
浙江大学实验报告

课程名称: Linux 程序设计 实验类型: 综合型

实验项目名称: 实验 3: 系统程序设计

学生姓名: 赵彬彬 专业: 计算机科学与技术 学号: 3140102395

电子邮件地址: 313220146@qq.com

实验日期: 2016 年 7 月 20 日

一、实验目的和要求

1. 学习如何使用 LINUX 的 C 语言工具完成代码编辑, 编译, 运行程序
2. 学习掌握 make 工具, Makefile 文件的 make 规则
3. 学习使用系统调用编写程序

二、主要仪器设备

计算机配置: 处理器 2.7GHz Intel Core i5, 内存 8GB 1867 MHz DDR3

操作系统环境: OS X EI Capitan 10.11.3 版本

LINUX 版本: Ubuntu 16.04 LTS & OS X EI Capitan 10.11.3

二、实验内容与结果

1. 在操作系统分析及实验课程中要对 linux 内核进行修改, 用 make 工具, 需要掌握 make 的规则。makfile 文件中的每一行是描述文件间依赖关系的 make 规则。本实验是关于 makefile 内容的, 你不需要在计算机上进行编程运行, 只要书面回答下面这些问题。

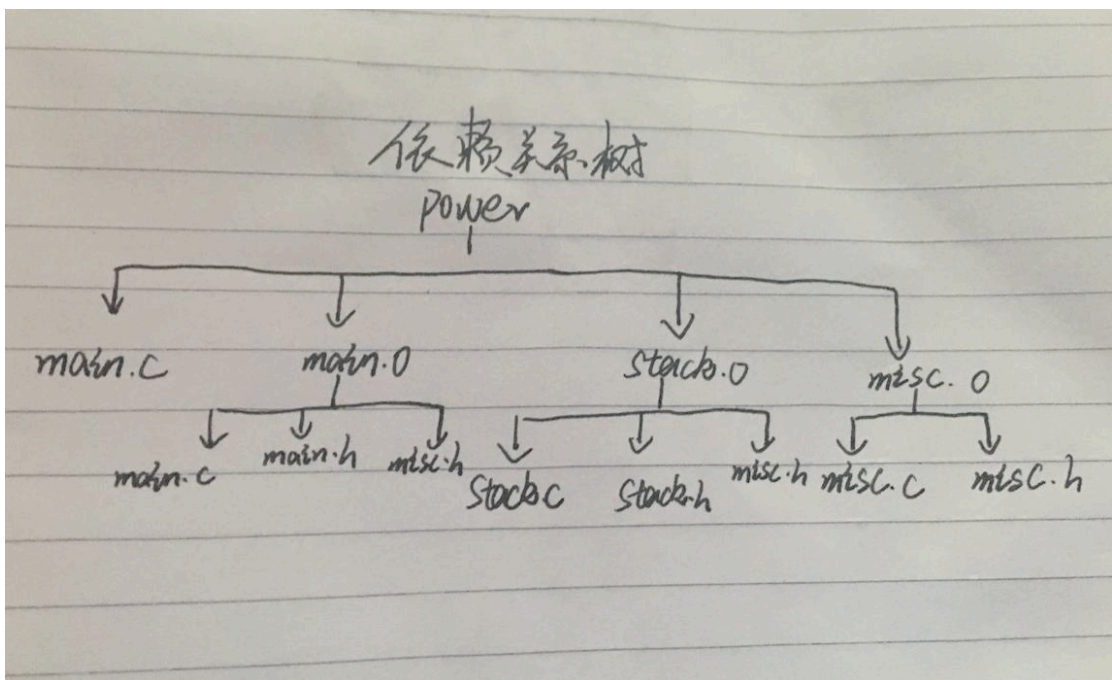
对于下面的 makefile:

```
CC = gcc
OPTIONS = -O3 -o
OBJECTS = main.o stack.o misc.o
SOURCES = main.c stack.c misc.c
HEADERS = main.h stack.h misc.h
power: main.c $(OBJECTS)
```

```
$(CC) $(OPTIONS) power $(OBJECTS) -lm
main.o: main.c main.h misc.h
stack.o: stack.c stack.h misc.h
misc.o: misc.c misc.h
```

回答下列问题

- 所有宏定义的名字
答: CC OPTIONS OBJECTS SOURCES HEADERS
- 所有目标文件的名字
答: main.o stack.o misc.o。
- 每个目标的依赖文件
答: main.o 的依赖文件: main.c main.h misc.h
stack.o 的依赖文件: stack.c stack.h misc.h
misc.o 的依赖文件: misc.c misc.h
- 生成每个目标文件所需执行的命令
答: gcc -c -o main.o main.c
gcc -c -o stack.o stack.c
gcc -c -o misc.o misc.c
- 画出 makefile 对应的依赖关系树。



- 生成 main.o stack.o 和 misc.o 时会执行哪些命令，为什么？
会执行预处理、编译、汇编命令，因为 gcc 的编译流程分为 4 个步骤，分别为预处理，编译，汇编和链接，而生成.o 文件时已经经过了预处理，编译和汇编三个阶段。
gcc -c -o main.o main.c
gcc -c -o stack.o stack.c
gcc -c -o misc.o misc.c

-
2. 用编辑器创建 main.c、compute.c、input.c、compute.h、input.h 和 main.h 文件。下面是它们的内容。注意 compute.h 和 input.h 文件仅包含了 compute 和 input 函数的声明但没有定义。定义部分是在 compute.c 和 input.c 文件中。main.c 包含的是两条显示给用户的提示信息。

\$ cat compute.h

```
/* compute 函数的声明原形 */  
double compute(double, double);
```

\$ cat input.h

```
/* input 函数的声明原形 */  
double input(char *);
```

\$ cat main.h

```
/* 声明用户提示 */  
#define PROMPT1 "请输入 x 的值: "  
#define PROMPT2 "请输入 y 的值: "
```

