

Linux 操作系统构建基础

第 1 课 Linux 操作系统构建简介

徐小东

统信软件

2022-10-10

课程目录

- ① Linux、GNU 与开源
- ② 编译器、链接器及如何构建软件
- ③ Linux 操作系统的构建方法
- ④ 补充阅读材料
- ⑤ 参考资料

Section 1

Linux、GNU 与开源

Linux

- ▶ 狭义
Linux kernel¹ (Linux 内核), 一种开源的类 Unix 操作系统宏内核
- ▶ 广义
基于 Linux 内核的操作系统
- ▶ 最初的开发者
Linus Torvalds (林纳斯·托瓦兹)
- ▶ 第一个公开版本
1991 年 9 月 17 日, 0.01 版
- ▶ 吉祥物
一只黄嘴的企鹅, 叫 Tux

¹<https://kernel.org>



图 1: Linus Torvalds

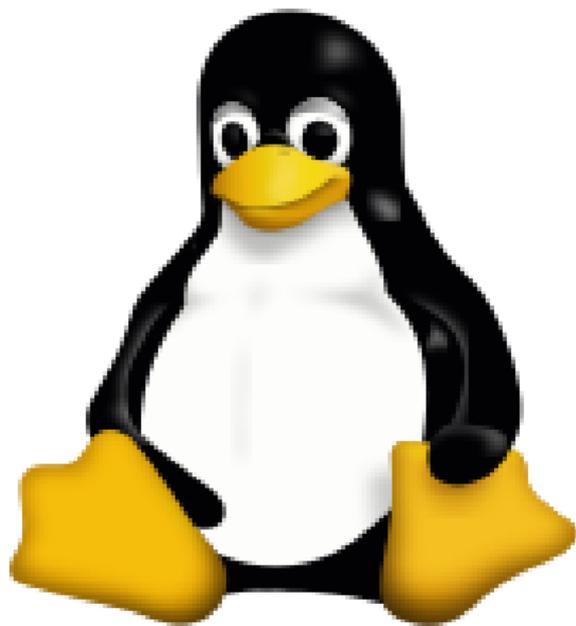


图 2: Tux

GNU

- ▶ GNU²
是 GNU's Not Unix! 的递归缩写
- ▶ 发起时间及人物
1983 年, Richard Stallman (理查德·斯托曼)
- ▶ 目标
提供一个和 Unix 兼容的 100% 自由软件的操作系统
- ▶ 支持方及用户权利
自由软件基金会, 通过 GPL (GNU General Public License, GNU 通用公共许可证) 授予程序的用户自由软件定义的权利
- ▶ 代表软件
GCC、Emacs、glibc、bash 等

²<https://www.gnu.org>



图 3: Richard Stallman



图 4: GNU Head

开源

- ▶ Open source, 开源
即源代码可自由使用以进行修改和重新分发
- ▶ Open-source software (OSS), 开源软件
根据许可协议发布的计算机软件, 在该许可协议中, 版权持有人授予用户使用、研究、更改和向任何人或为任何目的发布软件及其源代码的权利
- ▶ 开源模式
鼓励开放协作的去中心化软件开发模式, 埃里克·雷蒙称为“集市模式”

Linux 发行版

- ▶ Linux distro, Linux 发行版
Linux kernel + GNU + 其它自由或开源软件 = GNU/Linux 操作系统
- ▶ DistroWatch³, Linux 发行版数据库
目前仍然处于活跃开发状态的总计 251 个
- ▶ 流行的 Linux 发行版
Debian、Ubuntu、Fedora、Arch Linux、deepin、Manjaro Linux、openSUSE、Gentoo 等

³<https://distrowatch.com>



图 5: Linux distro

Linux 的优势

- ▶ 开源
获取免费，修改自由
- ▶ 安全可靠
足够多的眼睛，就可让所有问题浮现。— 林纳斯定律
- ▶ 开发者友好
编程语言（C/C++、Python、Ruby 等）、开发框架、工具.....
- ▶ 无处不在
桌面、服务器、路由器、嵌入设备、Android 等

