



OBS.D.R.U

[Home](#) > [Документация](#)

7 Настройка дисплея и клавиатуры

[printable page](#)

7.1 - Как я могу перенастроить клавиатуру? (*wscons*)

Архитектуры, на которые портирован консольный драйвер [wscons\(4\)](#): [alpha](#), [amd64](#), [hppa](#), [i386](#), [mac68k](#), [macppc](#), [sparc](#), [sparc64](#), [vax](#) и [zaurus](#).

В консолях [wscons\(4\)](#), большинство опций может быть изменено с помощью [wsconsctl\(8\)](#). К примеру, что бы изменить маппинг раскладки клавиатуры используя [wsconsctl\(8\)](#), вы должны сделать:

```
# wsconsctl -w keyboard.encoding=ru
```

В следующем примере, мы переназначим функцию клавиши "Caps Lock" на "Control L" (Control-L)

```
# wsconsctl -w keyboard.map+="keysym Caps_Lock = Control_L"
```

7.2 - Есть ли поддержка мыши в консоли OpenBSD?

[alpha](#), [amd64](#), и [i386](#), OpenBSD предоставляет [wsmoused\(8\)](#), который является портом [moused\(8\)](#) из FreeBSD. Вы можете включить его, изменив соответствующим образом ваш [rc.conf\(8\)](#).

7.3 - Очистка экрана консоли при завершении сеанса пользователя

Вам надо добавить строки в [/etc/gettytab\(5\)](#). Измените текущую секцию:

```
P|Pc|Pc console:\n      :np:sp#9600:
```

добавив ":cl=\E[H\E[2J:" в конец. У вас получится что-то вроде:

```
P|Pc|Pc console:\
      :np:sp#9600:\
      :cl=\E[H\E[2J:
```

7.4 - Доступ к буферу прокрутки консоли (*amd64, i386, некоторые Alpha*)

На некоторых платформах OpenBSD предоставляет буфер прокрутки экрана консоли. Он позволяет вам увидеть информацию, которая уже скрылась за пределами вашего экрана. Для перемещения вверх и вниз используйте комбинации клавиш [SHIFT]+[PGUP] и [SHIFT]+[PGDN] соответственно.

По умолчанию, размер буфера прокрутки - 8 экранов. Это стандартная фишка драйвера [vga\(4\)](#), и она не будет работать без VGA-совместимой карты на любой платформе (многие Alpha-системы имеют TGA).

7.5 - Как мне переключать консоли? (*amd64, i386, Zaurus, некоторые Alpha*)

На платформах amd64, i386 и Alpha с [vga\(4\)](#)-видеокартой, OpenBSD предоставляет шесть виртуальных консолей по умолчанию, с /dev/ttyC0 по /dev/ttyC5. ttyC4 по умолчанию зарезервирована для использования X Window System. Вы можете переключаться между ними, используя [CTRL]+[ALT]+[F1], [CTRL]+[ALT]+[F2], [CTRL]+[ALT]+[F3], [CTRL]+[ALT]+[F4] и [CTRL]+[ALT]+[F6].

X использует ttyC4, [CTRL]+[ALT]+[F5]. Нажав [CTRL]+[ALT]+[Fn], вы попадете в соответствующую консоль, нажав [CTRL]+[ALT]+[F5], вы попадете в X.

Если вы хотите больше консолей, чем шесть, используйте [wsconscfg\(8\)](#) чтобы создать ttyC6, ttyC7 или более. Пример:

```
# wsconscfg -t 80x25 6
```

позволит создать ttyC6, доступный при нажатии на [CTRL]+[ALT]+[F7]. Не забудьте добавить эту команду в [rc.local\(8\)](#), если вы хотите, чтобы эти консоли добавлялись каждый раз при загрузке системы.

Заметьте, что вы не получите приглашение "login:" на только-что созданной виртуальной консоли, пока не установите значение "on" в [/etc/ttys\(5\)](#), затем перезагрузитесь или пошлете процессу [init\(8\)](#) сигнал HUP, используя [kill\(1\)](#).

На машинах Zaurus существуют два терминала (/dev/ttyC0 и /dev/ttyC1) по умолчанию, доступных по нажатию [ALT]+[CALENDAR] и [ALT]+[ADDRESS] (Клавиша [ALT] находится справа от левой клавиши [CTRL]).

7.6 - Как изменить разрешение в консоли на 80x50? (amd64, i386, некоторые Alpha)

Как правило, пользователи amd64, i386 и VGA Alpha получают вывод на консоль 25 строк по 80 символов. Однако, некоторые VGA видеокарты позволяют выводить текст с большим текстовым разрешением 50 строк по 80 знаков.

