2015/1/5

IP加到/etc/hosts

mpirun --version

cd openmpi-1.8.3

cat /proc/cpuinfo

cat /proc/meminfo 是看内存

tar zxvf openmpi-\*\*\*

rpm -ivh \*\*\*.rpm(解压rpm文件）

./configure --prefix=/home/test/openmpi configure是配置生成Makefile

make是编译，生成可执行文件

make install是把程序装到你指定的目录下

chown -R test:test \*

从源文件安装就是 ./configure ; make; make install 三步

yum -y groupinstall "Development Tools"

/home/test/openmpi/bin就是可执行文件 mpicc mpirun的地方

/home/test/openmpi/lib 就是所有调用的MPI库的文件地方

这两个要加在.bashrc里，代替原来yum装的mpi

mpirun -np 2 hostname

mpirun -np 2 -host zhuangdizhu1,zhuangdizhu2 hostname

mpirun -np 2 -host zhuangdizhu1 ./transfer1

iptables on  
如闭防火墙则输入：  
chkconfig iptables off  
上述两条命令均要重启系统才能生效。  
如果不想通过重启系统而即时生效的话，可以用“service”命令。缺点是重启系统后设置会丢失。  
开启了防火墙：  
service iptables start  
关闭防火墙：  
service iptables stop

2015/1/14

可以使用rm删除目录中的文件后，使用rmdir删除目录。也可以使用rm -rf替代rmdir命令。

显示主机IP  
# hostname -i

显示主机名：

# hostname

 find / -name " "显示文件所在目录

rpm -qa | grep kernel查找kernel版本

yum remove kernel-devel-2.6.32-358.el6.i686

首先你要确定的是你的Centos有没有正确识别你的网卡，lsmod看看网卡驱动模块有没有正确加载或ifconfig eth0看没有该设备，如果有，vi编辑该配置文件（/etc/sysconfig/network-scripts/ifcfg-eth0）,把BOOTPROTO改为dhcp,ONBOOT=yes,保存退出，重启网络服务（service network restart）即可。也可用setup向导方式来设置，但这种方式仍需要修改ONBOOT选项；另一种情况就是没有驱动，相信你知道怎么处理了吧？安装驱动，后面的步骤差不多。

2015/01/17

ibv\_devices

lsmod | grep rdma

ibv\_devinfo

dmesg | grep eth

lsmod | grep mlx

ifconfig ib0 172.16.0.2 up

ficonfig ib0 172.16.0.3 up

ifconfig

iptables -L

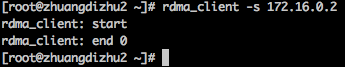
[15/01/2015 21:10:31] Michael Hines: server: ibv\_rc\_pingpong

[15/01/2015 21:10:36] Michael Hines: client ibv\_rc\_pingpong IPADDRESS

rdma\_server

rdma\_client -s IPADRESS





rping -s

rping -a IPADDRESS\_OF\_SERVER -d -v -V -c

[15/01/2015 21:29:35] salman\_baset: cat /proc/meminfo

[15/01/2015 21:29:39] salman\_baset: cat /proc/cpuinfo

rping -s -d

[15/01/2015 21:33:28] Michael Hines: ctrl-Z

[15/01/2015 21:33:30] Michael Hines: bg

[15/01/2015 21:34:28] Michael Hines: ctrl-C

[15/01/2015 21:34:31] Michael Hines: fg

[15/01/2015 21:34:34] Michael Hines: ctrl-C

cat /etc/selinux/config

2015/01/18

change locked memory limits

ulimit -l unlimited

dd if=/dev/zero of=/tem/test bs=1M count=1024 此命令在/tem/下建立一个名为test的1g的文件

#include <time.h>

clock\_t begin,end;

double time\_spent;

begin=clocka();

end=clock();

time=(double)(end-begin)/CLOCKS\_PER\_SEC;

[19/01/2015 15:51:04] Lei: struct timespec t1, t2;

double diff;

