

第三章 Ubuntu 操作入门

3.1 Ubuntu 下打开终端

我们安装的 Ubuntu 是桌面版本, 这样我们可以像在 windows 系统下操作一样, 相对于平时所说的 Linux 命令行下操作来说, 这种体验非常舒适。但是一般我们使用 Linux 都是在命令行下进行操作, 所有的操作我们的都可以通过输入命令来完成, 绝大多数情况下使用命令行来操作 Linux 系统比通过在 GUI 下操作的效率高很多, 虽然说我们使用的 Ubuntu 是包含了 GUI 的 Linux 发行版, 然而我们可以像在 windows 下那样唤出 Ubuntu 的终端, 打开 Ubuntu 的终端非常简单, 以我们使用的 Ubuntu18.04 为例, 有有方法可以直接在 Ubuntu 的用户界面下

3.1.1 用搜索框打开终端

我们要输入各种命令, 需要先打开终端。

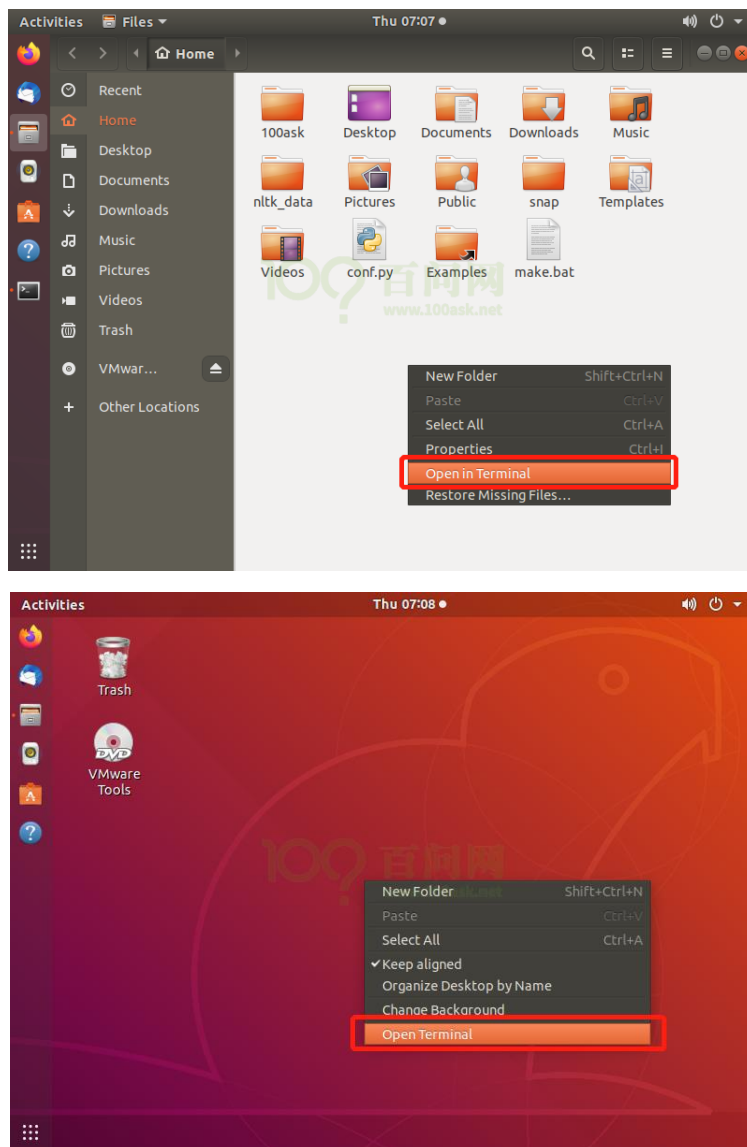
点击 Ubuntu 桌面左上角图标进入搜索框, 输入 “term” 可以弹出终端 “Terminal” 程序, 运行它, 如下图所示:



然后就可以在里面执行各种命令了。

3.1.2 使用右键打开终端

- 1) 首先是比较常规的方法：在桌面或者在文件浏览器的任何目录下右键鼠标后在弹出的菜单栏中选择“Open in Terminal”



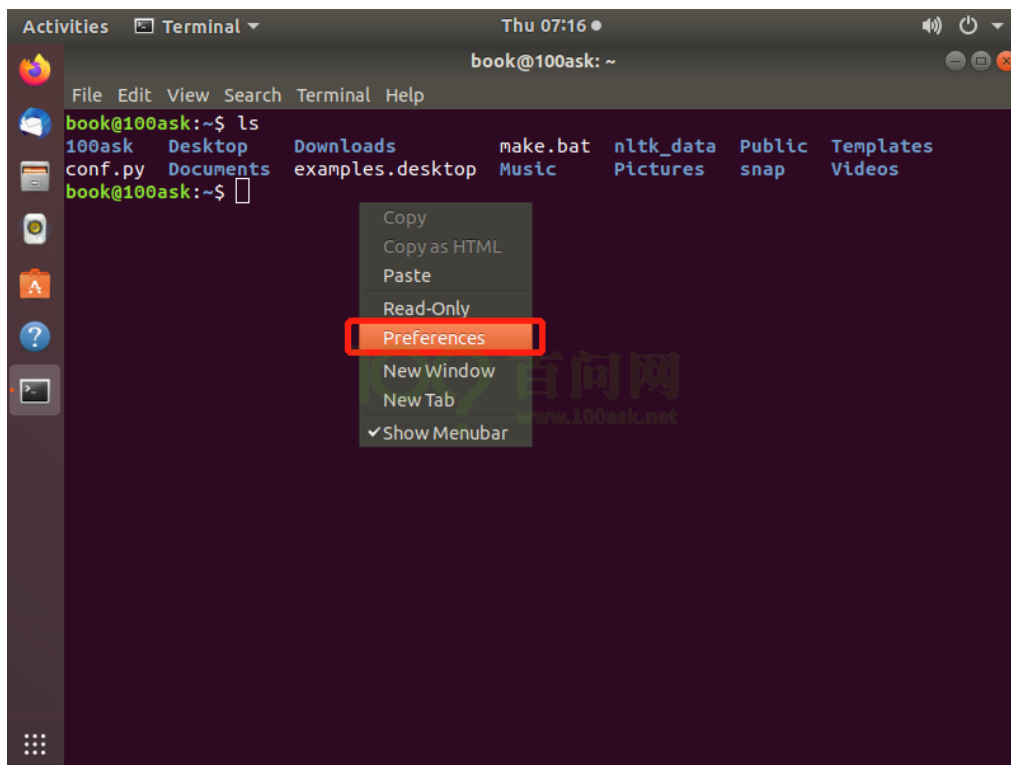
3.1.3 快捷键打开终端

这是个比较快捷方便的方法：使用快捷方式打开终端，快捷方式为”**Ctrl+Alt+T**”，使用快捷方式可在绝大多数情况下直接唤出 Ubuntu 的终端（无论你是在浏览器、文件管理器、查看邮件、甚至在一个已经打开的终端下工作，等等都可以直接唤出 Ubuntu 的终端）

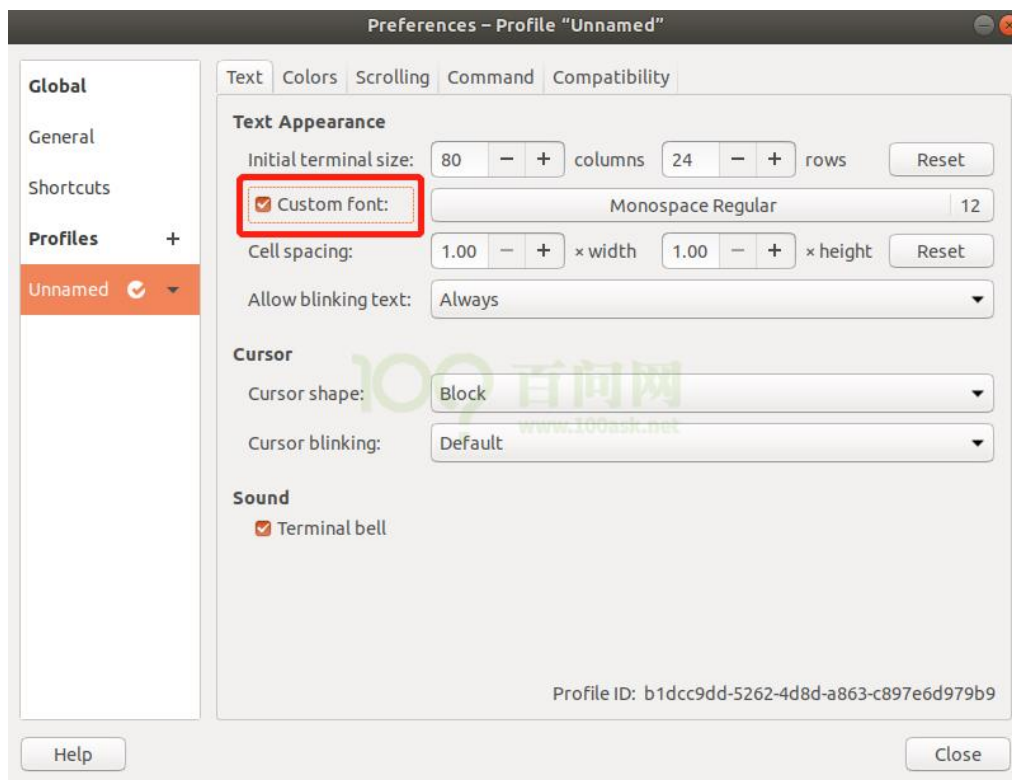
3.1.4 调节终端的字体大小及颜色

现在是有终端可以使用了，但是如果我们想调节终端的字体大小该怎么操作呢？这里也有两种方法可以调节终端字体的大小：

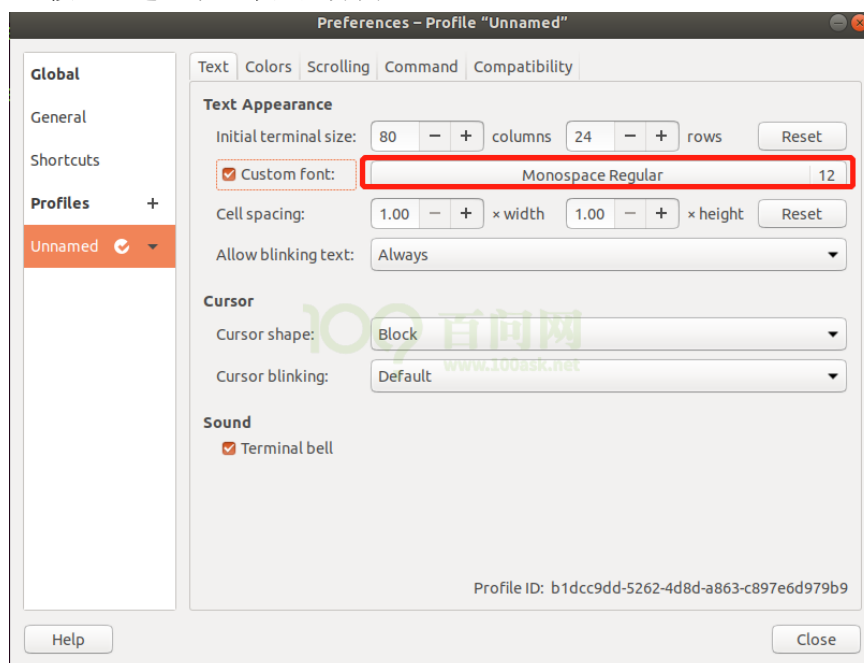
首先也是介绍比较常规的方法，在打开的终端下，在终端界面单击鼠标右键，选择“Preferences”



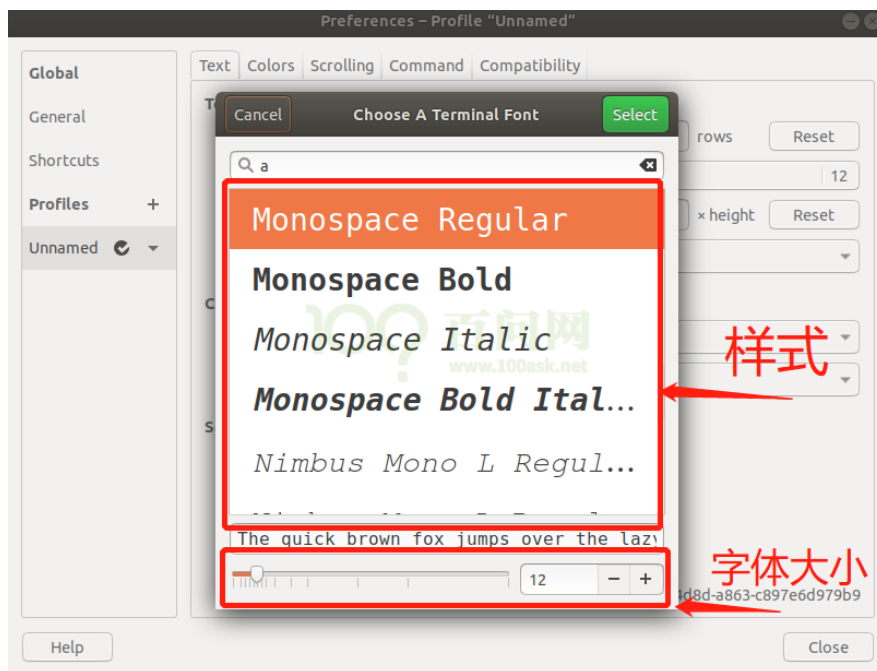
点击选择“Preferences”之后会进入，下面的配置界面，勾选“Custom font”：



勾选“Custom font”后，其右边的“Monospace Bold”按钮将会激活，我们再单击其右边的“Monospace Bold”按钮，进入下一个配置界面：



来到下面的界面后，我们就可以调节终端的字体大小甚至是字体的样式了，这里请根据实际的需求进行选择吧：



2) 如果我们仅仅是想调节终端字体的大小, 有一个非常方便快捷的快捷方式:

- 将字体调大: Ctrl + Shift + '+' (最后是符号+)
- 将字体调小: Ctrl + '-' (最后是符号-)

3.2 Ubuntu 系统初体验

3.2.1 Ubuntu 和 Windows 的最大差别: 目录

Windows 中每一个分区都对应一个盘符, 盘符下可以存放目录与文件:



注意: 目录就是文件夹。

Windows 下某个文件的绝对路径以盘符开始, 比如: C:\abc\def\hello.txt, 这是在 C 盘的 abc 目录下, 有 def 子目录; 而 def 中有 hello.txt 文件。

Ubuntu 中, 以树状结构表示文件夹与文件, 没有盘符的概念。比如: /abc/def/hello.txt, 这表示在根目录下有 abc 子目录, 而 abc 下又有 def 目录; def 中有 hello.txt 文件。

从名字 “/abc/def/hello.txt” 中你无法知道 hello.txt 文件位于磁盘哪一个分区。

注意: 要想查看某个分区挂载在哪一个目录下, 可以执行命令: df -h

对于普通用户, 在 Ubuntu 下不再关心分区、盘符。需要关心的是哪个目录存什么:



Ubuntu 中的目录遵循 FHS 标准 (Filesystem Hierarchy Standard, 文件系统层次标准)。它定义了文件系统中目录、文件分类存放的原则、定义了系统运行所需的最小文件、目录的集合, 并列举了不遵循这些原则的例外情况及其原因。FHS 并不是一个强制的标准, 但是大多的 Linux、Unix 发行版本遵循 FHS。

这些目录简单介绍如下:

/	
bin	所有用户都可以使用的、基本的命令
boot	启动文件, 比如内核等
dev	设备文件, Linux特有的
etc	配置文件
home	家目录
book	用户book的家目录
lib	库
media	插上U盘等外设时会挂载到该目录下
mnt	用来挂载其他文件系统
opt	Optional, 可选的程序
proc	用来挂载虚拟的proc文件系统, 可以查看各进程(process)的信息
root	root用户的家目录
sbin	基本的系统命令, 系统管理员才能使用
sys	用来挂载虚拟的sys文件系统, 可以查看系统信息: 比如设备信息
tmp	临时目录, 存放临时文件
usr	Unix Software Resource, 存放可分享的与不可变动的的数据
bin	绝大部分的用户可使用指令都放在这里(与开机无关), /bin中的命令跟开机有关
games	游戏
include	头文件
lib	库
local	系统管理员在本机自行安装、下载的软件
sbin	非系统正常运作所需要的系统命令
share	放置共享文件的地方, 比如/usr/share/man里存放帮助文件
src	源码
var	主要针对常态性变动的文件, 包括缓存(cache)、log文件等

3.2.2 Linux 文件属性

在终端执行“ls -al”命令显示当前目录下的所有文件及文件夹的详细信息。

文件属性示意图如下



第一个字符表示“文件类型”, 它是目录、文件或链接文件等。

d	目录
-	文件
l	链接文件
b	设备文件里的可供存储的接口设备
c	设备文件里的串行端口设备, 如鼠标、键盘等

文件类型后面的 9 个字符以 3 个为一组，第一组表示“文件所有者的权限”；第二组表示“用户组的权限”；第三组表示“其他非本用户组的权限”。每组都是 **rwX** 的组合，其中 **r** 代表可读，**w** 代表可写，**x** 代表可执行；如果没有对应的权限，就会出现减号 (-)。比如“**rw-r--r--**”表示：文件的所有者对该文件有读权限、写权限，但是没有执行权限；同一个用户组的其他用户对该文件只有读权限；其他用户对该文件也只有读权限。

连接数：表示有多少文件名连接到此节点。

文件所有者：表示这个文件的“所有者的账号”。

文件所属用户组。

文件大小：表示这个文件的大小，默认单位是 **B**(字节)。

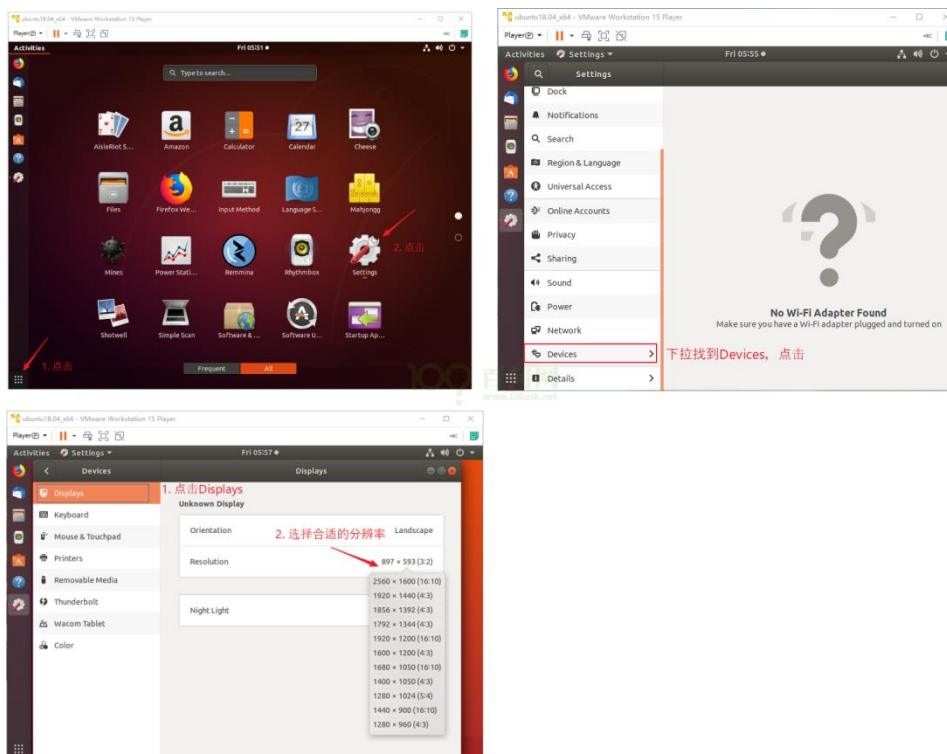
文件最后被修改的时间：这个文件的创建文件日期或者是最近的修改日期。

文件名：对应文件的文件名。

如果文件名之前多了一个“.”,则说明这个文件为“隐藏文件”，执行“**ls -a**”命令可以列出隐藏文件。

3.2.3 设置屏幕

在我们的后续学习过程中，很少使用 Ubuntu 的桌面系统，都是远程登录上去的。但是如果 Ubuntu 的桌面显示太小、太大，总是让人不舒服。这时可以修改屏幕分辨率，方法如下：

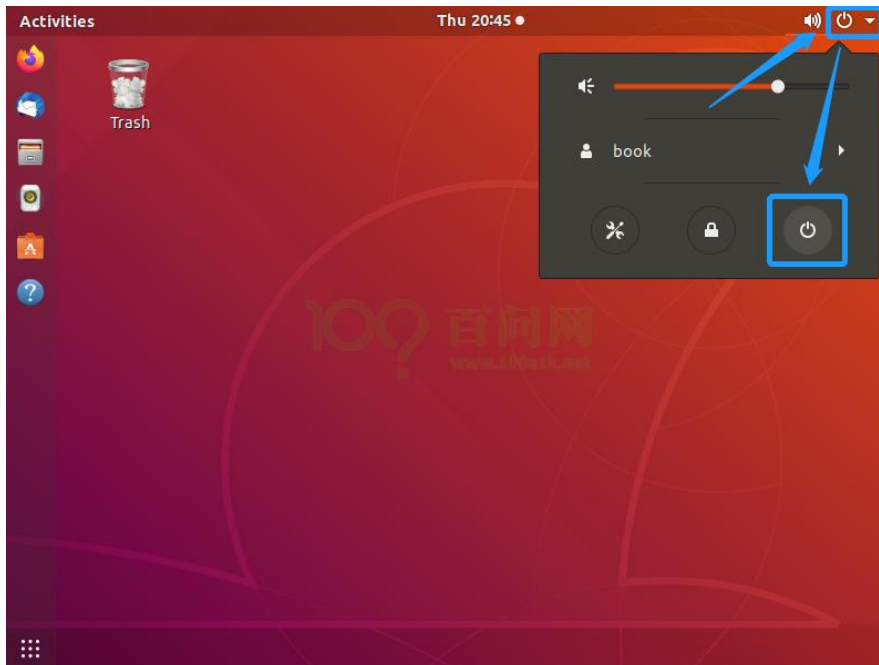


3.2.4 系统关机与重启

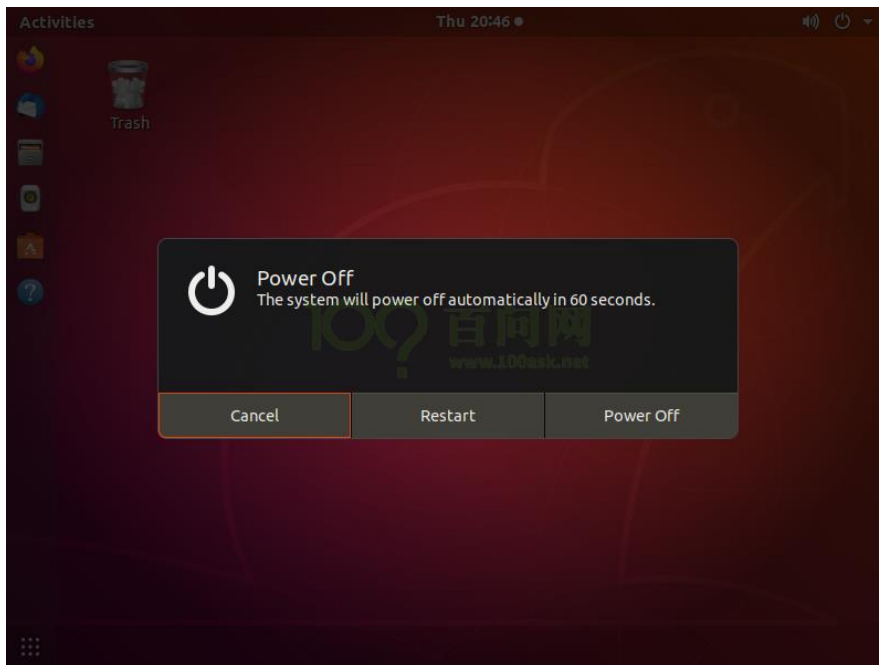
和我们使用 Windows 系统一样，当我们不使用 Ubuntu 系统以后就需要将其关机或者睡眠，千万不要通过直接退出 VMware 软件来关机!! 一般的步骤是：

