

本文档将引导读者如何在 Windows 操作系统上建立《专业嵌入式软件开发 — 全面走向高质高效编程》一书(后面简称本书)的学习环境。

首先,感谢 Cygwin 开源项目和来自 VMware 公司的 VMware 软件,它们使得在 Windows 操作系统上建立 Linux 操作系统环境成为可能。Cygwin 是一个在 Windows 操作系统上模拟 Linux 操作系统的开源软件,VMware 软件使得我们能在 Windows 操作系统上运行 Linux 虚拟机。通过这两个软件中的任一个,我们可以在 Windows 操作系统上从事大量以前只能在 Linux 操作系统上进行的软件开发工作。后者相比前者所需的内存和硬盘开销较大,但运行效率较前者更高。

本书中的很多实践都是在 Cygwin 环境中完成的。在继续介绍环境搭建之前,请确保读者手头计算机的处理器是 x86 或 x86-64,Windows 操作系统即可以是 32 位也可以是 64 位版本的。

## 1 安装 Cygwin 环境

安装 Cygwin 环境分为两大步。第一步是获取安装包,第二步是安装和配置。

### 1.1 获取安装包

当需要从 Cygwin 官网下载时可以参照本小节,否则读者可以跳过。Cygwin 的安装包在附书光盘的“Environment\Cygwin\Installation”目录中也能找到。

从 Cygwin 官网 [www.cygwin.org](http://www.cygwin.org) 上下载 setup.exe 安装程序文件并运行它,界面如图 1.1 所示。

