

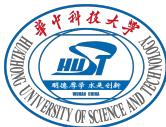
操作系统的生成

在 Ubuntu 系统编译一个新的内核

侯皓斐

华中科技大学软件学院 2003 班

2022 年 3 月 14 日



③ 参考资料

操作系统生成

定义

操作系统生成是指根据硬件环境的配置和用户的需求，重新配置、裁剪和构建操作系统的过程。

操作系统生成的前提

- 操作系统由可拆装模块构成
- 有交互式配置工具
- 有映像构建 (build) 工具

① 操作系统的生成

② Ubuntu20.04 编译内核

实验环境

实验目的

实验流程

③ 参考资料

① 操作系统的生成

② Ubuntu20.04 编译内核

实验环境

实验目的

实验流程

③ 参考资料

实验环境

- 使用虚拟机环境：VMWare Pro 16
- Ubuntu 20.04.4 LTS



① 操作系统的生成

② Ubuntu20.04 编译内核

实验环境

实验目的

实验流程

③ 参考资料

实验目的

- 将当前 Ubuntu 的内核更换至最新的稳定版内核。

① 操作系统的生成

② Ubuntu20.04 编译内核

实验环境

实验目的

实验流程

③ 参考资料

实验流程

- ① 查看当前系统内核版本
- ② 安装所需依赖，配置相关环境
- ③ 在 <https://www.kernel.org/> 下载稳定版内核
- ④ 解压缩内核
- ⑤ 净化内核源码
- ⑥ 复制现有配置（让事情变得简单）
- ⑦ 使用基于终端的配置工具（大概观察，并未做改动）
- ⑧ 编译内核
- ⑨ 安装内核
- ⑩ 更新 grub
- ⑪ 重新启动，重新查看系统内核版本，并对比

1. 查看当前系统内核版本

使用 `uname -r` 指令。

```
moonlight@moonlight-virtual-machine:~/桌面$ uname -r  
5.13.0-35-generic
```

2. 安装所需依赖，配置相关环境

安装相关依赖。

```
moonlight@moonlight-virtual-machine:~$ sudo apt update && sudo apt upgrade
```

```
moonlight@moonlight-virtual-machine:~$ sudo apt-get install git fakeroot build-essential ncurses-dev xz-utils libssl-dev bc flex libelf-dev bison
```

(请注意这一个)

```
moonlight@moonlight-virtual-machine:~/下载/linux-5.16.14$ sudo apt install dwarves
```

2. 安装所需依赖，配置相关环境

加速编译，配置好 ccache。安装 ccache。

```
moonlight@moonlight-virtual-machine:~$ sudo apt install ccache
```

配置 ccache。

```
moonlight@moonlight-virtual-machine:~$ sudo gedit ~/.bashrc
```

末尾回车添加语句。

```
111 if ! shopt -oq posix; then
112   if [ -f /usr/share/bash-completion/bash_completion ]; then
113     . /usr/share/bash-completion/bash_completion
114   elif [ -f /etc/bash_completion ]; then
115     . /etc/bash_completion
116   fi
117 fi
118
119 export USE_CCACHE=1
120 export CCACHE_DIR="/home/moonlight/.ccache"
121 export CC="ccache gcc"
122 export CXX="ccache g++"
123 export PATH="$PATH:/usr/lib/ccache"
```

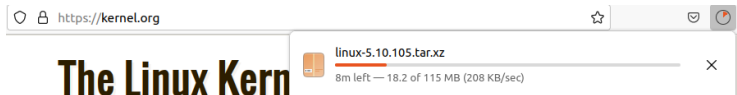
按 Ctrl+S 或点击“保存”按钮保存，然后使其生效。

```
moonlight@moonlight-virtual-machine:~$ source ~/.bashrc
moonlight@moonlight-virtual-machine:~$ ccache -M 40G
```

3. 在 <https://www.kernel.org/> 下载稳定版内核

在 <https://www.kernel.org/> 下载稳定版内核。

当前的稳定版内核为 2022-03-11 发行的 linux-5.16.14。



4. 解压缩内核

解压缩内核，并进入目录。

```
moonlight@moonlight-virtual-machine:~/下载$ tar xf linux-5.16.14.tar.xz  
moonlight@moonlight-virtual-machine:~/下载$ cd linux-5.16.14/
```

5. 净化内核源码

使用命令 `make mrproper`。
也可以在想重新编译时执行这个命令。

```
moonlight@moonlight-virtual-machine:~/下载/linux-5.16.14$ make mrproper
```