\$ cat compute.c

```
#include <math.h>  
#include <stdio.h>  
#include "compute.h"  
double compute(double x, double y)  
{  
    return (pow ((double)x, (double)y));  
}
```

\$ cat input.c

```
#include <stdio.h>  
#include "input.h"  
double input(char *s)  
{  
    float x;  
    printf("%s", s);  
    scanf("%f", &x);  
    return (x);  
}
```

\$ cat main.c

```
#include <stdio.h>  
#include "main.h"  
#include "compute.h"  
#include "input.h"  
  
main()  
{  
    double x, y;  
    printf("本程序从标准输入获取 x 和 y 的值并显示 x 的 y 次方.\n");  
    x = input(PROMPT1);  
    y = input(PROMPT2);
```

```
    printf("x 的 y 次方是:%6.3f\n",compute(x,y));  
}
```

提示：若您的 linux 系统没有中文系统，可以把程序中的汉字翻译成英文。

为了得到可执行文件 `power`，我们必须首先从三个源文件编译得到目标文件，并把它们连接在一起。下面的命令将完成这一任务。注意，在生成可执行代码时不要忘了连接上数学库。

```
$ gcc -c main.c input.c compute.c  
$ gcc main.o input.o compute.o -o power -lm  
$
```

相应的 Makefile 文件是：

```
$ cat Makefile  
power: main.o input.o compute.o  
    gcc main.o input.o compute.o -o power -lm  
  
main.o: main.c main.h input.h compute.h  
    gcc -c main.c  
  
input.o: input.c input.h  
    gcc -c input.c  
  
compute.o: compute.c compute.h  
    gcc -c compute.c  
$
```

- (1) 创建上述三个源文件和相应头文件，用 `gcc` 编译器，生成 `power` 可执行文件，并运行 `power` 程序。给出完成上述工作的步骤和程序运行结果。

```
bbge@ubuntu:~$ vim compute.h  
bbge@ubuntu:~$ vim input.h  
bbge@ubuntu:~$ vim main.h  
bbge@ubuntu:~$ vim compute.c  
bbge@ubuntu:~$ vim input.c  
bbge@ubuntu:~$ vim main.c
```

```
bbge@ubuntu:~$ gcc -c main.c input.c compute.c  
main.c:5:1: warning: return type defaults to 'int' [-Wimplicit-int]  
main()  
^  
bbge@ubuntu:~$ gcc main.o input.o compute.o -o power -lm
```

```
bbge@ubuntu:~$ ./power
display the value of x^y.
INPUT THE VALUE OF X 5
INPUT THE VALUE OF Y 6
x^y = :15625.000
```

- (2) 创建 Makefile 文件，使用 make 命令，生成 power 可执行文件，并运行 power 程序。
给出完成上述工作的步骤和程序运行结果。

```
bbge@ubuntu:~$ cat Makefile
power:main.o    input.o compute.o
    gcc      main.o  input.o compute.o    -o      power    -lm
main.o:main.c   main.h  input.h compute.h
    gcc      -c      main.c
input.o:input.c input.h
    gcc      -c      input.c
compute.o:compute.c compute.h
    gcc      -c      compute.c
```

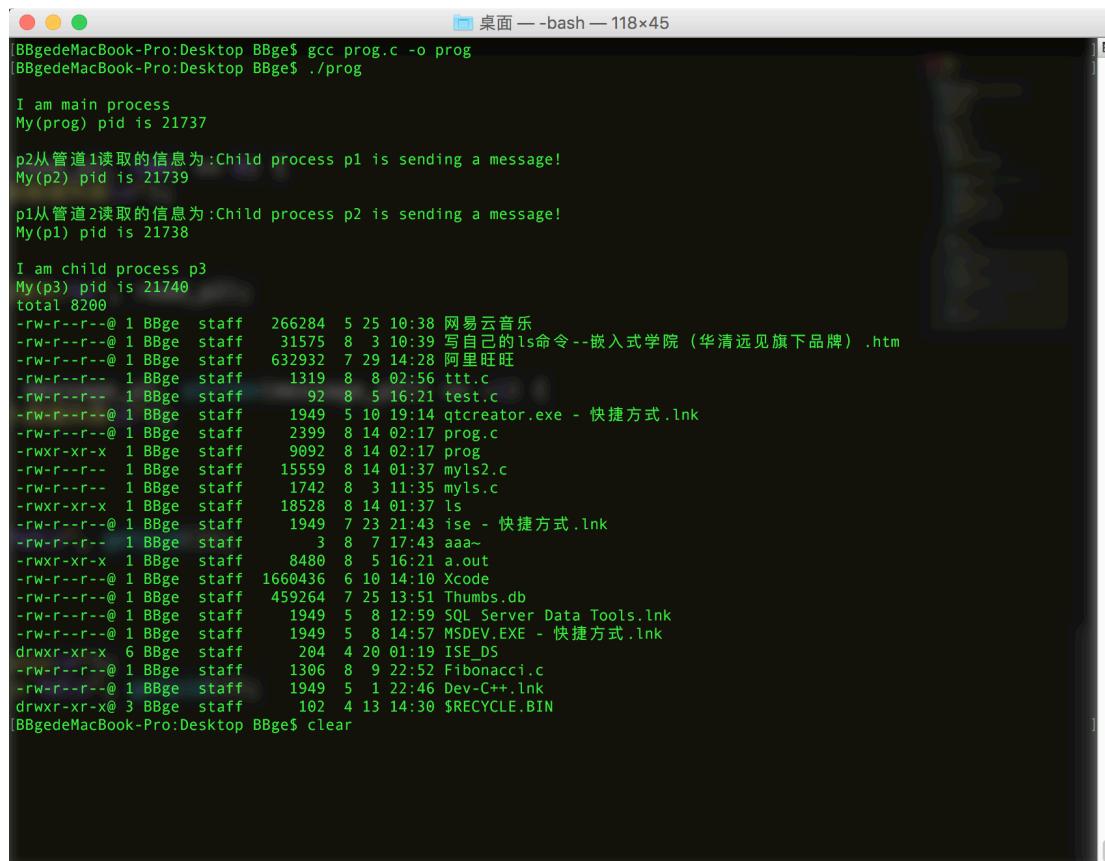
```
bbge@ubuntu:~$ make -f Makefile
gcc      main.o  input.o compute.o    -o      power    -lm
bbge@ubuntu:~$ ./power
display the value of x^y.
INPUT THE VALUE OF X 5
INPUT THE VALUE OF Y 6
x^y = :15625.000
```