...

clock\_gettime(CLOCK\_REALTIME, &t1);

rc\_client\_loop(argv[1], DEFAULT\_PORT, &ctx);

clock\_gettime(CLOCK\_REALTIME, &t2);

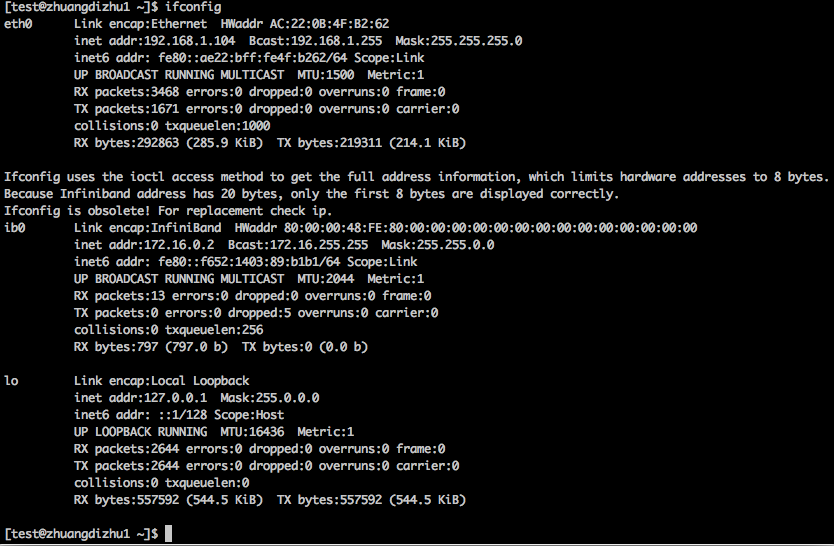
diff = (double) ((t2.tv\_sec-t1.tv\_sec) + (t2.tv\_nsec-t1.tv\_nsec)) / 1e9;

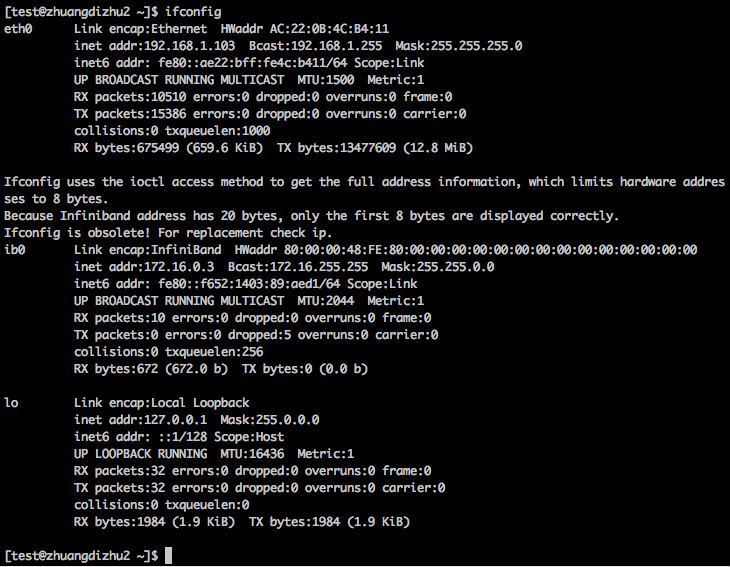
printf("%lf\n", diff);

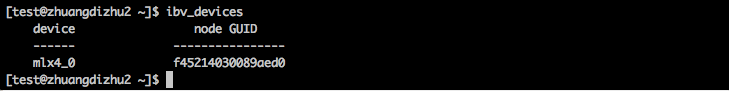
[19/01/2015 15:51:50] Lei: 然后 Makefile里

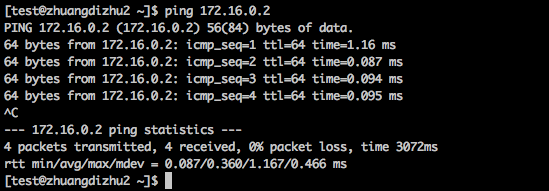
LDFLAGS := ${LDFLAGS} -lrdmacm -libverbs -lrt

