**一、安装虚拟机**

根据下列博客安装虚拟机和Linux环境

<https://blog.csdn.net/weixin_43465312/article/details/100233930>

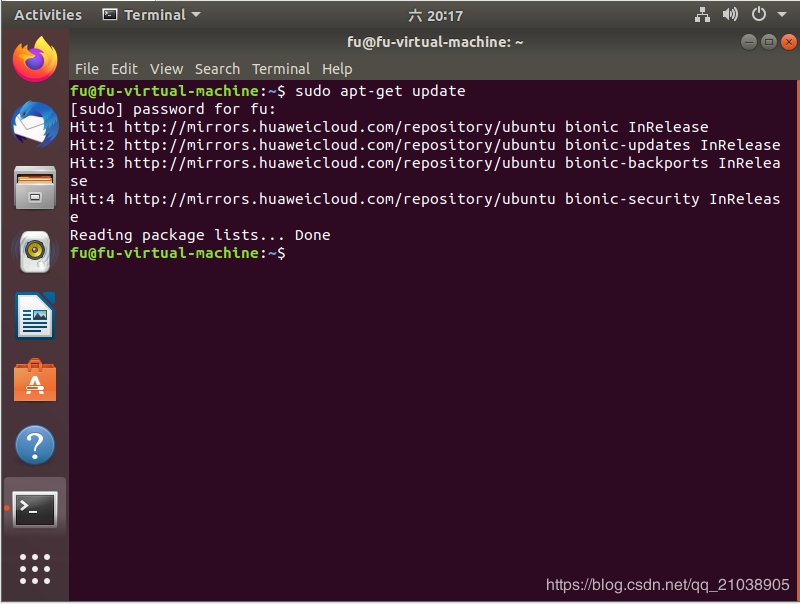
推荐使用Ubuntu18.04版本，喜欢尝试的可以使用最新的20.04版，但是会有很多包依赖的问题，需要多次指定包的旧版本号，才能解决依赖问题，比较繁琐。

**二、更新源**

新安装完成的虚拟机不要着急进行依赖安装，推荐更换源为国内的服务器，如下博客进行操作。

<https://blog.csdn.net/daerzei/article/details/84873964>

在桌面内进入终端，输入: sudo apt-get update,验证一下选择的源是否ok。



**三、安装依赖库**

推荐使用下列命令安装依赖库，可以傻瓜无脑安装，原实验指导书中的命令有依赖冲突问题，喜欢折腾的可以用实验指导书的命令，不过仍需要你们开动小脑瓜解决版本冲突问题。

以下命令来自ns3官方安装流程 <https://www.nsnam.org/wiki/Installation>

以下命令得逐条执行

sudo apt-get install gcc g++ python python3

sudo apt-get install gcc g++ python python3 python3-dev

sudo apt-get install python3-setuptools git mercurial

sudo apt-get install qt5-default mercurial

sudo apt-get install gir1.2-goocanvas-2.0 python-gi python-gi-cairo python-pygraphviz python3-gi python3-gi-cairo python3-pygraphviz gir1.2-gtk-3.0 ipython ipython3

sudo apt-get install openmpi-bin openmpi-common openmpi-doc libopenmpi-dev

sudo apt-get install autoconf cvs bzr unrar

sudo apt-get install gdb valgrind

sudo apt-get install uncrustify

sudo apt-get install doxygen graphviz imagemagick

sudo apt-get install texlive texlive-extra-utils texlive-latex-extra texlive-font-utils dvipng latexmk

sudo apt-get install python3-sphinx dia

sudo apt-get install gsl-bin libgsl-dev libgsl23 libgslcblas0

sudo apt-get install tcpdump

sudo apt-get install sqlite sqlite3 libsqlite3-dev

sudo apt-get install libxml2 libxml2-dev

sudo apt-get install cmake libc6-dev libc6-dev-i386 libclang-6.0-dev llvm-6.0-dev automake