3. 用 C 语言写一个程序名字为 **prog**，该程序运行过程中共有 4 个进程，**prog** 程序（父进程）创建 2 个子进程 p1 和 p2，p1 子进程再创建一个子进程 p3。4 个进程还需要完成如下工作：

- 1) 父进程打印字符串 “I am main process”；
- 2) 使用任何一种通信机制实现 p1 进程和 p2 进程之间的通信，可以使用的通信机制如：管道、消息队列、共享内存等。P1 进程发送 “Child process p1 is sending a message!” 信息给 p2 进程，P2 进程发送 “Child process p2 is sending a message!” 信息给 p1 进程，p1 和 p2 两个子进程接收信息后分别打印这两个字符串。
- 3) 子进程 p3 打印字符串 “I am child process p3”，然后使用 exec（族）系统调用打印当前目录下文件和子目录下的详细信息。
- 4) 每个进程的要打印自己的 pid。

(完成本题的有关资料请参考教材第 6、7 章)

实验结果如下图所示（代码在附件中）：



```
BBgedeMacBook-Pro:Desktop BBge$ gcc prog.c -o prog
BBgedeMacBook-Pro:Desktop BBge$ ./prog

I am main process
My(prog) pid is 21737

p2从管道1读取的信息为:Child process p1 is sending a message!
My(p2) pid is 21739

p1从管道2读取的信息为:Child process p2 is sending a message!
My(p1) pid is 21738

I am child process p3
My(p3) pid is 21740
total 8200
-rw-r--r--@ 1 BBge  staff   266284  5 25 10:38  网易云音乐
-rw-r--r--@ 1 BBge  staff   31575   8  3 10:39  写自己的ls命令--嵌入式学院（华清远见旗下品牌）.htm
-rw-r--r--@ 1 BBge  staff   632932  7 29 14:28  阿里旺旺
-rw-r--r--  1 BBge  staff    1319   8  8 02:56  ttt.c
-rw-r--r--  1 BBge  staff     92   8  5 16:21  test.c
-rw-r--r--@ 1 BBge  staff    1949  5 10 19:14  qtcreator.exe - 快捷方式.lnk
-rw-r--r--@ 1 BBge  staff    2399  8 14 02:17  prog.c
-rwxr-xr-x  1 BBge  staff    9092  8 14 02:17  prog
-rw-r--r--  1 BBge  staff   15559  8 14 01:37  myls2.c
-rw-r--r--  1 BBge  staff    1742   8  3 11:35  myls.c
-rwxr-xr-x  1 BBge  staff   18528  8 14 01:37  ls
-rw-r--r--@ 1 BBge  staff    1949  7 23 21:43  ise - 快捷方式.lnk
-rw-r--r--  1 BBge  staff     3   8  7 17:43  aaa~
-rwxr-xr-x  1 BBge  staff    8480   8  5 16:21  a.out
-rw-r--r--@ 1 BBge  staff  1660436  6 10 14:10  Xcode
-rw-r--r--@ 1 BBge  staff   459264  7 25 13:51  Thumbs.db
-rw-r--r--@ 1 BBge  staff    1949   5  8 12:59  SQL Server Data Tools.lnk
-rw-r--r--@ 1 BBge  staff    1949   5  8 14:57  MSDEV.EXE - 快捷方式.lnk
drwxr-xr-x  6 BBge  staff    204  4 20 01:19  ISE_DS
-rw-r--r--@ 1 BBge  staff    1306   8  9 22:52  Fibonacci.c
-rw-r--r--@ 1 BBge  staff    1949   5  1 22:46  Dev-C++.lnk
-rwxr-xr-x@ 3 BBge  staff    102  4 13 14:30  $RECYCLE.BIN
BBgedeMacBook-Pro:Desktop BBge$ clear
```

4. Fibonacci 序列为 0,1,1,2,3,5,8,..., 通常, 可表达为:

$$\text{fib}_0=0$$

$$\text{fib}_1=1$$

$$\text{fib}_n=\text{fib}_{n-1}+\text{fib}_{n-2}$$

编写一个多线程程序来生成 Fibonacci 序列。程序应该这样工作: 用户运行程序时在命令行输入要产生 Fibonacci 序列的个数, 然后程序创建一个新的线程来产生 Fibonacci 序列, 把这个序列放到线程共享的数据中(数组可能是一种最方便的数据结构)。当线程执行完成后, 父线程将输出由子线程产生的序列。由于在子线程结束前, 父线程不能开始输出 Fibonacci 序列, 因此父线程必须等待子线程的结束。[\(完成本题的有关知识请参考教材第8章\)](#)

```
桌面 — -bash — 80x48

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <pthread.h>
#include <unistd.h>

int number; /*全局变量number表示Fibonacci序列个数*/
void *calculate(void *a); /*Fibonacci的计算*/

int main(void)
{
    int i, temp, *p;
    pthread_t tid;

    printf("请输入Fibonacci序列的个数:");
    scanf("%d", &number);

    /*分配一块动态内存*/
    p = (int *)malloc(sizeof(int) * number);

    /*创建线程*/
    temp = pthread_create(&tid, NULL, calculate, p);
    /*无法创建线程时的错误提示*/
    if (temp != 0)
        fprintf(stderr, "无法创建线程: %s\n", strerror(temp));

    /*等待线程结束*/
    temp = pthread_join(tid, NULL);

    /*线程无法结束时的错误提示*/
    if (temp != 0)
        fprintf(stderr, "线程无法结束: %s\n", strerror(temp));

    printf("正在等待子进程结束...\n");

    /*子进程结束后, 父进程会输出Fibonacci序列*/
    for (i = 0; i < number; i++)
        printf("%d ", p[i]);

    free(p);
    printf("\n");

    return 0;
}

void *calculate(void *a)
```



```

void *calculate(void *a)
{
    int i;

    /*计算Fibonacci序列并赋给数组a*/
    if (number == 1) {
        *((int *)a) = 0;
    } else {
        *((int *)a) = 0;
        *((int *)a + 1) = 1;
        for (i = 2; i < number; i++)
            *((int *)a + i) = *((int *)a + i - 1) + *((int *)a + i -
2);
    }
}

