

# 上海交通大学

SHANGHAI JIAO TONG UNIVERSITY

## 实验报告

LAB REPORT



## 操作系统

### 实验一 Shell 程序设计

姓 名: 袁炜程

学 号: 516030910287

班 级: F1603602

## 一、实验要求

1. 编写一个 Shell 程序 findit, 该程序搜索参数 1 指定的目录树, 查找所有的以.c 和.h 结尾的文件, 如文件行中含有参数指定的字符串, 显示该行和相应的文件名。如目录参数 1 缺省, 则从当前目录中搜索。如: findit /home/wang/work searchstring  
搜索以/home/wang/work 为根的目录树中的 c 程序和头文件, 查找含有 searchstring 字符串的行, 显示文件名。

```
findit searchstring
```

从当前目录开始搜索。

2. 编一 Shell 程序, 以类似书本的目录结构的形式, 按层次输出当前目录树中的所有目录和文件, 要求每一层缩进 4 个空格。

## 二、实验原理及代码

### 1、findit

原本的思路是这样: 首先遍历目录树, 然后检查每一个文件是否为.h 或.c 文件, 如果是的话就进行字符串匹配, 如果有的话先输出该文件名, 然后输出字符串所在的行。这个方法在具体实现上来说, 先用一个递归函数进行遍历目录树, 然后用 case 语句检查文件类型是否符合, 符合的话用 grep 命令输出文件名和字符串所在行。因此代码如下:

```
function read_dir(){
    for file in `ls $1`          #获取当前目录的每一项
    do
        if [ -d $1/"$file" ]      #test -d 判断, 如果目录存在
        then
            read_dir $1/"$file" $2    #递归调用, 参数变为子目录
        else
            case $file in          #目录不存在, 即为文件
                *.c | *.h )
                    grep -l "test" $1/"$file" && grep -A 0 $2 $1/"$file" --
color #字符串匹配
                    ;;
            esac
        fi
    done
}

if [ $# -eq 2 ]                 #主函数, 判断有几个参数
```

```

then
    read_dir $1 $2 #2 个，函数参数为目录名和字符串
else
    read_dir . $1 #1 个，函数参数为当前目录名和字符串
fi

```

其中用到了 grep 命令，grep 的语法：

```

grep [-abcEFGhHilLnqrsVwxy][[-A<显示列数>][[-B<显示列数>][[-C<显示列数>][[-d<进行动作>][[-e<范本样式>][[-f<范本文件>][--help][范本样式][文件或目录...]

```

本实验中用到的参数是 -l：列出文件内容符合指定的样式的文件名称；-A：除了显示符合范本样式的那一列之外，并显示该行之后的内容。

但是感觉这有点繁琐。在继续查询命令后发现还有一个 find 命令，它可以代替递归函数在目录及其子目录中搜索目标文件。find 语法为：find path -option [-print] [-exec -ok command] {} \; 本实验中只要求获得.h 和.c 文件并打印文件名，用 -o 即“或”连接，；然后管道符连接 grep，用 xargs 将前一次 find 的输出变为后一次 grep 的输入，以达到同时输出文件名和字符串所在行的目的，--color 是为了突出显示匹配的字符串。这种方法的代码很简单，如下：

```

if [ -d $1 ]
then
    find $1 -name "*.c" -o -name "*.h" | xargs grep "$2" -A 0 --color
else
    find . -name "*.c" -o -name "*.h" | xargs grep "$1" -A 0 --color
fi

```

实验结果：(findit1 命令对应用 find 和 grep 的方法，findit2 命令对应递归和 grep 方法)

```

qwc@qwc-HP-Pavilion-Laptop-14-ce1xxx:~/shell-procedure$ sh findit test
./qwc/1.h:This is a test.
./qwc/1.h:This is a test1.
./qwc/1.h:This is a test2.
--
./qwc/2.h:This is a test.
./qwc/2.h:This is a test1.
./qwc/2.h:This is a test2.
--
./111.c:This is a test.
./111.c:This is a test1.
./111.c:This is a test2.
qwc@qwc-HP-Pavilion-Laptop-14-ce1xxx:~/shell-procedure$ sh findit /home/qwc/shell-procedure test
/home/qwc/shell-procedure/qwc/1.h:This is a test.
/home/qwc/shell-procedure/qwc/1.h:This is a test1.
/home/qwc/shell-procedure/qwc/1.h:This is a test2.
--
/home/qwc/shell-procedure/qwc/2.h:This is a test.
/home/qwc/shell-procedure/qwc/2.h:This is a test1.
/home/qwc/shell-procedure/qwc/2.h:This is a test2.
--
/home/qwc/shell-procedure/111.c:This is a test.
/home/qwc/shell-procedure/111.c:This is a test1.
/home/qwc/shell-procedure/111.c:This is a test2.

