



操作系统实验一

实验课程: 操作系统原理实验

实验名称: 编译内核/利用已有内核创建 OS

专业名称: 计算机科学与技术

学生姓名: 杨培凯

学生学号: 23336279

实验地点: 实验楼 B203

报告时间: 2025 年 3 月 3 日

Section 1 实验概述

- **实验任务一：**搭建OS内核开发环境，包括：代码编辑环境、编译环境、运行环境、调试环境等。
- **实验任务二：**下载Linux内核，并将内核编译成i386 32位版本。
- **实验任务三：**使用qemu启动内核并开启远程调试(gdb)。
- **实验任务四：**熟悉Initramfs制作方法。
- **实验任务五：**编译i386版本的Busybox，随内核启动，构建简单的OS。
- **实验任务六：**按照前五个实验任务的操作步骤，完成Linux 0.11内核的编译、启动和调试。

Section 2 实验步骤与实验结果

实验任务一

- 任务要求

- 安装虚拟机和Ubuntu系统
- 配置用户环境

- 实验步骤

- 在Windows环境中下载安装Virtualbox

下载Virtualbox，并为虚拟机分配内存、运存、处理器等。

- 安装Ubuntu，并配置清华安装源

下载ubuntu-22.04.5-desktop-amd64.iso,并将其加载到"虚拟光盘文件"中，登录Ubuntu系统。

- 配置C/C++环境，安装常用工具 在Ubuntu的命令行中依次输入以下命令，安装相应工具。

```
sudo apt install binutils
sudo apt install gcc

sudo apt install nasm
sudo apt install qemu
sudo apt install cmake
sudo apt install libncurses5-dev
sudo apt install bison
sudo apt install flex
sudo apt install libssl-dev
sudo apt install libc6-dev-i386
```

- 实验结果 (由于安装时忘记截图, 故只能后面重新截图, 仅显示部分安装结果)

```
Activities Terminal 3月1 15:31
peikaiyang@peikaiyang-VirtualBox: ~
peikaiyang@peikaiyang-VirtualBox:~$ dpkg -l bison
期望状态=未知(u)/安装(i)/删除(r)/清除(p)/保持(h)
| 状态=未安装(n)/已安装(i)/仅存配置(c)/仅解压缩(u)/配置失败(F)/不完全安装(H)/触发器等待(W)/触发器未决(T)
|/ 错误?=(无)/须重装(R) (状态, 错误: 大写=故障)
||/ 名称 版本 体系结构 描述
+++-----
ii bison 2:3.8.2+dfsg-1build1 amd64 YACC-compatible parser generator
peikaiyang@peikaiyang-VirtualBox:~$ dpkg -l flex
期望状态=未知(u)/安装(i)/删除(r)/清除(p)/保持(h)
| 状态=未安装(n)/已安装(i)/仅存配置(c)/仅解压缩(u)/配置失败(F)/不完全安装(H)/触发器等待(W)/触发器未决(T)
|/ 错误?=(无)/须重装(R) (状态, 错误: 大写=故障)
||/ 名称 版本 体系结构 描述
+++-----
ii flex 2.6.4-8build2 amd64 fast lexical analyzer generator
peikaiyang@peikaiyang-VirtualBox:~$ dpkg -l libssl-dev
期望状态=未知(u)/安装(i)/删除(r)/清除(p)/保持(h)
| 状态=未安装(n)/已安装(i)/仅存配置(c)/仅解压缩(u)/配置失败(F)/不完全安装(H)/触发器等待(W)/触发器未决(T)
|/ 错误?=(无)/须重装(R) (状态, 错误: 大写=故障)
||/ 名称 版本 体系结构 描述
+++-----
ii libssl-dev:amd64 3.0.2-0ubuntu1.19 amd64 Secure Sockets Layer toolkit - development files
peikaiyang@peikaiyang-VirtualBox:~$ dpkg -l libc6-dev-i386
期望状态=未知(u)/安装(i)/删除(r)/清除(p)/保持(h)
| 状态=未安装(n)/已安装(i)/仅存配置(c)/仅解压缩(u)/配置失败(F)/不完全安装(H)/触发器等待(W)/触发器未决(T)
|/ 错误?=(无)/须重装(R) (状态, 错误: 大写=故障)
||/ 名称 版本 体系结构 描述
+++-----
ii libc6-dev-i386 2.35-0ubuntu3.9 amd64 GNU C Library: 32-bit development libraries for AMD64
```

实验任务二

- 任务要求

- 下载Linux内核
- 将内核编译成i386 32位版本, 检查Linux压缩镜像和符号表是否生成

- 实验步骤

- 下载Linux 5.10.234内核压缩包到文件夹lab1中
- 解压内核压缩包, 并进入内核文件中

```
xz -d linux-5.10.234.tar.xz
tar -xvf linux-5.10.234.tar
cd linux-5.10.234
```

