

Non-OS 编译环境使用及编译环境搭建

该文档包括 Non-OS 编译环境的使用和编译环境的搭建，推荐使用我司提供的安装好编译环境的 Ubuntu 系统，如果你对 Ubuntu 系统比较熟悉或者你电脑上已经有安装好的 Ubuntu 系统，你也可以选择按照该文档第二部分 Non-OS 编译环境搭建来搭建编译环境。

一、Non-OS 编译环境使用说明

注：我司提供的 ubuntu 系统账户名和密码如下：

账户名：smart

密码：123456

1、解压 ubuntu 系统

将我司提供的 Ubuntu12.04.rar 解压到你的工作目录下，解压出来的是一个在虚拟机下安装好的 ubuntu 12.04 系统(**注意：这个系统是在 VMware10.0 下安装的，安装系统是没有勾选兼容 VMware10.0 以下的版本，所以 VMware10.0 以下的版本不能打开此系统**)，要打开这个 ubuntu 系统需要安装一个虚拟机，推荐使用 VMware Player，下面是 VMware Player 安装步骤。

2、安装 VMware Player

注意：如果你的电脑已经安装过其它 VMware 软件，需要将已经安装好的 VMware 软件卸载之后才能安装 VMware Player

双击 VMware_player_12.11.exe 开始安装 VMware Player



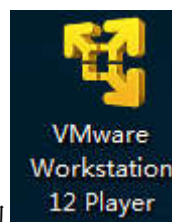




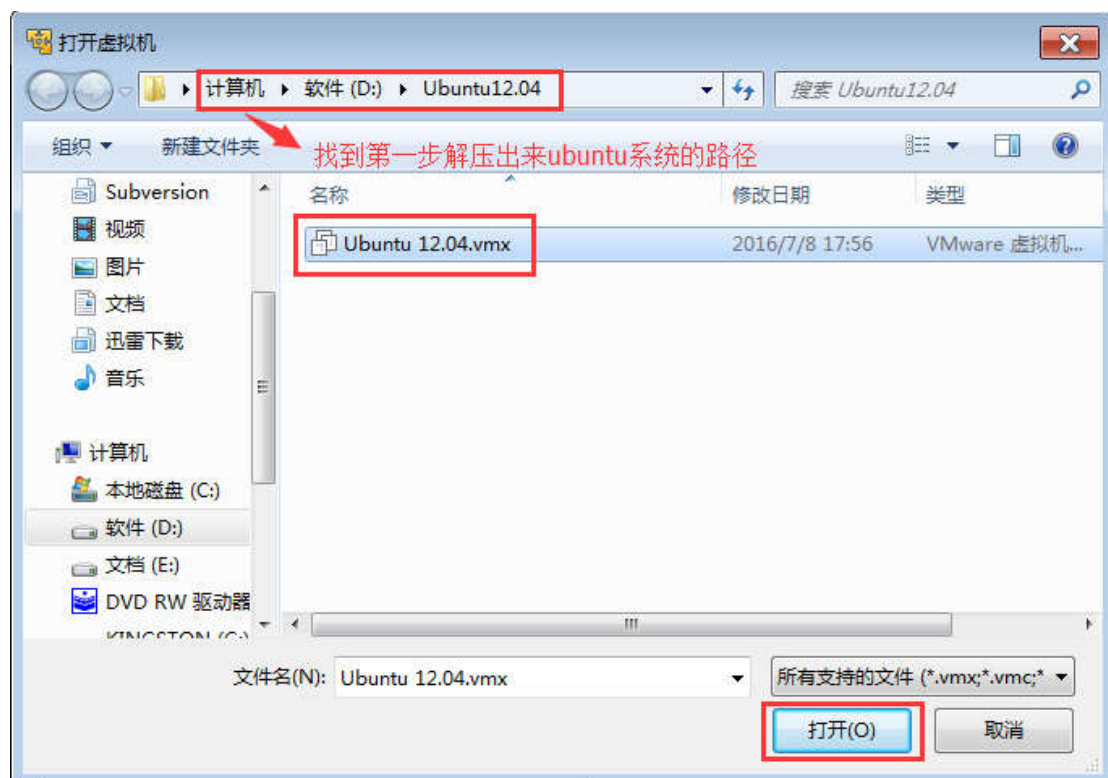


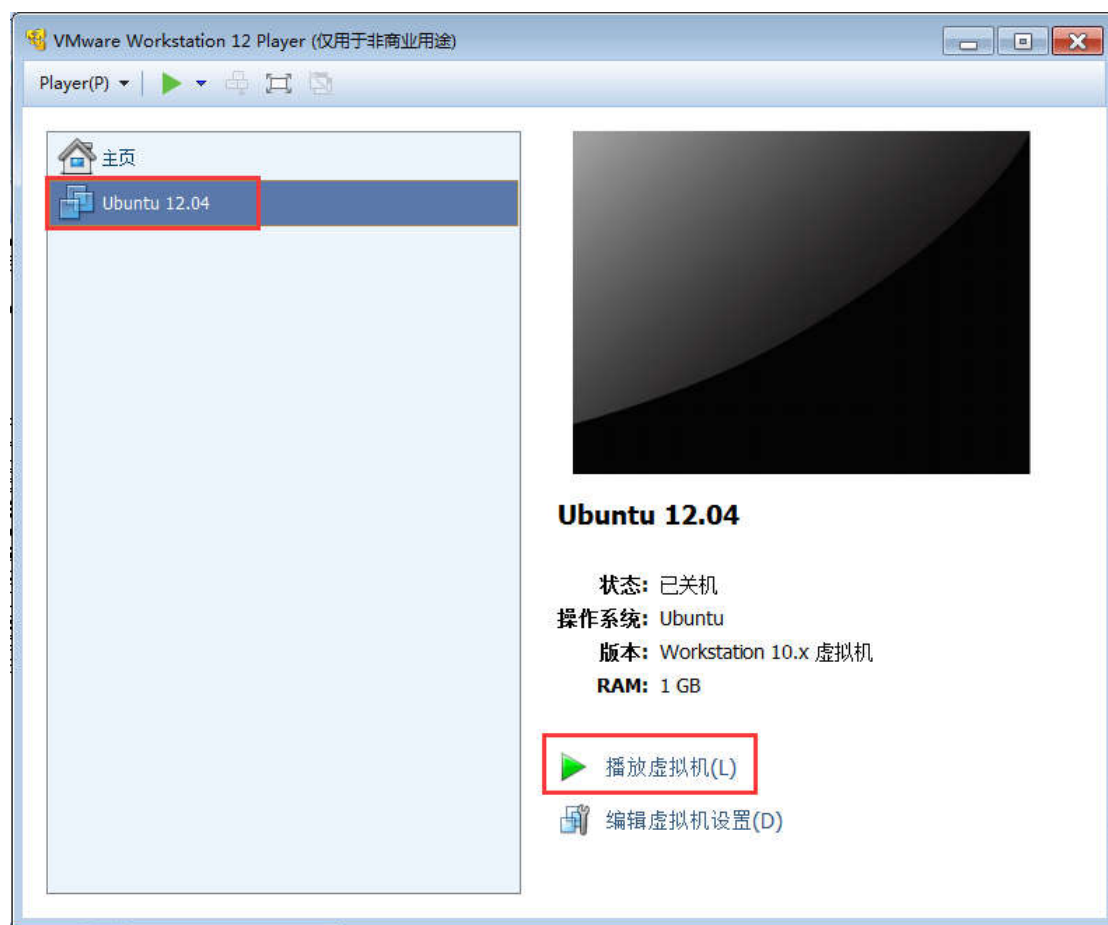
注意：安装完成后如果是第一次打开会提示提示要输入邮箱或许可证，你只需要任意输入一个邮箱然后点击继续就可以使用了

