逍遥剑客的专栏 scucj

个人资料

scucj

访问： 415789次
积分： 4324
等级： BLOG 5
排名： 第5630名

原创： 73篇 转载： 6篇
译文： 0篇 评论： 178条

文章搜索

文章分类

ASP.NET (9)
C# (14)
C/C++ (8)
Database (3)
Java (1)
Other (9)
UML (0)
UNIX (3)
Web Service (0)
Windows编程 (2)
XML (0)
心情随笔 (0)
数据结构与算法 (2)
计算机网络 (0)
设计模式 (24)
软件工程 (3)

文章存档

2012年04月 (1)
2011年12月 (1)
2010年12月 (5)
2010年07月 (1)
2010年05月 (1)

展开

GNU Autotools的使用方法

标签： makefile include compiler file gcc header

2010-12-16 02:03 20614人阅读 评论(4) 收藏 举报

分类： UNIX (2)

手工写Makefile是一件很有趣的事情，对于比较大型的项目，如果有工具可以代劳，自然是一件好事。在Linux系统开发环境中，GNU Autotools 无疑就充当了这个重要角色。(在Windows系统的开发环境中，IDE [illegible] Visual Studio，来管理项目也很方便。)

本文以一个简单项目为例子，来讲述GNU Autotools的一列工具及其命令的用法。

autotools是系列工具, 它主要由autoconf、automake、perl语言环境和m4等组成；所包含的命令有五个:

(1) aclocal
(2) autoscan
(3) autoconf
(4) autoheader
(5) automake

一、准备源代码

(1) 目录project包含一个main.c的文件和两个子目录lib与include；lib目录中包含一个test.c，include目录中包含一个test.h。在系统中，显示如下：

```
[c-sharp]

[root@localhost project]# ls
include  lib  main.c
[root@localhost project]#
[root@localhost project]# ls include/
test.h
[root@localhost project]# ls lib/
test.c
[root@localhost project]#
```

(2)源代码如下：

```
[cpp]

/* project/main.c */
#include <stdio.h>
#include "include/test.h"
int main()
{
    printf("main entrance./n");
    test_method();
    return 0;
```

```
09. }
```

```
[c-sharp]

01. /* project/lib/test.c */
02. #include <stdio.h>
03. #include "../include/test.h"
04. void test_method()
05. {
06.     printf("test method./n");
07. }
```

```
[cpp]

01. /* project/include/test.h*/
02. void test_method();
```

二、autotools 使用步骤

2.1 使用autoscan命令，它将扫描工作目录，生成 configure.scan 文件。

```
[c-sharp]

01. [root@localhost project]# autoscan
02. autom4te: configure.ac: no such file or directory
03. autoscan: /usr/bin/autom4te failed with exit status: 1
04. [root@localhost project]# ls
05. autoscan.log  configure.scan  include  lib  main.c
06. [root@localhost project]#
```

2.2 将configure.scan 文件重命名为configure.ac，并做适当的修改。在 configure.ac 中，# 号开始的行是注释，其他都是m4 宏命令；configure.ac里面的宏的主要作用是侦测系统。

```
[c-sharp]

01. [root@localhost project]mv configure.scan configure.ac
02. [root@localhost project]# ls
03. autoscan.log  configure.ac include  lib  main.c
04. [root@localhost project]#
05. [root@localhost project]# cat configure.ac
06. #                                               -*- Autoconf -*-
07. # Process this file with autoconf to produce a configure script.
08. AC_PREREQ(2.59)
09. AC_INIT(FULL-PACKAGE-NAME, VERSION, BUG-REPORT-ADDRESS)
10. AC_CONFIG_SRCDIR([main.c])
11. AC_CONFIG_HEADER([config.h])
12. # Checks for programs.
13. AC_PROG_CC
14. # Checks for libraries.
15. # Checks for header files.
16. # Checks for typedefs, structures, and compiler characteristics.
17. # Checks for library functions.
18. AC_OUTPUT
19. [root@localhost project]#
```

