

浙江大学实验报告

课程名称: 操作系统分析及实验 实验类型: 综合型/设计性

实验项目名称: 实验 2 重建 linux 内核

学生姓名: 胡亮泽 专业: 计算机科学与技术 学号: 3120102116

电子邮件地址 (必须): huliangze@yeah.net 手机: 18868103152

实验日期: 2014 年 12 月 12 日

一、实验目的

学习重建 Linux 内核, 掌握 Linux 内核和发行版本的区别。

二、实验内容

在 linux 操作系统环境下重建内核。实验主要内容:

- 查找并且下载一份内核源代码
- 配置内核
- 编译内核和模块
- 配置启动文件

三、主要仪器设备 (必填)

VAIO E 系列, Ubuntu 14.04,

3.13.0-40-generic #69-Ubuntu SMP Thu Nov 13 17:56:26 UTC 2014 i686 i686 i686

GNU/Linux

虚拟机软件为 VMware Play 6.0 Windows, 主机系统 Windows 8

四、操作方法和实验步骤

1. 前往官网下载内核源代码, 本次实验下载的源代码为 3.17.6 版本
2. 解压缩内核文件到根目录 `~/linux`
3. 用 `cd` 命令进入目录 `~/linux`, 并使用命令 `make mrproper` 清除之前在该目录下面的配置文件和生成的核心.o 文件

```
lk@wholanz:~/linux-3.17.6$ cd ~/linux
lk@wholanz:~/linux$ make mrproper
CLEAN    scripts/basic
CLEAN    scripts/kconfig
CLEAN    .config
```

4. 为了与正在运行的操作系统内核的运行环境匹配，可以先把当前已配置好的文件复制到当前目录下，新的文件名为.config 文件：

```
cp /boot/config-`uname -r` .config
```

5. 使用 make -j2 命令，使用两个进程编译内核。该过程需要等待大约两个小时的时间。

```
CHK      include/generated/uapi/linux/version.h
UPD      include/generated/uapi/linux/version.h
SYSHDR   arch/x86/syscalls/../include/generated/uapi/asm/unistd_32.h
SYSHDR   arch/x86/syscalls/../include/generated/uapi/asm/unistd_64.h
SYSHDR   arch/x86/syscalls/../include/generated/uapi/asm/unistd_x32.h
CHK      include/generated/utsrelease.h
UPD      include/generated/utsrelease.h
HOSTCC   scripts/genksyms/genksyms.o
SHIPPED  scripts/genksyms/lex.lex.c
SHIPPED  scripts/genksyms/keywords.hash.c
SHIPPED  scripts/genksyms/parse.tab.h
SHIPPED  scripts/genksyms/parse.tab.c
HOSTCC   scripts/genksyms/lex.lex.o
^[[A HOSTCC scripts/genksyms/parse.tab.o
HOSTLD   scripts/genksyms/genksyms
CC       scripts/mod/empty.o
HOSTCC   scripts/mod/mk_elfconfig
CC       scripts/mod/devicetable-offsets.s
MKELF    scripts/mod/elfconfig.h
GEN      scripts/mod/devicetable-offsets.h
HOSTCC   scripts/mod/modpost.o
HOSTCC   scripts/mod/sumversion.o
HOSTCC   scripts/mod/file2alias.o
```

