# OS开发环境配置

主要有两种方法：

1. [自行安装环境](#_自行安装环境)
2. [直接使用打包好的虚拟机](#_使用打包好的虚拟机)

## 自行安装环境

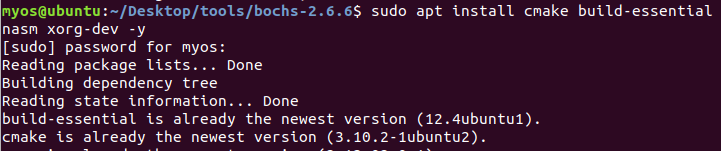
此教程中各个步骤是在Ubuntu 18.04下操作的。

1. 安装必要的软件包

Ctrl+Alt+t打开终端，并在终端中输入以下命令即可。

sudo apt install cmake build-essential nasm xorg-dev -y

其中nasm为汇编编译器，build-essential提供了C/C++的编译环境，cmake提供build system。

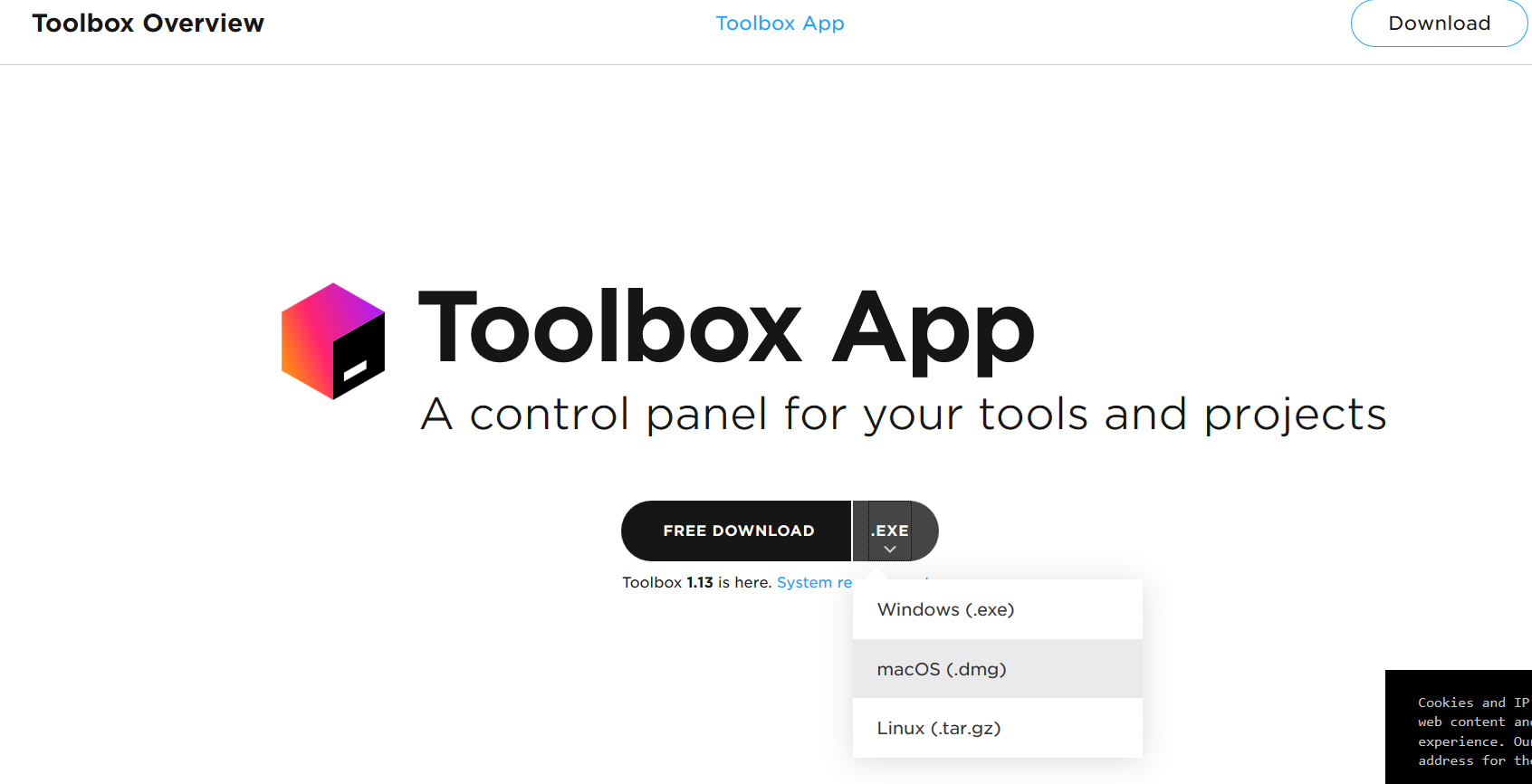


1. 安装IDE

为了使用cmake，我们将使用JetBrains Clion 来进行代码的编辑。此IDE对CMake提供了相当好的支持。这款IDE本来是收费的，但是我们凭借教育邮箱可以免费使用到本科毕业。

1. 下载安装包

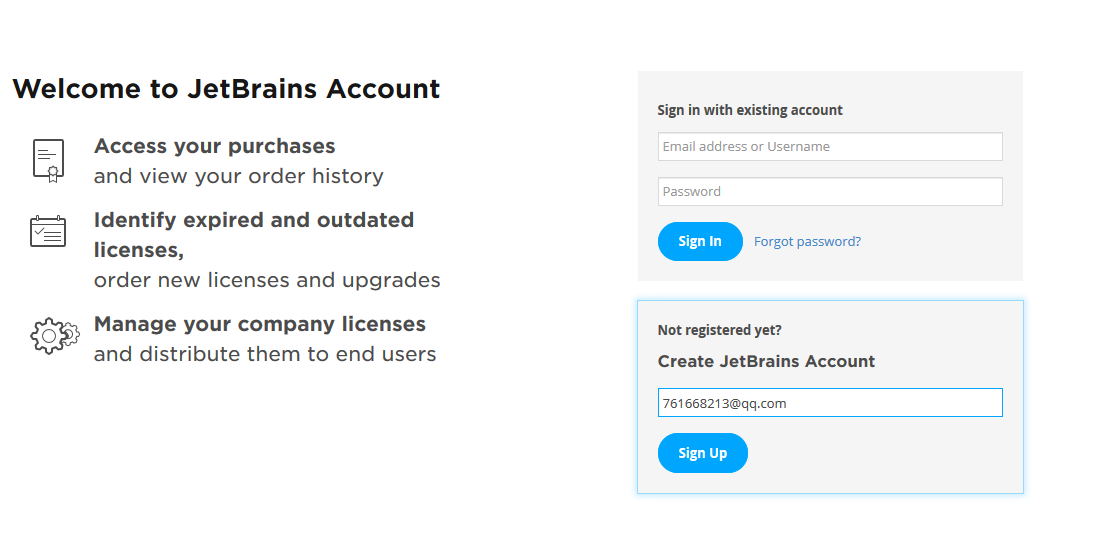
首先，我们需要从官网下载JetBrains Toolbox安装文件，下载地址为<https://www.jetbrains.com/toolbox/app>，注意选择Linux版本。



