Linux

# Part 1. Linux 简介

## 1.1 Linux 简介

Linux内核最初是由芬兰人李纳斯▪托瓦斯(Linus Torvalds)在赫尔辛基大学上学时处于个人爱好而编写的。Linux是一套免费试用和自由传播的类Unix操作系统，是一个基于POSI和UNIX的多用户、多任务、支持多线程和多CPU的操作系统。

Linux能运行主要的UNIX工具软件、应用程序和网络协议。它支持32位和64位硬件。Linux继承了Unix网络为核心的设计思想，是一个性能稳定的多用户网络操作系统。

## 1.2 Linux发行版

Linux的发行版就是将Linux内核与应用软件做一个打包。由于Linux是免费、高效、稳定的操作系统，所以发行版本比较多。

目前市面上较知名的发行版有: RedHat、Ubuntu、CentOS、Debian、Fedora、SuSE、OpenSUSE、Arch Linux、SolusOS等。

## 1.3 Linux应用领域

Linux版本众多，针对不同的应用场景有不同的发行版本，从嵌入式设备到超级计算机，并且在服务器领域确定了绝对地位，通常服务器使用LAMP(Linux + Apache + MySQL + PHP)或LNMP(Linux + Nginx + MySQL + PHP)组合。目前Linux在各个领域都有广泛的使用，特别实在服务器占有绝对地位。

## 1.4 Linux vs Windows

目前国内Linux更多的是应用于服务器上，而桌面操作系统更多使用的是 Windows。主要区别如下:

|  |  |  |
| --- | --- | --- |
| **vs** | **Windows** | **Linux** |
| 界面 | 界面统一，外壳程序固定所有Windows程序菜单几乎一致，快捷键也几乎相同 | 图形界面风格依发布版不同而不同，可能互不兼容。GNU/Linux的终端机是从UNIX传承下来，基本命令和操作方法也几乎一致。 |
| 驱动程序 | 驱动程序丰富，版本更新频繁。默认安装程序里面一般包含有该版本发布时流行的硬件驱动程序，之后所出的新硬件驱动依赖于硬件厂商提供。对于一些老硬件，如果没有了原配的驱动有时很难支持。另外，有时硬件厂商未提供所需版本的Windows下的驱动，也会比较头痛。 | 由志愿者开发，由Linux核心开发小组发布，很多硬件厂商基于版权考虑并未提供驱动程序，尽管多数无需手动安装，但是涉及安装则相对复杂，使得新用户面对驱动程序问题（是否存在和安装方法）会一筹莫展。但是在开源开发模式下，许多老硬件尽管在Windows下很难支持的也容易找到驱动。HP、Intel、AMD等硬件厂商逐步不同程度支持开源驱动，问题正在得到缓解。 |
| 使用 | 使用比较简单，容易入门。图形化界面对没有计算机背景知识的用户使用十分有利。 | 图形界面使用简单，容易入门。文字界面，需要学习才能掌握。 |
| 学习 | 系统构造复杂、变化频繁，且知识、技能淘汰快，深入学习困难。 | 系统构造简单、稳定，且知识、技能传承性好，深入学习相对容易。 |
| 软件 | 每一种特定功能可能都需要商业软件的支持，需要购买相应的授权。 | 大部分软件都可以自由获取，同样功能的软件选择较少。 |

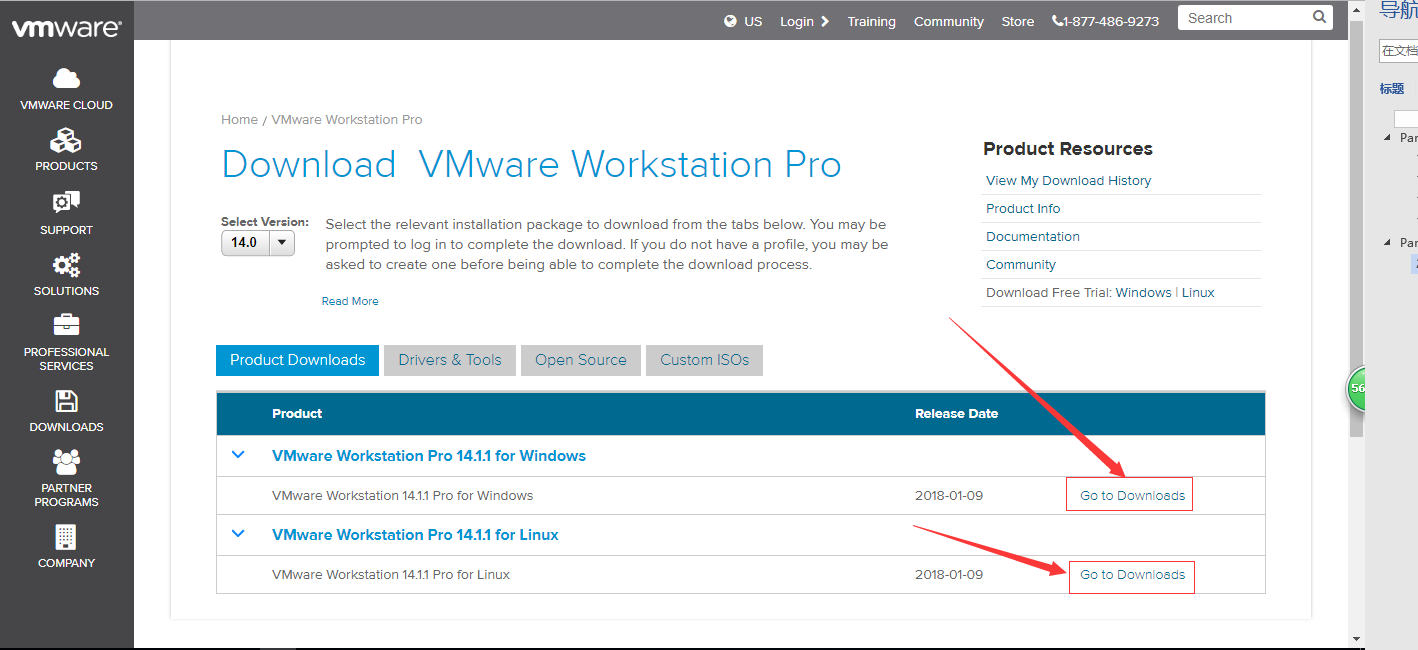
# Part 2. Linux安装

## 2.1 Linux安装方式

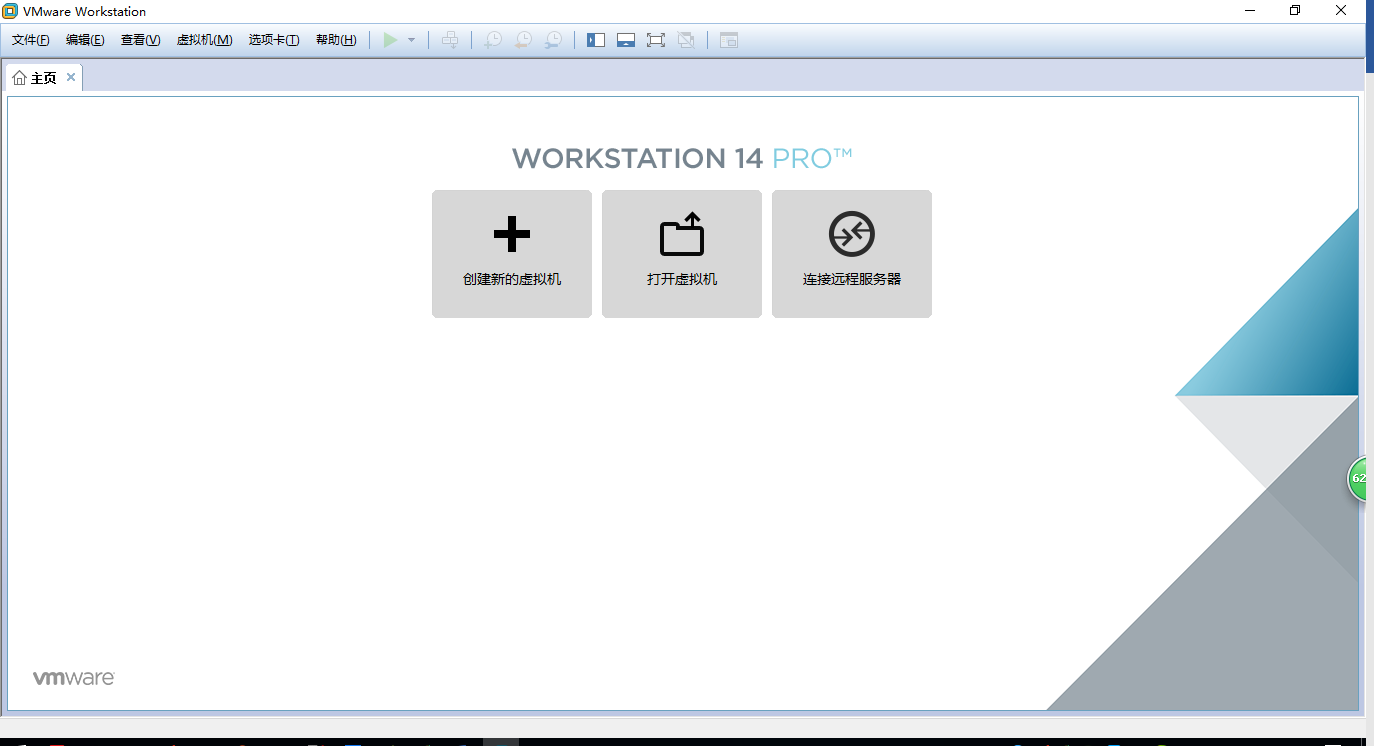
安装方式主要分为物理机安装和虚拟机安装，个人学习建议虚拟机安装，下面将介绍Vmware虚拟机进行安装，以Centos 7 为例进行安装。

下载Vmware虚拟软件：<https://my.vmware.com/en/web/vmware/info/slug/desktop_end_user_computing/vmware_workstation_pro/14_0>

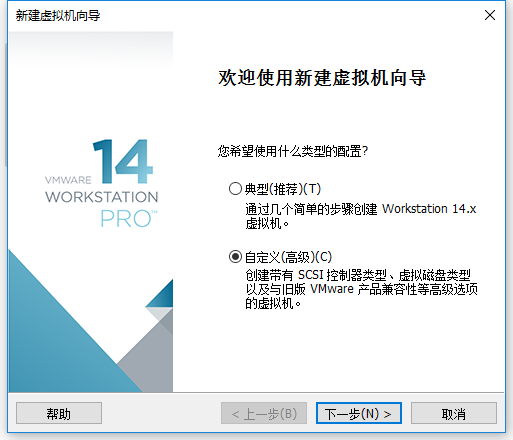
下载Centos：<https://www.centos.org/download/>



安装过程和普通软件安装过程类似，这里不再赘述，安装完成之后，打开Vmware，如图所示：



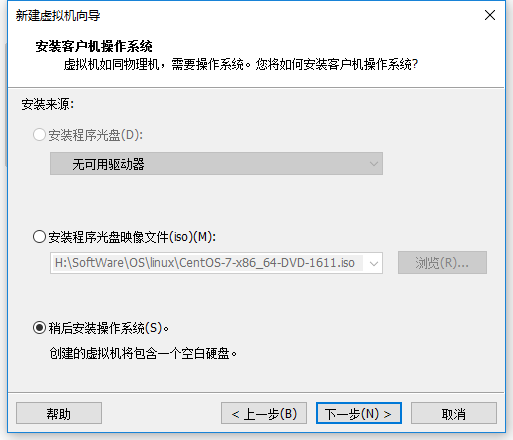
点击“创建新的虚拟机”后如图所示：



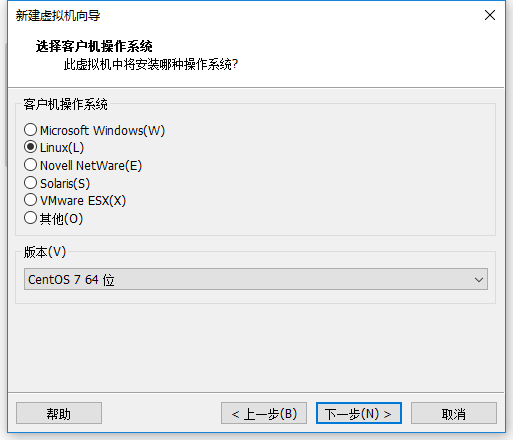
这里有两种安装方式“典型”和“自定义”，这里选择“自定义”进行安装，因为典型安装默认是最小化安装，对于Linux初学者，没有桌面是非常令人崩溃的，点击下一步会看到如下图所示：



这一步是选择Vmware的版本，如果你的个人电脑不是非常有年代的话，选择默认推荐的版本即可，点击下一步会看到如下图所示:



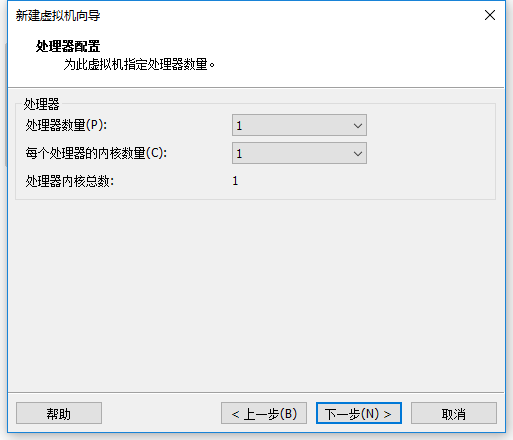
这一步是选择安装镜像(也就是linux的系统镜像文件)，这里选择“稍后安装操作系统”，点击下一步会看到如下图所示：



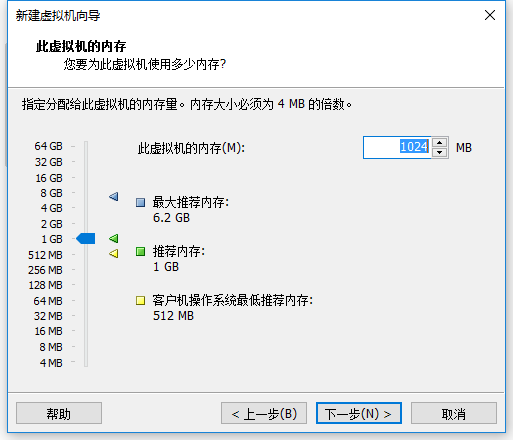
这一步是选择操作系统的类型，因为我们是安装Linux，所以选择Linux，然后在“版本”中选择CentOS 7 64位，如果你下载的不是Centos，也可以根据你所下载的版本进行选择，点击下一步会看到如下图所示：



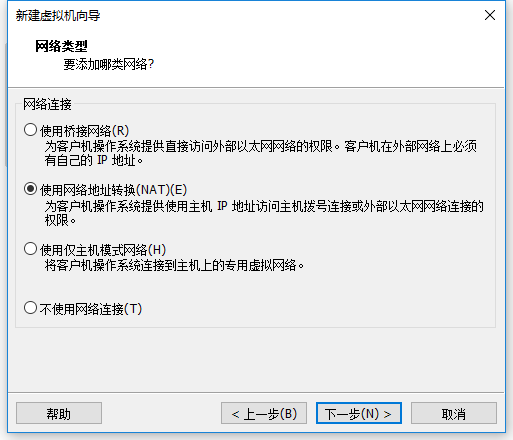
这一步是填写虚拟机名称和虚拟机安装位置，名称可以默认，也可以根据自己爱好设置一个名称，安装位置建议只要要安装到系统盘即可，点击下一步会看到如下图所示：



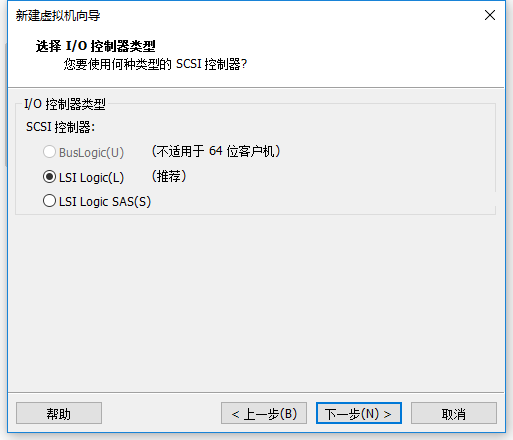
这一步是选择处理器数量和每个处理器的内核数量，一般选择默认即可，点击下一步会看到如下图所示：



这一步是为虚拟机分配运行内存，一般不要超过实体机的50%，但是Centos7 本身也要求至少1GB内存，所以分配2GB即可正常使用，点击下一步会看到如下图所示：



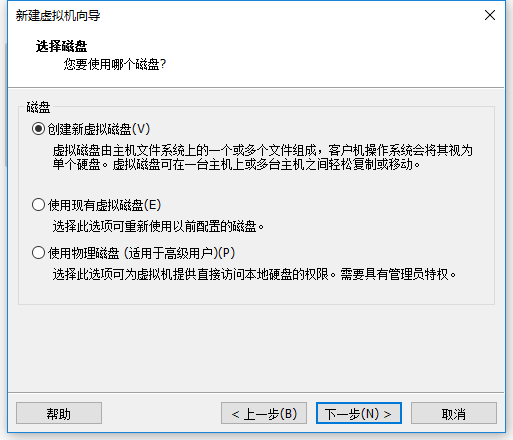
这一步是选择网络连接方式，对于初学者来说选择默认的网络连接方式即可，点击下一步会看到如下图所示：



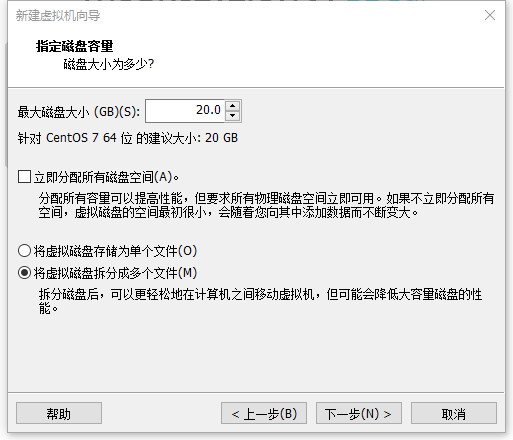
这一步是选择I/O控制器类型，默认即可，点击下一步会看到如下所示：



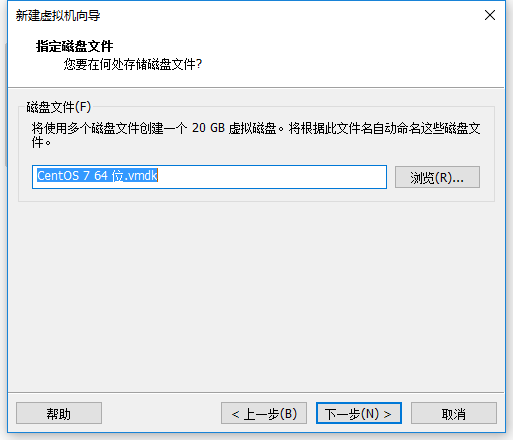
这一步是选择磁盘类型，默认即可，点击下一步会看到如下所示：



这一步是选择磁盘，三种类型，选择“创建新虚拟磁盘”，点击下一步会看到如下所示：



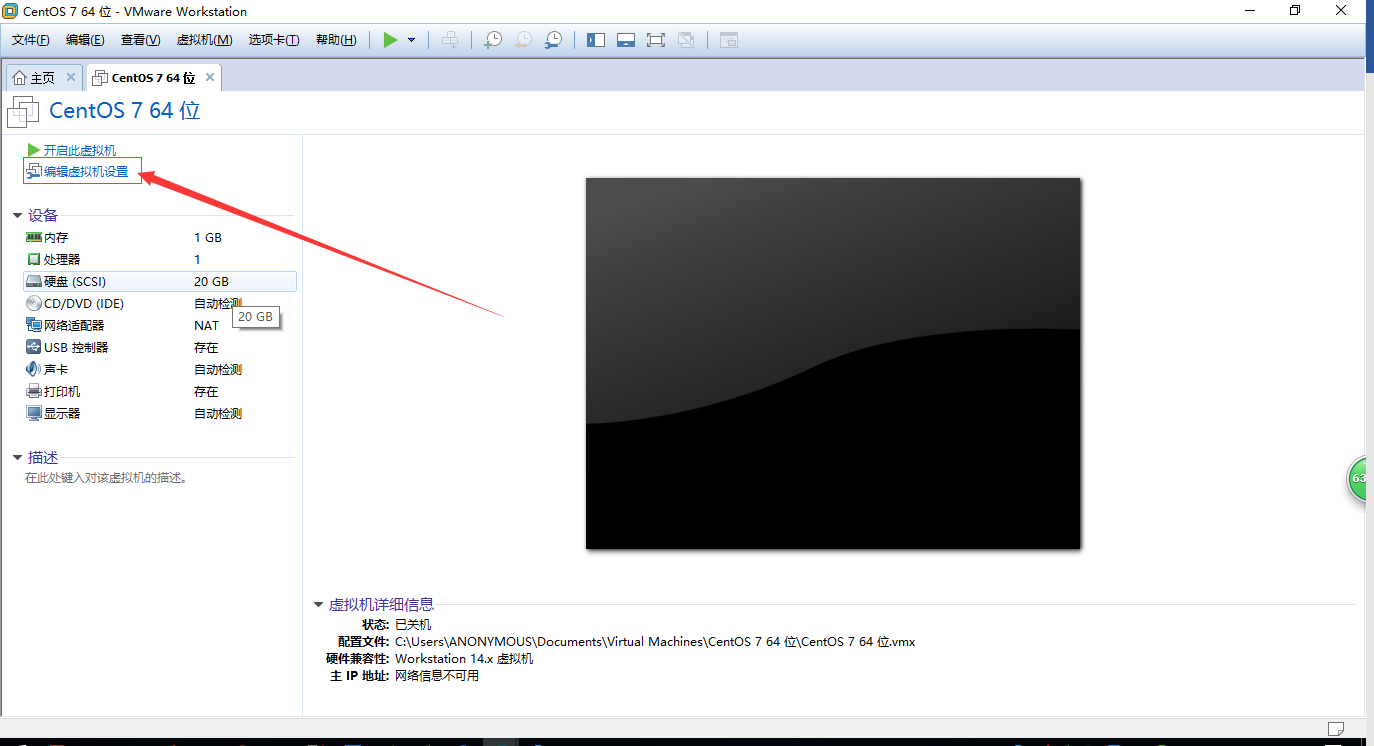
这一步是为虚拟机分配磁盘空间，一般默认的20GB空间是足够使用，在后边使用过程中如果磁盘空间不足，还可以为其分配磁盘更多的磁盘空间，点击下一步会看到如下图所示：



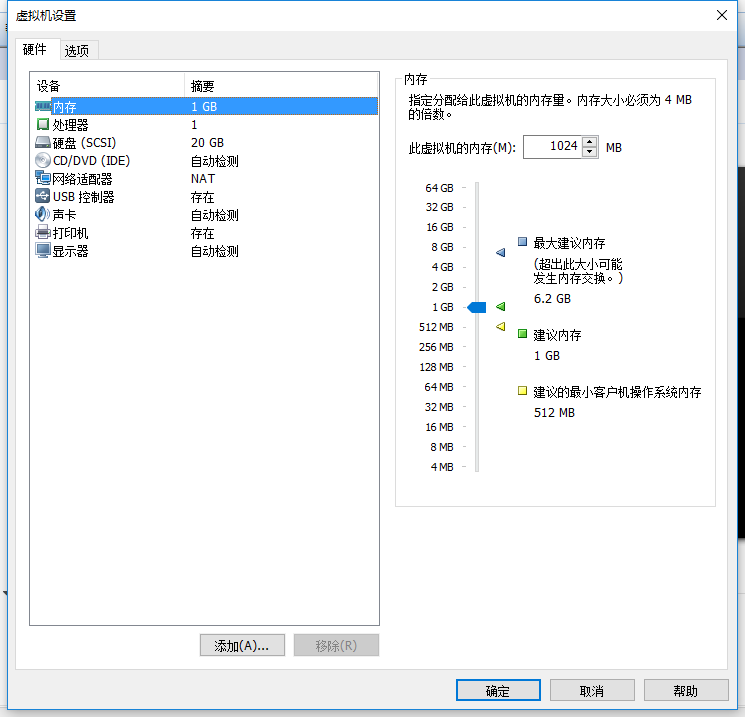
这一步是选择vmdk文件保存的位置，默认在该用户下边的“文档”目录下，可以选择其他位置，点击下一步会看到如下图所示：

