Linux

寒假训练项目-Week1

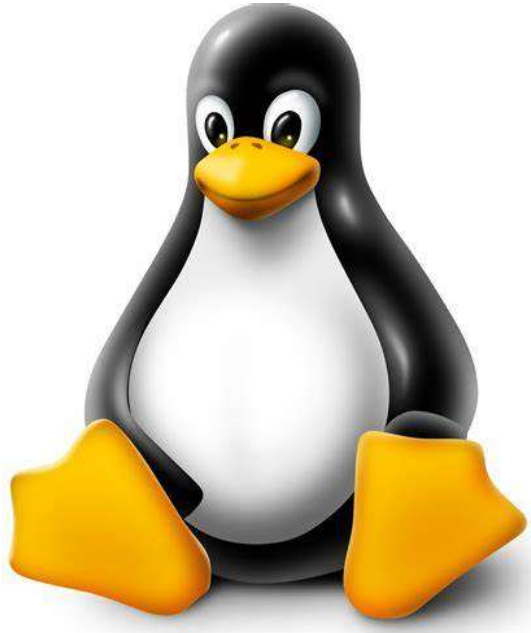
清华大学自动化系学生科协

王嘉辉

2023.1.16

目录

- Linux起源
- 为何选用Linux
- 如何在Windows下使用Linux
- 常见命令
- vim
- ssh
- 文件权限与chmod
- tmux
- 拓展资料



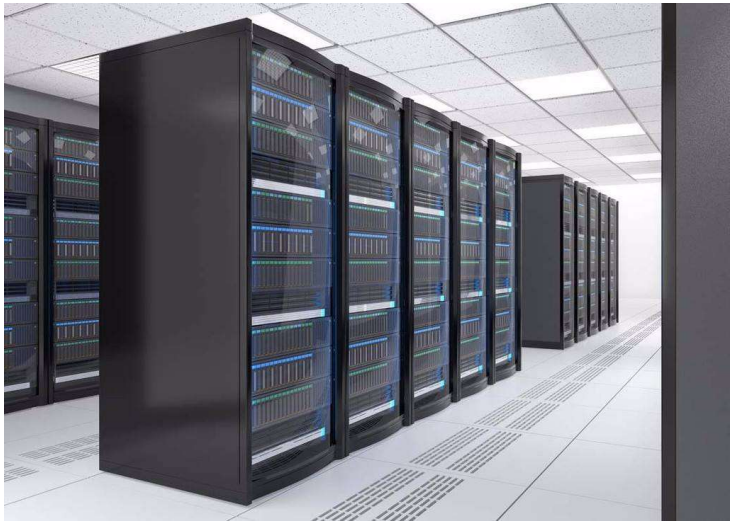
1.Linux起源

- 1969年，美国 AT&T 公司贝尔实验室开发了 UNIX 操作系统
- 1983年，理查德·斯托曼在MIT发起了GNU计划，并诞生了GPL
- 1991年，林纳斯·托瓦兹在大学时期编写并发布了Linux内核

- 随后诞生了许多基于Linux内核的操作系统，称作Linux发行版
- 例如：Debian,Red Hat等分支
- 以ubuntu-18.04 LTS为例

身边的Linux

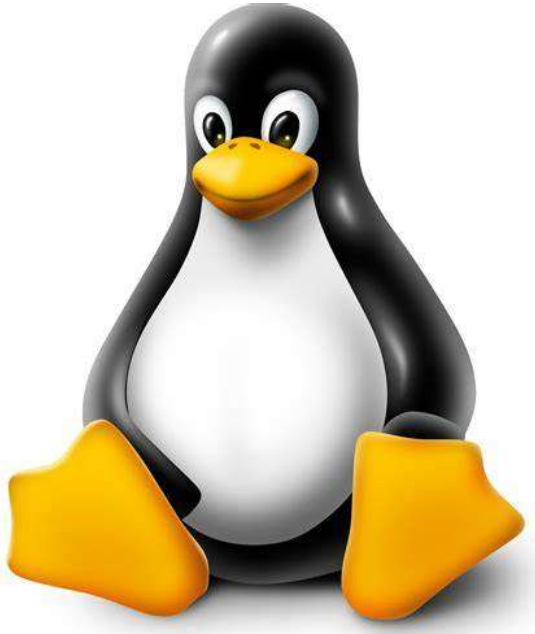
- Android，是Linux的一个发行版
- 服务器，大多数是Linux发行版
- 嵌入式开发板，常用Linux系统



2.为何选用Linux?

