



GDAL
(*GEOSPATIAL DATA*
***ABSTRACTION LIBRARY*)**

- ▣ GDAL -библиотека для чтения и записи растровых гео-пространственных форматов данных, выпускаемая под Open Source лицензией X/MIT организацией Open Source Geospatial Foundation(OSGeo).
- ▣ GDAL является проектом со проектом со свободным исходным кодом
- ▣ Первая версия была выпущена 4 мая 2006 года
- ▣ Языком написания является C++

- ▣ Некоторые программы используют библиотеки GDAL/OGR, чтобы читать и записывать различные ГИС-форматы:

Quantum GIS — свободная кроссплатформенная геоинформационная система
GRASS — свободная кроссплатформенная геоинформационная система с мощным интерфейсом командной строки.

GvSIG — свободная кроссплатформенная геоинформационная система, написанная на Java.

MapServer — открытая среда разработки геопространственных Интернет-приложений.

Google Earth — проект компании Google, в рамках которого в сети Интернет были размещены спутниковые фотографии всей земной поверхности.

R — язык программирования и среда для статистического анализа. Существуют расширения для анализа гео-пространственных данных.

ArcGIS — семейство программных продуктов американской компании ESRI. Начиная с версии 9.2 может использовать GDAL для работы с некоторыми форматами файлов:

- Библиотека GDAL состоит из 3-х основных частей:
CPL - базовые функции работы с файловой системой, строками, xml и т.п.
OGR - работа с векторными форматами.
GDAL - работа с растровыми форматами.
- OGR и GDAL обеспечивают не только поддержку форматов, но и операции работы с векторами и растрами. Т.е. GDAL позволяет не только считывать но и манипулировать с геоданными.
- Компонент CPL (исходники лежат в подпапке port) зависит от библиотек: zlib, curl.
OGR зависит от многих библиотек, но базовый набор от exrat, geos и proj.
GDAL зависит от tiff, jpeg, png.
- В зависимости от необходимости можно собрать GDAL именно в той конфигурации, что нужна (с определенным перечнем драйверов). Собирать будем пошагово: вначале все зависимые библиотеки, затем сам GDAL. Для простоты предлагается делать это все с использованием системы сборки CMake. Для большинства библиотек уже имеются скрипты CMake и сборка становится однотипной и простой. Заодно на выходе будут привычные проекты Visual Studio, где можно зайти в свойства и посмотреть, что и как.

Трансформирование снимков в системе GDAL производится значительно быстрее, чем во многих платных продуктах. К примеру, трансформация снимка ASTER в GDAL занимает 2-3 минуты, в то время как в ArcGIS Desktop – около получаса. GDAL корректно производит пересчет между системами координат использующими эллипсоиды и системами координат на сфере (обычная проблема, с которой сталкиваются пользователи ERDAS Imagine, который данную операцию производит не корректно).

Стандартная команда трансформации с помощью GDAL выглядит следующим образом:

```
gdalwarp -s_srs "+proj=longlat +datum=WGS84  
+no_defs" -t_srs "+proj=laea +lat_0=90 +lon_0=100  
+ellps=sphere +x_0=0 +y_0=0 +units=m +no_defs" -tr  
15 15 -rn -srcnodata 0 -dstnodata 0  
images\XXXXXX.jpg XXXXXX.tif
```

Стандартная команда трансформации с помощью GDAL выглядит следующим образом:

```
gdalwarp -s_srs "+proj=longlat +datum=WGS84 +no_defs" -t_srs  
"+proj=laea +lat_0=90 +lon_0=100 +ellps=sphere +x_0=0 +y_0=0  
+units=m +no_defs" -tr 15 15 -rn -srcnodata 0 -dstnodata 0  
images\XXXXXX.jpg XXXXXX.tif
```

gdalwarp – вызов программы трансформации

-s_srs "+proj=longlat +datum=WGS84 +no_defs" – описание исходной проекции/ системы координат снимков

-t_srs "+proj=laea +lat_0=90 +lon_0=100 +ellps=sphere +x_0=0 +y_0=0 +units=m +no_defs" – описание выходной системы координат и проекции. В данном примере мы используем проекцию Lambert azimuthal equal area с центром 90 с.ш., 100 в.д., основанную на сфере, а не эллипсоиде. Для других проекций набор параметров может отличаться.

