



Linux 系统使用介绍

潘建瑜

MATH@ECNU



華東師範大學 | 数学科学学院
School of Mathematical Sciences, East China Normal University

1 Linux 操作系统介绍

2 Linux 基本操作与管理

3 Linux 常用命令

4 Linux 程序开发环境

- ❑ 鸟哥的Linux私房菜: 基础学习篇, 第4版, 2018
- ❑ Linux命令行与shell脚本编程大全, 第4版, 2022
- ❑ Linux教程 http://c.biancheng.net/linux_tutorial/



1

Linux 操作系统介绍

- 1 Linux 操作系统介绍
- 2 Linux 基本操作与管理
- 3 Linux 常用命令
- 4 Linux 程序开发环境

操作系统

计算机 = 硬件 + 软件

计算机软件

□ 系统软件

管理计算机资源/硬件，并为应用软件提供一个统一的平台。

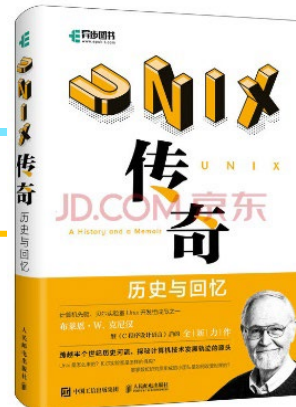
□ 应用软件

在系统软件的基础上实现用户所需要的各种功能。

操作系统 (Operating System, OS) 是最基本的系统软件，负责对电脑硬件直接控制及管理，包括处理器、存储、设备和作业等。

UNIX 操作系统

UNIX



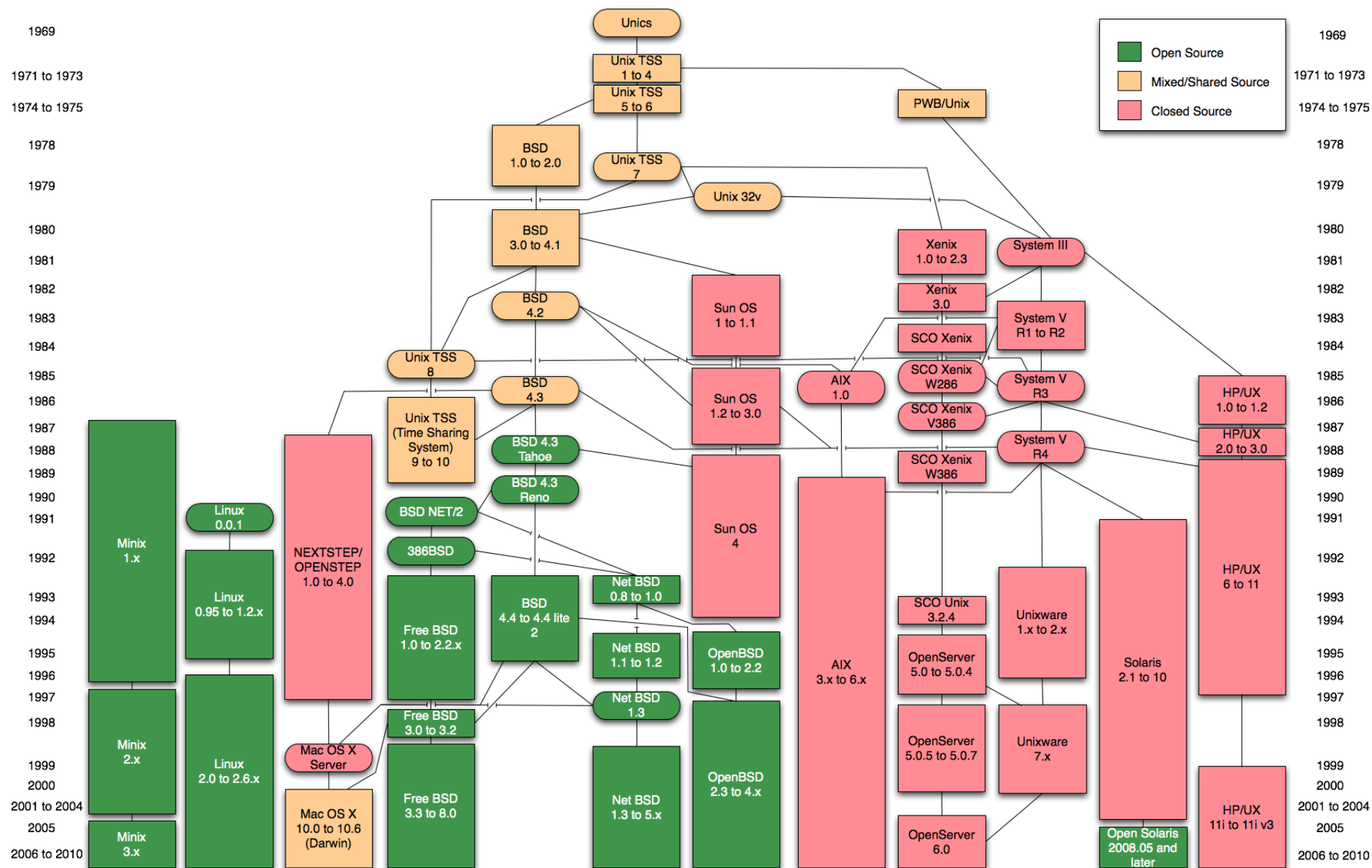
- ▶ 1969 年诞生于 AT&T Bell 实验室，主要用于服务器领域，大部分重要网络环节都是 Unix 构造，1973 年用 C 语言改写
- ▶ 1978 年 BSD 推出，成为 UNIX 另一主流分支
- ▶ 版本众多：AIX (IBM)、HP-UX (HP)、Solaris(SUN)、IRIX(SGI)、...
- ▶ UNIX 不是免费的，且不适用个人PC电脑



Unix 之父
Thompson & Ritchie



UNIX 操作系统发展史



Linux 操作系统

Linux

- ▶ 1991 年发布，免费使用和自由传播的类 Unix 操作系统
- ▶ 支持多种硬件平台，包括个人电脑和超级并行计算机
- ▶ 有良好的用户界面，同时提供字符界面和图形界面
- ▶ 版本众多：RHEL、CentOS、Fedora、Ubuntu、...