- 开源
- 安全
- 稳定

- 多数没有图形界面GUI



3.Windows下使用Linux

- 安装虚拟机
- 安装wsl
- 安装双系统
- 连接Linux服务器

安装虚拟机（推荐）

- 虚拟机：可视作一个安全可靠的沙盒
- 优点：
  - 不影响当前操作系统的使用
  - 无需考虑底层硬件兼容性
  - 利于试错，可以重置虚拟机镜像，而不会危害本机数据

安装虚拟机

- 主流虚拟机软件
  - VMware Workstation Player
  - VMware Fusion Player
  - VirtualBox
- 官网下载后，按照步骤安装即可

安装wsl（推荐）

- wsl: Windows Subsystem for Linux
- wsl可让开发人员直接在 Windows 上按原样运行 GNU/Linux 环境（包括大多数命令行工具、实用工具和应用程序），且不会产生传统虚拟机或双启动设置开销。
- wsl1 wsl2

功能	WSL 1	WSL 2
Windows 和 Linux 之间的集成	✅	✅
启动时间短	✅	✅
与传统虚拟机相比，占用的资源量少	✅	✅
可以与当前版本的 VMware 和 VirtualBox 一起运行	✅	✅
托管 VM	❌	✅
完整的 Linux 内核	❌	✅
完全的系统调用兼容性	❌	✅
跨 OS 文件系统的性能	✅	❌

安装wsl

- 参考文档：安装 WSL | Microsoft Learn

- 必须运行 Windows 10 版本 2004 及更高版本（内部版本 19041 及更高版本）或 Windows 11 才能使用以下命令。

- 以管理员身份打开powershell，执行wsl --install，随后重启计算机

- 打开wsl并设置用户名和密码，参考文档：设置 WSL 开发环境 | Microsoft Learn

安装双系统（不建议新手）

- 不建议新手使用，容易出现的问题：
  - 在安装过程中不理解关键的选项（如：磁盘分区、挂载、交换空间分配等）的意义，很容易做出错误的决定；
  - 错误的配置可能导致自己原先本机上的操作系统和数据遭到不可逆转的损坏；
  - 部分硬件可能对安装的发行版缺少兼容，从而导致意外安装失败；
  - 如果安装的过程中选择下载附加工具，可能会因为默认镜像在国外而导致下载十分缓慢，从而让安装流程变得很漫长；