2.3 对 configure.ac 文件做适当的修改，修改显示如下[1]：

```
[c-sharp]

01. [root@localhost project]# cat configure.ac
02. #                                               -*- Autoconf -*-
03. # Process this file with autoconf to produce a configure script.
04. AC_PREREQ(2.59)
05. #AC_INIT(FULL-PACKAGE-NAME, VERSION, BUG-REPORT-ADDRESS)
06. AC_INIT(hello,1.0,abc@126.com)
07. AM_INIT_AUTOMAKE(hello,1.0)
08. AC_CONFIG_SRCDIR([main.c])
09. AC_CONFIG_HEADER([config.h])
10. # Checks for programs.
```

```
11. AC_PROG_CC
12. # Checks for libraries.
13. # Checks for header files.
14. # Checks for typedefs, structures, and compiler characteristics.
15. # Checks for library functions.
16. AC_CONFIG_FILES([Makefile])
17. AC_OUTPUT
```

说明:

（1）以“#”号开始的行均为注释行。
（2）AC_PREREQ 宏声明本文要求的 autoconf 版本, 如本例中的版本 2.59。

（3）AC_INIT 宏用来定义软件的名称、版本等信息、作者的E-mail等。
（4）AM_INIT_AUTOMAKE是通过手动添加的, 它是automake所必备的宏, FULL-PACKAGE-NAME是软件名称,VERSION是软件版本号。
（5）AC_CONFIG_SCRDIR 宏用来侦测所指定的源码文件是否存在, 来确定源码目录的有效性.。此处为当前目录下main.c。

（6）AC_CONFIG_HEADER 宏用于生成config.h文件，以便 autoheader 命令使用。
（7）AC_PROG_CC用来指定编译器，如果不指定，默认gcc。
（8）AC_OUTPUT 用来设定 configure 所要产生的文件，如果是makefile，configure 会把它检查出来的结果带入makefile.in文件产生合适的makefile。使用 Automake 时，还需要一些其他的参数，这些额外的宏用aclocal工具产生。
（9）AC_CONFIG_FILES宏用于生成相应的Makefile文件。

2.4 使用 aclocal 命令，扫描 configure.ac 文件生成 aclocal.m4文件, 该文件主要处理本地的宏定义，它根据已经安装的宏、用户定义宏和 acinclude.m4 文件中的宏将 configure.ac 文件需要的宏集中定义到文件 aclocal.m4 中。[2]

```
[c-sharp]

01. [root@localhost project]# aclocal
02. [root@localhost project]# ls
03. aclocal.m4  autom4te.cache  autoscan.log  configure.in  include  lib  main.c
04. [root@localhost project]#
```

2.5 使用 autoconf 命令生成 configure 文件。这个命令将 configure.ac 文件中的宏展开，生成 configure 脚本。这个过程可能要用到aclocal.m4中定义的宏。

```
[c-sharp]

01. [root@localhost project]# autoconf
02. [root@localhost project]# ls
03. aclocal.m4  autom4te.cache  autoscan.log  configure  configure.in  include  lib  main.c
```

2.6 使用 autoheader 命令生成 config.h.in 文件。该命令通常会从 "acconfig.h” 文件中复制用户附加的符号定义。该例子中没有附加的符号定义, 所以不需要创建 "acconfig.h” 文件[2].

```
[cpp]

01. [root@localhost project]# autoheader
02. [root@localhost project]# ls
03. aclocal.m4  autom4te.cache  autoscan.log  config.h.in  configure  configure.in  include  lib  mair
04. [root@localhost project]#
```

2.7 手工创建Makefile.am文件。Automake工具会根据 configure.in 中的参量把 Makefile.am 转换成 Makefile.in 文件。

```
[cpp]

01. [root@localhost project]# cat Makefile.am
02. UTOMAKE_OPTIONS = foreign
03. bin_PROGRAMS = hello
04. hello_SOURCES = main.c include/test.h lib/test.c
```