- ❑ 1987 年，荷兰计算机科学家 A. Tanenbaum 专门写的简化的类 UNIX 系统 (mini-UNIX, Minix)，给 UNIX 爱好者学习使用。
- ❑ 1990 年，芬兰赫尔辛基大学的一名学生 Linus Torvalds 产生了写一个比 Minix 更好的 Minix 的想法，于是编写了一些硬件的设备驱动程序和一个小的文件系统，这就是Linux 0.0.1 版本，但它只有操作系统内核，必须在有 Minix 的机器上编译后才能用
- ❑ 1991年发布了 Linux 0.0.2 版本，可以独立运行 bash 和 gcc。

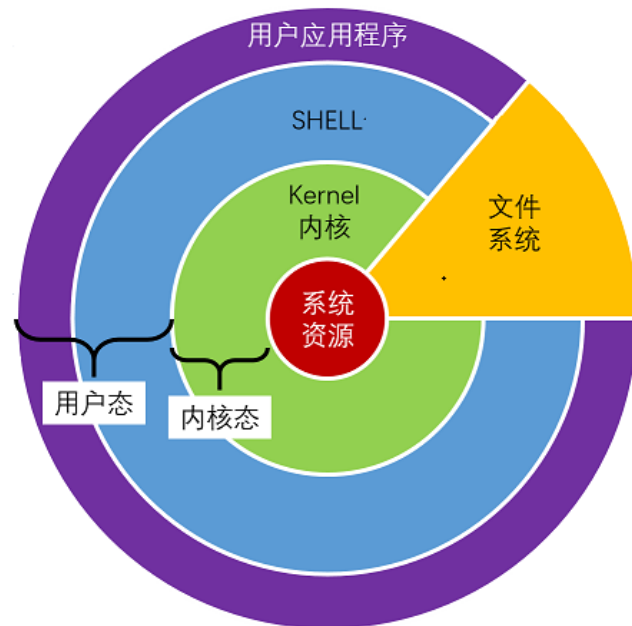


Linux 系统结构

Linux 系统结构大致分为：

内核、Shell、文件系统、应用程序

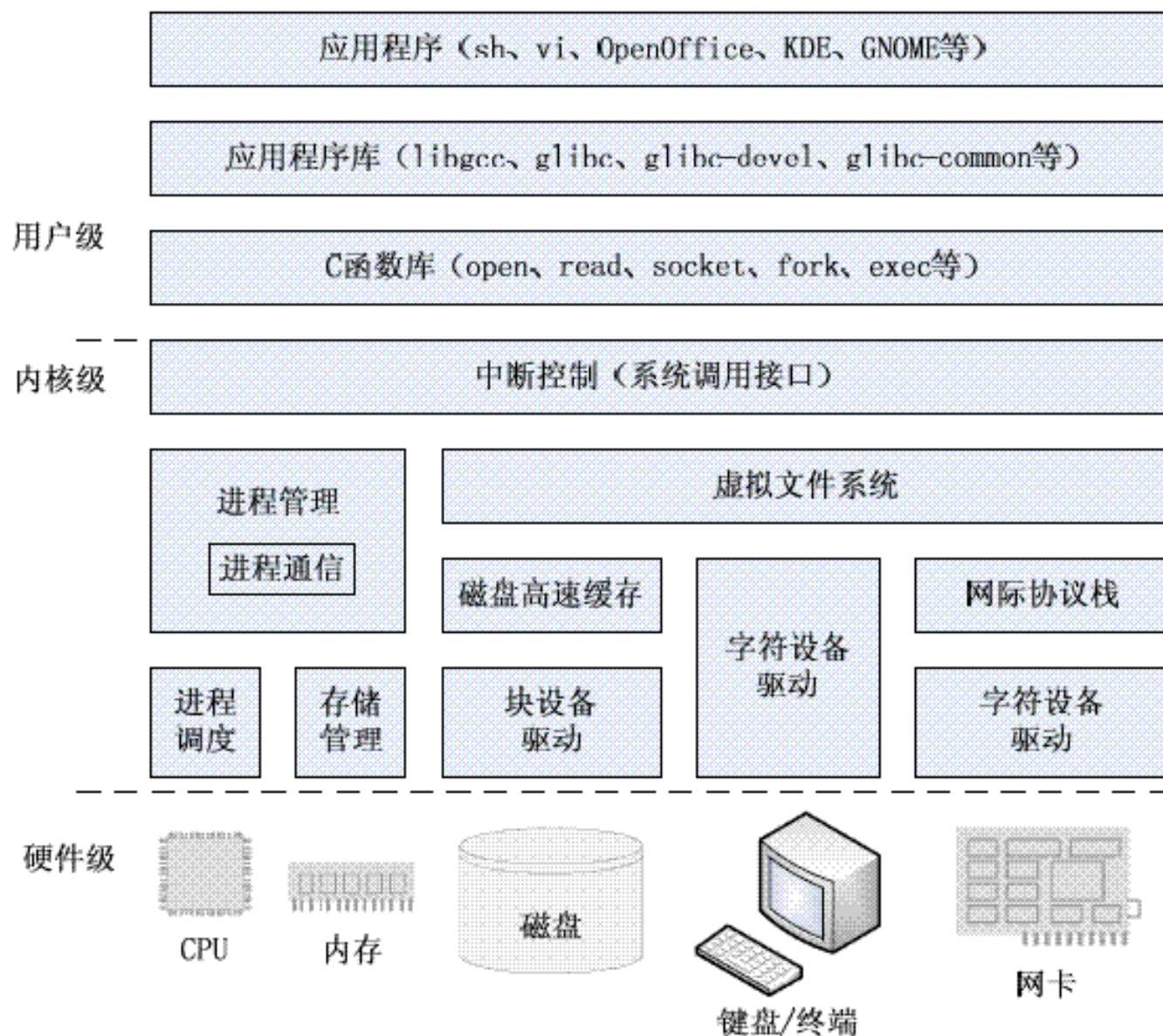
- ▶ 内核：
运行程序和管理各种硬件设备的核心程序
- ▶ Shell：
命令语言解释器，是用户与内核进行交互操作的接口
- ▶ 文件系统：
文件存放在存储设备上的组织方法，如 ext4、vfs、nfs 等
- ▶ 应用软件：
Linux 发行版都有一整套的应用程序集



