Name: V.P.Abishek Roll no: 205001002

Exercise 2 - Simulation of System commands using System calls

1)Implementation of cp and cp -i command using system calls.

Aim :

To copy the contents of one file into another file.

Algorithm :

1:Read the source and destination file with the operation to be performed using command line.

2:If argument length less than 3 or greater than 4, print invalid input.

3:If argument length is equal to 4 and 1st argument is -i, open file in 2nd argument and 3rd argument and store it.

3.1: If source file already exists, ask the user if he wants overwrite the contents. If yes, close the file. Else close the file and exit from the program.

3.2: Create a file in the name of 3rd argument.

3.3: Read the contents of source file and write it in destination file.

3.4: Close the source and the destination file.

4: Else

4.1: Open the source file and store it.

4.2: If source file does not exist, exit and terminate the program.

4.3: Create a file of name as in destination file and store it.

4.4: Read the contents of source file and write it in destination file.

4.5: Close both the files.

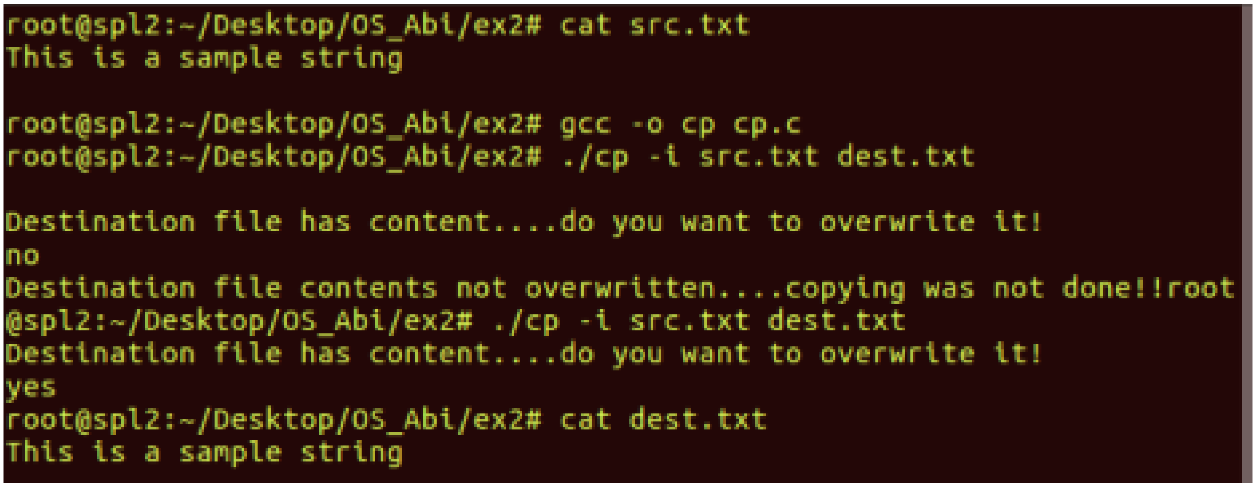
Source code:

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>
#include <string.h>
int main(int argc, char* argv[])
{
        if(argc < 3)
        printf("Too few arguements\n");
        else
        {
         char content[1024];
        int source_file,dest_file,count;

        if(argc==3)
        {
                        source_file= open(argv[1], O_RDONLY);
                dest_file= creat(argv[2], O_RDWR);
         }
         if(argc == 4 && strcmp(argv[1],"-i") == 0)
         {
                source_file= open(argv[2], O_RDONLY);
                read(dest_file, content, 26);
                if(strlen(content) !=0 )
                {
                        printf("Destination file has content....do you want to overwrite it!\n");
                        char str[10];
                        scanf("%str", str);
                        if(strcmp(str,"no") == 0)
                        {
                                printf("Destination file contents not overwritten....copying was not
done!!");
                                return 0;
                        }
                }
                dest_file= creat(argv[3], O_RDWR);
         }
         if(source_file == -1)
                printf("Cannot open %s",argv[1]);
         else
         {
                int r,w;
                read(source_file, content, 26);
                close(source_file);
                if(dest_file == -1)
                        printf("Destination file not created\n");
                else
                {
                        write(dest_file, content, strlen(content));
                        close(dest_file);
                }
        }

        }
        return 0;
}
```

Output:



```
root@spl2:~/Desktop/OS_Abi/ex2# cat src.txt
This is a sample string

root@spl2:~/Desktop/OS_Abi/ex2# gcc -o cp cp.c
root@spl2:~/Desktop/OS_Abi/ex2# ./cp -i src.txt dest.txt

