

# 实验一 编译 Linux 内核

07121502 班 1320151098 赵璐

**1.实验目的：**通过编译 Linux 内核，熟悉 Linux 操作系统及其基本命令的使用，掌握构建与启动 Linux 内核的方法。

**2.实验环境：**VMware – Ubuntu 16.04 LTS 64-bit

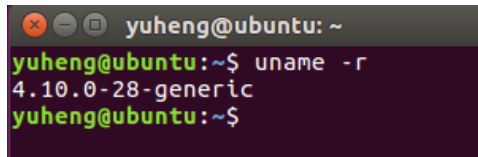
**3.实验内容：**下载 Linux 内核源码，编译并生成 Linux 内核，将新编译出来的内核更新到自己的 Linux 系统中。

**4.实验过程：**

## 4.1 查看现有内核版本

```
# uname -r
```

现有内核版本为 4.10.0-28



```
yuheng@ubuntu: ~  
yuheng@ubuntu:~$ uname -r  
4.10.0-28-generic  
yuheng@ubuntu:~$
```

## 4.2 下载新内核源码并解压

### 1.下载源码

起初在终端中下载，由于下载速度太慢，放弃了这种方法。

源码下载地址：<https://www.kernel.org> 下载最新的 4.15.7 版本内核。

通过虚拟机连接 U 盘，将压缩包拷贝到 Downloads 目录下。

### 2.解压到当前目录

```
# cd Downloads/
```

```
# cd linux-4.15.7/
```

```
# tar xzyf linux-4.15.7.tar.gz
```

## 4.3 配置内核编译选项

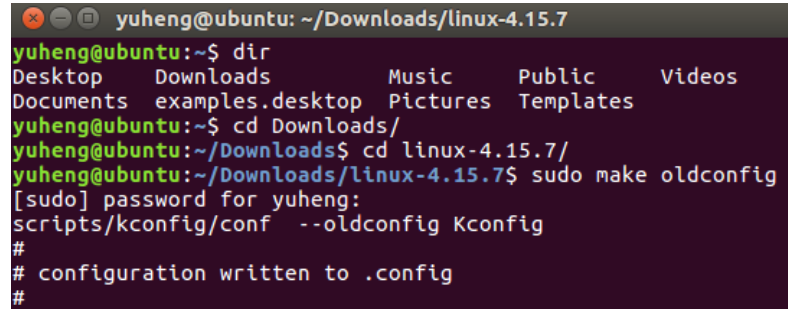
### 1.进入解压的内核源码目录下

```
# cd Downloads/
```

```
# cd linux-4.15.7/
```

## 2.基于.config 配置文件配置内核

```
# sudo make oldconfig
```

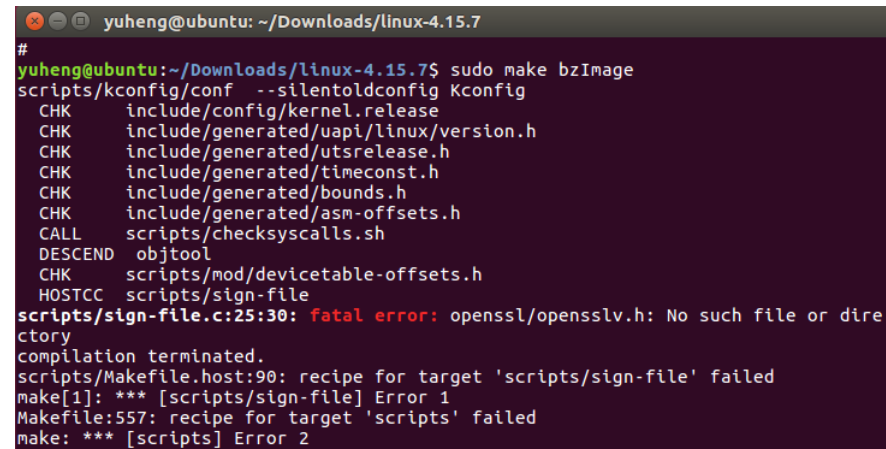


```
yuheng@ubuntu: ~/Downloads/linux-4.15.7
yuheng@ubuntu:~$ dir
Desktop    Downloads      Music          Public         Videos
Documents  examples.desktop Pictures        Templates
yuheng@ubuntu:~$ cd Downloads/
yuheng@ubuntu:~/Downloads$ cd linux-4.15.7/
yuheng@ubuntu:~/Downloads/linux-4.15.7$ sudo make oldconfig
[sudo] password for yuheng:
scripts/kconfig/conf --oldconfig Kconfig
#
# configuration written to .config
#
```

