

Power management/Suspend and hibernate (Русский)

< [Power management](#)

Состояние перевода: На этой странице представлен перевод статьи [Power management/Suspend and hibernate](#). Дата последней синхронизации: 22 июля 2017. Вы можете [помочь](#) синхронизировать перевод, если в английской версии произошли [изменения](#) (https://wiki.archlinux.org/index.php?title=Power_management/Suspend_and_hibernate&diff=0&oldid=482651).

Ссылки по теме

[Uswsusp](#)

[TuxOnIce](#)

[systemd \(Русский\)](#)

[Power management](#)

В настоящее время существует три метода приостановки работы компьютера: **suspend to RAM** обычно называемая просто **suspend**(приостановка, ждущий режим, сон, STR, S3), **suspend to disk** известный как **hibernate**(гибернация, спящий режим, STD, S4), и **hybrid suspend**(гибридная приостановка, гибридный спящий режим, иногда применяется название **suspend to both**):

- **Suspend to RAM** отключает питание большинства частей компьютера, кроме ОЗУ, что требуется для восстановления состояния машины. Из-за большой экономии энергии рекомендуется, чтобы ноутбуки автоматически входили в этот режим, когда компьютер работает от батарей, и крышка закрыта или пользователь неактивен в течение некоторого времени.
- **Suspend to disk** метод сохраняет состояние машины на диске [Swap \(Русский\)](#) и полностью отключает компьютер, потребления электроэнергии нет. Когда устройство включается, состояние восстанавливается.
- **Suspend to both** сохраняет состояние машины на диске в свопе, но не выключает ее. Вместо этого выполняется обычная приостановка в ОЗУ. Поэтому, если батарея не разряжена, система может возобновиться из ОЗУ. Если батарея разряжена, система может быть возобновлена с диска, что намного медленнее, чем возобновление работы из ОЗУ, но состояние машины не будет потеряно.

Существует несколько низкоуровневых интерфейсов, обеспечивающих базовые функции, а также некоторые интерфейсы высокого уровня, обеспечивающие трюки для обработки проблемных аппаратных драйверов / модулей ядра (например, повторная инициализация видеокарты).

Contents

Низкоуровневые интерфейсы

[kernel \(swsusp\)](#)[uswsusp](#)[Интерфейсы высокого уровня](#)[systemd](#)[Гибернация](#)[Про размер раздела/файла подкачки](#)[Необходимые параметры ядра](#)[Гибернация в файл подкачки](#)[Настройка initramfs](#)[Исправление проблем](#)[ACPI_OS_NAME](#)[Пользователям VAIO](#)[Ждущий/Спящий режим не работает или сбоят](#)[Wake-on-LAN](#)[Instantaneous wakeups from suspend](#)

Низкоуровневые интерфейсы

Хотя эти интерфейсы могут использоваться напрямую, рекомендуется использовать какой-либо из [#Интерфейсы высокого уровня](#) для ждущего / спящего режима. Использование низкоуровневых интерфейсов напрямую существенно быстрее, чем использование любого интерфейса высокого уровня, поскольку запуск всех хуков перед и после режима приостановки требует времени, но хуки могут правильно устанавливать аппаратные часы, восстанавливать беспроводное соединение и т.д.

kernel (swsusp)

Самый простой подход для входа в режим сна заключается в прямом информировании встроенного программного кода ядра (swsusp); точный метод и состояние зависят от уровня аппаратной поддержки. В современных ядрах основным механизмом переключения режимов является запись соответствующих значений в `/sys/power/state`.

Смотрите [документацию \(https://www.kernel.org/doc/Documentation/power/states.txt\)](https://www.kernel.org/doc/Documentation/power/states.txt) для подробностей.

uswsusp

Uswsusp ('Userspace Software Suspend') представляет собой оболочку ядерного механизма приостановки в ОЗУ, которая выполняет некоторые манипуляции с графическим адаптером из пользовательского пространства перед приостановкой и после возобновления.

Смотрите основную статью [Uswsusp](#).

Интерфейсы высокого уровня

Конечной целью этих пакетов является предоставление программ(двоичных файлов/скриптов), которые могут быть вызваны для выполнения приостановки компьютера. Фактическая привязка их к кнопкам питания, щелчкам меню или событиям крышки ноутбука обычно предоставляется другим инструментам. Чтобы автоматически приостановить работу при определенных событиях, таких как закрытие крышки ноутбука или процент истощения батареи, вам может потребоваться запустить [Acpid](#).

systemd

[systemd](#) предоставляет собственные команды для ждущего, спящего и гибридного режима приостановки, смотрите [Power management#Power management with systemd](#) для деталей. Это интерфейс по умолчанию, используемый в Arch Linux. Смотрите [Power management#Sleep hooks](#) для получения дополнительной информации о настройке хуков режимов сна. Также смотрите [systemctl\(1\)](#) (<https://jlk.fjfi.cvut.cz/arch/manpages/man/systemctl.1>), [systemd-sleep\(8\)](#) (<https://jlk.fjfi.cvut.cz/arch/manpages/man/systemd-sleep.8>) и [systemd.special\(7\)](#) (<https://jlk.fjfi.cvut.cz/arch/manpages/man/systemd.special.7>).

