



**Lab Manual**  
***(Operating Systems)***

Department of Computer Science  
FAST-NU, Lahore

# I/O Redirection

## Dup() System Call

The dup() system call creates a copy of a file descriptor.

- It uses the lowest-numbered unused descriptor for the new descriptor.
- If the copy is successfully created, then the original and copy file descriptors may be used interchangeably.
- They both refer to the same open file description and thus share file offset and file status flags.

```
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>

int main()
{
    // open() returns a file descriptor file_desc to a
    // the file "dup.txt" here
    int file_desc = open("input.txt", O_WRONLY | O_APPEND);
    if (file_desc < 0)
    {
        printf("Error opening the file\n");
    }

    // dup() will create the copy of file_desc as the copy_desc
    // then both can be used interchangeably.
    int copy_desc = dup(file_desc);

    // write() will write the given string into the file
    // referred by the file descriptors
    write(copy_desc, "This will be output to the file named dup.txt\n", 46);
    write(file_desc, "This will also be output to the file named dup.txt\n", 51);
    return 0;
}
```

The open() returns a file descriptor file\_desc to the file named "dup.txt". file\_desc can be used to do some file operation with file "dup.txt". After using the dup() system call, a copy of file\_desc is created copy\_desc. This copy can also be used to do some file operation with the same file "dup.txt". After two write operations one with file\_desc and another with copy\_desc, same file is edited i.e. "dup.txt".

## Dup2() System Call

The dup2() system call is similar to dup() but the basic difference between them is that instead of using the lowest-numbered unused file descriptor, it uses the descriptor number specified by the user.

- Include the header file unistd.h for using dup() and dup2() system call.
- If the descriptor newfd was previously open, it is silently closed before being reused.
- 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.

```

#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>

int main()
{
    // open() returns a file descriptor file_desc to a
    // the file "dup.txt" here"
    int file_desc = open("input.txt", O_WRONLY | O_APPEND);
    if (file_desc < 0)
    {
        printf("Error opening the file\n");
    }

    // dup() will create the copy of file_desc as the copy_desc
    // then both can be used interchangeably.
    dup2(file_desc, 1);

    // write() will write the given string into the file
    // referred by the file descriptors
    printf("Operating System");
    return 0;
}

```

Usual FDT in C

|   |  |
|---|--|
| 0 | STD_IN (Keyboard)                          |
| 1 | STD_OUT (Monitor)                          |
| 2 | STD_ERR (Monitor)                          |
| 3 | Some other file that you might have opened |
| . | .  |

## Tasks 1: (I/O REDIRECTION using filing)

(5 Marks)

- Write a program that reads text from a file input.txt (name passed through command line arguments) rather than a keyboard i.e. stdin. Like this:

```

// 0 represents the stdin in FDT
int status = read(0, data, 500);

```

- Use open() and read() system calls.
- Write this data to a new file output.txt rather than console, using **printf()**. Like this:

```
printf("%s", data);
```

- This whole process is to be done by Dup2 system call.

(Hint: Replace stdin (0) value with input file's file descriptor and stdout (1) value with output file's file descriptor)

Submit File with name as: YOUR\_ROLLNUMBER\_Q1.c and also submit output with name YOUR\_ROLLNUMBER\_Q1\_OUTPUT.jpg

## Task 2:

(5 Marks)

- Write a program that performs the task of the below shell command. **cat input.txt > output.txt**
- “cat” command prints all the contents of a file onto “stdout” i.e. monitor.
- “>” is used to redirect output when you are using shell/terminal.
- Now you have to redirect output of “cat” command into a file called “output.txt”. □ Write any data of your liking into input.txt file.

Submit File with name as: YOUR\_ROLLNUMBER\_Q2.c and also submit output with name YOUR\_ROLLNUMBER\_Q2\_OUTPUT.jpg