构建sqlite访问http://www.sqlite.org/ 下载源码包，尝试分析configure.ac和Makefile.am的内容，解答以下问题：

1. sqlite默认构建过程是否链接libsqlite库，链接该库应配置哪个特性？
2. 如果要修改指定类型文件的安装路径，如可执行程序，应配置哪个特性？
3. 分析Makefile.am语法结构，尝试手动编写Makefile完成整个构建过程。

**Makefile.am**

AM\_CFLAGS = @BUILD\_CFLAGS@

lib\_LTLIBRARIES = libsqlite3.la

libsqlite3\_la\_SOURCES = sqlite3.c

libsqlite3\_la\_LDFLAGS = -no-undefined -version-info 8:6:8

bin\_PROGRAMS = sqlite3

sqlite3\_SOURCES = shell.c sqlite3.h

EXTRA\_sqlite3\_SOURCES = sqlite3.c

sqlite3\_LDADD = @EXTRA\_SHELL\_OBJ@ @READLINE\_LIBS@

sqlite3\_DEPENDENCIES = @EXTRA\_SHELL\_OBJ@

sqlite3\_CFLAGS = $(AM\_CFLAGS) -DSQLITE\_ENABLE\_EXPLAIN\_COMMENTS -DSQLITE\_ENABLE\_DBPAGE\_VTAB -DSQLITE\_ENABLE\_STMTVTAB -DSQLITE\_ENABLE\_DBSTAT\_VTAB $(SHELL\_CFLAGS)

include\_HEADERS = sqlite3.h sqlite3ext.h

EXTRA\_DIST = sqlite3.1 tea Makefile.msc sqlite3.rc sqlite3rc.h README.txt Replace.cs Makefile.fallback

pkgconfigdir = ${libdir}/pkgconfig

pkgconfig\_DATA = sqlite3.pc

man\_MANS = sqlite3.1

**configure.ac**

#-----------------------------------------------------------------------

# Supports the following non-standard switches.

#

# --enable-threadsafe

# --enable-readline

# --enable-editline

# --enable-static-shell

# --enable-dynamic-extensions

#

AC\_PREREQ(2.61)

AC\_INIT(sqlite, 3.39.0, http://www.sqlite.org)

AC\_CONFIG\_SRCDIR([sqlite3.c])

AC\_CONFIG\_AUX\_DIR([.])

# Use automake.

AM\_INIT\_AUTOMAKE([foreign])

AC\_SYS\_LARGEFILE

# Check for required programs.

AC\_PROG\_CC

AC\_PROG\_LIBTOOL

AC\_PROG\_MKDIR\_P

# Check for library functions that SQLite can optionally use.

AC\_CHECK\_FUNCS([fdatasync usleep fullfsync localtime\_r gmtime\_r])

AC\_FUNC\_STRERROR\_R

AC\_CONFIG\_FILES([Makefile sqlite3.pc])

BUILD\_CFLAGS=

AC\_SUBST(BUILD\_CFLAGS)

#-------------------------------------------------------------------------

# Two options to enable readline compatible libraries:

#

# --enable-editline

# --enable-readline

#

# Both are enabled by default. If, after command line processing both are

# still enabled, the script searches for editline first and automatically

# disables readline if it is found. So, to use readline explicitly, the

# user must pass "--disable-editline". To disable command line editing

# support altogether, "--disable-editline --disable-readline".

#

# When searching for either library, check for headers before libraries

# as some distros supply packages that contain libraries but not header

# files, which come as a separate development package.

#

AC\_ARG\_ENABLE(editline, [AS\_HELP\_STRING([--enable-editline],[use BSD libedit])])

AC\_ARG\_ENABLE(readline, [AS\_HELP\_STRING([--enable-readline],[use readline])])

AS\_IF([ test x"$enable\_editline" != xno ],[

AC\_CHECK\_HEADERS([editline/readline.h],[

sLIBS=$LIBS

LIBS=""

AC\_SEARCH\_LIBS([readline],[edit],[

AC\_DEFINE([HAVE\_EDITLINE],1,Define to use BSD editline)

READLINE\_LIBS="$LIBS -ltinfo"

enable\_readline=no

],[],[-ltinfo])

AS\_UNSET(ac\_cv\_search\_readline)

LIBS=$sLIBS

])

])

AS\_IF([ test x"$enable\_readline" != xno ],[

AC\_CHECK\_HEADERS([readline/readline.h],[

sLIBS=$LIBS

LIBS=""

