

ArchLinux 内核编译

西安理工大学 Linux 小组指导教师 张 翔

<https://www.facebook.com/xautofzx/>

本教程主要面向计算机类专业大三及以上的大学生，最好学过《操作系统》课程。

读者应该有 Ubuntu/Fedora 等流行的 Linux 发行版的安装配置使用经验，或在教师指导下使用本教程。

1. 对 Linux 内核构建的模式

Arch Linux 提供了多种内核构建方式。推荐使用 Arch 构建系统，这样可以充分利用已有的 linux PKGBUILD 以及包管理系统。PKGBUILD 已经是结构化的，可以在下载源代码之后配置内核。

另外，也可以使用传统方式，手动下载内核源代码包，然后在自己的主目录里以普通用户的权限编译。传统方式的优点是在其他发行版中也可以使用。

一旦完成配置之后，有两种编译(安装)方式：传统的安装和适用 makepkg/pacman 的安装。

参考 <https://wiki.archlinux.org/index.php/Kernels/Compilation/Traditional>

2. 下载内核源代码

在终端窗口输入 firefox 启动浏览器，访问 www.kernel.org，下载右侧黄色块所示的最新 stable 稳定版内核，保存到 /root 目录下。如果网速太慢，就在 Windows 下用迅雷等工具下载，然后运行 Filezilla Server 架设 FTP 服务器，或者用 HFS 软件架设简易的 Web 服务器，然后再用 Firefox 下载。<http://www.rejetto.com/hfs/?f=dl>

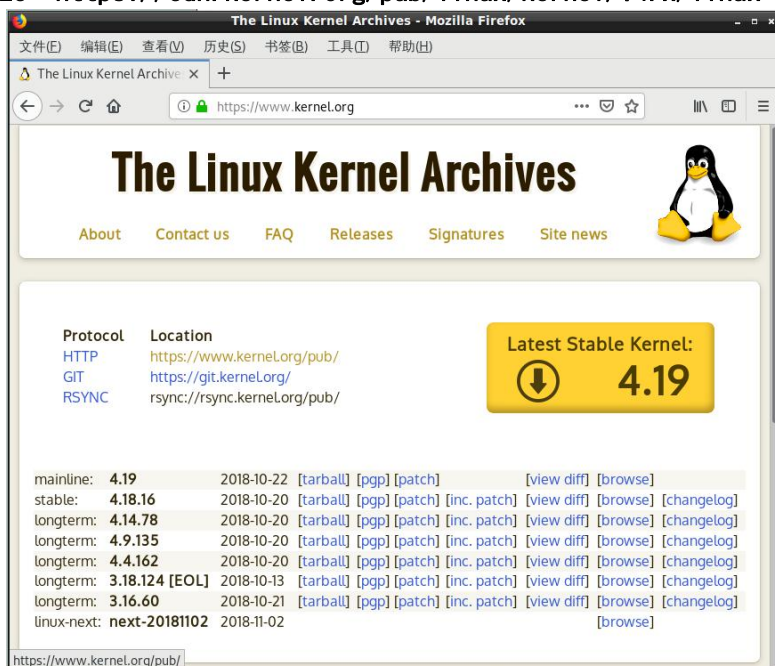
也可以使用命令行工具来下载。在命令行输入：`pacman -S wget aria2`，安装命令行支持断点续传和多线程的下载工具，二者任选其一。

一般用命令：`wget https://cdn.kernel.org/pub/linux/kernel/v4.x/linux-4.19.tar.xz` 就够了。

参考 <https://wiki.archlinux.org/index.php/Wget>。

如果网站有限速，则应采用 aria2c 多线程下载工具，参考 <https://wiki.archlinux.org/index.php/Aria2>

在命令行执行：`aria2c https://cdn.kernel.org/pub/linux/kernel/v4.x/linux-4.19.tar.xz`



将源代码解压缩到当前目录，`tar -xJvf linux-4.19.tar.xz`，注意第二个参数是大写“J”，指定后台用 xz 软件解压缩。由于虚拟机的性能较弱，大约需要 5 分钟左右，才能解压完毕。如果去掉 v 参数，就不会显示解压过程，能节省 1 分钟时间。

