

Оглавление

Общая информация (прочсть перед работой с инструкцией).....	2
1. Настройка удалённого рабочего стола.....	3
2. Установка Apache2.....	4
3. Установка MONO 4.6.2.....	5
3.1 Установка MonoDevelop.....	5
4. Установка с нуля nodejs и веб-сервера nodejs+express.....	6
5. Установка POSTGRES 9.6.....	8
6. Установка Oracle + sqlplus (32 bit).....	9
7. Установка Oracle + sqlplus (64 bit).....	10
8. Настройка автозапуска служб при старте системы.....	11
7.1 Настройка медиасервера.....	11
7.2 Настройка сервера приложений.....	11
7.3 Настройка сервера построения маршрутов.....	12
9. Установка red5 media server.....	13
10. Дисковое пространство LINUX.....	14
9.1 Подключение дополнительного диска.....	14
9.2 Отмонтирование диска.....	15
11. Отключение автоматических ежедневных обновлений в debian.....	17
12. Настройка HTTPS.....	18
12. Настройка сервера Apache для перенаправления запросов на интернет-ресурсы.....	20
13. Установка и настройка сайта: статический контент.....	23
4. Установка и настройка сайта: динамический контент.....	25
5. Распараллеливание сайтов TileTrees и WebSite.....	28
6. Настройка планировщика и SH-файлов.....	28
16.1 Общая информация, перечень SH-файлов.....	28
16.2 SH-файлы, которые конфигурируем перед запуском.....	31
16.2.1 SH-файлы, требующие настройки параметров БД.....	31
16.2.2 SH-файлы, требующие настройки параметров расчётов.....	31
16.2.3 SH-файлы, требующие настройки планировщика.....	31
7. Установка обновления приложения.....	34
8. Установка службы маршрутизации (полная).....	34
9. Установка службы маршрутизации(с подготовленным графом и исходниками).....	37
10. GDAL+настройка импорта/экспорта карт из административных задач через UTEService.....	38
11. Установка ghostScript.....	39
12. Установка curl и генерация файлов постраничности.....	39

Общая информация (прочитать перед работой с инструкцией)

Логин/пароль пользователя root для входа: **user/password**.

По параметрам безопасности редактирование/удаление файлов в системных папках разрешенно только под правами root.

Для работы с правами root в графическом интерфейсе в консоль вводим: *sudo nautilus*. Работать нужно в открывшемся окне. В противном случае опции по редактированию/удалению/вставке в системных папках будут неактивны.

Файлы открываем для редактирования в стандартном Text Editor.

Если требуется выполнить команду в консоли с правами root, то непосредственно перед командой вводим *sudo [команда]*. Затем вводим пароль во всплывающем окне.

Для настройки системы/сервера требуется **подключение к интернет**. Для настройки выполняем шаги

1. Заходим в System Settings -> Network -> Options. Далее во вкладку IPv4 Settings.
2. Устанавливаем Method: Manual.
3. Вводим параметры: Address – IP, Netmask – маска подсети, Gateway – Шлюз, DNS servers – сервер DNS. Нажимаем Save.
4. В окне Network переключаем выключатель в положение ON-OFF-ON, чтобы изменения вступили в силу.

Чтобы **подключиться к локальному диску** открываем Files -> Other Location -> и в окне connect to server указываем адрес с префиксом smb://, например: *smb://192.168.1.222*.
Примечание: для диска P в ГСИ при запросе пароля выбираем connect as Anonymous (без ввода логина и пароля).

При работе с **Apache**, при изменении параметров конфигурации, добавлению новых компонентов и т.д. для того чтобы изменения вступили в силу выполняем рестарт командой:
sudo systemctl restart apache2.service

1. Настройка удалённого рабочего стола

Варианты работы с системой:

А) через вкладку Console в VMware-клиенте. Для этого не требуется дополнительных настроек.

Б) через удалённый рабочий стол в Windows. Для этого требуется выполнить настройки в UBUNTU:

Б1) Установка необходимых утилит на UBUNTU. Последовательно выполняем команды в консоли:

```
sudo apt-get update
sudo apt-get install xrdp
sudo apt-get install xfce4
sudo apt-get install xfce4-terminal
sudo apt-get install gnome-icon-theme-full tango-icon-theme
echo xfce4-session > ~/.xsession
```

Б2) Изменяем файл /etc/xrdp/startwm.sh в текстовом редакторе.

Файл должен выглядеть так. (ВАЖНО! В КОНЦЕ ФАЙЛА ДОЛЖНА БЫТЬ ПУСТАЯ СТРОКА):

```
#!/bin/sh
if [ -r /etc/default/locale ]; then
. /etc/default/locale
export LANG LANGUAGE
fi
startxfce4
```

Б3) Перезапускаем сервис xrdp командой:

```
sudo service xrdp restart
```

Б4) Подключаемся к удалённому рабочему столу стандартными средствами Windows. Адрес IP UBUNTU можно узнать командой в консоли.

```
hostname -I
```

Логин/пароль для входа: user/password.


ВАЖНО! Проверка путей к репозиториям Debian

Необходимо убедиться, что пути к репозиториям Debian прописаны корректно

```
sudo nano /etc/apt/sources.list
```

Содержимое должно выглядеть так:

- закомментированные строки с доступом к cdrom
- основные пути – [debian.org](http://ftp.ru.debian.org/debian/) (не yandex или другие)



```
sources.list
#
# deb cdrom:[Debian GNU/Linux 9.3.0 _Stretch_ - Official i386 DVD Binary-1 20171209-13:03]/ stretch contrib main
#deb cdrom:[Debian GNU/Linux 9.3.0 _Stretch_ - Official i386 DVD Binary-1 20171209-13:03]/ stretch contrib main
deb http://ftp.ru.debian.org/debian/ stretch main
deb-src http://ftp.ru.debian.org/debian/ stretch main
deb http://security.debian.org/debian-security stretch/updates main contrib
deb-src http://security.debian.org/debian-security stretch/updates main contrib
# stretch-updates, previously known as 'volatile'
deb http://ftp.ru.debian.org/debian/ stretch-updates main contrib
deb-src http://ftp.ru.debian.org/debian/ stretch-updates main contrib
```

В случае, если при установке компонентов выдается сообщение «failed to start the dirmngr '/usr/bin/dirmngr': No such file or directory» - устанавливаем этот пакет

```
sudo apt install dirmngr
```

2. Установка Apache2

Устанавливаем модуль, добавляем в автозагрузку и запускаем сервер. Последовательно выполняем команды:

```
sudo apt-get install apache2
sudo systemctl enable apache2
sudo systemctl start apache2
```

3. Установка MONO 4.6.2

Выполняем команды:

```
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys
3FA7E0328081BFF6A14DA29AA6A19B38D3D831EF
echo "deb http://download.mono-project.com/repo/ubuntu wheezy/snapshots/4.6.2 main" |
sudo tee /etc/apt/sources.list.d/mono-official.listsudo apt-get update
sudo apt-get install aptitude
sudo aptitude install libapache2-mod-mono mono-apache-server4
sudo apt-get install mono-complete
```

Если возникает ошибка, переустанавливаем apache:

```
sudo apt-get remove apache2
sudo apt-get install apache2
```

Для корректной работы создаём папку, даём папке права командами:

```
sudo mkdir /etc/mono/registry
sudo chmod uog+rw /etc/mono/registry
```

Добавляем флаг перестарта mod-mono-server4 в файл конфигурации /etc/apache2/mods-enabled/mod_mono_auto.conf

```
sudo nano /etc/apache2/mods-enabled/mod_mono_auto.conf
```

Вписываем после строки “MonoAutoApplication enabled” 2 строки(перестарт mod-mono-server4 каждые 5 дней):

```
MonoAutoRestartMode Time
```

```
MonoAutoRestartTime 05:00:00:00
```

Сохраняем изменения ctrl+X => Y => Enter

Важно! Конфигурация применяется в группе с настройками запуска скрипта restart.sh в планировщике задач.

3.1 Установка MonoDevelop

В визуальной среде открываем окно «Центр приложений». Вписываем “MonoDevelop” и устанавливаем.

Установленная среда – индикатор того, что mono в системе установлена корректно.

4. Установка с нуля nodejs и веб-сервера nodejs+express

- 1) Удалить старый nodejs сервер, для этого в putty выполнить команды
Если в результате выполнения команд появляются ошибки – **игнорировать**.
 - 1.1 Удалить папку /home/websys53/gis_web77/nodejs-server

```
rm -rf /home/websys53/gis_web77/nodejs-server
```
 - 1.2 Удалить папку /home/websys53/nodejs

```
rm -rf /home/websys53/nodejs
```
 - 1.3 Выполнить команду

```
pm2 delete www  
pm2 save --force
```
- 2) в putty выполнить команду копирования библиотек libgdal

```
sudo cp /home/websys53/GIS_WEB60/UTEService/gdal-2.2.2/build/lib/libgdal.a /usr/local/lib  
sudo cp /home/websys53/GIS_WEB60/UTEService/gdal-2.2.2/build/lib/libgdal.la /usr/local/lib  
sudo cp /home/websys53/GIS_WEB60/UTEService/gdal-2.2.2/build/lib/libgdal.so /usr/local/lib  
sudo cp /home/websys53/GIS_WEB60/UTEService/gdal-2.2.2/build/lib/libgdal.so.20 /usr/local/lib  
sudo cp /home/websys53/GIS_WEB60/UTEService/gdal-2.2.2/build/lib/libgdal.so.20.3.1 /usr/local/lib  
  
sudo ldconfig
```
- 3) Убедиться, что в папке /usr/local/lib появились файлы libgdal.*(5 штук)

```
ls /usr/local/lib | grep libgdal
```
- 4) в putty выполнить команды для установки nodejs, предварительно выложить файлы nodejs/baseserver в папку /home/websys53/patch/
pm2-4.4.0.tgz –g в папку /website/Service/node/
node-v9.11.2-linux-x86.tar.xz в папку /website/Service/node/
Для удобства можно создать sh файл со всеми командами или выполнить их отдельно через putty.

```
#!/bin/bash  
WEB77_PATH="/home/websys53/gis_web77"  
echo "Начинаем установку nodejs"  
mkdir /home/websys53/nodejs  
cd /home/websys53/nodejs  
cp ${WEB77_PATH}/website/Service/node/node-v9.11.2-linux-x86.tar.xz /home/websys53/nodejs  
sudo mkdir /usr/local/lib/node  
sudo tar -xJvf node-v9.11.2-linux-x86.tar.xz  
sudo mv node-v9.11.2-linux-x86 nodejs  
sudo mv nodejs /usr/local/lib/node/nodejs  
sudo rm -f /home/websys53/nodejs  
echo "Выдаем права на папку /usr/local/lib/node/"  
sudo chmod -R 777 /usr/local/lib/node/  
echo "Добавляем пути nodejs в переменные среды NODEJS_HOME, PATH"  
echo 'export NODEJS_HOME=/usr/local/lib/node/nodejs' >> /home/websys53/.profile  
echo 'export PATH=$NODEJS_HOME/bin:$PATH' >> /home/websys53/.profile  
source /home/websys53/.profile  
echo "nodejs установлен."  
echo "Версия node "`node -v`  
echo "Версия npm "`npm -v`  
echo "Для детализации версии npm впишите в консоли npm version"
```

```
echo "Копируем серверную часть nodejs"
cp -r /home/websys53/patch/nodejs /home/websys53/gis_web77
echo "Начинаем установку PM2"
npm i ${WEB77_PATH}/website/Service/node/pm2-4.4.0.tgz -g
pm2 startup
```

После выполнения скрипта появится сообщение в консоли с предложением скопировать текст и выполнить его. Текст должен быть подобного плана: «sudo env PATH=\$PATH:/usr/local/lib/node/nodejs/bin /usr/lib/node/nodejs/lib/node_modules/pm2/bin/pm2 startup systemd -u websys53 --hp /home/websys53»

Скопировать текст правой кнопкой мыши, вставить его и выполнить.

Перезагрузить сервер

```
sudo reboot
```

- 5) В putty выполнить команды добавления веб-сервера nodejs в менеджер процессов PM2

5.1 старт веб-сервера

```
cd /home/websys53/gis_web77/nodejs/baseserver
```

```
pm2 start bin/www -i max --name "baseserver"
```

5.2 проверяем, что в списке присутствует несколько названий "baseserver".

```
pm2 list
```

5.3 сохраняем конфигурацию PM2

```
pm2 save --force
```

- 6) Отредактировать файл /etc/apache2/sites-available/000-default.conf.

```
sudo nano /etc/apache2/sites-available/000-default.conf
```

После строк:

```
ProxyPass /web77/tiletrees/ http://127.0.0.1/tiletrees/
```

```
ProxyPassReverse /web77/tiletrees/ http://127.0.0.1/tiletrees/
```

Добавить строки(проксирование запросов на web-сервер nodejs). timeout=28800 в секундах 8 часов – длительность рабочего дня:

```
ProxyPass /api/ http://127.0.0.1:3000/api/ timeout=28800
```

```
ProxyPassReverse /api/ http://127.0.0.1:3000/api/ timeout=28800
```

```
ProxyPass /web55/api/ http://127.0.0.1:3000/api/ timeout=28800
```

```
ProxyPassReverse /web55/api/ http://127.0.0.1:3000/api/ timeout=28800
```

```
ProxyPass /web77/api/ http://127.0.0.1:3000/api/ timeout=28800
```

```
ProxyPassReverse /web77/api/ http://127.0.0.1:3000/api/ timeout=28800
```

Сохранить ctrl+X. Далее нажать Y, Далее нажать enter

Перезагрузить сервер

```
sudo reboot
```

- 7) После перезагрузки убеждаемся, что nodejs веб-сервер запустился автоматически. Открываем путь:

<http://сервер:3000/api/log/ute-log> или <https://сервер:3000/api/log/ute-log> в зависимости от включенного протокола на сервере, и должны получить в браузере предложение скачать лог UTEService

5. Установка POSTGRES 9.6

1. Устанавливаем модуль. Последовательно выполняем команды:

```
sudo add-apt-repository "deb http://apt.postgresql.org/pub/repos/apt/ xenial-pgdg main"
wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -
sudo apt-get update
sudo apt-get install postgresql-9.6
```

2. Заходим в базу данных по умолчанию:

```
sudo -i -u postgres
```

3. Устанавливаем пароль для пользователя postgres

```
psql
ALTER ROLE postgres WITH PASSWORD 'gfhjkmubc';
ТОЧКА С ЗАПЯТОЙ В КОНЦЕ ОБЯЗАТЕЛЬНА!!!
```

4. С правами администратора открываем файл /etc/postgresql/9.6/main/postgresql.conf. Вносим правки: добавляем строку в теле CONNECTIONS AND AUTHENTICATION

```
listen_addresses = '*'
```

5. С правами администратора открываем файл /etc/postgresql/9.6/main/pg_hba.conf. Вносим правки: после строки #IPv4 local connections: добавляем строки с IP, которые смогут подключаться

host	all	all	192.168.1.231/32	md5
host	all	all	192.168.1.124/32	md5
host	all	all	192.168.1.31/32	md5

6. Перезапускаем postgres командой:

```
sudo /etc/init.d/postgresql restart
```

7. Проверяем подключение локально командой:

```
sudo -u postgres psql
```

запросит пароль, вводим пароль из пункта 3.

8. Настраиваем postgres, чтобы база была на отдельном диске:

8.1 останавливаем сервер командой

```
sudo service postgresql stop
```

8.2 перемещаем базу командой

```
sudo /var/lib/postgresql/9.6/main /home/websys53/postgres
```

8.3 создаем ссылку

```
sudo ln -s /home/websys53/postgres/main /var/lib/postgresql/9.6/main
```

8.4 запускаем сервис командой

```
sudo service postgresql start
```


6. Установка Oracle + sqlplus (32 bit)

1. Устанавливаем дополнительные модули. Требуется подключение к интернет. Последовательно выполняем команды:

```
sudo apt-get install libaio1
```

```
sudo apt-get install alien
```

2. Скачиваем с сайта Oracle www.oracle.com/technetwork/database/features/instant-client/index-097480.html необходимые пакеты в формате rpm под архитектуру (в нашем случае x86). Требуются пакеты basic и sqlplus (файлы *oracle-instantclient12.2-basic-12.2.0.1.0-1.i386.rpm* и *oracle-instantclient12.2-sqlplus-12.2.0.1.0-1.i386.rpm* лежат на диске *P:\distr\Oracle*). Сохраняем или копируем файлы на рабочий стол UBUNTU(для Debian кладем в корень /home/websys53).

3. В командной строке переделываем rpm пакеты в deb с помощью команды alien. Для этого последовательно выполняем команды:

```
sudo alien /home/websys53/Desktop/oracle-instantclient12.2-basic-12.2.0.1.0-1.i386.rpm
```

```
sudo alien /home/websys53/Desktop/oracle-instantclient12.2-sqlplus-12.2.0.1.0-1.i386.rpm
```

Где */home/websys53/Desktop/* - местонахождение файлов с расширением rpm(/home/websys53/ для Debian)

Файлы сгенерируются и будут лежать в папке /home/websys53/.

4. Устанавливаем полученные пакеты. В консоли выполняем команду

```
sudo dpkg -i oracle-instantclient12.2-basic_12.2.0.1.0-2_i386.deb oracle-instantclient12.2-sqlplus_12.2.0.1.0-2_i386.deb
```

При составлении команды следует обратить внимание, что после получения пакетов имена файлов изменились.

5. Экспортируем библиотеку, запустив следующие команды в консоли:

```
export LD_LIBRARY_PATH=/usr/lib/oracle/12.2/client/lib:$LD_LIBRARY_PATH
```

```
echo "/usr/lib/oracle/12.2/client/lib" |sudo tee /etc/ld.so.conf.d/oracle.conf
```

```
sudo ldconfig -v |grep oracle
```

Стоит обратить внимание, что, в зависимости от версии, путь /usr/lib/oracle/12.2/client/lib может меняться.

6. Создаём папку /etc/oracle. Копируем tns.names в папку /etc/oracle/.

7. Необходимо добавить в файл /etc/environment следующее содержимое:

```
TNS_ADMIN="/etc/oracle"
```

```
NLS_LANG="AMERICAN_RUSSIA.AL32UTF8"
```

8. Перезапускаем систему. Вход осуществляется из консоли командой:

```
sqlplus web_sys/gfhjkmubc@ИМЯ_БД
```

где

web_sys – логин

gfhjkmubc – пароль

7. Установка Oracle + sqlplus (64 bit)

1. Устанавливаем дополнительные модули. Требуется подключение к интернет. Последовательно выполняем команды:

```
sudo apt-get install libaio1
```

```
sudo apt-get install alien
```

2. Скачиваем с сайта Oracle www.oracle.com/technetwork/database/features/instant-client/index-097480.html необходимые пакеты в формате rpm под архитектуру (в нашем случае x64). Требуется пакеты basic и sqlplus (файлы *oracle-instantclient12.2-basic-12.2.0.1.0-1.x86_64.rpm* и *oracle-instantclient12.2-sqlplus-12.2.0.1.0-1.x86_64.rpm* лежат на диске *P:\distr\Oracle*). Сохраняем или копируем файлы на рабочий стол UBUNTU(для Debian кладем в корень /home/websys53).

3. В командной строке переделываем rpm пакеты в deb с помощью команды alien. Для этого последовательно выполняем команды:

```
sudo alien /home/websys53/Desktop/oracle-instantclient12.2-basic-12.2.0.1.0-1.x86_64.rpm
```

```
sudo alien /home/websys53/Desktop/oracle-instantclient12.2-sqlplus-12.2.0.1.0-1.x86_64.rpm
```

Где */home/websys53/Desktop/* - местонахождение файлов с расширением rpm(/home/websys53/ для Debian)

Файлы сгенерируются и будут лежать в папке /home/websys53/.

4. Устанавливаем полученные пакеты. В консоли выполняем команду

```
sudo dpkg -i oracle-instantclient12.2-basic_12.2.0.1.0-2_amd64.deb oracle-instantclient12.2-sqlplus_12.2.0.1.0-2_amd64.deb
```

При составлении команды следует обратить внимание, что после получения пакетов имена файлов изменились.

5. Экспортируем библиотеку, запустив следующие команды в консоли:

```
export LD_LIBRARY_PATH=/usr/lib/oracle/12.2/client64/lib:$LD_LIBRARY_PATH
```

```
echo "/usr/lib/oracle/12.2/client64/lib" |sudo tee /etc/ld.so.conf.d/oracle.conf
```

```
sudo ldconfig -v |grep oracle
```

Стоит обратить внимание, что, в зависимости от версии, путь /usr/lib/oracle/12.2/client/lib может меняться.

6. Создаём папку /etc/oracle. Копируем tns.names в папку /etc/oracle/.

7. Необходимо добавить в файл /etc/environment следующее содержимое:

```
TNS_ADMIN="/etc/oracle"
```

```
NLS_LANG="AMERICAN_RUSSIA.AL32UTF8"
```

8. Вход осуществляется из консоли командой:

```
sqlplus64 web_sys/gfhjkmubc@ИМЯ_БД
```

где

web_sys – логин

gfhjkmubc – пароль

8. Настройка автозапуска служб при старте системы

На серверах медиа, приложений и построения маршрутов должен быть настроен автозапуск служб при старте системы. Службы, которые должны запускаться при старте, перечислены ниже.

Сервер	Служба	Комментарий	Расположение
Медиа	RED5	Запускает медиа сервер	/home/websys53/Videos/red5/red5.sh
Приложений	Mapgen	Запускает Mapgen	/home/websys53/GIS_WEB60/MapGen/MRS_Service_switch_demon.sh
	UTEService	Запускаем UTEService	/home/websys53/GIS_WEB60/UTEService/Bin/UTEService_switch_demon.sh
Построение маршрутов	RoutingService	Запускает службу маршрутизации	osrm-routed /home/websys53/GIS_WEB60/RoutingService/RoadService/RU.osrm

7.1 Настройка медиасервера

Важно! Требуется доступ в интернет для скачивания пакета jsvc!

