**Linux debian中开发环境搭建：**

**框架编译运行环境依赖包：apt-get**

网络连通的条件下执行系统命令，自行安装：

1、安装lua包：

apt-get install lua5.2

apt-get install liblua5.2-dev

2、安装tingxml包：

apt-get install libtinyxml2.6.2

如果安装失败，见详细修改方法参考。

3、安装mysql包：

apt-get install mysql-server-5.5

apt-get install libmysql++-dev

apt-get install mysql-client-5.5

4、安装uuid包：

apt-get install uuid-dev

apt-get install libuuid1

5、安装tcl8.5包：

redis的依赖包

apt-get install tcl8.5

6、安装redis包：

命令自动安装版本比较旧(<http://redis.googlecode.com/files/redis-2.6.13.tar.gz>)

[wget http://download.redis.io/releases/redis-2.6.14.tar.gz](wget%20http://download.redis.io/releases/redis-2.6.14.tar.gz)

自行下载: redis-2.6.13.tar.gz

make linux

make install

修改redis.conf中daemonize = yes

解压后执行安装

7、安装hiredis

apt-get install libhiredis0.10

8、安装openssl

apt-get install openssl

9、安装readline6.x

apt-get install readline-dev

10、安装valGrind包：

内存检查工具

apt-get install valgrind

11、安装svn包：

apt-get install subversion

**Linux centos中开发环境搭建：yum**

首先确保系统已安装好make，g++（没有安装自行解决）

yum install make

yum install gcc-c++

1. 安装mysql5.1

yum install mysql(若没有自行关联其他安装包，执行下面的安装条件)

yum install mysql-server

yum install phy-mysql

yum install mysql-devel

1. 安装readline

yum install readline-devel（6.x）

1. 安装tingxml2.6.2

自行下载编译安装包，见**Tinyxml自编译安装方法。**

1. 安装uuid

yum install uuid

yum install uuid-devel

yum install libuuid

yum libevent-devel

yum libevent

1. 安装tcl

wget <http://downloads.sourceforge.net/tcl/tcl8.5.12-src.tar.gz>

tar /tcl8.5.12-src.tar.gz

cd tcl8.5.12

cd unix

./configure

make

make install

1. 安装lua5.2

wget <http://www.lua.org/ftp/lua-5.2.2.tar.gz>

tar xf lua-5.2.2.tar.gz

cd lua-5.2.2

make linux

make install

1. 安装redis2.6.x

与debian安装过程相同

wget <http://redis.googlecode.com/files/redis-2.6.13.tar.gz>

tar xf redis-2.6.13.tar.gz

cd redis-2.6.13

make linux

make test //测试是否正确安装

make install

1. 安装hiredis0.10

wget <http://dl.fedoraproject.org/pub/epel/6/x86_64/hiredis-0.10.1-3.el6.x86_64.rpm>

yum install hiredis-0.10.1-3.el6.x86\_64.rpm

wget <http://dl.fedoraproject.org/pub/epel/6/x86_64/hiredis-devel-0.10.1-3.el6.x86_64.rpm>

yum install hiredis-devel-0.10.1-3.el6.x86\_64.rpm

1. 安装openssl

wget <http://dl.fedoraproject.org/pub/epel/6/x86_64/> perl-Crypt-OpenSSL-AES-0.02-9.el6.x86\_64.rpm

yum install perl-Crypt-OpenSSL-AES-0.02-9.el6.x86\_64.rpm

wget <http://dl.fedoraproject.org/pub/epel/6/x86_64/> perl-Crypt-OpenSSL-DSA-0.13-14.el6.x86\_64.rpm

yum install perl-Crypt-OpenSSL-DSA-0.13-14.el6.x86\_64.rpm

wget <http://dl.fedoraproject.org/pub/epel/6/x86_64/> perl-Crypt-RC4-XS-0.02-2.el6.x86\_64.rpm

yum install perl-Crypt-RC4-XS-0.02-2.el6.x86\_64.rpm

**Tinyxml自编译安装方法：**

Linux下将Tinyxml编译为静态库

1.下载tinyxml库，直接从官方下载，http://sourceforge.net/projects/tinyxml/。

2.解压缩下载好的包，我解压缩到test目录下,进入项目目录。

3.使用vim编辑器修改Makefile文件：

将其中的OUTPUT := xmltest一行修改为：OUTPUT := libtinyxml.a

将xmltest.cpp从SRCS：=tinyxml.cpp tinyxml-parser.cpp xmltest.cpp tinyxmlerror.cpp tinystr.cpp中删除，，注释掉xmltest.o：tinyxml.h tinystr.h。因为不需要将演示程序添加到动态库中。

将${LD} -o $@ ${LDFLAGS} ${OBJS} ${LIBS} ${EXTRA\_LIBS}修改为：${AR} $@ ${LDFLAGS} ${OBJS} ${LIBS} ${EXTRA\_LIBS}。

保存退出。

4.执行make命令编译，即可在当前目录生成libtinyxml.a文件。

make时候遇到"common/defparser.cpp:16:29: fatal error: openssl/md5.h: No such file or directory"解决方法：

