# 新的Nachos4.1 安装教程

## 前言

原来的教程是上古时期的宝贝了，在下熬了两个通宵也没搞定他，索性自己做个新的。

如果你正常安装发现怎么都不行的话，

我们甚至可以从“重新安装一个虚拟机开始”。安装过程放在另一个文件当中。

## 1.首先为了方便后续安装我们换一个apt安装源

在终端输入：

sudo gedit /etc/apt/sources.list

换成清华的源

# 默认注释了源码镜像以提高 apt update 速度，如有需要可自行取消注释

deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy main restricted universe multiverse

# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy main restricted universe multiverse

deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-updates main restricted universe multiverse

# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-updates main restricted universe multiverse

deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-backports main restricted universe multiverse

# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-backports main restricted universe multiverse

deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-security main restricted universe multiverse

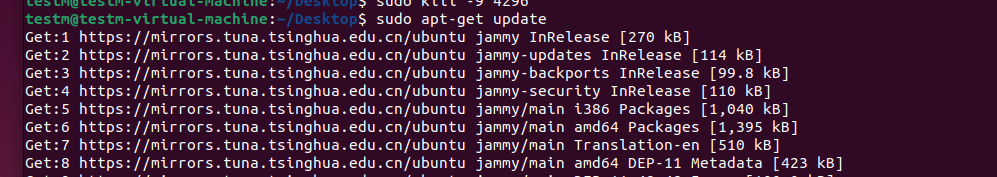
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-security main restricted universe multiverse

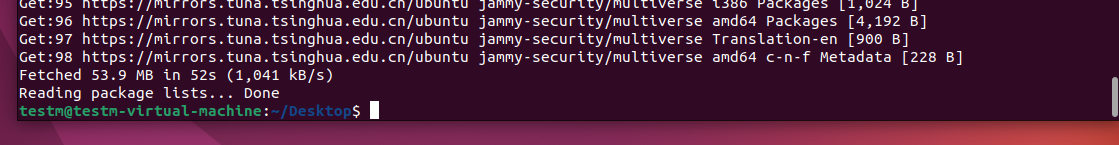
# 预发布软件源，不建议启用

# deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-proposed main restricted universe multiverse

# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-proposed main restricted universe multiverse

