

动态编译

什么叫动态编译？

动态编译是将应用程序需要的模块都编译成动态链接库，启动程序（初始化）时，这些模块不会被加载，运行时用到哪个模块就调用哪个。动态编译将.c文件编译成.o或.a或.lib格式的动态链接库。

.o .obj : object files, they contain the output of the compiler generated code. It is still in an intermediate format, for example, most references are still unresolved. Usually there is a one to one mapping between the source file and the object file.

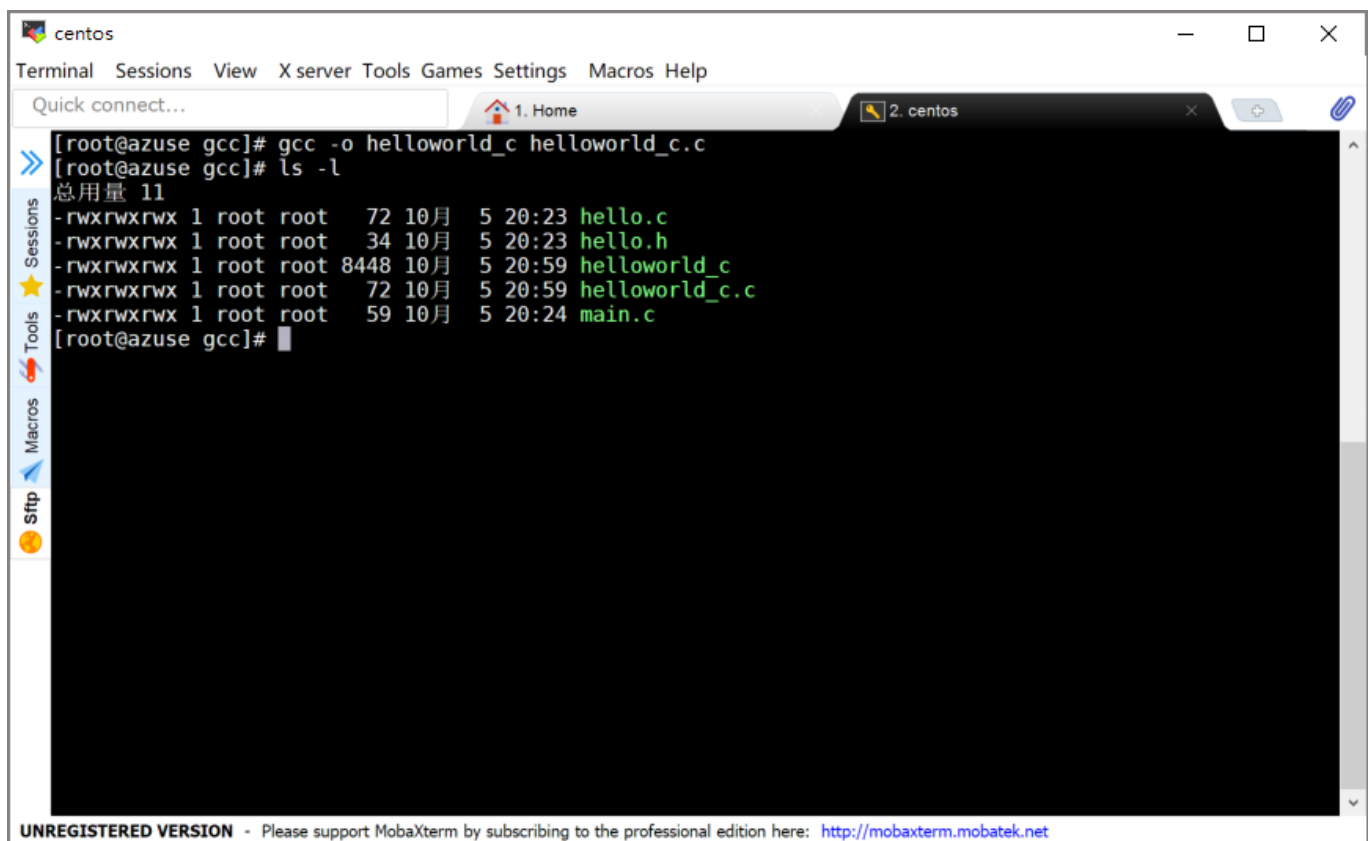
.a .lib : archives, also known as library, and are a set of object files.

一般动态编译的程序调用的系统函数默认位于系统的共享库中，而不在可执行文件中，gcc与g++默认都采用这种方式，无需外加参数。

动态编译printf("hello world");

```
gcc -o helloworld_c helloworld_c.c
```

生成的helloworld_c大小为8448字节

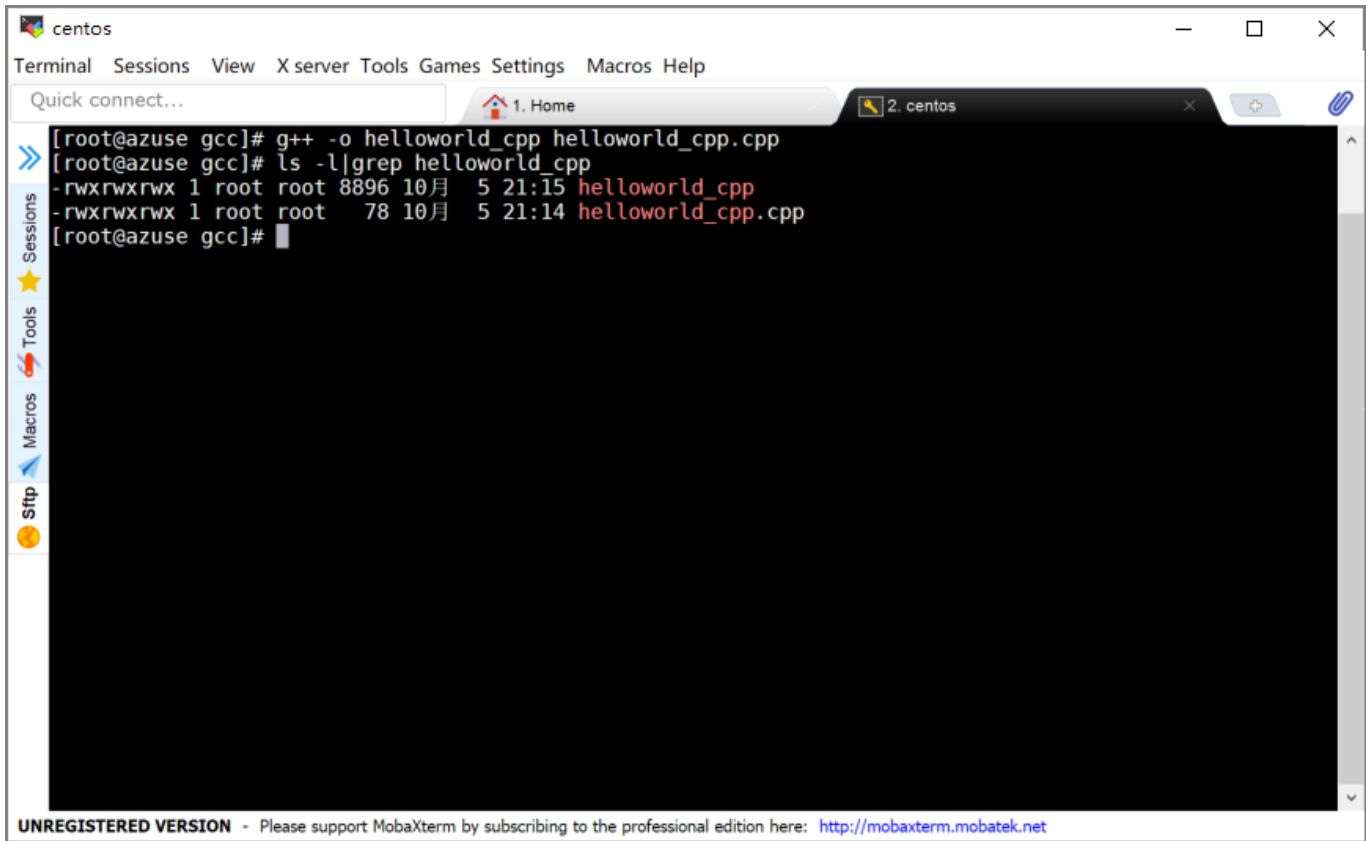


```
centos
Terminal Sessions View X server Tools Games Settings Macros Help
Quick connect...
1. Home 2. centos
[root@azuse gcc]# gcc -o helloworld_c helloworld_c.c
[root@azuse gcc]# ls -l
总用量 11
-rwxrwxrwx 1 root root 72 10月 5 20:23 hello.c
-rwxrwxrwx 1 root root 34 10月 5 20:23 hello.h
-rwxrwxrwx 1 root root 8448 10月 5 20:59 helloworld_c
-rwxrwxrwx 1 root root 72 10月 5 20:59 helloworld_c.c
-rwxrwxrwx 1 root root 59 10月 5 20:24 main.c
[root@azuse gcc]#
```

动态编译cout << "hello, world";

```
g++ -o helloworld_cpp helloworld_cpp.cpp
```

生成的helloworld_cpp大小为8896字节



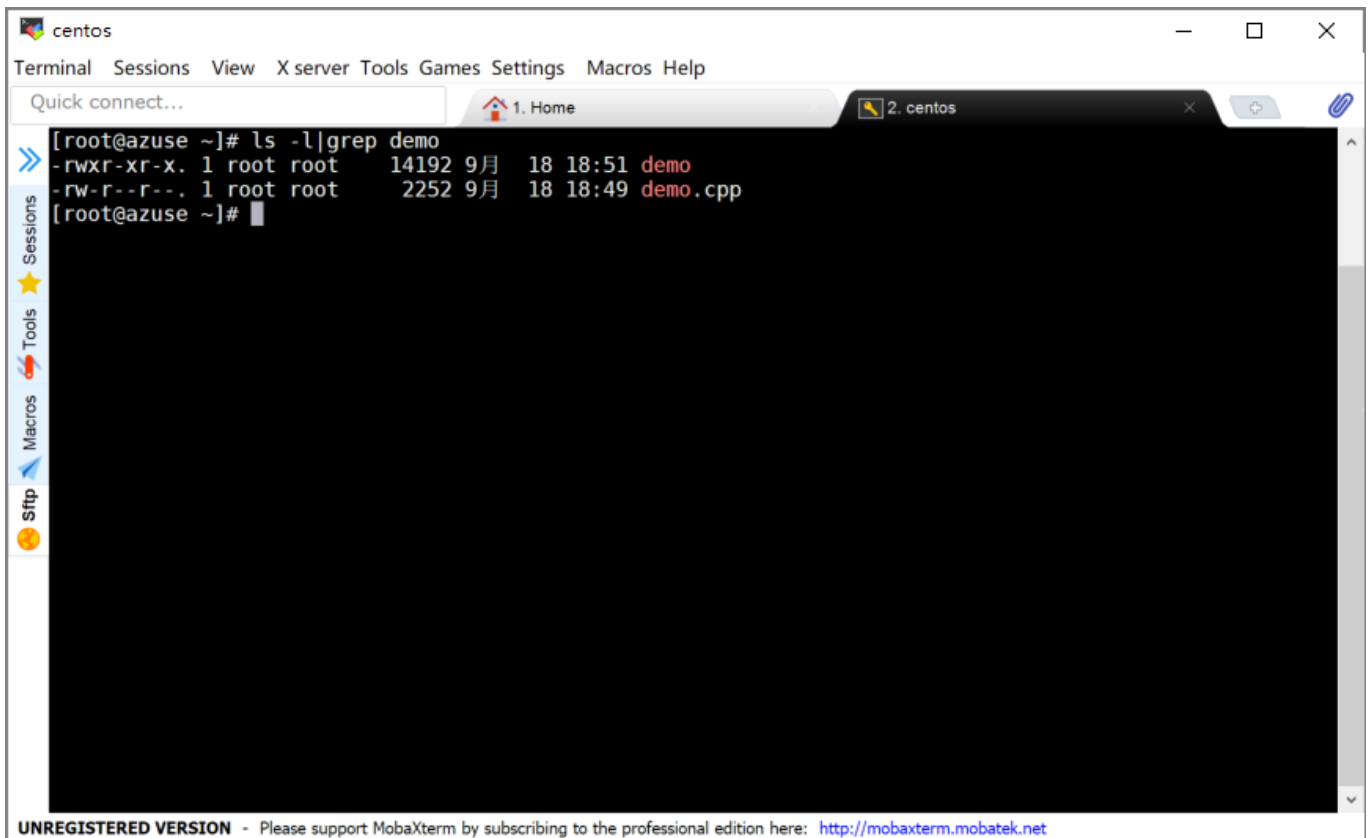
A screenshot of a MobaXterm terminal window. The terminal shows the following commands and output:

```
[root@azuse gcc]# g++ -o helloworld_cpp helloworld_cpp.cpp
[root@azuse gcc]# ls -l|grep helloworld_cpp
-rwxrwxrwx 1 root root 8896 10月 5 21:15 helloworld_cpp
-rwxrwxrwx 1 root root 78 10月 5 21:14 helloworld_cpp.cpp
[root@azuse gcc]#
```

The terminal window has a menu bar with 'Terminal', 'Sessions', 'View', 'X server', 'Tools', 'Games', 'Settings', 'Macros', and 'Help'. The status bar at the bottom says 'UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <http://mobaxterm.mobatek.net>'.

