



# Red Hat Enterprise Linux 6

## Управление энергопотреблением

Управление потреблением энергии в Red Hat Enterprise Linux 6  
Редакция 1.0



# Red Hat Enterprise Linux 6 Управление энергопотреблением

---

Управление потреблением энергии в Red Hat Enterprise Linux 6

Редакция 1.0

Don Domingo

Служба инженерной документации Red Hat

Rüdiger Landmann

Служба инженерной документации Red Hat

r.landmann@redhat.com

Red Hat Inc.

## Юридическое уведомление

Copyright © 2010 Red Hat Inc..

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](https://creativecommons.org/licenses/by-sa/3.0/). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## Аннотация

В этом документе рассматриваются способы эффективного управления энергопотреблением в Red Hat Enterprise Linux 6 и их влияние на производительность системы в целом.

---

## Содержание

<b>ГЛАВА 1. ОБЗОР</b> .....	<b>3</b>
1.1. ЗНАЧЕНИЕ УПРАВЛЕНИЯ ЭНЕРГОПОТРЕБЛЕНИЕМ	3
1.2. ОСНОВЫ УПРАВЛЕНИЯ ЭНЕРГОПОТРЕБЛЕНИЕМ	4
<b>ГЛАВА 2. ИНСТРУМЕНТЫ АНАЛИЗА И УПРАВЛЕНИЯ ЭНЕРГОПОТРЕБЛЕНИЕМ</b> .....	<b>6</b>
2.1. ОБЗОР АУДИТА И АНАЛИЗА	6
2.2. POWERTOP	6
2.3. DISKDEVSTAT И NETDEVSTAT	8
2.4. BLTK	12
2.5. TUNED И KTUNE	13
2.5.1. Файл tuned.conf	14
2.5.2. Tuned-adm	16
2.6. DEVICEKIT-POWER И DEVKIT-POWER	18
2.7. GNOME POWER MANAGER	19
2.8. ПРОЧИЕ СРЕДСТВА СБОРА СТАТИСТИКИ	19
<b>ГЛАВА 3. ОСНОВНАЯ ИНФРАСТРУКТУРА</b> .....	<b>21</b>
3.1. СОСТОЯНИЯ ЭНЕРГОСБЕРЕЖЕНИЯ ПРОЦЕССОРА	21
3.2. РЕГУЛЯТОРЫ CPUFREQ	21
3.2.1. Типы регуляторов CPUfreq	21
3.2.2. Настройка CPUfreq	23
3.2.3. Настройка частоты процессора с помощью CPUfreq	24
3.3. ПРИОСТАНОВКА И ВОЗОБНОВЛЕНИЕ РАБОТЫ	25
3.4. БЕЗТАКТОВОЕ ЯДРО	25
3.5. ASPM	25
3.6. ALPM	26
3.7. ОПТИМИЗАЦИЯ ДОСТУПА К ДИСКАМ RELATIME	27
3.8. ОГРАНИЧЕНИЕ ЭНЕРГОПОТРЕБЛЕНИЯ	27
3.9. УПРАВЛЕНИЕ ЭНЕРГОПОТРЕБЛЕНИЕМ ВИДЕОУСТРОЙСТВ	28
3.10. RFKILL	29
3.11. ОПТИМИЗАЦИЯ В ПРОСТРАНСТВЕ ПОЛЬЗОВАТЕЛЯ	30
<b>ГЛАВА 4. ПРИМЕРЫ</b> .....	<b>31</b>
4.1. СЕРВЕР	31
4.2. НОУТБУК	32
<b>ПРИЛОЖЕНИЕ А. РЕКОМЕНДАЦИИ ДЛЯ РАЗРАБОТЧИКОВ</b> .....	<b>35</b>
А.1. ИСПОЛЬЗОВАНИЕ ПОТОКОВ	35
А.2. ПРОБУЖДЕНИЯ	36
А.3. FSYNC	36
<b>ПРИЛОЖЕНИЕ В. ИСТОРИЯ ИЗМЕНЕНИЙ</b> .....	<b>38</b>



# ГЛАВА 1. ОБЗОР

При разработке Red Hat Enterprise Linux 6 эффективному управлению потреблением энергии уделялось особое внимание. Энергосбережение имеет огромное значение для реализации «зеленых» технологий, которые охватывают множество аспектов, включая переработку материалов, экологически чистое производство систем, разработку и планирование экологически чистых систем. В этом документе будут рассмотрены вопросы управления энергопотреблением систем Red Hat Enterprise Linux 6.

## 1.1. ЗНАЧЕНИЕ УПРАВЛЕНИЯ ЭНЕРГОПОТРЕБЛЕНИЕМ

В основе эффективного управления лежит понимание принципов потребления энергии индивидуальными компонентами системы. Для достижения максимальной производительности надо иметь представление о выполняемых системой задачах и способах настройки компонентов.

Основная цель управления энергопотреблением:

- уменьшение энергозатрат в целом.

Управление энергопотреблением позволяет достичь следующего:

- уменьшение тепловыделения серверов и вычислительных центров;
- уменьшение вторичных затрат на охлаждение, дополнительное пространство, проводку, генераторы и источники бесперебойного питания;
- продление срока службы батареи для ноутбуков;
- снижение выделения диоксида углерода;
- соответствие государственным и международным экологическим стандартам (например, Energy Star);
- соответствие экологическим стандартам отдельных предприятий.

Как правило, уменьшение потребления энергии определенного компонента или всей системы снижает не только тепловыделение, но и производительность. Поэтому сначала надо оценить, какие потери производительности можно позволить.

Изучение типов выполняемых системой задач и оптимальная настройка каждого компонента позволит сэкономить энергию, снизить тепловыделение и продлить срок работы батареи ноутбуков. В чем-то анализ и настройка энергопотребления системы схожи с настройкой производительности, но подходы к настройке системы отличаются, так как конечной целью является либо максимальная производительность, либо оптимальное потребление энергии. В этом документе рассматриваются методы оптимизации и соответствующие программные инструменты Red Hat.

Red Hat Enterprise Linux 6 включает множество инструментов управления энергопотреблением, используемых по умолчанию. Стандартные настройки должны вполне подойти для типичных серверов и настольных систем. Но в отдельных случаях, когда максимальная производительность с минимальной задержкой абсолютно необходима, может потребоваться их изменить.

Чтобы решить, действительно ли необходима оптимизация производительности компьютеров с использованием описанных здесь методов, ответьте на следующие вопросы:

**Вопрос:** Действительно ли нужна оптимизация?

**Ответ:** Необходимость в оптимизации энергопотребления может определяться требованиями вашего предприятия или законодательством.

---

**Вопрос:** Какая степень оптимизации необходима?

**Ответ:** Некоторые рассмотренные в этом документе методологии не требуют детального анализа системы, вместо этого предлагая predetermined набор способов оптимизации, позволяющий улучшить производительность в целом. Безусловно, максимальной эффективности можно достичь лишь при индивидуальном анализе и оптимизации системы.

---

**Вопрос:** Не снизится ли при оптимизации производительность системы до неприемлемого уровня?

**Ответ:** Многие рассмотренные в этом документе способы оптимизации оказывают влияние на производительность системы. Если стандартных методов управления энергопотреблением Red Hat Enterprise Linux 6 недостаточно, следите за производительностью системы после оптимизации и определите, какие потери производительности допустимы в вашем случае.

---

**Вопрос:** Оправдывают ли затраченные на оптимизацию ресурсы и время полученный результат?

**Ответ:** Оптимизация одной системы вручную не стоит затраченного времени и средств. С другой стороны, оптимизация нескольких тысяч офисных компьютеров с идентичной конфигурацией будет эффективной.

---

---

---

Далее будет рассказано о том, как оптимальная производительность оборудования может уменьшить потребление энергии.

## 1.2. ОСНОВЫ УПРАВЛЕНИЯ ЭНЕРГОПОТРЕБЛЕНИЕМ

Ниже перечислены принципы, на которых строится эффективное управление потреблением энергией.

### **Бездействующий процессор должен активироваться только при необходимости**

Раньше ядро Red Hat Enterprise Linux 5 для каждого процессора использовало таймер, предотвращавший переход процессора в энергосберегающий режим в силу необходимости обработки периодических событий таймера (с перерывом в несколько миллисекунд в зависимости от настроек), даже несмотря на отсутствие выполняющихся процессов. Необходимо отметить, что эффективное энергосбережение обеспечивается за счет уменьшения частоты пробуждения процессора.

По этой причине периодический таймер был исключен из ядра Red Hat Enterprise Linux 6, а бездействующее состояние процессора теперь является *безтактовым* и позволяет избежать излишнего потребления энергии. Но эффективность такого решения может пострадать, если в системе выполняются программы, инициирующие ненужные события таймера, такие как контроль изменения звука, перемещения мыши и т.п.



Red Hat Enterprise Linux 6 предоставляет инструменты для анализа потребления процессорных ресурсов программами (см. [Глава 2, Инструменты анализа и управления энергопотреблением](#)).

### **Неиспользуемое оборудование и устройства должны быть отключены**

Это особенно важно для устройств с подвижными элементами, таких как жесткие диски. Плюс ко всему некоторые приложения могут оставить незанятые устройства открытыми — в этом случае ядро будет подразумевать, что устройства продолжают использоваться и не могут перейти в энергосберегающий режим.

### **При низкой активности должно потребляться меньше энергии**

Во многих случаях это зависит от оборудования и настроек в BIOS. Более старые компоненты системы могут не поддерживать новые возможности Red Hat Enterprise Linux 6. Убедитесь, что в системах используются последние доступные микропрограммы и включены функции управления энергопотреблением BIOS. Эти функции включают:

- SpeedStep;
- PowerNow!;
- Cool'n'Quiet;
- ACPI (C-состояние);
- Smart.

Если оборудование поддерживает перечисленные возможности и они включены в BIOS, Red Hat Enterprise Linux будет их использовать по умолчанию.

### **Формы состояний процессора**

Современные процессоры в комбинации с интерфейсом ACPI (Advanced Configuration and Power Interface) поддерживают несколько состояний, характеризующихся уровнем потребляемой энергии:

- бездействие (C-состояния);
- частота (P-состояния);
- тепловыделение (T-состояния).

