



Linux

Linux进行ADS通讯入门

-暨Ubuntu使用入门

Version 1.00

2020年12月



ubuntu

前言

本书内容的架构安排如下：

第一章介绍了 Linux 概述

第二章介绍了 Linux 的安装

第三章介绍了 Ubuntu 的基本配置

第四章介绍了 Linux 的 shell 基本命令

第五章介绍了 Ubuntu 的网络配置

第六章介绍了 远程登录 Ubuntu 与向其传输文件

第七章介绍了 在 Linux 系统下使用 Python 与 TwinCAT ADS 通讯

V1.0→V1.0 更新内容：

第 X 章

- 更新说明

本书所有的内容都会不间断更新，如果想获取更新的教材可以通过访问虚拟学院获取到，当然本书所有配套的案例程序也会在此虚拟学院中供所有读者免费获取。

虚拟学院地址：<https://tr.beckhoff.com.cn/>

欢迎对本书的结构、内容提出意见和建议，请发邮件至：

gc.feng@beckhoff.com.cn

2020 年 12 月 12 日

目录

一、	Linux 概述	4
1	什么是 Linux	4
2	Linux 的发行版本	4
2.1	不同的发行版本	4
2.2	本书选择的发行版本	4
二、	Linux 安装	6
1	安装前的准备工作	6
1.1	获取 Linux 及硬件要求	6
1.2	硬件要求	6
2	虚拟机的使用	6
2.1	虚拟机的软件：VMware player	7
2.2	在 VMware player 中安装 Linux	9
三、	Ubuntu 的基本配置	13
1	超级用户 root	13
1.1	root 可以做什么	13
1.2	账户和系统安全问题	13
1.3	Ubuntu 的 root 用户	14
2	中文支持	15
四、	Shell 基本命令	19
1	Shell 简介	19
2	印刷约定	19
3	快速上手：浏览硬盘	20
4	快速上手：使用命令行补全和通配符	21
5	查看目录和文件	22
5.1	显示当前目录：pwd	23
5.2	改变目录：cd	23
5.3	列出目录内容：ls	23
5.4	列出目录内容：dir 和 vdir	25
6	建立文件和目录	26
6.1	建立目录：mkdir	26
6.2	建立一个空文件：touch	26
7	移动、复制和删除	27
7.1	移动和重命名：mv	27

本文纯属个人经验，非 Beckhoff 公司正式发行，禁止一切商用行为。如有疑问，请直接联系作者。

7.2	复制文件和目录: cp.....	27
7.3	删除目录和文件: rmdir 和 rm	27
8	VIM 编辑器.....	28
8.1	编辑和保存.....	28
9	软件包管理.....	30
9.1	高级软件包工具: apt.....	30
9.2	下载和安装软件包.....	30
五、	网络配置	32
1	获取 IP 地址.....	32
2	防火墙设置.....	33
2.1	Linux 防火墙基本概念	33
2.2	初始化防火墙设置.....	34
2.3	添加链规则.....	35
2.4	删除链规则.....	37
六、	远程登录和传输文件	38
1	安装 OpenSSH.....	38
2	从 Windows 登录这台 Linux (PuTTY)	38
3	传输文件 (WinSCP)	39
七、	ADS 通讯.....	41
1	在 Linux 中配置 Python 及其组件	41
1.1	安装 Python	41
1.2	安装 build 依赖包.....	41
1.3	安装 pip3.....	42
1.4	安装 pyads	42
1.5	关闭链规则.....	43
2	服务器版 Ubuntu 进行 ADS 通讯	43
2.1	软硬件版本.....	43
2.2	操作步骤.....	44
3	桌面版 Ubuntu 进行 ADS 通讯	47
3.1	软硬件版本.....	47
3.2	操作步骤.....	48
4	常见问题及解决方法	51

一、Linux 概述

1 什么是 Linux

简单地说，Linux 和 Windows 一样，它是一种系统，可以安装在包括服务器、个人电脑、乃至 PDA、手机、打印机等各类设备中。

2 Linux 的发行版本

Linux 实际上只定义了一个操作系统的内核，这个内核由 kernel.org 负责维护。不同的企业和组织在此基础上开发上一系列辅助软件，打包发布自己的“发行版本”。各种发行版本可以“非常不同”，但却是建立在同一个基础之上。

2.1 不同的发行版本

Linux 的发行版本非常多，下表只列出了其中比较著名的一些版本（这些发行版本是按照字母顺序排列，而不是推荐或者流程度度排列的）。

发行版本	官方网站	说明
CentOS	www.centos.org	模仿 Red Hat Enterprise Linux 的非商业发行版本
Debian	www.debian.org	免费的非商业发行版本
Fedora	Fedoraproject.org	Red Hat 公司赞助的社区项目免费发行版
Gentoo	www.gentoo.org	基于源代码编译的发行版本
Mandriva	www.mandriva.org	前身 Mandrakelinux，第一个为非技术类用户设计的 Linux 发行版本
OpenSUSE	www.opensuse.org	SUSE Linux 的免费发行版本
Rea Flag	www.redflag-linux.com	国内发展最好的 Linux 发行版本
Red Hat Enterprise	www.redhat.com	Red Hat 公司的企业级商业化发行版本
SUSE Linux Enterprise	www.suse.com/linux	Novell 公司的企业级商业化发行版本
TurboLinux	www.turbolinux.com	在中国和日本取得较大成功的发行版本
Ubuntu	www.ubuntu.com	类似于 Debian 的免费发行版本

2.2 本书选择的发行版本

众多的 Linux 发行版本的确丰富了 Linux 世界，但是也给了所有 Linux 的初学者出了一个大难题，就是究竟选择哪个发行版本来进行学习。

通过经验总结，本书选择了：Ubuntu Linux。不仅因为 Ubuntu 是目前 Linux 桌面市场占有率较高的发行版本，更重要的是，Ubuntu 是可以有桌面的系统，更容易让初学者上手。

本文纯属个人经验，非 Beckhoff 公司正式发行，禁止一切商用行为。如有疑问，请直接联系作者。



二、Linux 安装

1 安装前的准备工作

在安装这个全新的操作系统之前，需要做一些准备工作。比如，从哪里得到 Linux？对电脑配置有什么要求？安装会删除机器上原有的 Windows 吗？……这一系列问题，本节会逐一给予回答。

1.1 获取 Linux 及硬件要求

使用 Linux 本身不需要支付任何费用。读者可以在各个 Linux 发行版的官方网站上找到安装镜像。

以 Ubuntu 为例，进入它的官网下载地址：

<https://ubuntu.com/download/alternative-downloads>

本书下载的版本为：Ubuntu 18.04.5 Desktop (64-bit)

将来 Ubuntu 也肯定有新的版本出现，但是它们基于的内核是一样的，所以在基本指令和用法上也是一样的，读者不用纠结去下载哪个版本。

1.2 硬件要求

具体的硬件要求取决于所使用的发行版。本文后续将以 Ubuntu 18.04.5 Desktop (64-bit) 为例进行使用。

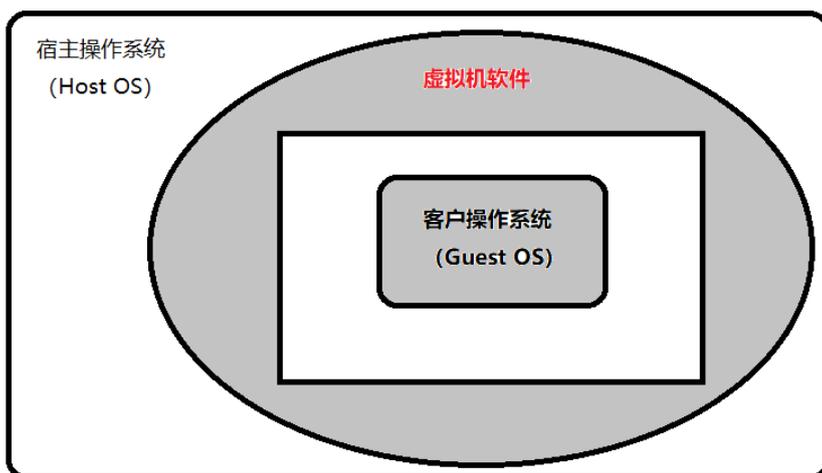
此版本的硬件要求：

CPU	2GHz 双核处理器或以上
内存	4G 内存
硬盘	25G 硬盘空间

根据此硬件需求，来选择对应电脑或虚拟机配置。

2 虚拟机的使用

虚拟机是一个非常方便以及学习系统的软件：它本身安装在一个操作系统中，却可以虚拟出整个硬件环境。在这个虚拟出来的硬件环境中，可以安装另一个操作系统。对于这两个操作系统，前者被称为宿主操作系统（Host OS），后者被称为客户操作系统（Guest OS）。如图所示。



使用虚拟机的优点在于,对客户操作系统的任何操作都不会对实际的硬件系统产生不良影响,因为其所依赖的硬件环境都是“虚拟”出来的。最终反映在硬盘上的,只是一系列文件。

虚拟机在服务器端拥有更广泛的应用。由于在控制成本、利用资源等方面展现出的巨大作用,虚拟机技术在最近几年获得了长足的进步。VMware、Sun、Microsoft 等公司都纷纷推出了自己的虚拟机产品。Intel 等芯片厂商也在 CPU 级别上提供了对虚拟化技术的支持。

2.1 虚拟机的软件：VMware player

对于 PC 用户而言,最常用到的虚拟机软件是 VMware。这款虚拟机产品可以在包括 Windows 和 Linux 在内的多个平台上运行。VMware 面向企业和个人开发了多个版本,其中一些需要用户购买许可证,例如 VMware workstation 等。另外一些,例如 VMware player,在仅非商业用途上是可以免费使用的。

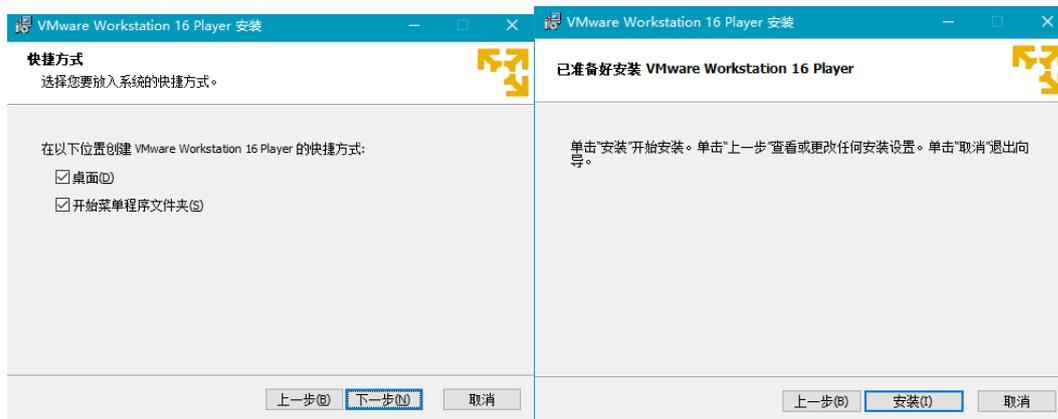
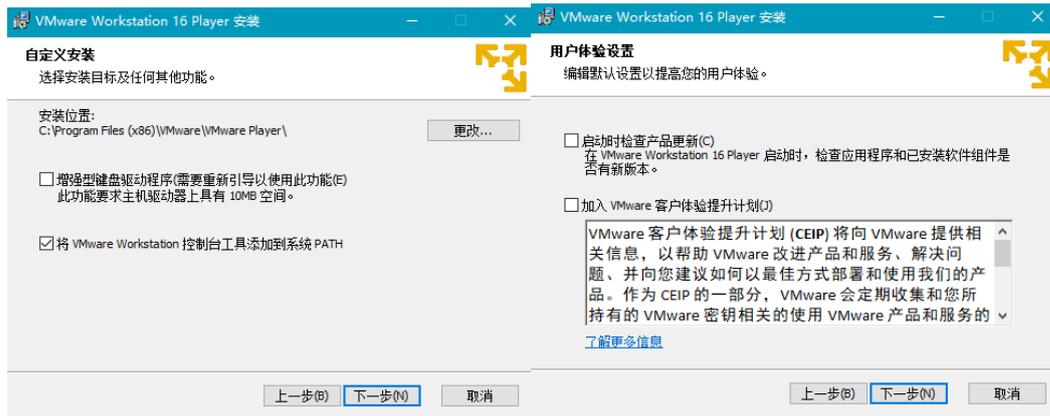
推荐读者使用 VMware player,所以这里介绍一下 VMware player 在 Windows 下的安装和使用方法。

VMware player 可以从 <https://www.vmware.com/cn/products/workstation-player/workstation-player-evaluation.html> 上下载。

本书安装的版本为: VMware-player-16.0.0-16894299.exe



本文纯属个人经验，非 Beckhoff 公司正式发行，禁止一切商用行为。如有疑问，请直接联系作者。



安装完成后，首次打开 VMware player，可以进行选择输入许可证或者免费用于非商业用途，之后就能使用虚拟机安装各类操作系统了。





2.2 在 VMware player 中安装 Linux

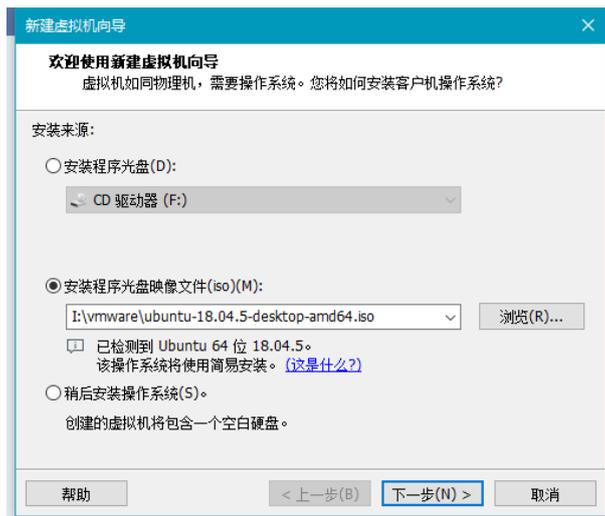
准备工作完成之后，就可以着手将 Linux 进行安装了。如今的 Linux 的安装过程已经非常简单了，只需轻点几下鼠标，就能够完成整个系统的安装。

1. 创建新虚拟机



2. 选择在官网下载到的 Ubuntu 系统的 ISO 文件，VMware 会自动检测文件中的系统内容，并将该系统使用简易安装

本文纯属个人经验，非 Beckhoff 公司正式发行，禁止一切商用行为。如有疑问，请直接联系作者。



3. 与 Windows 一样，Ubuntu 也需要设置其全名，用户名和密码。（因为使用了简易安装，所以这些配置都是在 VMware 中进行配置的）

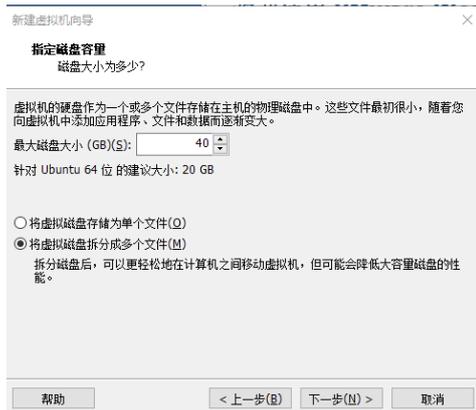


4. 设置该虚拟机的名称，以及设置虚拟机的磁盘文件所在的位置。



5. 根据以上找到的硬件要求，此版本的 Ubuntu 需要设置大于 25G 硬盘容量。

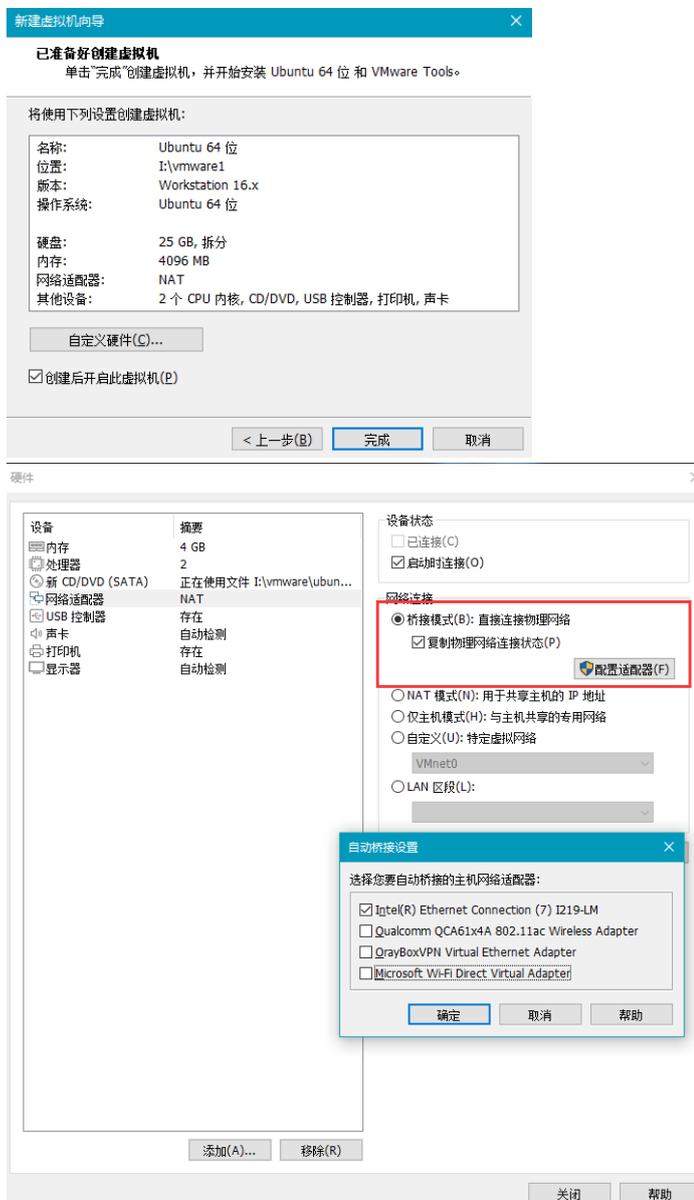
本文纯属个人经验，非 Beckhoff 公司正式发行，禁止一切商用行为。如有疑问，请直接联系作者。



