

H0Tpatch 介绍

一、什么是 H0Tpatch?

引入一段原文：

"static patching". In order to inject patched ACPI files, we extract native ACPI, disassemble them, make changes, then recompile and place the files in ACPI/patched, so that Clover injects the patched ACPI instead of native ACPI. With the techniques detailed in this guide, the changes can be made directly to the ACPI binaries provided by BIOS, skipping the extract, disassembly, and recompilation steps.

译文：

所谓“静态补丁”。为了修补 ACPI 文件，我们需要提取本地 ACPI，反汇编后进行修改，然后重新编译并将文件放在 ACPI/patched 中，然而，clover 注入这个打过补丁的 ACPI 却不是本机 ACPI。本文详述的技术（H0Tpatch），直接对 BIOS 提供的 ACPI 二进制文件进行更改（更名），跳过提取，反汇编和重新编译步骤。

理解：

1 “clover 注入这个打过补丁的 ACPI 却不是本机 ACPI”

首先，区分一下什么是**本机 ACPI**？什么是**打过补丁的 ACPI**。

ACPI 是一种规范，即高级配置与电源管理。ACPI 由很多描述文件组成（大约 20 几个），DSDT 和 SSDT 都在其中，这些都是二进制文件（像 BIOS），是机器本身提供的，这些 ACPI 就是**本机 ACPI**。

本机 ACPI 是否加载取决于 clover，当 clover 发现 patch 下有 DSDT 时，就不会加载**本机 ACPI** 的 **DSDT** 了。而 patch 下的这个 DSDT 就是你提取并**打过补丁的 DSDT**。

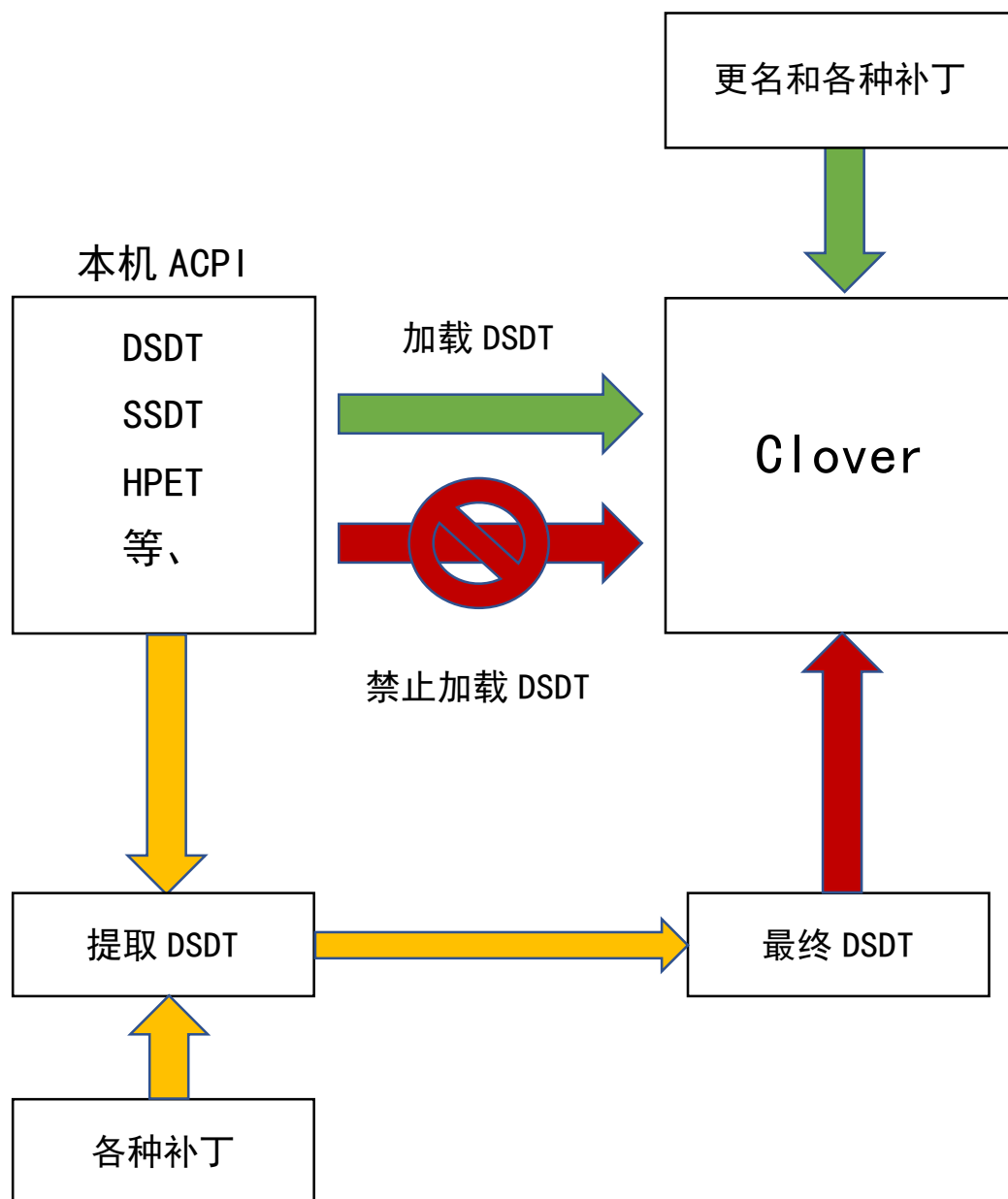
提取的 DSDT 是二进制文件，必须通过编译器进行反编译我们才能读懂。在重新编译时，会有一些错误和警告，需要修补才能重新编译成功。实际上，BIOS 提供的 DSDT 几乎没有错误，是反编译过程中产生的。

