LAB-7

-Saloni Patel

DUP2 system call:

```
#include int dup2(int oldfd, int newfd);
```

dup2() makes newfd be the copy of oldfd, closing newfd first if necessary.

- If oldfd is not a valid file descriptor, then the call fails, and newfd is not closed.
- If oldfd is a valid file descriptor, and newfd has the same value as oldfd, then dup2() does nothing, and returns newfd.

After a successful return from the system call, the old and new file descriptors may be used interchangeably.

EXECL functions:

```
#include int execl(const char *pathname, const char *arg, .../* (char *) NULL */);
```

The exec() family of functions replaces the current process image with a new process image. The functions described in this manual page are layered on top of execve(2).

The initial argument for these functions is the name of a file that is to be executed.

The const char *arg and subsequent ellipses can be thought of as arg0, arg1, ..., argn. Together they describe a list of one or more pointers to null-terminated strings that represent the argument list available to the executed program.

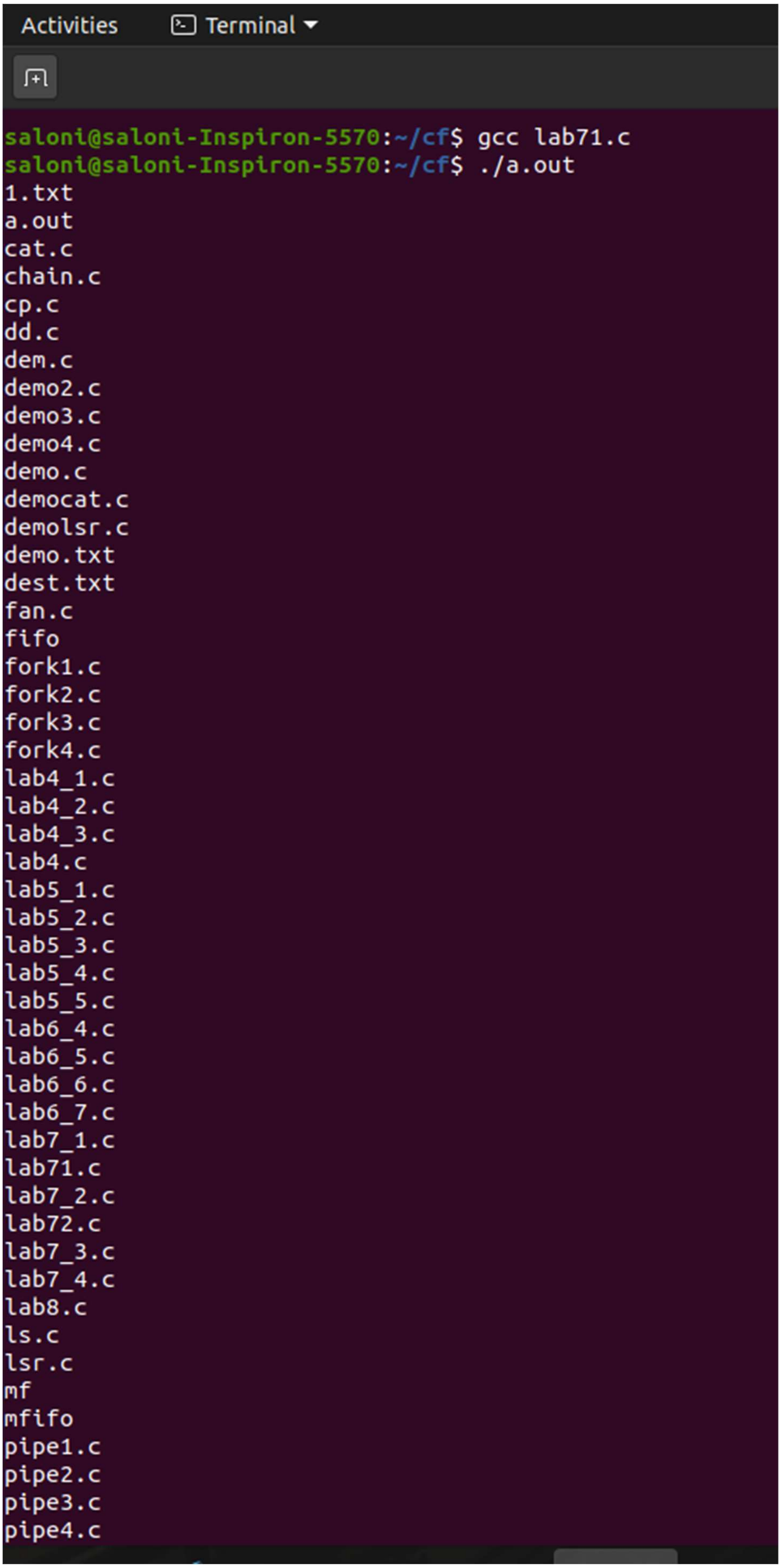
The first argument, by convention, should point to the filename associated with the file being executed. The list of arguments must be terminated by a null pointer, and, since these are variadic functions, this pointer must be cast (char *) NULL.

