

AOSCC 2024 | 欢迎

将塞爆进行到底

利用 OverlayFS 实现多版本 离线安装盘

杨欣辉 (老嘟嘟)

cyan@cyano.uk



《塞爆：2022》

- 2022 年，安同 OS 开始拥有基于 TUI 的安装程序，LiveKit 因此终于进入可用阶段
- 在那期间的 LiveKit 并没有图形环境，但体积不小，约 1.8GB 左右

```
Welcome to AOSC OS LiveKit!

Here you may find basic tools to install AOSC OS, or rescue other operating
systems installed on your computer. Here below is a basic guide to preinstalled
applications (in the form of commands) on LiveKit:

- deploykit: AOSC OS installer.
- cfdisk: Disk partition manager.
- nmtui: Network (Ethernet, Wi-Fi, etc.) connection manager.
- w3m: Web browser.

If you have encountered any issue, please get in touch with us:

- IRC: #aosc on Libera.Chat
- GitHub: https://github.com/AOSC-Dev/aosc-os-abbs/issues/new/

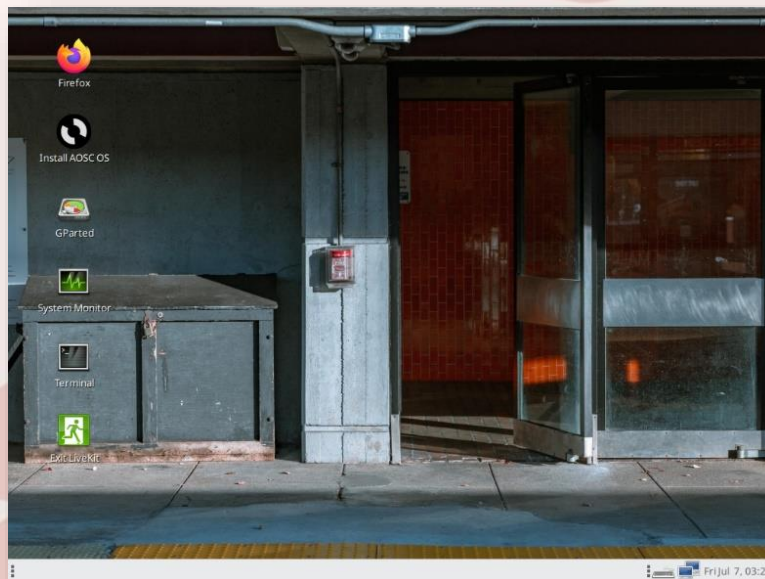
Enjoy your stay!

root@livekit [ ~ ] #
```



《塞爆：2023》

- 2023 年，白铭骢等人为 LiveKit 添加了基于 MATE 桌面的简易维护环境，LiveKit 从此“进入 GUI 时代”
- 由于引入了桌面环境，LiveKit 体积翻倍，达到了 3GB



《塞爆：2024》之前

- LiveKit 的文件系统是一个 ext4 镜像
- 存储在 squashfs 中的 /LiveOS/rootfs.img
- 由 dracut(8) 负责加载
 - dmsquashfs-live 模块挂载 squashfs
 - 之后挂载 /LiveOS/rootfs.img
 - 使用 Device Mapper 扩展成一个可读写层
 - 最终的可读写层是经过扩展的 ext4 文件系统



将塞爆进行到底！

- 自推出龙架构移植以来，安同 OS 的用户数量直线上升
- LiveKit 只是一个维护环境，并不自带系统发行
- 用户开始就 LiveKit 的体积问题灵魂拷问



“

为撒 3.5GB 的安装
器还得下载 8GB 的
系统？

”



