# DOL开发环境配置

**Distributed Operation Layer :**

The distributed operation layer (DOL) is a software development framework to program parallel applications. The DOL allows to specify applications based on the Kahn process network model of computation and features a simulation engine based on SystemC. Moreover, the DOL provides an XML-based specification format to describe the implementation of a parallel application on a multi-processor systems, including binding and mapping.

www.tik.ee.ethz.ch/~shapes/dol.html

# Make工具简介

- 在Linux和Ubuntu环境中，make工具主要被用来进行工程编译和程序链接
- Makefile文件：告诉make以何种方式编译源代码和链接程序
- make通过比较对应文件（规则的目标和依赖）的最后修改时间，来决定哪些文件需要更新、那些文件不需要更新。

http://blog.chinaunix.net/uid-9314244-id-2004686.html

# Ant工具简介

- Ant是一种基于Java的build工具。
- Ant用Java的类来扩展。
- Ant本身就是这样一个流程脚本引擎，用于自动化调用程序完成项目的编译，打包，测试等。

http://blog.163.com/qiangyongbin2000@126/blog/static/77517819201151653423687/

# Ant的优点

- 跨平台性。Ant是纯java语言编写的，所示具有很好的跨平台性。
- 操作简单。Ant是由一个内置任务和可选任务组成的。Ant运行时需要一个XML文件(构建文件)。
- 容易维护和书写，结构清晰。
- Ant可以集成到开发环境中。

# Java与javac简介

- 用途：编译或执行java代码
- javac命令用来编译java文件
- java命令可以执行生成的class文件

本次实验环境在linux下进行，建议使用虚拟机安装Ubuntu（

VMWARE教程：
http://jingyan.baidu.com/article/0320e2c1ef9f6c1b87507bf6.html
VIRTUALBOX教程：
http://jingyan.baidu.com/article/cdddd41c5eea3153ca00e160.html
Ubuntu下载：
http://www.ubuntu.com/download/desktop

）

安装一些必要的环境(ubuntu为例)：

```
$       sudo apt-get update
$       sudo apt-get install ant
$       sudo apt-get install openjdk-7-jdk
$       sudo apt-get install unzip
```

1.下载文件(使用Vmware虚拟机，也可以从主机拷贝到虚拟机中去 http://jingyan.baidu.com/article/c33e3f48a5c153ea15cbb5b2.html ):