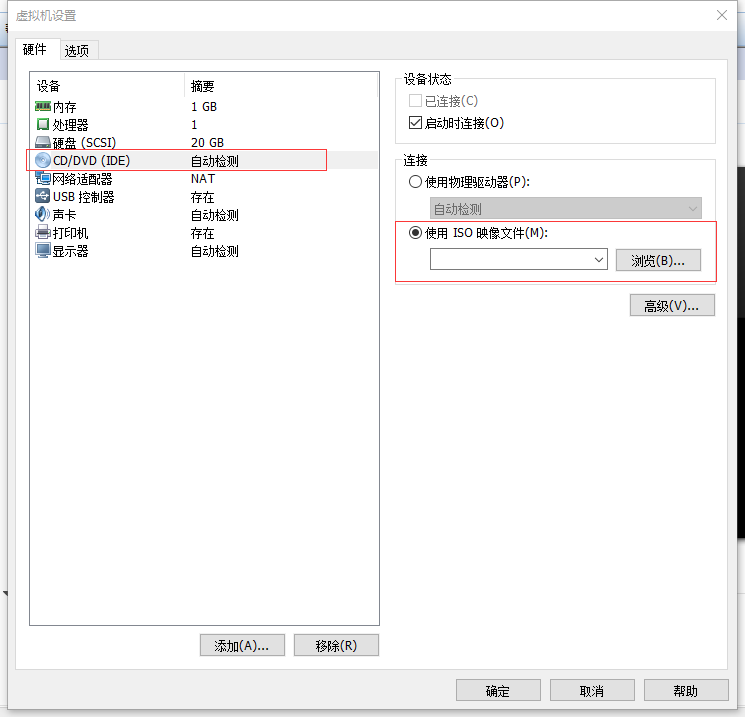


这一步点击完成即可，虚拟机基本创建完成，点击完成会看到如下图所示：

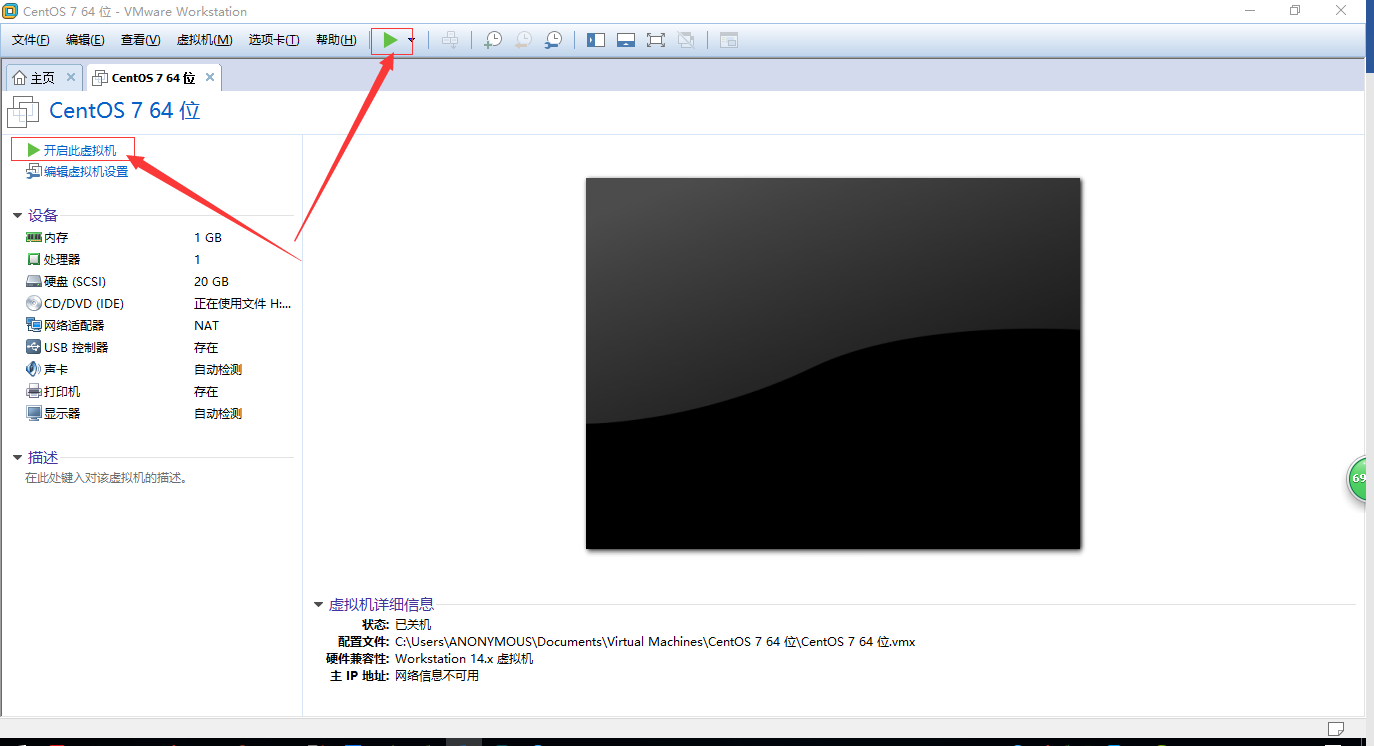


点击上图中的“编辑虚拟机设置”，会看到如下图所示：

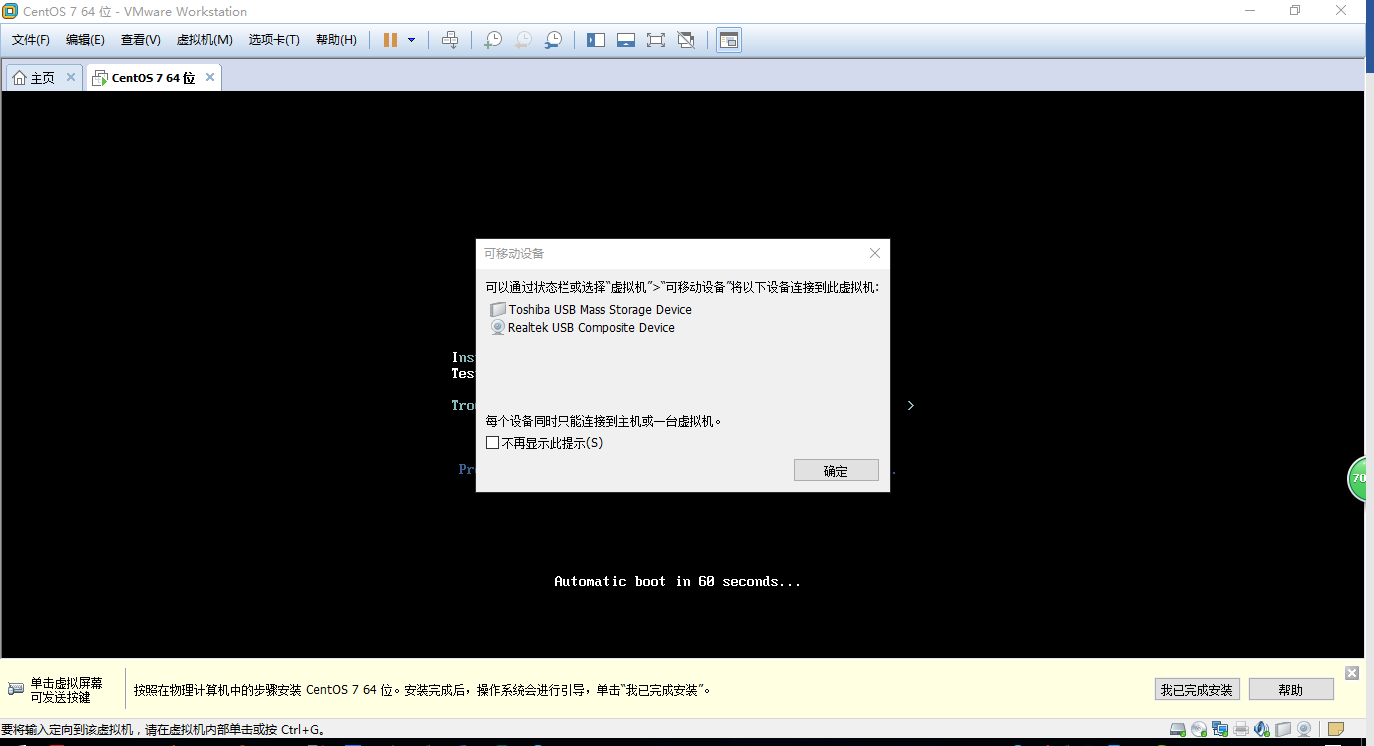




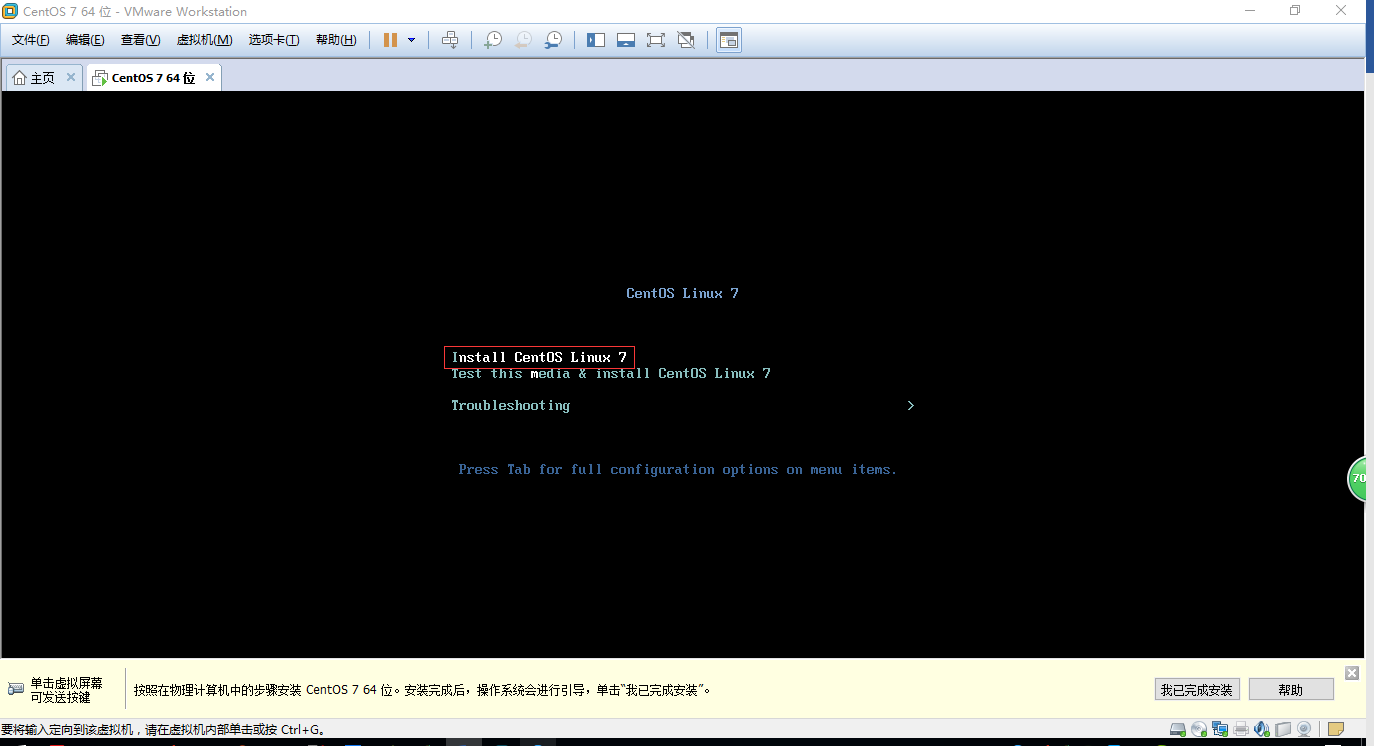
选择Linux系统镜像文件位置，点击“确定”会看到如下图所示：



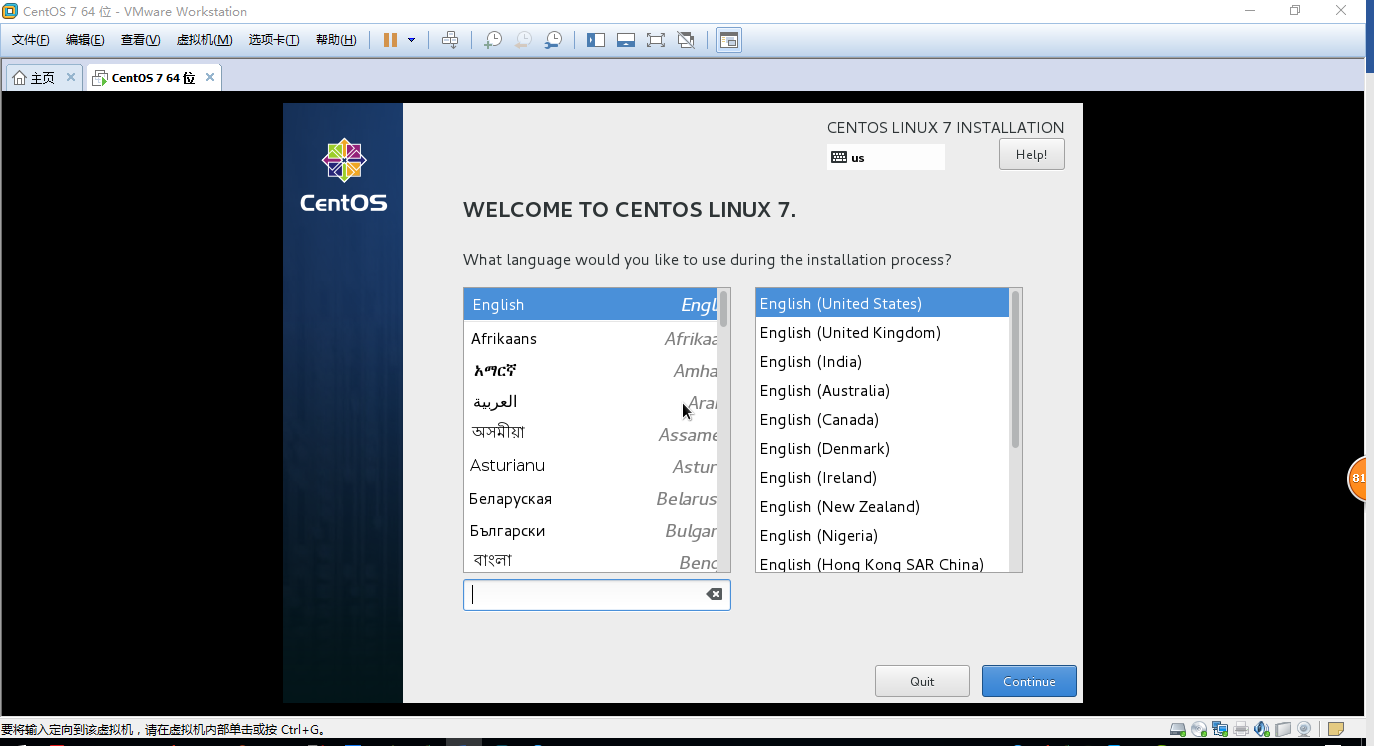
启动虚拟机，点击上图中指向的两个地方都可以启动，点击启动后会看到如下图所示：



弹出框，点击“确定”会看到如下图所示：



