Borealis 安装说明

编写人：吴西送

日期：2012-11-05

目　录

安装 Borealis

1、首先安装 Packages Needed For Borealis.

Borealis 依赖若干自由软件包，其中一些包是可选的（optional），所以只有当你有特定需要时才需要安装这些可选的包。

许多包的版本与 Borealis 不相容，或旧或新的版本都有可能造成平台的不稳定工作，因此我们需要安装指定版本高的包，这些版本会在下面一一给出。

在安装之前，要考虑到将这些依赖包安装在哪里，在此，我们在/opt 下新建了一个 borealis-tools/src 文件夹，将这些依赖包的源文件放在 src 文件夹里，将包的链接放在 borealis-tools 文件夹下。

下面是 borealis 外部以来包的链接，链接中提供了下载以及安装的具体方法：GNU Gcc, CCache, Antlr, Xerces, Libtool / Autoconf / Automake, Berkeley-Db, Glpk, Gsl, Open Computer Vision, and Doxygen.

1.1、Tools Used to Build Borealis

我们需要编译的外部依赖有：antlr ccache gcc db glpk gsl nmstl xercesc，这些类库都可以在网上下载到相应的版本，下载的源代码可以存放在 /opt/borealis-tools/src。还有其他的外部依赖可以在 Ubuntu 的软件源下安装，比如libtool等。(所以，在这里我们应该首先安装这几个需要独立编译的外部依赖，这样会减少不必要的时间浪费!)

1.1.1Gcc（GCC 要首先安装）

http://gcc.gnu.org

Ubuntu 软件源里的 gcc 版本是 4.5.2 无法用来编译 borealis,官方安装文档中推荐的版本为 4.1.1，所以我们要先安装 gcc 4.1.1，编译方法为：

```
cd /opt/borealis-tools/src
 解压 tar -vxjf gcc-4.1.1.tar.bz2
 cd  gcc-4.1.1
```

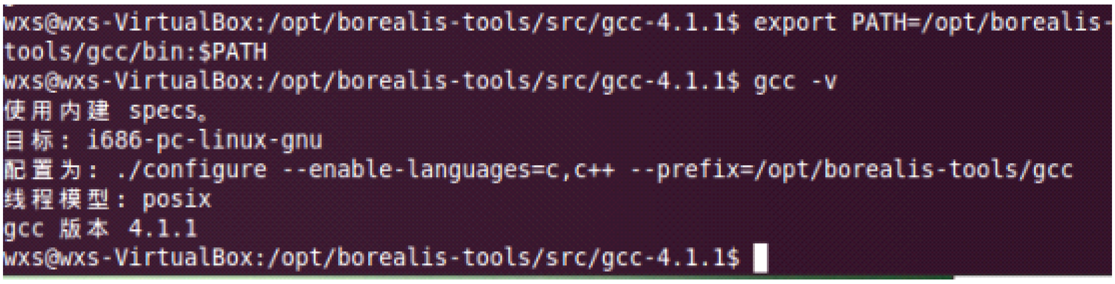
然后配置 configure 文件

```
./configure --enable-languages=c,c++ --prefix=/opt/borealis-tools/gcc
make
make install
```

gcc -v 查看当前版本

gcc 4.1.1 编译完后，应该设置环境变量，让后面的编译过程都使用 gcc 4.1.1

export PATH=/opt/borealis-tools/gcc/bin:$PATH



```
wxs@wxs-VirtualBox:/opt/borealis-tools/src/gcc-4.1.1$ export PATH=/opt/borealis-tools/gcc/bin:$PATH
wxs@wxs-VirtualBox:/opt/borealis-tools/src/gcc-4.1.1$ gcc -v
使用内建 specs。
目标：i686-pc-linux-gnu
配置为：./configure --enable-languages=c,c++ --prefix=/opt/borealis-tools/gcc
线程模型：posix
gcc 版本 4.1.1
wxs@wxs-VirtualBox:/opt/borealis-tools/src/gcc-4.1.1$
```

1.1.2、libtoolize（libtool）

版本：1.5.22

http://www.gnu.org/software/libtool/libtool.html

ftp://ftp.gnu.org/gnu/libtool/

直接在新立得包管理器中下载 libtool 的最新版本（我们这里安装的是最新的版本）

libtool	2.2.6b-2ubuntu3	2.2.6b-2ubuntu3	Generic library support

或者用一以下方法安装：

```
首先 cd /opt/borealis-tools/src
    tar -vxjf libtool-1.5.22.tar.gz
```

然后 cd libtool-1.5.22 在这个文件夹中有 configure 文件，我们要对这个文件进行配置。

```
cd libtool-1.5.22
./configure --prefix=/opt/borealis-tools/libtool
make
make install
```

1.1.3 autoconf

版本：2.60

http://www.gnu.org/software/autoconf/

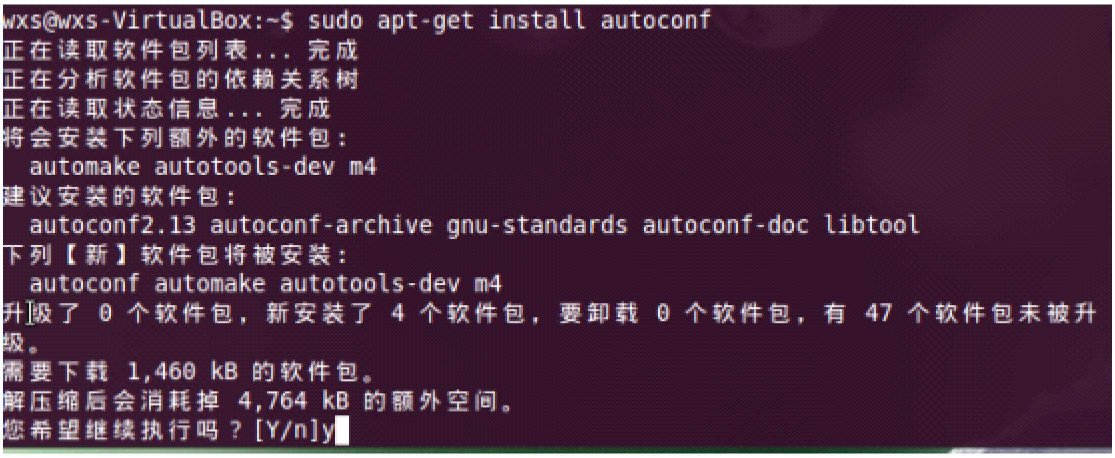
ftp://ftp.gnu.org/gnu/autoconf/

```
cd /opt/borealis-tools/src
```

```
tar -vxjf autoconf-2.60.tar.gz
```

