

Battery Powered Linux Mini-HOWTO

David Lechnyr

david@lechnyr.com

Revision History

Revision 2.0	2002-10-08	Revised by: drl
Major updates and conversion to DocBook SGML		
Revision 1.0	1997-12-21	Revised by: hm
Initial draft by Hanno Muller		

This document describes how to optimize and configure power management on a ready-configured Linux system for use on battery-powered laptops. This will be helpful for everyone who runs Linux on a portable computer system. APM and ACPI methods of power management will be discussed, along with general power saving tips and techniques. There is also some discussion about the different types of batteries available.

Table of Contents

<u>1. Introduction</u>	1
<u>1.1. Copyright Information</u>	1
<u>1.2. Disclaimer</u>	1
<u>1.3. About this document</u>	1
<u>2. Power Management</u>	2
<u>2.1. Advanced Power Management (APM)</u>	2
<u>2.2. Advanced Configuration and Power Interface (ACPI)</u>	2
<u>2.3. APM vs. ACPI: Which one?</u>	3
<u>3. Power Management Techniques</u>	5
<u>3.1. Power saving – The obvious stuff</u>	5
<u>3.2. Power saving – The semi-obvious stuff</u>	5
<u>3.3. Power saving – The not-so-obvious stuff</u>	6
<u>4. Types of Batteries</u>	8
<u>4.1. Nickel Cadmium (Ni-Cd)</u>	8
<u>4.2. Nickel Metal Hydride (Ni-MH)</u>	8
<u>4.3. Lithium Ion (Li-ion)</u>	8
<u>4.4. Smart Batteries</u>	8
<u>4.5. General Battery Care</u>	9

1. Introduction

The Linux Battery Powered Mini-HOWTO has undergone some significant enhancements from its last publication in 1997. Specific attention will be paid towards recent advances in the Linux realm with ACPI, specific power saving techniques, and general battery issues.

1.1. Copyright Information

This document is the result of the work and contribution of many individuals. As such, it is released under the terms of the GNU Free Documentation License. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation. A copy of the license is available at <http://www.gnu.org/licenses/fdl.txt>

1.2. Disclaimer

Power management can be tricky stuff. As such, no liability for the contents of this document can be accepted. Use the concepts, examples and other content at your own risk. This information is provided "as is" without any warranty, condition, or representation of any kind, either express or implied, including but not limited to, any warranty respecting non-infringement, and the implied warranties of conditions of merchantability and fitness for a particular purpose. In no event shall anyone connected to this document be liable for any direct, indirect, special, incidental, consequential or other damages howsoever caused whether arising in contract, tort, or otherwise, arising out of or in connection with the use or performance of the information contained in this document.

1.3. About this document

This document was lovingly handcrafted with vim on a Dell Latitude C400 running Linux 2.4.20-pre8 with ACPI Patch 20021002, in case anyone asks.

Many thanks to Andrew Grover of the ACPID project whose patience throughout many questions is much appreciated.

2. Power Management

If you have a relatively recent x86 laptop, odds are it supports either Advanced Power Management (APM) or Advanced Configuration and Power Interface (ACPI). ACPI is the newer of the two technologies and puts power management in the hands of the operating system, allowing for more intelligent power management than is possible with BIOS controlled APM. You can only have one power management interface in control of your machine at a time, so it's important you decide which method best suits your situation.

2.1. Advanced Power Management (APM)

Advanced Power Management (APM) is a specification for allowing the BIOS to control your system's power management without the knowledge of the operating system. The advantages to APM under Linux are that it's stable and has been around for some time, although not much development has happened with it over the past few years.

To use it, you'll need to enable APM in the kernel and fetch the APM daemon from <http://www.worldvisions.ca/~apenwarr/apmd/>

The APM daemon is actually made up of three primary programs:

- apmd – handles power management tasks
- apm – a command–line tool to print the current battery status or suspend the computer
- xapm – a simple battery meter for X

Currently, most major Linux distributions have not started to configure ACPI into their kernel, choosing to stick with APM. Therefore, configuration may be easier with APM depending on your distro's included utilities.

If you're looking for a simple, "works out of the box" approach to power management for your Laptop, APM is definitely the way to go.

