**2016级种子班**

**Linux初步**

**实验报告**

**实验名称： Linux初步课程实验**

**姓 名： 雷紫薇**

**学 号：** U201617155

**完成日期：** 2019/2/21

目录

[**1** **实验目的** 3](#_Toc1672679)

[**2** **实验内容** 3](#_Toc1672680)

[**2.1** **系统配置要求** 3](#_Toc1672681)

[**2.2** **主要功能** 3](#_Toc1672682)

[**3** **实验步骤** 3](#_Toc1672683)

[**3.1** **实验环境** 3](#_Toc1672684)

[**3.2** **实验原理** 3](#_Toc1672685)

[**3.3** **实验过程** 5](#_Toc1672686)

[**3.3.1** **外围文件系统** 5](#_Toc1672687)

[**3.3.2** **内核编译** 8](#_Toc1672688)

[**3.4** **实验测试** 8](#_Toc1672689)

[**4** **调试过程** 8](#_Toc1672690)

[**5** **分析总结** 8](#_Toc1672691)

1. **实验目的**

* 熟悉掌握Linux操作系统的使用
* 了解linux操作系统的运作过程，理解内核与外围支撑系统的关系
* 通过实验定制linux系统内核与外围支撑系统，加深对开源操作系统的认识
* 课程输出具有各自功能特色的自启动最小系统

1. **实验内容**
   1. **系统配置要求**

内核文件（bzImage）< 4MB

裁剪后的文件系统镜像（initrd.img）< 24MB

* 1. **主要功能**

通过U盘加载kernel和img启动进行验证

支持多用户登录（console界面和ssh网络方式）

系统支持通过ssh方式访问其他机器

可挂载U盘

可访问机器上的windows分区（ntfs-3g fs支持）

1. **实验步骤**
   1. **实验环境**

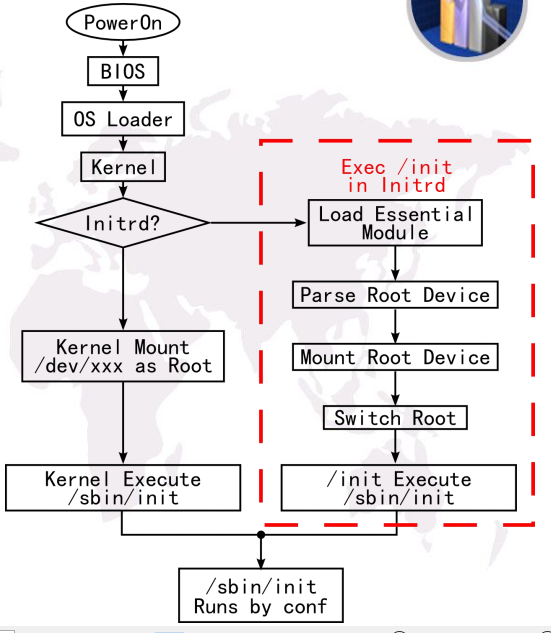
Centos6.9（内核版本：2.6.32-696.el6.x86\_64）

VMware 12 Pro

裁剪内核版本：linux-4.20.5-stable

* 1. **实验原理**

Linux启动过程如下



1. 开机后，BIOS首先取得主机的各项硬件配置，取得这些信息之后会进行开机自检的过程，这个过程其实就是检查CPU和内存，计算机最基本的组成单元(控制器、运算器和存储器)，还会检查其他硬件，若没有异常就开始加载BIOS程序到内存当中。BIOS主要的一个功能就是存储了磁盘的启动顺序，BIOS会按照启动顺序去查找第一个磁盘头的MBR信息，并加载和执行MBR中的Bootloader程序，若第一个磁盘不存在MBR，则会继续查找第二个磁盘(PS：启动顺序可以在BIOS的界面中进行设置)，一旦OS Loader程序被检测并加载内存中，BIOS就将控制权交接给了OS Loader程序。
2. OS Loader 把内核及 initrd 文件加载到内存的特定位置。Kernel是Linux系统最主要的程序，实际上，Kernel的文件很小，只保留了最基本的模块，并以压缩的文件形式存储在硬盘中，当GRUB将Kernel读进内存，内存开始解压缩内核文件。initrd是一个临时的根文件系统(rootfs)。因为Kernel为了精简，只保留了最基本的模块，因此，Kernel上并没有各种硬件的驱动程序，也就无法识rootfs所在的设备，故产生了initrd这个文件，该文件装载了必要的驱动模块，当Kernel启动时，可以从initrd文件中装载驱动模块，直到挂载真正的rootfs，然后将initrd从内存中移除。
3. Kernel会以只读方式挂载根文件系统，当根文件系统被挂载后，开始装载第一个进程(用户空间的进程)，执行/sbin/init，之后就将控制权交接给了init程序。
4. Init是进行OS初始化操作，这个文件执行的任务有：

激活udev和selinux

根据/etc/sysctl.conf文件，来设定内核参数

设定系统时钟

装载硬盘映射

启用交换分区

设置主机名

根文件系统检测，并以读写方式重新挂载根文件系统

激活RAID和LVM设备

启用磁盘配额

根据/etc/fstab，检查并挂载其他文件系统

清理过期的锁和PID文件

执行完后，根据配置的启动级别，执行对应目录底下的脚本，最后执行/etc/rc.d/rc.local这个脚本，至此，系统启动完成。

* 1. **实验过程**
     1. **外围文件系统**
        1. **V0.5 建立简单的文件系统**

最简单的文件系统，就是在启动时由initrd在内存中启动一个bash。需要我们创建自己的根目录生成相应的init文件，调用bash即可。最终打包成img文件并修改启动的grub.conf文件即可看到最终的效果。具体步骤如下：

1. 创建自己的根文件目录

选择/home/minilinux0.5作为自己的根文件目录



1. 添加原系统bash

查找bash在原系统的位置：which bash

查找bash的相关依赖：ldd /bin/bash

将bash和lib文件复制到相应的路径下

1. 生成init文件

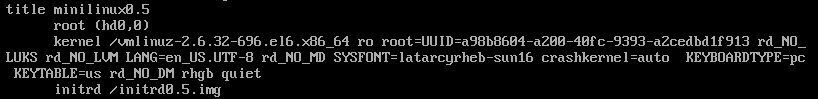
在自己的新根文件夹下创建init脚本，编辑内容如下：



获得一个bash

1. 修改grub.conf

在原系统下找到/boot/grub/grub.conf文件，添加如下内容



1. 打包生成intrid.img

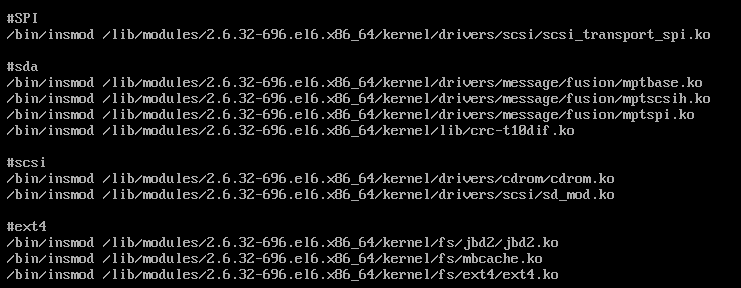
进入自己的新根文件的目录，将内容打包进intird.img

find . | cpio -H newc -o | gzip > /boot/intrid0.5.img

* + - 1. **V0.55 拥有可以挂载原系统的能力**

0.55版本要求在小系统中能够看到硬盘上的数据，这部分主要是要求系统拥有加载硬盘的各个驱动，为之后的udev自动调用驱动做准备。

首先是在虚拟机中硬盘读取需要的scsi相关驱动，另外MRT的驱动需要另外选择。由于模块比较难找，这一部分参考了同学的，最终的init文件如下：



* + - 1. **V0.6 完成拥有设备管理能力**

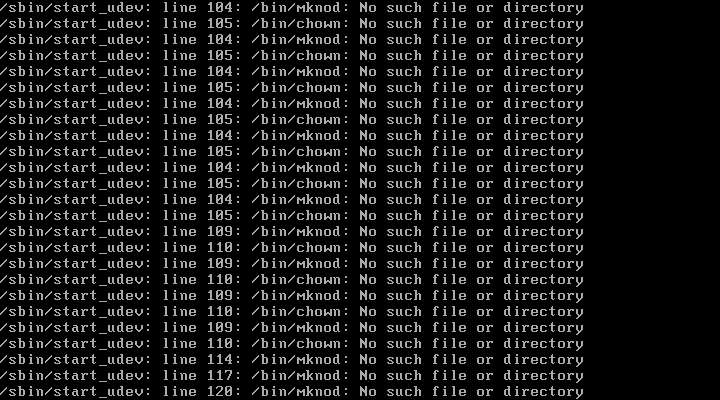
Udevd是主要的管理、监控支架设备的服务程序。通过udevd可以直接启动相应的驱动而不需要手动加载。