第一周给出的mysql_demo.cpp的动态编译命令的可执行文件字节数

生成的demo文件大小为14192字节



A screenshot of a MobaXterm terminal window. The terminal shows the following commands and output:

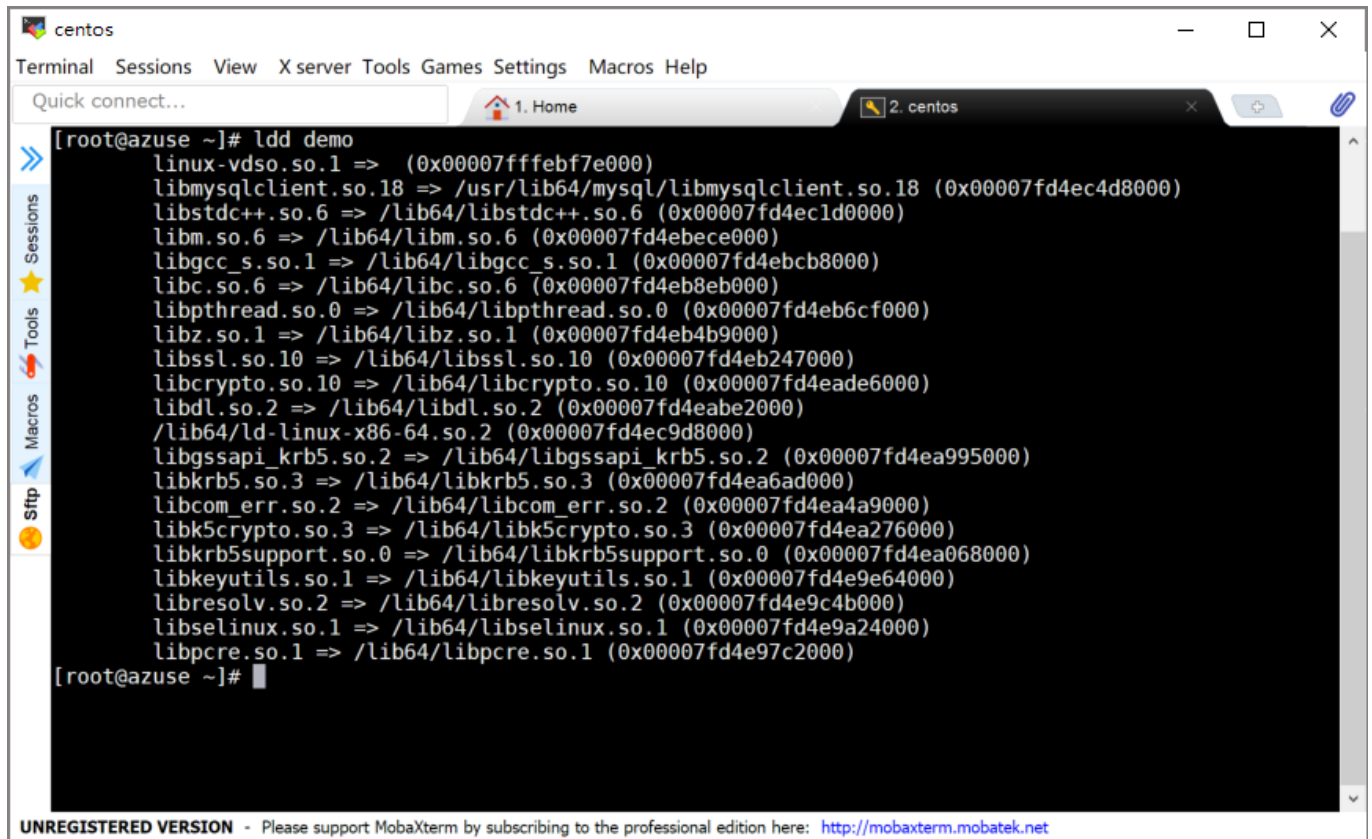
```
[root@azuse ~]# ls -l|grep demo
-rwxr-xr-x. 1 root root 14192 9月 18 18:51 demo
-rw-r--r-- 1 root root 2252 9月 18 18:49 demo.cpp
[root@azuse ~]#
```

The terminal window has a menu bar with 'Terminal', 'Sessions', 'View', 'X server', 'Tools', 'Games', 'Settings', 'Macros', and 'Help'. The status bar at the bottom says 'UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <http://mobaxterm.mobatek.net>'.

如何查找某个可执行文件的动态链接库？

使用ldd命令

demo文件的动态链接库ldd demo :

