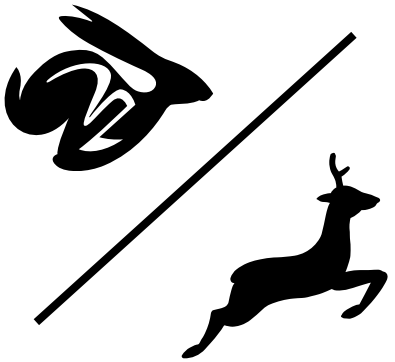
在 ThinkPad X200 上配置多 Payload 的 Libreboot 自由固件

Configuring Multi-Payload Libreboot Free-as-in-Freedom Firmware on ThinkPad X200

Junde Yhi <lmy441900@aosc.xyz>

Libreboot 和 Coreboot 的关系

- Libreboot 是 Coreboot 的一个平行分支
  - Coreboot 滚动发行， Libreboot 稳定发行
  - Libreboot 是去除 Blobs 的 Coreboot
    - Blob 即没有源码的二进制文件
      - 微码、固件、……
  - 但是我实际上操作的是 Coreboot
    - 对于 ThinkPad X200 这种本来就不会有 Blob 加入的平台，直接编译 Coreboot 应也是自由的
- 两者基础概念一致
  - Libreboot 怎么做， Coreboot 就能怎么做
  - Vise versa



Junde Yhi <lmy441900@aosc.xyz>

Liberalized Coreboot

- Coreboot 是一个模块化的固件系统
  - 开机之后， Coreboot 会执行（且仅执行）硬件初始化
  - 之后 Coreboot 会把执行权交给 **Payload**
    - Payload 是个 ELF 可执行文件
    - SeaBIOS (BIOS), TianoCore (UEFI), GRUB2 (?), …
  - Payload 再启动操作系统或引导器
    - 由于 GRUB2 本身就是个引导器，所以可以搞骚操作
      - 免除驱动器上安装的引导器
      - 进而加强安全性，比如真・全盘加密
      - 也是推荐的做法

Junde Yhi <lmy441900@aosc.xyz>

Payload

- （主） Payload 负责引导
  - 一般一次启动只选一个（当然也可以通过别的组合实现多 Payload）
- SeaBIOS 是一个 16 位 x86 BIOS 的开源实现
  - 常见于 QEMU / KVM
- TianoCore 是一个 UEFI 的开源实现
  - 甚至可以在本没有 UEFI 的 ThinkPad X200 上使用 UEFI
- GRUB2 是 GNU 著名的一款操作系统（？）
  - 与安装在硬盘上的 GRUB2 体验完全一致
  - 缺点：
    - 配置文件也得被写在固件里
    - 可能要使用一些高超的配置技术来尽量减少未来的刷写次数

Junde Yhi <lmy441900@aosc.xyz>

Payload

- Coreboot 还集成了些比较有趣的 Payload 的支持
  - Bayou 是个 Payload 选择器，可以在每次启动的时候选择需要的（主） Payload
    - 残念的是这个 Payload 好像已经不维护了，不工作
  - LinuxBoot 直接把 Linux 内核塞到固件里去
    - 用 kexec 引导
  - 或者也可以直接指定一个普通内核（固件空间够大的话）
  - Open Firmware, FILO, iPXE, ...
- 要使用 Windows 的话，选择 SeaBIOS 或 TianoCore
  - 理论上应该是可以的（通过一些微小的修改）
  - 实际上我并没有成功启动过我机器上的 Windows 7

Junde Yhi <lmy441900@aosc.xyz>

Subpayload

- Subpayload 一般执行除了引导以外的任务
  - `nvramcui` 用于修改 CMOS 参数供运行时调整 Coreboot
    - 即 BIOS 选项
  - `coreinfo` 用于查看机器信息
    - CPU 参数、内存参数、固件信息等
  - `memtest86+` 用于测试内存
    - 这个 Subpayload 只能在 VGA Text Mode 下使用（稍后再讲）
  - `tint` 是一个俄罗斯方块游戏（？）
- Subpayload 为 Payload 所链式调用（chainloader）
  - SeaBIOS 启动时按 Esc 可显示 Subpayload
  - GRUB2 通过配置文件或终端执行 Subpayload

Junde Yhi <lmy441900@aosc.xyz>

我做过的实验

- 取 Libreboot 给 Coreboot 、 SeaBIOS 等打的补丁， Rebase 到 master
- 各种尝试配置和 Payload 的组合（50+ 次）
  - 集成 SeaBIOS / 自行编译 SeaBIOS
  - 集成 TianoCore / 自行编译 TianoCore
  - 集成 GRUB2 / 自行编译 GRUB2
- 我可能是比较残念的那个
  - 我的机器在一些配置下会直接不能开机（屏幕不亮）
    - 甚至没办法打开调试
  - 我的机器在任何配置下重新启动都会死机

Junde Yhi <lmy441900@aosc.xyz>

准备（硬件）

石牌村人民醫院

本机会诊号 2019060114

- ThinkPad X200 一台
  - X200s 和 X200t 则不一定（而且拆得比较麻烦）
  - 这是性能最好的一款支持 Libreboot 的机器了（向前还有 X60）
    - Intel Core 2 Duo (Penryn) @ 2.26GHz – 2.66GHz, 3MB L2
- ~~程序员~~编程器 *
  - 可以是开发板的 SPI 接口（linux_spi）
  - 也可以是某宝烂大街土豪金（ch341a_spi）
- 烧录夹（SOIC-8 或 SOIC-16）
  - 推测：
    - 带 AMT （英特尔主动管理技术）的机器使用 SOIC-16 芯片（8MiB / 64Mib）
    - 不带 AMT 的机器使用 SOIC-8 芯片（4MiB / 32Mib）