输入sudo apt-get update更新

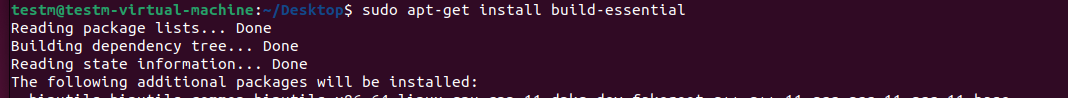


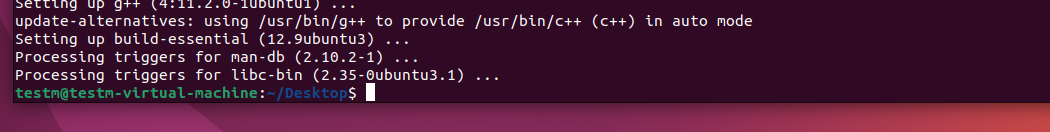


完成更新

## 1.安装编译基本工具包

sudo apt-get install build-essential





完成

## 2.检查系统是64位内核

dpkg --print-architecture



检查系统是否支持多 32 位

dpkg --print-foreign-architectures



I386表示支持

不支持的话

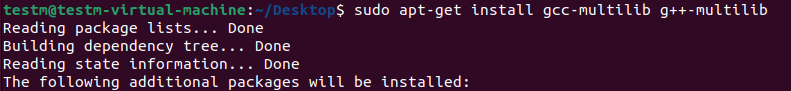
输入

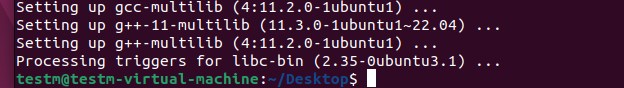
sudo dpkg --add-architecture i386

开启

## 3.安装C，C++多平台库

sudo apt-get install gcc-multilib g++-multilib

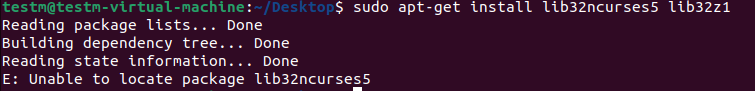




安装成功

## 4.安装32位环境库

sudo apt-get install lib32ncurses5 lib32z1

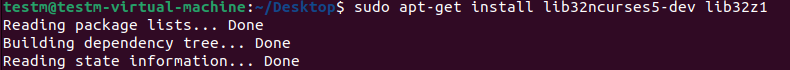


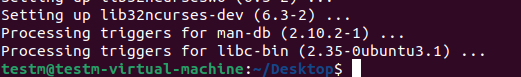
发现无法定位到包

这是由于不同版本的Kali Linux，它的源软件包名字会发生一些局部的变化，这都是很正常的

将命令换成

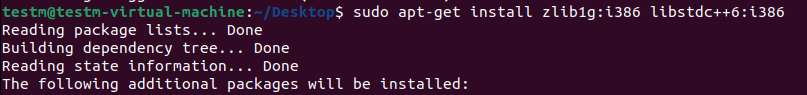
sudo apt-get install lib32ncurses5-dev lib32z1

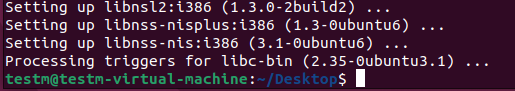




完成

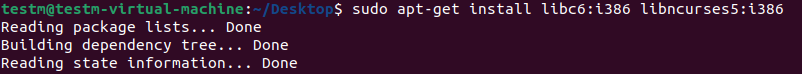
sudo apt-get install zlib1g:i386 libstdc++6:i386

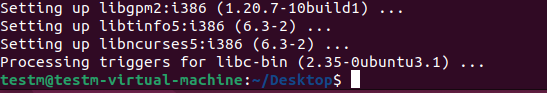




完成

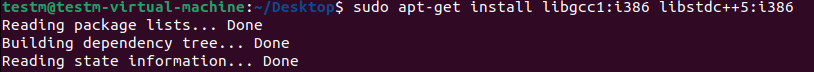
sudo apt-get install libc6:i386 libncurses5:i386

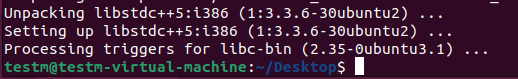




完成

sudo apt-get install libgcc1:i386 libstdc++5:i386

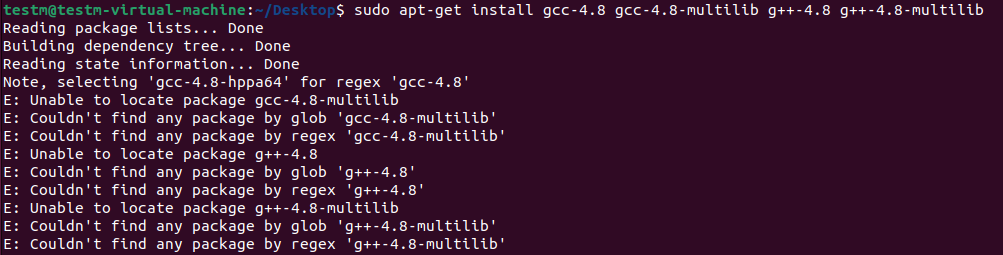




完成

## 5.安装低于 gcc5.0 的版本

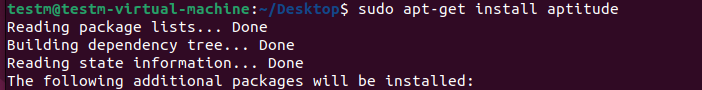
sudo apt-get install gcc-4.8 gcc-4.8-multilib g++-4.8 g++-4.8-multilib