```

实验结果如下所示：

```

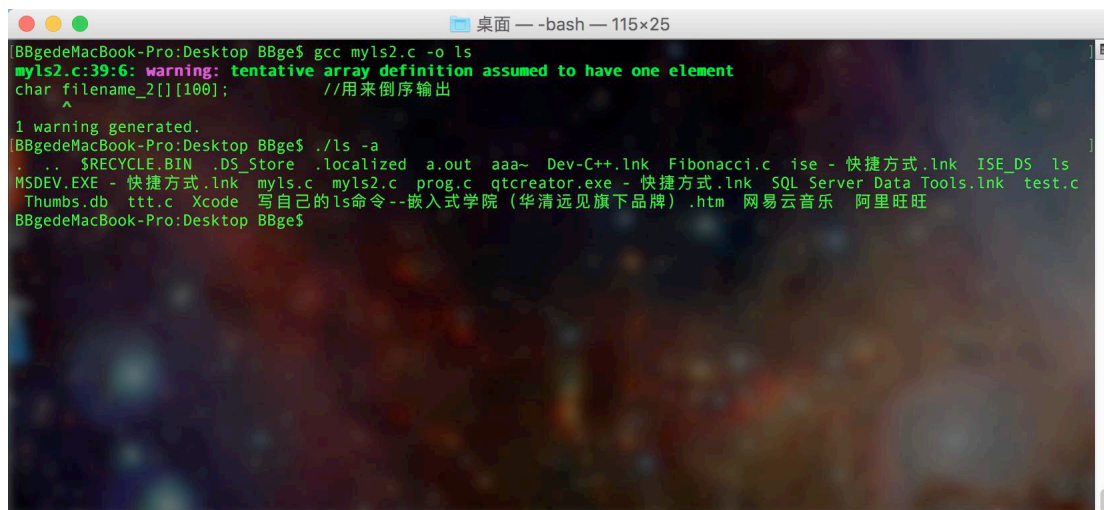
Last login: Sun Aug 14 02:35:08 on ttys001
[BBgedeMacBook-Pro:~ BBge$ cd Desktop/
[BBgedeMacBook-Pro:Desktop BBge$ gcc Fibonacci.c -o Fib
Fibonacci.c:61:1: warning: control reaches end of non-void function
[-Wreturn-type]
}
^
1 warning generated.
[BBgedeMacBook-Pro:Desktop BBge$ ./Fib
请输入Fibonacci序列的个数:25
正在等待子进程结束...
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181
6765 10946 17711 28657 46368
[BBgedeMacBook-Pro:Desktop BBge$ |

```


5.下面两题任选一题

- a) 用 C 语言写一个名字为 myls 程序，实现类似 Linux 的 ls 命令，其中 myls 命令至少实现 -a、-l、-i、-d、-t、-F、-R 等 20 个选项的功能。要求 myls 程序使用系统调用函数编写，不能使用 exec 系统调用或 system() 函数等调用 ls 命令来实现。命令 man ls 可以得到更多 ls 选项的含义。(完成本题的有关知识请参考教材第 5 章及“UNIX 环境高级编程”)

(1) ls -a 命令：显示隐藏文件



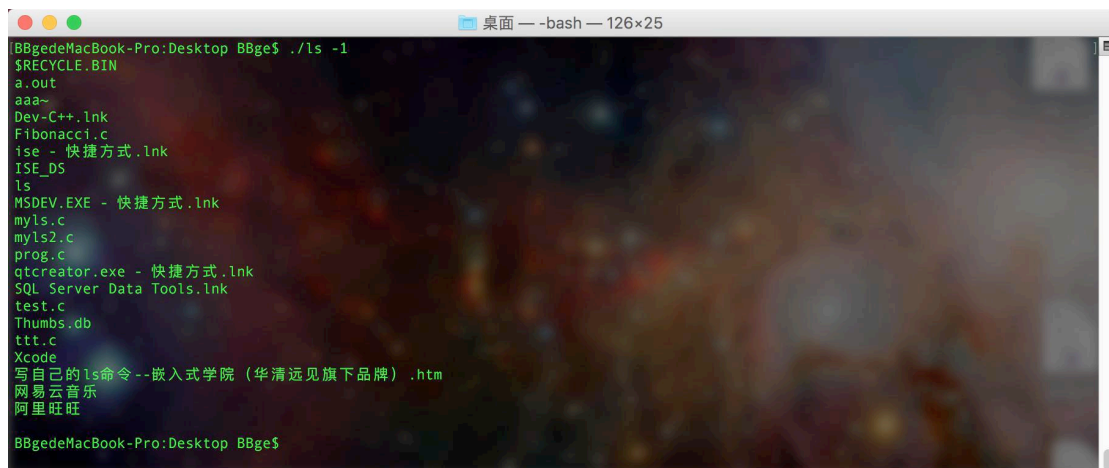
```
BBgedeMacBook-Pro:Desktop BBge$ gcc myls2.c -o ls
mysls2.c:39:6: warning: tentative array definition assumed to have one element
char filename_2[][100];           //用来倒序输出
      ^
1 warning generated.
BBgedeMacBook-Pro:Desktop BBge$ ./ls -a
.  .. $RECYCLE.BIN .DS_Store .localized a.out aaa~ Dev-C++.lnk Fibonacci.c ise - 快捷方式.lnk ISE_DS ls
MSDEV.EXE - 快捷方式.lnk myls.c myls2.c prog.c qtcreeator.exe - 快捷方式.lnk SQL Server Data Tools.lnk test.c
Thumbs.db ttt.c Xcode 写自己的ls命令--嵌入式学院（华清远见旗下品牌）.htm 网易云音乐 阿里旺旺
BBgedeMacBook-Pro:Desktop BBge$
```

