

如何在 Ubuntu 和 Linux Mint 下轻松升级 Linux 内核

摘要：这篇文章会向你展示如何使用 GUI 工具 Ukuu 来轻松升级到最新的内核。尽管这篇文章是用 Ubuntu 来测试，但应该也适用于别的基于 Ubuntu 的 Linux 发行版，比如 Linux Mint, elementary OS, Linux Lite 等等。

我假设你已经了解 Linux 内核是什么。Linux 内核是驱动任何 Linux 发行版的核心软件。所有的 Linux 发行版都在他们的核心处适用内核，最上面是 Shell 终端和 GUI 的元素。Linus Torvalds 在 25 年前创造了 Linux 内核，并且直到现在他还在使用的 Linux 内核。

基本上每几个月带有一些新特点(比如支持更多的硬件)并修成了 bug 的 Linux 新内核会发行出来。

你应该手工地升级你的 Linux 到最新的内核吗？



大多数的用户不会自己去升级 Linux 内核，他/她会等新的 Linux 发行版来提供内核升级。实际上，绝大部分的桌面 Linux 用户并不关心他们正使用什么 Linux 内核并且这不是一个坏事情。

事实是当一个新的 Linux 内核发布后，将会需要几周/月的时间，你的 Linux 发行版才会将它提供给你的系统，而且也依赖于这个因素这个 Linux 内核发布是不是一个 LTS(长期支持)。是的，可能你不知道，偶数版本的 Linux 内核发布具有 LTS 和非 LTS 版本。

Linux 发行版会负责你的系统的稳定性，这也是为什么他们不会发布一个新版本的 Linux 内核，除非他们确实在他们的系统上经过了 regression 测试。这能够确保你的系统不会由于硬件的不兼容或任何其他问题而一团糟。

我认为，除非给你提供足够的理由，我们没有真实的需求来升级到一个更新的 Linux 内核。

比如，我使用了一台 Dell XPS 13 Kaby Lake 的 Ubuntu 版本，我发现 Kernel 4.10 将会提供一个针对 Kaby Lake 处理器的性能提升。这会驱使我来升级 Ubuntu 到最新的 Linux 内核，但是根据以往的经验，Ubuntu 不会过早的提供这个内核。在这样的情况下，我可能会选择来手工来升级 Linux 内核。

在 Ubuntu 和 Linux Mint 上轻松升级 Linux 内核

你可以在 Linux 的命令行上使用一些 apt-get 的命令来自己升级 Linux 内核。但是使用一个叫作 Ukuu(Ubuntu 内核升级工具)的图形界面工具来升级内核的过程会更简单和便利。

这个 GUI 工具是由 Tony George 开发的，他为我们提供了一些其他的有用的 Ubuntu 工具，比如 Ubuntu 电池监测，app 备份工具 Aptik，等等。

警告：在我们向你展示如何用 Ukuu 升级 Ubuntu 里面的 Linux 内核，我必须提醒你你应该知道这里面的风险。如果发生一些错误，你有可能恢复到一个过去的内核版本，但你一定不要慌张。确保你将 Ubuntu 系统已经备份。如果你被故障修复阻扰了，避免适用手动升级的方法，可以用你的发行版本的系统升级。