在虚拟机系统中关闭系统或在 VMware 软件上挂起虚拟机->关闭 VMware 软件->关闭 windows 系统

Ubuntu 的关机与重启很简单，在主界面，点击右上角的图标，然后选项对应的选项即可，如下图所示：

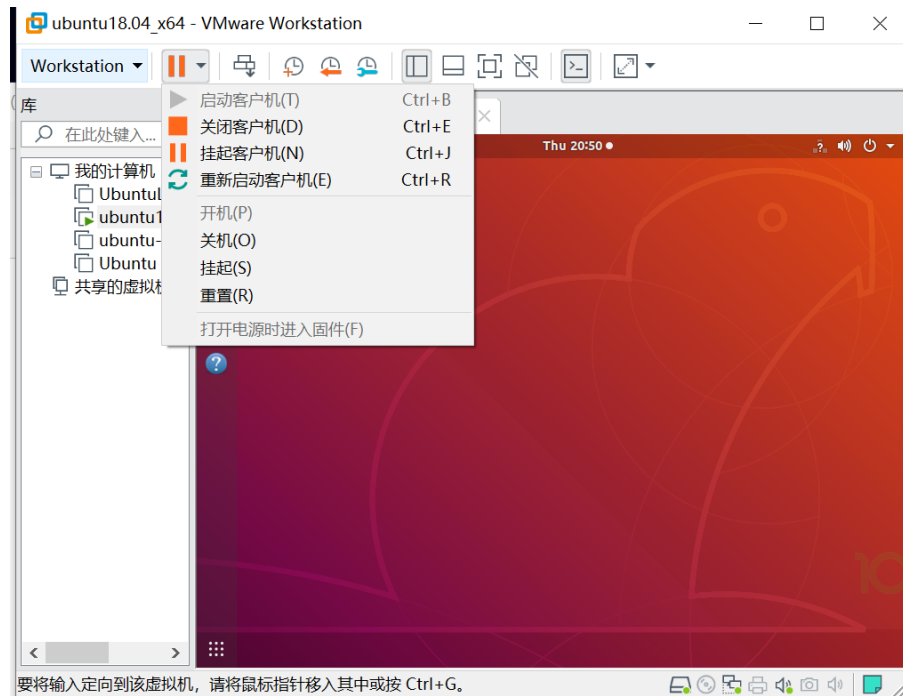


在弹出的对话框中我们可以进行重启或者关机操作，点击取消按钮可退出此对话框，如下图所示：

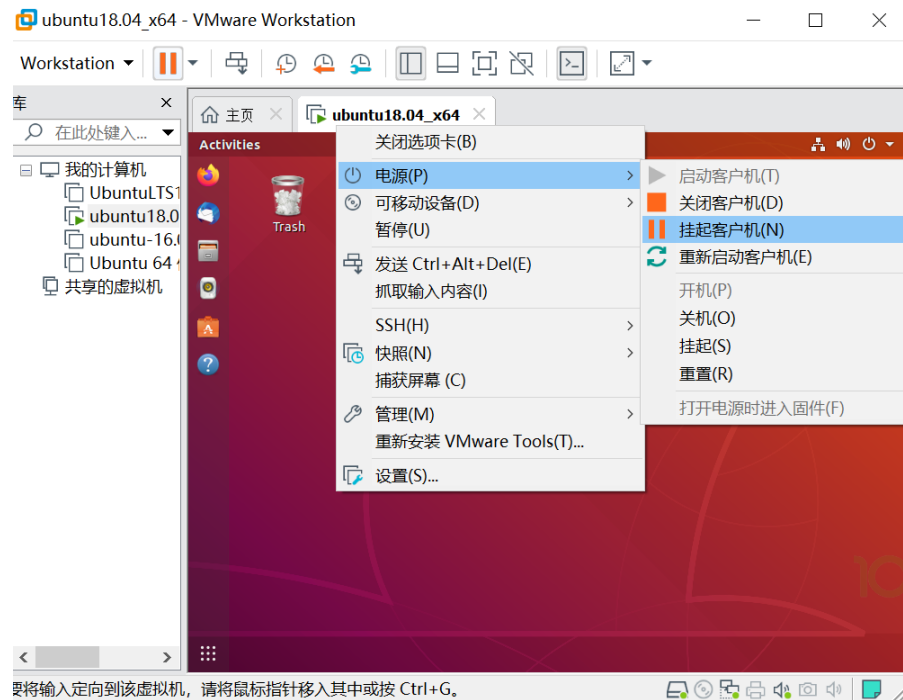


到这里，细心的读者可能会发现，在 windows 系统下我们还有一个选项就是“睡眠”，在 Ubuntu 中没有睡眠选项。其实我们可以通过 VMware 软件来实现虚拟机系统的睡眠操作，那就是挂起操作，将虚拟机系统挂起后，我们下次可以直接将虚拟机恢复到挂起时的状态。将虚拟机挂起非常简单，VMware 导航栏上的电源操作图标，或者在虚拟机的选项卡上右键唤出的菜单的电源选项中也有挂起操作，如下所示：

在 VMware 导航栏上的电源操作图标进行挂起：



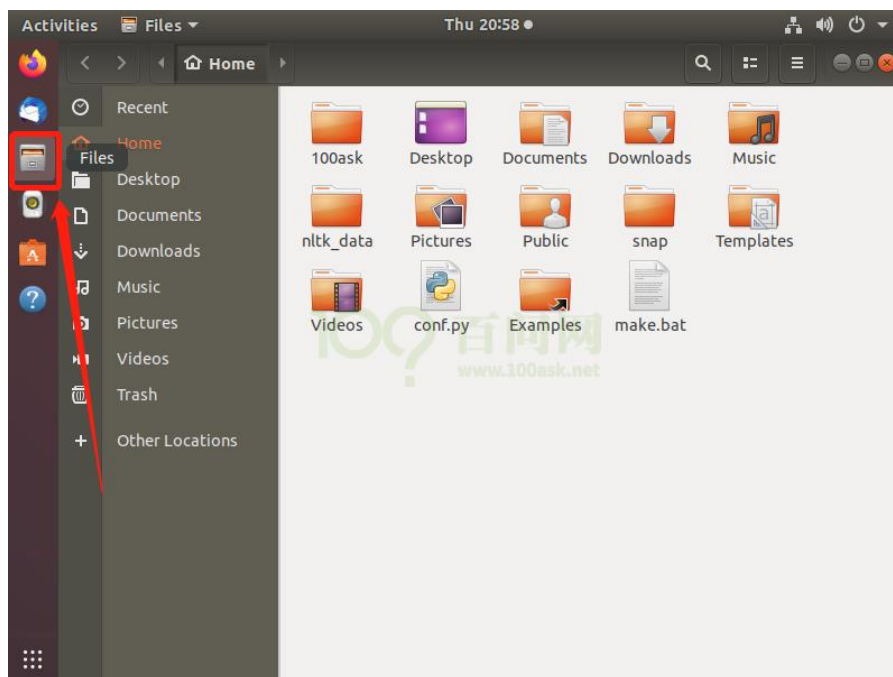
在虚拟机的选项卡上右键唤出的菜单的电源选项中进行挂起：



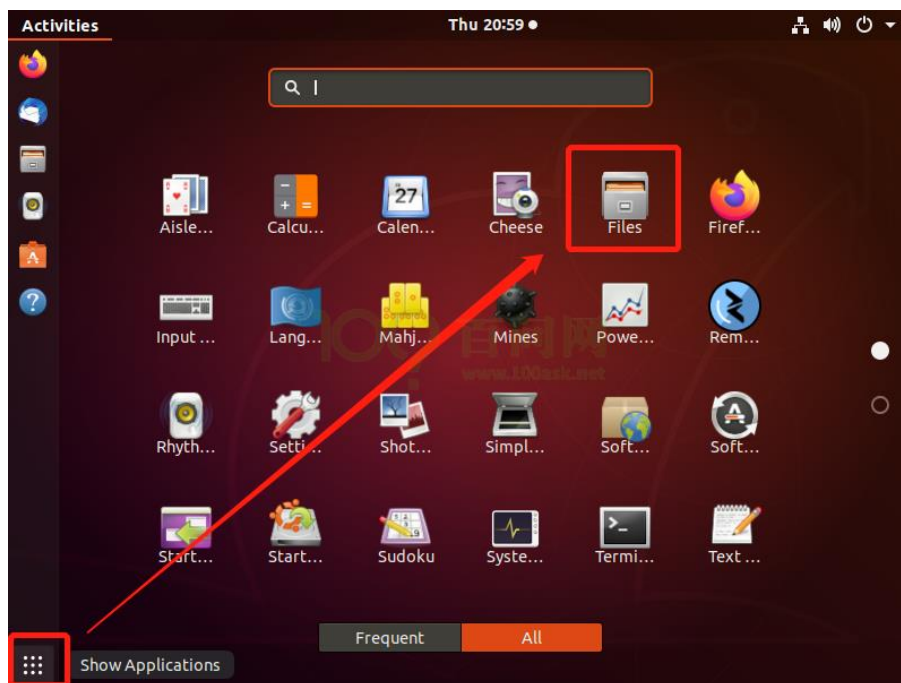
3.2.5 文件浏览器

每个带有 GUI 的系统都应该有文件浏览器，我们使用的桌面版本的 Ubuntu 也不例外，那么 Ubuntu 的文件浏览器怎么打开呢？

其实要打开 Ubuntu 的文件浏览器非常简单，文件浏览器在 Ubuntu 默认的左侧导航栏中可以直接打开，如下图所示：



或者我们可以在所有的应用中找到文件浏览器打开，如下图所示：



打开文件浏览器之后，我们就可以像在 windows 系统下利用文件资源管理器那样浏览磁盘中所有的文件。

3.3 Linux 常用命令

3.3.1 Linux 命令行介绍

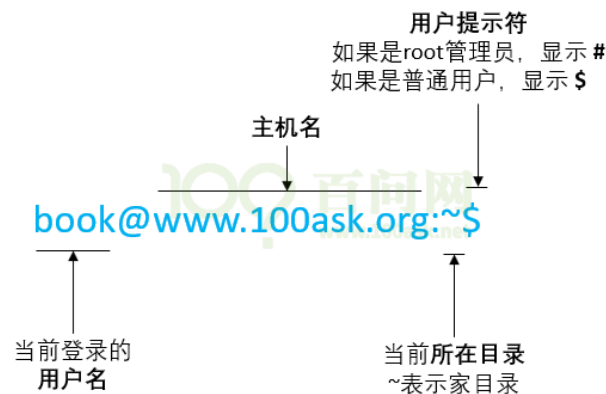
1. Linux Shell 简介

Shell 的意思是“外壳”，在 Linux 中它是一个程序，比如/bin/sh、/bin/bash 等。它负责接收用户的输入，根据用户的输入找到其他程序并运行。比如我们输入“ls”并回车时，shell 程序找到“ls”程序并运行，把结果打印出来。

