

《Linux 初步》课程设计报告

姓 名：_____伍圣晖_____

学 号：_____U201513488_____

2018 年 09 月 09 日

目录

一. 实验任务及要求	3
二. 实验环境	3
三、实验结果.	4
四、实现过程.	6
4.1 v0.5 版本（简单文件系统）	6
4.2 v0.55 版本（挂载原系统）	7
4.3 v0.6 版本（使用 udev 管理设备）	7
4.4 v0.7 版本（具有 login 功能）	8
4.5 v0.9 版本（使用/sbin/init）	8
4.6 v1.0 版本	9
4.7 编译内核	11
五. 总结	11

一. 实验任务及要求

裁剪 Linux 系统，制作自己的小系统，并满足以下要求：

- 1、bzImage（内核文件） < 4MB
- 2、initrd.img（系统镜像） < 24MB，且系统登录所使用的用户密码（用户名：root 密码：seedclass）
- 3、系统启动后，自动设置本机 ip 为 192.168.15.15。可通过 console（主机界面）和 ssh 登录（ssh 服务器），
- 4、支持ssh到其他主机（ssh客户端）
- 5、可挂载U盘
- 6、可访问windows NTFS分区（ntfs-3g）
- 7、可以正常关机（包括关闭电源）以及重启

二. 实验环境

虚拟机：VMware® Workstation 14 Pro

原系统：CentOS release 6.5

三、实验结果.

内核文件 bzImage 大小 3.0M，文件系统大小 14M。文件系统中包括两套驱动程序：原系统内核编译得到的驱动和裁剪内核对应的内核程序。

```
[root@localhost v1.0]# ll -h /boot/
total 41M
-rwxrw-rw-. 1 root root 3.0M Sep 11 11:03 bzImage
-rw-r--r--. 1 root root 103K Nov 22 2013 config-2.6.32-431.el6.x86_64
drwxr-xr-x. 3 root root 1.0K Nov 30 2013 efi
-rw-r--r--. 1 root root 163K Jul 20 2011 elf-memtest86+-4.10
drwxr-xr-x. 2 root root 1.0K Sep 11 09:24 grub
-rw-----. 1 root root 17M Aug 4 06:57 initramfs-2.6.32-431.el6.x86_64.img
-rw-r--r--. 1 root root 15M Sep 11 12:37 initrd.img
drwx-----. 2 root root 12K Sep 9 2017 lost+found
-rw-r--r--. 1 root root 162K Jul 20 2011 memtest86+-4.10
-rw-r--r--. 1 root root 190K Nov 22 2013 symvers-2.6.32-431.el6.x86_64.gz
-rw-r--r--. 1 root root 2.5M Nov 22 2013 System.map-2.6.32-431.el6.x86_64
-rwxr-xr-x. 1 root root 4.0M Nov 22 2013 vmlinuz-2.6.32-431.el6.x86_64
[root@localhost v1.0]#
```

在虚拟机内启动，可以使用 root 用户登陆，密码 seedclass，校验 ip 可见 eth0 网卡对应 ip 为 192.168.15.15：

```
localhost login: root
Password:
[root@localhost ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:DC:49:92
          inet addr:192.168.15.15  Bcast:192.168.15.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fedc:4992/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1 errors:0 dropped:0 overruns:0 frame:0
          TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:243 (243.0 b)  TX bytes:720 (720.0 b)
          Interrupt:19 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:4 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:344 (344.0 b)  TX bytes:344 (344.0 b)

[root@localhost ~]# _
```

ssh 客户端功能：以使用 ssh 命令登陆虚拟机子网下的另一台 ubuntu 虚拟机。

```
RX packets:4 errors:0 dropped:0 overruns:0 frame:0
TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:344 (344.0 b) TX bytes:344 (344.0 b)

[root@localhost ~]# ssh root@192.168.15.14
The authenticity of host '192.168.15.14 (192.168.15.14)' can't be established.
RSA key fingerprint is 26:55:14:05:2f:b9:18:30:32:e5:39:12:d5:3d:b8:59.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.15.14' (RSA) to the list of known hosts.
root@192.168.15.14's password:
Welcome to Ubuntu 14.04.5 LTS (GNU/Linux 4.4.0-31-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

Last login: Mon Sep 10 06:10:44 2018 from 192.168.15.15
root@ubuntu:~#
```