2 “直接对 BIOS 提供的 ACPI 二进制文件进行更改”

cover 提供了直接对**本机 ACPI**（二进制）进行更改的功能。这个功能为实现 HOTpatch 创造了条件。通过更名，我们可以让某个方法失效，而代替它的是我们的第三方文件，并参与和 ACPI 一起工作。

3 如果我们对**本机 ACPI** 二进制文件更改得当，HOTpatch 可以在更新 BIOS 后仍然很好的工作，甚至可以跨平台工作，除了 BIOS 更新后，更改了我们要修改的那个数据。

下图是 HOTpatch 和传统方法的对比：



HOTpatch 完整过程



传统方法：DSDT 提取过程和生成过程



传统方法：DSDT 加载过程

二、HOTpatch 优缺点

1 优点

可以说，HOTpatch 比传统方法更稳定，更可靠，适应性更强，适用范围也更广。

2 缺点

假如你想在 DSDT 的一个方法中加入一行代码，传统方法很简单，你想怎么做就怎么做。可是，HOTpatch 要完成这样的事情通常是屏蔽这个方法（更名），然后，复制这个方法所有内容，生成第三方文件，然后加入那一行代码。如果这个方法内容较多，HOTpatch 即笨拙，又无奈。当然，有很多取巧的办法可以完成，只是更名方法中某个名称失败率较高。

RM 的 HOTpatch 文件中唯独没有电池 patch，源于苹果按字节读取电池信息，而大多数机器的 ACPI 提供双字节数据。为了让电池工作正常，我们要拆分数据。被拆分数据的地址必须和内存地址一一对应，不像其他 HOTpatch 文件那样，只需对 DSDT 中的方法更名，再替换成第三方文件。电池的问题限制了跨平台工作的范围和可能性。虽然这样，HOTpatch 仍然很优秀。

三、HOTpatch 更名补丁（HOTpatch 核心一）

简单说就是对 DSDT 某个方法的名称更名，使它失效。然后，你的第三方文件中重新命名那个名称，让它按自己的方式工作。比如：DSDT 的唤醒名称是_WAK，我们将_WAK 更名为 ZWAK，原来 DSDT

关于_WAK 的部分失效了。而第三方文件重新命名_WAK 这个名称，内容我们自己重写，自然，工作过程就是我们想要的了。

有些更名不能简单更名，针对方法的更名才起作用。比如，有几处“sss”，只有一个方法是方法，其他是另外一个方法调用“sss”，全部更名不会起到任何作用，只能更名是方法的那个“sss”才奏效。

利用 clover 的强大功能就能实现 DSDT 更名操作，就在 config.plist 文件内完成。

四、HOTpatch 文件（HOTpatch 核心二）

SSDT-xxx.aml 就是 HOTpatch 第三方文件。SSDT-xxx.aml 不再依赖 BIOS 版本，甚至不依赖于机器平台。其中 SSDT-Config.aml 是 HOTpatch 文件中的重要配置文件，它决定其他第三方文件的工作方式。HOTpatch 是否能够跨平台工作，取决于 HOTpatch 更名补丁、配置文件和第三方文件的一致性。

五、HOTpatch 实现方法

通过 2 步即可实现 HOTpatch。第一步：在 config 文件中的 DSDT\Pahtchs\输入更名补丁；第二步：编制第三方文件，编译成 aml 格式放到 clover\ACPI\Patched。

RehabMan 提供了大量的第三方文件，为黑苹果做出了巨大贡献，本人由衷感谢和钦佩！在这些文件中，某些文件已经支持不同平台的机型。我们要做的是，针对自己的机型对个别补丁编制出第三方

文件。

六、HOTpatch 使用方法

参见 HOTpatch-使用方法。

=====

附 1：本 HOTpatch 文件命名规则：

1，文件名：

格式：SSDT-??? 或者 SSDT-???# 或者 SSDT-???_xxx

???：RehabMan 原文件名称

#：在原文件基础上略加修改

_xxx：表示特殊的 HOTpatch 文件，一般指某个系列机型

比如：SSDT-IGPU、SSDT-IMEI 是 RM 的原文件，未做任何修改。

SSDT-LPC#是电源管理文件，“#”表示修改了 ID 或者增加了适合本机的 ID。SSDT- FnKEY_Think 是 ThinkPad 机型关于 FN 快捷键的文件。SSDT-BATT_x30 表示 3 代机型，例如：230，430，530 等。

2，颜色

用了三种颜色来表示 HOTpatch 文件适用范围。

黄色：配置文件。

红色：有一定的特殊性，一定时期的 ThinkPad 笔记本。

绿色：通用性文件。

附 2： H0Tpatch 文件更新网站以及相关链接

1 RehabMan 的 H0Tpatch：

<https://github.com/RehabMan/OS-X-Clover-Laptop-Config/tree/master/hotpatch>

2 RehabMan 驱动：

<https://bitbucket.org/RehabMan>

3 其他参考文献：

<https://www.tonymacx86.com/threads/guide-using-clover-to-hotpatch-acpi.200137/>

<https://blog.neroxps.cn/blog/macOS/Hotpatch.html>