- 生成i386默认配置文件, 并在文本界面调整内核配置

```
make i386_defconfig
make menuconfig
```

- 并行编译Linux内核

```
make -j8
```

- 检查Linux内核是否编译成功

- 查看Linux压缩镜像

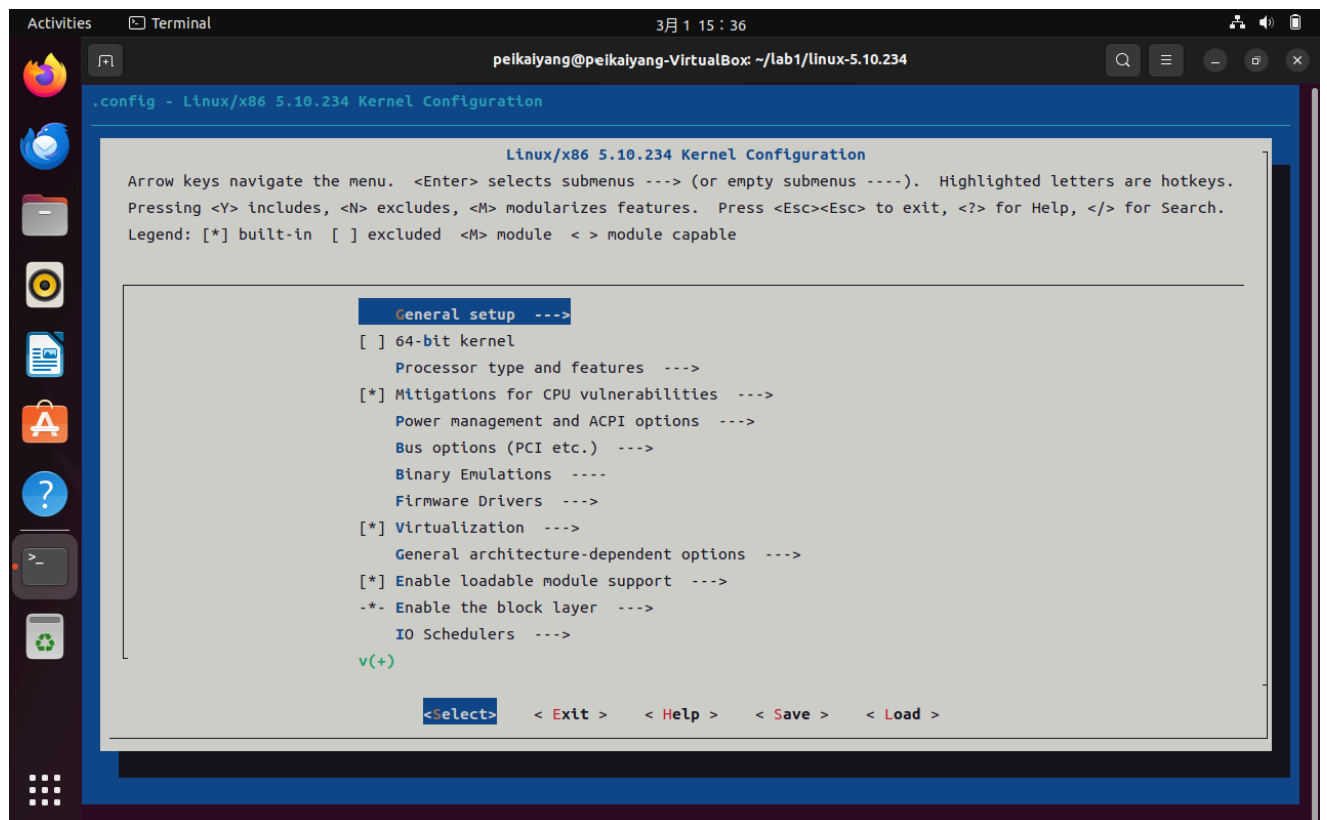
```
ls linux-5.10.234/arch/x86/boot/bzImage
```

- 查看Linux符号表

```
ls linux-5.10.234/vmlinux
```

• 实验结果

- 内核手动配置



- 并行编译内核

```
peikaiyang@peikaiyang-VirtualBox:~/lab1/linux-5.10.234$ make -j8
SYNC    include/config/auto.conf.cmd
CALL    scripts/atomic/check-atomics.sh
CALL    scripts/checksyscalls.sh
CHK     include/generated/compile.h
Kernel: arch/x86/boot/bzImage is ready (#1)
```

- Linux编译结果

```
peikaiyang@peikaiyang-VirtualBox:~/lab1/linux-5.10.234$ ls -l arch/x86/boot/bzImage
-rw-rw-r-- 1 peikaiyang peikaiyang 8871648  2月 27 00:06 arch/x86/boot/bzImage
peikaiyang@peikaiyang-VirtualBox:~/lab1/linux-5.10.234$ ls -l vmlinux
-rwxrwxr-x 1 peikaiyang peikaiyang 383178560  2月 27 00:06 vmlinux
```

实验任务三

- 任务要求

- 使用qemu启动内核
- 使用gdb远程调试内核运行

- 实验步骤

- 使用qemu运行Linux内核镜像，并等待gdb连接后再继续运行

```
qemu-system-i386 -kernel linux-5.10.19/arch/x86/boot/bzImage -s -S -
append "console=ttyS0" -nographic
```