3. 配置内核

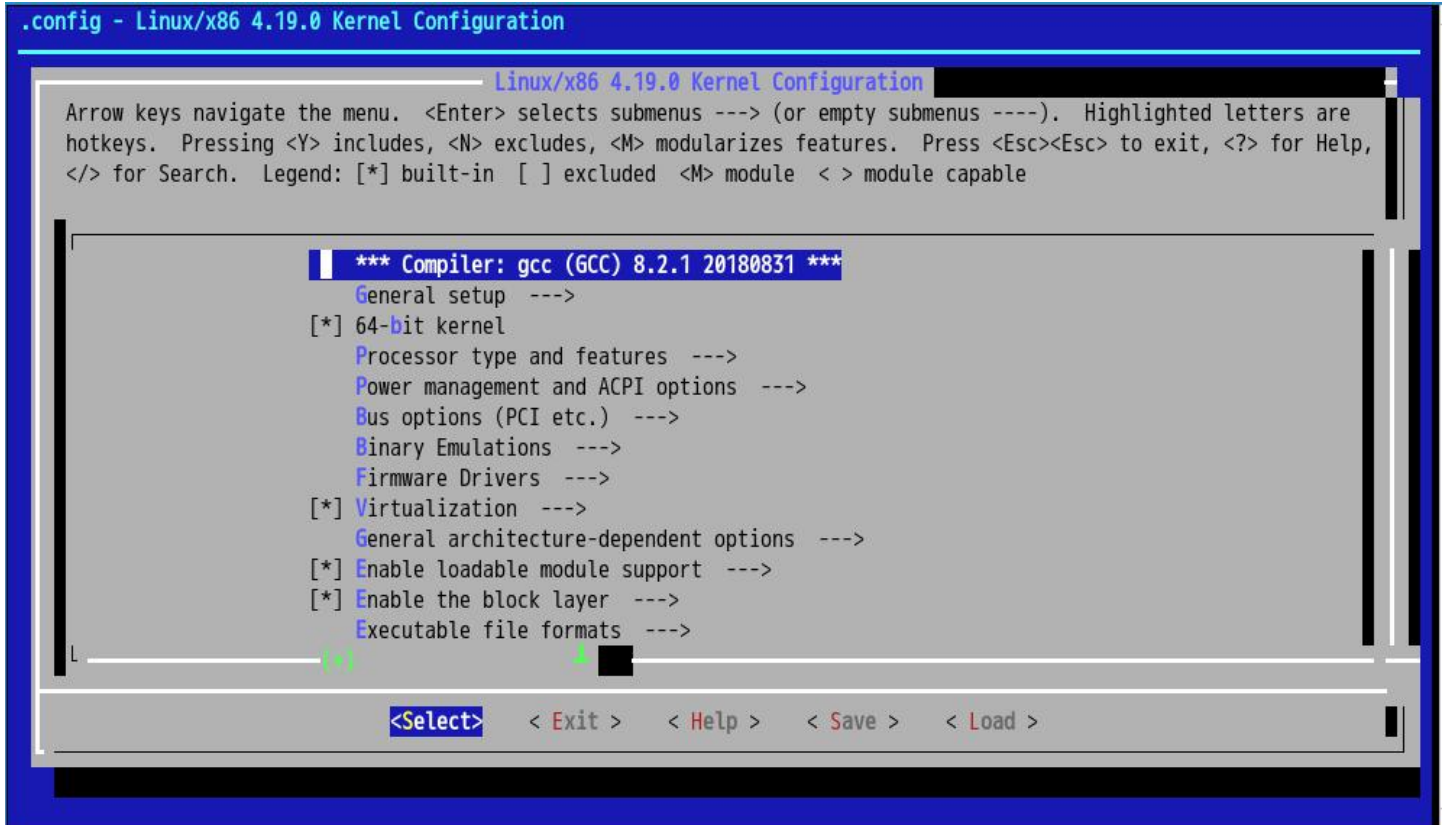
在终端窗口输入 `cd linux-4.19` 进入源代码目录。

```
[root@xautLinux linux-4.19]# ls -la
.      certs      CREDITS      firmware      .gitignore    Kbuild      LICENSES      mm      scripts      usr
..     .clang-format  crypto       fs            include       Kconfig     .mailmap      net      security    virt
arch   .coocciconfig  Documentation .get_maintainer.ignore  init          kernel      MAINTAINERS   README   sound       tools
block  COPYING        drivers      .gitattributes  ipc           lib          Makefile      samples  tools
```

源代码中不包含内核配置文件“.config”。为了节省配置时间，并避免因配置失误造成当前计算机启动不正常，把当前运行着的系统的配置拷贝到源代码目录，然后略加修改即可。

在命令行输入 `zcat /proc/config.gz > .config`，注意第一个字符是小数点。

接着输入 `make menuconfig`，等待几秒钟，就会出现图形配置界面。



在菜单“General setup”上按回车，在“Local version - append to kernel release”上回车，输入后缀名“-xaut”，回车即可。再按两次键盘左上角的 Esc 键，回到主菜单。再按两次 Esc 键，提示保存新的配置，选择 Yes，然后回车即可退出配置界面。为避免编译出现意想不到的错误，对于新手来说，在第一次编译内核时不建议修改其他参数。

