实验一

编译运行Linux内核并通过qemu+gdb调试

实验目的

- 熟悉Linux系统运行环境
- 掌握Linux内核编译方法
- 学习如何使用gdb调试内核
- 熟悉Linux下常用的文件操作指令

实验环境

- OS：Ubuntu 18.04LTS /16.04LTS (64bit)
- Linux内核版本：Kernel 0.11
- Hardware: 1 Core 2G RAM (若无电脑或本机配置较低，可尝试使用vlab进行实验 https://vlab.ustc.edu.cn/ 选用镜像vlab01-ubuntu-desktop-18.04.tar.gz 即可)
- Virtualization Software: Virtual box / VMware Workstation
- Kernel Version: 0.11 下载地址：课程主页
- 需要工具：gcc gdb qemu

  注意：由于原生源码由于版本较老，只能在gcc-1.4下编译，本次使用的Linux内核源码是Linux-0.11内核源码的改进版，可以在gcc-4.9以上版本编译。本次实验目的在于了解和使用qemu+gdb调试，故使用已有可编译版本，对源码有兴趣的同学可以尝试手动修改原生代码，本次实验不涉及此方面教程。源码地址 这里给出一些常见的编译错误，可自行参考。

  1. https://blog.csdn.net/hejinjing_tom_com/article/details/50294499
  2. https://blog.csdn.net/qq_42138566/article/details/89765781?depth_1-utm_source=distribute.pc_relevant.none-task&utm_source=distribute.pc_relevant.none-task

实验要求

1、需录制实验操作内容

- 利用shell命令实现以下操作：创建文件夹dir1，并在里面创建文件os.txt，往文件写入内容"Operating System expriment 1"，将该文件重命名为os_lab1.txt，最后在终端中输出文件内容
- 从Ubuntu主机端复制os_lab1.txt文件到Linux 0.11的硬盘镜像中
- 编译Linux 0.11源码并使用qemu启动，找到os_lab1.txt，并查看其内容
- 使用gdb加载符号表并远程调试，熟悉使用gdb常用命令

2、使用屏幕录制工具对实验操作进行录制，使用软件自由选择，需要达到以下要求

- 根据实验要求，配合口头说明演示关键的实验步骤
- 视频分辨率不可过低，需能看清视频中的字符
- 时间尽量控制在5分钟内
- 大小控制在100MB以内，文件过大请自行压缩视频

- 视频格式优先mp4格式，其他如rmvb,avi,wmv等也可

> 注：由于Zoom所录制视频文件体积和视频质量平衡较好，这里给出其录制视频教程，具体操作如下:
>
> 1. 开启新会议
> 2. 点击录制
> 3. 共享屏幕
> 4. 操作电脑进行实验演示
> 5. 结束会议

3、上传视频至ftp服务器

1. 服务器地址：ftp://OS2020:OperatingSystem2020@nas.colins110.cn:2001/
2. 上传至文件夹: **第一次实验**，命名格式为:**学号_姓名_实验1.mp4**，如果上传后需要修改，由于ftp服务器关闭了覆盖写入功能，需要将命名为**学号_姓名_实验1_修改n.mp4(n为修改版本)**，以最后修改版本为准。
3. 实验截止日期：**2020-04-12 23:59**

实验内容

一、编译Linux内核

1、下载并编译Linux内核

- 创建内核源代码文件夹，不妨称之为Linux源代码根目录(以下简称源码根目录)

```
$ mkdir ~/oslab
$ cd ~/oslab
```

- 下载Linux源代码，并解压

```
$ wget https://git.lug.ustc.edu.cn/gloomy/ustc_os/raw/master/Linux-0.11-lab1.tar.gz
$ tar -zvxf Linux-0.11-lab1.tar.gz
```

- 进入源代码根目录，并执行编译指令,直接执行make就可以编译内核，生成两个文件，一个是内核Image，一个是内核符号文件tools/system

```
$ cd ~/oslab/Linux-0.11
$ make help  # 获取帮助
$ make       # 编译内核
$ make start  # 在qemu中启动
$ make debug  # 通过qemu & gdb进行debug，需要打开gdb进行连接
$ make all  # 修改源码后需要重新编译内核
```

