Reg No: 18BCB0142
Name: David Bartolomeu Antonio De Vieira Velho
Subject: OS Lab DA1

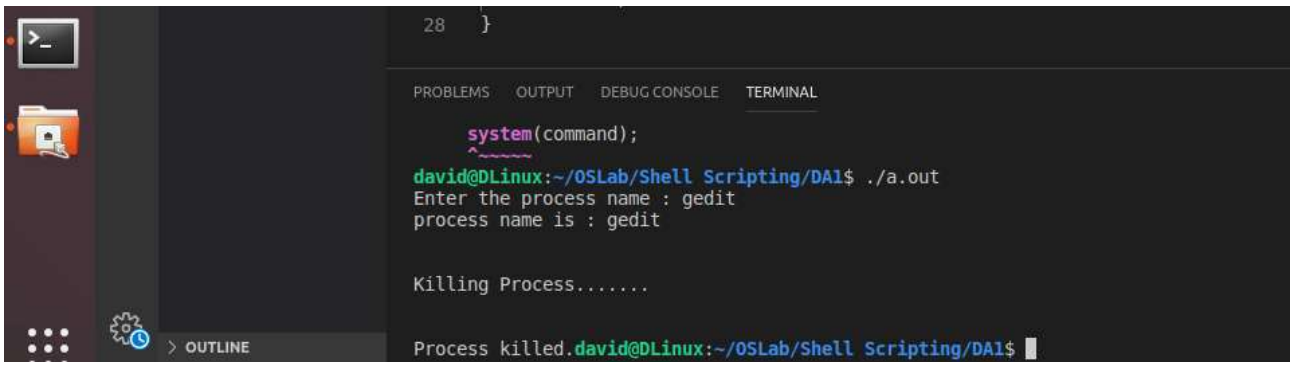
Q1) Write a C program to kill a process given its name.

Source:

```c
#include <stdio.h>
#include <sys/types.h>
#include <string.h>

int main(){

char process_name[30];
char command[50];
printf("Enter the process name : ");
scanf("%s", process_name);

strcpy(command, "ps -ef | grep ");
strcat(command, process_name);

printf("process name is : %s\n\n", process_name);

printf("\nKilling Process.....\n\n");
strcpy(command, "pkill ");
/*
* pkill kills the system process and takes input as name
*/
strcat(command, process_name);

system(command);
printf("\nProcess killed.");
return 0;
}
```

Output: (Before Execution) gedit process is running in the background

(After Execution)



Q2)
Create two POSIX threads. Let the first thread copy the contents of “file1.txt” to “file2.txt” and let the second thread collapse all spaces more than one to one space in an input string. Write the main function to test the working of the two threads.

Source:

```
#include <stdio.h>

#include <stdlib.h>
#include <sys/types.h>
#include <string.h>
#include <pthread.h>

#define MAX_STRING_SIZE 30


void thread1();
void thread2();



void thread1(){

system("cat file1.txt >> file2.txt");
}

void thread2(){

char s[MAX_STRING_SIZE];

printf("%s", "Enter the string: ");
gets(s);
printf("String %s before.\n", s);

int i;
int j;
int k;
```

```
for(k=0;k<MAX_STRING_SIZE;k++){
for(i=0;i<MAX_STRING_SIZE;i++){
if(s[i]==' ' && s[i+1]==' '){
//then we left shift every character

j=i;

for(j=i;j<MAX_STRING_SIZE-1;j++){
s[j]=s[j+1];
}

}
}
}
printf("String %s After", s);

}

int main(){

pthread_t t1,t2;

pthread_create(&t1, NULL,thread1, NULL);
pthread_create(&t2, NULL,thread2, NULL);
pthread_join(t1,NULL);
pthread_join(t2,NULL);
return 0;
}
```

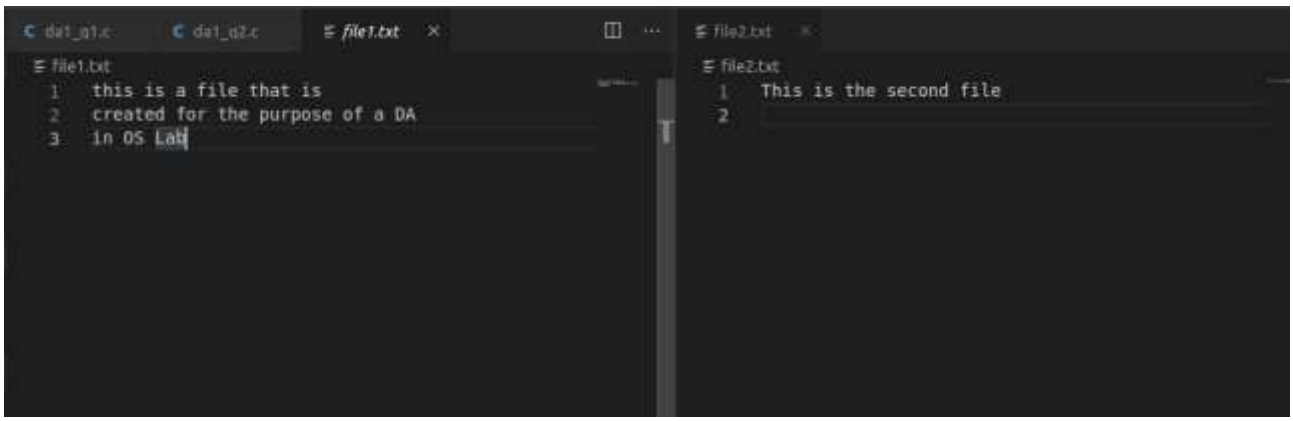
Additional Files:

file1.txt :

this is a file that is
created for the purpose of a DA
in OS Lab

file2.txt

this is the second file



Compile Instructions:

gcc <source code> -lpthread

Output: