

System time (Русский)

[Jump to navigation](#)[Jump to search](#)



Эта статья или раздел нуждается в [переводе](#)



Примечания: Перевод устарел. (обсуждение: [Talk:System time \(Русский\)#](#))



Эта страница нуждается в сопроводителе



Статья не гарантирует актуальность информации. Помогите русскоязычному сообществу поддержкой подобных страниц. См. [Команда переводчиков ArchWiki](#)

В операционной системе время (часы) определяется четырьмя понятиями: значение времени, стандарт времени, часовой пояс и, если необходимо, летнее время (Daylight Saving Time (DST)). В этой статье рассказывается о каждом из них, а также как прочитать/установить их значения.



Contents

- [1 Введение](#)
- [2 Стандарты времени](#)
- [3 Часовой пояс](#)
- [4 Установка времени](#)
- [5 Если нет файла rc.conf](#)
- [6 Погрешность часов](#)
- [7 Синхронизация времени](#)
- [8 Возможные проблемы](#)
- [9 См. также](#)

Введение

Аппаратные часы встроены в ваш компьютер, предоставляют информацию о годе, месяце, дне, часе, минуте и секунде и работают, даже когда он выключен. Они могут вести отсчет местному (локальному) времени или времени в стандарте **UTC**. Подробнее на [Википедии](#). Фактически, местное время - это то время, которое показывают ваши обычные часы (к примеру, любимые часы вашей бабушки), а время UTC, не вдаваясь в подробности, едино для всех жителей земного шара, точкой отсчета для которого является Гринвичский меридиан. По отношению ко времени UTC, ваше локальное время может отличаться на несколько часов. Поскольку вы читаете именно русскоязычный вариант статьи, то, скорее всего, ваше локальное время больше от трех до двенадцати часов по сравнению с **UTC**.

Все наиболее распространенные системы могут использовать локальное или UTC-время. По умолчанию, UNIX-подобные операционные системы (включая Mac OS) используют UTC, в то время как Windows использует локальное время. В случае если у вас на компьютере установлено несколько операционных систем, лучше всего будет использовать в них UTC чтобы избежать конфликтов.

В UNIX-подобных операционных системах аппаратные часы могут быть проверены и сконфигурированы командой `hwclock`.

Кроме аппаратных часов, Linux также имеет системные (программные) часы, которые работают независимо. Системные часы отслеживают время, часовой пояс и контролируют, используется ли в вашей локали (локаль, как вы помните, настраивалась при установке

системы в файле `/etc/locale.conf`) летнее время.

Демон `/etc/rc.d/hwclock` предоставляет работу с командой `hwclock` и устанавливает системное время по аппаратным часам при загрузке операционной системы и наоборот, корректирует аппаратное время при завершении ее работы (ведь при своей работе операционная система может узнать более точное время, например от [NTP-сервера](#)). Таким образом, пользователи должны включать демон 'hwclock' в массив DAEMONS в `/etc/rc.conf`, чтобы это происходило автоматически. В качестве альтернативы этому демону, пользователи часто используют NTP для синхронизации времени с т.н. серверами времени в интернете.

Стандарты времени

Все наиболее распространенные операционные системы могут использовать локальное или UTC-время. По умолчанию, UNIX-подобные операционные системы (включая Mac OS) используют UTC, в то время как Windows использует локальное время. В случае если у вас на компьютере установлено несколько операционных систем, лучше всего будет использовать в них UTC чтобы избежать конфликтов. GNU/Linux будет переходить на летнее время и обратно автоматически, только в том случае, если вы используете UTC, независимо от того, летнее или зимнее время сейчас. Если вы используете местное время, GNU/Linux не будет осуществлять переход, предполагая, что у вас несколько операционных систем и другая операционная система сделает это за нее. Если это не так, то вам придется переводить часы вручную.

Вы можете установить стандарт временного отсчета средствами командной строки. Для начала проверьте, какой стандарт вы выбрали, когда устанавливали систему:

```
$ grep ^HARDWARECLOCK /etc/rc.conf
```

Если вы используете также Windows, то лучшим выбором будет *localtime*. Чтобы изменить стандарт временного отсчета на локальное время, используйте:

```
# hwclock --localtime
```

Или установить стандарт UTC:

```
# hwclock --utc
```

После этих операций скорее всего придется обновить ваши аппаратные и системные часы. См. ниже [#Установка времени](#).