1. Через ssh putty отключаем вариант автозапуска медиасервера через rc.local(комментируем строки запуска):

```
sudo nano /etc/rc.local
```

2. Перезагружаем систему:

```
sudo reboot
```

3. Устанавливаем пакет jsvc для запуска red5, как демона :

```
sudo apt-get install jsvc
```

4. Копируем скрипт red5 в init.d директорию:

```
sudo cp /home/websys53/red5-server/red5 /etc/init.d
```

5. Редактируем скрипт:

```
sudo nano /etc/init.d/red5
```

В самом начале содержимого меняем в строке FILE_PATH="/usr/local/\$NAME" путь на "/home/websys53/red5-server"
сохраняем файл ctrl+X => Y => Enter

6. Даем права на выполнение скрипта:

```
sudo chmod +x /etc/init.d/red5
```

7. Добавляем скрипт старт медиасервера red5 в качестве сервиса в автозагрузку системы:

```
sudo update-rc.d red5 defaults
```

```
sudo update-rc.d red5 enable
```

8. Перезапускаем сервер:

```
sudo reboot
```

7.2 Настройка сервера приложений

1. В папке /home/websys53/GIS_WEB60/Tasks создаём файл service_autorun.sh

Заполняем файл содержимым:

```
#!/bin/sh
```

```
# version 6.1.2.0
```

```
sed -i 's/\r$//' /home/websys53/GIS_WEB60/UTEService/Bin/UTEService_switch_demon.sh
```

```
cd /home/websys53/GIS_WEB60/UTEService/Bin
```

```
bash UTEService_switch_demon.sh start
```

```
sed -i 's/\r$//' /home/websys53/GIS_WEB60/MapGen/MRS_Service_switch_demon.sh
```

```
cd /home/websys53/GIS_WEB60/MapGen
```

bash MRS_Service_switch_demon.sh start

2. Выставляем права на запуск скрипта командой:

```
chmod ugo+x /home/websys53/GIS_WEB60/Tasks/ service_autorun.sh
```

3. Входим в редактор файла автозапуска командой:

```
sudo nano /etc/rc.local
```

4. Внутри, перед строкой, "exit 0" пишем вызов sh скриптов под пользователем

```
cd /home/websys53/GIS_WEB60/Tasks
```

```
su websys53 -c 'bash service_autorun.sh'
```

5. Сохраняем файл, нажимаем: **ctrl+x**, затем **Y**, затем **enter**

6. Перезагружаем сервер. Убеждаемся, что автозапуск работает. В консоли выполняем команду:

```
gnome-system-monitor
```

Находим сервисы Main в списке. Нужный сервис запущен.

7.3 Настройка сервера построения маршрутов

1. Через ssh putty отключаем прошлый вариант автозапуска медиасервера:

```
sudo nano /etc/rc.local
```

2. Перезагружаем систему:

```
sudo reboot
```

3. Копируем файл osrm-routed.service из патча в директорию /home/websys53:

4. Копируем файл osrm-routed.service в /etc/systemd/system

```
sudo cp /home/websys53/osrm-routed.service /etc/systemd/system
```

5. Добавляем сервис маршрутизации в автозапуск системы:

```
sudo systemctl enable osrm-routed
```

6. Перезапускаем сервер:

```
sudo reboot
```

7. Проверяем работоспособность сервера(спустя 2-10 мин в зависимости от производительности сервера):

<http://192.168.1.128:8484/route/v1/driving/30.348887,60.010632;30.32459,59.873786?overview=false&steps=true>

9. Установка red5 media server

1. Скачиваем и устанавливаем последний релиз red5 с сайта <https://github.com/Red5/red5-server/releases>, например RED5 10.1.8 Release <https://github.com/Red5/red5-server/releases/download/v1.0.8-RELEASE/red5-server-1.0.8-RELEASE.tar.gz>
2. Архив представляет собой готовый сервер. Распаковываем, копируем содержимое в /home/websys53/videos/red5, где user – имя пользователя.
3. Дополнительно для работы нужны пакеты java. Для установки пакетов последовательно выполняем команды:

```
a) sudo apt-get update
б) sudo apt-get install java-package
в) sudo apt-get install ant
г) sudo apt-get install subversion
```

Комментарий: ANT: это библиотека Java и средство командной строки, задача которого — управлять процессами, описанными в файлах сборки, в качестве целей и точек расширения, зависящих друг от друга. Основным известным использованием Ant является сборка Java-приложений, позволяющих компилировать, собирать, тестировать и запускать приложения Java.

4. Для запуска медиасервера red5 вводим команды в консоль:

```
a) cd /home/websys53/videos/red5
б) sudo sh red5.sh
```

После запуска консоль не закрываем, а сворачиваем. В ней будет вестись log [INFO]. Если закрыть консоль, возможна остановка red5 медиасервера.

5. Для остановки медиасервера red5 вводим команды в консоль:

```
a) cd /home/websys53/videos/red5
б) sudo sh red5-shutdown.sh
```

6. Папка для видео: /home/websys53/videos/red5/webapps/vod/streams/. Новые видеозаписи добавляем в эту директорию.

7. При добавлении в ГИС ссылка на видеозапись будет выглядеть:
rtmp://192.168.1.123:1935/vod/01_09_16_1.mp4

где:

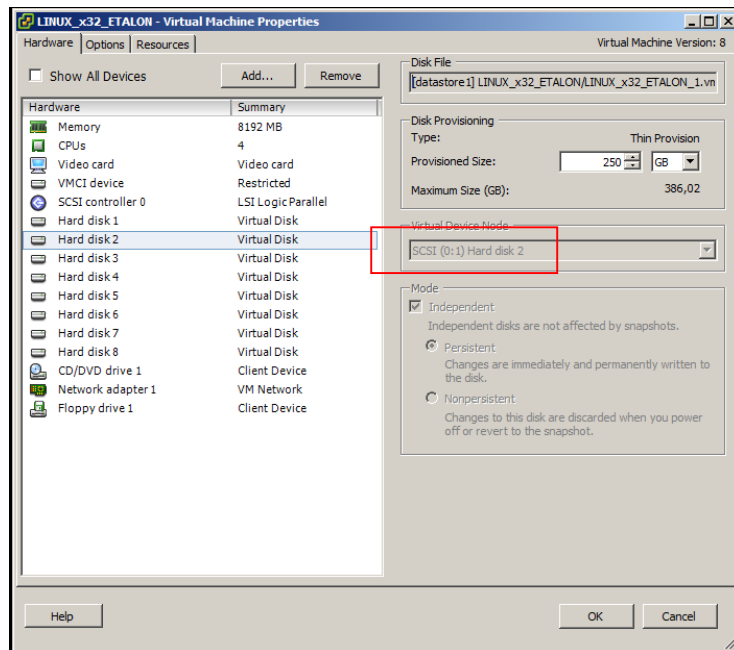
- 192.168.1.123 – IP-адрес сервера с медиа сервером red5
- 1935 – порт. Стандартные порты: RTMP:1935; Debug proxy:1936; HTTP servlet:5080; RTMPT:8088
- 01_09_16_1.mp4 – название видеозаписи с расширением.

8. Настраиваем автозапуск в соответствии с п. 7.1 Настройка медиасервера

10. Дисковое пространство LINUX

9.1 Подключение дополнительного диска

1. Подключаем дополнительный жёсткий диск в VMWare. Записываем номер диска, который виртуальная машина присваивает диску в свойствах (например, SCSI 0:1). Этот номер понадобится позже, при отмонтировании).



2. Заходим в консоль, смотрим список дисков в программе fdisk:

```
sudo fdisk -l
```

Отобразится список дисков, которые видит система.

Находим подключенный диск. В Ubuntu жесткие диски обозначаются /dev/sda, /dev/sdb, /dev/sdc и т.д. В рассматриваемом случае диск обозначается /dev/sdb.

```
did5@ubuntu:~$ sudo su
[sudo] password for did5:
root@ubuntu:/home/did5# fdisk -l | grep 'Disk /dev/sd'
Disk /dev/sdb doesn't contain a valid partition table
Disk /dev/mapper/ubuntu-root doesn't contain a valid partition table
Disk /dev/mapper/ubuntu-swap_1 doesn't contain a valid partition table
Disk /dev/sda: 21.5 GB, 21474836480 bytes
Disk /dev/sdb: 21.5 GB, 21474836480 bytes
```

3. Создаём раздел. Для разметки диска запускаем **fdisk** с указанием HDD, на котором нужно создать раздел:

```
sudo fdisk /dev/sdb
```

, где sdb – подключаемый диск

Далее в запущившейся программе набираем последовательно следующие команды:

- а) **p** [наличие разделов на диске, если диск новый, то покажет только общую информацию по диску]
- б) **n** [создаём новый раздел]
- в) **p** [тип раздела - основной]
- г) **1** [цифра от 1 до 4. Если диск новый, то 1]
- д) [спросит начальный цилиндр раздела, определит сам] нажимаем **Enter**
- е) [спросит конечный цилиндр раздела, определит сам] нажимаем **Enter**
- ё) **p** [проверяем список разделов, видим новый раздел, который для нашего примера обозначается как /dev/sdb1]
- ж) **w** [сохраняем изменения]

4. Форматируем новый раздел командой

ВАЖНО!!! Если на диске планируется размещать большое количество файлов (например, тайловая пирамида с десятками миллионов тайлов), то рекомендуется использовать файловую систему *btrfs*. Если же диск не предназначен для большого числа файлов, рекомендуется использовать стандартную файловую систему – *ext4*. В *ext4* используются *INODE*, количество файлов на диске ограничивается, и зависит от размера диска (например для 250 Gb диска это количество – 16,3 млн файлов)..

В зависимости от типа используемой файловой системы необходимо выполнять либо команду **sudo mkfs.btrfs /dev/sdb1** либо:

sudo mkfs.ext4 /dev/sdb1

Если при форматировании диска в систему *btrfs* и выдается сообщение, что команда *mkfs.btrfs* не найдена, то нужно установить утилиты *btrfs*

sudo apt-get install btrfs-tools

5. Создаём директорию, куда планируем примонтировать диск, например /home/websys53/HDD/OsmWms

sudo mkdir /home/websys53/HDD/OsmWms

6. Монтируем раздел в созданный каталог командой в консоли:

sudo mount /dev/sdb /home/websys53/HDD/OsmWms

, где *sdb* – монтируемый диск, /home/websys53/HDD/OsmWms – директорию монтируемого диска

7. Меняем права на общее чтение и редактирование файлов на монтируемом диске (примечание, если этого не сделать, возможны проблемы с чтением и записью файлов при работе без прав администратора). Выполняем команду в консоли:

sudo chmod -R 777 /home/websys53/HDD/OsmWms

8. Проверяем результат командой в консоли. В списке должен появиться примонтированный диск.

sudo df -h

```
user@ubuntu:~$ sudo df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            483M   0    483M   0% /dev
tmpfs           101M  6.4M   94M    7% /run
/dev/sda2       20G   3.4G   16G   18% /
tmpfs           501M  172K   500M   1% /dev/shm
tmpfs           5.0M   4.0K   5.0M   1% /run/lock
tmpfs           501M   0    501M   0% /sys/fs/cgroup
/dev/sdb1       16G   44M   15G    1% /home/user/Videos1
tmpfs           101M  48K   100M   1% /run/user/1000
```

9. Чтобы диск монтировался при старте системы добавляем строку в файл /etc/fstab. Открываем файл в консоли:

sudo nano /etc/fstab

и дописываем строку:

/dev/sdb /home/websys53/HDD/OsmWms ext4 defaults 1 2

, где *sdb* – монтируемый диск, /home/websys53/HDD/OsmWms – директорию монтируемого диска, *ext4* – формат файловой системы

10. Проверяем правильность выполнения – перезагружаем систему, запускаем команду из п.7. Диск должен быть в списке.

