# Practical – 2 & Assignment – 3

## Aim:

Implement copy command using open, create, read, write, access and close system call. Be sure to include all necessary error checking including, ensuring the source file exists.
Test your program with following specifications:
a. File extension with .txt, .c, .zip, .exe, .tar
b. Copy the whole directory.

## Program Code:

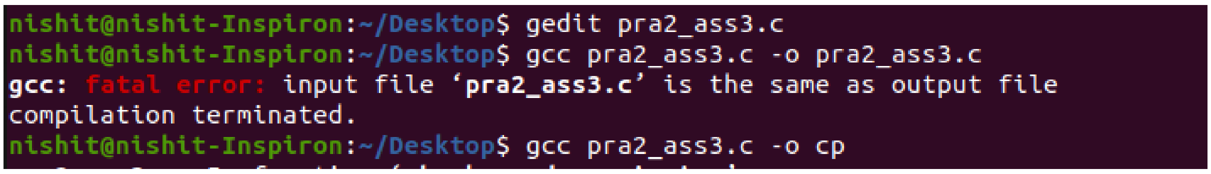
```
#include<stdio.h>
#include<fcntl.h>
#include<stdlib.h>
#include<sys/types.h>
#include<string.h>
#include<dirent.h>
#include <errno.h>
char buffer[1024];


int isFile(const char* name)
{
   DIR* directory = opendir(name);

   if(directory != NULL)
   {
    closedir(directory);
    return 0;
   }

   if(errno == ENOTDIR)
   {
    return 1;
   }

   return -1;
}

void check_read_permission(char *filename){

   int fdrok;
   fdrok=access(filename,R_OK);
   if(fdrok == -1){
      printf("file does not have read access");
   }
```

```
}
void check_write_permission(char *filename){

    int fdwok;
    fdwok=access(filename,W_OK);
    if(fdwok == -1){
        printf("file does not have write access");
    }

}

void copy(int srs,int dest){
    int count;
    while((count=read(srs,buffer,sizeof(buffer)))>0){
        write(dest,buffer,count);
    }
}
void directory_open(char* srsdir,char *destdir){

    DIR *srdr=opendir(srsdir);
    if(srdr==NULL){
       printf("Sorry! source dir does not exist");
       exit(0);
    }

    mkdir(destdir,0777);
    list_dir(srdr,srsdir,destdir);
    closedir(srdr);
}
void list_dir(DIR *dr,char *srcpath,char *destpath){
    //srcpath = ios
    char nest_src_path[100]="";
    char nest_dest_path[100]="";
    struct dirent *de;

    //de->d_name   f1  (ios/ios_new/ios_new_new)
    while((de=readdir(dr))!=NULL){
        if(de->d_type==DT_DIR && strcmp(de->d_name,".")!=0 && strcmp(de-
>d_name,"..")!=0){


         strcpy(nest_src_path,srcpath);
         strcat(nest_src_path,"/");
         strcat(nest_src_path,de->d_name);
         printf("src dir=%s\n",nest_src_path);

         strcpy(nest_dest_path,destpath);
         strcat(nest_dest_path,"/");
         strcat(nest_dest_path,de->d_name);
```

