

关于电源管理器

吕凯

内容:

一、一些概念的澄清

二、基本概念

三、电源管理举例

四、需要做的工作

一些概念的澄清:

=====

<http://ftp.gnome.org/pub/GNOME/sources/gnome-power-manager>
中的源代码 doc 里面的 sleep-names.html 里面详细介绍。

1.简介:

sleep 相关的术语在不同的人有不同的观点。

开发 gnome-power-manager (一个在 hal 中的电源管理)的时候许多人会混淆一些概念。

希望这篇文档能够让一些不正式的软件遵守最常用的命名方式。

2.Standby

Standby 是这样的一种动作: Cpu 以一种低电耗的状态运行着, 但是并没有数据被保存在 RAM 或者硬盘上。standby 和 resume 一般会耗费很少的一点时间, 如果你在便携式电脑上工作的时候移走了电池, 那么, 你的工作将会丢失。

推荐的术语:

正向动作:STANDBY

反向动作: CONTINUE(或者在用户手册或翻译的时候使用 RESUME)

3.Suspend

Suspend 是这样的一种动作: 电脑会冻结一切的活动, 把工作数据拷贝到 RAM 上面。然后关闭掉屏幕, 以非常低电耗的状态模式运行。suspend 一般会花费几秒钟, resume 也会花费几秒钟。

推荐的术语:

正向动作: SUSPEND

反向动作:RESUME

Suspend 非常适合描述计算机进入一个临时睡眠但是没有完全断电的状态。用户应该能理解, suspending 工作和 hibernating 是不一样的。你在 hibernated 的时候可以启动到其他的操作系统, 或者更换电池。

Resume 非常适合描述用户 un-suspends 电脑, 然后重新开始所有的工作。这之前也在 pm-tools, HAL, 和 kernel 以及 Microsoft Windows 里面使用。

不好的术语:

Sleep

Standby

Suspend-to-RAM

Sleep 没有涉及到时间相关的内容, 所以它并不是好的描述方式, 并且 sleep 在过去许多年来都用来表示 standby 和 suspend 以及 hibernate.

Standby 并没有指明究竟发生了什么, “standby 什么呢?”, 并且在内核中使用了, 在 ACPI 电源保存状态的时候, 很少会使用到。

Suspend-to-RAM 也是个不好的描述, 它让用户感觉需要知道抽象的硬件知识。s-t-r 可能对于程序员来说是不错的, 但是对于新用户来说并不直观。

4.Hibernate

Hibernate 是这样的一种动作: 计算机冻结所有的活动, 把数据转存到硬盘上面, 关闭掉显示器, 并且关闭电源。可能 Hibernate 会消耗超过一分钟的时间把数据从 ram 上面转存到硬盘上. 并且可能在 thaw 的时候会消耗将近 40 秒。

推荐的术语:

正向动作: HIBERNATE

反向动作: THAW(或者在用户手册或翻译的时候使用 RESUME)

Hibernate 已经在 pm-tools, HAL, kernel 和 Microsoft Windows 使用了。

反向动作的名字对于最终用户 IMO 并不是很重要了。用户可能会点击一个 Suspend 按钮, 但是不会是 Resume 按钮。这样, 把 thaw 翻译成为非英语的语言的困难, 我想可能是允许加入用户可见部分的 hibernate 和 resume 为合法的描述。

不好的术语:

Wake

Suspend-to-disk

Wake 和它的反义词 sleep (不是 hibernate) 一样是不好的描述, Wake 在过去用于从 standby 中 wakeup, 从 hibernate 中 wakeup 以及从 suspend 中 wakeup。

Suspend-to-disk 也是个不好的描述, 它让用户觉得应该明白硬件的知识, 用户凭什么需要知道写到 RAM 中要比写到硬盘中要快呢?