Udevd的规则文件在/lib/udev/目录下，配置文件在/etc/udev/目录下，将相应的配置文件个依赖的库文件拷贝到小系统根目录的相应地方。

通过在init文件中加入下面的命令启动udevd服务



在启动0.6版本的小系统时，很大的可能是无法看到udevd管理设备的效果，这时需要在小系统中加入strace命令，查找是哪些模块或者文件没有拷进小系统导致错误。



* + - 1. **V0.7 完成拥有login登录（多窗口）的能力**

登录涉及的文件有：/bin/login命令，涉及用户组管理/bin/chgrp、/bin/chown、/bin/chmod等，保存用户名的文件/etc/passwd、/etc/group，用户密码文件为/etc/shadow等/lib下的security文件夹，同时需要将这个文件夹的依赖也拷到相应的目录下。

而多窗口登录（ssh登录）需要网络功能，sshd服务，这些都在0.9之后用/sbin/init调度之后进行，具体操作在0.9之后详细说明。

但是在0.7看不到登录的效果，login模块是属于/sbin/init管理的一项，所以在0.9时才可以验证登录的效果

* + - 1. **V0.9 达到由/sbin/init管理的小系统原型**

0.7版本的login及多窗口，这些涉及到进程管理机制和进程组、控制台等许多方面，我们需要把这些模块的启动交给sbin/init来调用。

需要将/sbin/init拷进小系统，修改init脚本，使其指向/sbin/init，将控制权交给/sbin/init

配置文件在/etc/init中，同时需要/etc/rc\*.d /etc/rc /etc/rc.sysinit 等文件，这些文件是控制系统模式的重要文件，在/sbin/init之后，系统会选择runlevel。

/etc/system-release文件是可以更改系统启动时“welcome”的显示系统发行版本。在启动过程中可能出现一些问题，可以通过录屏开看缺少哪些文件。

在0.7中有几个问题没有验证：网络配置，ssh服务，下面一一解决。

**网络配置：**

拷贝网络配置相关的文件

/etc/sysconfig/network-scripts/ifcfg-eth0

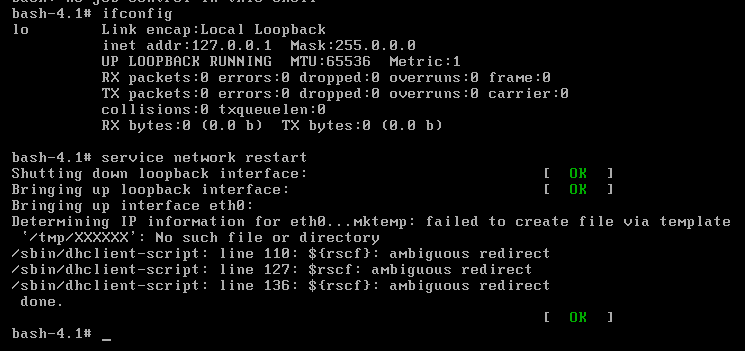
这是设置网卡参数的文件，网卡参数包括 network, ip, netmask, broadcast, gateway, 开机时的 ip 取得方式，是否在开机时启动等，其中 ifcfg-eth0 是指第一块网卡，ifcfg-eth1 指第二块，以此类推。

/bin/ipcalc /bin/cut /bin/mktemp /bin/netstat 是基本命令

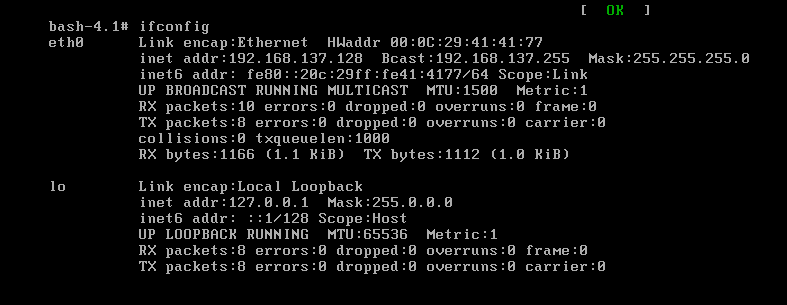
/sbin/ifconfig /sbin/ip /sbin/arping /sbin/dhclient /sbin/restorecon 也是命令

