```
who.s who.o:who.c
                          ../ include /unistd.h ../ include /errno.h
   ../ include /asm/segment.h
5. 编译内核;
6. 挂载虚拟机硬盘,打开目录"/home/shiyanlou/oslab/hdc/usr/include",添加头文件"usname.h",键入如图2所示代
码;
7. 将文件:
"/home/shiyanlou/oslab/linux -0.11/include/unistd.h"
复制到以下路径:
"/home/shiyanlou/oslab/hdc/usr/include";
将文件:
"/home/shiyanlou/oslab/linux -0.11/include/linux/sys.h"
复制到以下路径:
"/home/shiyanlou/oslab/hdc/usr/include/linux";
8. 在"/home/shiyanlou/oslab/hdc/usr/root"目录下编写程序"iam.c"和"whoami.c":
文件"iam.c"的内容:
   #include <usname.h>
   int main int char
        if (argc> 1) { if (iam(argv[ 1])< 0) return -1;}
        else return - 1;
        return 0;
文件"whoami.c"的内容:
   #include <usname.h>
   #include <stdio.h>
   int main void
        char str[ 128];
        if (whoami(str, 24)<0) return -1;
        else printf ("%s\n", str);
        return 0;
9. 从"/home/teacher"拷贝文件"testlab2.c"和"testlab.sh"至以下路径:
"/home/shiyanlou/oslab/hdc/usr/root"
10. 取消虚拟机硬盘挂载,运行虚拟机:
(1)键入以下三个命令:
   gcc -o testlab2 testlab2 .c
(2)
   gcc -o iam iam.c - Wall
(3)
   gcc -o whoami whoami c - Wall
(2)键入以下两个命令:
1
   ./iam NZGHDYTY
(2)
   ./whoami
之后键入:
   ./testlab2
运行结果如图3所示;
(3) 键入以下两个命令:
1
   chmod +x testlab2 .sh
(2)
   ./testlab2.sh
运行结果如图4所示;
五、实验截图:
冬1
                                             who.c (~/oslab/linux-0.11/kernel) - gedit
                                                                                                                            _ @ X
                                 Tools
  <u>File</u>
        Edit
               View
                       Search
                                                      <u>H</u>elp
                                         <u>D</u>ocuments
 who.c %
 #define _LIBRARY_
 #include<unistd.h>
 #include<errno.h>
 #include<asm/segment.h>
 char usnm[64]={0};
 int sys_iam(const char *name)
     int result=0; int cnt;
     while(get_fs_byte(name+result)!="\0'&&result<64) result++;
     if(result>23) return -EINVAL;
     else
         for(cnt=0;cnt<=result;cnt++) usnm(cnt)=get_fs_byte(name+cnt);
         return result;
int sys_whoami(char *name, unsigned int size)
     int result=0; int cnt;
     while(usnm[result]!="\0"&&result<64) result++;
     if(result>size) return -1;
     else
     {
         for(cnt=0;cnt<=result;cnt++) put_fs_byte(usnm[cnt], (name+cnt));
         return result;
                                                                                                       Ln 21, Col 34
                                                                              Tab Width: 8 ▼
 ♦ 应用程序菜单:● Bochs x86 emulat··· 🕝 who.c (~/oslab/lin··· 🖿 kernel - 文件管理器 💻 shiyanlou@94007 👀 com
冬2
                                          usname.h (hdc -/oslab/hdc/usr/include) - gedit
                                                                                                                            _ @ X
                                                      Help
        <u>E</u>dit
               View
                       Search
                                 Tools
  File
                                         <u>D</u>ocuments
 usname.h 💥
 #define _LIBRARY_
 #include<unistd.h>
 #include<errno.h>
 #include<asm/segment.h>
 _syscall1(int, iam, const char*, name)
 _syscall2(int, whoami, char *, name, unsigned int, size)
                                                                                                        Ln 6, Col 57
                                                     C/C++/ObjC Header ▼ Tab Width: 8 ▼
 ♦ 应用程序菜单 📴 usname.h (hdc ~/o··· 🖿 include - 文件管理器 💻 shiyanlou@94007···
                                                                                                               shiyanlou.com
图3
                     Bochs x86 emulator, http://bochs.sourceforge.net/
Booting Base Station X now, please wait...
Partition table ok.
38985/62000 free blocks
19509/20666 free inodes
3454 buffers = 3536896 bytes buffer space
Free mem: 12582912 bytes
 Ok.
                                                                                                            dbg-c
[/usr/root]# gcc -o testlab2 testlab2.c
[/usr/root]# gcc -o iam iam.c -Wall
[/usr/root]# gcc -o whoami whoami.c -Wall
[/usr/root]# ./iam NZGHDYTY
[/usr/root]# ./whoami
NZGHDYTY
                                                                                                           rungdb
                                                                                              un
[/usr/root]# ./testlab2
[/usr/rootl# ./testlab/]

Test case 1:name = "x", length = 1...PASS

Test case 2:name = "sunner", length = 6...PASS

Test case 3:name = "Twenty-three characters", length = 23...PASS

Test case 4:name = "123456789009876543211234", length = 24...PASS

Test case 5:name = "abcdefghijklmnopqrstuvwxy...", length = 26...PASS

Test case 6:name = "Linus Torvalds", length = 14...PASS

Test case 7:name = "NULL", length = 0...PASS

Test case 8:name = "who ami(0xhalabala, 10)", length = 22...PASS
Test case 8:name = "whoami(0xbalabala, 10)", length = 22...PASS
Final result: 50%
[/usr/root]#
CTRL + 3rd button enables nouse A: HD:0-H NUN CAPS SCRL
                   Build from CVS snapshot, on June 3, 2008
0000000000i[
                       ] reading configuration from ./bochs/bochsrc.bxrc
00000000001[
                       ] installing x module as the Bochs GUI
00000000001[
                       ] using log file ./bochsout.txt
12 个项目(72.4 MB), 可用空间: 9.1 GB

◆ 应用程序菜单 : ● Bochs x86 emulat… ■ oslab - 文件管理器
                                                                       shiyanlou@94007···
```

4. 创建"who.c"文件:

5. 修改"Makefile"文件:

signal .o mktime .o who.o

在原第50行和第51行之间增加如下代码:

进入"/home/shiyanlou/oslab/linux-0.11/kemel"目录,创建"who.c"文件,键入如图1所示代码;

打开"/home/shiyanlou/oslab/linux-0.11/kemel"目录,然后用"gedit"软件打开"Makefile",将文件第29行修改为: