

Linux & Linux Shell

I Linux 简介

[Linux系统简介 \(biancheng.net\)](#)

[Linux和UNIX的关系及区别 \(biancheng.net\)](#)

[Linux内核_百度百科 \(baidu.com\)](#)

[什么是 Linux 发行版？为什么它被称为“发行版”？ | Linux 中国 - 知乎 \(zhihu.com\)](#)

[Ubuntu_百度百科 \(baidu.com\)](#)

1 什么是 Linux

- Linux 也是一个操作系统软件，其 logo 是一只企鹅。
- Linux是一种开源电脑操作系统内核。它是一个用C语言写成，符合POSIX标准的类Unix操作系统。
- Linux 是 **Linux is not Unix** 的递归简写。



2 Linux 的优点

1. 免费且开源。不会受到任何商业化版本的制约。

Linux 是由世界各地成千上万的程序员设计和开发实现的，当初开发 Linux 系统的目的就是建立不受任何商业化软件版权制约的、全世界都能自由使用的类 Unix 操作系统兼容产品。

2. 与 Unix 系统兼容，具备 Unix 几乎所有的优秀特性。
3. 容易安装。

3 Unix 和 Linux 的关系

1. 关于 UNIX：

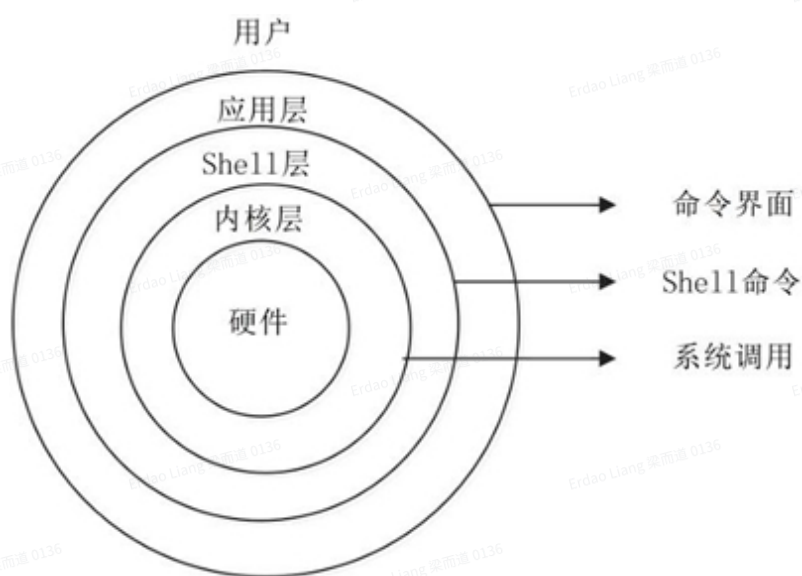
- UNIX 诞生于 20 世纪 60 年代末，是操作系统中的“老大哥”，后来的 Windows 和 Linux 都参考了 UNIX。
- UNIX 的部分技术来源可追溯到从 1965 年开始的 Multics 工程计划。以肯·汤普森为首的贝尔实验室研究人员吸取了 Multics 工程计划失败的经验教训，于 1969 年实现了一种分时操作系统的雏形，1970 年该系统正式取名为 UNIX。

- FreeBSD 是众多 UNIX 系统中的一种。

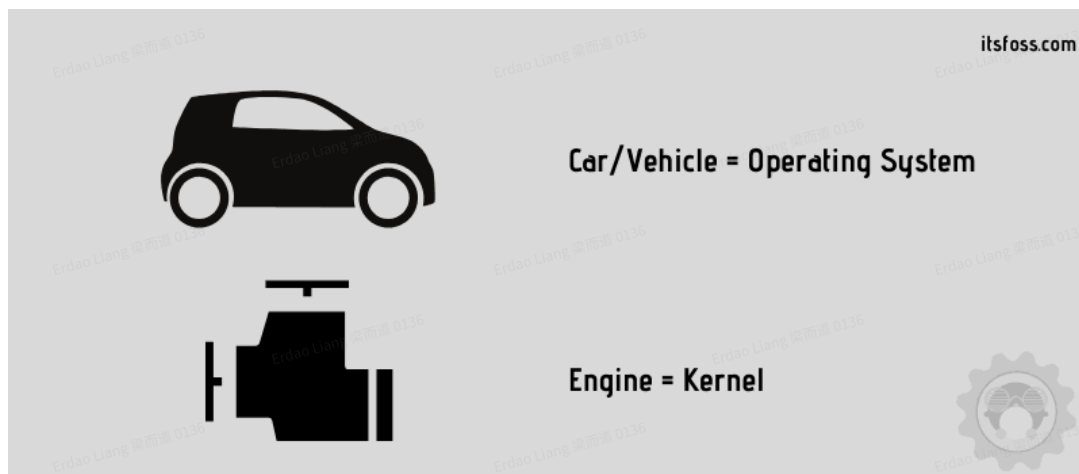
2. 关于 Linux:

- Linux 内核最初是由李纳斯·托瓦兹（**Linus Torvalds**）在赫尔辛基大学读书时出于个人爱好而编写的，他觉得教学用的迷你版 UNIX 操作系统 Minix 太难用了，于是决定自己开发一个操作系统。
- 因此，Linux 是一个**类 Unix** 的操作系统，Unix 要早于 Linux，Linux 的初衷就是要替代 UNIX，并在功能和用户体验上进行优化，所以 Linux 模仿了 UNIX（但并没有抄袭 UNIX 的源码），使得 Linux 在外观和交互上与 UNIX 非常类似。

4 UNIX/Linux 系统结构



1. **内核层 (Kernal)**：内核层是 UNIX/Linux 系统的核心和基础，它直接附着在硬件平台之上，控制和管理系统内各种资源（硬件资源和软件资源），有效地组织进程的运行。
2. **Shell 层**：Shell 层是与用户直接交互的界面。用户可以在提示符下输入命令行，由 Shell 解释执行并输出相应结果或者有关信息，所以我们也把 Shell 称作**命令解释器**（见第二部分）。
3. **应用层**



5 Linux 发行版

- Linux 发行版是一个由 Linux 内核开发的操作系统，它也可能包括显示服务器和桌面环境，以用作常规的桌面操作系统。
- 这个术语之所以是“Linux 发行版”，是因为像 Debian、**Ubuntu** 这样的机构“发行”了 Linux 内核以及所有必要的软件及实用程序，使其可以作为一个操作系统使用。

Ubuntu Desktop是由Canonical开发的Linux 发行版，由于其易用性，它是最受欢迎的发行版之一。

Windows 10 Linux (ubuntu) 根目录位置：**C:\Users\用户名\AppData\Local\Packages\CanonicalGroupLimited.UbuntuonWindows_79rhkp1fndgsc\LocalState\rootfs**

[Windows 10 的Linux \(ubuntu\) 根目录位置-CSDN博客](#)

- “Linux” 是内核，而“Linux 发行版”是操作系统。这就是为什么它们有时也被称为基于 Linux 的操作系统的的原因。

II Shell 简介

[Linux命令解释器：shell和bash](#)

[关于 Shell 解释器 介绍](#)

[什么是shell脚本？](#)

[Shell是什么？\(biancheng.net\)](#)

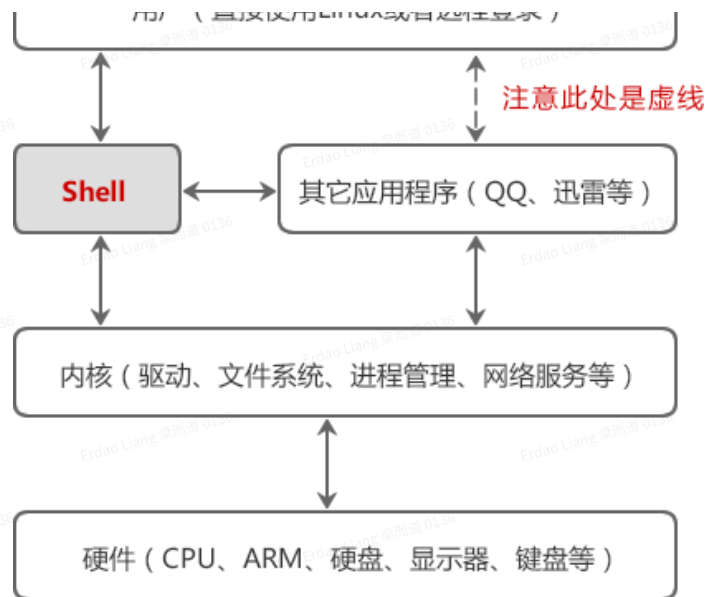
[UNIX 什么是 Shell 脚本_w3cschool](#)