使用上下箭头选择Install Centos Linux 7，鼠标点击进去后，按“Ctrl+Alt”即可将鼠标移动出来，回车后等待片刻会看到如下图所示：



这一步是选择安装的语言，如果英文较好可以选择安装默认(English)，也可以选择简体中文，点击“Continue”会看到如下图所示：



这一步是一些基本配置：

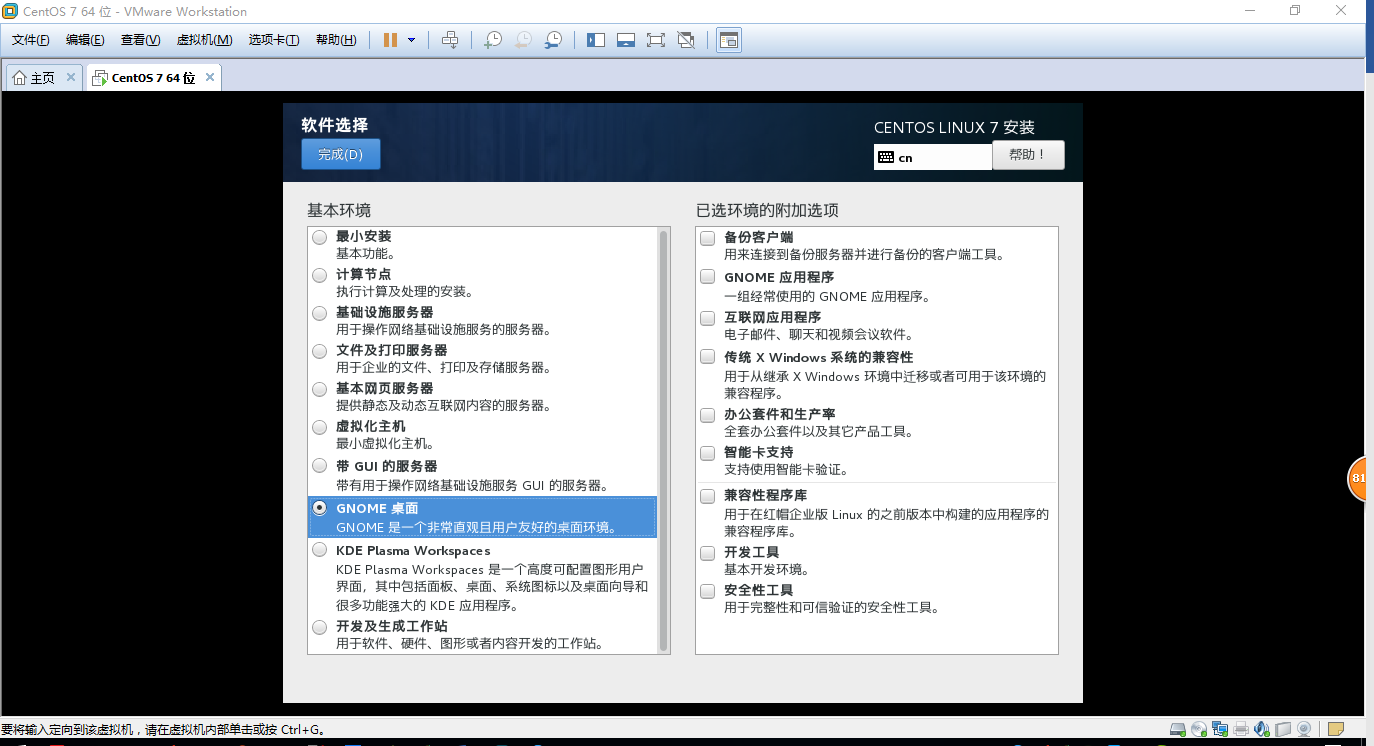
“日期和时间”：默认会自动识别出来，也可以点击去自己选择；

“键盘”：可以选择汉语；

“语言支持”：简体中文；

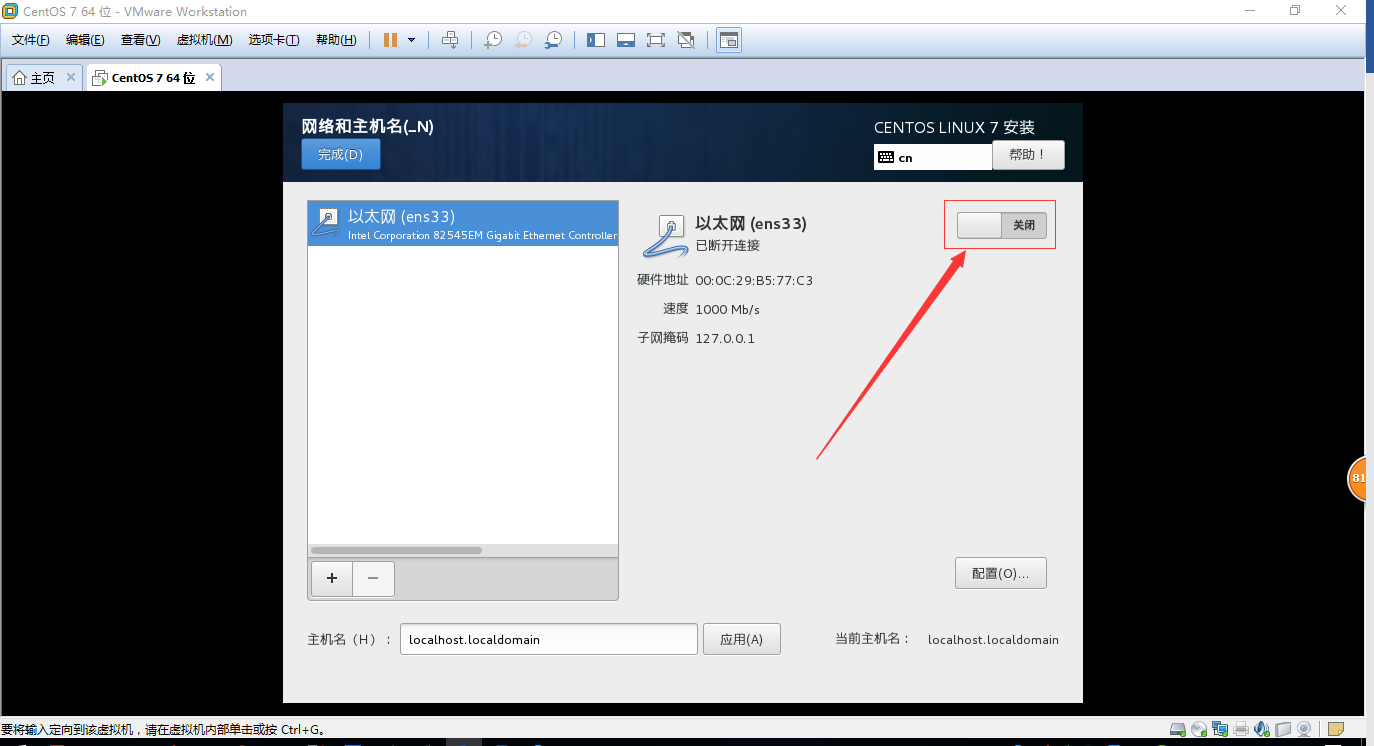
“安装源”：安装的来源，不用去管；

“选件选择”：需要手动选择，不然是最小化安装，一般选择GNOME桌面即可，如下图所示：

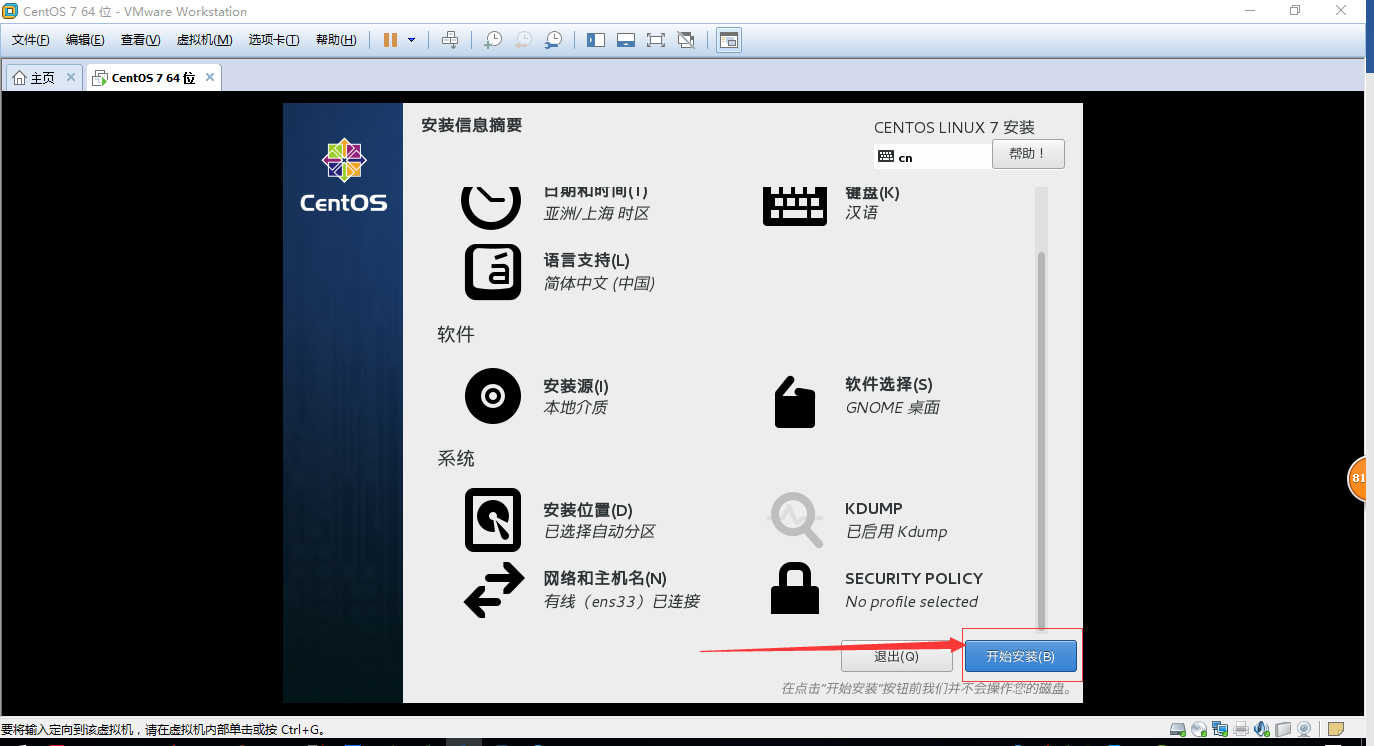


“安装位置”：需要配置，可以手动分区，也可以进入后直接点击完成，也就是自动分区。

“网络和主机名”：需要配置，点击进去点击连接，如下图所示：



然后点击“开始安装”，如下图所示：

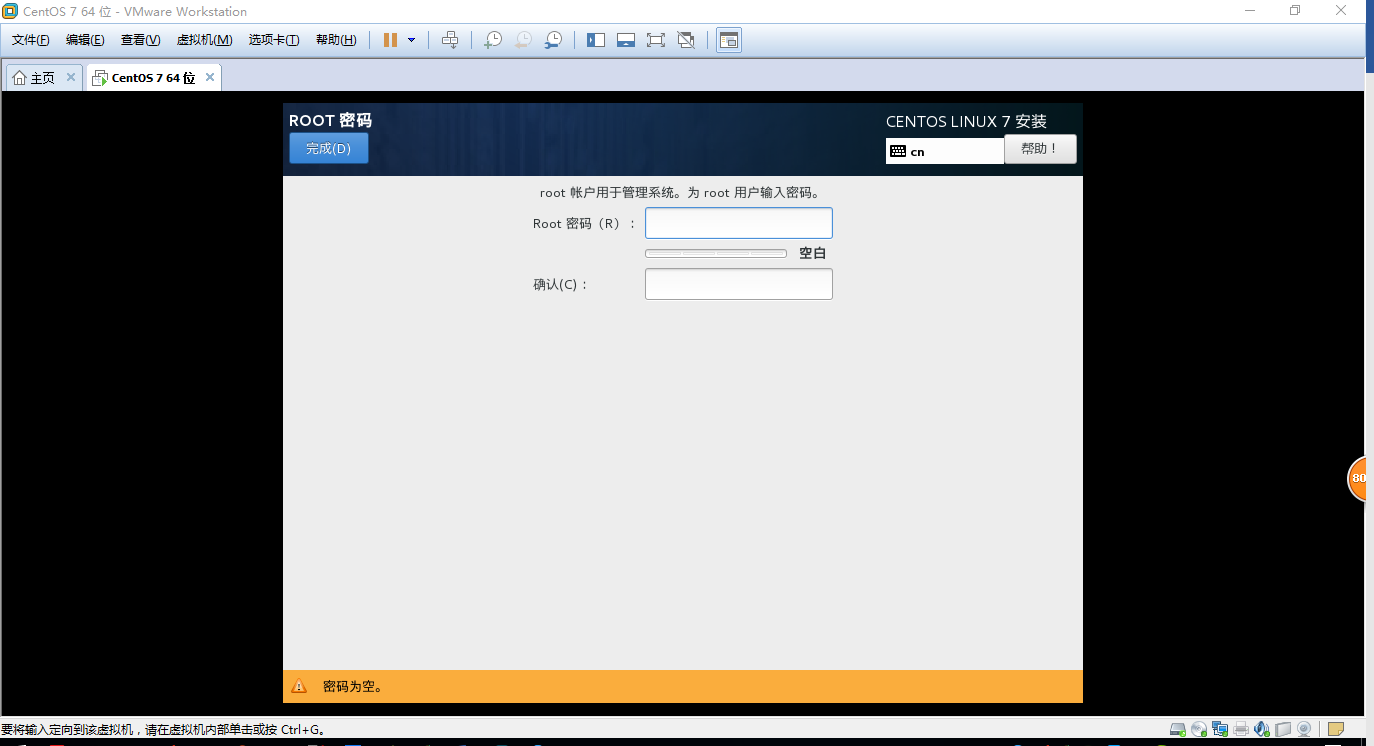


点击“开始安装”后，会看到如下图所示：

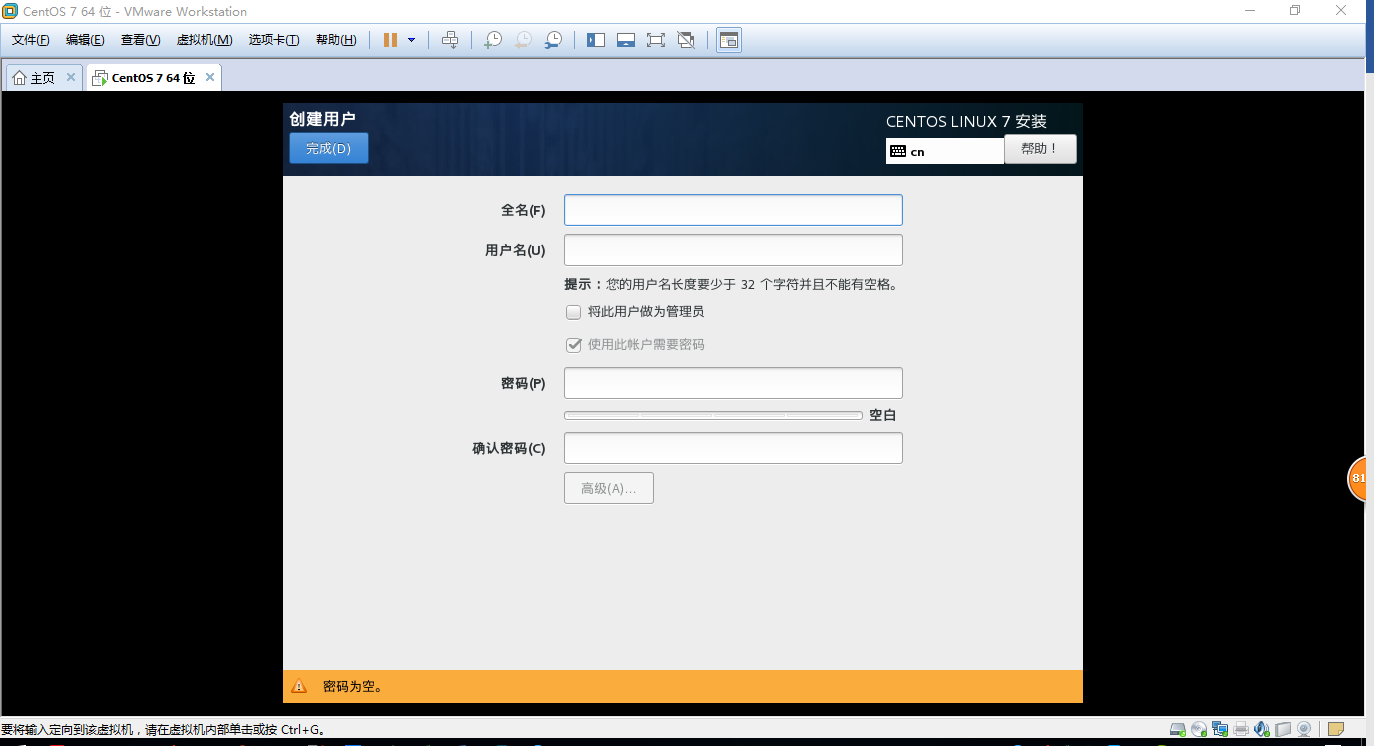


这里可以创建ROOT密码，和创建用户。

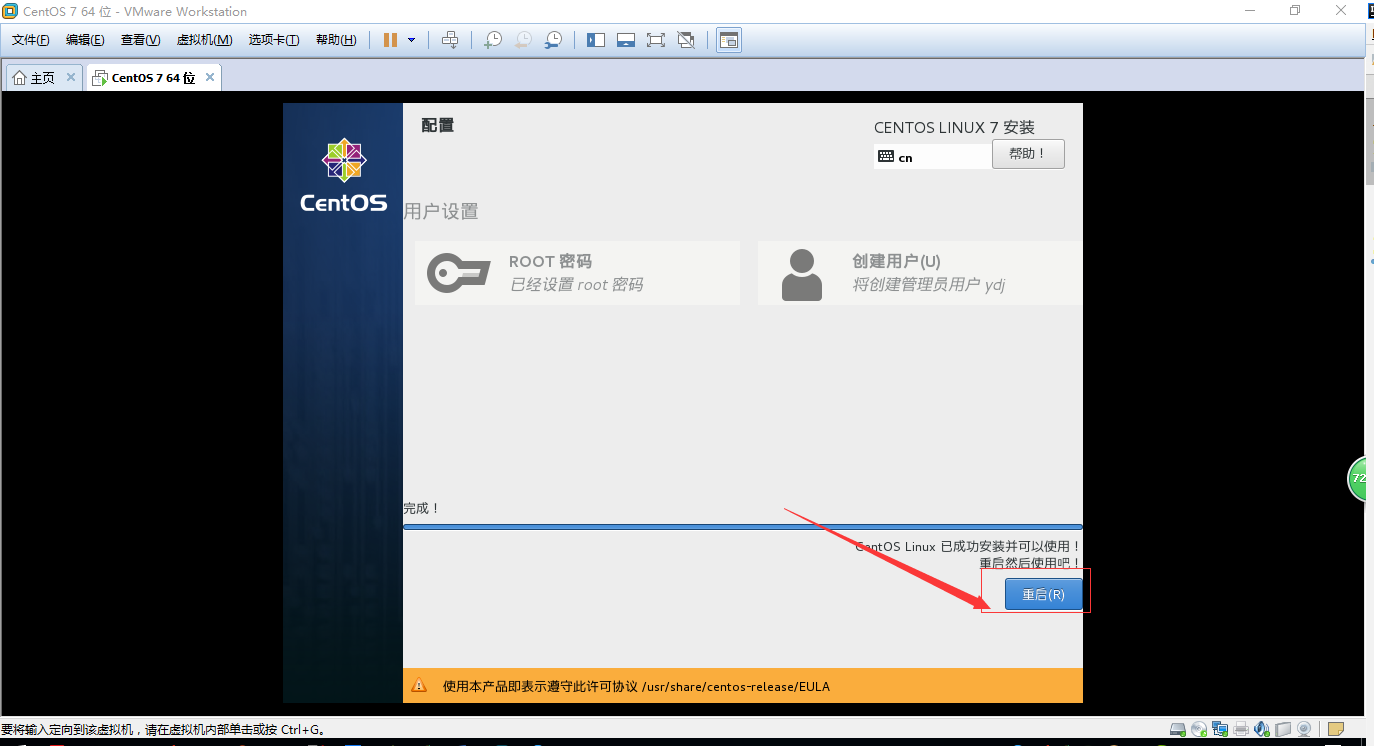
创建ROOT密码，如果密码太简单，需要双击“完成”按钮才能设置成功，如下图所示：