Shell 有很多种实现，我们常用 bash。

2. Linux 命令的提示符

在 Ubuntu 中打开终端后，即可看到类似下图的提示符：

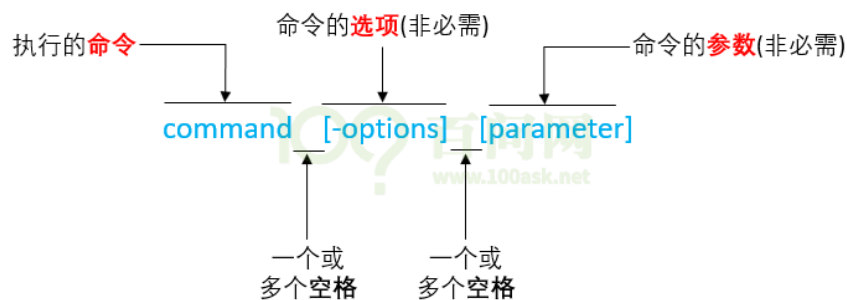


提示符中各项含义在上图中都列出来了。

3. Linux 命令的格式

Linux 命令一般由三部分组成：

- ① command 命令
- ② options 选项
- ③ parameter 参数



说明：

- ① [] 中括号表示 该部分可选，可有可无，需要根据命令的实际需要而添加
- ② 命令、选项、参数都以空格分隔，不管几个空格都算一个空格
- ③ 命令输入完毕后，按回车“Enter”键启动

示例：



4. 记住命令并不难，先背几个单词

序号	英语单词及含义	
1	directory	目录
2	change	改变
3	list	列出
4	print	打印
5	remove	删除
6	copy	复制
7	move	移动
8	clear	清除

5. 绝对路径和相对路径

Linux 下的根目录为“/”，从根目录下出发可以找到任意目录、任意文件。从根目录开始表示目录或文件的方法称为“绝对路径”。比如：

```
/home/book  
/home/book/1.txt  
/bin/pwd
```

有时候使用绝对路径太过麻烦，可以使用相对路径。假设当前正位于 `/home/book` 目录下，那么：

```
./1.txt      表示当前目录下的 1.txt，即 /home/book/1.txt；“.”表示当前目录  
../book/1.txt 表示当前目录的上一级目录里，book 子目录下的 1.txt  
“/home/book/..”就是“/home”目录，“..”表示上一级目录
```

- 使用 `.` 表示当前路径
- 使用 `..` 表示上一级路径
- 使用 `../..` 表示上上级路径，依此类推。

3.3.2 目录/文件操作命令

1. pwd

- 命令: pwd
- 英文来源: print working directory
- 功能: 打印当前所在路径
- 命令格式:

命令	选项	参数
pwd	\	\

注: pwd没有其他选项

◆ 示例: pwd

```
book@www.100ask.org:~$ pwd  
/home/book
```

示例功能: 显示当前所在路径

2. cd

- 命令: cd
- 英文来源: change directory
- 功能: 改变路径、切换路径
- 命令格式:

命令	选项	参数
cd	\	[目录]

◆ 示例: cd /home/

```
book@www.100ask.org:~$ pwd  
/home/book  
book@www.100ask.org:~$ cd /home/  
book@www.100ask.org:/home$ pwd  
/home
```

示例功能: 切换到/home路径

cd 命令有些缩略用法:

```
$ cd - // 进入上次目录, 比如先进入 a 目录再进入 b 目录, 执行此命令后即回到 a 目录  
$ cd ~ // 进入家目录
```

3. mkdir

- 命令: mkdir
- 英文来源: **make** **directory**
- 功能: 创建目录
- 命令格式:

命令	选项	参数
mkdir	-p	[目录]

◆ 示例: mkdir dir0

```
book@www.100ask.org:/work/001_linux_basic$ mkdir dir0
book@www.100ask.org:/work/001_linux_basic$ ls -l
total 4
drwxrwxr-x 2 book book 4096 7月 27 14:57 dir0
book@www.100ask.org:/work/001_linux_basic$
```

示例功能: 创建一个目录

◆ 示例: mkdir -p dir1/dir2

```
book@www.100ask.org:/work/001_linux_basic$ mkdir -p dir1/dir2
book@www.100ask.org:/work/001_linux_basic$ ls -l
total 4
drwxrwxr-x 3 book book 4096 7月 27 14:58 dir1
book@www.100ask.org:/work/001_linux_basic$ cd dir1/
book@www.100ask.org:/work/001_linux_basic/dir1$ ls -l
total 4
drwxrwxr-x 2 book book 4096 7月 27 14:58 dir2
book@www.100ask.org:/work/001_linux_basic/dir1$
```

示例功能: 创建目录及子目录

4. rmdir

- 命令: rmdir
- 英文来源: **remove** **directory**
- 功能: 删除目录
- 命令格式:

命令	选项	参数
rmdir	\	[目录]

◆ 示例: rmdir dir1

```
book@www.100ask.org:/work/001_linux_basic$ ls
dir1 dir2
book@www.100ask.org:/work/001_linux_basic$ rmdir dir1
book@www.100ask.org:/work/001_linux_basic$ ls
dir2
```

示例功能: 删除一个空目录

◆ 示例: rmdir dir1

```
book@www.100ask.org:/work/001_linux_basic/dir1$ ls
1.txt
book@www.100ask.org:/work/001_linux_basic/dir1$ cd ..
book@www.100ask.org:/work/001_linux_basic$ rmdir dir1
rmdir: failed to remove 'dir1': Directory not empty
```

示例功能: 不能删除一个非目录

5. ls

- 命令: ls
- 英文来源: list
- 功能: 列出目录内容
- 命令格式:

命令	选项			参数
ls	-l	-a	-h	[目录]

使用示例:

◆ 示例: ls

```
book@www.100ask.org:/work/001_linux_basic$ ls
dir1 dir2 file1 file2
book@www.100ask.org:/work/001_linux_basic$
```

示例功能: 显示当前目录下文件

◆ 示例: ls -l

```
book@www.100ask.org:/work/001_linux_basic$ ls -l
total 8
drwxrwxr-x 2 hceng hceng 4096 7月 27 11:31 dir1
drwxrwxr-x 2 hceng hceng 4096 7月 27 11:31 dir2
-rw-rw-r-- 1 hceng hceng 0 7月 27 11:31 file1
-rw-rw-r-- 1 hceng hceng 0 7月 27 11:31 file2
book@www.100ask.org:/work/001_linux_basic$
```

示例功能: 显示文件更完整信息
use a long listing format.

◆ 示例: ls -a

```
book@www.100ask.org:/work/001_linux_basic$ ls -a
. .dir0 dir1 dir2 .file0 file1 file2
book@www.100ask.org:/work/001_linux_basic$
```

示例功能: 显示当前目录下文件及隐藏文件
do not ignore entries starting with.

◆ 示例: ls -la

```
book@www.100ask.org:/work/001_linux_basic$ ls -la
total 20
drwxrwxrwx 5 hceng hceng 4096 7月 27 14:54 .
drwxrwxrwx 6 root root 4096 7月 27 11:30 ..
drwxrwxr-x 2 book book 4096 7月 27 11:51 .dir0
drwxrwxr-x 2 book book 4096 7月 27 14:54 dir1
drwxrwxr-x 2 book book 4096 7月 27 14:54 dir2
-rw-rw-r-- 1 book book 0 7月 27 11:51 .file0
-rw-rw-r-- 1 book book 0 7月 27 14:54 file1
-rw-rw-r-- 1 book book 0 7月 27 14:54 file2
book@www.100ask.org:/work/001_linux_basic$
```

示例功能: "-l" 和 "-a" 组合选项,
显示所有文件及完整信息。

◆ ls -lh // h表示--human-readable, 大小以K/M/G等可读方式列出来

```
book@www.100ask.org:/work/001_linux_basic$ ls -lh
total 8.0K
drwxrwxr-x 2 book book 4.0K 7月 27 14:54 dir1
drwxrwxr-x 2 book book 4.0K 7月 27 14:54 dir2
-rw-rw-r-- 1 book book 0 7月 27 14:54 file1
-rw-rw-r-- 1 book book 0 7月 27 14:54 file2
```

① 文件属性 ② 链接数 ③ 文件所有者 ④ 文件所属用户组 ⑤ 文件大小 ⑥ 最后修改时间 ⑦ 文件名

6. cp

- 命令: cp
- 英文来源: copy
- 功能: 复制文件或目录
- 命令格式:

命令	选项	参数
cp	-r -f -d	文件路径1 文件路径2

◆ 示例: cp file1 file2

```
book@www.100ask.org:/work/001_linux_basic$ ls
file1
book@www.100ask.org:/work/001_linux_basic$ cp file1 file2
book@www.100ask.org:/work/001_linux_basic$ ls
file1 file2
book@www.100ask.org:/work/001_linux_basic$
```

示例功能: 复制文件

◆ 示例: cp dir1/* dir2

```
book@www.100ask.org:/work/001_linux_basic$ ls dir1/
file1 file2
book@www.100ask.org:/work/001_linux_basic$ cp dir1/* dir2/
book@www.100ask.org:/work/001_linux_basic$ ls dir2/
file1 file2
book@www.100ask.org:/work/001_linux_basic$
```

示例功能: 拷贝dir1目录里的所有文件

◆ 示例: cp -r dir1/ dir2/

```
book@www.100ask.org:/work/001_linux_basic$ ls
dir1
book@www.100ask.org:/work/001_linux_basic$ cp -r dir1 dir2
book@www.100ask.org:/work/001_linux_basic$ ls
dir1 dir2
```

示例功能: 复制dir1文件夹

复制目录时, 常用如下命令:

```
$ cp -rfd dir_a dir_b
```

r: recursive, 递归地, 即复制所有文件

f: force, 强制覆盖

d: 如果源文件为链接文件, 也只是把它作为链接文件复制过去, 而不是复制实际文件

7. rm

- 命令: rm
- 英文来源: remove
- 功能: 删除文件或目录
- 命令格式:

命令	选项	参数
rm	-r	-p 文件 或 文件夹

◆ 示例: rm file1

```
book@www.100ask.org:/work/001_linux_basic$ ls
file1 file2
book@www.100ask.org:/work/001_linux_basic$ rm file1
book@www.100ask.org:/work/001_linux_basic$ ls
file2
book@www.100ask.org:/work/001_linux_basic$
```

示例功能: 删除文件

◆ 示例: rm -r dir1/

```
book@www.100ask.org:/work/001_linux_basic$ ls
dir1 dir2
book@www.100ask.org:/work/001_linux_basic$ rm -r dir1/
book@www.100ask.org:/work/001_linux_basic$ ls
dir2
book@www.100ask.org:/work/001_linux_basic$
```

示例功能: 删除文件夹

删除目录时, 常用如下命令:

```
$ rm -rf dir_a
```

r: recursive, 递归地, 即复制所有文件

f: force, 强制删除

8. cat

- 命令: cat
- 英文来源: \
- 功能: 串联文件的内容并打印出来
- 命令格式:

命令	选项	参数
cat	\	文件

◆ 示例: cat file1.txt file2.txt

```
book@www.100ask.org:/work/001_linux_basic$ ls
file1.txt file2.txt
book@www.100ask.org:/work/001_linux_basic$ cat file1.txt file2.txt
hello world.
www.100ask.org
book@www.100ask.org:/work/001_linux_basic$
```

示例功能: 串联文件并依次全部打印在标准输出中
Concatenate files and print on the standard output

9. touch

- **命令:** touch
- **英文来源:** \
- **功能:** 修改文件的时间, 如果文件不存在则创建空文件
- **命令格式:**

命令	选项	参数
touch	\	文件名

◆ 示例: touch file

```
book@www.100ask.org:/work/001_linux_basic$ touch file
book@www.100ask.org:/work/001_linux_basic$ ls -l
total 0
-rw-rw-r-- 1 book book 0  7月 27 15:00 file
book@www.100ask.org:/work/001_linux_basic$
```

示例功能: 创建一个文件file

3.3.3 改变文件的权限和属性

chgrp: 改变文件所属用户组

chown: 改变文件所有者

chmod: 改变文件的权限

1. chgrp

改变文件所属用户组

chgrp **【-R】** dirname/filename ...