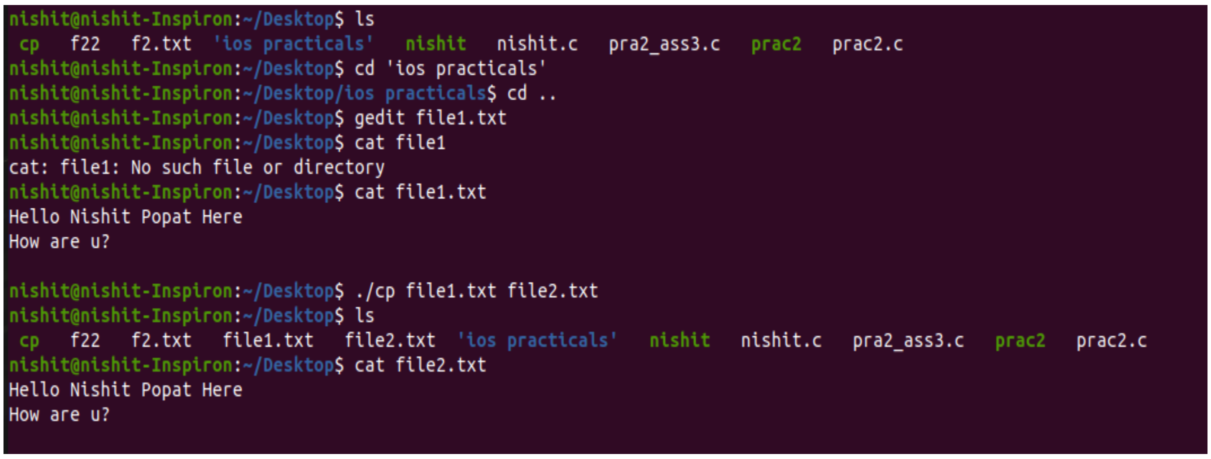
```
          printf("dest dir=%s\n",nest_dest_path);

         directory_open(nest_src_path,nest_dest_path);

        }
        else if(de->d_type!=DT_DIR && strcmp(de->d_name,".")!=0 && strcmp(de->d_name,"..")!=0)
        {
           int srsfd,destfd;
           strcpy(nest_src_path,srcpath);
           strcat(nest_src_path,"/");
           strcat(nest_src_path,de->d_name);
           strcpy(nest_dest_path,destpath);
           strcat(nest_dest_path,"/");
           strcat(nest_dest_path,de->d_name);
           printf("src dir=%s\n",nest_src_path);
           printf("dest dir=%s\n",nest_dest_path);


        srsfd=open(nest_src_path,O_RDONLY);
        if(srsfd== -1){
           printf("source file can't open");
           exit(0);
        }
        destfd=creat(nest_dest_path,0666);
        if(destfd== -1){
           printf("destination file can't open");
           exit(0);
        }
        copy(srsfd,destfd);
        close(srsfd);
        close(destfd);
        }
  }
}



void File_to_dic(char* srs,char* dest){

   int srsfd,destfd;
   char dest_path[100]="";
   srsfd=open(srs,O_RDONLY);
   strcpy(dest_path,dest);
   strcat(dest_path,"/");
   strcat(dest_path,srs);
   destfd=creat(dest_path,0666);
   copy(srsfd,destfd);
   close(srsfd);
   close(destfd);
```

```
}


void file_to_file(char* srs,char* dest){

        int srsfd,destfd;
    srsfd=open(srs,O_RDONLY);
    if(srsfd== -1){
        printf("source file can't open");
        exit(0);
    }
    check_read_permission(srs);
    destfd=creat(dest,0666);
    if(destfd== -1){
        printf("destination file can't open");
        exit(0);
    }
    check_write_permission(dest);
    copy(srsfd,destfd);
    close(srsfd);
    close(destfd);
}



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

    //argv[1] = ios
    //argv[2] = ios_bbatch
    if(isFile(argv[1]) == 1 && isFile(argv[2]) == 0)
    {
        File_to_dic(argv[1],argv[2]);
    }
    else if(isFile(argv[1]) == 0)
    {
        directory_open(argv[1],argv[2]);
    }
    else if(isFile(argv[1]) == 1)
    {
        file_to_file(argv[1],argv[2]);
    }
    else
    {
        printf("Enter Appropriate Argument.\n");
    }
}
```