(2) ls -l 命令：列出文件的详细信息




```
BBgedeMacBook-Pro:Desktop BBge$ ./ls -l
drwxr-xr-x  3 BBge  staff   102 2016  4 13 14:30 $RECYCLE.BIN
-rwxr-xr-x  1 BBge  staff   8480 2016  8  5 16:21 a.out
-rw-r--r--  1 BBge  staff    3 2016  8  7 17:43 aaa~
-rw-r--r--  1 BBge  staff  1949 2016  5  1 22:46 Dev-C++.lnk
-rw-r--r--  1 BBge  staff  1306 2016  8  9 22:52 Fibonacci.c
-rw-r--r--  1 BBge  staff  1949 2016  7 23 21:43 ise - 快捷方式.lnk
drwxr-xr-x  6 BBge  staff   204 2016  4 20  1:19 ISE_DS
-rwxr-xr-x  1 BBge  staff  18528 2016  8 13 20:36 ls
-rw-r--r--  1 BBge  staff  1949 2016  5  8 14:57 MSDEV.EXE - 快捷方式.lnk
-rw-r--r--  1 BBge  staff  1742 2016  8  3 11:35 myls.c
-rw-r--r--  1 BBge  staff  15381 2016  8 13 20:37 myls2.c
-rw-r--r--  1 BBge  staff  2416 2016  7 31 22:5 prog.c
-rw-r--r--  1 BBge  staff  1949 2016  5 10 19:14 qtcreeator.exe - 快捷方式.lnk
-rw-r--r--  1 BBge  staff  1949 2016  5  8 12:59 SQL Server Data Tools.lnk
-rw-r--r--  1 BBge  staff    92 2016  8  5 16:21 test.c
-rw-r--r--  1 BBge  staff  459264 2016  7 25 13:51 Thumbs.db
-rw-r--r--  1 BBge  staff   1319 2016  8  2 56 ttt.c
-rw-r--r--  1 BBge  staff  1660436 2016  6 10 14:10 Xcode
-rw-r--r--  1 BBge  staff   31575 2016  8  3 10:39 写自己的ls命令--嵌入式学院（华清远见旗下品牌）.htm
-rw-r--r--  1 BBge  staff   266284 2016  5 25 10:38 网易云音乐
-rw-r--r--  1 BBge  staff   632932 2016  7 29 14:28 阿里旺旺
BBgedeMacBook-Pro:Desktop BBge$
```

(3) ls -l 命令：一行只输出一个文件



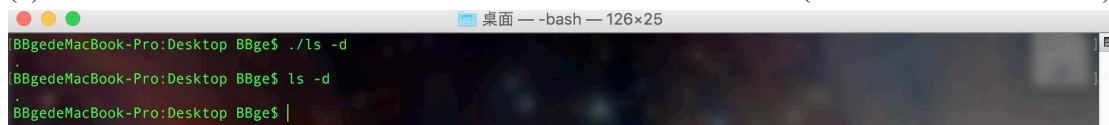
```
BBgedeMacBook-Pro:Desktop BBge$ ./ls -l
$RECYCLE.BIN
a.out
aaa~
Dev-C++.lnk
Fibonacci1.c
ise - 快捷方式.lnk
ISE_DS
ls
MSDEV.EXE - 快捷方式.lnk
mysls.c
mysls2.c
prog.c
qcreator.exe - 快捷方式.lnk
SQL Server Data Tools.lnk
test.c
Thumbs.db
ttt.c
Xcode
写自己的ls命令--嵌入式学院（华清远见旗下品牌）.htm
网易云音乐
阿里旺旺
BBgedeMacBook-Pro:Desktop BBge$
```

(4) ls -i 命令：输出文件的 i 节点的索引信息



```
BBgedeMacBook-Pro:Desktop BBge$ ./ls -i
$RECYCLE.BIN
a.out
aaa~
Dev-C++.lnk
Fibonacci1.c
ise - 快捷方式.lnk
ISE_DS
ls
MSDEV.EXE - 快捷方式.lnk
mysls.c
mysls2.c
prog.c
qcreator.exe - 快捷方式.lnk
SQL Server Data Tools.lnk
test.c
Thumbs.db
ttt.c
Xcode
写自己的ls命令--嵌入式学院（华清远见旗下品牌）.htm
网易云音乐
阿里旺旺
BBgedeMacBook-Pro:Desktop BBge$
```

(5) ls -d 命令：将目录像文件一样显示，而不是显示其下的文件(和自带的 ls -d 作了比较)



```
BBgedeMacBook-Pro:Desktop BBge$ ./ls -d
$RECYCLE.BIN
a.out
aaa~
Dev-C++.lnk
Fibonacci1.c
ise - 快捷方式.lnk
ISE_DS
ls
MSDEV.EXE - 快捷方式.lnk
mysls.c
mysls2.c
prog.c
qcreator.exe - 快捷方式.lnk
SQL Server Data Tools.lnk
test.c
Thumbs.db
ttt.c
Xcode
写自己的ls命令--嵌入式学院（华清远见旗下品牌）.htm
网易云音乐
阿里旺旺
BBgedeMacBook-Pro:Desktop BBge$
```