ssh 服务器功能：在宿主机 window 内可以用 putty 连接 192.168.15.15，即小系统。同时这也证明小系统支持多终端登陆。

```
RX bytes:344 (344.0 b) TX bytes:344 (344.0 b)

[root@localhost ~]# ssh root@192.168.15.14
The authenticity of host '192.168.15.14 (192.168.15.14)' can't be established.
RSA key fingerprint is 26:55:14:05:2f:b9:18:30:32:e5:39:12:d5:3d:b8:59.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.15.14' (RSA) to the list of known hosts.
root@192.168.15.14's password:
Welcome to Ubuntu 14.04.5 LTS (GNU/Linux 4.4.0-31-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

Last login: Mon Sep 10 06:10:44 2018 from 192.168.15.15
root@ubuntu:~#
```

```
root@localhost:~
login as: root
root@192.168.15.15's password:
Last login: Tue Sep 11 05:41:34 2018
[root@localhost ~]# ls /
bin  dev  init  lib64  mnt  root  sys  usr
boot  etc  lib  libCopy  proc  sbin  tmp  var
[root@localhost ~]#
```

挂载 U 盘：可以挂载 ntfs 格式的 U 盘，挂载后可以查看内部信息。由于编码格式等关系，读取 U 盘时会有报错。同时，支持 ntfs 格式说明可以挂载 windows 系统。

```
255 heads, 63 sectors/track, 3824 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0xe145c182
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sdb1	*	2	3825	38711936	7	HPFS/NTFS

```
[root@localhost ~]# mount -t ntfs /dev/sdb
sdb sdb1
[root@localhost ~]# mount -t ntfs /dev/sdb1 /mnt
ntfs: driver 2.1.32 [Flags: R/O MODULE].
ntfs: volume version 3.1.
[root@localhost ~]# ls /mnt
ntfs: (device sdb1): ntfs_ucstonls(): Unicode name contains characters that cannot be converted to character set default. You might want to try to use the mount option nls=utf8.
ntfs: (device sdb1): ntfs_filldir(): Skipping unrepresentable inode 0x48.
ntfs: (device sdb1): ntfs_ucstonls(): Unicode name contains characters that cannot be converted to character set default. You might want to try to use the mount option nls=utf8.
ntfs: (device sdb1): ntfs_filldir(): Skipping unrepresentable inode 0x49.
System Volume Information centos65 ksfywu.pif ntji.exe wsh-UM1
autorun.inf jianli linux-4.17.19.tar.xz olyriq.pif
[root@localhost ~]#
```

可以正常关机、重启：无法用截图说明，小系统支持 reboot 和 poweroff 指令，均能达到所需效果。

四、实现过程.

4.1 v0.5 版本（简单文件系统）

利用 initrd.img 机制。在 init 脚本中使用 exec 函数，使系统重启后可以获得一个 shell 界面。

使用 ldd 命令查看 bash、ls 等指令需要的动态库，复制到小系统的相应文件夹下。

打包生成新的 initrd.img 并在 grub 启动配置文件（/boot/grub/grub.conf）中新增入口。重启选择新系统，可以看到 shell 界面。

4.2 v0.55 版本（挂载原系统）