6. 使用 make modules 命令编译内核模块，并使用 make modules_install 命令安装内核模块。

```
lk@wholanz:~/linux$ make modules
CHK      include/config/kernel.release
CHK      include/generated/uapi/linux/version.h
CHK      include/generated/utsrelease.h
CALL     scripts/checksyscalls.sh
Building modules, stage 2.
MODPOST 4017 modules
WARNING: modpost: Found 5 section mismatch(es).
To see full details build your kernel with:
'make CONFIG_DEBUG_SECTION_MISMATCH=y'
```

```
make CONFIG_DEBUG_SECTION_MISMATCH=y
lk@wholanz:~/linux$ sudo make modules_install
[sudo] password for lk:
INSTALL arch/x86/crypto/aes-i586.ko
INSTALL arch/x86/crypto/aesni-intel.ko
INSTALL arch/x86/crypto/crc32-pclmul.ko
INSTALL arch/x86/crypto/glue_helper.ko
INSTALL arch/x86/crypto/salsa20-i586.ko
INSTALL arch/x86/crypto/serpent-sse2-i586.ko
INSTALL arch/x86/crypto/twofish-i586.ko
INSTALL arch/x86/kernel/apm.ko
INSTALL arch/x86/kernel/cpu/mcheck/mce-inject.ko
INSTALL arch/x86/kernel/cpuid.ko
INSTALL arch/x86/kernel/iosf_mbi.ko
INSTALL arch/x86/kernel/msr.ko
INSTALL arch/x86/kvm/kvm-amd.ko
INSTALL arch/x86/kvm/kvm-intel.ko
INSTALL arch/x86/kvm/kvm.ko
INSTALL arch/x86/oprofile/oprofile.ko
```

```
INSTALL drivers/ata/pata_cs5520.ko
INSTALL drivers/ata/pata_cs5530.ko
INSTALL drivers/ata/pata_cs5535.ko
INSTALL drivers/ata/pata_cs5536.ko
INSTALL drivers/ata/pata_cypress.ko
INSTALL drivers/ata/pata_efar.ko
INSTALL drivers/ata/pata_hpt366.ko
INSTALL drivers/ata/pata_hpt37x.ko
INSTALL drivers/ata/pata_hpt3x2n.ko
INSTALL drivers/ata/pata_hpt3x3.ko
INSTALL drivers/ata/pata_isapnp.ko
INSTALL drivers/ata/pata_it8213.ko
INSTALL drivers/ata/pata_it821x.ko
INSTALL drivers/ata/pata_jmicron.ko
INSTALL drivers/ata/pata_legacy.ko
INSTALL drivers/ata/pata_marvell.ko
INSTALL drivers/ata/pata_mpiix.ko
INSTALL drivers/ata/pata_netcell.ko
INSTALL drivers/ata/pata_ninja32.ko
INSTALL drivers/ata/pata_ns87410.ko
INSTALL drivers/ata/pata_ns87415.ko
INSTALL drivers/ata/pata_oldpiix.ko
INSTALL drivers/ata/pata_opti.ko
```