配置 configure 文件

```
cd autoconf-2.60
./configure --prefix=/opt/borealis-tools/autoconf
make
make install
```



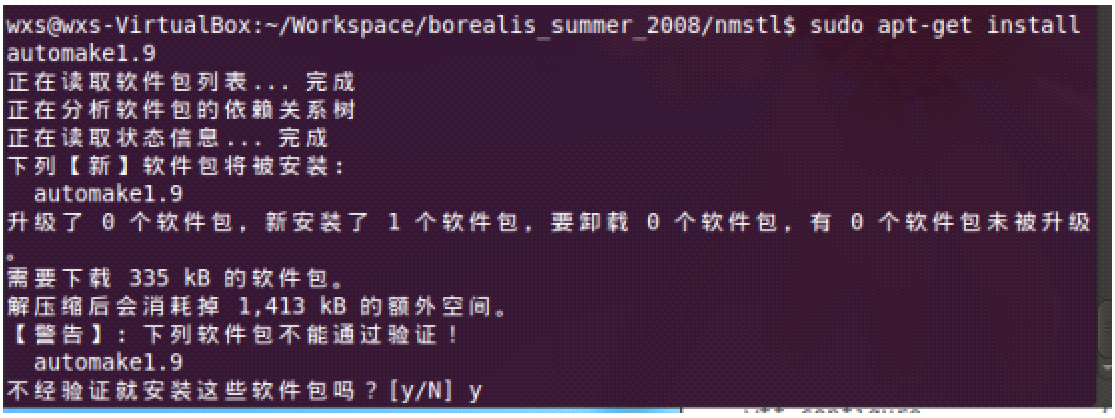
```
wxs@wxs-VirtualBox:~$ sudo apt-get install autoconf
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
将会安装下列额外的软件包：
  automake autotools-dev m4
建议安装的软件包：
  autoconf2.13 autoconf-archive gnu-standards autoconf-doc libtool
下列【新】软件包将被安装：
  autoconf automake autotools-dev m4
升级了 0 个软件包，新安装了 4 个软件包，要卸载 0 个软件包，有 47 个软件包未被升
级。
需要下载 1,460 kB 的软件包。
解压缩后会消耗掉 4,764 kB 的额外空间。
您希望继续执行吗？[Y/n]y
```

1.1.4 automake

版本：1.9.6 (要求 autoconf 2.6）

http://www.gnu.org/software/automake/

ftp://ftp.gnu.org/gnu/automake/



```
wxs@wxs-VirtualBox:~/Workspace/borealis_summer_2008/nmstl$ sudo apt-get install
automake1.9
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
下列【新】软件包将被安装：
  automake1.9
升级了 0 个软件包，新安装了 1 个软件包，要卸载 0 个软件包，有 0 个软件包未被升级
。
需要下载 335 kB 的软件包。
解压缩后会消耗掉 1,413 kB 的额外空间。
【警告】：下列软件包不能通过验证！
  automake1.9
不经验证就安装这些软件包吗？[y/N] y
```

1.1.5、opencv（ocv）

版本：2.4.2（下载最新版本）

http://downloads.sourceforge.net/opencvlibrary/
参考/docs/INSTALL 文件

```
cd /opt/borealis-tools/src
tar -zxvf OpenCV-2.0.0.tar.bz2
cd OpenCV-2.0.0
./configure --prefix=/opt/borealis-tools/opencv
make
make install
```

1.1.6 ccache

版本：2.4 (tested with gcc 4.1.1)

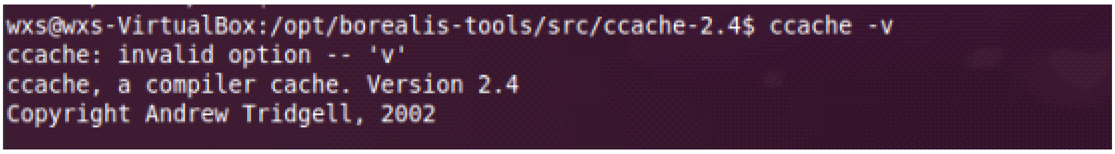
(optional, helps speed-up consecutive recompilations)

http://samba.org/ftp/ccache/

http://ccache.samba.org/

```
cd /opt/borealis-tools/src
tar -zxvf ccache-2.4.tar.gz
cd  ccache-2.4
./configure --prefix=/opt/borealis-tools/ccache
make
make install
```

安装完成后 ，用 ccache -v 查看是否安装成功，如下：



```
wxs@wxs-VirtualBox:/opt/borealis-tools/src/ccache-2.4$ ccache -v
ccache: invalid option -- 'v'
ccache, a compiler cache. Version 2.4
Copyright Andrew Tridgell, 2002
```