9.2 Отмонтирование диска

1. Сначала определяем, какой диск требуется отмонтировать. Выполняем команду, запоминаем букву диска, который отключаем (*sdc1*, *sdd1*...):

sudo df -h

2. Выполняем команду отмонтирования

sudo umount /dev/sdb

3. Выбрав диск, с правами администратора заходим в файл автозагрузки и убираем диск, который отключаем.

ВНИМАНИЕ! Если за диском с буквой, который мы отключаем, следуют другие диски (например, мы хотим извлечь диск *sdb1*, а в автозапуске ещё есть диски *sdcl* и *sddl*), то требуется в файле */etc/fstab* переименовать следующие за извлекаемым диски на одну букву алфавита меньше (в нашем примере диск *sdcl* станет *sdb1*, а диск *sddl* станет диском *sdcl*).

Правим файл */etc/fstab* командой:

```
sudo nano /etc/fstab
```

4. Выключаем виртуальную машину. Заходим в «Edit Settings...» виртуальной машины в VmWare, находим нужный диск.

5. Если требуется импортировать диск, определяем расположение диска на виртуализаторе, заходим в Datastore Browser, находим сервер, выбираем диск, осуществляем Импорт средствами VmWare.

ВНИМАНИЕ! Диски THIN PROVISION импортируются в ПОЛНОМ размере.

6. Отключить диск от виртуальной машины в свойствах. Заходим в «Edit Settings...» виртуальной машины в VmWare, находим нужный диск, нажимаем Remove.

11. Отключение автоматических ежедневных обновлений в debian

На всех серверах должно быть отключено автоматическое обновление. Для этого необходимо выполнить следующие настройки.

1. в /etc/apt/apt.conf.d/50unattended-upgrades комментируем в блоке Origin-Pattern:

Было:

```
origin=Debian,codename=${distro_codename},label=Debian-Security;
```

Стало:

```
// origin=Debian,codename=${distro_codename},label=Debian-Security;
```

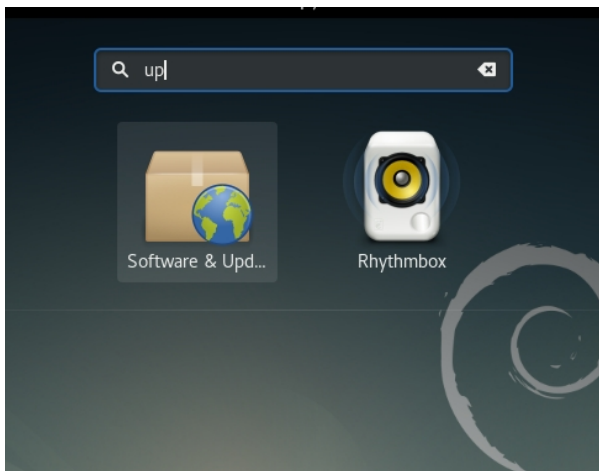
2. в /etc/apt/apt.conf.d/20auto-upgrades выставляем вместо "1" - "0" для двух параметров

APT::Periodic::Update-Package-Lists "0";

APT::Periodic::Unattended-Upgrade "0";

3. Отключаем автопроверку обновлений:

А. Открываем окно «Software & Updates» и вкладку «Updates» Через «Обзор»



Б. Выставляем автоматическую проверку обновлений в значение «Never»

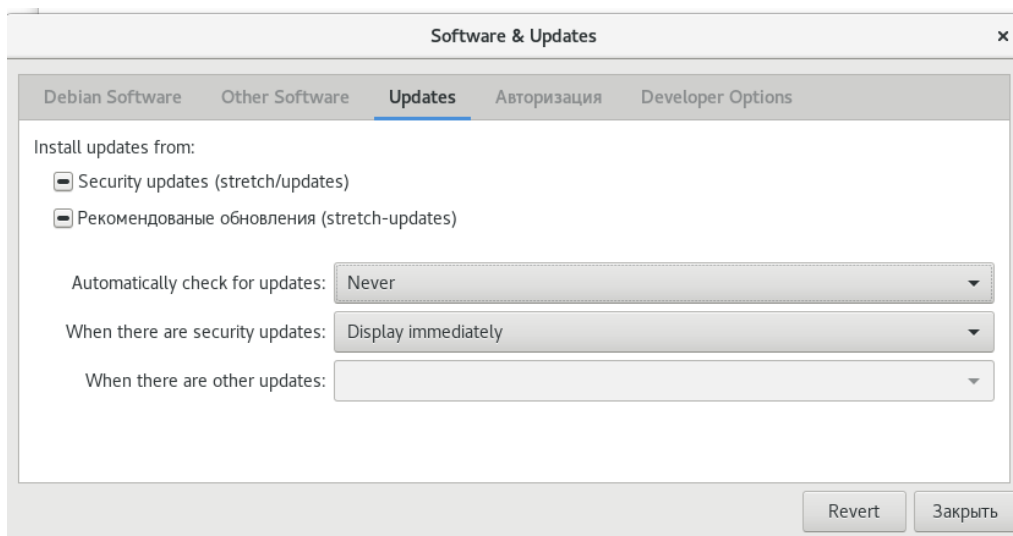
Было:

Automatically check for updates: Ежедневно

Стало:

Automatically check for updates: Never

В. Пункт «When there are security updates» должен быть выставлен в значение «Display immediately»



4. Перезагружаем систему

12. Настройка HTTPS

1. Если требуется создать самоподписанный сертификат выполняем действия пп.1.1-1.3, если нет – переходим к п.2:

1.1 Вводим команду

```
openssl req -new -x509 -days 365 -keyout server.key -out server.pem
```

На вопрос «Enter PEM pass phrase:» отвечаем паролем (например: test), подтверждаем.

Далее до вопроса Common Name просто нажимаем Enter.

На вопрос Common Name (eg, YOUR name) [] отвечаем адресом сайта (например: 192.168.1.123).

1.2 Снимаем пароль с ключа

```
cp server.key{,.orig}
```

```
openssl rsa -in server.key.orig -out server.key
```

```
rm server.key.orig
```

1.3 Копируем ключ и даём право чтения только администратору

```
sudo cp server.pem /etc/ssl/certs/
```

```
sudo cp server.key /etc/ssl/private/
```

```
sudo chmod 0600 /etc/ssl/private/server.key
```

2. Активируем мод SSL, включаем настройки сайта HTTPS по умолчанию

```
sudo a2enmod ssl
```

```
sudo a2ensite default-ssl
```

```
sudo a2enmod rewrite
```

3. Редактируем файл с настройками HTTPS `/etc/apache2/sites-enabled/default-ssl.conf`

После строки `<VirtualHost...` добавляем

```
DocumentRoot /home/websys53/GIS_WEB60/WebSite
```

```
Alias /web55 /home/websys53/GIS_WEB60/WebSite
```

После строки `SSLEngine on` добавляем

```
SSLProtocol all -SSLv2
```

Редактируем параметры

```
# Публичный сертификат сервера
```

```
SSLCertificateFile /etc/ssl/certs/server.pem
```

```
# Приватный ключ сервера
```

```
SSLCertificateKeyFile /etc/ssl/private/server.key
```

4. Отключение http, если настроен доступ по https:

а) отключаем сайт

```
sudo a2dissite 000-default
```

б) Отключаем 80-й порт

```
sudo nano /etc/apache2/ports.conf
```

комментируем строки:

```
#NameVirtualHost *:80
```

```
#Listen 80
```

5. Выполняем правку файлов приложения: Папка Core\Query во всех %Layers-файлах (baseLayers.xml, baseRasterLayers.xml, baseUserLayers.xml), а также в папке `gis_web77/Website/Core/Query` в файлах `%.json` файлах (`rasterlayers.json`, `pod equip_rasterlayers.json`). заменяем строку `"{SERVER}/tiletrees"` на `"https://IP_СЕРВЕРА_ПРИЛОЖЕНИЙ/tiletrees"`.

6. Убедиться что в файлах `/etc/apache2/sites-available/000-default.conf` и `default-ssl.conf` отсутствуют строки:

```
ProxyPass /web55/tiletrees/ http://127.0.0.1/tiletrees/
```

```
ProxyPassReverse /web55/tiletrees/ http://127.0.0.1/tiletrees/
```

Если эти строки есть - удалить и сохранить файлы.

7. Перезагружаем Apache

```
sudo service apache2 restart
```

12. Настройка сервера Apache для перенаправления запросов на интернет-ресурсы

Редирект (проксирование) – технология, которая в тех случаях, когда клиенту, который работает в своём браузере, запрещён доступ в интернет, позволяет организовать такой доступ из локальной сети через специальный выделенный сервер. Этот сервер отличается от других тем, что у него один диапазон прослушивает локальную сеть, а другой направлен в интернет. Благодаря этому он может принимать запросы, приходящие из локальной сети, отправлять их от своего имени в интернет, получать ответ и от своего имени внутри локальной сети распределять уже по клиентам. Клиенты получают доступ в интернет, но не напрямую, а через специальный сервер.

ВНИМАНИЕ! В силу ограничений, связанных с коммерческой и иными тайнами запрещается использовать эту технологию на том же сервере, на котором расположен сервер приложений. Следует использовать обособленный сервер.

В данном документе рассмотрена настройка такого сервера на базе UBUNTU 16.04 (32-bit) с установленным HTTP-сервером Apache. Для настройки последовательно выполняем операции:

1. Устанавливаем компоненты (если требуется), включаем модули Apache. Последовательно выполняем команды в консоли:

```
sudo aptitude install -y build-essential
sudo aptitude install -y libapache2-mod-proxy-html libxml2-dev
sudo a2enmod
proxy proxy_ajp proxy_http rewrite deflate headers proxy_balancer proxy_connect proxy_html
```

2. Редактируем файл «000-default.conf», который расположен в «/etc/apache2/sites-enabled» командой в консоли:

```
sudo nano /etc/apache2/sites-enabled/000-default.conf
```

3. Заполняем параметры проксирования для требуемых сайтов – Кадастр/росреестр, OpenStreetMap, Яндекс карты, Яндекс спутник, Google схемы и спутник. В теге <VirtualHost *:80>, после строки DocumentRoot... заполняем:

ProxyRequests Off

<Proxy *>

Order deny,allow

Allow from all

</Proxy>

SSLProxyEngine on

ProxyPassMatch ^/(.*openstreet/)(.*)\$ http://a.tile.openstreetmap.org/\$2

ProxyPassReverse ^/(.*openstreet/)(.*)\$ http://a.tile.openstreetmap.org/\$2

ProxyPassMatch ^/(.*cadastre/)(arcgis)(.*)\$ https://pkk5.rosreestr.ru/\$2\$3

ProxyPassReverse ^/(.*cadastre/)(arcgis)(.*)\$ https://pkk5.rosreestr.ru/\$2\$3

ProxyPassMatch ^/(.*cadastre/)(api)(.*)\$ https://pkk5.rosreestr.ru/\$2/features\$3

ProxyPassReverse ^/(.*cadastre/)(api)(.*)\$ **https://pkk5.rosreestr.ru/\$2/features\$3**

