

LAB ASSIGNMENT-4

Date of the Submission: 03/01/2022

Course Code: CSLR42

Subject: Operating Systems Lab

Submitted by:

Roll Number: 106120030

Name: Devipriya Sozharajan

Class: Btech CSE-B

1. Write a program to copy the content of one file to another. (Use open, read and write system call). – Implement the “CP” command.

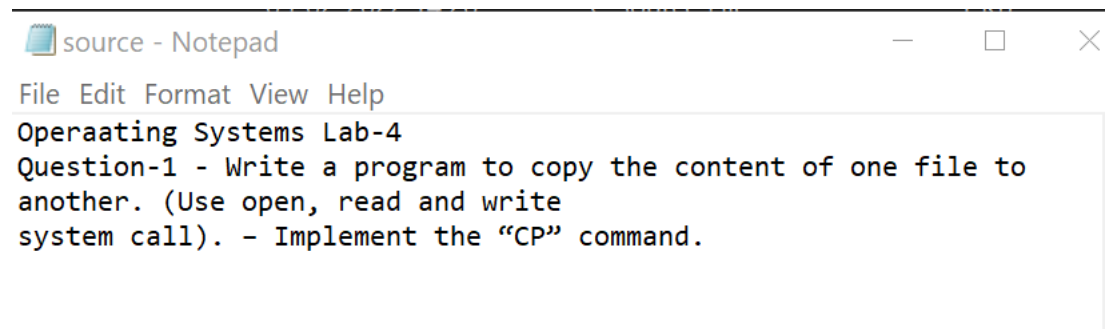
```
#include <stdio.h>

int main(int argc, char **argv)
{
    //printf("%s \n%s", argv[2], argv[1]);
    FILE *sourceFile, *destinationFile;
    sourceFile = fopen(argv[1], "r");
    destinationFile = fopen(argv[2], "w");
    if ((sourceFile == NULL))
        printf("File doesn't exist");
    char ch = fgetc(sourceFile);
    while (ch != EOF)
    {
        fputc(ch, destinationFile);
        ch = fgetc(sourceFile);
    }
    fclose(sourceFile);
    fclose(destinationFile);
}
```

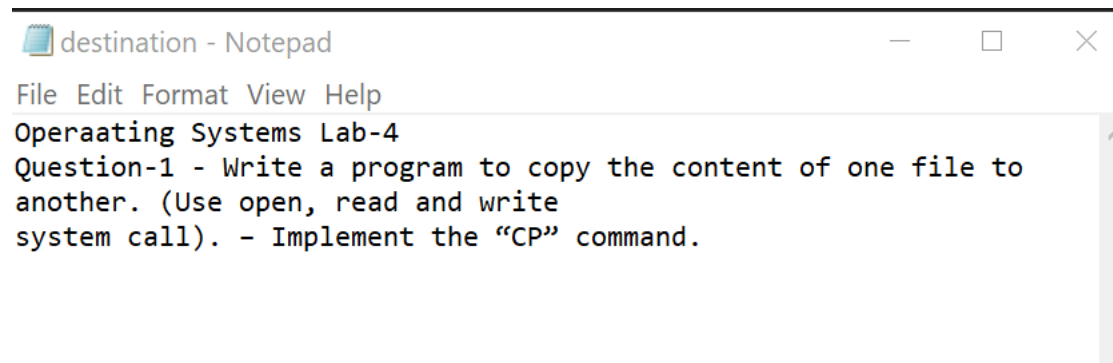
OUTPUT:

Content of source is copied to destination file:

Source:



Destination:



2. Write a C program to implement the cat command in Unix.

```
#include <stdio.h>

int main(int argc, char **argv)
{
    int count = argc;
    for (int i = 1; i <= count; i++)
    {
        FILE *inputFile;
        inputFile = fopen(argv[i], "r");
        if ((inputFile == NULL))
            printf("%s File doesn't exist\n", argv[i]);
        else
        {
            printf("\n\n%s:\n", argv[i]);
            char ch = fgetc(inputFile);
            while (ch != EOF)
            {
                printf("%c", ch);
                ch = fgetc(inputFile);
            }
        }
        fclose(inputFile);
    }
}
```

