

Сборка TauDEM в Linux

[Обсудить в форуме](#) Комментариев — 3

Эта страница опубликована в основном списке статей сайта по адресу <http://gis-lab.info/qa/taudem-linux-compile.html>

В статье поэтапно описывается процесс сборки из исходного кода набора программ TauDEM в Unix-подобных операционных системах.

Содержание

- [1 Кратко о TauDEM](#)
- [2 Подготовка к сборке](#)
 - [2.1 Ubuntu 9.10–11.04](#)
 - [2.2 Debian 5–6](#)
 - [2.3 Fedora 15](#)
 - [2.4 OpenSUSE 11.4](#)
- [3 Сборка TauDEM](#)
- [4 Тестирование TauDEM](#)
- [5 Ссылки по теме](#)
 - [5.1 Дополнительные материалы](#)

Кратко о TauDEM

[TauDEM](#) (Terrain Analysis Using Digital Elevation Models) — набор программ для геоморфологического и гидрологического анализа цифровых моделей рельефа. Автор — Дэвид Тарботон (David G. Tarboton), Исследовательская группа в области гидрологии, Университет шт. Юта. Программы написаны на языке программирования C++, распространяются под открытой лицензией (GPL), имеют консольный интерфейс. Входной и выходной растровый формат — GeoTIFF. Текущая версия на момент написания этой заметки — 5.0.5 (от 12.06.2011)

На официальном сайте [доступны](#) скомпилированные утилиты для Windows (32/64-bit), а также дополнение к ArcGIS 9.* в виде стандартного "тулбокса". Начиная с версии 5 все программы в составе TauDEM адаптированы для [параллельных вычислений](#), для обеспечения [многопоточности](#) по умолчанию используется открытая библиотека [MPICH2](#). При использовании такого подхода даже на однопроцессорных машинах задача всё равно выполняется в виде нескольких параллельных процессов.

Нужно отметить, что только в текущей версии TauDEM появились т.н. make-файлы для системы сборки программ [make](#) в Unix-системах.

Подготовка к сборке

На официальной странице TauDEM говорится, что для сборки программ и их нормальной работы должна быть обязательно установлена библиотека MPICH2. Экспериментально было выяснено, что другая, более распространённая в среде Linux свободная реализация MPI, — [OpenMPI](#) — также подходит для этой задачи. В этом случае будет использоваться именно OpenMPI, хотя основные дистрибутивы Linux содержат в своих репозиториях и MPICH2.

Итак, для успешной компиляции нам понадобятся программы:

- **make** — традиционная для Linux система сборки;
- **cmake** — более "продвинутая" система сборки;
- **gcc** — набор компиляторов, в том числе, для языка C++;
- **openmpi** — исполняемые файлы OpenMPI и набор файлов для разработчиков.

Внимание: все дальнейшие действия будут выполняться в командной строке и в основном с правами суперпользователя!

Все тестируемые системы — официальные LiveCD дистрибутивов.

Ubuntu 9.10–11.04

В пакетном менеджере должны быть подключены репозитории **universe** (установка через *Synaptic* или в консоли с помощью *apt-get* / *aptitude*)

- Открываем терминал
- Выполняем команды от суперпользователя (su / sudo):
 - sudo apt-get update
 - sudo apt-get install cmake openmpi-bin libopenmpi-devC учётом обновлений и зависимостей нужно будет загрузить 10-11Mb

Debian 5–6

В пакетном менеджере должны быть подключены репозитории **universe** (установка через *Synaptic* или в консоли с помощью *apt-get* / *aptitude*)

- Открываем терминал
- Выполняем команды от суперпользователя (su / sudo):
 - sudo aptitude update
 - sudo aptitude install gcc make cmake openmpi-bin libopenmpi-devC учётом обновлений и зависимостей нужно будет загрузить около 42Mb

Fedora 15

- Открываем терминал
- Выполняем команды от суперпользователя (su / sudo):
 - yum install gcc.i686 gcc-c++.i686 cmake openmpi.i686 openmpi-devel.i686 C учётом обновлений и зависимостей нужно будет загрузить около 60Mb.
 - updatedb Обновляем файловую базу данных
- Далее выполняем команды от обычного пользователя: export PATH="\$PATH:/usr/lib/openmpi/bin" Добавляем директорию с openmpi в системный PATH; рекомендуется соответственно отредактировать PATH в файле ~/.bashrc

OpenSUSE 11.4

В пакетном менеджере должен быть подключён репозиторий **Contrib** (установка через *YaST* или в командной строке с помощью *zypper*):

- Открываем терминал
- Выполняем команды от суперпользователя (su / sudo):
 - sudo zypper update
 - sudo zypper install gcc gcc-c++ make cmake openmpi openmpi-develC учётом обновлений и зависимостей нужно будет загрузить около 60Mb.