连接Linux服务器

- 较为简单，但是需要先有服务器和账号
- 随后演示

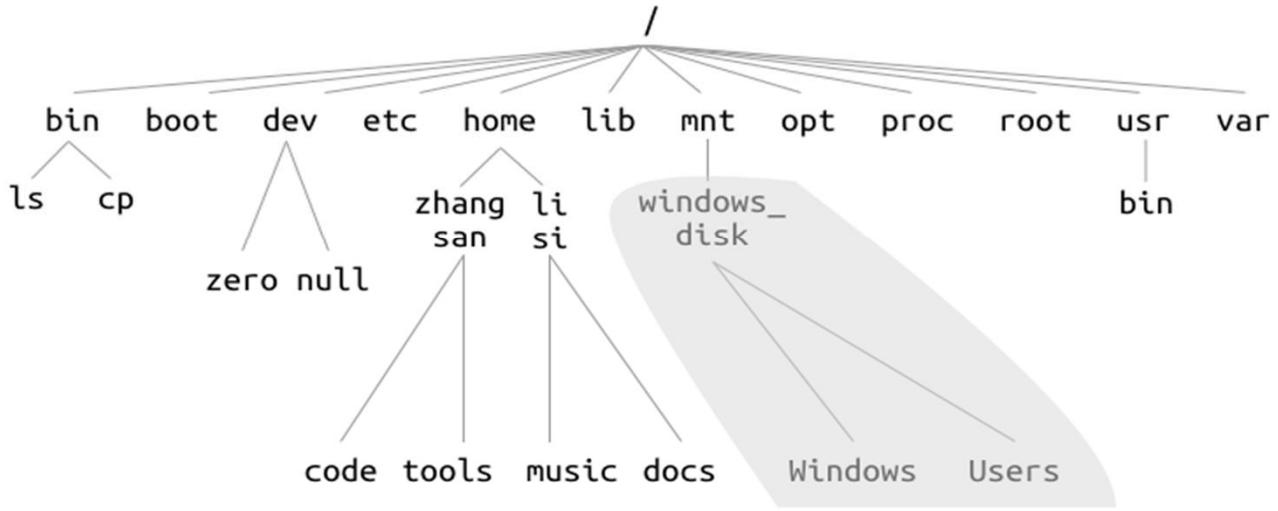
4.常见命令--问题引入

- 问题引入
- 1.我是谁，我在哪?
- 2.我下面这个文件夹怎么进去？怎么去上一级文件夹?
- 3.怎么看看当前文件夹下的兄弟姐妹?
- 4.这个文件怎么进不去啊(╯‾皿‾)╯
- 5.我也想当管理员行不行?

4.常见命令

- 目录结构

  /home/username/ASTA_test/



4.常见命令

- **显示当前目录**
  - `pwd`　　`pwd = print work directory`
- **当前目录下新建文件夹**
  - `mkdir ASTA_test`
- **当前目录下新建文件**
  - `touch a.cpp`
- **进入指定目录**
  - `cd ASTA_test`　　`cd = change directory`
- **进入上一层目录**
  - `cd ..`

4.常见命令

- 查看当前目录下的文件
  - ls　　ls = list
- 查看当前目录下的所有文件
  - ls -a
- 列出文件详细信息(如权限)
  - ls -l (filename)
  - ll

4.常见命令

- 删除指定文件
  - `rm a.cpp`
- 删除当前目录及目录下的所有文件
  - `rm -r ASTA_test`
- 删库跑路 ε=ε=ε=(~￣▽￣)~
  - `rm -rf /*`

4.常见命令

- 移动a.cpp到ASTA_test目录
  - mv a.cpp ASTA_test/
- 将a.cpp拷贝到ASTA_test目录
  - cp a.cpp ASTA_test/
- 将a目录下所有文件拷贝到ASTA_test目录
  - cp –r a ASTA_test

4.常见命令

- **清屏**
  - clear
- **显示历史记录和执行过的命令**
  - history
- **显示最近执行的n条命令**
  - history n
- **执行历史命令**
  - !id
  - !!

4.常见命令

- **查找文件所在位置**
  - `whereis bash`
  - **`-m` 说明文件　　`-b` 二进制文件　　`-s` 源代码文件**
- **查找命令是否存在及存放位置**
  - `which vim`

4.常见命令

➢查看文件内容

- cat a.cpp

➢输入输出重定向

- echo hello,world!
- echo “hello,\“linux\””　　# 转义\　-e
- echo “hello,world!” > a.txt　　# 覆盖>
- echo “I’m Tom” >> a.txt　　# 追加>>

- echo $USER
- echo $PATH
- echo *txt

4.常见命令

- **安装软件包**
  - `apt install g++`
- **更新软件包**
  - `apt update <package_name>`
- **查找软件包**
  - `apt search <package_name>`
- **删除软件包**
  - `apt remove <package_name>`

4.常见命令

➢提升权限

- sudo
- sudo apt install g++

- 要求当前用户拥有运行sudo的权限
- 系统首先会通过/etc/sudoers文件验证用户权限
- 确认权限后需要输入用户密码确认
- 密码正确，执行sudo命令