[19/01/2015 15:51:58] Lei: 加了-lrt









(这个是ib0的IP 可以ping通)

[15/01/2015 21:10:31] Michael Hines: server: ibv\_rc\_pingpong

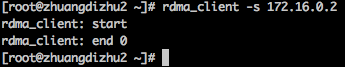
[15/01/2015 21:10:36] Michael Hines: client ibv\_rc\_pingpong IPADDRESS

（这个成功了我就不粘贴了）

rdma\_server

rdma\_client -s IPADRESS





2015/01/20

#vim /etc/hosts

#free

ud\_lat

UD one way latency

udp\_lat

UDP one way latency

udp\_bw

#rpm -qa | grep qperf

qperf-0.4.9-1.x86\_64

qperf-debuginfo-0.4.9-1.x86\_64

-i, --id Device:Port

Use RDMA Device and Port.

server:#qperf

client:# qperf -i mlx4\_0:1 172.16.1.2 ud\_bw

# ulimit -l unlimited

#lspci

[20/01/2015 21:12:30] salman\_baset: on server, type

[20/01/2015 21:12:33] salman\_baset: qperf

[20/01/2015 21:12:36] salman\_baset: and on client

[20/01/2015 21:12:42] salman\_baset: qperf -sr 172.16.1.2 ud\_bw

[20/01/2015 21:13:02] salman\_baset: qperf -sr 10 172.16.1.2 ud\_bw

[20/01/2015 21:17:43] salman\_baset: on server, do the following

[20/01/2015 21:17:44] salman\_baset: qperf

[20/01/2015 21:17:46] salman\_baset: on client

[20/01/2015 21:18:04] salman\_baset: qperf -lsr 10 -rsr 10 172.16.1.2 ud\_bw

[20/01/2015 21:24:21] salman\_baset: on tcp, we may want to check the impact of tcp rmem, wmem, and buffer size

[20/01/2015 21:24:24] salman\_baset: and jumbo frames etc

[20/01/2015 21:24:43] salman\_baset: we will like to sure that "packet size" is same for tcp and rdma

[20/01/2015 21:25:05] salman\_baset: we need to find the block size for rdma

man qperf

To run a TCP bandwidth and latency test:

qperf myserver tcp\_bw tcp\_lat

To run a SDP bandwidth test for 10 seconds:

qperf myserver -t 10 sdp\_bw

To run a UDP latency test and then cause the server to terminate:

qperf myserver udp\_lat quit

To measure the RDMA UD latency and bandwidth:

qperf myserver ud\_lat ud\_bw

To measure RDMA UC bi-directional bandwidth:

qperf myserver rc\_bi\_bw

To get a range of TCP latencies with a message size from 1 to 64K

qperf myserver -oo msg\_size:1:64K:\*2 -vu tcp\_lat

-i, --id Device:Port

Use RDMA Device and Port.

-li, --loc\_id Device:Port

Use local RDMA Device and Port.

-f, --flip OnOff

If non-zero, cause sender and receiver to play opposite roles.

-f1 Cause sender and receiver to play opposite roles.

-li, --loc\_id Device:Port

Use local RDMA Device and Port.

-ri, --rem\_id Device:Port

Use remote RDMA Device and Port.

-m, --msg\_size Size

Set the message size to Size. The default value varies by test. It is assumed that the value is specified in

bytes however, a trailing kib or K, mib or M, or gib or G indicates that the size is being specified in

kibibytes, mebibytes or gibibytes respectively while a trailing kb or k, mb or m, or gb or g indicates kilobytes,

megabytes or gigabytes respectively.