在 Ubuntu 和 Linux Mint 上安装 Ukuu

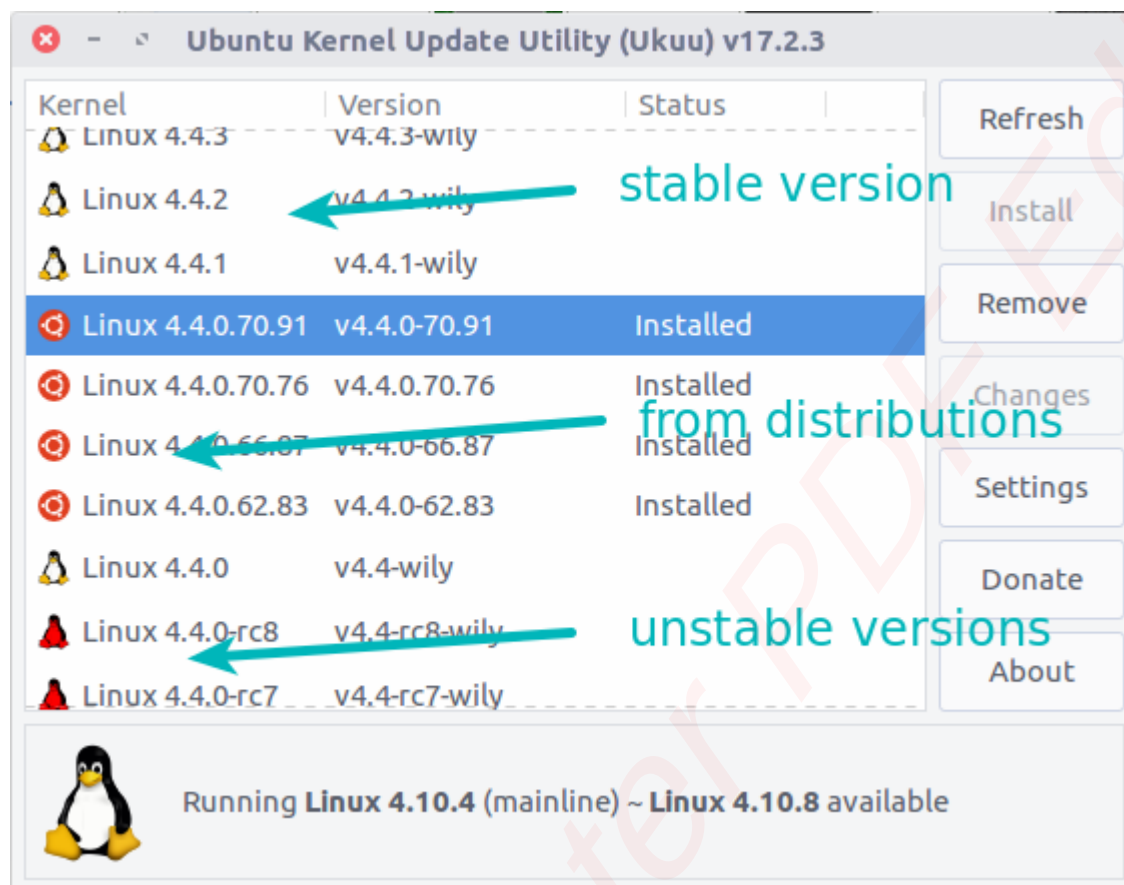
那有一个由开发者提供的官方 PPA 来安装这个工具。使用下面的命令来安装 Ukuu:

```
sudo add-apt-repository ppa:teejee2008/ppa
sudo apt-get update
sudo apt-get install ukuu
```

使用 Ukuu 在 Ubuntu 下安装最新的 Linux 内核

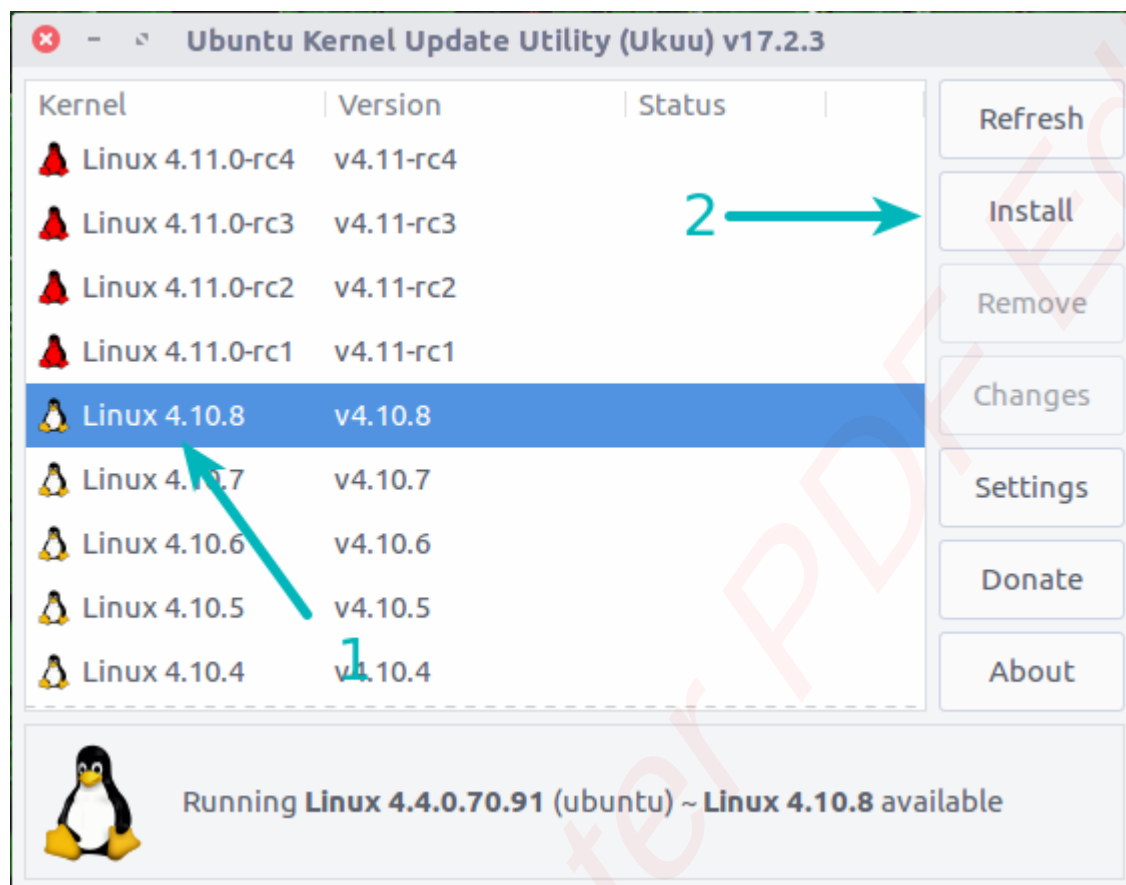
在这个知道里面我是使用 Ubuntu 16.04 来举例的，但是 Ukuu 在别的 Ubuntu 和 Linux Mint 版本里也是可用的。