-R: 进行递归的持续更改, 也连同子目录下的所有文件、目录都更新成为这个用户组之意。常常用在更改某一目录内所有文件的情况。

范例:

```
chgrp hy install.log
```

将 install.log 文件的用户组改为 hy 用户组。注意 hy 用户组必须要在/etc/group 文件内存在才可以。

2. chown

改变文件的所有者

chown **[-R]** 账号名 文件或目录

chown **[-R]** 账号名:组名 文件或目录

-R: 也是递归子目录。

范例:

```
chown bin install.log
```

```
chown book:book install.log
```

改变文件所有者和用户组的这两个命令的应用场景: 复制文件, 由于复制行为会复制执行者的属性和权限, 因此复制后需要改变文件所属用户、用户组等。

3. chmod

改变文件的权限

文件权限有两种设置方法：数字类型改变权限和符号改变权限。

首先说明各个权限对应的数字：

- ① r: 4 或 0
- ② w: 2 或 0
- ③ x: 1 或 0

这 3 种权限的取值相加后，就是权限的数字表示。

例如：文件 a 的权限为 “-rwxrwx---”，它的数值表示为：

- ① owner = rwx = 4+2+1 = 7
- ② group = rwx = 4+2+1 = 7
- ③ others = --- = 0+0 +0 = 0

所以在设置权限时，该文件的权限数字就是 770。使用数值改变文件权限的命令如下：

```
chmod [-R] xyz 文件或目录
```

- ① xyz : 代表权限的数值，如 770。
- ② -R : 以递归方式进行修改，比如修改某个目录下所有文件的属性。

范例：

```
chmod 777 .bashrc
```

将文件.bashrc 这个文件的所有权限设置都启用。

2) 符号类型改变文件权限方式

使用 u、g、o 三个字母代表 user、group、others 3 中身份。此外 a 代表 all，即所有身份。

范例：

```
chmod u=rwx,go=rx .bashrc
```

也可以增加或去除某种权限，“+”表示添加权限，“-”表示去除权限：

```
chmod a+w .bashrc
```

```
chmod a-x .bashrc
```

3.3.4 查找/搜索命令

1. find

在 Windows 中搜索文件，一般查找文件需要传入两个条件：

- a. 在哪些目录中查找；
- b. 查找的内容；

在 Linux 中，查找文件的也需要这两个条件，不同于 Windows 使用搜索框查找，Linux 中使用 find 命令查找文件。

find 命令格式为：

```
find 目录名 选项 查找条件
```

举例 1：

```
$ find /home/book/dira/ -name " test1.txt "
```

说明：

- a) /home/book/dira/指明了查找的路径。
- b) “-name”表明以名字来查找文件。
- c) “test1.txt”，就指明查找名为“test1.txt”的文件。

举例 2：

```
$ find /home/book/dira/ -name " *.txt "
```

说明：查找指定目录下面所有以“.txt”结尾的文件，其中“*”是通配符。

举例 3：

```
find /home/book/dira/ -name "dira"
```

说明：查找指定目录下面是否存在“dira”这个目录或文件，“dira”是名称。

注意：

- ① 如果没有指定查找目录，则为当前目录。

```
$ find . -name " *.txt " //其中.代表当前路径。
```

```
$ find -name " *.txt " //没加路径，默认是当前路径下查找。
```

- ② find 还有一些高级的用法，如查找最近几天(几个小时)之内(之前)有变动的文件

```
$ find /home/book -mtime -2 //查找/home目录下两天内有变动的文件。
```

2. grep

grep 命令的作用是查找文件中符合条件的字符串，其格式如下：

```
grep [选项] [查找模式] [文件名]。
```

假设 dira 目录的 test1.txt 和 dirb 目录的 test1.txt 都含有如下内容：aaa AAAAAA abc abcabcab abcabacba match_pattern nand->erase。

通过查找字符串，希望显示如下内容：

- 1) 所在的文件名-----grep 查找时默认已经显示目标文件名
- 2) 所在的行号-----使用-n 选项。

grep -rn “字符串” 文件名 r(recursive)：递归查找 n(number)：显示目标位置的行号 字符串：要查找的字符串 文件名：要查找的目标文件，如果是*则表示查找当前目录下的所有文件和目录。

举例：

```
//在test1.txt中查找字符串abc grep -rn "abc" * 在当前目录递归查找字符串abc  
$ grep -n "abc" test1.txt
```

注意：可以加入-w 全字匹配。

可以在 grep 的结果中再次执行 grep 搜索，比如搜索包含有 ABC 的头文件，可执行如下命令：

```
$ grep "ABC" * -nR | grep "\.h"
```

上述命令把第 1 个命令“grep “ABC” * -nR”通过管道传给第 2 个命令。

即第 2 个命令在第 1 个命令的结果中搜索。

3.3.5 压缩/解压命令

压缩的目的：在网络传递文件时，可以先将文件压缩，然后传递压缩后的文件，从而减少网络带宽。接收到文件后，解压即可。

压缩的类型有 2 种：有损压缩、无损压缩：

a. 有损压缩：

如 mp4 视频文件，在压缩过程中减少了很多帧的数据，但是对观看者而言没有影响。当然 mp3 音乐文件也是有损压缩。

b. 无损压缩：

如普通文件的压缩，为了保证信息的正确传递，不希望文件经过压缩或解压后，出现任何差异。后面讲解的都是无损压缩。

单个文件的压缩(解压)使用 gzip 和 bzip2，多个文件和目录使用 tar。

1. gzip

gzip 的常用选项：

- l(list) 列出压缩文件的内容。
- k(keep) 在压缩或解压时，保留输入文件。
- d(decompress) 将压缩文件进行解压缩。

举例：

1) 查看压缩文件

```
$ gzip -l pwd.1.gz
```

2) 解压文件

```
$ gzip -kd pwd.1.gz //该压缩文件是以.gz结尾的单个文件
```

3) 压缩文件

```
$ gzip -k mypwd.1 /得到了一个.gz结尾的压缩文件
```

注意：

1) 如果 gzip 不加任何选项，此时为压缩

压缩完该文件会生成后缀为.gz 的压缩文件，并删除原来的文件。

所以，推荐使用 gzip -k 来压缩源文件，这样会保留原来的文件。

2) 相同的文件内容，如果文件名不同，压缩后的大小也不同。

3) gzip 只能压缩单个文件，不能压缩目录。

2. bzip2

bzip2 的常用选项:

- k(keep) 在压缩或解压时, 保留输入文件;
- d(decompress) 将压缩文件进行解压缩;

1) 压缩文件

```
$ bzip2 -k mypwd.1 得到一个.bz2后缀的压缩文。
```

2) 解压文件

```
$ bzip2 -kd mypwd.1.bz2
```

注意:

- 1) 如果 bzip2 不加任何选项, 此时为压缩
压缩完该文件会生成后缀为 .bz2 的压缩文件, 并删除原来的文件。
所以说, 推荐使用 bzip2 -k 来压缩文件, 这样可以保留原来的文件。

- 2) bzip2 只能压缩单个文件, 不能压缩目录。

单个文件的压缩使用 gzip 或 bzip2, 压缩有两个参数:

- 1) 压缩时间
- 2) 压缩比。

一般情况下, 小文件使用 gzip 来压缩, 大文件使用 bzip2 来压缩。bzip2 的的压缩率更高。

3. tar

tar 常用选项:

- c(create): 表示创建用来生成文件包。
- x: 表示提取, 从文件包中提取文件。
- t: 可以查看压缩的文件。
- z: 使用gzip方式进行处理, 它与”c”结合就表示压缩, 与”x”结合就表示解压缩。
- j: 使用bzip2方式进行处理, 它与”c”结合就表示压缩, 与”x”结合就表示解压缩。
- v(verbose): 详细报告tar处理的信息。
- f(file): 表示文件, 后面接着一个文件名。-C <指定目录> 解压到指定目录。

例 1: tar 打包、gzip 压缩

1) 把目录 dira 压缩、打包为 dira.tar.gz 文件:

```
$ tar czvf dira.tar.gz dira.
```

注意: “tar -czvf”与“tar czvf”是一样的效果, 所以说, 后面统一取消“-”。

2) 查看压缩文件:

```
$ tar tvf dira.tar.gz
```

3) 解压文件, 可以用-C 指定解压到哪个目录:

```
$ tar xzvf dira.tar.gz //解压到当前目录
```

```
$ tar xzvf dira.tar.gz -C /home/book //解压到/home/book。
```

例 2: tar 打包、bzip2 压缩

1) 把目录 dira 压缩、打包为 dira.tar.bz2 文件

```
$ tar cjvf dira.tar.bz2 dira
```

2) 查看压缩文件

```
$ tar tvf dira.tar.bz2
```

3) 解压文件, 可以用-C 指定解压到哪个目录

```
$ tar xjvf dira.tar.bz2 //解压到当前目录:
```

```
$ tar xjvf dira.tar.bz2 -C /home/book //解压到/home/book
```

3.3.6 网络命令

1. ifconfig

查看网络、设置 IP。ifconfig 常用选项:

-a : 显示所有网卡接口

up: 激活网卡接口

down: 关闭网卡接口

address: xxx.xxx.xxx.xxx, IP地址

示例:

1) ifconfig: 查看当前正在使用的网卡

```
$ ifconfig
ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.137 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::4f45:59fb:ddb7:c274 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:ab:1d:05 txqueuelen 1000 (Ethernet)
    RX packets 998794 bytes 176687882 (176.6 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 801210 bytes 138020387 (138.0 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

2) ifconfig -a: 查看所有网卡

```
$ ifconfig -a
ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.137 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::4f45:59fb:ddb7:c274 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:ab:1d:05 txqueuelen 1000 (Ethernet)
    RX packets 998889 bytes 176699569 (176.6 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 801287 bytes 138033739 (138.0 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=8<LOOPBACK> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1000 (Local Loopback)
    RX packets 51460 bytes 3249553 (3.2 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 51460 bytes 3249553 (3.2 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

3) 设置网 IP:

```
$ sudo ifconfig ens160 192.168.1.137
```

2. route 和 DNS

确保 Windows 和 Ubuntu 的网络能互相 ping 通之后，如果 Ubuntu 无法上网，原因通常有 2 个：路由没设置好，DNS 没设置好。

如果执行以下命令不成功，表示路由没设置好：

```
$ ping 8.8.8.8  
connect: Network is unreachable
```

如果“ping 8.8.8.8”成功，但是“ping www.baidu.com”不成功，则是 DNS 没设置好：

```
$ ping www.baidu.com  
ping: unknown host www.baidu.com
```

DNS 的设置比较简单，8.8.8.8 是好记好用的 DNS 服务器，修改 Ubuntu 中的/etc/resolv.conf 文件，内容如下：

```
nameserver 8.8.8.8
```

路由信息使用 route 命令查看，其输出信息可以参考链接：

<https://akaedu.github.io/book/ch36s05.html>

摘录如下：

假设某主机上的网络接口配置和路由表如下：

```
$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:C2:8D:7E
          inet addr:192.168.10.223  Bcast:192.168.10.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:10 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:0 (0.0 b)  TX bytes:420 (420.0 b)
          Interrupt:10 Base address:0x10a0

eth1      Link encap:Ethernet  HWaddr 00:0C:29:C2:8D:88
          inet addr:192.168.56.136  Bcast:192.168.56.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:603 errors:0 dropped:0 overruns:0 frame:0
          TX packets:110 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:55551 (54.2 Kb)  TX bytes:7601 (7.4 Kb)
          Interrupt:9 Base address:0x10c0

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:37 errors:0 dropped:0 overruns:0 frame:0
          TX packets:37 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:3020 (2.9 Kb)  TX bytes:3020 (2.9 Kb)

$ route
Kernel IP routing table
Destination    Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.10.0   *               255.255.255.0   U      0      0      0 eth0
192.168.56.0   *               255.255.255.0   U      0      0      0 eth1
127.0.0.0      *               255.0.0.0       U      0      0      0 lo
default        192.168.10.1   0.0.0.0         UG     0      0      0 eth0
```

上述 route 命令输出信息中各项的含义请看下表：

Destination	目标网段或者主机
Gateway	网关地址，“*” 表示目标是本主机所属的网络，不需要路由
Genmask	网络掩码
Flags	标记。一些可能的标记如下：