Процессор в наиболее глубоком энергосберегающем режиме потребляет меньше всего энергии, но при этом его пробуждение занимает больше времени. В исключительно редких случаях пробуждение процессора происходит сразу после перехода в энергосберегающий режим, в результате чего он будет постоянно занят и эффективность энергосбережения падает, поэтому состояние процессора надо выбирать аккуратно.

### **Выключенный компьютер потребляет меньше всего энергии**

Впрочем, это очевидно. Выключение компьютера является самым эффективным способом экономии энергии. Ваша организация даже может разработать экологический план, подразумевающий выключение компьютеров на ночь и во время обеденного перерыва. Дополнительно с помощью технологии виртуализации Red Hat Enterprise Linux 6 можно обеспечить консолидацию нескольких физических серверов в один.

## ГЛАВА 2. ИНСТРУМЕНТЫ АНАЛИЗА И УПРАВЛЕНИЯ ЭНЕРГОПОТРЕБЛЕНИЕМ

### 2.1. ОБЗОР АУДИТА И АНАЛИЗА

Детальный ручной аудит, анализ и изменение настроек являются скорее исключением в силу неоправданной стоимости и затрат времени. В то же время их выполнение всего один раз для большого числа аналогичных систем может значительно сэкономить время и усилия по восстановлению. Представьте себе преимущества при настройке тысяч настольных систем или кластера НРС, объединяющего множество идентичных компьютеров. Аудит также может выполняться с целью определения исходного состояния, относительно которого будут регистрироваться изменения поведения системы. Результаты анализа помогут избежать неприятных сюрпризов в отношении потребления энергии, если оборудование, BIOS и программное обеспечение обновляются регулярно. Несомненно, подробный анализ позволит получить полное представление о происходящих в системе событиях.

Анализ энергопотребления является довольно сложным процессом даже во многих современных системах. Большинство систем попросту не предоставляют программные способы измерения затрат энергии. Но существуют и исключения, например управляющая консоль ILO серверов Hewlett Packard включает модуль управления энергопотреблением, который можно получить через Интернет. IBM предоставляет аналогичное решение в виде модуля BladeCenter, а в компьютерах Dell за управление энергопотреблением отвечает IT Assistant. Другие производители также могут предоставлять аналогичные инструменты, хотя универсального решения не существует. Если ваша система не включает такие функции, существует несколько вариантов: можно установить специальный источник питания с встроенными функциями измерения мощности. Например, для источника Gigabyte Odin GT 550 W можно загрузить специальные программы с <http://mgmt.sth.sze.hu/~andras/dev/gopsu/>. Если же и это недоступно, в крайнем случае можно подключить внешние ваттметры (например, Watts up? PRO).

Прямое измерение энергопотребления может требоваться лишь для обеспечения максимальной экономии. Существуют различные средства для определения изменений и отслеживания поведения системы. Они будут рассмотрены в этой главе.

### 2.2. POWERTOP

Наличие безтактового ядра в Red Hat Enterprise Linux 6 (см. [Раздел 3.4, «Безтактовое ядро»](#)) допускает более частый переход процессора в режим бездействия, тем самым снижая потребление энергии. Новая программа **PowerTOP** используется для идентификации пользовательских программ и компонентов ядра, наиболее часто пробуждающих процессор. Именно с помощью **PowerTOP** осуществлялось тестирование эффективности работы программ (см. [Раздел 3.11, «Оптимизация в пространстве пользователя»](#)), в результате чего в Red Hat Enterprise Linux 6 удалось уменьшить число ненужных пробуждений процессора в десятки раз.

Установите **PowerTOP**:

```
yum install powertop
```

Команда запуска **PowerTOP**:

```
powertop
```

Для полноценной работы **PowerTOP** необходимо ее запустить в режиме root.

**PowerTOP** выполнит сбор системной статистики и предоставит список компонентов, наиболее часто отправляющих сигналы пробуждения процессору. В нижней части экрана будут предложены способы оптимизации с целью снижения энергопотребления. Так как обновление статистики **PowerTOP** осуществляется довольно часто, рекомендации будут пополняться (см. [Рисунок 2.1, «PowerTOP в действии»](#)).

В верхней части списка будет показана длительность всех R и C-состояний процессора. Чем дольше процессор находится в более глубоком состоянии (**C4** или **C3**), тем лучше, так как это служит индикатором оптимального использования процессора. В идеальной ситуации в периоды бездействия процессор будет находиться в наиболее глубоком R или C-состоянии не меньше 90% времени.

Ниже вы увидите число пробуждений в секунду, что также характеризует степень эффективности энергопотребления. Чем больше число, тем больше было затрачено энергии.

Далее показана информация об энергопотреблении системы (если доступна). Обычно показывается для ноутбуков, работающих от батареи.

Затем следует подробный список компонентов, наиболее часто отправляющих запросы пробуждения процессору. В верхней части списка приведены компоненты, на которые следует обратить особое внимание. Если среди перечисленных есть компоненты ядра (имя заключено в скобки <>), это может означать, что пробуждения вызваны конкретным драйвером. Настройка драйверов обычно требует изменений ядра. Их рассмотрение выходит за рамки этого документа. В то же время управлять пользовательскими процессами намного проще, но сначала надо решить, есть ли необходимость в выполнении той или иной службы или приложения. Чтобы насовсем отключить службу, выполните

```
chkconfig служба off
```

Для получения информации о выполняемых компонентом действиях, выполните

```
ps -awux | grep компонент  
strace -p PID
```

Если действия повторяются, не исключено, что имеет место заикливание. Для его исправления потребуется модифицировать код компонента, обсуждение чего выходит за рамки данного документа.

Наконец, в нижней части экрана **PowerTOP** предложит способы оптимизации с целью снижения энергопотребления. Так как обновление статистики **PowerTOP** осуществляется довольно часто, рекомендации будут пополняться (см. [Рисунок 2.1, «PowerTOP в действии»](#)). Изменения будут активны только до перезагрузки. Можно сделать так, чтобы настройки сохранялись между перезагрузками — **PowerTOP** покажет точную команду. Добавьте ее в файл `/etc/rc.local` для активации при каждой загрузке системы.

```

PowerTOP version 1.11      (C) 2007 Intel Corporation

Cn          Avg residency      P-states (frequencies)
C0 (cpu running)      ( 4.4%)          2.81 Ghz    3.2%
polling          0.1ms ( 0.0%)          2.81 Ghz    0.2%
C1 mwait          0.0ms ( 0.0%)          2.14 Ghz    0.1%
C2 mwait          0.5ms ( 1.1%)          1.60 Ghz    0.4%
C4 mwait          4.3ms (94.5%)          800 Mhz    96.2%

Wakeups-from-idle per second : 245.5   interval: 15.0s
no ACPI power usage estimate available

Top causes for wakeups:
 38.3% (163.7)   <kernel core> : hrtimer_start_range_ns (tick_sched_timer)
  8.8% ( 37.8)   <interrupt> : iwlgagn
  8.6% ( 36.9)   <kernel IPI> : Rescheduling interrupts
  7.9% ( 33.9)   <interrupt> : extra timer interrupt
  7.9% ( 33.7)   firefox : hrtimer_start_range_ns (hrtimer_wakeup)
  4.6% ( 19.9)   popfile.pl : hrtimer_start_range_ns (hrtimer_wakeup)
  3.2% ( 13.8)   <kernel core> : hrtimer_start (tick_sched_timer)
  2.7% ( 11.7)   <interrupt> : i915
  2.6% ( 11.2)   <interrupt> : ahci
  2.2% (  9.5)   <interrupt> : ehci_hcd:usb1
  2.2% (  9.5)   USB device 1-5.1.2 : Microsoft 3-Button Mouse with IntelliEye(TM) (Microsoft)
  2.1% (  9.0)   <kernel core> : __mod_timer (ehci_watchdog)
  1.5% (  6.5)   thunderbird-bin : hrtimer_start_range_ns (hrtimer_wakeup)
  1.3% (  5.5)   simpres.b.in : hrtimer_start_range_ns (hrtimer_wakeup)
  1.3% (  5.5)   plasma-desktop : hrtimer_start_range_ns (hrtimer_wakeup)
  1.2% (  5.3)   <interrupt> : eth0
  0.9% (  4.0)   <kernel core> : __mod_timer (rh_timer_func)
  0.2% (  1.0)   klipper : hrtimer_start_range_ns (hrtimer_wakeup)
  0.2% (  1.0)   httpd : hrtimer_start_range_ns (hrtimer_wakeup)
  0.2% (  0.9)   konversation : hrtimer_start_range_ns (hrtimer_wakeup)

Suggestion: increase the VM dirty writeback time from 5.00 to 15 seconds with:
echo 1500 > /proc/sys/vm/dirty_writeback_centisecs
This wakes the disk up less frequently for background VM activity

Q - Quit   R - Refresh   W - Increase Writeback time

```

Рисунок 2.1. PowerTOP в действии

На сайте <http://www.lesswatts.org/projects/powertop/known.php> можно найти список приложений, которые программа **PowerTOP** идентифицировала как без необходимости поддерживающие активное состояние процессора.

## 2.3. DISKDEVSTAT И NETDEVSTAT

Для сбора статистики о дисковой и сетевой активности программ **SystemTap** использует **diskdevstat** и **netdevstat**. Они в чем-то схожи с **PowerTOP**, которая для каждой программы показывает количество попыток пробуждения процессора в секунду (см. [Раздел 2.2, «PowerTOP»](#)). С помощью **diskdevstat** и **netdevstat** можно идентифицировать приложения, нерационально потребляющие энергию за счет выполнения большого числа операций ввода/вывода вместо объединения их в группы и одновременного выполнения.

Команда их установки выглядит так:

```
yum install systemtap tuned-utils kernel-debuginfo
```

Команда запуска:

```
diskdevstat
```

или

netdevstat

Обе команды принимают максимум три аргумента:

**diskdevstat** *интервал\_обновлений* *длительность* *таблица*

**netdevstat** *интервал\_обновлений* *длительность* *таблица*

*интервал\_обновлений*

Время между обновлениями экрана (в секундах). По умолчанию **5** секунд.

*длительность*

Длительность полного цикла. По умолчанию **86400** (1 день).

*таблица*

Показывает сводную таблицу полученной статистики.

