

Как собрать пакет WHDD Mini-HOWTO

Shroom [OmskLUG]

14 декабря 2012 г.

Аннотация

В статье рассказывается, как быстро собрать пакет WHDD для Debian-based дистрибутивов GNU/Linux.

1 Что такое WHDD?

Собственно, эту информацию можно найти в README. Буквально там говорится следующее. «WHDD — это утилита для диагностики и восстановления блочных устройств (практически аналог MHDD для Linux)». Функционал пока весьма скромнен, однако вполне полезен. WHDD умеет распознавать подключенные блочные устройства, может выдать информацию о параметрах S.M.A.R.T., а также может протестировать диск в двух режимах: «только чтение» (данные остаются на месте) и «запись нулей» (соответственно, все данные при этом будут потеряны). Естественно, эта утилита умеет и отображать результаты тестирования, при этом интерфейс практически один к одному повторяет таковой у MHDD.

2 Где найти?

На момент написания этого Mini-HowTo (декабрь 2012 года) ни в одном дистрибутиве, кроме Gentoo, пакета WHDD не было. Если верить статье [WHDD как аналог MHDD под GNU/Linux \(http://syslinux.ru/node/1364\)](http://syslinux.ru/node/1364), в Gentoo «whdd с марта 2012 года находится в основном дереве портежей, и максимум, что нужно сделать — это размаскировать этот пакет». Поэтому для Debian придётся скачать исходники из [репозитория на github \(https://github.com/krieger-od/whdd/](https://github.com/krieger-od/whdd/) или сразу получить [архив с последней версией исходников](https://github.com/krieger-od/whdd/archive/master.zip) отсюда: <https://github.com/krieger-od/whdd/archive/master.zip>) и собрать программу на своей машине. Ничего страшного и сложного в этом нет; сейчас объясню, как это сделать.

3 Что понадобится?

0. Естественно, прежде всего для сборки нужны исходники. Предполагаем, что они уже скачаны. Затем их нужно куда-то распаковать. Пусть, например, это будет каталог *zoo* в домашнем каталоге пользователя. Поскольку исходники находятся в zip-архиве, распаковать их в *zoo*, я надеюсь, не составит никакого труда (иначе вся информация, изложенная здесь будет для вас абсолютно бесполезной).
1. Библиотеки для работы программы:
 - 1). `libmenuw.so` и `libncursesw.so` находятся в пакете **libncursesw5** («w» — это wide-character, то есть поддержка многобайтных символов);
 - 2). `libtinfo.so` находится в пакете **libtinfo5**;
 - 3). `libc.so`, `libm.so`, `libdl.so`, `librt.so`, `libpthread.so` являются частью пакета **libc6**, который ставится при установке, поскольку без него система просто не сможет работать. Поэтому о них можно не беспокоиться.
2. Пакеты для сборки:
 - 1). `libncursesw5-dev`;
 - 2). `libtinfo-dev`;
 - 3). `dialog`;
 - 4). `libncurses5-dev` (обратите внимание — это пакет заголовков для `libncurses` без поддержки многобайтных символов, без него не находится `unctrl.h`, на который ссылается `dialog.h`);
 - 5). `libc6-dev`[†] (если вы до этого момента никогда и ничего не собирали на своей машине, вам нужно его поставить; если же собирали, скорее всего, он у вас уже есть).
3. Инструменты для сборки:

gcc[†] Компилятор C от GNU, если кто не в курсах. По умолчанию обычно не ставится, поэтому, если вы не разработчик, скорее всего, придётся поставить и его.

make[†] Система для автоматического разрешения зависимостей и оптимизации процесса сборки. Полезна не только для программистов, поскольку изначально предназначена для описания процесса создания неких конечных файлов из некоторого количества исходных. Основная особенность — процессы обработки запускаются только для тех файлов, которые были изменены после последней сборки.