sudo apt-get install libgtk2.0-0 libgtk2.0-dev

sudo apt-get install vtun lxc uml-utilities

sudo apt-get install libboost-signals-dev libboost-filesystem-dev

**四、安装ns3基础模块**

推荐使用3.27版本，最新版是3.30.1版本，会有一些编译问题。

打开桌面终端执行如下命令

cd

mkdir tarballs

cd tarballs

wget <http://www.nsnam.org/release/ns-allinone-3.27.tar.bz2> #（在14上最好用3.26）

下载过于缓慢的可以直接拷贝我上传在群文件中的文件。

然后执行下列命令进行解压缩（你用图形化界面的提取也可以，一样的）。

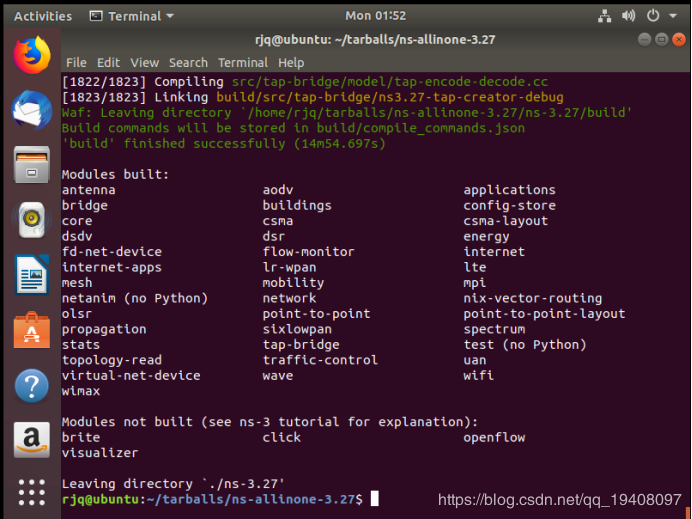
tar -xjvf ns-allinone-3.27.tar.bz2

然后进入文件夹，进行编译流程，有1800+文件，这个过程会比较缓慢，推荐设置虚拟机不要息屏，会产生中断，会有一些奇奇怪怪的问题。

cd ns-allinone-3.27

sudo ./build.py

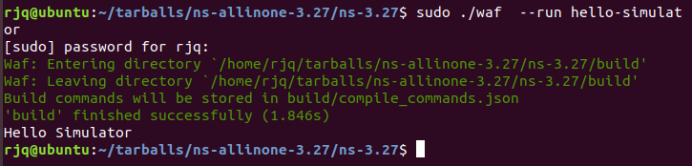
编译完成后会看见如下界面。



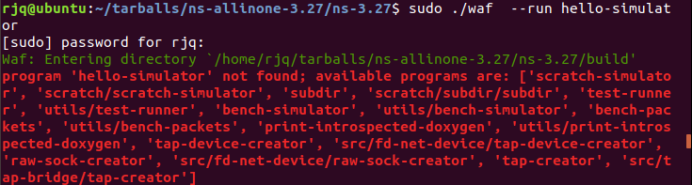
如果你执行下列命令，

cd ns-3.27

sudo ./waf --run hello-simulator

可以出现，下面界面，说明你安装成功了。

如果是如下界面，说明还有一些小问题，需要执行下列命令。



