

o/p :-

cat txt.txt
 This is the txt file
 cat txt.txt new.txt
 source file size is 20b
DONE
 cat new.txt
 diff txt eth si diff

Work-2

- 1.) Write a program to display the file content
 in reverse order using seek system call

Ans: #include < stdlib.h >
~~#include~~ include < stdio.h >
~~#include~~ include < stdlib.h >
~~#include~~ include < stdio.h >
~~#include~~ include < stdio.h >
~~#include~~ include < stdio.h >

```
int main(int argc, char *argv[])
{
    int i;
    char buff[100];
    int flag;
    int fd;
    fd =
```

```
if (argc != 3) {
    if (fprintf(stderr, "usage %s <source> <dest>\n",
```

```
    exit(-1);
}
```

```
if (source != open(<source>, O_RDONLY)) < 0>
    fprintf(stderr, "can't open source\n");
    exit(-1);
}
```

```
if (dest != open(<dest>, O_WRONLY | O_CREAT | O_TRUNC)) < 0>
```

```
    if (fprintf(stderr, "can't create dest in %s\n",  
               dest) == -1);  
}
```

```
    filesize = lseek(source, (off_t)0, SEEK_END);  
    printf("source file %s is %d bytes", filename, filesize);
```

```
    for (i = filesize - 1; i >= 0; i--)
```

```
        lseek(source, (off_t)i, SEEK_SET);
```

```
        if (n = read(source, &buf, 1)) != 1
```

```
            if (fprintf(stderr, "can't read byte %d\n",
```

```
                       buf) == -1);  
    }
```

```
    if (n = write(dest, &buf, 1)) != 1
```

```
        if (fprintf(stderr, "can't write byte %d\n",
```

```
                   buf) == -1);  
    }
```

```
}
```

```
    write(STDOUT_FILENO, "DONE", 5);
```

```
    close(source);
```

```
    close(dest);
```

```
    return 0;
```

```
}
```

Q.P:- Ques 1.1, 1.2, 1.3

Hello, good morning

This is VSP lab

It is also combined with (1)

a

cc pgm.c

1a.out

Hello, good morning

This is VSP lab

It is also combined with (1)

size offset is 92 bytes

Ques 1.2

a) to read file 20 bytes chars from file

b) seek to 10 bytes from beginning and

display 20 characters from there.

c) seek to 10 bytes, from current file offset

and display 20 characters.

d) display the file size.

Ans

#include < stdio.h >

#include < unistd.h >

#include < fcntl.h >

#include < sys/types.h >

int main ()

{ int file20, n;

char buffer[25];

if (file20 = open ("testfile.txt", "r", 000000)) <-1>

printf ("file open error\n");

if (read (file20, buffer, 20) != 20)

printf ("file read failed\n");

else

write (STDOUT_FILENO, buffer, 20);

printf ("\n");

if (lseek (file20, SEEK_SET) < 0)

printf ("seek op failed\n");

if (read (file20, buffer, 20) != 20)

printf ("read failed\n");

else

write (STDOUT_FILENO, buffer, 20);

printf ("\n");

Experiment No.

Date :

```
if (lseek (file, 10, SEEK_CUR) < 0)
    printf ("Seek operation to beginning of file failed")
if (read (file, buffer, 20) != 20)
    printf ("failed");
else
    write (STDOOUT - FILE NO, buffer, 20);
    printf ("In");
if (n == lseek (file, 0, SEEK_END) < 0)
    printf ("failed");
printf ("Size of file is %d", length);
close (file);
return 0;
```

Op: - ./a.out \$1. first

Info for \$1. first
 File Size: 52 bytes
 No. of links: 2
 File mode: 27
 File permission: r-w - r-w - r-w
 The file is not a symbolic link