该过程也需要等待一段时间才能完成

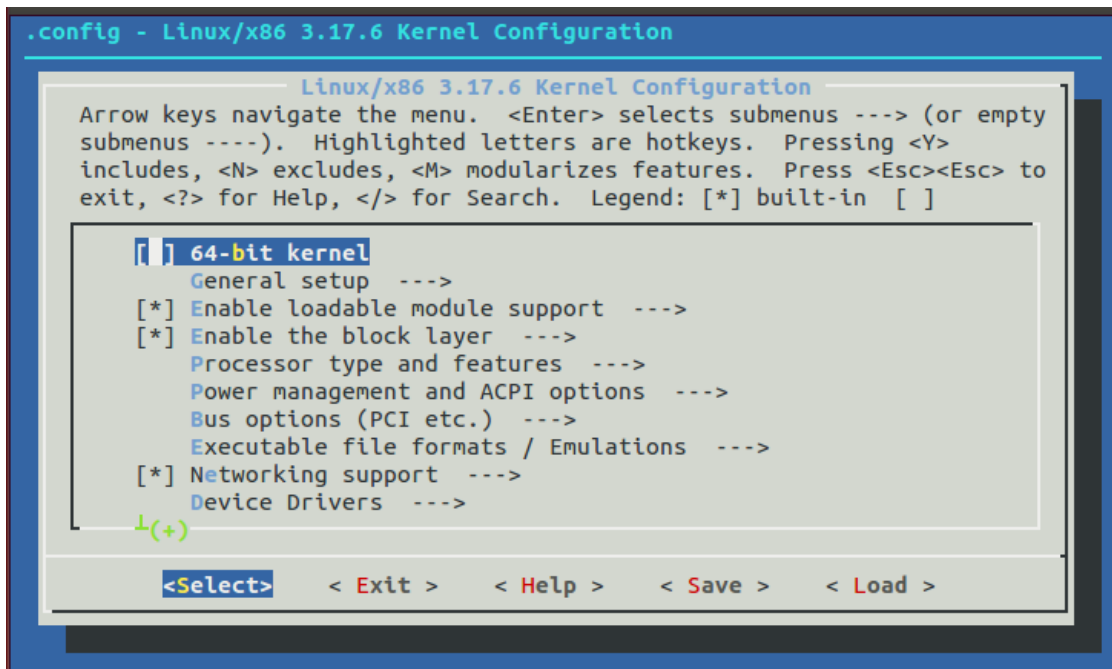
7. 使用 `sudo make install` 命令将自己的内核安装到 `/boot` 目录下，以便下次启动的时候使用自己的内核

```
lk@wholanz:~/linux$ sudo make install
[sudo] password for lk:
sh ./arch/x86/boot/install.sh 3.17.6 arch/x86/boot/bzImage \
    System.map "/boot"
run-parts: executing /etc/kernel/postinst.d/apt-auto-removal 3.17.6 /boot/vmlinuz-3.17.6
run-parts: executing /etc/kernel/postinst.d/initramfs-tools 3.17.6 /boot/vmlinuz-3.17.6
update-initramfs: Generating /boot/initrd.img-3.17.6
run-parts: executing /etc/kernel/postinst.d/pm-utils 3.17.6 /boot/vmlinuz-3.17.6
run-parts: executing /etc/kernel/postinst.d/update-notifier 3.17.6 /boot/vmlinuz-3.17.6
run-parts: executing /etc/kernel/postinst.d/zz-update-grub 3.17.6 /boot/vmlinuz-3.17.6
Generating grub configuration file ...
Warning: Setting GRUB_TIMEOUT to a non-zero value when GRUB_HIDDEN_TIMEOUT is set is no longer supported.
Found linux image: /boot/vmlinuz-3.17.6
Found initrd image: /boot/initrd.img-3.17.6
Found linux image: /boot/vmlinuz-3.17.6.old
Found initrd image: /boot/initrd.img-3.17.6
Found linux image: /boot/vmlinuz-3.13.0-40-generic
Found initrd image: /boot/initrd.img-3.13.0-40-generic
```

8. 重启系统，发现可以选择不同版本的内核进入操作系统

另外，可以用 `make menuconfig` 命令配置内核，但是本次实验都是采用默认的配置，不需要使用这个命令

```
lk@wholanz:~/linux$ make menuconfig
HOSTCC scripts/basic/fixdep
HOSTCC scripts/kconfig/conf.o
HOSTCC scripts/kconfig/lxdialog/checklist.o
HOSTCC scripts/kconfig/lxdialog/inputbox.o
HOSTCC scripts/kconfig/lxdialog/menubox.o
HOSTCC scripts/kconfig/lxdialog/textbox.o
HOSTCC scripts/kconfig/lxdialog/util.o
HOSTCC scripts/kconfig/lxdialog/yesno.o
HOSTCC scripts/kconfig/mconf.o
SHIPPED scripts/kconfig/zconf.tab.c
SHIPPED scripts/kconfig/zconf.lex.c
SHIPPED scripts/kconfig/zconf.hash.c
HOSTCC scripts/kconfig/zconf.tab.o
HOSTLD scripts/kconfig/mconf
scripts/kconfig/mconf Kconfig
.config:6443:warning: symbol value 'm' invalid for USB_DWC2
.config:6730:warning: symbol value 'm' invalid for VME_BUS
.config:6763:warning: symbol value 'm' invalid for GENERIC_PHY
```



五、实验结果和分析

使用 `uname -a` 命令查看内核版本号，原先的版本号为 3.13.0-40

```
lk@wholanz:~/linux$ uname -a
Linux Wholanz 3.13.0-40-generic #69-Ubuntu SMP Thu Nov 13 17:56:26 UTC 2014 i686
i686 i686 GNU/Linux
```