-mt, --mtu\_size Size

qperf myserver -oo msg\_size:1:64K:\*2 -vu tcp\_lat

qperf myserver -oo mtu\_size:1：

Set the MTU size. Only relevant to the RDMA UC/RC tests. Units are specified in the same manner as the

--msg\_size option.

Miscellaneous

conf Show configuration

quit Cause the server to quit

Socket Based

rds\_bw RDS streaming one way bandwidth

rds\_lat RDS one way latency

sctp\_bw SCTP streaming one way bandwidth

sctp\_lat SCTP one way latency

sdp\_bw SDP streaming one way bandwidth

sdp\_lat SDP one way latency

tcp\_bw TCP streaming one way bandwidth

tcp\_lat TCP one way latency

udp\_bw UDP streaming one way bandwidth

udp\_lat UDP one way latency

RDMA Send/Receive

rc\_bi\_bw RC streaming two way bandwidth

rc\_bw RC streaming one way bandwidth

rc\_lat RC one way latency

uc\_bi\_bw UC streaming two way bandwidth

uc\_bw UC streaming one way bandwidth

uc\_lat UC one way latency

ud\_bi\_bw UD streaming two way bandwidth

ud\_bw UD streaming one way bandwidth

ud\_lat UD one way latency

xrc\_bi\_bw XRC streaming two way bandwidth

xrc\_bw XRC streaming one way bandwidth

xrc\_lat XRC one way latency

RDMA

rc\_rdma\_read\_bw RC RDMA read streaming one way bandwidth

rc\_rdma\_read\_lat RC RDMA read one way latency

rc\_rdma\_write\_bw RC RDMA write streaming one way bandwidth

rc\_rdma\_write\_lat RC RDMA write one way latency

rc\_rdma\_write\_poll\_lat RC RDMA write one way polling latency

uc\_rdma\_write\_bw UC RDMA write streaming one way bandwidth

uc\_rdma\_write\_lat UC RDMA write one way latency

uc\_rdma\_write\_poll\_lat UC RDMA write one way polling latency

InfiniBand Atomics

rc\_compare\_swap\_mr RC compare and swap messaging rate

rc\_fetch\_add\_mr RC fetch and add messaging rate

Verification

ver\_rc\_compare\_swap Verify RC compare and swap

ver\_rc\_fetch\_add Verify RC fetch and add

NAME

sysctl - configure kernel parameters at runtime

SYNOPSIS

sysctl [-n] [-e] variable ...

sysctl [-n] [-e] [-q] -w variable=value ...

sysctl [-n] [-e] [-q] -p <filename>

sysctl [-n] [-e] -a

sysctl [-n] [-e] -A

DESCRIPTION

sysctl is used to modify kernel parameters at runtime. The parameters available are those listed under /proc/sys/.

Procfs is required for sysctl(8) support in Linux. You can use sysctl(8) to both read and write sysctl data.

PARAMETERS

variable

The name of a key to read from. An example is kernel.ostype. The ’/’ separator is also accepted in place of a

’.’.

variable=value

To set a key, use the form variable=value, where variable is the key and value is the value to set it to. If the

value contains quotes or characters which are parsed by the shell, you may need to enclose the value in double

quotes. This requires the -w parameter to use.

-n Use this option to disable printing of the key name when printing values.

-e Use this option to ignore errors about unknown keys.

-N Use this option to only print the names. It may be useful with shells that have programmable completion.

-q Use this option to not display the values set to stdout.

-w Use this option when you want to change a sysctl setting.

-p Load in sysctl settings from the file specified or /etc/sysctl.conf if none given. Specifying - as filename

means reading data from standard input.

-a Display all values currently available.