#apt-get install libssl-dev (这个必须做)

然后#find / -name md5.h 看看/usr/include/openssl/md5.h有没有

make时候遇到“dbmgr/myredis.h:5:29: fatal error: hiredis/hiredis.h: No such file or directory”解决方法：

#cp /usr/src/redis-2.6.14/deps/hiredis /usr/local/include -rf （看看hiredis文件夹在哪里）

make时候遇到“loginapp/world\_login.cpp:13:24: fatal error: curl/curl.h: No such file or directory”解决方法：

http://curl.haxx.se/download.html官网下载curl-7.41.0.tar.gz并rz到debian的/usr/src/解压缩

#cd curl-7.41.0

#./configure

#make

find一下curl.h看有没有，有就#cp /usr/src/curl-7.41.0/include/\* /usr/local/include/ -rf (我安装到的目录是/usr/src/curl-7.41.0/)

make时候遇到"g++: error: /usr/lib/libtinyxml.a: No such file or directory"解决方法：

3.进入到解压缩目录下，我们会发现Tinyxml在Windows 下是使用微软的VS 来生成的库，因为其中有tinyxml.sln，tinyxml\_lib.vcxproj，tinyxmlSTL.vcxproj等文件，当然，Tinyxml是开源的，所以它也有一个Makefile,用来生成Linux下的Tinyxml库。整个Tinyxml源码项目其实是由2个头文件和一个4个C++源文件（.cpp）组成：tinystr.h，tinyxml.h，tinystr.cpp，tinyxml.cpp，tinyxmlerror.cpp，tinyxmlparser.cpp。其中还有一个xmltest.cpp文件，只是一个测试代码，有兴趣的话，大家可以打开研究它。好了，现在介绍怎么修改它的Makefile：

(1)使用vim或者其他的编辑器打开Makefile文件

(2)将其中的注释为Targets of the build的下一行OUTPUT := xmltest一行修改为：OUTPUT := libtinyxml.a

(3)将其中的注释为Source files 的下一行SRCS：=tinyxml.cpp tinyxml-parser.cpp xmltest.cpp tinyxmlerror.cpp tinystr.cpp中的xmltest.cpp删除，因为它只是一个测试源文件，不需要编译。

(4)将其中的Output的下一行的${LD} -o $@ ${LDFLAGS} ${OBJS} ${LIBS} ${EXTRA\_LIBS}修改为：${AR} $@ ${LDFLAGS} ${OBJS} ${LIBS} ${EXTRA\_LIBS}。

大致改成这样:

${OUTPUT}: ${OBJS}

${AR} $@ ${LDFLAGS} ${OBJS} ${LIBS} ${EXTRA\_LIBS}

(5)将Makefile的倒数第二行 xmltest.o：tinyxml.h tinystr.h，注释掉，因为不需要将演示程序添加到静态库中。然后保存退出。

(6)在终端下进入Makefile所在目录，执行make命令编译，即可在Makefile所在目录下生成libtinyxml.a文件。

#cp libtinyxml.a /usr/lib

make时候遇到"g++: error: /usr/local/lib/libssl.a: No such file or directory"解决方法：

ln -s /usr/lib/x86\_64-linux-gnu/libssl.a /usr/local/lib/libssl.a (find / -name \*libssl\*看有没有，没有要下载安装)

make时候遇到"g++: error: /usr/local/lib/libcrypto.a: No such file or directory"解决方法：

ln -s /usr/lib/x86\_64-linux-gnu/libcrypto.a /usr/local/lib/libcrypto.a (find / -name \*libcrypto\*看有没有，没有要下载安装)

make时候遇到"/usr/bin/ld: cannot find -lhiredis"解决方法：

ln -s /usr/lib/x86\_64-linux-gnu/libhiredis.so.0.10 /usr/lib/libhiredis.so

make时候遇到"/usr/bin/ld: cannot find -lcurl"解决方法：

ln -s /usr/lib/x86\_64-linux-gnu/libcurl.so.4 /usr/lib/libcurl.so

make时候遇到"/usr/lib/libtinyxml.a: In function `\_start':

(.text+0x3658): multiple definition of `\_start'

/usr/bin/ld: /usr/lib/debug/usr/lib/x86\_64-linux-gnu/crt1.o(.debug\_info): relocation 0 has invalid symbol index 11

/usr/bin/ld: /usr/lib/debug/usr/lib/x86\_64-linux-gnu/crt1.o(.debug\_info): relocation 1 has invalid symbol index 12

/usr/bin/ld: /usr/lib/debug/usr/lib/x86\_64-linux-gnu/crt1.o(.debug\_info): relocation 2 has invalid symbol index 2

/usr/bin/ld: /usr/lib/debug/usr/lib/x86\_64-linux-gnu/crt1.o(.debug\_info): relocation 3 has invalid symbol index 2

/usr/bin/ld: /usr/lib/debug/usr/lib/x86\_64-linux-gnu/crt1.o(.debug\_info): relocation 4 has invalid symbol index 11