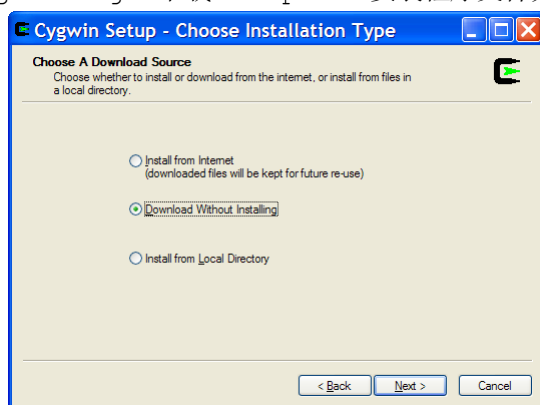


图 1.1

选择“Download Without Installing”,然后点击“Next”进入图 1.2 所示界面。

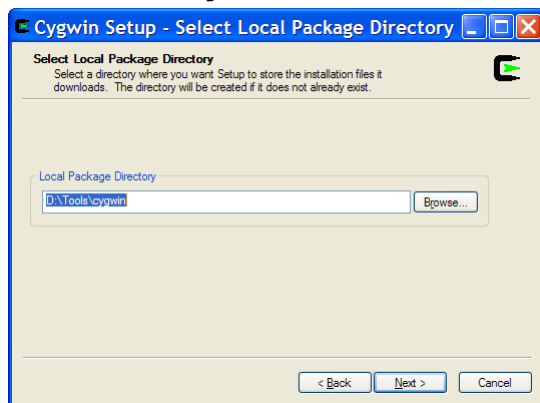


图 1.2

这一步需要我们选择被下载的软件包放到什么位置,默认路径为 setup.exe 文件所在目录。选择“Next”后,进入网络连接模式选择界面,如图 1.3 所示。

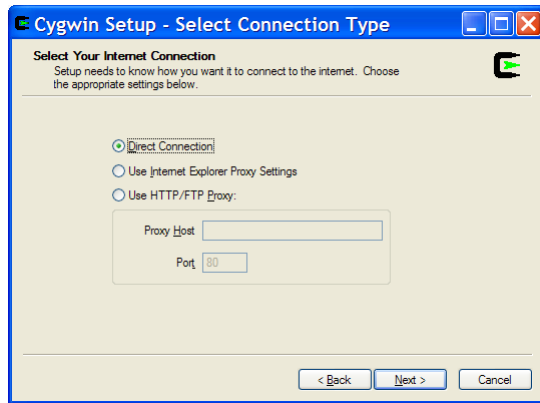


图 1.3

读者根据自己的情况做出选择。选择完后点击“Next”出现图 1.4 所示的界面。

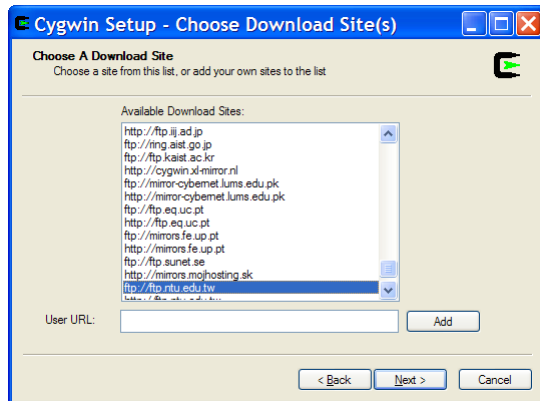


图 1.4

在这里我们需要选择从哪一个站点下载安装包。读者可以选择列表中的任一个，也可以尝试找到网速比较快的站点。继续点击“Next”，进入图 1.5 所示的界面。

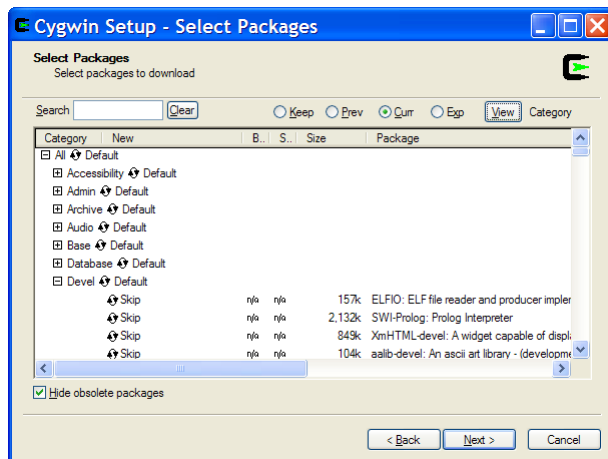


图 1.5

这一步需要选择所需下载的软件包。默认情况下，安装程序会为我们选择最基本的软件包，但我们得在这之上增加其他的软件包。安装程序所显示的所有软件包是采用树状结构按类别 (category) 进行管理的，通过展开每一个类别可以查看该类别中的软件包。图 1.6 示例了 Devel 类别中的 binutils 软件包被选中的情形。选择所需下载软件包的方法是点击软件包所在行的“Skip”，使其显示软件包的版本号。

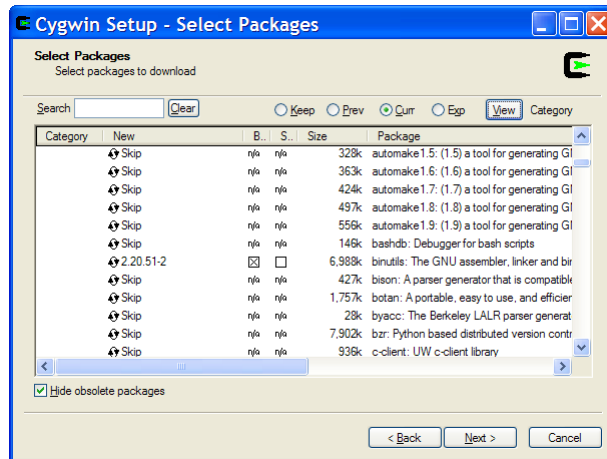


图 1.6

学习本书需要选择图 1.7 的表内所示的软件包。

安装包名	包所属类别	用途说明
binutils	Devel	GCC 编译器所需的开发工具集
gcc4		GCC 的 C/C++ 编译器
gcc4-core		
gcc4-g++		
gdb		来自 GNU 的调试器
make		用于实现编译环境
所有包	Perl	Perl 语言运行环境
vim	Editors	Linux 中通用的 vi 文本编辑器

图 1.7

注意，在选择“Perl”类别时可以通过点击“Perl”类别名之后的“Default”使其变成“Install”，从而实现对整个类别中软件包的选择，如图 1.8 所示。