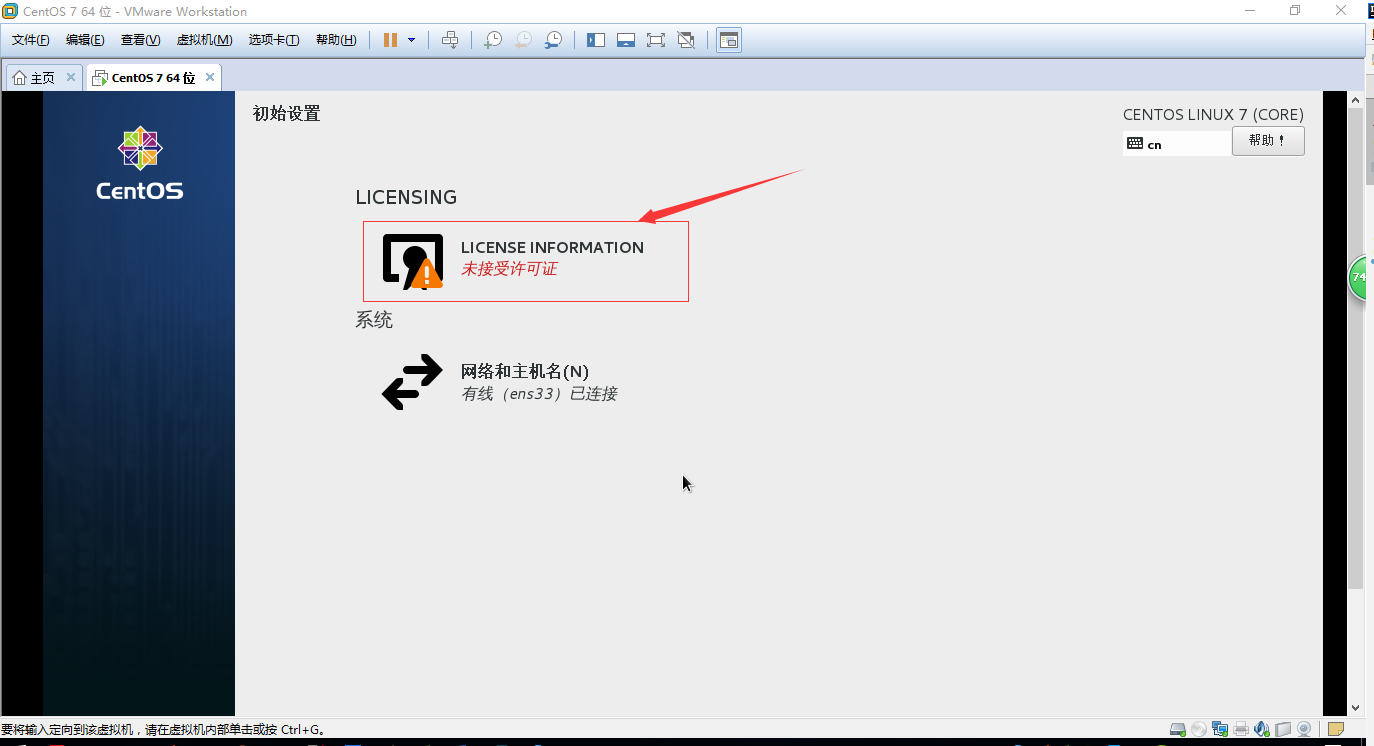
创建用户，如下图所示：



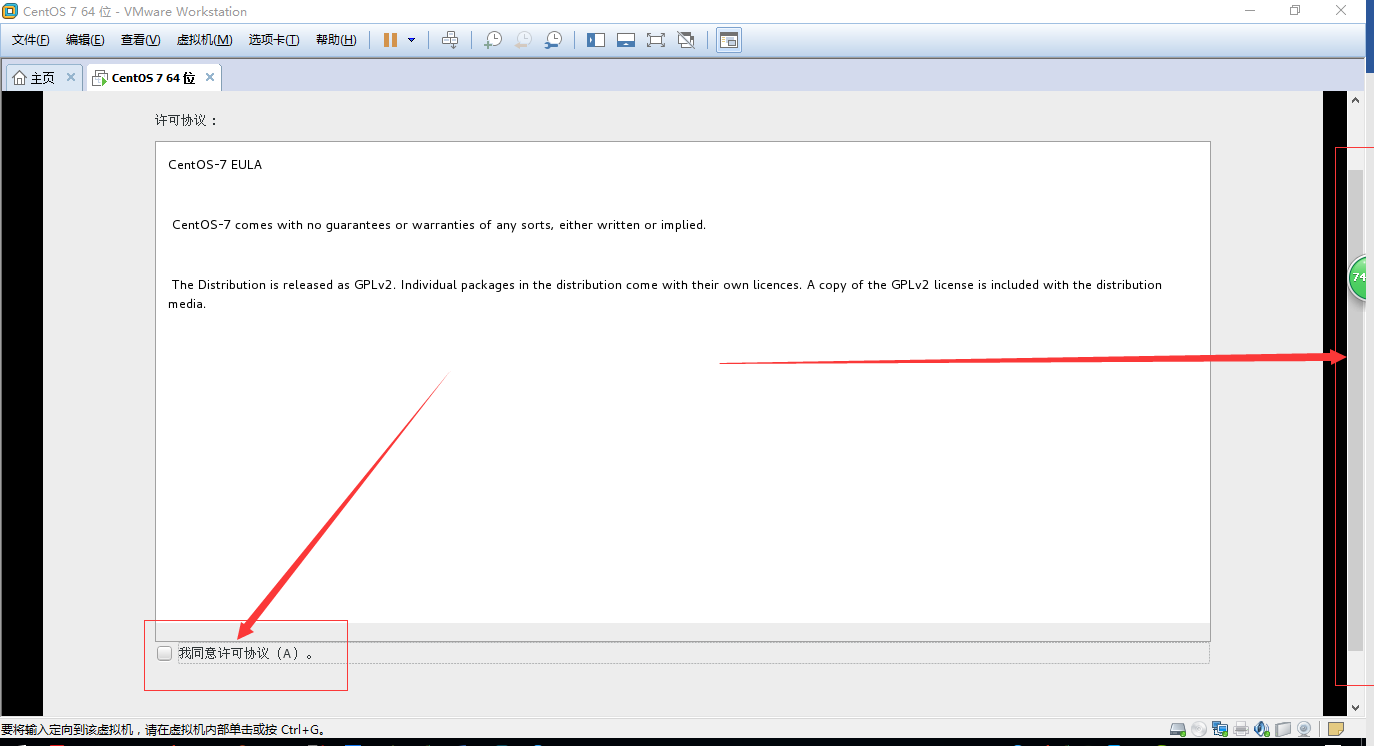
数据“全名”、“用户名”、“密码”、“确认密码”，如果你想将改用户设置为管理员可以选择上图中的“将此用户作为管理员”，然后双击“完成”按钮即可。然后就可以去喝一杯咖啡，静静的等待安装完成。然后会看到如下所示：