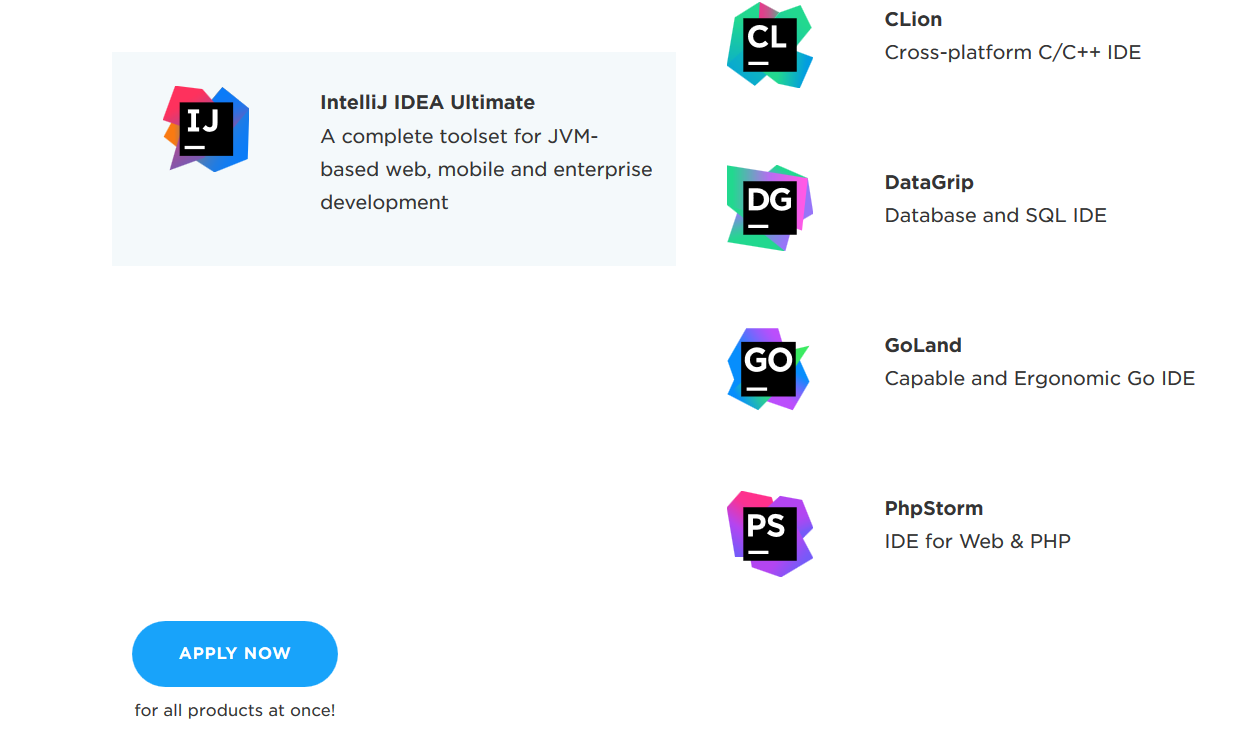
1. 注册JetBrain Account账号

为了使用JetBrain Toolbox，我们需要注册JetBrain Account账号，并使用教育邮箱申请一个免费的lisence。

首先，访问<https://account.jetbrains.com/login> 填写自己的邮箱，之后会收到JetBrain的邮件，在邮件中提供的网址上注册即可。

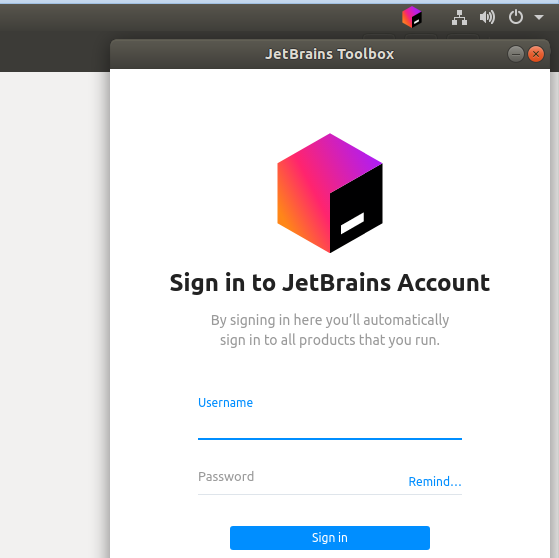


然后，使用注册的账号登录网站，并访问<https://www.jetbrains.com/student/>。点击页面最下方的Apply，在弹出的页面中填写相应信息即可获取到免费的lisence。