В OpenSUSE исполняемые файлы OpenMPI устанавливаются по умолчанию в директорию **/usr/lib/mpi/gcc/openmpi/bin**, её надо добавить в системный PATH. Также нужно добавить путь к разделяемым библиотекам OpenMPI в LD_LIBRARY_PATH, выполнив команду:

```
export LD_LIBRARY_PATH="$LD_LIBRARY_PATH:/usr/lib/mpi/gcc/openmpi/lib/"
```

Для дальнейшего использования OpenMPI желательно отредактировать строки PATH и

LD_LIBRARY_PATH в файле ~/.bashrc.

При использовании MPICH2 вместо OpenMPI для начала её надо установить средствами пакетного менеджера вашего дистрибутива или из исходного кода, затем нужно запустить "демон" **mpd** или **smpd**. Как это сделать, яснее всего скажет нам "подсказка" из командной строки:

```
A file named .mpd.conf file must be present in the user's home
directory (/etc/mpd.conf if root) with read and write access
only for the user, and must contain at least a line with:
MPD_SECRETWORD=
One way to safely create this file is to do the following:
  cd $HOME
  touch .mpd.conf
  chmod 600 .mpd.conf
and then use an editor to insert a line like
  MPD_SECRETWORD=mr45-j9z
into the file. (Of course use some other secret word than mr45-j9z.)
```

Сборка TauDEM

Для автоматизации процесса сборки и установки TauDEM в различных дистрибутивах Linux был написан небольшой [shell-скрипт](#). Скрипт делает следующее: проверяет, все ли необходимые компоненты установлены; скачивает архив с исходным кодом (около 260К); исправляет ошибки в нескольких файлах для сборки; запускает конфигурирование и компиляцию с помощью *stake*; устанавливает скомпилированные утилиты в выбранный каталог и переименовывает их в соответствии с Windows-версиями.

Работа скрипта была проверена на всех упомянутых выше дистрибутивах Linux.

В общем виде скрипт работает так.

- Сохраните скрипт на диск, дайте файлу права на исполнение: `chmod +x taudem_linux_compile.sh`
- При запуске без аргументов скрипт создаст в текущей директории каталог *TauDEM* и далее работает в этом каталоге, как описано выше. Исполняемые файлы будут установлены в *./TauDEM/bin*
- Можно запустить скрипт с аргументом в виде пути (если каталог существует, то скрипт будет работать в нём, если нет — создаст его). `taudem_linux_compile.sh ~/bin/TauDEM`

Желающим что-то поменять, улучшить и т.д. рекомендуется заглянуть в сам скрипт и его "под себя" :)

Тестирование TauDEM

Если сборка прошла успешно, настало время потестировать TauDEM.

Для этого можно использовать вспомогательный [скрипт](#) с выборочными командами из официального [руководства](#) пользователя. Нужно просто скопировать скрипт в каталог с исполняемыми файлами TauDEM, дать права на исполнение и запустить. Скрипт сам скачает пробный [набор данных](#) (~6М) и обработает их. При успешном завершении работы в каталоге *Logan* можно найти несколько растров *.tif и shp-файлов, в самом скрипте посмотреть использованные команды, а в руководстве почитать, что же всё это значит :)

Ссылки по теме

- [TauDEM installation](#)
- [TauDEM5 CommandLine Guide](#)
- [MPICH2](#)
- [OpenMPI](#)

Дополнительные материалы

- [Скрипт для сборки TauDEM в Linux](#)
- [Скрипт для тестирования TauDEM](#)

[Обсудить в форуме](#) Комментариев — 3

Последнее обновление: 2014-05-15 00:08

Дата создания: 20.06.2011

Автор(ы): [Александр Мурый \(amuriy\)](#)