Вывод команды аналогичен **PowerTOP**. Например, вывод **diskdevstat** в Fedora 10 с KDE 4.2 будет выглядеть так:

PID	UID	DEV	WRITE_CNT	WRITE_MIN	WRITE_MAX	WRITE_AVG	READ_CNT
READ_MIN	READ_MAX	READ_AVG	COMMAND				
2789	2903	sda1	854	0.000	120.000	39.836	0
0.000	0.000	0.000	plasma				
15494	0	sda1	0	0.000	0.000	0.000	758
0.000	0.012	0.000	0logwatch				
15520	0	sda1	0	0.000	0.000	0.000	140
0.000	0.009	0.000	perl				
15549	0	sda1	0	0.000	0.000	0.000	140
0.000	0.009	0.000	perl				
15585	0	sda1	0	0.000	0.000	0.000	108
0.001	0.002	0.000	perl				
2573	0	sda1	63	0.033	3600.015	515.226	0
0.000	0.000	0.000	auditd				
15429	0	sda1	0	0.000	0.000	0.000	62
0.009	0.009	0.000	crond				
15379	0	sda1	0	0.000	0.000	0.000	62
0.008	0.008	0.000	crond				
15473	0	sda1	0	0.000	0.000	0.000	62
0.008	0.008	0.000	crond				
15415	0	sda1	0	0.000	0.000	0.000	62
0.008	0.008	0.000	crond				
15433	0	sda1	0	0.000	0.000	0.000	62
0.008	0.008	0.000	crond				
15425	0	sda1	0	0.000	0.000	0.000	62
0.007	0.007	0.000	crond				
15375	0	sda1	0	0.000	0.000	0.000	62
0.008	0.008	0.000	crond				
15477	0	sda1	0	0.000	0.000	0.000	62
0.007	0.007	0.000	crond				
15469	0	sda1	0	0.000	0.000	0.000	62
0.007	0.007	0.000	crond				
15419	0	sda1	0	0.000	0.000	0.000	62

0.008	0.008	0.000	crond				
15481	0 sda1	0	0.000	0.000	0.000	0.000	61
0.000	0.001	0.000	crond				
15355	0 sda1	0	0.000	0.000	0.000	0.000	37
0.000	0.014	0.001	laptop_mode				
2153	0 sda1	26	0.003	3600.029	1290.730		0
0.000	0.000	0.000	rsyslogd				
15575	0 sda1	0	0.000	0.000	0.000	0.000	16
0.000	0.000	0.000	cat				
15581	0 sda1	0	0.000	0.000	0.000	0.000	12
0.001	0.002	0.000	perl				
15582	0 sda1	0	0.000	0.000	0.000	0.000	12
0.001	0.002	0.000	perl				
15579	0 sda1	0	0.000	0.000	0.000	0.000	12
0.000	0.001	0.000	perl				
15580	0 sda1	0	0.000	0.000	0.000	0.000	12
0.001	0.001	0.000	perl				
15354	0 sda1	0	0.000	0.000	0.000	0.000	12
0.000	0.170	0.014	sh				
15584	0 sda1	0	0.000	0.000	0.000	0.000	12
0.001	0.002	0.000	perl				
15548	0 sda1	0	0.000	0.000	0.000	0.000	12
0.001	0.014	0.001	perl				
15577	0 sda1	0	0.000	0.000	0.000	0.000	12
0.001	0.003	0.000	perl				
15519	0 sda1	0	0.000	0.000	0.000	0.000	12
0.001	0.005	0.000	perl				
15578	0 sda1	0	0.000	0.000	0.000	0.000	12
0.001	0.001	0.000	perl				
15583	0 sda1	0	0.000	0.000	0.000	0.000	12
0.001	0.001	0.000	perl				
15547	0 sda1	0	0.000	0.000	0.000	0.000	11
0.000	0.002	0.000	perl				
15576	0 sda1	0	0.000	0.000	0.000	0.000	11
0.001	0.001	0.000	perl				
15518	0 sda1	0	0.000	0.000	0.000	0.000	11
0.000	0.001	0.000	perl				
15354	0 sda1	0	0.000	0.000	0.000	0.000	10
0.053	0.053	0.005	lm_lid.sh				

Столбцы:

**PID**

Идентификатор процесса.

**UID**

Идентификатор пользователя, от лица которого выполняются приложения.

**DEV**

Устройство, где выполнялись операции ввода и вывода.

**WRITE\_CNT**

Общее число операций записи.

**WRITE\_MIN**

Минимальная длительность двух последовательных операций записи (в секундах).

**WRITE\_MAX**

Максимальная длительность двух последовательных операций записи (в секундах).

**WRITE\_AVG**

Средняя длительность двух последовательных операций записи (в секундах).

**READ\_CNT**

Общее число операций чтения.

**READ\_MIN**

Минимальная длительность двух последовательных операций чтения (в секундах).

**READ\_MAX**

Максимальная длительность двух последовательных операций чтения (в секундах).

**READ\_AVG**

Средняя длительность двух последовательных операций чтения (в секундах).

**COMMAND**

Имя процесса.

В этом примере следует обратить внимание на следующие процессы:

PID	UID	DEV	WRITE_CNT	WRITE_MIN	WRITE_MAX	WRITE_AVG	READ_CNT
2789	2903	sda1	854	0.000	120.000	39.836	0
0.000	0.000	0.000	0.000	plasma			
2573	0	sda1	63	0.033	3600.015	515.226	0
0.000	0.000	0.000	0.000	auditd			
2153	0	sda1	26	0.003	3600.029	1290.730	0
0.000	0.000	0.000	0.000	rsyslogd			

Следует обратить внимание на ненулевое значение счетчика **WRITE\_CNT**, что означает, что эти программы выполняли запись в процессе сбора статистики. Наихудший результат демонстрирует **plasma** — этот процесс выполнял запись больше всего с наименьшей задержкой между операциями. Поэтому в первую очередь надо обратить внимание именно на **Plasma** и продумать возможности оптимизации.

Более подробный анализ всех вызовов отдельных процессов можно выполнить с помощью **strace** и **ltrace**. Так, для нашего примера можно выполнить

```
strace -p 2789
```

Вывод **strace** в этом случае показал повторяющиеся действия каждые 45 секунд, открывающие файл значков KDE для записи и закрывающие его. При изменении метаданных, а точнее времени изменения, осуществлялась ненужная запись на жесткий диск. Для решения этой проблемы был наложен запрет на подобные вызовы в случае отсутствия обновлений значков.

## 2.4. BLTK

Red Hat Enterprise Linux 6 представляет комплект BLTK (**Battery Life Tool Kit**) для эмуляции нагрузки с целью последующего анализа срока действия батареи и производительности системы. BLTK выполняет различные задания, имитирующие деятельность разного рода пользователей, и позволяет анализировать полученные результаты. Несмотря на то, изначальным назначением BLTK служил анализ производительности ноутбуков, с помощью параметра **-a** его можно также использовать для тестирования настольных компьютеров.

BLTK позволяет генерировать большие объемы нагрузки, приближенные к реальным. Например, в режиме **office** будет осуществляться запись текста и его редактирование. Затем это будет повторено для электронной таблицы. BLTK в комбинации с **PowerTOP** или другой анализирующей программой позволяет проверять оптимизированные настройки в процессе активной работы, а не только в режиме ожидания. Можно изменять параметры для сравнения полученных результатов тестирования.

Установите BLTK:

```
yum install bltk
```

Команда запуска BLTK:

```
bltk режим параметры
```

Команда эмуляции простоя на протяжении 120 секунд будет выглядеть так:

```
bltk -I -T 120
```

Доступные режимы:

**-I, --idle**

Система бездействует. Используется в качестве точки отсчета.

**-R, --reader**

Эмуляция чтения документов (по умолчанию с помощью **Firefox**).

**-P, --player**

Эмуляция проигрывания мультимедийных файлов с CD или DVD (по умолчанию с помощью **mplayer**).

**-O, --office**

Эмуляция редактирования документов с помощью **OpenOffice.org**.

Другие параметры:

**-a, --ac-ignore**

Игнорирует проверку состояния адаптера AC (требуется для настольных компьютеров).

**-T секунды, --time секунды**



Время (в секундах), по истечении которого следует выполнить проверку. Используется с опцией **idle**.

**-F файл, --file файл**

Определяет файл для использования при эмуляции, например файл для проигрывания при эмуляции **player**.

**-W приложение, --prog приложение**

Задаёт приложение для использования при эмуляции, например **Firefox** для эмуляции **reader**.

BLTK поддерживает множество других опций. Подробную информацию можно найти на справочной странице **bltk**.

BLTK по умолчанию сохраняет полученные результаты в каталог `~/.bltk/режим.результаты.число/`, который можно изменить в файле `/etc/bltk.conf`. Представим, например, что результаты третьей проверки в режиме **reader** хранятся в разных файлах в каталоге `~/.bltk/reader.результаты.002/`. Чтобы объединить эти результаты для облегчения их анализа, выполните:

```
bltk_report путь
```

Полученный отчет будет сохранен в файл с именем **Report** в заданном каталоге. Для его просмотра используется параметр **-o**:

```
bltk_report -o путь
```

## 2.5. TUNED И KTUNE

Служба **tuned** отслеживает использование системных компонентов и динамически изменяет настройки системы, исходя из полученной информации о занятости компонентов в разное время. Так, например, нагрузка на жесткий диск увеличивается во время запуска компьютера и авторизации, но уменьшается в процессе использования приложений пользователя, таких как OpenOffice или почтовых программ. Аналогично может изменяться нагрузка процессора и сетевых устройств.

В качестве примера рассмотрим типичный офисный компьютер. Большую часть времени интерфейс Ethernet будет бездействовать, а использоваться лишь изредка для обмена почтой и загрузки веб-страниц. При такой нагрузке нет необходимости в поддержке максимальной скорости интерфейса. Существует дополнительный модуль для **tuned**, который автоматически снижает скорость интерфейса при низкой активности, что позволяет экономить энергию. При увеличении нагрузки на протяжении некоторого времени, например при загрузке образа DVD или открытии письма с большим вложением, **tuned** увеличит скорость интерфейса до максимума для достижения максимальной производительности. Дополнительные модули для процессора и жестких дисков работают по такому же принципу.