Output:

```
Admin@DESKTOP-4GFEIDQ MINGW64 /d/LAB-OS/Lab-4
$ ./Qn2 source.txt Qn2.c

source.txt:
Operating Systems Lab-4
Question-1 - Write a program to copy the content of one file to another. (Use open, read and write
system call). - Implement the "CP" command.

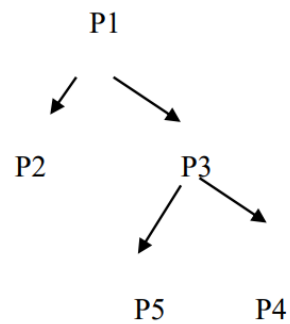
Qn2.c:
#include <stdio.h>

int main(int argc, char **argv)
{
    int count = argc;
    //printf("%d\n", count);

    for (int i = 1; i <= count; i++)
    {
        FILE *inputFile;
        inputFile = fopen(argv[i], "r");

        if ((inputFile == NULL))
            printf("%s File doesn't exist\n", argv[i]);
        else
        {
            printf("\n\n%s:\n", argv[i]);
            char ch = fgetc(inputFile);
            while (ch != EOF)
            {
                printf("%c", ch);
                ch = fgetc(inputFile);
            }
        }
        fclose(inputFile);
    }
}(null) File doesn't exist
```

3. Write a program to create the following hierarchy and print their Id and its parent ID using fork statements.



```
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>

int main(void)
{
    pid_t p1, p2, p3, p4, p5;
    p1 = fork();

    if (p1 == 0)
    {
        printf("P1 => Process ID: %d Parent ID: %d\n", getpid(), getppid());
        p2 = fork();
        if (p2 == 0)
        {
            printf("\nP2 => Process ID: %d Parent ID: %d\n", getpid(),
getppid());
            exit(EXIT_SUCCESS);
        }
        else if (p2 > 0)
        {
            wait(NULL);
            p3 = fork();
            if (p3 == 0)
            {
                printf("\nP3 => Process ID: %d Parent ID: %d\n", getpid(),
getppid());
                p5 = fork();
                //p4 = fork();
                if (p5 == 0)
                {
```

```

        printf("\nP5 => Process ID: %d Parent ID: %d\n", getpid(),
getppid());
        exit(EXIT_SUCCESS);
    }
    if (p5 > 0)
    {
        wait(NULL);
        p4 = fork();
        if (p4 == 0)
        {
            printf("\nP4 => Process ID: %d Parent ID: %d\n",
getpid(), getppid());
            exit(EXIT_SUCCESS);
        }
        if (p4 > 0)
        {
            wait(NULL);
        }
    }
}
else if (p3 > 0)
{
    wait(NULL);
}
}
exit(EXIT_SUCCESS);
}

else if (p1 > 0)
{
    wait(NULL);
}

return EXIT_SUCCESS;
}

```

OUTPUT:

```

P1 => Process ID: 5351 Parent ID: 5347

P2 => Process ID: 5352 Parent ID: 5351

P3 => Process ID: 5353 Parent ID: 5351

P5 => Process ID: 5354 Parent ID: 5353

P4 => Process ID: 5355 Parent ID: 5353

```

4. Create 2 processes (parent and child process) and a pipe. Child process read N characters and puts in to the pipe and Parent process reads and finds total no of vowels.

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>
#include <ctype.h>

int main(void)
{
    int n;
    char str[100];
    FILE *inputFile;
    inputFile = fopen("inputFile4.txt", "r");
    printf("Number of characters: ");
    scanf("%d", &n);

    int fd[2], i = 0;
    pipe(fd);
    pid_t pid = fork();

    if (pid > 0)
    {
        wait(NULL);
        close(0);
        close(fd[1]);
        dup(fd[0]);
        char str[100];
        int count = 0;
        int size = read(fd[0], str, sizeof(str));
        for (i = 0; i < size; i++)
        {
            str[i] = tolower(str[i]);
            if ((str[i] == 'a') || (str[i] == 'e') || (str[i] == 'i') ||
                (str[i] == 'o') || (str[i] == 'u'))
                count++;
        }
        printf("Number of vowels: %d", count);
    }
    else if (pid == 0)
    {
        printf("Enter string: ");
        scanf("%s", str);
        close(fd[0]);
        close(1);
        dup(fd[1]);
    }
}
```

```
    write(1, str, n);  
  }  
}
```

OUTPUT:

```
Number of characters: 10  
Enter string: operatings  
Number of vowels: 4
```