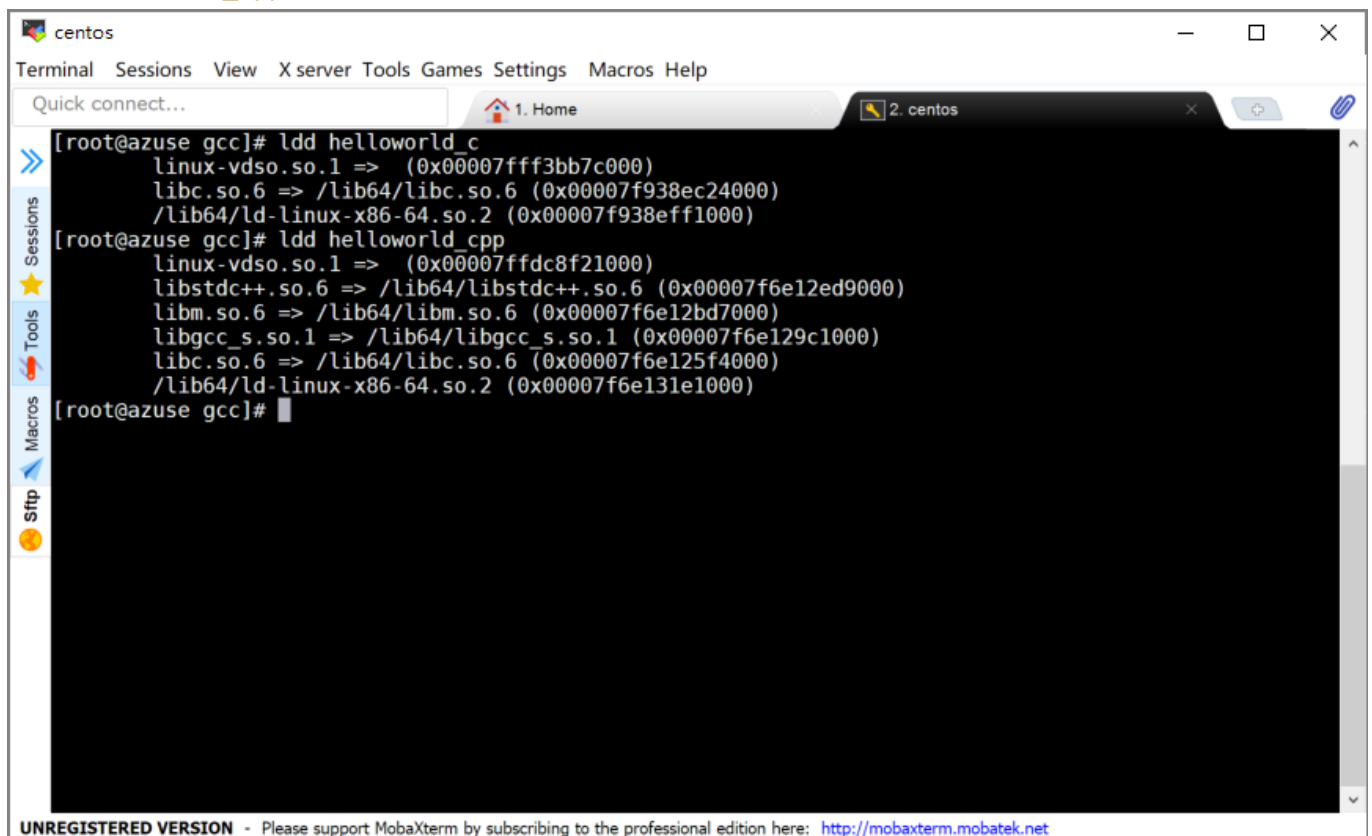


```
[root@azuse ~]# ldd demo
linux-vdso.so.1 => (0x00007fffefbf7e000)
libmysqlclient.so.18 => /usr/lib64/mysql/libmysqlclient.so.18 (0x00007fd4ec4d8000)
libstdc++.so.6 => /lib64/libstdc++.so.6 (0x00007fd4ec1d0000)
libm.so.6 => /lib64/libm.so.6 (0x00007fd4ebece000)
libgcc_s.so.1 => /lib64/libgcc_s.so.1 (0x00007fd4ebcb8000)
libc.so.6 => /lib64/libc.so.6 (0x00007fd4eb8eb000)
libpthread.so.0 => /lib64/libpthread.so.0 (0x00007fd4eb6cf000)
libz.so.1 => /lib64/libz.so.1 (0x00007fd4eb4b9000)
libssl.so.10 => /lib64/libssl.so.10 (0x00007fd4eb247000)
libcrypto.so.10 => /lib64/libcrypto.so.10 (0x00007fd4eade6000)
libdl.so.2 => /lib64/libdl.so.2 (0x00007fd4eabe2000)
/lib64/ld-linux-x86-64.so.2 (0x00007fd4ec9d8000)
libgssapi_krb5.so.2 => /lib64/libgssapi_krb5.so.2 (0x00007fd4ea995000)
libkrb5.so.3 => /lib64/libkrb5.so.3 (0x00007fd4ea6ad000)
libcom_err.so.2 => /lib64/libcom_err.so.2 (0x00007fd4ea4a9000)
libk5crypto.so.3 => /lib64/libk5crypto.so.3 (0x00007fd4ea276000)
libkrb5support.so.0 => /lib64/libkrb5support.so.0 (0x00007fd4ea068000)
libkeyutils.so.1 => /lib64/libkeyutils.so.1 (0x00007fd4e9e64000)
libresolv.so.2 => /lib64/libresolv.so.2 (0x00007fd4e9c4b000)
libselinux.so.1 => /lib64/libselinux.so.1 (0x00007fd4e9a24000)
libpcre.so.1 => /lib64/libpcre.so.1 (0x00007fd4e97c2000)
[root@azuse ~]#
```

helloworld_c和helloworld_cpp使用的动态链接库:

ldd helloworld_c

ldd helloworld_cpp



```
[root@azuse gcc]# ldd helloworld_c
linux-vdso.so.1 => (0x00007fff3bb7c000)
libc.so.6 => /lib64/libc.so.6 (0x00007f938ec24000)
/lib64/ld-linux-x86-64.so.2 (0x00007f938eff1000)
[root@azuse gcc]# ldd helloworld_cpp
linux-vdso.so.1 => (0x00007ffdc8f21000)
libstdc++.so.6 => /lib64/libstdc++.so.6 (0x00007f6e12ed9000)
libm.so.6 => /lib64/libm.so.6 (0x00007f6e12bd7000)
libgcc_s.so.1 => /lib64/libgcc_s.so.1 (0x00007f6e129c1000)
libc.so.6 => /lib64/libc.so.6 (0x00007f6e125f4000)
/lib64/ld-linux-x86-64.so.2 (0x00007f6e131e1000)
[root@azuse gcc]#
```

静态编译

什么叫静态编译？

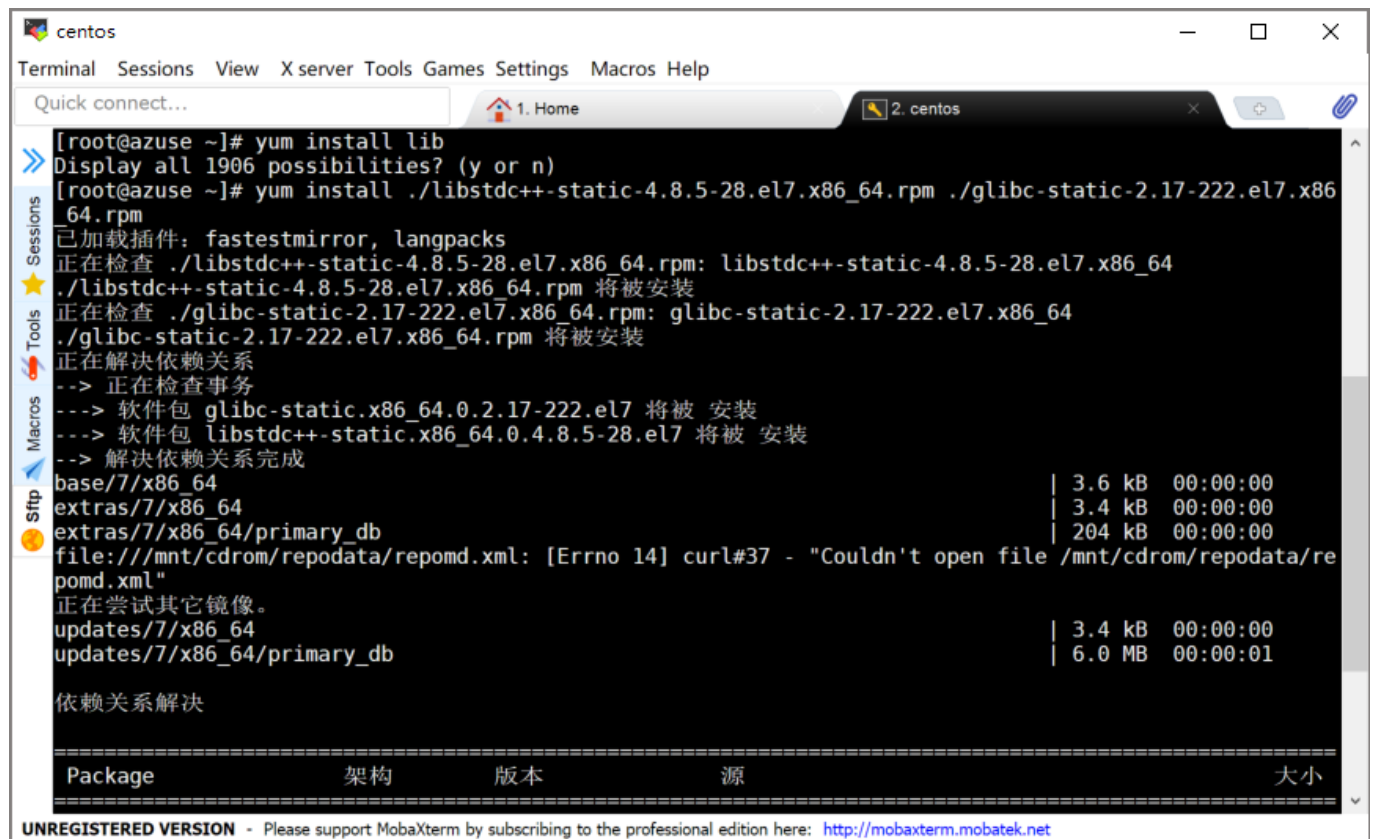
静态编译就是在编译时，把所有模块都编译进可执行文件里，当启动这个可执行文件时，所有模块都被加载进来。这样产生的可执行文件显然更浪费空间，但是优点是他们可以在不依赖于系统的共享库也能正常运行

