<https://www.cnblogs.com/tinywan/p/7230039.html>

这些都是典型的使用GNU的AUTOCONF和AUTOMAKE产生的程序的安装步骤

一、基本信息

　　1、./configure 是用来检测你的安装平台的目标特征的。比如它会检测你是不是有CC或GCC，并不是需要CC或GCC，它是个shell脚本。

　　2、make 是用来编译的，它从Makefile中读取指令，然后编译。

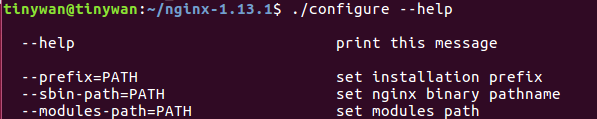
　　3、make install是用来安装的，它也从Makefile中读取指令，安装到指定的位置。

　　注意：AUTOMAKE和AUTOCONF是非常有用的用来发布C程序的东西。

二、详细解释

　　1、configure命令

　　这一步一般用来生成 Makefile，为下一步的编译做准备，你可以通过在 configure 后加上参数来对安装进行控制，比如代码:./configure –prefix=/usr 意思是将该软件安装在 /usr 下面，执行文件就会安装在 /usr/bin （而不是默认的 /usr/local/bin),资源文件就会安装在 /usr/share（而不是默认的/usr/local/share）。同时一些软件的配置文件你可以通过指定 –sys-config= 参数进行设定。有一些软件还可以加上 –with、–enable、–without、–disable 等等参数对编译加以控制，你可以通过允许 ./configure –help 察看详细的说明帮助。



　　2、make

　　这一步就是编译，大多数的源代码包都经过这一步进行编译（当然有些perl或python编写的软件需要调用perl或python来进行编译）。如果 在 make 过程中出现 error ，你就要记下错误代码（注意不仅仅是最后一行），然后你可以向开发者提交 bugreport（一般在 INSTALL 里有提交地址），或者你的系统少了一些依赖库等，这些需要自己仔细研究错误代码。

　　可能遇到的错误：make \*\*\* 没有指明目标并且找不到 makefile。 停止。问题很明了，没有Makefile，怎么办，原来是要先./configure 一下，再make。

　　3、make insatll

　　这条命令来进行安装（当然有些软件需要先运行 make check 或 make test 来进行一些测试），这一步一般需要你有 root 权限（因为要向系统写入文件）。

三、扩展说明

　　Linux的用户可能知道，在Linux下安装一个应用程序时，一般先运行脚本configure，然后用make来编译源程序，在运行make install，最后运行make clean删除一些临时文件。使用上述三个自动工具，就可以生成configure脚本。运行configure脚本，就可以生成Makefile文件，然后就可以运行make、make install和make clean。

　　configure是一个shell脚本，它可以自动设定源程序以符合各种不同平台上Unix系统的特性，并且根据系统叁数及环境产生合适的Makefile文件或是C的头文件(header file)，让源程序可以很方便地在这些不同的平台上被编译连接。

　　这时，就可运行configure脚本了，运行configure脚本，就可产生出符合GNU规范的Makefile文件了： $ ./configure

到此时，就可以运行make进行编译，在运行make install进行安装了，最后运行make clean删除临时文件。

$ make

$ make install (注：运行这个要有足够的权限)

$ make clean

　　利用configure所产生的Makefile文件有几个预设的目标可供使用，其中几个重要的简述如下：

　　make all：产生我们设定的目标，即此范例中的可执行文件。只打make也可以，此时会开始编译原始码，然后连结，并且产生可执行文件。

　　make clean：清除编译产生的可执行文件及目标文件(object file，\*.o)。

　　make distclean：除了清除可执行文件和目标文件外，把configure所产生的Makefile也清除掉。

　　make install：将程序安装至系统中。如果原始码编译无误，且执行结果正确，便可以把程序安装至系统预设的可执行文件存放路径。如果用bin\_PROGRAMS宏的话，程序会被安装至/usr/local/bin这个目录。

　　make dist：将程序和相关的档案包装成一个压缩文件以供发布。执行完在目录下会产生一个以PACKAGE-VERSION.tar.gz为名称的文件。 PACKAGE和VERSION这两个变数是根据configure.in文件中AM\_INIT\_AUTOMAKE(PACKAGE，VERSION)的定义。在此范例中会产生test-1.0.tar.gz的档案。

　　make distcheck：和make dist类似，但是加入检查包装后的压缩文件是否正常。这个目标除了把程序和相关文件包装成tar.gz文件外，还会自动把这个压缩文件解开，执行 configure，并且进行make all 的动作，确认编译无误后，会显示这个tar.gz文件可供发布了。这个检查非常有用，检查过关的包，基本上可以给任何一个具备GNU开发环境-的人去重新编译。

参考：<http://www.linuxidc.com/Linux/2011-02/32211.htm>