-A Same as -a

EXAMPLES

/sbin/sysctl -a

/sbin/sysctl -n kernel.hostname

/sbin/sysctl -w kernel.domainname="example.com"

/sbin/sysctl -p /etc/sysctl.conf

NOTES

Please note that modules loaded after sysctl is run may override the settings (example: sunrpc.\* settings are overridden

when the sunrpc module is loaded). This may cause some confusion during boot when the settings in sysctl.conf may be

overriden. To prevent such a situation, sysctl must be run after the particular module is loaded (e.g., from

/etc/rc.d/rc.local or by using the install directive in modprobe.conf)

FILES

/proc/sys /etc/sysctl.conf

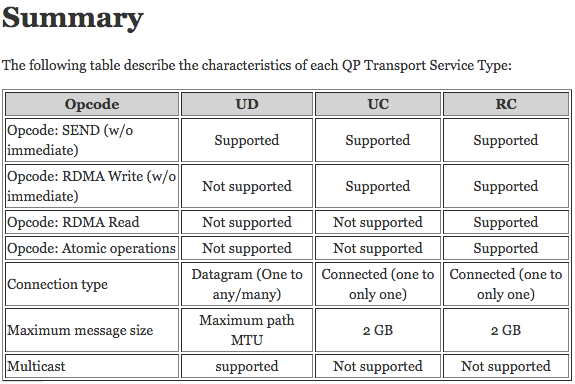
SEE ALSO

sysctl.conf(5), modprobe.conf(5)

AUTHOR

George Staikos, <staikos@0wned.org>

21 Sep 1999 SYSCTL(8)



2015/01/21

***/etc/sysctl.conf***

|  |
| --- |
| # allow testing with buffers up to 64MB  net.core.rmem\_max = 67108864  net.core.wmem\_max = 67108864  # increase Linux autotuning TCP buffer limit to 32MB net.ipv4.tcp\_rmem = 4096 87380 33554432 net.ipv4.tcp\_wmem = 4096 65536 33554432 # increase the length of the processor input queue net.core.netdev\_max\_backlog = 30000 # recommended default congestion control is htcp  net.ipv4.tcp\_congestion\_control=htcp # recommended for hosts with jumbo frames enabled net.ipv4.tcp\_mtu\_probing=1 |
| Also add this to /etc/rc.local (where N is the number for your 10G NIC):  /sbin/ifconfig ethN txqueuelen 10000 |

用此方法修改/proc/sys/net/目录下文件中的内容在系统重新启动后，所设置的内容会全部变为默认值，因此，如果要想设置的值永久有效，可以直接把这个命令加入到/ect/rc.d/rc.local文件中

SR-IOV:Single Root I/O virtualization

Single Root I/O Virtualization (SR-IOV) is a PCI feature which allows virtual functions (VF) to be created that share the resources of a physical function (PF).

SR-IOV is primarily useful in allowing a single PCI device to be shared amongst multiple [virtual machines](http://fedoraproject.org/wiki/Virtualization).

SR-IOV support is implemented in the kernel.

Full virtualization vs. paravirtualization

QEMU is a generic and open source machine emulator and virtualizer.

When used as a machine emulator, QEMU can run OSes and programs made for one machine (e.g. an ARM board) on a different machine (e.g. your own PC). By using dynamic translation, it achieves very good performance.

2015/01/24

查看是否使用Selinux:

#getenforce

查看硬盘

#df -h

安装KVM

#yum install kvm

#yum install virt-manager libvirt libvirt-python python-virtinst

Insert KVM module

#modprobe kvm-intel

Start the libvirtd daemon service:

#/etc/init.d/libvirtd start

2015/01/25

#ibv\_devices

#ibv\_devinfo

#dmesg | grep eth

#lsmod | grep mlx

#service NetworkManager stop

#history

# dmesg | grep eth

# service opensm restart

# dmesg | grep mlx

2015/1/25

#dmidecode -t baseboard

# dmidecode -t bios

#dmesg | grep mlx

[23/01/2015 10:02:42 am] Lei: 我今天早晨想你去on-site实习之前，你最好恶补一些Linux和编程的基本功，因为毕竟on-site，我没法时事帮你，而如果某些基础性的东西你没用过或没听过，可能不太好