Linux



用户



常见 Linux 发行版



- ▶ Red Hat 创建于1993年，是目前世界上最资深的 Linux 和开放源代码提供商
- ▶ 2018 年被 IBM 以 340 亿美元的价格收购
- ▶ 相关产品：RHEL, CentOS, Fedora



- ▶ 首次发布：2004 年 10 月，版本为 4.10
- ▶ 每六个月更新一次，当前版本为 22.10
- ▶ 注重桌面应用，对个人用户比较友好

常见 Linux 发行版



<http://www.debian.org>



<https://mxlinux.org/>



<https://manjaro.org>



<https://www.kali.org/>



<https://linuxmint.com/>



<https://pop.system76.com/>





2

Linux 基本操作与管理

- 1 Linux 操作系统介绍
- 2 Linux 基本操作与管理
- 3 Linux 常用命令
- 4 Linux 程序开发环境

- Linux 安装
- Linux 基础
- 文件系统
- 进程管理

Linux 安装

Linux 发行版选择



CentOS



也可以选择其他发行版，安装指南可以参见网络。

Linux 基础

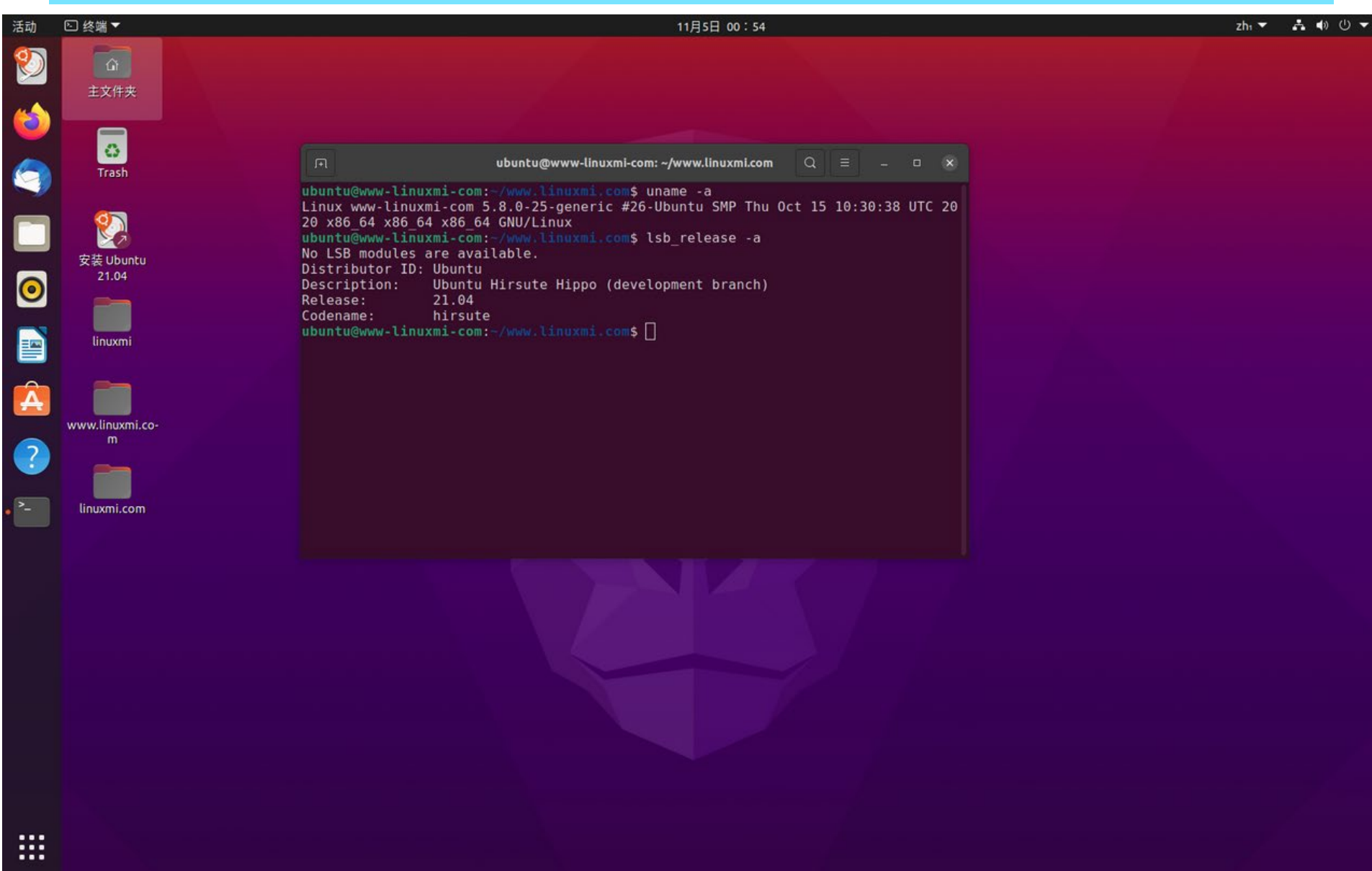
几个术语

- ▶ **命令 (Command) :**
给计算机的指令，通常使用键盘输入，Linux 区分大小写
- ▶ **命令行 (Command line) :**
在终端里输入命令的地方
- ▶ **Shell:**
一种命令解释器，提供用户和操作系统之间的交互接口；
当用户输入命令后，shell 进行解释，然后送往操作系统（内核）执行；
shell 可以执行 Linux 系统内部命令，也可以执行应用程序；
也可以利用 shell 编程，执行复杂的命令程序
- ▶ **超级用户/根用户/管理员 (root) :**
具有系统最高权限，对整个系统有完全的访问权；
只有超级用户才能更改系统的设置