说明:

（1）其中的AUTOMAKE_OPTIONS为设置automake的选项. 由于GNU对自己发布的软件有严格的规范, 比如必须附带许可证声明文件COPYING等，否则automake执行时会报错. automake提供了3中软件等级:foreign, gnu和gnits, 供用户选择。默认级别是gnu. 在本例中， 使用了foreign等级, 它只检测必须的文件。

（2）bin_PROGRAMS定义要产生的执行文件名. 如果要产生多个执行文件, 每个文件名用空格隔开。
（3）hello_SOURCES 定义"hello"这个可执行程序所需的原始文件。如果"hello"这个程序是由多个源文件所产生的, 则必须把它所用到的所有源文件都列出来，并用空格隔开。如果要定义多个可执行程序，那么需要对每个可执行程序建立对应的file_SOURCES。

2.8 使用 Automake 命令生成 Makefile.in 文件。使用选项 "--add-missing" 可以让 Automake 自动添加一些必需的脚本文件。

```
[cpp]

[root@localhost project]# automake --add-missing
configure.ac: installing `./install-sh'
configure.ac: installing `./missing'
Makefile.am: installing `./INSTALL'
Makefile.am: required file `./NEWS' not found
Makefile.am: required file `./README' not found
Makefile.am: required file `./AUTHORS' not found
Makefile.am: required file `./ChangeLog' not found
Makefile.am: installing `./COPYING'
Makefile.am: installing `./depcomp'
[root@localhost project]#
```

2.8.1 再次使用 automake ——add-missing 运行一次，可以辅助生成几个必要的文件。

```
[cpp]

[root@localhost project]# automake --add-missing
Makefile.am: required file `./NEWS' not found
Makefile.am: required file `./README' not found
Makefile.am: required file `./AUTHORS' not found
Makefile.am: required file `./ChangeLog' not found
[root@localhost project]# ls
aclocal.m4  autom4te.cache  autoscan.log  config.h.in  config.h.in~  configure  configure.ac  COPY
[root@localhost project]#
```

2.8.2 在当前目录创建上面未发现的四个文件，并再次使用 automake ——add-missing 运行一次。

```
[c-sharp]

[root@localhost project]# touch NEWS
[root@localhost project]# touch README
[root@localhost project]# touch AUTHORS
[root@localhost project]# touch ChangeLog
[root@localhost project]#
[root@localhost project]# automake --add-missing
[root@localhost project]# ls
aclocal.m4  autom4te.cache  ChangeLog    config.h.in~  config.status  configure.ac  depcomp  INSTA
AUTHORS     autoscan.log    config.h.in  config.log    configure      COPYING       include  insta
[root@localhost project]#
```

2.9 使用 configure 命令， 把 Makefile.in 变成最终的 Makefile 文件。

```
[c-sharp]

[root@localhost project]# ./configure
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking for gcc... gcc
```

```
checking for C compiler default output file name... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ANSI C... none needed
checking for style of include used by make... GNU
checking dependency style of gcc... gcc3
configure: creating ./config.status
config.status: creating Makefile
config.status: creating config.h
config.status: config.h is unchanged
config.status: executing depfiles commands
[root@localhost project]# ls
aclocal.m4  autom4te.cache  ChangeLog  config.h.in   config.log     configure     COPYING  hello
AUTHORS     autoscan.log    config.h   config.h.in~  config.status  configure.ac  depcomp  include
[root@localhost project]#
```