6. 点击自定义硬件信息:

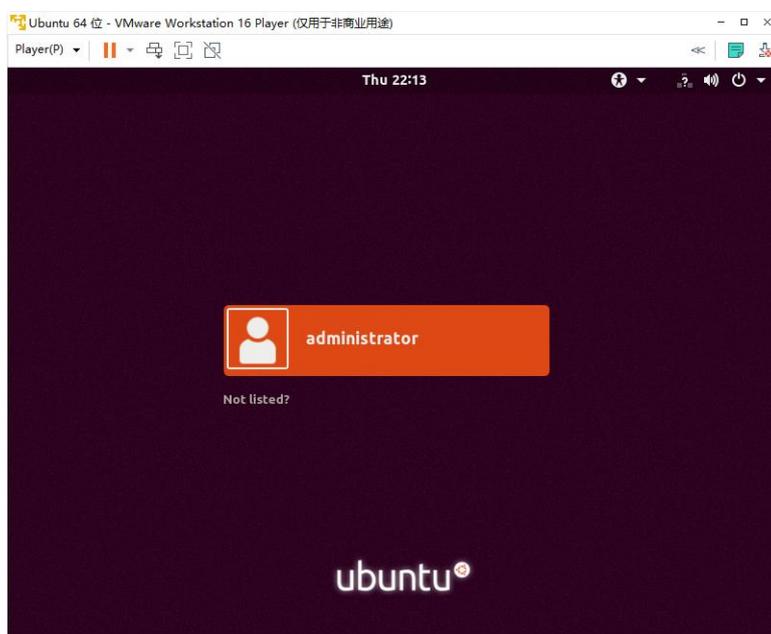
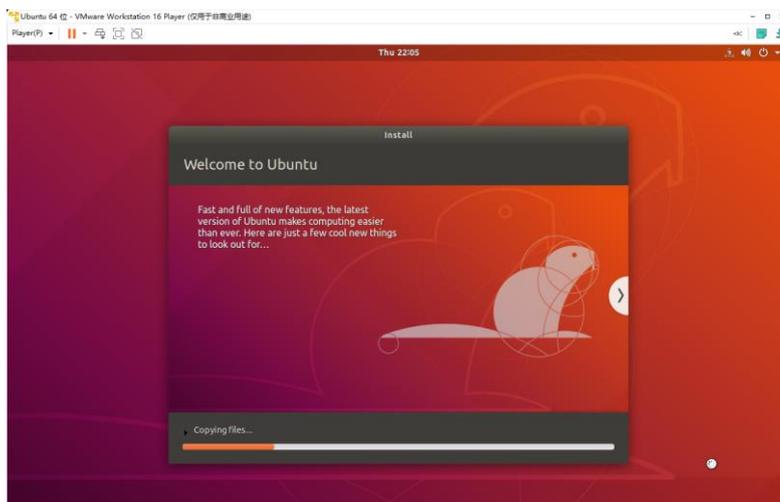
内存设置为 4G

网络适配器推荐设置为桥接模式，桥接有线网卡并复制物理网络连接状态（方便以后与 Twincat 进行 ADS 通讯）



7. 点击完成之后，VMware player 会自动配置 Ubuntu，仅需等待其自动完成即可。

本文纯属个人经验，非 Beckhoff 公司正式发行，禁止一切商用行为。如有疑问，请直接联系作者。



三、Ubuntu 的基本配置

1 超级用户 root

root 用户非常重要，Linux 系统设置都需要使用 root 用户来完成。root 从字面上解释是“根”的意思，所以超级用户也被称为根用户。从某种意义上它相当于 Windows 下的 Administrator 用户。

1.1 root 可以做什么

root 用户可以在 Linux 系统做任何事。root 可以访问、修改、删除系统中的任何文件和目录。另外，对于如下受限的操作，一般只有 root 用户才能够执行。

- a) 添加删除用户；
- b) 安装软件；
- c) 添加删除设备；
- d) 启动和停止网络服务；
- e) 某些系统调用（例如对内核的请求）；
- f) 关闭系统

Linux 系统上的每个文件和目录都属于某个特定的用户，没有得到许可，其他用户就不能访问这些对象。但 root 用户却可以访问所有用户的文件，就像使用自己的东西一样。因此，拥有 root 口令意味着更多的责任，特别是在一台多人协作的服务器上。

1.2 账户和系统安全问题

root 用户可以在系统上做任何事情。那么保证安全性就显得尤为重要。系统不会自己去判断命令是否合理而去执行。相反地，系统会乐滋滋地去执行这样一条命令，然后就把自己完完整整地删除了。

```
$ rm -fr /*          ## 删除根目录下所有的文件和目录
```

另外，妥善保管 root 口令也至关重要。因为任何得到 root 口令的人都能够完全控制系统。root 口令应该至少为 8 个字符，因为 7 个以下字符的密码特别容易被破解。从理论上讲，最安全的口令应该是由字母、标点符号和数字组成的足够长的随机序列。但是这样的密码往往难以记忆，如果为了使用这样所谓“最安全”的密码而不得不把它写在纸上，那么这是得不偿失的。一个比较好的建议是，使用拼音组成的一句话并穿插标点和数字。比如 jintian, qinglang（今天，晴朗），woshiyongUbuntu18.04.5（我使用 Ubuntu18.04.5）等都是不错的口令。

和普通用户一样，root 账户可以直接用来登录系统。但是一般不会这么去登录，因为任何一项误操作都有可能造成灾难性的后果，所以应该仅在必要的时候才使用 root 账户。

那么登陆了普通用户，又想要修改某些内容，却没有权限该怎么办呢？

用户可以执行不带参数的 su 命令将自己提升为 root 权限（需要输入 root 口令才能提升）。

```
root@ubuntu: /home/administrator
File Edit View Search Terminal Help
administrator@ubuntu:~$ su
Password:
root@ubuntu: /home/administrator#
```

另外一个方法可以用命令行工具，`sudo`，它可以临时使用 `root` 身份运行一个程序，并在程序执行完毕后返回普通用户状态。

```
administrator@ubuntu: ~
File Edit View Search Terminal Help
administrator@ubuntu:~$ sudo apt-get update
[sudo] password for administrator:
Hit:1 http://us.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu bionic-backports InRelease
Reading package lists... Done
administrator@ubuntu:~$
```

1.3 Ubuntu 的 root 用户

对于绝大多数的 Linux 发行版而言，安装的最后一步会要求设置两个用户口令：一个是 `root` 用户另一个是用户登录系统的普通用户。而对于 Ubuntu 而言，它只让你设置了一个普通用户（可以看上一章安装 Ubuntu，并没有强制设置 `root` 用户），而没有 `root`。实际上，这个在安装过程中设置的普通用户账户，在某种程度上充当了 `root`。平时，这个账户安分守己地做自己分内的事，没有任何特殊权限。在需要 `root` 的时候，则可以使用 `sudo` 命令来运行相关程序。`sudo` 命令运行时会要求输入口令，这个口令就是该普通账户的口令。

如果在 Ubuntu 中再建立一个用户，这个用户是不是也能够使用 `sudo` 命令呢？答案是否定的。`sudo` 通过读取 `/etc/sudoers` 来确定用户是否可以执行相关命令。这个文件默认需要有 `root` 权限才能修改。

上一节说过可以用 `su` 将自己提升成 `root` 用户，Ubuntu 中可以使用 `sudo` 的 `-s` 选项来进行提升。

```
root@ubuntu: ~
File Edit View Search Terminal Help
administrator@ubuntu:~$ sudo -s
[sudo] password for administrator:
root@ubuntu:~#
```

这两者虽然都是提升权限的，但是它们还是略有区别的。

`su` 输入的是 `root` 密码，`sudo -s` 输入的是普通用户密码

Ubuntu 默认没有 `root` 用户，那么它就没有密码了，所以一开始是用不了 `su` 的，要用 `su` 的话需要先给 `root` 用户创建密码。

本文纯属个人经验，非 Beckhoff 公司正式发行，禁止一切商用行为。如有疑问，请直接联系作者。

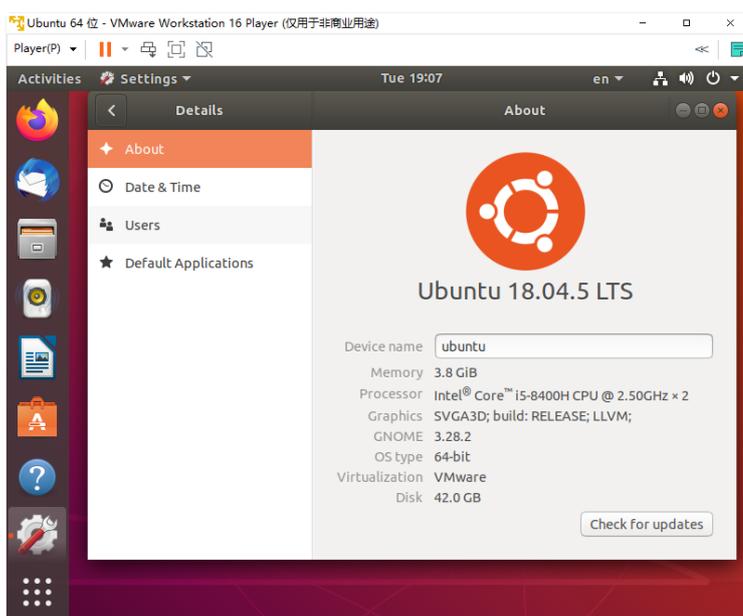
```
root@ubuntu: /home/administrator
File Edit View Search Terminal Help
administrator@ubuntu:~$ sudo passwd root
[sudo] password for administrator:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
administrator@ubuntu:~$ su
Password:
root@ubuntu: /home/administrator#
```

2 中文支持

在安装一个软件之后，总会想着这个软件是否有中文版，因为直接看中文会更加快捷，更加容易上手。Ubuntu 也有中文语言包可以进行安装。

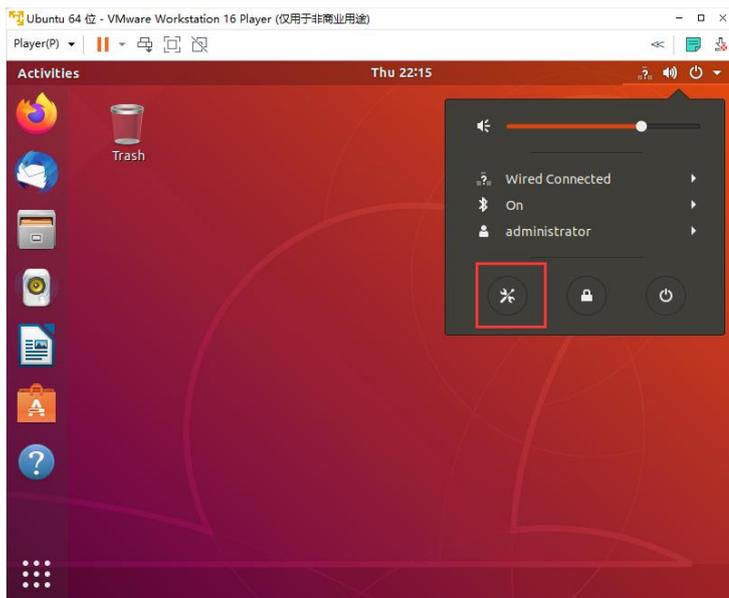
中文语言包虽然是自动下载的，但是首先还是要确保计算机连上网。Ubuntu 的下载服务器在外国，所以下载的时候特别慢。

这里以 Ubuntu18.04.5LTS 为例对汉化操作进行讲解

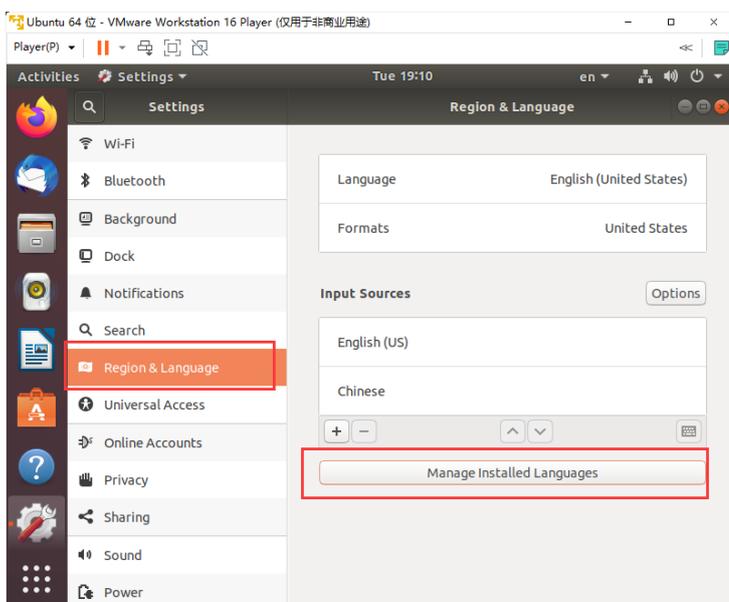


1. 点击右上角电源按钮，展开后选择工具按钮

本文纯属个人经验，非 Beckhoff 公司正式发行，禁止一切商用行为。如有疑问，请直接联系作者。

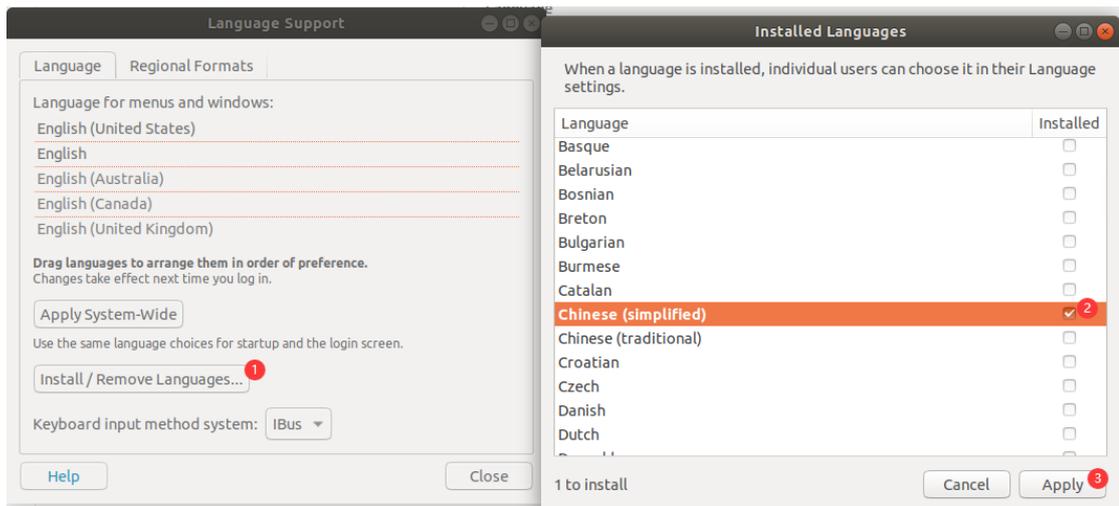


2. 选中 region & language 并点击 manage installed languages

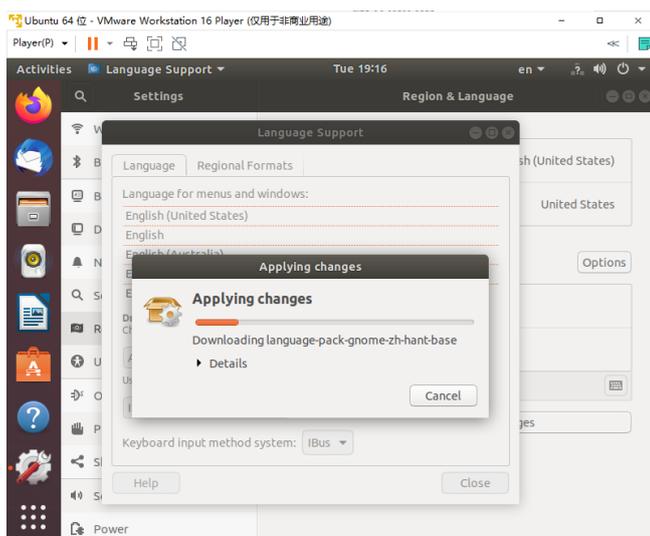
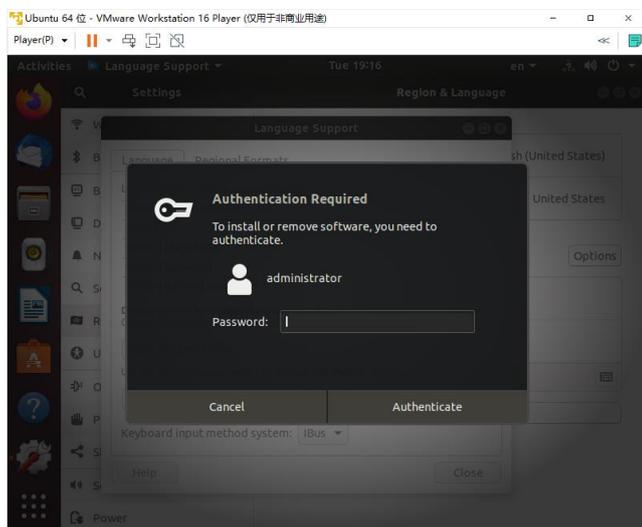