- **注意**：本次使用的Linux内核源码是Linux-0.11内核源码的改进版，可以在gcc-5.0以上版本运行，原生代码只能在gcc-1.4下编译，Linux0.11源码网址为http://www.oldlinux.org/Linux.old/kernel/0.1x/。使用原生代码编译会出现较多的编译问题，具体参见

1. https://blog.csdn.net/hejinjing_tom_com/article/details/50294499
2. https://blog.csdn.net/qq_42138566/article/details/89765781?depth_1-utm_source=distribute.pc_relevant.none-task&utm_source=distribute.pc_relevant.none-task

> 由于本次实验目的在于了解和使用qemu+gdb调试，而目前已有现成可编译版本，避免浪费时间在编译源代码的修改上。当然有兴趣的同学可以试一试。

2、准备模拟器qemu

- 直接安装qemu包即可

```
$ sudo apt-get install qemu
```

- 可能ubuntu官方镜像源上没有qemu包，将镜像源切换成ustc源即可，具体方法见下
- 更换apt-get源为ustc镜像源

3、熟悉linux简单指令

- 目标：掌握ls、touch、cat、echo、mkdir、mv、cd、cp等基本指令
- 在上一步"利用busybox生成根文件系统" 运行成功之后，在qemu窗口可以看到已进入shell环境。此时就可以在我们自己制作的根文件系统中执行指令了。如下指令创建写入一个txt文件并移动文件：

```
/ # ls          # 查看当前目录下的所有文件/文件夹
/ # touch 1.txt # 创建1.txt
/ # ls
/ # echo i am 1.txt > 1.txt  # 向1.txt写入内容
/ # cat 1.txt   # 查看1.txt内容
/ # ls -l       # 查看当前目录下的所有文件/文件夹的详细信息
/ # mkdir 1     # 创建目录1
/ # mv 1.txt 1  # 将1.txt移动到目录1
/ # cd 1        # 打开目录1
/ # ls
```

二、gdb+qemu调试内核

1、gdb简介

- gdb是一款终端环境下常用的调试工具
- 使用gdb调试程序
  - ubuntu下安装gdb：sudo apt install gdb
  - 编译程序时加入-g选项，如：gcc -g -o test test.c
  - 运行gdb调试程序：gdb test
- 常用命令

```
  r/run                  # 开始执行程序
  b/break <location>     # 在location处添加断点，location可以是代码行数或函数名
  b/break <location> if <condition> # 在location处添加断点，仅当caondition条件满足才中断运行
  c/continue             # 继续执行到下一个断点或程序结束
  n/next                 # 运行下一行代码，如果遇到函数调用直接跳到调用结束
  s/step                 # 运行下一行代码，如果遇到函数调用则进入函数内部逐行执行
  ni/nexti               # 类似next，运行下一行汇编代码（一行c代码可能对应多行汇编代码）
  si/stepi               # 类似step，运行下一行汇编代码
  list                   # 显示当前行代码
  p/print <expression>   # 查看表达式expression的值
```

> gdb 命令语法与参数详细介绍参见网址https://wangchujiang.com/linux-command/c/gdb.html

2、在qemu中启动gdb server

1. 第一种方法，在终端中执行以下指令启动qemu运行内核

```
$ cd ~/oslab/Linux-0.11    //进入源代码文件夹
$ qemu-system-i386 -m 16 -boot a -fda Image -hda hdc-0.11.img -s -S   # 可以看到qemu在等待gdb连接
```

关于qemu 选项的说明:

```
-fda Image：代表你把 Image 執行目錄下
-hda hdc-0.11.img：代表你把 HD img，是一个模拟硬盘的文件，本次实验已提供
-m：设定模拟的内存大小，本地设定为 16MB
-s：服务器开启1234端口，若不想使用1234端口，则可以使用-gdb tcp:xxxx来取代-s选项
-S：开始执行就挂住
```

2.第二种方法，在Linux源代码根目录使用make

```
$ make debug
```

3、建立gdb与gdb server之间的链接

在另外一个终端运行gdb，然后在gdb界面中运行如下命令：

```
$ gdb                    #这里一定是在另外一个终端运行，不能在qemu的窗口上输入
$ target remote:1234 #则可以建立gdb和gdbserver之间的连接
$ c                    #让qemu上的Linux继续运行
```