## 4.4 编译内核和模块

### 1.编译内核

```
# sudo make bzImage
```



```
yuheng@ubuntu: ~/Downloads/linux-4.15.7
#
yuheng@ubuntu:~/Downloads/linux-4.15.7$ sudo make bzImage
scripts/kconfig/conf --silentoldconfig Kconfig
CHK      include/config/kernel.release
CHK      include/generated/uapi/linux/version.h
CHK      include/generated/utsrelease.h
CHK      include/generated/timeconst.h
CHK      include/generated/bounds.h
CHK      include/generated/asm-offsets.h
CALL     scripts/checksyscalls.sh
DESCEND  objtool
CHK      scripts/mod/devicetable-offsets.h
HOSTCC   scripts/sign-file
scripts/sign-file.c:25:30: fatal error: openssl/opensslv.h: No such file or directory
compilation terminated.
scripts/Makefile.host:90: recipe for target 'scripts/sign-file' failed
make[1]: *** [scripts/sign-file] Error 1
Makefile:557: recipe for target 'scripts' failed
make: *** [scripts] Error 2
```

编译出现错误，根据命令行提示及网络查询，按以下命令成功修复错误：

```
# sudo apt-get install libssl-dev      Wrong
```

```
# sudo apt-get install libssl1.0.0=1.0.2g-1ubuntu4    Done
```

```
# sudo apt-get install libssl-dev      Wrong
```

```
# sudo apt-get install zlib1g-dev      Wrong
```

```
# sudo apt-get install zlib1g=1:1.2.8.dfsg-2ubuntu4    Done
```

```
# sudo apt-get install zlib1g-dev      Done
```

```
# sudo apt-get install libssl-dev      Done
```

此时，重新编译内核，成功。

```
yuheng@ubuntu: ~/Downloads/linux-4.15.7
Setting up libssl-dev:amd64 (1.0.2g-1ubuntu4) ...
Setting up libssl-doc (1.0.2g-1ubuntu4) ...
yuheng@ubuntu:~/Downloads/linux-4.15.7$ sudo make bzImage
CHK include/config/kernel.release
CHK include/generated/uapi/linux/version.h
CHK include/generated/utsrelease.h
CHK include/generated/timeconst.h
CHK include/generated/bounds.h
CHK include/generated/asm-offsets.h
CALL scripts/checksyscalls.sh
DESCEND objtool
CHK scripts/mod/devicetable-offsets.h
HOSTCC scripts/sign-file
HOSTCC scripts/extract-cert
HOSTCC scripts/insert-sys-cert
CC scripts/Makefile
```

## 2.编译模块

# sudo make modules

```
yuheng@ubuntu:~/Downloads/linux-4.15.7$ sudo make modules
[sudo] password for yuheng:
CHK include/config/kernel.release
CHK include/generated/uapi/linux/version.h
CHK include/generated/utsrelease.h
CHK include/generated/bounds.h
CHK include/generated/timeconst.h
CHK include/generated/asm-offsets.h
CALL scripts/checksyscalls.sh
DESCEND objtool
CHK scripts/mod/devicetable-offsets.h
CC [M] arch/x86/crypto/glue_helper.o
AS [M] arch/x86/crypto/aes-x86_64-asm_64.o
CC [M] arch/x86/crypto/aes_glue.o
LD [M] arch/x86/crypto/aes-x86_64.o
AS [M] arch/x86/crypto/des3_edc-asm_64.o
```

## 4.5 安装新内核

### 1.安装模块

# sudo make modules\_install

```
yuheng@ubuntu: ~/Downloads/linux-4.15.7
LD [M] virt/lib/irqbypass.ko
yuheng@ubuntu:~/Downloads/linux-4.15.7$ sudo make modules_install
[sudo] password for yuheng:
INSTALL arch/x86/crypto/aes-x86_64.ko
INSTALL arch/x86/crypto/aesni-intel.ko
INSTALL arch/x86/crypto/blowfish-x86_64.ko
INSTALL arch/x86/crypto/camellia-aesni-avx-x86_64.ko
INSTALL arch/x86/crypto/camellia-aesni-avx2.ko
INSTALL arch/x86/crypto/camellia-x86_64.ko
INSTALL arch/x86/crypto/cast5-avx-x86_64.ko
INSTALL arch/x86/crypto/cast6-avx-x86_64.ko
INSTALL arch/x86/crypto/chacha20-x86_64.ko
```