-tr 15 15 – размер выходного пикселя в метрах

-srcnodata 0 -dstnodata 0 – заливка нулями поля вокруг изображения.

images\XXXXXX.jpg – исходный снимок из папки images

XXXXXX.tif – выходной файл.

Пример: пакетное перепроектирование данных ASTER (Terralook)

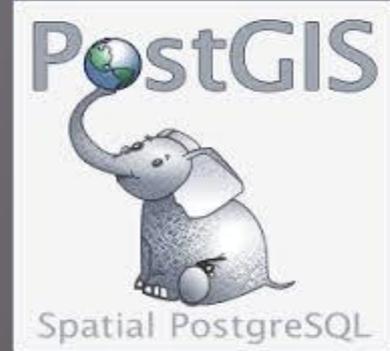
Эти данные поставляются в формате jpeg и имеют географическую привязку. Информация о привязке хранится в файле привязки jpgw. В комплект также входят текстовые файл с координатами углов для каждого снимка.

Для трансформации одного снимка из директории images скачанного набора данных TerraLook нужно, используя командную строку, выполнить следующее выражение (все пути даны относительно корневой папки распакованной коллекции):

```
gdalwarp -s_srs "+proj=longlat +datum=WGS84 +no_defs" -t_srs "+proj=laea +lat_0=90 +lon_0=100 +ellps=sphere +x_0=0 +y_0=0 +units=m +no_defs" -tr 15 15 -rn -srcnodata 0 -dstnodata 0 images\XXXXXX.jpg XXXXXX.tif
```

Для выполнения пакетного проецирования снимков ASTER используем прилагаемый скрипт на языке PERL (reproject.pl). Для выполнения трансформации достаточно скопировать скрипт в папку распакованных данных ASTER и запустить. Перед этим необходимо убедиться что все необходимые компоненты установлены (PERL, GDAL и PROJ.4). Скрипт создан для трансформации данных в проекцию Lambert azimuthal equal area.. При формировании имени выходного файла скрипт использует дату и время исходного снимка. Имя выходного файла будет выглядеть так: ГГГММЧЧ_ЧММСС.

POSTGIS



- PostGIS является расширением объектно-реляционной СУБД PostgreSQL, предназначенным для хранения в базе данных географической информации. PostGIS включает поддержку пространственных индексов R-Tree/GiST, а также функции обработки геоданных. PostGIS является свободным ПО с открытым исходным кодом.

- ▣ Первая версия была выпущена в 2001 году компанией Refrations Research под лицензией GNU General Public License. Стабильная версия «1.0» была выпущена 19 апреля 2005 года после 6-ти релиз-кандидатов. В 2006 году PostGIS была зарегистрирована OGC с формулировкой «реализует описанный стандарт» для «Simple Features для SQL».

- Использование связки PostgreSQL/PostGIS предоставляет довольно широкие возможности по работе с пространственными данными, однако широкого распространения этот союз до сих пор не получил. Одной из причин такого положения вещей можно считать практически полное отсутствие информации о PostGIS на русском языке.