- 在另一个终端启动gdb

```
gdb
```

- 进入Linux文件，在gdb下加载符号表

```
cd linux-5.10.234
file linux-5.10.234/vmlinux
```

- 在gdb中连接到qemu

```
target remote:1234
```

- 在gdb中为start_kernel设置断点

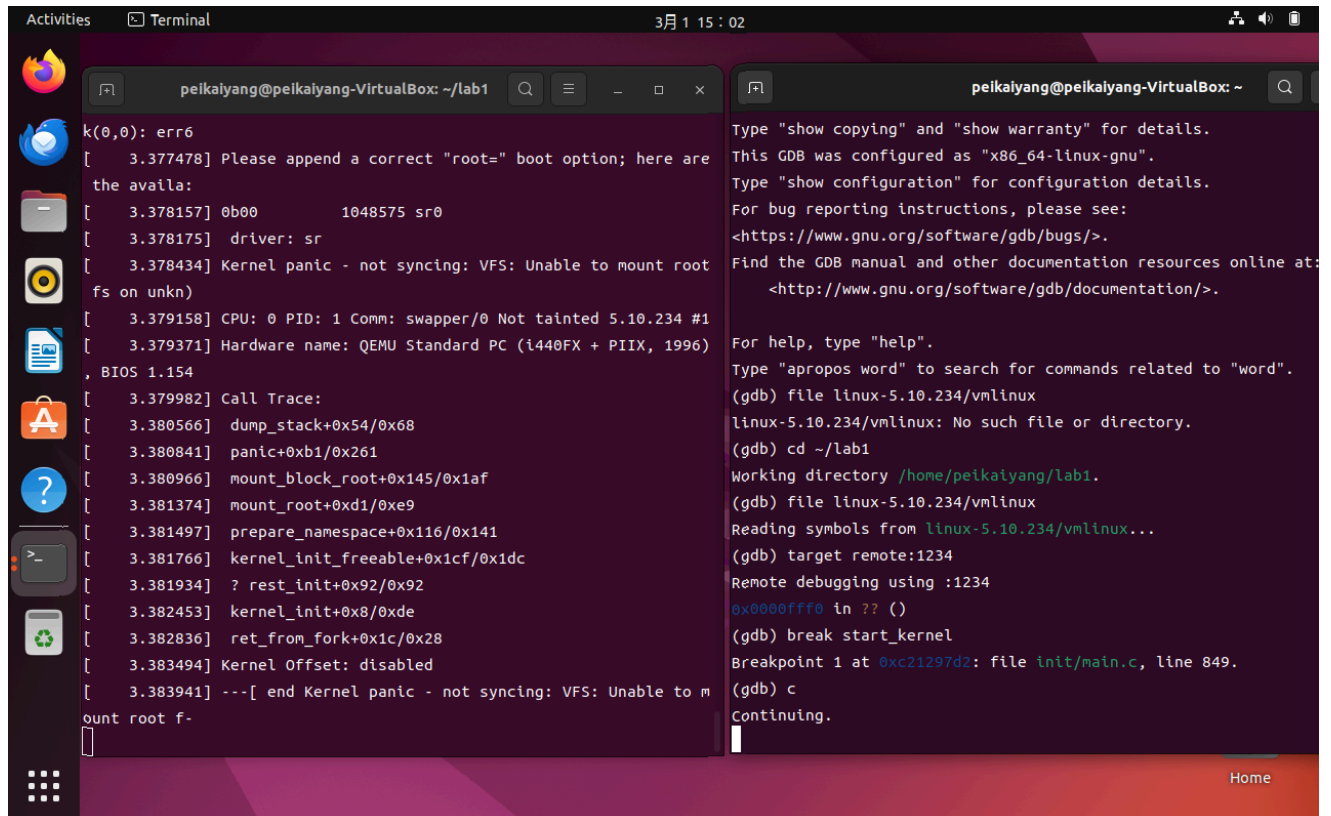
```
break start_kernel
```

- 在gdb中让调试程序继续运行

```
c
```

实验结果

Linux系统能够成功运行，但是最终以Kernel Panic结束。在终端输出的Call Trace指明了是在initrd_load阶段出错。因为在启动系统的时候，只指定了bzImage内核，但没有提供initrd文件，也没有initramfs文件系统，因此导致启动失败。



实验任务四

任务要求

- 编写简单的C语言程序制作initramfs
- 加载initramfs，使用gdb调试

• 实验步骤

- 编写C语言程序

- 使用nano编辑器创建helloworld.c文件

```
nano helloworld.c
```

- 在helloworld.c中写入一个简单程序

```
#include <stdio.h>

void main()
{
    printf("lab1: Hello World\n");
    fflush(stdout);
    while(1);
}
```

- 编译上述C语言程序

```
gcc -o helloworld -m32 -static helloworld.c
```

- 用cpio打包initramfs得到hwinitramfs

```
echo helloworld | cpio -o --format=newc > hwinitramfs
```