静态编译 `printf("hello, world");`

需要先安装 `libstdc++-static` 与 `glibc-static`

将下载的rpm包放到 `/root` 下，并使用 `yum` 安装

```
yum install ./libstdc++-static-4.8.5-28.el7.x86_64.rpm ./glibc-static-2.17-222.el7.x86_64.rpm
```



```
centos
Terminal Sessions View X server Tools Games Settings Macros Help
Quick connect...
1. Home 2. centos
[root@azuse ~]# yum install lib
Display all 1906 possibilities? (y or n)
[root@azuse ~]# yum install ./libstdc++-static-4.8.5-28.el7.x86_64.rpm ./glibc-static-2.17-222.el7.x86_64.rpm
已加载插件: fastestmirror, langpacks
正在检查 ./libstdc++-static-4.8.5-28.el7.x86_64.rpm: libstdc++-static-4.8.5-28.el7.x86_64
./libstdc++-static-4.8.5-28.el7.x86_64.rpm 将被安装
正在检查 ./glibc-static-2.17-222.el7.x86_64.rpm: glibc-static-2.17-222.el7.x86_64
./glibc-static-2.17-222.el7.x86_64.rpm 将被安装
正在解决依赖关系
--> 正在检查事务
--> 软件包 glibc-static.x86_64.0.2.17-222.el7 将被 安装
--> 软件包 libstdc++-static.x86_64.0.4.8.5-28.el7 将被 安装
--> 解决依赖关系完成
base/7/x86_64 | 3.6 kB 00:00:00
extras/7/x86_64 | 3.4 kB 00:00:00
extras/7/x86_64/primary_db | 204 kB 00:00:00
file:///mnt/cdrom/repodata/repomd.xml: [Errno 14] curl#37 - "Couldn't open file /mnt/cdrom/repodata/repomd.xml"
正在尝试其它镜像。
updates/7/x86_64 | 3.4 kB 00:00:00
updates/7/x86_64/primary_db | 6.0 MB 00:00:01

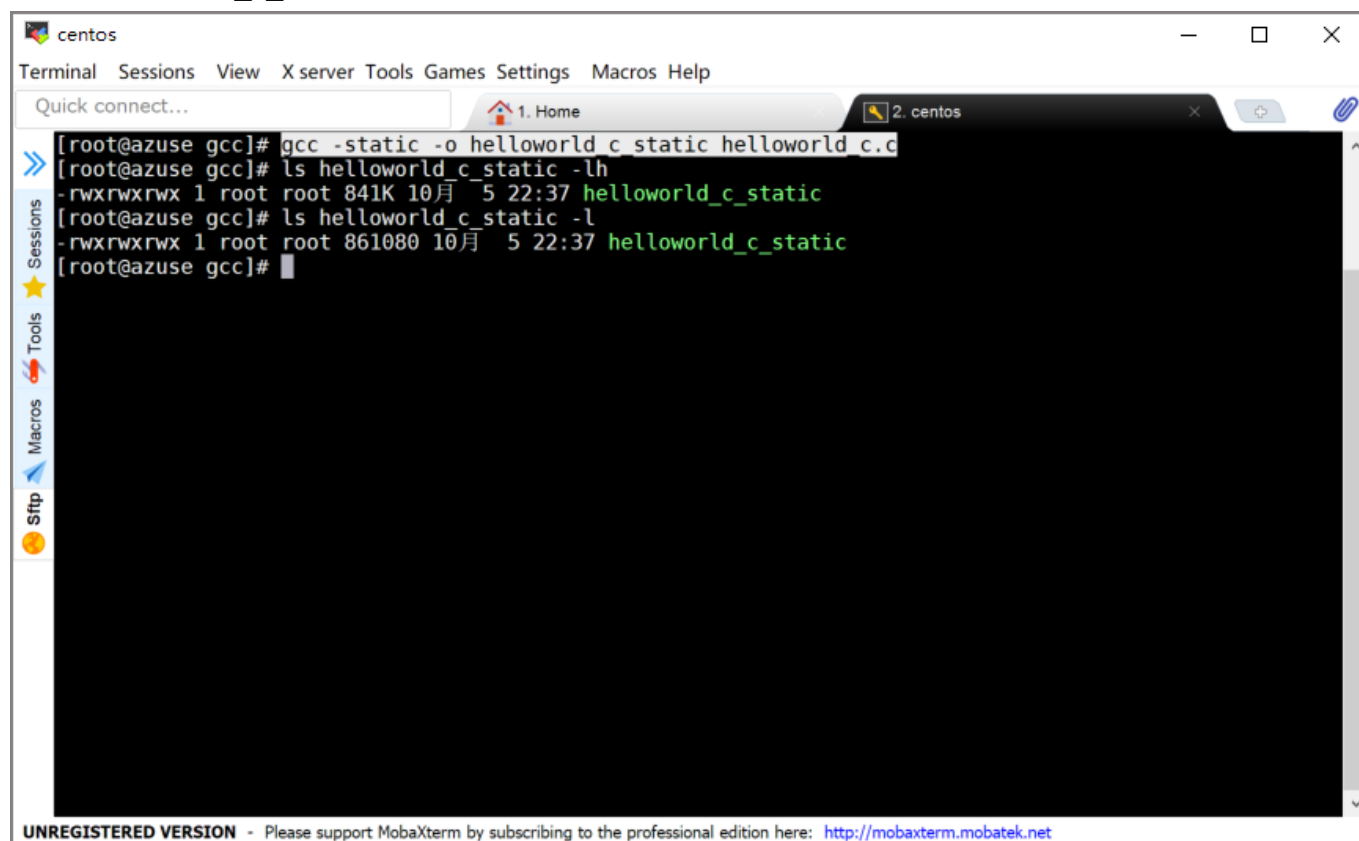
依赖关系解决

=====
Package          架构 版本      源      大小
=====
```

