苏州大学实验报告

院、系	计算机学院	年级专业	20计科	姓名	柯骅	学号	2027405033
课程名称	操作系统课程实践					成绩	
指导教师	李培峰	同组实验者	无		实验日期	2023.03.30	

实 验 名 称 实验5 Linux内核编译

一、 实验目的

1. 学习重新编译 Linux 内核的方法。
2. 理解 Linux 标准内核和发行版本内核的区别。

二、 实验内容

在 Linux 系统中下载同一发行版本的版本号较高的内核，编译之后运行自己编译的内核，并使用 uname -r 命令查看是否运行成功。

由于不同版本的内核在编译过程中可能出现不同的问题，本书推荐的内核版本为 4.16.10。从第 7 章开始的进阶实验篇，都可以选用该版本的内核。

三、 实验步骤

针对 4.16.10 内核版本，推荐编译空间为 30GB。

(1) 查看内核版本

```
# uname -r
```

若以上命令执行后现实的结果为 4.16.10-generitic，则说明此时内核版本为 4.16.10。

(2) 下载内核

既可以通过 Linux 官方网站下载内核，也可以通过国内的某些网站进行下载。

推荐内核版本：linux-4.16.10。下载后的压缩包：linux-4.16.10.tar.gz。

(3) 解压

将压缩包解压到/usr/src 目录下，使用如下命令：

```
# tar xf linux-4.16.10.tar.gz -C /usr/src
```

解压完成后跳转至/usr/src 目录，利用 ls 命令查看是否解压成功。

(4) 配置内核

```
# cd linux-4.16.10
# make menuconfig
```

执行以上命令进入解压后的内核版本目录，编译后，便可以图形化配置哪些功能需要直接编译进内核，哪些功能需要编译成模块，

哪些功能不需要编译。随后保存对应的配置文件。

对于每个配置选项，可以通过<Select>来对其进行选择：<*>或[*]表示将该功能编译进内核：[ ]表示不将该功能编译进内核；[M]表示将该功能编译成可以在需要时动态插入内核的模块。

说明：从此处开始，如果出现缺包错误，那么需要按照错误提示安装所需要的包。例如，ncurses 库可使用命令 apt-get install ncurses-devel 安装所需要的包

(5) 编译内核

```
# make -jn
```

利用 make 命令开始编译内核。为了提高编译速度，可以使用-j 选项进行多线程处理。在一台双核的机器上，可以使用 make -j4 使 make 最多允许 4 条编译命令同时执行，从而更有效地利用 CPU 资源。在一台 4 核的机器上，可以使用 make -j8 使 make 最多允许 8 条编译命令同时执行。

同样，如果出现缺包错误，那么需要安装所缺的包，比如 openssl（apt-get install openssl 和 libssl-dev（apt-get install libssl-dev）。

(6) 编译和安装模块

```
# make modules
# make modules_install
```

同样，可以使用-j 选项进行多线程处理。例如：

```
# make modules -j8
```

(7) 安装内核

```
# make install
```

(8) 重新启动，检查新内核

```
# reboot
```

重启 Linux 以开启新的内核。注意：有可能出现短暂死机的情况，可以多等待一些时间。

使用以下命令可以再次查看内核版本，并检查内核是否安装成功。

```
# uname -r
```

四、 实验结果

(1) 查看内核版本

执行 uname -r 命令后，结果为 5.4.0-144-generic：

```
kh@ubuntu:~$ uname -r
5.4.0-144-generic
```

内核版本为：5.4.0

(2) 下载内核

使用如下命令从镜像网站下载 linux-4.16.10.tar.gz：

```
# wget https://mirrors.edge.kernel.org/pub/linux/kernel/v4.x/linux-4.16.10.tar.gz
```

下载完成后：

```
kh@ubuntu:~$ wget https://mirrors.edge.kernel.org/pub/linux/kernel/v4.x/linux-4.16.10.tar.gz
--2023-04-02 22:46:04--  https://mirrors.edge.kernel.org/pub/linux/kernel/v4.x/linux-4.16.10.tar.gz
正在解析主机 mirrors.edge.kernel.org (mirrors.edge.kernel.org)... 147.75.80.249, 2604:1380:4601:e00::3
正在连接 mirrors.edge.kernel.org (mirrors.edge.kernel.org)|147.75.80.249|:443... 已连接。
已发出 HTTP 请求，正在等待回应... 200 OK
长度： 158943712 (152M) [application/x-gzip]
正在保存至：“linux-4.16.10.tar.gz”

linux-4.16.10.tar.gz 100%[===================>] 151.58M  2.27MB/s    用时 65s

2023-04-02 22:47:11 (2.34 MB/s) - 已保存 “linux-4.16.10.tar.gz” [158943712/158943712])
```