Q. Write a program to display known details of a file using
 struct structure (At least 5 fields).
 Ans. #include <unistd.h>
 #include <stdio.h>
 #include <sys/stat.h>
 #include <sys/types.h>

int main(int argc, char *argv)
{
 if (argc != 2)
 return 1;
}

struct stat fileStat;
 if (stat(argv[1], &fileStat) < 0)
 return 1;

```
printf ("Information for %s\n", argv[1]);  

printf ("File size: %d bytes\n", fileStat.st_size);  

printf ("Number of links: %d\n", fileStat.st_nlink);  

printf ("File mode: %o\n", fileStat.st_mode);  

printf ("uid :%d", fileStat.st_uid);  

printf ("file permission: %s\n");  

printf ("S_ISDIR (fileStat.st_mode) ? %u : %u\n",  

  S_ISDIR (fileStat.st_mode));  

printf ("fileStat.st_mode & S_IRUSR ? %u : %u\n",  

  fileStat.st_mode & S_IRUSR);  

printf ("fileStat.st_mode & S_IWUSR ? %u : %u\n",  

  fileStat.st_mode & S_IWUSR);  

printf ("fileStat.st_mode & S_IRGRP ? %u : %u\n",  

  fileStat.st_mode & S_IRGRP);  

printf ("fileStat.st_mode & S_IWGRP ? %u : %u\n",  

  fileStat.st_mode & S_IWGRP);
```

Experiment No.

Date :

```
printf ((filestat.st_mode & S_IROTH) ? "r" : "-");
printf ((filestat.st_mode & S_IWOTH) ? "w" : "-");
printf ((filestat.st_mode & S_IXOTH) ? "x" : "-");
printf ("lnln");
printf ("The file %s is a symbolic link\n",
(S_ISLINK(filestat.st_mode)) ? "is" : "is not");
return 0;
```

y

Reprint?

Experiment No.

Date:

(1) $n \neq 2$

Write a program to implement ls -li command which lists the files in a specified directory. Your program should print out 3 columns of files.

十一

~~Touch~~
Ls - Cu
empty
empty

touch empty sta. C
do - li empty tanks

emptyne-c

Q) Write a program to remove empty files from the directory.

```

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <dirent.h>
int main()
{
    DIR *dp;
    struct dirent *dir;
    int fd, n;
    dp = opendir(".");
    if (dp == NULL)
        perror("dir");
    while (dir = readdir(dp))
    {
        if (dir->d_ino == 0)
            continue;
        if (fd = open(dir->d_name, 0, RDRW, 0777))
        {
            n = lseek(fd, 0, SEEK_END);
            if (n < 0)
                perror("lseek");
            if (ftruncate(fd, n) == -1)
                perror("ftruncate");
            close(fd);
        }
    }
}

```

b) ~~5~~ program to demonstrate the creation of hard links and the various properties of hard links. Also, do the same for soft links.

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

Hardlink

1. a. out hard-link .c hardlink
2. hardlink hard-link .c and hardlink
Hard link created

ls -li hard-link .c hardlink
139603660 -rw-rw-rw -2 us6c46 cb0 Feb 15:49 2014

softlink
1. a. out soft-link .c softlink dummyarg
softlink soft-link .c and softlink
soft link created

ls -li soft-link .c softlink
139605635 lrwxrwxrwx 1 cs6c6b 11 Feb 20 softlink
softlink
139605660 -rw-rw-r-- 2 cs6c46 1603 Feb 20
softlink .c

Experiment No.

Date:

#include <iostream.h>
#include <sys/types.h>
#include <sys/stat.h>
int main(int argc , char* argv[])
{ if (argc <= 3)
{ if (link (argv[1] , argv[2]) == 0)
{ cout << "link created" << endl ;
} else
{ cout << "link not created" << endl ;
} } else if (link (argv[1] , argv[2]) == 0)
{ cout << "link created" << endl ;
} else
{ cout << "link not created" << endl ;
} }

7.) Write a C program to copy access and modification time of a file to another file using utime function.