/etc/dhcp 动态分配ip

若没有出现eth0，像下面的情况：



可以通过strace service network restart或者直接执行strace network restart来看缺少哪些文件，最后得到下效果：



**ssh服务：**

拷贝相应的文件和命令

/bin/env /bin/usleep

/usr/bin/dirname

/usr/sbin/sshd

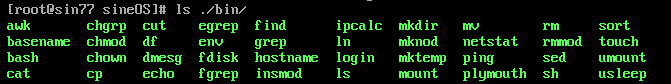
/etc/ssh /etc/ld.so.cache /etc/ld.so.conf /etc/profile.d

/usr/bin/ssh /usr/bin/ssh-keygen

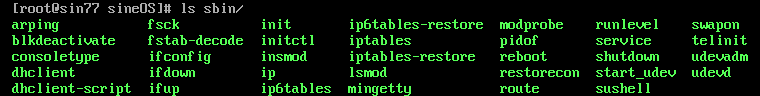
这些文件都是sshd服务需要的。

到目前为止，小系统的基本功能都已经完善，下面是各个文件夹的目录结构：

/bin



/sbin



/dev



/etc



/lib



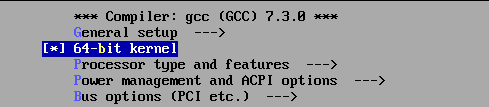
/usr