3. 点击 install/remove languages, 选中 Chinese, 点击 apply

本文纯属个人经验，非 Beckhoff 公司正式发行，禁止一切商用行为。如有疑问，请直接联系作者。

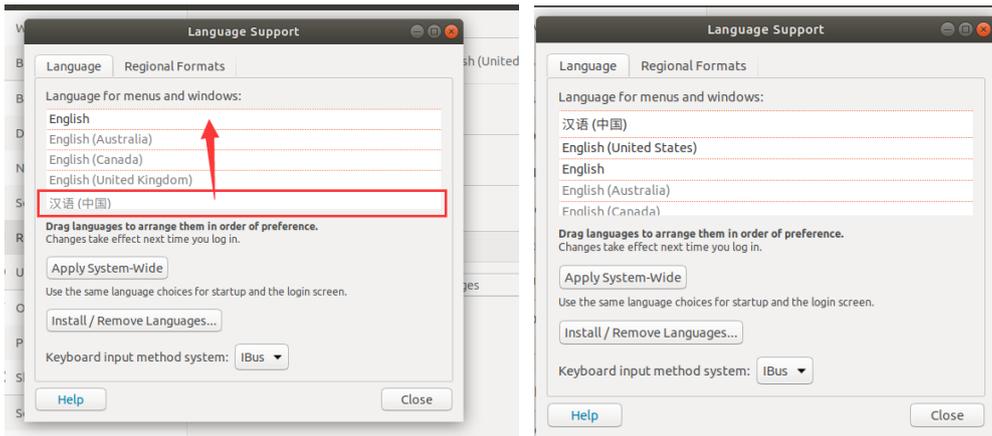


4. 输入密码，并等待下载

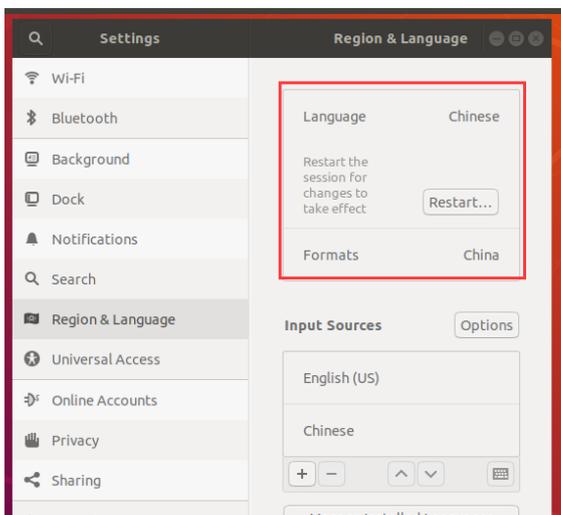


5. 下载完成后会回到该界面，将汉语拖至最上方

本文纯属个人经验，非 Beckhoff 公司正式发行，禁止一切商用行为。如有疑问，请直接联系作者。



6. 选择 language 为 Chinese, formats 为 china, 然后重启一下计算机



7. 最后就得到了一个汉化的 Ubuntu 系统了



四、Shell 基本命令

1 Shell 简介

一直以来 Shell 以其稳定、高效和灵活称为系统管理员的首选。命令行和 Shell 这两个概念常常是令人困惑的。在很多并不正式场合，这两个名词代表着相同的概念，即命令解释器。

然而从严格意义上，命令行指的是供用户输入命令的界面，其本身只是接收输入，然后把命令传递给命令解释器。后者就是 Shell。

从本质上讲，Shell 是一个程序，它在用户和操作系统之间提供了一个面向行的可交互接口。用户在命令行中输入命令，运行在后台的 Shell 把命令转换成指令代码发送给操作系统。Shell 提供了很多高级特性，使得用户和操作系统之间的交互变得简便和高效。

目前，在 Linux 环境下有几种不同类型的 Shell，常用的有 Bourne Again Shell (BASH)、TCSH Shell、Z-Shell 等。不同的 Shell 提供不尽相同的语法和特性，用户可以使用任何一种 Shell。在 Linux 上，BASH Shell 是默认安装和使用的 Shell。

2 印刷约定

Linux 命令行界面有一个输入行，用于输入命令。在 BASH 中，命令行以一个美元符号“\$”作为提示符，表示用户可以输入命令了。

如果正在以 root 身份执行命令，那么 Shell 提示符将称为“#”。

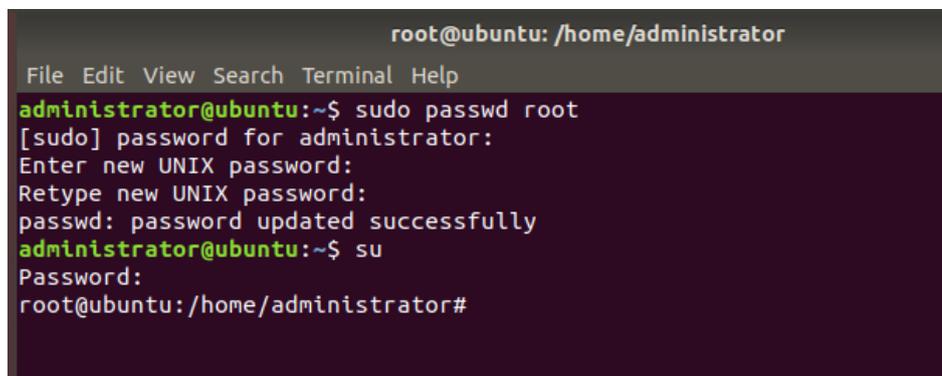
本文中的命令将以“提示符+命令+注释”的形式给出。例如：

```
$ sudo apt-get update      ##获取更新
```

“##”后面的文字是注释。在实际使用的时候，只需要输入命令部分即可。这里需要提醒的是，Linux 的命令和文件名都是区分大小写的，也就是说，SUDO 和 sudo 是不一样的。

现在重新看上一章的命令，在 BASH 的美元提示符前，一般还会有一段信息，包括用户名、主机名和当时目录。

比如：root@ubuntu:/home/administrator#

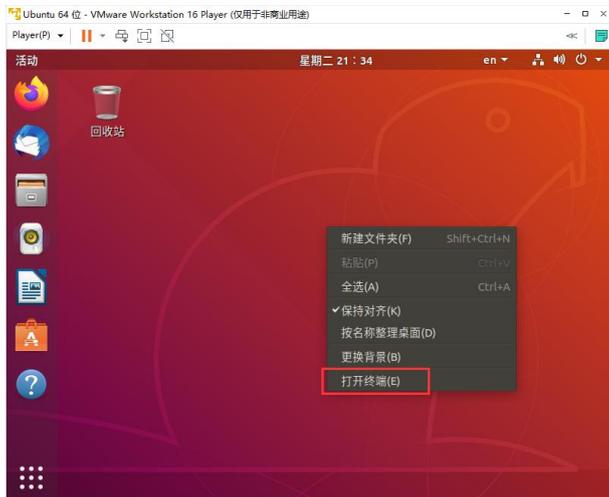


```
root@ubuntu: /home/administrator
File Edit View Search Terminal Help
administrator@ubuntu:~$ sudo passwd root
[sudo] password for administrator:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
administrator@ubuntu:~$ su
Password:
root@ubuntu:/home/administrator#
```

3 快速上手：浏览硬盘

本节将带领读者浏览 Ubuntu 中的文件系统。

首先，右键桌面打开终端，进入根目录看看里边有什么。



可以看到，Linux 安装完毕后自动在根目录下生成了大量目录和文件。

下面选择 home 进入。这个目录中存放着系统中所有用户的主目录。主目录的名字就是用户名。目前在此系统中，共有一个用户，叫做 administrator，是一开始创建的。



接着到/etc 目录中看一下。这个目录存放着系统以及绝大部分应用软件和配置文件（这里只列出了其中的一部分）。和 Windows 不同，Linux 使用纯文本文件来配置软件。修改配置文件就可以很方便地对软件进行订制。

```
administrator@ubuntu: /etc
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
administrator@ubuntu:/home$ cd /etc
administrator@ubuntu:/etc$ ls
acpi                hosts.allow         ppp
adduser.conf        hosts.deny          profile
alternatives        hp                  profile.d
anacrontab          ifplugd            protocols
apg.conf            ImageMagick-6      pulse
apm                 init                python2.7
apparmor            init.d              python3
apparmor.d          initramfs-tools    python3.6
appport             inputrc             rc0.d
appstream.conf      inserv.conf.d      rc1.d
apt                 iproute2            rc2.d
avahi               issue               rc3.d
bash.bashrc         issue.net           rc4.d
bash_completion     kernel              rc5.d
bash_completion.d   kernel-img.conf     rc6.d
bindresvport.blacklist kernelloops.conf    rc.local
```

4 快速上手：使用命令行补全和通配符

文件名是命令中最为常见的参数，然而每次完整输入文件名是一件很麻烦的事情，特别是当文件名还特别长的时候。BASH 提供了这样一种特性——命令行补全。在输入文件名的时候，只需要输入前面几个字符，然后按下 TAB 键，Shell 就会自动把文件名补全。

例如

```
$ cat fs<TAB>          ## <TAB>表示按下 TAB 键
```

Shell 会自动将其补全为:

```
$ cat fstab
```

如果以已键入的字符开头的文件不止一个，那么可以连续按两下 TAB，Shell 会以列表的形式给出所有该字符开头的文件，例如：

```
administrator@ubuntu:/etc$ cat b
bash.bashrc          bindresvport.blacklist  brlapi.key
bash_completion      binfmt.d/               brltty/
bash_completion.d/   bluetooth/               brltty.conf
administrator@ubuntu:/etc$
```

实际上，命令行补全也适用于所有 Linux 命令。例如输入 cat 并按两下 TAB 键。

```
administrator@ubuntu:/etc$ ca
cal                  canberra-gtk-play      case                    cautious-launcher
calendar/           cancel                  cat                    cat
calibrate_ppa       capsh                   catchsegv              catman
caller              captinfo
```

补充：因为系统命令本质上就是一些可执行文件，可以在 /usr/bin 目录下找到。所以说命令补全和文件名补全是同一回事。

另外，Shell 有一套被称作为通配符的专用符号，它们是“*”、“?”和“[]”。这些通配符可以搜索并匹配文件名的一部分，从而大大简化命令的输入，这使得多文件的时候可以进行批量操作。

“*”用于匹配文件名中任意长度的字符串。例如需要列出目录中所有.conf文件。

```
$ ls *.conf
```

```
administrator@ubuntu:/etc$ ls *.conf
adduser.conf          kernel-img.conf      pnm2ppa.conf
apg.conf              kerneloops.conf     popularity-contest.conf
appstream.conf        ld.so.conf           resolv.conf
brltty.conf           libao.conf           rsyslog.conf
ca-certificates.conf libaudit.conf        sensors3.conf
debconf.conf          logrotate.conf       sysctl.conf
deluser.conf          ltrace.conf          ucf.conf
fuse.conf             mke2fs.conf          updatedb.conf
gai.conf              mtools.conf          usb_modeswitch.conf
hdparm.conf           nsswitch.conf        vconsole.conf
host.conf             pam.conf
administrator@ubuntu:/etc$
```

和“*”相类似的通配符是“?”。但是不同的地方在于，*可以匹配任意长度的字符，一个“?”只能匹配一个字符。

例如，目前有 text1、text3、text5、textB、text_three、text2、text4、textA、text_one、text_two 这些文件名。

用“text?”只能找到 text1、text2、text3、text4、text5、textA、textB

```
administrator@ubuntu:~/Templates$ ls
text1 text3 text5 textB text_three
text2 text4 textA text_one text_two
administrator@ubuntu:~/Templates$ ls text?
text1:
text2:
text3:
text4:
text5:
textA:
textB:
```

“[]”用于匹配所有出现在方括号内的字符。例如，需要列出以 text 开头而仅以 1 或者 A 结束的文件名

```
administrator@ubuntu:~/Templates$ ls
text1 text3 text5 textB text_three
text2 text4 textA text_one text_two
administrator@ubuntu:~/Templates$ ls text[1A]
text1:
textA:
```

也可以使用短线“-”来指定一个字符集范围。所有包含在上下界之间的字符都会被匹配。例如，需要列出所有以 text 开头并以 1~3 中的某个字符结束的文件。

```
administrator@ubuntu:~/Templates$ ls
text1 text3 text5 textB text_three
text2 text4 textA text_one text_two
administrator@ubuntu:~/Templates$ ls text[1-3]
text1:
text2:
text3:
```

5 查看目录和文件

通过上述的快速上手环节，对于目前 Linux 和命令已经有了初步的认识和使用了，本节

将介绍目录和文件的操作指令，这是用户最常用到的命令了，虽然在其中的一些已经在上文尝试过了，但是这里将作进一步讲解，详细讨论命令各个常用选项。

5.1 显示当前目录：pwd

pwd 命令会显示当前所在位置，即工作目录。例如，执行如下命令：

```
administrator@ubuntu:~/Templates$ cd /usr/local/bin/  
administrator@ubuntu:/usr/local/bin$ pwd  
/usr/local/bin  
administrator@ubuntu:/usr/local/bin$
```

5.2 改变目录：cd

cd 命令是在 Linux 文件系统的不同部分之间移动的基本工作。当登录系统之后，总是会处于用户主目录中。这个目录有一个名字，也就是“路径名”，它是由/home/开头，后面跟着登录的用户名。

输入 cd 命令，后面跟着一个路径名作为参数，就可以直接进入另外一个子目录中去。举例来说，使用下面的命令进入/usr/bin 子目录。

```
administrator@ubuntu:/usr/local/bin$ cd /usr/bin  
administrator@ubuntu:/usr/bin$
```

在/usr/bin 子目录中，可以用以下命令进入/usr 子目录。（也就是说退到上一个文件夹中）

```
administrator@ubuntu:/usr/bin$ cd ..  
administrator@ubuntu:/usr$
```

在/usr/bin 子目录中还可以使用下面的命令直接进入根目录，即“/”目录。

```
administrator@ubuntu:/usr$ cd /  
administrator@ubuntu:/$
```

最后，总能够用下面的命令回到自己的用户主目录。

```
$ cd
```

或者

```
$ cd ~
```

总结一下，在 Shell 中，“..”代表当前目录的上一级目录。而“.”则代表当前目录。另外“~”代表用户主目录。

5.3 列出目录内容：ls

ls 命令是 list 的简化形式，ls 的命令选项非常之多，这里只讨论一些最常用的选项。ls 的基本语法如下：

```
ls [OPTION]... [FILE]...
```

不带任何参数的 ls 命令，用于列出当前目录下的所有文件和子目录。例如：

```
administrator@ubuntu:/$ ls
bin      etc      lib      mnt      run      swapfile  var
boot     home     lib64    opt      sbin     sys       vmlinuz
cdrom    initrd.img  lost+found  proc    snap    tmp       vmlinuz.old
dev      initrd.img.old  media     root    srv     usr
```

在这个列表中，可以方便地区分目录和文件。默认情况下，目录显示为蓝色；普通文件显示为白色；可执行文件显示为草绿色；淡蓝色则表示这个文件是一个链接文件（相当于 Windows 下的快捷方式）。

用于也可以使用 -F 选项的 ls 命令。

```
administrator@ubuntu:/$ ls -F
bin/     etc/     lib/     mnt/     run/     swapfile  var/
boot/    home/    lib64/   opt/     sbin/    sys/      vmlinuz@
cdrom/   initrd.img@  lost+found/  proc/   snap/    tmp/      vmlinuz.old@
dev/     initrd.img.old@  media/     root/   srv/     usr/
```

可以看到，-F 选项会在每个目录后加上 /，在可执行文件后加上 *，在链接文件后加上 @。这些选项在某些无法显示颜色的终端上会比较有用。

但是这些文件就是该目录下所有的文件了吗？可以尝试 -a 选项。如下：

```
administrator@ubuntu:/$ ls -F
bin/     etc/     lib/     mnt/     run/     swapfile  var/
boot/    home/    lib64/   opt/     sbin/    sys/      vmlinuz@
cdrom/   initrd.img@  lost+found/  proc/   snap/    tmp/      vmlinuz.old@
dev/     initrd.img.old@  media/     root/   srv/     usr/
administrator@ubuntu:/$ ls -a
.         cdrom  initrd.img  lost+found  proc  snap  tmp  vmlinuz.old
..        dev    initrd.img.old  media      root  srv   usr
bin       etc    lib         mnt         run  swapfile  var
boot      home  lib64       opt         sbin  sys      vmlinuz
```