AC\_SEARCH\_LIBS(tgetent, termcap curses ncurses ncursesw, [], [])

AC\_SEARCH\_LIBS(readline,[readline edit], [

AC\_DEFINE([HAVE\_READLINE],1,Define to use readline or wrapper)

READLINE\_LIBS=$LIBS

])

LIBS=$sLIBS

])

])

AC\_SUBST(READLINE\_LIBS)

#-----------------------------------------------------------------------

#-----------------------------------------------------------------------

# --enable-threadsafe

#

AC\_ARG\_ENABLE(threadsafe, [AS\_HELP\_STRING(

[--enable-threadsafe], [build a thread-safe library [default=yes]])],

[], [enable\_threadsafe=yes])

if test x"$enable\_threadsafe" == "xno"; then

BUILD\_CFLAGS="$BUILD\_CFLAGS -DSQLITE\_THREADSAFE=0"

else

BUILD\_CFLAGS="$BUILD\_CFLAGS -D\_REENTRANT=1 -DSQLITE\_THREADSAFE=1"

AC\_SEARCH\_LIBS(pthread\_create, pthread)

AC\_SEARCH\_LIBS(pthread\_mutexattr\_init, pthread)

fi

#-----------------------------------------------------------------------

#-----------------------------------------------------------------------

# --enable-dynamic-extensions

#

AC\_ARG\_ENABLE(dynamic-extensions, [AS\_HELP\_STRING(

[--enable-dynamic-extensions], [support loadable extensions [default=yes]])],

[], [enable\_dynamic\_extensions=yes])

if test x"$enable\_dynamic\_extensions" != "xno"; then

AC\_SEARCH\_LIBS(dlopen, dl)

else

BUILD\_CFLAGS="$BUILD\_CFLAGS -DSQLITE\_OMIT\_LOAD\_EXTENSION=1"

fi

AC\_MSG\_CHECKING([for whether to support dynamic extensions])

AC\_MSG\_RESULT($enable\_dynamic\_extensions)

#-----------------------------------------------------------------------

#-----------------------------------------------------------------------

# --enable-math

#

AC\_ARG\_ENABLE(math, [AS\_HELP\_STRING(

[--enable-math], [SQL math functions [default=yes]])],

[], [enable\_math=yes])

AC\_MSG\_CHECKING([SQL math functions])

if test x"$enable\_math" = "xyes"; then

BUILD\_CFLAGS="$BUILD\_CFLAGS -DSQLITE\_ENABLE\_MATH\_FUNCTIONS"

AC\_MSG\_RESULT([enabled])

AC\_SEARCH\_LIBS(ceil, m)

else

AC\_MSG\_RESULT([disabled])

fi

#-----------------------------------------------------------------------

#-----------------------------------------------------------------------

# --enable-fts4

#

AC\_ARG\_ENABLE(fts4, [AS\_HELP\_STRING(

[--enable-fts4], [include fts4 support [default=yes]])],

[], [enable\_fts4=yes])

AC\_MSG\_CHECKING([FTS4 extension])

if test x"$enable\_fts4" = "xyes"; then

BUILD\_CFLAGS="$BUILD\_CFLAGS -DSQLITE\_ENABLE\_FTS4"

AC\_MSG\_RESULT([enabled])

else

AC\_MSG\_RESULT([disabled])

fi

#-----------------------------------------------------------------------

#-----------------------------------------------------------------------

# --enable-fts3

#

AC\_ARG\_ENABLE(fts3, [AS\_HELP\_STRING(

[--enable-fts3], [include fts3 support [default=no]])],

[], [])

AC\_MSG\_CHECKING([FTS3 extension])

if test x"$enable\_fts3" = "xyes" -a x"$enable\_fts4" = "xno"; then

BUILD\_CFLAGS="$BUILD\_CFLAGS -DSQLITE\_ENABLE\_FTS3"

AC\_MSG\_RESULT([enabled])

else

AC\_MSG\_RESULT([disabled])

fi

#-----------------------------------------------------------------------

#-----------------------------------------------------------------------

# --enable-fts5

#

AC\_ARG\_ENABLE(fts5, [AS\_HELP\_STRING(

[--enable-fts5], [include fts5 support [default=yes]])],

[], [enable\_fts5=yes])

AC\_MSG\_CHECKING([FTS5 extension])

if test x"$enable\_fts5" = "xyes"; then

AC\_MSG\_RESULT([enabled])

AC\_SEARCH\_LIBS(log, m)