ProxyPassMatch ^/(.*yandexmaps/)(.*)\$ **http://vec01.maps.yandex.net/\$2**

ProxyPassReverse ^/(.*yandexmaps/)(.*)\$ **http://vec01.maps.yandex.net/\$2**

ProxyPassMatch ^/(.*yandexmapssat/)(.*)\$ **http://sat01.maps.yandex.net/\$2**

ProxyPassReverse ^/(.*yandexmapssat/)(.*)\$ **http://sat01.maps.yandex.net/\$2**

ProxyPassMatch ^/(.*googlemaps/)(.*)\$ **http://mt1.google.com/\$2**

ProxyPassReverse ^/(.*googlemaps/)(.*)\$ **http://mt1.google.com/\$2**

4. Сохраняем изменения, нажимаем *Ctrl+X*, затем *y*, затем *Enter*. Перезапускаем Apache командой:

```
sudo systemctl restart apache2.service
```

5. Вносим правки в файл *baseLayers.xml*, который расположен в */home/websys53/GIS_WEB60/WebSite/Core/Query/* и *rasterlayers.json*, который расположен в */home/websys53/gis_web77/website/Core/Query/*:

5.1. Для Росреестр (кадастр) с бланками (Требуется интернет):

В `<layer id="MixedTiles_Default">` заменить **http://pkk5.rosreestr.ru/** на **http://{SERVER}/cadastre/**, где {SERVER} – имя прокси-сервера, например 192.168.1.124

5.2. Для OpenStreetMap:

Если тег `<url>` присутствует в `<layer id="OSMapnikLayerINTERNAL">`, то заменить **https://{a-c}.tile.openstreetmap.org/** на **http://{SERVER}/openstreet/**, где {SERVER} – имя прокси-сервера, например 192.168.1.124

Если тега `<url>` в `<layer id="OSMapnikLayerINTERNAL">` нет, то после строки `<name>OpenStreetMapInternal</name>` требуется добавить строку: `<url>http://{SERVER}/openstreet/{z}/{x}/{y}.png</url>`.

5.3 Для Яндекс.Карты Схема:

В `<layer id="YANDEX_MAPS_SCHEMA_INTERNAL">` заменить **http://vec0{1-3}.maps.yandex.net/** на **http://{SERVER}/yandexmaps/**, где {SERVER} – имя прокси-сервера, например 192.168.1.124

5.4 Для Яндекс.Карты Спутник:

В `<layer id="YANDEX_MAPS_SATELITE_INTERNAL">` заменить **http://sat0{1-3}.maps.yandex.net/** на **http://{SERVER}/yandexmapssat/**, где {SERVER} – имя прокси-сервера, например 192.168.1.124

5.5 Для Google Схемы:

В `<layer id="GOOGLE_MAPS_SCHEMA">` заменить **http://mt{1-3}.google.com/** на **http://{SERVER}/googlemaps/**, где {SERVER} – имя прокси-сервера, например 192.168.1.124

5.6 Для Google Спутник:

В `<layer id="GOOGLE_MAPS_SCHEMA_PODS_RASTER">` заменить **http://mt{1-3}.google.com/** на **http://{SERVER}/googlemaps/**, где {SERVER} – имя прокси-сервера, например 192.168.1.124

6. В файле **Web.config** выполняем правку параметра Rosreestr_Url как показано ниже:
<add key="Rosreestr_Url" value="http://10.44.254.31/cadastre/api/" />

7. Перестартовываем Apache командой:

```
sudo systemctl restart apache2.service
```

13. Установка и настройка сайта: статический контент

Для настройки статического контента последовательно выполняем операции:

1. Создаём папку `/home/websys53/GIS_WEB60`, копируем в неё сборку сайта для linux.
2. Изменяем права доступа для папки с сайтом. Выполняем команду в консоли:

```
sudo chmod -R 777 /home/websys53/GIS_WEB60
```

3. Устанавливаем шрифты в систему. Для этого выполняем следующие операции:

3.1) Заходим в папку `/home/websys53`, нажимаем комбинацию клавиш `Ctrl+H`, чтобы отобразились скрытые файлы/папки. Заходим (или создаём, если отсутствует) в папку `.fonts`. Создаём в ней папку `truetype/mapgen` (общая структура каталога будет `/home/user/.fonts/truetype/mapgen`), копируем в эту папку шрифты. Также создаём папку `/home/websys53/.fonts/truetype/msttcorefonts`. Копируем в эту папку шрифты Microsoft из архива.

Кроме этого, необходимо добавить в папку `/home/websys53/.fonts/truetype/msttcorefonts` «спец» версии шрифтов Times New Roman CYR и Arial CYR. В противном случае, даже если шрифты Times New Roman, Arial и т.д. есть в системе, приложение не будет их использовать.

3.2) Запускаем установку шрифтов командой в консоли:

```
sudo fc-cache -fv
```

1. Редактируем файл `dir.conf`, который расположен в `/etc/apache2/mods-enabled`. В теге `<IfModule>` в строке «`DirectoryIndex`» добавляем «`main_js.html`» перед «`index.html`». В этой строке по порядку следования Apache2 будет выбирать, какую страницу открывать первой из директории (если находит такую страницу в папке).

2. Редактируем файл «`000-default.conf`», который расположен в `/etc/apache2/sites-enabled`.

Удаляем строку:

`DocumentRoot var/www/html`

Добавляем строки:

`DocumentRoot /home/websys53/gis_web77/website`

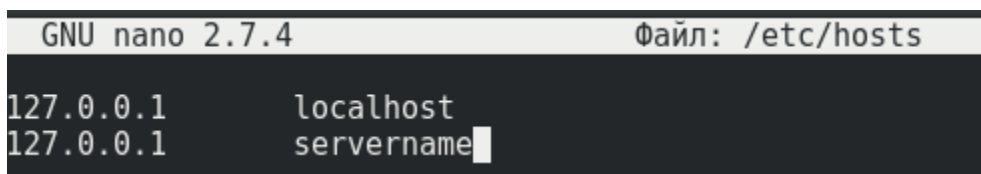
`Alias /web55 /home/websys53/gis_web77/website`

`Alias /web77 /home/websys53/gis_web77/website`

`Alias /old55 /home/websys53/GIS_WEB60/Website`

5.1 Чтобы исключить ситуацию с нерабочей ссылкой, содержащей `web55/` необходимо выполнить дополнительные настройки

Правим файл `/etc/hosts` (например командой `sudo nano`), в начало файла записываем (если она отсутствует) следующую строку `"127.0.0.1 название_сервера"` без кавычек (например - `servername`). Название сервера необходимо получить у системного администратора.



```
GNU nano 2.7.4                               Файл: /etc/hosts
127.0.0.1      localhost
127.0.0.1      servername
```

Сохраняем изменения: `Ctrl+x`, затем `y`, затем `Enter`.

Редактируем файл «`000-default.conf`»

sudo nano /etc/apache2/sites-available/000-default.conf

вместо строки

#ServerName www.example.com

пишем

ServerName localhost

Сохраняем изменения: *Ctrl+x*, затем *y*, затем *Enter*.

Выполняем команду:

cd /usr/bin

sudo ln -s mod-mono-server4 mod-mono-server2

3. Редактируем файл «*apache2.conf*», который расположен в */etc/apache2*.

Устанавливаем параметру **Timeout** значение **900** (в секундах, т.е. 15 минут).

Перенастраиваем Apache. Добавляем строки (в квадратных скобках описание):

```
<Directory /home/websys53/GIS_WEB60/WebSite> [ПУТЬ К ПАПКЕ С САЙТОМ]
Options FollowSymLinks [СЕРВЕР БУДЕТ
СЛЕДОВАТЬ ЗА СИМВОЛИЧЕСКИМИ СВЯЗЯМИ В КАТАЛОГЕ]
AllowOverride All [ДОСТУПНЫ ВСЕ ДИРЕКТИВЫ]
Require all granted [ВСЕ ПОЛЬЗОВАТЕЛИ ИМЕЮТ ДОСТУП К КАТАЛОГУ]
</Directory>
```

Сохраняем изменения: *Ctrl+x*, затем *y*, затем *Enter*.

4. Создаём ссылки на каталоги */home/websys53/GIS_WEB60/Log*, */home/websys53/GIS_WEB60/Help*, */home/websys53/GIS_WEB60/Mapgen/TileTrees*. Последовательно выполняем команды в консоли:

```
ln -s /home/websys53/GIS_WEB60/Log /home/websys53/GIS_WEB60/WebSite/Public
```

```
ln -s /home/websys53/GIS_WEB60/Help /home/websys53/GIS_WEB60/WebSite/Public
```

```
ln -s /home/websys53/GIS_WEB60/MapGen/TileTrees /home/websys53/GIS_WEB60/WebSite/Public/Data/Maps
```

5. Чтобы файлы не открывались как текст в браузере:

- запускаем консоль, вводим *sudo nautilus*, в открывшемся окне заходим в папку */etc*, открываем в текстовом редакторе файл *mime.types*.

Добавляем строки:

application/x-unknown img

application/gpx+kml gpx

6. Убеждаемся, что существуют папки. Если их нет, то создаём и оставляем пустыми.

/home/websys53/GIS_WEB60/WebSite/Public/Data/WMS

/home/websys53/GIS_WEB60/WebSite/Public/Data/TAB

/home/websys53/GIS_WEB60/WebSite/Log/UTEService

/home/websys53/GIS_WEB60/WebSite/Log/WebSite

7. Для переадресации с /web55 включаем модуль rewrite и перезагружаем Apache. Последовательно выполняем команды в консоли:

```
sudo a2enmod rewrite
sudo systemctl restart apache2.service
```

8. Убеждаемся, что файлы:

```
/home/websys53/GIS_WEB60/WebSite/Web.config
/home/websys53/GIS_WEB60/UTEService/Bin/UTEService.exe.config
/home/websys53/GIS_WEB60/UTEService/Bin/UTECall.exe.config
/home/websys53/GIS_WEB60/MapGen/MRS_Service.exe.config
```

в параметре *openTimeout* и *sendTimeout* содержат значения «20:00:00:00» - т.е. 20 дней. Чтобы отключить https(если был включен) при настроенном http необходимо выполнить команду:

```
sudo a2disssite 000-default
```

9. Добавление расширений файлов, которые браузер должен предлагать скачать вместо открытия их во вкладки

Строка со списком расширений файлов должна содержать сами расширения без точки, которые разделены вертикальной чертой, потом все расширения продублировать БОЛЬШИМИ БУКВАМИ

Пример:

```
pdf|gpx|kml|doc|img|rtf|gtar|gz|tar|zip|txt|vrml|exe|wrl|m_3d_scene|vrz|PDF|GPX|KML|DOC|IMG|RTF|GTAR|GZ|TAR|ZIP|TXT|VRML|EXE|WRL|M_3D_SCENE
```

- 1) Включить модуль apache(если был включен, то выдастся сообщение о том, что модуль уже был включен "Module headers already enabled")

```
sudo a2enmod headers
```

- 2) Отредактировать файлы 000-default.conf и default-ssl.conf (только в ЛТГ) из /etc/apache2/sites-available/

Вписываем в самый конец файла перед тегом </VirtualHost> следующий блок текста:

```
<FilesMatch "\.(РАСШИРЕНИЯ ФАЙЛОВ)$">
```

```
Header set Content-Disposition attachment
```

```
</FilesMatch>
```

- 3) Перезапустить apache:

```
sudo service apache2 restart
```

4. Установка и настройка сайта: динамический контент

1. Убедиться, что файлы:

```
- /home/websys53/GIS_WEB60/UTEService/Bin/UTEService.exe.config
- /home/websys53/GIS_WEB60/WebSite/web.config
```