如果你安装了 Ukuu，启动它。他将会刷新可用的针对 Ubuntu 的 Linux 内核。默认的，它将会向你显示所有可用的内核，包括不稳定的发行内核(会用 RC 和红 Tux 图表标识)。发行版里的内核版本会用 logo 来标识，别的版本仅仅会有完整的就的 Tux logo。



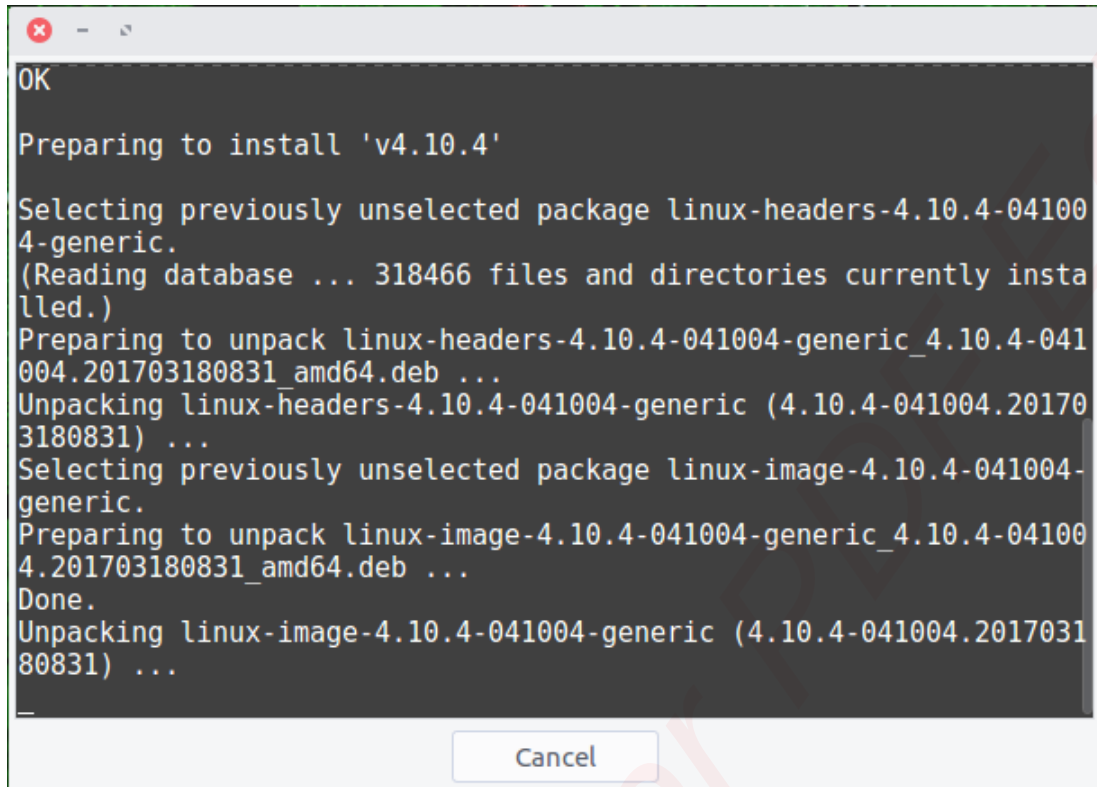
Ukuu 默认显示所有种类的内核

毋庸置疑地，你应该避免使用候选发行版。选择应该使用的版本，点击安装来安装新的 Linux 内核版本。



安装新的 Linux 内核

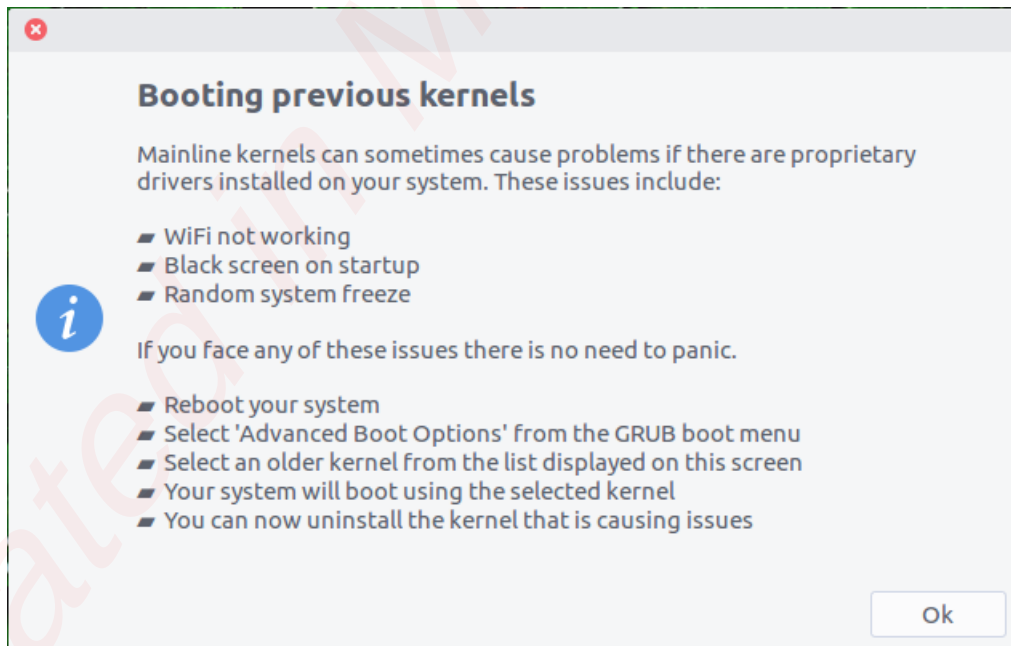
当然，这个动作需要管理员的密码。你一旦输入你的密码，你在这个工具里面看到安装的过程。关注最后的结果，你可以知道新的 Linux 内核是否安装成功了。



安装更新的 Linux 内核

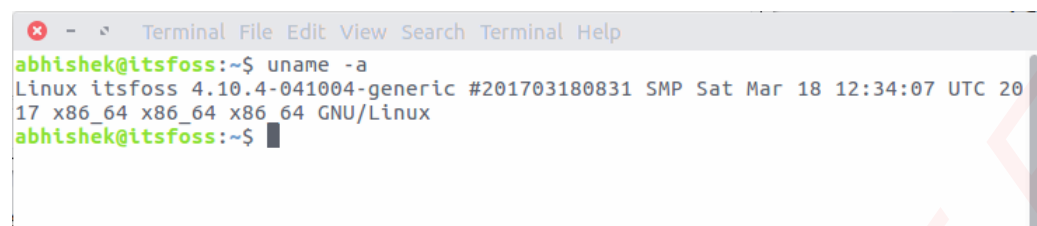
注意：如果这个安装失败了，没有必要恐慌。没有什么可以让系统出错。尝试一个不同的内核版本可能就可以了。

安装结束后，你会看到一个非常实用的屏幕，会告诉你如果新的 Linux 内核有任何错误，你通常可以在 GRUB 的菜单里面选择一个旧的内核并启动。



使用 Ukuu 进行内核升级过程的最后的一些有用信息。

当你下次启动进入系统后，你将会运行你刚刚安装的 Linux 内核。



```
Terminal File Edit View Search Terminal Help
abhishek@itsfoss:~$ uname -a
Linux itsfoss 4.10.4-041004-generic #201703180831 SMP Sat Mar 18 12:34:07 UTC 2017 x86_64 x86_64 x86_64 GNU/Linux
abhishek@itsfoss:~$
```

在 Ubuntu 里面检测 Linux 内核版本

这儿需要注意的一个事情是安装一个新的内核并不意味着旧的内核已经从系统里面移除了。旧的内核可以由你随意支配使用。默认地，Ubuntu 会启动进安装到系统上的最新的 linux 内核。

