



GNU/Linux for daily use

spinach/hehelego

March 14, 2021

这是什么

Linux 日常使用、开发环境、服务维护的相关教程、书籍、博客等资料已经非常多子遍地都是, 随便一搜就是一大把. 不要给 geekpie-workshop 灌水啊

这是一个关于 GNU/Linux OS 的基本介绍, 主要面向没有 Linux 使用经验, 想要学习或使用它 / 虽然了解 Linux 但是并未将其做为主力 OS 的人.

诚然,Linux 相关的资料、讨论、布道, 在国内国外、校内校外、线上线下已经非常多了. 比如ArchLinux wiki, Unix StackExchange, USTC linux101, TL;DR pages; 以及TheCW 在 B 站投稿的一些视频, MiraculousMoon 写的 manjaro-kde 安装与美化.

所以, 我做这次分享的目的并非在于丰富他们, 而是在于启发人们, 让人们敢于使用 GNU/Linux 为其日常主力操作系统, 并讲解基础使用技巧.

有些东西并不是那么难, 只是你需要一个契机.
让你意识到: 它很好玩, 开始玩它并没有什么门槛.
给你想要开始玩它的冲动. 这就是我要做的.

这次分享中不会有的东西

- 如何安装一个 GNU/Linux 发行版, 完成基础的配置¹
- 如何自己定制, 编译并安装 linux kernel²
- 详细的 GNU/Linux 开发, 运维指南³
- android,embedded OS,domain-specific OS⁴
- 如何在 GNU/Linux 环境下进行 iOS 开发, 进行 win32 开发
- 传教/圣战 你们啊, 不要总想着搞个大新闻, 把我批判一番

这次分享中包含的内容

- 对 GNU/Linux 的架构, 社区, 软件生态的简要介绍
- 为什么 GNU/Linux 可以满足你的日常使用需求
- 虽然不完全正确但并非错得离谱的入坑指南

¹如果用户文档讲不清楚, 那么它将没有用户
²这不是能提升日常使用体验的操作
³同上, 这次分享专注于日用
⁴它们确实基于 Linux kernel, 但并非 GNU/Linux 操作系统

Contents

1	GNU/Linux, a brief introduction	3
2	GNU/Linux, for daily use	4
3	GNU/Linux, beginner's guide	5
3.1	尝鲜体验	5
3.2	安装与配置	5
3.3	获取帮助与支持	5
3.4	软件生态	7
4	random thoughts on FOSS	8
4.1	FOSS, not only open source	8
4.2	FOSS, community driven	8
4.3	FOSS, contributing	8
A	reference & resources	9

1 GNU/Linux, a brief introduction

GNU/Linux 是目前唯一同时符合⁵end user& content creator&developer& system admin 使用需求的 system⁶

早都知道的历史故事

Ken Thompson&Dennis Ritchie 开发了 UNIX.
起初授权宽松快速流行. 后来 AT&T 变更授权, 打了多年官司...
RMS 发起 GNU 和 free software movement.
geek 们写出了 GCC,make,Emacs 等软件, 建立了 FSF 并起草 GPL.
芬兰网友 Linus Torvalds 写了个 GPL 授权的 kernel.
Linux kernel 和 GNU 已有生态结合,GNU/Linux 出现.
社区驱动模式和企业支持使 Linux 发展壮大.
开源文化和开源组织出现, 走向 IT 产业的更多分支甚至其他行业.

不会做为考点的操作系统层次

最底层的是 hardware 和上面的 firmware, 然后是 driver.
OS 会抽象硬件资源为虚拟软件资源管理他们并提供访问虚拟资源的接口.
kernel 还会管理进程, 控制其状态转换, 管理调度, 并提供 IPC.
之后是 shell, 例如我们熟悉的 bash,fish 和 gnome-shell,plasma-shell.
最高层的是 applications. 而用户就直接与 shell,app 交互⁷.

User mode	User applications	bash, LibreOffice, GIMP, Blender, 0 A.D., Mozilla Firefox, ...				
	System components	Daemons: systemd, logind, networkd, PulseAudio, ...	Window manager: X11, Wayland	Graphics: Mesa, AMD Catalyst, ...	Other libraries: GTK+, Qt, EFL, SDL, SFML, FLTK, GNUstep, ...	
	C standard library	open(), exec(), sbrk(), socket(), fopen(), calloc(), ... (up to 2000 subroutines) glibc aims to be fast, musl and uClibc target embedded systems, bionic written for Android, etc. All aim to be POSIX/SUS-compatible.				
Kernel mode	Linux kernel	stat, splice, dup, read, open, ioctl, write, mmap, close, exit, etc. (about 380 system calls) The Linux kernel System Call Interface (SCI, aims to be POSIX/SUS-compatible) ^[citation needed]				
		Process scheduling subsystem	IPC subsystem	Memory management subsystem	Virtual files subsystem	Network subsystem
		Other components: ALSA, DRI, evdev, LVM, device mapper, Linux Network Scheduler, Netfilter Linux Security Modules: SELinux, TOMOYO, AppArmor, Smack				
Hardware (CPU, main memory, data storage devices, etc.)						