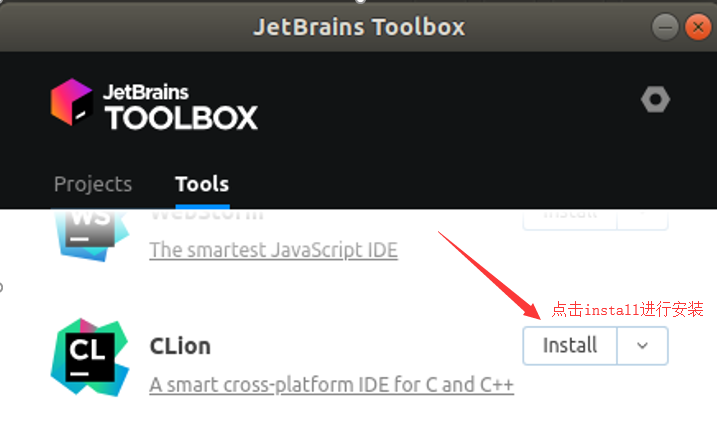


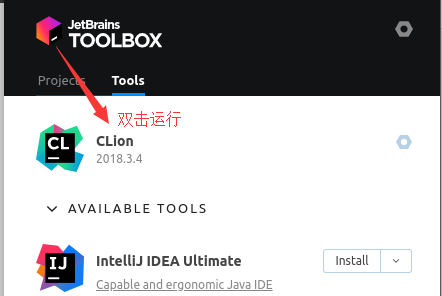
1. 运行Toolbox并安装CLion

首先，将安装文件放入Linux虚拟机并解压，之后得到一个名为jetbrains-toolbox的可执行文件。运行之，即进入ToolBox的登录界面。

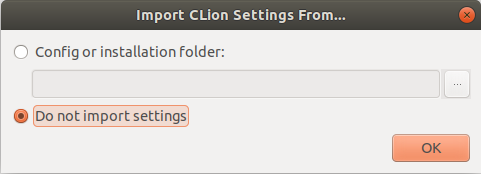


然后，在Tools中找到CLion，点击Install进行安装即可。安装完毕之后双击CLion运行之。

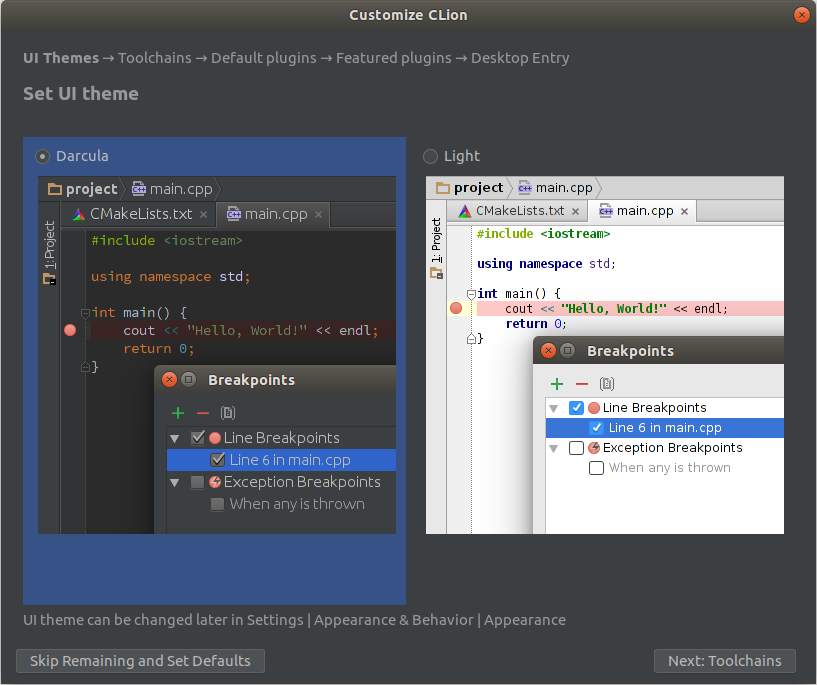




