小熊猫包管理维护心得

宇宙眼镜人

我是谁

- 小熊猫包管理、安同 OS 安装器作者
- Bug 缠身的应用开发者
- QQ: 宇宙眼镜人
- Telegram: 眼镜在哪
- GitHub: @eatradish

问题意识：为什么安同 OS 需要 oma

- 安同 OS 是个相当“怪异”的发行版
  - 不分包且依赖设定比较固化
  - 用户水平和目的多样化
    - 猎奇心理在安同 OS 新用户中较为普遍
    - 安同 OS 作为 .deb 发行版，使用习惯存在显著差异
      （那么用户的条件反射会是什么呢？）
  - 与众不同的多测试源设定、镜像源管理需求

我们的答案：重新设计包管理前端

- 核心思想：防呆和高度集成化
  - 清晰的审阅界面
  - 提供系统更改的撤销功能
  - 简化或隐藏晦涩及模棱两可的功能
    - apt upgrade/full-upgrade
    - apt remove/apt purge
      - 何不变成 oma remove/oma remove --remove-config?
  - 集成测试源、系统电源状态等的探测功能

待操作清单

oma 将执行如下操作，请仔细验证。

为应用您指定的更改，oma 可能 安装、卸载、更新、降级 或 重装 软件包。

按 [q] 结束审阅并应用更改
按 [Ctrl-c] 中止操作
按 [PgUp/Dn]、方向键或使用鼠标滚轮翻页

313 个软件包将被 卸载：

软件名	版本	备注
accountsservice	-2.05 MiB	清理未使用的依赖
adwaita-qt	-988.00 KiB	清理未使用的依赖
aha	-92.00 KiB	清理未使用的依赖
akonadi	-15.42 MiB	清理未使用的依赖
akonadi-contacts	-4.04 MiB	清理未使用的依赖
aosc-media-writer	-2.45 MiB	清理未使用的依赖
apparmor	-5.80 MiB	清理未使用的依赖
appstream-glib	-4.36 MiB	清理未使用的依赖
appstream-qt	-500.00 KiB	清理未使用的依赖
ark	-6.34 MiB	清理未使用的依赖
audiocd-kio	-4.25 MiB	清理未使用的依赖
baloo	-4.44 MiB	清理未使用的依赖
baloo-widgets	-1.26 MiB	清理未使用的依赖
bamf	-1.14 MiB	清理未使用的依赖
black-hole-solver	-136.00 KiB	清理未使用的依赖
bluedevil	-3.14 MiB	清理未使用的依赖
bluez-qt	-2.57 MiB	清理未使用的依赖
bolt	-624.00 KiB	清理未使用的依赖

按 [q] 结束审阅并应用更改，按 [Ctrl-c] 中止操作，按 [PgUp/Dn]、方向键或使用鼠标滚轮翻页。

```
saki@Daylily [ ~ ] $ sudo oma topics
  WARNING 您的电脑目前似乎正在使用电池供电。oma 在执行任务时可能会消耗大量电量
          ，推荐您接入交流电源以防断电导致数据损坏。
✔ 您确定要继续吗？ · yes
? 打开测试源以获取实验性更新，关闭测试源以回滚到稳定版本：
 ✔ crow-translate-2.11.1 (crow-translate: update to 2.11.1)
  ✔ cargo-audit-0.20.0
  ✔ spark-store-survey-20240403 (Spark Store Compatibility Survey (April, 2024))
  ✔ google-chrome-126.0.6478.126
    aarty-0.6.1
    abbs-update-checksum-0.2.0 (abbs-update-checksum: update to 0.2.0)
    abseil-cpp-20240116.2-rework3 (abseil-cpp: update to 20240116.2 (rework again and again))
    ack-3.7.0 (ack: update to 3.7.0)
    astyle-3.5.1 (astyle: update to 3.5.1)
    borgbackup-1.4.0 (borgbackup: update to 1.4.0)
    cabextract-1.11 (cabextract: update to 1.11)
    castxml-0.6.7
    checksec-2.7.1 (checksec: update to 2.7.1)
    chromaprint-1.5.1 (chromaprint: update to 1.5.1)
    cifs-utils-7.0 (cifs-utils: update to 7.0)
    clamav-1.3.1 (clamav: update to 1.3.1)
    clash-verge-rev-1.7.3 (clash-verge-rev (Clash Verge Rev): update to 1.7.3)
    cloud-init-24.2 (cloud-init: 24.2+git20240704)
    clzip-1.14 (clzip: update to 1.14)
▼   cmark-0.31.0 (cmark: update to 0.31.0)

按 [Space]/[Enter] 开关测试源，按 [Esc] 应用更改，按 [Ctrl-c] 退出。
```