- 启动内核，并加载initramfs

```
qemu-system-i386 -kernel linux-5.10.19/arch/x86/boot/bzImage -initrd  
hwinitramfs -s -S -append "console=ttyS0 rdinit=helloworld" -nographic
```

- 重复任务三中gdb调试步骤

• 实验结果

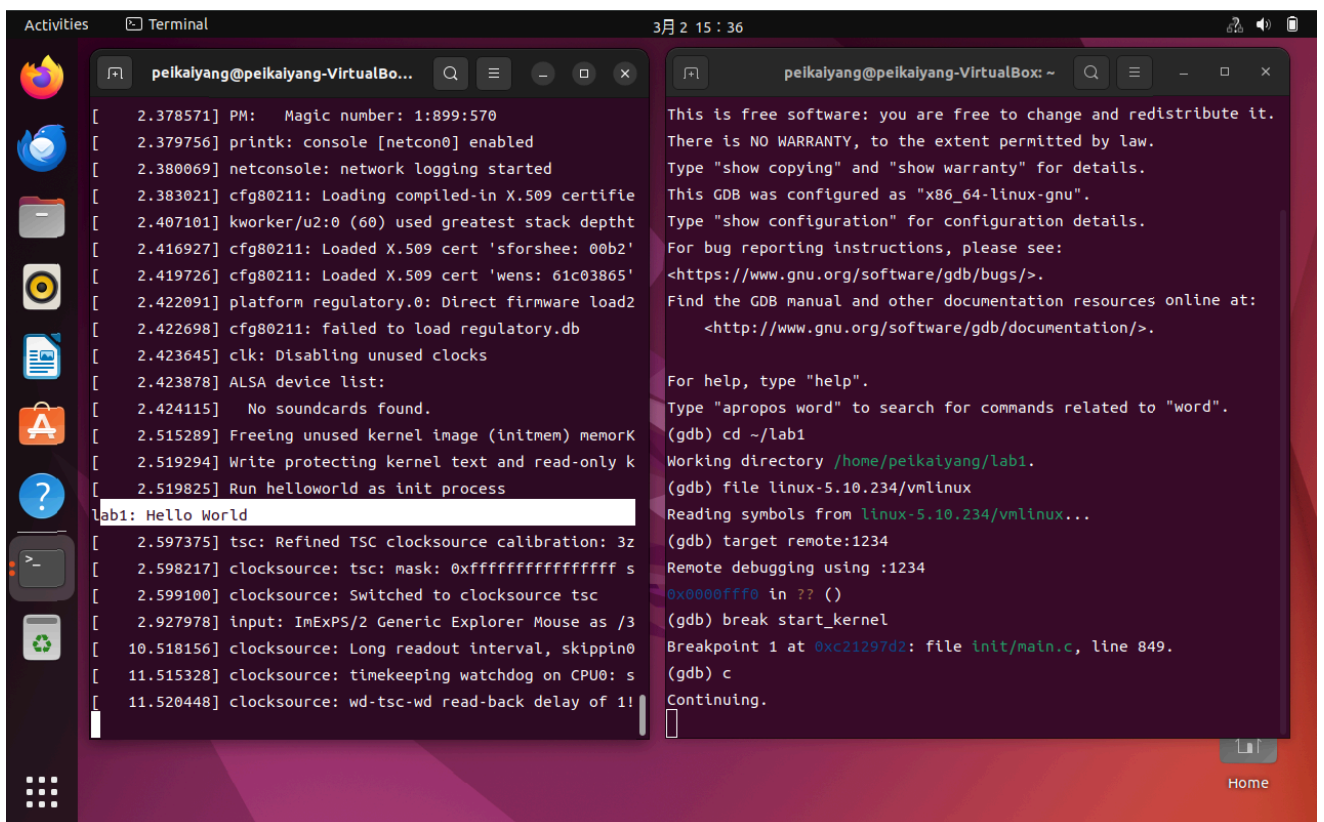
- 编写及编译C语言程序结果

```
peikaiyang@peikaiyang-VirtualBox:~/lab1$ cd ~/lab1
peikaiyang@peikaiyang-VirtualBox:~/lab1$ nano helloworld.c
peikaiyang@peikaiyang-VirtualBox:~/lab1$ ls
helloworld.c  linux-5.10.234  linux-5.10.234.tar
peikaiyang@peikaiyang-VirtualBox:~/lab1$ gcc -o helloworld -m32 -static helloworld.c
peikaiyang@peikaiyang-VirtualBox:~/lab1$ ls
helloworld  helloworld.c  linux-5.10.234  linux-5.10.234.tar
```