(3) 解压

切换到 root 管理员后，使用如下命令解压：

```
# tar -xzf linux-4.16.10.tar.gz -C /usr/src
```

完成后，使用 cd 命令跳转到/usr/src 目录，使用 ls 命令查看是否解压成功：

```
kh@ubuntu:/$ sudo -i
root@ubuntu:~# cd ../home/kh
root@ubuntu:/home/kh# tar -xzf linux-4.16.10.tar.gz -C /usr/src
root@ubuntu:/home/kh# cd ../../usr/src
root@ubuntu:/usr/src# ls
linux-4.16.10                      linux-hwe-5.4-headers-5.4.0-139
linux-headers-5.4.0-139-generic    linux-hwe-5.4-headers-5.4.0-144
linux-headers-5.4.0-144-generic    linux-hwe-5.4-headers-5.4.0-84
```

其中包含 linux-4.16.10 文件，说明解压成功

(4) 配置内核

使用如下命令：

```
# cd linux-4.16.10
# make menuconfig
```

发生错误：缺少 ncurses 库

```
root@ubuntu:/usr/src# cd linux-4.16.10
root@ubuntu:/usr/src/linux-4.16.10# make menuconfig
  HOSTCC  scripts/basic/fixdep
 *** Unable to find the ncurses libraries or the
 *** required header files.
 *** 'make menuconfig' requires the ncurses libraries.
 ***
 *** Install ncurses (ncurses-devel or libncurses-dev
 *** depending on your distribution) and try again.
 ***
scripts/kconfig/Makefile:206: recipe for target 'scripts/kconfig/dochecklxdialog
' failed
make[1]: *** [scripts/kconfig/dochecklxdialog] Error 1
Makefile:514: recipe for target 'menuconfig' failed
make: *** [menuconfig] Error 2
root@ubuntu:/usr/src/linux-4.16.10#
```

使用如下命令安装 ncurses 库：

```
# apt-get install ncurses-devel
# apt install libncurses-dev -y
```

```
root@ubuntu:/usr/src/linux-4.16.10# apt-get install ncurses-devel
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
E: 无法定位软件包 ncurses-devel
root@ubuntu:/usr/src/linux-4.16.10# apt install libncurses-dev -y
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
注意，选中 'libncurses5-dev' 而非 'libncurses-dev'
下列软件包是自动安装的并且现在不需要了：
  fonts-liberation2 fonts-opensymbol gir1.2-gst-plugins-base-1.0 gir1.2-gstreamer-1.0 gir1.2-gudev-1.0 gir1.2-udisks-2.0 grilo-plugins-0.3-base gstreamer1.0-gtk3 libboost-date-time1.65.1
  libboost-filesystem1.65.1 libboost-iostreams1.65.1 libboost-locale1.65.1 libcdr-0.1-1 libclucene-contribs1v5 libclucene-core1v5 libcmis-0.5-5v5 libcoland2 libdazzle-1.0-0 libe-book-0.1-1
  libedataserverui-1.2-2 libeot0 libepubgen-0.1-1 libetonyek-0.1-1 libevent-2.1-6 libexiv2-14 libfreerdp-client2-2 libfreerdp2-2 libgc1c2 libgee-0.8-2 libgexiv2-2 libgom-1.0-0 libgpgmepp6 libgpod-common
  libgpod4 liblangtag-common liblangtag1 liblirc-client0 liblua5.3-0 libmediaart-2.0-0 libmspub-0.1-1 libodfgen-0.1-1 libqqwing2v5 libraw16 librevenge-0.0-0 libsgutils2-2 libssh-4 libsuitesparseconfig5
  libvncclient1 libwinpr2-2 libxaplan30 libxmlsec1-nss linux-hwe-5.4-headers-5.4.0-84 lp-solve media-player-info python3-mako python3-markupsafe syslinux syslinux-common syslinux-legacy
  usb-creator-common
使用'apt autoremove'来卸载它(它们)。
将会同时安装下列软件：
  libtinfo-dev
建议安装：
  ncurses-doc
下列【新】软件包将被安装：
  libncurses5-dev libtinfo-dev
升级了 0 个软件包，新安装了 2 个软件包，要卸载 0 个软件包，有 69 个软件包未被升级。
需要下载 256 kB 的归档。
解压缩后会消耗 1,422 kB 的额外空间。
获取:1 http://us.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libtinfo-dev amd64 6.1-1ubuntu1.18.04 [81.3 kB]
获取:2 http://us.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libncurses5-dev amd64 6.1-1ubuntu1.18.04 [174 kB]
已下载 256 kB，耗时 2秒 (122 kB/s)
正在选中未选择的软件包 libtinfo-dev:amd64。
(正在读取数据库 ... 系统当前共安装有 179583 个文件和目录。)
正准备解包 .../libtinfo-dev_6.1-1ubuntu1.18.04_amd64.deb  ...
正在解包 libtinfo-dev:amd64 (6.1-1ubuntu1.18.04) ...
正在选中未选择的软件包 libncurses5-dev:amd64。
正准备解包 .../libncurses5-dev_6.1-1ubuntu1.18.04_amd64.deb  ...
正在解包 libncurses5-dev:amd64 (6.1-1ubuntu1.18.04) ...
正在设置 libtinfo-dev:amd64 (6.1-1ubuntu1.18.04) ...
正在设置 libncurses5-dev:amd64 (6.1-1ubuntu1.18.04) ...
正在处理用于 man-db (2.8.3-2ubuntu0.1) 的触发器 ...
```