Часовой пояс

Удостоверьтесь что вы правильно указали ваш часовой пояс в `/etc/rc.conf`, это значения используется не только для часов, но и для других программ. Чтобы проверить значение, введите:

```
$ grep ^TIMEZONE /etc/rc.conf
```

Вы можете просмотреть полный список часовых поясов в файле `/usr/share/zoneinfo/` и выберите крупный город в одном с вами часовом поясе. Если вы плохо разбираетесь в географии, то вы всегда сможете посетить страницу [в Википедии](#), на которой приведен полный список часовых поясов. Кроме указания ближайшего города, вы также можете выбрать часовой пояс непосредственно. Пример:

```
TIMEZONE="America/Chicago"
```

Выбранный часовой пояс будет подхвачен системой после перезагрузки. Для того, чтобы изменить часовой пояс немедленно, в должны создать ссылку или скопировать соответствующий файл в `/etc/localtime`:

```
# cp /usr/share/zoneinfo/America/Chicago /etc/localtime
```

Далее вам потребуется установить точное время на аппаратных часах.

Установка времени

Аппаратные часы могут быть установлены по системным. Посмотрите, какое время установлено на ваших аппаратных часах:

```
$ hwclock --show
```

И системных часах:

```
$ date
```

Чтобы установить аппаратные часы непосредственно:

```
# hwclock --set --date "MM/DD/YYYY hh:mm:ss"
```

Установить системные часы:

```
# date MMDDhhmmYYYY
```

Аппаратные часы можно установить по системным и наоборот:

```
# hwclock --systohc  
# hwclock --hctosys
```

Если нет файла `rc.conf`

Для того, чтобы сравнить системное время и UTC:

```
$ timedatectl status
```

Для установки системных часов:

```
# timedatectl set-time "2012-10-30 18:17:16"
```

Изменить Вашу зону и субзону:

```
# timedatectl set-timezone Zone/SubZone
```

Лист доступных зон:

```
$ timedatectl list-timezones
```

Пример установки зон:

```
# timedatectl set-timezone Europe/Moscow
```

Погрешность часов

Нет ничего идеального. В том числе и часов. Время на любых часах хоть немного но отличается от реального. Одними из наиболее точных считаются [атомные часы](#). Кварцевые часы, используемые в компьютерах, также не являются абсолютно точными, зато имеют относительно постоянную погрешность.

Каждый раз перед выключением демон `hwclock` сравнивает аппаратное и системное время (последнее принимается за наиболее точное) и вычисляет погрешность аппаратных кварцевых часов в день. Вычисленное значение погрешности записывается в файл `/var/lib/hwclock/adjtime`. Если `hwclock` не использовался больше суток, то он, на всякий случай, вычисляет новое значение погрешности.

Специальный скрипт корректирует аппаратные часы в соответствии с вычисленной погрешностью каждый час. Если вы видите, что ваши аппаратные часы идут все более и более неверно, то, вероятнее всего, в файле `/var/lib/hwclock/adjtime` записано неверное значение погрешности.

Это происходит если на аппаратных часах время изначально было установлено неверно или ваш временной стандарт (см. выше) не такой как у другой операционной системы, используемой на компьютере. Чтобы исправить это, удалите `/var/lib/hwclock/adjtime` и настройте время на всех ваших часах.

Примечание: Если вы всегда выключаете свой компьютер чаще чем раз в сутки, погрешность, записанная в `/var/lib/hwclock/adjtime` не будет перепроверяться `hwclock`. [Кварцевый генератор](#), конечно, надежная штука, но со временем параметры его могут меняться (например батарейка CMOS разряжается). Возможно, стоит иногда выключать компьютер реже чем раз в сутки. Тогда можно быть уверенным, что значение погрешности, по которому корректируются часы, будет самым новым, а часы самыми точными.