* Programmer

Junde Yhi <lmy441900@aosc.xyz>

准备（软件）

- Libreboot 源码一份
  - Libreboot 源码只有构建系统和补丁，实际代码都从上游 checkout
  - （小声） Libreboot 声称改进的构建系统有够难用
- 支持 GNAT (GCC Ada) 的编译器
  - 用于编译 Coreboot 的工具链（ GCC Ada 自举）
  - Coreboot 只支持自己的工具链配置
    - 也可以让 Coreboot 在任意工具链上编译，但是不推荐
  - 由于 AOSC OS 的 GCC 不支持 Ada ，所以并不能用 :)
    - 可以 debootstrap 一个 Debian / Ubuntu
    - 完整软件依赖我有时间完善脚本之后会放出来
- `flashrom`

Junde Yhi <lmy441900@aosc.xyz>

更新 Embedded Controller (EC)

- ThinkPad X200 出厂固件的 EC 版本号在 1.06
  - 1.07 EC 固件“修改了电池充电算法以平衡电池的充电和寿命”
  - EC 不开源，但是由于 EC 除了控制没有别的用处，暂且就不管了
- 更新 EC 和官方 BIOS 的安装程序需要使用 Windows 16 位环境
  - 因此 64 位 Windows 不能安装固件更新（不支持 16 位运行时）
    - 就算联想提供了 64 位安装程序也是如此
- 刷写 Coreboot 之后就没有更新 EC 的渠道了
  - 因此在刷 Coreboot 之前记得先更新 EC

Junde Yhi <lmy441900@aosc.xyz>

备份原厂 BIOS

- 备份原厂 BIOS 能让你刷坏之后快速恢复
- `flashrom -p <driver> -c MX25L6405D -r <file>`
  - 以上操作执行三次，以防编程器不稳定导致输出错误
    - 并算好哈希值，确保一致
- 将这份 BIOS 刷回机器就可以恢复出厂固件
  - 其中带的 CMOS 配置和 RTC 会损坏，不过无伤大雅

Junde Yhi <lmy441900@aosc.xyz>

获取 Libreboot

- Libreboot 二进制分发见其官网列出镜像
  - 截至目前最新的版本是 20160907
  - 对 ThinkPad X200 只有 GRUB 可供选择
- 源码
  - `https://notabug.org/libreboot/libreboot.git`
- Libreboot （新）的构建系统真的难用
  - 虽然他们声称自己改进了 Coreboot 的构建方式
  - 这也是为什么 Libreboot 的最新版本也非常老：他们希望在完成构建系统改进之后再发行新版本

Junde Yhi <lmy441900@aosc.xyz>

准备环境

- `debootstrap <codename> <sysroot> <mirror>`
  - 其实只要有带 Ada 支持的 GCC 就可以了
  - 完整的依赖列表还没整理出来
  - 然后 `chroot`
- 进入 Libreboot 目录下载源码
  - `./libreboot download` **`coreboot`**
  - `./libreboot download` **`ich9gen`**（稍后介绍）
  - `./libreboot download` **`vboot`**
  - 以上项目会被自动 checkout 打补丁，放在 sources 目录中

Junde Yhi <lmy441900@aosc.xyz>

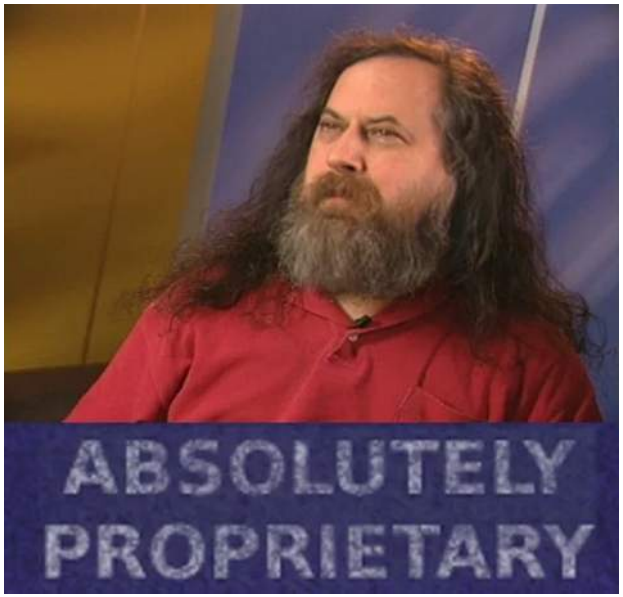
先行编译

- 钦定的工具链
  - `cd sources/coreboot`
  - `make` **`crossgcc-i386`**
    - 如果要编译 TianoCore，那么 amd64 工具链也是需要的：`make` **`crossgcc-x64`**
  - `make` **`iasl`**（ACPI 编译器）
  - 在 make 后使用 CPUS=# 打开并行编译
  - 在 make 后使用 DEST=<path> 安装到指定目录

Junde Yhi <lmy441900@aosc.xyz>

先行编译

- **`ich9gen`**
  - Libreboot 项目，非 Coreboot 项目
  - 生成 ICH9 芯片组初始化需要的前 12 KiB 字段
    - ROM 布局（固件）描述符
    - 板载网卡初始化代码
      - 根据 Libreboot ，这一段基本上是空的
    - 清空的 Intel Management Engine (Intel ME)
      - 通过在固件描述符中去除这一区域做到
  - 注意把板载网卡厂家指派的 MAC 地址写到 `projects/ich9gen/configs/macaddress` 文件中
    - 否则板载网卡将带有 Libreboot 默认的 MAC 地址
      - 一般情况下问题不大，但是一旦同一台交换机下有两台 Libreboot 机器……
    - 一般在机器底部有白色标签写出，也可以使用其它操作系统先看一看



Junde Yhi <lmy441900@aosc.xyz>

ich9gen

- `./libreboot build ich9gen`
  - 这一过程将生成三个 12 KiB 、三个 4 KiB 的二进制文件：
    - `ich9fdgbe_{4,8,16}m.bin` → 12 KiB
    - `ich9fd`**`no`**`gbe_{4,8,16}m.bin` → 4 KiB
  - 根据芯片容量大小，选择合适的固件描述符
    - 在 `build/ich9gen` 里
    - 如我的 X200 使用 SOIC-16 的 8MiB 闪存，则选用 `ich9fdgbe_`**`8m`**`.bin`
  - 带板载网卡的 ThinkPad X200 不能使用 nogbe 描述符
    - 总之就是不能开机
    - nogbe 是给没有板载网卡的 ICH9 机器用的

Junde Yhi <lmy441900@aosc.xyz>

变基 (Rebase)

- 由于 Libreboot checkout 的代码相当老，我们可能希望使用最新的 Coreboot 作为基准
- 变基操作中有不少冲突，需要手动解决
- Libreboot 的补丁主要去除了各平台的微码等没有可审计源代码的部件，并且避免 Coreboot 构建系统自动获取 Libreboot 另行 Deblob 过的项目
  - 然而，由于 ThinkPad X200 平台本身就不需要 Blob 可以构建，这一操作可能是多余的……

Junde Yhi <lmy441900@aosc.xyz>

配置 (Liberalized) Coreboot

- `cd sources/coreboot && make menuconfig`
  - Kconfig （和 Linux 内核配置使用相同的系统）
- General Setup
  - `CBFS prefix to use`
    - 每个 Payload 都存在 CBFS 的 <prefix>/payload 下
    - 默认加载 fallback/payload ，但是这一值需要在 CMOS 运行时配置中给出，所以仅更改这一项会导致 Coreboot 找不到 Payload （我没有试过正确地改动 prefix ）
  - `Add a bootsplash image` 可以为支持启动画面的 Payload 加入启动画面
    - SeaBIOS 、 TianoCore 和 GRUB2 都支持

Junde Yhi <lmy441900@aosc.xyz>

启动画面 (SeaBIOS)

Junde Yhi <lmy441900@aosc.xyz>

配置 (Liberalized) Coreboot

- Mainboard
  - `Mainboard vendor` 选择 `Lenovo`
  - `Mainboard model` 选择 `ThinkPad X200 / X200t`
  - `ROM chip size` 选择合适大小（我的是 `8 MB`）
  - `Size of CBFS filesystem in ROM: 0x7FD000`
    - 即 8 MiB − 12 KiB = 8180 KiB，最大化利用空间
      - 作为参考，TianoCore 约占用 1 - 2 MiB
    - ich9gen 描述的固件布局中 BIOS 区域就是这么大，所以对应地这里也把 BIOS 区域用到这么大
    - 而 Coreboot 默认的 `0x200000` (2 MiB) 是根据联想官方固件布局设定的（带 Intel ME 和 Intel AMT，选择不删除这些部件的情况下这个值没有问题，然而我们要把 ME 去掉）

Junde Yhi <lmy441900@aosc.xyz>

配置 (Liberalized) Coreboot

- Chipset
  - `Include CPU microcode in CBFS` 选择 `Do not`
    - 微码更新有助于处理器的健康运行，但是……
    - 少一个构建依赖（日后也可以往镜像里添加）
  - `Add Intel descriptor.bin file` 打开，然后在下面的选项里填写刚才 `ich9gen` 生成的 `ich9fdgbe_#m.bin` 文件
    - 这个操作实际上就是
      - `dd if=ich9fdgbe_8m.bin of=libreboot.rom bs=12k count=1 conv=notrunc`
      - 完全可以手动操作
  - `Bootblock behaviour` 控制 Coreboot 对 Payload 的选择
    - 假如存在 `normal/payload`，这里选择 `Switch to normal if CMOS says so`，并且 CMOS 里选择了 prefix 是 normal 的话， Coreboot 理论上就会选择它（我没有成功过）

Junde Yhi <lmy441900@aosc.xyz>

配置 (Liberalized) Coreboot

- Devices
  - `Graphics initialization` 选择 `Use libgfxinit`
    - 这是一个使用 Ada SPARK 撰写的显示初始化库
    - 我没有使用别的选项成功启动过（都没有显示）
  - Display 进入，`Framebuffer mode` 选择 `Linear "high-resolution" framebuffer`
    - `Legacy VGA text mode` 就是传统的 80x25 终端，不能显示多媒体内容（图片）
    - 要使用启动画面的话，这一选项必须选中
  - 几个 PCIe 选项开关无妨

Junde Yhi <lmy441900@aosc.xyz>

配置 (Liberalized) Coreboot

- Payload
  - Add a payload 选择一个 Payload
    - 这些 Payload 会在构建时从网络上拉取
  - Secondary Payloads
    - coreinfo 查看机器信息
    - Memtest86+ 测试内存
      - 此 Subpayload 只能在 VGA Text Mode 下正常显示
    - nvramcui 运行时配置
    - tint 俄罗斯方块