- initramfs的制作结果

```
peikaiyang@peikaiyang-VirtualBox:~/lab1$ echo helloworld | cpio -o --format=newc
> hwinitramfs
1459 blocks
peikaiyang@peikaiyang-VirtualBox:~/lab1$ ls
helloworld  helloworld.c  hwinitramfs  linux-5.10.234  linux-5.10.234.tar
peikaiyang@peikaiyang-VirtualBox:~/lab1$ qemu-system-i386 -kernel linux-5.10.234
/arch/x86/boot/bzImage -initrd hwinitramfs -s -S -append "console=ttyS0 rdinit=h
elloworld" -nographic
```

- initramfs的gdb调试结果



实验任务五

- 任务要求

- 下载并解压Busybox
- 编译Busybox
- 制作Initramfs
- 加载Busybox

- 实验步骤

- 在课程指定网站下载Busybox到lab1文件夹，并解压

```
wget -P /home/peikaiyang/lab1/  
http://172.18.167.174:7991/course/18/ware/383/download  
mv download Busybox  
tar -xjf Busybox
```

- 编译Busybox(具体操作同任务二)

```
make defconfig  
make menuconfig  
  
make -j8  
make install
```

- 制作Initramfs

- 将安装在_install目录下的文件和目录取出放在~/lab1/mybusybox

```
cd ~/lab1  
mkdir mybusybox  
mkdir -pv mybusybox/{bin,sbin,etc,proc,sys,usr/{bin,sbin}}  
cp -av busybox-1.33.0/_install/* mybusybox/  
cd mybusybox
```

- 编写init程序

- 用gedit打开init

```
gedit init
```

- 写一个简单的shell脚本作为init

```
#!/bin/sh
mount -t proc none /proc
mount -t sysfs none /sys
echo -e "\nBoot took $(cut -d' ' -f1 /proc/uptime) seconds\n"
exec /bin/sh
```

- 加上执行权限

```
chmod u+x init
```

- 将上述内容打包归档成cpio文件

```
find . -print0 | cpio --null -ov --format=newc | gzip -9 >
~/lab1/initramfs-busybox-x86.cpio.gz
```

- 加载Busybox，并查看当前文件夹

```
cd ~/lab1
qemu-system-i386 -kernel linux-5.10.19/arch/x86/boot/bzImage -initrd
initramfs-busybox-x86.cpio.gz -nographic -append "console=ttyS0" -m
size=2048
ls
```

- 实验结果

- 当前文件夹内容

```
peikaiyang@peikaiyang-VirtualBox:~$ cd ~/lab1
peikaiyang@peikaiyang-VirtualBox:~/lab1$ ls
Busybox          helloworld.c          linux-5.10.234
busybox-1.33.0   hwinitramfs           linux-5.10.234.tar
helloworld       initramfs-busybox-x86.cpio.gz  mybusybox
```

- Busybox运行结果

```
[ 1.953706] netconsole: network logging started
[ 1.955797] cfg80211: Loading compiled-in X.509 certificates for regulatory e
[ 1.995203] modprobe (59) used greatest stack depth: 6908 bytes left
[ 2.005211] cfg80211: Loaded X.509 cert 'sforshee: 00b28ddf47aef9cea7'
[ 2.009865] cfg80211: Loaded X.509 cert 'wens: 61c038651aabdcf94bd0ac7ff06c7'
[ 2.012988] platform regulatory.0: Direct firmware load for regulatory.db fa2
[ 2.013616] cfg80211: failed to load regulatory.db
[ 2.014543] clk: Disabling unused clocks
[ 2.015145] ALSA device list:
[ 2.015334]   No soundcards found.
[ 2.093413] Freeing unused kernel image (initmem) memory: 684K
[ 2.101598] Write protecting kernel text and read-only data: 15556k
[ 2.102150] Run /init as init process
[ 2.122499] mount (63) used greatest stack depth: 6860 bytes left
```

Boot took 2.08 seconds

```
/bin/sh: can't access tty; job control turned off
/ # [ 2.197289] tsc: Refined TSC clocksource calibration: 3792.776 MHz
[ 2.198297] clocksource: tsc: mask: 0xffffffffffffffff max_cycles: 0x6d5760cs
[ 2.198702] clocksource: Switched to clocksource tsc
[ 2.590986] input: ImExPS/2 Generic Explorer Mouse as /devices/platform/i8043
[ 7.571706] clocksource: Long readout interval, skipping watchdog check: cs_0
```

实验任务六

• 任务要求

- 下载Linux 0.11内核代码
- 编译32位版本的Linux 0.11内核
- 使用qemu-system-i386加载、启动内核
- 利用gdb进行远程调试
- 创建本地挂载目录，并在目录内创建文件

• 实验步骤

- 把Linux 0.11内核代码压缩包从主机传输到虚拟机中
 - 按照[3]中共享文件夹的说明，配置好文件传输环境，把压缩包通过Shared文件夹移动到lab1文件夹中
- 解压内核压缩包，并进入Linux-0.11文件

```
tar -xvzf Linux-0.11-lab1.tar.gz
cd Linux-0.11
```

- 查看Makefile，找到CFLAGS变量，添加-g、-m32参数

```
nano Makefile
```

- 编译32位版本的Linux-0.11内核

```
make -j8
```

- 使用qemu-system-i386加载启动内核

```
qemu-system-i386 -m 16 -boot a -fda Image -hda hdc-0.11.img -s -S
```

