Hard disk shock protection
=========================

Author: Elias Oltmanns <eo@nebensachen.de>
Last modified: 2008-10-03

0. Contents
----

1. Intro

ATA/ATAPI-7 specifies the IDLE IMMEDIATE command with unload feature.
Issuing this command should cause the drive to switch to idle mode and
unload disk heads. This feature is being used in modern laptops in
conjunction with accelerometers and appropriate software to implement
a shock protection facility. The idea is to stop all I/O operations on
the internal hard drive and park its heads on the ramp when critical
situations are anticipated. The desire to have such a feature
available on GNU/Linux systems has been the original motivation to
implement a generic disk head parking interface in the Linux kernel.
Please note, however, that other components have to be set up on your
system in order to get disk shock protection working (see
section 3. References below for pointers to more information about
that).

2. The interface

For each ATA device, the kernel exports the file
block/*/device/unload_heads in sysfs (here assumed to be mounted under
/sys). Access to /sys/block/*/device/unload_heads is denied with
-EOPNOTSUPP if the device does not support the unload feature.
Otherwise, writing an integer value to this file will take the heads
of the respective drive off the platter and block all I/O operations
for the specified number of milliseconds. When the timeout expires and
no further disk head park request has been issued in the meantime,
normal operation will be resumed. The maximal value accepted for a
timeout is 30000 milliseconds. Exceeding this limit will return
-EOVERFLOW, but heads will be parked anyway and the timeout will be
set to 30 seconds. However, you can always change a timeout to any
value between 0 and 30000 by issuing a subsequent head park request
before the timeout of the previous one has expired. In particular, the
total timeout can exceed 30 seconds and, more importantly, you can
cancel a previously set timeout and resume normal operation
immediately by specifying a timeout of 0. Values below -2 are rejected
with -EINVAL (see below for the special meaning of -1 and -2). If the
timeout specified for a recent head park request has not yet expired,

reading from /sys/block/*/device/unload_heads will report the number
of milliseconds remaining until normal operation will be resumed;
otherwise, reading the unload_heads attribute will return 0.

For example, do the following in order to park the heads of drive
/dev/sda and stop all I/O operations for five seconds:

echo 5000 > /sys/block/sda/device/unload_heads

A simple

cat /sys/block/sda/device/unload_heads

will show you how many milliseconds are left before normal operation
will be resumed.

A word of caution: The fact that the interface operates on a basis of
milliseconds may raise expectations that cannot be satisfied in
reality. In fact, the ATA specs clearly state that the time for an
unload operation to complete is vendor specific. The hint in ATA-7
that this will typically be within 500 milliseconds apparently has
been dropped in ATA-8.

There is a technical detail of this implementation that may cause some
confusion and should be discussed here. When a head park request has
been issued to a device successfully, all I/O operations on the
controller port this device is attached to will be deferred. That is
to say, any other device that may be connected to the same port will
be affected too. The only exception is that a subsequent head unload
request to that other device will be executed immediately. Further
operations on that port will be deferred until the timeout specified
for either device on the port has expired. As far as PATA (old style
IDE) configurations are concerned, there can only be two devices
attached to any single port. In SATA world we have port multipliers
which means that a user-issued head parking request to one device may
actually result in stopping I/O to a whole bunch of devices. However,
since this feature is supposed to be used on laptops and does not seem
to be very useful in any other environment, there will be mostly one
device per port. Even if the CD/DVD writer happens to be connected to
the same port as the hard drive, it generally *should* recover just
fine from the occasional buffer under-run incurred by a head park
request to the HD. Actually, when you are using an ide driver rather
than its libata counterpart (i.e. your disk is called /dev/hda
instead of /dev/sda), then parking the heads of one drive (drive X)
will generally not affect the mode of operation of another drive
(drive Y) on the same port as described above. It is only when a port
reset is required to recover from an exception on drive Y that further
I/O operations on that drive (and the reset itself) will be delayed
until drive X is no longer in the parked state.

Finally, there are some hard drives that only comply with an earlier
version of the ATA standard than ATA-7, but do support the unload
feature nonetheless. Unfortunately, there is no safe way Linux can
detect these devices, so you won't be able to write to the
unload_heads attribute. If you know that your device really does
support the unload feature (for instance, because the vendor of your

laptop or the hard drive itself told you so), then you can tell the
kernel to enable the usage of this feature for that drive by writing
the special value -1 to the unload_heads attribute:

echo -1 > /sys/block/sda/device/unload_heads

will enable the feature for /dev/sda, and giving -2 instead of -1 will
disable it again.

3. References

There are several laptops from different vendors featuring shock
protection capabilities. As manufacturers have refused to support open
source development of the required software components so far, Linux
support for shock protection varies considerably between different
hardware implementations. Ideally, this section should contain a list
of pointers at different projects aiming at an implementation of shock
protection on different systems. Unfortunately, I only know of a
single project which, although still considered experimental, is fit
for use. Please feel free to add projects that have been the victims
of my ignorance.

- http://www.thinkwiki.org/wiki/HDAPS
  See this page for information about Linux support of the hard disk
  active protection system as implemented in IBM/Lenovo Thinkpads.

4. CREDITS

This implementation of disk head parking has been inspired by a patch
originally published by Jon Escombe <lists@dresco.co.uk>. My efforts
to develop an implementation of this feature that is fit to be merged
into mainline have been aided by various kernel developers, in
particular by Tejun Heo and Bartlomiej Zolnierkiewicz.