

# Системный анабиоз

Многие пользователи уже давно знают про одну из интересных особенностей стандарта ACPI ([acpi.sourceforge.net](http://acpi.sourceforge.net)) — так называемые sleep states, или «состояния сна». Они описывают различные режимы энергосбережения, в которые можно переводить ACPI-совместимый компьютер. Кроме обычных и оставшихся со времен АPM, вроде отключения дисков или перевода устройств в режим «полусна», есть несколько новых и самых интересных: это ACPI S3 (или `suspend-to-ram`) и ACPI S4 (или `suspend-to-disk`). Как правило, еще в ходу сокращения STR для S3 и `hibernate` для S4. Мы рассмотрим эти состояния более подробно.

## STR, или `suspend-to-ram`

Это состояние, при котором все компоненты компьютера переводятся в режим максимально низкого энергопотребления (либо вообще отключаются), но питание для оперативной памяти сохраняется — таким образом, не происходит ее разрядка, и содержимое остается прежним. При включении компьютер делает лишь только повторную инициализацию отключенных или «усыпленных» устройств, не производя полной загрузки системы.

## S4, или `suspend-to-disk` aka `hibernate`

Самый интересный режим, позволяющий «заморозить» состояние всей системы, сохранить его на диск и полностью отключить компьютер. Этот режим, который, в отличие от STR, все-таки зависит от состояния электросети, позволяет «спать» компьютеру сколь угодно долго — до тех пор, пока диск не развалится.

Вернемся к пользователям Windows, они знают этот режим под названием `hibernate`, вариации которого можно изменять либо из BIOS (для STR), либо через средства управления самой Windows. Как правило, востребовано все это на переносных компьютерах: вроде как выключил и забыл, потом пришел домой и продолжил работу с того места, где закончил в офисе. Но и для обычных домашних пользователей это полезно — ведь на «засыпание» тратится гораздо меньше времени, чем на за-

грузку с нуля. Здорово, скажете вы, но причем тут Linux? А вот причем: в ней тоже это есть. Поддержка sleep states была добавлена в ядро 2.4 и серьезно улучшена в версии 2.6. Теперь там тоже поддерживаются спящие режимы, причем в несколько более полном варианте.

## Расширенное использование

### Особые возможности

Существует специальный скрипт, облегчающий подготовку компьютера к «усыпленнию». Он находится на сайте `swsusp2`, называется просто и незатейливо `hibernate` и представляет собой набор функций и полезных трюков. Например, с его помощью можно вызвать программу для восстановления видеорежима после пробуждения, настроить параметры сохранения через привычные названия опций, а не через `echo` «видеорежим» >

`/proc/software_suspend2`, выполнить какие-то действия перед засыпанием/после пробуждения и т. д. Вместе с этим скриптом можно использовать популярную программу `acpid`, которая умеет обрабатывать действия нажатия кнопок на корпусе (или датчика закрытия крышки у ноутбука). В общем, `hibernate` — очень полезная во всех отношениях программа, которая рекомендована каждому, кто будет пользоваться `swsusp/swsusp2`.

## | Swsusp, или первая волна |

Для начала немного скучной теории: чтобы компьютер смог сделать S4, необходимо каким-то образом записать «замороженное» состояние на диск. У Windows и большинства ноутбуков с этим проблем нет — нужна всего лишь «правильная» BIOS, умеющая делать запись на специальный раздел диска (как правило, его нельзя увидеть из Windows) и корректно выключать всю систему «по приказу» из Windows.