- 利用gdb进行远程调试

- 在另一终端启动gdb，加载Linux-0.11符号表

```
gdb
cd ~/lab1/Linux-0.11
(gdb)symbol-file tools/system
```

- 远程连接qemu调试

```
(gdb)target remote :1234
```

- 设置源码目录

```
(gdb)directory /home/peikaiyang/lab1/Linux-0.11
```

- 设置汇编代码的形式

```
(gdb)set disassembly-flavor intel
```

- 在关键位置设置断点(应到加载系统起始地址, 内核入口函数),并继续执行

```
(gdb)break *0x7c00  
(gdb)break main  
(gdb)continue
```

- 观察0x7DFE和0x7DFF地址存储的内容

```
(gdb)x/2x 0x7DFE
```

- 创建本地挂载目录hdc

- 查看磁盘的分区情况

```
cd ~/lab1/Linux-0.11  
fdisk -l hdc-0.11.img
```

- 创建本地挂载目录

```
mkdir hdc
```

- 显示磁盘空间

```
df -h
```

- 挂载Linux-0.11硬盘镜像

```
sudo mount -t minix -o loop,offset=512 /home/peikaiyang/lab1/Linux-0.11/hdc-0.11.img hdc
```

- 查看是否挂载成功

```
df -h
```

- 查看挂载后的hdc目录

```
ll hdc
```

- 在目录中创建文件hello.txt

- 进入hdc的usr目录

```
cd hdc/usr  
sudo touch hello.txt  
sudo nano hello.txt
```

- 卸载文件系统hdc

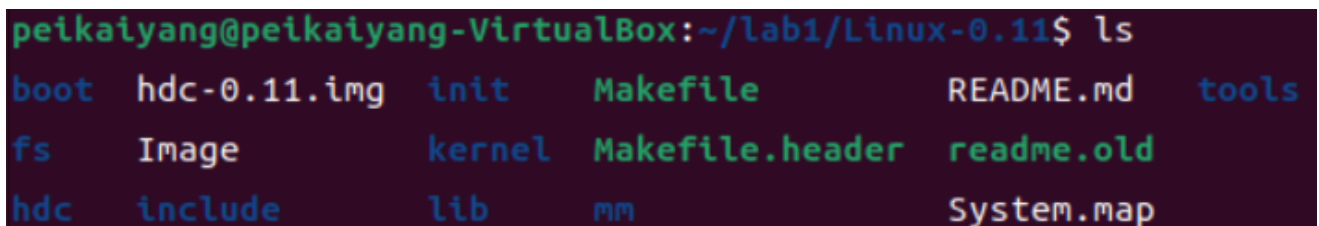
```
cd ../../  
sudo umount /dev/loop16  
df -h
```

- 重启Linux-0.11，观察/usr目录在是否还有hello.txt文件

```
qemu-system-i386 -m 16 -boot a -fda Image -hda hdc-0.11.img  
ls /usr/hello.txt
```

