

```

/*
 * shell.c
 *
 * Created on:
 * Author: root
 */
#include<stdio.h>
#include<string.h>
#include<sys/types.h>
#include<sys/wait.h>
#include<sys/stat.h>
#include<dirent.h>
#include<fcntl.h>
#include<unistd.h>
#include<stdlib.h>
char ss[40],tok1[10],tok2[10],tok3[10],tok4[10];
pid_t pid;

int fp;
char ch[1],ch1[100];
void sep()
{
    strcpy(tok1,"\0");
    strcpy(tok2,"\0");
    strcpy(tok3,"\0");
    strcpy(tok4,"\0");
    sscanf(ss,"%s%s%s%s",tok1,tok2,tok3,tok4);
}

void count()

```

```

{
    int cc=0,wc=0,lc=0,j;
    fp=open(tok3,O_RDONLY);
    if(fp== -1)
        printf("error");
    else
    {
        while(read(fp,ch,1)!=0)
        {
            printf("%c",ch[0]);
            if(ch[0]==' ' || ch[0]=='\n' || ch[0]=='\t')
                wc++;
            else
                cc++;
            if(ch[0]=='\n')
                lc++;
        } //while
        wc++;lc++;
        close(fp);
        if(strcmp(tok2,"c")==0)
            printf("total number of characters= %d\n",cc);
        else if(strcmp(tok2,"w")==0)
            printf("total number of words= %d\n",wc);
        else if(strcmp(tok2,"l")==0)
            printf("total number of lines= %d\n",lc);
        else if(strcmp(tok2,"cw")==0)
            printf("total number of characters= %d and
words=%d\n",cc,wc);
        else if(strcmp(tok2,"cl")==0)
            printf("total number of characters= %d and lines=%d\n",cc,lc);
        else if(strcmp(tok2,"wl")==0)

```

```

printf("total number of words= %d and lines=%d\n",wc,lc);
else if(strcmp(tok2,"wl")==0)
    printf("total number of words= %d and lines=%d\n",wc,lc);
else if(strcmp(tok2,"cwl")==0)
    printf("total number of characters= %d,words=%d,lines=%d\n",cc,wc,lc);
    }//else

}

//count
void typeline()
{
    if(strcmp(tok4,"\0")==0)
    {
        int charcount = 0,newlinechar1 = 0,newlinechar2 = 0,newlinechar;
        int ct,count=0;
        fp=open(tok3,O_RDONLY);
        if(fp== -1)
            printf("error");
        else
        {
            if(strcmp(tok2,"a")==0)
            {
                while(read(fp,ch,1))
                    printf("%c",ch[0]);

            }
            //if
            if(strcmp(tok2,"r")==0)
            {
                lseek(fp,0,SEEK_SET);
                while(read(fp,ch,1))
                {
                    if(ch[0]=='\n')
                        newlinechar1++;
                }
            }
        }
    }
}

```

```

    }//while
    newlinechar2=0;
    while(newlinechar1>=0)
    {
        newlinechar2=0;
        lseek(fp,0,SEEK_SET);
        while(read(fp,ch,1))
        {
            if(ch[0]=='\n')
                newlinechar2++;
            if(newlinechar2==newlinechar1)
                printf("%c",ch[0]);
        }
    }//while
    newlinechar1--;
    if(newlinechar1==0)
        printf("\n");
}
}

/***** for n>0*****/
if(atoi(tok2)>0)
{
    ct=atoi(tok2);
    while(read(fp,ch,1))
    {
        printf("%c",ch[0]);
        if(ch[0]=='\n')
            count++;
        if(ct==count)
            break;
    }
}
}

```

```
//if
```

```
/*for n<0*****/
```

```
if(atoi(tok2)<0)
```

 $\{$

```
count=0;
```

```
ct=atoi(tok2);
```

```
while(read(fp,ch,1))
```

 $\{$

```
if(ch[0]=='\n')
```

```
count++;
```

```
//while
```

```
int c1=count+ct;
```

```
count=0;
```

```
lseek(fp,0,SEEK_SET);
```

```
while(read(fp,ch,1))
```

 $\{$

```
if(ch[0]=='\n')
```

```
count++;
```

```
if(c1<count)
```

```
printf("%c",ch[0]);
```

```
//while
```

```
//if
```

```
//else
```

