

Linux笔记本基于“敲打”的命令 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/144/2021_2022_Linux_E7_AC_94_E8_AE_c103_144107.htm 有史以来第一次，您可以敲打一下计算机并得到有意义的响应！使用 Linuxreg.) ThinkPads 上的嵌入式加速器，然后处理加速器的数据来读取特定“敲打”事件序列（也就是您使用关节敲打笔记本的事件序列），并基于这些敲打事件运行一些命令。双击锁定屏幕，然后敲入密码来解锁。敲打显示屏一次就可以让 MP3 播放器前进一个音轨。这类可能事物是无穷无尽的。2003 年，IBM 开始发行集成了加速器和相关软件的 ThinkPad 笔记本，以便在笔记本掉到地上时对硬盘进行保护。来自 IBM 和其他地方的黑客已经为 Linux 内核开发了一些模块来利用这些传感器的优点。屏幕显示方向、桌面切换、甚至是游戏控制和实时的笔记本倾斜度 3D 模块现在都已经可以使用了。本文将展示“敲打代码”这种新技术和一个简单程序，该程序在检测到特定的敲打代码时会运行一些命令。使用带有 HDAPS 驱动的已更新的内核，我们就可以用一个简单程序 knockAge 来生成敲打代码了。我们也可以下载并使用一个 Perl 脚本来定制自己的敲打输入环境。请参阅本文最后的下载和参考资料部分给出的链接，其中包括了解 knockAge 操作的链接。硬件需求容易实现正如您从展示视频（请参见下面参考资料中的链接）可以看到的，敲打操作是由一系列指节敲打构成的。尽管 ThinkPad 的加速器是为保护它不受意外事件的影响，但太强力的振荡仍然会对硬盘造成损坏。因此我们必须小心。很多在 2003 年以及这以后生产的 IBM（现在是 Lenovo）的

ThinkPads 中都有 HDAPS 硬件。如果您不确定自己的硬件配置，可以检查 Lenovo 的 Web 站点上关于您自己型号的机器的技术细节。如果您的机器上没有 ThinkPad，那么这段代码可能无法在您的笔记本上正常工作。本文是在 x86 体系架构上编写的。本文中的代码是在 ThinkPad T42p 的两个不同模块上进行开发和测试的。有关 ThinkPad 硬件的链接，请参阅参考资料部分。如果您有一台 Apple MacBook，那么您可能也有这种加速器，并且可以使用相同的方法，通过内核访问它们。然而，本文中的代码并没有在 Apple 硬件上进行测试。软件要求 HDAPS 驱动程序必须包括在内核中才能启用对加速器的访问。试图对现有内核增加补丁也不会获得成功，因此我们建议从自己喜欢的镜像站点上下载最新的内核。新内核发行版中已经包含了对 HDAPS 驱动程序的支持。启动内核配置选择程序，并在配置中包含 HDAPS 驱动程序。HDAPS 驱动程序位于 Device Drivers > Hardware Monitoring Support > IBM Hard Drive Active Protection System (hdaps) 选项中。更多的内核配置和安装过程已经超出了本文的范围，但是在 Web 站点上有很多教程可以提供具体的帮助；有关可以帮助我们入门的链接，请参阅参考资料一节的内容。本文是在 2.6.15.1 版本的内核上进行开发和测试的。创建简单的敲打序列从下载一节的链接中下载源代码，并从中找到 knockAge.pl 脚本。这就是让我们可以创建敲打序列的主要 Perl 程序，它还允许监听特殊的敲打序列并运行命令。下面让我们来介绍一下这个用户空间程序的用法，以及 knockAge.pl 程序的配置，然后再对这个函数进行回顾。使用下面的命令运行 knockAge.pl 程序：`perl knockAge.pl -c` 这会启动 Perl 程序来监听敲打事件并

记录下它们之间的间距以供将来使用。一旦程序开始运行之后，对笔记本进行的敲打操作就会产生效果。我们并不需要在物理上移动自己的 ThinkPad 来注册敲打事件，如果 ThinkPad 在一个平面上，只要对其进行一些移动和滑行即可。我建议您用左右握住 ThinkPad 左边接近连接轴的地方，同时用右手在距离 LCD 底部 3 英寸的地方敲打显示屏即可。请参阅 下载 部分给出的视频展示，或参阅 参考资料 中用来创建敲打序列的例子。体验不同的敲打幅度和力度，从而了解 knockAge 程序能够捕获的事件判断率。对于创建复杂的敲打事件来说，这非常重要。第一次真正尝试敲打应该非常简单，两次双击之间停留 0.5 秒，然后再次运行 `perl knockAge.pl -c`，在看到“enter a knock sequence”时稳定地敲打 LCD 边上两次，中间停留 0.5 秒。在 4 秒之后会自动超时（这是可以配置的），您所敲打的序列会被打印出来，这类似于下面的例子：`0 540031 _#_ (command here) _#_` 让我们来分析一下这一行的内容：敲打序列，分隔符，命令区，分隔符，最后是注释区。我们的下一个步骤是将这行内容复制到 knockAge.pl 程序使用的默认配置文件 `{$HOME}/.knockFile` 中，该配置文件也可能是 `/home//.knockFile` 文件。在使用上面的敲打序列行创建好 .knockFile 文件之后，就可以对这行进行修改来运行程序了。将 `(command here)` 文本修改成 `/bin/echo "double tap"`，并将注释区的内容修改成更有意义的内容，例如：`0 540031 _#_ /bin/echo "double tap" _#_ Double tap event` 现在我们已经修改好这个配置文件，可以打印一条通知了，接下来使用下面的命令在守护模式下运行 knockAge 脚本：`perl knockAge.pl` 这个程序会在后台安静地监听 `~/.knockFile` 所罗列的事件。请使