再次执行 make menuconfig，出现错误：缺少 bison，使用如下命令安装：

```
# apt install bison -y
```

```
root@ubuntu:/usr/src/linux-4.16.10# make menuconfig
  HOSTCC  scripts/kconfig/mconf.o
  YACC    scripts/kconfig/zconf.tab.c
/bin/sh: 1: bison: not found
scripts/Makefile.lib:217: recipe for target 'scripts/kconfig/zconf.tab.c' failed
make[1]: *** [scripts/kconfig/zconf.tab.c] Error 127
Makefile:514: recipe for target 'menuconfig' failed
make: *** [menuconfig] Error 2
```

```
root@ubuntu:/usr/src/linux-4.16.10# apt install bison -y
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
下列软件包是自动安装的并且现在不需要了：
  fonts-liberation2 fonts-opensymbol gir1.2-gst-plugins-base-1.0 gir1.2-gstreamer-1.0 gir1.2-gudev-1.0 gir1.2-udisks-2.0 grilo-plugins-0.3-base gstreamer1.0-gtk3 libboost-date-time1.65.1
  libboost-filesystem1.65.1 libboost-iostreams1.65.1 libboost-locale1.65.1 libcdr-0.1-1 libclucene-contribs1v5 libclucene-core1v5 libcmis-0.5-5v5 libcoland2 libdazzle-1.0-0 libe-book-0.1-1
  libedataserverui-1.2-2 libeot0 libepubgen-0.1-1 libetonyek-0.1-1 libevent-2.1-6 libexiv2-14 libfreerdp-client2-2 libfreerdp2-2 libgc1c2 libgee-0.8-2 libgexiv2-2 libgom-1.0-0 libgpgmepp6 libgpod-common
  libgpod4 liblangtag-common liblangtag1 liblirc-client0 liblua5.3-0 libmediaart-2.0-0 libmspub-0.1-1 libodfgen-0.1-1 libqqwing2v5 libraw16 librevenge-0.0-0 libsgutils2-2 libssh-4 libsuitesparseconfig5
  libvncclient1 libwinpr2-2 libxaplan30 libxmlsec1-nss linux-hwe-5.4-headers-5.4.0-84 lp-solve media-player-info python3-mako python3-markupsafe syslinux syslinux-common syslinux-legacy
  usb-creator-common
使用'apt autoremove'来卸载它(它们)。
将会同时安装下列软件：
  libbison-dev
建议安装：
  bison-doc
下列【新】软件包将被安装：
  bison libbison-dev
升级了 0 个软件包，新安装了 2 个软件包，要卸载 0 个软件包，有 69 个软件包未被升级。
需要下载 605 kB 的归档。
解压缩后会消耗 1,811 kB 的额外空间。
获取:1 http://us.archive.ubuntu.com/ubuntu bionic/main amd64 libbison-dev amd64 2:3.0.4.dfsg-1build1 [339 kB]
获取:2 http://us.archive.ubuntu.com/ubuntu bionic/main amd64 bison amd64 2:3.0.4.dfsg-1build1 [266 kB]
已下载 605 kB，耗时 2秒 (256 kB/s)
正在选中未选择的软件包 libbison-dev:amd64。
(正在读取数据库 ... 系统当前共安装有 179632 个文件和目录。)
正准备解包 .../libbison-dev_2%3a3.0.4.dfsg-1build1_amd64.deb  ...
正在解包 libbison-dev:amd64 (2:3.0.4.dfsg-1build1) ...
正在选中未选择的软件包 bison。
正准备解包 .../bison_2%3a3.0.4.dfsg-1build1_amd64.deb  ...
正在解包 bison (2:3.0.4.dfsg-1build1) ...
正在设置 libbison-dev:amd64 (2:3.0.4.dfsg-1build1) ...
正在设置 bison (2:3.0.4.dfsg-1build1) ...
update-alternatives: 使用 /usr/bin/bison.yacc 来在自动模式中提供 /usr/bin/yacc (yacc)
正在处理用于 man-db (2.8.3-2ubuntu0.1) 的触发器 ...
```