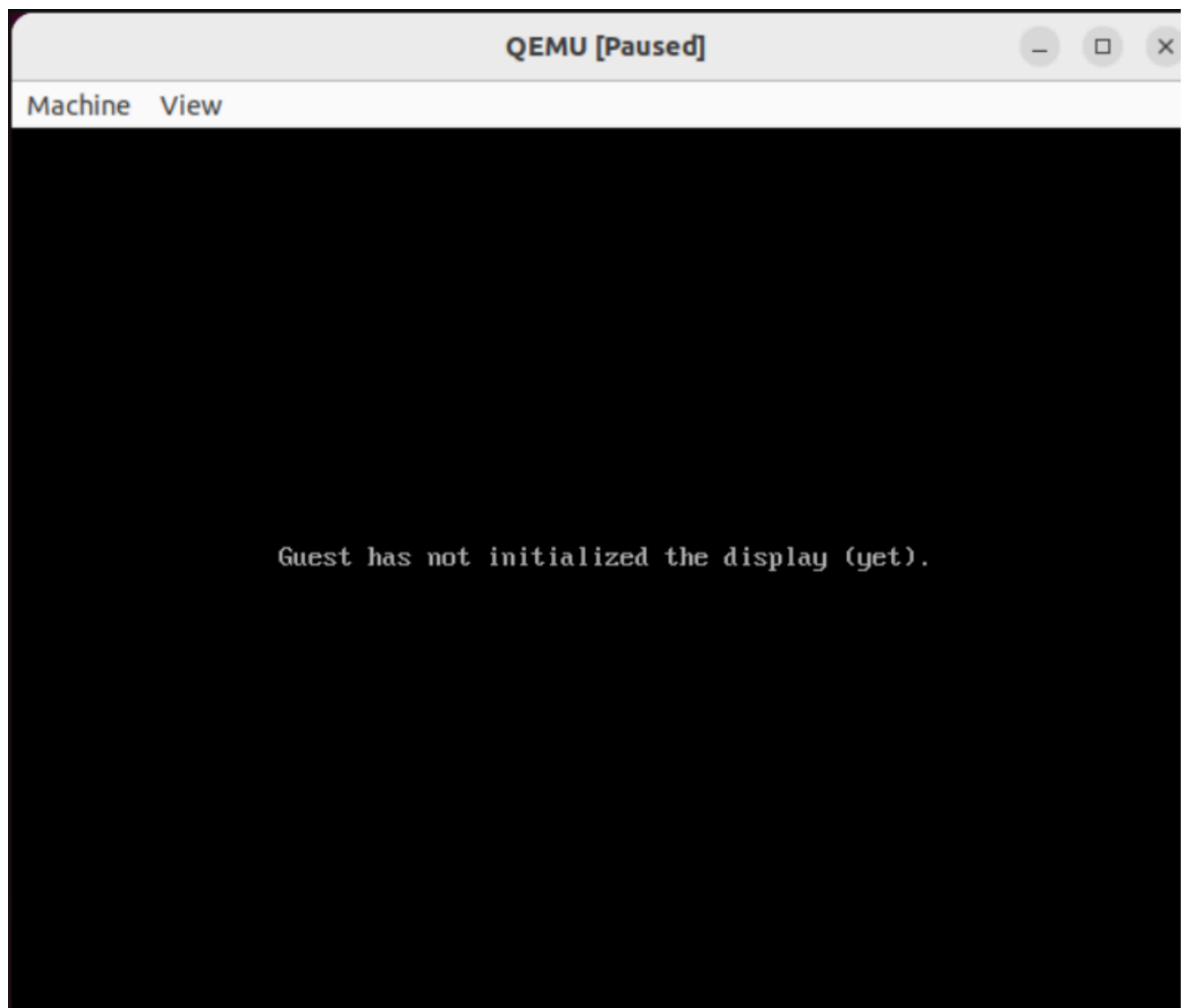
• 实验结果

- 编译Linux-0.11内核结果



```
peikaiyang@peikaiyang-VirtualBox:~/lab1/Linux-0.11$ ls  
boot  hdc-0.11.img  init      Makefile      README.md  tools  
fs    Image          kernel    Makefile.header  readme.old  
hdc   include        lib       mm             System.map
```

- 使用qemu-system-i386启动结果



- `gdb`调试过程及结果

```

(gdb) cd ~/lab1/Linux-0.11
Working directory /home/peikaiyang/lab1/Linux-0.11.
(gdb) symbol-file tools/system
Reading symbols from tools/system...
(gdb) target remote :1234
Remote debugging using :1234
warning: No executable has been specified and target does not support
determining executable automatically. Try using the "file" command.
0x0000ffff in do_execve (eip=0x0 <startup_32>, tmp=0,
      filename=0x0 <startup_32>, argv=0x0 <startup_32>, envp=0x0 <startup_32>)
      at exec.c:212
212          if (current->euid == inode->i_uid)
(gdb) directory /home/peikaiyang/lab1/Linux-0.11
Source directories searched: /home/peikaiyang/lab1/Linux-0.11:$cdir:$cwd
(gdb) set disassembly-flavor intel
(gdb) break *0x7c00
Breakpoint 1 at 0x7c00: file traps.c, line 76.
(gdb) break main
Breakpoint 2 at 0x6771: file init/main.c, line 107.
(gdb) continue
Continuing.

```

```

Breakpoint 1, die (str=0x6f5e <sleep_on+24> "", esp_ptr=31941, nr=0) at traps.c:
76
76          printk("%p ", get_seg_long(0x17, i+(long *)esp[3])
);
(gdb) x/2x 0x7DFE
0x7dfe <do_int3+105>:  0x0000aa55      0x00000000
(gdb) continue
Continuing.

Breakpoint 2, main () at init/main.c:107
107      {
/* The startup routine assumes (well, ...) this
*/
(gdb) x/2x 0x7DFE
0x7dfe <do_int3+105>:  0xff382474      0xff402474

```

- Linux-0.11硬盘镜像挂载结果


```
peikaiyang@peikaiyang-VirtualBox:~/lab1/Linux-0.11$ sudo mount -t minix -o loop,
offset=512 /home/peikaiyang/lab1/Linux-0.11/hdc-0.11.img hdc
[sudo] password for peikaiyang:
peikaiyang@peikaiyang-VirtualBox:~/lab1/Linux-0.11$ df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
tmpfs	591M	1.6M	590M	1%	/run
/dev/sda2	78G	17G	57G	23%	/
tmpfs	2.9G	0	2.9G	0%	/dev/shm
tmpfs	5.0M	4.0K	5.0M	1%	/run/lock
efivarfs	256K	131K	121K	53%	/sys/firmware/efi/efivars
tmpfs	2.9G	0	2.9G	0%	/run/qemu
/dev/sda1	511M	6.1M	505M	2%	/boot/efi
Shared	953G	448G	506G	47%	/media/sf_Shared
tmpfs	591M	136K	591M	1%	/run/user/1000
/dev/sr0	58M	58M	0	100%	/media/peikaiyang/VBox_GAs_7.1.6
/dev/loop16	58M	13M	46M	23%	/home/peikaiyang/lab1/Linux-0.11/hdc