塞爆，也要合理

因此需要一个终极解决方案，同时提供系统和安装环境，但也要讲究一定空间利用率

但基础版、桌面版、服务器版和 LiveKit 环境总共占用约 15GB，显然不合理

- 其中有大量重复组件，可以抽出来复用



塞爆进行时：系统设计

白铭骢提出了分层镜像的想法，尽可能复用系统组件，提升空间利用率

- 安同 OS 不同版本均为对基础版的增量
- 分离基础版，然后使用 OverlayFS 做不同版本的叠加层
- 白实现了最初的镜像生成步骤

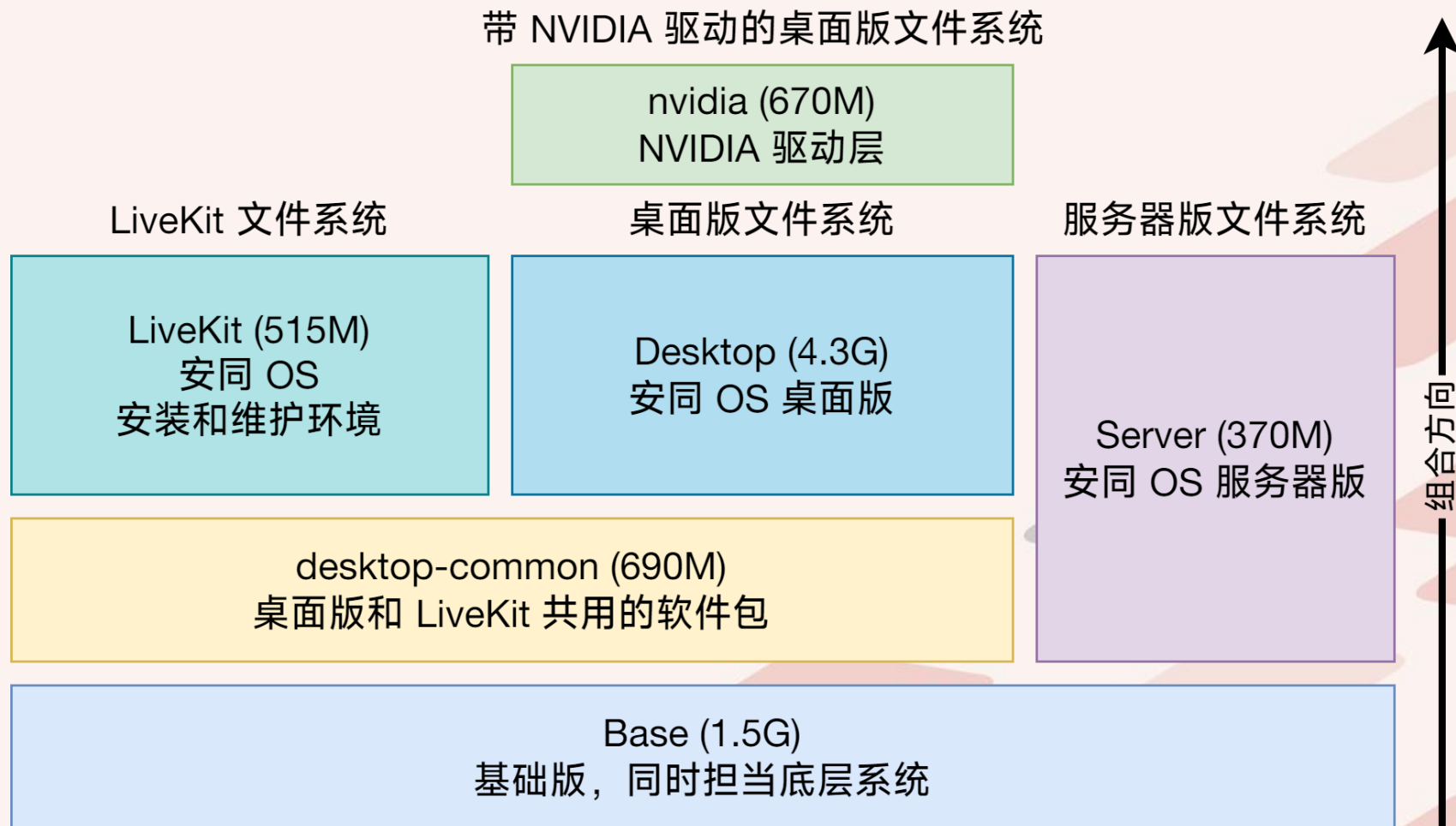


塞爆进行时：系统设计

- LiveKit 本质是 MATE 桌面
- 安装盘肯定包含桌面版
- 可以挑出和桌面版共同使用的软件包
- 这些软件包单独抽一层，进一步节省空间



塞爆进行时：系统设计



塞爆进行时：系统设计

- 生成基础发行，打包
- 叠加一层 desktop-common，装包
- 清理，打包 desktop-common 层的 upper 目录
- 再叠加一层 desktop，装包
- 清理，打包 desktop 层的 upper 目录
- 完成！



塞爆进行时：模块设计

dracut 没有模块能挂载 LiveKit 的分层镜像，因此还需解决谁来加载的问题

- 需要编写一个 dracut 模块来挂载这些镜像
- 同时提供一个可读写的临时 Live 环境
- 因此用户得以体验安同 OS 的桌面版
- 当然，用 Bash 编写
 - initrd 而已，不至于上其他编程语言