	U - 路由是活动的
	H - 目标是一个主机
	G - 路由指向网关
	R - 恢复动态路由产生的表项
	D - 由路由的后台程序动态地安装
	M - 由路由的后台程序修改
	! - 拒绝路由
Metric	路由距离，到达指定网络所需的中转数
Ref	路由项引用次数
Use	此路由项被路由软件查找的次数
Iface	该路由表项对应的输出接口

在上面的例子中，这台主机有两个网络接口：

- ① 一个网络接口连到 192.168.10.0/24 网络
- ② 另一个网络接口连到 192.168.56.0/24 网络。

如果要发送的数据包的目的地址是 192.168.56.3，跟第一行的子网掩码做与运算得到 192.168.56.0，与第一行的目的网络地址不符，再跟第二行的子网掩码做与运算得到 192.168.56.0，正是第二行的目的网络地址，因此从 eth1 接口发送出去，由于 192.168.56.0/24 正是与 eth1 接口直接相连的网络，因此可以直接发到目的主机，不需要经路由器转发。

如果要发送的数据包的目的地址是 202.10.1.2，跟前三行路由表条目都不匹配，那么就要按缺省路由条目，从 eth0 接口发出去，首先发往 192.168.10.1 路由器，再让路由器根据它的路由表决定下一跳地址。

可以使用 route 命令管理路由。

示例：

1) 添加路由：首先得确定网关 IP，假设为 192.168.1.1

```
$ sudo route add default gw 192.168.1.1
$ ping 8.8.8.8 // 验证
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=53 time=19.8 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=53 time=19.8 ms
```

2) 删除路由：

```
$ sudo route del default gw 192.168.1.1
```

3.3.7 vi 编辑器

vi 是一个命令，也是一个命令行下的编辑器，它有如下功能：

- 打开文件、新建文件、保存文件
- 光标移动
- 文本编辑
- (多行间|多列间)复制、粘贴、删除
- 查找和替换

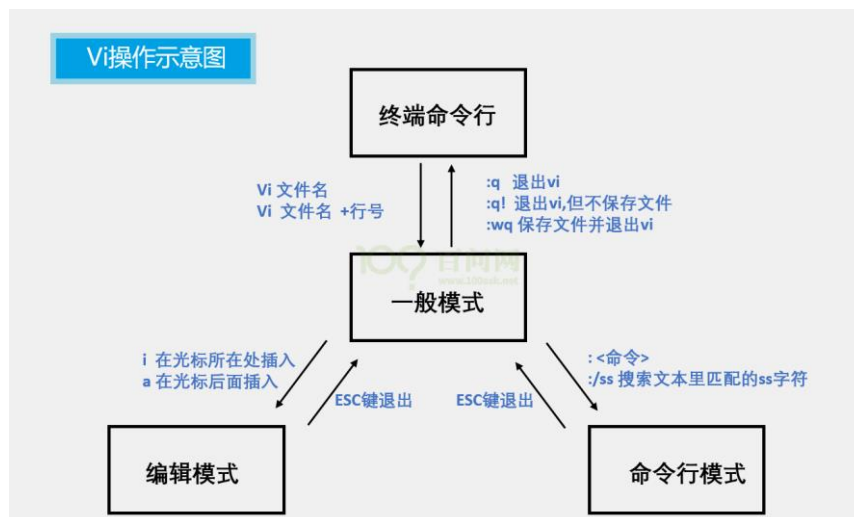
很多人不习惯在命令行下编辑文件，实际开发中也不会经常在命令行下编辑文件。但是在 Linux 系统 中对文件做些简单修改时，使用 vi 命令的效率非常高。并且在很多时候，比如现场调试时，并没有 GUI 形式的编辑工具，vi 是唯一选择。

1. 模式

vi 编辑器有三种模式，各个模式侧重点不一样：

- 一般模式（光标移动、复制、粘贴、删除）
- 编辑模式（编辑文本）
- 命令行模式（查找和替换）

vi 编辑器的三种模式间切换如下图所示



注意：

- 当不知道处于何种模式时，按 `ESC` 键返回到一般模式。
- `wq` (write quit)
- `i` (insert)

2. 文件的打开、新建、保存

打开文件、新建文件，命令如下(如果文件存在则打开文件，否则新建文件并打开)：

```
$ vi 文件名
```

修改结束之后，输入“:”进入命令行模式，再输入“wq”保存退出：

```
:wq 保存并退出文件
```

注意：如果文件不存在，也需要输入“:wq”才可以保存新文件，否则不会新建文件。

在编辑完成时，返回一般模式，方法如下：

- 输入“:w”则保存文件，如果已经保存文件，输入“:q”则退出文件
- 直接输入“:wq”保存并退出
- 如果不想保存被修改的内容，则输入“:q!”强制退出。

这些命令列表如下：

命令	描述
x	保存当前文档并且退出。
q	退出。
w	保存文档。
q!	退出 vi/vim，不保存文档。

3. 编辑文件

打开文件后，默认处于“一般模式”，这时可以输入以下字母：

指令	描述
i	在当前光标所在字符的前面，转为编辑模式。
I	在当前光标所在行的行首转换为编辑模式。
a	在当前光标所在字符的后面，转为编辑模式。
A	在光标所在行的行尾，转换为编辑模式。
o	在当前光标所在行的下方，新建一行，并转为编辑模式。
O	在当前光标所在行的上方，新建一行，并转为编辑模式。

4. 快速移动光标

在一般模式下，可以使用下面快捷键移动光标或是翻页：

移动光标	
h (或左方向键)	光标左移一个字符。
l (或右方向键)	光标右移一个字符。
j (或下方向键)	光标下移一行。
k (或上方向键)	光标上移一行。
nG 或 ngg	光标移动到第 n 行首。
n+	光标下移 n 行。
n-	光标上移 n 行。
屏幕翻滚	
Ctrl + f	屏幕向下翻一页，相当于下一页。
Ctrl + b	屏幕向上翻一页，相当于上一页。

详细介绍如下：

1) 快速的定位到某一行：文件头、文件尾、指定某一行

ngg //光标移至第n行的行首（n为数字, 想要跳转的行），

1gg //就跳到第一行的行首，就是文件头

2gg //就跳到第二行的行首

G //转至文件结尾

2) 在某一列如何快速定位到某一列：

0 //（数字零）光标移至当前行行首

\$ //光标移至当前行行末

fx //搜索当前行中下一个出现字母x的地方

注意：当你不知道 vi 当前处于何种模式时，使用 esc 键返回到一般模式。

5. 文本复制、粘贴、删除、撤销

在一般模式下，可以执行以下命令。

复制、删除和粘贴	
cc	删除整行，并且修改整行内容。
dd	删除该行，不提供修改功能。
ndd	删除当前行向下 n 行。
x	删除光标所在的字符。
X	删除光标前面的一个字符。
nyy	复制当前行及其下面 n 行。
p	粘贴最近复制的内容。
s	删除光标所在字符。
r	替换光标处字符。

1) 复制

```
yy //复制当前行(y:yank(复制))  
nyy //复制当前行及其后的n-1行(n是数字)
```

2) 粘贴

```
p //粘贴(p:paste)
```

3) 删除

```
dd //删除光标所在行(d:delete)  
ndd //删除当前行及其后的n-1行(n是数字)  
x //删除光标所在位置的字符
```

4) 撤销

```
u //撤销上一步操作
```

6. 文本查找和替换

在一般模式下，可以执行以下命令。

1) 查找

```
/pattern //从光标开始处向文件尾搜索pattern，后按下n或N
```

注意：

n 在同一个方向重复上一次搜索命令

N 在反方向重复上一次搜索命令

注意：

在/pattern 之前先跳到第一行则进行全文件搜索。

2) 替换

```
:%s/p1/p2/g //将文件中所有的p1均用p2替换
```

```
:%s/p1/p2/gc //替换时需要确认
```

“s “ 全称： substitute 替换；

“g “ 全称： global 全局；

“c “ 全称： confirm， 确认

参考 <https://www.runoob.com/linux/linux-command-manual.html>

7. vi 编辑器使用示例

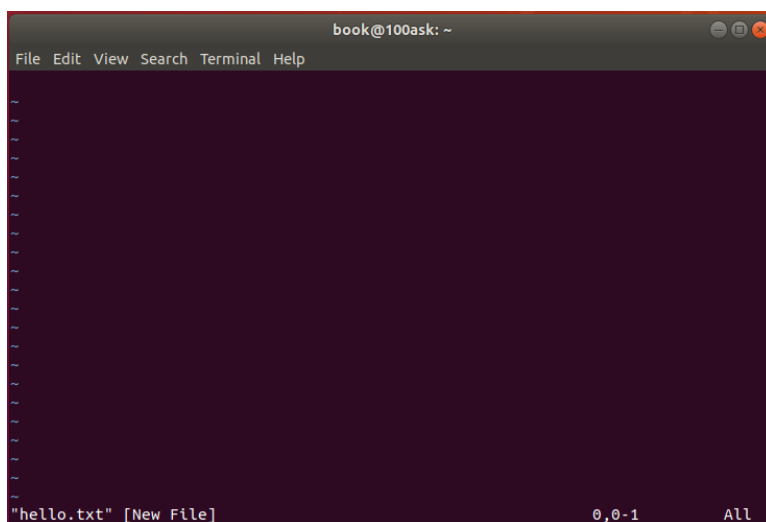
本例创建一个名为 `hello.txt` 的文件，添加如下 2 行文件字，再回退去删除第 2 行的字母“X”：

```
www.100ask.net  
wiki.100askX.net
```

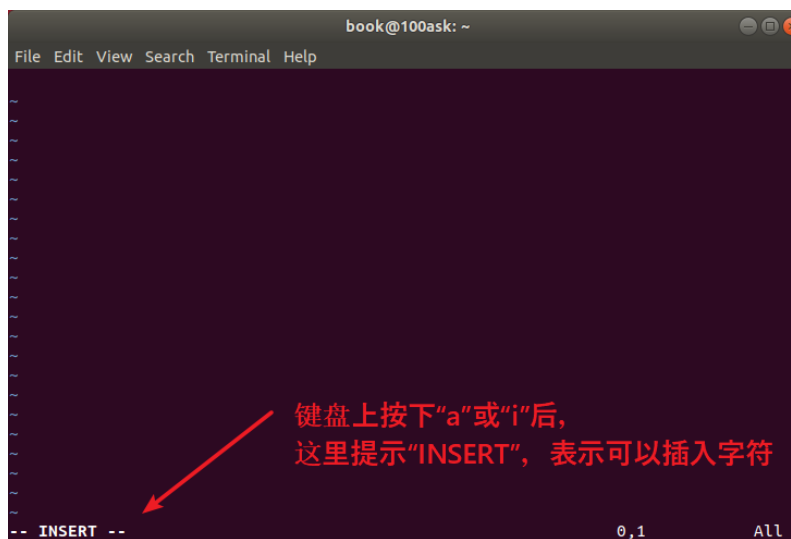
首先执行以下命令，如果当前目录下没有 `hello.txt` 它就会新建并打开该文件，否则直接打开该文件：

```
vi hello.txt
```

