
一、引言	2
二、开发环境快速搭建	3
2.1 windows 实验环境搭建.....	3
2.2 Linux 实验环境搭建.....	10
2.3 Mac 实验环境搭建	12
附录	20
2.1 实验环境详细搭建	20
2.2 使用 xshell 访问虚拟机.....	23
2.3 MIPS 相关知识.....	26

一、引言

操作系统是计算机系统的重要系统软件，也是计算机专业教学中的一个重要内容。该门课程内容复杂且抽象，我们认为掌握操作系统原理的最好方法就是自己编写一个操作系统，因此我们更希望的是同学们能够通过自己的努力去实现一个自己的操作系统，并在实现的过程中不断加深对操作系统的理解，而不是让大家只停留在理论课上的“纸上谈兵”。

通过该实验课，大家将由浅入深，从操作系统的引导到操作系统复杂的内部实现，一步一步，深刻理解操作系统中进程的管理、中断的处理、内存的管理、文件系统等相关知识，并将其实现出来。我们衷心希望每个选修完这门课的同学以后都能自豪的说：“我自己实现过一个完整的操作系统”。

最后，希望大家如果在实验的过程中发现有什么问题，请积极和老师、助教反映，提出自己的意见。对于学有余力的同学，非常欢迎大家能够加入我们，一起将国科大的操作系统实验课做的更好。接下来，我们将从环境搭建开始，一步一步实现我们的操作系统 UCAS-OS 吧！

二、开发环境快速搭建

为了让大家快速完成开发环境的搭建，我们已经将开发所需的环境集成到我们给所给的 VirtualBox 虚拟机镜像中，同学们只需要安装完成 VirtualBox 后导入我们所给的镜像，并简单的配置一下即可。当然，我们也在附录中给出了环境的具体流程，有兴趣的同学可以了解一下。

环境搭建所需的工具，我们已经拷贝到发给大家的 SD 卡中，此外我们将工具也上传到了百度云中，**链接为：<https://pan.baidu.com/share/init?surl=8KCrwEhsF1VuSFxCnHaPvg>，密码是：v5rb。**

2.1 windows 实验环境搭建

2.1.1 所需文件

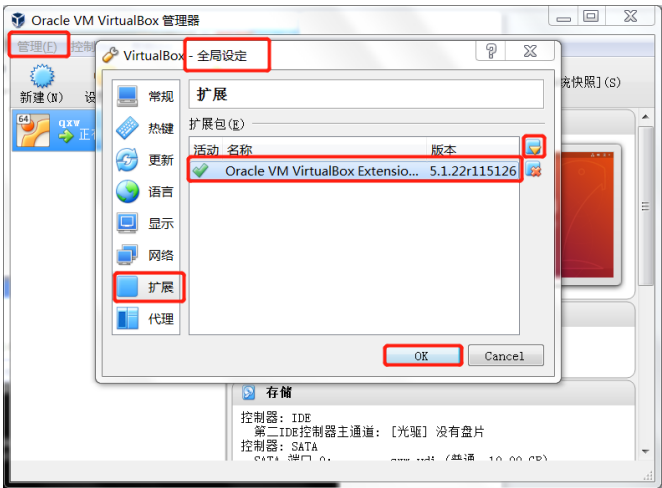
	文件名称	说明	对应网盘目录
1	VirtualBox-5.2.18-124319-Win.exe	VirtualBox windows 安装包	Windows_virtualbox
2	PL2303_Prolific_DriverInstaller_v110.exe	Windows7 串口驱动	windows 串口驱动
3	ubuntu_server_12.04.ova	VirtualBox 虚拟机镜像	已有交叉编译环境的镜像
4	Oracle_VM_VirtualBox_Extension_Pack-5.2.18.vbox-extpack	VirtualBox 拓展包	Windows_virtualbox

2.1.2 VirtualBox 软件的安装

(1) 右键点击安装包，选中以管理员方式运行安装包，完成快速安装。



(2) 安装 VirtualBox 扩展包。运行 VirtualBox->管理->全局设定->扩展->选择扩展包目录->安装->重启。

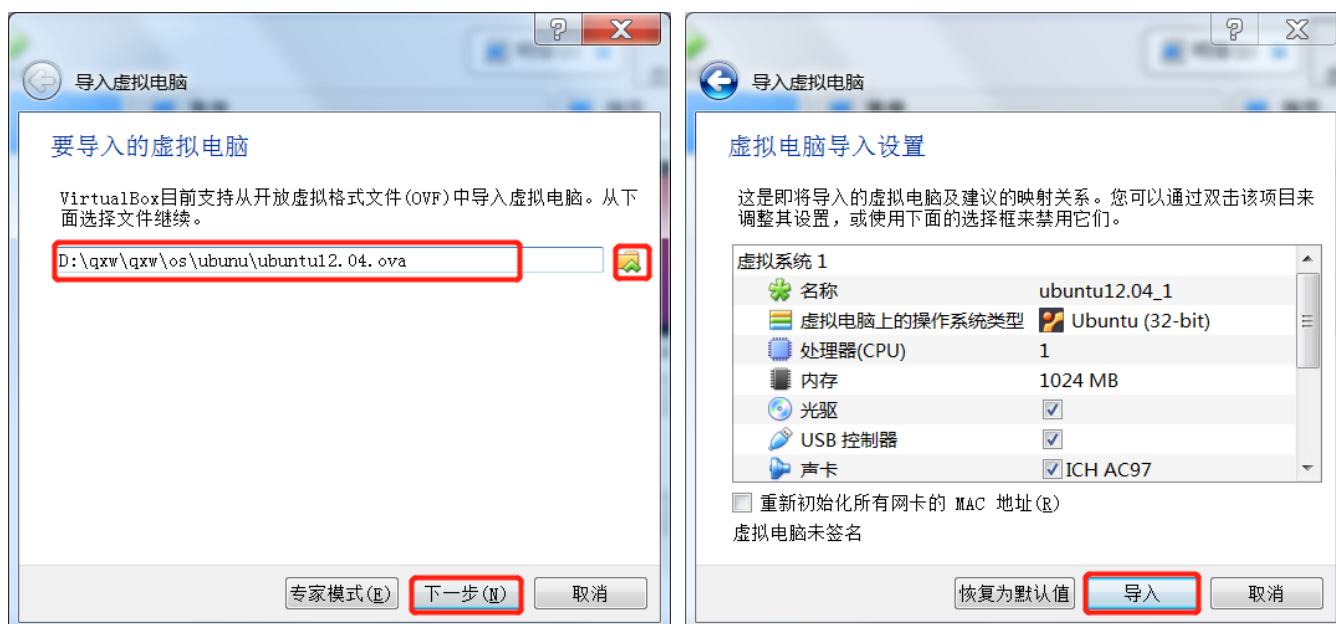


2.1.2 虚拟机镜像的导入

(1) 打开 VirtualBox 软件，点击管理选择导入虚拟电脑。



(2) 选择要导入的镜像文件 ubuntu12.04.ova，点击下一步，进行导入。



导入完成后，我们就可以登入已经配置好交叉编译环境的虚拟电脑了，**注意，用户名是 stu，密码是 123456！**

```
Ubuntu 12.04.4 LTS jinxu-VirtualBox tty1
jinxu-VirtualBox login: stu
Password:
Last login: Tue Sep 11 17:26:21 CST 2018 on tty1
Welcome to Ubuntu 12.04.4 LTS (GNU/Linux 3.11.0-15-generic i686)

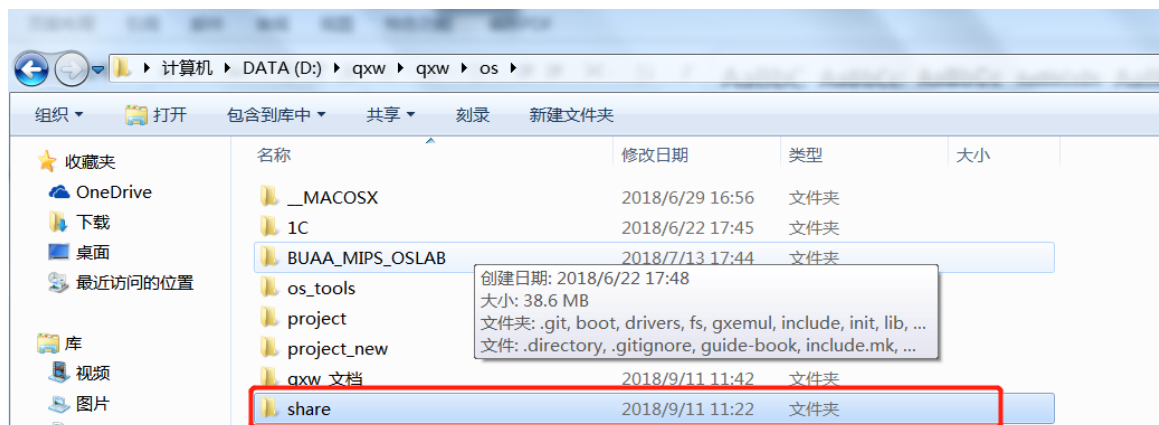
* Documentation:  https://help.ubuntu.com/

New release '14.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

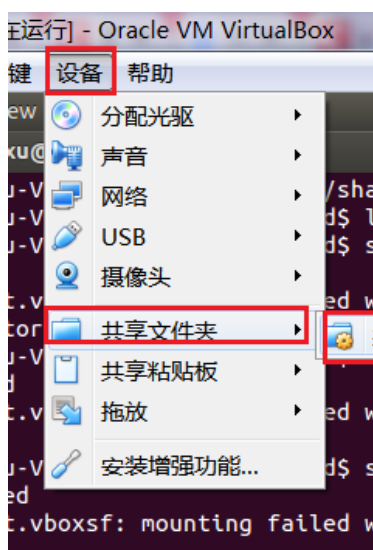
stu@jinxu-VirtualBox:~$ aa
```