Серия PostGIS 2+ обеспечивает:

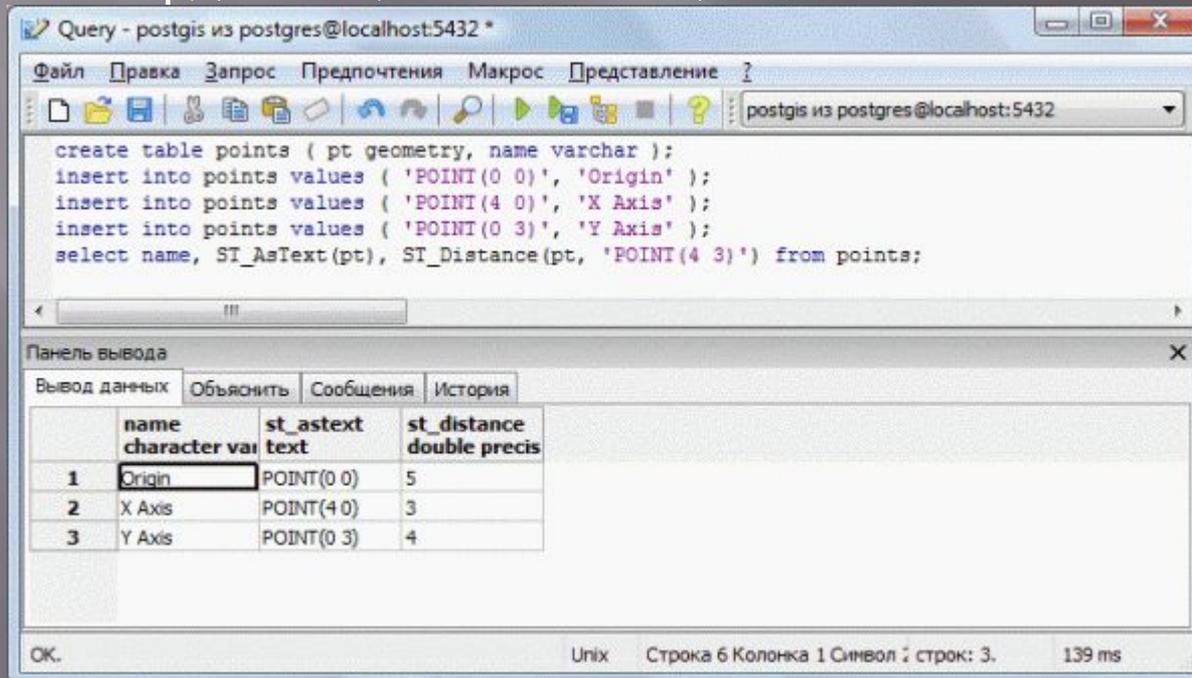
- ▣ Обработку векторных и растровых данных
- ▣ Пространственное перепроецирование SQL
- ▣ Поддержка импорта / экспорта шейпфайл векторных данных ESRI и поддержка большего количества форматов с помощью других инструментов с открытым исходным кодом третьих лиц
- ▣ командной строки для импорта растровых данных из многих стандартных форматов: GeoTIFF, NetCDF, PNG, JPG
- ▣ Предоставление и импорт вспомогательных функций векторных данных для стандартных текстовых форматов, таких как KML, GML, GeoJSON, GeoHash и WKT с помощью SQL
- ▣ Визуализация растровых данных в различных стандартных форматах GeoTIFF, PNG, JPG, NetCDF используя SQL
- ▣ Бесшовные растр/вектор запросы для выделения значений пикселей по геометрической области, рабочей статистики по регионам, вырезка растров по геометрии и векторизации растров
- ▣ Поддержку 3D объекта, пространственный индекс.

Таблицы метаданных OGC

- При создании пространственной базы данных автоматически создаются две таблицы метаданных - `SPATIAL_REF_SYS` и `GEOMETRY_COLUMNS`. Они создаются в соответствии со спецификацией «Open Geospatial Consortium Simple Features for SQL specification», выпущенной OGC и описывающей стандартные типы объектов ГИС, функции для манипуляции ими и набор таблиц метаданных.
- Таблица `GEOMETRY_COLUMNS` хранит информацию о таблицах базы данных, содержащих пространственную информацию. Её заполнение осуществляется вручную, либо как следствие выполнения специальной процедуры `OGC AddGeometryColumn()`.
- Таблица `SPATIAL_REF_SYS` содержит числовые идентификаторы и текстовые описания систем координат, используемых в пространственной базе данных. Одним из полей этой таблицы является поле `SRID` - уникальный идентификатор, однозначно определяющий систему координат. `SRID` представляет из себя числовой код, которому соответствует некоторая система координат. Например, распространенный код `EPSG 4326` соответствует географической системе координат `WGS84`.