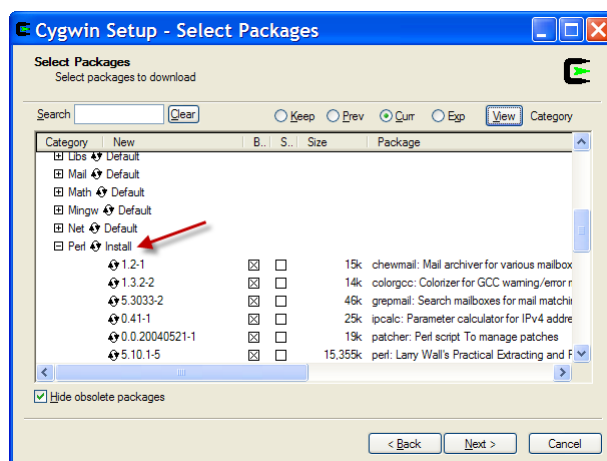


图 1.8

完成所有包的选择后选择“Next”，进入如图 1.9 所示界面。

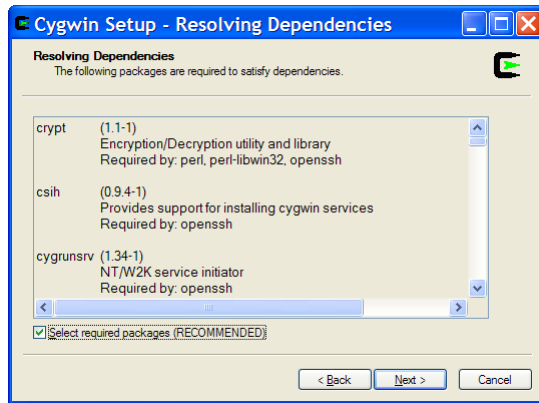


图 1.9

安装程序根据我们的选择告知其它依赖的软件包，我们只需信赖它并做出“Next”选择而开始软件包的下载。一旦全部软件包下载完了后将获得图 1.10 所示的界面。选择“Finish”结束安装包下载工作。

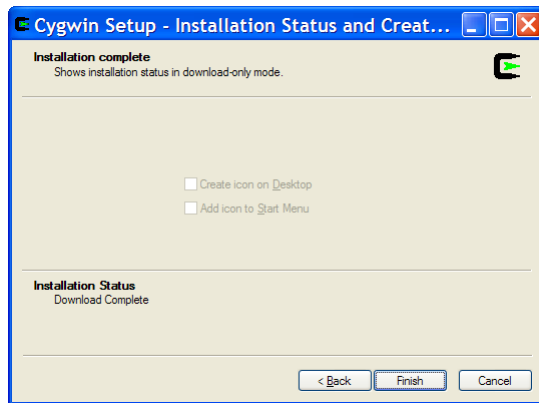


图 1.10

此时如果查看 setup.exe 文件所在目录中的内容会象图 1.11 那样。所有的软件包以压缩包的形式放在名称以“ftp”开头的目录内。

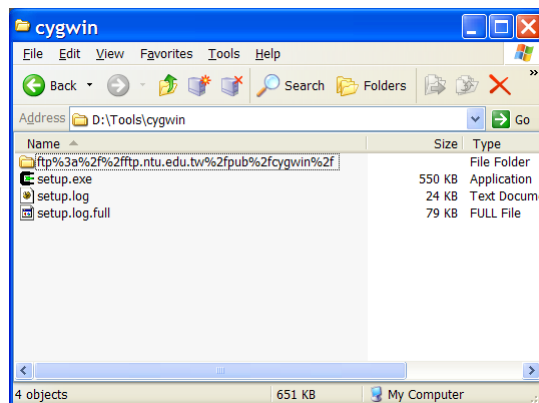


图 1.11

## 1.2 安装与配置

无论读者是自己从 Cygwin 官网下载安装包，或是直接使用本书光盘中 Environment/Cygwin/Installation 目录下的，都需要通过运行 setup.exe 安装程序来安装 Cygwin，如图 1.12 所示。

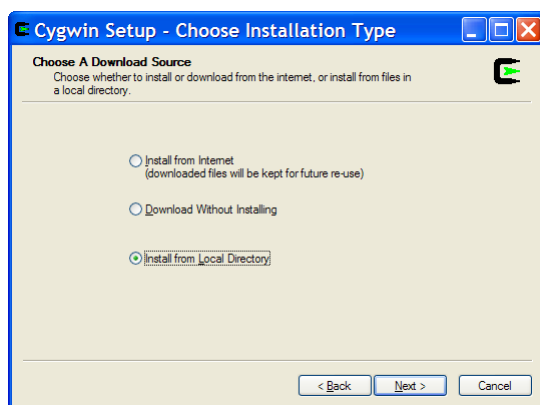


图 1.12

选择“Install from Local Directory”，然后点击“Next”获得图 1.13 所示的界面。

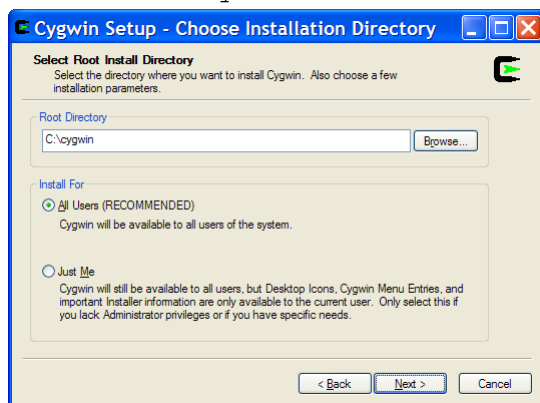


图 1.13

这一步需要选择将 Cygwin 安装到哪一个目录中。作者的习惯是使用缺省的“C:\cygwin”<sup>1</sup>。选择“Next”进入图 1.14 所示的界面。

