# 1. 流程

1. 下载内核和补丁
   rt补丁下载 https://www.kernel.org/pub/linux/kernel/projects/rt/
   稳定版补丁下载 https://wiki.linuxfoundation.org/realtime/preempt_rt_versions
   内核源码下载 https://mirrors.edge.kernel.org/pub/linux/kernel/

> **Note**
>
> 对于某些内核版本找不到的场合，寻找相近的内核源码就可以，但是要注意到RT补丁与所下载的内核源码版本号需要严格对应。
> 下载好后，将补丁包与解压后的内核源码文件夹放在同一级目录下。

2. 安转依赖

```
sudo apt-get install libncurses5-dev libssl-dev build-essential openssl zlibc libelf-dev minizip libidn11-dev libidn11 bison flex dwarves  libncurses-dev  zstd  -y
```

3. 开始安转
   1. 解压内核源码
      tar -zxvf  linux-5.15.49.tar.gz
   2. 解压补丁
      gunzip patch-5.15.49-rt47.patch.gz
   3. 打补丁
      cd linux-5.15.49
      patch -p1 < ../patch-5.15.49-rt47.patch
   4. 配置内存
      1. cp -v /boot/config-$(uname -r) .config
      2. 这一步是根据当前系统内核的配置设置配置文件， 我们一路enter即可
         make localmodconfig
      3. 进入一个图形界面，有一些地方需要修改
         make menuconfig
         1. 将内核设置为全抢占式的
            General setup - >Preemption Model->Fully Preemptible Kernel(RT)
         2. 关闭内存溢出检测(找不到就忽略此步)
            Kernel hacking —> Memory Debugging —> []Check for stack overflows
         3. 关闭Optimize very unlikely/likely branches(找不到就忽略此步)
            General setup —> Optimize very unlikely/likely branches 子选项Stack Protector buffer overflow detection设置为None，因为编译器不支持更强的堆栈保护
         4. Device Drivers ->[] staging drivers(如果默认开启，按N键取消) (可选择跳过)
         5. -> General setup -> Timers subsystem [*] High Resolution Timer Support
         6. -> General setup -> Timers subsystem -> Timer tick handling (Full dynticks system (tickless)) (X) Full dynticks system (tickless)
         7. -> Processor type and features -> Timer frequency (1000 HZ) (X) 1000 HZ
         8. -> Power management and ACPI options -> CPU Frequency scaling -> Default CPUFreq governor (performance)->performance
   5. 修改当前目录下配置文件./.config
      1. CONFIG_MODULE_SIG_KEY="certs/signing_key.pem"
      2. CONFIG_SYSTEM_TRUSTED_KEYS=""
      3. CONFIG_SYSTEM_BLACKLIST_HASH_LIST=""
      4. CONFIG_SYSTEM_REVOCATION_KEYS=""

> **Example**
>
> 一个ubuntu20.04可用的配置文件
> example

6. 开始编译
    1. sudo make -j核心数
    2. sudo make modules
    3. sudo make bzImage
    4. sudo make modules_install -j核心数
    5. sudo make install
    6. sudo gedit /etc/default/gurb
    7. sudo update-grub
    8. sudo reboot
7. 在引导界面选择刚搞定的内核即可

## 2. 参考博客