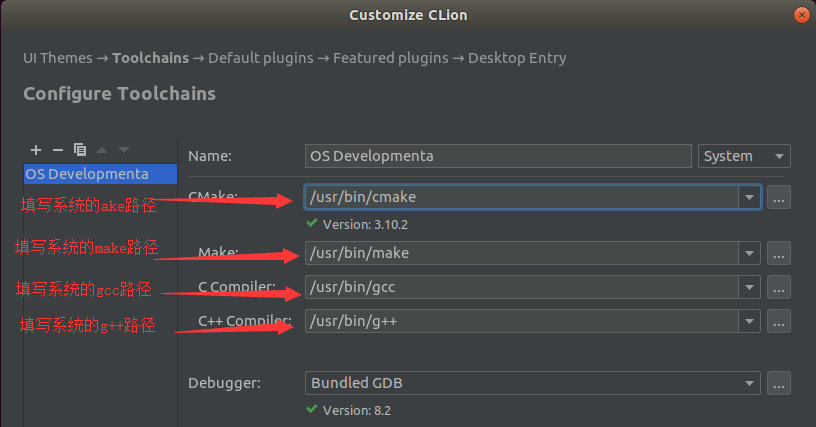
运行之后，在弹出的对话框中选择“Do not import settings”并点击OK。如下图所示。



接下来，在新的对话框中选取UI主题，然后点Next即可。



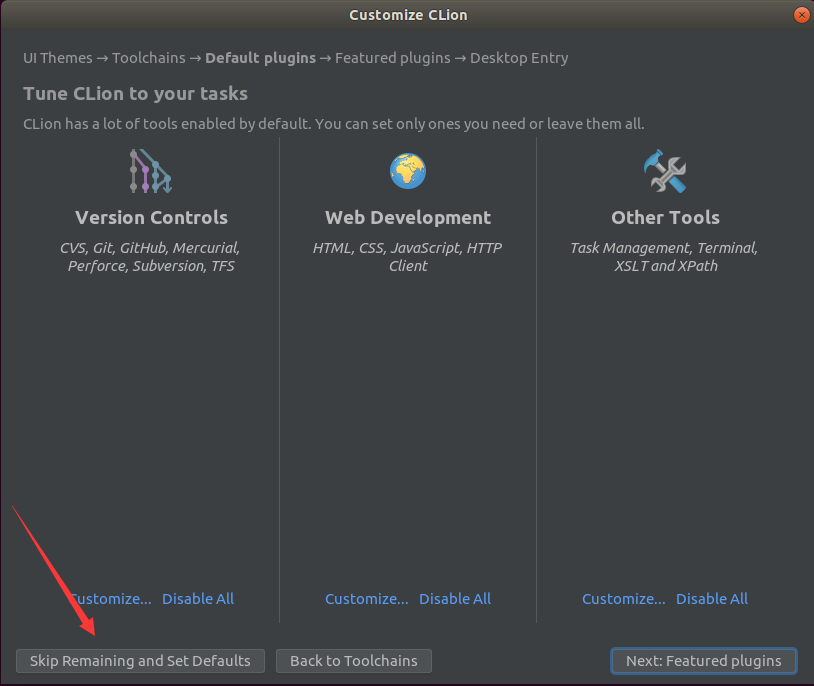
之后，对话框中设置要使用的工具，将CMake设置为cmake的路径，Make设置为make的路径，C Compiler设置为gcc的路径，C++ Compiler 设置为g++的路径（如图所示）。这些路径可以通过which命令来获取。之后点击“Next: Default Plugin”