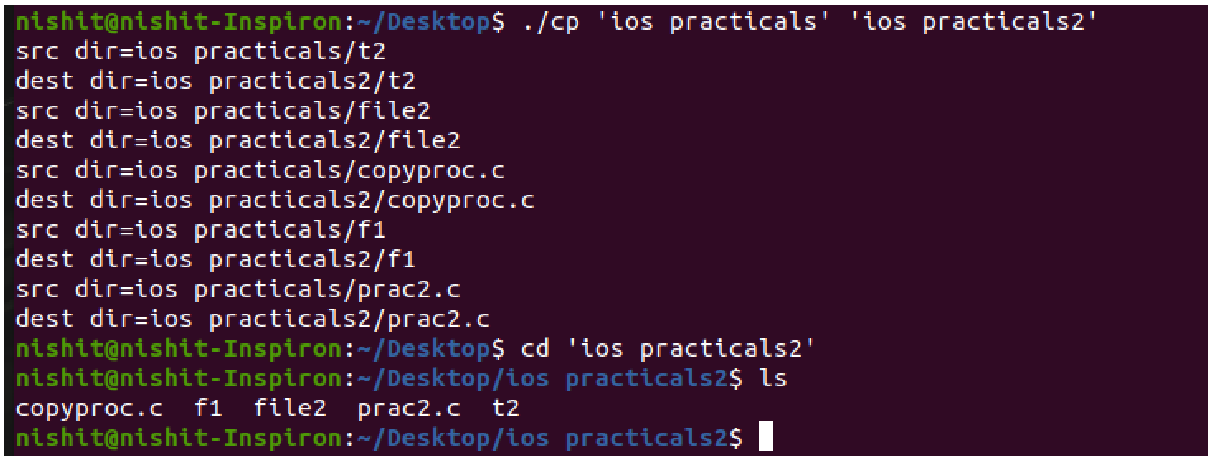
## Output:



```
nishit@nishit-Inspiron:~/Desktop$ gedit pra2_ass3.c
nishit@nishit-Inspiron:~/Desktop$ gcc pra2_ass3.c -o pra2_ass3.c
gcc: fatal error: input file ‘pra2_ass3.c’ is the same as output file
compilation terminated.
nishit@nishit-Inspiron:~/Desktop$ gcc pra2_ass3.c -o cp
```



```
nishit@nishit-Inspiron:~/Desktop$ ls
 cp   f22   f2.txt  'ios practicals'   nishit   nishit.c   pra2_ass3.c   prac2   prac2.c
nishit@nishit-Inspiron:~/Desktop$ cd 'ios practicals'
nishit@nishit-Inspiron:~/Desktop/ios practicals$ cd ..
nishit@nishit-Inspiron:~/Desktop$ gedit file1.txt
nishit@nishit-Inspiron:~/Desktop$ cat file1
cat: file1: No such file or directory
nishit@nishit-Inspiron:~/Desktop$ cat file1.txt
Hello Nishit Popat Here
How are u?

nishit@nishit-Inspiron:~/Desktop$ ./cp file1.txt file2.txt
nishit@nishit-Inspiron:~/Desktop$ ls
 cp   f22   f2.txt   file1.txt   file2.txt  'ios practicals'   nishit   nishit.c   pra2_ass3.c   prac2   prac2.c
nishit@nishit-Inspiron:~/Desktop$ cat file2.txt
Hello Nishit Popat Here
How are u?
```



```
nishit@nishit-Inspiron:~/Desktop$ ./cp 'ios practicals' 'ios practicals2'
src dir=ios practicals/t2
dest dir=ios practicals2/t2
src dir=ios practicals/file2
dest dir=ios practicals2/file2
src dir=ios practicals/copyproc.c
dest dir=ios practicals2/copyproc.c
src dir=ios practicals/f1
dest dir=ios practicals2/f1
src dir=ios practicals/prac2.c
dest dir=ios practicals2/prac2.c
nishit@nishit-Inspiron:~/Desktop$ cd 'ios practicals2'
nishit@nishit-Inspiron:~/Desktop/ios practicals2$ ls
copyproc.c  f1  file2  prac2.c  t2
nishit@nishit-Inspiron:~/Desktop/ios practicals2$ 
```