这次看到了一些带“.”的文件名。在 Linux 上，这些文件被称作隐含文件，在默认情况下并不会显示，除非指定了 -a 选项，用于显示所有文件。命令的选项可以组合使用，指定多个选项只需要使用一个短线。例如：

```
administrator@ubuntu:/$ ls -aF
./         dev/     lib/     opt/     snap/     usr/
../        etc/     lib64/   proc/    srv/      var/
bin/       home/    lost+found/  root/   swapfile  vmlinuz@
boot/      initrd.img@  media/   run/     sys/      vmlinuz.old@
cdrom/     initrd.img.old@  mnt/     sbin/    tmp/
```

另外一个常用选项就是 -l 选项。这个选项可以用来查看文件的各种属性。例如：

```
administrator@ubuntu:/$ ls -l
总用量 1943004
drwxr-xr-x  2 root root    4096 11月 24 17:32 bin
drwxr-xr-x  3 root root    4096 11月 24 17:36 boot
drwxrwxr-x  2 root root    4096 11月  5 22:08 cdrom
drwxr-xr-x 18 root root    4200 11月 24 21:26 dev
drwxr-xr-x 124 root root   12288 11月 24 19:17 etc
drwxr-xr-x  3 root root    4096 11月  5 22:10 home
lrwxrwxrwx  1 root root     32 11月 24 17:36 initrd.img -> boot/initrd.img
-S.4.0-53-generic
lrwxrwxrwx  1 root root     32 11月  5 22:04 initrd.img.old -> boot/initrd
.img-5.4.0-42-generic
drwxr-xr-x 21 root root    4096 11月  5 22:11 lib
drwxr-xr-x  2 root root    4096  8月  6 15:35 lib64
drwx----- 2 root root   16384 11月  6 06:04 lost+found
drwxr-xr-x  3 root root    4096  8月  6 15:34 media
drwxr-xr-x  2 root root    4096  8月  6 15:34 mnt
drwxr-xr-x  2 root root    4096 11月  5 22:13 opt
dr-xr-xr-x 330 root root     0 11月 24 21:26 proc
drwx----- 4 root root    4096 11月  6 00:29 run
drwxr-xr-x 27 root root     800 11月 24 21:26 run
drwxr-xr-x  2 root root   12288 11月 24 17:32 sbin
drwxr-xr-x 11 root root    4096 11月  5 22:13 snap
drwxr-xr-x  2 root root    4096  8月  6 15:34 srv
```

总共由 8 个不同的信息栏。从左至右一次表示：

- a) 文件的权限标志
- b) 文件的链接个数
- c) 文件所有者的用户名
- d) 该用户所在的用户组组名
- e) 文件的大小
- f) 最后一次被修改时的日期
- g) 最后一次被修改时的实际
- h) 文件名

在 ls 命令后跟上路径名可以查看该子目录中的内容。例如：

```
administrator@ubuntu:/$ ls /etc/init.d
acpid          cups-browsed  networking    speech-dispatcher
alsa-utils     dbus          network-manager spice-vdagent
anacron        dns-clean     open-vm-tools  udev
apparmor       gdm3          plymouth      ufw
apport         grub-common   plymouth-log   unattended-upgrades
avahi-daemon   hwclock.sh    pppd-dns       uuuid
bluetooth      irqbalance    procps         whoopsie
console-setup.sh kerneloops     rsync          x11-common
cron           keyboard-setup.sh rsyslog
cups           kmod          saned
```

5.4 列出目录内容：dir 和 vdir

Windows 用户可能更加熟悉与 dir 这个命令。在 Linux 中，dir 除了比 ls 的功能更少，其他的都是一样的。

```
administrator@ubuntu:/$ dir /etc/init.d/
acpid          cups-browsed  networking    speech-dispatcher
alsa-utils     dbus          network-manager spice-vdagent
anacron        dns-clean     open-vm-tools  udev
apparmor       gdm3          plymouth      ufw
apport         grub-common   plymouth-log   unattended-upgrades
avahi-daemon   hwclock.sh    pppd-dns       uuuid
bluetooth      irqbalance    procps         whoopsie
console-setup.sh kerneloops     rsync          x11-common
cron           keyboard-setup.sh rsyslog
cups           kmod          saned
```

vdir 相当于为 ls 命令加上了 -l 选项，默认情况下列出目录和文件的完整信息。

```
administrator@ubuntu:/$ vdir /etc/init.d/
总用量 172
-rwxr-xr-x 1 root root 2269 4月 22 2017 acpid
-rwxr-xr-x 1 root root 5336 1月 22 2017 alsa-utils
-rwxr-xr-x 1 root root 2014 5月 29 2017 anacron
-rwxr-xr-x 1 root root 4335 3月 22 2018 apparmor
-rwxr-xr-x 1 root root 2805 2月 26 2020 apport
-rwxr-xr-x 1 root root 2401 8月 22 2018 avahi-daemon
-rwxr-xr-x 1 root root 2968 9月 10 2019 bluetooth
-rwxr-xr-x 1 root root 1232 4月 18 2018 console-setup.sh
-rwxr-xr-x 1 root root 3049 11月 15 2017 cron
-rwxr-xr-x 1 root root 2804 3月 27 2018 cups
-rwxr-xr-x 1 root root 1961 2月 25 2018 cups-browsed
-rwxr-xr-x 1 root root 2813 11月 15 2017 dbus
-rwxr-xr-x 1 root root 1172 6月 5 2016 dns-clean
-rwxr-xr-x 1 root root 3033 10月 9 2018 gdm3
-rwxr-xr-x 1 root root 985 7月 20 09:27 grub-common
-rwxr-xr-x 1 root root 3809 2月 14 2018 hwclock.sh
-rwxr-xr-x 1 root root 2444 10月 25 2017 irqbalance
-rwxr-xr-x 1 root root 3131 5月 19 2017 kerneloops
-rwxr-xr-x 1 root root 1479 2月 15 2018 keyboard-setup.sh
-rwxr-xr-x 1 root root 2044 8月 15 2017 kmod
-rwxr-xr-x 1 root root 4597 11月 25 2016 networking
```

6 建立文件和目录

6.1 建立目录：mkdir

mkdir 命令可以一次建立一个或者几个目录。下面的命令在用户主目录下建立 document、picture 两个目录。

```
$ cd ~                ##进入用户主目录
$ mkdir document picture  ##新建两个目录
```

用户也可以使用绝对路径来新建目录。

```
$ mkdir ~/picture/temp  ##在主目录/picture 目录下新建 temp 目录
```

由于主目录下 picture 已经存在了，因此这条命令是合法的。但是当用户试图运行下面这条命令时，mkdir 将提示错误。

```
$ mkdir ~/tempx/job
```

```
mkdir: 无法创建目录"/home/administrator/tempx/job": 没有那个文件或目录
```

这是因为当前在用户主目录下并没有 tempx 这个目录，自然也就无法在 tempx 下创建 job 目录了。为此 mkdir 提供了 -p 选项，用于完整地创建一个子目录结构。

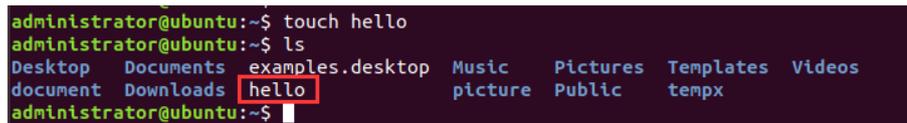
```
$ mkdir -p ~/tempx/job
```

在这个例子中，mkdir 会首先创建 tempx 目录，然后创建 job。在需要创建一个完整的目录结构的时候，这个选项非常有用。

6.2 建立一个空文件：touch

touch 命令的使用非常简单，只需要在后面跟上一个文件名作为参数。下面这个命令在当前目录下新建一个名为 hello 的文件

```
$ touch hello
```



```
administrator@ubuntu:~$ touch hello
administrator@ubuntu:~$ ls
Desktop  Documents  examples.desktop  Music  Pictures  Templates  Videos
document Downloads  hello             picture Public     tempx
```

用 touch 命令建立的文件是空文件。空文件对建立某些特定的实验环境是有用的。另外，当某些应用程序因为缺少文件而无法启动，而这个文件实际上并不那么重要时，可以建立一个空文件暂时“骗过”这个程序。

touch 命令的另一个用途是更新一个文件的建立日期和实际。

touch 命令的这个功能在自动备份和整理文件时非常有用，这使得程序可以决定哪些文件被备份或整理过了。

7 移动、复制和删除

7.1 移动和重命名：mv

mv 是 move 的缩写。这个命令可以用来移动文件。例如将 hello 文件移动到 bin 目录下

```
$ sudo mv hello /bin
```

当然也可以移动目录。下面这条命令把 photos 目录移动到 picture 目录

```
$ mv ./photos picture
```

mv 在移动的时候是不会有提示的，如果遇到一个相同名字的文件，那么默认情况下会直接将其覆盖掉。

为此 mv 命令提供了 -i 选项可以用于发现这样的情况

```
administrator@ubuntu:~$ mv -i test.php test
mv: 是否覆盖 'test/test.php'?
```

回答 y 表示覆盖，回答 n 表示跳过

另外一个比较有用的选项就是 -b。这个选项会用一种不同的方法来处理遇到重复文件的问题。在移动文件前，它会在同名文件的文件名后加一个“~”，从而避免这个文件被覆盖。

7.2 复制文件和目录：cp

cp 命令用来复制文件和目录。下面这条命令将文件 test2.php 复制到 test2 目录下。

```
$ cp test2.php test2
```

和 mv 命令一样，cp 默认情况下也会覆盖同名文件，也可以用 -i 选项对这种情况进行提示，也可以用 -b 对同名文件改名后再复制。

7.3 删除目录和文件：rmdir 和 rm

rmdir 命令用于删除目录。这个命令的使用非常简单，只需要在后面跟上要删除的目录名作为参数即可

```
$ mkdir test2    ##新建一个 test2 目录
```

```
$ rmdir test2    ##删除 test2 目录
```

但是 rmdir 命令只能删除空目录，如下

```
administrator@ubuntu:~$ cp test2.php test2
administrator@ubuntu:~$ rmdir test2
rmdir: 删除 'test2' 失败: 目录非空
administrator@ubuntu:~$
```

因此，在使用 rmdir 删除一个目录之前，首先要将这个目录下的文件和子目录都删除。删除文件需要用到 rm 命令。rm 同样也可以用来删除目录，而且比 rmdir 更高效。所以实际使用的时候 rmdir 很少用到。

rm 命令可以一次删除一个或者几个文件。例如下面这条指令是删除 test2 目录中所有的.php 文件。

本文纯属个人经验，非 Beckhoff 公司正式发行，禁止一切商用行为。如有疑问，请直接联系作者。

```
$rm ./test2/*.php
```

rm 命令和 mv 命令一样，不会做任何提示。通过 rm 命令删除的文件将永远地从系统中消失，而不会被放入一个称为“回收站”的临时目录下。一个比较安全地使用 rm 命令的方式是使用 -i 选项，这个选项会在删除文件前给出提示，并等待用户确认。

```
administrator@ubuntu:~$ rm -i ./test2/test2.php  
rm: 是否删除普通空文件 './test2/test2.php'?
```

带有 -r 参数的 rm 命令会递归地删除目录下所有的文件和子目录。例如

```
$ rm -r ./test2/
```

这样就会把 test2 目录完整地磁盘上移除了。

8 VIM 编辑器

在 Windows 中，记事本是最基础最常用的文本编辑器。所有的 Unix Like 系统都会内建 vi 文书编辑器，其他的文书编辑器则不一定会存在。

但是目前我们使用比较多的是 vim 编辑器。vim 具有程序编辑的能力，可以主动的以字体颜色辨别语法的正确性，方便程序设计

Vim 是从 vi 发展出来的一个文本编辑器。代码补完、编译及错误跳转等方便编程的功能特别丰富，在程序员中被广泛使用。

简单的来说，vi 是老式的字处理器，不过功能已经很齐全了，但是还是有可以进步的地方。Vim 则可以说是程序开发者的一项很好用的工具。

本文主要以实例介绍 Vim 的基本使用，编辑保存。更为详细的 Vim 使用请参考 Vim 手册。

8.1 编辑和保存

要编辑一个文件，可以在命令行下输入 vim file。如果 file 不存在，那么 Vim 会自动新建一个名为 file 的文件。如果使用不带任何参数的 vim 命令，那么就需要在保存的时候，指定文件名。

Vim 分为插入和命令两种模式。在插入模式下可以输入字符，命令模式下则执行除了输入字符之外的所有操作，包括保存、搜索、移动光标等。

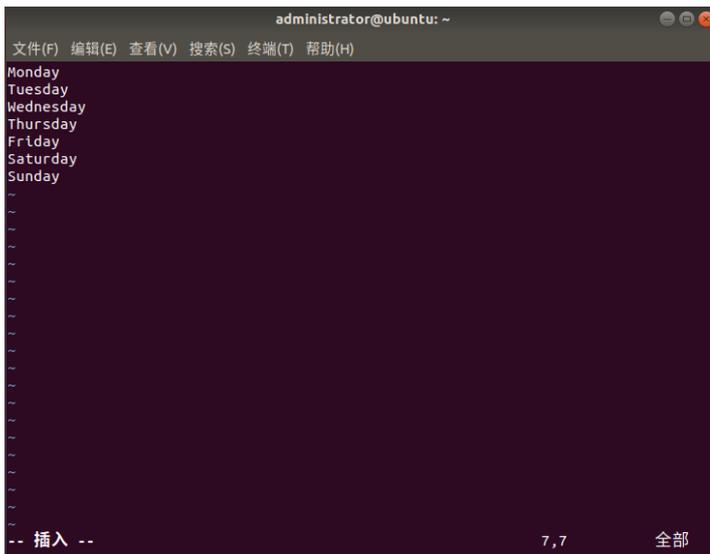


本文纯属个人经验，非 Beckhoff 公司正式发行，禁止一切商用行为。如有疑问，请直接联系作者。

启动 Vim 时自动处于命令模式。按 I 键可以进入插入模式，这个命令用于在当前光标所在处插入字符。Vim 会在左下角提示用户此时所处的模式。请确保没有开启键盘上的大写锁定，因为 Vim 的命令是严格区别大小写的。

接着尝试新建一个文件，然后输入下面一些字符，如果输错了，可以简单地使用退格键删除。（没有 vim 的话，可以尝试用 vi，也是一样的）

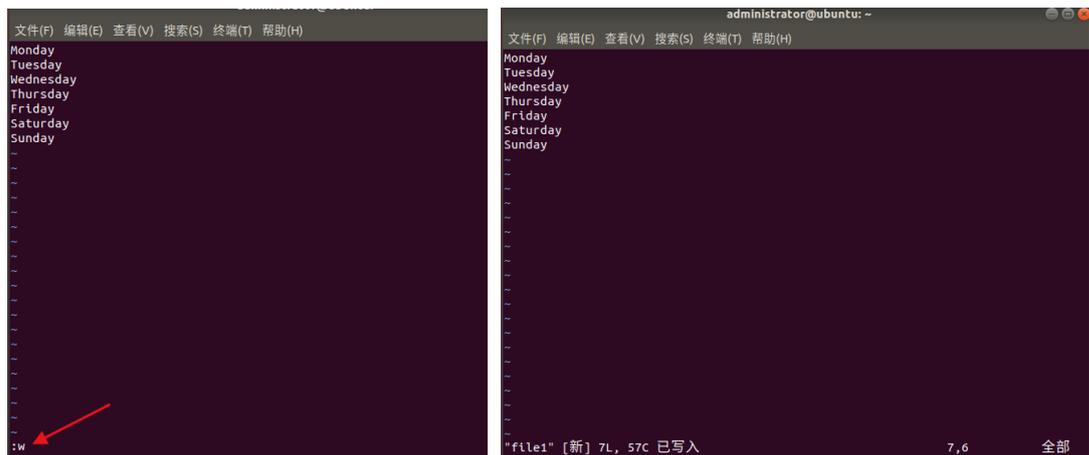
```
$ vim file1
```



输入完成之后，按下 ESC 键位可以回到命令模式，此时左下角的“—插入—”提示消失。

完成文本编辑后，需要保存这个文件。为此需要使用“:”命令在底部打开一个命令行，此时光标闪烁，等待用户输入命令。

输入“w”，然后按下 enter 键，文件保存成功。



如果在命令模式下输入“q”，即可退出

组合使用“:wq”可以保存文件并退出 Vim

如果用户在没有保存修改的情况，就使用命令“:q”，那么 Vim 会拒绝退出，并在底部显示一行提示信息。



如果确定要放弃修改，使用“:q!”命令退出 Vim，所做的修改就会全部失效。

9 软件包管理

顾名思义，软件包是将应用程序、配置文件和管理数据打包的产物。特定的软件包管理系统可以方便地安装和卸载软件包。如今，所有的 Linux 发行版都采用了某种形式的软件包系统，这使得在 Linux 上安装软件变得同在 Windows 下一样方便。常用的软件包格式有两种，这取决于所使用的发行版。SUSE、Red Hat、Fedora 等发行版本使用 RPM，而 Debian 和 Ubuntu 则使用 deb 格式的软件包。