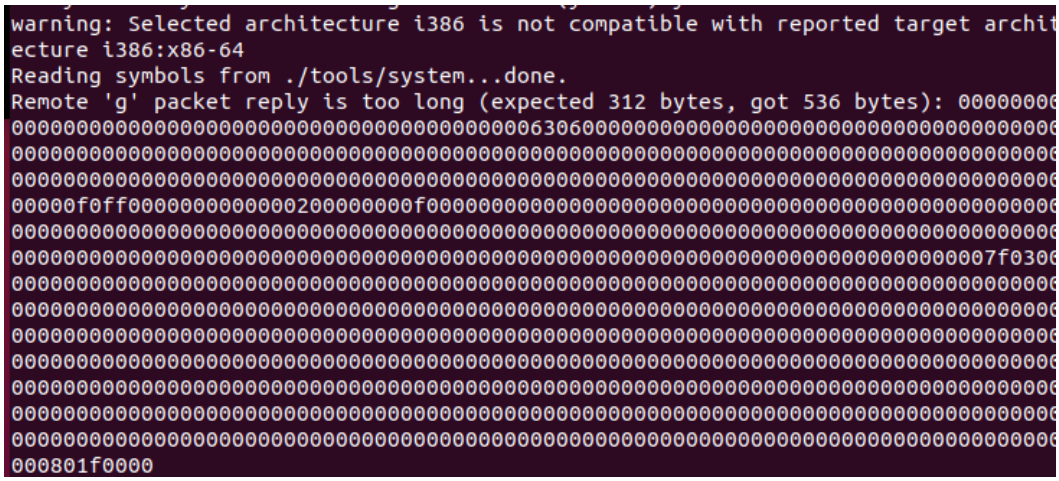
可以看到gdb与qemu已经建立了连接。但是由于没有加载符号表，无法根据符号设置断点。下面说明如何加入断点。

4、加载vmlinux中的符号表并设置断点

- 退出之前打开的qemu终端，重新执行第2步 ”在qemu中启动gdb server “

- 在另外一个终端输入如下指令运行gdb，加载符号表

```
$ gdb                       #这里一定是在另外一个终端运行，不能在qemu的窗口上输入
$ file ~/oslab/Linux-0.11/tools/system    #加载符号表
$ target remote localhost:1234 #建立gdb和gdbserver之间的连接
```

> 注意事项：若出现架构不兼容的现象，如下图
>
> 
>
>
> 在gdb中输入下面这条命令
>
> $ set architecture i386:x86-64

- 在gdb界面中设置断点

```
$ b main
$ c                         #继续运行到断点
```

三、文件交换

接下来讲解一下 Ubuntu 和 Linux 0.11 之间的文件交换操作。

> 在文件交换之前，务必关闭qemu虚拟机进程

oslab 下的 `hdc-0.11.img` 是 0.11 内核启动后的根文件系统镜像文件，相当于在 qemu 虚拟机里装载的硬盘。

```
$ cd ~/oslab/linux-0.11
$ pwd    #查看当前目录
```

1、挂载img镜像文件

1. 我们需要知道img磁盘文件，对应分区的开始位置。这样我们才好挂载。所以，先用fdisk命令查看磁盘的分区情况：

```
$ fdisk hdc-0.11.img
```

```
gesefudiao@ubuntu:~/Desktop/Linux-0.11$ fdisk hdc-0.11.img

Welcome to fdisk (util-linux 2.31.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help): p
Disk hdc-0.11.img: 59.6 MiB, 62447616 bytes, 121968 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x00000000

Device         Boot Start    End Sectors  Size Id Type
hdc-0.11.img1 *        1 120959  120959 59.1M 81 Minix / old Linux
```

2. 可以看到img文件系统类别属于Minix，有一个分区，分区是从1开始的，这里需要注意，需要转化一下：1*512=512（offset)
3. 在源码根目录下创建挂载目录

```
$ mkdir hdc
```

