编译安装 gcc 4.9并验证使用  
  
1. 准备环境(GCC 编译器)  
centOS 6.2  
cat /proc/version   
Linux version 2.6.32-279.el6.x86\_64 (mockbuild@c6b9.bsys.dev.centos.org) (gcc version 4.4.6 20120305 (Red Hat 4.4.6-4) (GCC) ) #1 SMP Fri Jun 22 12:19:21 UTC 2012  
gcc --version  
gcc (GCC) 4.4.6 20120305 (Red Hat 4.4.6-4)  
  
2. 准备文件  
gcc-4.9.0.tar.gz  
官网(http://gcc.gnu.org/)下载:

ftp://mirrors.kernel.org/gnu/gcc/gcc-4.9.0/gcc-4.9.0.tar.gz

http://gcc.petsads.us/releases/

3. 安装步骤  
3.1. 解压 gcc-4.9.0.tar.gz  
得到目录 gcc-4.9.0,进入目录  
#tar -xvzf gcc-4.9.0.tar.gz  
#cd gcc-4.9.0  
  
3.2. 下载编译准备文件  
主要是需要下面的库文件(需要完整版本,下载全部文件)  
MPFR=mpfr-2.4.2  
GMP=gmp-4.3.2  
MPC=mpc-0.8.1  
ISL=isl-0.12.2  
CLOOG=cloog-0.18.1  
执行./contrib/download\_prerequisites  
如果编译机器不能上网，直接打开文本文件 download\_prerequisites,把里面  
依赖的库下载回来后，放在指定目录，然后注释下载命令，再次执行  
contrib/download\_prerequisites,把相关文件解压，并建立链接。  
#vi contrib/download\_prerequisites  
注释 wget 相关的内容,手动把这些文件下载回来,拷贝到工作目录 gcc-4.9.0下。  
#./contrib/download\_prerequisites  
  
3.3. 创建编译目录并编译安装  
#mkdir ../gcc-build-4.9.0

#cd ../gcc-build-4.9.0

#../gcc-4.9.0/configure --prefix=/usr/local/gcc-4.9.0 --enable-stage1-checking=release --enable-stage1-languages=c,c++,go

#make -j 4

#make check 无需  
#make install  
编译后安装到  /usr/local/gcc-4.9.0

更加好的编译方法见下：

安装新的gcc后要把旧的切换为新的

update-alternatives –instal /usr/bin/gcc gcc /usr/local/bin/i686-pc-linux-gnu-gcc 40

/usr/bin/gcc为连接文件，gcc为名字,后面为路径,若连接文件存在则通过优先级如40来默认连接哪一个；

设置后可能需要关机重启，在查看gcc –v

Centos升级gcc4.4.7升级gcc4.8手记

Posted on: [2014年08月23日](http://www.mudbest.com/centos%e5%8d%87%e7%ba%a7gcc4-4-7%e5%8d%87%e7%ba%a7gcc4-8%e6%89%8b%e8%ae%b0/) by [hkshadow](http://www.mudbest.com/author/hkshadow/)

出于需求系统自带的是gcc4.4.7，版本太低无法编译，被逼无奈只能升级gcc了，碰到很多坑，下载这个玩意就用了不少时间。

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | [root@VM\_222\_92\_centos ~]# gcc -v  Using built-in specs.  Target: x86\_64-redhat-linux  Configured with: ../configure --prefix=/usr --mandir=/usr/share/man --infodir=/usr/share/info --with-bugurl=http://bugzilla.redhat.com/bugzilla --enable-bootstrap --enable-shared --enable-threads=posix --enable-checking=release --with-system-zlib --enable-\_\_cxa\_atexit --disable-libunwind-exceptions --enable-gnu-unique-object --enable-languages=c,c++,objc,obj-c++,java,fortran,ada --enable-java-awt=gtk --disable-dssi --with-java-home=/usr/lib/jvm/java-1.5.0-gcj-1.5.0.0/jre --enable-libgcj-multifile --enable-java-maintainer-mode --with-ecj-jar=/usr/share/java/eclipse-ecj.jar --disable-libjava-multilib --with-ppl --with-cloog --with-tune=generic --with-arch\_32=i686 --build=x86\_64-redhat-linux  Thread model: posix  gcc version 4.4.7 20120313 (Red Hat 4.4.7-4) (GCC) |

1.下载源码包

|  |  |
| --- | --- |
| 1  2 | wget http: //ftp.gnu.org/gnu/gcc/gcc-4.8.0/gcc-4.8.0.tar.bz2  tar -jxvf  gcc-4.8.0.tar.bz2 |

2.下载编译所需依赖库

|  |  |
| --- | --- |
| 1  2  3 | cd gcc-4.8.0  ./contrib/download\_prerequisites ----这里不能上网就只能参照上面的了  cd .. |