Подобное поведение сетевых устройств, процессоров и жестких дисков по умолчанию не настроено в силу того, что изменение скорости может занимать несколько секунд и быть вполне заметным. Если скорость вращения жесткого диска замедлена, ее наращивание может занять некоторое время, в течение которого система будет отвечать на запросы с задержкой. Задержка минимальна для процессора и вполне измерима, хоть и незаметна для пользователя.

В свою очередь, в Red Hat Enterprise Linux 5.3 была впервые представлена утилита **ktune** как средство оптимизации производительности в отдельных случаях. С тех пор **ktune** подверглась значительным улучшениям и обычно используется в predefined профилях (см. [Раздел 2.5.2, «Tuned-adm»](#)).

Установите пакет **tuned** и соответствующие сценарии **systemtap**:

```
yum install tuned
```

После установки **tuned** образец файла конфигурации можно будет найти в **/etc/tuned.conf**

Запустите **tuned**:

```
service tuned start
```

Чтобы запускать **tuned** каждый раз при загрузке системы, выполните

```
chkconfig tuned on
```

При запуске **tuned** можно указать следующие параметры:

**-d, --daemon**

Запускает **tuned** в фоновом режиме.

**-c, --conffile**

Использует заданный файл конфигурации (например, **--conffile=/etc/tuned2.conf**). По умолчанию используется файл **/etc/tuned.conf**.

**-D, --debug**

Использует максимальный уровень журналирования.

### 2.5.1. Файл **tuned.conf**

По умолчанию настройки **tuned** хранятся в файле **/etc/tuned.conf**, но с помощью параметра **--conffile** можно указать другой файл.

Файл конфигурации должен содержать секцию **[main]**, где определяются основные параметры **tuned**. Настройки каждого дополнительного модуля определяются в отдельной секции.

Секция **[main]** содержит параметры:

**interval**

Интервал отслеживания системы (в секундах). По умолчанию равен 10 секундам.

**verbose**

Подробный вывод. По умолчанию отключен (**False**).

**logging**

Минимальный приоритет регистрируемых в журнале сообщений. Допустимые значения в убывающем порядке: **critical**, **error**, **warning**, **info**, **debug**. По умолчанию используется **info**.

### **logging\_disable**

Максимальный приоритет регистрируемых в журнале сообщений. Сообщения с приоритетом меньше заданного не будут регистрироваться. Допустимые значения в убывающем порядке: **critical**, **error**, **warning**, **info**, **debug**. Значение **notset** отключает эту опцию.

Каждому отдельному модулю соответствует своя секция. Название секции состоит из имени модуля, заключенного в квадратные скобки, например **[CPUTuning]**. Параметры настройки разных модулей могут отличаться.

### **enabled**

Активация модуля. По умолчанию имеет значение **True**.

### **verbose**

Подробный вывод. Если не определен, будет унаследовано значение из **[main]**.

### **logging**

Минимальный приоритет регистрируемых в журнале сообщений. Если не определен, будет унаследовано значение из **[main]**.

Пример файла конфигурации:

```
[main]
interval=10
pidfile=/var/run/tuned.pid
logging=info
logging_disable=notset

# Disk monitoring section

[DiskMonitor]
enabled=True
logging=debug

# Disk tuning section

[DiskTuning]
enabled=True
hdparm=False
alpm=False
logging=debug

# Net monitoring section

[NetMonitor]
enabled=True
logging=debug
```

```
# Net tuning section

[NetTuning]
enabled=True
logging=debug

# CPU monitoring section

[CPUMonitor]
# Enabled or disable the plugin. Default is True. Any other value
# disables it.
enabled=True

# CPU tuning section

[CPUTuning]
# Enabled or disable the plugin. Default is True. Any other value
# disables it.
enabled=True
```

## 2.5.2. Tuned-adm

Довольно часто детальный анализ системы занимает довольно много времени и преимущество экономии нескольких ватт теряется на фоне затрат. Red Hat Enterprise Linux 6 включает предопределенные профили типичной нагрузки системы и утилиту **tuned-adm** для создания, изменения, удаления профилей и переключения между ними.

Команда просмотра доступных профилей и активного профиля:

```
tuned-adm list
```

Команда просмотра активного профиля:

```
tuned-adm active
```

Команда переключения профиля:

```
tuned-adm profile профиль
```

Например:

```
tuned-adm profile server-powersave
```

Команда отключения:

```
tuned-adm off
```

При установке **tuned** по умолчанию будет использоваться профиль **default**. Red Hat Enterprise Linux 6 предоставляет следующие профили:

### **default**

Стандартный профиль. Практически не оказывает влияния на уровень энергосбережения доступных профилей и включает модули **tuned** для процессора и дисков.

### **desktop-powersave**

Энергосберегающий профиль. Ориентирован на настольные системы. Включает возможности ALPM для адаптеров SATA (см. [Раздел 3.6, «ALPM»](#)) и дополнительные модули **tuned** для процессора, дисков и интерфейсов Ethernet.

### **server-powersave**

Энергосберегающий профиль. Ориентирован на серверы. Включает возможности энергосбережения ALPM для адаптеров SATA, с помощью **HAL** отключает опрос CD-ROM (см. справочную страницу `hal-disable-polling`) и включает дополнительные модули **tuned** для процессора и дисков.

### **laptop-ac-powersave**

Энергосберегающий профиль среднего уровня. Ориентирован на ноутбуки, работающие от сети. Включает возможности энергосбережения ALPM для адаптеров SATA, WiFi и дополнительные модули **tuned** для процессора, дисков и интерфейсов Ethernet.

### **laptop-battery-powersave**

Профиль глубокого энергосбережения. Ориентирован на ноутбуки, работающие от батареи. Включает все перечисленные выше механизмы энергосбережения и дополнительно активирует планировщик для пробуждения систем, отвечает за активность регулятора «ondemand» и энергосберегающих функций AC97. Этот профиль позволяет максимально снизить энергозатраты любых систем без ограничений, но за счет снижения производительности (задержки при обработке запросов ввода/вывода к дискам и сетевым устройствам).

### **throughput-performance**

Профиль для корректирования производительности прохождения трафика. Отключает механизмы **tuned** и **ktune**, а для улучшения производительности дискового и сетевого ввода/вывода использует **sysctl** и планировщик с алгоритмом **deadline**.

### **latency-performance**

Профиль для корректирования задержки ответа сервера. Отключает механизмы **tuned** и **ktune**, а для улучшения производительности сетевого ввода/вывода использует **sysctl**.

Все профили хранятся в отдельных подкаталогах `/etc/tune-profiles`. Так, например, `/etc/tune-profiles/desktop-powersave` содержит необходимые файлы и настройки профиля `desktop-powersave`.

### **tuned.conf**

Файл конфигурации службы `tuned` для заданного профиля.

### **sysctl.ktune**

Файл настроек **sysctl** для **ktune**. Формат аналогичен `/etc/sysconfig/sysctl` (см. справочные страницы `sysctl` и `sysctl.conf`).

### **ktune.sysconfig**

Файл конфигурации `/etc/sysconfig/ktune`.

### **ktune.sh**

Используемый службой **ktune** сценарий оболочки формата **init** для исполнения команд настройки системы в процессе ее загрузки.

Самый простой способ создания нового профиля состоит в копировании существующего. Для этого может подойти профиль **laptop-battery-powersave**, изначально включающий обширный набор параметров. Для создания нового профиля на его основе просто скопируйте его каталог:

```
cp -a /etc/tune-profiles/laptop-battery-powersave/ /etc/tune-profiles/myprofile
```

Измените файлы в соответствии с вашими требованиями. Так, например, если требуется отключить опрос CD-ROM, отметьте соответствующие строки как комментарий:

```
# Disable HAL polling of CDROMS
# for i in /dev/scd*; do hal-disable-polling --device $i; done > /dev/null
2>&1
```

## 2.6. DEVICEKIT-POWER И DEVKIT-POWER

В Red Enterprise Linux 6 функции управления энергопотреблением на уровне **HAL** и некоторые функции, за которые раньше отвечал **GNOME Power Manager** (см. [Раздел 2.7, «GNOME Power Manager»](#)), берет на себя **DeviceKit-power**. В состав **DeviceKit-power** входит API, набор утилит командной строки и, собственно, сама служба. Независимо от того, являются ли источники питания системы отдельными физическими устройствами, они будут представлены в **DeviceKit-power** как устройства. Так, например, и батарея ноутбука, и источник бесперебойного питания будут представлены как устройства.

Доступ к утилитам командной строки обеспечивается за счет **devkit-power**. Параметры **devkit-power** включают:

### **--enumerate, -e**

показывает путь к объектам всех устройств питания в системе. Например:

```
/org/freedesktop/DeviceKit/power/devices/line_power_AC
/org/freedesktop/UPower/DeviceKit/power/battery_BAT0
```

### **--dump, -d**

показывает параметры для всех устройств питания.

### **--wakeups, -w**

показывает информацию о пробуждении процессора.

### **--monitor, -m**

позволяет отслеживать изменения состояния устройств питания, например подключение и отключение источника бесперебойного питания или разряд батареи. Чтобы остановить наблюдение, нажмите **Ctrl+C**.

### **--monitor-detail**

позволяет отслеживать изменения состояния устройств питания, например подключение и

отключение источника бесперебойного питания или разряд батареи. Этот параметр предоставляет более подробную информацию по сравнению с `--monitor`. Чтобы остановить наблюдение, нажмите **Ctrl+C**.

### `--show-info` путь, `-i` путь

показывает все сведения об объекте устройства. Например, чтобы получить информацию о батарее компьютера, которой соответствует объект `/org/freedesktop/UPower/DeviceKit/power/battery_BAT0`, выполните:

```
devkit-power -i /org/freedesktop/UPower/DeviceKit/power/battery_BAT0
```

## 2.7. GNOME POWER MANAGER

**GNOME Power Manager** входит в стандартный комплект GNOME. Его основные функции управления энергопотреблением, которые предоставлялись ранними версиями Red Hat Enterprise Linux, были интегрированы в отдельный пакет **DeviceKit-power** в Red Hat Enterprise Linux 6 (см. [Раздел 2.6, «DeviceKit-power и devkit-power»](#)). На сегодняшний день **GNOME Power Manager** предоставляет интерфейс для доступа к этим возможностям и добавляет значок на панель задач, который сообщает пользователю об изменении уровня потребления энергии (например, при переходе от работы ноутбука от сети в автономный режим), заряда батареи и пр.