4. 显示磁盘空间统计信息

```
$ df -h
```

```
gesefudiao@ubuntu:~/Desktop/Linux-0.11$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            956M     0  956M   0% /dev
tmpfs           196M  1.8M  195M   1% /run
/dev/sda1        35G  7.3G   26G  23% /
tmpfs           980M     0  980M   0% /dev/shm
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
tmpfs           980M     0  980M   0% /sys/fs/cgroup
/dev/loop0      3.8M  3.8M     0 100% /snap/gnome-system-monitor/135
/dev/loop2       15M   15M     0 100% /snap/gnome-characters/495
/dev/loop1       90M   90M     0 100% /snap/core/8268
/dev/loop3      4.4M  4.4M     0 100% /snap/gnome-calculator/704
/dev/loop4      4.3M  4.3M     0 100% /snap/gnome-calculator/544
/dev/loop5       92M   92M     0 100% /snap/core/8689
/dev/loop6      1.0M  1.0M     0 100% /snap/gnome-logs/81
/dev/loop7      161M  161M     0 100% /snap/gnome-3-28-1804/116
/dev/loop8       55M   55M     0 100% /snap/core18/1668
/dev/loop9      3.8M  3.8M     0 100% /snap/gnome-system-monitor/127
/dev/loop10     1.0M  1.0M     0 100% /snap/gnome-logs/93
/dev/loop11      55M   55M     0 100% /snap/core18/1705
/dev/loop12      49M   49M     0 100% /snap/gtk-common-themes/1474
/dev/loop13      45M   45M     0 100% /snap/gtk-common-themes/1440
/dev/loop14      15M   15M     0 100% /snap/gnome-characters/399
tmpfs           196M   16K  196M   1% /run/user/121
tmpfs           196M   32K  196M   1% /run/user/1000
```

5. 挂载分区，需要使用第二步计算的参数（offset）

```
$ sudo mount -t minix -o loop,offset=512 ~/Desktop/Linux-0.11/hdc-0.11.img
~/Desktop/Linux-0.11/hdc
```

6. 显示磁盘空间统计信息

```
$ df -h
```

```
gesefudiao@ubuntu:~/Desktop/Linux-0.11$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            956M     0  956M   0% /dev
tmpfs           196M  1.8M  195M   1% /run
/dev/sda1        35G  7.3G   26G  23% /
tmpfs           980M     0  980M   0% /dev/shm
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
tmpfs           980M     0  980M   0% /sys/fs/cgroup
/dev/loop0      3.8M  3.8M     0 100% /snap/gnome-system-monitor/135
/dev/loop2       15M   15M     0 100% /snap/gnome-characters/495
/dev/loop1       90M   90M     0 100% /snap/core/8268
/dev/loop3      4.4M  4.4M     0 100% /snap/gnome-calculator/704
/dev/loop4      4.3M  4.3M     0 100% /snap/gnome-calculator/544
/dev/loop5       92M   92M     0 100% /snap/core/8689
/dev/loop6      1.0M  1.0M     0 100% /snap/gnome-logs/81
/dev/loop7      161M  161M     0 100% /snap/gnome-3-28-1804/116
/dev/loop8       55M   55M     0 100% /snap/core18/1668
/dev/loop9      3.8M  3.8M     0 100% /snap/gnome-system-monitor/127
/dev/loop10     1.0M  1.0M     0 100% /snap/gnome-logs/93
/dev/loop11      55M   55M     0 100% /snap/core18/1705
/dev/loop12      49M   49M     0 100% /snap/gtk-common-themes/1474
/dev/loop13      45M   45M     0 100% /snap/gtk-common-themes/1440
/dev/loop14      15M   15M     0 100% /snap/gnome-characters/399
tmpfs           196M   16K  196M   1% /run/user/121
tmpfs           196M   32K  196M   1% /run/user/1000
/dev/loop15      58M   13M   46M  23% /home/gesefudiao/Desktop/Linux-0.11/hdc
```