The exec() functions return only if an error has occurred. The return value is -1, and errno is set to indicate the error.

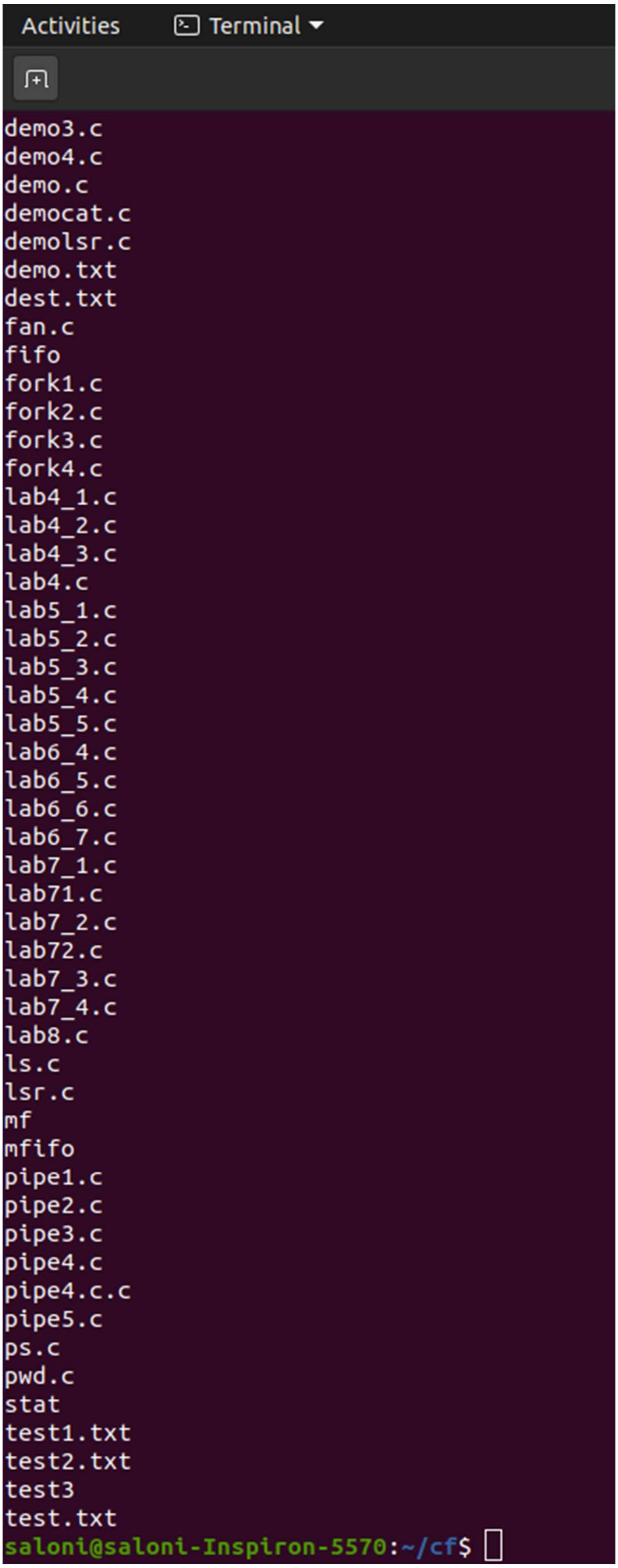
1. **Write a program to implement ls | sort functionality using the system calls and functions covered in the lab.**

```
#include<unistd.h>
#include<stdio.h>
#include<sys/types.h>
#include<sys/stat.h>
#include<fcntl.h>
#include<string.h>
#include<sys/wait.h>
#include<stdlib.h>
void main()
{
        int status,pid,pfd[2],p;
        p=pipe(pfd);
```

```
    if(p==0)
    {
        pid=fork();
        if(pid>0)
        {
            wait(&status);
            close(pfd[1]);
            dup2(pfd[0],0);
       execl("/bin/sort","sort",(char *)NULL);


        }
        else
        {
            close(pfd[0]);
            printf("child");
            dup2(pfd[1],1);
            execl("/bin/ls","ls","-1",(char *)NULL);

        }

    }
}
```