**GNOME Power Manager** позволяет выборочно изменить настройки энергопотребления. Для этого щелкните на его значке на панели задач и в открывшемся меню выберите **Параметры**.

Окно **Параметры управления питанием** содержит три вкладки:

- **От сети**
- **От батареи**
- **Общие**

На вкладках **От сети** и **От батареи** можно настроить время ожидания до отключения экрана неактивной системы, время ожидания до перевода неактивной системы в спящий режим, настроить снижение скорости вращения дисков во время бездействия. На вкладке **От батареи** дополнительно можно изменить яркость экрана и поведение системы в случае критического разряда батареи (например, переход компьютера в спящий режим). На вкладке **Общие** можно изменить поведение кнопки питания на корпусе и кнопки сна, а также настроить поведение апплета **GNOME Power Manager** на панели задач.

## 2.8. ПРОЧИЕ СРЕДСТВА СБОРА СТАТИСТИКИ

Red Hat Enterprise Linux 6 предоставляет другие инструменты сбора информации об активности системы, большинство которых используется в качестве дополнительных средств анализа или подтверждения уже известного результата.

### **vmstat**

**vmstat** позволяет получить подробную информацию об активности процессов, памяти, процессора, ввода/вывода устройств и пр.

### **iostat**

**iostat** выполняет те же функции что и **vmstat**, но только для оценки активности ввода/вывода блочных устройств. Предлагает подробный вывод и статистику.

### **blktrace**

**blktrace** позволяет получить исключительно подробные данные для отслеживания ввода/вывода блочных устройств. Часто используется вместе с **diskdevstat**.



## ГЛАВА 3. ОСНОВНАЯ ИНФРАСТРУКТУРА

### 3.1. СОСТОЯНИЯ ЭНЕРГОСБЕРЕЖЕНИЯ ПРОЦЕССОРА

Процессоры x86 могут переходить в состояния, где некоторые компоненты отключены или работают в режиме энергосбережения. Они называются *C-состояниями* и позволяют снизить потребление энергии за счет частичного отключения незанятых процессоров. Нумерация C-состояний начинается с нуля (состояние C0 характеризуется максимальным потреблением энергии и производительностью). Несмотря на схожесть C-состояний разных процессоров, они не являются абсолютно идентичными. Рассмотрим подробнее состояния C0-C3:

#### C0

Рабочий режим. Процессор работает без перехода в режим бездействия.

#### C1, остановлен

В этом состоянии процессор не обрабатывает задания, но и не переходит в режим энергосбережения. Поступающие задания обрабатываются без задержки. Если процессор поддерживает переход между состояниями, он должен поддерживать это состояние. Pentium 4 поддерживает расширенное состояние C1E, в котором дополнительно снижается потребление энергии.

#### C2, часы остановлены

Часы процессора остановлены. При этом поддерживается состояние регистров и кэша, поэтому после повторного запуска часов процессор сможет немедленно приступить к обработке заданий. Это состояние не является обязательным.

#### C3, режим сна

Процессор в спящем режиме не обновляет кэш. По сравнению с C2 пробуждение занимает значительно дольше. Это состояние не является обязательным.

Микропроцессорная архитектура Nehalem поддерживает новое состояние C6, позволяющее снизить напряжение до нуля. При этом потребление энергии снизится на 80-90%. Ядро Red Hat Enterprise Linux 6 обеспечивает оптимизацию этого C-состояния.

### 3.2. РЕГУЛЯТОРЫ CPUFREQ

Один из наиболее эффективных способов снижения потребления энергии и тепловыделения компонентов компьютера состоит в использовании подсистемы CPUfreq, позволяющей изменять скорость процессора на лету. Снижение скорости часов процессора помогает экономить энергию. Правила изменения частоты в большую или меньшую сторону определяются регуляторами.

Регулятор задает характеристики энергозатрат процессора, что оказывает влияние на его производительность. Каждый регулятор отвечает за определенное поведение, которое можно выбрать в зависимости от нагрузки. В этой секции рассказано о выборе и настройке регуляторов, их характеристиках и нагрузках, при которых они применяются.

#### 3.2.1. Типы регуляторов CPUfreq

В этой секции рассматриваются типы регуляторов CPUfreq в Red Hat Enterprise Linux 6.

### **cpufreq\_performance**

Регулятор «performance» заставляет процессор использовать максимальную частоту часов. Значение частоты не изменяется динамически, поэтому этот регулятор не решает задачи энергосбережения. Он подходит только для длительной непрерывной работы процессора.

### **cpufreq\_powersave**

Регулятор «powersave» заставляет процессор использовать минимальную частоту часов. Значение частоты не изменяется динамически, поэтому этот регулятор хоть и обеспечивает максимальную экономию энергии, но за счет *значительного снижения производительности процессора*.

Обычно медленный процессор при полной загрузке потребляет больше энергии по сравнению с быстрым, но не полностью загруженным процессором. Поэтому несмотря на рекомендации использования регулятора «powersave» в периоды ожидаемого бездействия, непредвиденная нагрузка может привести к высокому потреблению энергии вместо ожидаемой экономии.

Регулятор «powersave», по большому счету, ограничивает скорость процессора, а не экономит энергию. Его рекомендуется применять в окружениях с перегревающимися компонентами.

### **cpufreq\_ondemand**

Динамический регулятор «ondemand» позволяет достичь максимальной частоты процессора при высокой нагрузке и минимальной частоты при бездействии системы. Изменение частоты в зависимости от нагрузки происходит с задержкой, что при частом переключении между режимами может сказаться на общей экономии энергии.

Для большинства компьютеров регулятор «ondemand» позволяет достичь компромисса между тепловыделением, потреблением энергии и производительностью. Если нагрузка системы повышается в определенное время суток, «ondemand» будет автоматически переключаться между максимальной и минимальной частотой в зависимости от нагрузки.

### **cpufreq\_userspace**

Регулятор «userspace» позволяет пользовательским программам и процессам, выполняющимся в режиме root, задавать значение частоты. Обычно он используется в комбинации со службой **cpuspeed**, является наиболее гибким из всех перечисленных регуляторов в плане настройки и позволяет достичь оптимального баланса между производительностью системы и потреблением энергии.

### **cpufreq\_conservative**

Подобно «ondemand», регулятор «conservative» корректирует частоту процессора в зависимости от нагрузки, но переключается не между максимальной и минимальной частотой, а изменяет частоту постепенно.

Таким образом, «conservative» будет выбирать наиболее подходящую для уровня нагрузки частоту. Подобное изменение частоты происходит с еще большей задержкой по сравнению с использованием регулятора «ondemand».



## ПРИМЕЧАНИЕ

Регулятор можно включить с помощью заданий **cron** в определенное время. Так, например, можно настроить использование низкочастотного регулятора в периоды низкой активности (ночью), а в дневное время заменить его на высокочастотный регулятор.

Раздел 3.2.2, «Настройка CPUfreq» (см. Процедура 3.2, «Активация регулятора CPUfreq») содержит инструкции по активации регуляторов.

### 3.2.2. Настройка CPUfreq

Прежде чем выбрать и настроить регулятор CPUfreq, надо установить соответствующий драйвер.

#### Процедура 3.1. Добавление драйвера CPUfreq

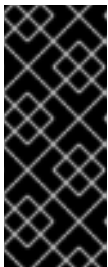
1. Проверьте наличие драйверов CPUfreq в системе:

```
ls
/lib/modules/[версия_ядра]/kernel/arch/[архитектура]/kernel/cpu/cpufreq/
```

2. Добавьте драйвер CPUfreq:

```
modprobe [драйвер_CPUfreq]
```

Не указывайте расширение **.ko**.



## ВАЖНО

При выборе драйвера всегда отдавайте предпочтение **acpi-cpufreq**, а не **p4-clockmod**. В то время как использование **p4-clockmod** снижает частоту часов процессора, оно не снижает потребление энергии. В свою очередь, **acpi-cpufreq** уменьшает потребление энергии и тепловыделение при уменьшении частоты процессора.

3. Завершив настройку драйвера CPUfreq, можно проверить, какой регулятор используется системой в настоящий момент:

```
cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
```

Можно также посмотреть доступные регуляторы для процессора:

```
cat
/sys/devices/system/cpu/[ID_процессора]/cpufreq/scaling_available_governors
```

Некоторые регуляторы могут быть недоступны. В таком случае с помощью команды **modprobe** добавьте модули ядра, необходимые для активации определенного регулятора. Модули ядра можно найти в **/lib/modules/[версия\_ядра]/kernel/drivers/cpufreq/**.

### Процедура 3.2. Активация регулятора CPUfreq

1. Для включения регулятора в список воспользуйтесь командой **modprobe**. Например, если вас интересует регулятор **ondemand**, выполните:

```
modprobe cpufreq_ondemand
```

2. Теперь он будет доступен. Команда активации выглядит так:

```
echo [регулятор] >
/sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
```

### 3.2.3. Настройка частоты процессора с помощью CPUfreq

После выбора регулятора CPUfreq можно дополнительно изменить скорость каждого процессора с помощью нижеперечисленных параметров в `/sys/devices/system/cpu/[ID_процессора]/cpufreq/`:

- **cpuinfo\_min\_freq** возвращает минимально допустимую частоту процессора (в кГц).
- **cpuinfo\_max\_freq** возвращает максимально допустимую частоту процессора (в кГц).
- **scaling\_driver** возвращает драйвер CPUfreq, используемый для настройки частоты заданного процессора.
- **scaling\_available\_governors** возвращает доступные для заданного ядра регуляторы. [Раздел 3.2.2, «Настройка CPUfreq»](#) (см. [Процедура 3.2, «Активация регулятора CPUfreq»](#)) содержит инструкции по добавлению регулятора, если он недоступен.
- **scaling\_governor** возвращает текущий используемый регулятор CPUfreq. Чтобы его изменить, выполните `echo [регулятор] > /sys/devices/system/cpu/[ID_процессора]/cpufreq/scaling_governor`. [Раздел 3.2.2, «Настройка CPUfreq»](#) (см. [Процедура 3.2, «Активация регулятора CPUfreq»](#)) содержит подробную информацию.
- **cpuinfo\_cur\_freq** возвращает текущую частоту процессора (в кГц).
- **scaling\_available\_frequencies** возвращает доступные значения частоты процессора (в кГц).
- **scaling\_min\_freq** и **scaling\_max\_freq** задают *ограничения политики* процессора (в кГц).