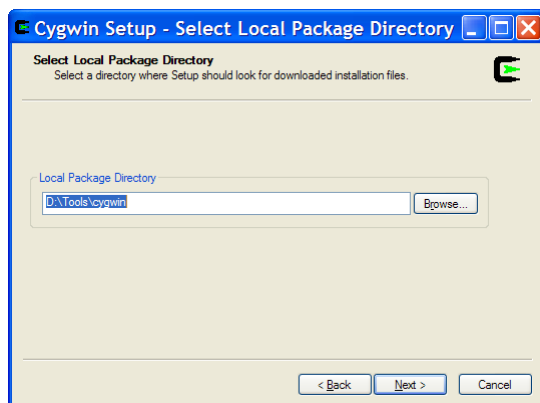


图 1.14

这步需要选择从 Cygwin 官网下载的软件包的存放位置的父目录。通常我们直接选择“Next”进入图 1.15 所示的界面。

<sup>1</sup> 如果读者选择了其他目录名而非这里演示的“C:\cygwin”，则文档后面所有提及“C:\cygwin”目录名的地方都用自己的实际目录加以替换。

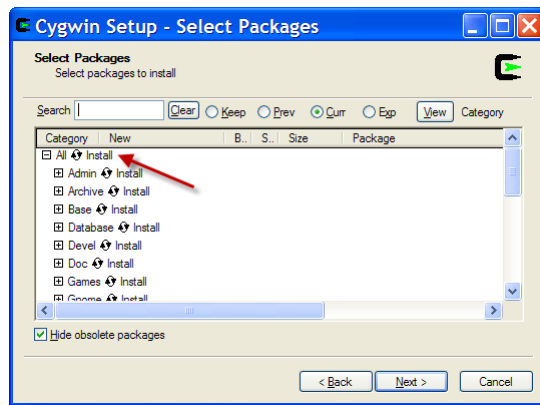


图 1.15

通过点击“ALL”类别之后的“Default”，使其变成“Install”以选择安装所有已下载的软件包。点击“Next”后安装程序将为我们完成安装工作，并在最后生成运行 Cygwin 的快捷方式。

安装完成后运行 Cygwin。Cygwin 在第一次运行时需要花几分钟时间做初始化。初始化时所生成的 .bashrc 文件需要做一些更改使 Cygwin 更友好。 .bashrc 文件的位置在“C:\cygwin\home”下的某子目录中。对其的更改点是图 1.16 内行号下面加了红色下划线的几处。作者计算机上使用的 .bashrc 文件可以从光盘的“Environment\Cygwin”目录中找到。

```

111 # Misc :)
112 # alias less='less -r'                # raw control characters
113 # alias whence='type -a'              # where, of a sort
114 alias grep='grep --color'            # show differences in colour
115
116 # Some shortcuts for different directory listings
117 alias ls='ls -hF --color=tty'         # classify files in colour
118 # alias dir='ls --color=auto --format=vertical'
119 # alias vdir='ls --color=auto --format=long'
120 # alias ll='ls -l'                    # long list
121 # alias la='ls -A'                    # all but . and ..
122 # alias l='ls -CF'                    #
123
124 alias embedded='cd /embedded/build'
125 alias clearrtos='cd /ClearRTOS/build'
126 export LD_LIBRARY_PATH=/usr/local/lib
127 export EDITOR=vi
128
129 PS1='\[\e]0;\w\a\]\n\[\e[32m\]yunli.blog.51CTO.com \[\e[33m\]\w\[\e[0m\]\n\$ '
130

```

图 1.16

114 和 117 行是在原有的内容中去掉开头的“#”（该符号所在行后面的内容为注释），这使得 ls 和 grep 两个命令所显示的结果具有色彩而更具可读性。

124 行定义一个名为 embedded 的别名，使得在 Cygwin 中运行 embedded 命令时进入“/embedded/build”目录。增加别名的目的是提高操作效率。125 行增加的 clearrtos 别名具有相同的目的。