2.2. Advanced Configuration and Power Interface (ACPI)

Advanced Configuration and Power Interface (ACPI) is the successor to APM, which places the responsibility of power management away from the BIOS and into the hands of the operating system. ACPI Linux is newer than APM Linux, more flexible in responding to power management events, has seen much more development as of late, and as a result of all this is prone to its own share of bugs from time to time.

If you're into cutting–edge development, are running a relatively recent (>2.4.10) kernel, and are not intimidated with kernel builds and applying patches against source code, ACPI is 0 the way to go.

There are two parts to ACPI under Linux: The ACPI driver built into the kernel itself, and the ACPI daemon (ACPID). ACPID in its current incarnation is pretty simple: monitor /proc/acpi/event and do things in response. Even if you don't load the daemon, you'll still get the benefit of ACPI features built into the kernel such as processor thermal support. You can fetch ACPID from <http://acpid.sourceforge.net>

When booting your ACPI–enabled system, you can determine which version of the ACPI driver you are using by looking for something similar to this:

ACPI development is progressing at a steady rate, so you might want to consider patching your kernel against any recent updates to the kernel-level ACPI code at <http://sourceforge.net/projects/acpi/>. This is also useful if you are running into problem getting ACPI to work on your system and are experiencing *Ps_execute: method failed* ACPI error messages on boot. You may need to enable ACPI debugging in the kernel in order to determine if you are having a problem, and is generally considered a good idea.

A bit of history... Microsoft was the first vendor to implement ACPI. This is both good and bad. It is good because when you buy a system, you can pretty much guarantee that it has passed Microsoft's hardware compliance tests, including the test of its ACPI implementation. However, these tests come up short in that they do not indicate compliance with the ACPI specification, but rather with Microsoft's implementation of ACPI. When that same machine is used with Linux, some classes of errors that did not manifest themselves under Windows may become apparent. To protect against this problem, the Linux ACPI driver maintains a "bad BIOS" blacklist of known BIOS's that are known to not be ACPI compliant, and as a result will refuse to enable ACPI if your system is listed.

Many manufacturers are now validating that their systems run on Linux. However, they use major Linux distributions with the default kernel. This means that it is somewhat difficult to get OEMs to ensure that their systems work with ACPI-enabled Linux until a major Linux distribution ships an ACPI kernel. This presents a slight dilemma in that Linux distributions want to ship kernels that run on as many systems as possible, but there have been some positive moves in this area lately.

Current goals for the ACPI project include the implementation of sleep support, improvement of power-related daemons and libraries to take advantage of ACPI, and of course continued improvements in the ACPI driver itself.

For more specific background information on ACPI itself, you can visit the ACPI website at <http://www.acpi.info>

2.3. APM vs. ACPI: Which one?

There are currently two competing standards for providing power management: APM and ACPI. Both cannot be used at the same time, so which one is best for your situation? If you have a relatively recent (>2.4.10) kernel and are not intimidated by kernel builds and patching source code, you'll find many benefits with the flexibility of ACPI. If you just want to enable generic power management, or are using an older machine, choose APM. Neither method spins down idle hard drives; use `hdparm` for that instead. Either way, your system's BIOS must correctly support the power management scheme you'd like to use as well; if your system does not fully support either standard, some of the power management options might crash your system and/or cause data loss. You have been warned!

Regardless, you should be aware that even if you don't enable any power management on your laptop, on the x86 architecture Linux will always issue the "hlt" instruction to your processor whenever nothing needs to be done. This results in lowering the power consumption of your CPU. Note that the system doesn't power down when it receives the hlt instruction; it just stops executing instructions until there is an interrupt.

Some system manufacturers may have omitted the pre-ACPI tables used for SMP configuration. In this case, ACPI is required. If a system supports HyperThreading, it must use ACPI tables to discover all the virtual processors present. IA64 machines require ACPI, and NUMA servers are starting to require it for proper initialization as well. There is generally no advantage to enabling either type of power management on servers

Battery Powered Linux Mini-HOWTO

or workstations that do not fall into these categories.

If your computer's BIOS does not offer any power saving settings, you can use `hdparm` to define your harddisk's standby period. This will still help you out a lot, since harddisk activity typically consumes a lot of power.