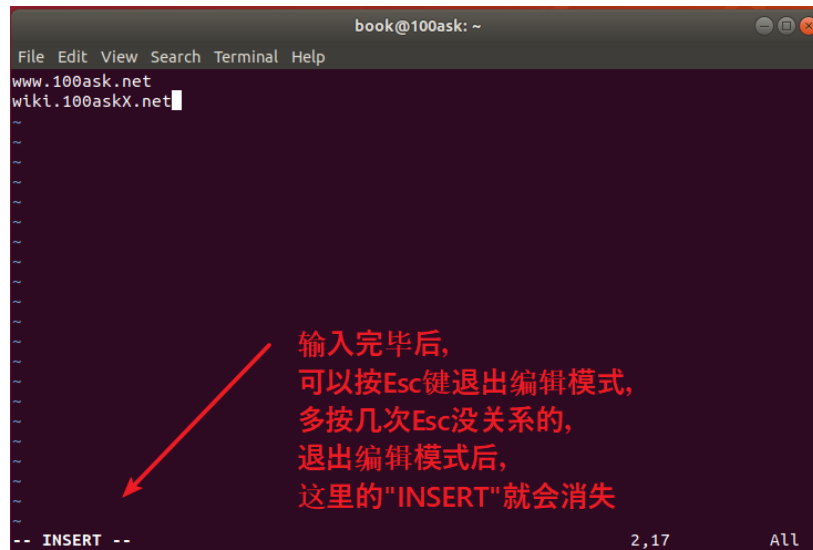
界面如下：



输入字符“i”，它表示“insert”，即在光标前输入字符，界面如下：



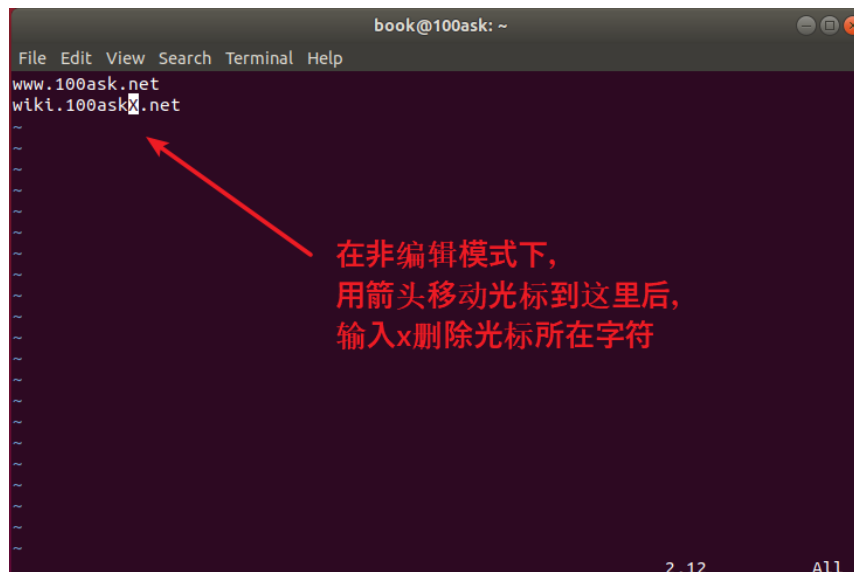
在上述界面中，可以输入字符了，可以使用回车键、删除键、箭头键等等，跟一般得文本工具没什么差别，如下：



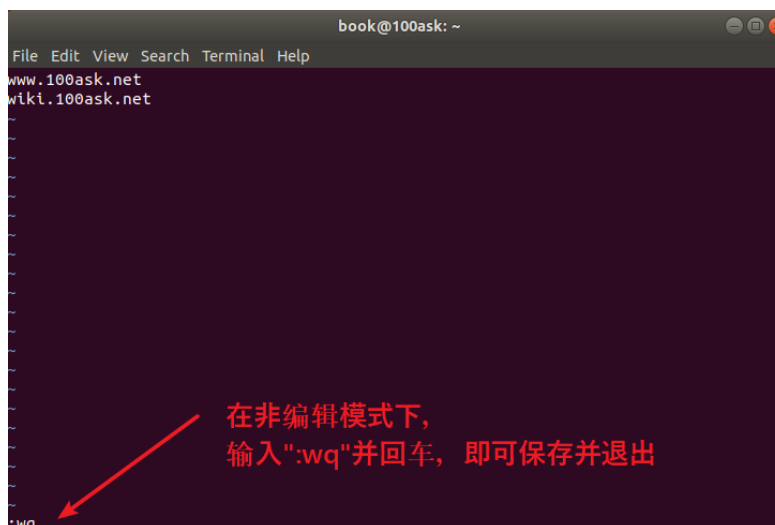
无论当前是否处于编辑模式，都可以使用箭头键移动光标。

在编辑模式下，使用删除键 (Backspace) 删除字符。

不在编辑模式下时，使用“x”删除光标所在的字符，如下：



最后输入“:wq”保存并退出，“:”表示进入命令行模式，“wq”表示“write and quit”，即写入并退出。如果不想保存则可以输入“:q!”，它表示“退出、不保存”。



最后执行“cat hello.txt”命令验证一下：

```
book@100ask:~$ cat hello.txt
www.100ask.net
wiki.100ask.net
book@100ask:~$
```

注意：本教程没有教你怎么在 Ubuntu 中安装中文输入法，因为我们是在 Windows 编辑源码，或是提倡多英文写注释。如果你需要在 Ubuntu 下使用中文，请自行百度。

3.3.8 其他命令

1. file

查看文件类型。其格式如下：

```
file 文件名
```

使用 gcc 编译得到的程序是运行于 PC 的，但是很多初学者经常把这些程序放到 ARM 板子上去运行，这时一般都会提示：

```
xxx not found
```

它并非“找不到”，而是格式不正确。

这时你可以执行“file xxx”查看它的类型，确定它是给 PC 还是给 ARM 编译的。

2. which 和 whereis

which 命令和 whereis 命令作用是查找命令或应用程序的所在位置，其格式如下：

```
which 命令名/应用程序名
```

```
whereis 命令名/应用程序名。
```

示例：

```
$ which pwd //定位到/bin/pwd
```

```
$ which gcc //定位到/usr/bin/gcc
```

```
$ whereis pwd //可得到可执行程序的位置和手册页的位置
```

3.4 Ubuntu 下包管理

3.4.1 软件包管理系统

像我们日常使用的 windows 提供的应用商店或者手机提供的的应用市场那样，大多数现代类 Unix 操作系统也都提供了一种中心化的机制用来搜索和安装软件。软件通常存放在存储库中，并通过包的形式对外进行分发。处理包的工作称为包管理。包提供了操作系统的基本组件，以及共享的库、应用程序、服务和文档。这个我们称为软件包管理系统，其除了安装软件外，它还提供了工具来更新已经安装的包。

大多数软件包系统都是围绕软件包文件的集合构建的。软件包文件通常是一个存档文件，它包含已编译的二进制文件和软件的其他资源，以及安装脚本。软件包文件同时也包含有价值的元数据，包括它们的依赖项，以及安装和运行它们所需的其他软件包的列表。

虽然这些包管理系统的功能和优点大致相同，但打包格式和工具会因平台（不同的 Linux 发行版）而异，如下表所示：

操作系统	格式	工具
Debian	.deb	apt, apt-cache, apt-get, dpkg
Ubuntu	.deb	apt, apt-cache, apt-get, dpkg
CentOS	.rpm	yum
Fedora	.rpm	dnf
FreeBSD	Ports, .txz	make, pkg

由上表可知，Debian 及其衍生版，如我现在使用的 Ubuntu，它们的包格式为 .deb。APT 这款先进的软件包管理工具提供了大多数常见的命令如：搜索存储库、安装软件包及其依赖项，并管理升级，等等。在

系统中，我们还可以使用 `dpkg` 程序来安装单个的 `deb` 文件，`APT` 命令作为底层 `dpkg` 的前端，有时也会直接调用它。

目前发布的 `debian` 衍生版大多数都包含了 `apt` 命令，它提供了一个简洁统一的接口，可用于通常由 `apt-get` 和 `apt-cache` 命令处理的常见操作。这个命令是可选的，但使用它可以简化一些任务。

3.4.2 apt 命令使用

上一小节介绍了软件包管理系统，本小节秉承上小节的内容，介绍软件包管理系统中 `apt` 命令。

我们习惯于使用 `windows` 系统以及智能手机，这两个下载和安装软件都是非常容易简单的。但是 `Ubuntu` 下该如何安装软件呢？`Ubuntu` 安装软件不像 `Windows` 下那样，直接双击 `.exe` 文件就开始安装了。一般来说 `Ubuntu` 下很多软件是需要先自行提供源码，使用源码自行编译，编译完成以后使用命令 “`install`” 来安装到系统中。当然 `Ubuntu` 下也有其它的软件安装方法，使用得最多的方法就是自行编译源码后进行安装，尤其是嵌入式 `Linux` 开发。命令 “`install`” 格式如下所示：

```
install [选项] ... [-T] 源文件 目标文件
install [选项] ... 源文件 ... 目录
install [选项] ... -t 目录 源文件 ... 或： install [选项] ... -d 目录 ...
```

“`install`” 命令是将文件（通常是编译后的文件）复制到目的位置，在上面得三种形式中，将源文件复制到目标文件或将多个源文件复制到一个已存在的目录中同时设置其所有权和权限模式。在第四种形式会创建指定的目录。命令 “`install`” 通常和命令 `apt-get`” 组合在一起使用的。

其实 `Ubuntu` 中我们可以直接安装很多的软件&游戏，但是对于刚刚接触 `Linux` 系统的我们往往会不知如何安装安装软件，其实我们利用软件包管理系统可以直接下载并安装所有通过认证的软件，其中 `Ubuntu` 下我们用的最多的下载工具：`APT` 下载工具，`APT` 下载工具可以实现软件自动下载、配置、安装二进制或者源码的功能。

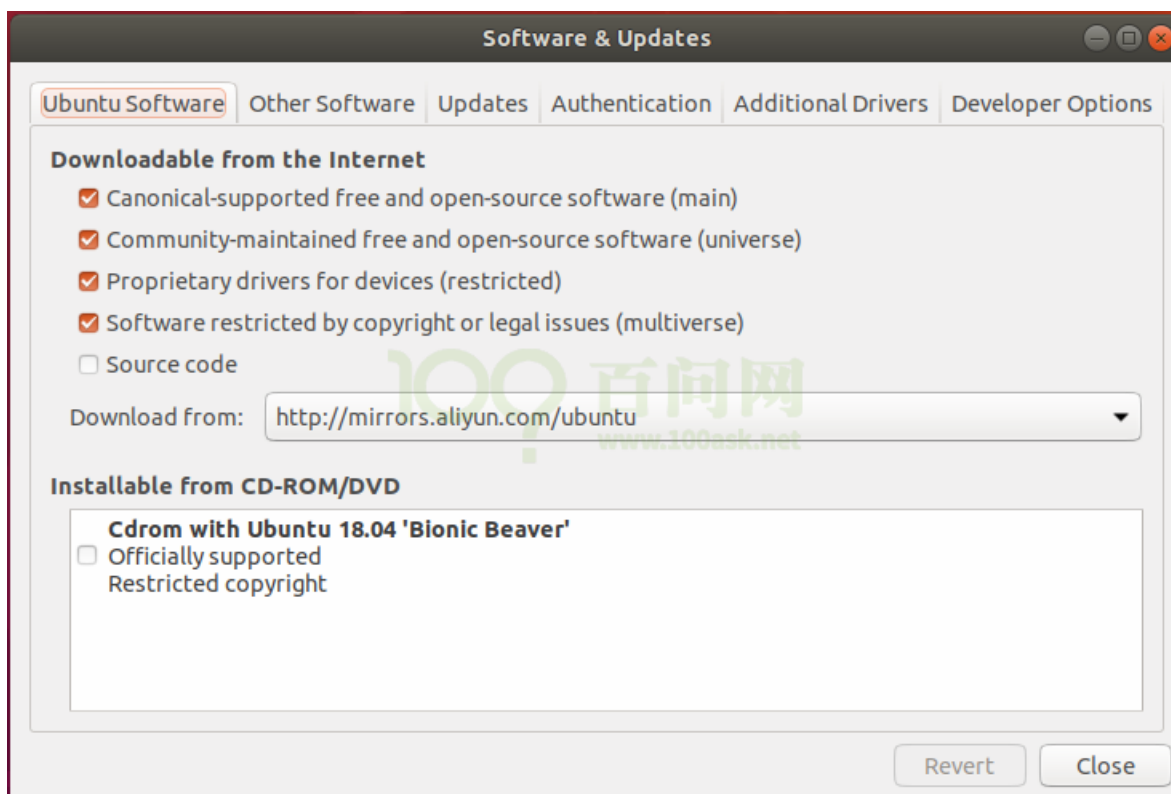
`APT` 下载工具和上面讲解到的 “`install`” 命令结合构成了 `Ubuntu` 下最常用的下载和安装软件方法。并且其解决了 `Linux` 平台下一安装软件的一个缺陷：即软件之间相互依赖问题。