撤回变化/降级 Linux 内核

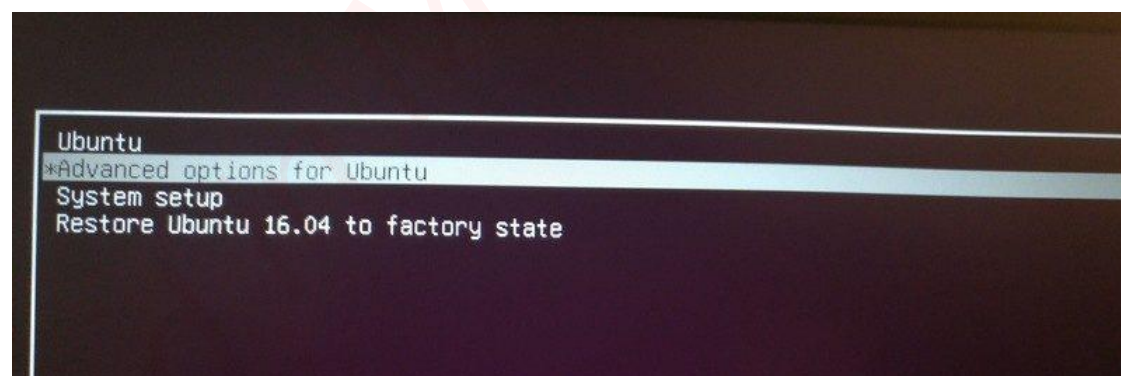
如果你不喜欢新的 Linux 内核，或者你发现它有一些问题，你可以很容易来降级内核。可以通过下面两个步骤来完成。

- 启动进入旧的内核
- 移除你不想要的 Linux 内核

让我们看如何来做吧。

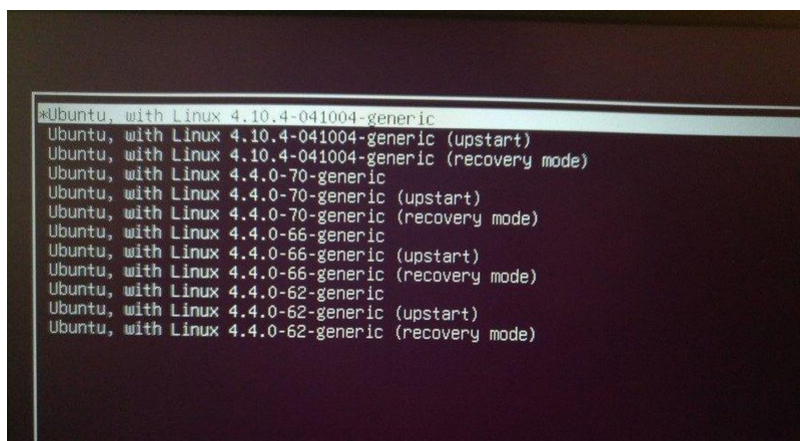
步骤 1：启动进旧的 Linux 内核

当你启动你的系统，在 grub 的菜单上，选择 Ubuntu 的高级选项。



选择 Ubuntu 的高级选项

从这儿你可以看到安装在你的系统上的所有的 Linux 内核，选择一个旧的。不要选择 upstart 或者 recovery 模式，选择正常的。我会在另一篇文章里单独来讨论。