(6) ls -t 命令：将文件以时间排序



```
BBgedeMacBook-Pro:Desktop BBge$ ./ls -t
$RECYCLE.BIN
a.out
aaa~
Dev-C++.lnk
Fibonacci1.c
ise - 快捷方式.lnk
ISE_DS
ls
MSDEV.EXE - 快捷方式.lnk
mysls.c
mysls2.c
prog.c
qcreator.exe - 快捷方式.lnk
SQL Server Data Tools.lnk
test.c
Thumbs.db
ttt.c
Xcode
写自己的ls命令--嵌入式学院（华清远见旗下品牌）.htm
网易云音乐
阿里旺旺
BBgedeMacBook-Pro:Desktop BBge$
```

(7) `ls -n` 命令：用数字的 UID，GID 代替名称

```
BBgedeMacBook-Pro:Desktop BBge$ ./ls -n
drwxr-xr-x  3  501    20      102 2016  4  13 14:30 $RECYCLE.BIN
-rwxr-xr-x  1  501    20      8480 2016  8  5 16:21 a.out
-rw-r--r--  1  501    20         3 2016  8  7 17:43 aaa~
-rw-r--r--  1  501    20     1949 2016  5  1 22:46 Dev-C++.lnk
-rw-r--r--  1  501    20     1306 2016  8  9 22:52 Fibonacci.c
-rw-r--r--  1  501    20     1949 2016  7 23 21:43 ise - 快捷方式.lnk
drwxr-xr-x  6  501    20      204 2016  4 20  1:19 ISE_DS
-rwxr-xr-x  1  501    20    18528 2016  8 13 20:40 ls
-rw-r--r--  1  501    20     1949 2016  5  8 14:57 MSDEV.EXE - 快捷方式.lnk
-rw-r--r--  1  501    20     1742 2016  8  3 11:35 myls.c
-rw-r--r--  1  501    20    15463 2016  8 13 20:40 myls2.c
-rw-r--r--  1  501    20     2416 2016  7 31 22:5 prog.c
-rw-r--r--  1  501    20     1949 2016  5 10 19:14 qtcreator.exe - 快捷方式.lnk
-rw-r--r--  1  501    20     1949 2016  5  8 12:59 SQL Server Data Tools.lnk
-rw-r--r--  1  501    20       92 2016  8  5 16:21 test.c
-rw-r--r--  1  501    20   459264 2016  7 25 13:51 Thumbs.db
-rw-r--r--  1  501    20     1319 2016  8  8  2:56 ttt.c
-rw-r--r--  1  501    20   1660436 2016  6 10 14:10 Xcode
-rw-r--r--  1  501    20     31575 2016  8  3 10:39 写自己的ls命令--嵌入式学院（华清远见旗下品牌）.htm
-rw-r--r--  1  501    20    266284 2016  5 25 10:38 网易云音乐
-rw-r--r--  1  501    20    632932 2016  7 29 14:28 阿里旺旺
total = 3109891
BBgedeMacBook-Pro:Desktop BBge$
```

(8) `ls -F` 命令：显示不同的符号来区别文件

```
BBgedeMacBook-Pro:Desktop BBge$ ./ls -F
$RECYCLE.BIN/
a.out*
aaa~
Dev-C++.lnk
Fibonacci.c
ise - 快捷方式.lnk
ISE_DS/
ls*
MSDEV.EXE - 快捷方式.lnk
mysls.c
mysls2.c
prog.c
qtcreator.exe - 快捷方式.lnk
SQL Server Data Tools.lnk
test.c
Thumbs.db
ttt.c
Xcode
写自己的ls命令--嵌入式学院（华清远见旗下品牌）.htm
网易云音乐
阿里旺旺
```

(9) `ls -s` 命令：在每个文件名前输出该文件的大小

```
BBgedeMacBook-Pro:Desktop BBge$ ./ls -s
 0 $RECYCLE.BIN
24 a.out
 8 aaa~
592 Dev-C++.lnk
 8 Fibonacci.c
336 ise - 快捷方式.lnk
 0 ISE_DS
40 ls
384 MSDEV.EXE - 快捷方式.lnk
 8 myls.c
32 myls2.c
 8 prog.c
304 qtcreator.exe - 快捷方式.lnk
432 SQL Server Data Tools.lnk
 8 test.c
904 Thumbs.db
 8 ttt.c
3248 Xcode
 64 写自己的ls命令--嵌入式学院（华清远见旗下品牌）.htm
528 网易云音乐
1240 阿里旺旺
total = 8176
BBgedeMacBook-Pro:Desktop BBge$
```


(10) `ls -m` 命令：横向输出文件名，并以","作分隔符

```
BBgedeMacBook-Pro:Desktop BBge$ ./ls -m
$RECYCLE.BIN, a.out, aaa~, Dev-C++.lnk, Fibonacci.c, ise - 快捷方式.lnk, ISE_DS, ls, MSDEV.EXE - 快捷方式.lnk,
mysl.c, mysl2.c, prog.c, qtcreator.exe - 快捷方式.lnk, SQL Server Data Tools.lnk, test.c, Thumbs.db, ttt.c,
Xcode, 写自己的ls命令--嵌入式学院（华清远见旗下品牌）.htm, 网易云音乐, 阿里旺旺,
BBgedeMacBook-Pro:Desktop BBge$ |
```