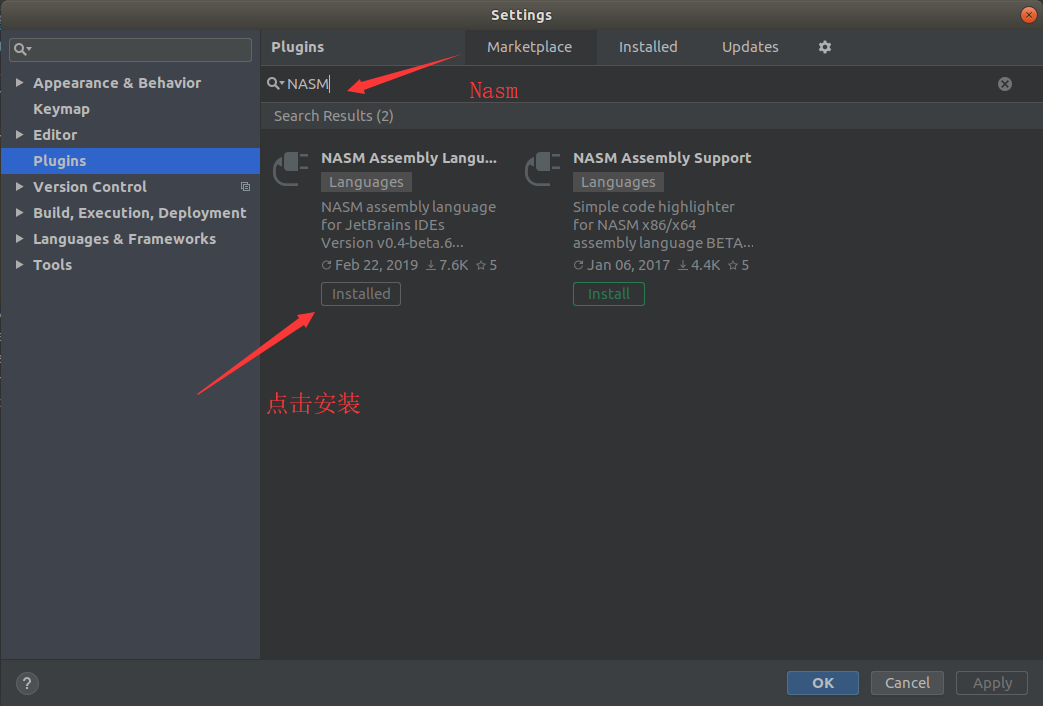
用which获取g++所在的路径

最后单击左下角的“Skip Remaining and Set Defaults”即完成CLion的配置。



1. 安装nasm语法高亮插件

任意新建一个项目，选择“”，在其中搜索”Nasm”。在搜索结果中选择“NASM Assembly Language”。安装完毕后重启CLion即可。



1. 编译并安装bochs

Bochs是一个x86架构的纯软件的虚拟机，其自带反汇编调试器同时支持远程GDB调试。由于两种调试模式互不兼容但是都很常用，因此我们将编译两个版本的bochs。

1） 下载Bochs源码

首先，我们需要下载Bochs的源码。Bochs的源码可以从sourceforge上下载,大家可以选择合适的版本进行下载。下载时注意下载后缀为src.rpm的源码。

SourceForge下载 : <https://sourceforge.net/projects/bochs/files/bochs/>

2）编译源码并安装

然后将下载的文件放到合适的位置并解压缩，得到一个压缩包。将得到的压缩包再解压缩得到包含了源码的文件夹bochs。打开终端，切换到源码的文件夹中就可以开始安装了。

**编译自带反汇编调试器的bochs**

在终端中输入以下命令：

./configure --prefix=/usr/local/share/bochs/disasm --enable-disasm --enable-all-optimizations --enable-cpu-level=6 --enable-debugger

make

sudo make install

**编译支持gdb远程调试的bochs**

在终端中输入以下命令

./configure --prefix=/usr/local/share/bochs/gdb --enable-gdb-stub --enable-all-optimizations --enable-cpu-level=6 --with-x