9.1 高级软件包工具：apt

rpm 和 dpkg 这些软件包管理的出现，大大减少了安装软件的工作量。但是系统管理员遗憾地发现，这些工具仍然不能有效地解决依赖性问题。为了安装某个软件，管理员不得不常常陷入“A 依赖于 B，B 依赖于 C，C 依赖于 D……”这类无休止的纠缠中。正是着眼于解决这类问题，以 APT、yum 等为代表的高级软件包管理工具应运而生。

APT，全称是 Advanced Package Tool，即高级软件包工具。这是现今最成熟的软件包管理系统。它可以自动检测软件依赖性问题，下载和安装所有文件；甚至只需要一条命令，就可以更新整个系统上所有的软件包。

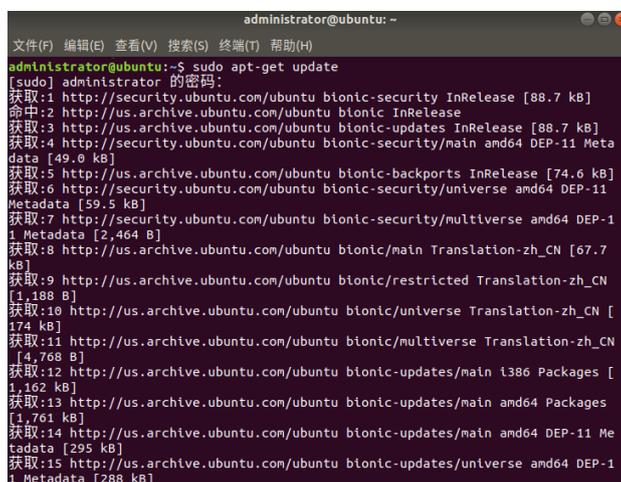
APT 最初被设计运行于 Debian 系统上，只能支持 .deb 格式的软件包文件。如今，APT 已经被移植到使用 RPM 软件包机制的发行版上，所以应用非常广泛。

APT 工具最常用的有两个命令：apt-get 和 apt-cache。前者用于执行和软件包安装有关的所有操作；后者主要用于查找软件包的相关信息。在大部分情况下，用户也可以使用 APT 工具。

9.2 下载和安装软件包

系统第一次启动时，需要运行 apt-get update 更新当前 apt-get 缓存中的软件包信息。此后，就可以使用 apt-get install 命令安装软件包了。实际上，推荐在每次安装和更新软件包之前都运行以下 apt-get update，以保证获得的软件包是最新的。现在来尝试安装 Linux 环境的泡泡龙。

```
$ sudo apt-get install frozen-bubble
```

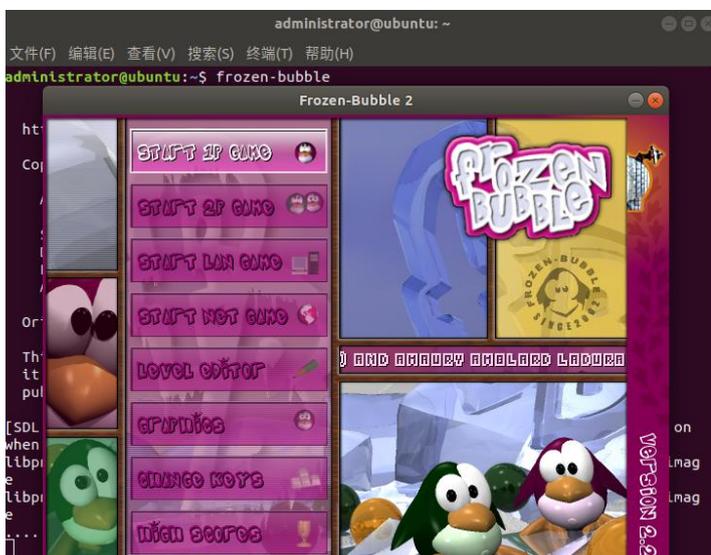


```
administrator@ubuntu: ~
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
administrator@ubuntu:~$ sudo apt-get update
[sudo] administrator 的密码:
获取:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
命中:2 http://us.archive.ubuntu.com/ubuntu bionic InRelease
获取:3 http://us.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
获取:4 http://security.ubuntu.com/ubuntu bionic-security/main amd64 DEP-11 Meta
data [49.0 kB]
获取:5 http://us.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
获取:6 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 DEP-11
Metadata [59.5 kB]
获取:7 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 DEP-1
1 Metadata [2,464 B]
获取:8 http://us.archive.ubuntu.com/ubuntu bionic/main Translation-zh_CN [67.7
kB]
获取:9 http://us.archive.ubuntu.com/ubuntu bionic/restricted Translation-zh_CN
[1,188 B]
获取:10 http://us.archive.ubuntu.com/ubuntu bionic/universe Translation-zh_CN [
174 kB]
获取:11 http://us.archive.ubuntu.com/ubuntu bionic/multiverse Translation-zh_CN
[4,768 B]
获取:12 http://us.archive.ubuntu.com/ubuntu bionic-updates/main i386 Packages [
1,162 kB]
获取:13 http://us.archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages
[1,761 kB]
获取:14 http://us.archive.ubuntu.com/ubuntu bionic-updates/main amd64 DEP-11 Me
tadata [295 kB]
获取:15 http://us.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 DEP-1
1 Metadata [288 kB]
```

本文纯属个人经验，非 Beckhoff 公司正式发行，禁止一切商用行为。如有疑问，请直接联系作者。

```
administrator@ubuntu: ~  
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)  
administrator@ubuntu:~$ sudo apt-get install frozen-bubble  
[sudo] administrator 的密码:  
正在读取软件包列表... 完成  
正在分析软件包的依赖关系树  
正在读取状态信息... 完成  
将会同时安装下列软件:  
  fonts-sil-gentium fonts-sil-gentium-basic frozen-bubble-data  
  libalien-sdl-perl libcapture-tiny-perl libclass-inspector-perl  
  libcompress-bzip2-perl libfile-sharedir-perl libfile-which-perl  
  libfluidsynth1 libmad0 libmikmod3 libopenal-data libopenal1 libSDL-gfx1.2-5  
  libSDL-image1.2 libSDL-mixer1.2 libSDL-pango1 libSDL-perl libSDL-ttf2.0-0  
  libSDL1.2debian libSDL2-2.0-0 libsndio6.1 libtie-simple-perl  
  timgm6mb-soundfont  
建议安装:  
  libportaudio2 sndiod fluid-soundfont-gm fluidsynth timidty musescore  
下列【新】软件包将被安装:  
  fonts-sil-gentium fonts-sil-gentium-basic frozen-bubble frozen-bubble-data  
  libalien-sdl-perl libcapture-tiny-perl libclass-inspector-perl  
  libcompress-bzip2-perl libfile-sharedir-perl libfile-which-perl  
  libfluidsynth1 libmad0 libmikmod3 libopenal-data libopenal1 libSDL-gfx1.2-5  
  libSDL-image1.2 libSDL-mixer1.2 libSDL-pango1 libSDL-perl libSDL-ttf2.0-0  
  libSDL1.2debian libSDL2-2.0-0 libsndio6.1 libtie-simple-perl  
  timgm6mb-soundfont  
升级了 0 个软件包，新安装了 26 个软件包，要卸载 0 个软件包，有 46 个软件包未被  
升级。  
需要下载 30.3 MB 的归档。  
解压缩后会消耗 42.3 MB 的额外空间。  
您希望继续执行吗？ [Y/n]
```

安装完成之后，直接输入 `frozen-bubble` 即可开启该应用软件。



可以看到，APT 提供了大量信息，并自动解决了包的依赖问题。按回车键执行下载和安装即可。

`apt-get` 还有其他一些命令，可以完成诸如升级、删除软件包等操作。以下是一些常用命令。

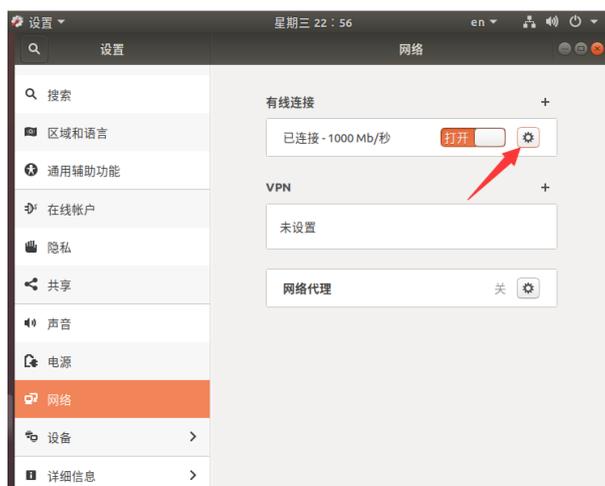
命令	描述
<code>apt-get install</code>	下载并安装软件包
<code>apt-get upgrade</code>	下载并安装在本系统上已有的软件包的最新版
<code>apt-get remove</code>	卸载特定的软件包
<code>apt-get source</code>	下载特定的软件源代码
<code>apt-get clean</code>	删除所有已下载的包文件

五、网络配置

1 获取 IP 地址

网络是人类最伟大的发明直以，这使人类进入了一个全新的时代。VMware 虚拟机平台也提供了非常便捷的桥接模式，可以使得内部的虚拟机直接复制物理网卡信息，成功接入网络中。

通过 Ubuntu 的图形界面，点击设置，然后也能很轻松以及方便地获取或者修改其 IP 地址。



如果没有图形界面的话，也可以使用命令行来实现。

```
$ ifconfig
```

将两张图对比看一下，就会发现，该系统 ipv4 地址为 192.168.100.23，子掩码 255.255.255.0

```
administrator@ubuntu:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.100.23 netmask 255.255.255.0 broadcast 192.168.100.255
    inet6 fdd3:3f1b:1ce:0:7518:fc1d:3213:c39c prefixlen 64 scopeid 0x0<gl
obal>
    inet6 fdd3:3f1b:1ce::960 prefixlen 128 scopeid 0x0<global>
    inet6 fdd3:3f1b:1ce:0:ffcc:89af:3568:5572 prefixlen 64 scopeid 0x0<gl
obal>
    inet6 fe80::d6:2d86:65:dafd prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:16:8c:8f txqueuelen 1000 (以太网)
    RX packets 640 bytes 130705 (130.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 215 bytes 25077 (25.0 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (本地环回)
    RX packets 201 bytes 15732 (15.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 201 bytes 15732 (15.7 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

2 防火墙设置

防火墙是网络安全的基本工具。通过在服务器和外部访客之间建立过滤机制，防火墙在网络层面上实现了安全防范。Linux 的防火墙工具是 IP Tables，这套防火墙系统也被作为很多其他专业网络设备的核心。

IP Tables 已经集成在 Linux 2.4 及以上版本的内核中了。同 Windows 下的防火墙不同的是，IP Tables 需要用户自己定制相关规则。

在初学的时候，可以把防火墙都关闭，但是在后续投入使用的时候，一定要注意网络安全。

2.1 Linux 防火墙基本概念

Linux 防火墙是一种典型的包过滤防火墙，通过检测到到达的数据包头中的信息，确定哪些数据包可以通过，哪些应该被丢弃。防火墙行为的依据主要是数据包的目的地址、端口号和协议类型，所有这些都应该由管理员指定。

Linux 中的包过滤引擎在 2.4 版内核中做了升级。防火墙工具最初叫做 ipchains，取这个名字的原因在于防火墙将一系列规则组成一些“链(chains)”应用到网络数据包上。iptables 则更进一步把一些功能相似的“链”组合成一个个“表(tables)”。

上面的说法有些抽象，现在举一个具体的例子。iptables 默认使用的表是“filter(过滤器)”，其中默认包含了 3 个链，分别是 FORWARD、INPUT 和 OUTPUT。

FORWARD 链中定义的规则作用于那些需要转发到另一个网络接口的数据包。

INPUT 链中定义的规则作用于发送到本机的数据包。

OUTPUT 链中定义的规则作用于从本机发送出去的数据包。

通常定义 filter 表就可以迎合大部分的安全需求，因为这个表包含了包过滤的所有内容。除了 filter，iptables 还包含有 nat 和 mangle 两个表。nat 用于网络地址转换(NAT)，mangle 则用于修改除了 NAT 和包过滤之外的网络包。

简便起见，本节只对 filter 表进行讨论。修改 nat 表和 mangle 表需要更多的网络知识，有兴趣的读者请参考相关的网络安全资料。

2.2 初始化防火墙设置

iptables 命令最常用的 5 个选项分别是 -F、-P、-A、-D 和 -L。大部分情况下，管理员只需要这 5 个选项就可以完成防火墙的规则设置。以下是这五个选项的含义

选项	含义
-F	清除链中所有的规则
-P	为链设置一条默认策略
-A	为链增加一条规则说明
-D	从链中删除一条规则
-L	查看当前表中的链和规则

举个例子-L 选项

```
$ sudo iptables -L
```

```
administrator@ubuntu:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source                               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                               destination
administrator@ubuntu:~$
```

现在这张表里 filter 表空空如也，并且所有链的默认行为都是 ACCEPT，这意味着所有的包都可以不受阻碍地通过防火墙。

接着可以用 iptables -P 用于给链设置默认策略，这条命令的基本语法如下：

```
iptables -P chain-name target
```

其中，chain-name 是链的名字，也就是要填 FORWARD、INPUT 和 OUTPUT 中的一个。target（目标）字段用于定义策略，filter 表中共有 9 个不同的策略可供使用，但最常用的只有 4 个。ACCEPT 表示允许包通过；DROP 丢弃一个包；REJECT 会在丢弃的同时返回一个 ICMP 错误消息。

下面这条命令将 INPUT 链的默认策略更改为 DROP（丢弃），通常对服务器而言，将所有的链的默认策略设置为 DROP（实际上就是把防火墙打开了，开启了白名单模式，默认全部丢弃，唯有链规则中的能接收）

```
$ sudo iptables -P INPUT DROP
```

执行完这套命令后，所有试图同本机建立连接的努力都会失败，因为所有从“外部”到达防火墙的包都被丢弃了，甚至连 ping 自己都不行。

```
$ ping localhost
```

```
administrator@ubuntu:~$ sudo iptables -P INPUT DROP
administrator@ubuntu:~$ sudo iptables -L
Chain INPUT (policy DROP)
target     prot opt source                               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source                               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                               destination
administrator@ubuntu:~$ ping localhost
PING localhost (127.0.0.1) 56(84) bytes of data.
^C
--- localhost ping statistics ---
79 packets transmitted, 0 received, 100% packet loss, time 79871ms
administrator@ubuntu:~$
```

2.3 添加链规则

完成防火墙规则的初始化后，就可以着手添加链规则了。假设当前防火墙所在的主机是一台 web 服务器，为此应该允许外部主机能够连接本机的 80 端口（对应 HTTP 服务器）和 22 端口（对应 SSH 服务）。使用 `iptables -A` 命令添加链规则，该命令的基本语法如下：

```
iptables -A chain-name -i interface -j target
```

其中，`chain-name` 表示链名字，`interface` 表示该规则用于哪个网络接口，`target` 用于定义策略。由于本节的例子中，防火墙主要用于保护本地主机，因此只要对 `INPUT` 链进行设置就可以了。为了简便起见，假设只有一个网络接口 `ens33` 通向外部（因为上一节获取的网络配置信息，请根据实际填写），`lo` 是本地环回接口。以下是用于防火墙设置的相关选项。

选项	含义
<code>-p proto</code>	匹配网络协议：tcp、udp、icmp
<code>--icmp-type -type</code>	匹配 ICMP 类型，和 <code>-p icmp</code> 配合使用
<code>-s source-ip</code>	匹配来源主机（或网络）的 IP 地址
<code>--sport port#</code>	匹配来源主机的端口，和 <code>-s source-ip</code> 配合使用
<code>-d dest-ip</code>	匹配目标主机的 IP 地址
<code>--dport port#</code>	匹配目标主机的端口，和 <code>-d dest-ip</code> 配合使用

下面这条命令添加了一条 `INPUT` 链的规则，允许所有通过 `lo` 接口的连接请求，这样防火墙就不会阻止“自己连接自己”的行为了。（可以 ping 通了）

```
$ sudo iptables -A INPUT -i lo -p ALL -j ACCEPT
```

```
administrator@ubuntu:~$ sudo iptables -A INPUT -i lo -p ALL -j ACCEPT
[sudo] administrator 的密码:
administrator@ubuntu:~$ ping localhost
PING localhost (127.0.0.1) 56(84) bytes of data:
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.123 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.054 ms
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.055 ms
64 bytes from localhost (127.0.0.1): icmp_seq=4 ttl=64 time=0.022 ms
64 bytes from localhost (127.0.0.1): icmp_seq=5 ttl=64 time=0.074 ms
64 bytes from localhost (127.0.0.1): icmp_seq=6 ttl=64 time=0.061 ms
^C
--- localhost ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5102ms
rtt min/avg/max/mdev = 0.022/0.064/0.123/0.032 ms
```

这条命令中还使用了 `-p` 选项。这个选项指定该规则应该匹配哪一种协议。支持的协议包括 `tcp`、`udp` 和 `icmp`。`ALL` 就把这三种全包括了。

通常来说，还应该让外部主机能勾 ping 到这台 web 服务器，这样当网站出现问题的时候，管理员可以简单地通过 `ping` 命令来确定这台服务器是否还在运行。