Гибернация

Чтобы использовать спящий режим, вам нужно создать [swap](#) раздел или файл. Вам нужно будет указать ядру на своп, используя параметр `resume=`, который настраивается через загрузчик. Вам также понадобится [настроить initramfs](#)^{[[broken link: invalid section](#)]}. Это говорит ядру попытаться возобновить работу с указанного свопа в раннем пользовательском пространстве. Эти три этапа подробно описаны ниже.

Про размер раздела/файла подкачки

Даже если ваш раздел подкачки меньше ОЗУ, у вас все еще есть большая вероятность успешно перейти в спящий режим. Согласно [ядерной документации \(https://www.kernel.org/doc/Documentation/power/interface.txt\)](https://www.kernel.org/doc/Documentation/power/interface.txt):

/sys/power/image_size управляет размером образа, создаваемого механизмом приостановки на диск. Это может быть строка, представляющая неотрицательное целое число, которое будет использоваться в качестве верхнего предела размера образа в байтах. Механизм приостановки сделает все возможное, чтобы размер образа не превышал это число. Однако, если это окажется невозможным, он попытается приостановить все равно, используя наименьший возможный размер образа. В частности, если в этот файл записать «0», размер образа будет настолько мал на сколько это возможно. Чтение из этого файла отображает текущее ограничение размера образа, которое по умолчанию установлено на 2/5 доступного ОЗУ.

Вы можете либо уменьшить значение `/sys/power/image_size`, чтобы сделать образ как можно меньшим (для небольших разделов подкачки) или увеличить его, чтобы ускорить процесс гибернации.

Необходимые параметры ядра

Должен быть использован параметр ядра `resume=swap_partition`. Либо имя, назначенное ядром для раздела, либо его [UUID](#), можно использовать как `swap_partition`. Например:

- `resume=/dev/sda1`
- `resume=UUID=4209c845-f495-4c43-8a03-5363dd433153`
- `resume=/dev/mapper/archVolumeGroup-archLogicVolume` -- если используется LVM

В общем, метод именования, используемый для параметра `resume`, должен быть таким же, как и для параметра `root`. Конфигурация зависит от используемого [boot loader](#), обратитесь к [Kernel parameters](#) за деталями.

Гибернация в файл подкачки

Важно: [Btrfs](#) не поддерживает файлы подкачки. Несоблюдение этого предупреждения может привести к повреждению файловой системы. В то же время когда файл подкачки монтируется через устройство `/dev/loop` его можно использовать и на Btrfs, но это приведет к значительному ухудшению производительности подкачки.

Использование файла вместо раздела подкачки требует дополнительного параметра для ядра `resume_offset=swap_file_offset`.

Значение `swap_file_offset` можно получить запустив `filefrag -v swap_file`, требуемое значение расположено в столбце `physical_offset` первого ряда таблицы выводимой командой. Например:

```
# filefrag -v /swapfile

Filesystem type is: ef53
File size of /swapfile is 4294967296 (1048576 blocks of 4096 bytes)
ext:      logical_offset:      physical_offset: length:  expected: flags:
  0:         0..          0:      38912..      38912:         1:
  1:         1..      22527:      38913..      61439:      22527:                unwritten
  2:      22528..      53247:      899072..      929791:      30720:      61440: unwritten
...
```

В этом примере значение `swap_file_offset` это первое число 38912 с двумя точками.

Значение `swap_file_offset` так же может быть получено с помощью `swap-offset swap_file`. Файл команды `swap-offset` предоставляется пакетом [uswsusp-git](#)

(<https://aur.archlinux.org/packages/uswsusp-git/>)^{AUR}.

Примечание:

- Параметр ядра `resume` определяет устройство, раздел которого содержит файл подкачки, а не сам файл подкачки. О местонахождении файла подкачки на устройстве возобновления систему информирует параметр `resume_offset`. Перед первой гибернацией требуется перезагрузка для их активации.
- Если вы используете **uswsusp**, то эти два параметра должны быть представлены в `/etc/suspend.conf` с помощью ключей `resume device` и `resume offset`. В этом случае перезагрузка не требуется.

Совет: Возможно, вы захотите уменьшить **Swap#Swappiness** для вашего файла подкачки, если единственная цель способность обеспечить спящий режим, а не расширение ОЗУ.