make

sudo make install

3）修改环境变量

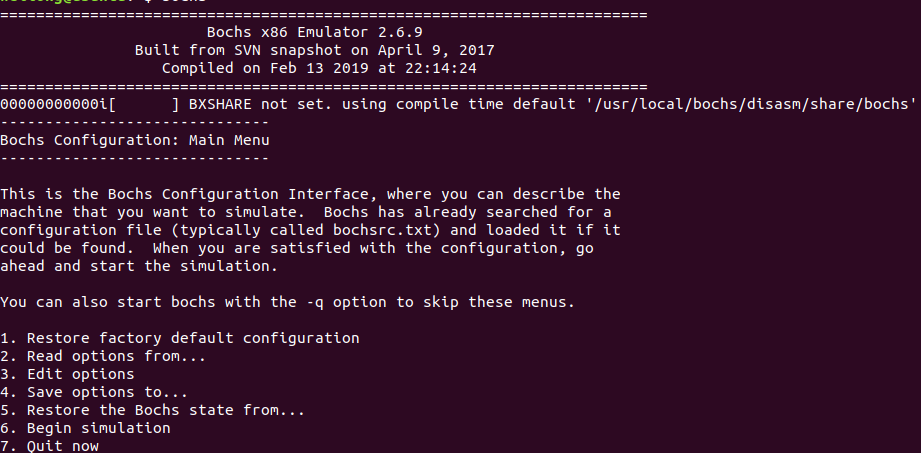
在前面的安装过程中，我们将bochs的两个版本分别安装到了/usr/local/share/bochs/disasm与/usr/local/share/bochs/gdb下。为了能够从终端中直接运行bochs，我们需要修改用户目录下的.bashrc，在终端中输入：

gedit ~/.bashrc

然后在该文件的最后面增加一行:

export PATH=${PATH}:/usr/local/share/bochs/disasm/bin/:/usr/local/share/bochs/gdb/bin/

最后保存文件并重新登录操作系统。之后打开终端，输入bochs检查是否可以正常运行，如果显示下图所示的结果，这说明bochs安装成功了。



虽然我们可以在终端中直接运行bochs，但是两个版本的bochs的可执行文件名称相同，我们需要对这两个版本的bochs进行区分。方法是为两个版本的bochs创建不同名的软链接。在终端中输入：

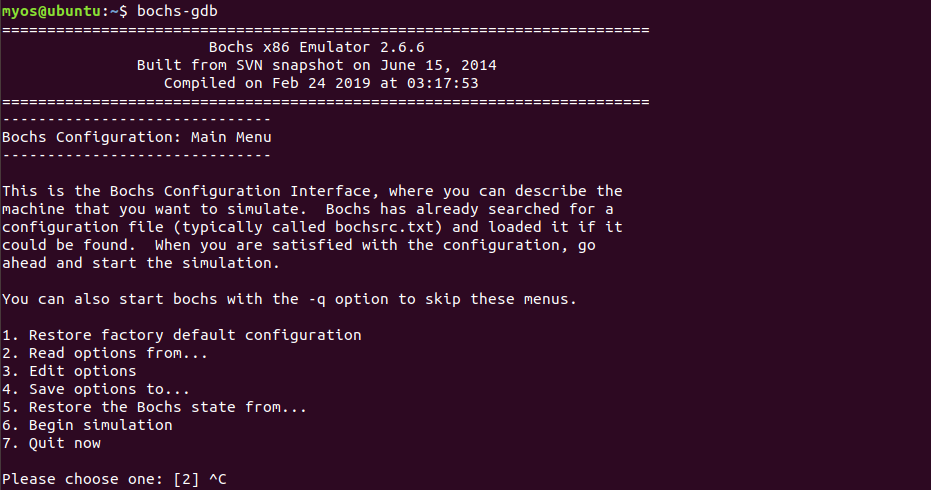
sudo ln -s /usr/local/share/bochs/gdb/bin/bochs /usr/local/share/bochs/gdb/bin/bochs-gdb

sudo ln -s /usr/local/share/bochs/gdb/bin/bximage /usr/local/share/bochs/gdb/bin/bximage-gdb

sudo ln -s /usr/local/share/bochs/disasm/bin/bochs /usr/local/share/bochs/disasm/bin/bochs-disasm

sudo ln -s /usr/local/share/bochs/disasm/bin/bximage /usr/local/share/bochs/disasm/bin/bximage-disasm