3、打开 ubuntu 系统

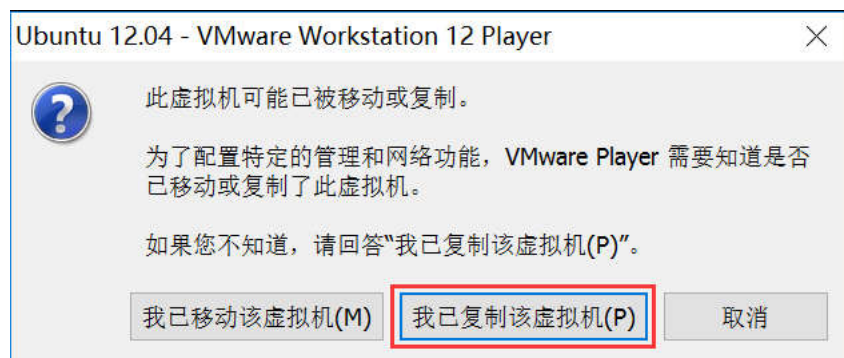


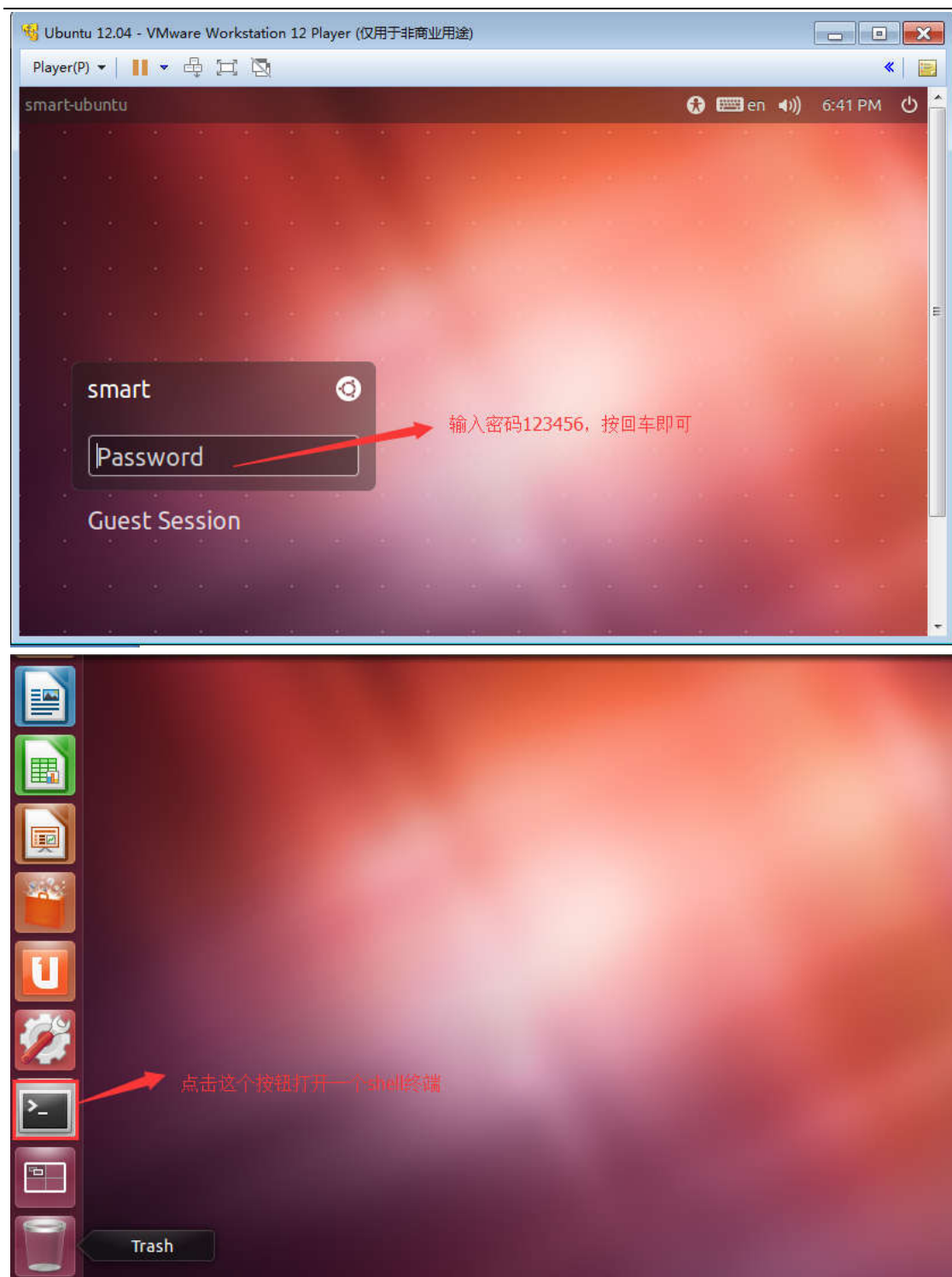
双击桌面上的图标



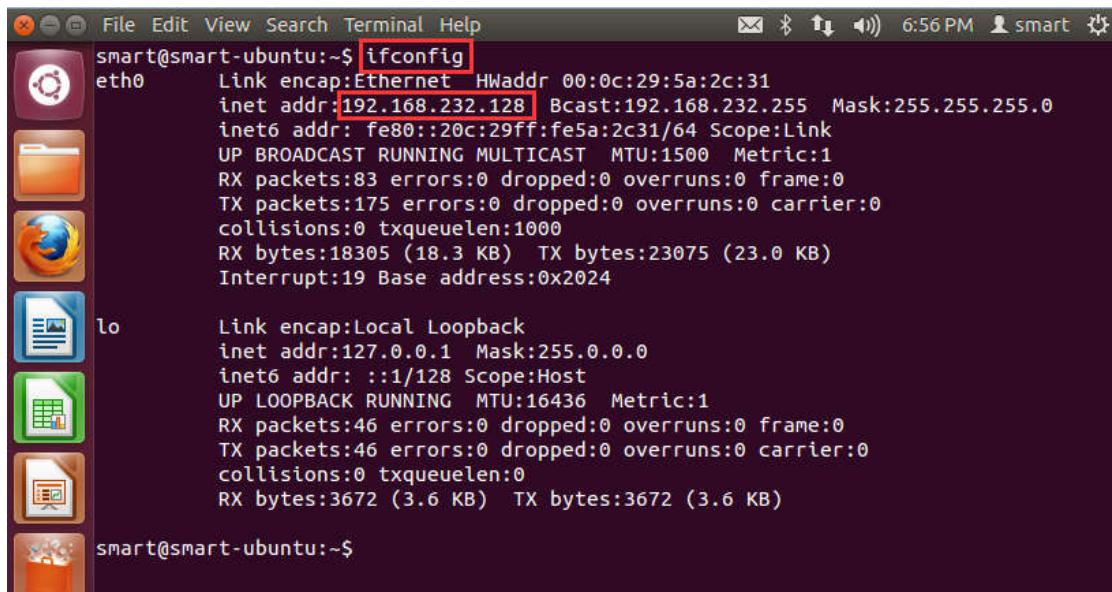


如果是第一次打开这个 Ubuntu 系统会有如下提示, 在弹出的窗口中选择 我已复制该虚拟机 (P) , 如果还有其它提示关闭窗口忽略即可





在终端用命令查看 ubuntu 系统的 IP 地址
在终端输入命令: ifconfig



```
smart@smart-ubuntu:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:5a:2c:31
          inet addr:192.168.232.128  Bcast:192.168.232.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe5a:2c31/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:83 errors:0 dropped:0 overruns:0 frame:0
          TX packets:175 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:18305 (18.3 KB)  TX bytes:23075 (23.0 KB)
          Interrupt:19 Base address:0x2024

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:46 errors:0 dropped:0 overruns:0 frame:0
          TX packets:46 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:3672 (3.6 KB)  TX bytes:3672 (3.6 KB)

smart@smart-ubuntu:~$
```

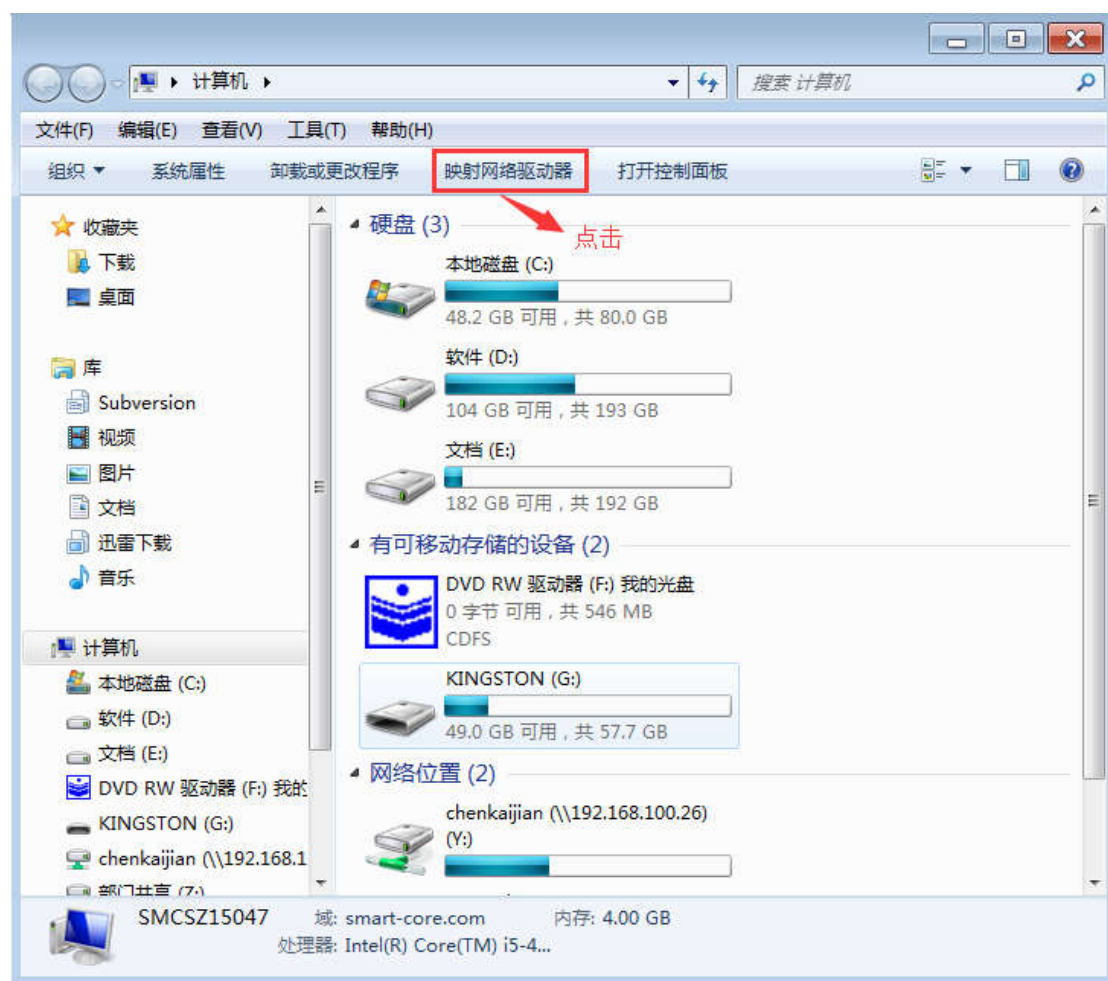
从上面的命令可以看出 ubuntu 系统的 IP 地址是：192.168.232.128

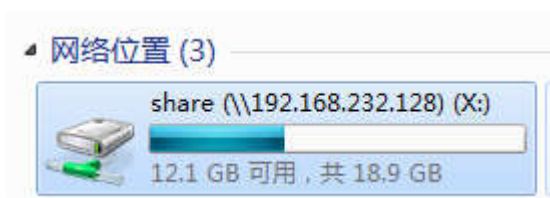
注意：这个 IP 地址在步骤 4 中要用到，先记录下来

4、建立 ubuntu 系统和 window 共享目录

建立共享目录的目的是为了可以方便的将要编译的工程从 Windows 系统拷贝到 ubuntu 系统中编译

在 windows 映射一个磁盘用来和 ubuntu 系统共享文件





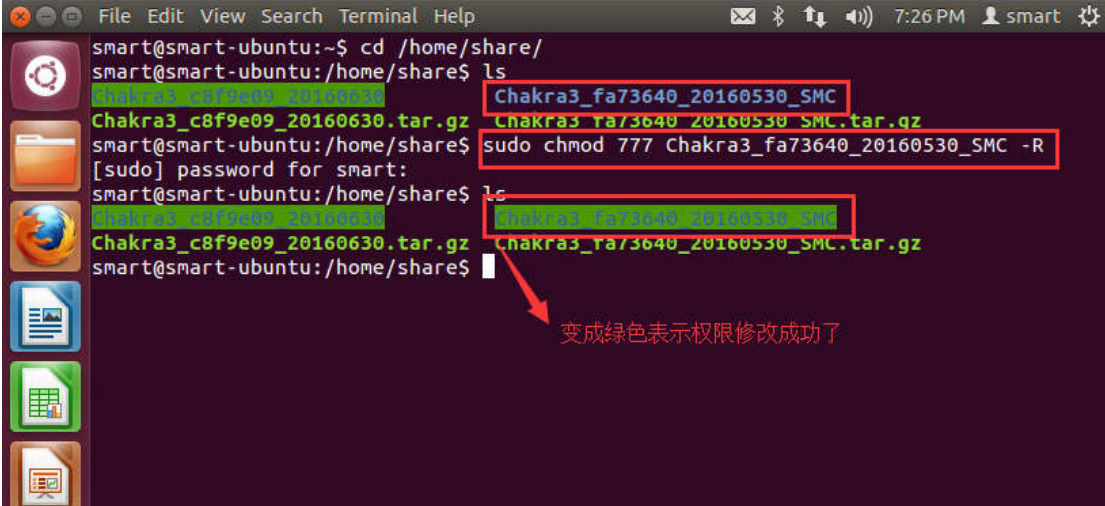
这时如果你的电脑上就会多了一个磁盘

这就表示共享已经设置成功了。这时在 ubuntu 系统的 `/home/share` 目录下你就可以看到你在 Windows 下拷贝到共享磁盘的文件了。

5、出现在 Windows 系统下不能编辑共享目录的文件解决办法

出现这种情况是因为权限不够导致无法修改文件, 为了避免在 Windows 下不能修改文件的麻烦, 可以将你的工程目录把权限改到最高。

```
sudo chmod 777 /工程目录 -R
```