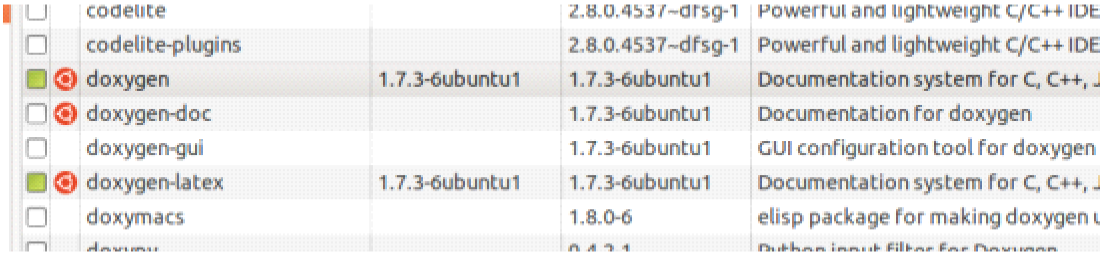
1.1.7 java jdk

1.1.8 Doxygen

版本：1.4.7(optional, serves to generate documentation from code)

http://www.stack.nl/~dimitri/doxygen/index.html

直接从　新立得软件包管理器　安装最新版本 1.7.3



☐		codelite		2.8.0.4537~dfsg-1	Powerful and lightweight C/C++ IDE
☐		codelite-plugins		2.8.0.4537~dfsg-1	Powerful and lightweight C/C++ IDE
🟩	◉	doxygen	1.7.3-6ubuntu1	1.7.3-6ubuntu1	Documentation system for C, C++, J
☐	◉	doxygen-doc		1.7.3-6ubuntu1	Documentation for doxygen
☐		doxygen-gui		1.7.3-6ubuntu1	GUI configuration tool for doxygen
🟩	◉	doxygen-latex	1.7.3-6ubuntu1	1.7.3-6ubuntu1	Documentation system for C, C++, J
☐		doxymacs		1.8.0-6	elisp package for making doxygen u
☐		doxypy		0.4.2.1	Python input filter for Doxygen

1.2、Packages Used By Borealis

1.2.1、Antlr

Antlr 需要安装 2.7.6 和 3.4 两个版本

安装 2.7.6 版本

版本：Linux: 2.7.6 (tested with gcc 4.1.1)

Used to parse: borealis/src/modules/queryProcessor/expr/expr.g

It uses java to run:

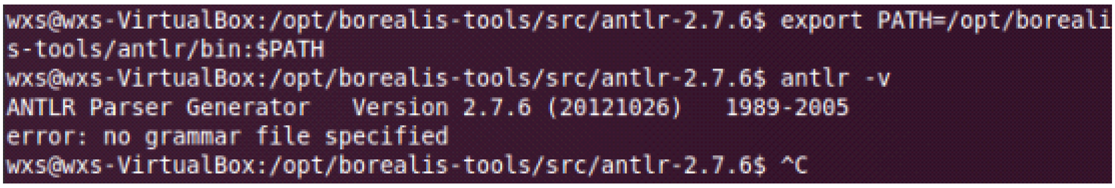
    $(JAVA) -cp $(ANTLR_JAR_FILE) antlr.Tool -o . expr.g

http://www.antlr.org

```
cd /opt/borealis-tools/src
tar -zxvf antlr-2.7.6.tar.gz
cd  antlr-2.7.6
./configure --prefix=/opt/borealis-tools/antlr2
make
make install
```

安装完成后，设置下路径 export PATH=/opt/borealis-tools/antlr2/bin:$PATH

然后测试下 antlr -v 如下图：



```
wxs@wxs-VirtualBox:/opt/borealis-tools/src/antlr-2.7.6$ export PATH=/opt/boreali
s-tools/antlr/bin:$PATH
wxs@wxs-VirtualBox:/opt/borealis-tools/src/antlr-2.7.6$ antlr -v
ANTLR Parser Generator   Version 2.7.6 (20121026)   1989-2005
error: no grammar file specified
wxs@wxs-VirtualBox:/opt/borealis-tools/src/antlr-2.7.6$ ^C
```

安装 3.4 版本

```
cd /opt/borealis-tools/src
tar -zxvf antlr-3.4.tar.gz
cd  antlr-3.4
./configure --prefix=/opt/borealis-tools/antlr3
make
make install
```

1.2.2 Xerces

版本：Linux: 2.7.0 (tested with gcc 4.1.1)

Used by Borealis to parse XML.

http://archive.apache.org/dist/xml/xerces-c/

http://xml.apache.org/xerces-c/

```
直接下载 xerces-c-src_2_7_0.tar.gz
然后直接解压在 /opt/borealis-tools/src
cd /opt/borealis-tools/src
export  XERCESCROOT=/opt/borealis-tools/src/xerces-c-src_2_7_0
cd $XERCESCROOT/src/xercesc
```

```
autoconf
./configure --prefix=/opt/borealis-tools/xercesc
make && make install
```

参考链接：

file:///opt/borealis-tools/src/xerces-c-src_2_7_0/doc/html/build-winunix.html#UNIX

1.2.3 BerkeleyDB（db）

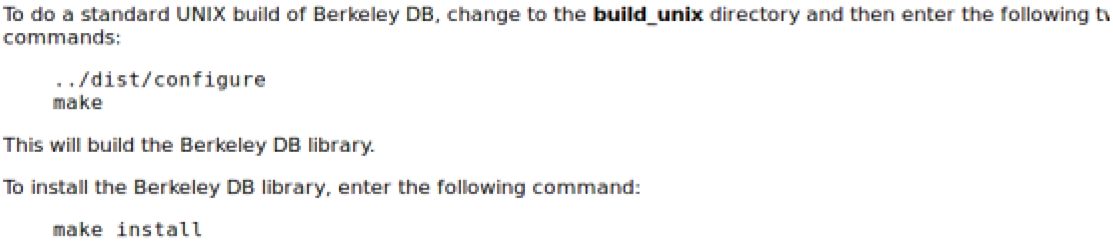
版本：(aka db4; optional)   Linux: db-4.4.20

http://www.sleepycat.com/

```
cd /opt/borealis-tools/src
tar -zxvf db-4.4.20.NC.tar.gz
cd  db-4.4.20.NC/build_unix
../dist/configure --prefix=/opt/borealis-tools/db
make
make install
```

参考链接：

file:///opt/borealis-tools/src/db-4.4.20.NC/docs/ref/build_unix/intro.html



To do a standard UNIX build of Berkeley DB, change to the **build_unix** directory and then enter the following tw commands:

```
../dist/configure
make
```

This will build the Berkeley DB library.