#### ВАЖНО

При настройке диапазона сначала необходимо определить **scaling\_max\_freq**, а уже затем **scaling\_min\_freq**.

- **affected\_cpus** показывает список процессоров, нуждающихся в координации частоты.
- **scaling\_setspeed** изменяет частоту процессора (в кГц). Выбранное значение должно лежать в диапазоне между **scaling\_min\_freq** и **scaling\_max\_freq**.

Просмотреть текущие значения параметров можно с помощью команды `cat [параметр]`. Например, команда просмотра текущей частоты процессора `cpu0` выглядит так:

```
cat /sys/devices/system/cpu/cpu0/cpufreq/cpuinfo_cur_freq.
```

Изменить значения параметров можно с помощью команды `echo [значение] > /sys/devices/system/cpu/[ID_процессора]/cpufreq/[параметр]`. Например, команда установки минимальной частоты процессора `cpu0` в 360 кГц будет выглядеть так:

```
echo 360000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling_min_freq
```

### 3.3. ПРИОСТАНОВКА И ВОЗОБНОВЛЕНИЕ РАБОТЫ

Когда система находится в режиме ожидания, драйверы сохраняют свое состояние, которое ядро сможет загрузить при возобновлении работы. Драйверы затем повторно настроят соответствующие им устройства.

В данной ситуации видеодрайверы представляют особую проблему, так как спецификация ACPI не требует повторной настройки видеоборудования. То есть если видеодрайверы не могут настроить оборудование с нуля, это может помешать возобновить работу системы.

Red Hat Enterprise Linux 6 обеспечивает поддержку многих последних моделей видеокарт, что позволяет использовать возможности приостановки и возобновления работы для широкого диапазона платформ. Так, например, значительно улучшена поддержка схем NVIDIA серии GeForce 8800.

### 3.4. БЕЗТАКТОВОЕ ЯДРО

Раньше ядро Linux периодически опрашивало каждый процессор с заданной частотой (100 Гц, 250 Гц или 1000 Гц в зависимости от платформы) на предмет выполняемых процессов. Полученные результаты использовались для распределения нагрузки. Подобные прерывания осуществлялись независимо от режима энергосбережения процессора, поэтому даже бездействующий процессор отвечал на множество таких запросов каждую секунду. В системах с механизмами энергосбережения эти запросы не позволяли процессорам оставаться в режиме бездействия достаточно долго.

Ядро Red Hat Enterprise Linux 6 является *безтактовым*, то есть вместо периодических прерываний оно осуществляет прерывания при необходимости. Таким образом, бездействующим процессорам не требуется возвращаться в рабочий режим до тех пор, пока они не получат задания для выполнения.

### 3.5. ASPM

С помощью технологии ASPM (Active-State Power Management) можно эффективно управлять потреблением энергии шин PCI Express (PCIe, Peripheral Component Interconnect Express) посредством их перевода в энергосберегающий режим, если подключенные через них устройства не используются. ASPM контролирует обе точки подключения и позволяет снизить потребление энергии, даже если подключенное устройство находится в рабочем режиме.

При активации ASPM задержка ответа устройства увеличивается из-за времени, затрачиваемого на переключение режимов шины. Можно определить три способа поведения ASPM:

**default**

Настраивает состояние энергопотребления шины PCI Express в соответствии с параметрами, определенными на микропрограммном уровне (например, в BIOS). Этот режим используется по умолчанию.

#### **powersave**

Максимальное энергосбережение независимо от воздействия на производительность системы.

#### **performance**

Отключает ASPM и обеспечивает максимальную производительность PCI Express.

Поведение ASPM можно изменить в файле `/sys/module/pci_aspm/parameters/policy` или во время загрузки с помощью параметра `pci_aspm`. Так, `pci_aspm=off` отключает, а `pci_aspm=force` принудительно включает ASPM даже на устройствах, не поддерживающих ASPM.



#### **ПРЕДУПРЕЖДЕНИЕ**

Если используется параметр `pci_aspm=force`, оборудование, не поддерживающее ASPM, может привести к тому, что система перестанет отвечать на запросы. Прежде чем установить этот параметр, убедитесь, что все аппаратные компоненты PCIe поддерживают ASPM.

## **3.6. ALPM**

Технология ALPM (Aggressive Link Power Management) позволяет снизить потребление энергии дисками за счет перевода интерфейса SATA в энергосберегающий режим в периоды отсутствия запросов ввода и вывода. При поступлении запросов ALPM автоматически переведет SATA в активное состояние.

При активации ALPM увеличивается задержка ответа дисков. Поэтому рекомендуется прибегать к помощи ALPM только в тех случаях, когда ожидаются длительные периоды бездействия.

ALPM подходит только для контроллеров SATA с интерфейсом AHCI (Advanced Host Controller Interface). За подробной информацией об AHCI обратитесь к <http://www.intel.com/technology/serialata/ahci.htm>.

Если доступен, механизм ALPM используется по умолчанию и предоставляет три режима:

#### **min\_power**

При отсутствии запросов ввода/вывода к диску этот режим переводит интерфейс в глубокий режим энергосбережения (SLUMBER). Лучше всего подходит для длительных периодов бездействия.

#### **medium\_power**

При отсутствии запросов ввода/вывода к диску этот режим переводит интерфейс во второй по эффективности после SLUMBER энергосберегающий режим (PARTIAL). Подходит для

переключения состояний с минимальными издержками производительности во время периодических изменений нагрузки ввода/вывода.

**medium\_power** позволяет переключаться между состояниями PARTIAL и ACTIVE в зависимости от нагрузки. Стоит отметить, что переход между состояниями PARTIAL и SLUMBER возможен только через состояние ACTIVE.

### max\_performance

ALPM отключен. Состояние интерфейса при отсутствии запросов ввода/вывода изменяться не будет.

Чтобы узнать, поддерживают ли адаптеры SATA возможности ALPM, проверьте наличие файла `/sys/class/scsi_host/host*/link_power_management_policy`. Изменить настройки можно напрямую в этом файле.



### ВАЖНО

Использование режима **min\_power** или **medium\_power** автоматически отключит возможности горячей замены.

## 3.7. ОПТИМИЗАЦИЯ ДОСТУПА К ДИСКАМ RELATIME

Стандарт POSIX требует, чтобы операционные системы поддерживали метаданные файловой системы с информацией о времени последнего доступа к файлам. Динамическое обновление меток времени (известных как **atime**) требует постоянного выполнения дополнительных операций записи на диск, что поддерживает накопители в активном состоянии. Поскольку данные **atime** используются ограниченным кругом приложений, их поддержка не оправдывает затрат энергии. Более того, запись на диск осуществляется даже при чтении файла из кэша, а не с диска. В свое время ядро Linux поддерживало опцию **noatime** для **mount**, которая отключала запись данных **atime** в файловую систему. Но совершенно отключить эту возможность тоже нельзя, так как многие приложения полагаются на данные **atime** и не смогут работать без них.

Ядро Red Hat Enterprise Linux 6 предлагает альтернативу — **relatime**. Эта опция обновляет данные **atime**, но не при каждом обращении к файлу. Вместо этого данные **atime** будут записаны на диск только в том случае, если файл был изменен с момента последнего обновления **atime** или к нему выполнялось обращение по истечении заданного периода (по умолчанию один день).

При подключении файловых систем опция **relatime** используется по умолчанию. Отменить ее для всех файловых систем одновременно можно с помощью загрузочного параметра **default\_relatime=0**. Чтобы отменить эту опцию для конкретной файловой системы, при ее подключении можно указать параметр **norelatime**. Наконец, с помощью параметра загрузки **relatime\_interval=** можно изменить время, по истечении которого система обновит данные **atime** файлов. Значение по умолчанию — **86400**.

## 3.8. ОГРАНИЧЕНИЕ ЭНЕРГОПОТРЕБЛЕНИЯ

Red Hat Enterprise Linux 6 позволяет ограничить энергопотребление оборудования за счет технологий HP *Dynamic Power Capping* (DPC) и Intel Node Manager (NM). С их помощью можно не только ограничить потребление энергии сервером, но и снизить риск перегрузки источников

питания, тем самым оптимизируя работу центров обработки данных. Вы сможете разместить большее число серверов на той же площади при полной уверенности, что потребление энергии не превысит допустимый лимит.

### HP Dynamic Power Capping

Возможности HP Dynamic Power Capping доступны на некоторых серверах ProLiant и BladeSystem и позволяют ограничить потребление энергии не только одним сервером, но и целой группой. Независимо от нагрузки, сервер не может затратить больше энергии чем заданный предел. Если предел потребления энергии достигнут, управляющий процессор откорректирует P-состояние и частоту часов в целях экономии.

Динамическое ограничение потребления энергии корректирует поведение процессора независимо от операционной системы. Тем не менее, механизм HP iLO2 (integrated Lights-Out 2) обеспечивает доступ операционной системы к управляющему процессору, поэтому программы пространства пользователя смогут к нему обращаться. Ядро Red Hat Enterprise Linux 6 включает драйвер для HP iLO и iLO2, что позволяет опрашивать управляющие процессоры в `/dev/hpilo/dXccbM`. Ядро также включает расширение интерфейса `hwmon sysfs` для поддержки ограничения энергопотребления и драйвер `hwmon` для устройств измерения мощности ACPI 4.0, использующих интерфейс `sysfs`. Все эти функции в совокупности позволяют операционной системе и инструментам пользователя прочитать заданное предельное значение и определить текущее потребление.

Подробную информацию о механизме HP Dynamic Power Capping можно найти по адресу <http://h20000.www2.hp.com/bc/docs/support/SupportManual/c01549455/c01549455.pdf>

### Intel Node Manager

Intel Node Manager позволяет установить предел потребления энергии для систем, используя P- и T-состояния процессора. Администраторы могут настроить снижение энергозатрат во время низкой нагрузки систем (например, ночью или на выходных).

Intel Node Manager корректирует поведение процессора с помощью механизма OSPM (Operating System-directed configuration and Power Management) согласно стандарту ACPI (Advanced Configuration and Power Interface). Как только Intel Node Manager сообщает драйверу OSPM об изменении T-состояния, драйвер соответственно изменит P-состояния процессора. И наоборот, когда Intel Node Manager сообщит об изменении P-состояния, драйвер OSPM откорректирует T-состояния. Это осуществляется автоматически и не требует вмешательства операционной системы. Для конфигурации и управления Intel Node Manager используется программный комплект *Intel Data Center Manager* (DCM).