Linux 的不足

- ▶ 硬件驱动
并非每种硬件都包含 Linux 驱动，使用某些硬件设备存在兼容问题
- ▶ 商业软件
有些大型软件开发商不为 Linux 发布软件，如：Adobe 设计套件
- ▶ 娱乐游戏
运行于 DirectX 技术上的某些流行游戏无法用于 Linux，但过去几年 Steam 已经做出了一些改观

Section 2

编译器、链接器及如何构建软件

编译器

- ▶ **Compiler, 编译器**
将便于人编写、阅读、维护的高级计算机语言所编写的源代码, 翻译为计算机能解读、运行的低级机器语言的程序
 - 高级语言: C、C++、C#、Java 等
 - 机器语言: 汇编语言或目标机器的目标代码
- ▶ **分类**
 - 本地编译器: 编译器输出的代码将运行于与编译器所在相同类型的计算机和操作系统上
 - 交叉编译器: 输出可以运行于不同的平台之上的编译器
- ▶ **著名编译器及工具链**
 - GCC
 - Clang/LLVM

编译器的执行阶段

- ① preprocessing, 预处理
- ② lexical analysis, 词法分析
- ③ parsing, 解析
- ④ semantic analysis, 语义分析
- ⑤ intermediate representation, 将输入程序转换为中间表示
- ⑥ code optimization, 代码优化
- ⑦ code generation, 代码生成

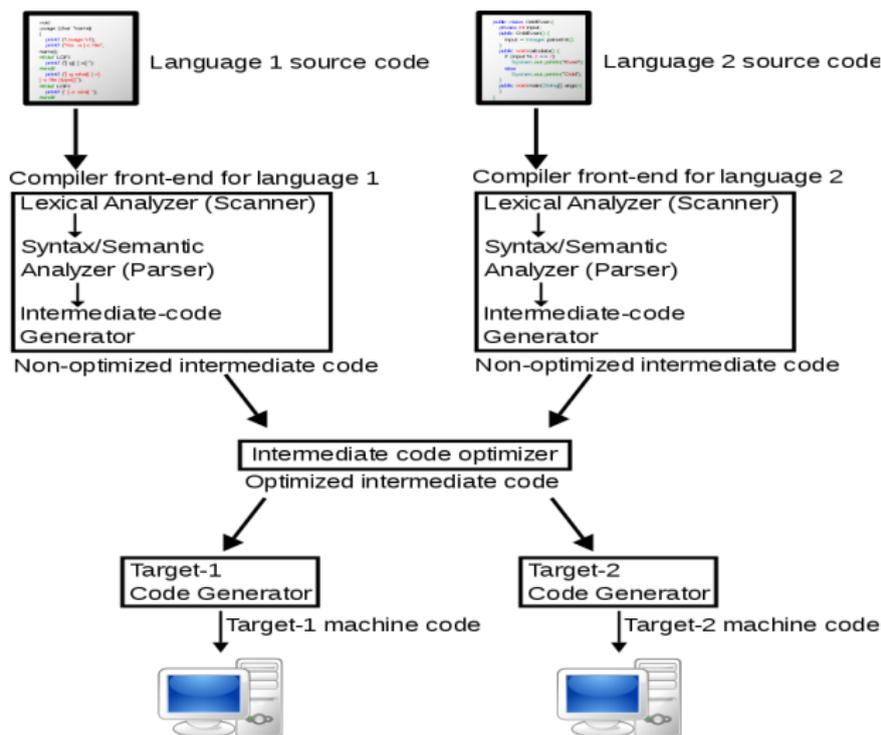


图 6: 编译器的执行阶段

GCC

- ▶ GNU Compiler Collection⁴，GNU 编译器套装
- ▶ 常被认为是跨平台编译器的标准
- ▶ 由理查德·斯托曼开发，1987 年 3 月发布首个版本，现由自由软件基金会负责维护
- ▶ 支持 C、C++、Fortran、Objective-C、Objective-C++、Ada、D、Go 等编程语言
- ▶ 支持非常广泛的处理器家族：x86、ARM、MIPS、Alpha 等
- ▶ 包含 C++ 标准库实现 `libstdc++`
- ▶ 许多类 Unix 操作系统（包括 Linux）的标准编译器