To install the Berkeley DB library, enter the following command:

```
make install
```

1.2.4 Glpk

版本：(optional)   Linux: 4.9（未找到该版本）

Used by the Load Manager component

ftp://ftp.gnu.org/gnu/glpk/

http://www.gnu.org/software/glpk/glpk.html

```
cd /opt/borealis-tools/src
tar -zxvf  glpk-4.47.tar.gz
cd  glpk-4.47
./configure --prefix=/opt/borealis-tools/glpk
make
make install
```

1.2.5 Gsl

版本：(optional)　Linux:　1.8
http://www.gnu.org/software/gsl/
ftp://ftp.gnu.org/gnu/gsl/

```
cd /opt/borealis-tools/src
tar -zxvf  gsl-1.8.tar.gz
cd  gsl-1.8
./configure --prefix=/opt/borealis-tools/gsl
make
make install
```

1.2.6 Ocv　（在 Tools Used to Build Borealis 已经安装）

1.2.7 TinyDb　（未安装）

版本：(optional; currently not in use)

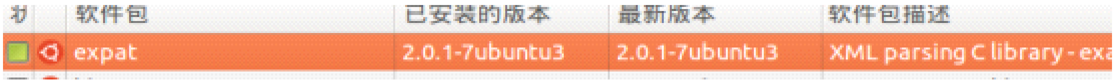
http://telegraph.cs.berkeley.edu/tinydb

http://www.tinydb.com

1.3、Software Integrated Into NMSTL

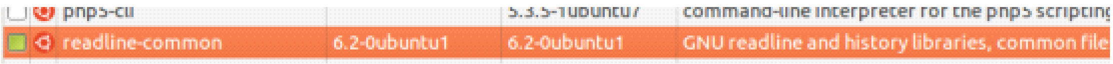
直接从　"新立得软件包管理器"安装。

1.3.1 安装 expat



状		软件包	已安装的版本	最新版本	软件包描述
☑		expat	2.0.1-7ubuntu3	2.0.1-7ubuntu3	XML parsing C library - exa

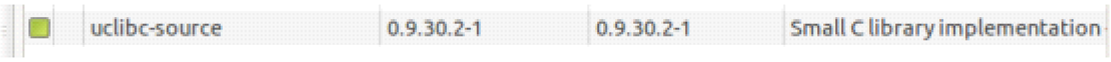
1.3.2 安装 readline



□		php5-cli		5.3.5-1ubuntu7	command-line interpreter for the php5 scripting
☑		readline-common	6.2-0ubuntu1	6.2-0ubuntu1	GNU readline and history libraries, common file

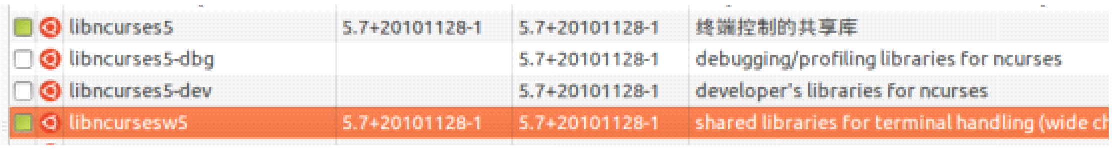
1.3.3 安装 atomicity.h

www.uclibc.org/lists/uclibc-cvs/2003-March/003194.html



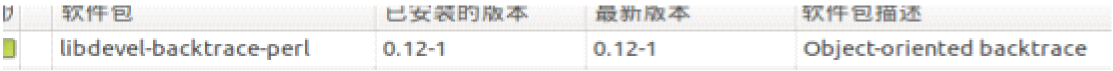
☑	uclibc-source	0.9.30.2-1	0.9.30.2-1	Small C library implementation

1.3.4 安装 libncurses



🟩	libncurses5	5.7+20101128-1	5.7+20101128-1	终端控制的共享库
□	libncurses5-dbg		5.7+20101128-1	debugging/profiling libraries for ncurses
□	libncurses5-dev		5.7+20101128-1	developer's libraries for ncurses
🟩	libncursesw5	5.7+20101128-1	5.7+20101128-1	shared libraries for terminal handling (wide ch

1.3.5 安装 backtrace from glibc



	软件包	已安装的版本	最新版本	软件包描述
🟩	libdevel-backtrace-perl	0.12-1	0.12-1	Object-oriented backtrace

安装 high resolution timer

安装 memalign

。。。

2、安装 nmstl

Borealis 使用修改过的 nmstl：下面我们安装nmstl，解压borealis_summer_2008源文件borealis_summer_2008.tar.gz，至如下文件夹：

```
cd $HOME/Workspace/borealis_summer_2008/nmstl
```

依次输入下面的命令进行安装；（在安装的过程中可能会出现一些错误，错误的解决办法见编译错误汇总6）

```
autoconf
wtf configure
```

（会提示未安装 wtf ，按提示输入 sudo apt-get install bsdgames）

```
./configure --prefix=/opt/borealis-tools/nmstl
make
make install
```

3、安装 Borealis

3.1 Borealis 简介

Borealis 是 Brandeis University, Brown University, MIT.三个大学联合开发的一个分布式的数据流处理引擎。Borealis 是基于 Aurora and Medusa 开发的。项目主页地址：

http://www.cs.brown.edu/research/borealis/public/

3.2 安装环境

操作系统：Ubuntu 11.04

Borealis 版本： borealis_summer_2008
下载地址：
http://www.cs.brown.edu/research/borealis/public/download/borealis_summer_2008.tar.gz

3.3 安装步骤

外部依赖的类库的编译在上面已经给出。

Borealis 源代码存放在 $HOME/Workspace/borealis_summer_2008
Borealis 依赖的外部类库存放在 /opt/borealis-tools
文中没有说明的相对路径都是相对于
$HOME/Workspace/borealis_summer_2008/borealis