发现定位不到这些包

我们安装aptitude

sudo apt-get install aptitude



之后使用aptitude安装

并且因为这些包版本太老了，我们需要换源

在终端输入：

sudo gedit /etc/apt/sources.list

打开文件

换成下列内容

deb http://mirrors.aliyun.com/ubuntu/ xenial main

deb-src http://mirrors.aliyun.com/ubuntu/ xenial main

deb http://mirrors.aliyun.com/ubuntu/ xenial-updates main

deb-src http://mirrors.aliyun.com/ubuntu/ xenial-updates main

deb http://mirrors.aliyun.com/ubuntu/ xenial universe

deb-src http://mirrors.aliyun.com/ubuntu/ xenial universe

deb http://mirrors.aliyun.com/ubuntu/ xenial-updates universe

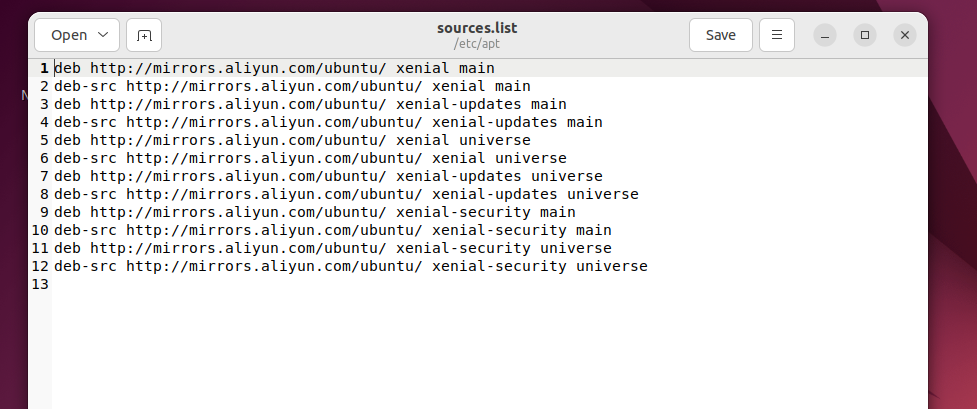
deb-src http://mirrors.aliyun.com/ubuntu/ xenial-updates universe

deb http://mirrors.aliyun.com/ubuntu/ xenial-security main

deb-src http://mirrors.aliyun.com/ubuntu/ xenial-security main

deb http://mirrors.aliyun.com/ubuntu/ xenial-security universe

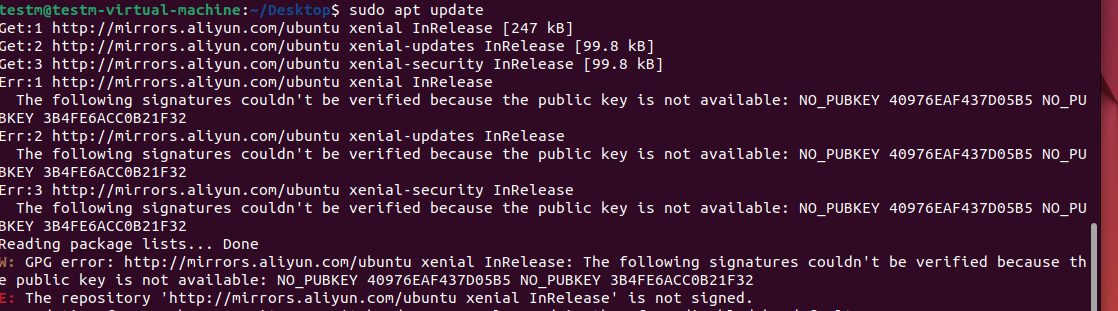
deb-src http://mirrors.aliyun.com/ubuntu/ xenial-security universe



保存后关闭

添加源后更新，输入

sudo apt update

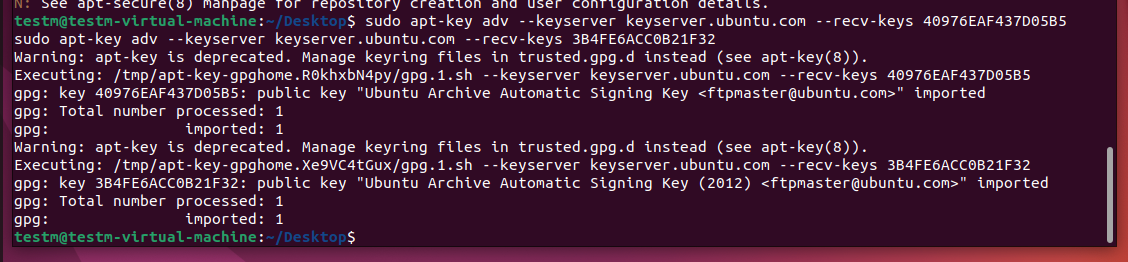


发现报错，我们缺少那个公钥

那就添加，输入

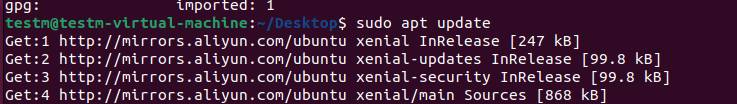
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 40976EAF437D05B5