}

```
//typeline
```

```
void list()
```

 $\{$

```
int i;
```

```

DIR *dp;

struct dirent *entry;

struct stat info;

char pname[40];

puts(tok1);

puts(tok2);

puts(tok3);

if((dp=opendir(tok3))==NULL)
    printf("Open directory error");
else
{
    if(strcmp(tok2,"f")==0)
    {
        while(entry=readdir(dp))
        {
            if((strcmp(entry->d_name,"")==0) || (strcmp(entry-
>d_name,"..")==0))
                continue;
            strcpy(pname,tok3);
            strcat(pname,"/");
            strcat(pname,entry->d_name);
            if(stat(pname,&info)==0)
            {
                if(S_ISREG(info.st_mode))
                    printf("%s\n",entry->d_name);

            }//if
        }//while
    }//if
    else if(strcmp(tok2,"n")==0)
    {

```

```

        i=0;
        while(entry=readdir(dp))
        {
            if((strcmp(entry->d_name,".")==0) || (strcmp(entry-
>d_name,"..")==0))
                continue;
            i++;
        }//while
        printf("Number of entries in the directory = %d",i);

    }//else if
    else if(strcmp(tok2,"i")==0)
    {
        printf("filename: inode: devid: mode: links: uid: gid: size: created:\n");
        while(entry=readdir(dp))
        {
            if((strcmp(entry->d_name,".")==0) || (strcmp(entry->d_name,"..")==0))
                continue;
            strcpy(pname,tok3);
            strcat(pname,"/");
            strcat(pname,entry->d_name);
            if(stat(pname,&info)!=0)
                printf("stat error");
            else
            {
                printf("%s",entry->d_name);
                printf("%d",(int)info.st_ino);
                printf("%d",(int)info.st_dev);
                printf("%08x",info.st_mode);
                printf("%d",info.st_nlink);
                printf("%d",(int)info.st_uid);
            }
        }
    }
}

```

```

        printf("%d", (int)info.st_gid);

        printf("%ld", info.st_size);

        printf("%s\n", ctime(&info.st_ctim));
    } //else

    } //while
} //else if
closedir(dp);
} //else

} //list
int main()
{
    while(1)
    {
        printf("\nMyShell $");
        gets(ss);
        sep();
        pid=fork();
        if(pid==0)
        {
            if(strcmp(tok1, "count")==0)
                count();

            else if(strcmp(tok1, "typeline")==0)
                typeline();

            else if(strcmp(tok1, "list")==0)
                list();

            else if(strcmp(tok1, "exit")==0)
                exit(0);
        }
    }
}

```



```

        else

            printf("BAD COMMAND");

    } //if
    else
        if(pid!=0)
        {
            waitpid(pid,NULL,0);
            printf("parent process completed");
            exit(0);
        } //if
    } //while
} //main

```

/*****output*****/

MyShell \$count c abc.txt

aaaaaaaa bbbbbbbbbbbbbbbb

cccccc ddddddddddd

eeeeeeee

fffffffffff

ggggggggggggggtotal number of characters= 74

MyShell \$count w abc.txt

aaaaaaaa bbbbbbbbbbbbbbbb

cccccc ddddddddddd

eeeeeeee

fffffffffff

ggggggggggggggtotal number of words= 7

MyShell \$count l abc.txt

aaaaaaaa bbbbbbbbbbbbbbbb

cccccc ddddddddddd

eeeeeeee

fffffffffff

ggggggggggggggtotal number of lines= 5

MyShell \$list f Debug

list

f

Debug

shell.d

objects.mk

shell

shell.o

makefile

subdir.mk

sources.mk

MyShell \$list n Debug

list

n

Debug

Number of entries in the directory = 7

MyShell \$list i Debug

list

i

Debug

filename: inode: devid: mode: links: uid: gid: size: created:

shell.d51121002054000081a410028Fri Dec 14 23:00:25 2012

objects.mk51120742054000081a4100230Mon Dec 12 05:34:40 2011

shell51138382054000081ed10070781Fri Dec 14 23:00:25 2012

shell.o51128002054000081a410063384Fri Dec 14 23:00:25 2012

makefile51120732054000081a4100949Sat Dec 15 10:28:36 2012

subdir.mk51120722054000081a4100646Sat Dec 15 10:28:36 2012

sources.mk51120702054000081a4100390Sat Dec 15 10:28:36 2012

MyShell \$exit

parent process completedparent process completedparent process completedparent process
completed

*****/