编译完这些外部类库后，同样要设置环境变量，新建文件 /opt/borealis-tools/rc 内容为：

```
export PATH=/opt/borealis-tools/gcc/bin:$PATH
export PATH=/opt/borealis-tools/antlr2/bin:$PATH
export PATH=/opt/borealis-tools/ccache/bin:$PATH
export PATH=/opt/borealis-tools/db/bin:$PATH
export PATH=/opt/borealis-tools/glpk/bin:$PATH
export PATH=/opt/borealis-tools/antlr3/bin:$PATH
export PATH=/opt/borealis-tools/gsl/bin:$PATH
export PATH=/opt/borealis-tools/nmstl/bin:$PATH

export INSTALL_NMSTL=/opt/borealis-tools/nmstl
export INSTALL_ANTLR=/opt/borealis-tools/antlr2
export INSTALL_BDB=/opt/borealis-tools/db
export INSTALL_XERCESC=/opt/borealis-tools/xercesc
export INSTALL_GLPK=/opt/borealis-tools/glpk
export INSTALL_GSL=/opt/borealis-tools/gsl
#export INSTALL_OCV=/opt/borealis-tools/ocv

export LD_LIBRARY_PATH=/opt/borealis-tools/nmstl/lib:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH=/opt/borealis-tools/antlr2/lib:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH=/opt/borealis-tools/db/lib:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH=/opt/borealis-tools/xercesc/lib:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH=/opt/borealis-tools/glpk/lib:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH=/opt/borealis-tools/gsl/lib:$LD_LIBRARY_PATH
#export LD_LIBRARY_PATH=/opt/borealis-tools/ocv/lib:$LD_LIBRARY_PATH

export CVS_SANDBOX=$HOME/Workspace/borealis_summer_2008
```

每次编译或运行 Borealis 之前都应该运行下面的命令来初始化环境变量

```
source /opt/borealis-tools/rc
或者 source /opt/borealis-tools/login.bashrc
```

新建文件 $HOME/Workspace/borealis_summer_2008/borealis/init.sh 内容为

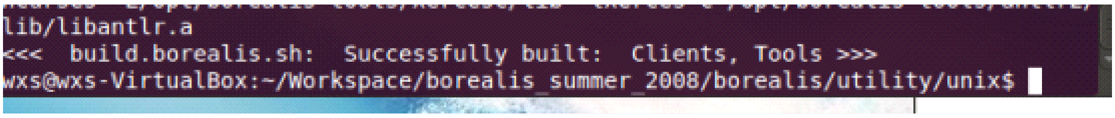
```
export PATH=$HOME/Workspace/borealis_summer_2008/borealis/tool/marshal:$PATH
export PATH=$HOME/Workspace/borealis_summer_2008/borealis/tool/head:$PATH
```

每次运行 Borealis 都应该运行下面这个命令初始化变量

```
source $HOME/Workspace/borealis_summer_2008/borealis/init.sh
```

编译 Borealis

```
cd $HOME/Workspace/borealis_summer_2008/borealis/utility/unix/
./build.borealis.sh
./build.borealis.sh -client -tool.marshal -tool.head
```



```
lib/libantlr.a
<<<  build.borealis.sh:  Successfully built:  Clients, Tools >>>
wxs@wxs-VirtualBox:~/Workspace/borealis_summer_2008/borealis/utility/unix$
```

统一一下：

我把 login.bashrc 和 init.sh 的内容都放在 init.sh 文件里，所以以后再运行 borealis 之前须键入下面的初始化语句：

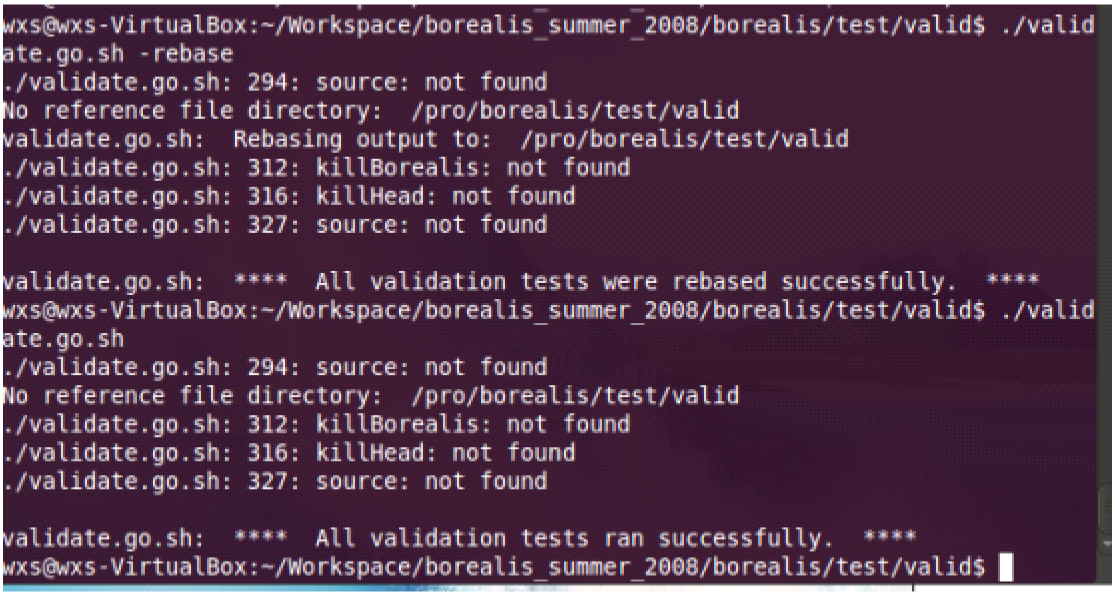
```
source $HOME/Workspace/borealis_summer_2008/borealis/init.sh
```

3.4 运行测试

运行 Borealis 中的 test， 验证 Borealis 是否正确编译。

```
cd $HOME/Workspace/borealis_summer_2008/borealis/utility/unix/
./build.borealis.sh -test.valid
cd $HOME/Workspace/borealis_summer_2008/borealis/test/valid
./validate.go.sh -rebase
./validate.go.sh
```