[shell和shell脚本的区别 - 知乎 \(zhihu.com\)](#)

[linux bash 是什么意思 - PHP中文网](#)

1 Shell 是什么

- 命令解释器：解释器是一种电脑程序，能够把高级编程语言一行一行直接转译运行。
- Shell 是实现 Linux 内核与用户之间沟通的**命令解释器**。它连接了用户和 Linux 内核，让用户能够更加高效、安全、低成本地使用 Linux 内核。
- Shell 本身并不是内核的一部分，它只是站在内核的基础上编写的一个应用程序。用户通过 Shell 来使用 Linux，不启动 Shell 的话，用户就没办法使用 Linux。
- Shell 为用户提供了一个操作界面，接收用户的输入，遵循一定的语法将输入的**命令**加以解释并传给系统，然后基于该输入执行程序。程序执行完后，结果会显示在显示器上。
- Shell 也连接其他程序：在 Shell 中输入的命令，有一部分是 Shell 本身自带的，这叫做**内置命令**；有一部分是其它的应用程序（一个程序就是一个命令），这叫做**外部命令**。Shell 可以调用其他的程序，每个程序就是一个命令。



2 Shell 的种类

Linux 提供了多种命令解释器，一般默认 bash 作为默认的解释器。

- bourne again shell (/bin/**bash**) —— 最常用
- bourne shell (/usr/bin/sh 或 /bin/sh)
- C shell (/usr/bin/csh)
- K shell (/usr/bin/ksh)

3 Shell 编程与 Shell 脚本

- Shell 既是一种命令语言，又是一种**程序设计语言**（也就是 **shell 脚本**/shell 编程）。作为命令语言，它交互式地解释和执行用户输入的命令；作为程序设计语言，它定义了各种变量和参数，并提供了许多在高阶语言中才具有的控制结构，包括循环和分支。
- Shell 脚本 (Shell Scripting) 是由 Shell 命令组成的**文件**，是利用 shell 的功能所写的一个**程序**。
- Shell 脚本是解释型语言，所以他们并不会被编译。作为编程语言时，解释器是 Shell 这个程序。

III 环境变量

什么是环境变量？ - 知乎 (zhihu.com)

什么是环境变量，Linux环境变量及作用 (biancheng.net)

1 环境变量是什么

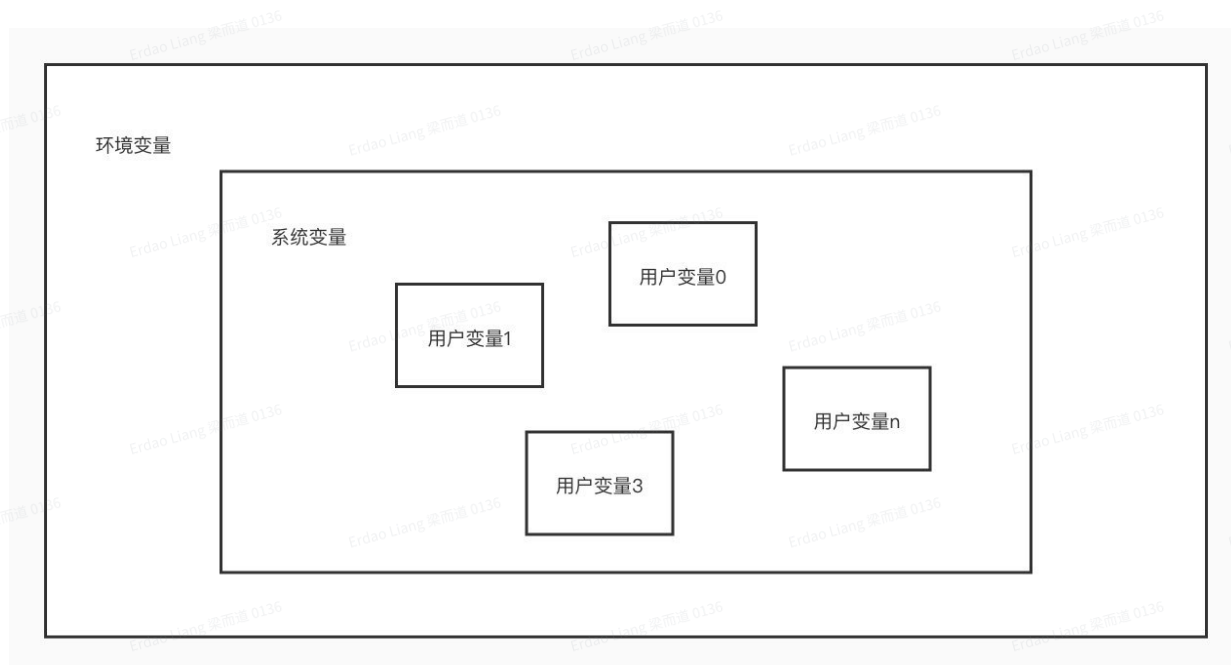
- 环境变量是**描述环境的变量**，是指在操作系统中用来指定操作系统运行环境的变量。它包含了一个或者多个应用程序所将使用到的信息。