```
$ sudo iptables -A INPUT -i ens33 -p icmp --icmp-type 8 -j ACCEPT
```

```
administrator@ubuntu:~$ sudo iptables -A INPUT -i ens33 -p icmp --icmp-type 8 -j ACCEPT
administrator@ubuntu:~$ sudo iptables -L
Chain INPUT (policy DROP)
target    prot opt source                destination
ACCEPT    all  --  anywhere              anywhere
ACCEPT    icmp --  anywhere              anywhere          icmp echo-request

Chain FORWARD (policy ACCEPT)
target    prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target    prot opt source                destination
```

```
C:\Users\jackfeng>ping 192.168.100.22

正在 Ping 192.168.100.22 具有 32 字节的数据:
来自 192.168.100.22 的回复: 字节=32 时间<1ms TTL=64

192.168.100.22 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 0ms, 最长 = 0ms, 平均 = 0ms
```

命令的-p 选项指定了该规则匹配协议 ICMP，紧跟的-icmp-type 指定了 ICMP 的类型代码。ping 命令对应的类型代码是 8。

接下来的两条命令增加了对 22 和 80 端口的访问许可。注意这次-p 选项指定的协议类型是 tcp，这是因为 SSH 服务和 HTTP 服务都是基于 TCP 协议的。

```
$ sudo iptables -A INPUT -i ens33 -p tcp --dport 22 -j ACCEPT
```

```
$ sudo iptables -A INPUT -i ens33 -p tcp --dport 80 -j ACCEPT
```

```
administrator@ubuntu:~$ sudo iptables -A INPUT -i ens33 -p tcp --dport 22 -j ACCEPT
administrator@ubuntu:~$ sudo iptables -A INPUT -i ens33 -p tcp --dport 80 -j ACCEPT
administrator@ubuntu:~$ sudo iptables -L
Chain INPUT (policy DROP)
target     prot opt source                destination
ACCEPT    all  --  anywhere              anywhere
ACCEPT    icmp --  anywhere              anywhere      icmp echo-request
ACCEPT    tcp  --  anywhere              anywhere      tcp dpt:ssh
ACCEPT    tcp  --  anywhere              anywhere      tcp dpt:http

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
administrator@ubuntu:~$
```

很多时候，管理员想要做的并不仅仅是把别人挡在门外，还希望指定有哪些人正在试图访问服务器。下面这条命令给 INPUT 链添加一条 LOG（日志记录）策略。

```
$ sudo iptables -A INPUT -i ens33 -j LOG
```

默认情况下，防火墙记录会把访问信息保存在/var/log/messages 中。这是一个文本文件，可以使用任何文本查看命令查看。实际上，/var/log/messages 记录了系统中的大部分行为，这是 Linux 主要的系统日志文件。（Ubuntu 系统默认不开 log，所以没有 /var/log/messages，要开启的话，可以看一下这个方法

https://blog.csdn.net/qq_40938301/article/details/89175623)

```
administrator@ubuntu:/var/log$ cat /var/log/messages
Nov 26 00:25:29 ubuntu kernel: [ 3676.353346] IN=ens33 OUT= MAC=ff:ff:ff:ff:ff:ff:ff:c8:f7:50:15:de:ae:08:00 SRC=192.168.100.20 DST=192.168.100.255 LEN=291 TOS=0x00 PREC=0x00 TTL=128 ID=47296 PROTO=UDP SPT=54915 DPT=54915 LEN=271
Nov 26 00:25:29 ubuntu rsyslogd: [origin software="rsyslogd" swVersion="8.32.0" x-pid="653" x-info="http://www.rsyslog.com"] exiting on signal 15.
Nov 26 00:25:29 ubuntu rsyslogd: imuxsock: Acquired UNIX socket '/run/systemd/journal/syslog' (fd 3) from systemd. [v8.32.0]
Nov 26 00:25:29 ubuntu rsyslogd: rsyslogd's groupid changed to 106
Nov 26 00:25:29 ubuntu rsyslogd: rsyslogd's userid changed to 102
Nov 26 00:25:29 ubuntu rsyslogd: [origin software="rsyslogd" swVersion="8.32.0" x-pid="2333" x-info="http://www.rsyslog.com"] start
Nov 26 00:25:30 ubuntu kernel: [ 3677.343873] IN=ens33 OUT= MAC=ff:ff:ff:ff:ff:ff:ff:c8:f7:50:15:de:ae:08:00 SRC=192.168.100.20 DST=192.168.100.255 LEN=291 TOS=0x00 PREC=0x00 TTL=128 ID=47297 PROTO=UDP SPT=54915 DPT=54915 LEN=271
Nov 26 00:25:31 ubuntu kernel: [ 3678.359132] IN=ens33 OUT= MAC=ff:ff:ff:ff:ff:ff:ff:c8:f7:50:15:de:ae:08:00 SRC=192.168.100.20 DST=192.168.100.255 LEN=291 TOS=0x00 PREC=0x00 TTL=128 ID=47298 PROTO=UDP SPT=54915 DPT=54915 LEN=271
```

本文纯属个人经验，非 Beckhoff 公司正式发行，禁止一切商用行为。如有疑问，请直接联系作者。

```
ff:c8:f7:50:15:de:ae:08:00 SRC=192.168.100.20 DST=192.168.100.255 LEN=291 TOS=0x00  
PREC=0x00 TTL=128 ID=47298 PROTO=UDP SPT=54915 DPT=54915 LEN=271
```

比如这一条，就是从 192.168.100.20（是我的 Windows 电脑发送过来的），其中比较常用的记录字段有 IN（接收数据包的网络接口）、SRC（数据包来源的 IP 地址）、DST（数据包的目的 IP 地址）以及开头的日期和时间等。一般来说，要分析这些日志的话，会借助一些日志分析工具，比如 `swatch` 和 `logcheck`。

2.4 删除链规则

在大部分情况下，管理员在改变防火墙设置之前总时清空整条链规则，因为这样可以避免一些不必要的冲突。但是如果只是想修改一两条，就把整个清空的话就有点得不偿失了。`iptables` 提供了 `-D` 选项来删除链规则，有 2 种不同的语法用于删除一条规则。

```
iptables -D chain rule-specification
```

```
iptables -D chain ruleum
```

第 1 种语法使用规则描述来匹配某条链规则，为此，用户必须一字不差地照搬当初使用 `-A` 选项添加时使用的描述。下面这条命令删除了对 `lo` 环回接口的规则设置。

```
$ sudo iptables -D INPUT -i lo -p ALL -j ACCEPT
```

很少有人会使用这样冗长的命令。`iptables -D` 命令的第 2 种形式（接收规则对应的编号）能够更高效删除链规则。首先需要通过 `--line-numbers` 选项查看链规则的编号。

```
$ sudo iptables -L --line-numbers
```

```
administrator@ubuntu:/var/log$ sudo iptables -L --line-numbers
Chain INPUT (policy DROP)
num target      prot opt source      destination
1  ACCEPT      all  --  anywhere    anywhere
2  ACCEPT      icmp --  anywhere    anywhere          icmp echo-re
quest
3  ACCEPT      tcp  --  anywhere    anywhere          tcp dpt:ssh
4  ACCEPT      tcp  --  anywhere    anywhere          tcp dpt:http
5  LOG         all  --  anywhere    anywhere          LOG level wa
rning
6  LOG         all  --  anywhere    anywhere          LOG level wa
rning

Chain FORWARD (policy ACCEPT)
num target      prot opt source      destination

Chain OUTPUT (policy ACCEPT)
num target      prot opt source      destination
```

下面这条命令删除了 INPUT 中编号为 2 的链规则（是上图的 2 号链规则），然后再查看一下。

```
$ sudo iptables -D INPUT 2
```

```
administrator@ubuntu:/var/log$ sudo iptables -D INPUT 2
administrator@ubuntu:/var/log$ sudo iptables -L --line-numbers
Chain INPUT (policy DROP)
num target      prot opt source      destination
1  ACCEPT      all  --  anywhere    anywhere
2  ACCEPT      tcp  --  anywhere    anywhere          tcp dpt:ssh
3  ACCEPT      tcp  --  anywhere    anywhere          tcp dpt:http
4  LOG         all  --  anywhere    anywhere          LOG level wa
rning
5  LOG         all  --  anywhere    anywhere          LOG level wa
rning

Chain FORWARD (policy ACCEPT)
num target      prot opt source      destination

Chain OUTPUT (policy ACCEPT)
num target      prot opt source      destination
```

最后，如果想要做一些网络测试的话，建议先不要使用链规则表，比如后面章节需要用到的 `SSH`、`winscp`、`ADS 通讯` 等等，等测试通了，在知道其网络信息的情况下，再将链规则加入。

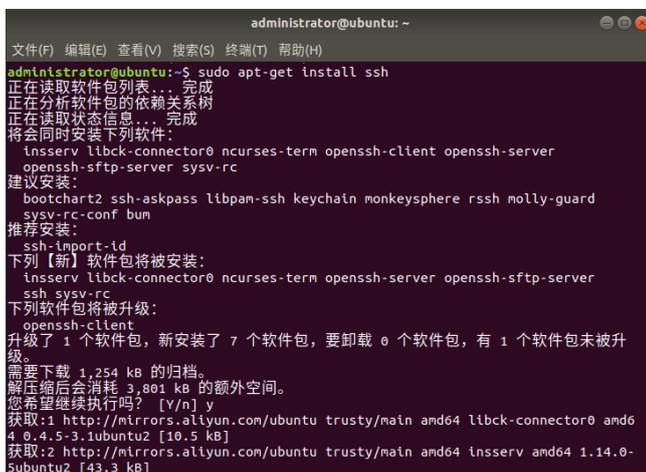
六、 远程登录和传输文件

服务器和 PLC 一样，一般安装到了现场之后，一般不会特意插个显示器鼠标键盘，看它的桌面，用键盘去码字，同样的也没有一个网络管理员会选择把机房作为自己的办公室。为此，管理员们总是使用“远程登录”的方式管理服务器。

1 安装 OpenSSH

OpenSSH 是 Linux 下最常用的 SSH 服务器/客户端软件，Linux 发行版一般都附带了这个软件，可以简单地通过发行版的安装源安装。Ubuntu 用户可以通过下面的命令安装 OpenSSH。（上一章开了防火墙，APT 是需要联网下载软件的，记得把防火墙关掉）

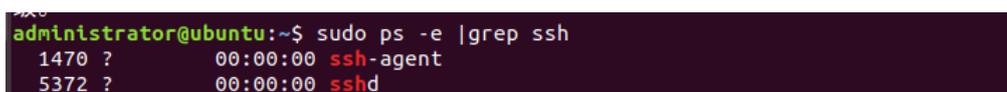
```
$ sudo apt-get install ssh
```



```
administrator@ubuntu: ~
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
administrator@ubuntu:~$ sudo apt-get install ssh
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
将会同时安装下列软件：
  insserv libck-connector0 ncurses-term openssh-client openssh-server
  openssh-sftp-server sysv-rc
建议安装：
  bootchart2 ssh-askpass libpan-ssh keychain monkeysphere rssh molly-guard
  sysv-rc-conf bum
推荐安装：
  ssh-import-id
下列【新】软件包将被安装：
  insserv libck-connector0 ncurses-term openssh-server openssh-sftp-server
  ssh sysv-rc
下列软件包将被升级：
  openssh-client
升级了 1 个软件包，新安装了 7 个软件包，要卸载 0 个软件包，有 1 个软件包未被升级。
需要下载 1,254 kB 的归档。
解压缩后会消耗 3,801 kB 的额外空间。
您希望继续执行吗？ [Y/n] y
获取:1 http://mirrors.aliyun.com/ubuntu trusty/main amd64 libck-connector0 amd64
0.4.5-3.1ubuntu2 [10.5 kB]
获取:2 http://mirrors.aliyun.com/ubuntu trusty/main amd64 insserv amd64 1.14.0-
5ubuntu2 [43.3 kB]
```

输入下面命令，查看是否安装并启动成功了，出现了 sshd 就说明已经启动了。

```
$ sudo ps -e |grep ssh
```



```
administrator@ubuntu:~$ sudo ps -e |grep ssh
1470 ? 00:00:00 ssh-agent
5372 ? 00:00:00 sshd
```

如果没有启动，或者想要停止的话可以用下面命令

启动 ssh 命令：sudo service sshd start

停止 ssh 命令：sudo service sshd stop

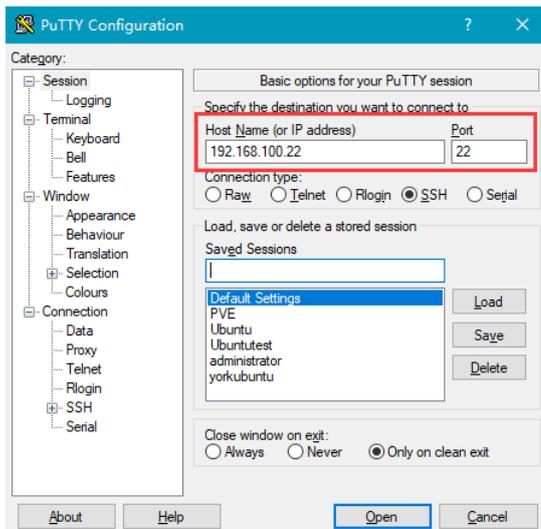
2 从 Windows 登录这台 Linux (PuTTY)

一些任务不得不在 Windows 下完成，而 Linux 作为一款优秀的服务器操作系统又被部署在机房中。这种情况下，要么安装双系统，并且为了短暂的应用而不停地重启计算机，所以可以直接从 Windows 登录到 Linux 服务器中成为了必要的需求。

Windows 上有几种不同的 SSH 客户端，其中开放源代码的 PuTTY 是使用最为广泛、也是最受好评的一个 (<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>)。这是一个绿色软件，不需要安装。下载并允许其主程序 putty.exe，填写远程主机的主机名（或者

本文纯属个人经验，非 Beckhoff 公司正式发行，禁止一切商用行为。如有疑问，请直接联系作者。

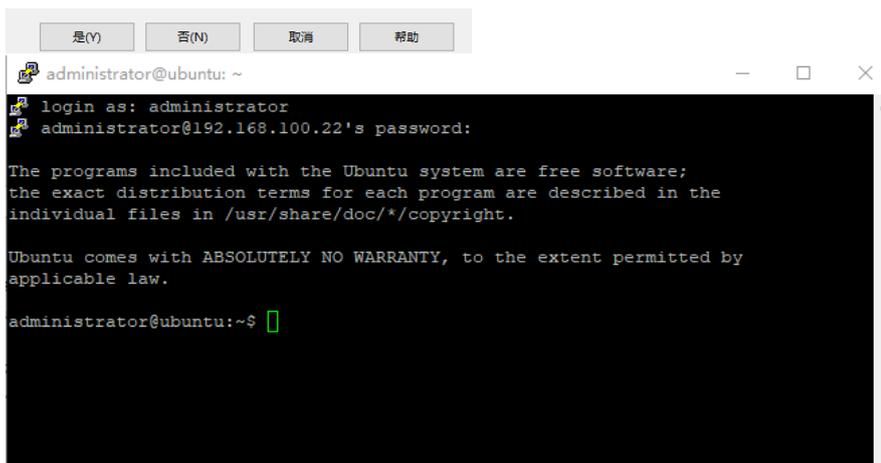
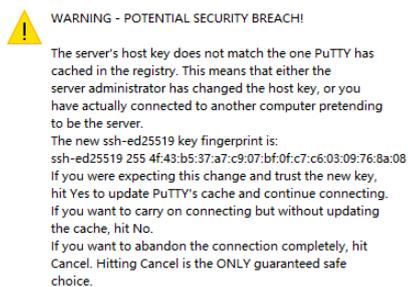
IP 地址) 和登录端口。



单击 Open 按钮，即可建立连接。如果是初次登录，会出现一个提示框，单击“是”按钮继续登录。

PuTTY 将打开一个类似于 Shell 终端的命令行窗口，输入用户名和口令即可完成登录，然后就和 Linux 中的终端是一样的使用方法了。

PuTTY Security Alert

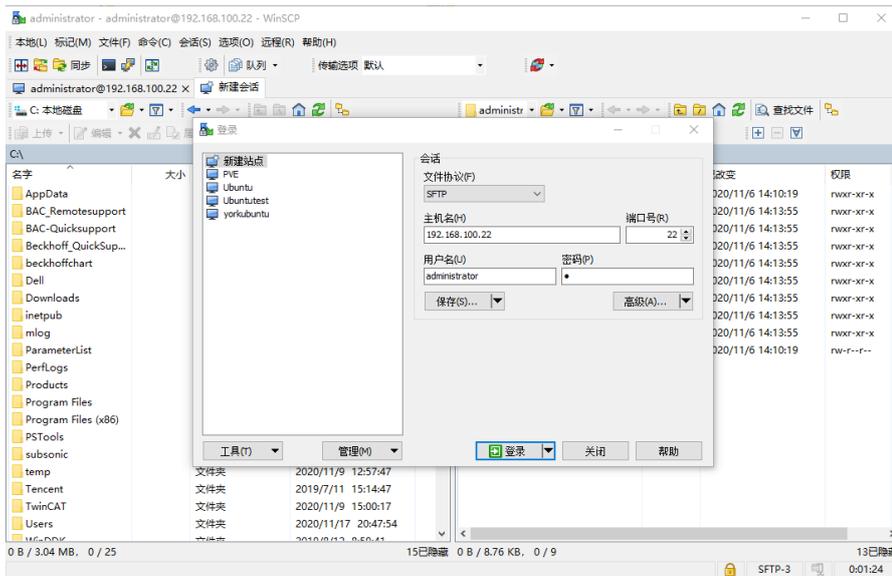


3 传输文件 (WinSCP)

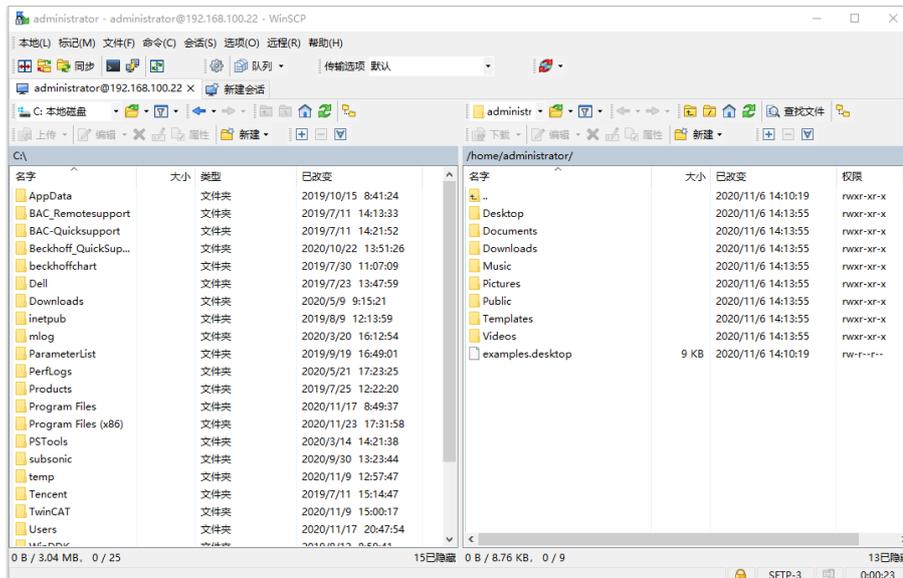
有了远程登录的功能之后，还有个迫切的需求就是，怎么才能将自己的电脑中的文件传输到这台 Linux 中呢。用 U 盘拷过去当然是一种非常好的方法，这里介绍另外一种通过网络

本文纯属个人经验，非 Beckhoff 公司正式发行，禁止一切商用行为。如有疑问，请直接联系作者。

传输的软件——WinSCP (<https://winscp.net/eng/docs/lang:chs>)



新建站点，输入主机名或者 IP 地址，用户名和密码之后，就能访问控制器中的文件系
统



可以将本地的文件传输进入目标系统中，或者将其内部文件拷出来。

七、ADS 通讯

上面讲述的全是一些 Linux 的基本应用，全是为了后面应用而做铺垫的。本章节将利用 Python 的 Pyads 库，配置方便，语言简洁，可以实现在 Linux 系统下与倍福 TwinCAT 进行 ADS 通讯。Pyads 帮助文档：<https://pyads.readthedocs.io/en/latest/index.html>

本节将用两个例子来展示 Linux 下与 TwinCAT:

- 1、用服务器版的 Ubuntu，用最简单的代码完成通讯测试。
- 2、用桌面版的 Ubuntu，做一个图形界面完成通讯测试。

1 在 Linux 中配置 Python 及其组件

这里的设置，用服务器或者桌面版，设置都是同样的，以下截图是在 Ubuntu 18.04.3 LTS 版本中截图的

1.1 安装 Python

Ubuntu 默认是安装 Python 的，而且因为很多内置的软件，程序都是依赖于 Python，所以请**不要**用 `apt-get remove python`，这样一键卸载 python 的话，后果很严重。