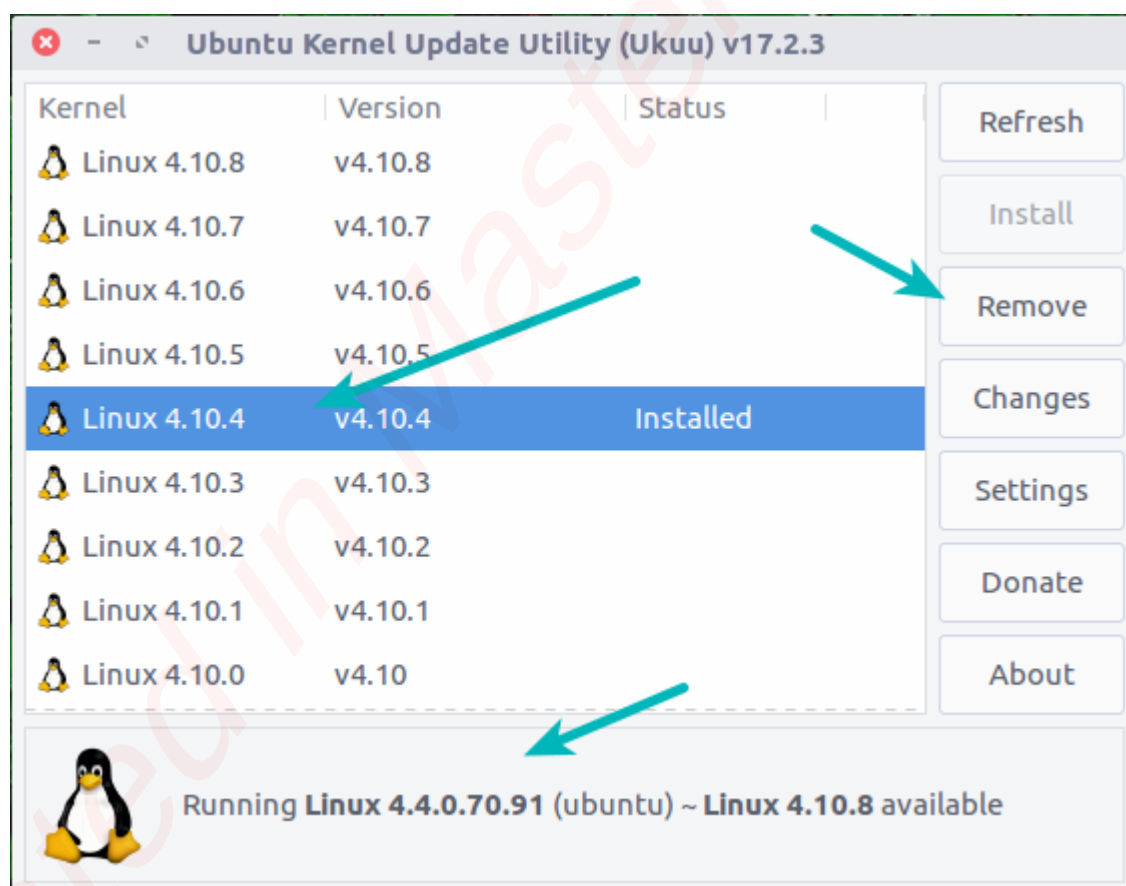


启动进 Ubuntu 里面的一个旧的 Linux 内核

步骤 2: 回退 Linux 内核

你用旧的 Linux 内核来启动你的系统后,再次启动 Ukuu。确保你没有删除你正在使用的内核。

选择你不再需要的新内核然后点删除。

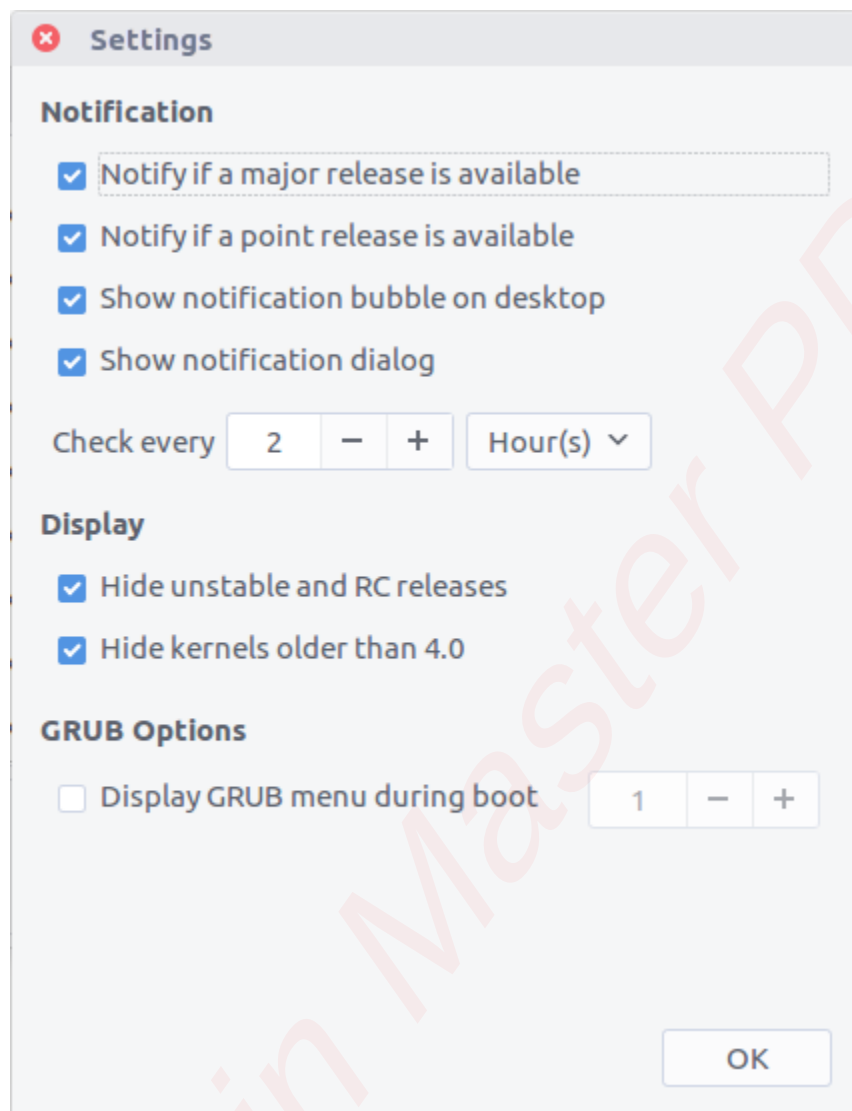


使用 Ukuu 来移除 Linux 内核

这就是你所需要做的所有部分来 downgrade Ubuntu 的 Linux 内核。

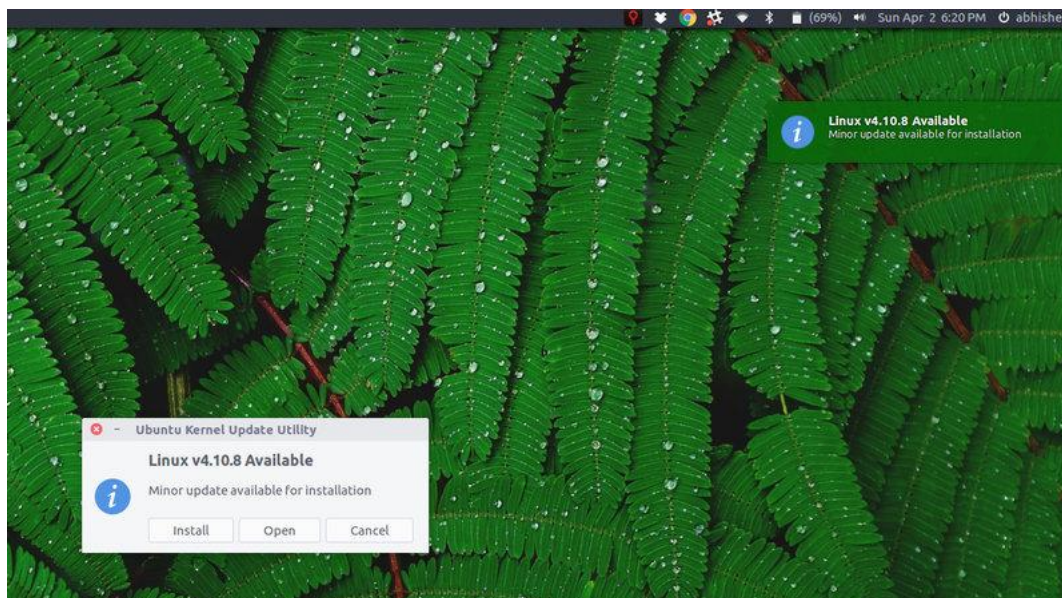
Ukuu 的一些别的特性

当我们讨论这个的时候，我想指出 Ukuu 的更多的一些特性。Ukuu 具有一个设置选项可用让你来不显示列表里面的候选内核。你也可以隐藏早于 4.0 之前的 Linux 内核版本。



Ukuu 里的一些额外的选项

你也可以选择如果有新的 Linux 内核可用就有桌面通知的选项。



Desktop notification for new Linux kernel availability

移除 Ukuu

为了移除 Ukuu，你必须根据下面的程序来从 Ubuntu 移除一个 PPA，使用下面的命令即可：

```
sudo apt-get remove ukuu
```

从软件的源列表里面移除 PPA，使用下面的命令：

```
sudo add-apt-repository --remove ppa:teejee2008/ppa
```

就这些。

你如何来升级你的 Linux 内核？

Ukuu 是一个不错的图形工具用来在基于 Ubuntu 的 Linux 发行版上升级 Linux 的内核。它用起来像一个魔法并且会专注于它的用途。我希望这个指导手册是有用的，能向你演示如何轻松升级 Linux 内核。

所以，你是否经常自己升级 Linux 内核，或者你会等你的发行版来提供升级？你是怎么做的呢？