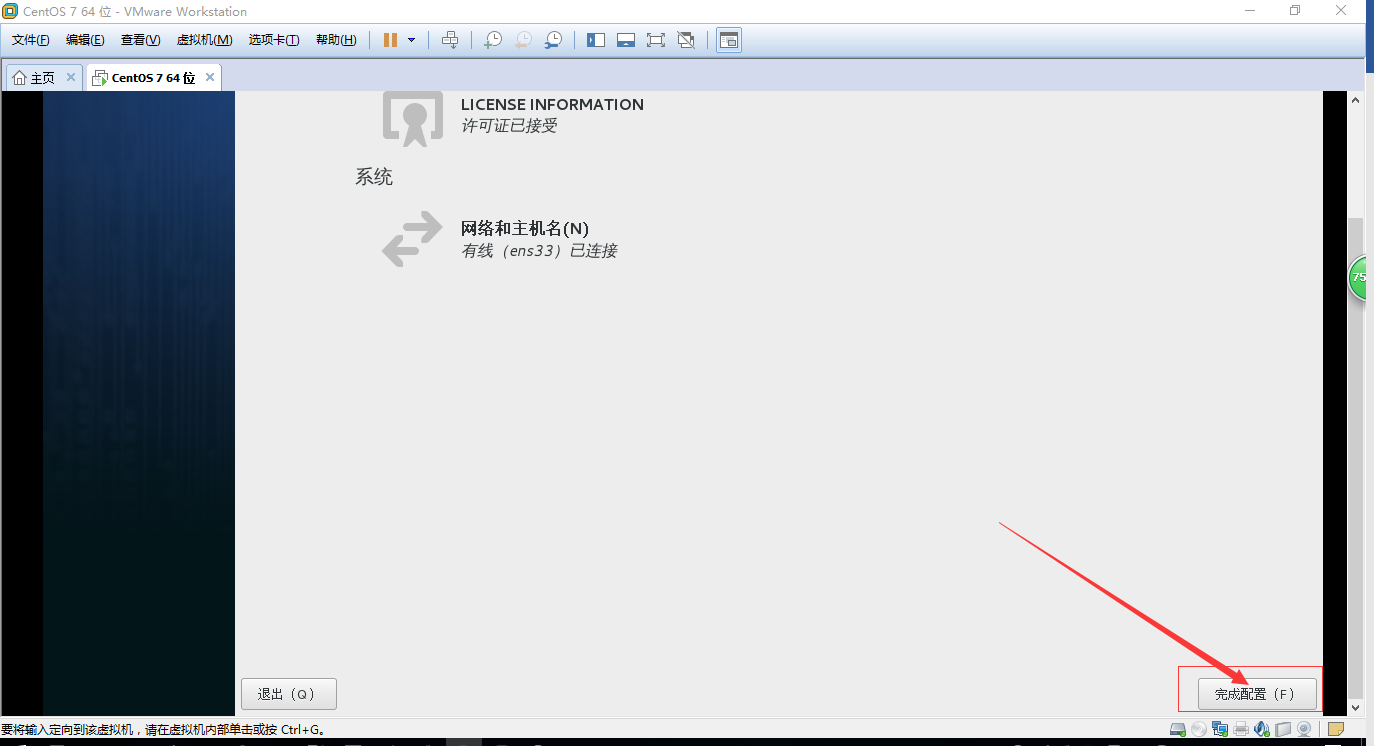
重启后会看到如下所示：



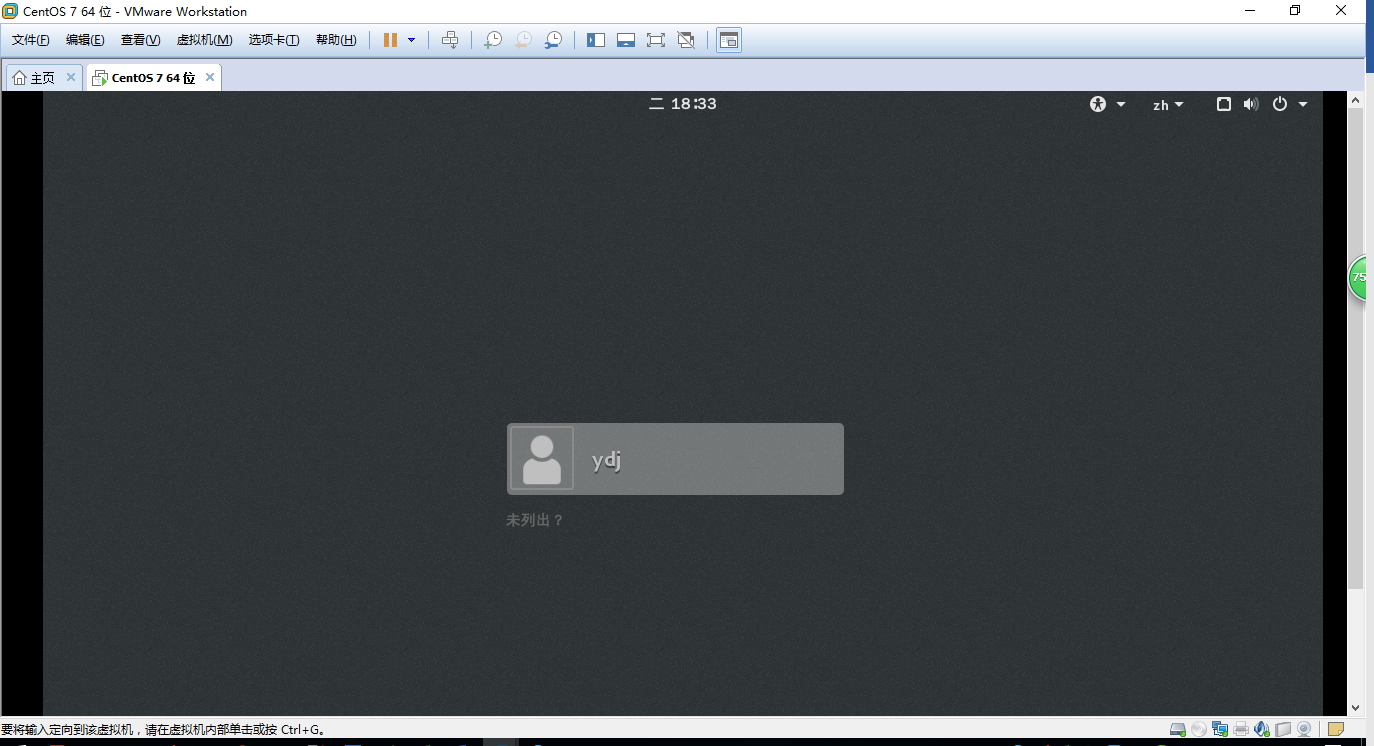
配置许可证，点击进去会看到如下所示：



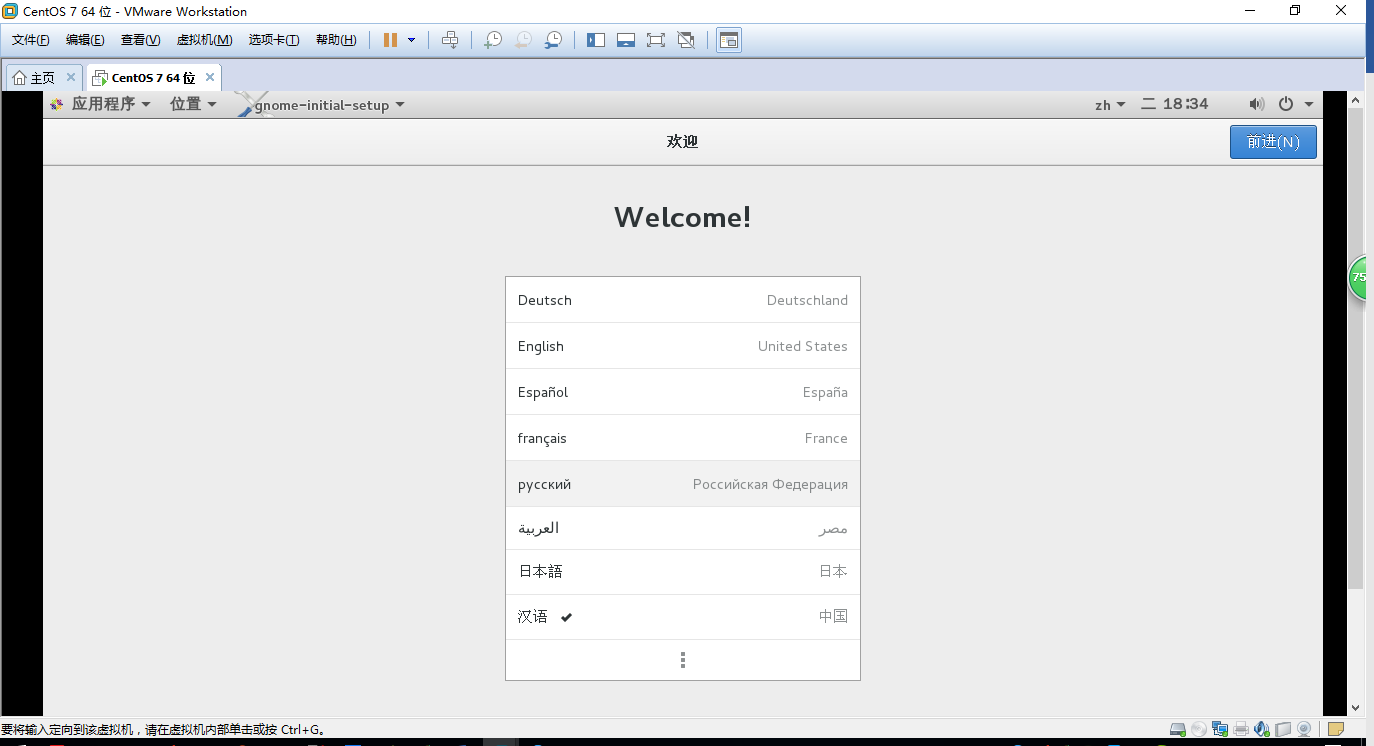
这个页面需要拖动右侧滚动条才能看到“我同意许可协议”，然后勾选上，然后完成即可，



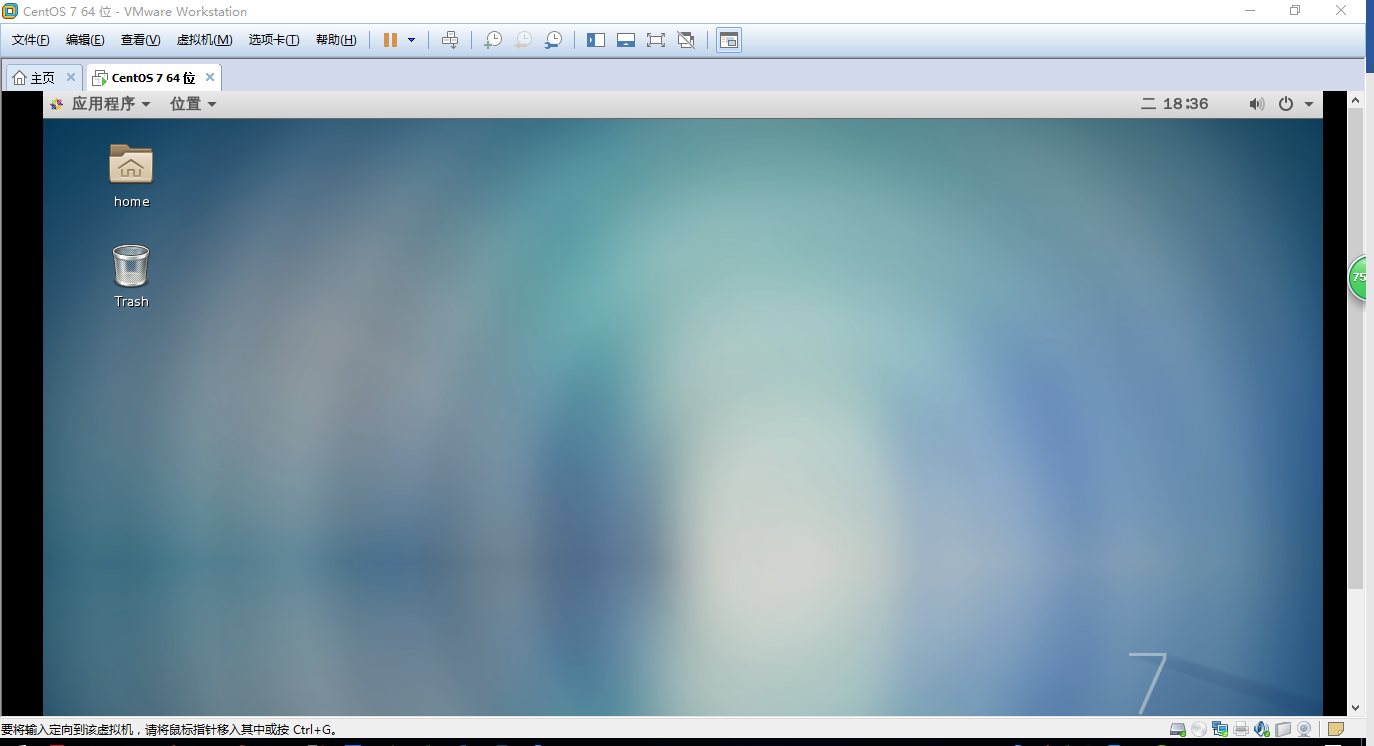
拖动右侧滚动条，点击右下角的“完成配置”，即可启动系统，然后就会到登录页面，如下图所示：



点击用户名输入密码回车即可登录到系统桌面，首次登录会看到如下所示：



根据提示信息，点击完成后即可愉快的开始使用Centos 系统啦，桌面如下图所示：



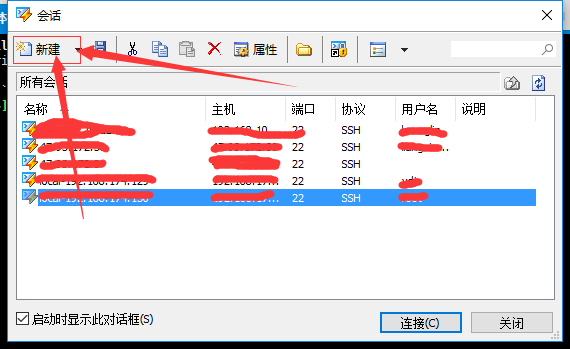
Centos 7 默认是安装了Vmware Tools的，可以点击如下所示，最大化窗口：

## 2.2 Xshell连接Linux服务器

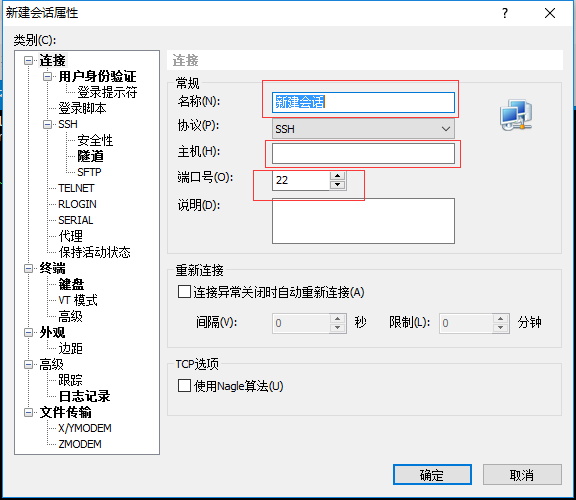
Xshell安装，最新版的Xshell，最近好像没办法使用免费版，可以下载一个老版本的xshell，我使用的版本如下图所示，此版本可以使用免费家庭版：



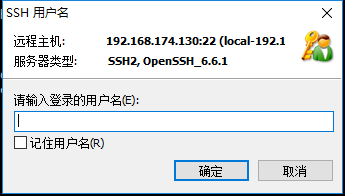
安装xshell之后启动会出现如下图所示：



点击“新建”会出现如下所示：



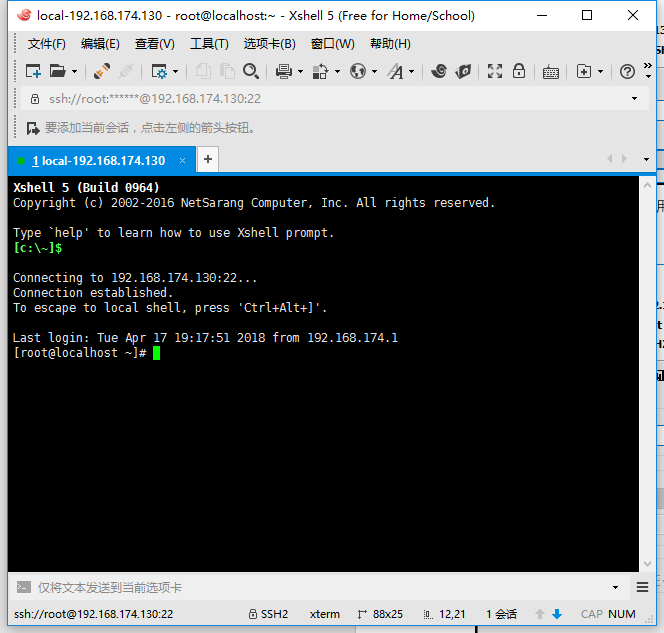
输入“名称(随便写你喜欢的)”，“主机(ip地址)”，端口(默认22)，点击“确定”后会出现如下所示：



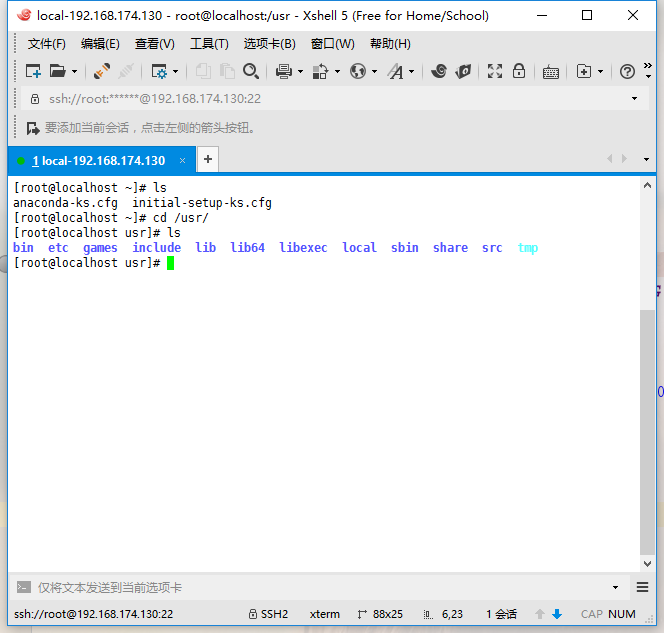
输入“用户名”，可以点击“记住用户名”，下次可以不用再次输入用户名，点击“确定”会出现如下所示：



输入“密码”，可以点击“记住密码”，下次登录时不用再次输入密码，点击确定后会就会连接成功，并出现如下所示：

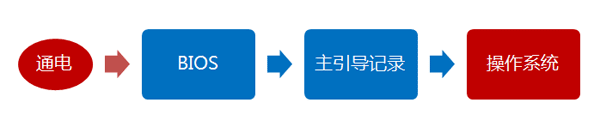


如果进入如上图所示页面表示连接成功，如果不喜欢这种黑色的窗口可以调节窗口的颜色，在页面中右键->配色方案->Pastel on White->确定，就会变成白色风格的主题，如下图所示：



# Part 3. Linux系统启动过程

## 3.1 Linux系统启动过程

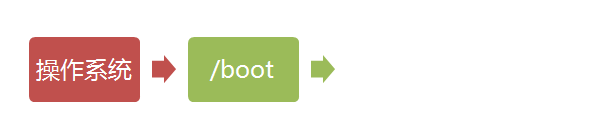


这个部分比较有意思。因为在BIOS阶段，计算机的行为基本上被写死了，程序员可以做的事情并不多；但是，一旦进入操作系统，程序员几乎可以定制所有方面。所以，这个部分与程序员的关系更密切。