静态编译命令：

```
gcc -static -o helloworld_c_static helloworld_c.c
```

生成的helloworld_c_static文件大小为841KB



```
centos
Terminal Sessions View X server Tools Games Settings Macros Help
Quick connect... 1. Home 2. centos
[root@azuse gcc]# gcc -static -o helloworld_c_static helloworld.c
[root@azuse gcc]# ls helloworld_c_static -lh
-rwxrwxrwx 1 root root 841K 10月 5 22:37 helloworld_c_static
[root@azuse gcc]# ls helloworld_c_static -l
-rwxrwxrwx 1 root root 861080 10月 5 22:37 helloworld_c_static
[root@azuse gcc]#
```

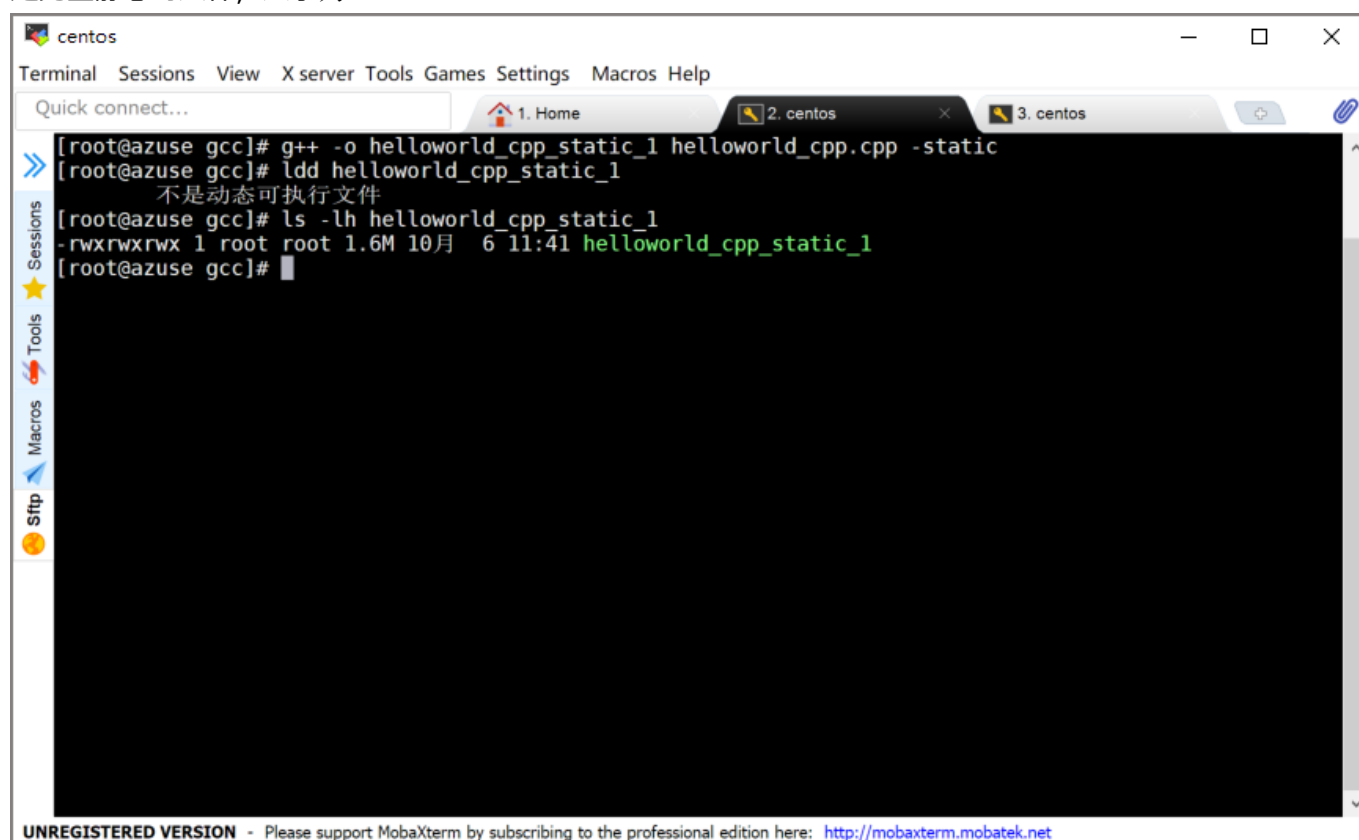
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <http://mobaxterm.mobatek.net>

静态编译`cout << "hello, world";`

需要libstc++-static静态c++库和glibc-static, 生成helloworld_cpp_staic_1

```
g++ -o helleworld_cpp_static_1 helloworld_cpp.cpp -static
```