以上的命令都应该通过，如果没有全部通过，那么就是编译不正确



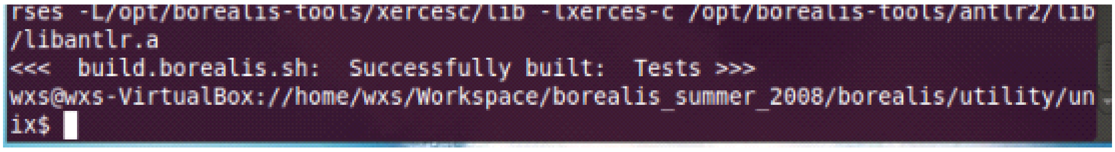
```
wxs@wxs-VirtualBox:~/Workspace/borealis_summer_2008/borealis/test/valid$ ./valid
ate.go.sh -rebase
./validate.go.sh: 294: source: not found
No reference file directory:  /pro/borealis/test/valid
validate.go.sh:  Rebasing output to:  /pro/borealis/test/valid
./validate.go.sh: 312: killBorealis: not found
./validate.go.sh: 316: killHead: not found
./validate.go.sh: 327: source: not found

validate.go.sh:  ****  All validation tests were rebased successfully.  ****
wxs@wxs-VirtualBox:~/Workspace/borealis_summer_2008/borealis/test/valid$ ./valid
ate.go.sh
./validate.go.sh: 294: source: not found
No reference file directory:  /pro/borealis/test/valid
./validate.go.sh: 312: killBorealis: not found
./validate.go.sh: 316: killHead: not found
./validate.go.sh: 327: source: not found

validate.go.sh:  ****  All validation tests ran successfully.  ****
wxs@wxs-VirtualBox:~/Workspace/borealis_summer_2008/borealis/test/valid$ 
```

3.5 运行简单的实例程序

我们可以用 C++或者 JAVA 编写 Borealis 应用程序。

所有的实例程序位于 borealis/test/下，我们可以按下面的命令来构建所有的实例程序：

```
cd $HOME/Workspace/borealis_summer_2008/borealis/utility/unix
./build.borealis.sh  -test.simple
# Build tests in:  borealis/test/simple/
```



```
rses -L/opt/borealis-tools/xercesc/lib -lxerces-c /opt/borealis-tools/antlr2/lib
/libantlr.a
<<<  build.borealis.sh:  Successfully built:  Tests >>>
wxs@wxs-VirtualBox://home/wxs/Workspace/borealis_summer_2008/borealis/utility/un
ix$ 
```

或者我们可以使用下面的命令:

要想了解怎样 run 简单的 C++ Borealis 应用程序，需要查看 borealis/test/simple/ 目录下的实例。README 文件解释的很详细。（一定要看的）

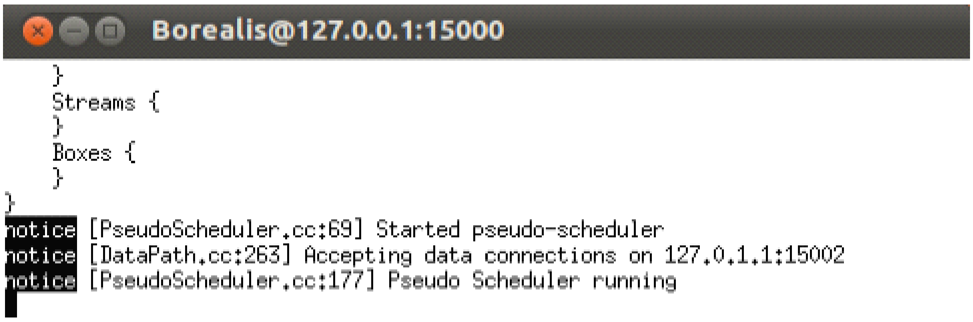
```
cd $HOME/Workspace/borealis_summer_2008/borealis/test
./setup
wtf configure --with-antlr=/opt/borealis-tools/antlr2
--with-xercesc=/opt/borealis-tools/xercesc --with-bdb=/opt/borealis-tools/db
--with-gsl=/opt/borealis-tools/gsl --with-nmstl=/opt/borealis-tools/nmstl
--with-borealis=$HOME/Workspace/borealis_summer_2008/borealis/src/
make
```

(在 make 时出现错误，具体见编译错误汇总 10)

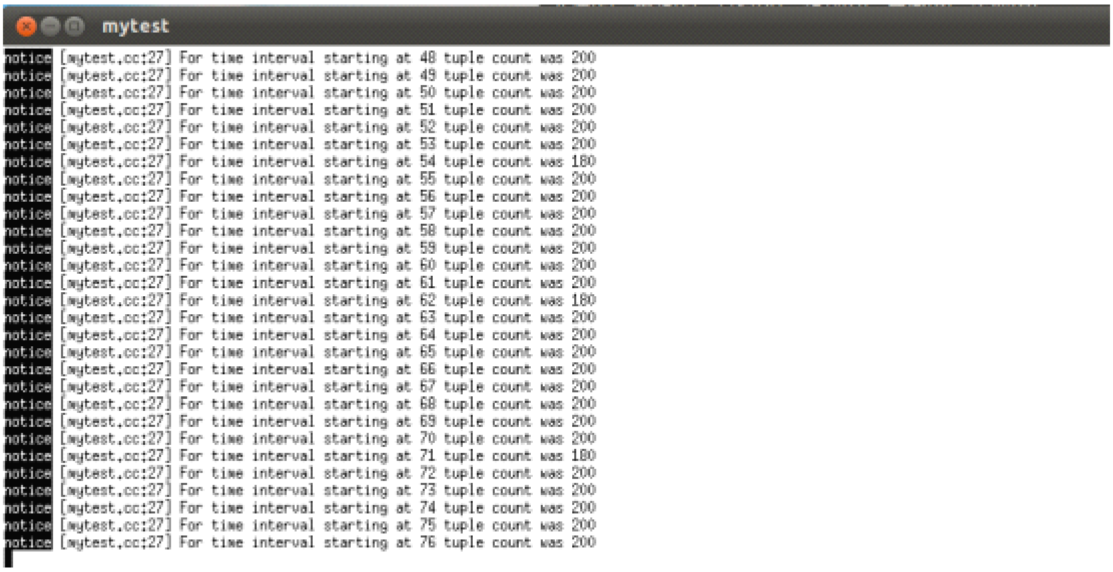
3.6 最后测试：

```
cd $HOME/Workspace/borealis_summer_2008/borealis/test/simple
```

```
./runtest mytest
```