(11) `ls -o` 命令：显示文件的除组信息外的详细信息

```
BBgedeMacBook-Pro:Desktop BBge$ ./ls -o
drwxr-xr-x  3 BBge      102 2016  4 13 14:30 $RECYCLE.BIN
-rwxr-xr-x  1 BBge     8480 2016  8  5 16:21 a.out
-rw-r--r--  1 BBge       3 2016  8  7 17:43 aaa~
-rw-r--r--  1 BBge    1949 2016  5  1 22:46 Dev-C++.lnk
-rw-r--r--  1 BBge    1306 2016  8  9 22:52 Fibonacci.c
-rw-r--r--  1 BBge    1949 2016  7 23 21:43 ise - 快捷方式.lnk
drwxr-xr-x  6 BBge      204 2016  4 20  1:19 ISE_DS
-rwxr-xr-x  1 BBge    18528 2016  8 13 20:40 ls
-rw-r--r--  1 BBge    1949 2016  5  8 14:57 MSDEV.EXE - 快捷方式.lnk
-rw-r--r--  1 BBge    1742 2016  8  3 11:35 mysl.c
-rw-r--r--  1 BBge   15463 2016  8 13 20:40 mysl2.c
-rw-r--r--  1 BBge    2416 2016  7 31 22:5 prog.c
-rw-r--r--  1 BBge    1949 2016  5 10 19:14 qtcreator.exe - 快捷方式.lnk
-rw-r--r--  1 BBge    1949 2016  5  8 12:59 SQL Server Data Tools.lnk
-rw-r--r--  1 BBge     92 2016  8  5 16:21 test.c
-rw-r--r--  1 BBge   459264 2016  7 25 13:51 Thumbs.db
-rw-r--r--  1 BBge    1319 2016  8  8  2:56 ttt.c
-rw-r--r--  1 BBge  1660436 2016  6 10 14:10 Xcode
-rw-r--r--  1 BBge   31575 2016  8  3 10:39 写自己的ls命令--嵌入式学院（华清远见旗下品牌）.htm
-rw-r--r--  1 BBge   266284 2016  5 25 10:38 网易云音乐
-rw-r--r--  1 BBge   632932 2016  7 29 14:28 阿里旺旺
BBgedeMacBook-Pro:Desktop BBge$
```

(12) `ls -r` 对目录反向排序

```
BBgedeMacBook-Pro:Desktop BBge$ ./ls -r
阿里旺旺  网易云音乐  写自己的ls命令--嵌入式学院（华清远见旗下品牌）.htm  Xcode  ttt.c  Thumbs.db  test.c
SQL Server Data Tools.lnk  qtcreator.exe - 快捷方式.lnk  prog.c  mysl2.c  mysl.c  MSDEV.EXE - 快捷方式.lnk
ls  ISE_DS  ise - 快捷方式.lnk  Fibonacci.c  Dev-C++.lnk  aaa~  a.out  $RECYCLE.BIN
BBgedeMacBook-Pro:Desktop BBge$ |
```

(13) `ls -Q` 命令：把输出的文件名用双引号括起来

```
BBgedeMacBook-Pro:Desktop BBge$ ./ls -Q
"$RECYCLE.BIN" "a.out" "aaa~" "Dev-C++.lnk" "Fibonacci.c" "ise - 快捷方式.lnk" "ISE_DS" "ls" "MSDEV.EXE - 快捷方式.lnk"
"mysl.c" "mysl2.c" "prog.c" "qtcreator.exe - 快捷方式.lnk" "SQL Server Data Tools.lnk" "test.c" "Thumbs.db" "ttt.c" "Xc
ode" "写自己的ls命令--嵌入式学院（华清远见旗下品牌）.htm" "网易云音乐" "阿里旺旺"
BBgedeMacBook-Pro:Desktop BBge$ |
```

(14) `ls -g` 命令：列出文件的详细信息，但不列出文件拥有者

```
BBgedeMacBook-Pro:Desktop BBge$ ./ls -g
drwxr-xr-x  3 BBge      102 2016  4 13 14:30 $RECYCLE.BIN
-rwxr-xr-x  1 BBge     8480 2016  8  5 16:21 a.out
-rw-r--r--  1 BBge       3 2016  8  7 17:43 aaa~
-rw-r--r--  1 BBge    1949 2016  5  1 22:46 Dev-C++.lnk
-rw-r--r--  1 BBge    1306 2016  8  9 22:52 Fibonacci.c
-rw-r--r--  1 BBge    1949 2016  7 23 21:43 ise - 快捷方式.lnk
drwxr-xr-x  6 BBge      204 2016  4 20  1:19 ISE_DS
-rwxr-xr-x  1 BBge    18528 2016  8 14  1:22 ls
-rw-r--r--  1 BBge    1949 2016  5  8 14:57 MSDEV.EXE - 快捷方式.lnk
-rw-r--r--  1 BBge    1742 2016  8  3 11:35 mysl.c
-rw-r--r--  1 BBge   15558 2016  8 14  1:22 mysl2.c
-rw-r--r--  1 BBge    2416 2016  7 31 22:5 prog.c
-rw-r--r--  1 BBge    1949 2016  5 10 19:14 qtcreator.exe - 快捷方式.lnk
-rw-r--r--  1 BBge    1949 2016  5  8 12:59 SQL Server Data Tools.lnk
-rw-r--r--  1 BBge     92 2016  8  5 16:21 test.c
-rw-r--r--  1 BBge   459264 2016  7 25 13:51 Thumbs.db
-rw-r--r--  1 BBge    1319 2016  8  8  2:56 ttt.c
-rw-r--r--  1 BBge  1660436 2016  6 10 14:10 Xcode
-rw-r--r--  1 BBge   31575 2016  8  3 10:39 写自己的ls命令--嵌入式学院（华清远见旗下品牌）.htm
-rw-r--r--  1 BBge   266284 2016  5 25 10:38 网易云音乐
-rw-r--r--  1 BBge   632932 2016  7 29 14:28 阿里旺旺
BBgedeMacBook-Pro:Desktop BBge$ |
```