是完全静态的文件, 大小为1.6MB



```
centos
Terminal Sessions View X server Tools Games Settings Macros Help
Quick connect... 1. Home 2. centos 3. centos
[root@azuse gcc]# g++ -o helloworld_cpp_static_1 helloworld_cpp.cpp -static
[root@azuse gcc]# ldd helloworld_cpp_static_1
不是动态可执行文件
[root@azuse gcc]# ls -lh helloworld_cpp_static_1
-rwxrwxrwx 1 root root 1.6M 10月 6 11:41 helloworld_cpp_static_1
[root@azuse gcc]#
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <http://mobaxterm.mobatek.net>

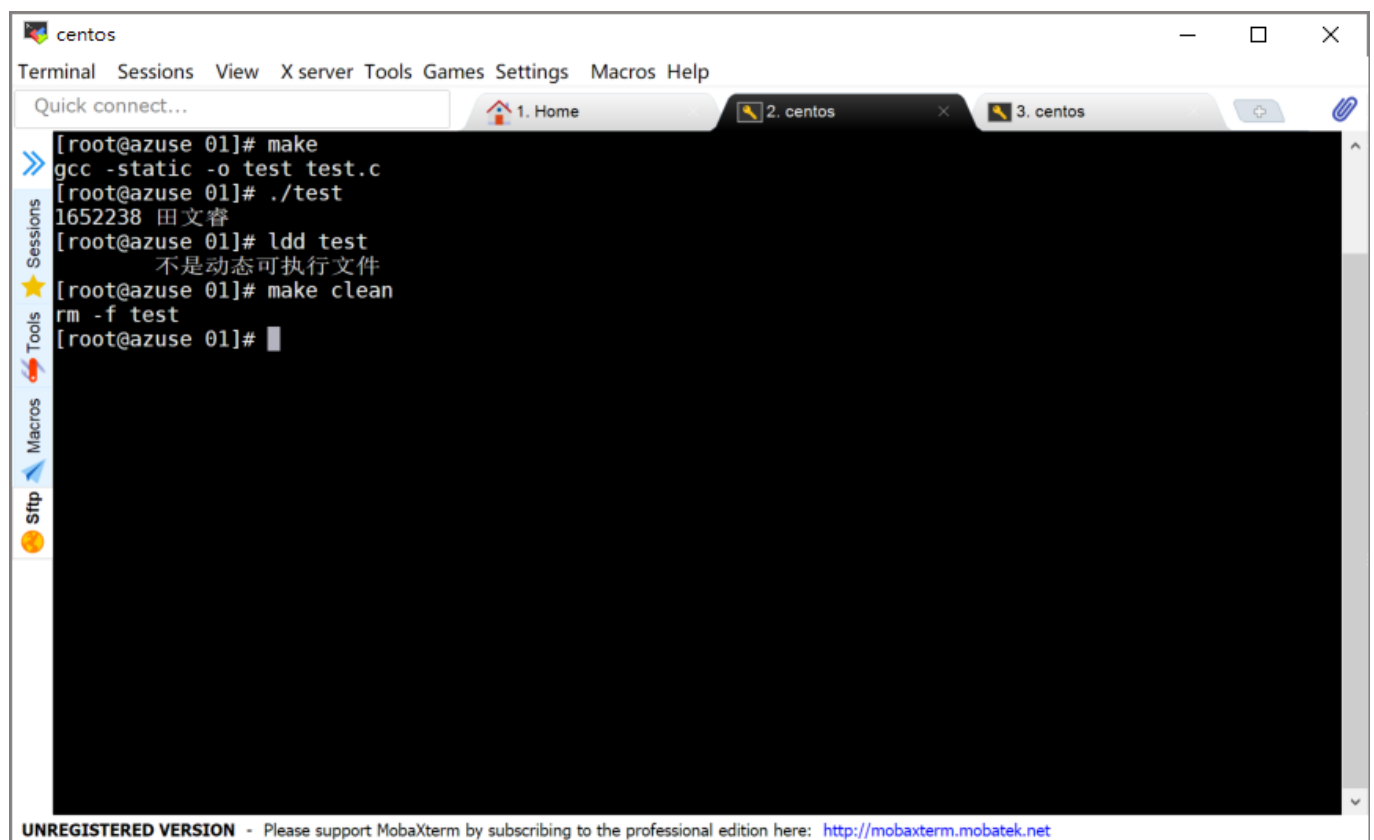
4 写出几种常用情况的静态编译测试样例

4.2 在01目录下建立test.c, 使用make静态编译

makefile:

```
# makefile 01
test : test.o
    gcc -static -o test test.c
clean :
    rm -f test test.o
```

编译过程：



The screenshot shows a MobaXterm terminal window with the following commands and output:

```
[root@azuse 01]# make
gcc -static -o test test.c
[root@azuse 01]# ./test
1652238 田文睿
[root@azuse 01]# ldd test
不是动态可执行文件
[root@azuse 01]# make clean
rm -f test
[root@azuse 01]#
```

The terminal window has a menu bar with 'Terminal', 'Sessions', 'View', 'X server', 'Tools', 'Games', 'Settings', 'Macros', and 'Help'. It also shows a 'Quick connect...' field and several tabs for different sessions. A vertical sidebar on the left contains icons for 'Sessions', 'Tools', 'Macros', and 'Sftp'. At the bottom, a message states: 'UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <http://mobaxterm.mobatek.net>'.

正常编译出静态可执行文件, 使用make clean可以清除掉可执行文件

4.3 在子目录03下建立test.cpp, 并写出满足要求的makefile文件

makefile:

```
# makefile 02
test : test.o
    g++ -o test test.cpp -static
.PHONY : clean
clean :
    rm -f test test.o
```

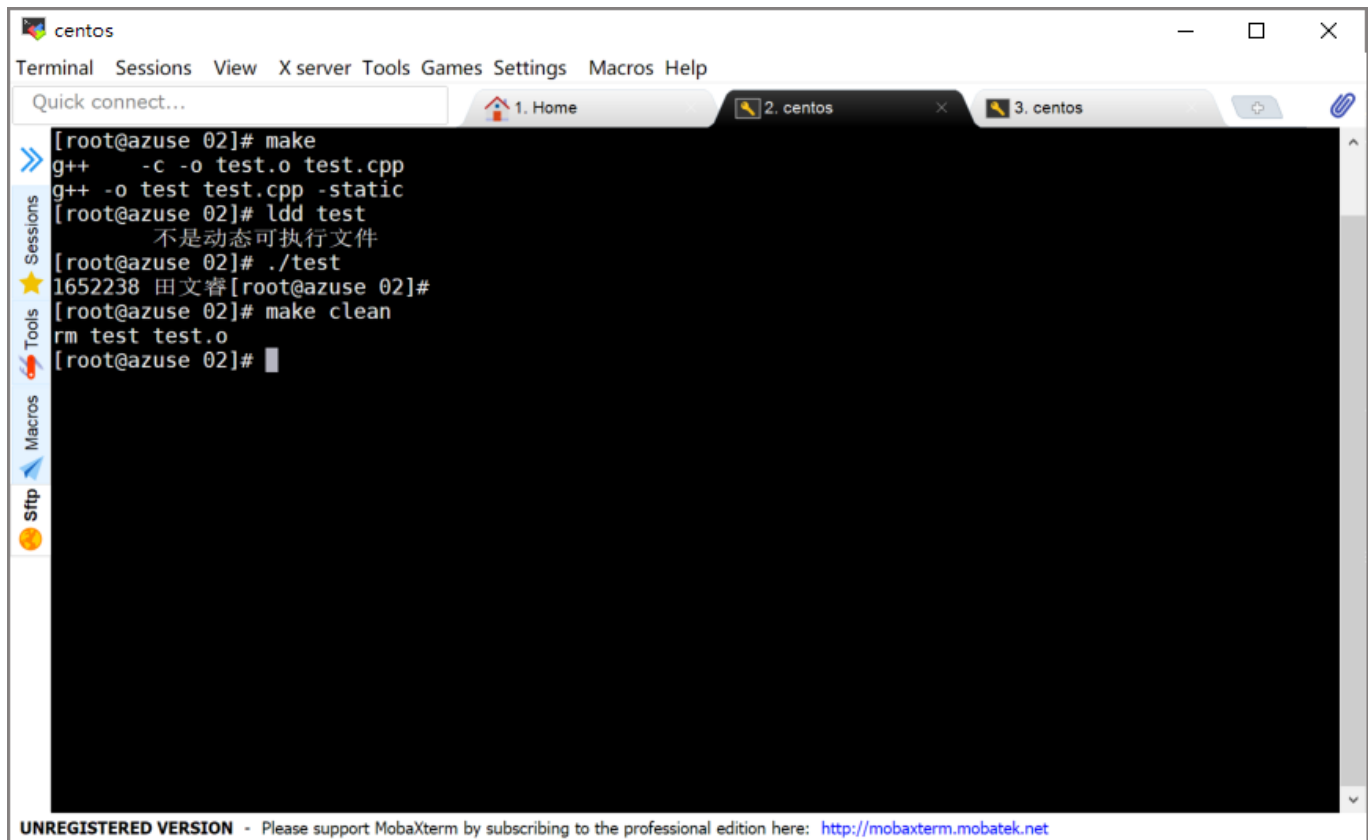
安装的静态库：

libstdc++-static-4.8.5-28.el7.x86_64.rpm

glibc-static-2.17-222.el7.x86_64.rpm

安装方法：`yum install`本地安装

编译过程：