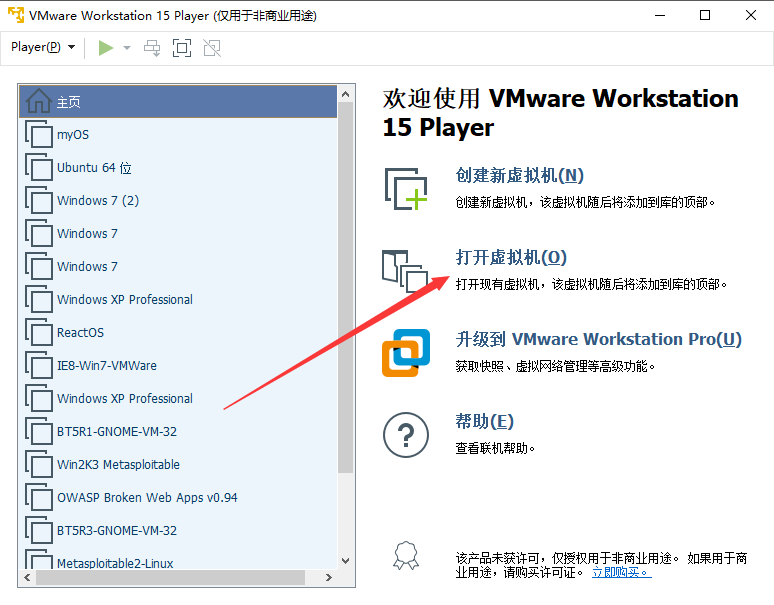
之后，在终端中分别输入bochs-gdb，bochs-disasm 检查是否可以运行。



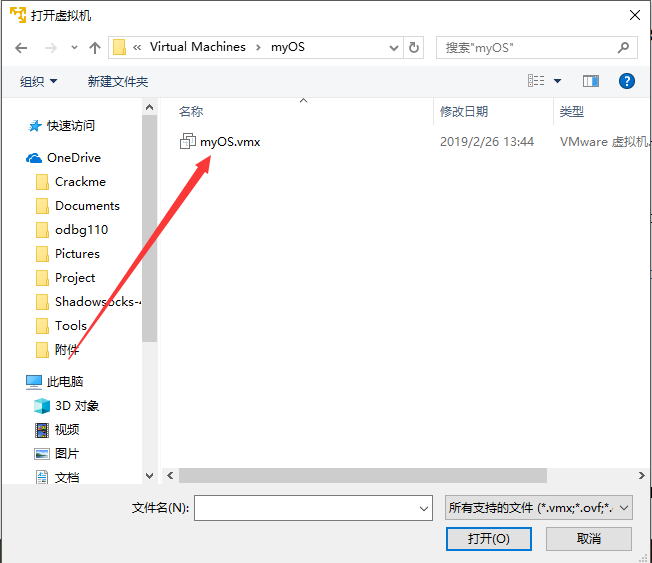
## 使用打包好的虚拟机

我们提供了已经装好环境的虚拟机，大家可以下载后使用。下面简单介绍一下如何使用。

1. 对打包好的虚拟机进行解压缩。
2. 打开VMware 选择“打开虚拟机”选项。



1. 在压缩出来的文件夹中，选择vmx文件并打开，然后开启虚拟机。



1. 开机过程中可能会有以下提示，选择“我已复制虚拟机“即可。