重启进入系统后的内核版本号为 3.17.6

```
lk@wholanz:~$ uname -a
Linux Wholanz 3.17.6 #1 SMP Fri Dec 26 02:33:01 CST 2014 i686 i686 i686 GNU/Linu
x
lk@wholanz:~$
```

另外，实验过程中曾经出现如下问题：

```

BusyBox v1.21.1 (Ubuntu 1:1.21.0-1ubuntu1) built-in shell (ash)
Enter 'help' for a list of built-in commands.

(initramfs)
(initramfs)
(initramfs)
(initramfs) help
Built-in commands:
-----
. : [ alias break cd chdir command continue echo eval exec exit
export false getopt hash help let local printf pwd read readonly
return set shift test times trap true type ulimit umask unalias
unset wait [ [[ ash awk basename blockdev cat chmod chroot chvt
clear cmp cp cut deallocvt df dnsdomainname du dumpkmap echo
egrep env expr false fbset fdflush fgrep find grep gunzip gzip
hostname ifconfig ip kill ln loadfont loadkmap ls mkdir mkfifo
mknod mkswap mktmp modinfo more mount mv openvt pidof printf
ps pwd readlink reset rm rmdir sed seq setkeycodes sh sleep sort
stat static-sh stty switch_root sync tail tee test touch tr true
tty umount uname uniq wc wget which yes zcat

(initramfs) ls
dev      bin      run      scripts  conf      sys      tmp
root     init     etc      sbin     lib       proc     var
(initramfs) _

```

原因是驱动没有正常安装，忘记在编译内核模块后使用 `make modules_install` 命令安装内核模块，进入操作系统，重新使用该命令安装相应的内核模块后，重启成功进入相应版本内核的操作系统

六、思考题

1. Linux 内核目录下有一个 `.config` 文件，请说明这个文件的作用？

内核配置参数都保存在这里。在执行 `make xconfig` 或者 `make menuconfig` 之后，修改过之后的配置就保存在这个 `.config` 文件里了。所有 `Makefile` 文件都是根据 `.config` 来确定使用那些文件的

2. 在 Linux 内核代码树中，很多子目录有 `Makefile` 文件和 `Kconfig` 文件，请分别解释这两个文件的作用？

Makefile 定义了内核的编译规则。

linux 内核中所有配置工具都是通过读取 `arch/arm/Kconfig` 文件来生成配置界面的，这个文件是所有配置文件的入口，它会包含其他目录的 `Kconfig` 文件进来。我们内核中每个子目录中都会有一个 `Makefile` 和 `Kconfig` 文件，每个目录下的 `Kconfig` 文件是配置界面的源文件。

3. 浏览 `/boot` 目录，你一定发现了 `System.map-3.17.1` 文件，以及 `initrd.img-3.17.1` 文件。如果打开 `/boot/grub` 目录下面的 `grub.cfg` 文件，不难发现命令：

`initrd /boot/ initrd.img-3.17.1`

这两个文件分别起什么作用？你能否设计一个实验来验证你的判断？

System.map 是内核的符号表，通过它可以知道所有符号和其地址的对应关系。

Initrd 是一种启动 **Linux** 系统的方式

当前流行的 **Linux** 版本一般都采用模块化的内核,这种方式可以在不重新编译构建内核的情形下增加功能模块

但是如果 **Linux** 的 **root** 文件系统所在设备的驱动是一个模块(没有编译进内核映像),就不能被引导程序(例如 **loadlin**)直接加载，这时会用到 **initrd** 方式来启动 **Linux** 系统

实验设计：

用#将该行注释，观察是否能够正常启动

七、讨论、心得

- 1.了解了如何简单重建 linux 内核的方法步骤。
2. 了解了重建内核过程中某些命令的使用方法，以及某些文件如 **Makefile**, **Kconfig**, **.config** 的作用。
- 3.学会了应对某些可能的编译内核过程中出现的错误，比如忘记安装相应的驱动内核模块