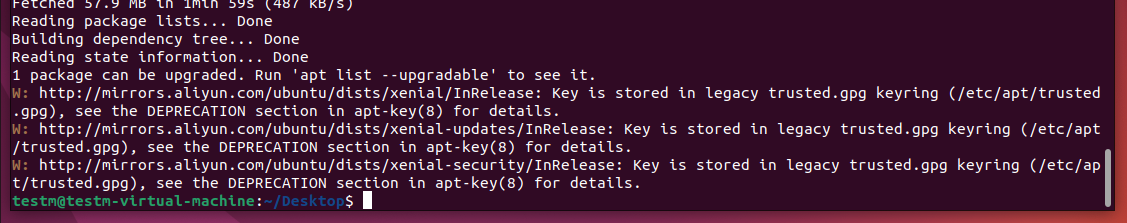
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 3B4FE6ACC0B21F32



添加成功，我们再次更新

输入sudo apt update





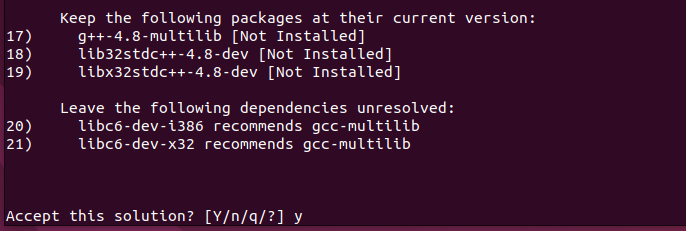
可以看到完成更新了，有3个警告，无所谓~

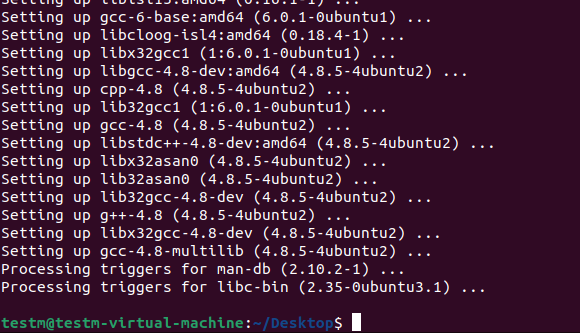
输入

sudo aptitude install gcc-4.8 gcc-4.8-multilib g++-4.8 g++-4.8-multilib

sudo apt-get install gcc-4.8-multilib g++-4.8-multilib

他提供了方案



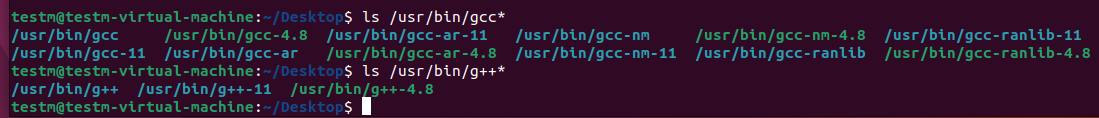


成功！

查看安装的gcc 和g++

ls /usr/bin/gcc\*

ls /usr/bin/g++\*

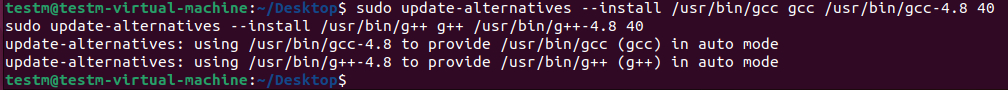


将低版本 gcc 与 g++切换为当前活跃版本

sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-4.8 40

sudo update-alternatives --install /usr/bin/g++ g++ /usr/bin/g++-4.8 40

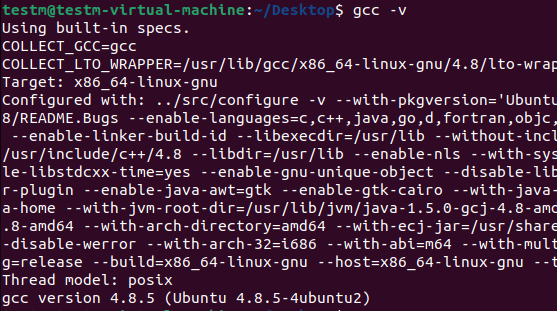
最后那个数字是优先级 越大越优先

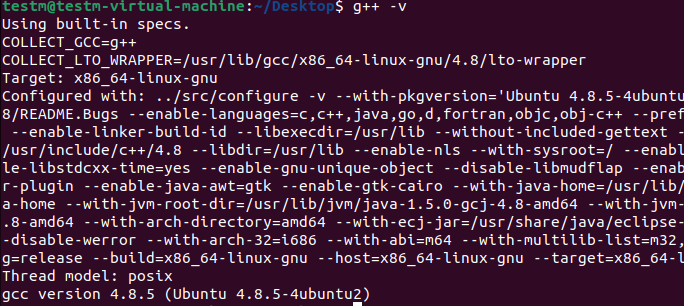


查看当前gcc，g++版本

gcc -v

g++ -v





好的我们已经完成最麻烦的一步了(好累！)

