Pktgen模块：

Sys.argv[1] 即此程序跟着的参数，假如此程序后面跟这的参数

Test.py what 输出的就是what

Sys.arg[2: ] test.py a b c d e f ================输出b c d e f

Pktgen命令:

Modprobe pktgen //加载模块

Rmmod pktgen //卸载模块

在一个多核系统中，此路径目录下回自动创建多个文件，

路径 =/proc/net/pktgen/

kpktgend\_0、kpktgend\_1和pgctrl kpktgend\_x即有几个核就多少

通过cat + 路径 即可查看此路径下的网口数据

Start 所有线程开始发送

Stop 停止

线程命令 :

add\_device 添加某个端口到线程

Rem\_device\_all 删除绑定在某个线程的所有端口

Max\_before\_softirp 在最多发送多少个数据包后，执行do\_softirp()

端口命令: 在add\_device之后，执行相关的命令

通过shell设置下面的值，即可执行

Debug 调试

clone\_skb 对每个skb复制，0表示不复制

Clear\_counters 清空计数器，自动清空

Pkt\_size 链路包的大小

Min\_pkt\_size 数据包最小值

Max\_pkt\_size 最大值

flags 包的分片数量

Count 发送数据包的个数，count 0表示一直发送

Delay 发送两个数据包之间的延时

Dst 目的ip

Dst\_min 目的ip最小值 dst\_max 目的ip最大值

Src\_min 源ip最小值 src\_max 源ip最大值

Dst6 目的ipv6地址 src6源ipv6地址

Dstmac 目的mac srcmac 源mac

系统运行步骤：

-n –nico 运行两个参数

运行步骤：sudo su

输入账号密码

**测试环境工作流程：**

**切换到根目录，cp –r a/ b/ 将目录下面的程序全部复制到另外一个目录**

**Cp config.ini.example config.ini 复制一个完全相同的文件**

**执行 python3 linux\_prd.py –t –n**

**如果不需要的话，rm -rf 文件或者文件夹**

Awk获取指定列：ls -L | awk ‘{print $1} ’获取指定列 { }中填写的是需要的参数，此命令获取第一列 ，$1

输出第1列非字母d开头的行

Awk ‘{ hello}’ 1.txt 逐行读取1.txt文件内容

每读取一行打印一个hello

查看网口命令：

cmd = "ethtool %s | grep Speed | awk '{print $2}'" % eth

查看某个网口的速度

加载命令，cmd = ‘modprobe pktgen’

elif self.cpu\_type == cu.CPU\_TYPE\_FT64:  
 cmd\_group\_1 = 'bash /root/tools/pktgen.conf-2-2 -a %s -b %s ' \  
 '-c %s -d 4 -e 5 -r 100 > /dev/null' % (eth\_list[0], eth\_list[1], packets)  
 cmd\_group\_2 = 'bash /root/tools/pktgen.conf-2-2 -a %s -b %s ' \  
 '-c %s -d 4 -e 5 -r 100 > /dev/null' % (eth\_list[2], eth\_list[3], packets)

文件的地址在此目录下：

cat /proc/net/pktgen/

查看网口lsmod | grep pktgen

查看启动的网口数量

设置串口服务器的ip地址的方法：

Ifconfig 查看网口



Route add default gw 10.3.0.1 ip地址需要自己设置

/etc/init.d/network-manager stop

加载驱动：操作步骤

Sh select\_board\_type.sh

.env.sh

Sh show\_cyclone\_version.sh

对网口数据进行设置：

*/etc/init.d/network-manager stop*

*ifconfig enp5s0f0 10.3.22.33 netmask 255.255.0.0*

*route add default gw 10.3.0.1*

*Select\_board\_type.sh //选择板卡类型*

*ifconfig enp5s0f0 10.10.120.0 netmask 255.255.0.0*

*route add default gw 10.0.0.1*

数据库操作：

DDL 创建数据库用的语句

DCL 控制用户的访问权限，以及安全级别

DML 用来操作数据库，对其中的数据进行更新，insert, update

DQL 查询数据库中的表的记录，select from

增：insert into 表（列名1，列名2，列名3..）values(值1，值2，值3..);

INSERT INTO product (pname,price) VALUE('洗衣机',800)

删：delete 表名

改：update 列 from 表 + 处理函数

查：

DDL：\* A: 创建用户表: 需求:创建用户表,用户编号,姓名,用户的地址 \* B: SQL语句 CREAT TABLE users ( uid INT, uname VARCHAR(20), uaddress VARCHAR(200) );

主键的约束作用，标识当前记录的字段，在相应字段后面加上

Primary key；primary key(id)

表中添加数据set 键名=’相应的赋值’

Mysql 使用的不等号 <>

高级操作：两表联查，左右关联，全关联

排序--->根据组排序，根据某个关键字进行排序

Group by 根据某个关键字或者词进行分组

Order by 根据某个值进行排序

order by 列名 asc/desc asc降序，desc升序

例子：

在表person中查找相同的某个值，并将其展示

select Email from Person group by Email having count(Email) > 1;

解决先排序后分组的问题：子查询

SELECT \* FROM (SELECT \* FROM posts ORDER BY dateline DESC) as testbiao GROUP BY tid ORDER BY dateline DESC LIMIT 10

----------------------------------------------------------------------------------------------------

SELECT \* from posts a where datline in (SELECT max(dateline) from posts b where a.tid=b.tid) LIMIT 10

---------------------------------------------------------------------------

update salary set sex=IF(sex='f','m','f')

更新表中数据，当sex为f时 变成M，否则依旧使用f

在mysql中，对于单列来说，or使用是没有问题的，但是or涉及多个列的时候，每次select 只能选取一个index，如果选择两列或者两列以上的话，需要全表扫描，可能会浪费时间

但是union会进行排序以及去重；

Union和 or 的意思都是或，两个都是可以进行选择的

Mod（）函数，可以用来去余数,mod(id, 2) = 1

取得余数

DISTINCT 去重

Select distinct 列名称 from 表

在 SQL 中增加 HAVING 子句原因是，WHERE 关键字无法与聚合函数一起使用。

HAVING 子句可以让我们筛选分组后的各组数据。

聚合函数可以直接对整组数据进行处理，但是where 只能处理逐条数据

聚合函数对一组值执行计算，并返回单个值。 除了 COUNT(\*) 外，聚合函数都会忽略 Null 值。 聚合函数经常与 SELECT 语句的 GROUP BY 子句一起使用。

CASE函数：

两种形式，一种简单的case函数，另一种是case搜索函数



else一定要加，原有值，否则所有的没有在when上面的话，会变成null值

select 选择项目 from 表名 设定条件，进行排序

Limit x,y 参数，x 为跳过x条数据，y为取y条数据

即为跳过前x条，从x+1开始取y条数据

Limit x 只有一个参数的话，那么为取x条数据

Limit x offset y 即为取出x条数据，并且跳过第y条数据

选择这张表，索引为子查询

SELECT \* FROM table WHERE id >=(SELECT id FROM table LIMIT 1000000,1) LIMIT 10;

查询这张表，取出其中id大于1000001的所有数据

数据表名字的简写和复制

Select e1 from elmp el

将elmp 简写成el



选择目标表名，并且将对应的值赋予对应的简写值；

对表名进行赋值的过程，

Sql赋值机制：

From 表名 x1，表名 x2；

此机制将对应的表用x1,x2代替，x1.id 即 为目标表的id

select \* from table1 a, table2 b

简写方式 条件可以写成where a.id = b.id