3. Power Management Techniques

The basic goal of any power management technique is to reduce an entity's consumption. In the case of laptop power management, our focus is on decreasing CPU and hard drive usage. To make things a bit simpler, this is broken down into *obvious*, *semi-obvious*, and *non-obvious* techniques. Granted, your mileage may vary.

3.1. Power saving – The obvious stuff

There are some obvious things that you can do to reduce your system's power consumption:

- Decrease or turn off your display's backlight when you don't need it
 - While on battery, reducing the CPU clock speed will decrease power consumption, if your laptop's BIOS allows you to control this
 - Avoid using external devices (printer, crt screen, zip drive, portable camera etc.) with your computer while on battery
 - Avoid using any built in device such as a cdrom, floppy drive, or PCMCIA devices unless necessary. PCMCIA cards can also consume a lot of power, so don't leave your modem or network adapter plugged in when it is not in use
 - Use simple software. A full blown multimedia application will create a lot more system load and disk activity than a small simple word processor
-

3.2. Power saving – The semi-obvious stuff

hdparm is a Linux shell utility that can be used to improve the performance and power management of various IDE drives. If it's not included with your system, you can fetch the source from http://freshmeat.net/redirect/hdparm/4062/url_homepage/hardware For example, running ``hdparm -B128 -c1 -d1 -m16 -S60 -u1 -K1 -k1 -W1 /dev/hda`` gains me tremendous performance with added power savings. Note that your mileage may vary, and you'll want to adjust this for your specific system to prevent data loss (especially the `-B` and `-m` flags!).

Pop open an xterm window and try running ``top -i`` while your system isn't doing much of anything. Look for any active processes; any that show up are potential battery drainers. Identification is the first step; the tricky part is determining whether or not you should disable a given daemon or simply modify its behavior.

Examine your `syslog.conf` file for unnecessary logging. If you don't want to log any system activity, consider disabling `syslogd` and `klogd` entirely. If you tend to debug programs from time to time, you can modify or minimize the amount of logging your system will do. For example, to log only messages with a priority of `info` or higher except for the mail facility, which will be logged only with priorities of `error` or higher, you can try this:

```
*.info;mail.none      -/var/log/messages
mail.err              -/var/log/messages
```

The hyphen before the logfile tells the system not to sync the log file each time after logging a system message. While you could loose logfile data in the event of a system crash, buffering your data to minimize writes to your hard drive will grant multiple blessings upon your battery life!

If you want your screen to actually turn off rather than use the fake blanking effect, you'll have to edit your `XF86Config` file. In the `ServerLayout` section, you can add four options:

Battery Powered Linux Mini-HOWTO

```
Option "BlankTime"      "time" # Blank the screen
Option "StandbyTime"    "time" # Turn off screen
Option "SuspendTime"    "time" # Full hibernation
Option "OffTime"        "time" # Turn off DPMS monitor
```

If you have a Dell Inspiron or Latitude, try enabling CONFIG_I8K when recompiling the kernel and you'll be able to access information about your laptop's performance from /proc/i8k. For example, running `cat /proc/i8k` on my Dell Latitude C400 reveals:

```
1.0 A06 5X1F21 63 -22 0 1 3540 -1 0
```

Which can roughly be translated to [i8k ver] [BIOS] [serial #] [CPU temp] [Lfan status] [Rfan status] [Lfan rpm] [Rfan rpm] [ac status] [buttons status]. Significant information from this example would indicate that my CPU is currently running at 63 degrees 0 and that my right fan is on and running at 3,540 rpm.

After recompiling the kernel to support either APM or ACPI, you can gain added power-savings benefits by recompiling the Linux PCMCIA drivers as well. The package will detect that you have APM enabled (even if you are using ACPI, go figure) and will automatically add power management support within the PCMCIA system itself. This should be evidenced during the PCMCIA `make config` step with an output similar to, "Power management (APM) support is enabled". The PCMCIA source files are available from <http://sourceforge.net/projects/pcmcia-cs/>

Whether or not power management support is configured in your PCMCIA package, you should use `cardctl suspend` before suspending your laptop, and `cardctl resume` after resuming to cleanly shut down and restart your PCMCIA cards. This will not work with a modem that is in use, because the serial driver isn't able to save and restore the modem operating parameters. Some drivers, notably the PCMCIA SCSI drivers, cannot recover from a suspend/resume cycle. When using a PCMCIA SCSI card, always use `cardctl eject` prior to suspending the system. Many people refer to this as "apmcontinue" and you can use this as a key word search with your favorite search engine to find custom-made scripts for this.

