

操作系统实践

10152130137 汪贻俊

Project 0

Introduction

本次作业主要是了解了 shell 命令的用法，还有如何使用 gdb 调试程序，还有就是 awk 的用法。

Part1

s1.sh

```
#!/bin/bash
```

```
read file
mkdir $file
touch name.txt stno.txt
echo 'Wang Yijun' > name.txt
echo '10152130137' > stno.txt
cp name.txt $file"/name.txt"
cp stno.txt $file"/stno.txt"
```

Solution:

按照题目的要求，首先就是 read, 读入文件名，然后使用 mkdir 命令新建目标文件夹，touch 命令新建文件，然后使用 echo 和 > 重定向输出流，输出名字和学号到相应的文件，然后使用 cp 命令 copy 两个文件到相应目录下。

s2. sh

```
#!/bin/bash
#find file start with letter "b" in /bin
find /bin -name "b*" > tmp.txt
#exclude '/bin'
cat tmp.txt | awk -F '/' '{if(NF>2) {print $0;}}' > tmp2.txt
#ls -l these file
cat tmp2.txt | xargs ls -l > tmp.txt
#create output.txt
cat /dev/null > output.txt
#get name, owner and permission of these file
cat tmp.txt | awk '{ split($9,str,"/"); print str[3], $3,$1; }' >
output.txt
#sort by filename
sort output.txt -o output.txt
rm tmp.txt tmp2.txt
#change premission of output.txt
chmod 644 output.txt
```

Solution:

首先使用 find 的命令在/bin 目录下查找，匹配的正则表达式是“b*”来查找文件名以 b 开头的文件，获得文件路径后在会多余一个/bin 目录，然后使用 awk 以 “/” 为分隔符删去/bin，然后将结果使用通道直接给 xargs ls -l 来获取以 b 开头的文件的 owner,permission,然后再使用 awk 输出每一行的这三个字段，然后使用 sort 命令对文件内容进行排序，最后修改 output.txt 文件的权限即可。

Part 2

1. 用命令 gcc -g -o set_op set_operation.c 出现编译错误, 发现 26 行有两个 -> , 删掉一个后继续编译, 编译成功。

```
wyj@ubuntu: ~/oslab
wyj@ubuntu:~/oslab$ gcc -g -o set_op set_operation.c
set_operation.c: In function 'check':
set_operation.c:26:12: error: expected identifier before '->' token
    if((p->->next)->number==num)
           ^
wyj@ubuntu:~/oslab$
```

2. 然后启动 gdb 开始调试,在 26 行 `if((p->next)->number==num)` 设置断点开始调试,发现程序中断 91 行出错,找到错误是循环的结束条件错误,

`for(i=0;i<=A_size;i++)` 改为 `for(i=0;i<A_size;i++)`

```
wyj@ubuntu: ~/oslab
23     struct node *p;
24     p=head;
25     while(p!=NULL){
26         if((p->next)->number==num)
27             sign=1;
28         p=p->next;
(gdb) b 26
Breakpoint 1 at 0x400770: file set_operation.c, line 26.
(gdb) r
Starting program: /home/wyj/oslab/set_op
-----Computing (A-B)union(B-A)-----
-----input the number of elements of A: 3
1-th element: 3
2-th element: 4
3-th element: 5
-----input the number of elements of B: 2
1-th element: 1
2-th element: 4

Program received signal SIGSEGV, Segmentation fault.
0x0000000000400993 in main () at set_operation.c:91
91         p2->number=p3->number;
(gdb) p p->next
No symbol "p" in current context.
(gdb)
```

3. 修改后重新编译再次调试设置 26 行断点,发现 `p->next` 会为 NULL,
修改 `p->next` 为 `p`

```
wyj@ubuntu: ~/oslab
2-th element: 4
3-th element: 5
----input the number of elements of B: 2
1-th element: 1
2-th element: 4

Breakpoint 1, check (num=3, head=0x603890) at set_operation.c:26
26      if((p->next)->number==num)
(gdb) p p->next
$1 = (struct node *) 0x6038b0
(gdb) c
Continuing.

Breakpoint 1, check (num=3, head=0x603890) at set_operation.c:26
26      if((p->next)->number==num)
(gdb) display p->next
1: p->next = (struct node *) 0x0
(gdb) c
Continuing.

Program received signal SIGSEGV, Segmentation fault.
0x0000000000400778 in check (num=3, head=0x603890) at set_operation.c:26
26      if((p->next)->number==num)
1: p->next = (struct node *) 0x0
(gdb) q
A debugging session is active.
```

4. 再次调试时设置 144 行断点, 打印 `p1->number` 发现值为 4, 和程序逻辑违背, 修改 `if(!check(p1->number, B_head))` 为
`if(check(p1->number, B_head))`, 同时发现 133 行存在同样的问题
`if(!check(p1->number, A2_head))` 修 改 为
`if(check(p1->number, A2_head))`

```

wyj@ubuntu: ~/oslab
http://www.gnu.org/software/gdb/bugs/>.
find the GDB manual and other documentation resources online at:
http://www.gnu.org/software/gdb/documentation/>.
or help, type "help".
type "apropos word" to search for commands related to "word".
gdb) file set_op
loading symbols from set_op...done.
gdb) b 144
Breakpoint 1 at 0x400b23: file set_operation.c, line 144.
gdb) r
Starting program: /home/wyj/oslab/set_op
-----
---Computing (A-B)union(B-A)---
-----
---input the number of elements of A: 3
1-th element: 3
2-th element: 4
3-th element: 5
---input the number of elements of B: 2
1-th element: 1
2-th element: 4

Breakpoint 1, main () at set_operation.c:144
44      if(sign==0){
gdb) p p1->number
1 = 4
gdb)

```

5. 再次编译后运行结果正确。

```

wyj@ubuntu: ~/oslab
(gdb) p p1->number
$1 = 4
(gdb) q
A debugging session is active.

        Inferior 1 [process 4227] will be killed.

Quit anyway? (y or n) y
wyj@ubuntu:~/oslab$ vim set_operation.c
wyj@ubuntu:~/oslab$ gcc -g -o set_op set_operation.c
wyj@ubuntu:~/oslab$ ./set_op
-----
----Computing (A-B)union(B-A)----
-----
----input the number of elements of A: 3
1-th element: 3
2-th element: 4
3-th element: 5
----input the number of elements of B: 2
1-th element: 1
2-th element: 4
---- elements of (A-B)union(B-A) ----
1-th element: 3
2-th element: 5
3-th element: 1
sh: 1: pause: not found
wyj@ubuntu:~/oslab$

```

Bonus 1

b1. sh

```
#!/bin/bash
read file
cat $file | awk '{ if(NR == 10) { print $0 } }'
```

Solution:

就是 awk 的 NR 变量代表行号，NR=10 时输出即可

Bonus 2

b2. sh

```
#!/bin/bash
read file
cat $file | awk 'BEGIN {row=0; col=0;}
{for(i=1;i<=NF;i++)str[i,row]=$i;row+=1;col=NF;}
END {for(i=1;i<=col;++i) { for(j=0;j<row;++j) {printf("%s ",
str[i,j]);}printf("\n");}}'
```

Solution:

使用 awk 将读到的每一行的每一个域转置存放在一个二维数组中，然后输出二维数组的每一个元素即可，然后注意使用 printf 格式化输出。