Первоначально должен быть загружен шрифт для поддержки данного режима с помощью команды [wsfontload\(8\)](#). Стандартный вывод текста 80x25 использует шрифт размером 8x16 пикселей; для удвоения числа строк мы используем шрифт размером 8x8 пикселей.

После этого, мы должны удалить и заново создать виртуальные консоли в нужном нам решении экрана, используя команду [wsconscfg\(8\)](#).

Это может быть сделано автоматически во время загрузки, с помощью добавления следующих строк в конец вашего файла [rc.local\(8\)](#):

```
wsfontload -h 8 -e ibm /usr/share/misc/pcvtfonts/vt2201.808
wsconscfg -dF 5
wsconscfg -t 80x50 5
```

Как и при любом другом изменении системной конфигурации, мы рекомендуем вам провести некоторое время за чтением страниц соответствующих руководств для того, чтобы вы поняли что делают эти команды.

Первая строчка загружает шрифты 8x8. Вторая строка удаляет 5 экран (который будет доступен по комбинации клавиш [CTRL]+[ALT]+[F6]). Третья строка заново создает 5 экран с 50 линиями и 80 символами на каждой линии. Если вы сделаете это, то увидите, что основной и остальные три экрана остаются в прежнем стандартном режиме 80x25, но новая пятая виртуальная консоль в режиме 80x50 доступна по [CTRL]+[ALT]+[F6].

Запомните, что [CTRL]+[ALT]+[F1] – это экран 0 (ttyC0). Если вы желаете изменить другие экраны, просто повторите шаги удаления и добавления виртуальных консолей для всех экранов, которые вы хотите перевести в разрешение 80x50.

Вы должны избегать изменение разрешения экрана 4 (ttyC4, [CTRL]+[ALT]+[F5]), который используется X в качестве графического. Также не рекомендуется изменять разрешение основной консоли (т. е. ttyC0).

Все эти команды могут быть введены в командную строку от имени пользователя root или (рекомендуется) используя [sudo\(8\)](#).

Примечание: это не будет работать на всех видеокартах. К сожалению, не все видеокарты поддерживают загружаемые шрифты, которые необходимы [wscons\(4\)](#) для применения разрешения 80x50 в текстовом режиме. В этих случаях вы можете рассмотреть возможность запуска X.

7.7 - Как использовать консоль на последовательном порту?

Существует много причин, почему вы можете захотеть использовать консольный порт:

- Запись вывода консоли (для документации).
- Удаленное управление.
- Удобную поддержку большого числа машин.
- Предоставление вывода `dmesg` с машин, который, возможно, тяжело получить другим путем.
- Предоставление точного вывода `"trace"` и `"ps"` для разработчиков, в случае краха системы.

OpenBSD поддерживает консоль на последовательном порту на большинстве платформ, однако существует детальное различие для разных платформ.

Отметим, что работа с консольным портом НЕ простая задача -- у вас часто будет возникать необходимость в бесполезных проводах, порты не будут стандартизированы на различных машинах, в некоторых случаях не будут совместимы даже на одной. Предположим, что вы знаете как выбрать подходящий кабель для связи вашего компьютера и устройства, выступающего в роле последовательного терминала, поэтому не станем заострять на этом внимание. Полное описание последовательного интерфейса выходит за пределы этой статьи, однако напомним: обычное подключение концов провода не будет означать что все будет работать.

Изменения в `/etc/ttys`

Есть две вещи, которые необходимо сделать для того чтобы заработал консольный порт на OpenBSD. Во-первых, у вас должна уже быть машина на OpenBSD, использующая последовательный порт в качестве консольного для `"status"` и однопользовательского режима. Реализация этой части сильно зависит от платформы. Во-вторых, вы должны разрешить использование последовательного порта в качестве терминального, так чтобы пользователь мог зайти в систему в многопользовательском режиме. Эта часть в некоторой степени одинаково выполнима на различных платформах, и расписана здесь подробно.