cmake Кросс-платформенный генератор make-файлов. При установке пакета тянет за собой **emacsen-common** (опосредованно, через зависимость от **cmake-data**), так что не пугайтесь.

checkinstall Весьма полезная утилита, которая позволяет отслеживать файлы и каталоги, которые получаются в процессе сборки и установки софта из исходников (с помощью `make && make install`), и делать из них полноценный `.deb`-пакет. Собственно для процесса сборки не нужна, поэтому её можно не ставить. Однако я настоятельно рекомендую использовать именно её вместо традиционного `make install`, поскольку установленный пакет вычистить из системы значительно проще, нежели рыскать по дереву каталогов в поисках хвостов установленной когда-то программы, особенно если исходников уже давно нет и `make uninstall` сделать просто невозможно.

[†]Если вы время от времени намерены собирать что-нибудь из исходников, вероятно, имеет смысл поставить пакет **build-essential**, в зависимостях которого указаны и `libc6-dev`, и `gcc`, и `make`.

Итак, резюмируя вышесказанное одной командой,

```
# apt-get install build-essential libncursesw5-dev \  
libtinfo-dev libncurses5-dev dialog libc6-dev cmake \  
checkinstall libncursesw5 libtinfo5
```

Обратите внимание на символ «\» (бэкслэш) на концах строк: здесь он приведён, чтобы показать, что вторая и третья строка являются продолжением первой. Никакой другой цели, кроме эстетической (избежать слишком длинной строки), в его использовании не было. В настоящем терминале можно обойтись и без него (записать команду со всеми параметрами просто в одну строчку), хотя там он выполняет ту же функцию — говорит оболочке о том, что строка не закончилась и будет продолжена дальше.

4 Как собрать?

Для сборки проекта необходимо перейти в каталог с исходными текстами и выполнить всего две команды:

```
$ cmake . && make
```

Обратите внимание на то, что эти две команды записаны в одной строке, причём оператор `&&` гарантирует выполнение второй команды только после успешного завершения первой. **Точка** после `cmake` важна, так как задаёт рабочий каталог, в котором находятся параметры для `cmake` и в котором будет сгенерирован make-файл. Также можно запустить сборочный скрипт, который находится в этом же каталоге с исходниками:

```
$ ./build.sh
```

Но по сути этот скрипт запускает те же две команды сборки и предлагает выполнить `make install` для установки.

Если все нужные пакеты были установлены на предыдущем шаге, то сборка пройдёт успешно. Если что-то пошло не так, следует выяснить, что является источником ошибки. Скорее всего, проблема вызвана неудовлетворённой зависимостью от каких-либо заголовочных файлов или библиотек и решается установкой соответствующих пакетов.

5 Как установить?

Если вы выбрали путь наименьшего сопротивления и согласились выполнить `make install` (из-под административного аккаунта, то есть, от имени `root`'а), можете пропустить этот раздел и сразу запустить `whdd`:

```
# whdd-cli
```

или

```
# whdd-curses
```

Если же вы являетесь адептом строгого и рационального *Debian-way*, то сначала вы соберёте `.deb`-пакет, а затем установите его, и ваша система останется прозрачной, логичной и управляемой.

Итак, `checkinstall`... Как вы, вероятно, заметили, сборка программы идёт с правами обычного пользователя, поскольку процесс происходит в пользовательском каталоге. То же самое касается и сборки пакета. Установка же предполагает запись в системные директории, что потребует наличия прав администратора. Сам по себе `checkinstall`, запущенный без параметров, соберёт пакет и попытается его установить. Таким образом, если вы хотите собрать пакет и сразу же установить его (автоматически), вы можете сделать это, выполнив

```
# checkinstall
```

с правами администратора. Если же вы хотите только собрать пакет без непосредственной установки, вы можете задать параметр `--install=no`, выполнив

```
$ checkinstall --install=no
```

с правами обычного пользователя.