в секции «netTcpBinding» содержат следующие строки:

```
<netTcpBinding>
  <binding openTimeout="00:20:00" sendTimeout="00:20:00"
    name="NetTcpBinding_IMyService">
    <security mode="None">
    </security>
  </binding>
</netTcpBinding>
```

2. Убедиться, что файл /home/websys53/GIS_WEB60/WebSite/web.config

- 1.1 В секции <client> в элементе <endpoint> содержит следующие строки:

```
<endpoint name="tcp" contract="ASL.IUTEService"
address="net.tcp://localhost:8003/UTEService/tcp" binding="netTcpBinding"
bindingConfiguration="NetTcpBinding_IMyService" />
```

2.2 В секции <netTcpBinding>:

```
<binding openTimeout="00:20:00" maxReceivedMessageSize="2147483647"
sendTimeout="00:20:00" name="NetTcpBinding_IMapGenService">
  <readerQuotas maxDepth="2147483647" maxStringContentLength="2147483647"
maxArrayLength="2147483647" maxBytesPerRead="2147483647"
maxNameTableCharCount="2147483647" />
  <security mode="None">
  </security>
</binding>
```

1.3 В секции <client>:

```
<endpoint name="tcp" contract="ASL.IMapGenService"
address="net.tcp://localhost:8004/MapGenService/tcp" binding="netTcpBinding"
bindingConfiguration="NetTcpBinding_IMapGenService" />
```

1.4 В секции <add_key> указываем корректный Query_Path:

```
<add_key="Query_Path" value="/home/websys53/GIS_WEB60/WebSite/Core/Query" />
```

3. Убедиться, что файл GIS_WEB60/MapGen/MRS_Service.exe.config в разделе <configuration> содержит следующие строки:

```
<system.serviceModel>
  <bindings>
<netTcpBinding>
  <binding maxReceivedMessageSize="2147483647" openTimeout="00:20:00"
sendTimeout="00:20:00" name="NetTcpBinding_IMapGenService">
    <readerQuotas maxDepth="2147483647" maxStringContentLength="2147483647"
maxArrayLength="2147483647" maxBytesPerRead="2147483647"
maxNameTableCharCount="2147483647" />
    <security mode="None">
    </security>
  </binding>
</netTcpBinding>
</bindings>

  <services>
    <service
      name="WcfService.MapGenService"
      behaviorConfiguration="MyServiceBehaviors" >
      <host>
        <baseAddresses>
          <add baseAddress="net.pipe://localhost/MapGenService/" />
        </baseAddresses>
      </host>
      <endpoint
        contract="WcfService.IMapGenService"
        binding="netTcpBinding"
        bindingConfiguration="NetTcpBinding_IMapGenService"
        address="net.tcp://localhost:8004/MapGenService/tcp" />
      </service>
    </services>
  <behaviors>
    <serviceBehaviors>
```

```
<behavior name="MyServiceBehaviors" >  
    <serviceMetadata httpGetEnabled="true" />  
    </behavior>  
</serviceBehaviors>  
</behaviors>  
</system.serviceModel>
```

4. Запускаем службу UTEService и Mapgen. Вводим команды:

```
cd /home/websys53/GIS_WEB60/UTEService/Bin  
bash UTEService_switch_demon.sh start  
cd /home/websys53/GIS_WEB60/MapGen  
bash MRS_Service_switch_demon.sh start
```

После запуска MapGen консоль нужно оставить открытой.

5. Для остановки UTEService выполняем команды:

```
cd /home/websys53/GIS_WEB60/UTEService/Bin  
bash UTEService_switch_demon.sh stop
```

Для остановки MapGen выполняем команды:

```
cd /home/websys53/GIS_WEB60/MapGen  
bash MRS_Service_switch_demon.sh stop
```

После остановки служб, заходим в папку /tmp, удаляем файлы из списка ниже, если они есть:

- MRS_Service.lock
- UTEService.lock

Только при отсутствии этих файлов можно заново запускать службы.

5. Распараллеливание сайтов TileTrees и WebSite.

В целях оптимизации очереди запросов в рамках одного сервера apache предусмотрено распараллеливание на 2 подсайта TileTrees и WebSite. В таком случае каждый сайт получает и обрабатывает свою очередь запросов.

Если имеется предварительно настроенный сайт, то необходимо включить модули проксирования в apache2 и выполнить следующую настройку файлов 000-default.conf и default-ssl.conf.

1. Включить модули проксирования в apache2 командой:

```
sudo a2enmod proxy proxy_http
```

2. Редактирование файла /etc/apache2/sites-available/000-default.conf

Войти в редактор файла командой:

```
sudo nano /etc/apache2/sites-available/000-default.conf
```

Дописать в файл новые директории и алиасы:

```
Alias /tiletrees /home/websys53/GIS_WEB60/MapGen/TileTrees
```

```
<Directory /home/websys53/GIS_WEB60/WebSite>
```

```
Options -Indexes +FollowSymLinks
```

```
AllowOverride All
```

```
Require all granted
```

```
</Directory>
```

```
<Directory /home/websys53/GIS_WEB60/MapGen>
```

```
Options -Indexes +FollowSymLinks
```

```
AllowOverride All
```

```
Require all granted
```

```
</Directory>
```

Сохранить файл:

нажать **ctrl+x**, затем **Y**, затем **enter**

3. Редактирование файла /etc/apache2/sites-available/default-ssl.conf

Войти в редактор файла командой:

```
sudo nano /etc/apache2/sites-available/default-ssl.conf
```

Дописать в файл новые директории и алиасы:

```
Alias /tiletrees /home/websys53/GIS_WEB60/MapGen/TileTrees
```

```
<Directory /home/websys53/GIS_WEB60/WebSite>
```

```
Options -Indexes +FollowSymLinks
```

```
AllowOverride All
```

```
Require all granted
```

```
</Directory>
```

```
<Directory /home/websys53/GIS_WEB60/MapGen>
```

```
Options -Indexes +FollowSymLinks
```

```
AllowOverride All
```

```
Require all granted
```

```
</Directory>
```

Сохранить файл:

нажать **ctrl+x**, затем **Y**, затем **enter**

6. Настройка планировщика и SH-файлов

16.1 Общая информация, перечень SH-файлов

В Windows часть задач, выполняемых на сервере, была оформлена в виде командных BAT-файлов. Некоторые задачи добавлялись в планировщик IIS и выполнялись с указанной периодичностью. В LINUX такого же рода файлы имеют расширение «*.sh».

Перечень SH-файлов указан в таблице 1. Во всех SH-файлах как корневой каталог приложения указан директорий: **/home/websys53/GIS_WEB60**.

Если сайт расположен в другом месте, то во всех файлах следует исправить значения **/home/websys53/GIS_WEB60** на выбранный путь.

Таблица 1 Список SH-файлов

№	Расположение	Название	Описание
1	MapGen	create_run_updateMaps.sh	Создание заданий планировщика для обновления карт
2		delete_run_updateMaps.sh	Удаление заданий планировщика для обновления карт
3		updateMaps.sh	Обновление карт растеризатора
4	RoutingService\	Start.sh	Запуск службы маршрутизации
5	RoadService	Stop.sh	Остановка службы маршрутизации
6	Tasks\Calc\Batch	run_calc_cluster_all.sh	Пакетная группировка в кластеры по всем отчетам ВТД
7		run_calc_cp_all.sh	Пакетный расчет по всем отчетам ВТД
8		run_calc_hodo_all.sh	Пакетный расчет ХОДО по всем отчетам ВТД
9		run_calc_ili_all.sh	Пакетный расчет по всем отчетам ВТД
10		run_calc_ili_sto_all.sh	Пакетный расчет СТО по всем отчетам ВТД
11		run_calc_ili_sto_ehz_all.sh	Пакетный расчет СТО ЭХЗ по всем отчетам ВТД
12		run_calc_pressure_all.sh	Пакетный расчет давления по всем отчетам ВТД
13		run_calc_single_ili.sh	Расчет одного отчета ВТД
14		run_gen_img_all.sh	Пакетная генерация IMG-карт по всем ЛПУ
15		run_gen_vrml_all.sh	Пакетная генерация 3D-сцен по всем ЛПУ
16		run_group_event_on_routes_all.sh	Пакетное проецирование
17		run_group_MARKER_on_routes_all.sh	Пакетное проецирование Marker
18		run_group_structures_on_lines_all.sh	Пакетная группировка всех крановых площадок по газопроводам
19		run_group_testleads_on_routes_all.sh	Пакетное проецирование всех КИП-ов и КИК-ов
20		run_group_valves_on_structures_all.sh	Пакетная группировка всех кранов по крановым площадкам
21		run_km_calc_all.sh	Пакетный расчет километража по всем участкам
22		run_lrs_calc_all.sh	Пакетный расчет линейной дистанции по всем участкам
23		run_ps_index_all.sh	Пакетный расчет индексации по всем участкам
24		run_batch.sh	Аналог «run_batch.cmd». Пакетные расчёты.
25		run_batch_STO.sh	Аналог «run_batch_STO.cmd». Пакетные расчёты.
26	Tasks\Calc\Corridor	run_cor_calc.sh	Запуск переиндексации по коридорам
27	Tasks\DBClean	create_db_clean.sh	Создание задачи запуска Удаление неиспользуемых данных PODS
28		delete_db_clean.sh	Удаление задачи запуска Удаление неиспользуемых данных PODS
29		run_db_clean.sh	Запуск удаления неиспользуемых данных PODS

№	Расположение	Название	Описание
30	Tasks\ DBStat	create_db_stat_refresh.sh	Создание задачи запуска Пересбора статистики
31		delete_db_stat_refresh.sh	Удаление задачи запуска Пересбора статистики
32		run_db_stat_refresh.sh	Запуск пересбора статистики
33	Tasks\Integration	create_run_integration_type_1_link.sh	Создание задачи запуска Интеграция Связывание объектов
34		create_run_integration_type_2_transfer.sh	Создание задачи запуска Интеграция Перенос объектов ВТД
35		create_run_integration_type_2_transfer_ehz.sh	Создание задачи запуска Интеграция Перенос объектов ЭХЗ
36		delete_run_integration_type_1_link.sh	Удаление задачи запуска Интеграция Связывание объектов
37		delete_run_integration_type_2_transfer.sh	Удаление задачи запуска Интеграция Перенос объектов ВТД
38		delete_run_integration_type_2_transfer_ehz.sh	Удаление задачи запуска Интеграция Перенос объектов ЭХЗ
39		run_integration_type_1_link.sh	Запуск интеграции связывания объектов
40		run_integration_type_2_transfer.sh	Запуск интеграции переноса объектов ВТД
41		run_integration_type_2_transfer_ehz.sh	Запуск интеграции переноса объектов ЭХЗ
42	Tasks\MapInfo_ T50_Refresh	create_mapinfo_t50_data_refresh.sh	Создание задачи запуска Заполнение данных для MapInfo
43		delete_mapinfo_t50_data_refresh.sh	Удаление задачи запуска Заполнение данных для MapInfo
44		run_mapinfo_t50_data_refresh.sh	Запуск заполнения данных для MapInfo
45	Tasks\MapTiles	map_tiles_gen.sh	Задания для работы с картами
46		OSM_tiles_gen.sh	Перетайливание слоев OSM.
47		podsequip_tiles_gen.sh	Генерирование векторной пирамиды тайлов
48		rosreestr_tiles_gen.sh	Тайлогенерация пирамиды тайлов РосРеестра с ID 4
49	Tasks\Warmup Bot	restart.sh	Скрипт для перезагрузки службы растеризации и перезагрузки Apache.
50	WebSite	clear.sh	Запуск очистки временных папок
51		create_run_cleanTempData.sh	Создание задачи запуска очистки временных папок
52		delete_run_cleanTempData.sh	Удаление задачи запуска очистки временных папок
53		backup_and_clear.sh	Скрипт чистки и создания бэкапов
54		install_patch.sh	Скрипт установки патчей.
55		revert_patch.sh	Скрипт откатывания патча
56		set_htaccess.sh	Скрипт копирования файла прав для apache, который разрешает либо запрещает доступ к директориям

57		set_rights.sh	Скрипт установки прав на папки
58	Tasks	service_autorun.sh	Скрипт автозагрузки служб
59	Tasks	generate_http_data.sh	Скрипт генерации файлов постраничности для сайта настроенного по протоколу http
60	Tasks	generate_https_data.sh	Скрипт генерации файлов постраничности для сайта настроенного по протоколу https

В некоторых SH-файлах требуется выполнить настройку параметров, таких как IP сервера баз данных, название БД, имя и пароль пользователя БД, идентификатор отчёта, периодичность запуска и т.д. Перечень задач, которые необходимо конфигурировать перед запуском указаны в следующем пункте.