嵌入式资源免费下载

总线协议:

1. [基于 PCIe 驱动程序的数据传输卡 DMA 传输](#)
2. [基于 PCIe 总线协议的设备驱动开发](#)
3. [CANopen 协议介绍](#)
4. [基于 PXI 总线 RS422 数据通信卡 WDM 驱动程序设计](#)
5. [FPGA 实现 PCIe 总线 DMA 设计](#)
6. [PCI Express 协议实现与验证](#)
7. [VPX 总线技术及其实现](#)
8. [基于 Xilinx FPGA 的 PCIE 接口实现](#)
9. [基于 PCI 总线的 GPS 授时卡设计](#)
10. [基于 CPCI 标准的 6U 信号处理平台的设计](#)
11. [USB30 电路保护](#)
12. [USB30 协议分析与框架设计](#)
13. [USB 30 中的 CRC 校验原理及实现](#)
14. [基于 CPLD 的 UART 设计](#)
15. [IPMI 在 VPX 系统中的应用与设计](#)
16. [基于 CPCI 总线的 PMC 载板设计](#)
17. [基于 VPX 总线的工件台运动控制系统研究与开发](#)

VxWorks:

1. [基于 VxWorks 的多任务程序设计](#)
2. [基于 VxWorks 的数据采集存储装置设计](#)
3. [Flash 文件系统分析及其在 VxWorks 中的实现](#)
4. [VxWorks 多任务编程中的异常研究](#)
5. [VxWorks 应用技巧两例](#)
6. [一种基于 VxWorks 的飞行仿真实时管理系统](#)
7. [在 VxWorks 系统中使用 TrueType 字库](#)
8. [基于 FreeType 的 VxWorks 中文显示方案](#)
9. [基于 Tilcon 的 VxWorks 简单动画开发](#)
10. [基于 Tilcon 的某武器显控系统界面设计](#)
11. [基于 Tilcon 的综合导航信息处理装置界面设计](#)