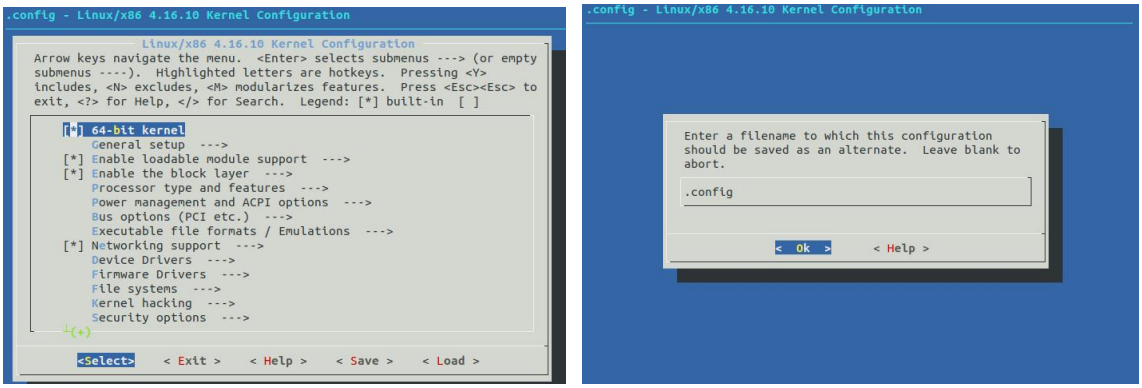
教务处制

再次执行 make menuconfig，出现错误：缺少 flex，使用如下命令安装：

```
# apt install flex -y
```

```
root@ubuntu:/usr/src/linux-4.16.10# make menuconfig
  YACC    scripts/kconfig/zconf.tab.c
  LEX     scripts/kconfig/zconf.lex.c
/bin/sh: 1: flex: not found
scripts/Makefile.lib:202: recipe for target 'scripts/kconfig/zconf.lex.c' failed
make[1]: *** [scripts/kconfig/zconf.lex.c] Error 127
Makefile:514: recipe for target 'menuconfig' failed
make: *** [menuconfig] Error 2
root@ubuntu:/usr/src/linux-4.16.10# apt install flex -y
```

再次执行 make menuconfig，成功进入图形化界面，根据需求选择需要编译进内核的功能后，选择<Save>进行保存：



(5) 编译内核

执行以下命令：

```
# make -j8
```

出现错误：缺少 openssl 和 libssl-dev

```
Makefile:976: "Cannot use CONFIG_STACK_VALIDATION=y, please install libelf-dev, libelf-devel or elfutils-libelf-devel"
```

```
scripts/sign-file.c:25:10: fatal error: openssl/opensslv.h: 没有那个文件或目录
 #include <openssl/opensslv.h>
          ^~~~~~~~~~~~~~~~~~~~
compilation terminated.
```

执行如下命令安装 openssl 和 libssl-dev：

```
# apt install openssl
# apt install libssl-dev
```

再次执行 make -j8，编译成功：

```
  CC      kernel/crash_dump.o
  CC      kernel/jump_label.o
  CC      kernel/memremap.o
  CC [M]  kernel/torture.o
  AR      kernel/built-in.o
root@ubuntu:/usr/src/linux-4.16.10#
```