Пространственные запросы

- В данном примере создали таблицу `points`, содержащую два поля: поле `pt` типа `geometry` и поле `name` типа `varchar`, после чего добавили (`insert`) в нее три записи, содержащих информацию о точках. Затем осуществили выборку с использованием функций PostGIS: `ST_Distance()` и `ST_AsText()`. В качестве параметра обе функции используют объект типа `geometry`. Функция `ST_Distance()` рассчитывает расстояние между двумя указанными точками плоскости, а `ST_AsText()` возвращает геометрию объекта в текстовом формате WKT (Well-Known Text). Формат WKT включает информацию о типе объекта и координаты, составляющие объект.



The screenshot shows a PostgreSQL query window titled "Query - postgres из postgres@localhost:5432". The query text is as follows:

```
create table points ( pt geometry, name varchar );
insert into points values ( 'POINT(0 0)', 'Origin' );
insert into points values ( 'POINT(4 0)', 'X Axis' );
insert into points values ( 'POINT(0 3)', 'Y Axis' );
select name, ST_AsText(pt), ST_Distance(pt, 'POINT(4 3)') from points;
```

The output panel, titled "Панель вывода", displays the following table:

	name character vari	st_astext text	st_distance double precis
1	Origin	POINT(0 0)	5
2	X Axis	POINT(4 0)	3
3	Y Axis	POINT(0 3)	4

At the bottom of the window, the status bar shows "Unix Строка 6 Колонка 1 Символ ; строк: 3. 139 ms".

Работа с shаре-файлами

- Для этого сначала преобразуем данные из shаре-файла в формат понятный PostgreSQL с помощью утилиты shp2pgsql, а затем загрузим их в базу данных. Данные хранятся в географической системе координат WGS84, для которой SRID имеет значение 4326. Загружается в базу данных файл bnd-political-boundary-a.shp. Составляется небольшой командный bat-файл, содержащий следующие строки:
SET PATH="C:\Program Files\PostgreSQL\8.3\bin"
shp2pgsql -i -D -s 4326 bnd-political-boundary-a.shp bnd-political-boundary-a > bnd-political-boundary-a.sql psql -U postgres -f bnd-political-boundary-a.sql -d postgis. Данный набор команд осуществляет конвертацию (shp2pgsql создает файл sql) и загрузку данных в БД (psql).
- Для того, чтобы добавить данные к существующей БД, можно использовать ключ -a, при этом схемы существующей БД и загружаемых данных должны совпадать.
- Далее вводится пароль для пользователя postgres, открывается база данных. В pgAdmin заходим в: «Базы/postgis/Схемы/public/Таблицы», щелкаем правой кнопкой мыши и в контекстном меню выбираем «Обновить», после чего в списке таблиц должна появиться загруженная нами таблица bnd-political-boundary-a.



Браузер объектов

- Сервера (1)
 - PostgreSQL Database Server 8.3 (localhost:5432)
 - Базы (4)
 - nonspatial
 - postgis
 - Каталоги (2)
 - Схемы (1)
 - public
 - Домены (2)
 - Функции (1426)
 - Последовательности (1)
 - Таблицы (4)
 - bnd-political-boundary-a
 - geometry_columns
 - points
 - spatial_ref_sys
 - Триггерные функции (10)
 - Виды (3)
 - Репликация (0)
 - postgres
 - template_postgis
 - Tablespaces (2)
 - Групповые роли (0)
 - Роли входа (1)

Свойства

Статистика

Зависимости

Зави

Таблица	Комментарий
bnd-political-bounda...	
geometry_columns	
points	
spatial_ref_sys	

SQL панель