```
orange@orange:~$ python3 --version
Python 3.6.9
orange@orange:~$
```

很多 Linux 系统依赖的都是 python2.7 版本，python2 和 python3 是两个大版本，有些库仅支持 python2，或者仅支持 python3，所以 Ubuntu 中如果没有 python3 的话，可以用这个命令安装 `$ sudo apt-get install python3`

```
orange@orange:~$ sudo apt-get install python3
[sudo] password for orange:
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3 is already the newest version (3.6.7-1~18.04).
python3 set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 66 not upgraded.
orange@orange:~$
```

1.2 安装 build 依赖包

很多 pip 安装的包都需要 libssl 和 libevent 编译环境，所以用一下命令进行安装。如果跳过这一步的话，pip 都装不上去。

```
$ sudo apt install build-essential libssl-dev libevent-dev libjpeg-dev libxml2-dev libxslt-dev
```

```
orange@orange:~$ sudo apt install build-essential libssl-dev libevent-dev libjpe
g-dev libxml2-dev libxslt-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'libxslt1-dev' instead of 'libxslt-dev'
The following additional packages will be installed:
 binutils binutils-common binutils-x86-64-linux-gnu cpp cpp-7 dpkg-dev
 fakeroot g++ g++-7 gcc gcc-7 gcc-7-base glibc-harfbuzz-0.0 icu-devtools
 libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl
 libasan4 libatomic1 libbinutils libc-dev-bin libc6 libc6-dev libc6-i386
 libc6-kernel libc6-pkg-perl libevent-core-2.1-6 libevent-extra-2.1-6
 libevent-openssl-2.1-6 libevent-pthreads-2.1-6 libfakeroot
 libfile-fcntllock-perl libgcc-7-dev libglib2.0-bin libglib2.0-dev
 libglib2.0-dev-bin libgomp1 libgraphite2-3 libgraphite2-dev libharfbuzz-dev
 libharfbuzz-gobject0 libharfbuzz-icu0 libharfbuzz0b libicu-dev
 libicu-le-hb-dev libicu-le-hb0 libicu60 libisl19 libitm1 libjpeg-turbo8
 libjpeg-turbo8-dev libjpeg8 libjpeg8-dev liblsan0 libmpc3 libmpx2
 libpcre16-3 libpcre3-dev libpcre3-3 libpcrecpp0v5 libquadmath0
 libstdc++-7-dev libtsan0 libubsan0 linux-libc-dev make manpages-dev
 pkg-config python3-distutils python3-lib2to3 zlib1g-dev
Suggested packages:
 binutils-doc cpp-doc gcc-7-locales debian-keyring g++-multilib
 g++-7-multilib gcc-7-doc libstdc++6-7-dbg gcc-multilib autoconf automake
```

1.3 安装 pip3

pip3 是 python3 用来管理包的工具，可以用来安装、升级、卸载第三方库，当然也可以查看包信息，更加可以发布自己所写的包到 Pypi 社区中。

要用到 pyads 的话，可以手动去拷库到对应文件夹，也可以直接使用 pip3 来进行安装，和之前将的 APT 是差不多的。

```
$ sudo apt install python3-pip
```

注意：是 python3-pip，否则默认安装 python2 的 pip

```
orange@orange:~$ sudo apt install python3-pip
[sudo] password for orange:
Sorry, try again.
[sudo] password for orange:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
 dh-python libexpat1-dev libpython3-dev libpython3.6-dev python-pip-whl
 python3-crypto python3-dev python3-keyring python3-keyrings.alt
 python3-secretstorage python3-setuptools python3-wheel python3-xdg
 python3.6-dev
Suggested packages:
 python-crypto-doc gnome-keyring libkf5wallet-bin glibc-harfbuzz-1.0
 python-secretstorage-doc python-setuptools-doc
The following NEW packages will be installed:
 dh-python libexpat1-dev libpython3-dev libpython3.6-dev python-pip-whl
 python3-crypto python3-dev python3-keyring python3-keyrings.alt python3-pip
 python3-secretstorage python3-setuptools python3-wheel python3-xdg
 python3.6-dev
0 upgraded, 15 newly installed, 0 to remove and 65 not upgraded.
Need to get 48.0 MB of archives.
After this operation, 84.1 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

1.4 安装 pyads

利用 pip3 就能很快速便捷地安装 pyads 了，如果觉得下载的较慢的话，可以将 pip3 的源切换成清华镜像或者阿里云镜像的

```
$ pip3 install pyads
```

或者

```
$ pip3 install pyads -i https://pypi.tuna.tsinghua.edu.cn/simple ##清华镜像
```

或者

```
$ pip3 install pyads -i https://mirrors.aliyun.com/pypi/simple ##阿里云镜像
```

本文纯属个人经验，非 Beckhoff 公司正式发行，禁止一切商用行为。如有疑问，请直接联系作者。

```
orange@orange:~$ pip3 install pyads
Collecting pyads
  Downloading https://files.pythonhosted.org/packages/42/60/293dd529d25b93d77315396756abd85912d81d14f17663ae5117cf962240/pyads-3.3.0.tar.gz (469kB)
    100% |#####| 471kB 61kB/s
Building wheels for collected packages: pyads
  Running setup.py bdist_wheel for pyads ... done
  Stored in directory: /home/orange/.cache/pip/wheels/32/1f/4a/944a6d0d817225b66a8af62e6b5cdeed121532d12ed03af636
Successfully built pyads
Installing collected packages: pyads
Successfully installed pyads-3.3.0
orange@orange:~$
```

1.5 关闭链规则

ADS 通讯需要用到 48898 和 48899 端口，为了方便通讯测试，这里默认将 iptables 链规则全设置成 ACCEPT，命令详情请看第五章防火墙设置

```
root@orange:/home/orange# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
```

2 服务器版 Ubuntu 进行 ADS 通讯

2.1 软硬件版本

控制器：

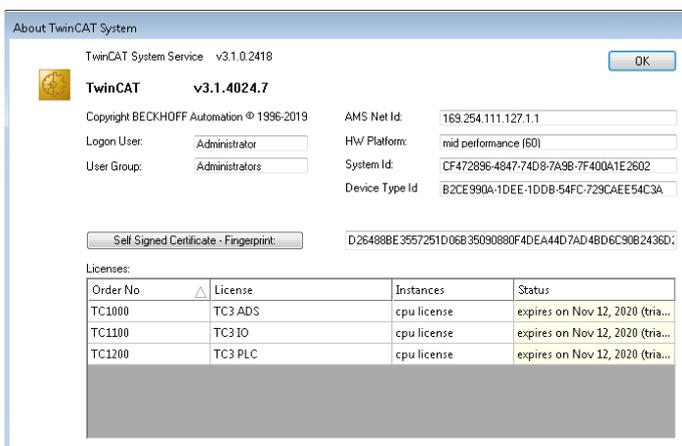
工控机：C6920

TwinCAT 3.1 Build 4024.7

IP 地址：192.168.100.22

子掩码：255.255.255.0

AMS NET ID：169.254.111.127.1.1



本文纯属个人经验，非 Beckhoff 公司正式发行，禁止一切商用行为。如有疑问，请直接联系作者。

Linux:

Ubuntu 18.04.3 LTS

```
Distributor ID: Ubuntu
Description:   Ubuntu 18.04.3 LTS
Release:       18.04
Codename:      bionic
```

IP 地址: 192.168.100.21

子掩码: 255.255.255.0

AMD NETID: 192.168.100.21.1.1 (后面会在 Python 中设置)

Python 3.6.9

```
root@orange:/home/orange# python3 -V
Python 3.6.9
```

2.2 操作步骤

- 1 打开 putty 连接 Ubuntu，并输入 python3 进入 python3 的 shell 模式

```
root@orange:/home/orange# python3
Python 3.6.9 (default, Oct  8 2020, 12:12:24)
[GCC 8.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> █
```

- 2 导入库并开启 Ubuntu 的 ads 端口

```
import pyads
pyads.open_port()
```

```
root@orange:/home/orange# python3
Python 3.6.9 (default, Oct  8 2020, 12:12:24)
[GCC 8.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import pyads
>>> pyads.open_port()
30000
>>> █
```

- 3 在 PLC 端加 Ubuntu 的路由。

方法一：用代码，然后观察一下 PLC 中的路由信息

sending_net_id = '169.254.111.127.1.1' # PLC 的 netid

adding_host_name = '192.168.100.21' # 填 Ubuntu 的 IP 地址

ip_address = '192.168.100.22' # PLC 的 IP

username = 'Administrator' # PLC 用户名

password = '1' # PLC 密码

route_name = 'liunxtest' # Ubuntu 的名字

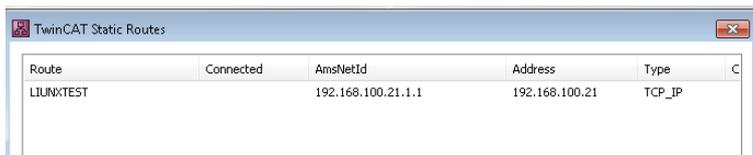
added_net_id = '192.168.100.21.1.1' # Ubuntu 的 netid

控制 PLC 加路由的函数

```
pyads.add_route_to_plc(sending_net_id,adding_host_name,ip_address,username,password,route_name,added_net_id)
```

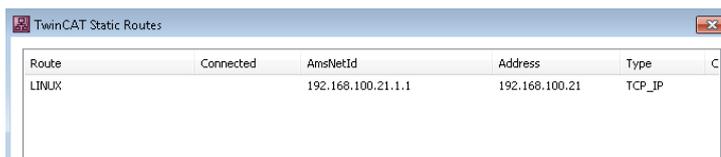
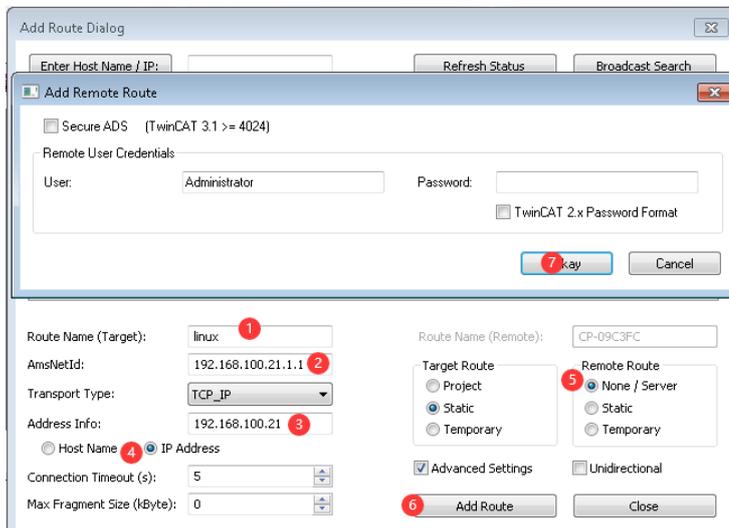
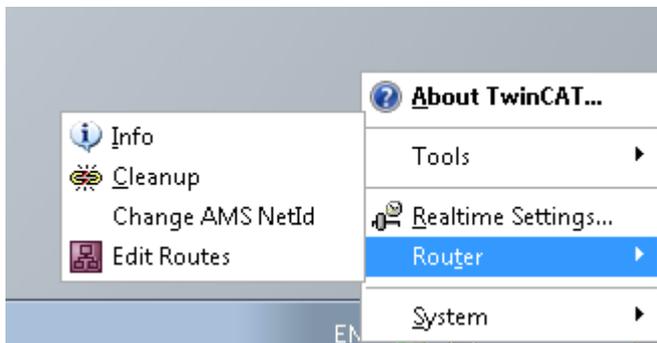
本文纯属个人经验，非 Beckhoff 公司正式发行，禁止一切商用行为。如有疑问，请直接联系作者。

```
root@orange:/home/orange# python3
Python 3.6.9 (default, Oct 8 2020, 12:12:24)
[GCC 8.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import pyads
>>> pyads.open_port()
30000
>>> sending_net_id = '169.254.111.127.1.1'# PLC的netid
>>> adding_host_name = '192.168.100.21'# 填Linux的IP地址
>>> ip_address = '192.168.100.22'# PLC的IP
>>> username = 'Administrator'# PLC用户名
>>> password = '1'# PLC密码
>>> route_name = 'liunxtest'# Linux的名字
>>> added_net_id = '192.168.100.21.1.1'# Linux的netid
>>> pyads.add_route_to_plc(sending_net_id,adding_host_name,ip_address,username,p
assword,route_name,added_net_id)
True
>>>
```



方法二:

用 edit routes 工具手动添加一个静态的路由，不用在意用户密码，直接点确定即可



4 在 Ubuntu 中加 PLC 的路由。

```
pyads.ads
def add_route(adr: Union[str, AmsAddr],
              ip_address: str) -> None
```

pyads.add_route('169.254.111.127.1.1', '192.168.100.22')

```
root@orange:/home/orange# python3
Python 3.6.9 (default, Oct 8 2020, 12:12:24)
[GCC 8.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import pyads
>>> pyads.open_port()
30000
>>> sending_net_id = '169.254.111.127.1.1'# PLC的netid
>>> adding_host_name = '192.168.100.21' # 填Linux的IP地址
>>> ip_address = '192.168.100.22'# PLC的IP
>>> username = 'Administrator'# PLC用户名
>>> password = '1' # PLC密码
>>> route_name = 'linuxtest'# Linux的名字
>>> added_net_id = '192.168.100.21.1.1' # Linux的netid
>>> pyads.add_route_to_plc(sending_net_id,adding_host_name,ip_address,username,p
assword,route_name,added_net_id)
True
>>> pyads.add_route(sending_net_id, ip_address)
2020-11-04T09:05:45+0000 Info: Connected to 192.168.100.22
>>>
```

