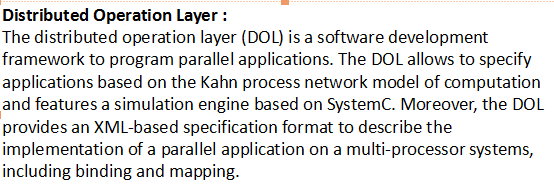
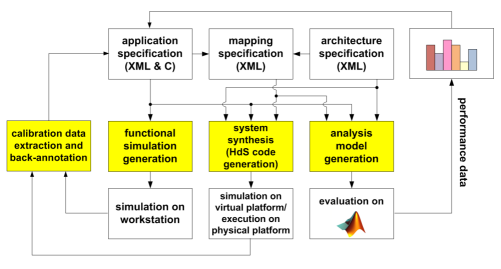
**DOL框架描述**

因为是初学者，所以说并不是很懂，只能随着后续的实验陆陆续续的研究出来的它的含义及用途。现在也就只能借用一下实验文档中的内容了。

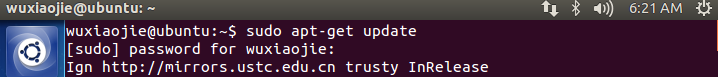


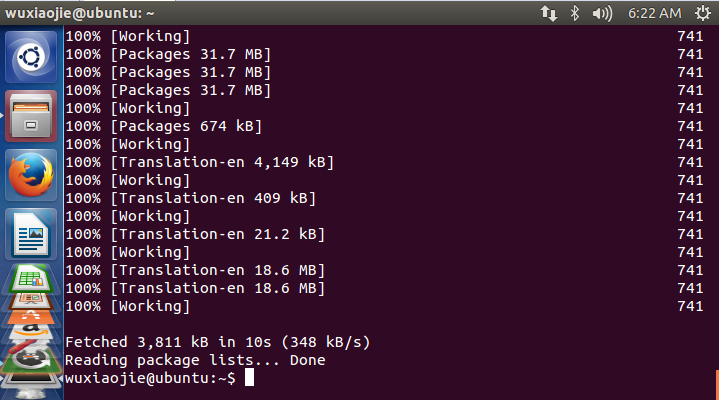


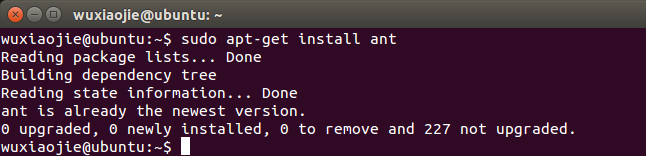
**DOL安装笔记**

**1.安装一些必要的环境：**

截图如下：







上述命令意为安装Ant工具，其为一种基于Java的build工具；

Ant用Java的类来扩展；

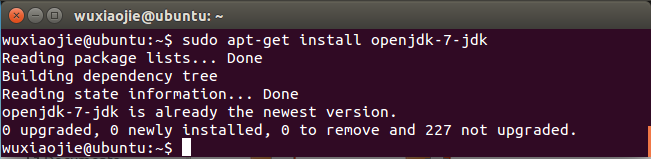
Ant本身就是这样一个流程脚本引擎，用于自动化调用程序完成项目的编译，打包，测试等。其优点有以下四点：

跨平台性。Ant是纯java语言编写的，所示具有很好的跨平台性。

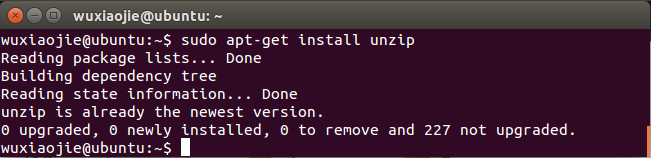
操作简单。Ant是由一个内置任务和可选任务组成的。Ant运行时需要一个XML文件(构建文件)。