Ans. #include < stdio.h >

#include < sys/types.h >

#include < sys/types.h >

#include < utime.h >

#include < utime.h >

#include < time.h >

#include < fcntl.h >

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

int fd;

struct stat statbuf_1;

struct stat statbuf_2;

struct utimbuf times;

if (stat(argv[1], &statbuf_1) < 0)

printf ("Error\n");

if (stat(argv[2], &statbuf_2) < 0)

printf ("Error\n");

Dept.

gdbt prog1.c
cc prog1.c
cc prog1.c -o prog1
1 ping1 later, but prog. won't give 2 filnames as cmd args
Before copying: ---
Access Time: Thu Apr 19 15:04:46 2018

Modification Time Thu Apr 19 15:04:46 2018

After copying ..
Sun, Jun 17, 2018 12:11:11 2018

Modification Time Tue Feb 20 12:41:11 2018

These Lab 4. txt's time is changed to fossil.html's time

```

printf ("Before copying... \n");
printf ("Access Time is in Modification Time \n");
printf ("Time ( < statbuf.st_ctime), time ( > statbuf.st_mtime),
       st_mtime2);
```

time, modtime = statbuf.st_mtime, st_mtime;

times.ctime > statbuf.st_mtime;

if (ctime(&arg[1], ×) < 0)

printf ("On Error copying time \n");

if (stat (&arg[1], &statbuf) < 0)

printf ("Error \n");

printf ("After copying... \n");

printf ("Access Time is in Modification Time \n");
 time (< statbuf.st_ctime), time (> statbuf.st_mtime),
 st_mtime2);

100

Experiment No.

Date: _____

Output:

```
gdbt prog.c
cc prog.c -o prog
./prog
Output:
in f1():
global = 0.5, autovar = 9.6,0, regvar = 9.7, valvar = 9.8, statvar = 9.9
after longjmp:
global = 10.000, autovar = 9.6, regvar = 9.7, valvar = 9.8,
statvar = 9.9
```

```
8.) Write a C program to illustrate effect of setting and
longjmp functions on register and volatile variables
a)
#include <setjmp.h>
#include <stdio.h>
#include <stdlib.h>
static void f1(int, int, int, float);
static void f2 (void);
int main (void)
{
    int autovar;
    register int regvar;
    volatile int valvar;
    static int statvar;
    int main (void)
    {
        int autovar;
        register int regvar;
        volatile int valvar;
        static int statvar;
        if (longjmp(jmp_buffer) != 0)
        {
            printf ("after longjmp: (%d\n",);
            printf ("global (%.1f), autovar = %.1f, regvar = %.1f,
valvar = %.1f, statvar = %.1f\n", global, autovar,
regvar, valvar, statvar);
            exit(0);
        }
    }
}
```

global = 95; autoval = 96; regival = 97; uolval = 98;
statval = 99;

f1(autoval, regival, uolval, statval); /* here returns */
exit(0);

y
static void f1(int i, int j, int k, int l)

{

printf("in f1(): %n");

printf("global = %.d, autoval = %.d, regival = %.d,
statval = %.d\n", global, i, j, k, l);

global = 10000;

j = 10000;

f2();

y

static void f2(void)

{ longjmp(jmpbuffer, 1);

y

④

Output
/Hari1. txt
Hello this is a testfile

Op: -
.1a.out test1.txt test2.txt
File 1 opened successfully
File 2 opened successfully
File size of file 1 obtained
File copied successfully

cat test2.txt

Hello file is testfile!

Work-5

Q.) C program to simulate copy command by accepting
the filenames from commandline. Report all errors,
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
int main (int argc, char *argv[])
{
char buf[100];
int fd1, fd2;
if (argc < 3)
size = readfd1, writefd2; i;
fd1 = open (arg[1], O_RDONLY); //open file 1
if (fd1 == -1)
printf ("\"ERROR IN OPENING FILE : FILE DOES
NOT EXIST\\n\"");
else
printf ("\"FILE1 OPENED SUCCESSFULLY\\n\"");
fd2 = open (arg[2], O_WRONLY | O_CREAT | O_TRUNC,
0666); //
if (fd2 == -1)
printf ("\"ERROR IN OPENING FILE\\n\"");
else
printf ("\"FILE2 OPENED SUCCESSFULLY\\n\"");

```
size = lseek (fd1, 0L, SEEK-END);
```

```
if (size == -1)
```

```
    printf ("ERROR: COULD NOT OBTAIN FILE SIZE\n");
```

```
else
```

```
    printf ("FILE SIZE OF FILE1 OBTAINED\n");
```

```
ret = lseek (fd1, 0L, SEEK-SET);
```

```
if (ret == -1)
```

```
    printf ("RETRACE FAILED\n");
```

```
readdata = read (fd1, buf, size);
```

```
if (readdata == -1)
```

```
    printf ("ERROR IN READING FILE CONTENTS\n");
```

```
writedata = write (fd2, buf, size);
```

```
if (writedata != size)
```

```
    printf ("ERROR IN COPYING FILE\n");
```

```
else
```

```
    printf ("FILE COPIED SUCCESSFULLY\n");
```

```
return 0;
```

```
}
```

~~Experiment No.~~
~~Date:~~

Explanation and Q.P.:

If a child is created properly, the parent process is killed and the child is orphaned and will run in the background.