- 5 设置 Ubuntu 的 netid，注意一定要设置 netid，不然后面读写会报错

pyads.set_local_address('192.168.100.21.1.1')

```
>>> route_name = 'linuxtest'# Linux的名字
>>> added_net_id = '192.168.100.21.1.1' # Linux的netid
>>> pyads.add_route_to_plc(sending_net_id,adding_host_name,ip_address,username,p
assword,route_name,added_net_id)
True
>>> pyads.add_route(sending_net_id, ip_address)
2020-11-04T09:16:41+0000 Info: Connected to 192.168.100.22
>>> pyads.set_local_address('192.168.100.21.1.1')
```

- 6 连接 PLC 的 851 端口并开启

plc = pyads.Connection('169.254.111.127.1.1', 851, '192.168.100.22')

plc.open()

```
pyads
>>> pyads.add_route_to_plc(sending_net_id,adding_host_name,ip_address,username,p
assword,route_name,added_net_id)
True
>>> pyads.add_route(sending_net_id, ip_address)
>>> plc = pyads.Connection(sending_net_id,851,ip_address)
>>> plc.open()
```

```
pyads.ads.Connection
def __init__(self,
              ams_net_id: str,
              ams_net_port: int,
              ip_address: str = None) -> None
```

- 7 读写 PLC 变量，至此通讯成功，其余 pyads 中的方法一样适用于 Ubuntu 平台中使用

plc.read_by_name('MAIN.test', pyads.PLCTYPE_INT)

plc.write_by_name('MAIN.writetest', 100, pyads.PLCTYPE_INT)

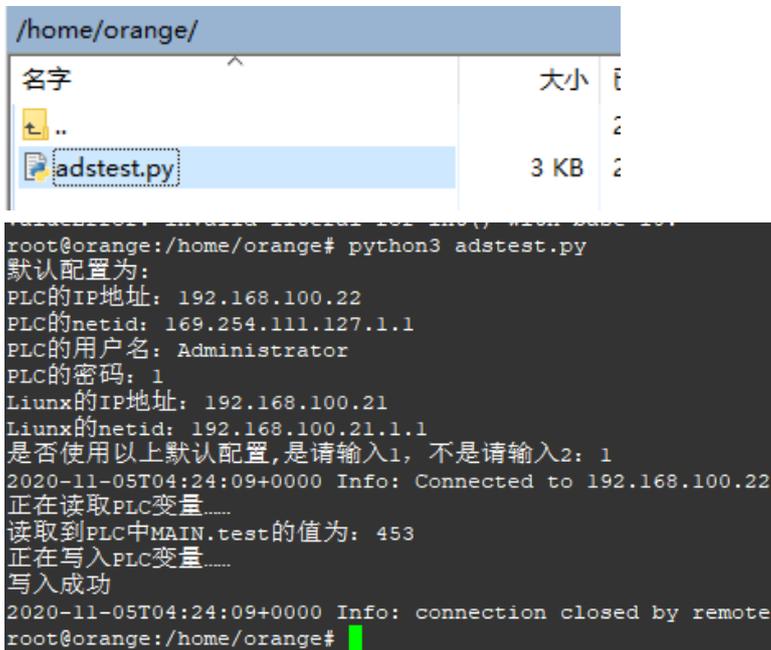
```
>>> plc.write_by_name('MAIN.writetest', 100, pyads.PLCTYPE_INT)
>>> plc.read_by_name('MAIN.test', pyads.PLCTYPE_INT)
453
>>> plc.write_by_name('MAIN.writetest', 100, pyads.PLCTYPE_INT)
>>>
```

LinuxADS.ADS.MAIN		
Expression	Type	Value
test	INT	453
writetest	INT	100

本文纯属个人经验，非 Beckhoff 公司正式发行，禁止一切商用行为。如有疑问，请直接联系作者。

- 8 以上是通过 shell 窗口分步运行，现在将这些程序集成写在一个文件中 `adstest.py`，然后把文件放到对应目录并行

```
root@orange:/home/orange# python3 adstest.py
```



3 桌面版 Ubuntu 进行 ADS 通讯

3.1 软硬件版本

控制器:

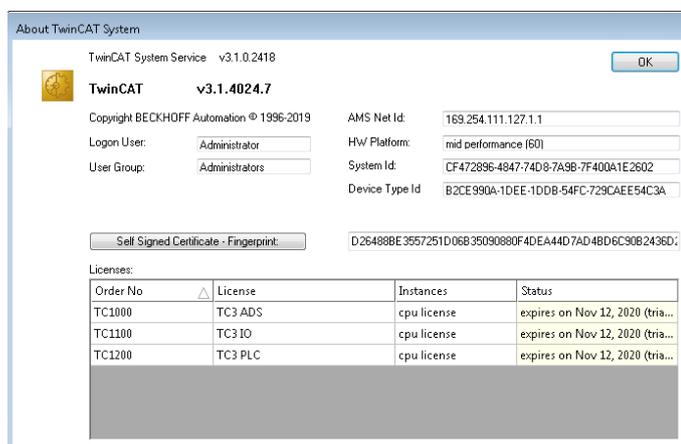
工控机: C6920

TwinCAT 3.1 Build 4024.7

IP 地址: 192.168.100.22

子掩码: 255.255.255.0

AMS NET ID: 169.254.111.127.1.1



本文纯属个人经验，非 Beckhoff 公司正式发行，禁止一切商用行为。如有疑问，请直接联系作者。

Linux:

Ubuntu 18.04.3 LTS

```
administrator@ubuntu:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:   Ubuntu 18.04.5 LTS
Release:       18.04
Codename:      bionic
```

IP 地址: 192.168.100.23

子掩码: 255.255.255.0

AMD NETID: 192.168.100.23.1.1 (后面会在 Python 中设置)

Python 3.6.9

```
administrator@ubuntu:~$ python3 -V
Python 3.6.9
administrator@ubuntu:~$
```

3.2 操作步骤

上面一节已经初步教了在 Linux 中 ADS 通讯需要哪些操作，如果是在桌面版的 Ubuntu 中实现 ADS 通讯的话，注意也要安装 python 和 pyads 等组件的，详情请看在上节的“在 Linux 中配置 python 及其组件”

这里就直接将代码写好，然后拷到 Ubuntu 中即可。因为想要在 Ubuntu 中做一个界面显示，所以这里需要利用 python 的 tkinter 库来实现，关于 python 的 tkinter 代码不做解读，请有余力者自行学习。

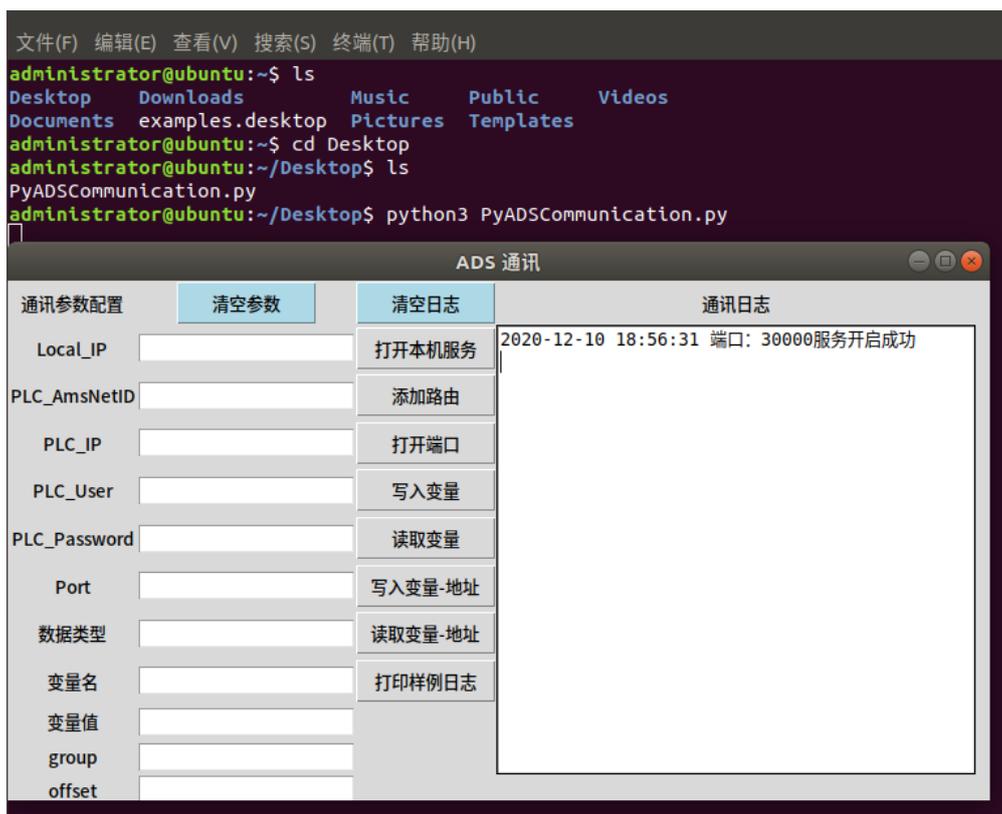
1. tkinter 与 pyads 安装不一样，需要用 APT 来安装，所以命令如下

```
$ sudo apt-get install python3-tk
```

2. 将例程拷入 Ubuntu 中，如果是虚拟机的话可以直接复制粘贴进去，也可以用使用 winscp 进行拷入



3. 在文件所在目录，打开终端，并运行 python3 PyADSCommunication.py



4. 填写对应参数

local_ip (Ubuntu 的 ip 地址);

PLC_AMSNetID (PLC 的 AMSNetID);

PLC_IP (PLC 的 IP 地址);

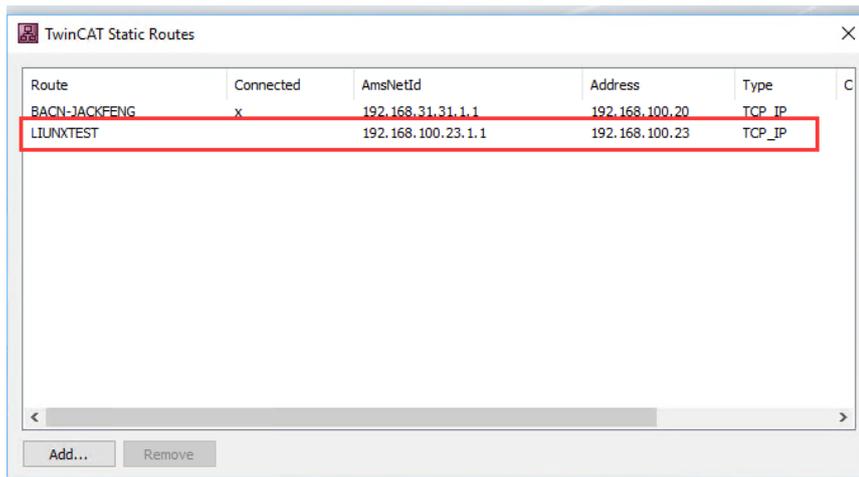
PLC_User (PLC 的用户名);

PLC_Password (PLC 的密码);

然后添加路由，添加成功后，看一下 PLC 的路由表，也多出来一项路由信息了



本文纯属个人经验，非 Beckhoff 公司正式发行，禁止一切商用行为。如有疑问，请直接联系作者。



5. 填写程序所在的 PORT 口，并点击打开端口



6. 填写数据类型，和变量名，点击读取变量即可读取变量；
或者将变量值填好，点击写入变量也可以进行写入操作



7. 如果不想用变量名，也可以填写变量的 group 和 offset 进行读取写入操作

本文纯属个人经验，非 Beckhoff 公司正式发行，禁止一切商用行为。如有疑问，请直接联系作者。



4 常见问题及解决方法

1. ADS error 报错号 1286，报错 1286 解释是 port 已经被使用，根本原因是端口没有找到。在 windows 端中用 pyads.Connection 填了 netid 和 port 号就可以通上，但是在 Linux 端需要多填一个参数（目标的 IP 地址）才能通讯成功。

```
>>> import pyads
>>> pyads.open_port()
30000
>>> remote_ip = '192.168.100.22'
>>> remote_ads = '169.254.111.127.1.1'
>>> pyads.add_route(remote_ads, remote_ip)
2020-11-04T06:41:43+0000 Info: Connected to 192.168.100.22
>>> pyads.set_local_address('192.168.100.21.1.1')
>>> plc = pyads.Connection(remote_ads, 852)
>>> plc.open()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/usr/local/lib/python3.6/dist-packages/pyads/ads.py", line 492, in open
    adsAddRoute(self, adr.netIdStruct(), self.ip_address)
  File "/usr/local/lib/python3.6/dist-packages/pyads/pyads_ex.py", line 155, in
  wrapper
    return fn(*args, **kwargs)
  File "/usr/local/lib/python3.6/dist-packages/pyads/pyads_ex.py", line 177, in
  adsAddRoute
    raise ADSError(error_code)
pyads.pyads_ex.ADSError: ADSError: ROUTERRR_PORTALREADYINUSE (1286).
>>> plc = pyads.Connection(remote_ads, 852, remote_ip)
>>> plc.open()
<bound method Connection.open of <pyads.ads.Connection object at 0x7f81bb63a0b8>
>
>>> plc.open()
>>> a = plc.read_by_name('MAIN.test', pyads.PLCTYPE_INT)
>>> a
-10010
>>>
```

Hex	Dec	Name	Description
0x500	1280	ROUTERRR_NOLOCKEDMEMORY	No locked memory can be allocated
0x501	1281	ROUTERRR_RESIZEMEMORY	The size of the router memory could not be changed
0x502	1282	ROUTERRR_MAILBOXFULL	The mailbox has reached the maximum number of possible messages. The current sent message was rejected
0x503	1283	ROUTERRR_DEBUGBOXFULL	The mailbox has reached the maximum number of possible messages. The sent message will not be displayed in the debug monitor
0x504	1284	ROUTERRR_UNKNOWNPORTTYPE	Unknown port type
0x505	1285	ROUTERRR_NOTINITIALIZED	Router is not initialized
0x506	1286	ROUTERRR_PORTALREADYINUSE	The desired port number is already assigned
0x507	1287	ROUTERRR_NOTREGISTERED	Port not registered
0x508	1288	ROUTERRR_NOMOREQUEUES	The maximum number of Ports reached
0x509	1289	ROUTERRR_INVALIDPORT	Invalid port
0x50A	1290	ROUTERRR_NOTACTIVATED	TwincAT Router not active

2. ADS error 报错号 1861 超时报错，超时报错的原因有很多，比如目标没找到，比如防火墙开着等。

下图是因为 Linux 设置的 netid(1.2.3.4.1.1)和在 PLC 添加 Linux 路由信息的 netid(192.168.100.21.1.1)不匹配导致的。

本文纯属个人经验，非 Beckhoff 公司正式发行，禁止一切商用行为。如有疑问，请直接联系作者。

```
133
>>> pyads.set_local_address('1.2.3.4.1.1')
>>> plc = pyads.Connection(sending_net_id,851,ip_address)
>>> plc.open()
>>> plc.read_by_name('MAIN.test', pyads.PLCTYPE_INT)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/usr/local/lib/python3.6/dist-packages/pyads/ads.py", line 768, in read_
by_name
    check_length=check_length,
  File "/usr/local/lib/python3.6/dist-packages/pyads/pyads_ex.py", line 977, in
adsSyncReadByNameEx
    handle = adsGetHandle(port, address, data_name)
  File "/usr/local/lib/python3.6/dist-packages/pyads/pyads_ex.py", line 777, in
adsGetHandle
    PLCTYPE_STRING,
  File "/usr/local/lib/python3.6/dist-packages/pyads/pyads_ex.py", line 638, in
adsSyncReadWriteReqEx2
    raise ADSError(err_code)
pyads.pyads_ex.ADSError: ADSError: timeout elapsed (1861).
>>> |
```

本文纯属个人经验，非 Beckhoff 公司正式发行，禁止一切商用行为。如有疑问，请直接联系作者。

上海（中国区总部）

中国上海市静安区汶水路 299 弄 9 号（市北智汇园）

电话：021-66312666 传真：021-66315696 邮编：200072

北京分公司

北京市西城区新街口北大街 3 号新街高和大厦 407 室

电话：010-82200036 传真：010-82200039 邮编：100035

广州分公司

广州市天河区珠江新城珠江东路16号高德置地G2603室

电话：020-38010300/1/2 传真：020-38010303 邮编：510623

成都分公司

成都市锦江区东御街18号 百扬大厦2305 房

电话：028-86202581 传真：028-86202582 邮编：610016



请用微信扫描二维码
通过公众号与技术支持交流

倍福中文官网：

<https://www.beckhoff.com.cn/>

倍福虚拟学院：

<https://tr.beckhoff.com.cn/>

招贤纳士：job@beckhoff.com.cn

技术支持：support@beckhoff.com.cn

产品维修：service@beckhoff.com.cn

方案咨询：sales@beckhoff.com.cn