```

```

qwc@qwc-HP-Pavilion-Laptop-14-ce1xxx:~/shell-procedure$ sh findit2 test
./111.c
This is a test.
This is a test1.
This is a test2.
./qwc/1.h
This is a test.
This is a test1.
This is a test2.
./qwc/2.h
This is a test.
This is a test1.
This is a test2.
qwc@qwc-HP-Pavilion-Laptop-14-ce1xxx:~/shell-procedure$ sh findit2 /home/qwc/shell-procedure test
/home/qwc/shell-procedure/111.c
This is a test.
This is a test1.
This is a test2.
/home/qwc/shell-procedure/qwc/1.h
This is a test.
This is a test1.
This is a test2.
/home/qwc/shell-procedure/qwc/2.h
This is a test.
This is a test1.
This is a test2.

```

## 2、catalog 打印目录

总体思路比较简单，跟上一个实验一样先是遍历目录及其子目录的所有文件，然后在每一次调用递归函数的时候缩进 4 个空格，以达到目录的视觉效果，代码：

```

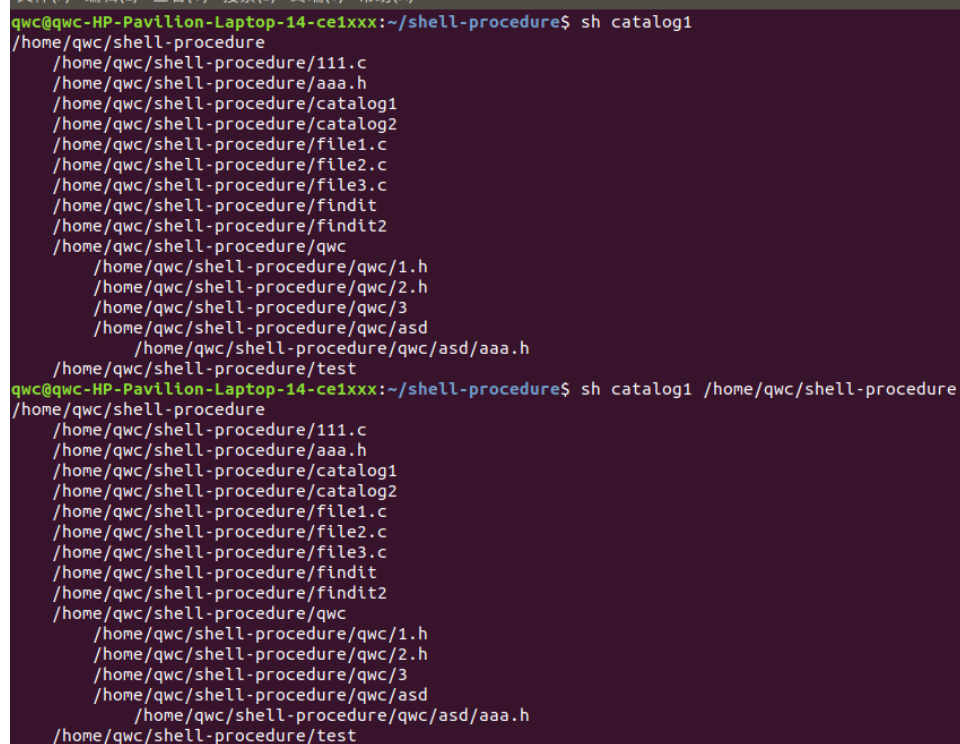
function read_dir(){
    local i=$2      #i 用来记录某文件的递归轮数
    for file in `ls $1`
    do
        if [ -d $1/"$file" ]
        then
            spaces=$(seq -s ' ' 0 $i | sed 's/[0-9]//g')    #确定空格的
            #个数并形成字符串
            echo "$spaces"$1/"$file"    #切换到子目录前打印该目录名
            read_dir $1/"$file" `expr $i + 1`
        else
            spaces=$(seq -s ' ' 0 $i | sed 's/[0-9]//g')
            echo "$spaces"$1/"$file"    #打印该文件
        fi
    done
}

if [ $# -eq 0 ]
then
    project_path=$(cd `dirname $0`; pwd)    #获取当前目录名
    echo $project_path    #先打印一下父目录
    read_dir $project_path 1    #该目录填补到递归函数的第一个参数，1 是递归
    #第二个参数用来记轮数
else

