**Assignment No. 04**

**Problem Statement :** Write a program to simulate inter process communication mechanism using pipes and redirection.

**Code :**

#include<stdio.h>

#include<unistd.h>

#include<sys/types.h>

#include<sys/stat.h>

#include<fcntl.h>

int main(int argc, char\* argv[]) {

    if(argc != 3) {

        printf("You can not enter more or less than 3 arguments.\n");

        return 0;

    }

    int fd[2];

    int fork\_val;

    char\* file\_1 = argv[1];

    char\* file\_2 = argv[2];

    int src\_file;

    int dest\_file;

    src\_file = open(file\_1, O\_RDONLY);

    if(src\_file == -1) {

        printf("Unable to open source file!!!\n");

        return 1;

    } else {

        dup2(src\_file, STDIN\_FILENO);

        close(src\_file);

    }

    pipe(fd);

    fork\_val = fork();

    if(fork\_val > 0) {

        dup2(fd[1], STDOUT\_FILENO);

        close(fd[0]);

        close(fd[1]);

        execl("/usr/bin/sort", "sort", NULL);

        perror("execl failed for sort");

    }

    else if(fork\_val == 0) {

        dup2(fd[0], STDIN\_FILENO);

        dest\_file = open(file\_2, O\_WRONLY | O\_CREAT | O\_TRUNC, 0644);

        if(dest\_file == -1) {

            printf("Unable to open destination file!!!\n");

            return 1;

        } else {

            dup2(dest\_file, STDOUT\_FILENO);

            close(dest\_file);

            close(fd[0]);

            close(fd[1]);

            execl("/usr/bin/uniq", "uniq", NULL);

            perror("execl failed for uniq");

        }

    } else {

        perror("Fork failed!!!");

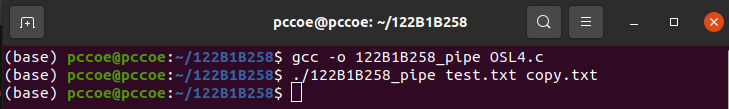
        return 1;

    }

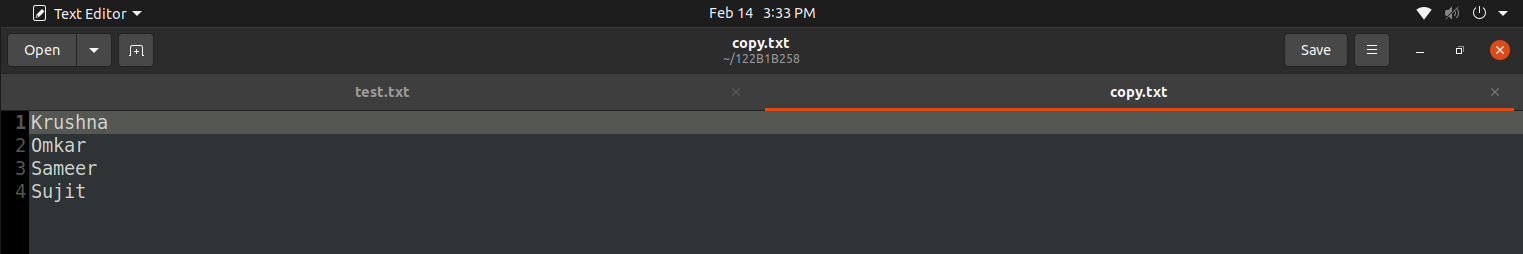
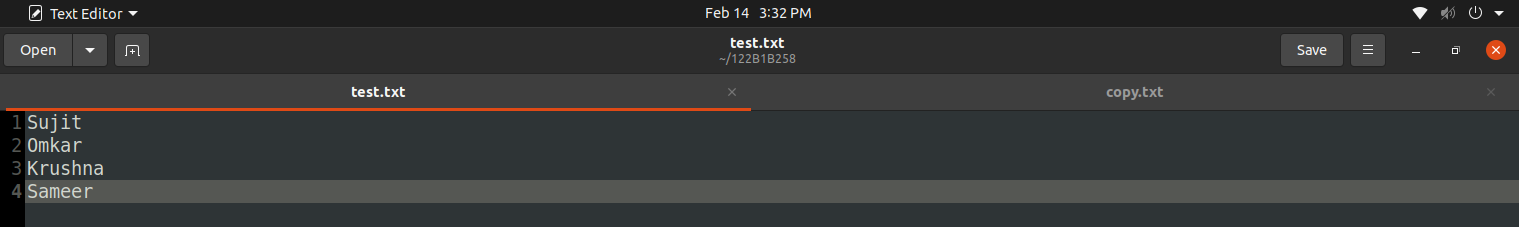
    return 0;

}

**Complilation :**

****

**Output :**

****