要实现挂载原系统，需要系统能够识别原系统的 ext4 文件系统，并执行 mount 等指令。因此需要在 0.5 版本基础上增加新的命令和驱动程序。

关于识别原文件系统所需的驱动，可以通过 lsmod 和 modinfo 两条命令找到相关驱动和所在位置。将相关的驱动放到小系统对应的位置。

将原系统的 initrd.img 解压，参考 init 脚本的执行流程，修改小系统的 init 脚本，可以实现重启后挂载原文件系统。

4.3 v0.6 版本（使用 udev 管理设备）

想要启动 udev 服务，只需在 init 中添加 start_udev 或 udevd -daemon; udevadm trigger 指令即可。这一步成功的关键在于找到 udev 服务的依赖项。

udev 的规则文件在 /lib/udev/ 目录下，配置文件在 /etc/udev/ 目录下，同时还需要 /etc/nsswitch.conf 配置的名称服务交换，其依赖的库为 /lib 目录下以 libnss 开头的文件，复制上述文件至小系统。。

udev 指令运行时还需要很多其他在 /bin、/sbin 内的指令。缺乏这些命令会在重启时出错，可以通过查看系统启动时打印在屏幕上的 log 排查缺少的指令。还可以使用 strace 指令跟踪 udev 的执行过程。但 strace 的输出结果较长，不太容易寻找。

重启后，若前述过程没有错误，可以在 /dev 下面看到所有的硬件设备。插入 U 盘则会动态加载相关驱动。

4.4 v0.7 版本（具有 login 功能）

系统的登录功是由/bin/login 指令实现的。与之相关的是/etc/pam.d，对应 login 时认证的配置；/etc/shadow、/etc/passwd、/etc/group，对应系统的用户、组、哈希后的密码等数据。

在 init 脚本最后、exec /bin/bash 之前，增加 login 语句，即可在此版本开机后，先验证登陆、后获得 shell 界面。

4.5 v0.9 版本（使用/sbin/init）

在经历过上述几个版本的开发后，小系统已经基本具备了启动所需的大部分驱动、程序和动态库。因此，之前在 init 中做的工作可以直接交由/sbin/init 完成。即在 init 脚本中，可以直接 exec /sbin/init。此时，/sbin/init 是以 pid=1 的情况启动的。

/sbin/init 的工作流程涉及到这样几个方面：udev 的启动，各项服务 service 的启动，和 mingetty+/bin/login 的启动。udev 依赖于/etc/rc.sysinit，各项服务的启动顺序和配置在/etc/rc*.d、/etc/init 中，登陆所需的 login 在上一版中已经实现了，因此只需复制 mingetty 相关的命令和动态库。

此外，与启动相关的很多配置文件此时也同样需要放入/etc 中。这其中包括配置默认启动级别的/etc/inittab，记录系统版本信息的/etc/system-release，配置 bash 界面的/etc/bashrc，等等。

完成了上述任务之后，小系统就基本完成了，此时的系统全部运行在内存中，

启动顺序与原系统类似。剩余工作是完善相关功能，如网络相关功能。

4.6 v1.0 版本

最终版本的主要任务是完善网络相关任务，使小系统在开机时能够适配好网卡等配置，并上网。同时允许 ssh 等方式连接小系统。因此主要工作围绕/etc 中 network 相关的文件展开。

安装相关的网卡驱动；安装常用网络命令如 ping 和 ifconfig 及网络服务启动时需要的相关命令；将/etc/init.d/network, /etc/sysconfig/network 等文件复制进小系统；最后编辑好相关配置文件，即可在小系统内 ping 通 127.0.0.1 和本机 ip。

这里遇到过一个问题，NetworkManager 服务会与 network 服务冲突，导致 /etc/init.d/network 服务无法开机启动，开机时只有 lo 网卡，而没有 eth0 网卡。删去 NetworkManager 相关的文件，并手动执行/etc/init.d/network 即可开启 eth0。因此在 rc.d/rc.local 中增加了这一条指令，使之开机启动。

在/etc/sysconfig/network-scripts/ifcfg-eth 中固定 IPADDR 等参数，即可实现在开机时固定 ip 为 192.168.15.15。

另一个任务是 ssh 的服务端和客户端。有了网络服务后，宿主机和虚拟机之间可以 ping 通，但不能使用 ssh。ssh 功能包括是 ssh 命令和 sshd 命令及相关服务与配置文件，分散在/etc/、/var/、/usr/sbin 文件夹下。在 sshd 服务和 ssh 命令执行无误后，发现试图登陆时发现 permission denied。尝试修改 sshd_cnfig 文

件无效。仔细检查登陆逻辑，发现问题出在 pam.d 系统中。实现 login 功能时，简化了 pam.d 的校验逻辑，只依赖于 /lib64/security/pam_unix.so。仿照 login 重写 sshd 文件，ssh 功能可以正常实现。