导入完虚拟机后，我们需要配置主机和虚拟机的共享文件夹，方便虚拟机和主机的文件传输。

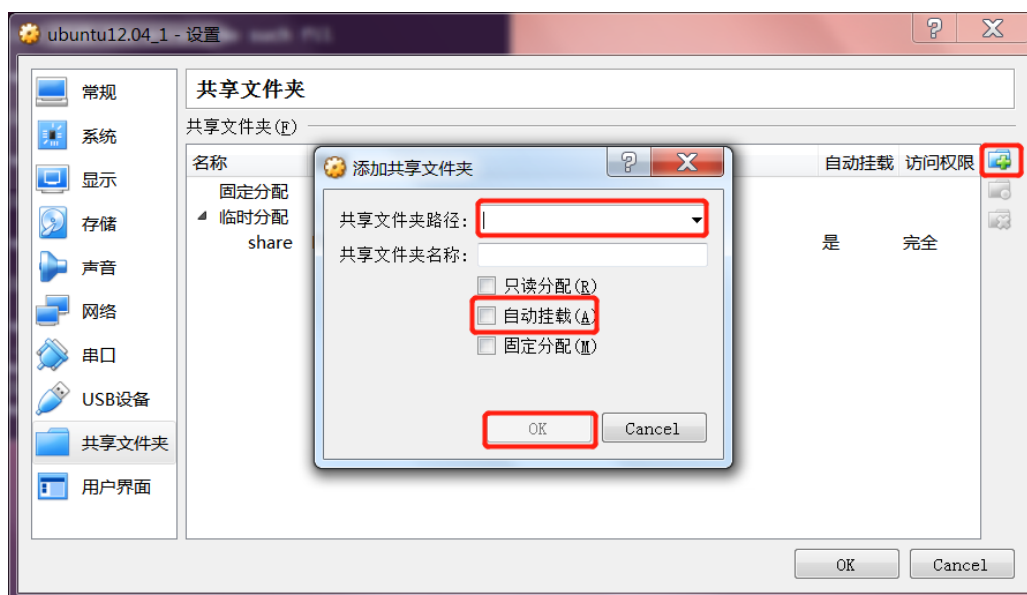
(3) 首先，在主机中找一个位置创建 share 文件夹，如图示例：



点击 VirtualBox 设备->共享文件夹->共享文件夹，如图所示



点击 ，在共享文件夹路径输入创建好的 share 文件夹路径，选择自动挂载，点击 ok。



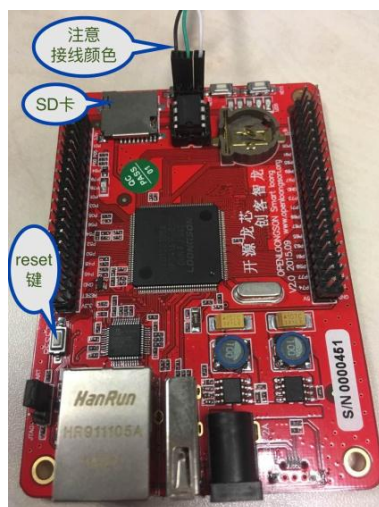
(4) 在虚拟机 terminal 中输入命令：`sudo mount -t vboxsf share /mnt/shared`。注意，每次虚拟机开机后都需要输入此

命令! 之后我们就可以在主机中的 share 文件夹中放入需要编译的文件, 在虚拟机 terminal 中输入 `cd /mnt/shared` 就可以进入我们共享的文件夹, 在 terminal 输入 `ls`, 就可以列出共享文件夹中的文件了。

```
minicom: cannot open /dev/ttyUSB0: No such file or directory
stu@stu-VirtualBox:~$ sudo mount -t vboxsf share /mnt/shared
sudo: unable to resolve host stu-VirtualBox
stu@stu-VirtualBox:~$ sudo mount -t vboxsf share /mnt/shared
sudo: unable to resolve host stu-VirtualBox
[sudo] password for stu:
stu@stu-VirtualBox:~$ cd /mnt/shared/finished_code1/
stu@stu-VirtualBox:/mnt/shared/finished_code1$ aaaaa_
```

2.1.3 开发板的连接

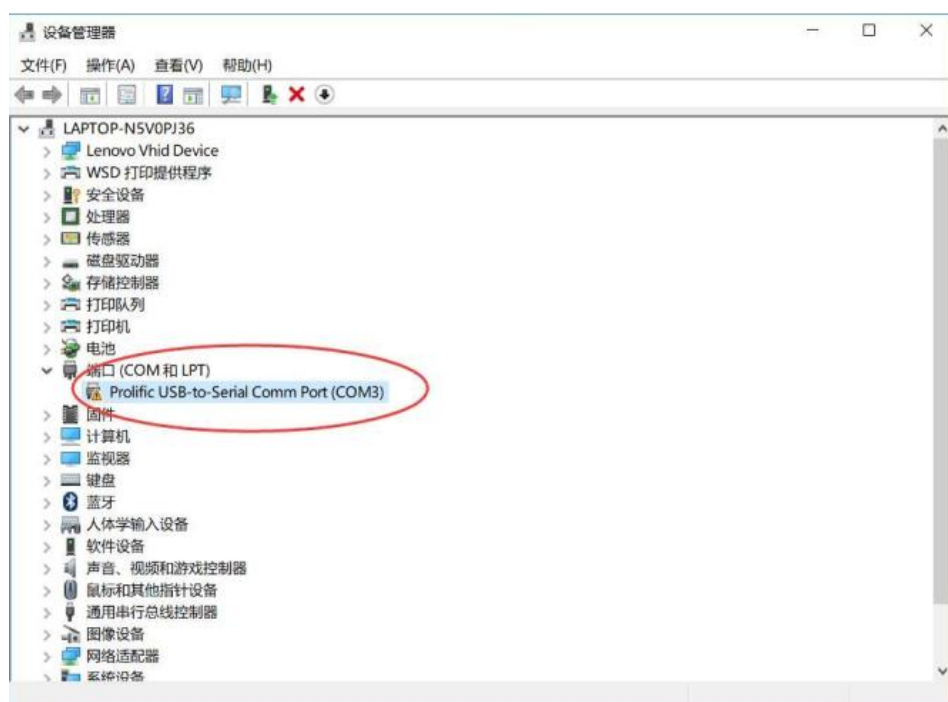
安装开发板串口驱动, 将串口线顺着 SD 卡的方向以黑、绿、白的顺序插入开发板, 同时将串口线 USB 端插入电脑。看到开发板上的两盏 LED 灯亮起说明开发板正常工作。



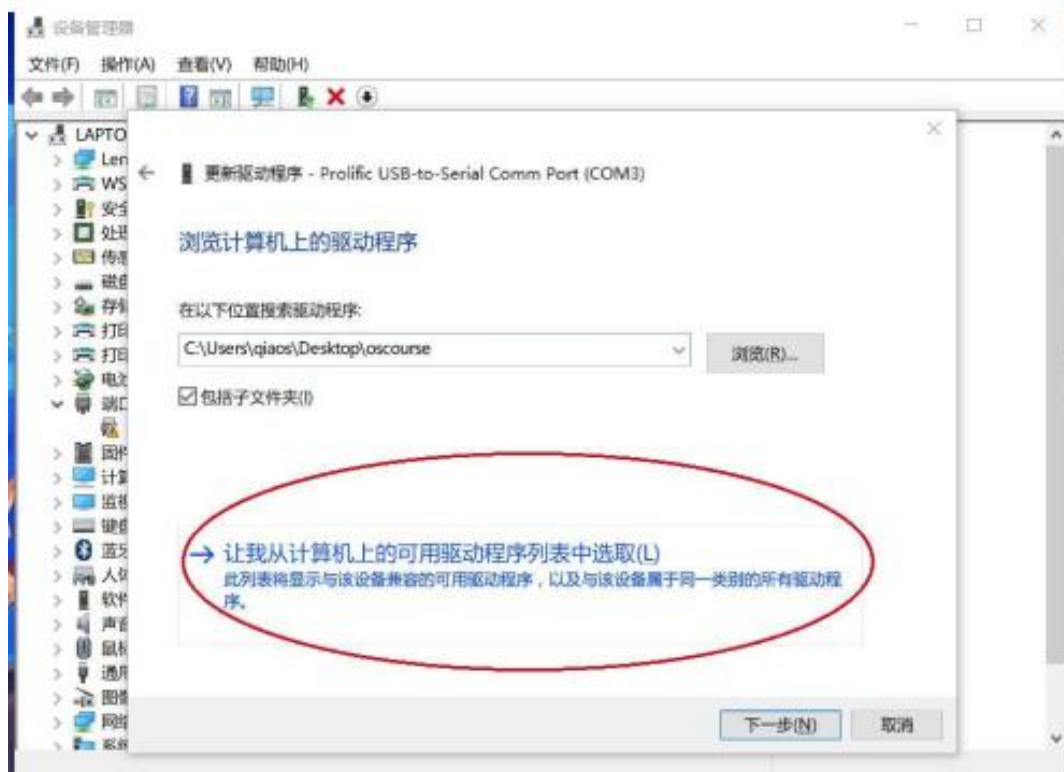
2.1.4 串口配置

Windows10 的驱动安装流程:

(1) 打开 windows 设备管理器, 下图黄色叹号表示设备异常, 驱动未(正确)安装。



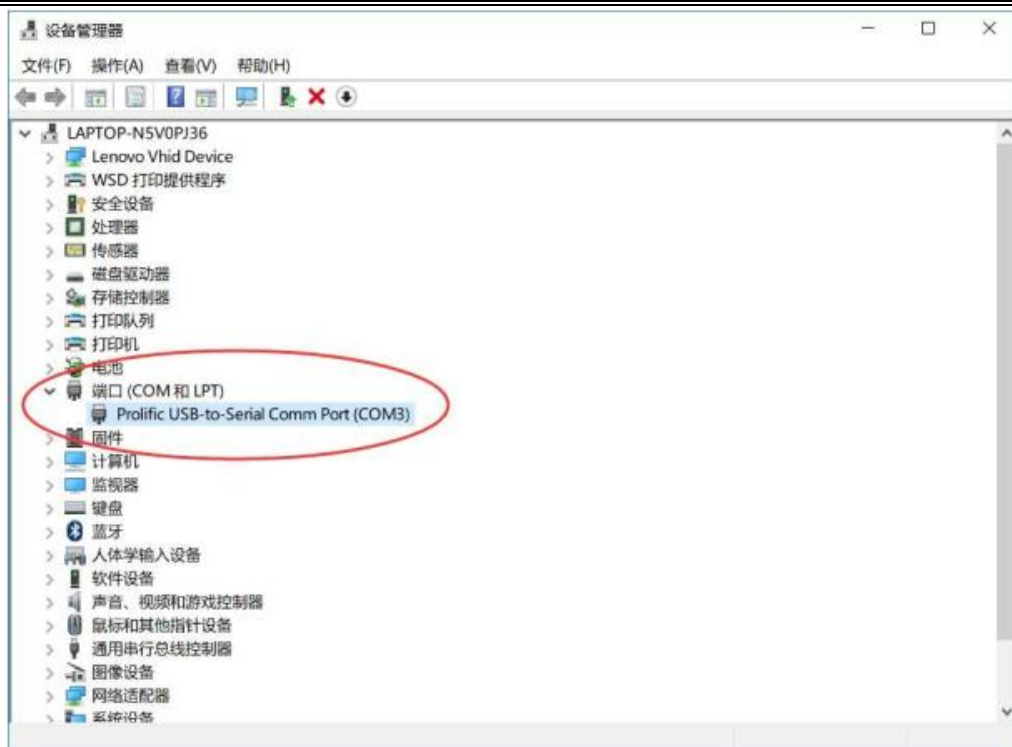
(2) 右键选中串口设备，点击更新驱动程序



(3) 选择 2009/11/19 的驱动安装。

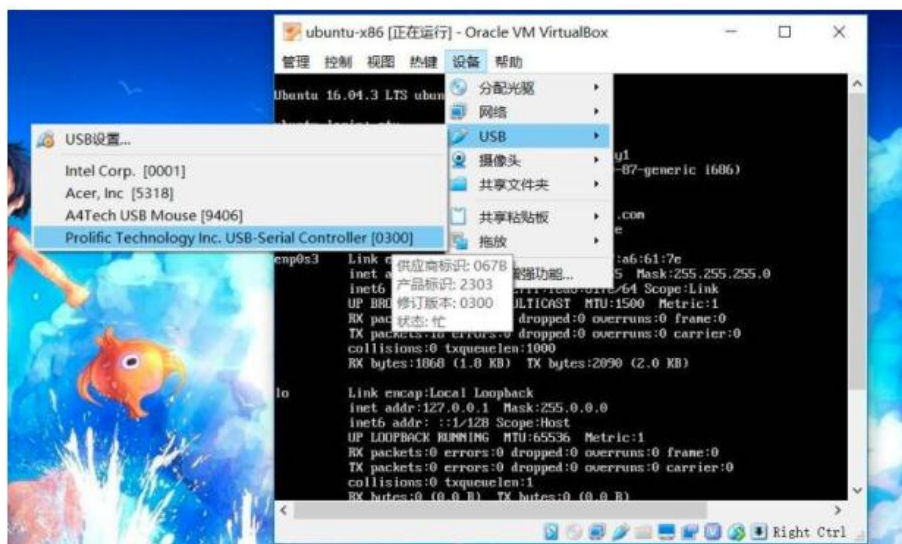


(4) 安装完成之后，开发板正确工作，黄色叹号消失。

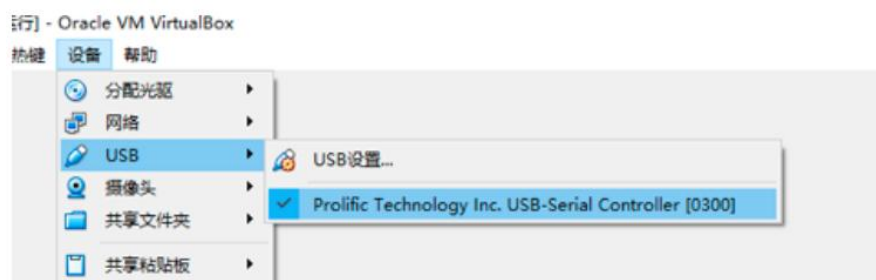


Windows7 的安装:

(5) 使用 PL2303_Prolific_DriverInstaller_v110.exe 安装串口驱动。将开发板串口线连接至电脑，同时检查驱动是否正常工作。打开 virtualbox，并打开虚拟机，并登陆。在虚拟机菜单的设备选项-->USB 中选择 USB-Serial Controller。



成功连接之后，串口设备前会出现选中标识。



(6) 在虚拟机的命令行中输入 `sudo minicom -s` 命令便进入了 minicom 的配置界面。

```
+-----[configuration]-----+
| Filenames and paths          |
| File transfer protocols      |
| Serial port setup            |
| Modem and dialing            |
| Screen and keyboard          |
| Save setup as dfl             |
| Save setup as..              |
| Exit                         |
| Exit from Minicom            |
+-----+

```

(7) 进入到 Serial port setup 选项，修改 Serial Device 为 `/dev/ttyUSB0`，完成按回车返回主菜单即可。

```
+-----+
| A - Serial Device      : /dev/ttyUSB0 |
| B - Lockfile Location  : /var/lock    |
| C - Callin Program     :              |
| D - Callout Program    :              |
| E - Bps/Par/Bits       : 115200 8N1   |
| F - Hardware Flow Control : Yes       |
| G - Software Flow Control : No        |
|                               |
| Change which setting?          |
+-----+
| Screen and keyboard          |
| Save setup as dfl             |
| Save setup as..              |
| Exit                         |
| Exit from Minicom            |
+-----+

```

```
+-----[configuration]-----+
| Filenames and paths          |
| File transfer protocols      |
| Serial port setup            |
| Modem and dialing            |
| Screen and keyboard          |
| Save setup as dfl             |
| Save setup as..              |
| Exit                         |
| Exit from Minicom            |
+-----+

```

```
+-----[configuration]-----+
| Filenames and paths          |
| File transfer protocols      |
| Serial port setup            |
| Modem and dialing            |
| Screen and keyboard          |
| Save setup as dfl             |
| Save setup as..              |
| Exit                         |
| Exit from Minicom            |
+-----+

```

(9) 返回主菜单后，选择“Save setup as dfl”，将其保存为默认设置，然后选择 Exit 退出。在命令行中用 root 权限重新打开 minicom: `sudo minicom`，当屏幕出现以下界面代表环境配置成功。

```
Configuration [FCR,EL,NET]
GitHashNumber: bfd91c56aede4ecffd23199fa30e583bbe4f16f9
CommitAuthor: hammer19
CommitDate: Wed Jul 12 15:07:26 2017
userIP:
UserName: zhaoyangyang
MakeTime: Tue Aug 8 11:36:24 CST 2017.
Supported loaders [srec, elf, bin]
Supported filesystems [sdcard, mtd, net, fat, fs, disk, socket, tty, ram]
This software may be redistributed under the BSD copyright.
Copyright 2000-2002, Opsycon AB, Sweden.
Copyright 2005, ICT CAS.
CPU Loongson 1C300A OpenLoongson V2.0 @ 252.00 MHz / Bus @ 126.00 MHz
Memory size 32 MB ( 32 MB Low memory, 0 MB High memory) .
Primary Instruction cache size 16kb (32 line, 4 way)
Primary Data cache size 16kb (32 line, 4 way)
BEV in SR set to zero.
PMON>