3.4.3 如何更换软件源

`APT` 采用 `C/S` 模式，也就是客户端/服务器模式，一般来说我们的 `PC` 机作为客户端，当需要下载软件的时候就向服务器请求，因此我们需要知道服务器的地址，也叫做软件源或者更新源，这个一般默认使用的是国外的软件源（服务器）。因为我们是在中国，如果没有选择中国的服务器，可能会导致下载速度非常慢甚至是下载失败。所以我们需要修改软件源为国内的服务器，这里由两种方法可以进行修改：

- 1) 我们使用的是桌面版的 `Ubuntu`，先介绍在桌面环境中如何修改软件源：

打开系统设置，打开“软件和更新”设置，打开以后如下图所示：



上图中的“Ubuntu 软件”选项卡下面的“下载自”就是 APT 工具的安装源。

2) 下面介绍一种更加方便常用的修改方式，我们直接修改配置文件

Ubuntu 配置的默认源并不是国内的服务器，下载更新软件都比较慢。首先备份源列表文件 sources.list:

首先备份源列表

```
sudo cp /etc/apt/sources.list /etc/apt/sources.list_backup
```

打开 sources.list 文件修改选择合适的源，替换原文件的内容，保存编辑好的文件，以阿里云更新服务器为例：

打开 sources.list 文件

```
sudo vim /etc/apt/sources.list
```

编辑 /etc/apt/sources.list 文件，在文件最前面添加阿里云镜像源：

```
# 阿里源
deb http://mirrors.aliyun.com/Ubuntu/ bionic main restricted universe multiverse
deb http://mirrors.aliyun.com/Ubuntu/ bionic-security main restricted universe multiverse
deb http://mirrors.aliyun.com/Ubuntu/ bionic-updates main restricted universe multiverse
deb http://mirrors.aliyun.com/Ubuntu/ bionic-proposed main restricted universe multiverse
deb http://mirrors.aliyun.com/Ubuntu/ bionic-backports main restricted universe multiverse
deb-src http://mirrors.aliyun.com/Ubuntu/ bionic main restricted universe multiverse
deb-src http://mirrors.aliyun.com/Ubuntu/ bionic-security main restricted universe multiverse
deb-src http://mirrors.aliyun.com/Ubuntu/ bionic-updates main restricted universe multiverse
deb-src http://mirrors.aliyun.com/Ubuntu/ bionic-proposed main restricted universe multiverse
deb-src http://mirrors.aliyun.com/Ubuntu/ bionic-backports main restricted universe multiverse
```

在我们使用 APT 工具下载安装或者更新软件的时候，首先会在下载列表中与本机软件进行对比，看一下需要下载哪些软件，或者升级哪些软件，默认情况下 APT 会下载并安装最新的软件包，被安装的软件包所依赖的其它软件也会被下载安装或者更新，非常智能省心。

3.5 Ubuntu 下如何使用编辑器

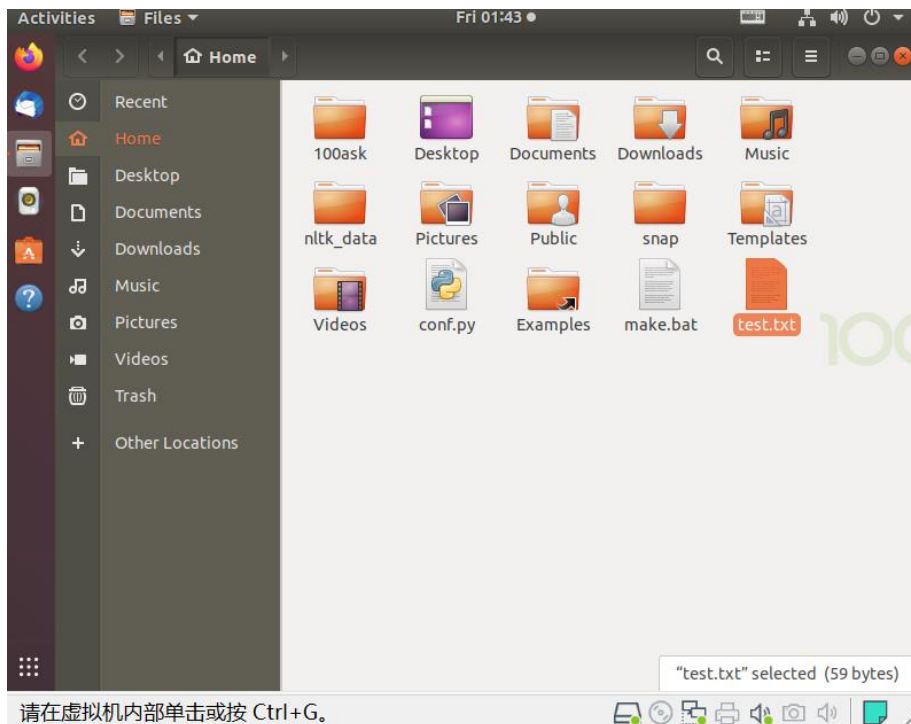
进行文本编辑是非常常用的操作,在 windows 系统为我们默认提供了记事本让我们可以进行文本编辑,Ubuntu 系统下也为我们提供一些文本编辑工具来让我们使用,如 gedit、vi/vim。

3.5.1 gedit 编辑器使用

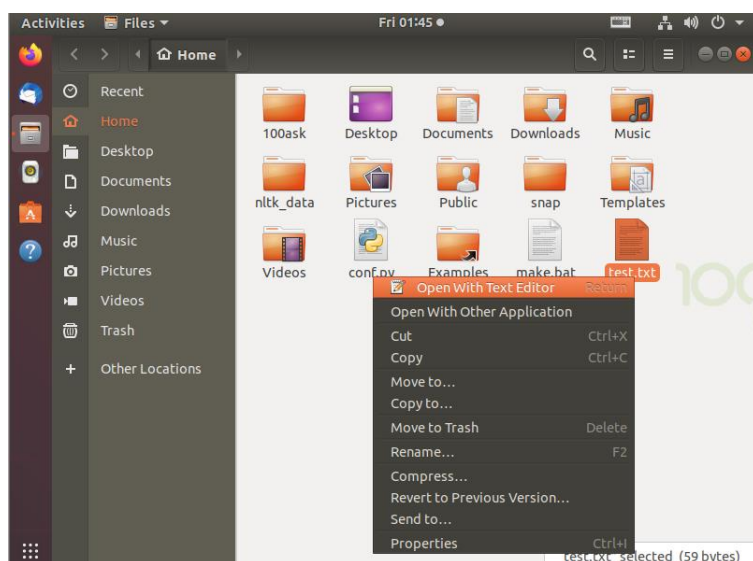
gedit 是一个窗口式的编辑器,只能在 Ubuntu 桌面环境下使用。

1. 在 GUI 环境下打开 gedit

我们一般在使用 Ubuntu 的文件管理器时,直接双击文本文件默认使用的就是“Gedit”编辑器。



或者单击右键需要编辑的文本文件,选择应用打开:



2. 在终端里打开 gedit

在终端里，可以直接运行 gedit 命令打开编辑器，也可以运行“gedit 文件名”打开指定文件，比如：

```
book@100ask: ~  
File Edit View Search Terminal Help  
book@100ask:~$  
book@100ask:~$ gedit hello.txt  
book@100ask:~$
```

如果要修改其他用户的文件，比如/etc/fstab，如下：

```
book@100ask: ~  
File Edit View Search Terminal Help  
book@100ask:~$  
book@100ask:~$ sudo gedit /etc/fstab  
[sudo] password for book:
```

← 可以使用sudo修改系统文件，
输入密码：123456

3. gedit 的使用

gedit 跟 Windows 下记事本的用法没什么差别。

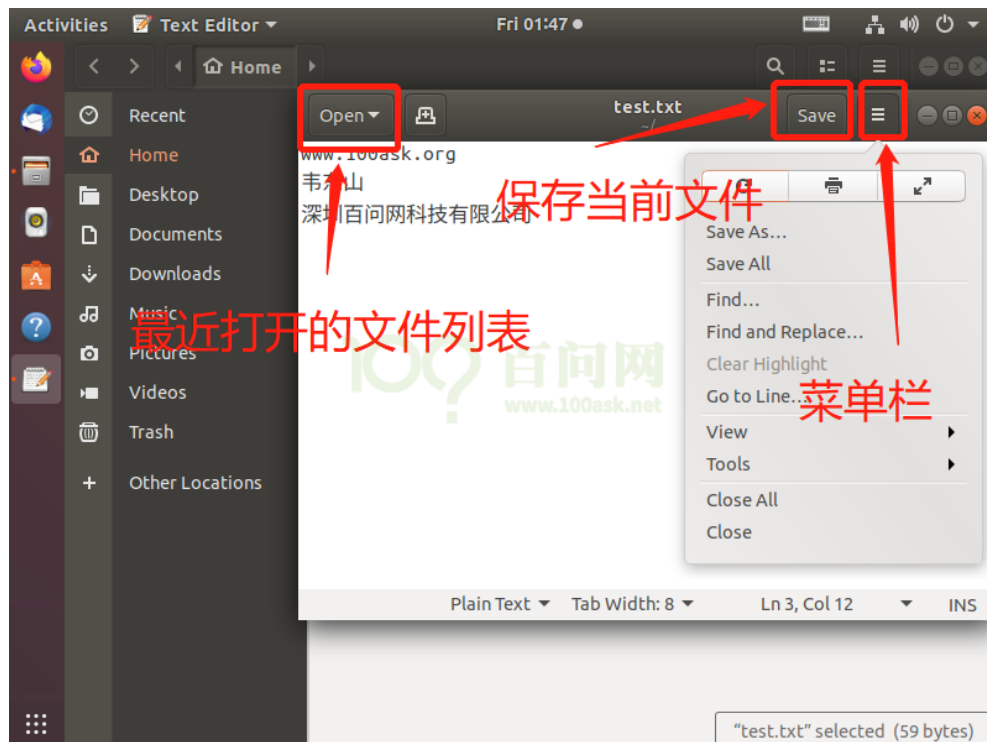
在编辑器中我们可以点击 “Open” 按钮浏览最近打开过的文件列表并打开文件；点击 “Save” 按钮可以保存当前正在编辑的文件；点击右侧的菜单栏进行更多的操作等等。

快捷键也跟 windows 下一样：

保存文件：Ctrl + s

另存为：Ctrl + Shift + s

搜索文本内容：Ctrl + f



3.5.2 vi 编辑器使用

请参考前面介绍的 vi 命令。

3.6 强烈建议使用纯英文的 Ubuntu

编译程序时，在纯英文环境下错误信息是以英文来描述的，比如：

```
book@book-virtual-machine:~/3_minutes/head_file$ gcc -o test main.c  
/tmp/ccZGM6a5.o: In function `main':  
main.c:(.text+0x28): undefined reference to `add' 英文环境下错误信息  
collect2: error: ld returned 1 exit status
```

这些错误信息在网上一贴，就有很多答案，百度找不到就用 bing，再不行就用 google，国内找不到可以找国外。

如果使用中文环境的话，错误信息是以中文来描述的，你只能在百度上搜答案。

3.7 默认不能使用 root 用户登录

我在开发过程中从没用过 root 用户，要使用 root 权限时可以在命令前加上“sudo”，比如“sudo ps -a”。

如果你就是喜欢用 root 用户，可以按下图操作，先给 root 用户设置密码，以后就可以用 root 用户登录了：

```
book@book-virtual-machine:~$ sudo passwd root 1. 给root设置密码  
[sudo] password for book: 2. 输入book用户密码：123456  
Enter new UNIX password:  
Retype new UNIX password: 3. 给root用户设置新密码，输入2次  
passwd: password updated successfully  
book@book-virtual-machine:~$ su root 4. 切换到root用户，  
Password: 以后可以用 su book 切换回来  
root@book-virtual-machine:/home/book#
```