(6) 编译和安装模块

执行以下命令进行编译和安装：

```
# make modules_install
```

运行结果如下：

```
  INSTALL sound/usb/snd-usbmidi-lib.ko
  INSTALL sound/usb/usx2y/snd-usb-us122l.ko
  INSTALL sound/usb/usx2y/snd-usb-usx2y.ko
  INSTALL sound/x86/snd-hdmi-lpe-audio.ko
  DEPMOD  4.16.10
```

(7) 安装内核

执行以下命令进行安装：

```
# sudo make install
```

运行结果如下：

```
root@ubuntu:/usr/src/linux-4.16.10# sudo make install
Makefile:976: "Cannot use CONFIG_STACK_VALIDATION=y, please install libelf-dev, libelf-devel or elfutils-libelf-devel"
sh ./arch/x86/boot/install.sh 4.16.10 arch/x86/boot/bzImage \
        System.map "/boot"
run-parts: executing /etc/kernel/postinst.d/apt-auto-removal 4.16.10 /boot/vmlinuz-4.16.10
run-parts: executing /etc/kernel/postinst.d/initramfs-tools 4.16.10 /boot/vmlinuz-4.16.10
update-initramfs: Generating /boot/initrd.img-4.16.10
run-parts: executing /etc/kernel/postinst.d/unattended-upgrades 4.16.10 /boot/vmlinuz-4.16.10
run-parts: executing /etc/kernel/postinst.d/update-notifier 4.16.10 /boot/vmlinuz-4.16.10
run-parts: executing /etc/kernel/postinst.d/xx-update-initrd-links 4.16.10 /boot/vmlinuz-4.16.10
I: /vmlinuz.old is now a symlink to boot/vmlinuz-5.4.0-146-generic
I: /initrd.img.old is now a symlink to boot/initrd.img-5.4.0-146-generic
I: /vmlinuz is now a symlink to boot/vmlinuz-4.16.10
I: /initrd.img is now a symlink to boot/initrd.img-4.16.10
run-parts: executing /etc/kernel/postinst.d/zz-update-grub 4.16.10 /boot/vmlinuz-4.16.10
Sourcing file `/etc/default/grub'
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-5.4.0-146-generic
Found initrd image: /boot/initrd.img-5.4.0-146-generic
Found linux image: /boot/vmlinuz-5.4.0-144-generic
Found initrd image: /boot/initrd.img-5.4.0-144-generic
Found linux image: /boot/vmlinuz-4.16.10
Found initrd image: /boot/initrd.img-4.16.10
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
done
root@ubuntu:/usr/src/linux-4.16.10#
```

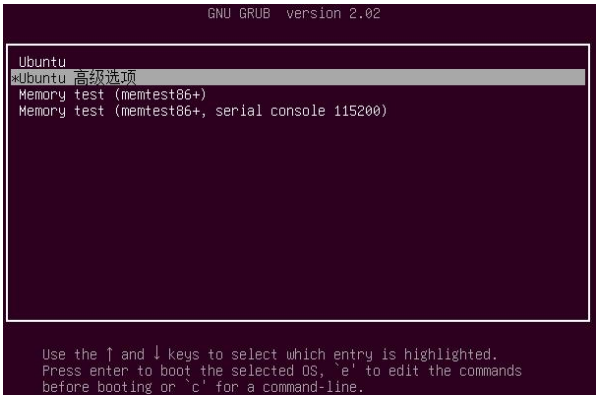
(8) 重新启动，检查新内核

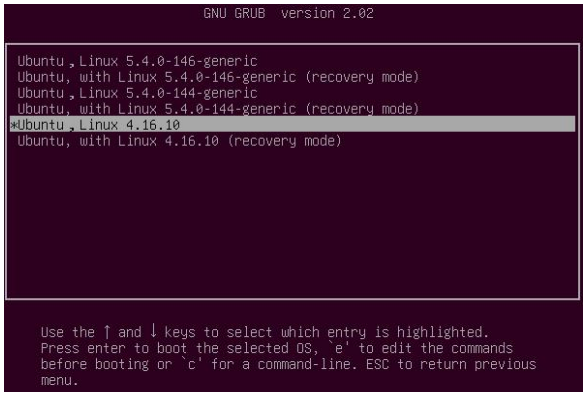
输入指令“reboot”重新启动，并长按“Shift”键进入 GRUB 菜单：

```
# reboot
# //长按 shift
```