很无聊啊, 这关我锤子事

⁵正如你所想的那样, 这是我钦定的
⁶不是指 operating system, 而是整个计算机系统
⁷这里可能有些 confusing, 请自行了解 shell, tty, pty, terminal emulator 的信息

2 GNU/Linux, for daily use

为什么 (现在) 你会想要使用 GNU/Linux

在上一个 section 中, 我们钦定 GNU/Linux 是适合日常办公学习; 并且, 可以在其上轻松搭建开发者, 内容创作者, 系统维护者所需的环境.

- 100 percent controllable
- 100 percent customizable
- embrace FOSS,community driven,decentralized
- for fun, for working efficiently, or for anything.

为什么 (现在) 你可以完全使用 GNU/Linux

- 靠谱且稳定的 kernel
- 完备的软件生态
- 友好、强大的社区 + 企业支持

已经是 2021 年. 不是那个 canonical 给全世界的网友邮寄光盘的年代; 不是那个在 slackware 上面自己管理 PHP,Apache http server,MySQL DB 依赖的年代; 不是那个每天 kernel panic 的年代; 不是那个资料查不到, 问人问不到, 只能去邮件组里面被人喷, 完全没有中文社区年代; 不是那个分辨调节不了, 天天屏幕撕裂的年代.

安装一个 GNU/Linux 发行版现在十分简单⁸; 包管理器发展为强大的全系统维护工具, 硬件驱动、字体、开发环境、GUI 应用等等都可以无脑安装/更新/替换/卸载; kernel 越来越稳定且高效, 反倒是隔壁 windows10 天天蓝屏; 桌面环境现在开箱即用, 且定制性和集成度都有了巨大飞跃, 不少发行版可以做到无 tty, 无 termial emulator 使用; 跨平台技术和基于 web 的技术大势所趋, 专业内容生产创作或日常 office 办公, 甚至影音娱乐, 都有了足够的软件支持; arch wiki/gentoo wiki 多年积累, 成为 linux 世界的 comprehensive unofficial user manual+trouble shooting guide; 有了 github/gitlab 等靠谱托管开发平台, 开发者用户维护者交流成本大幅下降; Linux 使用的论坛不再冷清, 中文社区也有了足够的用户基础⁹.

好家伙, 比某厂手机营销还离谱. 我们不妨看一看网友们的使用经历, 以此, 初步检验现在完全使用 GNU/Linux 的可行性. 我在 2020 年初开始使用 Linux 环境工作, 在 VirtualBox 中运行 manjaro, 并使用 msys2 模拟 linux 环境. 到 6 月时, 我确信自己积累了足够的经验, 于是把 manjaro 安装到了物理机上. 后来切换到了 arch. 遇到过一些小麻烦, 但都成功解决, 稳定使用至今¹⁰.

为了学会开车, 你并不需要了解内燃机工作原理和制造工艺

⁸gentoo 这种需要 compile the whole world 的 distro 除外

⁹我说的当然不是 CSDN

¹⁰虽然目前没有出大问题, 但定期备份还是要做的

3 GNU/Linux, beginner’s guide

3.1 尝鲜体验

TODO: 个一些视频, 真机现场演示.

3.2 安装与配置

TODO: 一个物理机安装视频.

3.3 获取帮助与支持

讲解正确地搜索与提问方式. 或者说, 出了锅怎么修

我想 trouble shooting 中, 最重要的是敢乱搞、敢提问的自信以及能全面认真地查阅资料的谦虚. 当然, 也需要一些技术水准, 以解决那些前人未发现、未明确或为解决的问题.
必须说明的是, 尽管 arch wiki 中文翻译做得不错, 中文社区现在初具规模, 使用英文仍是不可避免的.

step 1: 尝试复现问题, 形成初步问题描述

尝试复现问题的同时, 也要尽量定位问题来源, 确定问题的严重性.
清楚的问题描述和初步的定位, 可以帮助你更有效的进行搜索, 以了解潜在的解决方案或者进一步定位问题.