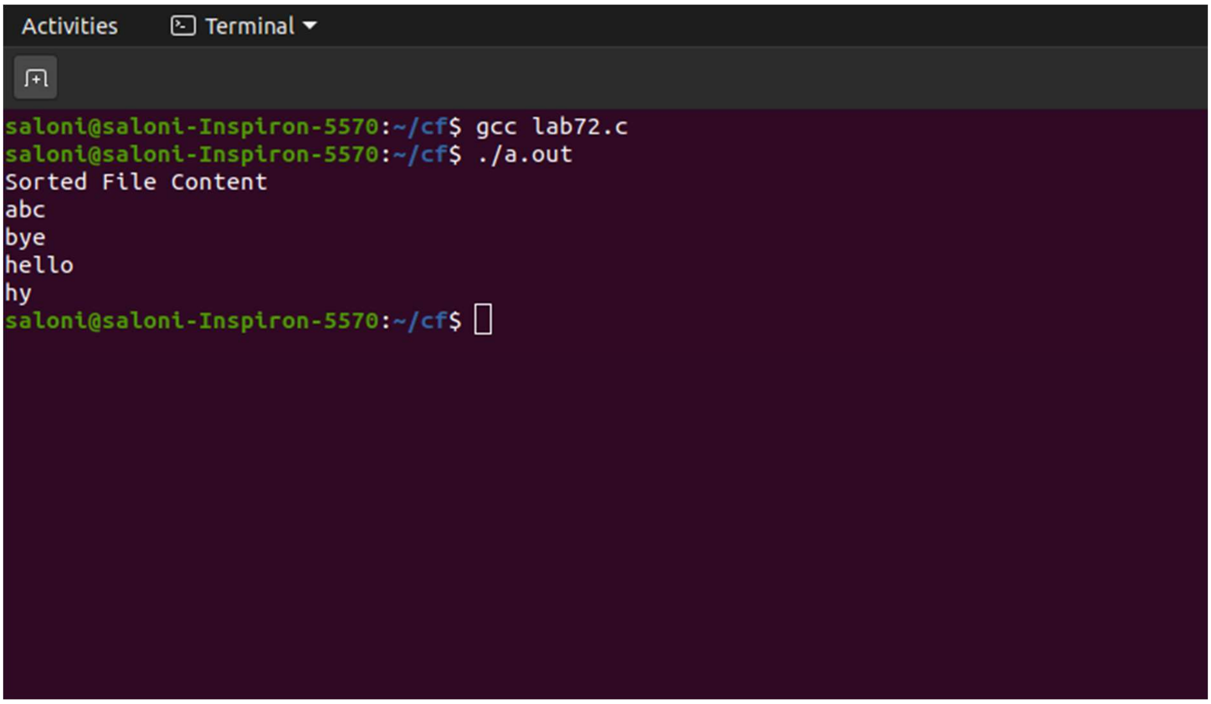
```
saloni@saloni-Inspiron-5570:~/cf$ gcc lab71.c
saloni@saloni-Inspiron-5570:~/cf$ ./a.out
1.txt
a.out
cat.c
chain.c
cp.c
dd.c
dem.c
demo2.c
demo3.c
demo4.c
demo.c
democat.c
demolsr.c
demo.txt
dest.txt
fan.c
fifo
fork1.c
fork2.c
fork3.c
fork4.c
lab4_1.c
lab4_2.c
lab4_3.c
lab4.c
lab5_1.c
lab5_2.c
lab5_3.c
lab5_4.c
lab5_5.c
lab6_4.c
lab6_5.c
lab6_6.c
lab6_7.c
lab7_1.c
lab71.c
lab7_2.c
lab72.c
lab7_3.c
lab7_4.c
lab8.c
ls.c
lsr.c
mf
mfifo
pipe1.c
pipe2.c
pipe3.c
pipe4.c
```



```
demo3.c
demo4.c
demo.c
democat.c
demolsr.c
demo.txt
dest.txt
fan.c
fifo
fork1.c
fork2.c
fork3.c
fork4.c
lab4_1.c
lab4_2.c
lab4_3.c
lab4.c
lab5_1.c
lab5_2.c
lab5_3.c
lab5_4.c
lab5_5.c
lab6_4.c
lab6_5.c
lab6_6.c
lab6_7.c
lab7_1.c
lab71.c
lab7_2.c
lab72.c
lab7_3.c
lab7_4.c
lab8.c
ls.c
lsr.c
mf
mfifo
pipe1.c
pipe2.c
pipe3.c
pipe4.c
pipe4.c.c
pipe5.c
ps.c
pwd.c
stat
test1.txt
test2.txt
test3
test.txt
saloni@saloni-Inspiron-5570:~/cf$
```

2. **Write a program to achieve following: o Child process should open a file with the contents to be sorted, pass the contents to parent process. o Parent process should sort the contents of the file and display.**

```
#include<unistd.h>
#include<stdio.h>
#include<sys/types.h>
#include<sys/stat.h>
#include<fcntl.h>
#include<string.h>
#include<sys/wait.h>
#include<stdlib.h>
void main()
{
    int status,pid,pfd[2],p,fd,n;
       char buff[50];
    p=pipe(pfd);
    if(p==0)
    {
        pid=fork();
        if(pid>0)
        {
            wait(&status);
             close(pfd[1]);
            dup2(pfd[0],0);
                    printf("Sorted File Content\n");
            execl("/bin/sort","sort",(char *)NULL);

             }
        else
        {
            close(pfd[0]);
                    fd=open("dest.txt",O_RDONLY);
                    n=read(fd,buff,sizeof(buff));
                    write(pfd[1],buff,n);

        }

    }
}
```



```
saloni@saloni-Inspiron-5570:~/cf$ gcc lab72.c
saloni@saloni-Inspiron-5570:~/cf$ ./a.out
Sorted File Content
abc
bye
hello
hy
saloni@saloni-Inspiron-5570:~/cf$ 
```