Терминальные сессии управляются файлом `/etc/ttys`. Для того, чтобы OpenBSD вывело приглашение `"login:"` на устройство, его (устройство) необходимо разрешить в `/etc/ttys`, в конце концов есть другие способы применения последовательного порта, нежели в качестве терминала. На платформах, с подключаемыми клавиатурой и монитором, в качестве консоли, последовательный терминал отключен по-умолчанию. Приведем пример для платформы i386. В этом случае, замени строку:

```
tty00  "/usr/libexec/getty std.9600"  unknown off
```

на:

```
tty00  "/usr/libexec/getty std.9600"  vt100  on secure
```

Здесь `tty00` - последовательный порт, используемый в качестве консольного. `vt220` наден `[url=http://www.openbsd.org/cgi-bin/man.cgi?query=termcap&sektion=5]termcap(5)` для ВАШЕГО терминала (могут быть другие - `vt100`, `xterm` и т.д.). "он" стимулируют `getty` к выводу приглашения "login:", "secure" разрешает логиниться пользователю `root (uid=0)` на этой консоли (что может соответствовать или не соответствовать вашим желаниям), и "9600" - максимальная скорость двоичной передачи. Попытка поднять скорость обмена до величины, максимально поддерживаемой вашим оборудованием скорее создаст проблемы, нежели сделает работу более комфортной. Most systems have a "default" speed (supported by default by the boot ROM and/or the boot loader, often 9600), use this unless you have real reason to use something different.

Отметим, что при установке вы можете использовать консольный порт без этих шагов, т.к. в это время система работает в однопользовательском режиме и не использует `getty` для входа.

На некоторых платформах и при некоторых конфигурациях, вы должны запустить систему в однопользовательском режиме для внесения этих изменений, если последовательная консоль - все что вам доступно.

amd64 и i386

Для того чтобы сразу во время загрузки использовать последовательный порт в качестве консоли, создайте или отредактируйте ваш файл `/etc/boot.conf`. Добавьте туда строку:

```
set tty com0
```

для использования первого последовательного порта в качестве вашей консоли. Скорость передачи информации по умолчанию равна 9600bps. Её можно изменить, модифицировав в `/etc/boot.conf` строку, использующую опцию `stty`. Этот файл помещен на ваш загрузочный носитель, которым может быть и инсталляционная дискета (флоппи), либо команда может быть введена в командной строке [загрузчика OpenBSD](#) для одноразового (или в первый раз) использования последовательной консоли.

Замечания по amd64 и i386:

OpenBSD нумерует последовательные порты начиная с `tty00`, DOS/Windows помечает их начиная с `COM1`. Поэтому, помните, что `tty02` это `COM3`, а не `COM2`.

Некоторые системы позволяют работать без видеокарты на машине, но не все -- многие системы рассматривают это как ошибку. Некоторые машины также отказываются работать без подключенной клавиатуры.

Некоторые системы способны перенаправлять все действия BIOS клавиатуры на последовательный порт через опцию в конфигурации. Поэтому машина вполне может работать через последовательный порт. Однако результаты могут измениться -- когда вы используете эту фишу, некоторые BIOSы могут мешать загрузчику видеть последовательный порт, и, таким образом, ядру не будут указывать на его использование. Некоторые BIOSы имеют опцию "Continue Console Redirection after POST" (Power On Self Test), она должна быть выключена, чтобы загрузчик и ядро могли использовать их собственную консоль. К сожалению, эта фиша не

универсальна.

PC-совместимые компьютеры не создавались для загрузки с последовательной консоли, в отличие от некоторых других платформ. Даже те системы, которые поддерживают последовательную консоль, обычно имеют это как опцию в конфигурации BIOS -- и если информация о конфигурации повреждена, вы получите систему, ищущую стандартный монитор и клавиатуру снова и снова. Вы вообще должны иметь какой-нибудь способ для получения монитора и клавиатуры к вашим amd64 и i386 системам в критической ситуации.

Вам необходимо редактировать */etc/ttys* как это рассматривалось выше.

SPARC и UltraSPARC

Эти машины поддерживают возможность полного управления с serial console. Просто отключите клавиатуру.

Замечания по SPARC и UltraSPARC

Последовательные порты на SPARC помечаются как *ttya*, *ttyb*, и т. д.

В отличие от некоторых других платформ, вовсе необязательно модифицировать файл */etc/ttys*, чтобы serial console заработала.

Системы SPARC/UltraSPARC понимают сигнал BREAK как команду STOP -A, что останавливает все приложения и операционную систему. Это удобно, когда это действительно нужно, но, к сожалению, некоторые serial terminal'ы при определенных обстоятельствах посылают сигналы, которые компьютер понимает как сигнал прерывания операции и останавливает машину. Пожалуйста, протестируйте, прежде чем использовать.

Если клавиатура и монитор подключены к машине, вы можете заставить использовать вместо них serial console с помощью следующих команд:

```
ok prompt:
ok setenv input-device ttya
ok setenv output-device ttya
ok reset
```

Если клавиатура и монитор (ttyC0) прописаны в */etc/ttys* (выше), вы можете использовать их в X.