Junde Yhi <lmy441900@aosc.xyz>

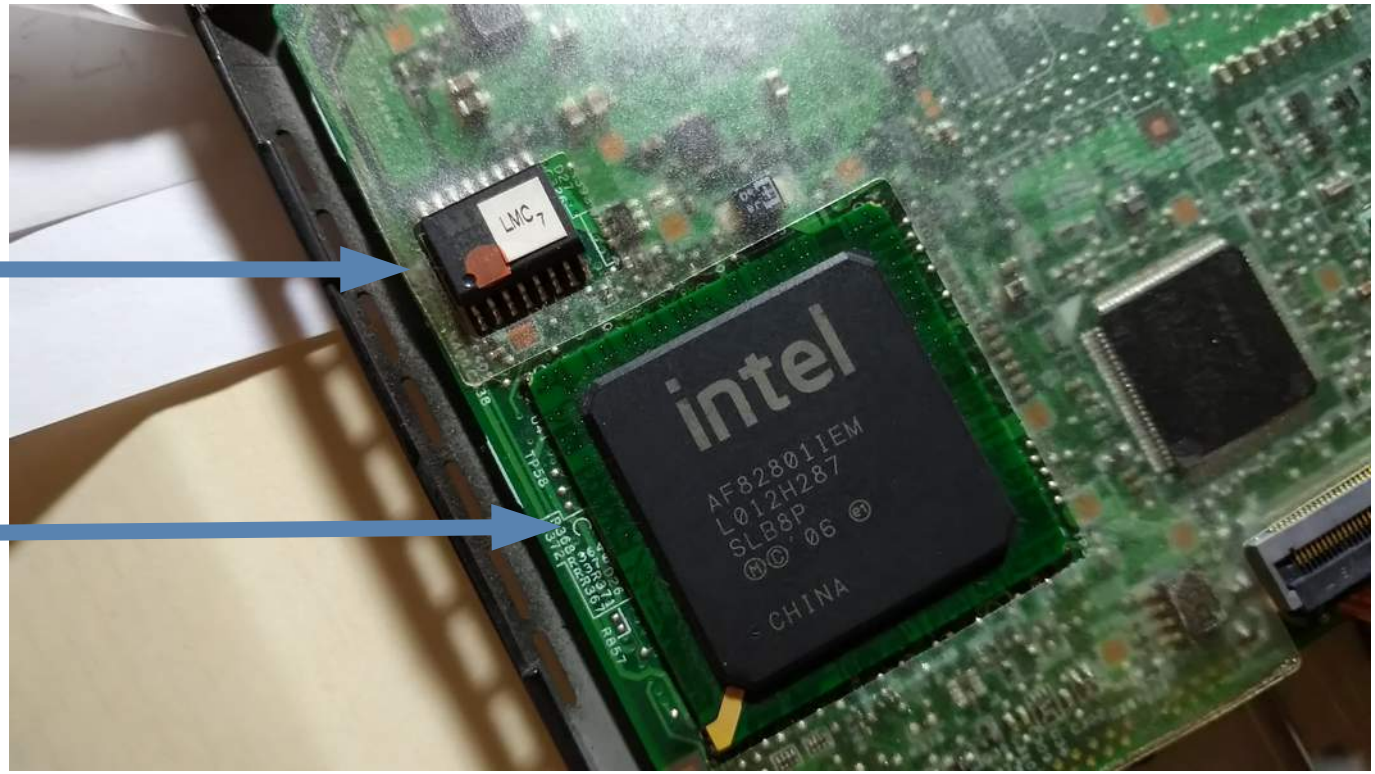
编译 (Liberalized) Coreboot

- 配置保存后，执行 make 即可
  - 不支持多线程（会出问题）
- 成品在 `build/coreboot.rom`
- 一些 Coreboot 工具：
  - `build/cbfstool` 用于修改固件文件内的 Coreboot File System （添加删除 ROM 内文件）
  - `build/ifwitool`
  - `build/rmodtool`