容易维护和书写，结构清晰。

Ant可以集成到开发环境中。



以上是安装openjdk-7-jdk工具。

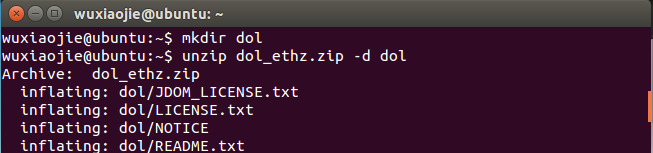


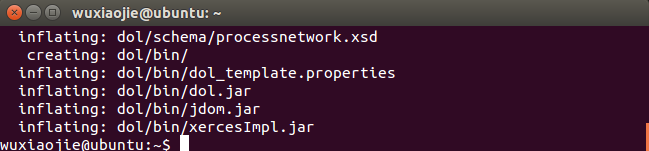
以上是安装解压工具。

**2.将已下载的文件压缩包dol\_ethz.zip和systemc-2.3.1.tgz拷贝到虚拟机中去。**

**3.解压文件。**

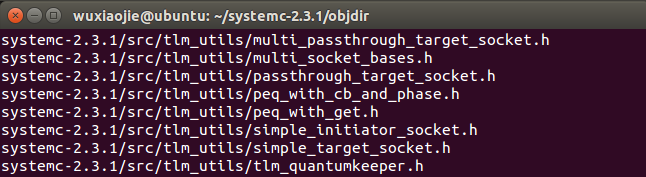
新建dol的文件夹以及将dolethz.zip解压到dol文件夹中，文件过多，没有全部截出来





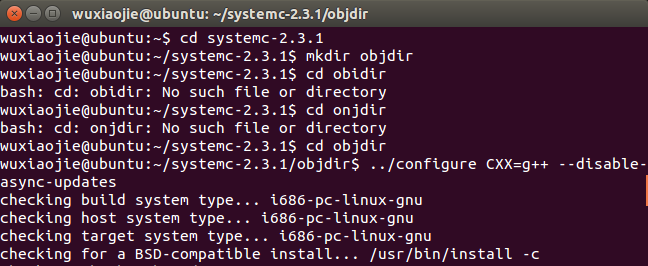
解压systemc，同样因为文件过多，没有全部进行截图



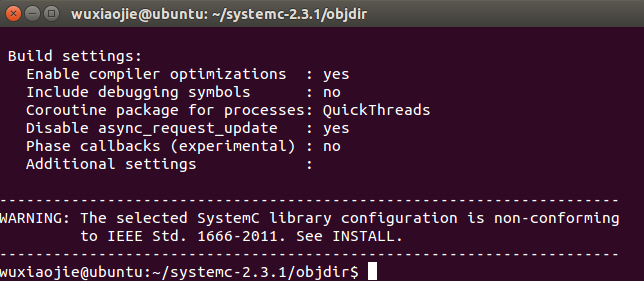


**4.编译systemc。**

进入systemc-2.3.1的目录下，新建一个临时文件夹objdir，进入该文件夹objdir，运行configure。



下图为运行configure之后的截图。



安装Make工具，下为简介：

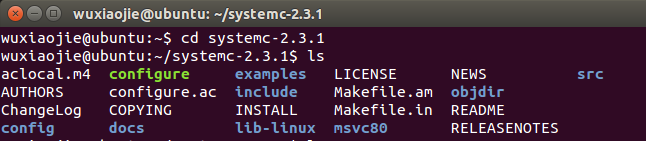
在Linux和Ubuntu环境中，make工具主要被用来进行工程编译和程序链接

Makefile文件：告诉make以何种方式编译源代码和链接程序

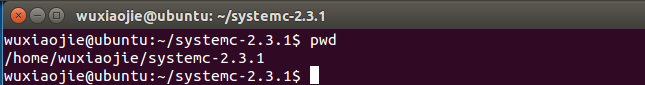
make通过比较对应文件（规则的目标和依赖）的最后修改时间，来决定哪些文件需要更新、那些文件不需要更新。