Makefile文件已经生成成功。

三、Makefile的用法

3.1 make 命令，用来编译代码， 默认执行"make all"命令，可以看到生成了"hello"的可执行文件，

```
[c-sharp]

[root@localhost project]# make
make  all-am
make[1]: Entering directory `/home/chenjie/project'
gcc  -g -O2   -o hello  main.o test.o
make[1]: Leaving directory `/home/chenjie/project'
[root@localhost project]#
[root@localhost project]# ls
aclocal.m4  autom4te.cache  ChangeLog  config.h.in   config.log     configure     COPYING  hello
AUTHORS     autoscan.log    config.h   config.h.in~  config.status  configure.ac  depcomp  include
[root@localhost project]#
```

3.2 make clean 命令清除编译时的obj文件，它与 make 命令是对应关系，一个是编译，一个清除编译的文件

3.3 运行"./hello"就能看到运行结果:

```
[c-sharp]

[root@localhost project]# ./hello
main entrance.
test method.
[root@localhost project]#
```

3.4 make install 命令把目标文件安装到系统中。这一，直接输入hello, 就可以看到程序的运行结果。

```
[c-sharp]

[root@localhost project]# make install
make[1]: Entering directory `/home/chenjie/project'
test -z "/usr/local/bin" || mkdir -p -- "/usr/local/bin"
  /usr/bin/install -c 'hello' '/usr/local/bin/hello'
make[1]: Nothing to be done for `install-data-am'.
make[1]: Leaving directory `/home/chenjie/project'
[root@localhost project]#
[root@localhost project]# hello
main entrance.
```

```
test method.
[root@localhost project]#
```

3.5 make uninstall 命令把目标文件从系统中卸载。

3.6 make dist 命令将程序和相关的文档打包为一个压缩文档以供发布，在本例子中，生成的打包文件名为：hello-1.0.tar.gz。

```
[c-sharp]

[root@localhost project]# make dist
{ test ! -d hello-1.0 || { find hello-1.0 -type d ! -perm -200 -exec chmod u+w {} ';' && rm -fr he
mkdir hello-1.0
find hello-1.0 -type d ! -perm -755 -exec chmod a+rwx,<a href="http://lib.csdn.net/base
/go" class='replace_word' title="Go知识库" target='_blank' style='color:#df3434; font-
weight:bold;'>Go</a>+rx {} /; -o /
          ! -type d ! -perm -444 -links 1 -exec chmod a+r {} /; -o /
          ! -type d ! -perm -400 -exec chmod a+r {} /; -o /
          ! -type d ! -perm -444 -exec /bin/sh /home/chenjie/project
/install-sh -c -m a+r {} {} /; /
        || chmod -R a+r hello-1.0
tardir=hello-1.0 && /bin/sh /home/chenjie/project
/missing --run tar chof - "$tardir" | GZIP=--best gzip -c >hello-1.0.tar.gz
{ test ! -d hello-1.0 || { find hello-1.0 -type d ! -perm -200 -exec chmod u+w {} ';' && rm -fr he
[root@localhost project]# ls
aclocal.m4  autom4te.cache  ChangeLog  config.h.in   config.log     configure     COPYING  hello
AUTHORS     autoscan.log    config.h   config.h.in~  config.status  configure.ac  depcomp  hello-
1.0.tar.gz  INSTALL  lib         main.o  Makefile.am  missing      README  test.o
[root@localhost project]#
```