BUILD\_CFLAGS="$BUILD\_CFLAGS -DSQLITE\_ENABLE\_FTS5"

else

AC\_MSG\_RESULT([disabled])

fi

#-----------------------------------------------------------------------

#-----------------------------------------------------------------------

# --enable-rtree

#

AC\_ARG\_ENABLE(rtree, [AS\_HELP\_STRING(

[--enable-rtree], [include rtree support [default=yes]])],

[], [enable\_rtree=yes])

AC\_MSG\_CHECKING([RTREE extension])

if test x"$enable\_rtree" = "xyes"; then

BUILD\_CFLAGS="$BUILD\_CFLAGS -DSQLITE\_ENABLE\_RTREE -DSQLITE\_ENABLE\_GEOPOLY"

AC\_MSG\_RESULT([enabled])

else

AC\_MSG\_RESULT([disabled])

fi

#-----------------------------------------------------------------------

#-----------------------------------------------------------------------

# --enable-session

#

AC\_ARG\_ENABLE(session, [AS\_HELP\_STRING(

[--enable-session], [enable the session extension [default=no]])],

[], [])

AC\_MSG\_CHECKING([Session extension])

if test x"$enable\_session" = "xyes"; then

BUILD\_CFLAGS="$BUILD\_CFLAGS -DSQLITE\_ENABLE\_SESSION -DSQLITE\_ENABLE\_PREUPDATE\_HOOK"

AC\_MSG\_RESULT([enabled])

else

AC\_MSG\_RESULT([disabled])

fi

#-----------------------------------------------------------------------

#-----------------------------------------------------------------------

# --enable-debug

#

AC\_ARG\_ENABLE(debug, [AS\_HELP\_STRING(

[--enable-debug], [build with debugging features enabled [default=no]])],

[], [])

AC\_MSG\_CHECKING([Build type])

if test x"$enable\_debug" = "xyes"; then

BUILD\_CFLAGS="$BUILD\_CFLAGS -DSQLITE\_DEBUG -DSQLITE\_ENABLE\_SELECTTRACE -DSQLITE\_ENABLE\_WHERETRACE"

CFLAGS="-g -O0"

AC\_MSG\_RESULT([debug])

else

AC\_MSG\_RESULT([release])

fi

#-----------------------------------------------------------------------

#-----------------------------------------------------------------------

# --enable-static-shell

#

AC\_ARG\_ENABLE(static-shell, [AS\_HELP\_STRING(

[--enable-static-shell],

[statically link libsqlite3 into shell tool [default=yes]])],

[], [enable\_static\_shell=yes])

if test x"$enable\_static\_shell" = "xyes"; then

EXTRA\_SHELL\_OBJ=sqlite3-sqlite3.$OBJEXT

else

EXTRA\_SHELL\_OBJ=libsqlite3.la

fi

AC\_SUBST(EXTRA\_SHELL\_OBJ)

#-----------------------------------------------------------------------

AC\_CHECK\_FUNCS(posix\_fallocate)

AC\_CHECK\_HEADERS(zlib.h,[

AC\_SEARCH\_LIBS(deflate,z,[BUILD\_CFLAGS="$BUILD\_CFLAGS -DSQLITE\_HAVE\_ZLIB"])

])

AC\_SEARCH\_LIBS(system,,,[SHELL\_CFLAGS="-DSQLITE\_NOHAVE\_SYSTEM"])

AC\_SUBST(SHELL\_CFLAGS)

#-----------------------------------------------------------------------

# UPDATE: Maybe it's better if users just set CFLAGS before invoking

# configure. This option doesn't really add much...

#

# --enable-tempstore

#

# AC\_ARG\_ENABLE(tempstore, [AS\_HELP\_STRING(

# [--enable-tempstore],

# [in-memory temporary tables (never, no, yes, always) [default=no]])],

# [], [enable\_tempstore=no])

# AC\_MSG\_CHECKING([for whether or not to store temp tables in-memory])

# case "$enable\_tempstore" in

# never ) TEMP\_STORE=0 ;;

# no ) TEMP\_STORE=1 ;;

# always ) TEMP\_STORE=3 ;;

# yes ) TEMP\_STORE=3 ;;

# \* )

# TEMP\_STORE=1

# enable\_tempstore=yes

# ;;

# esac

# AC\_MSG\_RESULT($enable\_tempstore)

# AC\_SUBST(TEMP\_STORE)

#-----------------------------------------------------------------------

AC\_OUTPUT