5. 总结

把这 6 个词汇弄清楚, 会对整体的框架有一个很大的改观, 交流起来也会更方便不会引起歧义。

六个词汇从省电级别由低到高, 来简单归纳如下:

1) STANDBY, CONTINUE: cpu。

2) SUSPEND, RESUME: cpu, 显示器关闭, ram

3) HIBERNATE, THAW: cpu, 显示器, disk

基本概念

=====

1.DPMS:

是 X Display Power Management Signaling 的简称。

DPMS extension 设计的目标是提供一个传统 screen saver 的逻辑扩展。

它的执行独立于 screensaver，它和 screen saver 交互，并且 DPMS 能够服从用户或者 screen saver 应用程序。

2.VESA:

是 Video Electronics Standards Association 的简称，规定了显示器的四种电源状态：

0 DPMSModeOn 显示器打开状态中

1 DPMSModeStandby 最轻度的节省电源，切换较快，这种状态会把显示器的水平同步信号关掉。

2 DPMSModeSuspend 中度节省电源，切换速度稍慢，这种状态会把显示器的垂直同步信号关掉。

3 DPMSModeOff 最大程度节省电源，切换速度慢于前两个，它会关闭垂直和水平同步信号。

如果把 1，2，3 状态的 timeout 设置为 0 表示不使用。这三种状态的 timeout 延迟是递增的，后面的不会小于前面的，否则就是错误的。

3.APM:

高级电源管理。它使用分层的方法来管理设备。APM-aware applications 和 OS-specific 的 APM driver 通信，然后这个 driver 能够直接控制硬件设备。

APM 定义了两种接口：一种接口是 /dev 目录下面的 bios 接口，如果用户应用程序有权限，它就能够通过这个接口控制设备。另一种是 /proc 目录下的接口，用户应用程序可以通过这个接口，获得 APM 的状态和电池的信息。

尽管上面 APM 已经提供了一些信息，但是还不足以支持 HAL，所以还需要在 APM 的 proc 接口中另外一些信息。

4.HAL:

硬件抽象层。HAL 的守护进程是系统范围的服务，它负责维护一个设备的对象的数据库。这个守护进程不仅从 APM 收集电源状态信息，也会从设备信息文件 merging 信息以及管理设备的对象的生存期。这个服务以一个守护进程的形式执行，任何用户可以请求获取特定的设备信息。

HAL 可以提供 D-Bus 服务，电源管理的接口在 HAL 中进行注册，这个接口在 org.freedesktop.Hal。

当 HAL 提供系统内的 service 的时候，一个 hald 进程就会启动。

5.D-BUS:

可以实现程序之间的通信。D-bus 是 HAL 和 power policy application 的桥梁。D-bus 也提供了一些方便的工具，使得用户能够方便的与它进行通信。

6.gnome-power-manager:

这是我们的电源管理程序，它实际是一个守护进程，是 session 级别的，通过它提供的一些 D-Bus 接口，我们也可以实现电源管理功能。

7.总结

把上面的内容简单总结如下：

DPMS，是和屏幕电源管理相关的，VESA 规定了显示器的四种电源状态；APM 是较为底层的电源管理机制，电源

管理方面 HAL 是在 APM 之上的提供更方便的 D-Bus 接口控制相关电源管理信息;gnome-power-manager 是电源管理守候进程，它运行在 HAL 之上，它接受用户发送的请求，然后把这个请求传到 hal 或者 apm 中，调用 hal 或者 apm 提供的相应接口。

电源管理控制举例:

=====

1, 关闭屏幕的方法:

```
Sxset dpms force off
```

这里最后一个参数可以是 standby, suspend, off 这样都会“关闭屏幕”，移动鼠标会恢复幕。

2, apm 命令实现电源管理的步骤如下:

1) 加载 apm 模块:

```
# modprobe apm_power
```

2) 测试 suspend 功能:

```
$apm -s
```

3, 根据 hal 信息，利用 D-Bus 接口发送命令实现 suspend 功能:

```
dbus-send --system --print-reply --type=method_call --dest=org.freedesktop.Hal /org/freedesktop/Hal/devices/computer org.freedesktop.Hal.Device.SystemPowerManagement.Suspend int32:0
```

如果提供了 hal 层次的接口那么这样就会使机器进入 suspend 状态。

=====

1. 深入 Hal 研究

2. 提取 gnome-power-manager 的接口

3. 利用 gnome-power-manager 接口实现电源管理

4 深入理解 gnome-power-manager 的工作机制