2、文件读写交换

1. 查看hdc目录结构

```
$ ll ./hdc  #查看内容
```

```
gesefudiao@ubuntu:~/Desktop/Linux-0.11$ ll ./hdc
total 13
drwxr-xr-x 10 root       root        176 Mar 21  2004 ./
drwxr-xr-x 12 gesefudiao gesefudiao 4096 Mar 27 22:21 ../
drwxr-xr-x  2 root       root        912 Mar 21  2004 bin/
drwxr-xr-x  2 root       root        336 Mar 21  2004 dev/
drwxr-xr-x  2 root       root        224 Mar 21  2004 etc/
drwxr-xr-x  8 root       root        128 Mar 21  2004 image/
drwxr-xr-x  2 root       root         32 Mar 21  2004 mnt/
drwxr-xr-x  2 root       root         64 Mar 21  2004 tmp/
drwxr-xr-x 10 root       root        224 Mar 27 22:42 usr/
drwxr-xr-x  2 root       root         32 Mar 21  2004 var/
```

> hdc 目录下就是和 0.11 内核一模一样的文件系统了，可以读写任何文件（可能有些文件要用 sudo 才能访问）。

2. 创建文件hello.txt

```
$ cd ~/oslab/Linux-0.11/hdc/usr
$ sudo touch hello.txt   # 创建文件
$ sudo vim hello.txt   # 向文件写入hello world!
```

```
gesefudiao@ubuntu:~/Desktop/Linux-0.11/hdc/usr$ ll
total 12
drwxr-xr-x 10 root root  224 Mar 27 23:19 ./
drwxr-xr-x 10 root root  176 Mar 21  2004 ../
drwx--x--x  2 root root 1200 Mar 21  2004 bin/
drwx--x--x  2 root root   96 Feb 14  1992 docs/
-rw-r--r--  1 root root   13 Mar 27 23:19 hello.txt
drwx--x--x  5 root   31  624 Mar 29  2004 include/
drwx--x--x  4 root root   64 Mar 29  2004 local/
drwxr-xr-x  3 root root  256 Mar 29  2004 root/
drwx--x--x  4 root root   96 Mar 21  2004 src/
drwx--x--x  2 root root   64 Mar 29  2004 tmp/
drwxr-xr-x  2 root root   32 Feb 14  1992 var/
```

3. 卸载文件系统hdc

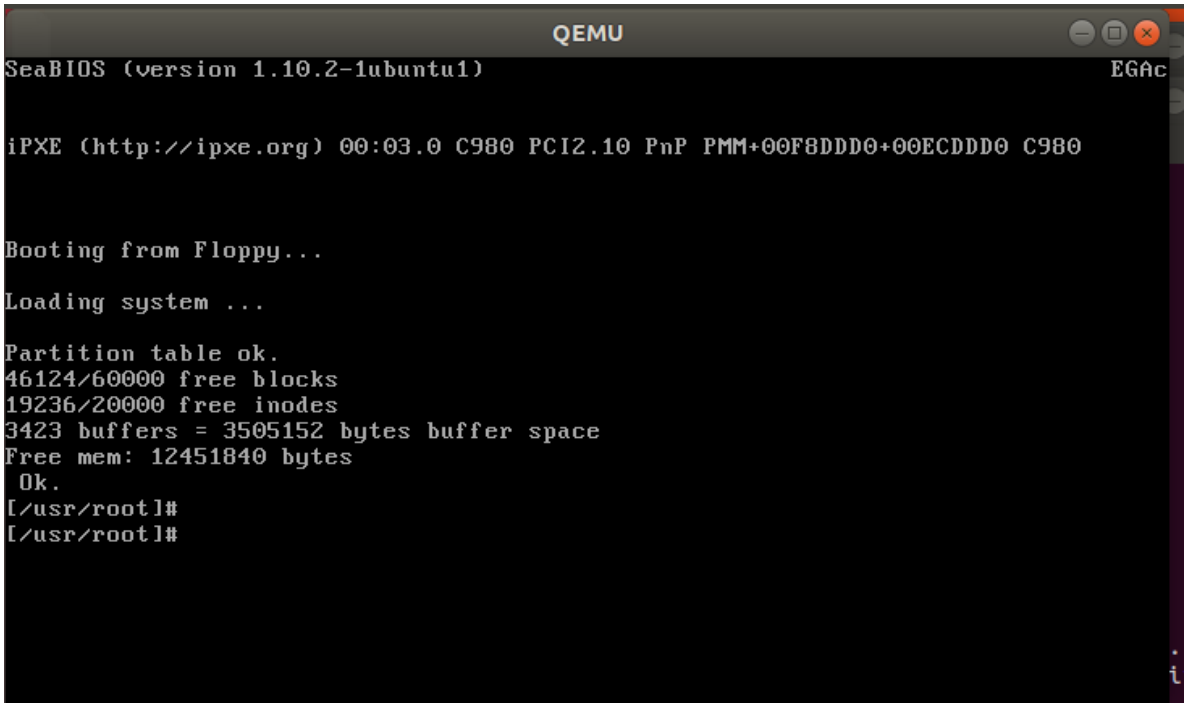
```
$ sudo umount /dev/loop15
$ df -h
```