Подробную информацию об Intel Node Manager можно найти по адресу <http://communities.intel.com/docs/DOC-4766>

## 3.9. УПРАВЛЕНИЕ ЭНЕРГОПОТРЕБЛЕНИЕМ ВИДЕОУСТРОЙСТВ

Red Hat Enterprise Linux 6 позволяет снизить энергопотребление видеоустройств и дисплеев за счет исключения ненужных затрат.

### Изменение частоты обновлений с помощью LVDS

*Низковольтная дифференциальная передача сигналов* (LVDS, Low-voltage differential signalling) характеризует способ передачи электрических сигналов по медному проводу и широко применяется для передачи пиксельной информации жидкокристаллическим (ЖК) мониторам ноутбуков. Для начала следует упомянуть о такой характеристике монитора как *частота обновления* — это частота получения новых данных от графического контроллера и прорисовки экрана заново. Стандартная частота — 60 обновлений в секунду (60 Гц). Если монитор и



графический контроллер соединены посредством LVDS, энергия будет потребляться при каждом цикле обновления, а в режиме бездействия частота обновлений многих ЖК-мониторов может падать до 30 Гц без каких-либо последствий (в отличие от катодно-лучевых мониторов, которые начнут мерцать). В ядро Red Hat Enterprise Linux 6 встроен драйвер для графических адаптеров Intel, снижающий частоту обновления автоматически и позволяющий сэкономить примерно 0.5 Вт во время бездействия монитора.

### Самообновление памяти

Синхронная динамическая память с произвольным доступом (SDRAM, Synchronous dynamic random access memory) используется для организации видеопамати графических адаптеров и обновляется тысячи раз в секунду с целью поддержки данных в отдельных ячейках памяти. Помимо основной функции управления поступающими и исходящими данными, контроллер памяти отвечает за инициализацию циклов обновления. Надо отметить, что режим самообновления SDRAM является энергосберегающим. В этом режиме для генерации собственных циклов обновления память использует внутренний таймер, что позволяет отключить контроллер без риска потери данных, находящихся в данный момент в памяти. Ядро Red Hat Enterprise Linux 6 позволяет перевести графические адаптеры Intel в режим самообновления памяти в периоды бездействия и тем самым сэкономить примерно 0.8 Вт.

### Снижение частоты графического процессора

Типичные графические процессоры (GPU, Graphical Processing Unit) оборудованы внутренними часами, отвечающими за управление другими компонентами. Ядро Red Hat Enterprise Linux 6 позволяет уменьшить частоту внутренних часов некоторых графических процессоров Intel и ATI с целью экономии энергии. Ядро автоматически уменьшает частоту часов в периоды бездействия графического процессора и увеличивает при росте активности. Такой способ позволяет в целом сэкономить до 5 Вт.

### Выключение графического процессора

Графические драйверы Intel и ATI в Red Hat Enterprise Linux 6 способны определить отсутствие подключения монитора к адаптеру и полностью отключить графический процессор.

## 3.10. RFKILL

Многие современные компьютеры оборудованы устройствами WiFi, Bluetooth и 3G, которые потребляют энергию даже во время бездействия.

Подсистема ядра Linux под названием *RFKill* предоставляет интерфейс для управления подобного рода устройствами. Отключаемые устройства могут быть переведены в состояние, из которого их можно будет позднее повторно активировать (*временное блокирование*) или нельзя (*постоянное блокирование*).

RFKill включает интерфейс API, с помощью которого драйверы ядра, отвечающие за работу RFKill, могут осуществлять регистрацию в ядре. Эти драйверы позволяют включать и отключать устройства, опрашивать их состояние и уведомлять пользовательские программы.

Интерфейс RFKill определен в файле `/dev/rfkill`, который содержит текущее состояние всех встроенных устройств передачи. Состояние RFKill каждого устройства зарегистрировано в `sysfs`. При изменении состояния устройства RFKill генерирует событие.

Для получения доступа к программе **Rfkill** потребуется установить одноименный пакет.

Выполните команду `rfkill list` для получения списка устройств и их *индексов* (начиная с нуля). Индекс можно использовать для блокирования устройств. Например:

```
rfkill block 0
```

Эта команда заблокирует первое устройство.

Можно заблокировать отдельные категории устройств. Например:

```
rfkill block wifi
```

Эта команда заблокирует все устройства WiFi. Чтобы полностью заблокировать все устройства в системе, выполните

```
rfkill block all
```

Команда **rfkill unblock** разблокирует устройства. Полный список параметров можно посмотреть с помощью команды **rfkill help**.

### 3.11. ОПТИМИЗАЦИЯ В ПРОСТРАНСТВЕ ПОЛЬЗОВАТЕЛЯ

Уменьшение объема работ, выполняемых оборудованием, позволяет пропорционально снизить потребляемую энергию. В то время как системные компоненты могут работать в разных режимах энергосбережения (см. [Глава 3, Основная инфраструктура](#)), пользовательские программы, запрашивающие ненужную обработку заданий, могут помешать переходу оборудования в энергосберегающий режим. С целью снижения подобных энергозатрат при разработке Red Hat Enterprise Linux 6 особое внимание уделялось следующему:

#### Уменьшение числа попыток пробуждения оборудования

Red Hat Enterprise Linux 6 использует *безтактовое ядро* (см. [Раздел 3.4, «Безтактовое ядро»](#)), которое позволяет дольше поддерживать процессоры в энергосберегающем режиме. Но тактовые сигналы не являются единственной причиной пробуждения процессора — программные вызовы функций также приводят к выходу процессора из энергосберегающего режима. Red Hat удалось уменьшить число ненужных вызовов более чем в 50 программах.

#### Уменьшение числа операций ввода и вывода

Операции ввода и вывода через сетевые интерфейсы и накопители приводят к потреблению энергии, которую можно сэкономить при переводе этих устройств в энергосберегающий режим при отсутствии нагрузки. Red Hat удалось значительно уменьшить число лишних запросов обращения к накопителям в программах, особенно тех, которые предотвращали замедление скорости вращения дисков при снижении активности.

#### Оптимизация сценариев инициализации

Запускающиеся автоматически службы (обязательные или дополнительные), могут нерационально потреблять системные ресурсы. Рекомендуется по умолчанию отключать ненужные службы и запускать их по мере необходимости. Например, служба активации поддержки Bluetooth **BlueZ** раньше запускалась автоматически во время загрузки системы даже при отсутствии устройств Bluetooth. Теперь прежде чем запустить службу, сценарий инициализации **BlueZ** проверяет наличие устройства Bluetooth.

## ГЛАВА 4. ПРИМЕРЫ

В этой главе рассмотрены примеры анализа производительности типичного сервера и ноутбука.

### 4.1. СЕРВЕР

Red Hat Enterprise Linux 6 поддерживает оборудование большинства современных серверов. Решение оптимизации с целью снижения энергозатрат принимается, исходя из нагрузки на сервер.

Обычно нет необходимости в оптимизации работы видеоадаптеров серверов, поэтому настройки графического процессора можно не изменять.

#### Веб-сервер

Веб-сервер обычно обрабатывает большой объем сетевых и локальных запросов ввода/вывода. В зависимости от скорости внешнего соединения, 100 Мбит/с должно быть достаточно. Если сервер предоставляет статическое содержимое, производительность процессора не настолько критична. Таким образом, оптимизация энергопотребления достигается за счет следующего:

- нет необходимости в дополнительных дисковых или сетевых модулях для **tuned**;
- использование ALPM;
- включение регулятора **ondemand**;
- скорость сетевой карты ограничена 100 Мбит/с.

#### Сервер вычислений

Такой сервер активно использует ресурсы процессора. Оптимизация энергопотребления достигается за счет следующего:

- В зависимости от вида решаемых задач и расположения данных могут потребоваться дисковые или сетевые модули для **tuned**. Для выполнения пакетной обработки утилиты **tuned** должна быть активна.
- Может потребоваться регулятор **performance**.

#### Почтовый сервер

Почтовый сервер обычно обрабатывает запросы ввода/вывода и использует процессорные ресурсы. Оптимизация энергопотребления достигается за счет следующего:

- регулятор **ondemand** включен, так как последние несколько процентов производительности процессора не так важны;
- нет необходимости в дополнительных дисковых или сетевых модулях для **tuned**;
- не должно быть ограничений на скорость сети, так как для обмена почтой часто используется внутренняя передача, а при внутреннем обмене можно использовать соединение 1 Гбит/с или 10 Гбит/с.

#### Файловый сервер

Требования к файловому серверу аналогичны почтовому, но может потребоваться больше процессорных ресурсов в зависимости от используемого протокола. Так, типичные серверы Samba более требовательны к ресурсам по сравнению с NFS, а NFS — по сравнению с iSCSI.

При этом вы сможете использовать регулятор **ondemand**.

### Сервер каталогов

Требования сервера каталогов к дисковому вводу/выводу невысоки, но необходимо обратить внимание на сетевую задержку. Тщательно тестируйте все изменения скорости соединения.

## 4.2. НОУТБУК

Управление энергопотреблением ноутбуков может оказать существенное влияние на время автономной работы ноутбуков. Несмотря на то, что ноутбуки изначально потребляют меньше энергии по сравнению с настольными компьютерами и серверами, при автономной работе любая экономия поможет продлить ценное время работы батареи. В этой секции рассматриваются методы энергосбережения, которые также могут применяться при работе от сети.

Снижение энергозатрат отдельных компонентов особенно важно для ноутбуков. Например, для сетевого интерфейса 1 Гбит/с, функционирующего на скорости 100 Мбит/с, экономия в среднем составляет 3-4 ватт. Для типичного сервера, потребляющего около 400 ватт, потребление энергии снизится примерно на 1 %, а для ноутбука, потребляющего 40 ватт, экономия лишь одного компонента составит целых 10 %.