```

```
echo $1    #先打印一下父目录
read_dir $1 1    #输入的的第一个参数给递归函数的第一个参数，1 是递归第二个
参数用来记轮数
fi
```

在写这段的时候碰到的问题一个是如何在合适的位置插入 4 个空格，二是如何打印 n 个空格。解决第一个问题时主要想到了在 C++ 编程时使用递归函数记递归轮次的例子，想到可以在函数中加一个参数，在 shell 中这个参数是 \$n，调用递归函数时引用就可以，并且是一个仅在一轮递归中有效的值，因此要声明 local，这样每一次递归，轮数就加一，达到子目录缩进 4 个，子子目录缩进 8 个，以此类推；第二个问题在网上查找时发现一个命令 seq，来输出序列的，只要在输出序列时两个数字之间加空格，最后把数字都去掉就可以了，sed 可以去掉数字。catalog1 中每次输出都是绝对路径，catalog2 中只有当前目录或指定目录是相对路径，其他都是绝对路径（仅文件名）。实验结果：



```
qwc@qwc-HP-Pavillion-Laptop-14-ce1xxx:~/shell-procedure$ sh catalog1
/home/qwc/shell-procedure
/home/qwc/shell-procedure/111.c
/home/qwc/shell-procedure/aaa.h
/home/qwc/shell-procedure/catalog1
/home/qwc/shell-procedure/catalog2
/home/qwc/shell-procedure/file1.c
/home/qwc/shell-procedure/file2.c
/home/qwc/shell-procedure/file3.c
/home/qwc/shell-procedure/findit
/home/qwc/shell-procedure/findit2
/home/qwc/shell-procedure/qwc
/home/qwc/shell-procedure/qwc/1.h
/home/qwc/shell-procedure/qwc/2.h
/home/qwc/shell-procedure/qwc/3
/home/qwc/shell-procedure/qwc/asd
/home/qwc/shell-procedure/qwc/asd/aaa.h
/home/qwc/shell-procedure/test
qwc@qwc-HP-Pavillion-Laptop-14-ce1xxx:~/shell-procedure$ sh catalog1 /home/qwc/shell-procedure
/home/qwc/shell-procedure
/home/qwc/shell-procedure/111.c
/home/qwc/shell-procedure/aaa.h
/home/qwc/shell-procedure/catalog1
/home/qwc/shell-procedure/catalog2
/home/qwc/shell-procedure/file1.c
/home/qwc/shell-procedure/file2.c
/home/qwc/shell-procedure/file3.c
/home/qwc/shell-procedure/findit
/home/qwc/shell-procedure/findit2
/home/qwc/shell-procedure/qwc
/home/qwc/shell-procedure/qwc/1.h
/home/qwc/shell-procedure/qwc/2.h
/home/qwc/shell-procedure/qwc/3
/home/qwc/shell-procedure/qwc/asd
/home/qwc/shell-procedure/qwc/asd/aaa.h
/home/qwc/shell-procedure/test
```

```

/home/qwc/shell-procedure/test
qwc@qwc-HP-Pavilion-Laptop-14-ce1xxx:~/shell-procedure$ sh catalog2
/home/qwc/shell-procedure
111.c
aaa.h
catalog1
catalog2
file1.c
file2.c
file3.c
findit
findit2
qwc
  1.h
  2.h
  3
  asd
    aaa.h
  test
qwc@qwc-HP-Pavilion-Laptop-14-ce1xxx:~/shell-procedure$ sh catalog2 /home/qwc/shell-procedure
/home/qwc/shell-procedure
111.c
aaa.h
catalog1
catalog2
file1.c
file2.c
file3.c
findit
findit2
qwc
  1.h
  2.h
  3
  asd
    aaa.h
  test
qwc@qwc-HP-Pavilion-Laptop-14-ce1xxx:~/shell-procedure$ █

```

### 三、实验总结

本实验主要是学习 shell 编程，了解基本语法并进行简单的应用。总体来说不复杂，但还是花了很多时间，处理一些语法上的问题。在这次作业以后，感觉对 if 语句、test 命令、case 语句、for 语句、参数、字符串处理等有了很深刻的理解，并自己在网上或者书上查到 find、grep、seq、sed 等命令来简化代码，感觉学习到了很多东西，对自己的编程能力也是一个锻炼。最重要的当然还是把之前自己看书学到的 Linux 命令、Shell 编程语法运用了一下，弥补了自学上的很多问题。