```

2.2 Linux 实验环境搭建

对于 Linux 的环境配置比较简单，我们不需要使用虚拟机，只需要简单的安装交叉编译环境即可。

2.2.1 所需文件

	文件名称	说明	对应网盘目录
1	gcc-4.3-ls232.tar.gz	交叉编译工具	交叉编译工具链

2.2.2 串口配置

关于串口的相关驱动已经附加在 Linux 内核里，不需要安装额外的驱动，只需要安装 minicom 串口调试软件即可。

步骤如下：

- 1) 安装 minicom，一般而言 Linux 是自带 minicom 的，如果没有的话，需要我们手动安装一下，直接在终端输入指令，进行安装：**sudo apt-get install minicom**
- 2) 在连接板子之前，我们需要配置一下 minicom 对应的设备文件：**sudo minicom -s**，设置 Port 为 **/dev/ttyUSB0**。

```
+-----+
| A -   Serial Device       : /dev/ttyUSB0 |
| B -   Lockfile Location   : /var/lock     |
| C -   Callin Program      :               |
| D -   Callout Program     :               |
| E -   Bps/Par/Bits        : 115200 8N1    |
| F -   Hardware Flow Control : Yes         |
| G -   Software Flow Control : No          |
+-----+
| Change which setting? |
+-----+
| Screen and keyboard |
| Save setup as dfl   |
| Save setup as..    |
| Exit                |
| Exit from Minicom  |
+-----+
```

- 3) 运行 minicom，连接龙芯开发板，如果开发板亮起 2 盏 LED 灯说明开发板连接正常（一个红色灯代表串口链接正常，一个绿色灯代表电源键供电正常），按下 restart 复位键，不出意外的话，屏幕会打印出初始化信息，并进入 PMON 界面。

```
Configuration [FCR,EL,NET]
GitHashNumber: bfd91c56aede4ecffd23199fa30e583bbe4f16f9
CommitAuthor: hammer19
CommitDate: Wed Jul 12 15:07:26 2017
userIP:
UsrName: zhaoyangyang
MakeTime: Tue Aug 8 11:36:24 CST 2017.
Supported loaders [srec, elf, bin]
Supported filesystems [sdcard, mtd, net, fat, fs, disk, socket, tty, ram]
This software may be redistributed under the BSD copyright.
Copyright 2000-2002, Opsycon AB, Sweden.
Copyright 2005, ICT CAS.
CPU Loongson 1C300A OpenLoongson V2.0 @ 252.00 MHz / Bus @ 126.00 MHz
Memory size 32 MB ( 32 MB Low memory, 0 MB High memory) .
Primary Instruction cache size 16kb (32 line, 4 way)
Primary Data cache size 16kb (32 line, 4 way)

BEV in SR set to zero.
PMON>
```

2.2.2 交叉编译环境安装

由于我们编译的程序是基于龙芯开发板的，所以需要进行交叉编译，关于交叉编译需要的工具，我们已经放到了 gcc-4.3-ls232.tar.gz 中，请根据以下步骤搭建环境：

- 1) 解压压缩包到/opt 目录下：**sudo tar zxvf gcc-4.3-ls232.tar.gz -C /opt**

- 2) 添加加环境变量：

打开文件 bashrc：**vi ~/.bashrc**

在文件末尾添加语句：**export PATH=/opt/gcc-4.3-ls232/bin:\$PATH**

3) 重新刷新配置路径: `source ~/.bashrc`

4) 安装依赖库: `sudo apt-get install zlib1g:i386`

进行完以上步骤之后, 运行命令, 如果屏幕输出相关信息, 则说明交叉编译环境配置已经成功。

```
parallels@ubuntu:~$ mipsel-linux-gcc -v
Using built-in specs.
Target: mipsel-linux
Configured with: ../gcc-4.3.0/configure --prefix=/usr --target=mipsel-linux --host=i486-pc-linux-gnu --with-sysroot=/usr --enable-__cxa_atexit --enable-c99 --enable-shared --enable-threads=posix --enable-__cxa_atexit --enable-c99 --enable-shared --enable-threads=posix --enable-__cxa_atexit --enable-c99 --enable-shared --enable-threads=posix
Thread model: posix
gcc version 4.3.0 (GCC)
```

2.3 Mac 实验环境搭建

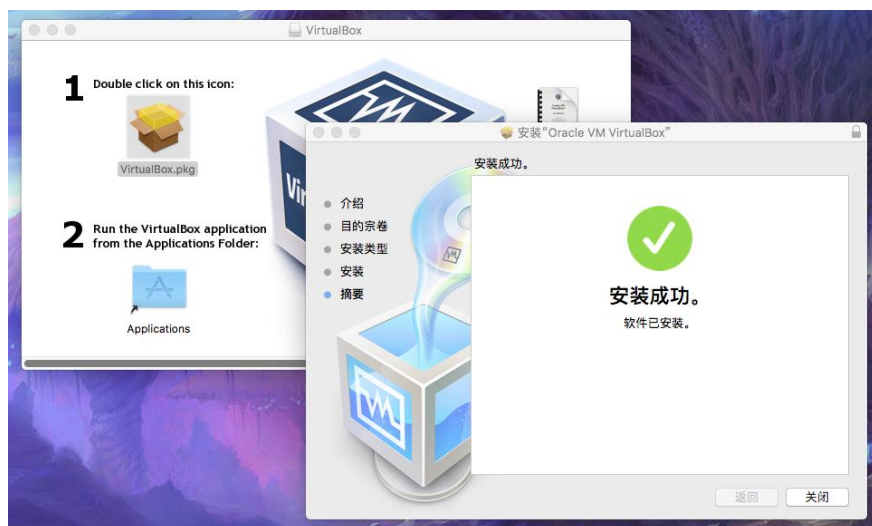
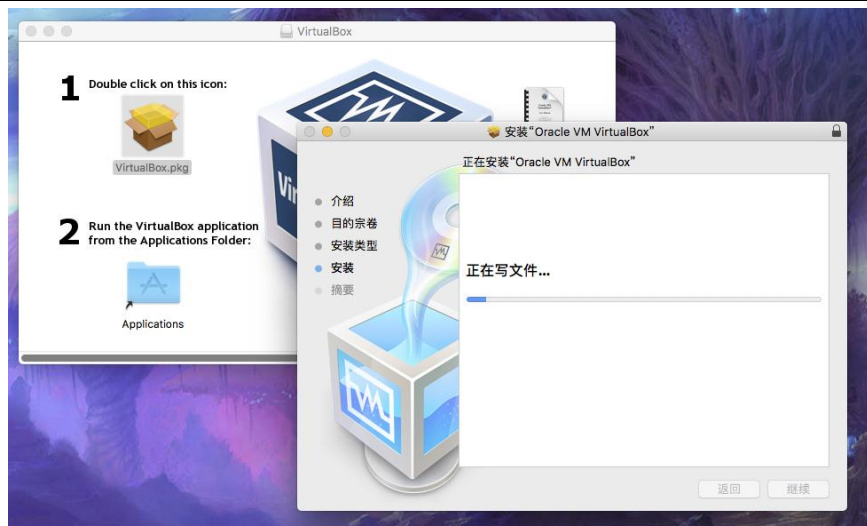
2.3.1 文件列表

	文件名称	说明	对应网盘目录
1	VirtualBox-5.2.18-124319-OSX.dmg	VirtualBox Mac 安装包	Mac_virtualbox
2	PL2303_MacOSX_1_6_1_20170620.zip	Mac 串口驱动	Mac 串口驱动
2	ubuntu_server_12.04.ova	VirtualBox 虚拟机镜像	已有交叉编译环境的镜像