6. 复制现有配置（让事情变得简单）

使用命令 `make xconfig`，会自动选取当前内核的配置文件当做 `.config` 文件，也可以在这个基础上修改。

点击 `save` 然后退出，最好把搞出来的 `.config` 备份一下。

```
moonlight@moonlight-virtual-machine:~/下载/linux-5.16.14$ make xconfig
HOSTCC      scripts/basic/fixdep
HOSTCC      scripts/kconfig/images.o
HOSTCC      scripts/kconfig/confdata.o
HOSTCC      scripts/kconfig/expr.o
LEX          scripts/kconfig/lexer.lex.c
YACC         scripts/kconfig/parser.tab.[ch]
HOSTCC      scripts/kconfig/lexer.lex.o
HOSTCC      scripts/kconfig/menu.o
HOSTCC      scripts/kconfig/parser.tab.o
HOSTCC      scripts/kconfig/preprocess.o
HOSTCC      scripts/kconfig/symbol.o
HOSTCC      scripts/kconfig/util.o
UPD          scripts/kconfig/qconf-cfg
HOSTCXX     scripts/kconfig/qconf.o
MOC          scripts/kconfig/qconf-moc.cc
HOSTCXX     scripts/kconfig/qconf-moc.o
HOSTLD      scripts/kconfig/qconf

#
# using defaults found in /boot/config-5.13.0-35-generic
#
.config:8757:warning: symbol value 'm' invalid for ASHMEM
.config:9809:warning: symbol value 'm' invalid for ANDROID_BINDER_IPC
.config:9810:warning: symbol value 'm' invalid for ANDROID_BINDERFS
#
```

7. 使用基于终端的配置工具（大概观察，并未做改动）

使用命令 `make menuconfig`，但我只进行了大概地观察，并未做改动。

```
moonlight@moonlight-virtual-machine:~/下载/linux-5.16.14$ make menuconfig
UPD      scripts/kconfig/mconf-cfg
HOSTCC   scripts/kconfig/mconf.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
HOSTLD   scripts/kconfig/mconf

*** End of the configuration.
*** Execute 'make' to start the build or try 'make help'.
```

7. 使用基于终端的配置工具（大概观察，并未做改动）

使用命令 `make menuconfig`，但我只进行了大概地观察，并未做改动。

```
.config - Linux/x86 5.16.14 Kernel Configuration

Linux/x86 5.16.14 Kernel Configuration
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty
submenus ----). Highlighted letters are hotkeys. Pressing <Y>
includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to
exit, <?> for Help, </> for Search. Legend: [*] built-in [ ]

General setup --->
[*] 64-bit kernel
    Processor type and features --->
        Power management and ACPI options --->
        Bus options (PCI etc.) --->
        Binary Emulations --->
[*] Virtualization --->
    General architecture-dependent options --->
[*] Enable loadable module support --->
[*] Enable the block layer --->
    Executable file formats --->
    Memory Management options --->
[*] Networking support --->
    Device Drivers --->
    File systems --->
    Security options --->
-*- Cryptographic API --->
v(+)
```

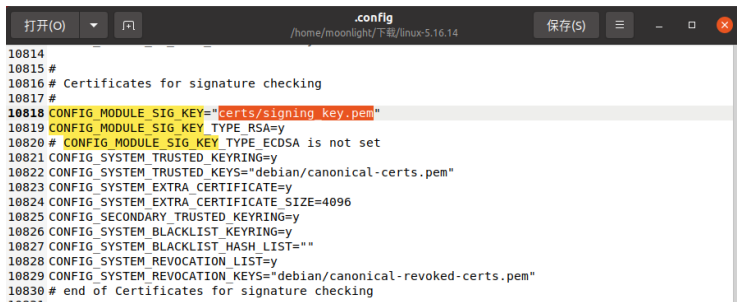
8. 编译内核

编译主体前注意对.config 文件进行修改。
打开形成的.config 文件。

```
moonlight@moonlight-virtual-machine:~/下载/linux-5.16.14$ sudo gedit .config
```

8. 编译内核

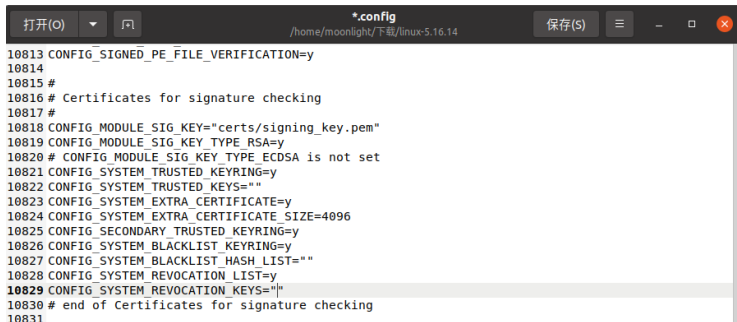
将如下设置：



```
.config
/home/moonlight/下载/linux-5.16.14
10814
10815 #
10816 # Certificates for signature checking
10817 #
10818 CONFIG_MODULE_SIG_KEY="certs/signing key.pem"
10819 CONFIG_MODULE_SIG_KEY_TYPE_RSA=y
10820 # CONFIG_MODULE_SIG_KEY_TYPE_ECDSA is not set
10821 CONFIG_SYSTEM_TRUSTED_KEYRING=y
10822 CONFIG_SYSTEM_TRUSTED_KEYS="debian/canonical-certs.pem"
10823 CONFIG_SYSTEM_EXTRA_CERTIFICATE=y
10824 CONFIG_SYSTEM_EXTRA_CERTIFICATE_SIZE=4096
10825 CONFIG_SECONDARY_TRUSTED_KEYRING=y
10826 CONFIG_SYSTEM_BLACKLIST_KEYRING=y
10827 CONFIG_SYSTEM_BLACKLIST_HASH_LIST=""
10828 CONFIG_SYSTEM_REVOCATION_LIST=y
10829 CONFIG_SYSTEM_REVOCATION_KEYS="debian/canonical-revoked-certs.pem"
10830 # end of Certificates for signature checking
10831
```

