2016/11/8 ES2016_14353404/readme.md at master · sysuzyc/ES2016_14353404

This repository Search Pull requests Issues Gist

sysuzyc / ES2016_14353404 Watch 0 Star 1 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Pulse Graphs Settings

Branch: master ES2016_14353404 / readme.md Find file Copy path

sysuzyc Update readme.md d85cea3 8 minutes ago

2 contributors

Executable File 120 lines (71 sloc) 4.83 KB Raw Blame History

DOL开发环境配置

姓名：张亚琛　 学号：14353404

　本次实验进行的是在Linux环境下的DOL的配置，由于我使用的windows系统，所以，利用的是虚拟机VMware平台进行的实验，安装的是Ubuntu16.04，在这个平台上进行实验的操作。

为了完成这次的实验，我们需要分几步完成：

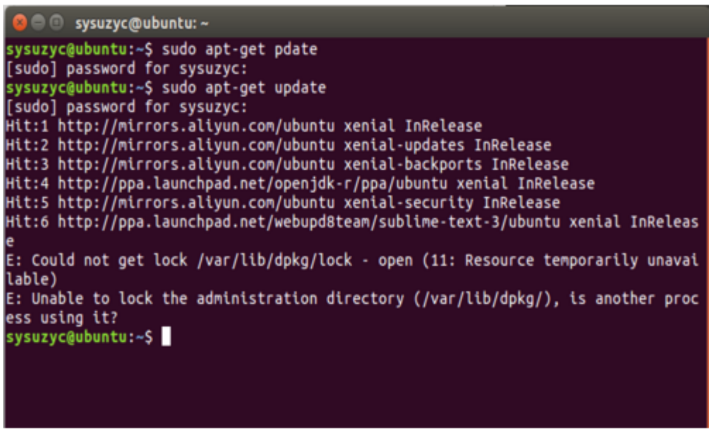
```
1、安装必要的环境
2、进行dol文件的解压和安装
3、编译systemc
4、编译dol
```

所以，我们按照步骤进行相关的实验：

1、安装必要的环境

1）我们需要做的是进行一些适当库的更新。首先需要的是先进行update，看下那些源可以使用：

```
sudo apt-get update
```

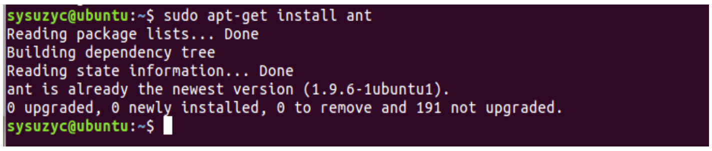


```
sysuzyc@ubuntu:~$ sudo apt-get pdate
[sudo] password for sysuzyc:
sysuzyc@ubuntu:~$ sudo apt-get update
[sudo] password for sysuzyc:
Hit:1 http://mirrors.aliyun.com/ubuntu xenial InRelease
Hit:2 http://mirrors.aliyun.com/ubuntu xenial-updates InRelease
Hit:3 http://mirrors.aliyun.com/ubuntu xenial-backports InRelease
Hit:4 http://ppa.launchpad.net/openjdk-r/ppa/ubuntu xenial InRelease
Hit:5 http://mirrors.aliyun.com/ubuntu xenial-security InRelease
Hit:6 http://ppa.launchpad.net/webupd8team/sublime-text-3/ubuntu xenial InReleas
e
E: Could not get lock /var/lib/dpkg/lock - open (11: Resource temporarily unavai
lable)
E: Unable to lock the administration directory (/var/lib/dpkg/), is another proc
ess using it?
sysuzyc@ubuntu:~$ 
```

在进行了源的更新之后，我们可以继续进行下面的配置。

2）我们看到这次的实验是利用java进行编译的，那么我们就需要配置一些必要的环境。Ant是我们需要配置的一个环境，它是一个编译java的平台。

```
sudo apt-get install ant
```



```
sysuzyc@ubuntu:~$ sudo apt-get install ant
Reading package lists... Done
Building dependency tree
Reading state information... Done
ant is already the newest version (1.9.6-1ubuntu1).
0 upgraded, 0 newly installed, 0 to remove and 191 not upgraded.
sysuzyc@ubuntu:~$
```