12. [VxWorks 的内存配置和管理](#)
13. [基于 VxWorks 系统的 PCI 配置与应用](#)
14. [基于 MPC8270 的 VxWorks BSP 的移植](#)
15. [Bootrom 功能改进经验谈](#)
16. [基于 VxWorks 嵌入式系统的中文平台研究与实现](#)
17. [VxBus 的 A429 接口驱动](#)
18. [基于 VxBus 和 MPC8569E 千兆网驱动开发和实现](#)
19. [一种基于 vxBus 的 PPC 与 FPGA 高速互联的驱动设计方法](#)
20. [基于 VxBus 的设备驱动开发](#)

Linux:

1. [Linux 程序设计第三版及源代码](#)
2. [NAND FLASH 文件系统的设计与实现](#)
3. [多通道串行通信设备的 Linux 驱动程序实现](#)
4. [Zsh 开发指南-数组](#)
5. [常用 GDB 命令中文速览](#)
6. [嵌入式 C 进阶之道](#)
7. [Linux 串口编程实例](#)
8. [基于 Yocto Project 的嵌入式应用设计](#)
9. [Android 应用的反编译](#)
10. [基于 Android 行为的加密应用系统研究](#)
11. [嵌入式 Linux 系统移植步步通](#)
12. [嵌入式 CC++ 语言精华文章集锦](#)
13. [基于 Linux 的高性能服务器端的设计与研究](#)
14. [S3C6410 移植 Android 内核](#)
15. [Android 开发指南中文版](#)
16. [图解 Linux 操作系统架构设计与实现原理（第二版）](#)

Windows CE:

1. [Windows CE.NET 下 YAFFS 文件系统 NAND Flash 驱动程序设计](#)
2. [Windows CE 的 CAN 总线驱动程序设计](#)
3. [基于 Windows CE.NET 的 ADC 驱动程序实现与应用的研究](#)
4. [基于 Windows CE.NET 平台的串行通信实现](#)
5. [基于 Windows CE.NET 下的 GPRS 模块的研究与开发](#)
6. [win2k 下 NTFS 分区用 ntldr 加载进 dos 源代码](#)

7. [Windows 下的 USB 设备驱动程序开发](#)
8. [WinCE 的大容量程控数据传输解决方案设计](#)
9. [WinCE6.0 安装开发详解](#)
10. [DOS 下仿 Windows 的自带计算器程序 C 源码](#)
11. [G726 局域网语音通话程序和源代码](#)
12. [WinCE 主板加载第三方驱动程序的方法](#)
13. [WinCE 下的注册表编辑程序和源代码](#)
14. [WinCE 串口通信源代码](#)
15. [WINCE 的 SD 卡程序\[可实现读写的源码\]](#)
16. [基于 WinCE 的 BootLoader 研究](#)

PowerPC:

1. [Freescale MPC8536 开发板原理图](#)
2. [基于 MPC8548E 的固件设计](#)
3. [基于 MPC8548E 的嵌入式数据处理系统设计](#)
4. [基于 PowerPC 嵌入式网络通信平台的实现](#)
5. [PowerPC 在车辆显控系统中的应用](#)
6. [基于 PowerPC 的单板计算机的设计](#)
7. [用 PowerPC860 实现 FPGA 配置](#)

ARM:

1. [基于 DiskOnChip 2000 的驱动程序设计及应用](#)
2. [基于 ARM 体系的 PC-104 总线设计](#)
3. [基于 ARM 的嵌入式系统中断处理机制研究](#)
4. [设计 ARM 的中断处理](#)
5. [基于 ARM 的数据采集系统并行总线的驱动设计](#)
6. [S3C2410 下的 TFT LCD 驱动源码](#)
7. [STM32 SD 卡移植 FATFS 文件系统源码](#)
8. [STM32 ADC 多通道源码](#)
9. [ARM Linux 在 EP7312 上的移植](#)
10. [ARM 经典 300 问](#)
11. [基于 S5PV210 的频谱监测设备嵌入式系统设计与实现](#)

12. [Uboot 中 start.S 源码的指令级的详尽解析](#)

Hardware:

1. [DSP 电源的典型设计](#)
2. [高频脉冲电源设计](#)
3. [电源的综合保护设计](#)
4. [任意波形电源的设计](#)
5. [高速 PCB 信号完整性分析及应用](#)
6. [DM642 高速图像采集系统的电磁干扰设计](#)
7. [使用 COMExpress Nano 工控板实现 IP 调度设备](#)