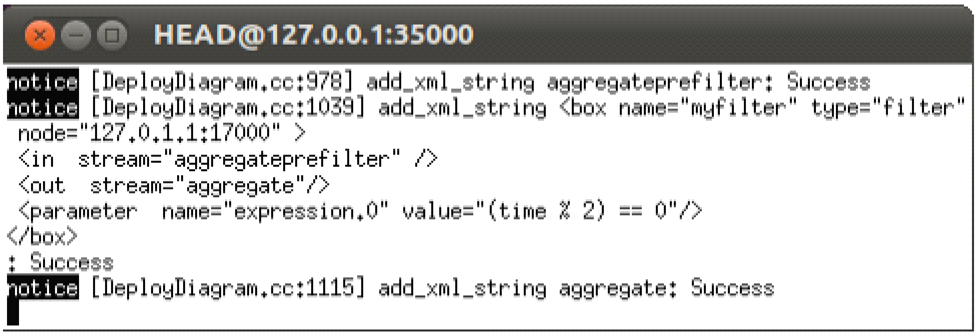
```
Borealis@127.0.0.1:15000
    }
    Streams {
    }
    Boxes {
    }
}
notice [PseudoScheduler.cc:69] Started pseudo-scheduler
notice [DataPath.cc:263] Accepting data connections on 127.0.1.1:15002
notice [PseudoScheduler.cc:177] Pseudo Scheduler running
```



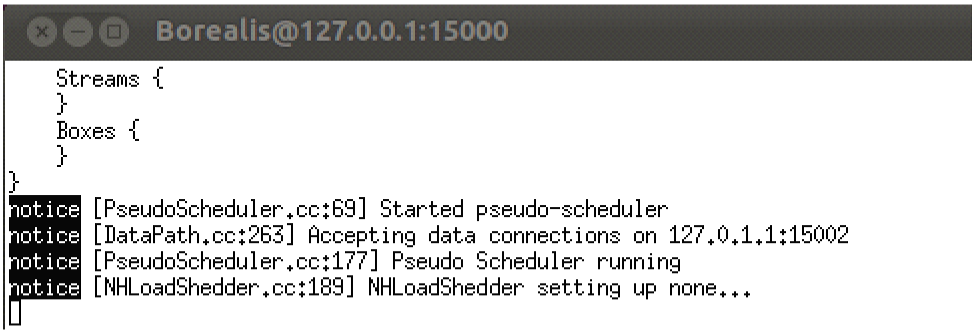
```
mytest
notice [mytest.cc:27] For time interval starting at 48 tuple count was 200
notice [mytest.cc:27] For time interval starting at 49 tuple count was 200
notice [mytest.cc:27] For time interval starting at 50 tuple count was 200
notice [mytest.cc:27] For time interval starting at 51 tuple count was 200
notice [mytest.cc:27] For time interval starting at 52 tuple count was 200
notice [mytest.cc:27] For time interval starting at 53 tuple count was 200
notice [mytest.cc:27] For time interval starting at 54 tuple count was 180
notice [mytest.cc:27] For time interval starting at 55 tuple count was 200
notice [mytest.cc:27] For time interval starting at 56 tuple count was 200
notice [mytest.cc:27] For time interval starting at 57 tuple count was 200
notice [mytest.cc:27] For time interval starting at 58 tuple count was 200
notice [mytest.cc:27] For time interval starting at 59 tuple count was 200
notice [mytest.cc:27] For time interval starting at 60 tuple count was 200
notice [mytest.cc:27] For time interval starting at 61 tuple count was 200
notice [mytest.cc:27] For time interval starting at 62 tuple count was 180
notice [mytest.cc:27] For time interval starting at 63 tuple count was 200
notice [mytest.cc:27] For time interval starting at 64 tuple count was 200
notice [mytest.cc:27] For time interval starting at 65 tuple count was 200
notice [mytest.cc:27] For time interval starting at 66 tuple count was 200
notice [mytest.cc:27] For time interval starting at 67 tuple count was 200
notice [mytest.cc:27] For time interval starting at 68 tuple count was 200
notice [mytest.cc:27] For time interval starting at 69 tuple count was 200
notice [mytest.cc:27] For time interval starting at 70 tuple count was 200
notice [mytest.cc:27] For time interval starting at 71 tuple count was 180
notice [mytest.cc:27] For time interval starting at 72 tuple count was 200
notice [mytest.cc:27] For time interval starting at 73 tuple count was 200
notice [mytest.cc:27] For time interval starting at 74 tuple count was 200
notice [mytest.cc:27] For time interval starting at 75 tuple count was 200
notice [mytest.cc:27] For time interval starting at 76 tuple count was 200
```

使用./runtest stop 停止运行程序

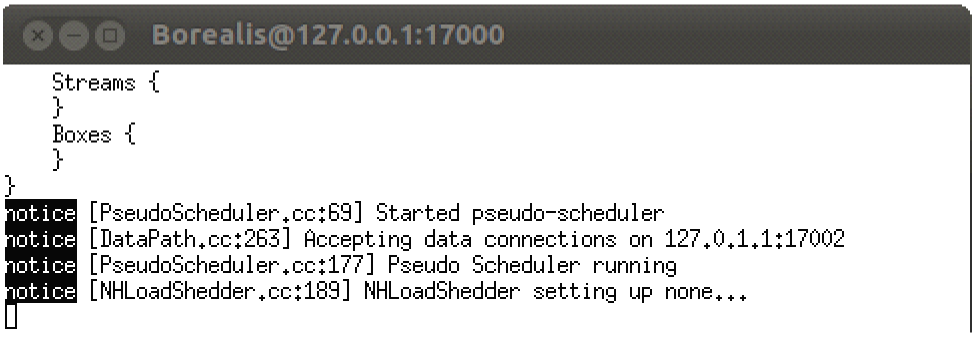
下面测试 mytestdist，输入命令./runtest mytestdist，运行如下：