```

[root@azuse 02]# make
g++ -c -o test.o test.cpp
g++ -o test test.cpp -static
[root@azuse 02]# ldd test
不是动态可执行文件
[root@azuse 02]# ./test
1652238 田文睿[root@azuse 02]#
[root@azuse 02]# make clean
rm test test.o
[root@azuse 02]#
  
```

正常编译出静态可执行文件，使用`make clean`可以清楚可执行文件

4.4 1652238-000104目录下的makefile文件

可以直接将1652238-000103目录下的makefile文件拿来用

```

# makefile 1652238-000104

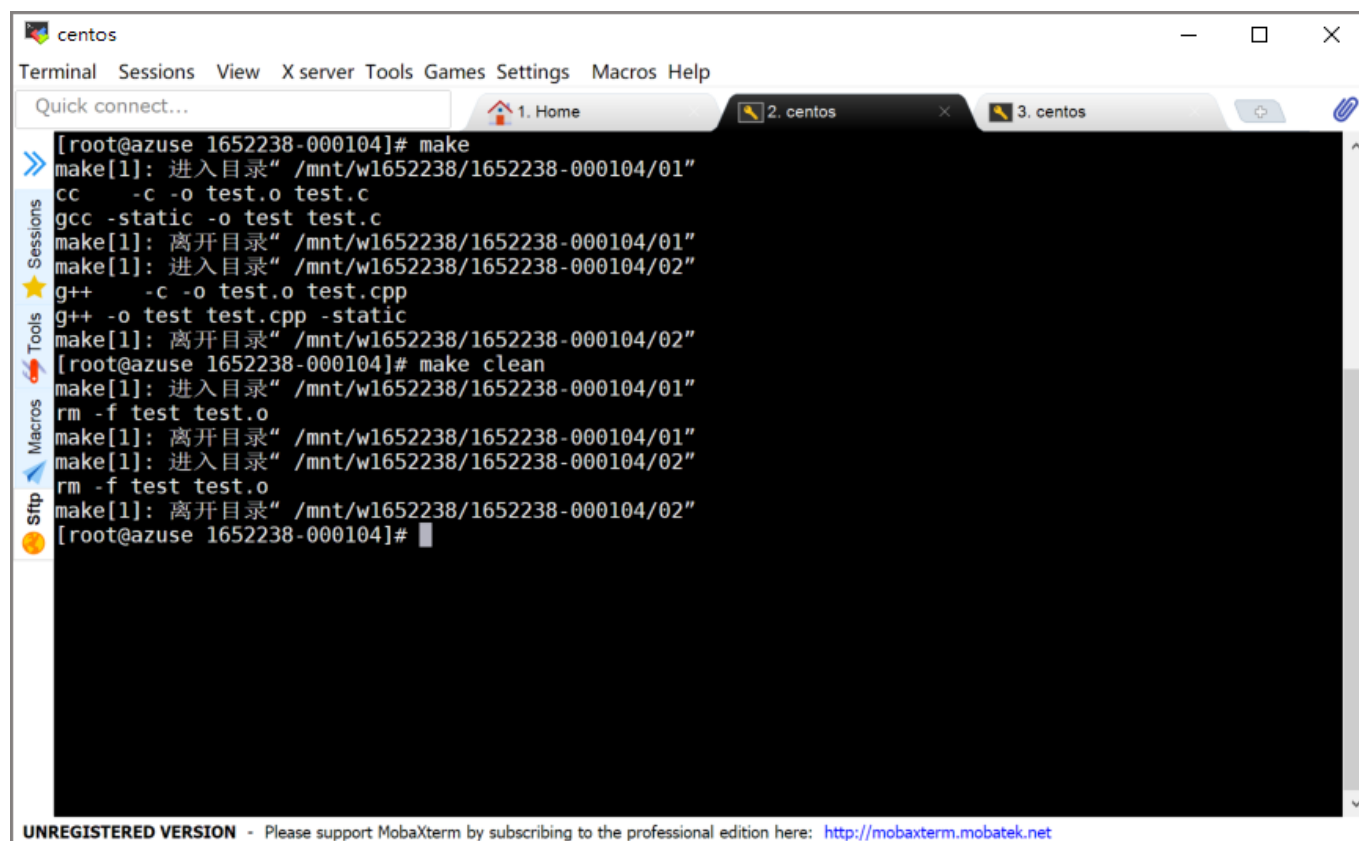
subdir=$(shell find . -maxdepth 1 -type d)
dirs:=$(basename $(patsubst ./%,%, $(subdir)))

.PHONY: $(dirs) clean

$(dirs):
    @for dir in $(dirs); do \
        $(MAKE) -C $$dir; \
    done

clean:
    @for dir in $(dirs); do \
        $(MAKE) -C $$dir clean; \
    done
  
```


编译过程：



```
centos
Terminal Sessions View X server Tools Games Settings Macros Help
Quick connect... 1. Home 2. centos 3. centos
[root@azuse 1652238-000104]# make
make[1]: 进入目录 "/mnt/w1652238/1652238-000104/01"
cc -c -o test.o test.c
gcc -static -o test test.c
make[1]: 离开目录 "/mnt/w1652238/1652238-000104/01"
make[1]: 进入目录 "/mnt/w1652238/1652238-000104/02"
g++ -c -o test.o test.cpp
g++ -o test test.cpp -static
make[1]: 离开目录 "/mnt/w1652238/1652238-000104/02"
[root@azuse 1652238-000104]# make clean
make[1]: 进入目录 "/mnt/w1652238/1652238-000104/01"
rm -f test test.o
make[1]: 离开目录 "/mnt/w1652238/1652238-000104/01"
make[1]: 进入目录 "/mnt/w1652238/1652238-000104/02"
rm -f test test.o
make[1]: 离开目录 "/mnt/w1652238/1652238-000104/02"
[root@azuse 1652238-000104]#
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <http://mobaxterm.mobatek.net>

make可以正常编译每个子目录下的文件，make clean可以清除子目录下的.o和可执行文件