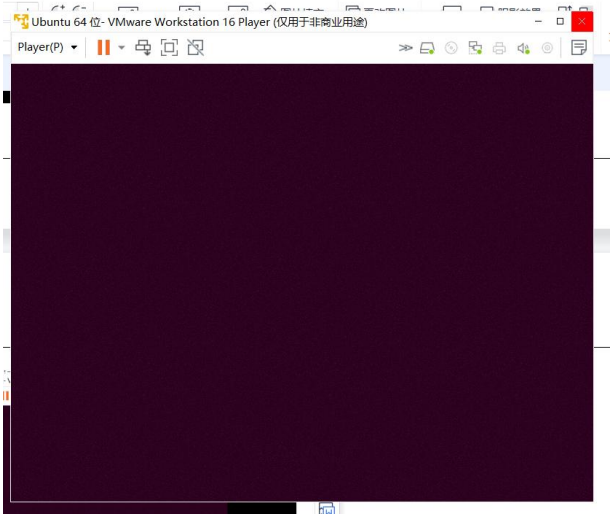
使用“⬆”“⬇”键选择，使用“Enter”键进入“Ubuntu 高级选项”

选择选项“Ubuntu，Linux 4.16.10”按下“Enter”回车后，使用 linux4.16.10 内核启动。



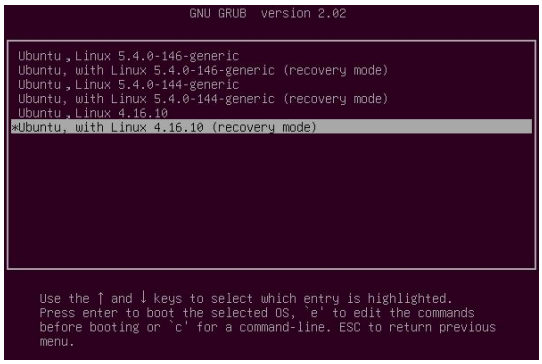


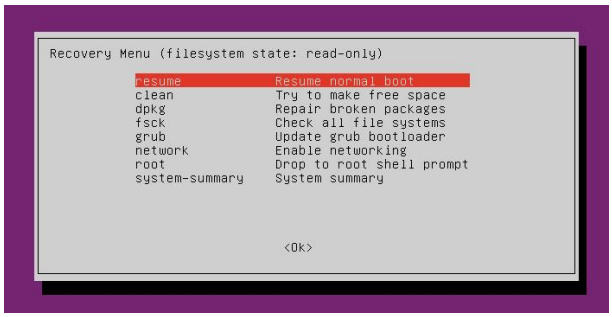
发现卡在如下界面



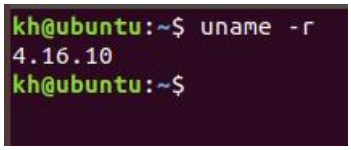
教务处制

重新启动，按照同样的步骤选择 Linux4.16.10 的 recovery mode 恢复模式，选择第一个选项（resume--Resume normal boot），“恢复正常启动”进行修复，最终成功进入系统





输入指令“uname -r”查看内核版本号：

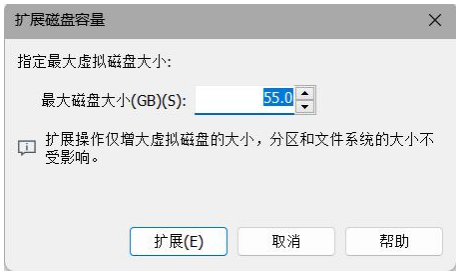


更换内核成功。

五、 实验思考与总结

1. 思考1：磁盘空间不足

已分配给ubuntu虚拟机的磁盘空间不足时，不仅需要在vmware的设置中扩展磁盘空间，还需要在ubuntu系统中使用Gparted将新分配的空间划分给系统分区。



2. 思考2：缺少依赖

在编译过程中，系统报错：缺少相关依赖库，可以按照错误提示安装相应的库，在熟悉了编译过程，积累了足够的经验后，可以预先将需要的依赖库提前安装好。

3. 实验总结

本次实验是Linux内核编译实验，在本次实验中，学习了重新编译Linux内核的方法，手动编译了Linux内核，同时对于Linux标准内核和发行版本内核的区别有了更深理解：linux核心只有内核部分，安装完后，用户界面/软件都没有；linux发行版就是在内核的基础上，加入用户界面和各种软件的支持。在遇到问题时，可以通过问题的提示进行操作或搜索相关技术来解决，对问题的处理有了更多的经验积累。