Настройка initramfs

- Когда используется **initramfs** с хуком `base`, а по умолчанию это так, хук `resume` требуется в `/etc/mkinitcpio.conf`. Будь то по метке или по UUID раздел подкачки ссылается на файл устройства создаваемый `udev`, поэтому хук `resume` должен идти «после» хука `udev`. Этот пример был сделан на основе конфигурации хуков по умолчанию:

```
HOOKS="base udev resume autodetect modconf block filesystems keyboard fsck"
```

Не забудьте **пересобрать образ initramfs** чтобы эти изменения вступили в силу.

Примечание: **LVM** пользователи должны добавить `resume` хук после `lvm2`.

- Когда используется `initramfs` с хуком `systemd`, механизм возобновления уже предоставлен и дополнительные хуки не нужны.

Исправление проблем

ACPI_OS_NAME

Возможно, вы захотите настроить свою **таблицу DSDT**, чтобы заставить ее работать. Смотрите статью **DSDT**

Пользователям VAIO

Добавьте `acpi_sleep=nonvs` параметр ядра в ваш загрузчик и возьмите себе с полки

пирожок!

Ждущий/Спящий режим не работает или сбоит

There have been many reports about the screen going black without easily viewable errors or the ability to do anything when going into and coming back from suspend and/or hibernate. These problems have been seen on both laptops and desktops. This is not an official solution, but switching to an older kernel, especially the LTS-kernel, will probably fix this.

Sometimes the screen goes black due to device initialization from within the initramfs. Removing any modules you might have in [Mkinitcpio#MODULES](#) and rebuilding the initramfs, can possibly solve this issue, specially graphics drivers for [early KMS](#). Initializing such devices before resuming can cause inconsistencies that prevents the system resuming from hibernation. This does not affect resuming from RAM. Also, check this [article \(https://01.org/blogs/rzhang/2015/best-practice-debug-linux-suspend/hibernate-issues\)](https://01.org/blogs/rzhang/2015/best-practice-debug-linux-suspend/hibernate-issues) for the best practices to debug suspend/hibernate issues.

For Intel graphics drivers, enabling early KMS may help to solve the blank screen issue. Refer to [Kernel mode setting#Early KMS start](#) for details.

Wake-on-LAN

If [Wake-on-LAN](#) is active, the network interface card will consume power even if the computer is hibernated.

Instantaneous wakeups from suspend

For some Intel Haswell systems with the LynxPoint and LynxPoint-LP chipset, instantaneous wakeups after suspend are reported. They are linked to erroneous BIOS ACPI implementations and how the `xhci_hcd` module interprets it during boot. As a work-around reported affected systems are added to a blacklist (named `XHCI_SPURIOUS_WAKEUP`) by the kernel case-by-case. [\[1\] \(https://bugzilla.kernel.org/show_bug.cgi?id=66171#c6\)](https://bugzilla.kernel.org/show_bug.cgi?id=66171#c6)

Instantaneous resume may happen, for example, if a USB device is plugged during suspend and ACPI wakeup triggers are enabled. A viable work-around for such a system, if it is not on the blacklist yet, is to disable the wakeup triggers. An example to disable wakeup through USB is described as follows. [\[2\] \(https://bbs.archlinux.org/viewtopic.php?pid=1575617\)](https://bbs.archlinux.org/viewtopic.php?pid=1575617)

To view the current configuration:

```
$ cat /proc/acpi/wakeup
```

Device	S-state	Status	Sysfs node
...			
EHC1	S3	*enabled	pci:0000:00:1d.0
EHC2	S3	*enabled	pci:0000:00:1a.0
XHC	S3	*enabled	pci:0000:00:14.0
...			

The relevant devices are EHC1 , EHC1 and XHC (for USB 3.0). To toggle their state you have to echo the device name to the file as root.

```
# echo EHC1 > /proc/acpi/wakeup  
# echo EHC2 > /proc/acpi/wakeup  
# echo XHC > /proc/acpi/wakeup
```

This should result in suspension working again. However, this settings are only temporary and would have to be set at every reboot. To automate this take a look at [systemd#Writing unit files](#). See [BBS thread \(https://bbs.archlinux.org/viewtopic.php?pid=1575617#p1575617\)](https://bbs.archlinux.org/viewtopic.php?pid=1575617#p1575617) for a possible solution and more information.

Retrieved from "[https://wiki.archlinux.org/index.php?title=Power_management/Suspend_and_hibernate_\(Русский\)&oldid=583403](https://wiki.archlinux.org/index.php?title=Power_management/Suspend_and_hibernate_(Русский)&oldid=583403)"

This page was last edited on 20 September 2019, at 07:00.

Content is available under [GNU Free Documentation License 1.3 or later](#) unless otherwise noted.