我主要关心的是Linux操作系统，它是目前服务器端的主流操作系统。下面的内容针对的是[Debian](http://en.wikipedia.org/wiki/Debian)发行版，因为我对其他发行版不够熟悉。

**第一步、加载内核**

操作系统接管硬件以后，首先读入 /boot 目录下的内核文件。



以我的电脑为例，/boot 目录下面大概是这样一些文件：

　　$ ls /boot

　　config-3.2.0-3-amd64

　　config-3.2.0-4-amd64

　　grub

　　initrd.img-3.2.0-3-amd64

　　initrd.img-3.2.0-4-amd64

　　System.map-3.2.0-3-amd64

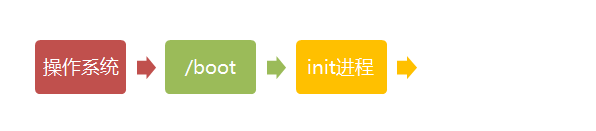
　　System.map-3.2.0-4-amd64

　　vmlinuz-3.2.0-3-amd64

　　vmlinuz-3.2.0-4-amd64

**第二步、启动初始化进程**

内核文件加载以后，就开始运行第一个程序 /sbin/init，它的作用是初始化系统环境。

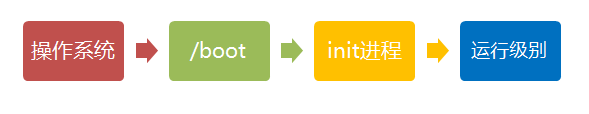


由于init是第一个运行的程序，它的进程编号（pid）就是1。其他所有进程都从它衍生，都是它的子进程。

**第三步、确定运行级别**

许多程序需要开机启动。它们在Windows叫做"服务"（service），在Linux就叫做"[守护进程](http://zh.wikipedia.org/wiki/%E5%AE%88%E6%8A%A4%E8%BF%9B%E7%A8%8B)"（daemon）。

init进程的一大任务，就是去运行这些开机启动的程序。但是，不同的场合需要启动不同的程序，比如用作服务器时，需要启动Apache，用作桌面就不需要。Linux允许为不同的场合，分配不同的开机启动程序，这就叫做"[运行级别](http://zh.wikipedia.org/wiki/%E8%BF%90%E8%A1%8C%E7%BA%A7%E5%88%AB)"（runlevel）。也就是说，启动时根据"运行级别"，确定要运行哪些程序。



Linux预置七种运行级别（0-6）。一般来说，0是关机，1是单用户模式（也就是维护模式），6是重启。运行级别2-5，各个发行版不太一样，对于Debian来说，都是同样的多用户模式（也就是正常模式）。

init进程首先读取文件 /etc/inittab，它是运行级别的设置文件。如果你打开它，可以看到第一行是这样的：

　　id:2:initdefault:

initdefault的值是2，表明系统启动时的运行级别为2。如果需要指定其他级别，可以手动修改这个值。

那么，运行级别2有些什么程序呢，系统怎么知道每个级别应该加载哪些程序呢？......回答是每个运行级别在/etc目录下面，都有一个对应的子目录，指定要加载的程序。

　　/etc/rc0.d

　　/etc/rc1.d

　　/etc/rc2.d

　　/etc/rc3.d

　　/etc/rc4.d

　　/etc/rc5.d

　　/etc/rc6.d

上面目录名中的"rc"，表示run command（运行程序），最后的d表示directory（目录）。下面让我们看看 /etc/rc2.d 目录中到底指定了哪些程序。

　　$ ls /etc/rc2.d

　　README

　　S01motd

　　S13rpcbind

　　S14nfs-common

　　S16binfmt-support

　　S16rsyslog

　　S16sudo

　　S17apache2

　　S18acpid

　　...

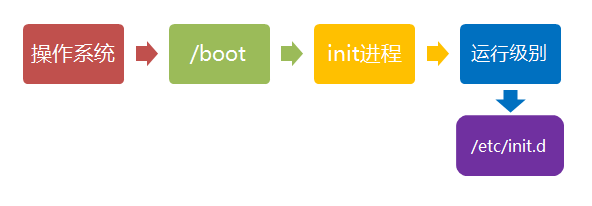
可以看到，除了第一个文件README以外，其他文件名都是"字母S+两位数字+程序名"的形式。字母S表示Start，也就是启动的意思（启动脚本的运行参数为start），如果这个位置是字母K，就代表Kill（关闭），即如果从其他运行级别切换过来，需要关闭的程序（启动脚本的运行参数为stop）。后面的两位数字表示处理顺序，数字越小越早处理，所以第一个启动的程序是motd，然后是rpcbing、nfs......数字相同时，则按照程序名的字母顺序启动，所以rsyslog会先于sudo启动。

这个目录里的所有文件（除了README），就是启动时要加载的程序。如果想增加或删除某些程序，不建议手动修改 /etc/rcN.d 目录，最好是用一些专门命令进行管理（参考[这里](http://www.debianadmin.com/manage-linux-init-or-startup-scripts.html)和[这里](http://www.debianadmin.com/remove-unwanted-startup-files-or-services-in-debian.html)）。

**第四步、加载开机启动程序**

前面提到，七种预设的"运行级别"各自有一个目录，存放需要开机启动的程序。不难想到，如果多个"运行级别"需要启动同一个程序，那么这个程序的启动脚本，就会在每一个目录里都有一个拷贝。这样会造成管理上的困扰：如果要修改启动脚本，岂不是每个目录都要改一遍？

Linux的解决办法，就是七个 /etc/rcN.d 目录里列出的程序，都设为链接文件，指向另外一个目录 /etc/init.d ，真正的启动脚本都统一放在这个目录中。init进程逐一加载开机启动程序，其实就是运行这个目录里的启动脚本。



下面就是链接文件真正的指向。

　　$ ls -l /etc/rc2.d

　　README

　　S01motd -> ../init.d/motd

　　S13rpcbind -> ../init.d/rpcbind

　　S14nfs-common -> ../init.d/nfs-common

　　S16binfmt-support -> ../init.d/binfmt-support

　　S16rsyslog -> ../init.d/rsyslog

　　S16sudo -> ../init.d/sudo

　　S17apache2 -> ../init.d/apache2

　　S18acpid -> ../init.d/acpid

　　...

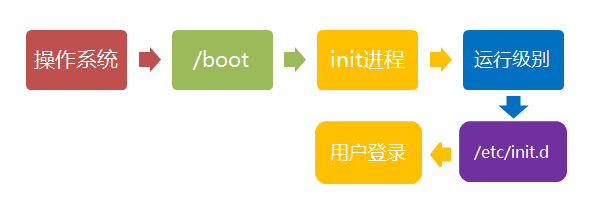
这样做的另一个好处，就是如果你要手动关闭或重启某个进程，直接到目录 /etc/init.d 中寻找启动脚本即可。比如，我要重启Apache服务器，就运行下面的命令：

　　$ sudo /etc/init.d/apache2 restart

/etc/init.d 这个目录名最后一个字母d，是directory的意思，表示这是一个目录，用来与程序 /etc/init 区分。

**第五步、用户登录**

开机启动程序加载完毕以后，就要让用户登录了。



一般来说，用户的登录方式有三种：

　　（1）命令行登录

　　（2）ssh登录

　　（3）图形界面登录

这三种情况，都有自己的方式对用户进行认证。

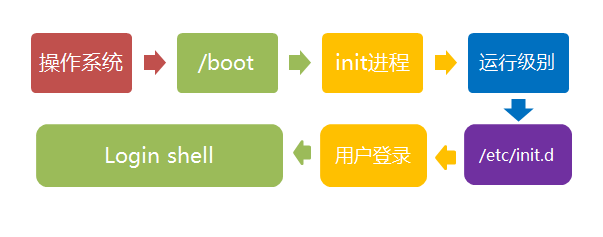
（1）命令行登录：init进程调用getty程序（意为get teletype），让用户输入用户名和密码。输入完成后，再调用login程序，核对密码（Debian还会再多运行一个身份核对程序/etc/pam.d/login）。如果密码正确，就从文件 /etc/passwd 读取该用户指定的shell，然后启动这个shell。

（2）ssh登录：这时系统调用sshd程序（Debian还会再运行/etc/pam.d/ssh ），取代getty和login，然后启动shell。

（3）图形界面登录：init进程调用显示管理器，Gnome图形界面对应的显示管理器为gdm（GNOME Display Manager），然后用户输入用户名和密码。如果密码正确，就读取/etc/gdm3/Xsession，启动用户的会话。

**第六步、进入 login shell**

所谓shell，简单说就是命令行界面，让用户可以直接与操作系统对话。用户登录时打开的shell，就叫做login shell。



Debian默认的shell是[Bash](http://zh.wikipedia.org/wiki/Bash)，它会读入一系列的配置文件。上一步的三种情况，在这一步的处理，也存在差异。

（1）命令行登录：首先读入 /etc/profile，这是对所有用户都有效的配置；然后依次寻找下面三个文件，这是针对当前用户的配置。

　　~/.bash\_profile

　　~/.bash\_login

　　~/.profile

需要注意的是，这三个文件只要有一个存在，就不再读入后面的文件了。比如，要是 ~/.bash\_profile 存在，就不会再读入后面两个文件了。

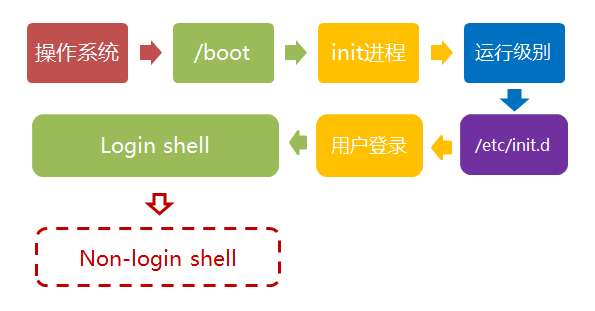
（2）ssh登录：与第一种情况完全相同。

（3）图形界面登录：只加载 /etc/profile 和 ~/.profile。也就是说，~/.bash\_profile 不管有没有，都不会运行。

**第七步，打开 non-login shell**

老实说，上一步完成以后，Linux的启动过程就算结束了，用户已经可以看到命令行提示符或者图形界面了。但是，为了内容的完整，必须再介绍一下这一步。

用户进入操作系统以后，常常会再手动开启一个shell。这个shell就叫做 non-login shell，意思是它不同于登录时出现的那个shell，不读取/etc/profile和.profile等配置文件。



non-login shell的重要性，不仅在于它是用户最常接触的那个shell，还在于它会读入用户自己的bash配置文件 ~/.bashrc。大多数时候，我们对于bash的定制，都是写在这个文件里面的。

你也许会问，要是不进入 non-login shell，岂不是.bashrc就不会运行了，因此bash 也就不能完成定制了？事实上，Debian已经考虑到这个问题了，请打开文件 ~/.profile，可以看到下面的代码：

　　if [ -n "$BASH\_VERSION" ]; then

　　　　if [ -f "$HOME/.bashrc" ]; then

　　　　　　. "$HOME/.bashrc"

　　　　fi

　　fi

上面代码先判断变量 $BASH\_VERSION 是否有值，然后判断主目录下是否存在 .bashrc 文件，如果存在就运行该文件。第三行开头的那个点，是source命令的简写形式，表示运行某个文件，写成"source ~/.bashrc"也是可以的。

因此，只要运行～/.profile文件，～/.bashrc文件就会连带运行。但是上一节的第一种情况提到过，如果存在～/.bash\_profile文件，那么有可能不会运行～/.profile文件。解决这个问题很简单，把下面代码写入.bash\_profile就行了。

　　if [ -f ~/.profile ]; then

　　　　. ~/.profile

　　fi

这样一来，不管是哪种情况，.bashrc都会执行，用户的设置可以放心地都写入这个文件了。

Bash的设置之所以如此繁琐，是由于历史原因造成的。早期的时候，计算机运行速度很慢，载入配置文件需要很长时间，Bash的作者只好把配置文件分成了几个部分，阶段性载入。系统的通用设置放在 /etc/profile，用户个人的、需要被所有子进程继承的设置放在.profile，不需要被继承的设置放在.bashrc。

顺便提一下，除了Linux以外， Mac OS X 使用的shell也是Bash。但是，它只加载.bash\_profile，然后在.bash\_profile里面调用.bashrc。而且，不管是ssh登录，还是在图形界面里启动shell窗口，都是如此。

# Part 4. Linux系统目录结构

## 4.1 Linux系统目录结构