However, if child is not created, the process is made the parent leader and becomes parent of itself.

```
10.) C program to check zombie status of a process
#include <sys/types.h>
#include <sys/conf.h>
#include <sys/conf.h>

int main (void) {
    pid_t pid;
    if ((pid = fork ()) < 0)
        err - sys ("fork error");
    else if (pid == 0) {
        if ((pid = fork ()) < 0)
            err - sys ("fork error");
        else if (pid > 0)
            exit (0);
        sleep (2);
        print ("second child, parent pid - '%d'\n", (long) getpid);
        exit (0);
    }
}
```

Q/P:

- /a.out
output from child
- output from parent
- /a.out
output from parent
- output from child

Experiment No.

Date:

11) C program to demonstrate New condition among parent and child processes.

#include <stdio.h>

#include <stdlib.h>

static void charatime (char *);

```
int main (void) {  
    pid + pid';  
    if (pid == fork ()) < 0  
        printf ("fork error");  
    else if (pid == 0) {  
        charatime ("Output from child in");  
        exit (0);  
    }  
}
```

```
static void charatime (char *str) {  
    char *ptr; int c;  
    setbuf (stdout, NULL);  
    for (ptr = str; *ptr != '\0'; ptr++)  
        puts (c, *ptr);  
}
```

Experiment No.

Date :

12) Write a C program which initializes itself as a daemon process.

```

#include <sys/types.h>
#include <sys/stat.h>
#include <sys/conf.h>
#include <sys/conf.h>
void daemonize() {
    pid = fork();
    if (pid < 0)
        if (fork() < 0)
            if (chdir("/") < 0)
                perror("Error changing directory\n");
            if (setsid() < 0)
                printf("Error creating session\n");
            exit(0);
    }
    umask(0);
    if (chdir("/") < 0)
        perror("Error changing directory\n");
    if (setsid() < 0)
        printf("Error creating session\n");
    if (fork() < 0)
        printf("Error creating process\n");
    exit(0);
}

int main() {
    daemonize();
    system("ps -axg > /dev/null");
}

```

Q: .1a.out
Ctrl+C
Interrupt received
Ctrl+C
Exit

Experiment No.

Date:

13) Write a program using disruption system call which calls a signal handler on SIGINT signal and then meet the default action of the signal signal.

```
#include <sys/types.h>
#include <sys/conf.h>
#include <sys/signal.h>
void handler(int val)
{
    printf("Signal Received\n");
    sigset(SIGINT, handler);
    sigdel(SIGINT, handler);
}

main()
{
    sigset(SIGINT, handler);
    sleep(1);
    printf("Program is Happening!\n");
    sleep(1);
}
```

Op: If choice is 1, it works normally. If choice is 2, then before the tenth time if choice is 1, it does not interrupt.

Q) Write a C program:
(i) Which takes a signal handler on SIGINT signal
and then quit the default action of the SIGINT
signal.
(ii) Which ignores SIGINT signal and then quit the default
action of SIGINT.

```

#include <statis.h>
#include <unctrl.h>
#include <stroref.h>
void callalt();
printf("Interrupt received\n");
(void)signal(SIGINT, SIG_DEL);
}
int main()
{
    char i=0;
    printf("Enter choice\n");
    scanf("%d", &i);
    switch(i)
    {
        case 1: (void)signal(SIGINT, callalt); break;
        case 2: (void)signal(SIGINT, SIG_DEL); break;
    }
    while(1)
    {
        sleep(1);
        printf("Press ctrlC\n");
        i++;
        if(i>10 kch=2)
            (void)signal(SIGINT, SIG_DEL);
    }
}

```