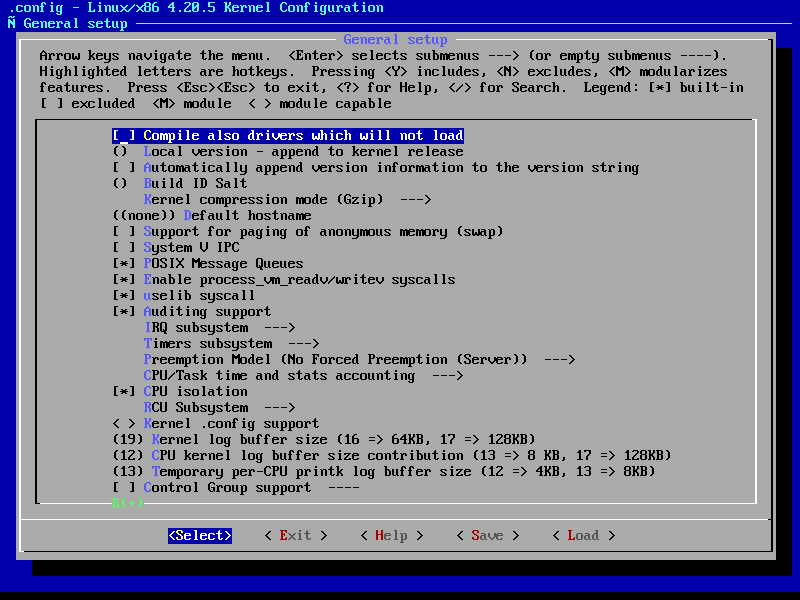
* + 1. **内核编译**

下载稳定版本的Linux内核4.20.5的版本，使用make menuconfig命令进行内核的配置

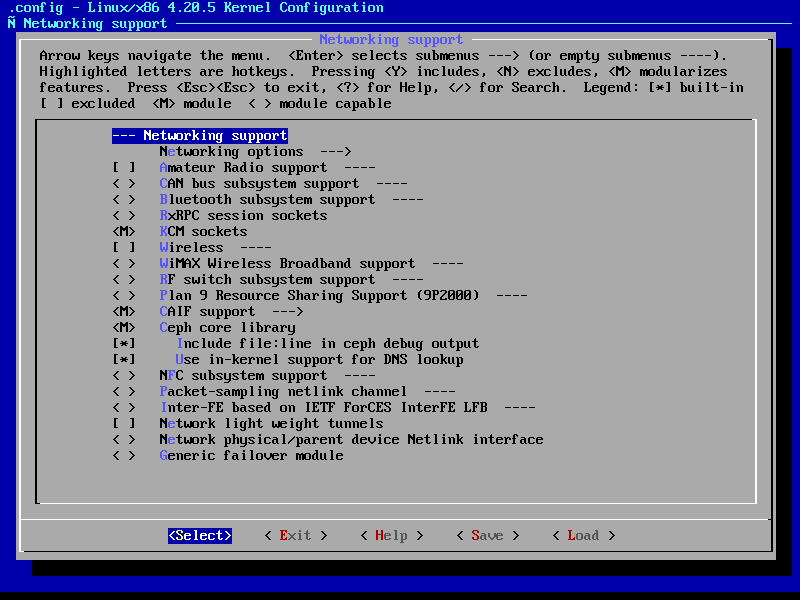
1. 64位系统



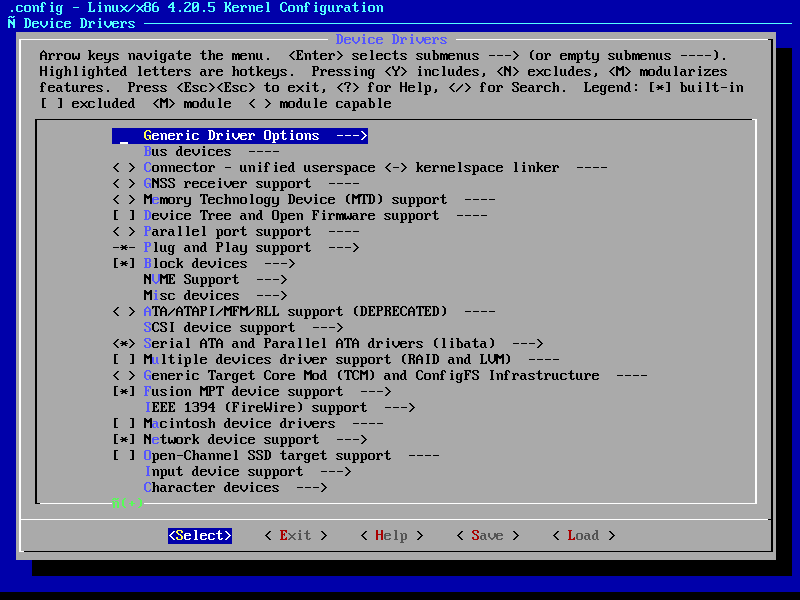
1. 通用配置



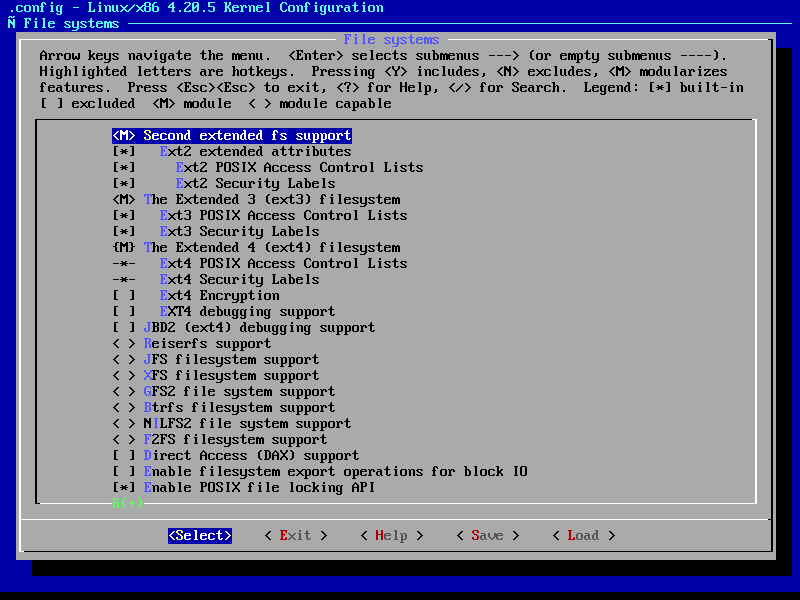
1. 网络相关配置



1. 驱动设备



1. 文件系统



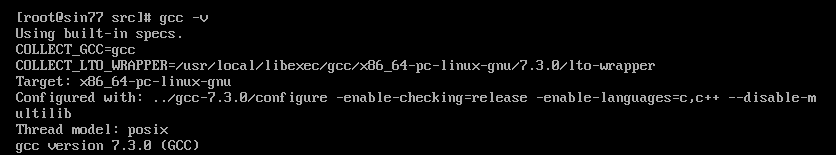
配置完成之后保存会自动生成.config文件，接下来进行编译

make -j4

在编译过程中遇到两个问题，一个是时钟问题，虚拟机的物理时间和系统时间存在偏差，导致编译无法展开；另外一个比较头疼的问题是gcc版本的问题，由于内核编译对gcc的要求是7.x以上，而centos直接通过yum得到的最新gcc的版本只停留在4.x上，所以需要手动编译gcc，编写脚本对gcc进行编译：