4.常见命令

- 管道 |
  - |command1 | command2
  - 管道|左边的输出成为右边的输入
  - 语法紧凑简单

4.常见命令

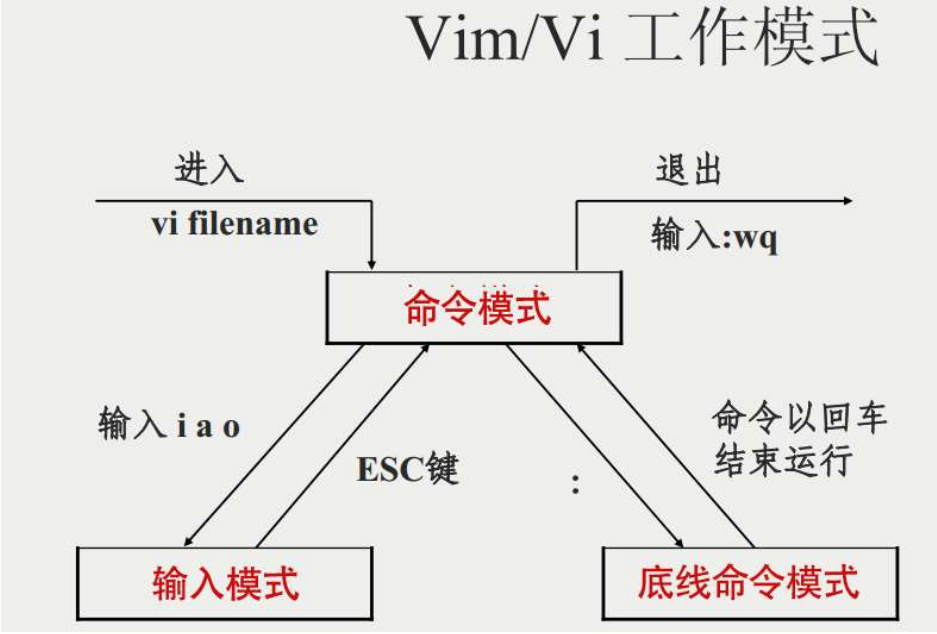
- 查找文件里符合条件的字符串
  - grep ASTA hello_ASTA.txt
  - grep ASTA *ASTA.txt

- 查找文件、历史指令等
  - ls | grep hi*
  - history | grep “g++”