(15) ls -B 命令：不输出以~结尾的备份文件

```
BBgedeMacBook-Pro:Desktop BBge$ ./ls -B
$RECYCLE.BIN  a.out  Dev-C++.lnk  Fibonacci.c  ise - 快捷方式.lnk ISE_DS  ls  MSDEV.EXE - 快捷方式
.lnk myls.c      myls2.c  prog.c      qtcreeator.exe - 快捷方式.lnk SQL Server Data Tools.lnk test.c  Thumbs.db
ttd.c  Xcode      写自己的ls命令--嵌入式学院 (华清远见旗下品牌) .htm 网易云音乐 阿里旺旺
BBgedeMacBook-Pro:Desktop BBge$
```

(16) ls -N 命令：输出不限文件长度

```
BBgedeMacBook-Pro:Desktop BBge$ ./ls -N
$RECYCLE.BIN  a.out  aaa~  Dev-C++.lnk  Fibonacci.c  ise - 快捷方式.lnk  ISE_DS  ls  MSDEV.EXE - 快捷方式
.lnk myls.c      myls2.c  prog.c      qtcreeator.exe - 快捷方式.lnk  SQL Server Data Tools.lnk  test.c  Thumbs.db
ttd.c  Xcode      写自己的ls命令--嵌入式学院 (华清远见旗下品牌) .htm  网易云音乐  阿里旺旺
BBgedeMacBook-Pro:Desktop BBge$
```

(17) ls -S 命令：以文件大小排序 (从大到小)

```
BBgedeMacBook-Pro:Desktop BBge$ ./ls -S
Xcode 阿里旺旺 Thumbs.db 网易云音乐 写自己的ls命令--嵌入式学院 (华清远见旗下品牌) .htm  ls  myls2.c
a.out  prog.c  MSDEV.EXE - 快捷方式.lnk  ise - 快捷方式.lnk  Dev-C++.lnk  qtcreeator.exe - 快捷方式.lnk
SQL Server Data Tools.lnk  myls.c  ttd.c  Fibonacci.c  ISE_DS  $RECYCLE.BIN  test.c  aaa~
BBgedeMacBook-Pro:Desktop BBge$
```

(18) ls -A 命令：显示除 . 和 .. 外的所有文件

```
BBgedeMacBook-Pro:Desktop BBge$ ./ls -A
$RECYCLE.BIN  a.out  aaa~  Dev-C++.lnk  Fibonacci.c  ise - 快捷方式.lnk  ISE_DS  ls  MSDEV.EXE - 快捷方式
.lnk myls.c      myls2.c  prog.c      qtcreeator.exe - 快捷方式.lnk  SQL Server Data Tools.lnk  test.c  Thumbs.db
ttd.c  Xcode      写自己的ls命令--嵌入式学院 (华清远见旗下品牌) .htm  网易云音乐  阿里旺旺
BBgedeMacBook-Pro:Desktop BBge$
```

(19) ls -u 命令：以文件上一次的访问时间排序

```
BBgedeMacBook-Pro:Desktop BBge$ ./ls -u
ls  myls2.c  Xcode  写自己的ls命令--嵌入式学院 (华清远见旗下品牌) .htm  网易云音乐  阿里旺旺  MSDEV.EXE -
快捷方式.lnk  ise - 快捷方式.lnk  qtcreeator.exe - 快捷方式.lnk  SQL Server Data Tools.lnk  prog.c  ttd.c
Fibonacci.c  ISE_DS  $RECYCLE.BIN  Thumbs.db  Dev-C++.lnk  aaa~  a.out  test.c  myls.c
BBgedeMacBook-Pro:Desktop BBge$
```

(20) ls -R 命令：列出所有子目录下的文件

```
Dev-C++.lnk
Fibonacci.c
ISE_DS
MSDEV.EXE - 快捷方式.lnk
SQL Server Data Tools.lnk
Thumbs.db
Xcode
a.out
aaa~
ise - 快捷方式.lnk
ls
mysls.c
mysls2.c
prog.c
qtcreeator.exe - 快捷方式.lnk
test.c
ttd.c
阿里旺旺
写自己的ls命令--嵌入式学院 (华清远见旗下品牌) .htm
网易云音乐

./$RECYCLE.BIN:

./ISE_DS:
EDK          ISE          PlanAhead   SysGen

./ISE_DS/EDK:
eclipse      gnuwin      lib          sw
gnu          hw          msg          third_party

./ISE_DS/EDK/eclipse:
nt

./ISE_DS/EDK/eclipse/nt:
eclipse

./ISE_DS/EDK/eclipse/nt/eclipse:
plugins readme

./ISE_DS/EDK/eclipse/nt/eclipse/plugins:
com.xilinx.sdk.docs.cheatsheets_1.0.0
com.xilinx.sdk.docs.user_1.0.0
com.xilinx.sdk.product_1.0.0
gnu.io.rxtx.win32.x86_2.1.7.3_v20071015
org.apache.ant_1.8.3.v201301120609
```

四、讨论、心得

实验 3 主要是 Linux 的系统编程，学到了很多自己之前没接触到过的知识，比如通信和进程这两个概念。在以前 C 语言的学习中，`argc` 和 `argv` 两个参数一直没用过，不知道这两个参数是什么意思。但在写 `myls` 的时候，自己知道了 `argc` 和 `argv` 这两个命令行参数的用法，对 C 语言的理解又更深了一层。

本次实验个人感觉比第二次实验要简单一些，shell 编程需要掌握一些没学过的语法，相当于是一门全新的语言，而此次实验用到的 C 语言对我们而言已经是熟练掌握了。同时，这次我都是用 `gcc` 来编译写好的源程序文件。完成本次实验，我基本上都是在空白文本中写好自己的代码，然后在命令行输入 `gcc` 来编译，没有用到 IDE，我感觉这样非常方便，`gcc` 给的错误提示也非常明确。

Linux 的实验到此也结束了，在完成这几个实验的过程中，自己学到了很多知识，但越学越觉得 Linux 的博大精深，感觉自己也只是学了个皮毛。希望自己在操作系统课中能够更深入的了解 Linux。