- 例如 Windows 和 DOS 操作系统中的 PATH 环境变量，当要求系统运行一个程序而没有告诉它程序所在的完整路径时，系统除了在当前目录下面寻找此程序外，还应到 PATH 中指定的路径去找。用户通过设置环境变量，来更好的运行进程。

Linux 系统中环境变量的名称一般都是大写的，这是一种约定俗成的规范。

2 系统变量与用户变量

系统变量就是系统级别的变量，用户需要使用系统变量。如果系统变量被修改了，而任何系统用户都在用系统变量，因此每个系统用户都将受到影响。用户变量运行在系统变量之上的，每个用户拥有不同的用户变量，不同用户的用户变量之间是并列的，也是互不干扰的。



IV Shell 命令

1 Shell 常用命令

[Linux 命令大全 | 菜鸟教程 \(runoob.com\)](#)

[UNIX 常用命令_w3cschool](#)

[unix 的基本操作命令_li_snow的博客-CSDN博客_unix操作](#)

用 `man [命令]` 查看各个命令的使用文档。

2 文件基本属性

[Linux 文件基本属性 | 菜鸟教程 \(runoob.com\)](#)

[linux查看文件夹权限_码农先锋A的博客-CSDN博客_linux查看文件夹权限](#)

[解决 Linux Bash 中的权限被拒绝错误 | D栈 - Delft Stack](#)

[Linux下出现permission denied的解决办法_StudentWang_的博客-CSDN博客](#)

Linux下设置文件夹以及子文件夹中所有文件权限_wangqingchuan92的博客-CSDN博客_linux 子文件夹权限

Linux 系统是一种典型的多用户系统，不同的用户处于不同的地位，拥有不同的权限。为了保护系统的安全性，Linux 系统对不同的用户访问同一文件（包括目录文件）的权限做了不同的规定。

属性信息表示

文件类型	属主权限			属组权限			其他用户权限		
0	1	2	3	4	5	6	7	8	9
d	rwX			r-X			r-X		
目录文件	读	写	执行	读	写	执行	读	写	执行

每个文件的属性由 10 个字符来确定。

1. 最前面的字母代表文件类型：d 目录，- 文件。
2. 后面的九个字母代表权限：
 - Linux/Unix 的文件调用权限分为三级：文件所有者（Owner）、用户组（Group）、其它用户（Other Users）。
 - 均为 **rwX** 的三个参数的组合。其中，**r** 代表可读(read)、**w** 代表可写(write)、**x** 代表可执行(execute)，没有相应权限时为 -。

修改用户权限

Chmod（Change mode）命令：修改用户的权限。

1. 语法： `chmod [-R] xyz 文件或目录`
2. 参数xyz：使用数字的累加来代表各个权限，各权限的分数对照表如下：r:4，w:2，x:1。如当权限为： **-rwxrwx---** 分数则是：
 - owner = rwx = 4+2+1 = 7
 - group = rwx = 4+2+1 = 7
 - others= --- = 0+0+0 = 0，权限数字为770。
3. 选项：
 - **-R**：进行递归(recursive)的持续变更，以及连同次目录下的所有文件都会变更。

3 文件与目录管理

[Linux 文件与目录管理 | 菜鸟教程 \(runoob.com\)](#)

[Linux ls命令 | 菜鸟教程 \(runoob.com\)](#)