[23/01/2015 10:03:36 am] Lei: 最起码你要知道是什么，大概怎么用，人家说起来你可以答得上来，最好能做简单评论或者最基本使用，然后再看文档查资料看具体怎么用

[23/01/2015 10:08:58 am] Lei: 我给你列个List，你看看

[23/01/2015 10:17:30 am] Lei: 第一个，我想到的是版本管理，你可能之前都没接触过，一般我们都用git，当然也有不少项目还用svn，我建议你最起码了解一下git的基本使用，包括github(不知道国内墙了没有)。

一般企业编程都需要每次把修改git push进repository，然后下次再pull出来，保证改错能轻易回到之前版本，也方便多人协同在一个项目工作。

http://rogerdudler.github.io/git-guide/ 这个是一个快速入门；

[23/01/2015 10:25:08 am] Lei: 第二个，vim将可能是你长期的编辑器，需要掌握基本的删除，复制，粘贴，批量替换，分屏，设置行号，缩进等常用功能，

我找了个还可以的资料 http://blog.csdn.net/wangdingqiaoit/article/details/7757386

其他可以自己搜。

[23/01/2015 10:33:13 am] Lei: 第三个，make和Makefile编译，从源代码安装软件的configure，make, make install，基本的gcc link, -L -I -l 等的使用，

Makefile的入门 http://david-je.iteye.com/blog/1877250

Linux下compile的基本教程: http://luv.asn.au/overheads/compile.html

[23/01/2015 10:35:56 am] Lei: 第四个，Linux下Bash的基本使用，学会基本的 ls/chmod/chown/rm/find/ln/cat/mount/mkdir/tar/gzip/man 基本命令，会用基本的管理命令 ps/top/lsof/netstat/kill/tcpdump/iptables/dd 了解 /etc 下一些配置文件以及 /proc 下系统运行信息；能够用bash编写小程序，完成比较复杂的功能

[23/01/2015 10:41:30 am] Lei: 第五个，regular expression正则表达式，这个理论很复杂，但基本使用很简单，学会用 grep， awk和sed做基本的文件文字处理，知道怎么查找，组织文本文件。一些资源如下：

http://tldp.org/LDP/abs/html/x17129.html

http://www.opensourceforu.com/2012/06/beginners-guide-gnu-grep-basics-regular-expressions/

http://awk.readthedocs.org/en/latest/chapter-one.html

自己可以专门搜

[23/01/2015 10:45:30 am] Lei: 第六个，gdb debug程序，以后调试的程序会越来越大，不像自己几十行小程序，所以还是要接触一下gdb调试，当然有更多更好的专用debugger，但大同小异，gdb是个基础，有时间接触一下，别人提起要知道。有需要专门花时间学习。

2015/1/27

docker pull

docker images

docker ps -a

docker log <imageID>

docker version

docker commit -m="Added json gem" -a="Kate Smith" \ 0b2616b0e5a8 ouruser/sinatra:v2

[27/01/2015 19:56:01] Michael Hines: docker ps -a | grep -v CONTAINER | cut -d " " -f 1 | while read ID ; do echo "Delete this container: $ID" ; done

[27/01/2015 19:56:03] Michael Hines: $ docker ps -a | grep -v CONTAINER | cut -d " " -f 1 | while read ID ; do echo "Delete this container: $ID" ; done

docker save jude1 > jude1.tar

docker load < file\_you\_sent.tar

docker images -a

2015/01/28

docker inspect -f "{{ .Name }}" aed84ee21bde

2015/2/1

Communication between containers and the wider world

$ cat /proc/sys/net/ipv4/ip\_forward

0

$ sudo echo 1 > /proc/sys/net/ipv4/ip\_forward

$ cat /proc/sys/net/ipv4/ip\_forward

1

root@5cf459a63eea:~/qperf-0.4.9# cat README

To build

./cleanup

./autogen.sh

./configure

make

Changing version

\* src/qperf.c: Change VER\_MAJ, VER\_MIN and VER\_INC.