2.3.2 VirtualBox 的安装

VritualBox 的安装在 Mac 上的安装同 windows 一样，只需要使用我们提供的安装包即可。



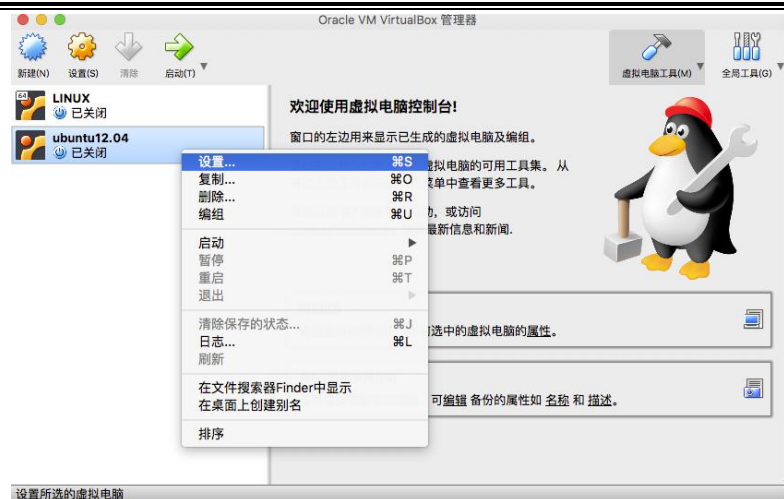


2.3.3 虚拟机镜像的导入

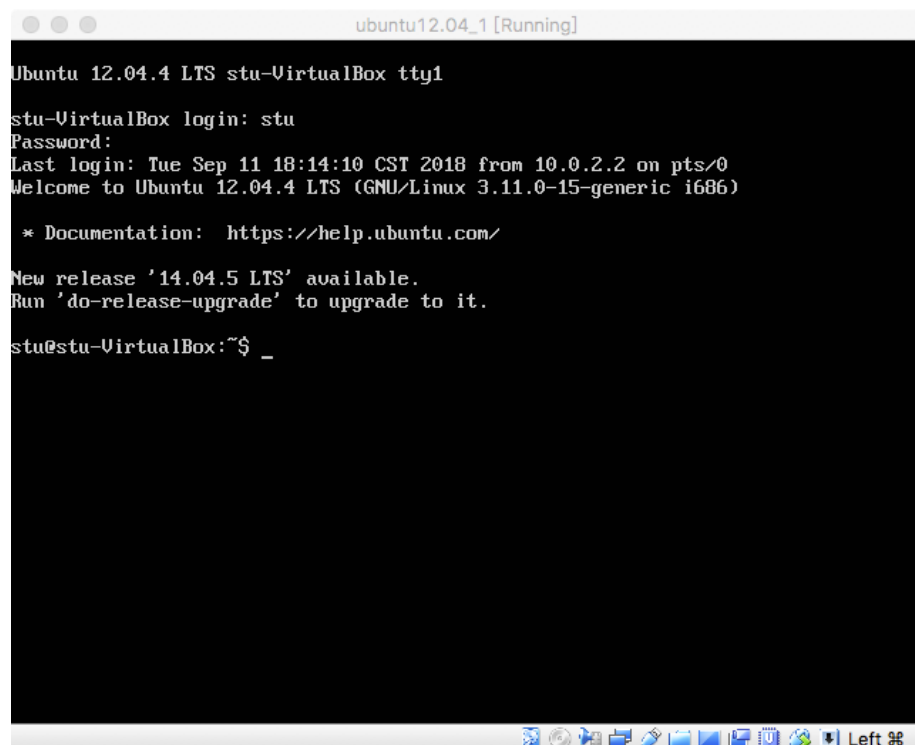
使用我们给出的已经配置完成的虚拟机镜像导入 VirtualBox 中。





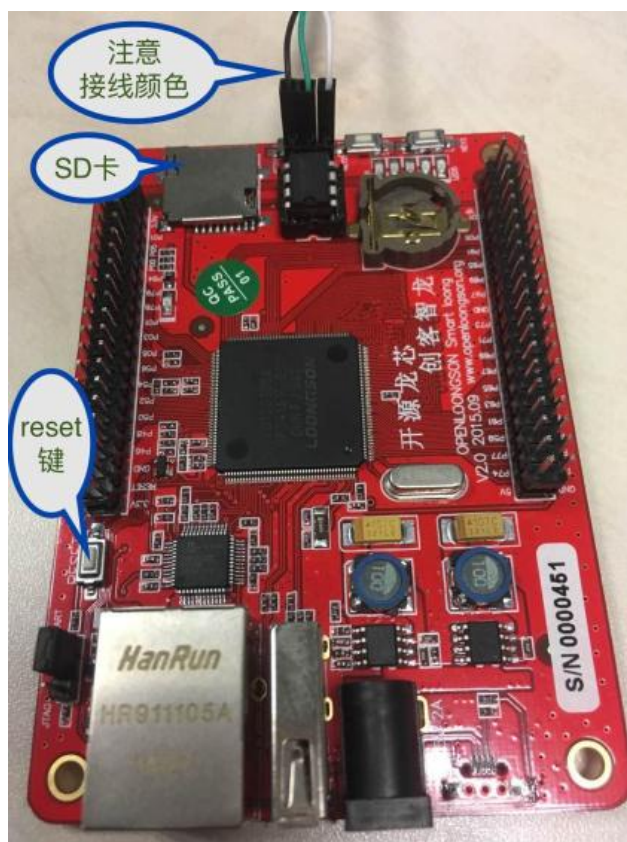


安装完成后登陆，账号为 **stu**，密码为 **123456**，如下图：



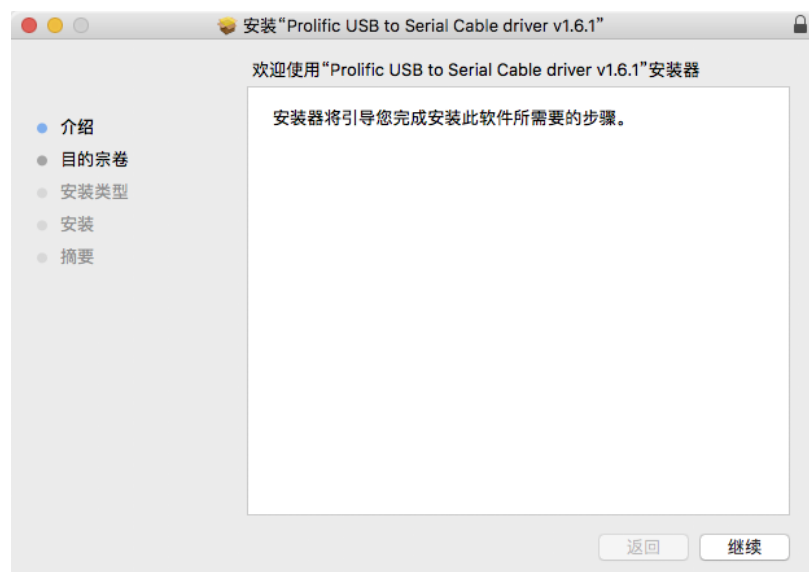
2.3.4 开发板的连接

安装开发板串口驱动，将串口线顺着 SD 卡的方向以黑、绿、白的顺序插入开发板，同时将串口线 USB 端插入电脑。看到开发板上的**两盏 LED 灯**亮起说明开发板正常工作。