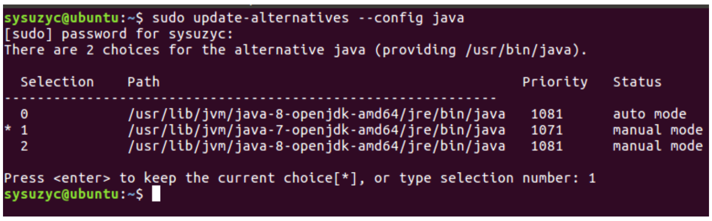
看到上面的一些信息，则说明我们的ant是配置成功的。

3）在安装好平台之后，我们的编译语言java的最重要的部分jdk需要安装了，我们用的是jdk7，所以，我们就配置jdk7。由于我用的是Ubuntu16.04的版本，所以，我的jdk是不可以直接进行安装的，需要加入一些适当的库，才可以进行jdk的安装。

```
sudo add-apt-repository ppa:openjdk-r/ppa
sudo apt-get update
sudo apt-get install openjdk-7-jdk  // OpenJdk 7安装：
```

这样的话，就可以进行jdk的安装了，但是有的时候，还是不可以进行运行，那么我们需要看的是自己的jdk到底是什么版本的，选择正确的版本即可。

```
sudo update-alternatives --config java
```

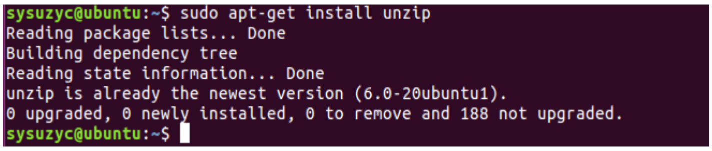


```
sysuzyc@ubuntu:~$ sudo update-alternatives --config java
[sudo] password for sysuzyc:
There are 2 choices for the alternative java (providing /usr/bin/java).

  Selection    Path                                            Priority   Status
------------------------------------------------------------
  0            /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java   1081      auto mode
* 1            /usr/lib/jvm/java-7-openjdk-amd64/jre/bin/java   1071      manual mode
  2            /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java   1081      manual mode

Press <enter> to keep the current choice[*], or type selection number: 1
sysuzyc@ubuntu:~$
```

这里看到的是，我们选择了jdk7就可以正确运行了，这里是配置好了对应的环境。

4）为了能够解压我们拷贝进来的压缩包，我们也要安装unzip工具：

```
sudo apt-get install unzip
```



```
sysuzyc@ubuntu:~$ sudo apt-get install unzip
Reading package lists... Done
Building dependency tree
Reading state information... Done
unzip is already the newest version (6.0-20ubuntu1).
0 upgraded, 0 newly installed, 0 to remove and 188 not upgraded.
sysuzyc@ubuntu:~$
```