A terminal window on a smart-ubuntu system. The user is in the /home/share directory. They list files, showing two tar.gz files and one SMC directory. They then run 'sudo chmod 777 Chakra3_fa73640_20160530_SMC -R'. After a password prompt, they list the files again. The SMC directory and its contents are now green, indicating successful permission changes. A red arrow points to the green text with the label '变成绿色表示权限修改成功了' (Turned green indicates permission modification was successful).

```
smart@smart-ubuntu:~$ cd /home/share/
smart@smart-ubuntu:/home/share$ ls
Chakra3_c8f9e09_20160630 Chakra3_fa73640_20160530_SMC
Chakra3_c8f9e09_20160630.tar.gz Chakra3_fa73640_20160530_SMC.tar.gz
smart@smart-ubuntu:/home/share$ sudo chmod 777 Chakra3_fa73640_20160530_SMC -R
[sudo] password for smart:
smart@smart-ubuntu:/home/share$ ls
Chakra3_c8f9e09_20160630 Chakra3_fa73640_20160530_SMC
Chakra3_c8f9e09_20160630.tar.gz Chakra3_fa73640_20160530_SMC.tar.gz
smart@smart-ubuntu:/home/share$
```

二、Non-OS 编译环境搭建

1、安装 Ubuntu 系统

注意：选择安装的 Ubuntu 系统必须是 32 位，不能用 64 位的 Ubuntu 系统

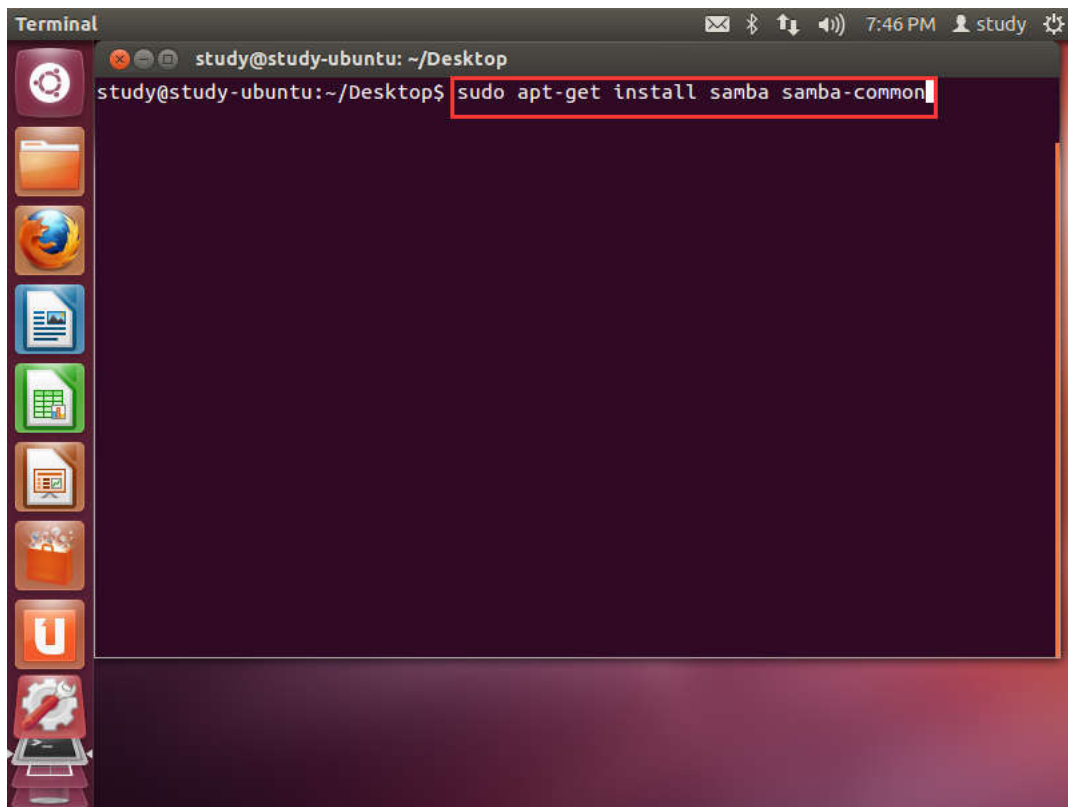
具体 Ubuntu 系统的安装方法在此就不做一一介绍，可自行在网上查找安装方法安装即可。如果你电脑上已经有安装好的 Ubuntu 系统，请务必确认下你的 Ubuntu 系统是 32 位还是 64 位，因为我司提供的交叉编译环境是在 32 位系统下运行的，因此 64 位的系统不适用。

2、在 Ubuntu 下安装 Samba 共享服务

安装完 Ubuntu 后要想能够和 Windows 系统方便的拷贝文件，这时你需要在 Ubuntu 系统下安装一个 Samba 共享服务，下面介绍一个简单的在 windows 不要账号密码就能实现文件共享的方法：

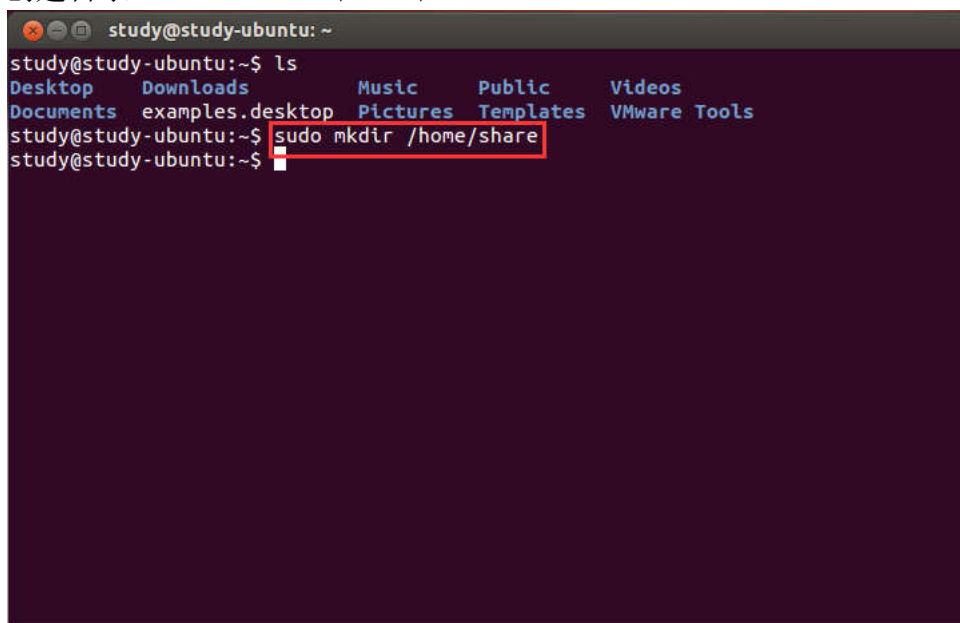
（1）用命令安装 Samba 服务

在终端输入命令：`sudo apt-get install samba samba-common`
这时系统就会自动安装 Samba 共享服务了，不过前提是要在你的虚拟机能上网的情况下