Ubuntu 桌面

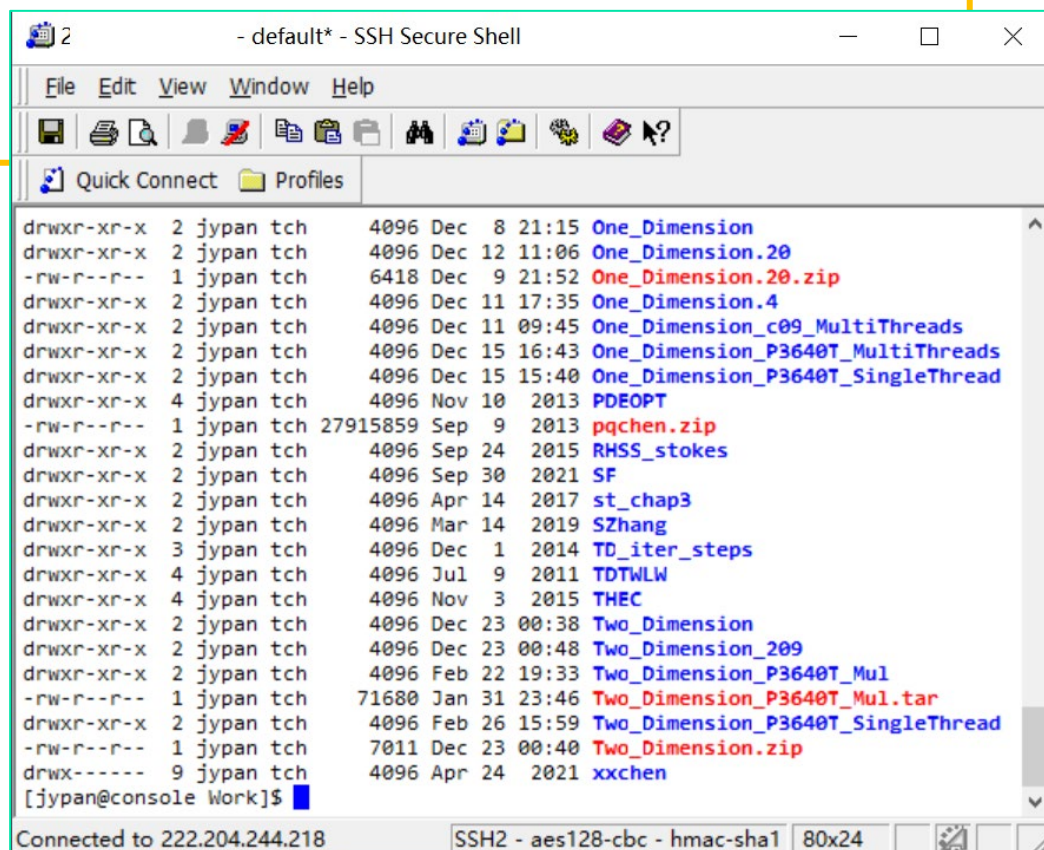


Terminal



Linux 字符操作环境

- ▶ **终端** (Terminal)：人机交互的窗口，纯字符界面
- ▶ 用户可以通过终端访问系统资源，可远程登录
- ▶ 远程登录：SSH
- ▶ 退出终端：exit



The screenshot shows an SSH Secure Shell terminal window titled "- default* - SSH Secure Shell". The window has a menu bar (File, Edit, View, Window, Help) and a toolbar. Below the toolbar is a "Quick Connect" section with a "Profiles" button. The main area displays a directory listing of files and folders, including:

- One_Dimension
- One_Dimension.20
- One_Dimension.20.zip
- One_Dimension.4
- One_Dimension_c09_MultiThreads
- One_Dimension_P3640T_MultiThreads
- One_Dimension_P3640T_SingleThread
- PDEOPT
- pqchen.zip
- RHSS_stokes
- SF
- st_chap3
- SZhang
- TD_iter_steps
- TDTWLW
- THEC
- Two_Dimension
- Two_Dimension.209
- Two_Dimension_P3640T_Mul
- Two_Dimension_P3640T_Mul.tar
- Two_Dimension_P3640T_SingleThread
- Two_Dimension.zip
- xxchen

The prompt is [jypan@console Work]\$.

Linux 用户管理

用户管理

□ 账号/用户 (user)

用户须有合法账号才能登录和使用服务器，用户管理是超级用户的日常工作

- ▶ 每个用户名对应一个**用户 ID 号** (uid, 输入命令 `id` 可显示相关信息)
- ▶ 每个用户都被分配到一个指定的**组**中，每个组有独立的 ID 号 (gid)
- ▶ 用户登录后，会进入系统指定的专属目录，即用户的**主目录** (home) ，主目录的名字通常与用户帐号相同
- ▶ 用户对自己主目录下的文件拥有**所有权**，可以进行相关操作
- 在创建新用户时，管理员会给每个用户建立一个主目录

提醒：当使用超级用户登录后，进行相关操作时需非常谨慎，否则可能会对系统或用户文件造成不可挽回的损失。

Linux 命令一般格式

command [选项] [文件或目录列表]

- ▶ 其中 **选项** 通常以 “-” 开始, 如 `ls -l`
- ▶ 多个选项可用一个 “-”, 如 `ls -l -a` 与 `ls -la` 相同
- ▶ 所有的命令从**标准输入**接受输入, 输出结果显示在**标准输出**, 而错误信息则显示在**标准错误**输出设备
- ▶ 可使用重定向功能对这些设备进行**重定向**
- ▶ 命令在正常执行结果后返回一个 0 值, 否则返回一个非零值 (在shell中可用变量 `$?` 查看: `echo $?`)