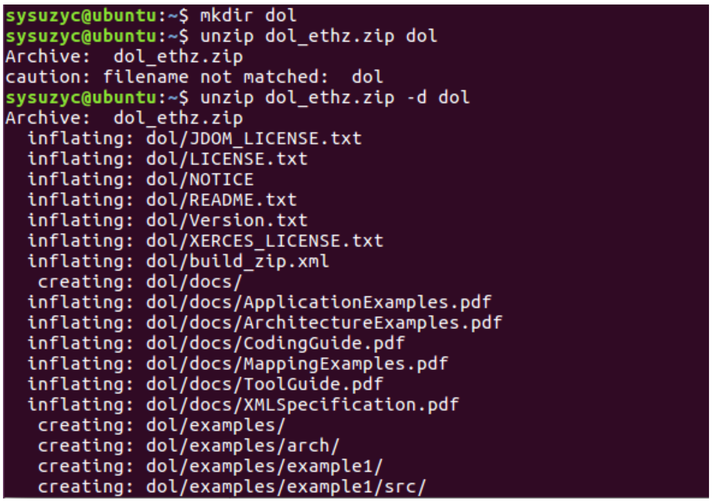
这里，我们就可以进行下面的操作了。

2、进行dol文件的解压和安装

我们将文件中的dol的压缩包拷贝到Ubuntu中，进行解压。

```
mkdir dol
unzip dol_ethz.zip -d dol
```

敲入上面的命令之后，可以看到下面的图片：



```
sysuzyc@ubuntu:~$ mkdir dol
sysuzyc@ubuntu:~$ unzip dol_ethz.zip dol
Archive:  dol_ethz.zip
caution: filename not matched:  dol
sysuzyc@ubuntu:~$ unzip dol_ethz.zip -d dol
Archive:  dol_ethz.zip
  inflating: dol/JDOM_LICENSE.txt
  inflating: dol/LICENSE.txt
  inflating: dol/NOTICE
  inflating: dol/README.txt
  inflating: dol/Version.txt
  inflating: dol/XERCES_LICENSE.txt
  inflating: dol/build_zip.xml
   creating: dol/docs/
  inflating: dol/docs/ApplicationExamples.pdf
  inflating: dol/docs/ArchitectureExamples.pdf
  inflating: dol/docs/CodingGuide.pdf
  inflating: dol/docs/MappingExamples.pdf
  inflating: dol/docs/ToolGuide.pdf
  inflating: dol/docs/XMLSpecification.pdf
   creating: dol/examples/
   creating: dol/examples/arch/
   creating: dol/examples/example1/
   creating: dol/examples/example1/src/
```

等虚拟机跑完了之后，我们就会发现自己的dol文件解压好了。

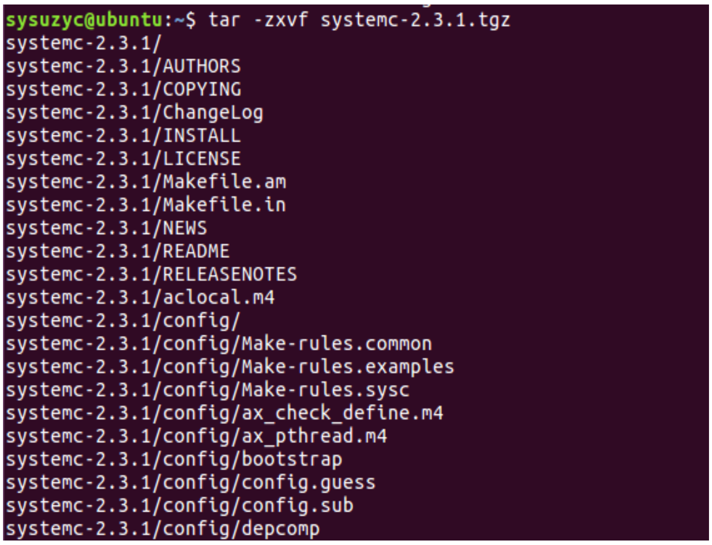
3、编译systemc

我们首先是进行systemc文件的解压，然后再进行编译:

```
tar -zxvf systemc-2.3.1.tgz
cd systemc-2.3.1
mkdir objdir
cd objdir
../configure CXX=g++ --disable-async-updates
sudo make install
```