/usr/bin/ld: /usr/lib/debug/usr/lib/x86\_64-linux-gnu/crt1.o(.debug\_info): relocation 5 has invalid symbol index 13

/usr/bin/ld: /usr/lib/debug/usr/lib/x86\_64-linux-gnu/crt1.o(.debug\_info): relocation 6 has invalid symbol index 13

/usr/bin/ld: /usr/lib/debug/usr/lib/x86\_64-linux-gnu/crt1.o(.debug\_info): relocation 7 has invalid symbol index 13

/usr/bin/ld: /usr/lib/debug/usr/lib/x86\_64-linux-gnu/crt1.o(.debug\_info): relocation 8 has invalid symbol index 2

/usr/bin/ld: /usr/lib/debug/usr/lib/x86\_64-linux-gnu/crt1.o(.debug\_info): relocation 9 has invalid symbol index 2

/usr/bin/ld: /usr/lib/debug/usr/lib/x86\_64-linux-gnu/crt1.o(.debug\_info): relocation 10 has invalid symbol index 12

/usr/bin/ld: /usr/lib/debug/usr/lib/x86\_64-linux-gnu/crt1.o(.debug\_info): relocation 11 has invalid symbol index 13

/usr/bin/ld: /usr/lib/debug/usr/lib/x86\_64-linux-gnu/crt1.o(.debug\_info): relocation 12 has invalid symbol index 13

/usr/bin/ld: /usr/lib/debug/usr/lib/x86\_64-linux-gnu/crt1.o(.debug\_info): relocation 13 has invalid symbol index 13

/usr/bin/ld: /usr/lib/debug/usr/lib/x86\_64-linux-gnu/crt1.o(.debug\_info): relocation 14 has invalid symbol index 13

/usr/bin/ld: /usr/lib/debug/usr/lib/x86\_64-linux-gnu/crt1.o(.debug\_info): relocation 15 has invalid symbol index 13

/usr/bin/ld: /usr/lib/debug/usr/lib/x86\_64-linux-gnu/crt1.o(.debug\_info): relocation 16 has invalid symbol index 13

/usr/bin/ld: /usr/lib/debug/usr/lib/x86\_64-linux-gnu/crt1.o(.debug\_info): relocation 17 has invalid symbol index 13

/usr/bin/ld: /usr/lib/debug/usr/lib/x86\_64-linux-gnu/crt1.o(.debug\_info): relocation 18 has invalid symbol index 13

/usr/bin/ld: /usr/lib/debug/usr/lib/x86\_64-linux-gnu/crt1.o(.debug\_info): relocation 19 has invalid symbol index 13

/usr/bin/ld: /usr/lib/debug/usr/lib/x86\_64-linux-gnu/crt1.o(.debug\_info): relocation 20 has invalid symbol index 20

/usr/bin/ld: /usr/lib/debug/usr/lib/x86\_64-linux-gnu/crt1.o(.debug\_line): relocation 0 has invalid symbol index 2"解决方法：

证明libtinyxml.a还不是静态库：请按照http://blog.csdn.net/yasi\_xi/article/details/38872467来做

有时候，在通过apt-get install 安装软件的时候，会出现：

更换介质：请把标有“Debian GNU/Linux 7.1.0 \_Wheezy\_ - Official amd64 DVD Binary-1 20130615-23:06”的盘片插入驱动器“/media/cdrom/”再按回车键

类似的提示，如果不放入CD，那么安装是不能继续下去的。

为了解决此问题，我们可以这像下面这么做：

root权限修改/etc/apt/sources.list文件，注释掉deb cdrom:开头的行。即修改成：

#deb cdrom:……

然后执行：apt-get update即可。

vi 上下左右变ABCD问题解决方法

执行命令 sudo apt-get remove vim-common

执行命令 sudo apt-get install vim

apt-get 找不到安装包

修改 /etc/apt/sources.list, 加入  
deb <http://ftp.us.debian.org/debian> stable main contrib non-free   
deb <http://ftp.us.debian.org/debian-non-US> stable/non-US main contrib non-free  
deb <http://ftp.us.debian.org/debian> testing main contrib non-free  
deb <http://ftp.us.debian.org/debian-non-US> testing/non-US main contrib non-free   
deb <http://ftp.us.debian.org/debian> unstable main contrib non-free  
deb <http://ftp.us.debian.org/debian-non-US> unstable/non-US main contrib non-free  
  
执行 apt-get update 更新软件库

g++ 未找到命令

apt-get install g++

没有找到liblua.a

下载lua5.2.1

解压 make linux

找不到lua.hpp……等文件

在lua,tinyxml文件夹将相信的文件复制到server/common文件夹下

修改bin/cfg.ini

Loginapp对应的iP改为eth0对应的ip

Baseapp对应的ip改为eth0对应的ip

并添加进Trusted

Db设定，并在mysql下创建相应的表

Log\_db也做相应设定

修改bin/run.sh

Baseapp ./cfg.ini 6改成 baseapp ./cfg.ini 11

Cellapp ./cfg.ini 7 改成 cellapp ./cfg.ini 12

执行bin/run.sh启动服务器

注意不能用root身份启动