查看文件目录。

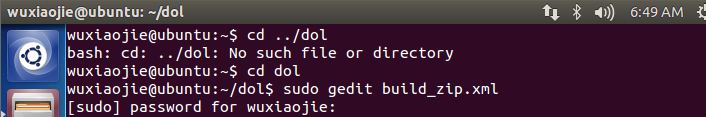


记录当前的工作路径，有用，以下表示我当前的工作路径为/home/wuxiaojie/systemc-2.3.1

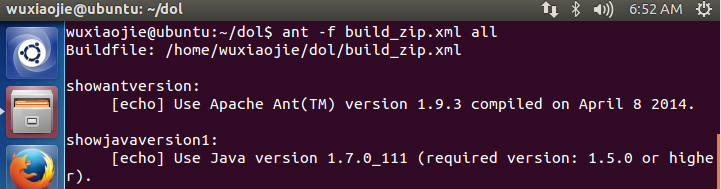


**5.编译dol。**

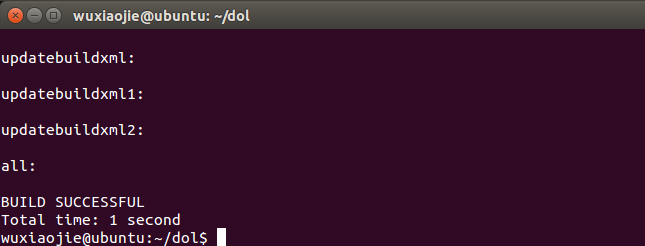
进入刚刚dol的文件夹，修改build\_zip.xml文件。



接下来就是编译了。

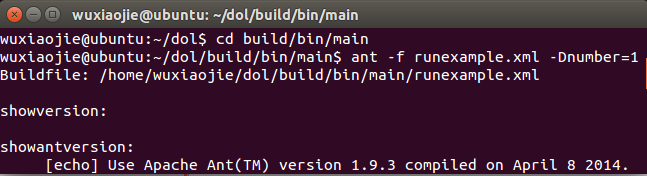


下图是编译成功的截图。

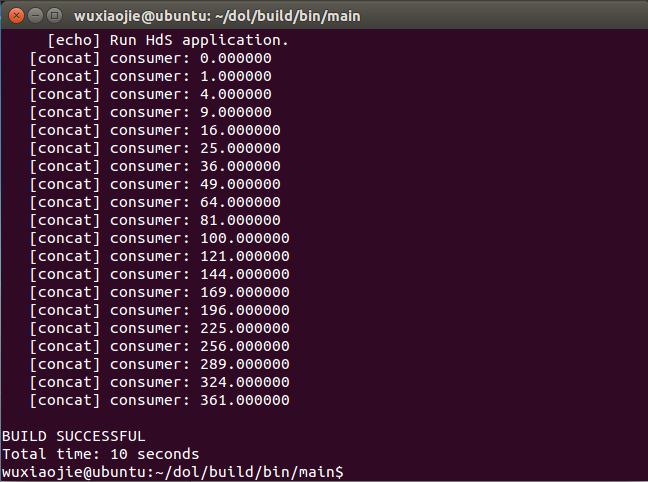


**6.试试运行第一个例子**

进入build/bin/main路径下并运行第一个例子。



下图是运行成功的截图。



**实验感想与心得**

本次实验（其实应该是上次实验）并没有太大的问题，毕竟只是简单的配置环境，其目的也只是为了后续实验做铺垫。不过到让我感受到了实验中一些工具的强大，比如Ant,Make之类的，然而现在还不是很会用，毕竟都是初学者。当然实验中也碰到了些许问题，比如因为不熟悉Ubuntu而导致在修改build\_zip.xml文件中的内容时陷入僵局，依靠同学的帮助才解决这个问题。