

一般在linux上编译一个工程需要先调用./configure,大部分linux开源项目都使用autotools编译架构，都会有configure这个脚本。通过执行./configure，产生Makefile,src/Makefile,config.h，为下一步编译做准备。随后调用make进行编译。一般步骤如下：

```
./configure
make
make check
su
Password:***
make install
make installcheck
```

文件结构

make install将编译的程序，库或者头文件安装到系统中。
安装的路径由prefix决定，默认prefix=/usr/local

路径变量	默认值
prefix	/usr/local
exec-prefix	prefix
bindir	exec-prefix/bin
libdir	exec-prefix/lib
includedir	prefix/include
datarootdir	prefix/share
datadir	datarootdir
mandir	datarootdir/man
infodir	datarootdir/info

比如你可以这样改变prefix值

```
./configure --prefix ~/usr
make
make install
```

标准配置变量

./configure可以自动侦测这些变量设置。但是你也可以强制设置这些值。

变量	含义
CC	C编译命令
CFLAGS	C编译器参数
CXX	C++编译命令
CXXFLAGS	C++编译器参数
LDFLAGS	链接器参数
CPPFLAGS	C/C++预处理器参数

./configure -help可以列出全部的选项

```
./configure --prefix ~/usr CC=gcc-3 CPPFLAGS=-I$HOME/usr/include LDFLAGS=-L$HOME/usr/lib
```

make的一些常用参数

make all
等价于make，编译程序，库，文档等。

make install
安装编译好的程序和库，头文件到系统中。
make install-strip
和make install相同，但是剥离掉调试符号信息。
make uninstall
make install的反向操作。
make clean
删除编译产生的文件（make all的反向操作）
make distclean
删除一些./configure产生的文件。
make check
运行测试项
make installcheck
检查安装程序或库，如果支持的话。
make dist
创建一个压缩包文件 PACKAGE-VERSION.tar.gz

产生configure需要三个文件

configure.ac

Makefile.am

src/Makefile.am

（参考我的另外两篇文章[编写configure.ac](#)和[编写Makefile.am](#)）

编辑好这几个文件就可以产生configure脚本，和config.h.in，Makefile.in，src/Makefile.in

configure.ac主要初始化一些信息，检查编译器，检查头文件，检查库，检查函数等。

下面是一个configure.ac样本

```
AC_INIT([amhello], [1.0], [bug-report@address])
AM_INIT_AUTOMAKE([foreign -Wall -Werror])
AC_PROG_CC
AC_CONFIG_HEADERS([config.h])
AC_CONFIG_FILES([Makefile src/Makefile])
AC_OUTPUT
AC_INIT初始化工程名，版本，bug报告地址一个email
AM_INIT_AUTOMAKE 初始化automake
AC_PROG_CC检查编译器
AC_CONFIG_HEADERS产生config.h.in
AC_CONFIG_FILES 产生Makefile
AC_OUTPUT 输出这些文件 configure,config.h.in,Makefile
```

Makefile.am 指定建造目标程序，库还是动态库等

下面看一个例子

根目录下的Makefile.am

```
SUBDIRS=src
```

• 1

SUBDIRS表示建造路径为src，去到src下寻找Makefile.am

src/Makefile.am

```
bin_PROGRAMS=hello
```

```
hello_SOURCES=main.c
```

• 1

• 2

bin_PROGRAMS = hello表示建造一个应用程序hello，bin表示安装路径，PROGRAMS表示建造的是一个程序。

hello_SOURCES=main.c表示建造hello需要的源文件。

有两这些文件你输入下面的命令将会产生configure,config.h.in等文件

autoreconf -install

注意autoreconf其实会为你以一种正确的顺序去启动一系列的工具有。

工具	描述
autoconf	从configure.ac创建configure
autoheader	从configure.ac创建config.h.in
autoreconf	以正确的顺序运行全部命令
autoscan	搜索源代码中的移植问题，最终configure.ac中没提及的宏
autoupdate	跟新configure.ac中废弃的宏
ifnames	收集源代码中的#if/#ifdef/...并且标示定义
autom4te	autoconf的核心，它驱动M4通过下面的工具实现功能。
automake	从Makefile.am和configure.ac创建Makefile.in文件
aclocal	扫描configure.ac中的第三方宏，并且标示定义到aclocal.m4中