127 行为 gdb 指定在其内编辑程序时所使用的工具，这里指定的是 vi。

129 行的目的是让 Cygwin 能在命令行上显示作者的博客地址，读者可以根据自己的需要进行定制。

.bashrc 文件一旦更新过后如要使其生效，需要重新启动 Cygwin，或者象图 1.17 那样通过运行 source 命令使 .bashrc 中的内容立即生效。

图 1.17

为了让 Cygwin 的窗口看起来比较舒服，以及能查看更多历史命令，需要对 Cygwin 的窗口属性做一定的设置（右键点击 Cygwin 快捷方式，选中属性）。在作者的电脑上采用图 1.18 所示的设置使得 Cygwin 运行时刚好占满整个桌面且有 9999 行缓冲命令。

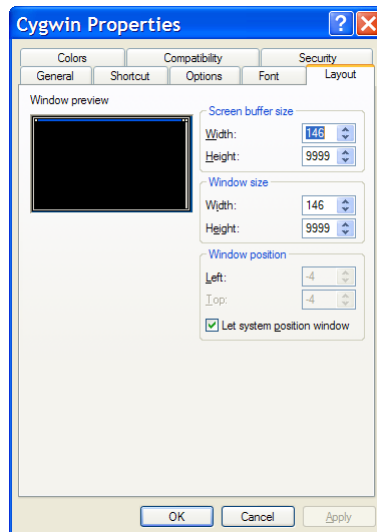


图 1.18

### 1.3 验证环境

Cygwin 安装完后可以通过编译附书光盘中 Project 目录下的 embedded 项目加以验证。先将 embedded.tar.gz 拷贝到 “C:\cygwin” 目录下，并在该目录下运行 “tar xzvf embedded.tar.gz” 命令解压。解压时出现的信息请忽略。**注意，读者不能直接拷贝光盘中已解压的代码，否则在有些 Windows 操作系统中会出现文件权限问题。**解压之后，运行图 1.19 所示的 embedded 和 make 命令来检查 embedded 项目是否能成功编译。必须确保 .bashrc 内 embedded 别名中的路径设置与 embedded 项目所拷贝到的目录路径是一致的，Windows 中的 “C:\cygwin” 目录，对应于 Cygwin 中的 “/” 根目录。

```
yunli.blog.51CTO.com ~
$ embedded

yunli.blog.51CTO.com /embedded/build
$ make
make[1]: Entering directory `/embedded/code/tools/err2str'
g++ -Wall -Weffc++ -Woverloaded-virtual -gdwarf-2 -g3 -O2 -o robjs/err2str.o -c err2str.c
g++ -L/embedded/build/release -o /embedded/build/release/err2str.exe robjs/err2str.o
make[1]: Leaving directory `/embedded/code/tools/err2str'
make[1]: Entering directory `/embedded/code/platform/common/src'
```

图 1.19

编译成功的标志是后面能看到 “:-) Completed” 字符串，如图 1.20 所示。

```
make[1]: Entering directory `/embedded/code/application/sync/recursive'
gcc -std=gnu99 -Wall -gdwarf-2 -g3 -O2 -I/embedded/code/platform/common/inc -I/embedded/code/platform/device/inc -I/embedded/code/platform/timer/v3/inc -I/embedded/code/platform/task/common/inc -I/embedded/code/platform/sync/v3/inc -o robjs/main.o -c main.c
gcc -L/embedded/build/release -o /embedded/build/release/recursive.exe robjs/main.o -Wl,-Ma
timerv3 -ldevice -lboard -lctrlc -lsyncv3 -lcpu -lcommon
make[1]: Leaving directory `/embedded/code/application/sync/recursive'

:-) Completed
```

图 1.20



## 2 准备 Linux 虚拟机

本书的第 31 和 32 章因为所使用的 Valgrind 工具必须在 Linux 操作系统上完成学习。如果读者手头已经有一台安装于 x86 或 x86-64 处理器的 32 位 Linux 操作系统的话，可以跳过本章。否则请继续阅读以了解如何在 Windows 上准备 Linux 虚拟机。

Linux 操作系统有很多发布，作者因为易用性而比较喜欢其中的 Ubuntu。为了在 Windows 操作系统上安装 Ubuntu 虚拟机需要用到 VMware 公司提供的“VMware Workstation”，但这一软件是商用软件。好在 VMware 公司还提供免费的“VMware Player”，通过它可以运行已安装好的虚拟机，为此作者为读者准备好了一个可以直接运行的 Ubuntu 虚拟机。从光盘的 Environment/Linux 目录下可找到其 ZIP 压缩包。