⁴<https://gcc.gnu.org>

Clang/LLVM

- ▶ Clang⁵
适用于 LLVM 的 C 语言家族编译器前端
- ▶ LLVM⁶
模块化及可重用的编译器和工具链技术
- ▶ 模块化架构，能集成、重用或替换 LLVM 的独立部分
- ▶ IR (intermediate representation, 中间表示) 是一大创新
- ▶ 解析器设计对 warning 和 error 消息改进很多
- ▶ 能够 JIT (Just in time, 及时) 编译代码
- ▶ 用户: FreeBSD、macOS、Rust、Zig 等

⁵<https://clang.llvm.org>

⁶<https://llvm.org>

链接器

- ▶ Linker, 链接器
将一个或多个目标文件（由编译器或汇编器生成）组合成单个可执行文件、库文件或另一个“目标”文件的计算机系统程序
- ▶ Static linking, 静态链接
链接器将程序中使用的所有库例程复制到可执行映像中的结果
 - 优点：更便携，不需要库存在；能阻止“DLL 地狱”
 - 缺点：相比动态链接需要更多的磁盘空间和内存

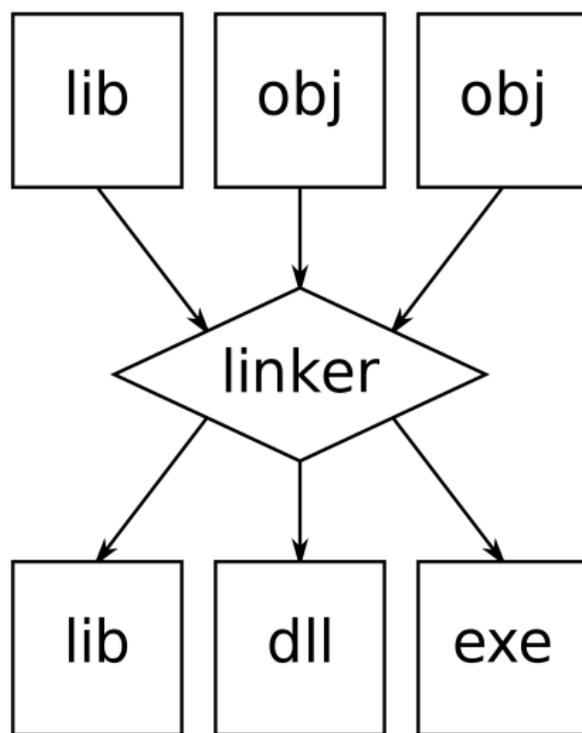


图 7: 链接过程

链接器 - 续

- ▶ Dynamic linking, 动态链接
可执行代码仍包含未定义的符号及为这些符号提供定义的对象或库列表, 加载程序将加载这些对象或库, 并执行最终链接
 - 优点: 常用的库只需存储在一个位置, 从而节省有限的内存和磁盘空间; 通过替换库即可修正库函数的 bug
 - 缺点: 不兼容的更新库将破坏可执行文件
- ▶ 常用链接器
 - GNU ld⁷: GNU linker, 为 GNU Binary Utilities (binutils) 的一部分; 包含两个版本: 基于 bfd 的经典版和仅用于 ELF 的 gold 版
 - LLVM lld⁸: LLVM 项目的 linker