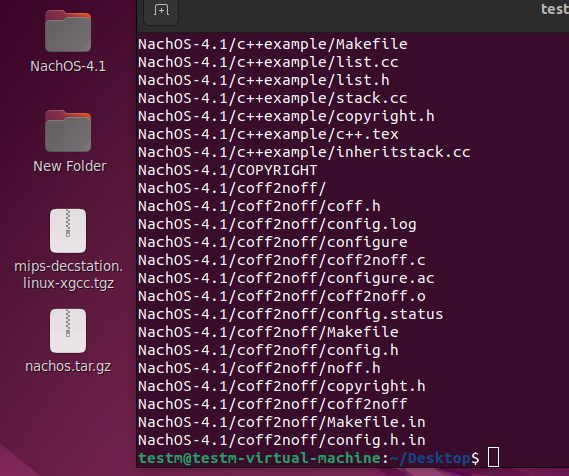
## 6.解压 nachos 源码

当然你要先把这个压缩文件拖到桌面，如果不能拖的话，需要进行VMtools的安装。

[VMware——VMware Tools的介绍及安装方法\_William.csj的博客-CSDN博客](https://blog.csdn.net/williamcsj/article/details/121019391)

[安装 VMware Tools](https://docs.vmware.com/cn/VMware-Tools/12.3.0/com.vmware.vsphere.vmwaretools.doc/GUID-D8892B15-73A5-4FCE-AB7D-56C2C90BD951.html)

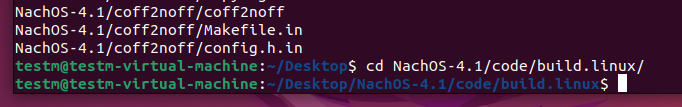
tar xvzf nachos.tar.gz



解压完成，产生了一个叫NachOS-4.1的文件夹

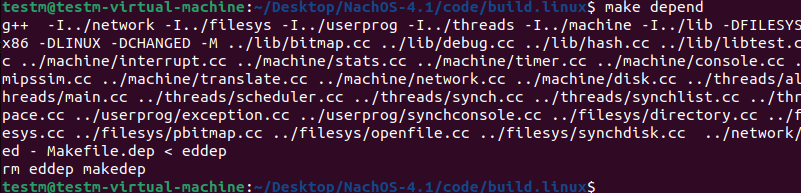
进入目录

cd NachOS-4.1/code/build.linux/



进行编译

make depend



Make

这个地方会狂报错，各种缺文件

我选择执行了一下前面的命令

sudo apt-get install build-essential

sudo apt-get install lib32ncurses5 lib32z1

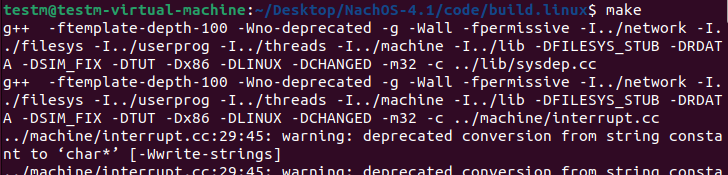
sudo apt-get install zlib1g:i386 libstdc++6:i386

sudo apt-get install libc6:i386 libncurses5:i386

sudo apt-get install libgcc1:i386 libstdc++5:i386

sudo aptitude install gcc-multilib g++-multilib

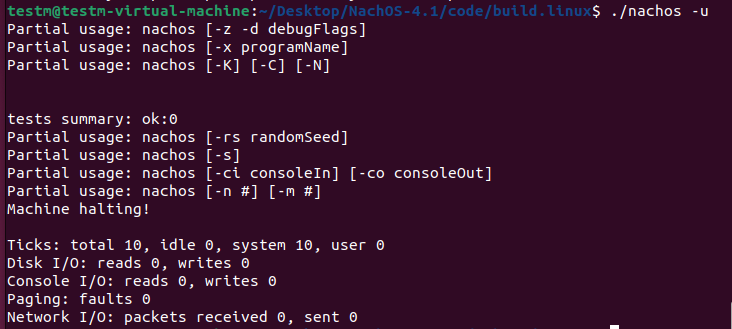
然后还是不行，之后我重启了一下客户机，居然成功了



如果尝试重启没有成功，可以考虑换一个版本的ubuntu，重新进行一边上述步骤，本次是在22版本的Ubuntu下成功，到这一步我重新执行了一遍以下指令，然后就可以编译了

sudo apt-get install gcc-4.8-multilib g++-4.8-multilib

[Index of /ubuntu-releases/22.04.3/ (ustc.edu.cn)](http://mirrors.ustc.edu.cn/ubuntu-releases/22.04.3/)