Destination file has content....do you want to overwrite it!
no
Destination file contents not overwritten....copying was not done!!root
@spl2:~/Desktop/OS_Abi/ex2# ./cp -i src.txt dest.txt
Destination file has content....do you want to overwrite it!
yes
root@spl2:~/Desktop/OS_Abi/ex2# cat dest.txt
This is a sample string
```

2)Implementation of ls and ls -a command using system calls.

Aim :

To list all files and directories of a given directory.

Algorithm :

```
1: Read the directory name with operation to be performed from command line.
2: If argument length is less than 1 or greater than 3, print invalid and terminate from the program.
3: Create a pointer to the structure dirent.
4: If argument length is equal to 3,
    4.1: If directory is not present, terminate the program.
    4.2: Open directory and store it in a variable.
    4.3: While the contents of directory is not null
        4.3.1: Print the directory or file name.
        4.3.2: Check if the file or directory is readable ‘r’, writable ‘w’ and executable ‘x’. If it is not, print -.
        4.3.3: Print the size of it along with the date of creation of it.
    4.4: Close the directory that was opened.
    5: Else
        5.1: Open the directory and store it in a variable.
        5.2: While the contents of the directory is not null, print the contents of the directory.
        5.3: Close the directory that was opened.
```

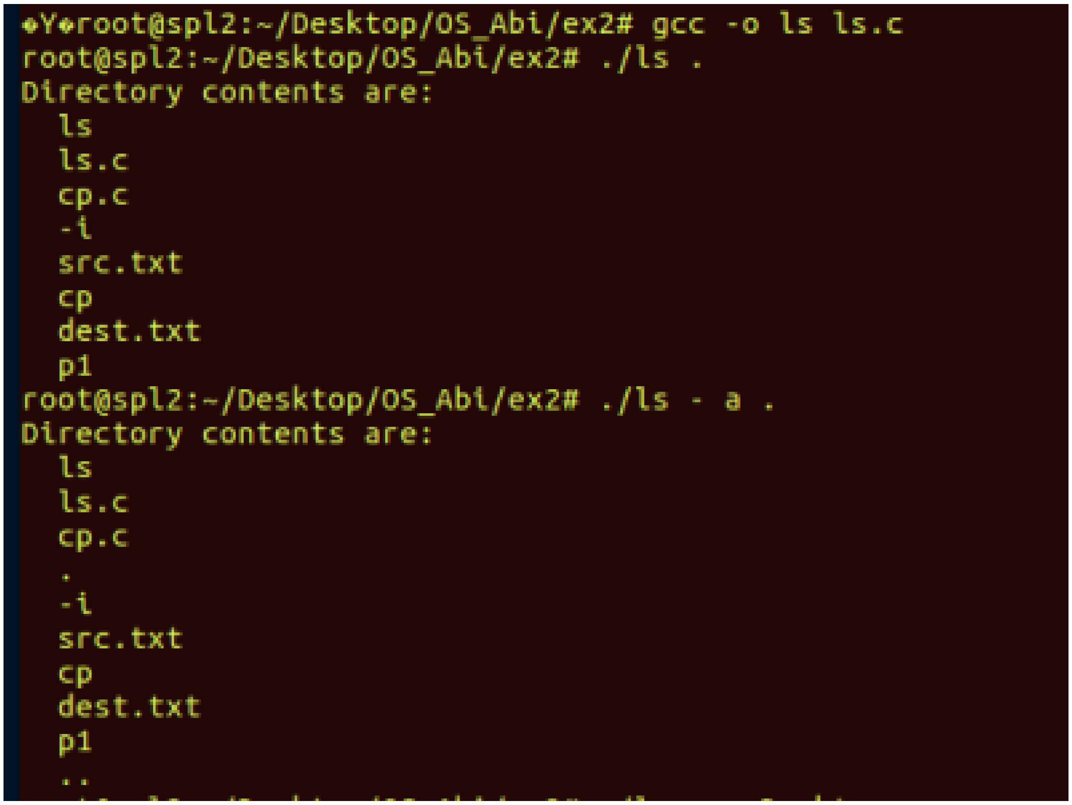
Source code:

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <dirent.h>
#include <fcntl.h>
#include<string.h>

DIR *dir, *temp;
struct dirent *tmp;
int main(int argc, char *argv[])
{
	if (argc > 4)
	printf("Too many arguements\n");
	else if (argc < 1)
	printf("Too few arguements\n");
	else
	{
	if (argc == 2)
	{
	struct dirent *ptr;
	if ((dir = opendir(argv[1])) == NULL)
		printf("Given directory cannot be opened\n");
	else
	{
		printf("Directory contents are:\n");
```

```
        while ((ptr = readdir(dir)) != NULL)
        {
        if (ptr->d_name[0] == '.')
        continue;
        printf(" %s\n", ptr->d_name);
        }
        closedir(dir);
    }
    }
    else if (argc > 1 && strcmp(argv[2], "a") == 0)
    {
    struct dirent *ptr;
    if ((dir = opendir(argv[3])) == NULL)
         printf("Given directory cannot be opened\n");
    else
    {
        printf("Directory contents are:\n");
        while ((ptr = readdir(dir)) != NULL)
        {
        printf(" %s\n", ptr->d_name);
        }
        closedir(dir);
    }
    }
    }
    return 0;
}
```