Suspend-to-Disk is a tricky subject. The chief problem with it is twofold: First, it doesn't work properly under Linux ACPI (yet). Second, every laptop vendor seems to vary the methods used to create a "suspend-to-disk" or "hibernate" file, so there is no uniformity. For example, the Dell Inspiron 8000 uses Dell's phdisk utility, while the Latitude C400 requires Dell's mks2d utility. If you're really interested in getting suspend-to-disk working under Linux, find out if your vendor provides a suspend-to-disk (s2d) utility. Determine if it requires it's own partition, if that partition must be primary (not extended), etc. Most reports of successful s2d instances involve using a s2d partition on the first cylinder of the primary hard drive (e.g., /dev/hda1). If you use a boot manager that writes to the Master Boot Record (MBR), you'll probably need to modify it to write to the boot partition instead as some s2d utilities mark the s2d partition as active when suspending, which will 0 confuse both you and your boot manager. If you already have installed Linux and have never made a suspend partition, there is still hope. Using fips, parted, Partition Magic, or other similar utility to move your partition ahead leaving free space at the first cylinder of your hard drive. Then use your vendor supplied utility to format the partition.

3.3. Power saving – The not-so-obvious stuff

Disable your swap file in /etc/fstab to reduce hard drive access. If you've got lots of memory, this is definitely the way to go. One way to tell if you need your swap file is to enable it, use your system for a period of time, and examine /proc/meminfo and /proc/swaps (or just run `top` if this is unsettling to you) to determine how much free memory you've got on average, and whether or not your swap file is even being utilized. For

Battery Powered Linux Mini-HOWTO

example, my laptop has a ridiculous 512 MB RAM, and on any given day still has an average of 150 MB RAM unused with no swap file whatsoever.

Compile your kernel with TMPFS (temporary file system) enabled and mount your /tmp directory using it. The useful bit here is that nothing will be written to your hard drive on this mount point as it will act like a RAM disk (however nothing will be saved either!) The advantage of tmpfs over the more traditional ramfs is that it lives in the kernel internal cache and grows and shrinks to accommodate the files placed there. See your kernel's Documentation/filesystems/tmpfs.txt for full information. An example /etc/fstab with 100MB temporary ram file mounted on /tmp would look like:

tmpfs	/tmp	tmpfs	size=100m	0	0
-------	------	-------	-----------	---	---

However this might create problems if you're downloading a large program and your download manager uses the /tmp directory to hold files prior to full download.

Boot your system and list the currently loaded modules with lsmod. Anything listed here most likely needs to be loaded on a regular basis; compiling these in as part of your kernel rather than as loadable modules may help to decrease the amount of time they must be loaded from disk, and to a very minor degree, decrease the amount of disk access required to start your system.

Examine your crontab settings to see if anything is being run on a regular basis. Comment out any unnecessary items. Don't forget to examine every user's crontab, including the user 'nobody'. If you don't need to schedule any background activity, consider disabling crond altogether. The same advice goes for atd.

If you run httpd to test and/or develop web pages, try altering the values of MinSpareServers and StartServers to 1. Don't define any CustomLogging or at least increase the value of LogLevel to warn. If you're really sure of yourself, you can change the ErrorLog directive to point to /dev/null.

One popular practice in the past involved modifying the update/bdflush daemon to change the frequency of flushing the filesystem buffers (sync). However, newer kernels do not use a user space update daemon, so any modifications you make here will simply never take effect.

4. Types of Batteries

There are currently three types of batteries commonly used for laptops: Nickel Cadmium, Nickel Metal Hydride, and Lithium Ion.