**sudo wget http://www.accellera.org/images/downloads/standards/systemc/systemc-2.3.1.tgz**

**sudo wget http://www.tik.ee.ethz.ch/~shapes/downloads/dol_ethz.zip**

# 2.解压文件

新建dol的文件夹

**$ sudo mkdir dol**

将dolethz.zip解压到 dol文件夹中

**$ sudo unzip dol_ethz.zip -d dol**

解压systemc

**$ sudo tar -zxvf systemc-2.3.1.tgz**

# 2.编译systemc

解压后进入systemc-2.3.1的目录下

**$ cd systemc-2.3.1**

新建一个临时文件夹objdir

**$ sudo mkdir objdir**

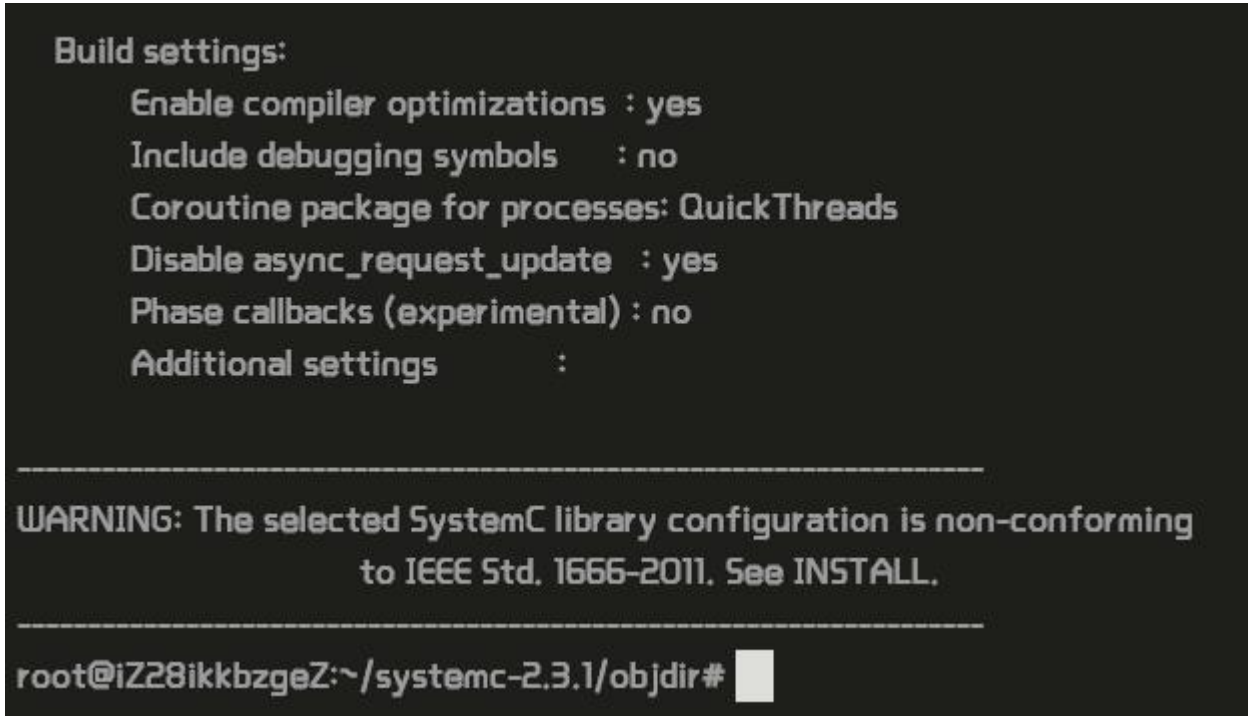
进入该文件夹objdir

**$ cd objdir**

运行configure(能根据系统的环境设置一下参数，用于编译)

**$ sudo ../configure CXX=g++ --disable-async-updates**

**右图为运行configure之后的截图**



# 2.编译systemc（续上页

编译

**$ sudo make install**

编译完后文件目录如下($ cd .. $ ls

```
root@iZ28ikkbzgeZ:~/systemc-2.3.1# ls
aclocal.m4  config      COPYING   include      LICENSE      msvc80  README
AUTHORS     configure   docs      INSTALL      Makefile.am  NEWS    RELEASENOTES
ChangeLog   configure.ac examples  lib-linux64  Makefile.in  objdir  src
```

记录当前的工作路径(会输出当前所在路径，记下来，待会有用)

**$ sudo pwd**

```
root@iZ28ikkbzgeZ:~/systemc-2.3.1# pwd
/root/systemc-2.3.1
```

这里表示我当前的工作路径为 /root/systemc-2.3.1

# 3.编译dol

进入刚刚dol的文件夹

**$ cd ../dol**

修改build_zip.xml文件

**找到下面这段话，就是说上面编译的systemc位置在哪里，**

<property name="systemc.inc" value="YYY/include"/>

<property name="systemc.lib" value="YYY/**lib-linux**/libsystemc.a"/>

把YYY改成上页pwd的结果，皇上不会不记得了吧（**注意，对于64位 系统的机器，lib-linux要改成lib-linux64**）

# 3.编译dol（续上页

然后是编译

**$ sudo ant -f build_zip.xml all**

约成功会显示build successful

接着可以试试运行第一个例子

进入build/bin/mian路径下

$ cd build/bin/main

然后运行第一个例子

$ sudo ant -f runexample.xml -Dnumber=1

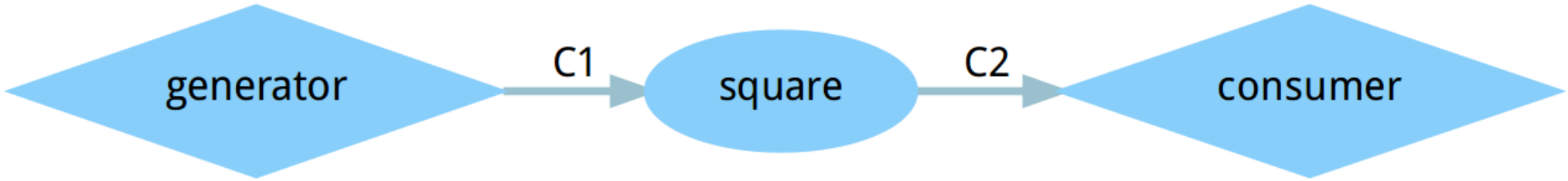
成功结果如图

```
    [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: 19 seconds
```

Run example1：

```
$ cd build/bin/main
$ ant -f runexample.xml -Dnumber=1
```



# 实验报告提交及要求：

暂无