\* configure.in: Change in AC\_INIT and AM\_INIT\_AUTOMAKE

\* qperf.spec: Change line beginning with Version:

\* Note ensure that qperf.spec is modified last so that cleanup does not

delete it.

Notes

\* If the library ibverbs is not found, a version of qperf is built that

does not support the RDMA tests.

\* Running "make clean" does not seem to clean up everything. Run

"./cleanup" instead.

[root@5cf459a63eea:~/qperf-0.4.9#](mailto:root@5cf459a63eea:~/qperf-0.4.9)

2015/2/1

git fetch --all

git reset --hard origin/master

virsh nodedev-reattach pci\_0000\_02\_00\_0

virsh nodedev-dettach

virsh nodedev-list --tree

virsh nodedev-list | grep pci

: ctrl-Z

: bg

：ctrl-C

: fg

: ctrl-C

一般计算机主板上里都有个芯片是 MMU（Memory Management Unit），他是给CPU来操纵内存的，IOMMU就是跟这个很像，但是提供给PCI的设备，比如IB网卡来直接操纵内存的

Force InfiniBand static rate. Rate can be one of: 2.5, 5, 10, 20, 30, 40, 60, 80, 120, 1xSDR (2.5 Gbps), 1xDDR (5 Gbps), 1xQDR (10 Gbps), 4xSDR (2.5 Gbps), 4xDDR (5 Gbps), 4xQDR (10 Gbps), 8xSDR (2.5 Gbps), 8xDDR (5 Gbps), 8xQDR (10 Gbps).

2015/03/31

$(CC) -o server server.o $(LIBS) -luuid

uuid cannot found

-- yum install uuid uuid-devel

-- cd /usr/include; mkdir uuid; cp uuid.h uuid/

-- yum install e2fsprogs-devel

-- yum install libuuid libuuid-devel

-- add -luuid in Makefile

[31/03/2015 11:16:15] shangjob: 假设一个文件叫 aaa.c

[31/03/2015 11:16:45] shangjob: 它里面调用了 libzhuangdizhu.so 这个库文件

[31/03/2015 11:16:58] shangjob: 当然也引用了 zhuangdizhu.h 这个头文件

[31/03/2015 11:17:17] shangjob: 并且这两个文件不在系统文件夹，也不在aaa.c所在目录

[31/03/2015 11:17:21] shangjob: 那怎么办呢？

[31/03/2015 11:18:06] 庄翟 朱: 而且 如果没有Makefile 就是普通的 gcc 编译 又怎么办呢 现在是改Makefile

[31/03/2015 11:18:39] shangjob: 首先，**gcc -c aaa.c -o aaa.o -I /home/abcd/**，把C源文件编译成 .o 的object文件，其中 /home/abcd是zhuangdizhu.h 所在的目录

[31/03/2015 11:19:05] 庄翟 朱: wow

[31/03/2015 11:19:09] 庄翟 朱: then？

[31/03/2015 11:19:44] shangjob: 然后，**gcc aaa.o -o aaa -lzhuangdizhu -L /home/abcdef** ，把这个object文件链接上库文件，其中 /home/abcdef 是 libzhuangdizhu.so 所在位置

[31/03/2015 11:20:23] shangjob: 任何的C程序都是经过这两个步骤编译完成，最终生成 aaa 这个binary二进制文件的

[31/03/2015 11:20:32] 庄翟 朱: expected specifier-qualifier-list before ‘uuid\_t’

[31/03/2015 11:21:44] shangjob: 只不过，如果所需要的头文件 库文件，可能正好在aaa.c所在目录，或者系统默认的查找目录，比如 /usr/include /usr/lib64 之类的，就免加了

[31/03/2015 11:21:54] shangjob: 先把这点知识记下来，

[31/03/2015 11:21:58] 庄翟 朱: 第一步的 -l 是什么意思

[31/03/2015 11:21:59] shangjob: 别忘了，很重要的

[31/03/2015 11:22:07] shangjob: Include 的首字母啊