### 2.安装程序

# sudo make install

```

yuheng@ubuntu: ~/Downloads/linux-4.15.7
yuheng@ubuntu:~/Downloads/linux-4.15.7$ sudo make install
sh ./arch/x86/boot/install.sh 4.15.7 arch/x86/boot/bzImage \
System.map "/boot"
run-parts: executing /etc/kernel/postinst.d/apt-auto-removal 4.15.7 /boot/vmlinu
z-4.15.7
run-parts: executing /etc/kernel/postinst.d/initramfs-tools 4.15.7 /boot/vmlinuz
-4.15.7
update-initramfs: Generating /boot/initrd.img-4.15.7

run-parts: executing /etc/kernel/postinst.d/pm-utils 4.15.7 /boot/vmlinuz-4.15.7
run-parts: executing /etc/kernel/postinst.d/unattended-upgrades 4.15.7 /boot/vml
inuz-4.15.7
run-parts: executing /etc/kernel/postinst.d/update-notifier 4.15.7 /boot/vmlinuz
-4.15.7

```

## 4.6 修改启动设置

### 1.制作启动盘

```
# sudo mkinitramfs 4.15.7 -o /boot/initrd.img-4.15.7
```

### 2.更新启动引导程序

```
# sudo update-grub2
```

```

yuheng@ubuntu:~/Downloads/linux-4.15.7$ sudo mkinitramfs 4.15.7 -o /boot/initrd.
img-4.15.7

yuheng@ubuntu:~/Downloads/linux-4.15.7$
yuheng@ubuntu:~/Downloads/linux-4.15.7$ sudo update-grub2
Generating grub configuration file ...
Warning: Setting GRUB_TIMEOUT to a non-zero value when GRUB_HIDDEN_TIMEOUT is se
t is no longer supported.
Found linux image: /boot/vmlinuz-4.15.7
Found initrd image: /boot/initrd.img-4.15.7
Found linux image: /boot/vmlinuz-4.10.0-28-generic
Found initrd image: /boot/initrd.img-4.10.0-28-generic
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
done

```

## 4.7 重启虚拟机

### 1.导入启动项

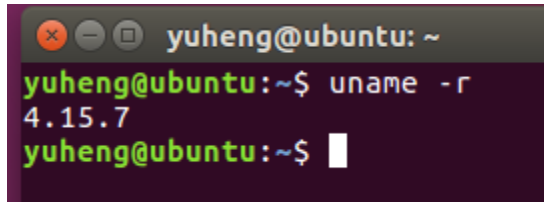
```

GNU GRUB version 2.02~beta2-36ubuntu3.12

*Ubuntu, with Linux 4.15.7
Ubuntu, with Linux 4.15.7 (upstart)
Ubuntu, with Linux 4.15.7 (recovery mode)
Ubuntu, with Linux 4.10.0-28-generic
Ubuntu, with Linux 4.10.0-28-generic (upstart)
Ubuntu, with Linux 4.10.0-28-generic (recovery mode)

```

### 2.查看更新后内核版本



```
yuheng@ubuntu: ~  
yuheng@ubuntu:~$ uname -r  
4.15.7  
yuheng@ubuntu:~$
```

## 5.心得体会

用惯了 Windows 的图形化界面，对本次实验使用的 Linux 命令行显得有些陌生，甚至都不了解一些基本的命令。所以一听到这个实验作业感觉有些心虚，好在之前的课程中就使用过虚拟机中的 Ubuntu 系统，省去了安装实验环境的环节，并通过翻阅课本，查询网络对实验整体有了大概的掌握，也慢慢有了进展，到最后才发现实验其实并不困难。

收获 1：通过本次实验我熟练掌握了 Linux 系统中的一些常用命令，如：

# dir	查看当前目录
# cd ..	返回上一级
# cd Download/	进入某个目录下
# sudo apt-get install...	自动获取安装...
# sudo update-...	更新...

收获 2：掌握了通过编译并安装新内核的方法来更新系统内核版本的方法。

收获 3：最大化利用网络共享资源，发现获取资源的途径。

收获 4：面对从未遇到过的问题，不要产生畏难心理。