```
root@localhost:/home/ws
File Edit View Search Terminal Help
#%PAM-1.0
#auth [user_unknown=ignore success=ok ignore=ignore default=bad] pam_securetty.so
auth      required      /lib64/security/pam_unix.so
account   required      /lib64/security/pam_unix.so
password  required      /lib64/security/pam_unix.so
session   required      /lib64/security/pam_unix.so
```

/etc/pam.d/sshd 文件

挂载 U 盘、识别 NTFS 分区：U 盘相关驱动有 usb-storage, fat-32 分区对应驱动有 vfat、fat, NTFS 驱动有 ntfs, 复制相关驱动并在新系统中重新 depmod 即可。

至此，v1.0 文件系统完成。各文件夹下的文件内容如下：

```
[root@localhost v1.0]# pwd
/home/wsh/Documents/v1.0
[root@localhost v1.0]# ls
bin boot etc init lib lib64 libCopy mnt proc root sbin sys usr var
[root@localhost v1.0]# ls bin
awk bash chgrp chown cut dmesg fgrep grep ipcalc ln ls mknod mount ping ps sed sleep touch vi
basename cat chmod cp df egrep find hostname kill login mkdir mktemp mv plymouth rm sh sort umount
[root@localhost v1.0]# ls etc
bashrc group init mtab pam.d profile.d rc1.d rc4.d rc.d readahead ssh udev
fcoe host.conf init.d networks passwd rc rc2.d rc5.d rc.local readahead.conf sysconfig
fstab hosts inittab nsswitch.conf profile rc0.d rc3.d rc6.d rc.sysinit shadow system-release
[root@localhost v1.0]# ls lib
modules terminfo udev
[root@localhost v1.0]# ls sbin
arping depmod fsck.ext4 ifconfig init ip lsmod modprobe readahead runlevel swapon telinit
chkconfig fdisk fstab-decode ifdown initctl ip6tables lspci pidof readahead-collector shutdown switch_root udevadm
consoletype fsck halt ifup insmod iptables mingetty poweroff reboot start_udev sysctl udevd
[root@localhost v1.0]# ls usr/bin
clear file id ldd ssh
[root@localhost v1.0]# ls usr/sbin
fcoeadm fcoemon NetworkManager pcsd sshd update-reader.conf
[root@localhost v1.0]# ls lib64
ld-linux-x86-64.so.2 libdbus-1.so.3 libgobject-2.0.so.0 libm.so.6 libnss_dns-2.12.so libnss_winbind.so.2 libproc-3.2.8.so libuuid.so.1
libacl.so.1 libdl.so.2 libgssapi_krb5.so.2 libncurses.so.5 libnss_dns.so.2 libnss_wins.so.2 libpthread.so.0 libwrap.so.0
libattr.so.1 libe2p.so.2 libgthread-2.0.so.0 libnih-dbus.so.1 libnss_files-2.12.so libpam_misc.so.0 libresolv-2.12.so libxtables.so.4
libaudit.so.1 libext2fs.so.2 libidn.so.11 libnih.so.1 libnss_files.so.2 libpam.so.0 librt.so.1 libresolv.so.2 libz.so.1
libauparse.so.0 libfipscheck.so.1 libip4tc.so.0 libnl.so.1 libnss_hesiod-2.12.so libpci.so.3 librt.so.1 libselinux.so.1 libz.so.1
libblkid.so.1 libfreebl3.so libip6tc.so.0 libnsl-2.12.so libnss_hesiod.so.2 libpci.so.3 librt.so.1 libselinux.so.1 libz.so.1
libcap.so.2 libgcc_s.so.1 libk5crypto.so.3 libnsl.so.1 libnss_nis-2.12.so libplc4.so librt.so.1 libselinux.so.1 libz.so.1
libcom_err.so.2 libgio-2.0.so.0 libkeyutils.so.1 libnspr4.so libnss_nisplus-2.12.so libplds4.so libtinfo.so.5
libcrypt.so.1 libglib-2.0.so.0 libkrb5.so.3 libnss_compat-2.12.so libnss_nisplus.so.2 libply.so.2 libudev.so.0
libc.so.6 libgmodule-2.0.so.0 libkrb5support.so.0 libnss_compat.so.2 libnss_nis.so.2 libpopt.so.0 libutil.so.1
```

4.7 编译内核

使用内核版本 4.13.4 定制内核（此版本内核是课上下载的，最新版的 4.17.x 在虚拟机上编译时会提示报错，因此退而求其次）。参照已有文件系统中存在的驱动和虚拟机硬件配置，在 menuconfig 界面选择所必须的模块和驱动，执行编译。即可获得定制内核 bzImage。执行 make modules_install 命令安装驱动，成功后即可以在 /lib/modules/\${kernel_version} 下找到新内核对应的驱动程序*.ko。将之复制到 v1.0 中的 /libmodules 中。

更改 /boot/grub/grub.conf，替换 kernel 和 initrd 为自己定制的部分，即可启动自己的新系统。

编译出来的驱动程序有一个问题，其中不包含虚拟机网卡的 pcnet32 驱动程序。因此修改虚拟机的配置文件，指定网卡类型为 e1000，即可正常联网。

五. 总结

本课程设计实现了对 Linux 文件系统和内核的裁剪与定制化，深入理解 Linux，逐步搭建起自己可用的小系统。通过课上的任务分解，原本无从下手的任务有了可行性，一步步迭代中，对 Linux 系统的理解也从模糊到清晰，使用从生涩到熟练。

这次课设拖了很久，最后终于有了阶段性的成果，此后也可以假装说自己是研究过 Linux 系统内核的技术人员了。虽然最后的成果相比于真正的一个好用的、顺手的迷你小系统还有很长一段路，虽然有不少功能点是在自己不经意间“撞”出

来的，但是这个过程上的攻坚克难的经验、不断试错之后顿悟的收获、一点点突破自己的成就感，值得久久回味。