MacPPC

Машины MacPPC настроены для использования serial console с помощью OpenFirmware.

Используйте следующие команды:

```
ok setenv output-device scca
ok setenv input-device scca
ok reset-all
```

Установите скорость передачи информации на 57600bps, 8N1.

Заметки по MacPPC:

К сожалению, использование serial console на всех MacPPC невозможно. Хотя большинство этих машин действительно имеют последовательные устройства, они недоступны для использования извне. К счастью, некоторые компании предложили подключаемые устройства для нескольких моделей Macintosh, которые позволяют использовать этот порт для serial console или для других целей. Поищите в вашем любимом поисковике информацию об этом, используя ключевую фразу "Macintosh internal serial port".

Вам придется включить устройство tty00 в /etc/ttys и установить его скорость в 57600, вместо стандартных 9600, в однопользовательском режиме, прежде чем произойдет вход в многопользовательский режим, как это описывалось выше. В этом случае serial console будет работать нормально.

Mac68k

Включение консоли на последовательном порту производится выбором в программе загрузки элемента меню "Options" -> "Serial Ports". Проверьте кнопку "Serial Console", затем выберите порт модема или принтера. Для того чтобы подключиться к последовательным портам Mac, вам нужен кабель от модема или принтера Macintosh. Чтобы сделанные вами изменения применялись при каждом запуске, сохраните текущие настройки в программе загрузки.

Замечания по Mac68k:

Порт модема -- *tty00*, а порт принтера *tty01*.

Mac68k не включает свой последовательный порт до тех пор, пока ему на это явно не "укажут". Поэтому, вы можете не видеть передаваемые последовательным портом сигналы до тех пор, пока не начнется процесс загрузки OpenBSD.

Вам придется включить порт (*tty00* или *tty01*), указанный выше.

7.8 - Как мне очистить/отключить экран в консольном режиме? (wscons)

Если вы хотите, чтобы экран отключался после определенного периода неактивности консоли (без использования X), можете установить следующие переменные [wscons\(4\)](#):

Включение **display.vblank** отключает вертикальное обновление экрана, что многие мониторы воспринимают как переход в режим "экономии энергии". После установления этой опции, экрану может потребоваться значительное время для возврата в обычный режим, но уменьшит потребление энергии и

увеличит производительность. Когда опция отключена, экран станет пустым, но монитор будет продолжать горизонтальное и вертикальное обновление экрана. Таким образом, экран восстановится мгновенно.

Переменная **display.screen_off** определяет время в тысячных долях секунды, через которое очищается экран. Например, значение 60000 установит таймаут в одну минуту.

display.kbdact указывает на то, стоит ли восстанавливать экран в случае получения активности с клавиатуры. Желательно включить эту опцию.

display.outact определяет, будет ли восстанавливаться экран пустым.

Вы можете установить эти переменные в командной строке, используя [wsconsctl\(8\)](#):

```
# wsconsctl -w display.screen_off=60000
display.screen_off -> 60000
```

или установить их перманентно, отредактировав [/etc/wsconsctl.conf](#) и изменения вступят в силу после перезагрузки:

```
display.vblank=on           # enable vertical sync blank
display.screen_off=600000   # set screen blank timeout to 10 minutes
display.kbdact=on          # Restore screen on keyboard input
display.outact=off         # Restore screen on display output
```

Очищение/отключение экрана включается, если одна из двух переменных -- `display.kbdact` или `display.outact` -- установлена на "on".

7.9 - ВСЕ ЧТО Я ВВОЖУ ПРИ ВХОДЕ В СИСТЕМУ ОТОБРАЖАЕТСЯ В ВЕРХНЕМ РЕГИСТРЕ!

Это не баг, а фишка.

Фактически, все Unix-команды и имена пользователей вводятся строчными буквами. Однако некоторые очень старые терминалы могли писать только заглавными, что делало невозможным их использование с Unix. Если ввести имя пользователя полностью заглавными символами, [getty\(8\)](#) предположит, что вы используете один из подобных старых терминалов и интерпретирует все, введенное вами, в виде строчных букв, хотя и будет показывать на экране прописные символы. Если вы используете пароль в разном регистре или он содержит только прописные буквы, то это сделает невозможным вход в систему.

Нажав комбинацию клавиш CTRL+D при получении приглашения для входа в систему, вы прервете работу [getty\(8\)](#), а [init\(8\)](#) запустит новый процесс, который будет правильно понимать как прописные, так и строчные буквы.

Перевод соответствует \$OpenBSD: faq7.html,v 1.78 2008/03/01 11:31:08 steven Exp \$