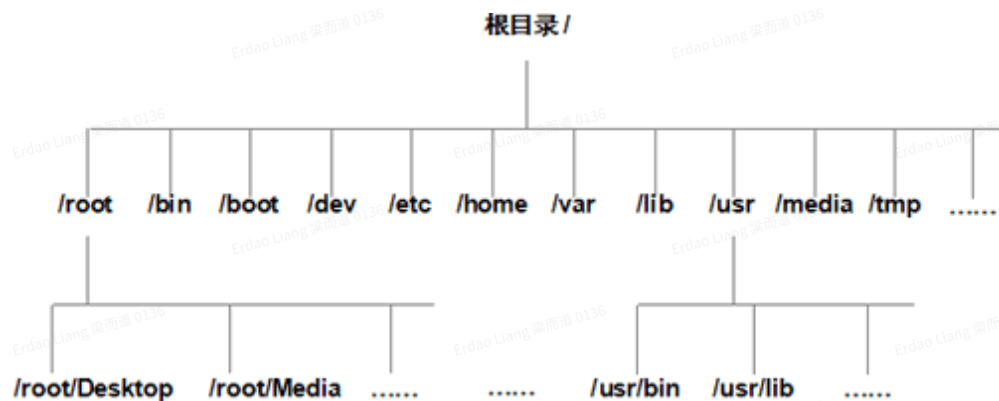
[Linux cd命令：切换目录 \(biancheng.net\)](#)

Linux rm命令：删除文件或目录 (biancheng.net)

linux中 ~ / ./分别表示什么目录_小小鸟008的博客-CSDN博客_./目录

Linux 文件系统层次结构

为了方便管理文件和目录，Linux 系统将它们组织成一个以根目录 / 开始的倒置的**树状结构**。Linux 中的目录，和 Windows 系统中的文件夹类似，不同之处在于，Linux 系统中的目录也被当做文件看待。



重要目录简写

- / 是指根目录：就是所有目录最顶层的目录。
- ~ 表示代码主目录，也就是当前登录用户的用户目录。
- ./ 表示当前目录。
- ../ 表示上级目录。

绝对路径和相对路径

1. 绝对路径的写法：由根目录 / 写起。 /usr/share/doc。
2. 相对路径的写法：../ 写起。 ../man。

常用命令

1. ls (List files)：列出目录和文件名
 - -a 显示所有文件及目录 (. 开头的隐藏文件也会列出)
 - -l 除文件名称外，亦将文件型态、**权限**、拥有者、文件大小等资讯详细列出
 - -R 若目录下有文件，则以下之文件亦皆依序列出
 - -d 仅列出目录本身，而不是列出目录内的文件数据(常用)
2. cd (Change directory)：切换目录 cd 相对路径或绝对路径
3. pwd (Print work directory)：显示目前目录
4. mkdir (Make directory)：创建新的目录 mkdir 目录名称
5. rmdir (Remove directory)：删除一个空的目录 rmdir 目录名称
6. cp (Copy file)：复制文件或目录 cp [options] source1 source2 directory

7. `rm` (Remove) : 删除文件或目录

- `-r` 递归删除, 涉及删除目录时必须使用 `-r`
- `-f` 强制删除, 无需询问

8. `mv` (Move file) : 移动文件与目录, 或修改文件与目录的名称 `mv [options] source1 source2 directory`

4 压缩和解压

[Linux tar打包命令详解 \(biancheng.net\)](http://biancheng.net)

[linux中的tar命令的使用 - 博客园 \(cnblogs.com\)](http://cnblogs.com)

[Linux zip命令: 压缩文件或目录 \(biancheng.net\)](http://biancheng.net)

[Linux - 压缩指定后缀文件 \(压缩FIND命令找到的文件\) - 博客园 \(cnblogs.com\)](http://cnblogs.com)

zip

1. 命令基本格式: `zip [选项] 压缩包名 源文件或源目录列表`

2. 选项

- `-r` 递归压缩目录, 及将制定目录下的所有文件以及子目录全部压缩。
- `-m` 将文件压缩之后, 删除原始文件, 相当于把文件移到压缩文件中。
- `-v` 显示详细的压缩过程信息。

tar

使用 `tar` 命令归档的包通常称为 `tar` 包 (`tar` 包文件都是以 “.tar” 结尾的)。

1. 命令基本格式:

Shell

```
1 [root@localhost ~]$ tar [选项] 源文件或目录
2
3 [root@localhost ~]$ tar -cvf anaconda-ks.cfg.tar anaconda-ks.cfg
4 $ #把anacondehks.cfg打包为 anacondehks.cfg.tar文件
```

2. 选项

- `-c` 将多个文件或目录进行打包。
- `-A` 追加 `tar` 文件到归档文件。
- `-f` 包名 指定包的文件名。包的扩展名是用来给管理员识别格式的, 所以一定要正确指定扩展名。
- `-v` 显示打包文件过程。