在附书光盘的“Environment\Linux”目录下可以找到“VMware Player”的安装包，或者读者可以从 VMware 的官网下载一个最新的。安装“VMware Player”不在这里进行介绍，因为它的安装过程没有任何特别之处，除了不要忘记在安装完成后重新启动计算机。

运行“VMware Player”后，打开解压获得的虚拟机文件 Ubuntu.vmx 运行虚拟机。一开始会弹出图 2.1 所示的对话框询问虚拟机的来源。

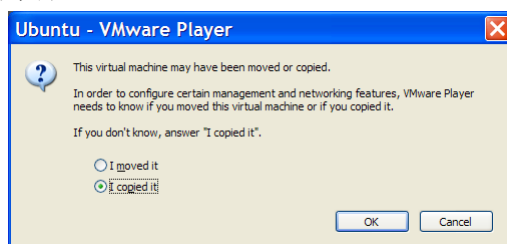


图 2.1

选择“I copied it”并点击“确定”。当虚拟机准备好后，将获得如图 2.2 所示的登录界面。

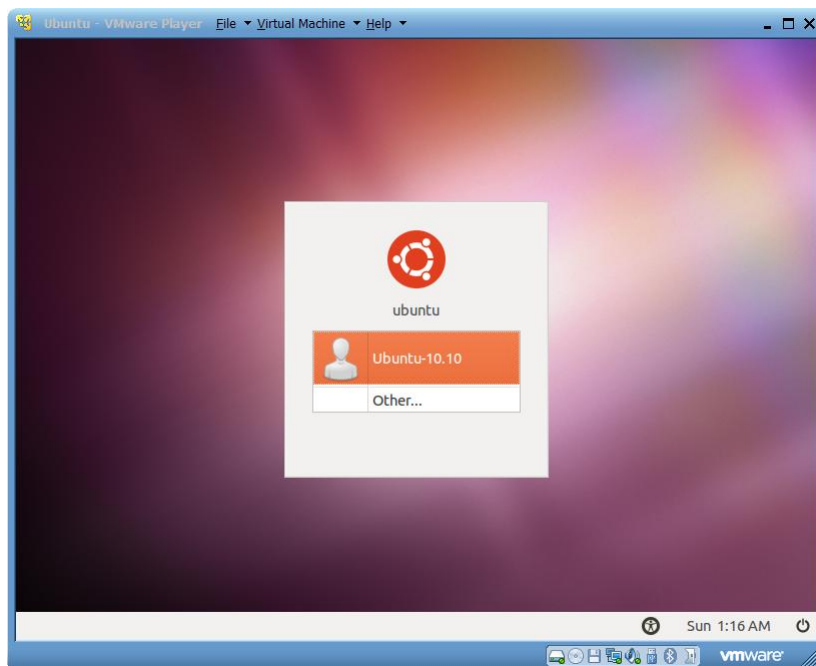


图 2.2

在“VMware Player”软件的虚拟机空间内点击鼠标左键让虚拟机获得焦点<sup>2</sup>。之后，点击位于图中间的 Ubuntu-10.10 以 iubuntu 用户名登录，登录密码是“12345678”（引号不包含在内）。完成登录后