最后gcc升级成功：



完成编译之后执行：

make modules

之后进行模块安装：

make modules\_install

这步完成之后会在生成内核文件bzImage，并且在/lib/modules下可以看到新编译成功的模块驱动



将驱动拷贝到小系统的对应目录下，并且将生成的bzImage内核文件拷贝到boot下，再修改grub.conf文件，这样时候就可以启动新编译成功的内核。

* + 1. **U盘启动**

格式化U盘，输入命令： mkfs.vfat /dev/sdb1

挂载U盘： mount /dev/sdb1 /USB

安装grub到U盘上： grub-install --root-directory=/USB --no-floppy /dev/sdb

拷贝当前系统的grub.conf到U盘grub下： cp /boot/grub/grub.conf /USB/boot/grub/

删去其他系统启动的配置，并将其中所有的根目录/改为/boot。

拷贝/boot/grub/目录下的stage1、stage2到U盘的/boot/grub/目录下，拷贝生成的内核bzImage和initrd.img文件到U盘的/boot目录下。

然后安装grub到MBR：

[root@localhost /USB]# grub

grub > find /boot/grub/stage1

(hd0,0)

(hd1,0)

grub > root (hd0,0)

grub > setup (hd0)

grub > quit

就完成了grub的安装，可以实现在真机上启动U盘小系统。

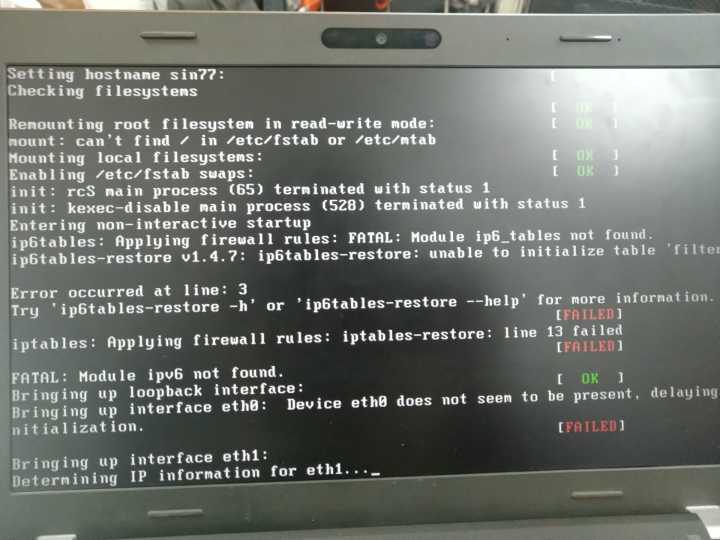
1. **实验结果**

bzImage大小：3.6M