用相同的间隔再次双击屏幕，您会看到在屏幕上打印出了“double tap”消息。如果我们希望更详细地了解 knockAge.pl 脚本是如何工作的，那么我们可以使用下面的命令在守护模式下运行它：`perl knockAge.pl -v`使用 xscreensaver 锁定屏幕或打开屏幕创建“password”序列使用下面的命令在“create”模式下运行 knockAge.pl 程序：`perl knockAge.pl -c`现在我们需要创建一个解锁的密码序列；我建议使用“刮脸和理发的动作”。请确保每次您都可以以一贯精确的方式执行这个动作。尽管您可以通过修改参数来控制输入密码敲打操作所需要的精度，但是这仍然很难匹配精确的时间。“刮脸和理发动作”除了可以提供稳定的击打顺序之外，其复杂性和简单性对于屏保解锁密码来说也非常适合。下面是一个“刮脸和理发动作”的击打序列示例：`0 564025 1185795 621350 516038 960035 444421` `_#_ /bin/echo "shave the haircut" _#_ two bits` 在进行下一步操作之前，您应该体验一下上面的命令和 `~/.knockFile` 配置文件中的双击命令。这可以在屏保运行时提供很好的帮助，它更难检测出敲打是否正确。xscreensaver 所使用的命令配置 以下设置假设您已经登录到了窗口管理器中，并且已经使用您的 `userid` 启动了 xscreensaver 程序。例如，如果您正在运行 Fedora Core 4，并且使用 `gdm` 登录到 KDE 中，那么 xscreensaver 就会自动启动。因此，要激活它，则需要将双击命令从：`/bin/echo "double tap"`修改为：`xscreensaver-command -activate amp`. 这个命令会停止当前运行的所有 xscreensaver 程序，然后在后台再重新启动 xscreensaver。现在我们可以通过敲打屏幕边来重复加锁和解锁计算机屏保的过程。这比蓝牙提供的近似度加锁更加安全

或更方便吗？答案可能是否定的。它更酷吗？当然！更多例子HDAPS 传感器和 knockAge.pl 程序提供了另外一种用户输入设备，我们可以使用它们以独特的方式进行输入。例如：如果计划在一个基础上测试新的 X 配置文件，可以将双击条目更改为重新启动配置好的 X 服务器。这样就不需要敲任何其他键来强制重启了。在命令区中可以放上我们喜欢使用的任何 shell 脚本，这样就可以使用双击来查看 e-mail。以最新的组合节拍进行敲打，让 ThinkLight 显示 WWII 代在 Kinakuta 的黄金存储设备的 Morse 密码位置。敲入 Morse 编码，防止键盘输入被记录。请参阅参考资料部分给出的有关将 ThinkPad 的倾斜度用于游戏、显示工具的例子。或者直接跳过这部分内容，将 Threshold 变量设置为 15，这样您使劲踢一脚 ThinkPad，它就会自动重启了。knockAge.pl 代码历史和策略 Jeff Molofee 所编写的 hdaps-gl.c 是 knockAge.pl 代码的基础。Hdaps-gl.c 是一个非常好的展示程序，可以展示如何使用倾斜传感器来实时地显示有关 ThinkPad 的方向的信息。二者之间的区别是本例将时间上隔离的事件组织在一起创建了敲打事件，同时提供了相关的代码来创建并监听敲打事件序列。参数配置下面让我们来使用对时间和传感器敏感的一些参数来启动 knockAge.pl：清单 1. 主程序参数

```
require
sys/syscall.ph. # for subsecond timing
my $option = $ARGV[0] || "".
# simple option handling
# filename for hdaps sensor reads
my $hdapsFN = "/sys/devices/platform/hdaps/position".my
$update_threshold = 4. # threshold of force that indicates a
knock
my $interval_threshold = 100000. # microseconds
of time required between knock # events
my $sleep_interval =
```

0.01. # time to pause between hdaps readsmy
\$MAX_TIMEOUT_LENGTH = 4. # maximum length in seconds
of knock pattern # lengthmy \$MAX_KNOCK_DEV = 100000. #
maximum acceptable deviation between recorded # pattern values
and knocking valuesmy \$LISTEN_TIMEOUT = 2. # timeout value
in seconds between knock # events when in listening mode 100Test
下载频道开通，各类考试题目直接下载。详细请访问
www.100test.com