Для Linux подобный механизм работает плохо — во-первых, «неправильных» BIOS все-таки больше, да и знает он про Linux немного. Поэтому максимум, на что можно рассчитывать, — это вполне корректное выключение/усыпление устройств, и не более того. А все остальные задачи придется решать программно и своими силами. Прибавьте к этому обилие компонентов ядра, изначально не спроектированных на «усыпление», драйверы устройств, в которых отсутствует поддержка восстановления после «просыпания», и т. д. В общем, картина не совсем приятная. Часть этих проблем может решить созданный механизм под названием Swap Suspend, или swsusp, который позволяет выполнять большинство операций (остановка всех выполняемых процессов ядра, запись состояния регистров процессора, перевод драйверов в энергосберегающий режим), а также записывать «замороженный» образ в раздел, предназначенный для своппинга (или подкачки — говоря языком пользователя Windows). К сожалению, первые версии swsusp работали крайне нестабильно: это было связано с плохим тестированием (просто не было достаточной аппаратной базы), да и пользователи особого интереса не испытывали, потому что пользователи могут появиться только у стабильно работающего решения.

## | Swsusp2, или революционный метод запуска вашего компьютера |

К счастью, добрые люди не дали умереть swsusp, и появился форк от основного swsusp под названием Software Suspend 2, или swsusp2 ([www.suspend2.net](http://www.suspend2.net)). Адрес Wiki по swsusp2 — <http://wiki.suspend2.net>. Все части оригинального «заморажи-

### Варианты использования

#### Когда дорога каждая секунда

► **Сервер приложений с критичным временем доступа и важными данными. Вдруг вам понадобилось срочно остановить сервер, и у вас нет времени на его нормальное завершение (например, осталось 30 секунд до окончательной разрядки UPS).**

**Без проблем — «усыпляет» сервер, выключаете его, потом снова включаете, когда неприятности позади, и сервер снова работает!**

► **У вас старый ноутбук, у которого никогда не будет**

**STR или S4. Ставите swsusp2 и «усыпляет» его, когда хотите, так как swsusp2 больше зависит от ядра, а не от версии BIOS.**

► **Нужно срочно заменить вышедший из строя компонент (например, сетевую плату), а у вас сервер с критичным временем доступа (предположим, там расположен хостинг или почта). «Усыпляет» его, заменяете сетевую плату, и, если вы все сделали быстро, никто ничего и не заметит.**



Различные варианты заставок во время «засыпания»

вателя» подверглись серьезной доработке: было затронуто само ядро (в частности, планировщик процессов и различные ключевые компоненты), код многочисленных драйверов (видеоадаптеры, сетевые платы, устройства мультимедиа и т. д.). Появилась возможность сжимать образ памяти перед сохранением, подгонять его под размер свободного места на диске (в документации к swsusp2 этот процесс называется поеданием памяти), стало возможным использовать различные плагины для режимов сохранения (swapwriter, filewriter), появилась поддержка boot splash (крутящийся на экране во время «засыпания» пингвин на сноуборде — это гораздо интереснее, чем синий экран Windows). И самое главное — можно «усыплять» не только ноутбуки, но и серьезные серверы, так как теперь поддерживаются SMP и HIGHMEM (то есть количество памяти превышает 1 Гбайт)! Есть патчи для поддержки PPC и x86-64, но пока они не признаны очень стабильными. На сегодняшний день осталось еще много недостатков в работе swsusp2, но большинство из них можно решить — например, не пользоваться закрытыми коммерческими драйверами (для видеоплат ATI или NVIDIA) или выбирать компоненты от более-менее известных производителей, так как гораздо больше шансов, что, например, «кривая» BIOS быстрее исправит ASUS, чем Elitegroup.

## | Заключение |

Если вы заинтригованы и решили проверить, как работает swsusp2 на вашем компьютере с дистрибутивом Linux, к тому же этот дистрибутив — от ALT Linux (например, Master-2.4), то вам крупно повезло. Так как для него ничего дополнительно собирать или скачивать не надо, достаточно поставить из backports специальное ядро (из серии -wks26) и сам скрипт управления hibernate. Если в вашем дистрибутиве таких возможностей нет, обратитесь к документации на сайте swsusp2 или расположенному там же списку рассылки для пользователей. А если вы разработчик и желаете помочь проекту своими патчами или отчетами о замеченных ошибках, присоединяйтесь к обществу создателей swsusp2. В любом случае вы поможете и себе, и другим. |