休息5min

5.vim

- vim是从vi发展而来的一个文本编辑器，对于编写程序比较方便。



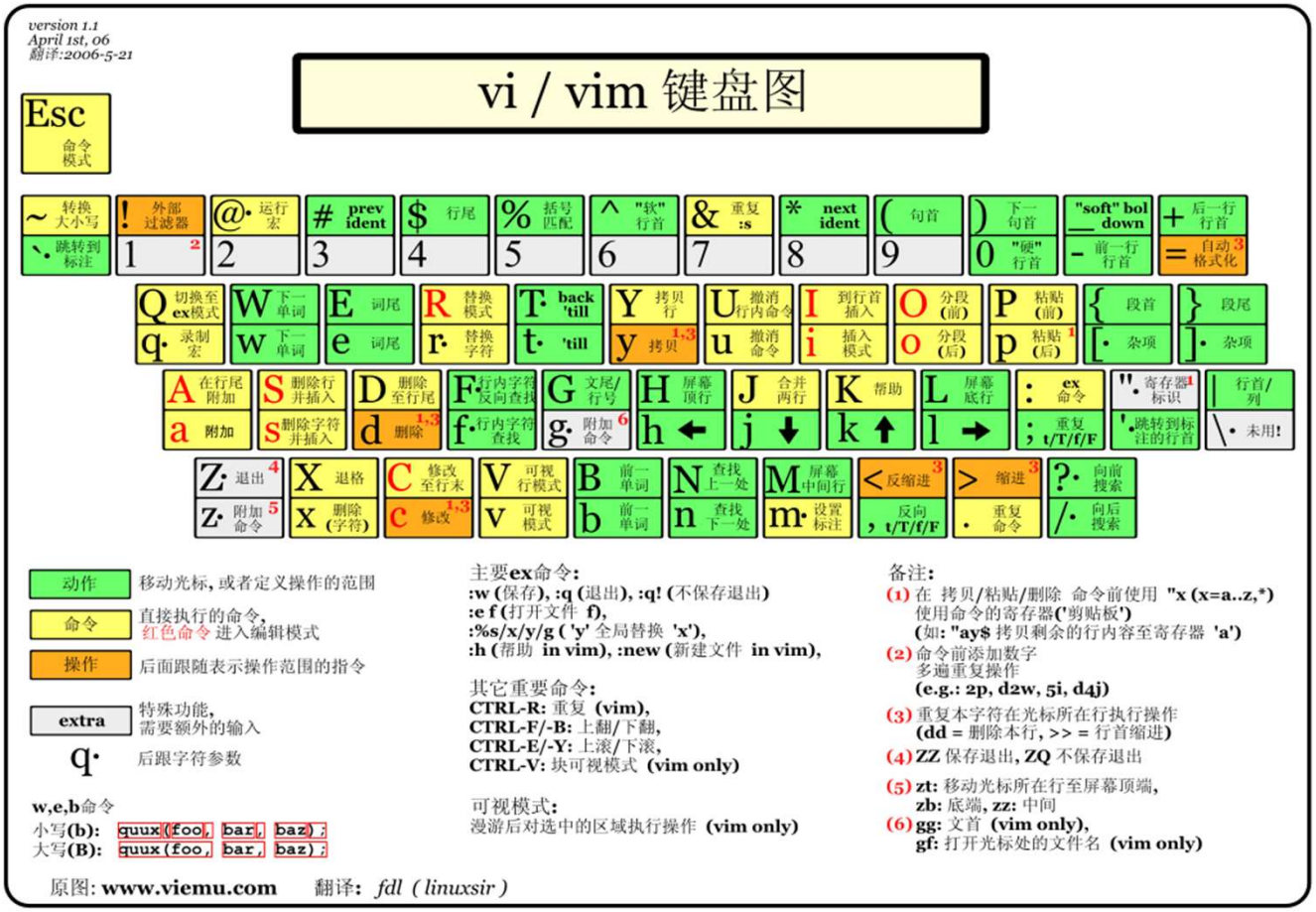
5.vim

➢常用操作:

- i 切换到输入模式，可以修改文件内容
- r 切换到取代模式，可以修改文件内容
- Esc从输入模式退出到命令模式

- 在命令模式下，
- 输入:q　　不保存退出
- 输入:w　　保存
- 输入:wq　保存并退出
- 输入:wq!　强制保存并退出

5.vim


version 1.1
April 1st, 06
翻译:2006-5-21
vi / vim 键盘图
Esc 命令模式
动作 移动光标, 或者定义操作的范围
命令 直接执行的命令, 红色命令 进入编辑模式
操作 后面跟随表示操作范围的指令
extra 特殊功能, 需要额外的输入
q· 后跟字符参数
w,e,b命令
小写(b): quux(foo, bar, baz);
大写(B): quux(foo, bar, baz);
主要ex命令:
:w (保存), :q (退出), :q! (不保存退出)
:e f (打开文件 f),
:%s/x/y/g ('y' 全局替换 'x'),
:h (帮助 in vim), :new (新建文件 in vim),
其它重要命令:
CTRL-R: 重复 (vim),
CTRL-F/-B: 上翻/下翻,
CTRL-E/-Y: 上滚/下滚,
CTRL-V: 块可视模式 (vim only)
可视模式:
漫游后对选中的区域执行操作 (vim only)
备注:
(1) 在 拷贝/粘贴/删除 命令前使用 "x (x=a..z,*)
使用命令的寄存器('剪贴板')
(如: "ay$ 拷贝剩余的行内容至寄存器 'a')
(2) 命令前添加数字
多遍重复操作
(e.g.: 2p, d2w, 5i, d4j)
(3) 重复本字符在光标所在行执行操作
(dd = 删除本行, >> = 行首缩进)
(4) ZZ 保存退出, ZQ 不保存退出
(5) zt: 移动光标所在行至屏幕顶端,
zb: 底端, zz: 中间
(6) gg: 文首 (vim only),
gf: 打开光标处的文件名 (vim only)
原图: www.viemu.com 翻译: fdl (linuxsir)

6.***

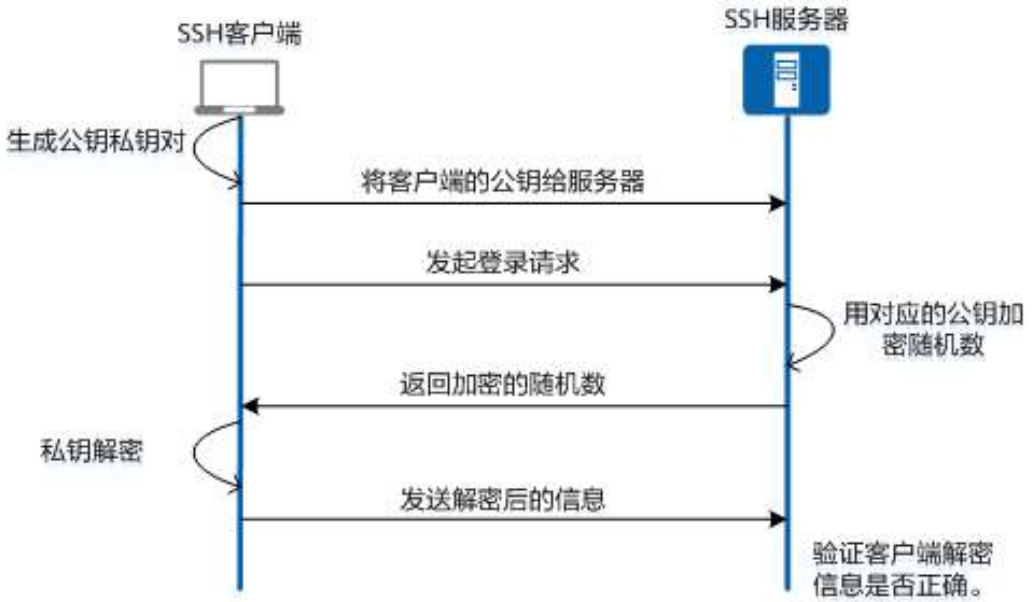
- 学长学姐跟我说服务器有八张A100,可是服务器在哪呢？我怎么连接呢?

