**一．PostgreSQL安装，操作数据库说明：**

**虚拟机账户密码：123456 ， root 密码：123456**

1. 编译前的准备：

**注意：如果使用的是老师直接给的镜像文件，就不用做这一部分工作了，因为镜像文件已经安装好了相应的编译环境。**

sudo yum install gcc

sudo yum install readline-devel

sudo yum install zlib-devel

1. 创建用户：

**注意：如果使用的是老师直接给的镜像文件，就不用创建用户了，因为已经给创建好了，密码是：123456，直接切换到 postgres 用户下后解压文件即可。**

* 添加用户：adduser postgres
* 修改密码：passwd postgres 输入2次密码

**以下所有操作都在postgres用户下进行，所以一定要切换到 postgres 用户**

su postgres

输入密码

1. 解压：

进入存放下载完成的源代码的目录：

cd /home/postgres/

tar -zxvf postgresql-10.6.tar.gz

**解压的时候需要保证源文件的所有者是当前操作用户，否则会报错。**

1. 进入目录，在终端上输入以下命令

* 首先创建data目录来存储数据库文件

mkdir -p /home/postgres/data(data目录创建在/home/postgres/data文件夹下)

* 改变data文件夹的权限，保证data文件夹的所有者是postgres

chmod 777 data

chown postgres data

* 安装编译数据库

首先进入到解压后的数据库源码的目录

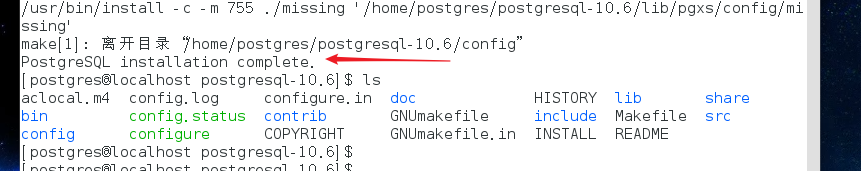
cd postgresql-10.6/

./configure --prefix=/home/postgres/postgresql-10.6

make

make install

查看postgresql目录,我的主目录位于/home/postgres/postgresql-10.6，如果有bin,config,lib,share几个目录表示编译成功



1. 创建postgresql数据库前的准备操作，创建数据库操作用户：

* 编辑~/.bash\_profile文件：vi /home/postgres/.bash\_profile
* 设置以下环境变量，将下面配置复制到文件的末尾：（这里要用到 vim ，不熟悉 vim 操作的自行上网查阅）

export PGHOME=/home/postgres/postgresql-10.6

export PGDATA=~/data

export PATH=$PATH:$HOME/bin:$PGHOME/bin

* 使环境变量生效：source ~/.bash\_profile

1. 数据库操作：

* 进入postgresql的主目录，我的主目录是 /home/postgres/postgresql-10.6

cd /home/postgres/postgresql-10.6

* 初始化postgres,出现如图的这表示初始化成功,初始化的命令为：

cd bin

./initdb -D /home/postgres/data

如果提示编码错误，可以使用 ：

./initdb -E UTF-8 -D /home/postgres/data --locale=zh\_CN.UTF-8

具体问题参考：http://blog.sina.com.cn/s/blog\_785b434d01010ekq.html

* 启动数据库服务器：./postgres -D /home/postgres/data
* 打开**新的终端**进入主目录(之前的终端不能关闭，新终端打开之后要注意切换到 postgres 用户)：

su postgres

之后输入密码

cd /home/postgres/postgresql-10.6/bin

* 执行命令创建数据库,名字为mydb：./createdb mydba
* 执行命令,连接到数据库mydb当中：./psql mydba

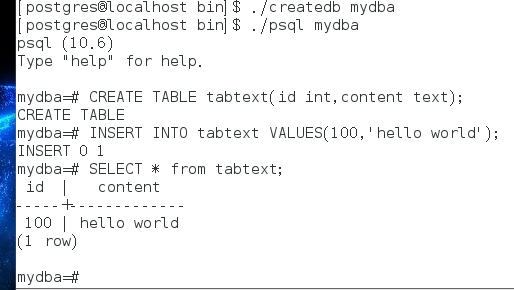
现在我们安装完成了，并且连接上了数据库，下面我们可以进行数据库中表的一些列操作了，我们现有的SQL语句，PG应该已经支持。

* 比我们可以简单的创建一个关系表并且完成查询。

CREATE TABLE tabtext(id int,content text);