Junde Yhi <lmy441900@aosc.xyz>

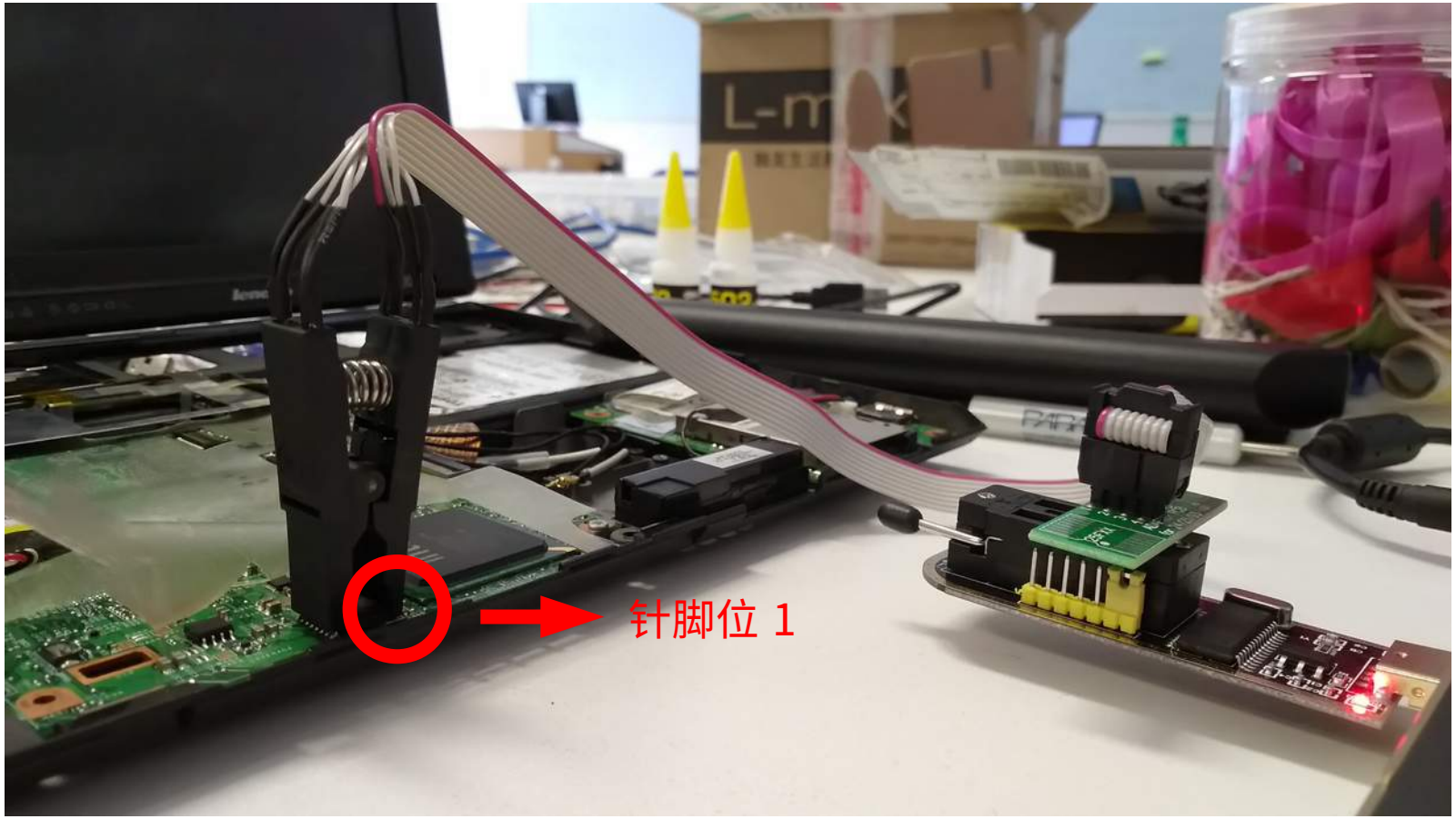
拆开电脑

- ThinkPad X200 的 BIOS 闪存位于左掌托下
  - 卸下 D 面的 9 颗对应的螺丝即可卸下键盘和掌托



Junde Yhi <lmy441900@aosc.xyz>

夹住



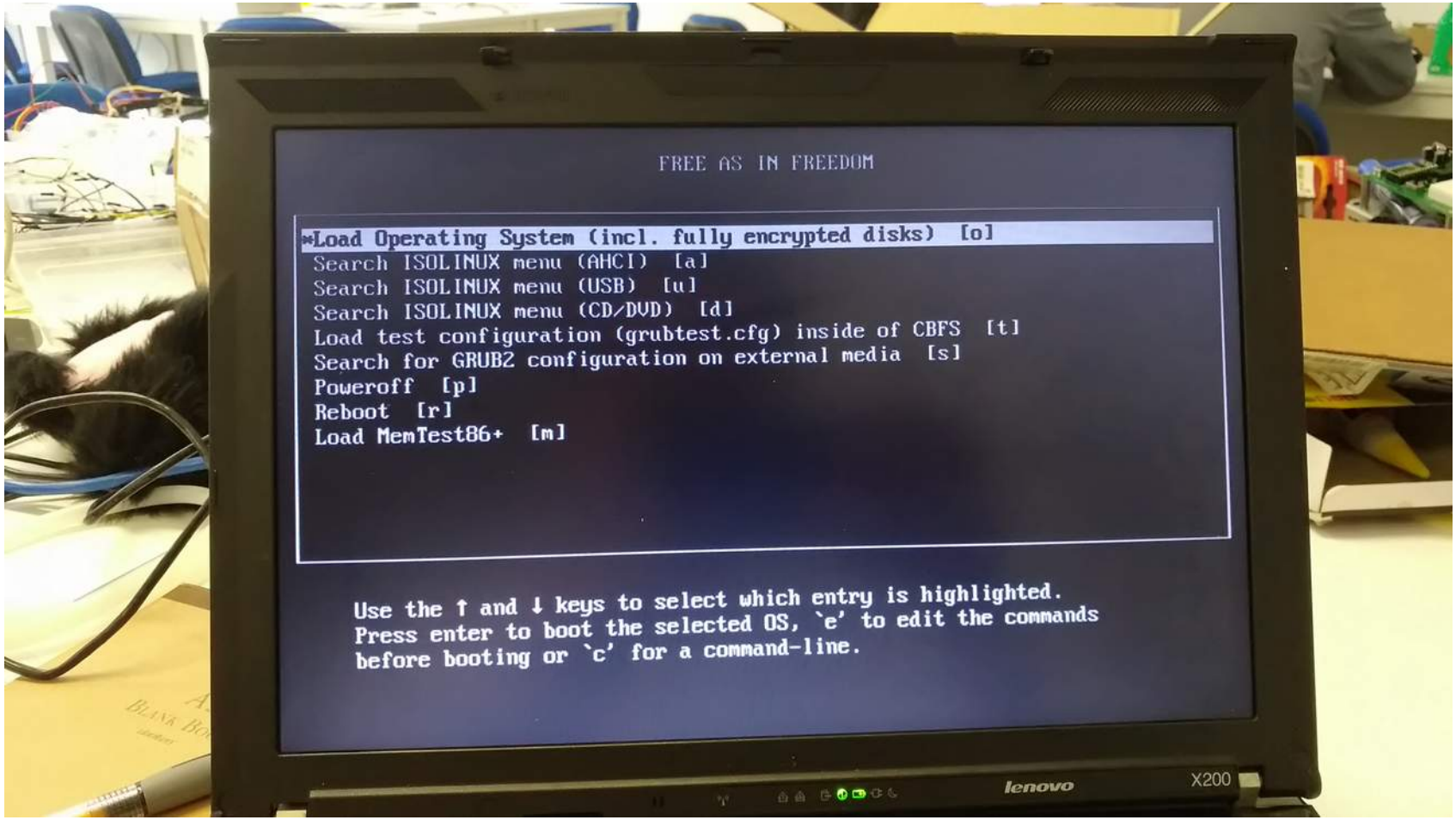
Junde Yhi <lmy441900@aosc.xyz>

烧写

- `flashrom -p ch341a_spi` 读取当前闪存型号
  - 或 `-p linux_spi`，取决于使用的 SPI 烧写器
- 盘它
  - `flashrom -p <driver> -c <chip> -w <img>`
    - `<chip>` 在我的电脑上是 `MX25L6405D`
    - flashrom 会读一次 SPI Flash，然后作差，只将前后不同的内容写入
- 约 5 分钟完成后就可以取掉编程夹开机
  - Flashrom 会显示 VERIFYING... OK.
  - 注意烧写的时候最好不要通电（拔掉电池），更不要开机……
    - 否则机器会砖