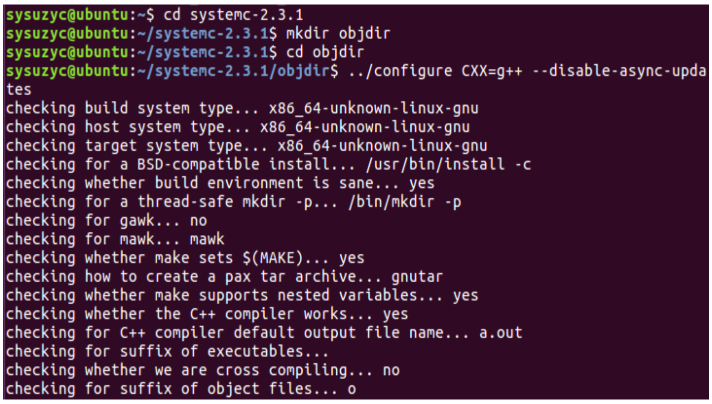
按照上面的步骤，我们就可以对systemc进行编译。

首先是进行解压：



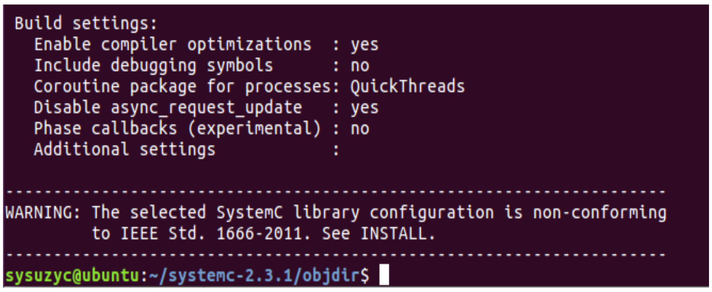
```
sysuzyc@ubuntu:~$ tar -zxvf systemc-2.3.1.tgz
systemc-2.3.1/
systemc-2.3.1/AUTHORS
systemc-2.3.1/COPYING
systemc-2.3.1/ChangeLog
systemc-2.3.1/INSTALL
systemc-2.3.1/LICENSE
systemc-2.3.1/Makefile.am
systemc-2.3.1/Makefile.in
systemc-2.3.1/NEWS
systemc-2.3.1/README
systemc-2.3.1/RELEASENOTES
systemc-2.3.1/aclocal.m4
systemc-2.3.1/config/
systemc-2.3.1/config/Make-rules.common
systemc-2.3.1/config/Make-rules.examples
systemc-2.3.1/config/Make-rules.sysc
systemc-2.3.1/config/ax_check_define.m4
systemc-2.3.1/config/ax_pthread.m4
systemc-2.3.1/config/bootstrap
systemc-2.3.1/config/config.guess
systemc-2.3.1/config/config.sub
systemc-2.3.1/config/depcomp
```

然后，进入我们解压后的文件夹，新建一个文件夹，然后进入其中，进行下面的编译操作：



```
sysuzyc@ubuntu:~$ cd systemc-2.3.1
sysuzyc@ubuntu:~/systemc-2.3.1$ mkdir objdir
sysuzyc@ubuntu:~/systemc-2.3.1$ cd objdir
sysuzyc@ubuntu:~/systemc-2.3.1/objdir$ ../configure CXX=g++ --disable-async-upda
tes
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking target system type... x86_64-unknown-linux-gnu
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... no
checking for mawk... mawk
checking whether make sets $(MAKE)... yes
checking how to create a pax tar archive... gnutar
checking whether make supports nested variables... yes
checking whether the C++ compiler works... yes
checking for C++ compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
```

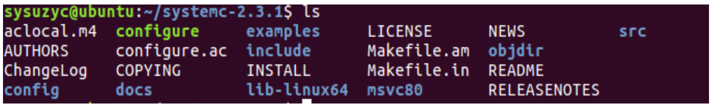
等到运行完了之后，我们就可以对比下课件，看下自己跑出来的结果是不是正确的。



```
Build settings:
  Enable compiler optimizations  : yes
  Include debugging symbols      : no
  Coroutine package for processes: QuickThreads
  Disable async_request_update   : yes
  Phase callbacks (experimental) : no
  Additional settings            :

---------------------------------------------------------------------
WARNING: The selected SystemC library configuration is non-conforming
         to IEEE Std. 1666-2011. See INSTALL.
---------------------------------------------------------------------
sysuzyc@ubuntu:~/systemc-2.3.1/objdir$
```

我们看到最终的结果是正确的，所以，我们就可以看到这次的systemc编译是正确的。

在看到上面的结果之后，我们需要再次检查下看下是不是有什么其他的被遗漏的东西：



```
sysuzyc@ubuntu:~/systemc-2.3.1$ ls
aclocal.m4  configure     examples     LICENSE      NEWS          src
AUTHORS     configure.ac  include      Makefile.am  objdir
ChangeLog   COPYING       INSTALL      Makefile.in  README
config      docs          lib-linux64  msvc80       RELEASENOTES
```