(2) 在 Ubuntu 下创建一个和 Windows 共享目录

创建目录: `sudo mkdir /home/share`



把创建的目录权限修改为 777

修改权限: `sudo chmod 777 /home/share`

```
study@study-ubuntu: ~  
study@study-ubuntu:~$ ls  
Desktop  Downloads  Music  Public  Videos  
Documents examples.desktop Pictures Templates VMWare Tools  
study@study-ubuntu:~$ sudo mkdir /home/share  
study@study-ubuntu:~$ sudo chmod 777 /home/share/
```

（3）修改 Samba 配置文件 smb.conf

建议在修改配置文件之前先做备份，以备不时之需

备份配置文件：sudo cp /etc/samba/smb.conf /etc/samba/smb.conf.bak

在终端输入命令：sudo vi /etc/samba/smb.conf

```
study@study-ubuntu: ~/Desktop  
study@study-ubuntu:~/Desktop$ sudo vi /etc/samba/smb.conf
```

在文件的最后面加上如下配置：

[share]

```
comment      = share with windows  
path          = /home/share  
writable      = yes  
browseable   = yes
```



```
guest ok = yes
```

```
study@study-ubuntu: ~
; locking = no
; path = /cdrom
; guest ok = yes

# The next two parameters show how to auto-mount a CD-ROM when the
# cdrom share is accessed. For this to work /etc/fstab must contain
# an entry like this:
#
# /dev/scd0 /cdrom iso9660 defaults,noauto,ro,user 0 0
#
# The CD-ROM gets unmounted automatically after the connection to the
#
# If you don't want to use auto-mounting/unmounting make sure the CD
# is mounted on /cdrom
#
; preexec = /bin/mount /cdrom
; postexec = /bin/umount /cdrom
[share]
comment = share with windows
path = /home/share
writable = yes
browseable = yes
guest ok = yes
```

注意 path 的值为刚刚创建共享目录的绝对路径（/home/share），修改完后保存退出

（4）重启 samba 服务器

在终端输入命令：sudo /etc/init.d/smbd restart

```
study@study-ubuntu: /home
study@study-ubuntu:~$ ls
Desktop  Downloads  Music  Public  Videos
Documents  examples.desktop  Pictures  Templates  VMware Tools
study@study-ubuntu:~$ sudo mkdir /home/share
study@study-ubuntu:~$ sudo chmod 777 /home/share/
study@study-ubuntu:~$ ls
Desktop  Downloads  Music  Public  Videos
Documents  examples.desktop  Pictures  Templates  VMware Tools
study@study-ubuntu:~$ cd ..
study@study-ubuntu:/home$ ls
share  study
study@study-ubuntu:/home$ sudo /etc/init.d/smbd restart
Rather than invoking init scripts through /etc/init.d, use the service(8)
utility, e.g. service smbd restart

Since the script you are attempting to invoke has been converted to an
Upstart job, you may also use the stop(8) and then start(8) utilities,
e.g. stop smbd ; start smbd. The restart(8) utility is also available.
smbd stop/waiting
smbd start/running, process 3935
study@study-ubuntu:/home$
```

这代表Samba服务已经启动

（5）在 Ubuntu 下查看 Ubuntu 系统的 IP 地址

在终端输入命令：ifconfig

```
study@study-ubuntu: /home
study@study-ubuntu: /home$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:23:a7:46
          inet addr:192.168.220.129  Bcast:192.168.220.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe23:a746/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:19976 errors:0 dropped:0 overruns:0 frame:0
          TX packets:12247 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:23939454 (23.9 MB)  TX bytes:679043 (679.0 KB)
          Interrupt:19 Base address:0x2024

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128  Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:106 errors:0 dropped:0 overruns:0 frame:0
          TX packets:106 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:8170 (8.1 KB)  TX bytes:8170 (8.1 KB)

study@study-ubuntu: /home$
```

从上面我们可以看出 IP 地址为：192.168.220.129

(5) 在 Windows 端访问共享目录

在 Windows 端映射一个盘来和虚拟机共享文件

← 映射网络驱动器

要映射的网络文件夹:

请为连接指定驱动器号，以及你要连接的文件夹:

驱动器(D): Y: ▾

文件夹(O): \\192.168.220.129\share ▾

浏览(B)...

示例: \\server\share

☒ 登录时重新连接(R)

☐ 使用其他凭据连接(C)

[连接到可用于存储文档和图片的网站。](#)

完成(F)

