转 Linux configure 参数解释

2013年10月19日 17:01:27　阅读数：2360

Linux环境下的软件安装，并不是一件容易的事情；如果通过源代码编译后在安装，当然事情就更为复杂一些；现在安装各种软件的教程都非常普遍；但万变不离其中，对基础知识的扎实掌握，安装各种软件的问题就迎刃而解了。Configure脚本配置工具就是基础之一，它是autoconf的工具的基本应用。

与一些技巧相比，Configure显得基础一些，当然使用和学习起来就显得枯燥乏味一些，当然要成为高手，对基础的熟悉不能超越哦。

为此我转载了一篇关于Configure选项配置的详细介绍。供大家参考

'configure'脚本有大量的命令行选项.对不同的软件包来说,这些选项可能会有变化,但是许多基本的选项是不会改变的.带上'--help'选项执行'configure'脚本可以看到可用的所有选项.尽管许多选项是很少用到的,但是当你为了特殊的需求而configure一个包时,知道他们的存在是很有益处的.下面对每一个选项进行简略的介绍:

--cache-file=FILE
'configure'会在你的系统上测试存在的特性(或者bug!).为了加速随后进行的配置,测试的结果会存储在一个cache file里.当configure一个每个子树里都有'configure'脚本的复杂的源码树时,一个很好的cache file的存在会有很大帮助.

--help
输出帮助信息.即使是有经验的用户也偶尔需要使用使用'--help'选项,因为一个复杂的项目会包含附加的选项.例如,GCC包里的'configure'脚本就包含了允许你控制是否生成和在GCC中使用GNU汇编器的选项.

--no-create
'configure'中的一个主要函数会制作输出文件.此选项阻止'configure'生成这个文件.你可以认为这是一种演习(dry run),尽管缓存(cache)仍然被改写了.

--quiet
--silent
当'configure'进行他的测试时,会输出简要的信息来告诉用户正在作什么.这样作是因为'configure'可能会比较慢,没有这种输出的话用户将会被扔在一旁疑惑正在发生什么.使用这两个选项中的任何一个都会把你扔到一旁.(译注:这两句话比较有意思,原文是这样的:If there was no such output, the user would be left wondering what is happening. By using this option, you too can be left wondering!)

--version
打印用来产生'configure'脚本的Autoconf的版本号.

--prefix=PEWFIX
'--prefix'是最常用的选项.制作出的'Makefile'会查看随此选项传递的参数,当一个包在安装时可以彻底的重新安置他的结构独立部分. 举一个例子,当安装一个包,例如说Emacs,下面的命令将会使Emacs Lisp file被安装到"/opt/gnu/share":
$ ./configure --prefix=/opt/gnu

--exec-prefix=EPREFIX
与'--prefix'选项类似,但是他是用来设置结构倚赖的文件的安装位置.编译好的'emacs'二进制文件就是这样一个问件.如果没有设置这个选项的话,默认使用的选项值将被设为和'--prefix'选项值一样.

--bindir=DIR
指定二进制文件的安装位置.这里的二进制文件定义为可以被用户直接执行的程序.

--sbindir=DIR
指定超级二进制文件的安装位置.这是一些通常只能由超级用户执行的程序.

--libexecdir=DIR
指定可执行支持文件的安装位置.与二进制文件相反,这些文件从来不直接由用户执行,但是可以被上面提到的二进制文件所执行.

--datadir=DIR
指定通用数据文件的安装位置.

--sysconfdir=DIR
指定在单个机器上使用的只读数据的安装位置.

--sharedstatedir=DIR
指定可以在多个机器上共享的可写数据的安装位置.

--localstatedir=DIR
指定只能单机使用的可写数据的安装位置.

--libdir=DIR
指定库文件的安装位置.

--includedir=DIR
指定C头文件的安装位置.其他语言如C++的头文件也可以使用此选项.

--oldincludedir=DIR

指定为除GCC外编译器安装的C头文件的安装位置.

--infodir=DIR
指定Info格式文档的安装位置.Info是被GNU工程所使用的文档格式.

--mandir=DIR
指定手册页的安装位置.

--srcdir=DIR
这个选项对安装没有作用.他会告诉'configure'源码的位置.一般来说不用指定此选项,因为'configure'脚本一般和源码文件在同一个目录下.