Output:



```
◆Y◆root@spl2:~/Desktop/OS_Abi/ex2# gcc -o ls ls.c
root@spl2:~/Desktop/OS_Abi/ex2# ./ls .
Directory contents are:
  ls
  ls.c
  cp.c
  -i
  src.txt
  cp
  dest.txt
  p1
root@spl2:~/Desktop/OS_Abi/ex2# ./ls - a .
Directory contents are:
  ls
  ls.c
  cp.c
  .
  -i
  src.txt
  cp
  dest.txt
  p1
  ..
```

3)Implementation of grep and grep -c command using system calls.

Aim :

To develop a C program to implement grep command.

Algorithm :

```
1: Id argument count greater than 4
2: Print too many arguments
3: Else argument count less than 2
4: Print less arguments
5: Else open file in argv[2] using open() command
6: Id file found
```

```
    7: Check for the word/ pattern argv[1] for file each line
    8: If line found, print line
    9: If argv[3] is c
        9.1: If a word is found in each line print each line and increment count
    10: Print count
```

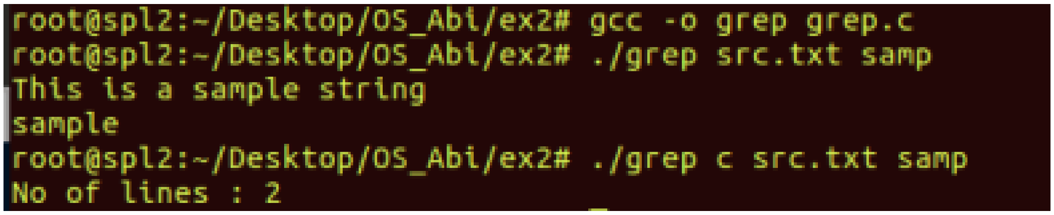
Source code:

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <dirent.h>
#include <fcntl.h>
int main(int argc, char *argv[])
{
    if (argc > 4)
    printf("Too many arguements\n");
    else if (argc < 2)
    printf("Too few arguements\n");
    else
    {
        if (argc == 3)
            {
                    int file_desc = open(argv[1], O_RDONLY);
                    if (file_desc == -1)
                    {
                        printf("File does not exist\n");
                        return 0;
                    }
                    char line[100], buff[1024];
                    int l = 0, i = 0,num;
                    num = read(file_desc, buff, 1024);
                    close(file_desc);
                    while (l < num)
                    {
                            for (i=0; buff[l] != '\n' && l < num; i++, l++)
                                line[i] = buff[l];

                            line[i] = '\0';
                            l++;
                            if (strstr(line, argv[2]))
                                    printf("%s\n", line);
                    }
            }
            else if (argc == 4 && strcmp(argv[1], "c") == 0)
            {
                    //grep -c
                    int file_desc = open(argv[2], O_RDONLY);
                    if (file_desc == -1)
                    {
                        printf("File does not exist\n");
                        return 0;
                    }
                    int cnt = 0;
                    char line[100], buff[1024];
                    int l= 0, i=0,num;
                    num = read(file_desc, buff, 1024);
                    close(file_desc);
                    while (l < num)
                    {
                             for(i=0; buff[l] != '\n' && l <num; i++,l++)
                                line[i] = buff[l];

                            line[i] = '\0';
                            l++;
                            if (strstr(line, argv[3]))
                                    cnt++;
                    }
                    printf("No of lines : %d\n", cnt);
            }
    }
    return0;
}
```

<u>Output:</u>



```
root@spl2:~/Desktop/OS_Abi/ex2# gcc -o grep grep.c
root@spl2:~/Desktop/OS_Abi/ex2# ./grep src.txt samp
This is a sample string
sample
root@spl2:~/Desktop/OS_Abi/ex2# ./grep c src.txt samp
No of lines : 2
```

Learning Outcomes:

Learnt to implement system commands like cp, grep, ls in C using system calls.