Системное время рассчитывается ядром, начиная с полуночи первого января 1st 1970 по стандарту UTC. Начальное значение системных часов рассчитывается исходя из показаний часов аппаратных (после корректировки погрешности и пересчета на местный часовой пояс UTC) при включении и идут независимо от аппаратных. Ядро Linux отслеживает системные часы, подсчитывая прерывания таймера. Как и аппаратные, они не являются идеально точными. Иногда системные часы могут потерять точность, если ядро сбрасывает прерывания. Но показания системных часов можно сделать точными с помощью [NTP](#). NTP будет регулировать частоту прерываний и количество тактов в секунду для уменьшения дрейфа системных часов. Эти значения могут быть скорректированы с помощью приложения `adjtimex` из AUR.

Примечание: Если вы не используете NTP, то ваши аппаратные часы будут более точными, чем системные. Соответственно описанный механизм корректировки будет только вносить дополнительную погрешность и его лучше отключить. Для этого можно закомментировать строку `hwclock --systohc` в файле `{/etc/rc.d/hwclock`. Теперь при выключении компьютера система не будет изменять значение ваших аппаратных часов.

Синхронизация времени

Протокол синхронизации времени (NTP) - это протокол для синхронизации часов компьютера с помощью пакетов в сетях с переменными задержками доставки данных. Вот реализации этого протокола:

- **Network Time Protocol daemon (Русский)** - это [Wikipedia:ru:эталонная реализация](https://ru.wikipedia.org/wiki/Network_Time_Protocol_daemon) протокола, особенно рекомендуется к использованию на серверах времени. Он также регулирует частоту прерываний и количество тактов в секунду, чтобы уменьшить дрейф системных часов, и будет снова синхронизировать аппаратные часы каждые 11 минут.
- **ntpd** - это [wikipedia:ru:SNTP](https://ru.wikipedia.org/wiki/ntpd) клиент, который входит в пакет [ntp](https://ru.wikipedia.org/wiki/ntp). Он заменяет собой *ntpd* и рекомендуется к использованию на не серверах.
- **systemd-timesyncd (Русский)** - это простой [wikipedia:ru:SNTP](https://ru.wikipedia.org/wiki/systemd-timesyncd) демон, реализующий только клиентскую часть, ориентируется только на запрос времени с одного удалённого сервера. Этот вариант подходит большинству пользователей.
- **OpenNTPD** - это часть проекта OpenBSD, реализует как клиентскую, так и серверную часть.
- **Chrony** - это клиент и сервер, который дружелюбен к роумингу и специально разработан для систем, которые не всегда онлайн.

Возможные проблемы

На некоторых ноутбуках (HP Compaq 6715s) была выявлена [1](#), [2](#) проблема. При этом системные часы спешили на два часа. После ввода команды:

```
$ hwclock --debug
```

Появляется следующий вывод консоли:

```
hwclock из util-linux 2.19.1
Используется /dev interface to clock.
Дата последней корректировки отклонения - 1311726134 секунд после 1969
Дата последней калибровки - 1310756173 секунд после 1969
Аппаратные часы выставлены по местному времени
Подразумевается, что аппаратные часы выставлены по местному времени.
Ожидается тиканье часов...
Истекло время ожидания тиканья часов select() в /dev/rtc # Собственно, строка
с ошибкой
...synchronization failed
```

Подобная ошибка возникает при загрузке и выключении системы (ядро не может прочитать/изменить аппаратное время), а также при последующих операциях по конфигурированию времени. Вероятно, проблема в драйвере, включенном в ядро по умолчанию. Некоторые пользователи других дистрибутивов решали проблему использованием другого драйвера или добавлением ключей `--directisa` в скрипт демона.

Добавление демона `hwclock` в `rc.conf` по крайней мере решило проблему со спешащими часами. Тем не менее, любые запросы к аппаратным часам проходят один раз из семи. Если найдется решение проблемы, пожалуйста, свяжитесь с первым автором перевода этой статьи.

См. также

- [Личная служба точного времени: hwclock и ntp](#) на posix.ru.
- [Время](#) (Википедия)
- [Источники данных, определяющие часовые пояса и переход на летнее время для пакета tzdata](#)

Category:

- [Русский](#)