⁷<https://sourceware.org/binutils/>

⁸<https://lld.llvm.org>

如何构建软件

① 下载软件源码包

软件包常以源代码的压缩存档格式发行，如：`.tar.gz`、`.tgz`、`.tar.bz2`、`.tar.xz` 等

② 验证软件源码包

- ▶ 下载数据既能被拦截及修改，也有可能损坏，故验证下载数据很有必要
- ▶ 若软件发行者提供的是 MD5 值，则用 `md5sum <file>` 计算校验和
- ▶ 若软件发行者提供的是 `.sig` 或 `.asc` 签名文件，则用 `gpg --import <public-key>` 和 `gpg --verify <file>` 验证签名

③ 提取软件源码包

一般用 `tar xvf <file>` 命令

如何构建软件 - 续

- ④ 阅读常见文件 README 或 INSTALL
这些文档包含如何编译、安装及配置源代码的重要信息
- ⑤ 给源代码打补丁（可选）
软件在你的环境不工作，且有人已经解决了问题，发布的工作称为 patch（补丁）文件，patch 工具能应用补丁
- ⑥ 配置、编译及安装
 - ▶ 管理源代码编译及安装使用最广泛地工具是 make，configure 脚本将 makefile 模板转换为真正的 Makefile，make 执行编译过程，make install 执行安装
 - ▶ 除了 make 外，还包括其它构建工具，如：cmake，其工作方式与 configure 类似

如何构建软件 - 续

7 构建及安装文档（可选）

- ▶ 有些项目提供文档，如：man 或 info
- ▶ 有些是 HTML 形式，一般安装到 /usr/share/doc
- ▶ 有些是 raw（原始）形式，需要处理后才能安装

8 安装后配置（可选）

有些程序在启动时可通过 configuration（配置）文件定制行为

如何构建软件 - 示例：构建 sed

- 1 下载源码包 `sed-4.8.tar.gz`⁹、签名文件 `sed-4.8.tar.gz.sig`¹⁰ 及 GNU 公钥 `gnu-keyring.gpg`¹¹
- 2 验证源码包 `sed-4.8.tar.gz`
 - ▶ 导入公钥: `gpg --import gnu-keyring.gpg`
 - ▶ 验证签名: `gpg --verify sed-4.8.tar.gz.sig`
- 3 提取源码包: `tar xvf sed-4.8.tar.gz`
- 4 阅读 `INSTALL` 了解安装介绍: `less INSTALL`
- 5 配置、编译及安装
 - ▶ 配置: `./configure --prefix=/tmp`
 - ▶ 编译: `make`
 - ▶ 安装: `make install`

⁹<https://ftp.gnu.org/gnu/sed/sed-4.8.tar.gz>

¹⁰<https://ftp.gnu.org/gnu/sed/sed-4.8.tar.gz.sig>

¹¹<https://ftp.gnu.org/gnu/gnu-keyring.gpg>

Section 3

Linux 操作系统的构建方法

Linux 操作系统的构建方法

- ① 准备工作
满足构建要求的宿主系统、所用的软件包和补丁、以及基本的环境配置
- ② 构建交叉工具链和临时工具
先构建 GCC 交叉编译工具链，再利用工具链来交叉编译临时工具
- ③ 构建 Linux 系统
安装基本的系统软件，对系统进行配置，使 Linux 系统可引导

Section 4

补充阅读材料

补充阅读材料

- ① 《程序员的自我修养——链接、装载与库》
 - ▶ 第 2 章 编译和链接
 - ▶ 第 4 章 静态链接
 - ▶ 第 7 章 动态链接
- ② 《An Introduction to GCC》
 - ▶ 第 2 章 Compiling a C program
- ③ 《GNU Make 项目管理》
 - ▶ 第 1 章 如何编写一个简单的 Makefile

Section 5

参考资料

参考资料

- ▶ Wikipedia: Linux kernel
- ▶ Wikipedia: Open source
- ▶ Wikipedia: Compiler
- ▶ Wikipedia: Linker
- ▶ About the GNU Operating System
- ▶ 《只是为了好玩：Linux 之父林纳斯自传》
- ▶ 《若为自由故：自由软件之父理查德·斯托曼传》