改良和优化——解决 APT 的普遍痛点

- 多线程下载和高性能 HTTP 库缩短下载耗时
- command-not-found 的高性能实现
  - 与 Ubuntu 的 cnf 实现相比体验入侵性更低
  - oma 的 cnf 基本没有延迟！

```
saki@Daylily [ ~ ] $ oma download vscodium go qt-5 texlive
 Progress    77.34 MiB / 3.79 GiB @ 44.24 MiB/s            00:01:25   [>--------------------------]  2%
 (1/4) vscodium 1.90.2.24171 (amd64)                        87.50 MiB   [=======>                  ] 30%
 (2/4) go 1.22.3+tools0.21.0+net... (amd64)                 107.34 MiB  [=====>                    ] 23%
 (3/4) qt-5 1:5.15.13+webengine5.1... (amd64)               161.63 MiB  [⇒                         ]  8%
 (4/4) texlive 20220321-6 (amd64)                           3.44 GiB    [>                        ]  0%
```

开发历史

- 早期 oma 的设计和维护上有显著问题
  - 代码结构混乱且效率较低
  - 没有系统性的测试流程
  - 设计上存在较为明显的个人偏好
- 在过去一年，oma 通过引入特性和设计修改尽可能地迎合用户的反馈

版本历史：1.1

- 模块化
  - 整理代码，方便其他开发者利用 oma 特性
  - 支持功能扩展及第三方特性集成
- 系统集成
  - 自动检查电源状态，控制电源及会话管理，有效避免意外故障
- 界面进化
  - 重新设计历史及撤销功能，操作界面更简明易懂
  - 各项操作确认后，向终端回显操作内容，以便查阅
- 性能优化
  - 优化下载及解压逻辑，源数据刷新大幅度增速

版本历史：1.2

- 大幅度改善报错和调试信息
  - 方便用户和开发者解决使用过程中遇到的问题
- 新增 --sysroot= 参数
  - 允许管理设备中安装的其他 AOSC OS 系统或容器

版本历史：1.3

- oma 历史上维护时间最长的分支
  - 说明 oma 已经基本进入稳定使用的阶段
  - 可在大多数情况下替代 apt
- 小熊猫 TUI：提供直观的搜索、安装和管理功能
- 改善进程管理，更加直观地向用户展示进程锁
- 改善依赖报错的准确性
- 初步实现了“泛 Debian 兼容性”
  - 支持 Debian 及 Ubuntu 使用的大多数软件源
  - 配合 Spiral 兼容性标记的需求，支持更多不同配置的软件源