INSERT INTO tabtext VALUES(100,'hello world');

SELECT \* from tabtext;



* 退出 psql 是输入 \q 之后按回车

**二．GDB安装**

1.解压：

cd /home/postgres/

tar -zxvf gdb-7.12.tar.gz

2.安装：

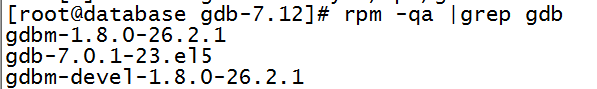
 用CD命令进入解压缩后的目录

 输入编译文件命令：./configure（有的压缩包已经编译过，这一步可以省去）

 然后是命令：make

 再是安装文件命令：sudo make install（install 时可能会有部分报错信息出现）

3.检查：rpm -qa |grep gdb，如果出现如下样式，则安装成功！



**三．自定义函数**

1. 编辑文件

(1)在主目录 /home/postgres/postgresql-10.6 下创建文件：

cd /home/postgres/postgresql-10.6

vim add\_func.c

(2)在创建的文件夹下添加内容：

#include "postgres.h"

#include "fmgr.h"

PG\_MODULE\_MAGIC;

PG\_FUNCTION\_INFO\_V1(add\_ab);

Datum

add\_ab(PG\_FUNCTION\_ARGS)

{

int32 arg\_a=PG\_GETARG\_INT32(0);

int32 arg\_b=PG\_GETARG\_INT32(1);

PG\_RETURN\_INT32(arg\_a+arg\_b);

}

(3)编辑 Makefile 文件

vim Makefile

在Makefile 文件末尾添加内容：

MODULES = add\_func

PG\_CONFIG = pg\_config

PGXS := $(shell $(PG\_CONFIG) --pgxs)

include $(PGXS)

1. 编译：

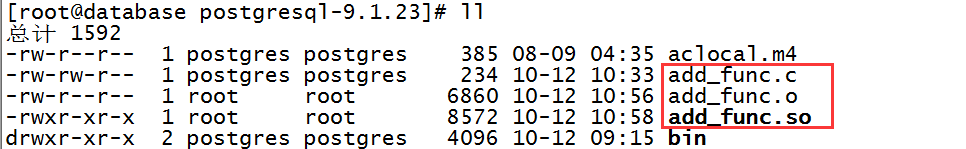
（1）编译 add\_func.c

gcc -O2 -Wall -Wmissing-prototypes -Wpointer-arith -Wdeclaration-after-statement -Wendif-labels -Wmissing-format-attribute -Wformat-security -fno-strict-aliasing -fwrapv -g -fpic -I. -I. -I/home/postgres/postgresql-10.6/include/server -I/home/postgres/postgresql-10.6/include/internal -D\_GNU\_SOURCE -c -o add\_func.o add\_func.c

（2）编译 add\_func.o

gcc -O2 -Wall -Wmissing-prototypes -Wpointer-arith -Wdeclaration-after-statement -Wendif-labels -Wmissing-format-attribute -Wformat-security -fno-strict-aliasing -fwrapv -g -fpic -I. -I. -I/home/postgres/postgresql-10.6/include/server -I/home/postgres/postgresql-10.6/include/internal -D\_GNU\_SOURCE --enable-new-dtags -shared -o add\_func.so add\_func.o

(3)查看文件夹下的文件：ll，有如下图所示的三个文件



(4)添加到lib目录下：cp add\_func.so lib/

(5)启动数据库服务器：./psql -d postgres

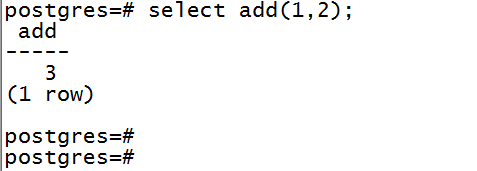
1. 测试：

(1)创建函数如下：

CREATE FUNCTION add(int,int) RETURNS int AS '/home/postgres/postgresql-10.6/lib/add\_func', 'add\_ab' LANGUAGE C STRICT;

(2)检验创建的函数：

select add(1,2);



参考自：<http://blog.sina.com.cn/s/blog_67d069a90102v3ov.html>

**四．自定义类型：**

（1）修改Makefile文件的内容：

cd /home/postgres/postgresql-10.6

vim Makefile

将之前自定义函数时添加的部分**删除**之后，在文件的末尾添加下面的内容并保存：

PG\_MODULE\_MAGIC = complex

PG\_CONFIG = pg\_config

PGXS := $(shell $(PG\_CONFIG) --pgxs)

include $(PGXS)