这个地方我做了两次，都是缺文件，然后乱操作一通，莫名其妙就可以编译了，我也不清楚原因，大家多尝试一下吧。

总之我们已经成功安装了

## 7.建立交叉编译环境

将mips-decstation.linux-xgcc.tgz压缩文件拷贝到根目录

sudo mv mips-decstation.linux-xgcc.tgz /.



进入根目录

解压

cd /

sudo tar xvzf mips-decstation.linux-xgcc.tgz



解压完成

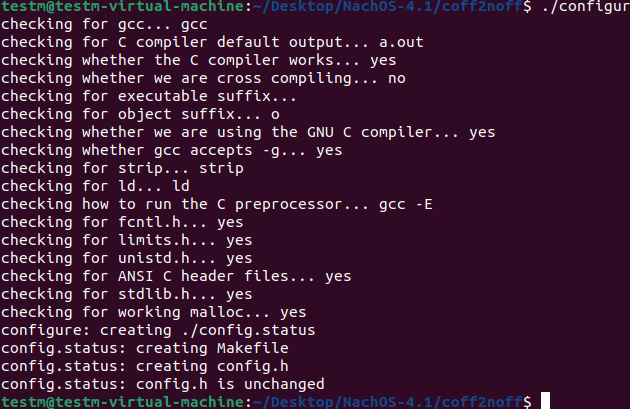
下面编译可执行文件格式转换器

进入到NachOS文件夹下面的coff2noff文件中



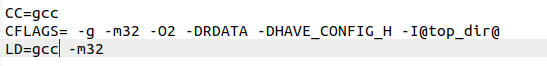
运行文件

./configure



然后打开生成的Makefile文件

修改为



注意有两行要添加 -m32

修改之后

终端输入

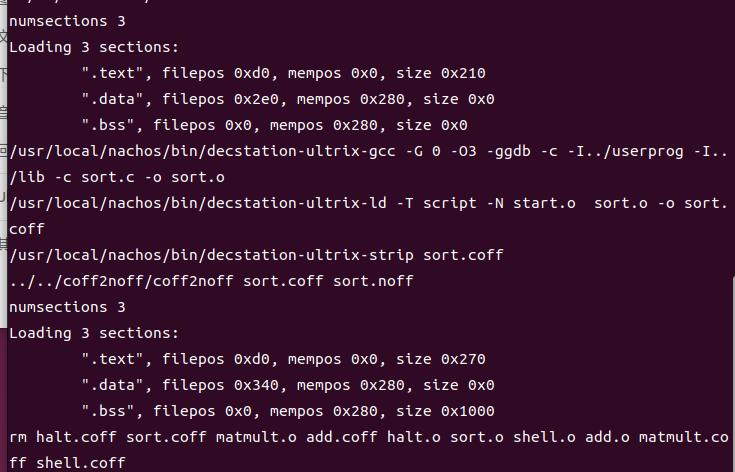
make clean

make

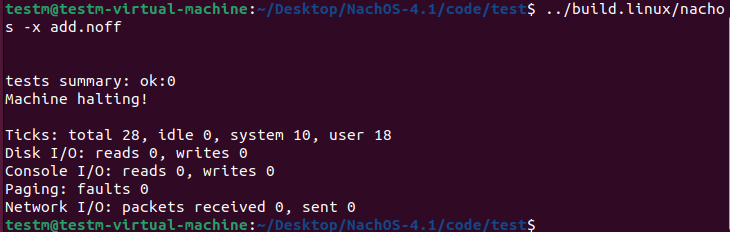
之后进入到Nachos目录下code子目录下的test目录

检查Makefile.dep下的交叉编译器路径是否正确

编译test目录下的用户程序 make



测试用户态程序是否能在nachos 上运行



### 如果可以看到系统运行界面，则说明程序运行成功