做问题复现请务必记住: 步骤最简, 环境一致.
在这个过程中, 你会对问题原因有初步认知, 比如是 user space 的问题还是 kernel 的问题, 是那个软件包/服务的问题, 是否与硬件平台等相关信息. 它们非常重要.
至于问题严重性的, 新手常常会误判问题严重性. 正如你所想, 常常会高估问题严重性... 像 kernel 和 fallback image 同时挂了, 没法 boot; partial upgrade 导致 system break; grub 无法引导等等, 是很严重的问题, 而什么进不了桌面环境, 屏幕撕裂了, 某个配置不生效, 某应用启动失败, 是不那么严重的问题...

step 2: 尝试自己 workaround

当然, 如果问题严重, 别乱搞... 转移工作空间的数据要紧.
具体而言: 用最新的 arch iso 做个 live USB, boot 到 liveCD 环境; 用外挂存储设备/网络连接拿走你的工作环境.

step 3: (二手信息) 查阅 arch wiki/gentoo wiki

事实上, 多数你遇到的锅 (不论什么发行版, 什么软件环境, 什么硬件平台), 是被相当多的人发现, 被前人准确定位, 并找到了合适解决方案的. 这些信息被记录在各种神奇的地方, 当然 arch wiki 集中了 most of them
arch wiki/gentoo wiki 可以说无所不包, 别看顶着 arch/gentoo 发行版的名字, 实际上包含整个 GNU/Linux 生态的信息, 甚至还有不少硬件相关的信息. 它们由用户群自发创建、自发维护, 以及翻译. arch wiki 是非常亲切的, 因为记录下那些信息的人和你我一样, 都是常规用户, 用户最清楚用户需要什么.

step 3: (二手信息) 使用搜索引擎

如果你不知道去 wiki 上面的什么词条找方案: 那么你应该先用 step1 中的信息进行搜索, 以更好地定位问题, 然后回到 wiki.
如果认真仔细阅读了所有相关 wiki 词条, 仍然没有解决问题: 那么你应当扩大信息搜集范围, 利用搜索引擎找 user manual, blog, github issue, forum 中的相关信息.

搜索引擎使用技巧... 我懂得不多, 只能给出一些零星的 idea.

- 使用关键词, 而不是整句, 防止 SE 的 NLP 错误理解问题;
- 使用必须出现/必须不出某关键词的约束;
- 使用 and/or 逻辑约束;
- 使用 site,title 约束缩小范围;
- 一次搜索很难解决问题, 往往需要分析查看结果, 调整检索式, 再次搜索, 多次重复;
- 注意语言的使用, 多数情况下, 英文资料最多最全, 但中国特供的产品/仅在中国流行的东西, 显然要用中文, 必要时也可配合翻译工具搜索日文, 德文等语种的资料.

step 3: 查官方文档

有些过于细节的问题, 或者过于繁琐的问题, 需要在官方性质的文档中查阅. wiki 不会搬运这种东西.

step 4: 问人

描述问题, 给出环境和复现方式; 描述你查阅到的相关信息, 和进行的尝试. 注意排版... 有大段 log/code 可以使用 ubuntu pastebin. 然后, 耐心等待, 耐心和愿意提供帮助者进一步交流情况. 并且记住, 没有人有义务帮助你; 以及, 你时常会遇到当前不存在较好解决方案的问题, 甚至是目前没有明确原因的问题.

为什么人们 (尤其是开发者和维护者) 讨厌伸手党? 因为它们比起数量增长的推广和扩张, 更关心功能改进、问题修复、稳定性提升和技术创新的质量提升. 而复制粘贴解决方案给伸手党, 最多是拉拢一些不可能成长为贡献者的初级用户.

step 5: 自己动手, 丰衣足食

到了这里, 你可能进入无人区了. 你遇到的问题, 可能是未发现/未明确/未解决状态. 具体是那种得去上游找. 请首先想办法 workaround, 以继续正常工作. 比如 “问题不大, 可以忍”, “从某个路径下、在启动某个服务之后运行它, 就不会有问题”, “写一个脚本 wrap 它, hook 一些 syscall 做些特殊处理”. 之后找上游¹¹看看问题是属于那种情况:

- 1. 已经修复, 尚未跟进
 - 2. 已经被发现, 尚未修复
 - 3. 没有被遇到过/没有被报告
- 1 请联系发行版的软件包打包者.
2 可以汇报问题再现的情况 其实就是催更.
3 请上报给开发者. 可能是 github/gitlab 发 issue, bugzilla 汇报, 也可能是其他渠道, 比如 trac ticket. 具体如何上报问题一般会在 user manual / how to contribute 文档中描述.
当然如果您能写个 patch 修锅, 或者 fork 出去修掉再 pr/mr 回去最好.

step ∞: 记录与分享

我们进行 trouble shooting 用到的资料, 都是其他用户无偿分享的. 所以做为回报, 我们也要做对等的记录与分享. 比如写一篇对搜索引擎友好的 blog, 让其他人在搜索时, 能够少走弯路. 或是帮助完善 arch wiki, 使它更加完备, 维持其 “a comprehensive unofficial guide for GNU/Linux” 的作用.

¹¹指软件开发者 / 硬件、驱动、固件提供商

3.4 软件生态

更全面的介绍见 arch wiki 的两个词条^{1,2}.
以及桌面环境配套应用[KDE](#) [Gnome](#).

- 选择 KDE plasma5 或者 Gnome3 获得完整的桌面环境.
也可以用精简的 i3,sway,bspwm 等窗口管理器
- 想要安装新软件, 却不会用包管理器的 CLI?
试试 KDE 下的 discover 或者 Gnome 下的 Gnome software center.
- 都用上 GNU/Linux, 总不能还微软雅黑 + 时代新罗马吧¹².
试试 google noto,adobe source 等开源字体吧.
在 KDE/Gnome 中配置字体非常容易, 不需要手写 fontconfig.
- 安装 fcitx5-im 组 (包含 fcitx5,fcitx5-qt,fcitx5-gtk,fcitx5-configtool),
之后按照 wiki 指示安装你喜欢的输入法并进行配置.
- 浏览器当然选择 firefox.¹³
- 什么时候有 SIST seminar? 需要给老板发邮件?outlook 没了?
安装 kmail/evolution 获得邮件与日程管理¹⁴
- 聊天社交使用 telegram,IRC,tox. 腾讯家的 wechat,qq 比较难搞.
- 安装 LibreOffice / OpenOffice 代替 MS office.¹⁵ 没了拼写纠错? 安装 aspell,hunspell 和语言包就行啦.
- 使用 okular 阅读多种文档; 配合 pandoc 进行文档格式转换.
- 安装 vlc 获得多媒体支持¹⁶, 甚至还有你熟悉的网易云音乐
图片/相册管理使用 EOE/Gwenview.
- 懒得自己做数学题? sagemath 提供完整数学环境,octave 替代 MATLAB core; 还有你熟悉的 scipy 以及 R.
- 备份和同步 timeshift,syncthing. 当然也有 dropbox,onedrive,google drive.
- XX 手机助手没了? 使用 screpy + kde-connect 来融移动平台和桌面平台.
- Wine+Qemu(with KVM) 解决 Linux 无法使用的软件 腾讯全家桶;~~steam;~~ ~~垃圾~~ ~~multisim~~.
- 编辑器当然用 vim/Emacs. 懒得自己配置用 spaceVim/spaceEmacs.
也可以试试你们熟悉的 VS code 与 typora.
- 有专业需求? 是一名软件开发者, 内容创作者, 或是科研人员.
参考 arch wiki list of applications, 你们熟悉的环境, 这里都有.
- 跨过长城, 走向世界? 你熟悉的暗影袜子/第二光线都可以轻松搭建.
- 夹带私货 一些有趣的 TUI/CLI 应用, 轻松帮你工作效率翻倍:
 - 基础工具: core-utils,gnu-toolchain,git,tmux,openssh,vim...
 - 包管理器: arch 系使用 pacman;ubuntu 使用 apt;RPM 系使用 dnf.
 - init 系统: systemd 或者 open rc, 进行服务管理/定时任务/hook 脚本.
 - fish,zsh: 更友好的 shell, 提供补全、纠错、快速跳转等特性, 吊打 bash.
对于脚本编程, 我推荐 python 而非任何一种 shell.
 - nmcli,iwctl: network manager 和 iwctl 的前端, 快速管理网络连接.
 - tldr¹⁷: 帮助文档速查, 部分代替 man pages.
 - rsync: 远程同步工具. 还可以用 vim+rsync 干掉 VSC remote dev.
 - ranger,broot: TUI 文件管理器.
 - FuzzyFileFinder(fzf)+theSilverSearch(ag)+ripgrep(rg)
 - curl: 访问 HTTP/FTP/redis/memcached.
 - aria2: 多线程下载器, 支持断点续传.
 - proxychains: 环境变量 http_proxy,socks5_proxy 不生效, 怎么办.
 - pandoc: 构建不同文档格式为顶点的完全图
 - aspell: 拼写检查工具.
 - rofi: 最棒的 app launcher.