`-f` 参数是最后一个参数, 后面只能接档案名。 **`-f`为必选。**

例: 利用 `find` 命令, 压缩找到的所有指定后缀的文件:

Shell

```
1 $find . -name "*.html" -type f -mtime +1 | xargs tar -czvf html.tar.gz
```

5 文本编辑器 nano

试试 GNU nano，一个轻量级的 Vim 替代品 | [Linux 中国 - 知乎 \(zhihu.com\)](#)

详解linux中nano命令-CSDN博客

许多 Linux 发行版都捆绑了 Vim 作为默认的文本编辑器。GNU nano 是 Vim 之外的另一种轻量级终端文本编辑器，比 vim 要简单得多，比较适合 Linux 初学者使用。

1. 语法：

Shell

```
1 $nano [选项] [[+行,列] 文件名]...
2 $nano example.txt
```

2. 快捷键：

- **Ctrl+S** 保存你的工作
- **Ctrl+W** 另存为
- **Ctrl+R** 加载文件（读取）
- **Ctrl+X** 退出
- **Ctrl+G** 获得帮助

3. 光标控制

- 移动光标：使用方向键移动。
- 选择文字：按住鼠标左键拖到。

6 文件处理

cat (concatenate)

Linux cat命令：连接文件并打印输出到标准输出设备 ([biancheng.net](#))

1. 功能1：显示文件的内容

a. 语法：**cat** [选项] 文件名

b. 选项：

- **-n** 对输出的所有行进行编号
- **-b** 同 **-n** 不同，此选项表示只对非空行进行编号。
- **-s** 当遇到有连续 2 行以上的空白行时，就替换为 1 行的空白行。

2. 功能2：合并多个文件内容并输出到另一个文件

a. 语法: `cat 文件1 文件2 ... > 文件3`

cut 输出特定文字

输出文件中特定位置的文本到标准输出。

1. 选项: cut 命令必须要指定选项才能执行。

- -c: 提取指定的字符。
- -d: 自定义分隔符, 默认为 Tab 制表符。
- -f: 与-d一起使用, 提取指定的字段。

2. 规则: [Linux cut命令 | 菜鸟教程\(runoob.com\)](#)

sed 处理文本数据

[Linux sed 命令 | 菜鸟教程\(runoob.com\)](#)

1. 功能: 根据脚本命令来处理文本文件中的数据。

2. 规则: [Linux sed命令完全攻略\(超级详细\)\(biancheng.net\)](#)

wc 统计

[linux shell wc命令 - 李泽 - 博客园\(cnblogs.com\)](#)

[Shell命令统计文件行数方法_Hoking的博客-CSDN博客](#)

统计文件中的字节数、字数和行数。wc同时也给出所有指定文件的总统计数。字是由空格字符区分开的最大字符串。如果不指定文件, 则统计标准输入。

1. 语法: `wc [选项] 文件1 文件2 ...`

2. 选项: -c 统计字节数。-l 统计行数。-w 统计字数。

选项可以组合使用

Shell

```
1 $ wc -lcw file1 file2
2 4 33 file1
3 7 52 file2
4 11 11 85 total
```

7 其他常见命令

Read

Alias

Xargs

Echo

V Shell 编程



SilverFOCS Incubator

Training Materials (Set 1)

Donnell & Ines — SFOCS (Summer 2022)

S-FOCS (tech)

- Pull the practice support materials
- Always include comments in the code
- Before starting think of the program structure
- Keep testing and improving your code
- Write a single README file per practice
- For any altered code list changes in a changelog
- Push you code on your repository and create a release

This goal of this training set is to help you getting acquainted with shell scripting.

Modern computers feature a wide variety of interfaces to interact with them, ranging for fancy graphical user interfaces to voice interfaces, or even AR/VR. Unfortunately those most common interfaces are often fundamentally restricted in what they allow you to do. For instance important buttons could be missing or some voice command might not have been programmed.

Another approach to communicating with computers consists in using a *command line* or *text interface* called a *Shell*. Many type of shells exist, each displaying different characteristics and being more or less advanced. A basic shell allows the user to at least launch programs, process input, and inspect their output in a semi-structured way, but many also embed an internal interpreter allowing to develop shell scripts. In this training set, we will focus on the Bourne Again Shell, often referred to as "bash".

📄 t1.pdf

1 管道

输入输出重定向