

EX NO: 2	SIMULATION OF UNIX COMMANDS (GREP, CP, LS)
DATE:	

A. SIMULATION OF GREP UNIX COMMAND

AIM:

To write a c program to simulate grep unix command.

ALGORITHM:

Step 1: Include necessary header files.

Step 2: Make necessary declarations.

Step 3: Read the file name from the user and open the file in the read only mode.

Step 4: Read the pattern from the user.

Step 5: Read a line of string from the file and search the pattern in that line.

Step 6: If pattern is available, print the line.

Step 7: Repeat the step 4 to 6 till the end of the file.

PROGRAM:

```

#include<stdio.h>
#include<unistd.h>
#include<stdlib.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include<string.h>
#include <fcntl.h>

void match_pattern(char *argv[])
{
    int fd,r,j=0;    char
temp,line[100];
    if((fd=open(argv[2],O_RDONLY)) != -1)
    {
        while((r=read(fd,&temp,sizeof(char)))!= 0)
        {
            if(temp!='\n')
            {
                line[j]=temp;
                j++;
            }
            else
            {
                if(strstr(line,argv[1])!=NULL)
                printf("%s\n",line);
                memset(line,0,sizeof(line));
                j=0;
            }
        }
    }
}

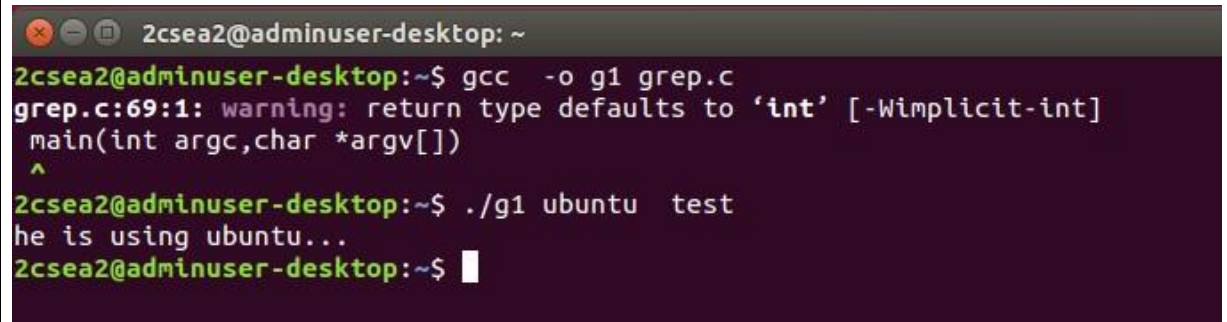
main(int argc,char *argv[])
{
    struct stat
stt;
    if(argc==3)
    {
        if(stat(argv[2],&stt)==0)
            match_pattern(argv);
        else
        {
            perror("stat()");
            exit(1);
        }
    }
}

```


test.txt

he is using ubuntu...

OUTPUT:



```
2csea2@adminuser-desktop: ~  
2csea2@adminuser-desktop:~$ gcc -o g1 grep.c  
grep.c:69:1: warning: return type defaults to 'int' [-Wimplicit-int]  
  main(int argc,char *argv[])  
  ^  
2csea2@adminuser-desktop:~$ ./g1 ubuntu test  
he is using ubuntu...  
2csea2@adminuser-desktop:~$
```

RESULT:

Thus the program for simulating grep unix command has been executed and verified successfully.

B. SIMULATION OF CP UNIX COMMAND

AIM:

To write a c program to simulate cp unix command.

ALGORITHM:

Step 1: Include necessary header files for manipulating directory.

Step 2: Declare and initialize required objects.

Step 3: Read matrix rows and columns.

Step 4: Read the matrix elements from the user.

Step 5: Compute sum of diagonal matrix.

Step 6: Stop the program.

PROGRAM:

```
#include<stdio.h>
void get_matrix(int m[20][20],int n)
{
int i,j;
for(i=0;i<n;i++)
for(j=0;j<n;j++)
{
scanf("%d",&m[i][j]);
} }
void print_matrix(int m[20][20],int n)
{
int i,j;
for(i=0;i<n;i++)
{ printf("\n\t");
for(j=0;j<n;j++)
)
{
printf("%d\t",m[i][j]);
```


}}

```
printf("\n");
} int main()
{
int m[20][20],n,i,trace=0;
printf("\n enter the no.of.R&c for the square matrix (n*n):"); scanf("%d",&n);
printf("\n enter the elements for matrix (%d*%d)\n",n,n);
get_matrix(m,n); printf("\n matrix read:");
print_matrix(m,n); for(i=0;i<n;i++)
{
trace = trace +m[i][i];
}
printf("\n sum of diagonal elements(trace) of the  matrix :%d\n",trace); return
0;
}
```

OUTPUT:

```
2csea2@adminuser-desktop:~$ cc test2.c
2csea2@adminuser-desktop:~$ ./a.out

enter the no.of.R&c for the square matrix (n*n):3

enter the elements for matrix (3*3)
1 2 3
4 5 6
7 8 9

matrix read:
      1      2      3
      4      5      6
      7      8      9

sum of diagonal elements(trace) of the matrix :15
2csea2@adminuser-desktop:~$
```

RESULT:

Thus the program for simulating cp unix command has been executed and verified successfully.

C. SIMULATION OF LS UNIX COMMAND

AIM:

To write a c program to simulate ls unix command.

ALGORITHM:

Step 1: Include necessary header files for manipulating directory.

Step 2: Declare and initialize required objects.

Step 3: Read the directory name form the user.

Step 4: Open the directory using opendir() system call and report error if the directory is not

available.

Step 5: Read the entry available in the directory.

Step 6: Display the directory entry ie., name of the file or sub directory.

Step 7: Repeat the step 6 and 7 until all the entries were read.

PROGRAM:

```
#include<stdio.h>
#include<dirent.h>

int main(void)
{
    struct dirent *de;
    DIR *dr = opendir(".");
    if(dr==NULL)
    {
        printf("could not open current directory");
        return 0;
    }
    while ((de=readdir(dr))!=NULL)
        printf("%s\t",de->d_name);
    closedir(dr);
    return 0;
}
```

OUTPUT:

```
2csea2@adminuser-desktop:~$ cc ls1.c
2csea2@adminuser-desktop:~$ ./a.out
.xsession-errors    examples.desktop    ls1.c    .cache    .sjf.c.swp    semaphore.c    Documents    Videos    Pictures    Public
.profile            a.out    .viminfo    .ICEauthority    .gconf    .local    .rr.c.swp    .bashrc    ls.c    ..    .Xauthority    .    .bash_
logout    student    .factorial.sh.swp    fcfs.c    Desktop    .xsession-errors.old    .config    .bash_history    factorial.sh    Downloads    Music
dmrc    .semaphore.c.swo    Templates    fcfs.c~    .job.c.swp    .shortest.c.swp    2csea2@adminuser-desktop:~$
```

RESULT:

Thus the program for simulating ls unix command has been executed and verified successfully.