2.3.5 串口驱动安装与配置

同 windows 一样，mac 也需要安装串口驱动，只需要直接双击安装驱动程序即可。安装完需要重启电脑，之后重新连接开发板。

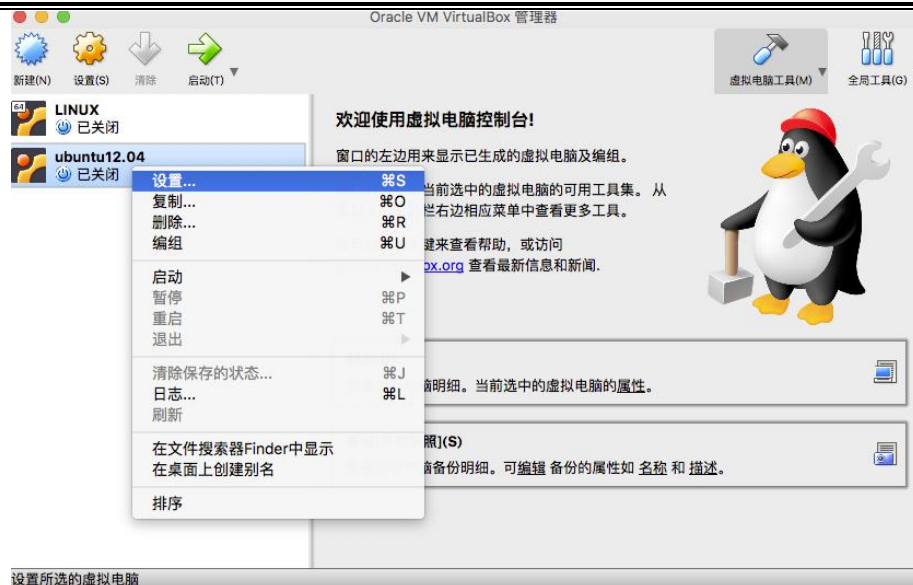




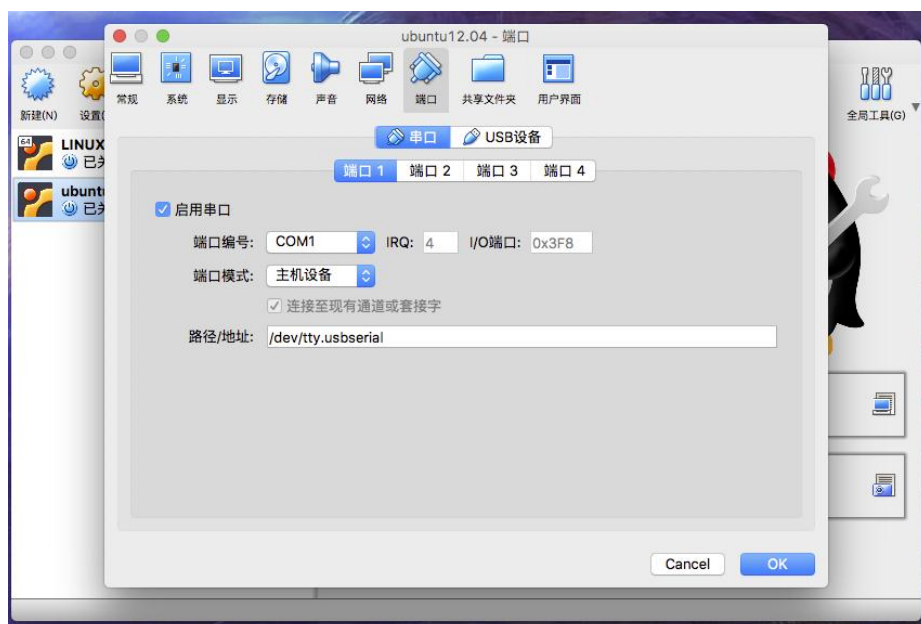
2.3.6 虚拟机串口配置

为了让虚拟机可以连接到串口设备，我们需要对虚拟机进行配置。

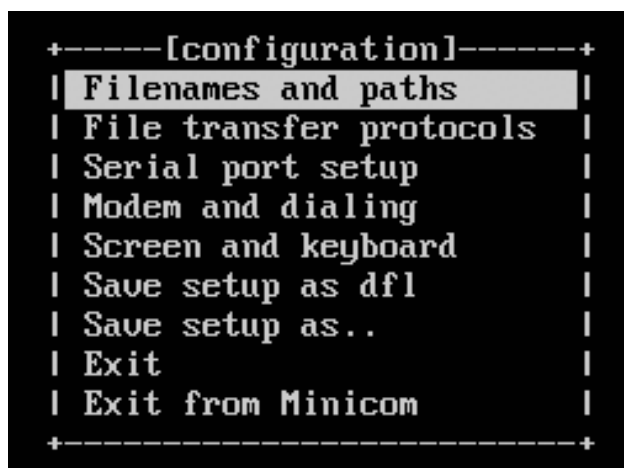
(1) 右键虚拟机，进入设置。



(2) 将串口 1 (COM1) 启用，并将端口模式设为主机设备，设置路径/地址为 `/dev/tty.usbserial`。



(3) 配置 minicom 的串口地址，输入 `sudo minicom -s` 进入到配置界面，进入 serial port setup 选项，修改 Serial Device 为 `/dev/ttyS0`。



```

ubuntu12.04_1 [Running]
+-----+
| A -   Serial Device       : /dev/tty8      |
| B - Lockfile Location    : /var/lock       |
| C -   Callin Program     :                 |
| D -   Callout Program    :                 |
| E -   Bps/Par/Bits       : 115200 8N1     |
| F - Hardware Flow Control : Yes           |
| G - Software Flow Control: No            |
|                                     |
|   Change which setting? _             |
+-----+
| Screen and keyboard |
| Save setup as dfl   |
| Save setup as..    |
| Exit                |
| Exit from Minicom   |
+-----+

```

(4) 修改 minicom 设置后，退出 minicom，重新打开 minicom: `sudo minicom`，对开发板进行复位（reset），如果出现以下界面，代表环境配置成功。

```

ubuntu12.04_1 [Running]
bootp=8000bbd8
Version = 0x1237
MacAddr = 0x0 0x55 0x7b 0xb5 0x7d 0xf7
====>enter synopGMAC_mac_init:1000
====>full duplex
====>100M
====>enter synopGMAC_mac_init:1000
====>full duplex
====>100M

Configuration [FCR,EL,NET]
GitHashNumber:
CommitAuthor:
CommitDate:
userIP: 10.0.2.15
UsrName: jinxu
MakeTime: Thu Sep 6 17:13:02 CST 2018.
Supported loaders [srec, elf, bin]
Supported filesystems [sdcard, mtd, net, fat, fs, disk, socket, tty, ram]
This software may be redistributed under the BSD copyright.
Copyright 2000-2002, Opsycon AB, Sweden.
Copyright 2005, ICT CAS.
CPU Loongson 1C300A OpenLoongson V2.0 @ 252.00 MHz / Bus @ 126.00 MHz
Memory size 32 MB ( 32 MB Low memory, 0 MB High memory) .
Primary Instruction cache size 16kb (32 line, 4 way)
Primary Data cache size 16kb (32 line, 4 way)

BEU in SR set to zero.
PMON>
CTRL-A Z for help |115200 8N1 | NOR | Minicom 2.5 | UT102 | Online 00:00

```

附录

2.1 实验环境详细搭建

在第二节我们给出了配置环境的快速搭建，在 windows 和 mac 环境下，大家只需要安装完 VirtualBox 后导入我们已经配置完成的虚拟机镜像即可，但是有些同学已经有了虚拟机或者想自己去一步一步完成环境的配置。因此我们在这一节给出更为详细的环境配置。有需要的同学可以通过这一节的内容去一步一步的配置虚拟机及交叉编译环境。当然，对于已经完成快速搭建且对环境配置不感兴趣的同学可以跳过本节内容。

搭建环境所需工具的百度云链接为：<https://pan.baidu.com/s/1d6oGfGjAiSkoFQurTj1Y5g>，密码：0q64。

2.1.1 所需文件

以下的文件大家请根据自己的系统自己选择：

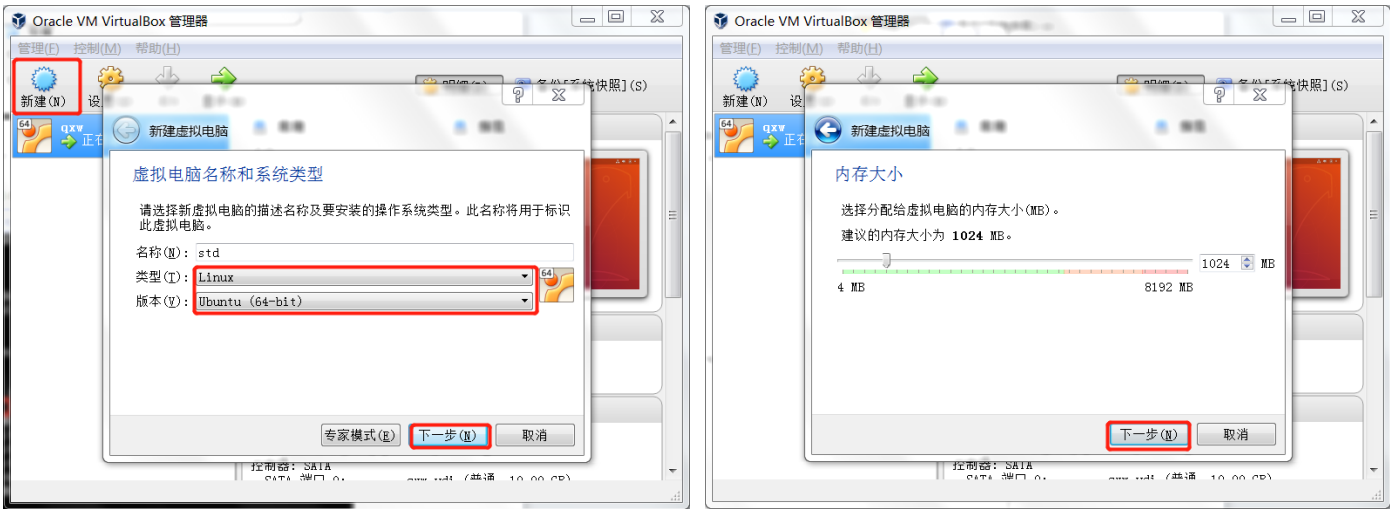
	文件名称	说明
1	VirtualBox-5.2.18-124319-OSX.dmg	VirtualBox Mac 安装包
2	VirtualBox-5.2.18-124319-Win.exe	VirtualBox Windows 安装包
3	Oracle_VM_VirtualBox_Extension_Pack-5.2.18.vbox-extpack	VirtualBox 拓展包
4	gcc-4.3-ls232.tar.gz	交叉编译工具
5	ubuntu-14.04.3-desktop-amd64.iso	Ubuntu 镜像（64bits）
6	PL2303_MacOSX_1_6_1_20170620.zip	Mac 串口驱动
7	PL2303_Prolific_DriverInstaller_v110.exe	Windows7 串口驱动