```
peikaiyang@peikaiyang-VirtualBox:~/lab1/Linux-0.11$ ll hdc
total 13
drwxr-xr-x 10 root      root      176  3月 22  2004 ./
drwxrwxr-x 12 peikaiyang peikaiyang 4096  3月  5 18:48 ../
drwxr-xr-x  2 root      root      912  3月 22  2004 bin/
drwxr-xr-x  2 root      root      336  3月 22  2004 dev/
drwxr-xr-x  2 root      root      224  3月 22  2004 etc/
drwxr-xr-x  8 root      root      128  3月 22  2004 image/
drwxr-xr-x  2 root      root       32  3月 22  2004 mnt/
drwxr-xr-x  2 root      root       64  3月 22  2004 tmp/
drwxr-xr-x 10 root      root      192  3月  5 15:56 usr/
drwxr-xr-x  2 root      root       32  3月 22  2004 var/
```

- hdc卸载后hello.txt观察结果

```
QEMU - Press Ctrl+Alt+G to release grab

Machine View
SeaBIOS (version 1.15.0-1)

iPXE (https://ipxe.org) 00:03.0 CA00 PCI2.10 PnP PMM+00F8B590+00ECB590 CA00

Booting from Floppy...

Loading system ...

Partition table ok.
46124/60000 free blocks
19236/20000 free inodes
3419 buffers = 3501056 bytes buffer space
Free mem: 12451840 bytes
Ok.
[/usr/root]# ls /usr/hello.txt
/usr/hello.txt
[/usr/root]# _
```

Section 3 实验总结与心得体会

- 第一次使用虚拟机，加上不熟悉各种命令行命令，在配置环境这一步花费了很多时间。在Virtualbox中安装Ubuntu系统时，我试了很多次都不行。我把Virtualbox删了重下，把Ubuntu ISO文件删了，换清华源，都没成功。在询问了deepseek后，我才知道不需要在主机打开Ubuntu ISO文件，只需要直接在虚拟机中加载Ubuntu ISO文件就行了。
- 而实验任务二到实验任务五在详细的操作文档的指导下，都比较顺利地完成了。
- 在完成实验任务六的时候，碰到了较大的阻力，有四个步骤卡地比较久。第一个步骤是在传递Linux内核文件的时候，我试了两个方法：(1)直接把文件拖拽到虚拟机中、(2)尝试去Linux官网重新下载0.11版本的内核，但都失败了，翻了往年的实验指导文档，发现往年都没有任务六，因此搁置了一段时间，机缘巧合之下，在CSDN中发现了虚拟机的共享文件夹，才得以解决这个问题。第二个步骤是在编译Linux内核的时候，不知道该怎么添加-g -m32参数，询问了张皓昱同学和周宏杰同学之后，才知道要去Makefile中修改。第三个步骤是在用fdisk命令查看磁盘情况时，在终端中输入fdisk hdc-0.11.img总是没有显示正确的结果，然后是在问了deepseek后，才知道正确的命令是fdisk -l hdc-0.11.img。第四个步骤是在卸载文件系统hdc时，终端总是显示target is busy，我不知道这是什么意思，在询

问了deepseek后，我才知道因为当前文件正在运行，所以无法删除，需要先退出当前文件夹，才能顺利卸载。

- 第一次接触操作系统的实验，由于缺乏命令行、Makefile、虚拟机、操作系统等知识，感觉还是比较难的，用了差不多一个星期才完成全部实验任务。还是得尽快熟悉相关知识，希望在以后的操作系统实验中能够更顺利地完成实验。

Section 4 对实验的改进建议和意见

- 实验任务六：需要把Linux 0.11内核压缩包从主机传输到虚拟机中，这里涉及共享文件夹的使用。希望能够补充 共享文件夹 的使用说明。
- 实验任务六：查看磁盘分区情况的命令fdisk hdc-0.11.img有误，应是fdisk -l hdc-0.11.img。
- 实验任务六：在卸载文件系统hdc时，直接输入命令sudo umount /dev/loop16，终端会提示target is busy，无法卸载。因此要先在上述命令输入前输入cd ../..退出当前文件夹。

Section 5 参考资料清单

[1] 课程实验文档：<https://gitee.com/apshuang/sysu-2025-spring-operating-system/tree/master/lab1#%E7%BC%96%E8%AF%91linux%E5%86%85%E6%A0%B8>

[2] 在Ubuntu使用命令行下载网站内容说明：<https://www.oryoy.com/news/ru-he-zai-ubuntu-zhong-shi-yong-ming-ling-xing-xia-zai-wang-zhan-nei-rong-wget-yu-curl-ji-qiao-xiang.html>

[3] Virtualbox共享文件夹使用说明：
<https://blog.csdn.net/danshiming/article/details/117997558>