--program-prefix=PREFIX
指定将被加到所安装程序的名字上的前缀.例如,使用'--program-prefix=g'来configure一个名为'tar'的程序将会使安装的程序被命名为'gtar'.当和其他的安装选项一起使用时,这个选项只有当他被`Makefile.in'文件使用时才会工作.

--program-suffix=SUFFIX
指定将被加到所安装程序的名字上的后缀.

--program-transform-name=PROGRAM
这里的PROGRAM是一个sed脚本.当一个程序被安装时,他的名字将经过`sed -e PROGRAM'来产生安装的名字.

--build=BUILD
指定软件包安装的系统平台.如果没有指定,默认值将是'--host'选项的值.

--host=HOST
指定软件运行的系统平台.如果没有指定,将会运行`config.guess'来检测.

--target=GARGET
指定软件面向(target to)的系统平台.这主要在程序语言工具如编译器和汇编器上下文中起作用.如果没有指定,默认将使用'--host'选项的值.

--disable-FEATURE
一些软件包可以选择这个选项来提供为大型选项的编译时配置,例如使用Kerberos认证系统或者一个实验性的编译器最优配置.如果默认是提供这些特性,可以使用'--disable-FEATURE'来禁用它,这里'FEATURE'是特性的名字.例如:
$ ./configure --disable-gui

-enable-FEATURE[=ARG]
相反的,一些软件包可能提供了一些默认被禁止的特性,可以使用'--enable-FEATURE'来起用它.这里'FEATURE'是特性的名字.一个特性可能会接受一个可选的参数.例如:
$ ./configure --enable-buffers=128
`--enable-FEATURE=no'与上面提到的'--disable-FEATURE'是同义的.

--with-PACKAGE[=ARG]
在自由软件社区里,有使用已有软件包和库的优秀传统.当用'configure'来配置一个源码树时,可以提供其他已经安装的软件包的信息.例如,倚赖于Tcl和Tk的BLT器件工具包.要配置BLT,可能需要给'configure'提供一些关于我们把Tcl和Tk装的何处的信息:
$ ./configure --with-tcl=/usr/local --with-tk=/usr/local
'--with-PACKAGE=no'与下面将提到的'--without-PACKAGE'是同义的.

--without-PACKAGE
有时候你可能不想让你的软件包与系统已有的软件包交互.例如,你可能不想让你的新编译器使用GNU ld.通过使用这个选项可以做到这一点:
$ ./configure --without-gnu-ld

--x-includes=DIR
这个选项是'--with-PACKAGE'选项的一个特例.在Autoconf最初被开发出来时,流行使用'configure'来作为Imake的一个变通方法来制作运行于X的软件.'--x-includes'选项提供了向'configure'脚本指明包含X11头文件的目录的方法.

--x-libraries=DIR
类似的,'--x-libraries'选项提供了向'configure'脚本指明包含X11库的目录的方法.

在源码树中运行'configure'是不必要的同时也是不好的.一个由'configure'产生的良好的'Makefile'可以构筑源码属于另一棵树的软件包.在一个独立于源码的树中构筑派生的文件的好处是很明显的:派生的文件,如目标文件,会凌乱的散布于源码树.这也使在另一个不同的系统或用不同的配置选项构筑同样的目标文件非常困难.建议使用三棵树:一棵源码树(source tree),一棵构筑树(build tree),一棵安装树(install tree).这里有一个很接近的例子,是使用这种方法来构筑GNU malloc包:
$ gtar zxf mmalloc-1.0.tar.gz
$ mkdir build && cd build
$ ../mmalloc-1.0/configure
creating cache ./config.cache
checking for gcc... gcc
checking whether the C compiler (gcc) works... yes
checking whether the C compiler (gcc) is a cross-compiler... no
checking whether we are using GNU C... yes
checking whether gcc accepts -g... yes
checking for a BSD compatible install... /usr/bin/install -c
checking host system type... i586-pc-linux-gnu
checking build system type... i586-pc-linux-gnu
checking for ar... ar
checking for ranlib... ranlib

checking how to run the C preprocessor... gcc -E
checking for unistd.h... yes
checking for getpagesize... yes
checking for working mmap... yes
checking for limits.h... yes
checking for stddef.h... yes
updating cache ../config.cache
creating ./config.status
这样这棵构筑树就被配置了,下面可以继续构筑和安装这个包到默认的位置'/usr/local':
$ make all && make install?

文章标签： Linux configure 参数

个人分类： 纯编程