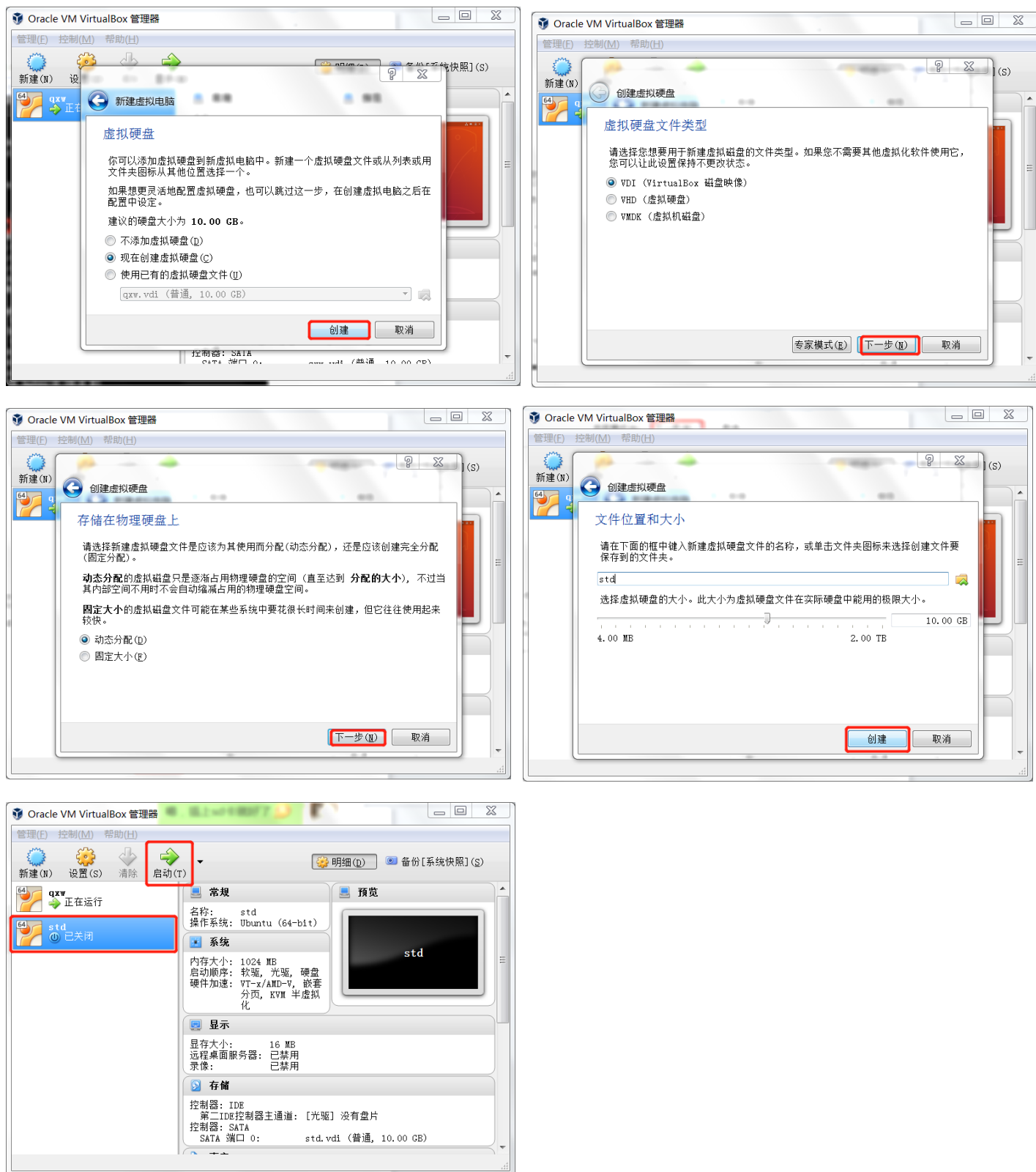
2.1.2 安装 VirtualBox 软件

在之前的小节中我们已经介绍了 windows 和 mac 下 VirtualBox 软件的安装，请大家根据自己的系统进行安装，在这里不进行赘述。

2.1.3 Ubuntu 安装

（1）打开 VirtualBox 程序，新建一个虚拟电脑。选择类型为 Linux，版本为 Ubuntu(64-bit)。如下图所示创建新建一个虚拟电脑。



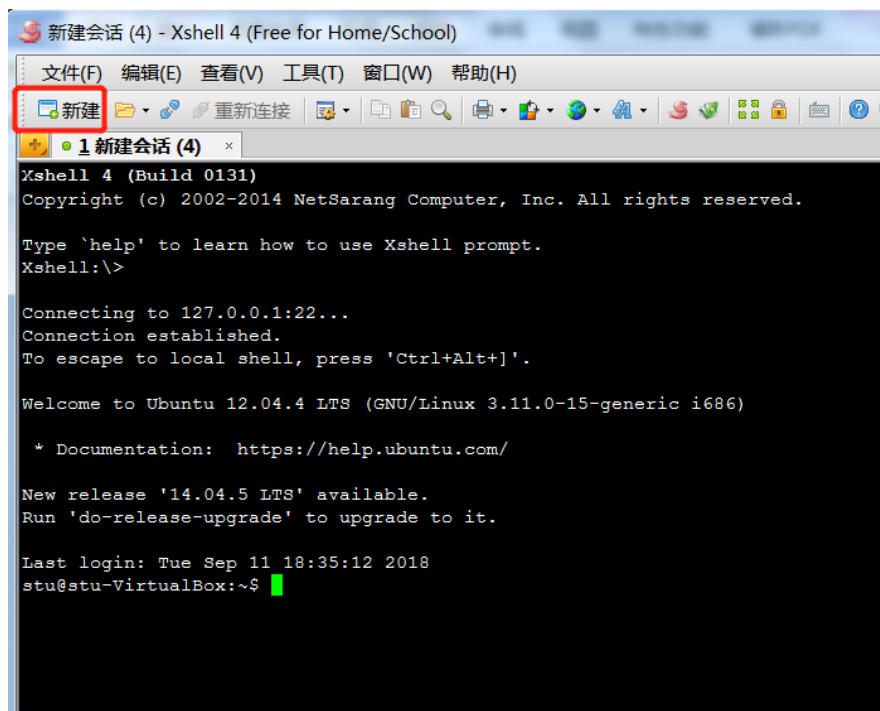


(2) 启动虚拟机，选择启动盘，即下载好的 Ubuntu 系统镜像，点击启动，安装 Ubuntu 系统，设置好用户名和密码。

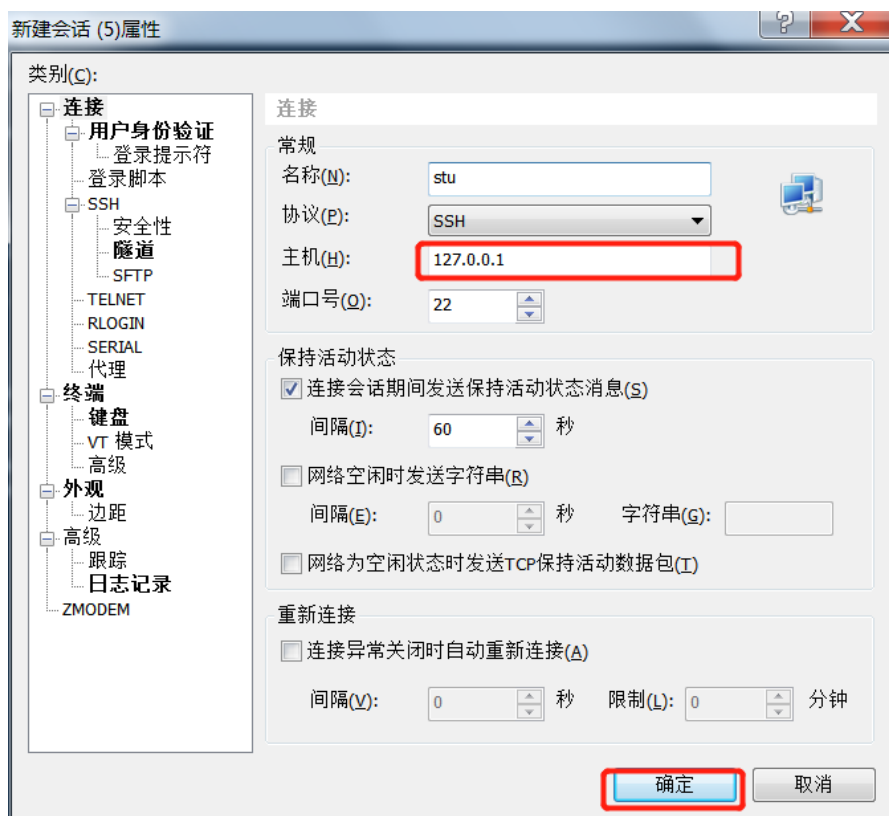
2.2 使用 xshell 访问虚拟机

在之前的教程中，我们都是直接在虚拟机中进行操作，但是考虑到有的同学电脑配置问题，可能出现虚拟机太卡的问题，因此我们在打开虚拟机后，可以在主机使用 xshell 工具访问虚拟机。以下为使用 xshell 建立主机和虚拟机的连接过程：