16.2 SH-файлы, которые конфигурируем перед запуском

16.2.1 SH-файлы, требующие настройки параметров БД

В некоторых sh-файлах перед запуском необходимо указать такие параметры, как IP сервера БД, имя БД, логин и пароль пользователя БД, идентификатор отчёта и т.д. Такие параметры вынесены в начало файла. Перечень таких файлов:

- \Tasks\Calc\Batch\run_calc_single_ili_.sh
- \Tasks\MapInfo_T50_Refresh\run_mapinfo_t50_data_refresh.sh
- \Tasks\DBClean\run_db_clean.sh
- \Tasks\DBStat\run_db_stat.sh
- \Tasks\Integration\run_integration_type_1_link.sh
- \Tasks\Integration\run_integration_type_2_transfer.sh
- \Tasks\Integration\run_integration_type_2_transfer_ehz.sh
- \MapGen\updateMaps.sh

16.2.2 SH-файлы, требующие настройки параметров расчётов

В некоторых sh-файлах изначально прописаны усреднённые параметры расчётов. При необходимости следует изменять эти параметры, указав требуемые значения. Усреднённые параметры присутствуют в файлах:

- /Tasks/Calc/Batch/run_calc_cluster_all.sh
- /Tasks/Calc/Batch/run_calc_ili_sto_all.sh
- /Tasks/Calc/Batch/run_calc_ili_sto_ehz_all.sh
- /Tasks/Calc/Batch/run_calc_pressure_all.sh

16.2.3 SH-файлы, требующие настройки планировщика

Некоторые sh-файлы предназначены для добавления или исключения задач из планировщика.

Планировщик – программа, выполняющая задачи с указанной периодичностью. Перечень задач указан в файле, расположенном: **/var/spool/cron/crontabs/root**. В файле каждая строка – задача, которая выполняется по определенным правилам. В порядке следования идут:

- периодичность, задача выполняется раз в (указываются минуты (значения 0-59), часы (значения 0-23), дни (значения 1-31), месяцы (значения 1(январь)-12(декабрь)), дни недели (значения 1(понедельник)-7(воскресенье));

- пользоваель;
- файл или задача.

Например:

a)0 1 1-7 * 7 root

/home/websys53/GIS_WEB60/MapGen/run_updateMaps.sh

каждое первое воскресенье месяца в 01:00 будет запускаться Задача Обновления карт растрезатора;

б)0 2 * * 6 root / home/user/GIS_WEB60/WebSite/clean.sh

каждую субботу в 02:00 ночи будет запускаться Задача Очистки временных папок

в)0 22 * * * root / home/user/GIS_WEB60/Tasks/DBClean/run_db_clean.sh
 ежедневно в 22:00 ночи будет запускаться Задача Удаление неиспользуемых данных PODS

Рассмотрим добавление и удаление задачи планировщика на примере «Обновления карт растеризатора»:

- создание задачи запуска (перед запуском устанавливаем периодичность в файле `/home/websys53/GIS_WEB60/MapGen/create_run_updateMaps.sh`, по умолчанию установлено раз в год, 1 января)

```
cd /home/websys53/GIS_WEB60/MapGen
sudo bash create_run_updateMaps.sh
```

- удаление задачи запуска

```
cd /home/websys53/GIS_WEB60/MapGen
sudo bash delete_run_updateMaps.sh
```

Перечень задач, пути и значения по умолчанию указаны в таблице 2.

№	Название задачи	Папка	Файл	Эталонные периодичность и время запуска
1	ГИС 54 БД Заполнение данных для доступа MapInfo	Tasks\MapInfo_T50_Refresh\	run_mapinfo_t50_data_refresh.sh	Раз в месяц по пятницам, 23-55
2	ГИС 54 БД Интеграция Перенос данных и расчет ВТД. Из Oracle в Postgress	Tasks\Integration\	run_integration_type_2_transfer_pg.sh	раз в неделю, по субботам, 23-55
3	ГИС 54 БД Интеграция Перенос данных и расчет ЭХЗ	Tasks\Integration\	run_integration_type_2_transfer_ehz.sh	никогда
4	ГИС 54 БД Интеграция Связывание объектов	Tasks\Integration\	run_integration_type_1_link.sh	никогда
5	ГИС 54 БД Интеграция. Перенос сведений по ремонтам из ССД Инфотех	Tasks\Integration\	ВАТ-файл отсутствует. Скрипт переноса: integration_type_2_repair.sql	никогда
6	ГИС 54 БД Пересбор статистики	Tasks\DBStat\	run_db_stat_refresh.sh	раз в неделю, по воскр., 21-00
7	ГИС 54 БД Удаление неиспользуемых файлов PODS	Tasks\DBClean\	Tasks\DBClean\run_db_clean.sh	раз в неделю, по воскр., 9-00
8	ГИС 54 Сервер приложений Обновление карт растеризатора	Разные в Linux и Oracle	Tasks\map_refresh\run_refresh.sh	раз в неделю, по воскр., 23-55
9	ГИС 54 Сервер приложений Переиндексация по коридорам	Tasks\Calc\Corridor\	run_cor_calc.sh	никогда
10	ГИС 70. Перестарт Apache, служб растеризатора	Tasks\WarmupBot\	restart.sh	ежедневно, 23.55

После выполнения команд проверяем, что задачи прописаны в планировщике командой:

```
sudo crontab -l
```

В консоли появятся строки вида:

```
* * 1 1 * root  
/home/websys53/GIS_WEB60/MapGen/run_updateMaps.sh  
50 1 8 8 * root /home/user/GIS_WEB60/WebSite/clean.sh
```

, где в порядке следования идут: минуты, часы, дни месяца, месяцы, дни недели запуска, пользователь и запускаемый файл/команда (в примере в планировщике добавлены две задачи: а) 1 января будет запускаться Задача Обновления карт растеризатора и б) в 01:50 ночи 8 августа будет запускаться Задача Очистки временных папок).

7. Установка обновления приложения

Вариант 1. Через скрипты

Предполагается, что файлы патча скрипты запуска находятся в директории /home/websys53/patch/.

1. Запускаем скрипт установки патча:
`bash /home/websys53/patch/install_patch.sh`
2. Если требуется откатить изменения, выполняем команду:
`bash /home/websys53/patch/revert_patch.sh`

Вариант 2. Ручная установка обновления

Если требуется установить обновление (получен патч от разработчика), следует выполнить обновление в следующей последовательности:

3. Останавливаем Apache, UTEService, MapGen командами:

```
sudo systemctl stop apache2.service
cd /home/websys53/GIS_WEB60/UTEService/Bin
bash UTEService_switch_demon.sh stop
cd /home/websys53/GIS_WEB60/MapGen
bash MRS_Service_switch_demon.sh stop
```

После остановки служб, заходим в папку /tmp, удаляем файлы из списка ниже, если они есть:

- MRS_Service.lock
- UTEService.lock

4. Заменяем файлы в папке /home/websys53/GIS_WEB60/ (путь расположения приложения) файлами из патча.

5. Даём права редактирования папке /home/websys53/GIS_WEB60/ командой:

```
sudo chmod -R 777 /home/websys53/GIS_WEB60
```

6. Запускаем Apache, UTEService, MapGen командами:

```
cd /home/websys53/GIS_WEB60/UTEService/Bin
bash UTEService_switch_demon.sh start
cd /home/websys53/GIS_WEB60/MapGen
bash MRS_Service_switch_demon.sh start
sudo systemctl start apache2.service
```

Внимание! После старта или перезапуска **apache** необходимо подождать 3 минуты до попытки захода на сайт.

8. Установка службы маршрутизации (полная)

1. Подготовка графа дорог

Сервис маршрутизации использует подготовленный граф дорог, который получается обработкой исходного файла в формате .pbf (обновляются файлы графа России и стран СНГ на ресурсе http://gis-lab.info/projects/osm_dump/index.html). Оттуда необходимо скачать свежий файл osm.pbf для России. Можно скачать файл для всех стран СНГ или для конкретного региона (на странице ресурса описано подробнее).

2. Подготовка инфраструктуры для службы маршрутизации

- а) Устанавливаем необходимые пакеты для компиляции исходников службы маршрутизации <https://github.com/Project-OSRM/osrm-backend/wiki/Building-on-Ubuntu>

```
sudo apt install build-essential git cmake pkg-config \
libbz2-dev libstxxl-dev libstxxl1v5 libxml2-dev \
libzip-dev libboost-all-dev lua5.2 liblua5.2-dev libtbb-dev
```

б) Сборка из исходников(Сборка проекта из исходных текстов длится около 10-15 минут(х64 – около 12 минут) <https://github.com/Project-OSRM/osrm-backend/wiki/Building-OSRM>

- Если папка osrm-backend существует и в ней установлена старая служба, то удаляем эту папку
- копируем архив osrm-backend-5.12.0.tar.gz в папку /home/websys53/GIS_WEB60/RoutingService и переходим в эту папку в консоли

```
cd /home/websys53/GIS_WEB60/RoutingService
```

- распаковываем архив и созданную папку переименовываем в osrm-backend

```
tar -xzf osrm-backend-5.12.0.tar.gz
```

```
mv osrm-backend-5.12.0 osrm-backend
```

в) Компиляция и установка в систему сервера маршрутизации

```
cd osrm-backend
```

```
mkdir -p build
```

```
cd build
```

```
cmake .. -DCMAKE_BUILD_TYPE=Release
```

```
cmake --build .
```

```
sudo cmake --build . --target install
```

3) Подготовка графа дорог для запуска сервера

С помощью утилиты osrm-extract, данные извлекаются из карты и сохраняются для дальнейшей загрузки. Перед тем как извлекать данные, необходимо указать файл профиля (profile.lua), в котором описано какие именно маршруты необходимы. Все профили лежат в папке profiles в корневой папке с исходниками и подключаются созданием ссылки на папку