然后执行下列命令，再次进行编译，2500+文件。

cd ns-3.27

sudo ./waf clean

sudo ./waf -d debug --enable-example --enable-tests configure

sudo ./waf

再次执行上文的sudo ./waf --run hello-simulator ，应该就没问题了。

**五、编译NetAnim可视化部分**

返回目录ns-allinone-3.27，进入目录netanim-3.108，执行下列命令

sudo make clean

sudo qmake NetAnim.pro

sudo make

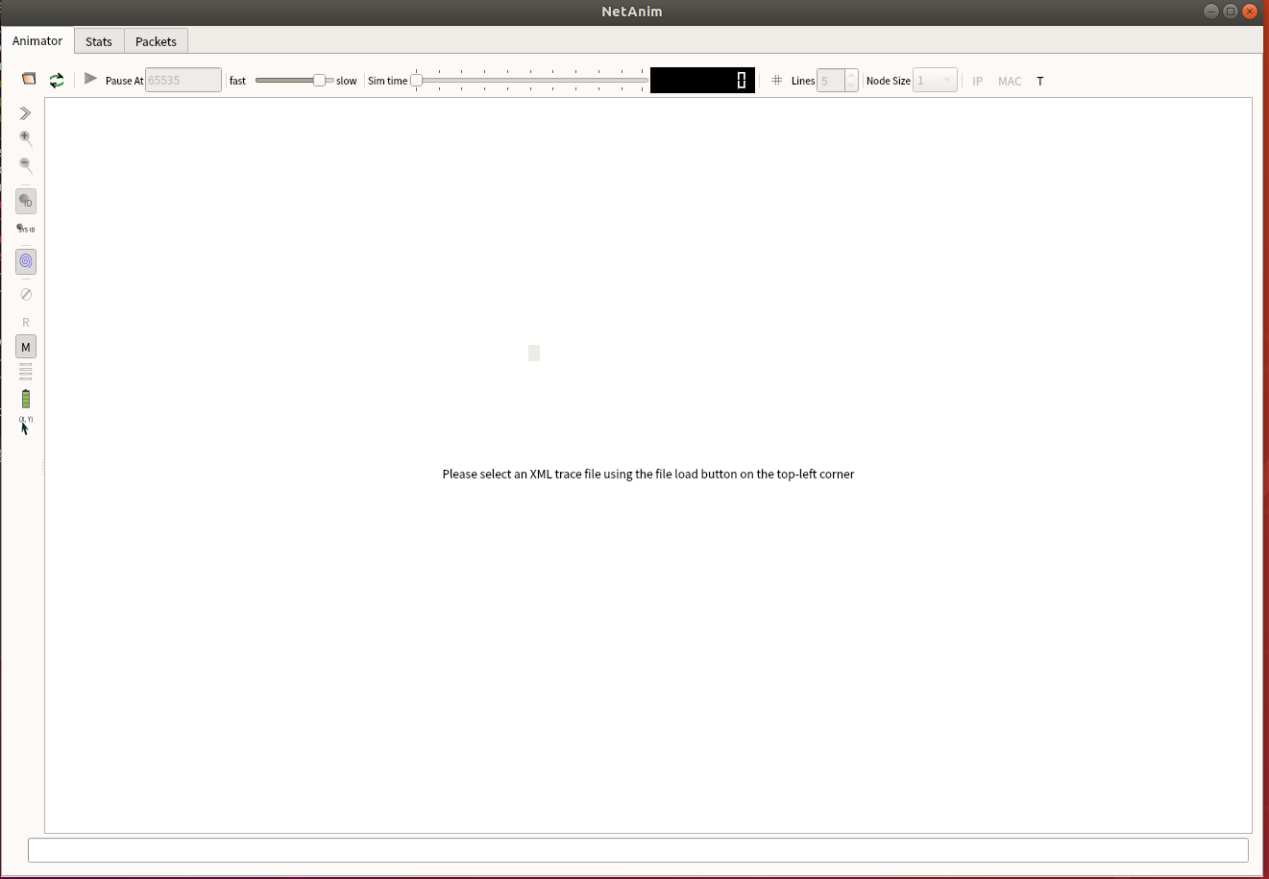
编译结束后执行

sudo ls

有NetAnim的话，执行

sudo ./NetAnim

打开如下仿真界面，



安装全部完成

附录：你可以运行下tutorial文件夹下的官方小例子，流程如下。

这里使用的NS3自带例子，所以可以直接进入ns-3.xx/examples/tutorial目录会发现first.cc脚本，要运行自己的脚本，你所需要做的仅仅是把你的脚本放到scratch目录下，通过waf,这样你的脚本就会被编译。

cp examples/tutorial/first.cc scratch/myfirst.cc

现在使用waf命令来编译自己的第一个实例脚本：

sudo ./waf

现在你能够运行这个例子（注意如果你在scratch目录编译了你的程序，你必须在scratch目录外运行它）：

注意这里是myfirst而不是myfirst.cc!

sudo ./waf --run scratch/myfirst

会出现如下的结果。