更改口令

passwd

- ▶ 直接在命令行中输入 `passwd`，然后根据提示，先输入原来的口令，回车后，再输入新口令，并确认一次
- ▶ 提醒：输入密码时，屏幕上不会有任何输出或提示！
- ▶ 为确保高安全性，Linux 对口令有一定的要求，而且通常需要在口令中加入一些特殊符号，如加号、减号等

yppasswd

- ▶ 修改 NIS 口令，网络用户需要使用该命令来修改密码
(通常服务器上需使用 `yppasswd` 来修改口令)

获取帮助

man # 获取相关命令的帮助信息

info # 获取相关命令的更多信息

▶ 退出 man 或 info 按 q

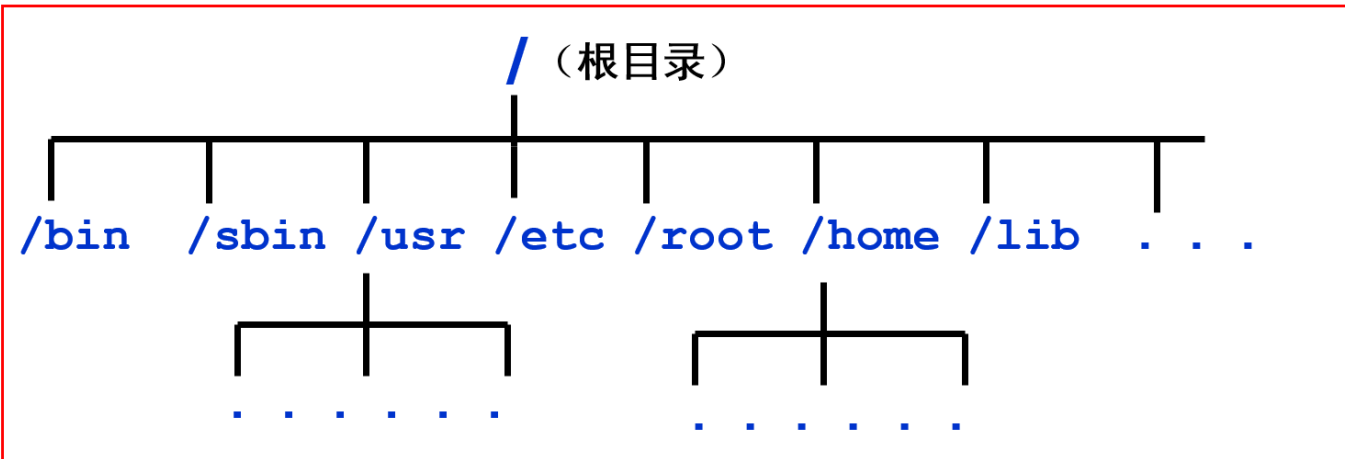
```
man ls
```

```
info ls
```

文件系统结构

目录树结构

- ▶ 从一个根目录 “/” 开始，根目录下可以有任意多个文件和子目录，子目录中又可以有任意多个文件和子目录，形成一个树状结构
- ▶ Linux 文件结构是以根目录为起点，向下展开
- ▶ Linux 的这种文件系统结构使得一个目录和它包含的文件/子目录之间形成一个父子关系



当前工作目录

当前工作目录

- ▶ 用户目前所处的目录
- ▶ 用户登录后进入的目录通常是自己的主目录
- ▶ 可用 `pwd` 命令查看用户的当前目录
- ▶ 可用 `cd` 命令来切换目录
- ▶ 一些特殊字符的特殊含义：
 - ✓ “.” 表示当前目录
 - ✓ “..” 表示当前目录的上一级目录（父目录）
 - ✓ “-” 表示用 `cd` 命令切换目录前所在的目录
 - ✓ “~” 表示用户主目录的绝对路径名

路径 (path)

路径

- ▶ 文件或目录在文件系统中所处的位置
- ▶ 路径分为“绝对路径”和“相对路径”
 - ▶ 绝对路径：从根目录开始的完整的路径
 - ▶ 相对路径：相当于当前工作目录的路径
- ▶ 任何不以 `/` 和 `~` 开始的路径均为相对路径

例如：设用户当前工作目录为：`/home/jypan/Teaching/`
则该目录的子目录 `ParaComp/` 中的文件 `lect01.pptx` 的绝对路径为
`/home/jypan/Teaching/ParaComp/`
相对路径为
`./ParaComp/`

文件访问权限

文件类型

- ▶ 普通文件、目录、符号链接、设备文件等

访问权限

- ▶ 每个文件或目录都有访问权限，这些权限决定了谁能访问和如何访问这些文件或目录
- ▶ 通过设定权限，可以实现以下三种方式来限制访问权限：
 - ✓ 只允许**用户自己**访问
 - ✓ 允许**与自己同组**的用户访问
 - ✓ 允许系统内的**所有用户**访问

文件访问权限

访问权限（续）

- ▶ 一个文件或目录可以有读、写和执行的权限

当一个文件或目录被创建时，系统会自动赋予文件所有者的读和写的权限，这样就允许文件所有者能够阅读和修改该文件。

- ▶ 一个文件或目录可以被三种不同类型的用户访问：

所有者、同组用户和其它用户

通常每个用户在创建时都被分配到一个特定的用户组，系统中存在许多不同的用户组。文件所有者可以通过修改文件的属性而授予其所在用户组的其他成员访问该文件的权限，同样也可以将自己的文件向系统内的所有用户开放。