小熊猫包管理 v1.4.0-alpha.0

搜索软件包

game

软件包列表（5 可更新，0 可删除，1983 可安装）

```
  UPGRADE discord 0.0.58 → 0.0.59
          All-in-one voice and text chat for gamers
    AVAIL 0ad 0.0.26-3 [调试符号可用]
          RTS (Real Time Strategy) Game of Ancient Warfare
    AVAIL aisleriot 3.22.24 [调试符号可用]
          A collection of patience games written in guile scheme
    AVAIL allegro+32 4.4.2-1
          Portable library mainly aimed at video games and multimedia programs
          (optenv32)
    AVAIL atomix 3.34.0-1 [调试符号可用]
          A puzzle game in which full molecules have to be built
    AVAIL blinken 23.08.5 [调试符号可用]
          A game of pattern memory
    AVAIL bomber 23.08.5 [调试符号可用]
          A single player arcade bomber game
    AVAIL bovo 23.08.5 [调试符号可用]
          A Gomoku like game for two players
    AVAIL bullet 2.88-3 [调试符号可用]
          A 3D Collision Detection and Rigid Body Dynamics Library for games
          and animation
    AVAIL cataclysm 0.G [调试符号可用]
          A turn-based survival game set in a post-apocalyptic world
    AVAIL corsixth 1:0.67 [调试符号可用]
          A Theme Hosptial clone which uses the original game's resource files.
    AVAIL darkradiant 3.8.0-2 [调试符号可用]
          Level editor for The Dark Mod and other idTech4/Doom3-based games
```

Quicknav: TAB / F1 / ESC / Space / / / Ctrl+C

做泛 Debian 兼容时的妙妙故事

- 1.3 是 oma 初步涉猎不同系统和生态场景的版本
  - 维护时遇到问题的复杂度不断上升
- 两个案例：oma 如何解决业务需求
  - 如何解析 InRelease 的 Date 和 Valid-Until 栏目
  - 如何正确地处理流式 gzip 数据

日期，RFC 2822 日期，与 APT 软件源

- 日期在 InRelease 等元数据文件中的作用
  - 软件源在何时刷新?
  - 证书的有效期?
- RFC 2822 日期

```
Fri, 12 Jul 2024 15:24:10 +0000
```

- 眼见为虚：APT 使用 RFC 2822 格式的日期吗?
  - 安同 OS 的软件源一直假设了 APT 源使用这一规范，因此 oma 也使用了这一规范的标准实现

APT 软件源并不一定遵从 RFC 2822！

- Debian 的软件源格式规范文档中的范例使用的也是 RFC 2822 日期，但在后续的详细描述中也违反了该 RFC 的规定
  - 因此，Aptly 生成的元数据文件并不完全遵从这一规范！
    - 个位数小时无 0 前缀（RFC 2822 要求 09:00，Aptly 源中可能为 9:00）
    - RFC 2822 对国际协调时的标记为 UT/GMT/+0000，但 Aptly 会生成 UTC（Debian 文档指定支持 UTC/GMT/+0000/Z）
- oma 怎么办?

连锁反应

- 在 Ubuntu 刷新软件源，总是报告错误
  - 由于编码错误，oma 总是在下载未压缩的源文件
    - 而 Ubuntu 不提供未压缩的 Packages 文件
- 解决这一 bug 暴露了更多 bug
  - 下载 Gzip 元数据时总是报错（一边下载一边解压）

```
tabnine: test | explain | document | ask
async fn demo1() -> Result<()> {
    let client: Client = Client::builder().user_agent("oma").build()?;
    let url: &str = "https://mirrors.bfsu.edu.cn/ubuntu/dists/lunar/main/binary-amd64/Packages.gz";
    let mut f: File = fs::File::create(path: "demo1").await?;
    let mut decoder: GzipDecoder<&mut File> = async_compression::tokio::write::GzipDecoder::new(read: &mut f);
    let mut resp: Response = client.get(url).send().await?.error_for_status()?;

    while let Some(v: Bytes) = resp.chunk().await? {
        decoder.write_all(src: &v).await?;
    }

    decoder.shutdown().await?;

    Ok(())
}
```

```rust
tabnine: test | explain | document | ask
async fn demo2() -> Result<()> {
    let client: Client = Client::builder().user_agent("oma").build()?;
    let url: &str = "https://mirrors.bfsu.edu.cn/ubuntu/dists/lunar/main/binary-amd64/Packages.gz";
    let mut f: File = fs::File::create(path: "demo2").await?;
    let stream: IntoAsyncRead<MapErr<impl Stream<Item = …… = client
        .get(url) RequestBuilder
        .send() impl Future<Output = Result<…, …>>
        .await? Response
        .error_for_status()? Response
        .bytes_stream() impl Stream<Item = Result<…, …>>
        .map_err(|e: Error| io::Error::new(kind: ErrorKind::Other, error: e)) MapErr<impl Stream<Item =…
        .into_async_read();

    let decoder: GzipDecoder<IntoAsyncRead<…>> = async_compression::futures::bufread::GzipDecoder::new(read: stream);
    let mut decoder: Compat<GzipDecoder<IntoAsyncRead<…>>> = decoder.compat();

    let mut buf: Vec<u8> = vec![0u8; 8 * 1024];
    loop {
        let n: usize = decoder.read(&mut buf).await?;
        if n == 0 {
            break;
        }

        f.write_all(src: &buf[..n]).await?;
    }

    f.shutdown().await?;

    Ok(())
} fn demo2
```