- Возвращаемся в директорию службы маршрутизации

```
cd ../../RoadService
```

- Создаем ссылку на папку с профилями

```
ln -s osrm-backend/profiles
```

- Для извлечения данных потребуется много ресурсов, поэтому желательно создать swapfile в случае, если в консоли выдается подобная ошибка:

```
[info] RAM: peak bytes used: 4419612672
```

```
[error] [exception] std::bad_alloc
```

```
[error] Please provide more memory or consider using a larger swapfile
```

- Если файл подкачки уже существует (/home/websys53/GIS_WEB60/RoutingService/RoadServiceSwapFile), то нужно его удалить(либо не трогать. Предпочтительно не трогать)

```
sudo swapoff /home/websys53/GIS_WEB60/RoutingService/RoadServiceSwapFile
```

```
rm /home/websys53/GIS_WEB60/RoutingService/RoadServiceSwapFile
```

- Создаем файл подкачки

```
fallocate -l 20G /home/websys53/GIS_WEB60/RoutingService/RoadServiceSwapFile
```

```
chmod 600 /home/websys53/GIS_WEB60/RoutingService/RoadServiceSwapFile
```

```
mkswap /home/websys53/GIS_WEB60/RoutingService/RoadServiceSwapFile
```

```
sudo swapon /home/websys53/GIS_WEB60/RoutingService/RoadServiceSwapFile
```

- Дополнительно создаем файл в директории, где лежит программа экспорта, если его там нет

```
nano ../osrm-backend/build/.stxxl
```

```
disk=/tmp/stxxl,20G,syscall
```

- Удаляем все файлы в папке /home/websys53/GIS_WEB60/RoutingService/RoadService. Там должна остаться только ссылка на папку profiles
- Копируем скаченный файл RU.osm.pbf в папку /home/websys53/GIS_WEB60/RoutingService/RoadService
- Запускаем команду извлечения данных (занимает около 30 минут)

```
osrm-extract RU.osm.pbf
```

либо с прямым указанием профиля

osrm-extract RU.osm.pbf -p profiles/car.lua

- Запускаем команду для создания иерархии графа дорог(занимает около часа)

osrm-contract RU.osrm

Результатом будет создание 24-х файлов RU*.* общим размером около 5,5гб

4) Запуск службы(проверка работоспособности созданного графа)

- Запуск службы на порте 8484(по умолчанию порт 5000), -t=8 – количество потоков

osrm-routed -p 8484 -t 8 RU.osrm

• ждём сообщение в консоли, что служба готова к приёму запросов
[info] running and waiting for requests

- Проверяем, что служба работает с подготовленным графом дорог(указываем ip адрес, где запущена служба. 127.0.0.1 , если внутри линукса в firefox проверить)

<http://192.168.1.126:8484/route/v1/driving/30.348887,60.010632;30.32459,59.873786?overview=false&steps=true>

5) Сохраняем 24 сгенерированных файла для дальнейшего использования при обновлении графа у клиентов

9. Установка службы маршрутизации(с подготовленным графом и исходниками)

1. Подготовка инфраструктуры для службы маршрутизации

- Устанавливаем необходимые пакеты для компиляции исходников службы <https://github.com/Project-OSRM/osrm-backend/wiki/Building-on-Ubuntu>

```
sudo apt install build-essential git cmake pkg-config \
libbz2-dev libstxxl-dev libstxxl1v5 libxml2-dev \
libzip-dev libboost-all-dev lua5.2 liblua5.2-dev libtbb-dev
```

б) Сборка из исходников(Сборка проекта из исходных текстов длится около 10-15 минут(х64 – около 12 минут) <https://github.com/Project-OSRM/osrm-backend/wiki/Building-OSRM>

- Если папка osrm-backend существует и в ней установлена старая служба, то удаляем эту папку
- копируем архив osrm-backend-5.12.0.tar.gz в папку /home/websys53/GIS_WEB60/RoutingService и переходим в эту папку в консоли

```
cd /home/websys53/GIS_WEB60/RoutingService
```

- распаковываем архив и созданную папку переименовываем в osrm-backend

```
tar -xzf osrm-backend-5.12.0.tar.gz
mv osrm-backend-5.12.0 osrm-backend
```

в) Компиляция и установка в систему сервера маршрутизации

```
cd osrm-backend
mkdir -p build
cd build
cmake .. -DCMAKE_BUILD_TYPE=Release
cmake --build .
sudo cmake --build . --target install
```

- Возвращаемся в директорию службы маршрутизации

```
cd ../../RoadService
```

- Если файл подкачки уже существует (/home/websys53/GIS_WEB60/RoutingService/RoadServiceSwapFile), то нужно его удалить(либо не трогать. Предпочтительно не трогать)

```
sudo swapoff /home/websys53/GIS_WEB60/RoutingService/RoadServiceSwapFile
rm /home/websys53/GIS_WEB60/RoutingService/RoadServiceSwapFile
```

- Создаем файл подкачки

```
fallocate -l 20G /home/websys53/GIS_WEB60/RoutingService/RoadServiceSwapFile
chmod 600 /home/websys53/GIS_WEB60/RoutingService/RoadServiceSwapFile
mkswap /home/websys53/GIS_WEB60/RoutingService/RoadServiceSwapFile
sudo swapon /home/websys53/GIS_WEB60/RoutingService/RoadServiceSwapFile
```

- Дополнительно создаем файл в директории, где лежит программа экспорта, если его там нет

```
nano ../osrm-backend/build/.stxxl
disk=/tmp/stxxl,20G,syscall
```

- Копируем 24 файла, которые получились после osrm-extract, osrm-contract (около 6гб)
- ### 2. Запуск службы(проверка работоспособности созданного графа)
- Запуск службы на порте 8484(по умолчанию порт 5000), -t=8 – количество потоков

```
osrm-routed -p 8484 -t 8 RU.osrm
```

- ждем сообщение в консоли, что служба готова к приёму запросов [info] running and waiting for requests

- Проверяем, что служба работает с подготовленным графом дорог(указываем ip адрес, где запущена служба. 127.0.0.1 , если внутри линукса в firefox проверить)

```
http://192.168.1.126:8484/route/v1/driving/30.348887,60.010632;30.32459,59.873786?overview=false&steps=true
```

3. Настраиваем автозапуск в соответствии с п. 7.3

10. GDAL+настройка импорта/экспорта карт из административных задач через UTEService

1. Устанавливаем SWIG для компиляции:

```
sudo apt-get install swig 2.0
```

2. Компиляция и установка GDAL

- Кладем архив gdal-2.2.2.tar.gz в папку /home/websys53/GIS_WEB60/UTEService и запускаем команду распаковки

```
cd /home/websys53/GIS_WEB60/UTEService
tar -xzf gdal-2.2.2.tar.gz
```

- По рекомендациям из https://github.com/jealie/binaries_GDAL_Csharp/blob/master/compilationNotes.txt компилируем и устанавливаем gdal в папку build

```
cd gdal-2.2.2/
./configure --prefix /home/websys53/GIS_WEB60/UTEService/gdal-2.2.2/build --without-python CFLAGS="-fPIC"
make
make install
```

- Заходим в директорию gdal2.2.2/swig/csharp и делаем правки в файлах:

А) Файл gdal/Band.cs

```
### 17 public Band(IntPtr cPtr, bool cMemoryOwn, object parent) :
base(GdalPINVOKE.BandUpcast(cPtr), cMemoryOwn, parent) {
```

В

```
### 17 public Band(IntPtr cPtr, bool cMemoryOwn, object parent) :
base(GdalPINVOKE.Band_SWIGUpcast(cPtr), cMemoryOwn, parent) {
```

Б) Файл gdal/Dataset.cs

```
### 17 public Dataset(IntPtr cPtr, bool cMemoryOwn, object parent) :
base(GdalPINVOKE.DatasetUpcast(cPtr), cMemoryOwn, parent) {
```

В

```
### 17 public Dataset(IntPtr cPtr, bool cMemoryOwn, object parent) :
base(GdalPINVOKE.Dataset_SWIGUpcast(cPtr), cMemoryOwn, parent) {
```

В) Файл gdal/Driver.cs

```
### 17 public Driver(IntPtr cPtr, bool cMemoryOwn, object parent) :
base(GdalPINVOKE.DriverUpcast(cPtr), cMemoryOwn, parent) {
```

В

```
### 17 public Driver(IntPtr cPtr, bool cMemoryOwn, object parent) :
base(GdalPINVOKE.Driver_SWIGUpcast(cPtr), cMemoryOwn, parent) {
```

Г) Файлы ogr/OgrPINVOKE.cs ; ogr/OsrPINVOKE.cs ; osr/OsrPINVOKE.cs;
gdal/GdalPINVOKE.cs

удаляем лишнее определение конструкторов OgrPINVOKE, OsrPINVOKE, GdalPINVOKE
соответственно подобного вида

```
### 188 static OsrPINVOKE() {
### 189 }
```

- Всё готово для компиляции

```
cd /swig/csharp
make
```

После компиляции в корне gdal-2.2.2/swig/csharp появятся новые dll, exe, ...

- Теперь нужно создать ссылки на библиотеки, чтобы UTEService корректно их подхватил и заработал импорт/экспорт карт

```
ln -s /home/websys53/GIS_WEB60/UTEService/gdal-2.2.2/swig/csharp/ogr_csharp.dll
/home/websys53/GIS_WEB60/UTEService/Bin
ln -s /home/websys53/GIS_WEB60/UTEService/gdal-2.2.2/swig/csharp/osr_csharp.dll
/home/websys53/GIS_WEB60/UTEService/Bin
```

Если потребуется почистить и заново проверить компиляцию /swig/csharp необходимо:

```
make veryclean
make interface
```

делаем правки в .cs, описанные выше и компилируем

```
make
```

11. Установка ghostScript

ghostScript - пакет библиотек, необходимых для работы с графикой. Существует как в версии Windows так и в версии Linux. Используется для постраничного доступа к PDF-файлам в задаче электронного архива.

Для установки необходимо:

1. Выполнить проверку, что пакет не установлен, для этого выполнить команду **gs**.
2. Если в ответ на выдастся сообщение, что команда не найдена, то устанавливаем пакет командой: **sudo apt-get install ghostscript**.
3. После установки еще раз выполнить **gs** и убедиться, что сообщение, что пакет не установлен, больше не выдается.

12. Установка curl и генерация файлов постраничности.

libcurl3_7.52.1-5+deb9u9_i386.deb и **curl_7.52.1-5+deb9u9_i386.deb** - библиотеки для получения данных посредством протоколов SSL certificates, HTTP POST, HTTP PUT, FTP uploading, HTTP form based upload, proxies, HTTP/2, cookies, user+password authentication (Basic, Plain, Digest, CRAM-MD5, NTLM, Negotiate and Kerberos)

Установка curl необходима для генерации xml-файлов, которые используются для работы постраничности.

Для установки

1) кладем **libcurl3_7.52.1-5+deb9u9_i386.deb** и **curl_7.52.1-5+deb9u9_i386.deb** в домашнюю папку и выполняем команду

```
cd ~
sudo dpkg -i libcurl3_7.52.1-5+deb9u9_i386.deb
sudo dpkg -i curl_7.52.1-5+deb9u9_i386.deb
```

2) Запускаем батник по генерации файлов

```
bash /home/websys53/Tasks/generate_http_data.sh
```

Сгенерированные файлы будут помещены в папку /home/websys53/private/geo.