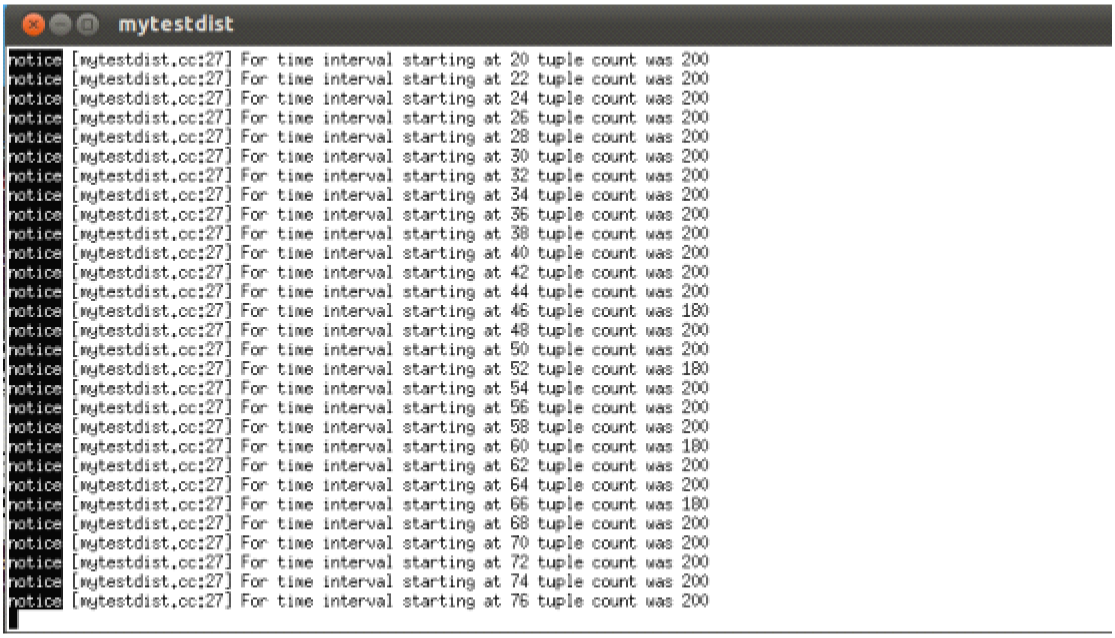
```
HEAD@127.0.0.1:35000
notice [DeployDiagram.cc:978] add_xml_string aggregateprefilter: Success
notice [DeployDiagram.cc:1039] add_xml_string <box name="myfilter" type="filter"
 node="127.0.1.1:17000" >
 <in  stream="aggregateprefilter" />
 <out  stream="aggregate"/>
 <parameter  name="expression.0" value="(time % 2) == 0"/>
</box>
: Success
notice [DeployDiagram.cc:1115] add_xml_string aggregate: Success
```



```
Borealis@127.0.0.1:15000
    Streams {
    }
    Boxes {
    }
}
notice [PseudoScheduler.cc:69] Started pseudo-scheduler
notice [DataPath.cc:263] Accepting data connections on 127.0.1.1:15002
notice [PseudoScheduler.cc:177] Pseudo Scheduler running
notice [NHLoadShedder.cc:189] NHLoadShedder setting up none...
```



```
    Streams {
    }
    Boxes {
    }
}
notice [PseudoScheduler.cc:69] Started pseudo-scheduler
notice [DataPath.cc:263] Accepting data connections on 127.0.1.1:17002
notice [PseudoScheduler.cc:177] Pseudo Scheduler running
notice [NHLoadShedder.cc:189] NHLoadShedder setting up none...
```



```
notice [mytestdist.cc:27] For time interval starting at 20 tuple count was 200
notice [mytestdist.cc:27] For time interval starting at 22 tuple count was 200
notice [mytestdist.cc:27] For time interval starting at 24 tuple count was 200
notice [mytestdist.cc:27] For time interval starting at 26 tuple count was 200
notice [mytestdist.cc:27] For time interval starting at 28 tuple count was 200
notice [mytestdist.cc:27] For time interval starting at 30 tuple count was 200
notice [mytestdist.cc:27] For time interval starting at 32 tuple count was 200
notice [mytestdist.cc:27] For time interval starting at 34 tuple count was 200
notice [mytestdist.cc:27] For time interval starting at 36 tuple count was 200
notice [mytestdist.cc:27] For time interval starting at 38 tuple count was 200
notice [mytestdist.cc:27] For time interval starting at 40 tuple count was 200
notice [mytestdist.cc:27] For time interval starting at 42 tuple count was 200
notice [mytestdist.cc:27] For time interval starting at 44 tuple count was 200
notice [mytestdist.cc:27] For time interval starting at 46 tuple count was 180
notice [mytestdist.cc:27] For time interval starting at 48 tuple count was 200
notice [mytestdist.cc:27] For time interval starting at 50 tuple count was 200
notice [mytestdist.cc:27] For time interval starting at 52 tuple count was 180
notice [mytestdist.cc:27] For time interval starting at 54 tuple count was 200
notice [mytestdist.cc:27] For time interval starting at 56 tuple count was 200
notice [mytestdist.cc:27] For time interval starting at 58 tuple count was 200
notice [mytestdist.cc:27] For time interval starting at 60 tuple count was 180
notice [mytestdist.cc:27] For time interval starting at 62 tuple count was 200
notice [mytestdist.cc:27] For time interval starting at 64 tuple count was 200
notice [mytestdist.cc:27] For time interval starting at 66 tuple count was 180
notice [mytestdist.cc:27] For time interval starting at 68 tuple count was 200
notice [mytestdist.cc:27] For time interval starting at 70 tuple count was 200
notice [mytestdist.cc:27] For time interval starting at 72 tuple count was 200
notice [mytestdist.cc:27] For time interval starting at 74 tuple count was 200
notice [mytestdist.cc:27] For time interval starting at 76 tuple count was 200
```

使用./runtest stop 停止运行程序