<sup>2</sup>如果要让鼠标从虚拟机回到 Windows 操作系统，需按“Ctrl + Alt”组合键。



将出现图 2.3 所示的桌面。

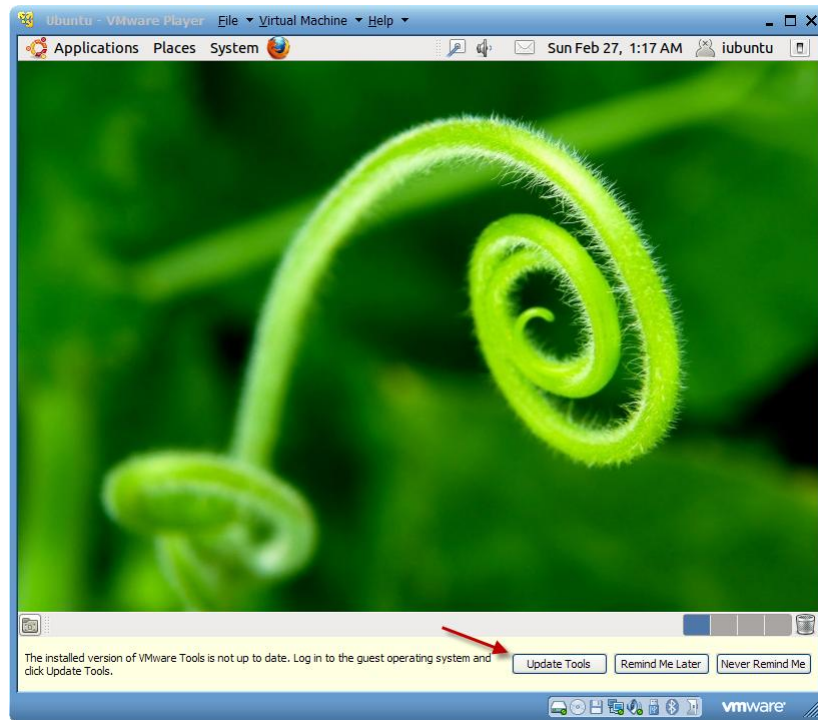


图 2.3

## 2.1 安装“VMware Tools”

“VMware Tools”通过在 Linux 虚拟机中安装驱动的方式，实现虚拟机与 Windows 操作系统之间的无缝通讯。安装好“VMware Tools”工具后，我们可以直接将位于 Windows 操作系统中的文件或目录拖到虚拟机中实现两个“机器”之间的拷贝，也可以自由地在两个操作系统之间使用拷贝和粘贴。附书光盘内的虚拟机上已经安装好了“VMware Tools”工具，如不想了解该安装过程，读者可以选择跳过阅读本节后面的内容。

当“VMware Player”检测到虚拟机中没有安装“VMware Tools”时，会在下方进行提示(如图 2.3 所示)。如果没有出现提示，可以通过使用“VMware Player”的菜单选择安装“VMware Tools”，如图 2.4 所示。

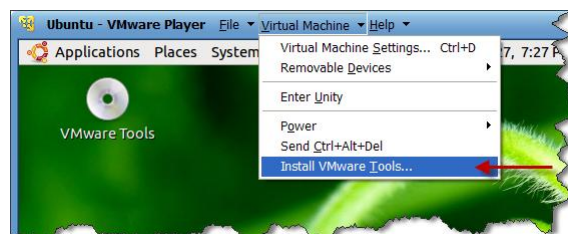


图 2.4

通过选择图 2.3 下方的“Update Tools”按钮，“VMware Player”会弹出图 2.5 的询问框以告知需要从官网下载安装包。在继续之前，请确保计算机能接入 Internet，然后选择“Download and Install”。

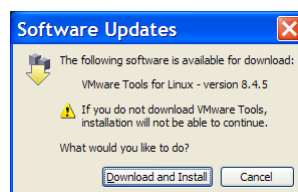


图 2.5

当安装包下载完成后会在虚拟机中出现一个虚拟的光盘，如图 2.6 所示。

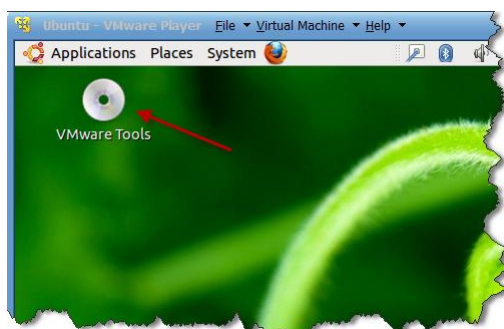


图 2.6

在 Ubuntu 中选择“Applications”菜单下的“Terminal”子菜单(如图 2.7 所示)打开一个命令行窗口。

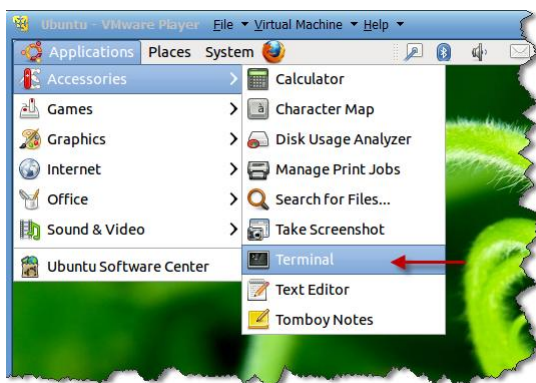


图 2.7

命令窗口的默认工作路径是登录用户的 home 目录，这里是 /home/iubuntu。直接运行“`sudo tar xzvf /media/VMware\ Tools/VMwareTools-8.4.5-324285.tar.gz`”<sup>3</sup>命令将“VMware Tools”的安装包进行解压缩，如图 2.8 所示。Sudo 命令会要求提供 root 账号的密码，请输入与 iubuntu 用户名相同的密码<sup>4</sup>。