3.建立编译输出目录

|  |  |
| --- | --- |
| 1 | mkdir gcc-build-4.8.0 |

4.进入此目录，执行以下命令，生成makefile文件

|  |  |
| --- | --- |
|  | cd  gcc-build-4.8.0  ../gcc-4.8.0/configure --enable-checking=release --enable-languages=c,c++ --disable-multilib |

5.编译

|  |  |
| --- | --- |
| 1  2 | # j 后面的是核心数，编译速度会比较快  make -j4 |

6.安装

|  |  |
| --- | --- |
| 1 | sudo make install |

7.切换GCC到新版  
确定新安装的GCC的路径,一般默认在/usr/local/bin下。可以先updatedb,然后locate gcc-4.8|tail找一下

|  |  |
| --- | --- |
| 1 | ls /usr/local/bin | grep gcc |

添加新GCC到可选项，倒数第三个是名字，倒数第二个参数为新GCC路径，最后一个参数40为优先级，设大一些之后就自动使用新版了

|  |  |
| --- | --- |
| 1 | update-alternatives --install /usr/bin/gcc gcc /usr/local/bin/i686-pc-linux-gnu-gcc 40 |

8.确认当前版本已经切换为新版.

|  |  |
| --- | --- |
| 1 | gcc -v |



现在来说下坑吧，以上的步骤完全不需要画蛇添足，几个步骤不多不少，我之前对第二步进行单独的make,make check，结果悲剧了，各种报错。碰到第二坑就是磁盘问题，上面说系统盘只有8G，编译过程中就报了磁盘不足，刚刚发现这个新服务器的数据盘没有挂载。

**挂在部分：**

关于update-alternatives连接的使用：

 update-alternatives是ubuntu系统中专门维护系统命令链接符的工具，通过它可以很方便的设置系统默认使用哪个命令、哪个软件版本，比如，我们在系统中同时安装了open jdk和sun jdk两个版本，而我们又希望系统默认使用的是sun jdk，那怎么办呢？通过update-alternatives就可以很方便的实现了。

  sudo update-alternatives --install /usr/bin/java java /opt/java/jdk1.6.0\_27/bin/java 1062

其中1062 为alternative更改后的优先级，新更改的优先级需要大于当前的，通过update-alternatives --display java 可以查看。

现在介绍下 update-alternatives 命令中的几个主要选项，diplay、install、remove、config；

   1、display选项的功能就是查看一个命令链接组的所有信息，包括链接的模式(自动还是手动)、链接priority值、所有可 用的链接命令等等。使用语法：update-alternatives --display name ,其中name为命令链接名称，如java，执行命令: update-alternatives --display java 将会出现类似下面的结果

java - auto mode

        link currently points to /opt/java/jdk1.6.0\_27/bin/java

/opt/java/jdk1.6.0\_27/bin/java - priority 1062

/usr/lib/jvm/java-6-openjdk/jre/bin/java - priority 1061

  slave java.1.gz: /usr/lib/jvm/java-6-openjdk/jre/man/man1/java.1.gz

Current 'best' version is '/opt/java/jdk1.6.0\_27/bin/java'.

   2、install选项的功能就是增加一组新的系统命令链接符了，使用语法为：update-alternatives --install link name path priority [--slave link name path]... 其中link为系统中功能相同软件的公共链接目录，比如/usr/bin/java(需绝对目录); name为命令链接符名称,如java； path为你所要使用新命令、新软件的所在目录； priority为优先级，当命令链接已存在时，需高于当前值，因为当alternative为自动模式时,系统默认启用priority高的链接; --slave为从alternative。

    alternative有两种模式：auto和manual，默认都为auto模式，因为大多数情况下update-alternatives命令都被postinst (configure) or prerm (install)调用的，如果将其更改成手动的话安装脚本将不会更新它了。

   3、remove选项的功能是删除一个alternative及相关从alternative,使用语法为:update-alternatives --remove name path,其中name与path与install中的一致，如果所删除的链接组中还有其他链接的话，系统将会自动从其他中选择一个priority高的链接作为默认为链接。

   4、config选项功能为在现有的命令链接选择一个作为系统默认的，使用语法为：update-alternatives --config name,如执行：sudo update-alternatives --config java，将会有类似以下信息：

There are 2 choices for the alternative java (providing /usr/bin/java).

  Selection    Path                                      Priority   Status

------------------------------------------------------------

\* 0            /opt/java/jdk1.6.0\_27/bin/java             1062      auto mode

  1            /opt/java/jdk1.6.0\_27/bin/java             1062      manual mode

  2            /usr/lib/jvm/java-6-openjdk/jre/bin/java   1061      manual mode

Press enter to keep the current choice[\*], or type selection number:

   若是想将manual改回自动模式的话，执行命令: update-alternatives --auto name即可，如

 update-alternatives --config  java