塞爆进行时：实现

老嘟嘟随后编写了 dracut 模块，并成功根据白铭骢提供的初版镜像启动 Live 环境

- 模块挂载所有的分层
- 根据配置将分层组合成完整的系统环境 (sysroot)
- 最后根据启动参数挂载一个可读写的 Live 环境



塞爆进行时：结果

```
[INFO]: Welcome to AOSC OS LiveKit!  
[INFO]: Command arguments are:  
[INFO]: /dev/disk/by-label/LiveKit  
[INFO]: Creating temporary filesystem ...  
MOUNT -t tmpfs -o rw,size=7893M,relatime livekit /run/livekit  
[INFO]: Creating directory structure ...  
[INFO]: Mounting LiveKit ...  
MOUNT -o ro /dev/disk/by-label/LiveKit /run/livekit/livemnt  
[INFO]: Reading config files (if any) ...  
[INFO]: Mounting base sysroot ...  
MOUNT -t squashfs -o ro,threads=4 /run/livekit/livemnt/squashfs/base.squashfs /run/livekit/layers/base  
MOUNT --bind /run/livekit/layers/base /run/livekit/sysroots/base  
[INFO]: Mounting layers ...  
MOUNT -t squashfs -o ro,threads=4 /run/livekit/livemnt/squashfs/layers/desktop-common.squashfs /run/livekit/layers/desktop-com  
MOUNT -t squashfs -o ro,threads=4 /run/livekit/livemnt/squashfs/layers/desktop.squashfs /run/livekit/layers/desktop  
MOUNT -t squashfs -o ro,threads=4 /run/livekit/livemnt/squashfs/layers/desktop-nvidia.squashfs /run/livekit/layers/desktop-nvid  
MOUNT -t squashfs -o ro,threads=4 /run/livekit/livemnt/squashfs/layers/livekit.squashfs /run/livekit/layers/livekit  
MOUNT -t squashfs -o ro,threads=4 /run/livekit/livemnt/squashfs/layers/server.squashfs /run/livekit/layers/server  
[INFO]: Mounting sysroots ...  
MOUNT -t overlay sysroot:desktop -o lowerdir=/run/livekit/layers/desktop:/run/livekit/layers/desktop-common:/run/livekit/layers  
MOUNT -t overlay sysroot:desktop-nvidia -o lowerdir=/run/livekit/layers/desktop-nvidia:/run/livekit/layers/desktop:/run/liveki  
MOUNT -t overlay sysroot:server -o lowerdir=/run/livekit/layers/server:/run/livekit/layers/base,redirect_dir=on /run/livekit/s  
MOUNT -t overlay sysroot:livekit -o lowerdir=/run/livekit/layers/livekit:/run/livekit/layers/desktop-common:/run/livekit/layers  
MOUNT -t overlay sysroot:livekit-nvidia -o lowerdir=/run/livekit/layers/livekit:/run/livekit/layers/desktop-nvidia:/run/liveki  
[INFO]: Sysroots are set up successfully.  
[INFO]: Using default template file for boot target .  
[INFO]: Booting into livekit ...  
[INFO]: Setting up live environment template ...  
MOUNT -t squashfs -o ro,threads=4 /run/livekit/livemnt/squashfs/templates/livekit.squashfs /run/livekit/template  
MOUNT -t overlay live-sysroot:livekit -o lowerdir=/run/livekit/template:/run/livekit/sysroots/livekit,upperdir=/run/livekit/up  
[INFO]: Finishing up ...
```



模块功能

- 读取介质内存放的配置文件
- 根据配置文件挂载基础层和所有分层
- 根据配置文件描述的依赖信息组合 sysroot
- 决定需要启动哪一个 sysroot，然后
- 挂载对应系统的 Live 模板和临时空间
- 组合成 Live 系统环境，告诉 dracut 启动系统



模块配置

```
#!/bin/bash
# All available layers.
LAYERS=("desktop-common" "desktop" "desktop-nvidia" "livekit" "server")
# All possible sysroots these layers combine to.
SYSROOTS=("desktop" "desktop-nvidia" "server" "livekit" "livekit-nvidia")
# What layers combine into a sysroot.
# Only dashes are allowed in the layer names - they are converted into
# underscores.
SYSROOT_DEP_desktop=("base" "desktop-common" "desktop")
SYSROOT_DEP_livekit=("base" "desktop-common" "livekit")
SYSROOT_DEP_server=("base" "server")
# it does nothing to the loader's behaviour, even if nvidia is not supported.
SYSROOT_DEP_desktop_nvidia=("base" "desktop-common" "desktop" "desktop-nvidia")
SYSROOT_DEP_desktop_latx=("base" "desktop-common" "desktop" "desktop-latx")
SYSROOT_DEP_livekit_nvidia=("base" "desktop-common" "desktop-nvidia" "livekit")

TEMPLATE_desktop_nvidia="desktop.squashfs"
TEMPLATE_livekit_nvidia="livekit.squashfs"
```