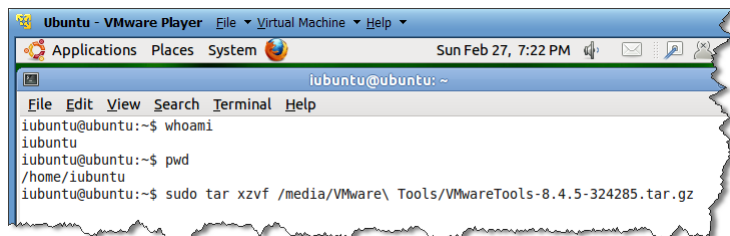


图 2.8

解压完成后在当前目录下能找到名为 vmware-tools-distrib 的子目录，象图 2.9 所示的那样运行该目录下的 vmware-install.pl 脚本启动“VMware Tools”的安装。除了输入密码外，我们可以以直接回车的方式选择安装过程中所提问题的默认答案。

<sup>3</sup> 请注意，文件名可与之有所不同。

<sup>4</sup> Ubuntu 中不允许以 root 账号登录，所有需要 root 权限的地方都得使用 sudo 命令。

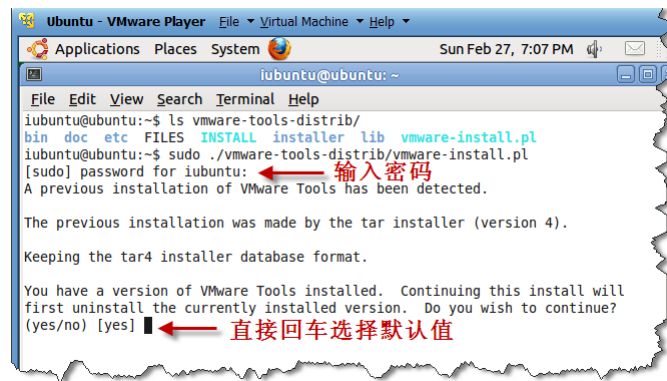


图 2.9

安装完成后为了使“VMware Tools”生效需要注销用户并重新登录。重新登录后，读者可以尝试在 Windows 操作系统和 Ubuntu 虚拟机之间使用拷贝和粘贴的方式验证“VMware Tools”的功能是否起作用，也可以尝试将 Windows 操作系统中的文件拖到 Ubuntu 虚拟机内加以验证。

## 2.2 验证环境

虚拟机内的/home/iubuntu 目录下已包含学习本书所需的 embedded 和 ClearRTOS 两个项目的代码，如图 2.10 所示。

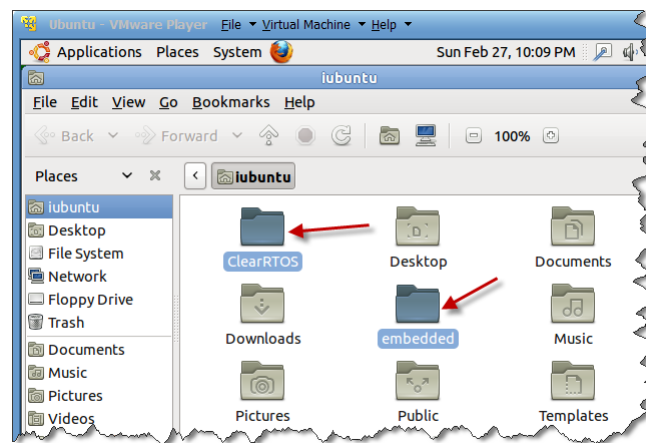


图 2.10

在命令行终端上进入 embedded/build 子目录，并分别运行“make unittest”和“make dcheck”进行验证。如果在两个命令的最终都能看到“:-)”则说明环境已经好了。

\*\*\*\*\*

至此，所需的学习环境已经搭建好了。让我们开始本书的学习之旅吧！

## 作者联系方式

邮箱: [yunli.book@gmail.com](mailto:yunli.book@gmail.com)

微博: [weibo.com/UltraEmbedded](http://weibo.com/UltraEmbedded)

博客: [yunli.blog.51CTO.com](http://yunli.blog.51CTO.com)