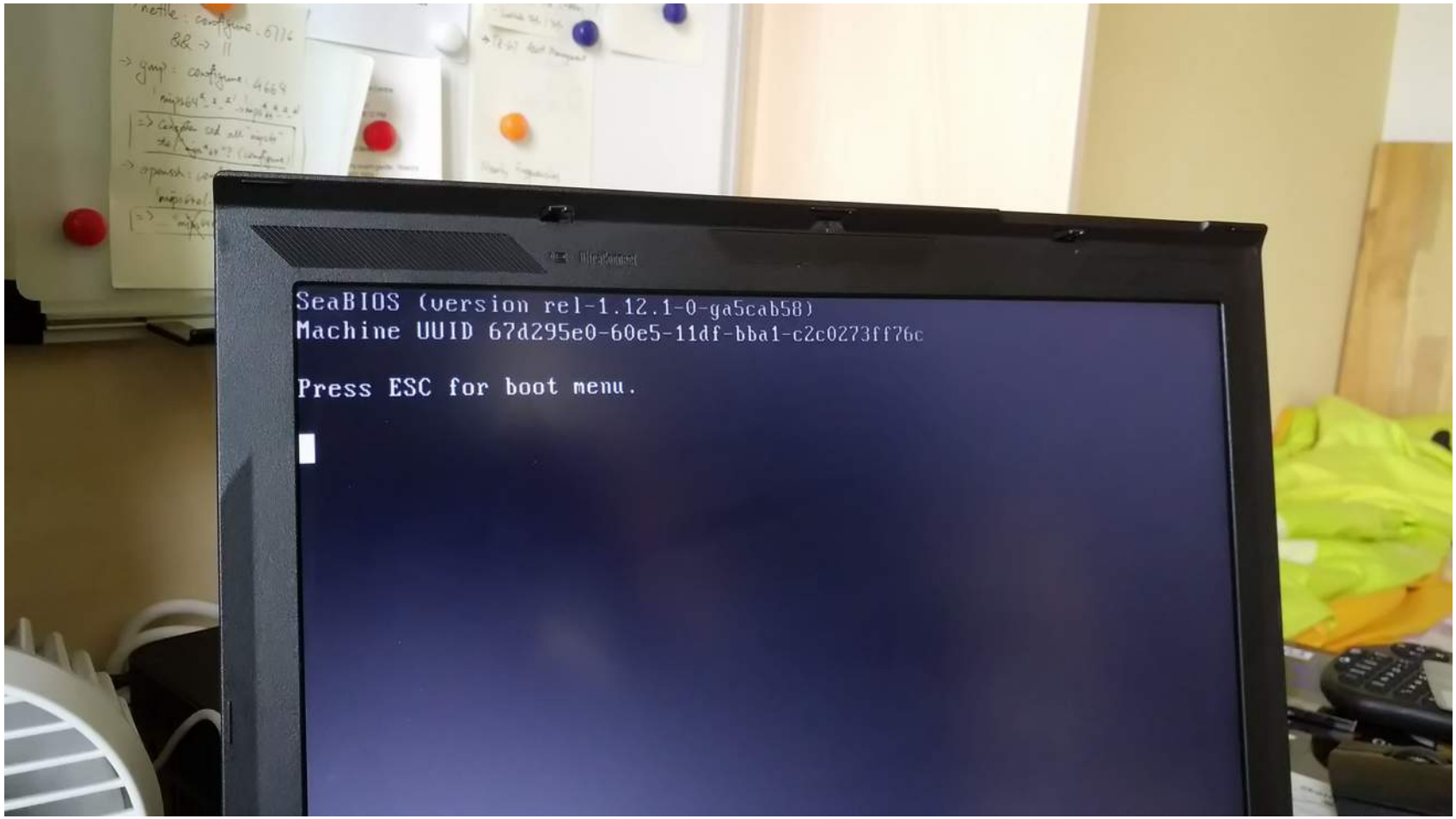
Junde Yhi <lmy441900@aosc.xyz>

效果（ Libreboot 二进制）



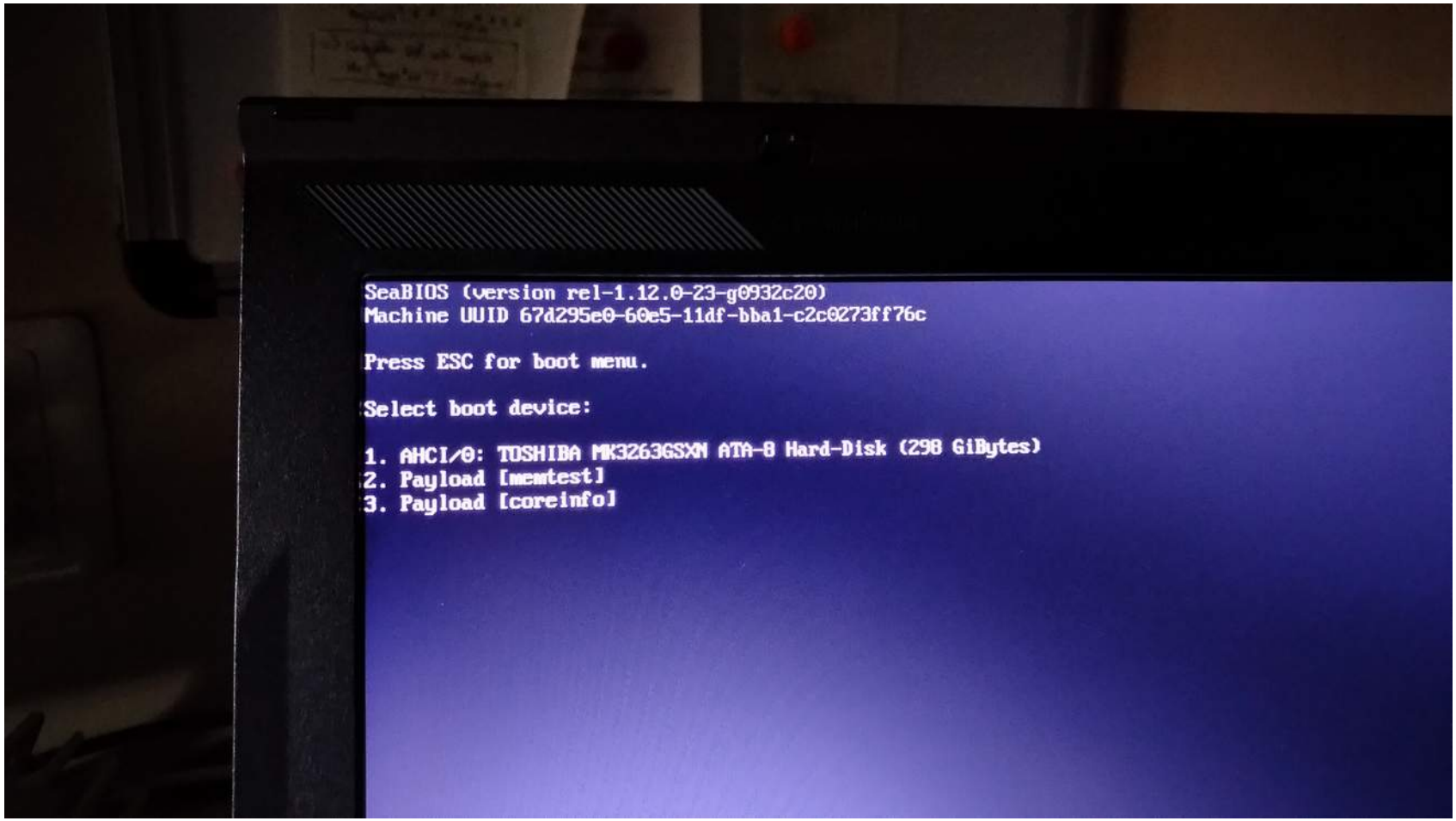
Junde Yhi <lmy441900@aosc.xyz>

效果 (SeaBIOS / VGA Text Mode)