四 如何使用已发布的压缩文档

4.1 下载到“hello-1.0.tar.gz”压缩文档

4.2 使用“ tar -zxvf hello-1.0.tar.gz ”命令解压

4.3 使用 “./configure” 命令，主要的作用是对即将安装的软件进行配置，检查当前的环境是否满足要安装软件的依赖关系。

4.4 使用“ make ” 命令编译源代码文件生成软件包。

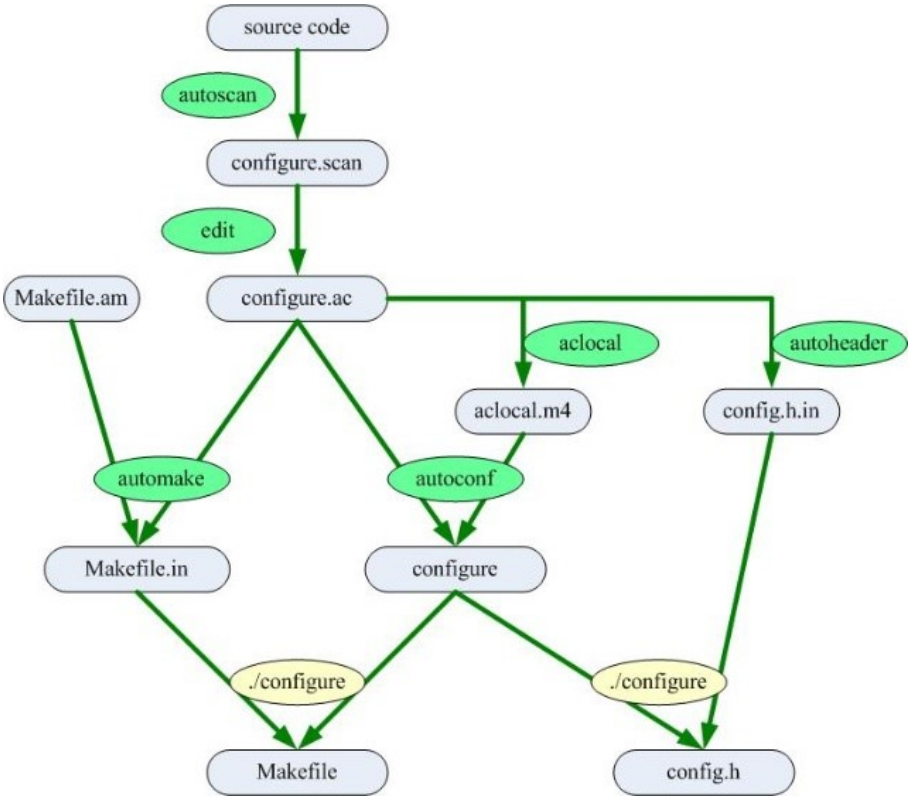
4.5 使用 “ make install ”命令来安装编译后的软件包。

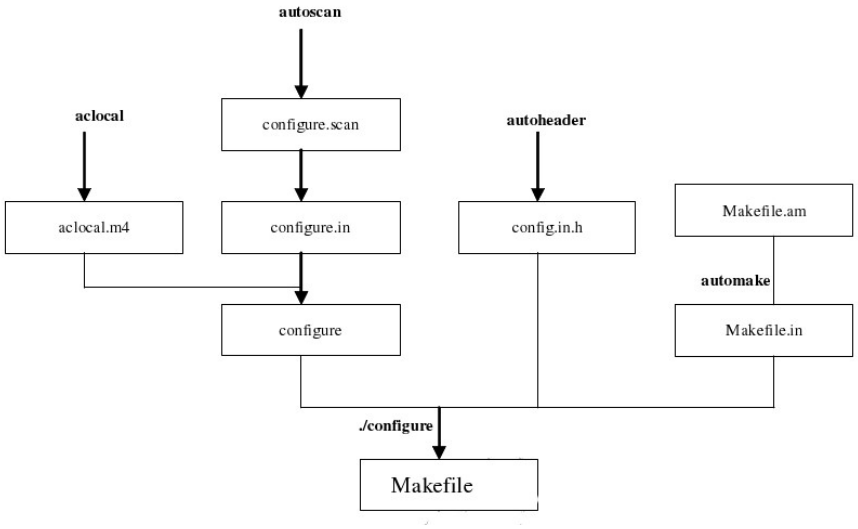
```
[c-sharp]

[root@localhost chenjie]# ls
hello-1.0.tar.gz
[root@localhost chenjie]# tar -zxvf hello-1.0.tar.gz
[root@localhost chenjie]# ls
hello-1.0  hello-1.0.tar.gz
[root@localhost chenjie]# cd hello-1.0
[root@localhost hello-1.0]# ls
aclocal.m4  AUTHORS  ChangeLog  config.h.in  configure  configure.ac  COPYING  depcomp  include  I
[root@localhost hello-1.0]#
[root@localhost hello-1.0]#
[root@localhost hello-1.0]# ./configure
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking for gcc... gcc
checking for C compiler default output file name... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
```

```
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ANSI C... none needed
checking for style of include used by make... GNU
checking dependency style of gcc... gcc3
configure: creating ./config.status
config.status: creating Makefile
config.status: creating config.h
config.status: executing depfiles commands
[root@localhost hello-1.0]#
[root@localhost hello-1.0]# make
make  all-am
make[1]: Entering directory `/home/chenjie/hello-1.0'
if gcc -DHAVE_CONFIG_H -I. -I. -I.     -g -O2 -MT main.o -MD -MP -MF ".deps/main.Tpo" -c -o main.o
        then mv -f ".deps/main.Tpo" ".deps/main.Po"; else rm -f ".deps/main.Tpo"; exit 1; fi