文件访问权限

查看文件和目录的属性: `ls -l`

```
[jypan@euler jypan]$ ls -l
```

```
total 1208
```

-rw-r--r--	1	jypan	tch	702019	6月	7	13:29	2ndECpde2.zip
-rw-r--r--	1	jypan	tch	472956	3月	15	11:52	2ndECpde.zip
drwx-----	2	jypan	tch	4096	5月	13	12:22	backup
drwxr-xr-x	3	jypan	tch	4096	5月	3	01:35	bin
lrwxrwxrwx	1	root	root	11	12月	27	2003	Desktop -> .Desk

文件类型

文件权限

链接个数

文件所有者

文件所有者所在的用户组

文件大小

文件上次修改的时间和日期

文件名

文件访问权限

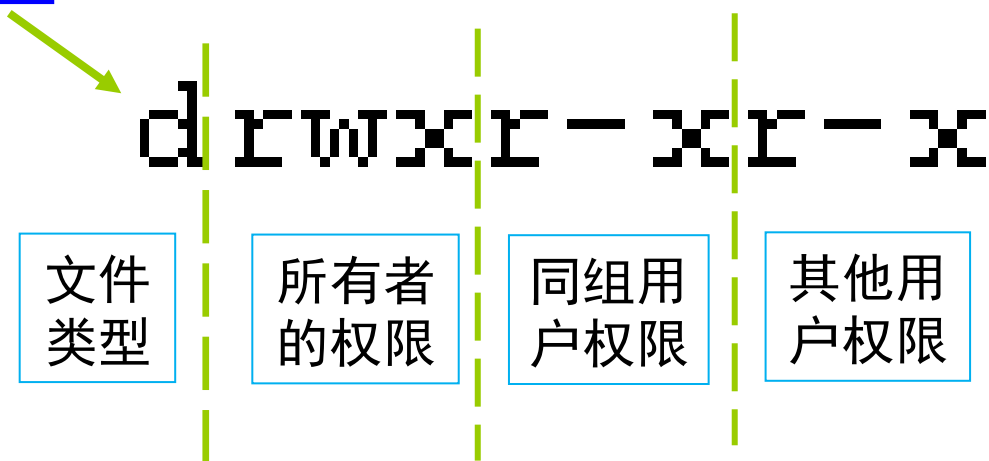
```
drwxr-xr-x    3 jypan    tch                4096  5月  3 01:35 bin
```

□ 在显示的结果中，第一个字段中的第一个字符是表示文件的类型，具体含义为：

字符	代表的文件类型
-	普通文件
d	目录
l	链接文件
b、c	分别代表区块设备和其它外围设备，特殊类型文件
s、p	这些文件类型关系到系统的数据结构和管道，很少见

文件访问权限

```
drwxr-xr-x  3 jypan  tch          4096  5月  3 01:35 bin
```



- ▶ 在显示的结果中，第一个字段的第 2~10 个字符是用来表示权限。这 9 个字符每 3 个一组，组成 3 套权限控制：
- ▶ 第一套控制文件所有者的访问权限
- ▶ 第二套控制所有者所在用户组的其他成员的访问权限
- ▶ 第三套是控制系统内其他用户的访问权限

文件访问权限

d rwx r-x r-x

- 每套控制中三个字符按顺序排列，所代表的含义如下：

r	Read （读取）：对文件而言，具有阅读文件内容的权限， 对目录而言，具有浏览目录内容的权限。
w	Write （写入）：对文件而言，具有修改文件内容的权限， 对目录而言，具有移动、删除目录内文件的权限。
x	eXecute （执行）：对文件而言，具有执行文件的权限， 对目录而言，具有进入目录的权限。
-	表示不具有该权限，或取消该权限

文件访问权限

只有文件所有者具有读、写和执行权限。

-rwx---

普通
文件

系统内其它用户没有
读、写和执行权限

用户组其它成员没有读、写和执行权限

用户有读、写和执行权限

文件访问权限

<code>-rw-r--r--</code>	普通文件，用户有读、写权限， 组用户和其他用户只有读权限。
<code>drwx--x--x</code>	目录，用户有读、写和执行权限， 组用户和其他用户只有执行权限。
<code>drwx-----</code>	目录，用户有读、写和执行权限， 组用户和其他用户不能访问。
<code>-rwxr-xr-x</code>	普通文件，用户有读、写和执行权限， 组用户和其他用户有读和执行权限。

- 用户主目录的默认权限通常为： `rwx-----`

修改访问权限

更改已有文件或目录的访问权限的命令

chmod

□ 两种用法

▶ 文字设定法

使用字母和操作符表达式来修改或设定文件的访问权限

▶ 数字设定法

利用数字来设定文件的访问权限

chmod 文字设定法

chmod [who] [+|-|=] [mode] 文件或目录列表

操作对象

操作符

权限

● 操作对象可以是下述字母中的任一个或者它们的组合：

- ✓ u 表示“用户（user）”，即文件或目录的所有者
- ✓ g 表示“同组（group）用户”
- ✓ o 表示“其他（others）用户”
- ✓ a 表示“所有（all）用户”
- ✓ 如果没有指定操作对象 who，则缺省值为 a

chmod 文字设定法

- 操作符可以是：

- ✓ + 添加某个权限
- ✓ - 取消某个权限
- ✓ = 赋予给定权限

- 访问权限可用下述字母的任意组合：

- ✓ r 可读
- ✓ w 可写
- ✓ x 可执行

- 文件或目录列表：可以单个或多个文件/目录，支持通配符

chmod 文字设定法

```
chmod o-rx mydata/
```

```
chmod o=rx mydata/
```

```
chmod go=r mydata/
```

```
chmod +x filename # 等价于: chmod a+x filename
```

```
chmod +w filename # 等价于: chmod u+w filename
```

chmod 数字设定法

chmod num 文件或目录列表

权限	对应数字
r	4
w	2
x	1
-	0

-rw-r--r-- ↔ 644

drwx--x--x ↔ 711

drwx----- ↔ 700

-rwxr-xr-x ↔ 755

Example

chmod 644 filename # 设定权限为: rw-r--r--

chmod 750 filename # 设定权限为: rwxr-x---

chmod 700 filename # 设定权限为: rwx-----

进程与进程管理

- ▶ 正在运行的程序叫做**进程** (process) , 程序只有被系统载入内存并运行后才能称为进程。
- ▶ Linux 允许同时运行多个程序, 为了区分每一个运行的程序, 系统会给每个进程赋标号, 称为**进程号** (process ID) , 每个进程的进程号唯一
- ▶ Linux 是**多用户多进程**操作系统
- ▶ 查看当前进程及其进程号的命令: `ps`
- ▶ 终止一个进程: `ctrl+c` / `kill` / `killall`