> 注意：出现以下情况
>
> ```
> gesefudiao@ubuntu:~/Desktop/Linux-0.11/hdc/usr$ sudo umount /dev/loop15
> umount: /home/gesefudiao/Desktop/Linux-0.11/hdc: target is busy.
> ```
>
> ```
> $ cd ~/oslab/Linux-0.11   # 退出文件系统挂载的目录文件夹
> $ sudo umount /dev/loop15
> $ df -h
> ```

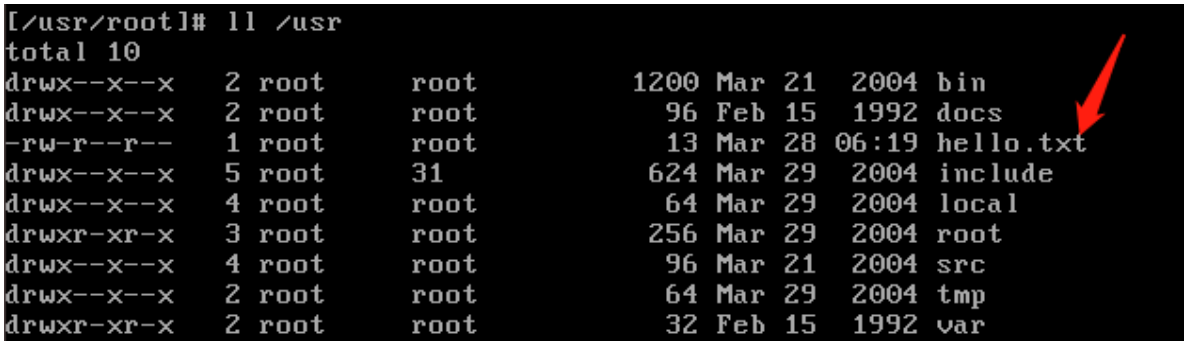
```
gesefudiao@ubuntu:~/Desktop/Linux-0.11$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            956M     0  956M   0% /dev
tmpfs           196M  1.8M  195M   1% /run
/dev/sda1        35G  7.3G   26G  23% /
tmpfs           980M     0  980M   0% /dev/shm
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
tmpfs           980M     0  980M   0% /sys/fs/cgroup
/dev/loop0      3.8M  3.8M     0 100% /snap/gnome-system-monitor/135
/dev/loop2       15M   15M     0 100% /snap/gnome-characters/495
/dev/loop1       90M   90M     0 100% /snap/core/8268
/dev/loop3      4.4M  4.4M     0 100% /snap/gnome-calculator/704
/dev/loop4      4.3M  4.3M     0 100% /snap/gnome-calculator/544
/dev/loop5       92M   92M     0 100% /snap/core/8689
/dev/loop6      1.0M  1.0M     0 100% /snap/gnome-logs/81
/dev/loop7      161M  161M     0 100% /snap/gnome-3-28-1804/116
/dev/loop8       55M   55M     0 100% /snap/core18/1668
/dev/loop9      3.8M  3.8M     0 100% /snap/gnome-system-monitor/127
/dev/loop10     1.0M  1.0M     0 100% /snap/gnome-logs/93
/dev/loop11      55M   55M     0 100% /snap/core18/1705
/dev/loop12      49M   49M     0 100% /snap/gtk-common-themes/1474
/dev/loop13      45M   45M     0 100% /snap/gtk-common-themes/1440
/dev/loop14      15M   15M     0 100% /snap/gnome-characters/399
tmpfs           196M   16K  196M   1% /run/user/121
tmpfs           196M   32K  196M   1% /run/user/1000
```