¹²MS font 可以搞, 但是没必要

¹³chrome,chromnium,edge,brave 可选, 但是我不推荐它

¹⁴evolution 支持 exchange 协议

¹⁵WPS office 有原生 Linux 版,MS office 可以通过 wine 完美运行, 但是我不推荐它

¹⁶常规的 codec,ffmpeg 等依赖不需要手动安装, 会被包管理器安排好

¹⁷too long, don't read. 或者说太长不看

4 random thoughts on FOSS

我不想讨论 fres software 和 open source 的区别, 不想讲解 RMS 和 FSF 的执念, 也不想画大饼构想 totally free and open 的未来. 我想, 当前人类还未有物质极大丰富而走向共产主义, 自由软件运动也仅仅是初级阶段. 不如现实一点, 实用意义大于意识形态, 只要我们心中还有自由并且用实际行动促进开放, FOSS 就不会死去.

4.1 FOSS, not only open source

继续之前, 先来谈谈自由软件和开源软件
首先是自由软件,GNU 定义的 free software: respects users’ four essential freedoms. 其着眼点在于用户的自由权利.

- 0 The freedom to run the program as you wish, for any purpose.
- 1. The freedom to study how the program works, and change it so it does your computing as you wish.
- 2. The freedom to redistribute copies so you can help others.
- 3. The freedom to distribute copies of your modified versions to others.
By doing this you can give the whole community a chance to benefit from your changes.

而开源软件, 要求使用 open source licenses¹⁸授权. 任何人都可以获得开源软件, 它注重软件透明可信和社区协作的可能性.

4.2 FOSS, community driven

什么是社区? 就是用户 + 开发者 + 维护者的人群与信息交流公开渠道 + 组织¹⁹.
GNU/Linux 的社区乃至整个 FOSS 世界的特点, 我认为有这样三点: 自由, 开放, 对等.

4.3 FOSS, contributing

社区特点中的对等, 是指用户, 开发者, 维护者这三个角色没有实质区别.
我们应当意识到 FOSS 来之不易, 且其发展面临压力, 所以用户应当做开发者和维护者的工作. 而此外,“使用”本身也可以是一种贡献

- 仍然有人在使用他们的软件, 他们的软件将来也会有更多的用户.
用户是开发者和维护者继续贡献的动力来源之一.
- 用户在使用中会发现 bug, 定位问题, 并且可能辅助修复; 使用者也会提供强化功能和优化体验的建议甚至解决方案.
用户是 FOSS 进步的来源之一.

¹⁸例如 GPL,Apache license,MIT Lincense
¹⁹我们还是不能太理想化, 现实一点, 这里不可避免地会有企业巨头和资本的影响.
看看 kernel 代码贡献来源就知道了

A reference & resources

本次分享的参考资料

- [USTC linux101](#)
- [GNU official site](#)
- [OpenSourceInitiative](#)
- [CreativeCommons lincense](#)
- [distrowatch](#)
- wikipedia: [UNIX](#),[Linux](#)

可以随便看看, 找找有趣东西的视频

- [拉我入坑的 TheCW at bilibili](#)
- [Houge Langley 的乐享 linux 生活/linux 头脑风暴系列视频](#)
- [brain 的 linux 体验视频](#)

获取发行版的 ISO 镜像

- 各种发行版的官网: [ubuntu](#),[manjaro](#),[fedora](#),[Pop!_OS...](#)
- 国内的镜像、反代站: [清华的 Tuna mirrors](#),[中科大镜像站](#),[张江理工镜像站](#)

虚拟机, POSIX 兼容层, win32 兼容层

- [Qemu](#); [VirtualBox](#)
- [Cygwin](#) [msys2](#)²⁰,[WSL](#), [wine](#)

出了问题去哪里问

1. ask your self / try to fix it by yourself
2. [bing/google/duckduckgo](#) search engine
3. Arch Linux wiki; [arch wiki:FAQ](#)
4. the software's official documentation; man page; tldr pages; github.
5. ArchLinux forum& Unix Stackexchange& Ask Ubuntu& Stack Overflow ; telegram 群组,IRC 群聊, 邮件组.

²⁰[msys2](#) 早就可以完全取代 [Cygwin](#) 并且比其做得更好了