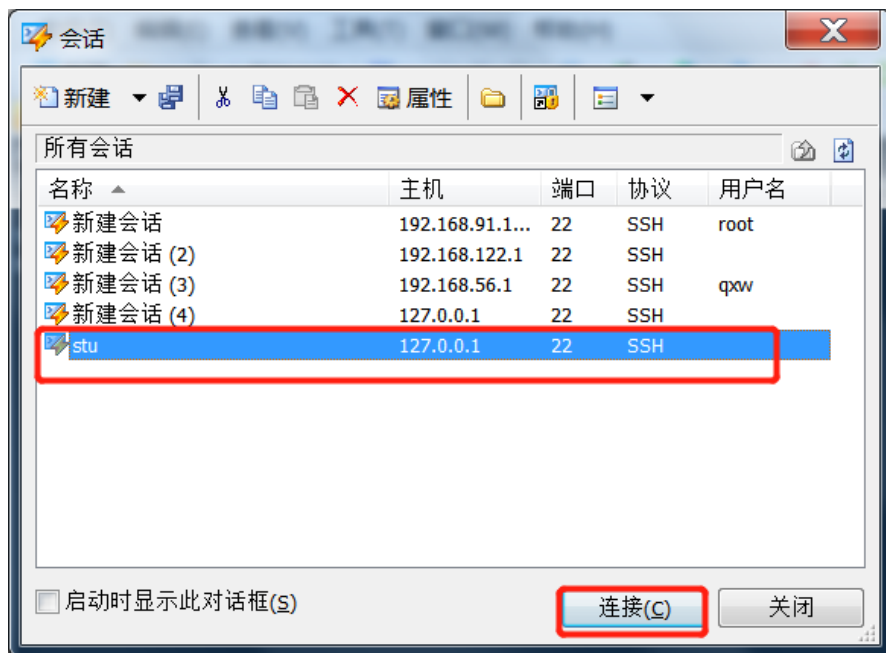
(1) 打开 xshell，点击新建，如图：



(2) 在弹出的窗口中主机输入 127.0.0.1，端口:22，点击确定。



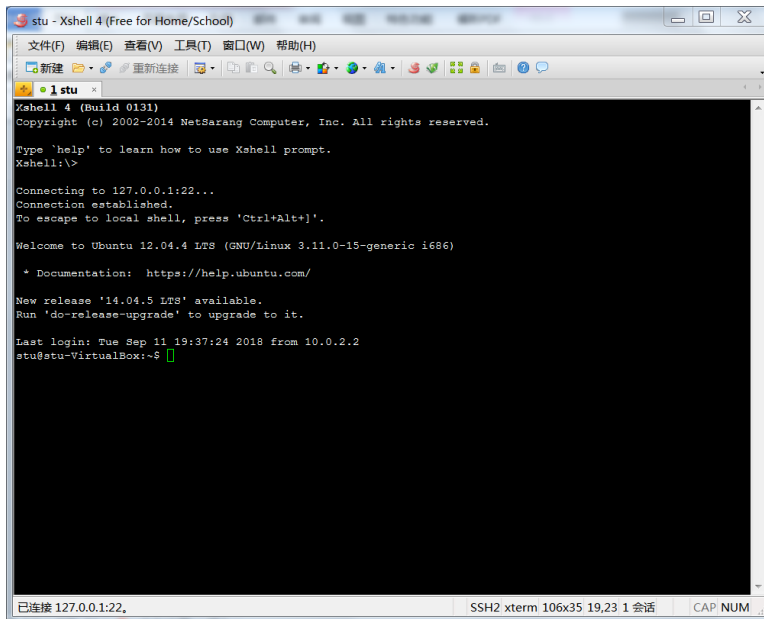
(3) 在弹出的会话框中选择刚建的 stu，点击连接。



(4) 输入用户名：stu，输入密码：123456，点击确定



(5) 完成 xshell 连接虚拟机，如图



2.3 MIPS 相关知识

2.3.1 MIPS 通用寄存器

在 MIPS 处理器中，算术指令的操作数只能为寄存器，MIPS 中的寄存器都是 32 位操作数。MIPS 提供了 32 个通用寄存器，其用法如下表所示：

寄存器编号	寄存器名	用法
\$0	\$zero	常数 0
\$1	\$at	保留给寄存器处理大常数
\$2~\$3	\$v0~\$v1	存放函数的返回值
\$4~\$7	\$a0~\$a3	存放函数调用参数
\$8~\$15	\$t0~\$t7	临时寄存器
\$16~\$23	\$s0~\$s7	存储寄存器，可存放变量
\$24~\$25	\$t8~\$t9	临时寄存器
\$26~\$27	\$k0~\$k1	用于保存中断信息
\$28	\$gp	全局指针
\$29	\$sp	栈指针
\$30	\$fp	帧指针
\$31	\$ra	子程序调用返回地址

2.3.2 MIPS 常用指令

MIPS 汇编指令格式一般如下：

标号：操作码 操作数 1 操作数 2 操作数 3 #注释

操作码表示执行什么操作，操作数表示执行的对象。其中 “标号：” 与 “#注释” 可以不写。

MIPS 常用的汇编指令如下表所示：

指令类型	指令	例子	含义
算术运算指令	加法	add \$s0,\$s1,\$s2	$\$s0 = \$s1 + \$s2$
	减法	sub \$s0,\$s1,\$s2	$\$s0 = \$s1 - \$s2$
	加立即数	addi \$s0,\$s1,10	$\$s0 = \$s1 + 10$
数据存入与读取	读出字	lw \$s0,offset(\$s1)	$\$s0 = \text{memory}[\text{offset} + \$s1]$
	存入字	sw \$s0,offset(\$s1)	$\text{memory}[\text{offset} + \$s1] = \$s0$
	读出半字	lh \$s0,offset(\$s1)	$\$s0 = \text{memory}[\text{offset} + \$s1]$

	读出无符号半字	lhu \$s0,offset(\$s1)	\$s0=memory[offset+\$s1]
	存储字节	sb \$s0,offset(\$s1)	memory[offset+\$s1]= \$s0
	读取字节	lb \$s0,offset(\$s1)	\$s0=memory[offset+\$s1]
	读取立即数到高半字	lui \$s0,30	\$s0=30*2^16
	读取无符号字节	lbu \$s0,offset(\$s1)	\$s0=memory[offset+\$s1]
逻辑运算	与	and \$s0,\$s1,\$s2	\$s0=\$s1&\$s2
	或	or \$s0,\$s1,\$s2	\$s0=\$s1 \$s2
	或非	nor \$s0,\$s1,\$s2	\$s0=~(\$s1 \$s2)
	与立即数	andi \$s0,\$s1,10	\$s0=\$s1&10
	或立即数	ori \$s0,\$s1,10	\$s0=\$s1 10
	逻辑左移	sll \$s0,\$s1,10	\$s0=\$s1<<10
	逻辑右移	srl \$s0,\$s1,10	\$s0=\$s1>>10
条件跳转	相等转移	beq \$s0,\$s1,15	if(\$s0==\$s1) goto PC=PC+4+4*15
	不相等转移	bne \$s0,\$s1,15	if(\$s0!= \$s1) goto PC=PC+4+4*15
	小于设置	slt \$s0,\$s1,\$s2	if(\$s1<\$s2) \$s0=1 else \$s0=0
	低于设置	sltu \$s0,\$s1,\$s2	if(\$s1<\$s2) \$s0=1 else \$s0=0
	小于常数设置	slti \$s0,\$s1,20	if(\$s1<20) \$s0=1 else \$s0=0
	低于常数设置	sltiu \$s0,\$s1,10	if(\$s1<10) \$s0=1 else \$s0=0
无条件跳转	直接跳转	j 2000	goto 2000*4
	间接跳转	jr \$ra	goto \$ra
	跳转并连接	jal 2000	\$ra=PC+4 goto 2000*4

其中有符号与无符号的差别是：有符号运算产生溢出时微处理器会产生异常，无符号运算发生溢出时不会产生异常。

2.3.3 CP0 寄存器及指令

在 MIPS 体系结构中，最多支持 4 个协处理器(Co-Processor)。其中，协处理器 CP0 是体系结构中必须实现的。它起到控制 CPU 的作用。MMU、异常处理、乘除法等功能，都依赖于协处理器 CP0 来实现。它是 MIPS 的精髓之一，也是打开 MIPS 特权级模式的大门。

CP0 寄存器用于控制处理器的状态改变并报告处理器的当前状态。这些寄存器通过 MFC0 指令来读或 MTC0 指令来写。对 CP0 的主要操作有以下的指令：

mfc0 rt, rd 将 CP0 中的 rd 寄存器内容传输到 rt 通用寄存器；

mtc0 rt, rd 将 rt 通用寄存器中内容传输到 CP0 中寄存器 rd；

2.3.4 伪指令

如下表所示，列出了 MIPS 常用的伪指令。

伪指令	含义
.rdata	只读数据
.align 2	两字节对齐
.globl status	全局变量 status
.set	设置代码的属性
.type @function	定义的是函数，这里的话可以看出是数据还是函数
.set noreorder	不让代码优化
.set reorder	优化代码
.extern g_data	对外部变量的引用
main:	标号 main
.set push	将当前的属性保存起来，是的设置的属性仅作用于当前的代码
.set pop	回复之前保存的属性
.set noat	不使用\$at 寄存器，当用到\$at 寄存器时不会报警
.set at	使用\$at 寄存器，当用到\$at 寄存器时会报警,但是不会报错
.text	代码段
.data	数据段
.ent	用于标记函数的起点
.end	用于标记函数结束的位置
.set macro A	定义宏 A
.endm	结束宏定义
.ascii "HELLO\0"	定义字符串不附带空结束符

.asciz "HELLO"	定义字符串附带空结束符
----------------	-------------

2.3.5 MIPS 程序的一个例子

我们给大家提供一个 MIPS 程序的例子。例子如下：

```

22  .globl asm_start           #定义全局变量asm_start
23  .align 2                  #2字节对齐
24  .type asm_start, @function #定义的asm_start是函数
25  .ent asm_start, 0         #开始asm_start
26  asm_start:                #asm_start标号
27  .frame sp, 0, ra
28      mtc0    $0, CP0_STATUS #将$0通用寄存器中内容传输到CP0中寄存器CP0_STATUS中
29      mtc0    $0, CP0_WATCHLO #将$0通用寄存器中内容传输到CP0中寄存器CP0_WATCHLO中
30      mtc0    $0, CP0_WATCHHI #将$0通用寄存器中内容传输到CP0中寄存器CP0_WATCHHI中
31
32      mfc0    t0, CP0_CONFIG  #将CP0中的CP0_CONFIG寄存器内容传输到t0通用寄存器
33      and     t0, ~0x7        #将t0通用寄存器中内容与上~0x7，并将结果存入t0中
34      ori     t0, 0x2         #将t0通用寄存器中内容或上0x2，并将结果存入t0中
35      mtc0    t0, CP0_CONFIG  #将t0通用寄存器中内容传输到CP0中寄存器CP0_CONFIG中
36
37      jr      ra              #跳转到ra寄存器指向的位置
38  .end asm_start
39  .size asm_start, .- asm_start
40

```