细节：如何利用 OverlayFS

我们用到了 OverlayFS 的两种模式：

- 只读 overlay：只指定底层文件系统
 - 用于组合只读的 sysroots，避免意外读写污染系统
- 读写 overlay：需要指定底层和上层文件系统，以及临时工作目录
 - 用于组合成 Live 可读写环境
 - 底层文件系统是组合出的 sysroots（只读）
 - 上层文件系统存放在 tmpfs 中，可读写



细节：如何利用 OverlayFS

- 先以只读方式组合挂载所有的 sysroots
 - 这些只读 sysroots 会由安装器直接使用（LiveKit 除外）
- 寻找需要引导的系统所对应的 Live 环境模板
- 挂载 Live 模板，组合成 Live sysroot
- 挂载一个 tmpfs，存放可读写的 Live 环境
- 组合 Live sysroot 和 tmpfs，挂载到 /sysroot
- 启动！



细节：如何利用 OverlayFS

- 只读 OverlayFS

```
mount -o ro squashfs/base.squashfs /layer/base
mount -o ro squashfs/desktop-common.squashfs /layer/desktop-common
mount -o ro squashfs/livekit.squashfs /layer/livekit
mount -t overlay \
  -o lowerdir=/layer/livekit:/layer/desktop-common:/layer/base \
  sysroot:livekit /sysroots/livekit
```



细节：如何利用 OverlayFS

- 可读写 OverlayFS

```
mount -t overlay \  
-o lowerdir=/layer/livekit:/layer/desktop-common:/layer/base,\  
upperdir=/live-rw,workdir=/work,redirect_dir=on \  
sysroot-rw:livekit /sysroot # for dracut
```

或者嵌套挂载

```
mount -t overlay \  
-o lowerdir=/sysroots/livekit,\  
upperdir=/live-rw,workdir=/work,redirect_dir=on \  
sysroot-rw:livekit /sysroot # for dracut
```



《塞爆：2024》

- 是一堆打包成 squashfs 的 OverlayFS 上层
- 启动时现场挂载，按顺序组合出完整系统
- 同时使用 tmpfs 创建一个读写 overlay
- 所有 overlay 和 sysroot 都在启动后可见
- 安装程序可以更简便地安装系统
- 带全了系统版本，同时尽可能复用组件



《塞爆：2024》

最终，我们做到了：

- 8.7GB 大小 (amd64)
- 带有 LiveKit 安装和维护环境
- 带有基础版、桌面版和服务器的离线安装包
- amd64 和 arm64 带有预置 NVIDIA 驱动的桌面版
- 龙架构带有预置 LATX+Wine 的桌面版



《塞爆：2024》

- 显然一张 DVD-9 是容不下了
- 蓝光烧进去又有些浪费
- 要同时用到所有内容，无法分盘
- 比较吃随机读写
- 所以还是用 U 盘吧！这东西也不贵了现在



《塞爆：2024》外传

- 语言选择可以从 GRUB 下手
- GRUB 的菜单可以被翻译 (gettext)

```
Select Your Language:
```

```
中文（简体）  
中文（繁體）  
English  
日本語  
한국어
```

```
安装安同 OS  
安装安同 OS（基本图形模式）  
试用安同 OS 桌面版  
试用安同 OS 桌面版（随附 NVIDIA 图形驱动程序）  
安装安同 OS（命令行模式）  
实用工具 >>  
启动默认操作系统或硬盘  
重启  
关机
```

```
Install AOSC OS  
Install AOSC OS (Basic graphics)  
Try AOSC OS Desktop  
Try AOSC OS Desktop (With NVIDIA graphics)  
Install AOSC OS (Command Line Only)  
Utilities >>  
Boot Default OS  
Restart  
Power Off
```



《塞爆：2024》外传

- 编写启动配置文件，使用 Shell 的 Gettext 语法：

```
menuentry "Welcome to LiveKit!" { ... }
```

- 使用 xgettext 导出翻译模板并翻译
- msgfmt 生成 .mo 文件，放置于 GRUB 可读的位置
- 使用 secondary_locale_dir 环境变量
- 编写或生成语言选择菜单
- 大功告成！



《塞爆：2024》外传

grub.cfg:

```
set secondary_locale_dir=/boot/locales_menu

menuentry “中文（简体）” {
    set lang=zh_CN
    configfile /boot/grub/installer.cfg
}

menuentry “English” {
    set lang=en_US
    configfile /boot/grub/installer.cfg
}
```



《塞爆：2024》外传

在这之后，LiveKit 就可以：

- 在每个启动项给内核传递语言设置参数
- 在 initrd 期间设置 Live 环境的语言
- 让安装程序跳过选择语言的步骤
- 让语言设置影响所有方面（包括安装的系统）



总结

- 合理利用 OverlayFS 打造多功能启动盘
- OverlayFS 的组合方式
- dracut 非常灵活，且非常容易扩展
- GRUB 有完整的多语言支持



Q&A



Thanks

