

16. Write a C program that behaves like a shell (command interpreter). It has its own prompt say "NewShell\$". Any normal shell command is executed from your shell by starting a child process to execute the system program corresponding to the command. It should additionally interpret the following command.

i) count c <filename> - print number of characters in file

ii) count w <filename> - print number of words in file

iii) count l <filename> - print number of lines in file

```
#include<stdio.h>
#include<sys/types.h>
#include<sys/stat.h>
#include<unistd.h>
#include<dirent.h>
#include<fcntl.h>

void count(char c, char *fn)
{
    int lc=0,wc=0,cc=0,handle;
    char ch;
    if((handle=open(fn,O_RDONLY))== -1)
    {
        printf("File %s not found\n",fn);
        return;
    }
    while(read(handle,&ch,1)!=0)
    {
        if(ch=='\n')
            lc++;
        if(ch==' ' || ch == '\t' || ch == '\n' || ch ==
'\0')
            wc++;
        else
            cc++;
    }
    close(handle);
}
```

```

switch(c)
{
case 'c':
    printf("Total No.of Characters = %d\n",cc);
    break;
case 'w':
    printf("Total No.of Words = %d\n",wc);
    break;
case 'l':
    printf("Total No.of Lines = %d\n",lc);
    break;
}
}

```

```

int main()
{
    char command[80],t1[20],t2[20],t3[20],t4[20];
    int n;
    system("clear");
    while(1)
    {
        printf("\nmyShell$");
        fflush(stdin);
        fgets(command,80,stdin);
        n = sscanf(command,"%s %s %s %s",t1,t2,t3,t4);
        switch(n)
        {
            case 1:
                if(!fork())
                {
                    execlp(t1,t1,NULL);
                    perror(t1);
                }
                break;
            case 2:
                if(!fork())
                {
                    execlp(t1,t1,t2,NULL);
                    perror(t1);
                }
                break;
            case 3:
                if(strcmp(t1,"count")==0)
                    count(t2[0],t3);

```

```

        else
        {
            if(!fork())
            {
                execlp(t1,t1,t2,t3,NULL);
                perror(t1);
            }
        }
        break;
case 4:
    if(!fork())
    {
        execlp(t1,t1,t2,t3,t4,NULL);
        perror(t1);
    }
}
}
return 0;
}

/*
shubham@shubham:~/Documents/C Programs$ gcc 16shellcount.c -o
test
16shellcount.c: In function 'main':
16shellcount.c:46:2: warning: implicit declaration of function
'system' [-Wimplicit-function-declaration]
    system("clear");
    ^~~~~~
16shellcount.c:70:7: warning: implicit declaration of function
'strcmp' [-Wimplicit-function-declaration]
    if(strcmp(t1,"count")==0)
       ^~~~~~
shubham@shubham:~/Documents/C Programs$ ./test

myShell$count c demo.docx
Total No.of Characters = 51

myShell$count w demo.docx
Total No.of Words = 9

myShell$count l demo.docx
Total No.of Lines = 6
*/

```