Оптимизация энергопотребления ноутбуков достигается за счет следующего:

- Отключение в BIOS неиспользуемого оборудования (параллельных и последовательных портов, веб-камер, WiFi, Bluetooth и т.п.).
- Уменьшение яркости монитора при ослабленном освещении. Для этого в главном меню надо выбрать **Система+Параметры** → **Управление питанием (GNOME)** или в меню запуска выбрать **Компьютер+Параметры системы+Расширенные** → **Управление питанием (KDE)**. Другие способы состоят в выполнении команд **gnome-power-manager**, **xbacklight** или изменения настроек с помощью функциональных клавиш.
- Профиль **laptop-battery-powersave** утилиты **tuned-adm** предоставляет доступ к целому набору механизмов энергосбережения. Стоит помнить, что их использование может негативно сказаться на производительности жесткого диска и сетевого интерфейса.

Дополнительные методы изменения системных настроек включают:

- использование регулятора **ondemand** (активен по умолчанию в Red Hat Enterprise Linux 6);
- активация режима ноутбука (в рамках профиля **laptop-battery-powersave**):

```
echo 5 > /proc/sys/vm/laptop_mode
```

- увеличение интервала между пробуждениями для периодической записи данных на диск (в рамках профиля **laptop-battery-powersave**):

```
echo 1500 > /proc/sys/vm/dirty_writeback_centisecs
```

- отключение контроля немаскируемых прерываний (в рамках профиля **laptop-battery-powersave**):

```
echo 0 > /proc/sys/kernel/nmi_watchdog
```

- активация энергосбережения для аудиокарт AC97 (активно по умолчанию в Red Hat Enterprise Linux 6):

```
echo Y > /sys/module/snd_ac97_codec/parameters/power_save
```

- активация энергосбережения в многоядерных архитектурах (в рамках профиля **laptop-battery-powersave**):

```
echo 1 > /sys/devices/system/cpu/sched_mc_power_savings
```

- автоматический переход USB в режим ожидания:

```
for i in /sys/bus/usb/devices/*/power/autosuspend; do echo 1 > $i; done
```

Обратите внимание, что автоматический переход в режим ожидания работает не со всеми устройствами USB.

- активация минимального уровня энергопотребления для ALPM (в рамках профиля **laptop-battery-powersave**):

```
echo min_power > /sys/class/scsi_host/host*/link_power_management_policy
```

- подключение файловой системы с использованием `relatime` (активно по умолчанию в Red Hat Enterprise Linux 6):

```
mount -o remount,relatime точка_подключения
```

- активация максимально эффективного режима энергосбережения для жестких дисков (в рамках профиля **laptop-battery-powersave**):

```
hdparm -B 1 -S 200 /dev/sd*
```

- отключение опроса CD-ROM (в рамках профиля **laptop-battery-powersave**):

```
hal-disable-polling --device /dev/scd*
```

- уменьшение яркости экрана до **50** и меньше, например:

```
xbacklight -set 50
```

- активация DPMS в периоды бездействия монитора:

```
xset +dpms; xset dpms 0 0 300
```

- уменьшение уровня потребления энергии устройств WiFi (в рамках профиля **laptop-battery-powersave**):

```
for i in /sys/bus/pci/devices/*/power_level ; do echo 5 > $i ; done
```

- отключение WiFi:

```
echo 1 > /sys/bus/pci/devices/*/rf_kill
```

- ограничение максимальной скорости проводного соединения 100 мегабитами в секунду (в рамках профиля **laptop-battery-powersave**):

```
ethtool -s eth0 advertise 0x0F
```

## ПРИЛОЖЕНИЕ А. РЕКОМЕНДАЦИИ ДЛЯ РАЗРАБОТЧИКОВ

В любой серьезном пособии по программированию рассматриваются вопросы производительности функций и выделения памяти. При разработке программ следует обращать внимание на их влияние на энергопотребление систем. Несомненно, это не затрагивает каждую строку кода, но поможет уменьшить вероятность неэффективного потребления.

Что может оказывать негативный эффект на энергопотребление системы?

- Использование потоков.
- Излишнее или неэффективное пробуждение процессора. Если пробуждение необходимо, выполните необходимые действия быстро и в одно и то же время.
- Излишнее использование `[f]sync()`.
- Излишние периодические опросы (лучше применять тактику реагирования на события).
- Низкая эффективность пробуждений.
- Неэффективный доступ к дискам. Рекомендуется использование буферов большего объема для уменьшения частоты запросов доступа к диску и выполнение записи больших блоков за раз.
- Неэффективное использование таймеров. По возможности рекомендуется группировать таймеры для программ (или даже систем);
- Излишние операции ввода и вывода, потребления энергии и использования ресурсов памяти (включая утечку памяти).
- Выполнение вычислений, в которых нет необходимости.

Ниже об этом будет рассказано более подробно.

### А.1. ИСПОЛЬЗОВАНИЕ ПОТОКОВ

Использование потоков не всегда ускоряет работу приложений.

#### Python

В Python используются возможности глобальной блокировки интерпретатора (GIL, Global Interpreter Lock) <sup>[1]</sup>, поэтому выполнение большого числа потоков имеет смысл только для больших объемов операций ввода/вывода.

#### Perl

Потоки Perl изначально предназначались для выполнения программ без ветвления (например, в 32-битных системах Windows). При этом данные копируются в каждый отдельный поток, а совместный доступ к ним по умолчанию не обеспечивается, так как пользователи должны иметь возможность контролировать предоставление общего доступа. Для этого используется модуль `threads::shared`, который дополнительно создает переменные для данных, что и замедляет обработку. <sup>[2]</sup>

#### C

Потоки C используют память совместно, при этом каждому потоку соответствует отдельный

стек, а необходимости в создании ядром новых дескрипторов файлов и выделении дополнительной памяти нет. При этом в параллельных вычислениях может использоваться большее число потоков и принимать участие большее число процессоров. Для оптимизации производительности потоков рекомендуется использовать язык низкого уровня программирования — С или С++. Если же вы предпочитаете работать со сценариями, рекомендуется создавать С-проекции. Специальные инструменты анализа помогут идентифицировать низкопроизводительные секции кода [3].

## A.2. ПРОБУЖДЕНИЯ

Многие приложения проверяют наличие изменений в файлах конфигурации с заранее определенной частотой (например, каждую минуту). Это приводит к пробуждению диска в случае его бездействия, поэтому рекомендуется откорректировать частоту проверки или использовать для этих целей подсистему **inotify**.

Например:

```
int fd;
fd = inotify_init();
int wd;
/* checking modification of a file - writing into */
wd = inotify_add_watch(fd, "./myConfig", IN_MODIFY);
if (wd < 0) {
    inotify_cant_be_used();
    switching_back_to_previous_checking();
}
...
fd_set rdfs;
struct timeval tv;
int retval;
FD_ZERO(&rdfs);
FD_SET(0, &rdfs);

tv.tv_sec = 5;
value = select(1, &rdfs, NULL, NULL, &tv);
if (value == -1)
    perror(select);
else {
    do_some_stuff();
}
...
```

Преимущество такого подхода состоит в возможности выполнения широкого диапазона проверок.

Недостатком является то, что число выполняемых проверок ограничено. Это число задано в файле `/proc/sys/fs/inotify/max_user_watches` и изменять его не рекомендуется. Следует также учесть, что в случае сбоя **inotify** необходимо обеспечить другой способ проверки, что обычно подразумевает включение в код множества секций `#if #define`.

Дальнейшую информацию можно найти на странице помощи `inotify`.

## A.3. FSYNC



Считается, что функция **fsync** выполняет множество операций ввода и вывода, но это не всегда верно. Подробнее об этом можно прочитать в статье по адресу <http://think.org/tytso/blog/2009/03/15/dont-fear-the-fsync/>.

**Firefox** раньше обращался к библиотеке **sqlite** при нажатии на любую ссылку. В свою очередь **sqlite** вызывала **fsync**, что и служило причиной довольно длительной задержки в ext3 (до 30 секунд), если другой процесс в это время осуществлял копирование большого файла.

Но в других случаях, когда функция **fsync** не использовалась, возникали проблемы в ext4. Для сравнения, состояние памяти в ext3 сохранялось на диск каждые несколько секунд, в то время как в ext4 (в режиме `laptop_mode`) интервал значительно увеличивался и возрастал риск потери данных в случае неожиданного отключения компьютера. Эта проблема была исправлена в ext4, но все же стоит соблюдать осторожность при разработке приложений и использовании **fsync**.

Ниже приведен простой пример чтения и записи файла конфигурации, демонстрирующий создание резервной копии. В этом случае не исключен риск потери данных:

```
/* открытие и чтение файла ~/.kde/myconfig */
fd = open("./kde/myconfig", O_WRONLY|O_TRUNC|O_CREAT);
read(myconfig);
...
write(fd, bufferOfNewData, sizeof(bufferOfNewData));
close(fd);
```

Более оптимальный вариант будет выглядеть так:

```
open("./kde/myconfig", O_WRONLY|O_TRUNC|O_CREAT);
read(myconfig);
...
fd = open("./kde/myconfig.suffix", O_WRONLY|O_TRUNC|O_CREAT);
write(fd, bufferOfNewData, sizeof(bufferOfNewData));
fsync; /* paranoia - optional */
...
close(fd);
rename("./kde/myconfig", "./kde/myconfig~"); /* paranoia - optional */
rename("./kde/myconfig.suffix", "./kde/myconfig");
```

[1] <http://docs.python.org/c-api/init.html#thread-state-and-the-global-interpreter-lock>

[2] [http://www.perlmonks.org/?node\\_id=288022](http://www.perlmonks.org/?node_id=288022)

[3] <http://people.redhat.com/drepper/lt2009.pdf>

## ПРИЛОЖЕНИЕ В. ИСТОРИЯ ИЗМЕНЕНИЙ

<b>Издание 1.0-10.400</b> Rebuild with publican 4.0.0	<b>2013-10-31</b>	<b>Rüdiger Landmann</b>
<b>Издание 1.0-10</b> Rebuild for Publican 3.0	<b>2012-07-18</b>	<b>Anthony Towns</b>
<b>Издание 1.0-2</b> Исправлены опечатки	<b>Fri Oct 22 2010</b>	<b>Rüdiger Landmann</b>
<b>Издание 1.0-1</b> Снята метка чернового варианта	<b>Thu Oct 7 2010</b>	<b>Rüdiger Landmann</b>
<b>Издание 1.0-0</b> Публикация основной версии	<b>Thu Oct 7 2010</b>	<b>Rüdiger Landmann</b>