8. 编译内核

删除全部 debian。



```
10813 CONFIG_SIGNED_PE_FILE_VERIFICATION=y
10814
10815 #
10816 # Certificates for signature checking
10817 #
10818 CONFIG_MODULE_SIG_KEY="certs/signing_key.pem"
10819 CONFIG_MODULE_SIG_KEY_TYPE_RSA=y
10820 # CONFIG_MODULE_SIG_KEY_TYPE_ECDSA is not set
10821 CONFIG_SYSTEM_TRUSTED_KEYRING=y
10822 CONFIG_SYSTEM_TRUSTED_KEYS=""
10823 CONFIG_SYSTEM_EXTRA_CERTIFICATE=y
10824 CONFIG_SYSTEM_EXTRA_CERTIFICATE_SIZE=4096
10825 CONFIG_SECONDARY_TRUSTED_KEYRING=y
10826 CONFIG_SYSTEM_BLACKLIST_KEYRING=y
10827 CONFIG_SYSTEM_BLACKLIST_HASH_LIST=""
10828 CONFIG_SYSTEM_REVOCATION_LIST=y
10829 CONFIG_SYSTEM_REVOCATION_KEYS=""
10830 # end of Certificates for signature checking
10831
```


8. 编译内核

编译主体。

```
moonlight@moonlight-virtual-machine:~/下载/linux-5.16.14$ make bzImage -j2
```

编译模块。(耗时最长)

```
moonlight@moonlight-virtual-machine:~/下载/linux-5.16.14$ make modules -j2
```

8. 编译内核

编译主体中途又出现了问题。

```
/bin/sh: 1: zstd: not found
make[2]: *** [arch/x86/boot/compressed/Makefile:139: arch/x86/boot/compressed/vmlinux.bin.zst] 错误 127
make[2]: *** 正在删除文件“arch/x86/boot/compressed/vmlinux.bin.zst”
make[2]: *** 正在等待未完成的任务....
make[1]: *** [arch/x86/boot/Makefile:115: arch/x86/boot/compressed/vmlinux] 错误 2
make: *** [arch/x86/Makefile:253: bzImage] 错误 2
```

解决方式是，安装如下软件包。

```
moonlight@moonlight-virtual-machine:~/下载/linux-5.16.14$ sudo apt install zstd
```

8. 编译内核

编译主体成功后会产生如下界面：

```
Kernel: arch/x86/boot/bzImage is ready (#2)
```

9. 安装内核

安装模块。

```
moonlight@moonlight-virtual-machine:~/下载/linux-5.16.14$ sudo make INSTALL_MOD_  
STRIP=1 modules_install -j4
```

此时/lib/modules 下应该会出现内核版本号文件夹。

```
moonlight@moonlight-virtual-machine:~/下载/linux-5.16.14$ cd /lib/modules  
moonlight@moonlight-virtual-machine:/lib/modules$ ls -al  
总用量 28  
drwxr-xr-x  7 root root 4096 3月 13 11:35 .  
drwxr-xr-x 124 root root 4096 3月 13 09:12 ..  
drwxr-xr-x  5 root root 4096 2月 23 16:51 5.13.0-30-generic  
drwxr-xr-x  5 root root 4096 3月 12 22:40 5.13.0-35-generic  
drwxr-xr-x  3 root root 4096 3月 13 15:37 5.16.14
```

9. 安装内核

复制三个文件。

```
moonlight@moonlight-virtual-machine:/lib/modules$ sudo mkinitramfs /lib/modules/  
5.16.14 -o /boot/initrd.img-5.16.14
```

```
moonlight@moonlight-virtual-machine:~/下载/linux-5.16.14$ sudo cp arch/x86/boot/  
bzImage /boot/vmlinuz-5.16.14
```

```
moonlight@moonlight-virtual-machine:~/下载/linux-5.16.14$ sudo cp System.map /bo  
ot/System.map-5.16.14
```

10. 更新 grub

使用命令 `sudo update-grub2`。

```
moonlight@moonlight-virtual-machine:~/下载/linux-5.16.14$ sudo update-grub2
```

11. 重新启动，重新查看系统内核版本，并对比

重新启动后，再次使用 `uname -r` 指令。

```
moonlight@moonlight-virtual-machine:~/桌面$ uname -r  
5.16.14
```

- ◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡ 🔍 ↺ ↻

侯皓斐

操作系统的生成