Struct flate2::write::GzDecoder

source · [−]

```
pub struct GzDecoder<W: Write> { /* private fields */ }
```

[−] A decoder for a single member of a gzip file.

This structure exposes a `Write` interface, receiving compressed data and writing uncompressed data to the underlying writer.

After decoding a single member of the gzip data this writer will return the number of bytes up to to the end of the gzip member and subsequent writes will return Ok(0) allowing the caller to handle any data following the gzip member.

To handle gzip files that may have multiple members, see `MultiGzDecoder` or read more in the introduction.

Struct flate2::bufread::GzDecoder

source · [−]

```
pub struct GzDecoder<R> { /* private fields */ }
```

[−] A decoder for a single member of a gzip file.

This structure implements a `Read` interface. When read from, it reads compressed data from the underlying `BufRead` and provides the uncompressed data.

After reading a single member of the gzip data this reader will return Ok(0) even if there are more bytes available in the underlying reader. If you need the following bytes, call `into_inner()` after Ok(0) to recover the underlying reader.

- write::GzipDecoder 遇到终止符时（终止符不一定在数据的结尾，也可能在一段片段里的头），终止写入流
- chunk 写入的带有终止符的 buffer 有可能不是完整的 buffer
- 后续还有数据，但流已经终止，程序报错退出

- bufread::GzipDecoder 与 write::GzipDecoder 不同，它内部维护了一个 buffer，当输入的片段未组成完整有效的 Gzip 片段时，它会将读取到的数据写入到这个 buffer 中，当数据再次写入而形成有效的 Gzip 片段时则继续传递
- 此方法能够一直写入 Gzip 流直至完成

教训

- API 文档必须仔细查阅
  - 遇到问题时要首先找文档和实现代码进行交叉诊断
- 综合参考其他实现有时也能找到答案
- 测试，测试，测试

oma 1.4 展望

- 进一步完善 TUI 界面
  - 将大多数常用界面从命令行迁移至 TUI
    - 安装、更新、镜像源与测试源
    - 确认、历史与撤销界面
- 重新设计系统更新的展现方式
  - 优先显示「更新集合」，而不是一味列出所有受影响的软件包
  - 以类似 Windows Update 的方式对系列更新进行概括介绍
- 进一步增强单元与使用测试的覆盖率
  - 相对于前三个维护分支延长开发周期，以便进行更充分的用户测试

```
>> 系统更新

您的系统有如下 2 个可用更新：

  更新项目                                                  |  受影响的软件包
------------------------------------------------------------------------------
  KDE 桌面更新（2023 年冬）                                 |  127 个更新、2 个新增
  核心运行时 12.0.2 版                                      |  7 个更新

共计将更新 134 个、安装 2 个软件包。

要继续执行系统更新吗（输入 review 即可查阅更新详情）？(yes/no/review)
```

致谢、问答与后续讨论