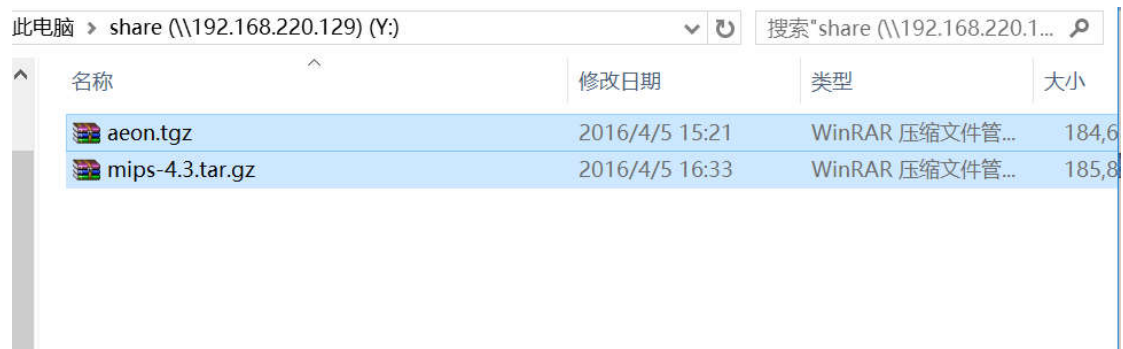
取消

如果没有其它错误提示，那么恭喜你已经实现共享了，这时你就可以从拷贝粘贴文件到这个目录了

3、Non-OS 编译环境的搭建

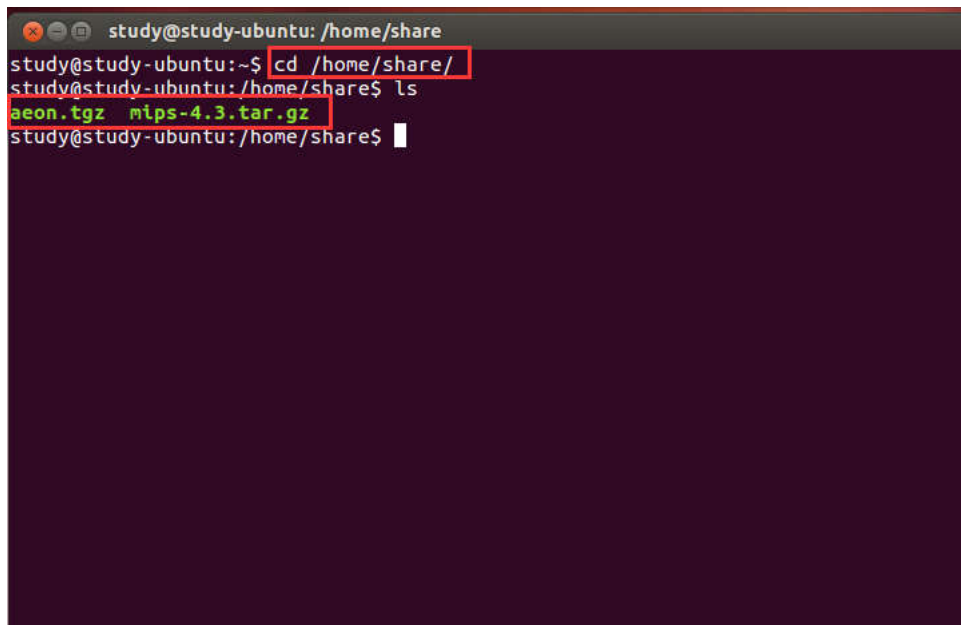
(1) 拷贝编译环境安装包到 Ubuntu 系统

从 Windows 将 aeon.tgz 和 mips-4.3.tar.gz 这两个文件拷贝到 Ubuntu 共享目录下



(2) 在 Ubuntu 下解压安装包

进入到共享目录下确认是否有这两个安装包



将安装包放到指定的目录下(指定目录可以根据自己的需要来定),下面以/home/目录为例

```
study@study-ubuntu: /home/share
study@study-ubuntu:~$ cd /home/share/
study@study-ubuntu:/home/share$ ls
aeon.tgz  mips-4.3.tar.gz
study@study-ubuntu:/home/share$ sudo mv aeon.tgz mips-4.3.tar.gz /home
```

解压 aeon.tgz 文件

在终端输入命令：sudo tar -xzvf aeon.tgz

```
study@study-ubuntu: /home
study@study-ubuntu:/home$ ls
aeon.tgz  mips-4.3.tar.gz  share  study
study@study-ubuntu:/home$ tar -xzvf aeon.tgz
```

解压完后

```
study@study-ubuntu: /home
aeon/lib/gcc/aeon/4.1.2/aeonR2f1/0s/libgcov.a
aeon/lib/gcc/aeon/4.1.2/aeonR2f1/0s/libaeonsim.a
aeon/lib/gcc/aeon/4.1.2/aeonR2f1/0s/libgcc.a
aeon/lib/gcc/aeon/4.1.2/aeonR2f1/0s/crtbegin.o
aeon/lib/gcc/aeon/4.1.2/crtbegin.o
aeon/lib/gcc/aeon/4.1.2/0s/
aeon/lib/gcc/aeon/4.1.2/0s/crti.o
aeon/lib/gcc/aeon/4.1.2/0s/crtend.o
aeon/lib/gcc/aeon/4.1.2/0s/crtn.o
aeon/lib/gcc/aeon/4.1.2/0s/libgcov.a
aeon/lib/gcc/aeon/4.1.2/0s/libaeonsim.a
aeon/lib/gcc/aeon/4.1.2/0s/libgcc.a
aeon/lib/gcc/aeon/4.1.2/0s/crtbegin.o
aeon/lib/tk8.4/
aeon/lib/tkConfig.sh
aeon/lib/libtclstub8.4.a
aeon/lib/libitcl3.2.a
aeon/lib/libtcl8.4.a
aeon/lib/tclConfig.sh
aeon/lib/insight1.0/
aeon/lib/insight1.0/plugins.tcl
study@study-ubuntu:/home$ ls
aeon  aeon.tgz  mips-4.3.tar.gz  share  study
study@study-ubuntu:/home$
```

解压出来的目录

解压 mips-4.3.tar.gz

在终端输入命令：sudo tar -xzvf mips-4.3.tar.gz

```
study@study-ubuntu: /home
study@study-ubuntu:/home$ ls
aeon  aeon.tgz  mips-4.3.tar.gz  share  study
study@study-ubuntu:/home$ sudo tar -xzvf mips-4.3.tar.gz
```

解压完后


```
study@study-ubuntu: /home
mips-4.3/bin/mips-sde-elf-objdump
mips-4.3/bin/mips-linux-gnu-ranlib
mips-4.3/bin/mips-sde-elf-strings
mips-4.3/bin/mips-linux-gnu-ar
mips-4.3/bin/mips-sde-elf-run
mips-4.3/bin/mips-linux-gnu-gcov
mips-4.3/bin/mips-sde-elf-as
mips-4.3/bin/mips-linux-gnu-gdb
mips-4.3/bin/mips-sde-elf-conv
mips-4.3/bin/mips-linux-gnu-objcopy
mips-4.3/bin/mips-linux-gnu-nm
mips-4.3/bin/mips-linux-gnu-addr2line
mips-4.3/bin/mips-sde-elf-readelf
mips-4.3/bin/mips-linux-gnu-strip
mips-4.3/bin/mips-linux-gnu-gprof
mips-4.3/bin/mips-sde-elf-gcov
mips-4.3/bin/mips-linux-gnu-ld
mips-4.3/bin/mips-linux-gnu-objdump
mips-4.3/bin/mips-linux-gnu-as
mips-4.3/bin/mips-linux-gnu-c++
mips-4.3/include/
study@study-ubuntu:/home$ ls
aeon aeon.tgz mips-4.3 mips-4.3.tar.gz share study
study@study-ubuntu:/home$
```

解压出来的目录

刚刚解压出来的两个目录下都各有一个 bin 目录，分别将这两个目录的绝对路径记下来

```
study@study-ubuntu: /home/aeon/bin
study@study-ubuntu:/home$ ls
aeon aeon.tgz mips-4.3 mips-4.3.tar.gz share study
study@study-ubuntu:/home$ cd aeon/bin/
study@study-ubuntu:/home/aeon/bin$ ls
aeon-addr2line aeon-cpp aeon-gdb aeon-objcopy aeon-size
aeon-ar aeon-g++ aeon-gdbtui aeon-objdump aeon-strings
aeon-as aeon-gcc aeon-gprof aeon-ranlib aeon-strip
aeon-c++ aeon-gcc-4.1.2 aeon-insight aeon-readelf tcsh8.4
aeon-c++filt aeon-gccbug aeon-ld aeon-sim wish8.4
aeon-conv aeon-gcov aeon-nm aeon-sim.dat
study@study-ubuntu:/home/aeon/bin$ pwd
/home/aeon/bin
study@study-ubuntu:/home/aeon/bin$
```

