**Write a C program to create an unnamed pipe. The child process will write following three messages to pipe and parent process display it.**

**Message1 = “Hello World”**

**Message2 = “Hello SPPU”**

**Message3 = “Linux is Funny”**

#include<stdio.h>

#include<unistd.h>

int main() {

int pipefds[2];

int returnstatus;

char writemessages[3][20]={"Hello World", "Hello SPPU","Linux is Funny"};

char readmessage[20];

returnstatus = pipe(pipefds);

if (returnstatus == -1) {

printf("Unable to create pipe\n");

return 1;

}

int child = fork();

if(child==0){

printf("Child is Writing to pipe - Message 1 is %s\n", writemessages[0]);

write(pipefds[1], writemessages[0], sizeof(writemessages[0]));

printf("Child is Writing to pipe - Message 2 is %s\n", writemessages[1]);

write(pipefds[1], writemessages[1], sizeof(writemessages[1]));

printf("Child is Writing to pipe - Message 3 is %s\n", writemessages[2]);

write(pipefds[1], writemessages[2], sizeof(writemessages[2]));

}

else

{

read(pipefds[0], readmessage, sizeof(readmessage));

printf("Parent Process is Reading from pipe – Message 1 is %s\n", readmessage);

read(pipefds[0], readmessage, sizeof(readmessage));

printf("Parent Process is Reading from pipe – Message 2 is %s\n", readmessage);

read(pipefds[0], readmessage, sizeof(readmessage));

printf("Parent Process is Reading from pipe – Message 3 is %s\n", readmessage);

}

}