if gcc -DHAVE_CONFIG_H -I. -I. -I.     -g -O2 -MT test.o -MD -MP -MF ".deps/test.Tpo" -c -o test.o
/test.c; /
        then mv -f ".deps/test.Tpo" ".deps/test.Po"; else rm -f ".deps/test.Tpo"; exit 1; fi
gcc  -g -O2   -o hello  main.o test.o
make[1]: Leaving directory `/home/chenjie/hello-1.0'
[root@localhost hello-1.0]#
[root@localhost hello-1.0]# make install
make[1]: Entering directory `/home/chenjie/hello-1.0'
test -z "/usr/local/bin" || mkdir -p -- "/usr/local/bin"
  /usr/bin/install -c 'hello' '/usr/local/bin/hello'
make[1]: Nothing to be done for `install-data-am'.
make[1]: Leaving directory `/home/chenjie/hello-1.0'
[root@localhost hello-1.0]#
[root@localhost hello-1.0]# hello
main entrance.
test method.
```

五、命令使用的整个流程图

图我就不画了，转载两个图[2][3]，对比着看，或许更明白一些。





六、总结

本文描述了如果使用GNU Autotools的来管理源代码，发布源代码包，以及获得源代码包后如何编译、安装。由于这个例子过于简单，GNU Autotools的用法还未完全描述清楚，主要体现在以下几点：

（1）在创建 Makefile.am 文件中，描述的很简单。在实际的项目中，文件关系很复杂，而且还有引用其他动态库、第三方动态库等关系。

（2）虽然 makefile 是自动生成的，但是了解它的规则是非常重要的。makefile 涉及到的规则本文并未加以描述。

有空的时候再写一篇blog来描述上述两个问题。

[1] http://book.chinaitlab.com/linux/777286.html

[2] http://blog.ossxp.com/2010/04/954/

上一篇 Windows系统和Linux系统中的静态链接库与动态链接库(三)
下一篇 Source Insight的一点小技巧

我的同类文章

UNIX（2）

- Source Insight的一点小技巧 2010-12-18 阅读 4169
- linux下的FTP,SAMBA配置笔记 2008-10-21 阅读 3192

查看评论

3楼 叶子还在书上 2016-04-17 11:05发表

请问我可以转载到我的博客作为学习笔记吗？

2楼 zhangwiwi 2014-03-25 14:37发表

2.2 将configure.scan 文件重命名为configure.ac，并做适当的修改 楼主这边有个小错误,必须得将configure.scan 文件重命名为configure.in才行。

Re: ic499747 2016-02-29 10:13发表

回复zhangwiwi：重命名为configure.ac或configure.in都可以，用configure.ac是ok的，ubuntu环境

1楼 on_1y 2013-11-05 22:58发表

不错，很有帮助。