这个路径为aeon.tgz编译安装包的绝对路径，把这个路径记录下来，后面添加环境变量要用到


```
study@study-ubuntu: /home/mips-4.3/bin
study@study-ubuntu: /home$ ls
aeon  aeon.tgz  mips-4.3  mips-4.3.tar.gz  study
study@study-ubuntu: /home$ cd mips-4.3/bin/
study@study-ubuntu: /home/mips-4.3/bin$ ls
mips-linux-gnu-addr2line  mips-linux-gnu-objdump  mips-sde-elf-gcov
mips-linux-gnu-ar         mips-linux-gnu-ranlib   mips-sde-elf-gdb
mips-linux-gnu-as         mips-linux-gnu-readelf  mips-sde-elf-gdbtui
mips-linux-gnu-c++        mips-linux-gnu-size     mips-sde-elf-gprof
mips-linux-gnu-c++filt    mips-linux-gnu-strings  mips-sde-elf-ld
mips-linux-gnu-cpp        mips-linux-gnu-strip    mips-sde-elf-nm
mips-linux-gnu-g++        mips-sde-elf-addr2line  mips-sde-elf-objcopy
mips-linux-gnu-gcc        mips-sde-elf-ar         mips-sde-elf-objdump
mips-linux-gnu-gcc-4.3.2  mips-sde-elf-as         mips-sde-elf-ranlib
mips-linux-gnu-gcov       mips-sde-elf-c++        mips-sde-elf-readelf
mips-linux-gnu-gdb        mips-sde-elf-c++filt    mips-sde-elf-run
mips-linux-gnu-gdbtui     mips-sde-elf-conv       mips-sde-elf-size
mips-linux-gnu-gprof      mips-sde-elf-cpp        mips-sde-elf-strings
mips-linux-gnu-ld         mips-sde-elf-g++        mips-sde-elf-strip
mips-linux-gnu-lld        mips-sde-elf-gcc        mips-sde-elf-strip
mips-linux-gnu-objcopy    mips-sde-elf-gcc-4.3.2
study@study-ubuntu: /home/mips-4.3/bin$ pwd
/home/mips-4.3/bin
study@study-ubuntu: /home/mips-4.3/bin$
```

重上面两个截图我们可以得出两个编译器对应的两个 bin 路径：

`/home/aeon/bin`

`/home/mips-4.3/bin`

这两个路径是编译环境搭建的成功与否的关键，一定不要弄错了，否则后面的步骤就没有意义了

(3) 添加 Ubuntu 系统环境变量

添加 Ubuntu 系统环境变量的方法有 3 中：

- 1) 在 `/etc/profile` 文件中添加
- 2) 在 `~/.profile` 文件中添加
- 3) 在 `~/.bashrc` 文件中添加

下面以第 1 种方法添加为例，其它两种方法可按照第 1 中方法来添加

在终端输入命令打开 profile 文件：`sudo vi /etc/profile`

```
study@study-ubuntu: ~  
study@study-ubuntu:~$ sudo vi /etc/profile
```

在文件的最后面添加在步骤（2）中的两个路径

```
study@study-ubuntu: ~  
else  
    if [ "`id -u`" -eq 0 ]; then  
        PS1='# '  
    else  
        PS1='$ '  
    fi  
fi  
fi  
# The default umask is now handled by pam_umask.  
# See pam_umask(8) and /etc/login.defs.  
if [ -d /etc/profile.d ]; then  
    for i in /etc/profile.d/*.sh; do  
        if [ -r $i ]; then  
            . $i  
        fi  
    done  
    unset i  
fi  
export PATH=$PATH:/home/aeon/bin  
export PATH=$PATH:/home/mips-4.3/bin
```

将aeon和mips这两个编译器添加到环境变量中

添加完之后保存退出
重启系统

（4）检测编译环境是否搭建成功

在终端输入：aeon-
然后连续按两次 tab 键

```
study@study-ubuntu: ~  
study@study-ubuntu:~$ aeon-  
aeon-addr2line  aeon-g++      aeon-gprof      aeon-readelf  
aeon-ar         aeon-gcc      aeon-insight    aeon-sim  
aeon-as         aeon-gcc-4.1.2 aeon-ld         aeon-sim.dat  
aeon-c++       aeon-gccbug   aeon-nm        aeon-size  
aeon-c++filt   aeon-gcov    aeon-objcopy   aeon-strings  
aeon-conv      aeon-gdb     aeon-objdump   aeon-strip  
aeon-cpp       aeon-gdbtui  aeon-ranlib  
study@study-ubuntu:~$ aeon-
```

出现如上图所示，则表明 aeon 编译器安装成功

在终端输入：mips-

然后连续按两次 tab 键

```
study@study-ubuntu: ~  
study@study-ubuntu:~$ mips-  
mips-linux-gnu-addr2line mips-linux-gnu-objdump mips-sde-elf-gcov  
mips-linux-gnu-ar       mips-linux-gnu-ranlib  mips-sde-elf-gdb  
mips-linux-gnu-as       mips-linux-gnu-readelf mips-sde-elf-gdbtui  
mips-linux-gnu-c++      mips-linux-gnu-size    mips-sde-elf-gprof  
mips-linux-gnu-c++filt  mips-linux-gnu-strings mips-sde-elf-ld  
mips-linux-gnu-cpp      mips-linux-gnu-strip   mips-sde-elf-nm  
mips-linux-gnu-g++      mips-sde-elf-addr2line mips-sde-elf-objcopy  
mips-linux-gnu-gcc      mips-sde-elf-ar        mips-sde-elf-objdump  
mips-linux-gnu-gcc-4.3.2 mips-sde-elf-as        mips-sde-elf-ranlib  
mips-linux-gnu-gcov     mips-sde-elf-c++       mips-sde-elf-readelf  
mips-linux-gnu-gdb      mips-sde-elf-c++filt   mips-sde-elf-run  
mips-linux-gnu-gdbtui   mips-sde-elf-conv      mips-sde-elf-size  
mips-linux-gnu-gprof    mips-sde-elf-cpp       mips-sde-elf-strings  
mips-linux-gnu-ld       mips-sde-elf-g++       mips-sde-elf-strip  
mips-linux-gnu-nm       mips-sde-elf-gcc  
mips-linux-gnu-objcopy  mips-sde-elf-gcc-4.3.2  
study@study-ubuntu:~$ mips-
```

出现如上图所示，则表明 mips 编译器安装成功

如果上面两个编译器都安装成功，则说明编译环境搭建成功

（5）make 工程时会出现报错问题解决办法

搭建好编译环境后编译 Non-OS 工程时可能会出现以下报错

```
study@study-ubuntu: /home/share/Chakra3_fa73640_20160530_SMC
study@study-ubuntu: /home/share/Chakra3_fa73640_20160530_SMC$ ls
boot  config_by_chip.mk  core  include  project  secure
chip  config.tpl         device  Makefile  scripts  tv-ap
study@study-ubuntu: /home/share/Chakra3_fa73640_20160530_SMC$ make
[Rule - PROJ_R2_MELODY_ATV_MM_64MB]
make loader
make[1]: Entering directory `/home/share/Chakra3_fa73640_20160530_SMC'
make[1]: Nothing to be done for `loader'.
make[1]: Leaving directory `/home/share/Chakra3_fa73640_20160530_SMC'
make setup
make[1]: Entering directory `/home/share/Chakra3_fa73640_20160530_SMC'
-e \x1b[32;01m
#@ls -l core/bin/maya/SecurityR2/secu_r2.bin.128.sign
Force build SysInit.c for timestamp...
-e \x1b[0m
[SETUP START] - PROJ_R2_MELODY_ATV_MM_64MB
Tool Chain - acon-gcc
/bin/sh: 1: /bin/gawk: not found
make[1]: *** [setup] Error 1
make[1]: Leaving directory `/home/share/Chakra3_fa73640_20160530_SMC'
make: *** [R2_MELODY_ATV_MM_64MB] Error 2
study@study-ubuntu: /home/share/Chakra3_fa73640_20160530_SMC$
```