4. 查看Linux0.11文件

```
$ cd ~/oslab/Linux-0.11
$ make start
```




```
$ ll /usr  # 列举文件
```




```
$ more hllo.txt  # 查看文件内容
```

> 进入 Linux 0.11（即 run 启动 qemu 以后）就会看到这个 hello.txt（即如上图所示），这样就避免了在 Linux 0.11 上进行编辑 文件的麻烦，因为 Linux 0.11 作为一个很小的操作系统，没有便捷的编辑工具。

5. 修改Linux0.11系统中的文件hello.txt

```
$ echo hello > hello.txt
$ head hello.txt
```

```
[/usr]# ll
total 10
drwx--x--x   2 root     root         1200 Mar 21  2004 bin
drwx--x--x   2 root     root           96 Feb 15  1992 docs
-rw-r--r--   1 root     root            6 Mar 28 06:40 hello.txt
drwx--x--x   5 root     31            624 Mar 29  2004 include
drwx--x--x   4 root     root           64 Mar 29  2004 local
drwxr-xr-x   3 root     root          256 Mar 29  2004 root
drwx--x--x   4 root     root           96 Mar 21  2004 src
drwx--x--x   2 root     root           64 Mar 29  2004 tmp
drwxr-xr-x   2 root     root           32 Feb 15  1992 var
[/usr]# head hello.txt
hello
```

6、关闭Linux0.11系统，并在主机挂载img镜像

```
$ sudo mount -t minix -o loop,offset=512 ~/Desktop/Linux-0.11/hdc-0.11.img
~/Desktop/Linux-0.11/hdc
$ cd /Desktop/Linux-0.11/hdc/usr
$ ll
$ tail hello.txt
```

```
gesefudiao@ubuntu:~/Desktop/Linux-0.11$ sudo mount -t minix -o loop,offset=512 ~
/Desktop/Linux-0.11/hdc-0.11.img ~/Desktop/Linux-0.11/hdc
[sudo] password for gesefudiao:
gesefudiao@ubuntu:~/Desktop/Linux-0.11$ cd ~/D
Desktop/   Documents/ Downloads/
gesefudiao@ubuntu:~/Desktop/Linux-0.11$ cd ~/Desktop/Linux-0.11/hdc/usr/
gesefudiao@ubuntu:~/Desktop/Linux-0.11/hdc/usr$ ll
total 12
drwxr-xr-x 10 root root  224 Mar 27 23:19 ./
drwxr-xr-x 10 root root  176 Mar 21  2004 ../
drwx--x--x  2 root root 1200 Mar 21  2004 bin/
drwx--x--x  2 root root   96 Feb 14  1992 docs/
-rw-r--r--  1 root root    6 Mar 27 23:40 hello.txt
drwx--x--x  5 root   31  624 Mar 29  2004 include/
drwx--x--x  4 root root   64 Mar 29  2004 local/
drwxr-xr-x  3 root root  256 Mar 29  2004 root/
drwx--x--x  4 root root   96 Mar 21  2004 src/
drwx--x--x  2 root root   64 Mar 29  2004 tmp/
drwxr-xr-x  2 root root   32 Feb 14  1992 var/
gesefudiao@ubuntu:~/Desktop/Linux-0.11/hdc/usr$ tail hello.txt
hello
```

> 在 Linux 0.11 上产生的文件，可以按这种方式 “拿到” Ubuntu 下用 python 程序进行处理，某些文件(python文件等)在 Linux 0.11 上显然是不好处理，因为 Linux 0.11 上搭建不了 python 解释环境。

- 注意点：不要在 0.11 内核运行的时候 mount 镜像文件，否则可能会损坏文件系统。同理，也不要在已经 mount 的时候运行 0.11 内核。

本节参考资料

- gdb调试工具
- Linux0.11内核编译与调试
- 实验楼操作系统原理与实践实验课一