Linux 进程

前台与后台

- ▶ **前台进程**：指一个程序控制着标准输入/输出，在程序运行时，shell 被暂时挂起，直到该程序运行结束后，才退回到 shell。在这个过程中，用户不能再执行其它程序。
- ▶ **后台进程**：用户不必等待程序运行结束就可以执行其它程序。
- ▶ 在一个终端里只能同时存在一个前台任务，但可以有多多个后台任务。
- ▶ 运行后台进程：在命令最后加上 “&”
- ▶ 如果程序已经在前台运行，需要将其改为后台运行，可以先按组合键 **ctrl+z**，将任务挂起，然后利用 **bg** 命令将该程序转为后台运行
- ▶ 若要将一个后台进程转到前台运行，可以使用 **fg** 命令
- ▶ 相关命令：**jobs, bg, fg**

查看进程

ps [选项]

-A, -e	显示所有进程
-f	完全显示
-l	长格式显示, 可查看各个进程的优先权值
-u	查看指定用户的进程
u	增加显示用户名, 进程起始时间, CPU 和内存占用百分比等信息
a	显示终端机下用户执行的进程, 包含其它用户
r	显示正在运行的进程

```
ps af
ps -u jypan -l
ps u
```

Linux 进程

列标志的含义

PID	进程 ID	CMD	命令名 (COMMAND)
UID	用户 ID	START	进程启动时间
USER	用户名	%CPU	进程所用CPU时间百分比
TIME	执行时间	%MEM	进程所有MEM百分比
STAT	进程状态	NI	优先权值 / nice 值

TTY	启动进程的终端	RSS	进程所用内存块数
PGID	进程组 ID	VSZ	所用虚拟内存块数

注 更多列标志的含义: `man ps`

Linux 进程

相关命令

`id` # 显示 uid, gid

`w` # 显示登录用户

`top` # 显示计算机资源使用情况

`hostname` # 计算机名称

`uname` # 操作系统相关信息



3 Linux 常用命令

- 1 Linux 操作系统介绍
- 2 Linux 基本操作与管理
- 3 Linux 常用命令
- 4 Linux 程序开发环境

□ 更多 Linux 命令可参见课程主页

□ 详细命令用法:

<https://wangchujang.com/linux-command/>

Linux 常用命令

ls	显示目录内容
cd	切换工作目录
pwd	显示工作目录
cp	拷贝文件
mkdir	创建目录
rmdir	删除空目录
mv	移动文件
rm	删除文件
chmod	改变文件的权限

cat	查看文件内容
more	查看文件内容
less	查看文件内容
ln	创建文件链接
tar	文件打包
unzip unrar	文件压缩和解压

更多参见课程主页或相关资料

Linux Shell

终端命令行一些使用技巧

- ▶ 自动补全功能: `Tab`
- ▶ 历史记录: `history`、上下箭头键
- ▶ 通配符: `*` `?` `[]`
- ▶ 别名: `alias`
- ▶ 重定向和管道: `>` `>>` `<` `<<` `|`
- ▶ 环境变量查看: `env` 查看所有环境变量

ls 显示当前目录下的文件

ls [选项] [目录或是文件]

- 对于目录，该命令将列出其中的所有子目录与文件
- 对于文件，ls 将输出其文件名以及所要求的其他信息

常用
选项

- a 列出目录下的所有文件，包括以 . 开头的隐藏文件
- l 列出文件的详细信息，通常称为“长格式”
- t 以时间排序
- d 输入参数是目录时，只显示该目录名，而不是列出该目录下的文件
- R 列出所有子目录下的文件
- S 以文件大小排序
- r 反序输出
- help 显示简短帮助信息
- version 输出版本信息

```
ls -l
```

```
ls -al
```

cd / pwd 切换和打印工作目录

cd 目录名

- 进入指定的目录，即将工作目录改为指定的目录

pwd

- 输出当前的工作目录，即当前在哪个目录下

CP 复制文件和目录

cp [选项] 源文件或目录 目标文件或目录

- 把源文件复制为目标文件，若目标文件已存在，则被覆盖
- 把多个源文件复制到目标目录中
- 把源目录复制为指定的目标目录（目标目录不存在）
- 把源目录复制到指定的目标目录中

常用选项

- f 若目标文件与源文件同名，则直接覆盖而不提示
- i 交互式拷贝，即覆盖目标文件之前将给出提示
- p 保留源文件的修改时间和访问权限等属性
- r 递归，**复制目录**时必须加此选项

```
cp file1 /home/jypan/data/
```

```
cp -pr Teaching/ParaComp/Codes/ Temp/
```

mv 移动文件或目录/重命名

mv [选项] 源文件或目录 目标文件或目录

- 将源文件改名为目标文件，若目标文件已存在，则被覆盖
- 将多个源文件移至目标目录中
- 将源目录改名为指定的目标目录（目标目录不存在）
- 将源目录移至指定的目标目录中

常用选项

- f 若目标文件与源文件同名，则直接覆盖而不提示
- i 在覆盖目标文件之前要求用户确认（交互式移动）
- u 如果要被覆盖的文件或目录比源文件要新，则不覆盖

```
mv file1 dir/ // 将当前目录下的 file1 移至 dir 目录中
```

```
mv file1 file2 // 将 file1 改名为 file2
```

```
mv dir1/ dir2 // 若 dir2 存在，则表示移动，否则表示改名
```

rm 删除文件或目录

rm [选项] 文件或目录名

- 删除指定的文件或目录

常用
选项

- f 强制删除，并且不给提示
- I 删除每个文件时给用户提示
- r 递归删除，即删除整个目录，包括文件和子目录

```
rm file1 file2
```

```
rm file*
```

```
rm -rf dir1/
```

