National University of Computer and Emerging Sciences

Laboratory Manual

for

Operating Systems Lab

(CL-2006)

Course Instructor	Ms. Rubab Anam
Lab Instructor (s)	Muhammad Faheem
Section	BCS-4F1
Semester	Spring 2025

Department of Computer Science

FAST-NU, Lahore, Pakistan

Objectives:

• Input, and output redirection using Dup, Dup2 system calls. • Use of pipe and dup together

1 dup" System Call Manual

1.1 NAME

dup - Duplicate an open file descriptor

1.2 SYNOPSIS

```
#include <unistd.h
          oldfd)
in t
dup(int
```

>

1.3 DESCRIPTION

The **dup** system call creates a new file descriptor that refers to the same open file description as the **oldfd** file descriptor. The new file descriptor is the lowest-numbered available descriptor.

1.4 PARAMETERS

- **oldfd**: The file descriptor to be duplicated.

1.5 RETURN VALUE

- On success, **dup** returns a new file descriptor that refers to the same file as **oldfd**. If an error occurs, it returns -1, and **errno** is set to indicate the error.

1.6 ERRORS

- **EBADF**: **oldfd** is not a valid file descriptor.
- **EMFILE**: The process has too many open file descriptors.
- Other errors as described in the **errno** documentation.

2 “dup2” System Call Manual

2.1 NAME

dup2 - Duplicate an open file descriptor to a specified file descriptor

number 2.2 SYNOPSIS

```
include
<unistd.h>
```

```
int
```

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

2.3 DESCRIPTION

The dup2 system call duplicates the file descriptor oldfd to newfd, allowing you to specify a particular file descriptor number for the duplication. If newfd is already in use, it is closed before the duplication occurs.

2.4 PARAMETERS

- **oldfd**: The file descriptor to be duplicated.
- **newfd**: The desired file descriptor number for the duplication.

2.5 RETURN VALUE

On success, **dup2** returns **newfd**, which is the duplicated file descriptor. If an error occurs, it returns -1, and **errno** is set to indicate the error.

2.6 ERRORS

- **EBADF**: **oldfd** is not a valid file descriptor, or **newfd** is negative or exceeds the maximum allowed file descriptor value.
- **EMFILE**: The process has too many open file descriptors.
- Other errors as described in the **errno** documentation.

Lab Tasks

Question 1: Using the dup Function

Write a C/C++ program that reads text from a file named "original.txt" and converts all the characters to uppercase. Then, update the content of the file with the modified text, ensuring that all letters are in uppercase. Utilize the **dup** function to duplicate the file descriptor for performing the file operations. Ensure proper error handling throughout the program. **You must use printf/scanf for reading & writing in original.txt.**

Question 2: Using the dup2 Function

Develop a C/C++ program to find even numbers within a given range specified in a file named "input.txt". The file contains two numbers separated by a space, indicating the start and end of the range, respectively. Instead of using read or write system calls for file operations, employ the **dup2** function to provide these two numbers as input via standard input (stdin). Similarly, redirect the output generated on the terminal to a file named "output.txt" without using read or write system calls. Ensure error handling for file operations and follow proper coding standards. The task to find even numbers must be done by child process and the parent must communicate with the child through a pipe.

Evaluation will be in the lab but you need to submit it online within time.