

Unix/Linux 体系及编程实验

实验报告

实验二：Linux 环境下的 C 语言实训

姓名：李毓琪

学号：2220191195

专业班级：网络工程 2019-2

选课序号：86

一、实验目的

1. 用 C/C++构造一个简单的 shell。。
2. 理解 shell 程序的功能。
3. 学会 shell 的使用。
4. 掌握基本的 makefile 方法。

二、实验内容

基本任务 1：用 C/C++编写一个简单的 shell 程序，实现以下基本的命令。

- 1) 浏览目录和文件的各种属性 ls（可以不支持参数）。
- 2) 回现命令 echo。
- 3) 显示文件内容 cat。
- 4) 创建目录 mkdir。
- 5) 删除文件 rm。
- 6) 切换目录 cd。
- 7) 显示当前目录 pwd。
- 8) 文字统计 wc。

基本任务 2：每一条命令单独对应一个源程序文件，不允许所有命令一个源文件。写一个 makefile 来管理这些源文件。

基本任务 3：写清楚 make 编译和运行的过程。

选做任务

任何你想实现的其它命令 提供友好的人机界面:例如提供命令列表、命令查询、命令帮助等。 Advanced shell 功能, 例如:

- 1) 输入输出重定向, 例如 `./a.out > output.dat`。
- 2) 支持多管道命令, 例如 `./a.out | ./b.out | ./c.out` 。
- 3) 支持后台运行命令, 例如 `ls | wc &`。
- 4) 其它:参考 GNU bash 提供的功能。

三、实验步骤

1. 通过 vim 编写源程序代码。
2. 通过 g++/gcc 对源程序进行编译测试。
3. 通过 vim 编写 makefile 文件。
4. 通过 make 创建可执行文件。

四、实验结果

cd.c

```
#include <sys/stat.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <dirent.h>
#include <unistd.h>
#include <pwd.h>
char* get_mode(mode_t m, char* str)
{
    if(S_ISREG(m))
        strcpy(str, "-");
    else if(S_ISDIR(m))
        strcpy(str, "d");
    else if(S_ISCHR(m))
        strcpy(str, "c");
    else if(S_ISBLK(m))
        strcpy(str, "b");
    else if(S_ISFIFO(m))
        strcpy(str, "f");
    else if(S_ISLNK(m))
        strcpy(str, "l");
    else if(S_ISSOCK(m))
        strcpy(str, "n");
    strcat(str, m&S_IRUSR?"r":"-");
    strcat(str, m&S_IWUSR?"w":"-");
    strcat(str, m&S_IXUSR?"x":"-");
    strcat(str, m&S_IRGRP?"r":"-");
    strcat(str, m&S_IWGRP?"w":"-");
    strcat(str, m&S_IXGRP?"x":"-");
    strcat(str, m&S_IROTH?"r":"-");
    strcat(str, m&S_IWOTH?"w":"-");
    strcat(str, m&S_IXOTH?"x":"-");
    return str;
}

int ls()
{
    DIR* dp=opendir(".");
    if(NULL == dp)
    {
        perror("opendir");
        return -1;
    }
    struct dirent* de=readdir(dp);
    for(; de; de=readdir(dp))
    {
        if('.')==de->d_name[0]) continue;
        struct stat s;
        int ret = lstat(de->d_name, &s);
        if(0 > ret)
        {
            perror("stat");
            return -1;
        }
        char str[11] = {};
        printf("%s ", get_mode(s.st_mode, str));
        printf("%5ld ", s.st_size);
        printf("%s\t", de->d_name);
        printf("\n");
    }
    closedir(dp);
}
```

cat.c

```
#include <stdio.h>
int cat( char * argv){
    FILE *fp = fopen(argv,"r");
    int read_ret;
    while(!feof(fp)){
        read_ret = fgetc(fp);
        fputc(read_ret,stdout);
    }
    return 0;
}
```

clear.c

```
#include <stdio.h>
int clear()
{
    printf("\033[ 2J");
    printf("\033[ 0; 0H");
    return 0 ;
}
```

echo.c

```
#include<stdio.h>
int echo(int argc, char* argv[]){
    int i;
    for(i = 1 ; i < argc ; i++){
        printf("%s ",argv[i]);
    }
    printf("\n");
    return 0;
}
```

help.c

```
#include"stdio.h"
int help(){
    printf("\t👉 This is the Help document 👉\n");
    printf("\t\tls: List directory contents\n");
    printf("\t\tclear: Clear the terminal screen\n");
    printf("\t\tcd: Change directory\n");
    printf("\t\trm: Remove directory entries\n");
    printf("\t\tmkdir: Make directories\n");
    printf("\t\twc: Word count\n");
    printf("\t\tcat: Concatenate and print files\n");
    printf("\t\techo: Write arguments to the standard output\n");
    printf("\t\tpwd: Return working directory name\n");
    printf("\t👉 All command couldn't carry parameters 👉\n");
    return 0;
}
```

mkdir.c

```
#include<sys/stat.h>
#include<sys/types.h>
#include<stdio.h>
int mymkdir(char argv[]){
int i;
    if(mkdir(argv,0775)){
        printf("error! This File Has Exists! Or Wrong Path\n");
    }
}
```

mycd.c

```
#include<stdio.h>
#include<unistd.h>
int cd(char argv[]){
    if(chdir(argv) == -1){
        printf("Direction Not Found\n");
        return -1;
    }
    return 0;
}
```

pwd.c

```
#include<stdio.h>
#include<unistd.h>
#define MAXPATH 50
int pwd(char * path){
    puts(getcwd(path,MAXPATH));
    return 0;
}
```

rm.c

```
#include<stdio.h>
#include<unistd.h>
int rm(char argv[]){
    if(unlink(argv)==-1){
        printf("The File has't Exists");
    }
    return 0;
}
```