4.1. Nickel Cadmium (Ni–Cd)

Nickel Cadmium (Ni–Cd) batteries were the standard technology for years, but today they are out of date and new laptops don't use them anymore. They are heavy and very prone to the "memory effect". When recharging a NiCd battery that has not been fully discharged, it "remembers" the old charge and continues there the next time you use it. The memory effect is caused by crystallization of the battery's substances and can permanently reduce your battery's lifetime, even make it useless. To avoid it, you should completely discharge the battery and then fully recharge it again at least once every few weeks. As this battery contains cadmium, a toxic material, it should always be recycled or disposed of properly.

NiCad batteries, and to a some degree NiMH batteries, suffer from what's called the *memory effect*. Memory Effect means that if a battery is repeatedly only partially discharged before recharging, the battery will forget that it can further discharge. The best way to prevent this situation is to fully charge and discharge your battery on a regular basis.

4.2. Nickel Metal Hydride (Ni–MH)

Nickel Metal Hydride (Ni–MH) batteries are the cadmium–free replacement for NiCad. They are less affected by the memory effect than NiCd and thus require less maintenance and conditioning. However, they have problems at very high or low room temperatures. And even though they use less hazardous materials (i.e., they do not contain heavy metals), they cannot be fully recycled yet. Another main difference between NiCad and NiMH is that NiMH battery offers higher energy density than NiCads. In other words, the capacity of a NiMH is approximately twice the capacity of its NiCad counterpart. What this means for you is increased run–time from the battery with no additional bulk or weight.

4.3. Lithium Ion (Li–ion)

Lithium Ion (Li–ion) are the new standard for portable power. Li–ion batteries produce the same energy as NiMH but weighs approximately 20%–35% less. They do not suffer from the memory effect unlike their NiMH and Ni–Cd counterparts. Their substances are non–hazardous to the 0. Because lithium ignites very easily, they require special handling. Unfortunately, few consumer recycling programs have been established for Li–ion batteries at this point in time.

4.4. Smart Batteries

Smart batteries are not really a different type of battery, but they do deserve special mention. Smart batteries have internal circuit boards with chips which allow them to communicate with the laptop and monitor battery performance, output voltage and temperature. Smart batteries will generally run 15% longer due to their increased efficiency and also give the computer much more accurate "fuel gauge" capabilities to determine how much battery run time is left before the next recharge is required.

4.5. General Battery Care

Even if the battery case looks the same, you cannot just upgrade to another battery technology unless your laptop has been pre–configured from the manufacturer to accept more than one type of battery type, since the recharging process is different for each of the three types of batteries.

A battery that is not used for a long time will slowly discharge itself. Even with the best of care, a battery needs to be replaced after 500 to 1000 recharges. But still it is not recommended to run a laptop without the battery while on ac power — the battery often serves as a big capacitor to protect against voltage peaks from your ac outlet.

As the manufacturers change the shapes of their batteries every few months, you might have problems to find a new battery for your laptop in a few years from now. This is somewhat of a concern only if you anticipate using the same laptop several years from now. If in doubt, buy a spare battery now – before it's out of stock.

New batteries come in a discharged condition and must be fully charged before use. It is recommended that you fully charge and discharge the new battery two to four times to allow it to reach its maximum rated capacity. It is generally recommend that you perform an overnight charge (approximately twelve hours) for this. Note: It is normal for a battery to become warm to the touch during charging and discharging. When charging the battery for the first time, the device may indicate that charging is complete after just 10 or 15 minutes. This is a normal with rechargeable batteries. New batteries are hard for the device to charge; they have never been fully charged and are not broken in. Sometimes the device's charger will stop charging a new battery before it is fully charged. If this happens, remove the battery from the device and then reinsert it. The charge cycle should begin again. This may happen several times during the first battery charge. Don't worry; it's perfectly normal. Keep the battery healthy by fully charging and then fully discharging it at least once every two to three weeks. Exceptions to the rule are Li–Ion batteries which do not suffer from the memory effect.

Batteries should be stored in a discharged state since they can self–discharge and may become inactive after a long storage period. They should not be stored for any length of time while connected to the laptop. High humidity and temperatures can cause the battery to deteriorate, so these should be avoided during storage.

Do not remove and carry a battery pack in your pocket, purse, or other container where metal objects (such as car keys or paper clips) could short–circuit the battery terminals. The resulting excessive current flow can cause extremely high temperatures and may result in damage to the battery pack or cause fire or burns.