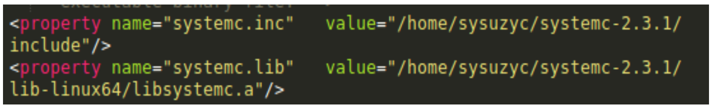
可以看到一共21个文件，没有遗漏的，所以，我们可以说是编译成功了。

4、编译dol

在前面，我们进行了dol的解压，但是并没有进行编译，所以，这里进行编译：

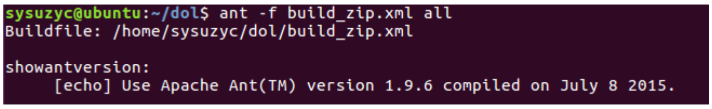
```
cd ../dol
ant -f build_zip.xml all
cd build/bin/main
ant -f runexample.xml -Dnumber=1
```

首先进入第一步中解压好的文件夹dol，然后，我们去修改对应的xml文件，修改一些路径，改为我们需要的xml文件：

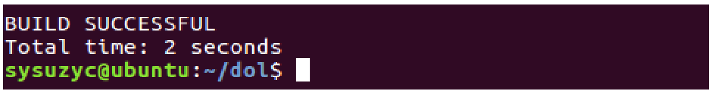


```
<property name="systemc.inc"   value="/home/sysuzyc/systemc-2.3.1/
include"/>
<property name="systemc.lib"   value="/home/sysuzyc/systemc-2.3.1/
lib-linux64/libsystemc.a"/>
```

在修改好了之后，我们就可以编译它了。

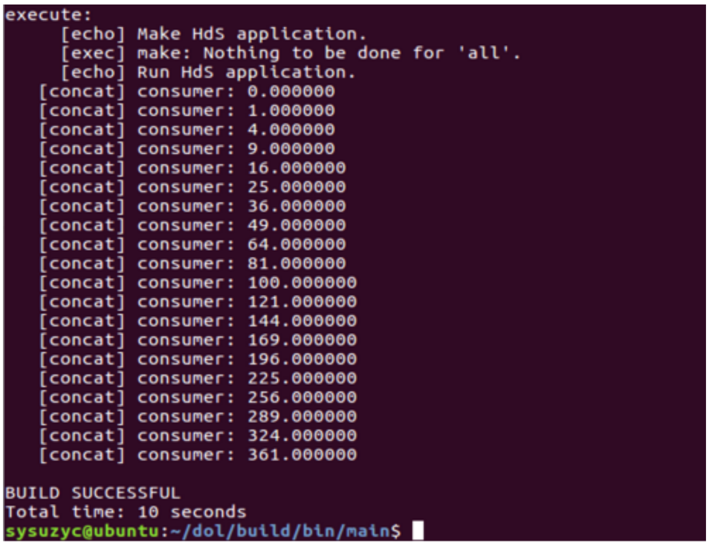


```
sysuzyc@ubuntu:~/dol$ ant -f build_zip.xml all
Buildfile: /home/sysuzyc/dol/build_zip.xml

showantversion:
     [echo] Use Apache Ant(TM) version 1.9.6 compiled on July 8 2015.
```



```
BUILD SUCCESSFUL
Total time: 2 seconds
sysuzyc@ubuntu:~/dol$
```

所以，我们看到，这个时候，buidl_zip.xml是编译成功了，然后，我们可以尝试下运行这次的实验，结果如下所示：



```
execute:
     [echo] Make HdS application.
     [exec] make: Nothing to be done for 'all'.
     [echo] Run HdS application.
   [concat] consumer: 0.000000
   [concat] consumer: 1.000000
   [concat] consumer: 4.000000
   [concat] consumer: 9.000000
   [concat] consumer: 16.000000
   [concat] consumer: 25.000000
   [concat] consumer: 36.000000
   [concat] consumer: 49.000000
   [concat] consumer: 64.000000
   [concat] consumer: 81.000000
   [concat] consumer: 100.000000
   [concat] consumer: 121.000000
   [concat] consumer: 144.000000
   [concat] consumer: 169.000000
   [concat] consumer: 196.000000
   [concat] consumer: 225.000000
   [concat] consumer: 256.000000
   [concat] consumer: 289.000000
   [concat] consumer: 324.000000
   [concat] consumer: 361.000000

BUILD SUCCESSFUL
Total time: 10 seconds
sysuzyc@ubuntu:~/dol/build/bin/main$
```

看到上面的结果的话，说明我们这次的实验配置成功了。

😊