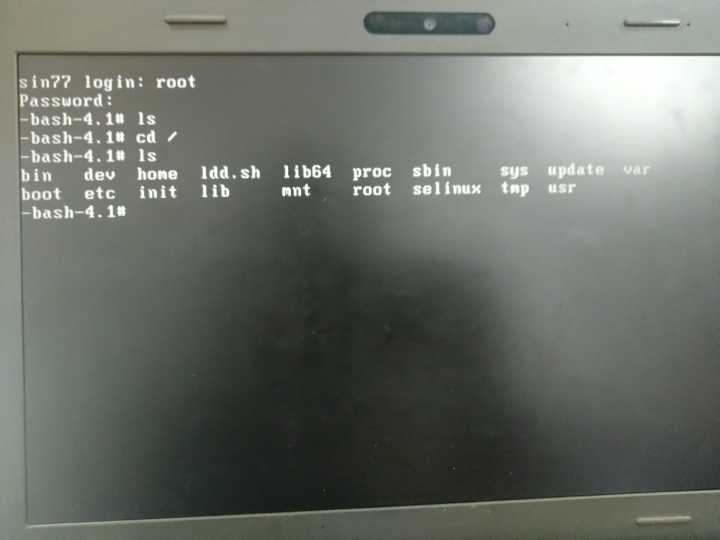
intrid.img大小：19M

系统启动：

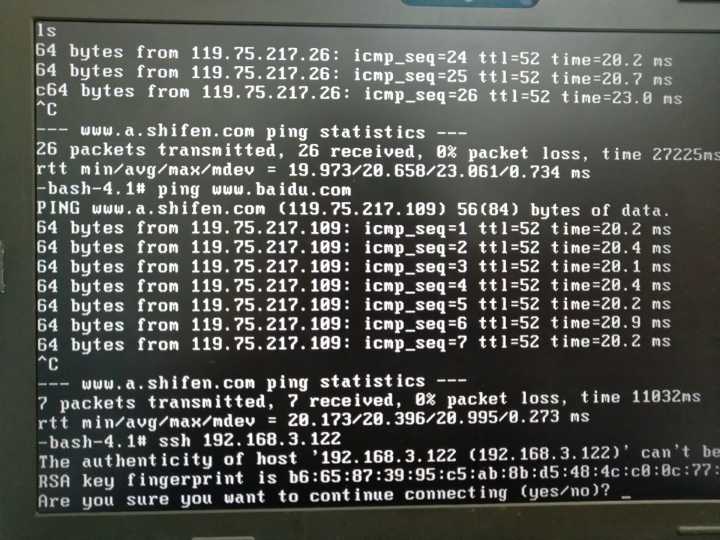




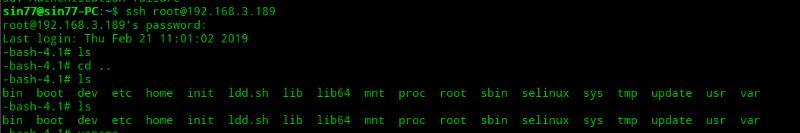
控制台登录：



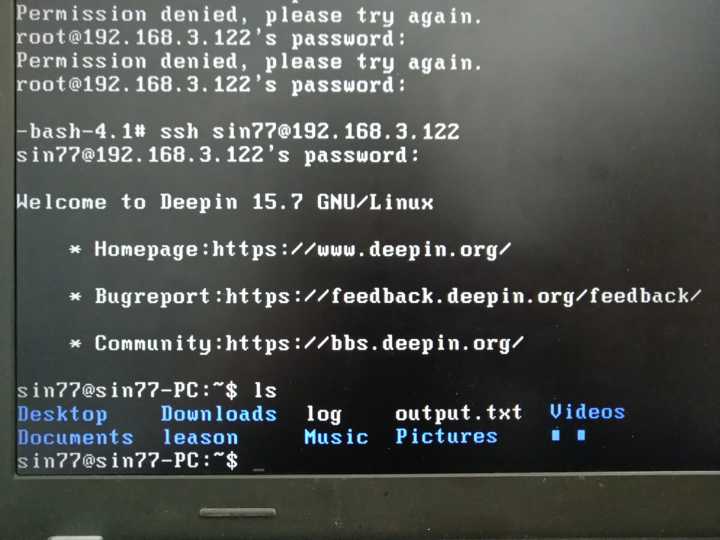
网络：



ssh登录：



ssh访问其他机器：



1. **分析总结**

通过这次实验，对liunx有进一步深刻的了解。我在大二学习操作系统课程的时候已经学习过编译linux内核，相比起之前的学习经历，这一次对Linux内核的定制，小系统的裁剪可以说是印象深刻。

深刻的原因一个是时间跨度很大，从9月份到寒假。其实9月份自己的完成度相比其他同学来说并不高，而当时做文件系统的裁剪时并没有找到方法，很多时候是一头雾水，盲目操作。寒假再次上手这个实验的时候仍然很排斥，因为我还是不知道从何下手，现在回想起来，我当时对整个实验结构并不了解，对如何达到每一步骤的要求没有清晰的认识，之后我重新梳理了liunx初步的课程，学习了一些liunx的基本知识，向同学请教实验的方法与心得，慢慢自己也有了一些门道，渐渐对实验有信心了。

令我印象深刻的第二个原因是实验过程中踩的坑实在是太多了。在裁剪文件系统时各种需要拷贝的文件可能只能通过strace来发现，网络配置问题，减小intrid.img文件的大小，内核配置的选择，内核编译过程中时钟错误以及gcc版本问题，U盘启动后网络问题等等，实验的每一步都不简单。

从这次实验中收获很大，不仅深入了解了linux系统，对裁剪系统有了初步的实践，而且提高了解决陌生问题的能力，从中学习到了方法和技巧。