注 删除后很难恢复，谨慎使用，特别是以超级用户身份进行相关操作时

mkdir / rmdir 创建和删除目录

mkdir 目录名

rmdir 目录名

```
mkdir newdir // 创建一个新目录
```

```
rmdir olddir // 删除一个空目录
```

注 **rmdir** 只能删除空目录，非空目录可用 **rm -r** 删除

cat/more/less 查看文本文件内容

cat [选项] 文件列表

- 功能一：显示文件内容
- 功能二：将两个或多个文件连接起来（重定向）

常用选项

-n 在文件的每行前面显示行号

```
cat cpi.c // 把文件 cpi.c 的内容全部显示到屏幕上
```

```
cat file1.txt file2.txt > file.txt // 合并文件
```

more 文件名

less 文件名

- 逐屏显示文本，按空格键显示下一屏，退出按 **q**
- **more** 只能往前翻，**less** 可以相前翻，也可以向后翻

tar/unzip/unrar 打包解包

tar [主选项+辅选项] 文件或者目录

- 将多个文件或目录打包成一个文件
- **tar** 有主选项和辅选项，主选项是必须的，辅选项是可选的

常
用
主
选
项

- c 创建新的档案文件（打包）
- x 从档案文件中释放文件（解包）
- r 把要存档的文件追加到档案文件的末尾
- t 列出档案文件的内容，查看已经备份了哪些文件
- u 更新档案中的已有文件

常用
辅助
选项

- f 使用档案文件或设备，这个选项通常是必选的
- v 详细列出 **tar** 处理的文件信息
- z 用 **gzip** 来压缩/解压缩文件包

tar/unzip/unrar 打包解包

```
tar -cvf data1.tar file1 file2 file3    // 打包
tar -czvf data1.tgz file1 file2 file3    // 打包时进行压缩
tar -xzvf data1.tgz    // 解包
```

注 tar 的选项前的 符号 “-” 可以不加

注 用 tar 打包的文件在 windows 下可以用 winrar 打开

注 Windows 下的 zip/rar 文件, 在 Linux 下可以用 unzip/unrar 解开

```
unzip data.zip
unrar x data.rar
```

ln 链接

ln [选项] 目标 [链接名]

- 为**目标**创建一个链接（类似快捷键）
- 删除链接，不会破坏原来的文件和目录
- 链接有两种：**硬链接**和**符号链接**，缺省是建立硬链接
- 若没有指定链接名，则链接名与原文件名相同

常用
选项

-s 建立符号链接/软链接

```
ln -s /opt/MATLAB/R2022/bin/matlab.exe ./ // 创建文件链接
```

```
ln -s /export/home/szhang ./sz // 创建目录链接，链接名与原名不同
```



4

Linux 程序开发环境

- 1 Linux 操作系统介绍
- 2 Linux 基本操作与管理
- 3 Linux 常用命令
- 4 Linux 程序开发环境

文本编辑工具



`vi` 文件名 / `vim` 文件名

- ▶ vim 是 Linux 下最常用的文本编辑器之一
- ▶ 文件若存在，则打开该文件，否则将新建一个文件

vim 工作模式

工作模式

→ 命令模式

该模式下，所有的输入都被看成是**命令**

→ 插入模式 / 编辑模式

在该模式下可以输入文件内容

● 工作模式的切换



● 启动 vim 编辑器时，首先进入的是**命令模式**

● 在任何模式下，按 **ESC** 键 便可返回到命令模式

vim 工作模式

退出 vim

→ 先保存，后退出

→ 具体步骤：

第一步：切换成命令模式

第二步：使用下面的命名保存文件或退出 vim

✓ **:w** 保存当前编辑文件，但并不退出vim

✓ **:q** 退出 vim

✓ **:wq** 表示存盘并退出vim

✓ **:q!** 放弃所作的修改而直接退出vim

一些基本操作

▶ 光标移动：可使用方向键或命令，但不能使用鼠标

▶ 删除文本：插入模式：**Delete**、**Backspace**；命令模式：**x**、**dd**、重复因子

Linux 编程环境

► 编译: gcc

```
gcc -O2 -o hello hello.c // C
```

```
g++ -O2 -o hello hello.cpp // C++
```

```
gfortran -O2 -o hello hello.f95 // Fortran
```

► 运行可执行文件

```
./hello
```

GCC - GNU Compiler Collection

gcc [选项] 源代码文件

□ 常用选项

- ▶ -o : 指定输出文件名, 缺省为 **a.out**
- ▶ -c : 只编译不链接, 即只生产目标文件 (.o 文件)
- ▶ -O1, -O2, -O3 : 优化开关
- ▶ -w : 关闭所有警告信息
- ▶ -Ipath : 指定或增加包含文件 (如 *.h) 的搜索路径
- ▶ -Lpath : 指定 (增加) 库文件的搜索路径
- ▶ -lname : 与库文件 libname.a 链接
- ▶ -g : 在目标码中加入更多信息, 用于程序调试
- ▶ -std=c99 -pedantic-errors : C99标准
- ▶ **-std=c11** : C11标准
- ▶ -std=c17 : C17标准 (与C11基本一样)

Linux 编程环境

(1) 编写源代码：可以在个人电脑上用自己喜欢的 IDE 编写和小规模测试

如：Dev C++, VS Code

(2) 远程登录服务器，上传源代码

远程登录软件：SSH Secure Shell Client, SecureCRT, XshellPlus

(3) 用 gcc 编译，可根据需要适当修改源代码，将结果输出到文件中

(4) 运行可执行文件，等待程序运行结束

LINUX