（2）编译部分：

自定义类型用到的源文件complex.c 已经在源码包中提供，所以直接使用：

cd /home/postgres/postgresql-10.6/src/tutorial

下面开始编译：

gcc -O2 -Wall -Wmissing-prototypes -Wpointer-arith -Wdeclaration-after-statement -Wendif-labels -Wmissing-format-attribute -Wformat-security -fno-strict-aliasing -fwrapv -g -fpic -I. -I. -I/home/postgres/postgresql-10.6/include/server -I/home/postgres/postgresql-10.6/include/internal -D\_GNU\_SOURCE -c -o complex.o complex.c

gcc -O2 -Wall -Wmissing-prototypes -Wpointer-arith -Wdeclaration-after-statement -Wendif-labels -Wmissing-format-attribute -Wformat-security -fno-strict-aliasing -fwrapv -g -fpic -I. -I. -I/home/postgres/postgresql-10.6/include/server -I/home/postgres/postgresql-10.6/include/internal -D\_GNU\_SOURCE --enable-new-dtags -shared -o complex.so complex.o

（3）添加编译后的文件到lib目录下：

cp complex.so /home/postgres/postgresql-10.6/lib/

（4）启动数据库服务器：

cd /home/postgres/postgresql-10.6/bin/

启动服务器：./postgres -D /home/postgres/data （注意启动之后不能关闭当前终端，要再开一个终端，切换到 postgres 用户之后再进入到该目录下之后执行下边的命令）

进入 sql shell：./psql -d postgres

（5）测试：

下面的几个命令一起复制到终端运行即可：

CREATE FUNCTION complex\_in(cstring) RETURNS complex AS ' /home/postgres/postgresql-10.6/lib/complex' LANGUAGE C IMMUTABLE STRICT;

CREATE FUNCTION complex\_out(complex) RETURNS cstring AS '/home/postgres/postgresql-10.6/lib/complex' LANGUAGE C IMMUTABLE STRICT;

CREATE FUNCTION complex\_recv(internal) RETURNS complex AS '/home/postgres/postgresql-10.6/lib/complex' LANGUAGE C IMMUTABLE STRICT;

CREATE FUNCTION complex\_send(complex) RETURNS bytea AS '/home/postgres/postgresql-10.6/lib/complex' LANGUAGE C IMMUTABLE STRICT;

CREATE TYPE complex ( internallength = 16, input = complex\_in, output = complex\_out, receive = complex\_recv, send = complex\_send, alignment = double);

我们已经成功创建好了自定义的类型 complex，下面来验证一下：

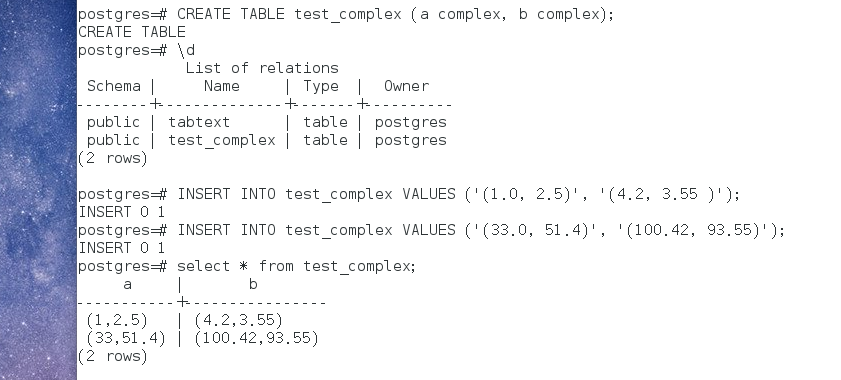
CREATE TABLE test\_complex (a complex, b complex);

INSERT INTO test\_complex VALUES ('(1.0, 2.5)', '(4.2, 3.55 )');

INSERT INTO test\_complex VALUES ('(33.0, 51.4)', '(100.42, 93.55)');

SELECT \* FROM test\_complex;

查询结果：



更多测试内容查看complex.source文件

附录：

1. 如何让 visualbox 与主机之间共享剪切板？

参见博文：https://www.cnblogs.com/ayseeing/p/4287571.html