用命令：`ls -l .config`，就能看到该文件 220KB 左右，用命令：`vi .config`，可以看到里面的信息非常繁多，所以不建议手工修改。

提示：如果想编译最小的内核，则输入 `make allnoconfig`，才可能生成 880KB 左右的内核，而且后面无需 `make module install` 和 `mkinitcpio` 这两步骤，直接拷贝到/boot 目录，再更新 Grub 即可，可惜它启动时会死机，没有实用价值。另外，在源代码目录下，可以输入 `make help` 可以查看其它的、编译内核时可用的参数。

提示：在 Ubuntu 下没有 /proc/config.gz 文件，却可以 `cp /boot/config-`uname -r`` 文件。也可以执行 `modprobe configs` 来创建。如果想恢复 config.gz 文件，这需要在编译内核时，/proc support 设为 enabled，CONFIG_IKCONFIG 和 CONFIG_IKCONFIG_PROC 都是 yes。

```
General Setup --->
  <*/M> Kernel .config support
    [*] Enable access to .config through /proc/config.gz
```

但是 Ubuntu 下可以：`cp /boot/config-`uname -r``

参考文档：<https://blog.fpmurphy.com/2015/10/what-is-proconfig-gz.html>

<https://help.ubuntu.com/community/Kernel/Compile>

4. 编译内核

在命令行输入 `pacman -S bc`，安装字符界面的计算器，随后的 make 中会用到一次。

在 ubuntu 中，需要安装更多软件：`sudo apt install bison flex bc libncurses-dev libssl-dev`

在命令行输入 `lscpu`，第四行 CPU(s) 就是处理器内核的数量，make 命令的参数 -j 后的数字一般是内核数加一。

```
[root@xautLinux linux-4.19]# lscpu
架构: x86_64
CPU 运行模式: 32-bit, 64-bit
字节序: Little Endian
CPU: 1
在线 CPU 列表: 0
每个核的线程数: 1
每个座的核数: 1
座: 1
NUMA 节点: 1
厂商 ID: GenuineIntel
CPU 系列: 6
型号: 76
型号名称: Intel(R) Celeron(R) CPU J3160 @ 1.60GHz
步进: 4
CPU MHz: 1599.953
BogoMIPS: 3201.23
超管理器厂商: VMware
虚拟化类型: 完全
L1d 缓存: 24K
L1i 缓存: 32K
L2 缓存: 1024K
NUMA 节点0 CPU: 0
标记: fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ss sysca
ll nx rdtscp lm constant_tsc arch_perfmon nopl xtopology tsc_reliable nonstop_tsc cpuid pni pclmulqdq sse3 cx16 sse4_1 sse
4_2 x2apic movbe popcnt tsc_deadline_timer aes rdrand hypervisor lahf_lm 3dnowprefetch pti tsc_adjust smep arat
```

在命令行输入: `make -j 2`，即可开始编译内核源代码。对于双核 CPU 可用 `make -j 3` 加快编译速度。经过十几分钟之后，make 命令运行结束，如果没有出错，就会成功的生成内核文件。

将 make 生成的 64 位的内核文件拷贝到启动目录下：

```
cp -v arch/x86_64/boot/bzImage /boot/vmlinuz-4.19-xaut
```

注意：初次编译内核非常耗费时间，通常少则十几分钟，多则一两个小时，主要取决于所用计算机的运算性能，其次是要编译的模块数量。以后再编译时，只对修改过的那一小部分进行编译，就只需几分钟即可。

提示：建议编译成功后，用 tar 命令将源代码目录打包保存，注意第二个参数是大写“J”，指定后台用 xz 压缩成单一文件：`tar cJvf linux-4.19-prebuilt.tar.xz linux-4.19`。压缩成功后，用 filezilla 传送压缩包到别的计算机上保存。在 Windows 上用 FileZilla Server 建立私人的 FTP 服务器。<https://filezilla-project.org/>

5. 准备模块

输入 `make modules_install`，会根据当前的内核配置文件，自动拷贝驱动模块文件，存入指定目录下。

输入 `ls /lib/modules/`，能看到多出了一个名为 4.19.0-xaut 的目录。

6. 准备启动文件

再生成启动时初始化用的内存虚拟盘：

```
mkinitcpio -k 4.19.0-xaut -c /etc/mkinitcpio.conf -g /boot/initramfs-4.19-xaut.img
```

生成好的文件，会自动拷贝到 /boot 目录下。输入命令 `ls /boot -l`，就能看到多出两个含 4.19 的文件。

7. 配置 Grub 启动管理器

在终端窗口输入 `grub-mkconfig -o /boot/grub/grub.cfg`，会自动生成 grub 启动菜单。

提示：由于会自动扫描所有的硬盘、光盘、软盘等设备，会提示“没有那个设备或地址”，“找不到介质”，请忽略这类无关紧要的错误提示。参考 <https://wiki.archlinux.org/index.php/GRUB#Configuration>

8. 重启进入新内核

在终端窗口输入 `reboot`，重启后就能看到启动菜单多了两行，选择第三行带 xaut 的内核即可启动。