WC.C

```
#include<stdio.h>
#include<stdlib.h>
int wc(char argv[]){
    FILE *fp;
    if((fp = fopen(argv,"r")) == NULL ){
        printf(" Cannot Open The File!\n");
        return -1;
    }
    int line = 1;
    int word = 0;
    int character= 0;
    char c;
    printf(" FileName: %s\n", argv);
    while(!feof(fp)){
        int tmp = 0;
        while ((c = getc(fp)) != EOF)
        {
            if (c == '\n')
            {
                line++;
            }

            if (c != '\n' && c != '\t' && c != ' ')
            {
                tmp = 1;
            }
            else if (tmp == 1)
            {
                word++;
                tmp = 0;
            }
            character++;
        }
    }
    if(tmp==1){
        word++;
    }
    fclose(fp);
    printf(" WordNumber: %d\n", word);
    printf(" LineNumber: %d\n", line);
    printf(" CharacterNumber: %d\n", character);
    return 0;
}
```

将上述的源码在 main.h 中引用一下

```
[stu1195@ubuntu:~/lab2$ cat main.h
#include"ls.c"
#include"pwd.c"
#include"rm.c"
#include"wc.c"
#include"mkdir.c"
#include"echo.c"
#include"mycd.c"
#include"cat.c"
#include"clear.c"
#include"help.c"
```

main.c

```
#include "main.h"
#include "stdio.h"
#include "string.h"
#include "iostream"
#define MAX 128
int main(){
    printf("Welcome to Mybash.\n");
    printf("Now you can use some basic command like 'ls'.\n");
    printf("put 'help' to get more command\n");
    while(1){
        printf(">>>");
        char buff[128]={0};
        fgets(buff, 128, stdin);
        buff[strlen(buff)-1]=0;
        char *s=strtok(buff, " ");
        if(s==NULL)
        {
            continue;
        }
        char* myargv[MAX] = {0};
        myargv[0]=s;
        int i = 1;
        while((s=strtok(NULL, " "))!=NULL)
        {
            myargv[i++]=s;
        }
        if(strcmp(myargv[0], "help") == 0){
            help();
        }else if(strcmp(myargv[0], "ls") == 0){
            ls();
        }else if(strcmp(myargv[0], "pwd") == 0){
            pwd(myargv[0]);
        }else if(strcmp(myargv[0], "exit") == 0){
            exit(-1);
        }else if(strcmp(myargv[0], "cat") == 0){
            cat(myargv[1]);
        }else if(strcmp(myargv[0], "cd") == 0){
            cd(myargv[1]);
        }else if(strcmp(myargv[0], "echo") == 0){
            echo(i, myargv);
        }else if(strcmp(myargv[0], "wc") == 0){
            wc(myargv[1]);
        }else if(strcmp(myargv[0], "rm") == 0){
            rm(myargv[1]);
        }else if(strcmp(myargv[0], "mkdir") == 0){
            mymkdir(myargv[1]);
        }else if(strcmp(myargv[0], "clear") == 0){
            clear();
        }else{
            printf("Wrong Command\n");
        }
    }
    return 0;
}
```


利用 vim 编写 makefile 文件

```
2220191195_mybash: main.o
    g++ main.o -o 2220191195_mybash
main.o: main.c
    g++ -c main.c -o main.o
.PHONY: clean
clean:
    rm *.o
```

使用 make 语句创建可执行文件

```
[stu1195@ubuntu: ~/lab2$ make
g++ -c main.c -o main.o
```

使用./2220191195_mybash 运行 bash

```
[stu1195@ubuntu: ~/lab2$ ./2220191195_mybash
Welcome to Mybash.
Now you can use some basic command like 'ls'.
put 'help' to get more command
>>>
```

在里面可以输入命令

```
[stu1195@ubuntu: ~/lab2$ ./2220191195_mybash
Welcome to Mybash.
Now you can use some basic command like 'ls'.
put 'help' to get more command
[>>>pwd
/home/stu1195/lab2
[>>>ls
-rw-rw-r-- 572 help.c
-rw-rw-r-- 171 cat.c
-rw-rw-r-- 180 mkdir.c
-rw-rw-r-- 856 wc.c
-rw-rw-r-- 171 mycd.c
-rwxrwxr-x 18896 2220191195_mybash
-rw-rw-r-- 11208 main.o
-rw-rw-r-- 120 pwd.c
-rw-rw-r-- 125 makefile
-rw-rw-r-- 1173 main.c
-rw-rw-r-- 132 rm.c
-rw-rw-r-- 1373 ls.c
-rw-rw-r-- 164 main.h
-rw-rw-r-- 89 clear.c
-rw-rw-r-- 140 echo.c
[>>>wc cat.c
FileName: cat.c
WordNumber: 22
lineNumber: 11
CharacterNumber: 171
>>>
```

五、实验分析

通过该实验复习了 C/C++ 的语法。同时可以使用 C/C++ 创建一个简单的 bash。了解并掌握 g++ 与 gcc 编译器对编写好的 .c 文件进行编译以及调试。理解了 shell 程序的功能。了解并掌握 makefile 文件的编写和使用。通过不断的学习我们可以实现更多的指令，这样可以在我们的日常使用中提供极大的便利。