){

int x=fork();

if(x==0)

execlp("play","play",argv[2],NULL);

else{

printf("\n\n\t<<<<<<<<<<Enter ' s ' to stop Music : >>>>>>>>>>\n\n");

scanf("%s",&choice);

if(choice=='s' || choice=='S'){

kill(x,SIGKILL);

}

wait(NULL);

}

}

if(choice=='q' || choice=='Q'){

break;

}

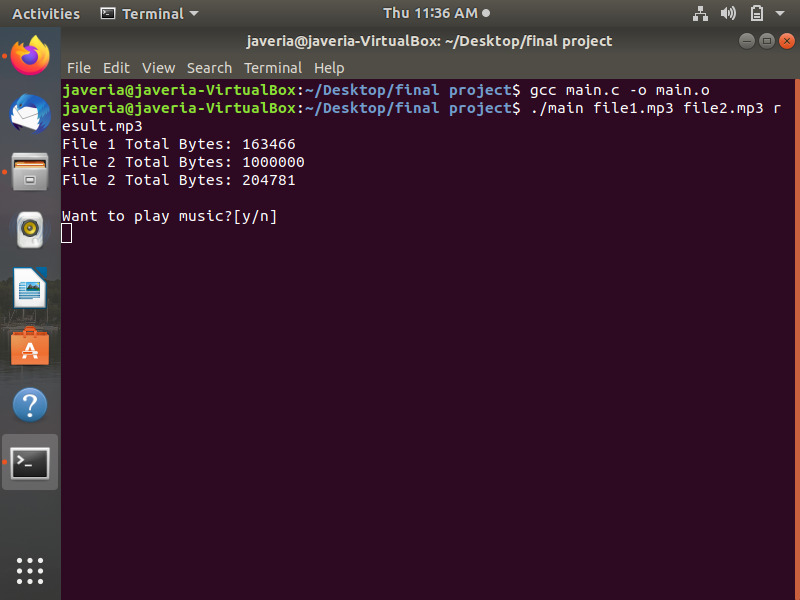
return 0;

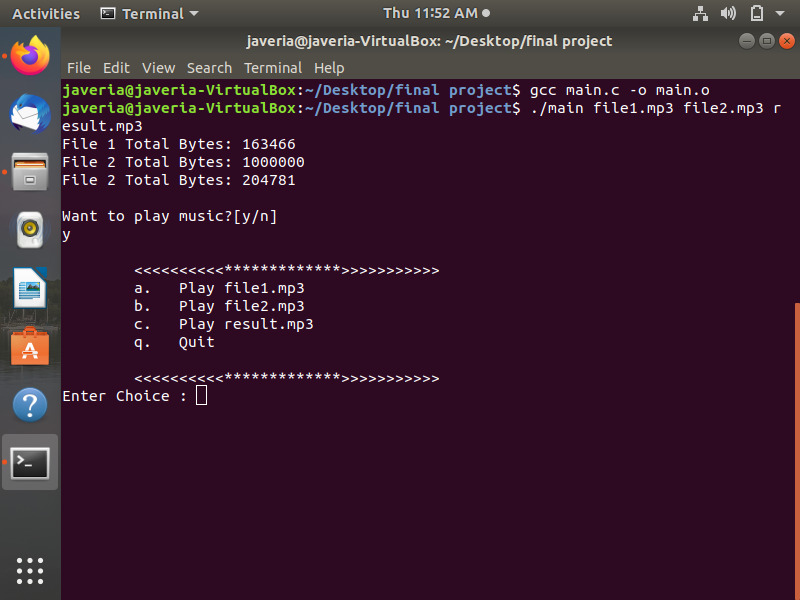
}

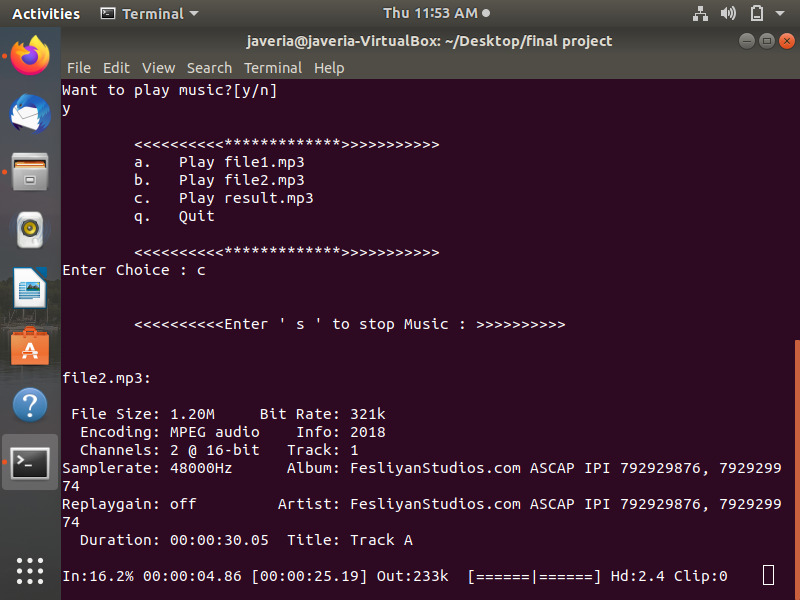
return 0;

}

Output:







What we learned:

We learned the difference between the new and old linux audio structure in new linux Sound System is Alsa “Advance Linux Sound Architecture”.

Presentation: *(improved)\**  file attached