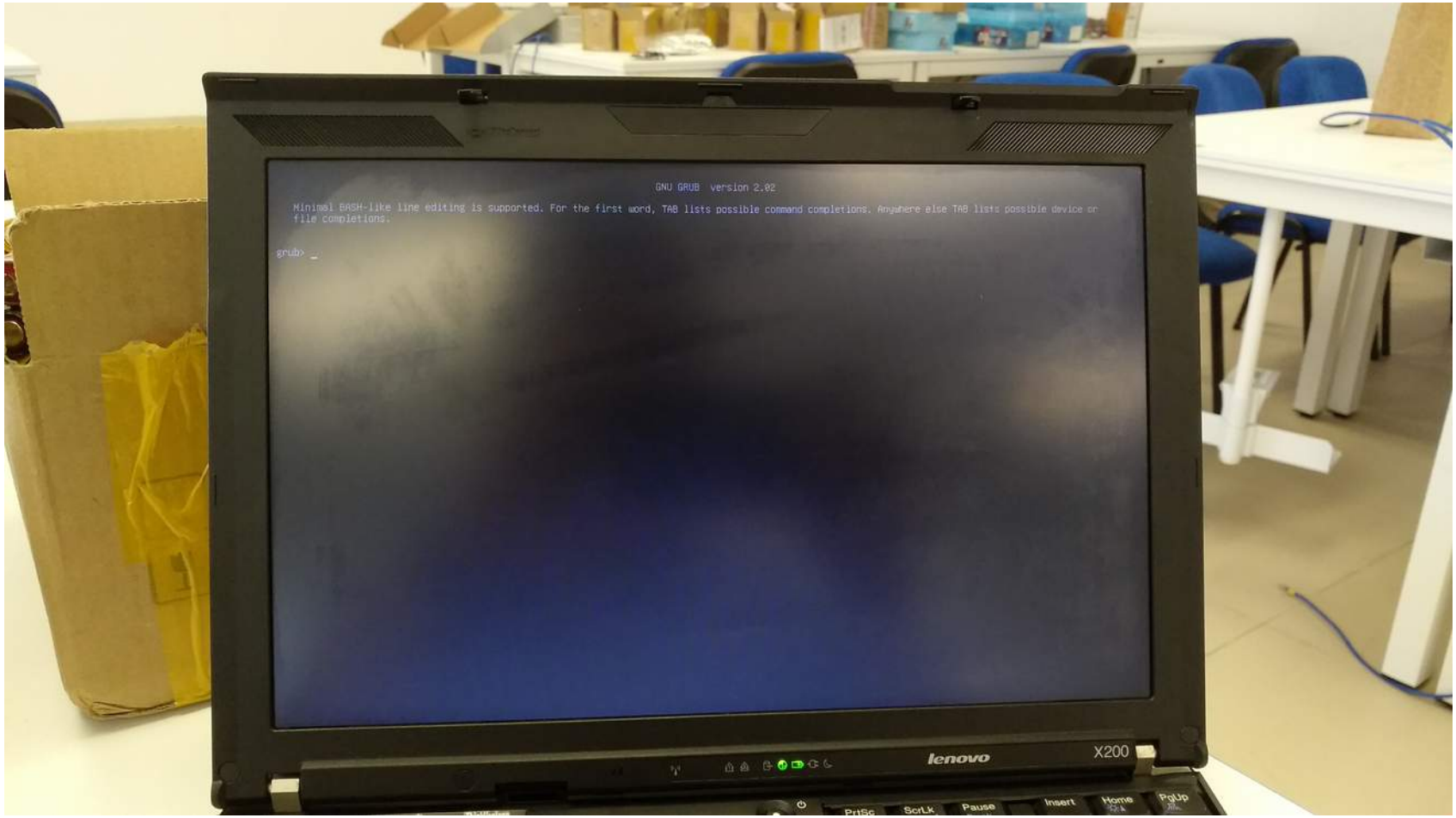
Junde Yhi <lmy441900@aosc.xyz>

效果 (SeaBIOS / VGA Framebuffer)



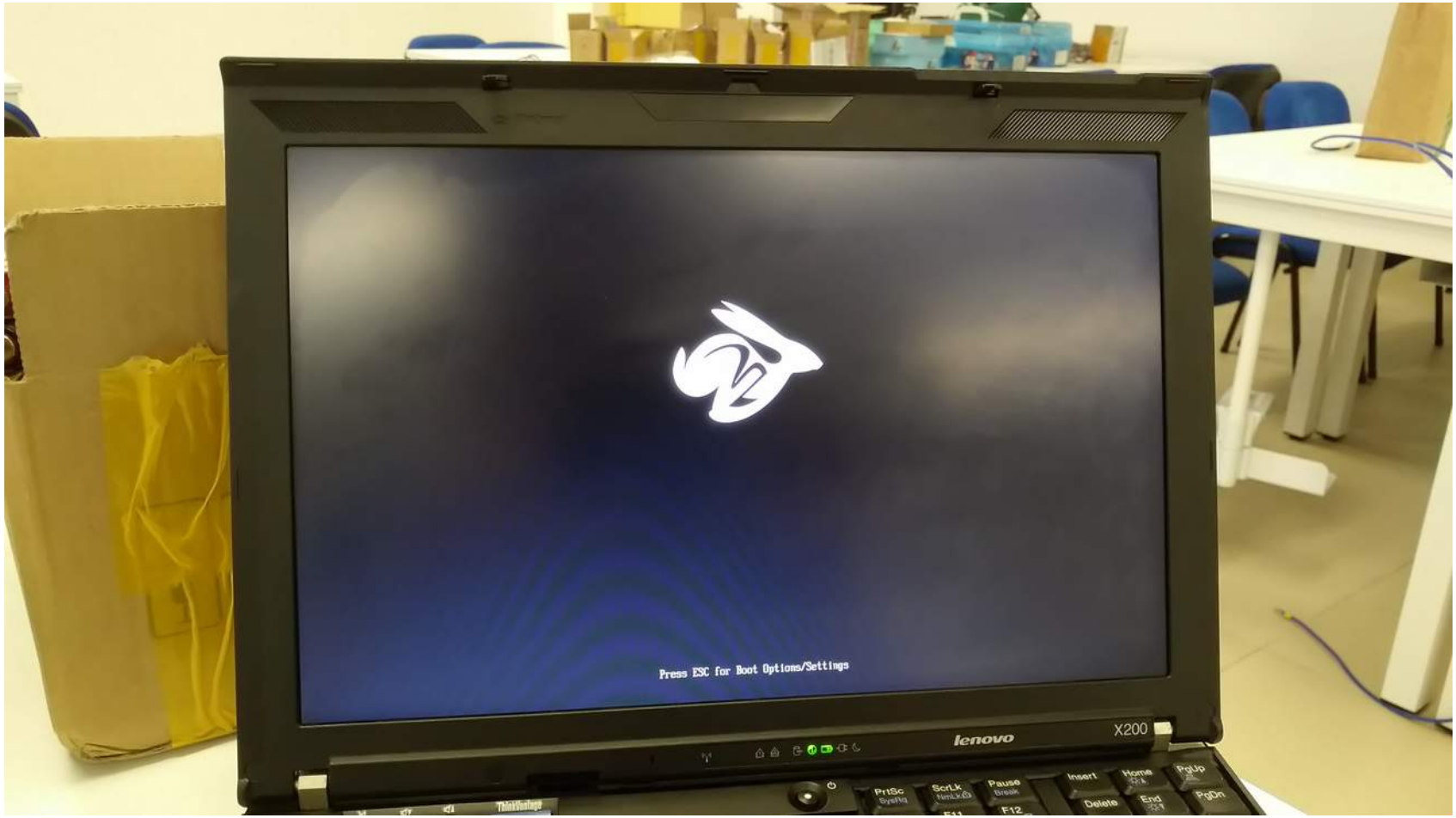
Junde Yhi <lmy441900@aosc.xyz>

效果 (GRUB2)



Junde Yhi <lmy441900@aosc.xyz>

效果 (TianoCore)



Junde Yhi <lmy441900@aosc.xyz>

可能性

- Libreboot + GRUB2 + TianoCore (SeaBIOS as CSM)
  - GRUB2 启动高安全要求的 Linux 操作系统
  - TianoCore 补充 UEFI 平台特性
    - 配合 SeaBIOS 作为兼容支持模块 (CSM) 获得 BIOS 功能
- Libreboot 为用户改进便捷的同时损失了定制度
  - Libreboot 只给 X200 平台提供了某些特性供定制
  - 更复杂的组合还是需要通过手动构建 Coreboot 达到
- 暂时未能实现，可能我比较逊啦

Junde Yhi <lmy441900@aosc.xyz>

Ask Me Anything!

Next (14:15 - 15:00):

Pine64, Its History, and Mainlining Efforts

TL Lim