出现这个报错导致不能继续编译下去

注意：出现上面报错不能工程代码的问题，是当前 Ubuntu 在/bin/目录下没有 gawk 这个命令

问题解决办法：

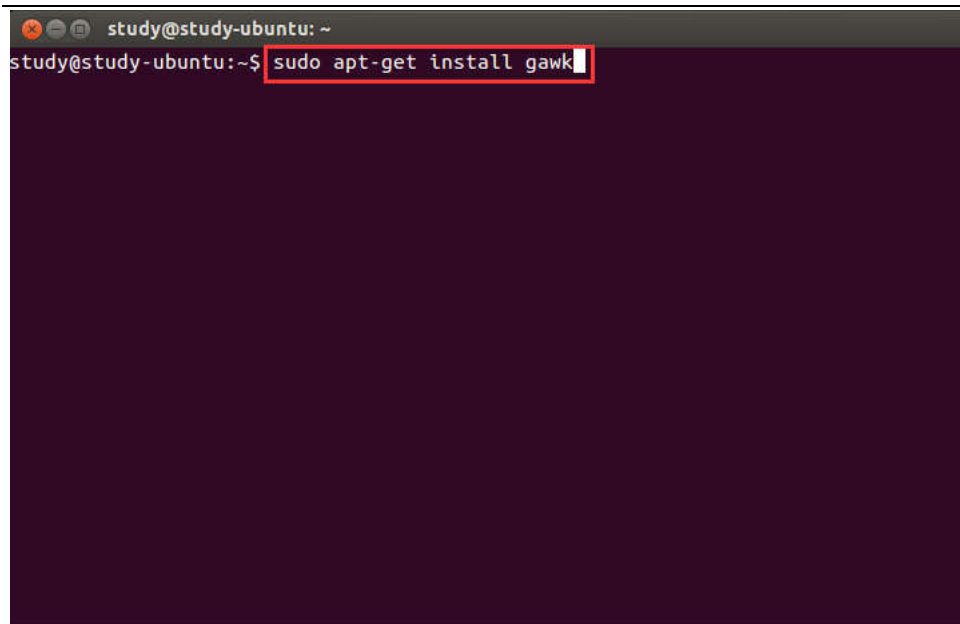
- 1) 确认 Ubuntu 是否安装 gawk

用 type gawk 确认是否安装 gawk，如果安装了则显示 gawk 所在相应的路径

```
study@study-ubuntu: /home/share/Chakra3_fa73640_20160530_SMC
study@study-ubuntu: /home/share/Chakra3_fa73640_20160530_SMC$ type gawk
bash: type: gawk: not found
study@study-ubuntu: /home/share/Chakra3_fa73640_20160530_SMC$
```

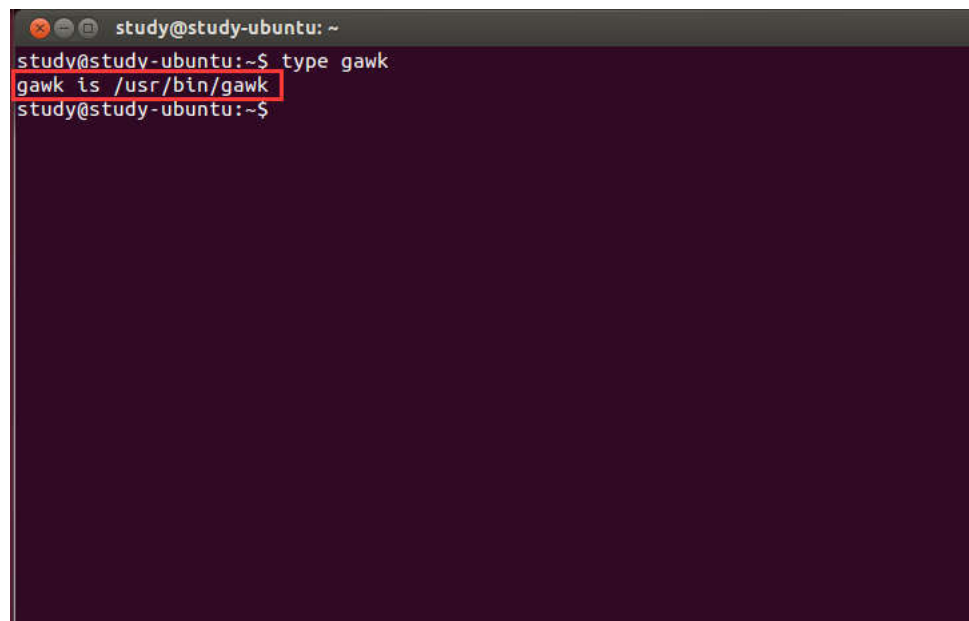
出现这个提示说明当前系统中没装 gawk

- 2) 如果 Ubuntu 下没有安装 gawk 则可通过下面的命令安装
sudo apt-get install gawk



```
study@study-ubuntu: ~  
study@study-ubuntu:~$ sudo apt-get install gawk
```

3) 安装完后在终端输入: type gawk



```
study@study-ubuntu: ~  
study@study-ubuntu:~$ type gawk  
gawk is /usr/bin/gawk  
study@study-ubuntu:~$
```

出现上面的提示说明 gawk 安装成功了，这是我们只需要将/usr/bin/gawk 拷贝到/bin/目录下就 OK 了


```
study@study-ubuntu: ~  
study@study-ubuntu:~$ type gawk  
gawk is /usr/bin/gawk  
study@study-ubuntu:~$ sudo cp /usr/bin/gawk /bin/
```

现在重新 make 以下工程就不会有刚才的报错了

```
study@study-ubuntu: /home/share/Chakra3_fa73640_20160530_SMC  
study@study-ubuntu:/home/share/Chakra3_fa73640_20160530_SMC$ make  
[Rule - PROJ_R2_MELODY_ATV_MM_64MB]  
make loader  
make[1]: Entering directory `/home/share/Chakra3_fa73640_20160530_SMC'  
make[1]: Nothing to be done for `loader'.  
make[1]: Leaving directory `/home/share/Chakra3_fa73640_20160530_SMC'  
make setup  
make[1]: Entering directory `/home/share/Chakra3_fa73640_20160530_SMC'  
-e \x1b[32;01m  
#@ls -l core/bin/maya/SecurityR2/secu_r2.bin.128.sign  
Force build SysInit.c for timestamp...  
-e \x1b[0m  
[SETUP START] - PROJ_R2_MELODY_ATV_MM_64MB  
Tool Chain = aeon-gcc  
GNU Awk 3.1.8  
rm -f ../Bin_R2_MELODY_ATV_MM_64MB/AP.elf.tmp  
rm -f  
rm -f  
cp -f ./project/loader/nos_aeon_default.ld ../Bin_R2_MELODY_ATV_MM_64MB/loader_a  
eon_R2_MELODY_ATV_MM_64MB.ld; chmod 744 ../Bin_R2_MELODY_ATV_MM_64MB/loader_aeon  
_R2_MELODY_ATV_MM_64MB.ld  
[SETUP END]  
make[1]: Leaving directory `/home/share/Chakra3_fa73640_20160530_SMC'  
make sbboot  
make[1]: Entering directory `/home/share/Chakra3_fa73640_20160530_SMC'  
[CC] core/api/utl/NoOS.c  
[CC] core/driver/power/drvPower_std.c  
[CC] core/driver/pq/drvPQ.c
```

到此整个编译环境就已经安装完了，非常的不容易啊。。。。。