После запуска вам предложат создать описание пакета. Оно может содержать более одной строки, но если вы не собираетесь ни с кем делиться, можете оставить его пустым. После этого вас попросят принять как есть или изменить по своему усмотрению несколько параметров:

Этот пакет был создан с использованием данных значений:

```
0 - Maintainer: [ root@localhost ]
1 - Summary: [ ]
2 - Name: [ kriegler-od-whdd-c8a8691 ]
3 - Version: [ c8a8691 ]
4 - Release: [ 1 ]
5 - License: [ GPL ]
```

6 - Group: [checkinstall]
7 - Architecture: [amd64]
8 - Source location: [kriegler-od-whdd-c8a8691]
9 - Alternate source location: []
10 - Requires: []
11 - Provides: []
12 - Conflicts: []
13 - Replaces: []

Введите номер для изменения параметра
или нажмите ВВОД для продолжения:

Опять же, если вы не собираетесь распространять этот пакет, можете только изменить на что-то вменяемое имя пакета (2 - Name), версию (3 - Version) и предоставляемые пакеты (11 - Provides). Впрочем, последнее необязательно. Имя должно отражать реальное название соф-та, версия должна быть адекватной для того, чтобы можно было легко апгрейдиться после выхода новых и/или официальных версий (для whdd последняя версия на момент написания этого текста была 1.1), поле предоставляемых пакетов сообщает пакетному менеджеру информацию для разрешения зависимостей.

Если хочется всё сделать правильно, тогда очень желательно дополнительно изменить поля «Ответственный за поддержку» (0 - Maintainer, обычно пишется имя, а затем адрес электронной почты в угловых скобках), «Краткое описание» (1 - Summary, сюда можно вставить первое предложение из README), «Раздел» (6 - Group, ветка дерева пакетов, в которую попадёт этот пакет; для whdd можно выставить util или admin), «Зависимости» (10 - Requires, здесь указываются пакеты, без наличия которых в системе приложение не запустится, и их версии). Вот что получилось в результате у меня (версии библиотек указаны для Debian 7 Wheezy/testing):

Этот пакет был создан с использованием данных значений:

0 - Maintainer: [Philaret Nikonov <phil@hyperion.cc>]
1 - Summary: [WHDD is a diagnostic and recovery tool for
block devices (near to replace MHDD for Linux).]
2 - Name: [whdd]
3 - Version: [1.1-git20121213-c8a8691]
4 - Release: [1]
5 - License: [GPL]
6 - Group: [util]
7 - Architecture: [amd64]
8 - Source location: [kriegler-od-whdd-c8a8691]

```
9 - Alternate source location: [ ]
10 - Requires: [ libc6 (= 2.13-37), libncursesw5 (>=5.7),
    libtinfo5 (>=5.7) ]
11 - Provides: [ whdd ]
12 - Conflicts: [ ]
13 - Replaces: [ ]
```

Введите номер для изменения параметра
или нажмите ВВОД для продолжения:

Installing with make install...

В сущности, это всё. После нажатия <Enter> `checkinstall` соберёт пакет и (если ему этого не запретили явно) попытается его установить. После того, как пакет был успешно установлен, дерево исходников можно с лёгким сердцем удалять до следующего релиза.

6 Кого спросить?

Итак, с вопросом «что делать?» мы (надеюсь) разобрались, остался вопрос «кто виноват?» То есть, «кому задавать вопросы, если всё пошло не так?»

Во-первых, конечно же, авторам программы WHDD. Их координаты можно найти в файле `README`, который лежит в каталоге `doc` дерева исходных текстов, или в указанном [выше репозитории на github](#).

Во-вторых, автору этого текста. Его можно найти в джаббер-конференции омских линуксоидов (`omsklug@conference.jabber.ru`) под ником Shroom.

Ну и, наконец, всем участникам вашего местного (или ближайшего к вам) LUG'а и вообще любым первым встречным линуксоидам.

Спасибо за внимание!