- 连上了服务器，但是我怎么把我的程序上传上去呢？又怎么拉下来呢?

6.SSH

- SSH(Secure Shell) 是建立在应用层基础上的安全协议，可以用其远程登录服务器/其他电脑.

- SSH登录流程



6.SSH

- 本机生成密钥
  - ssh-keygen
  - cd ~/.ssh
  - cat id_rsa.pub
- 将公钥添加到/home/user/.ssh/authorized_keys

  即可免密登录

- ssh remote_username@remote_address
  - remote_username是服务器端的用户名
  - Remote_address是服务器地址,可以是ip/域名

6.SSH

- SCP(Secure Copy)是Linux系统下基于SSH登陆进行安全的远程文件拷贝命令。

- 从服务器拷贝到本地

```
scp remote_username@remote_address:remote_filename local_folder
```

- 从本地拷贝到服务器

```
scp local_file remote_username@remote_address:remote_folder
```

- 对于文件夹需加参数-r

```
scp –r local_file remote_username@remote_address:remote_folder
```

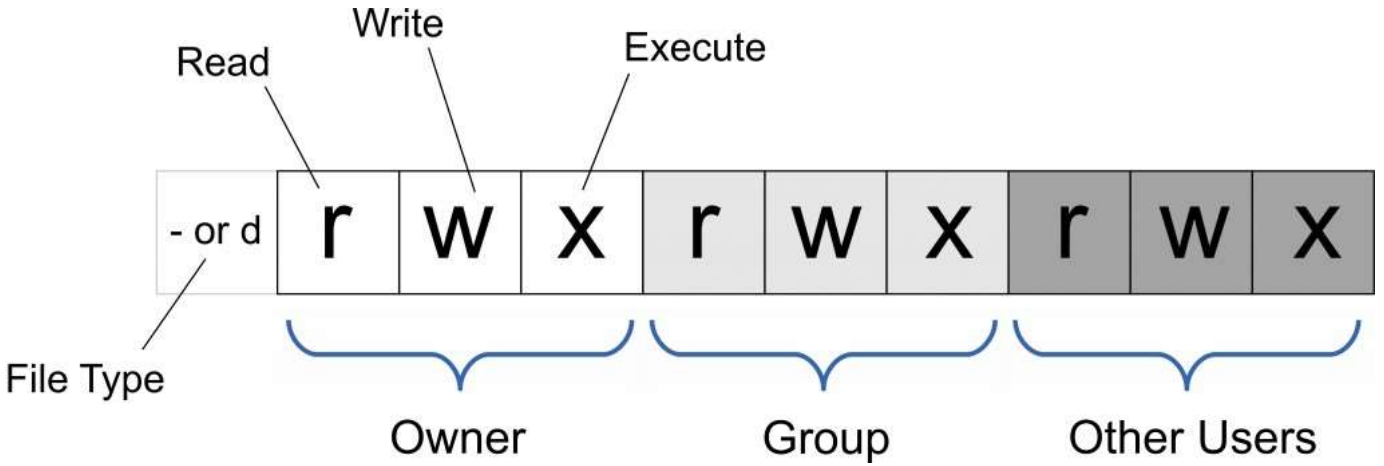
7.文件权限

- 问题引入
- 实验资料很贵重，怎么能设置一下，让合适的人访问特定的文件?

7.文件权限

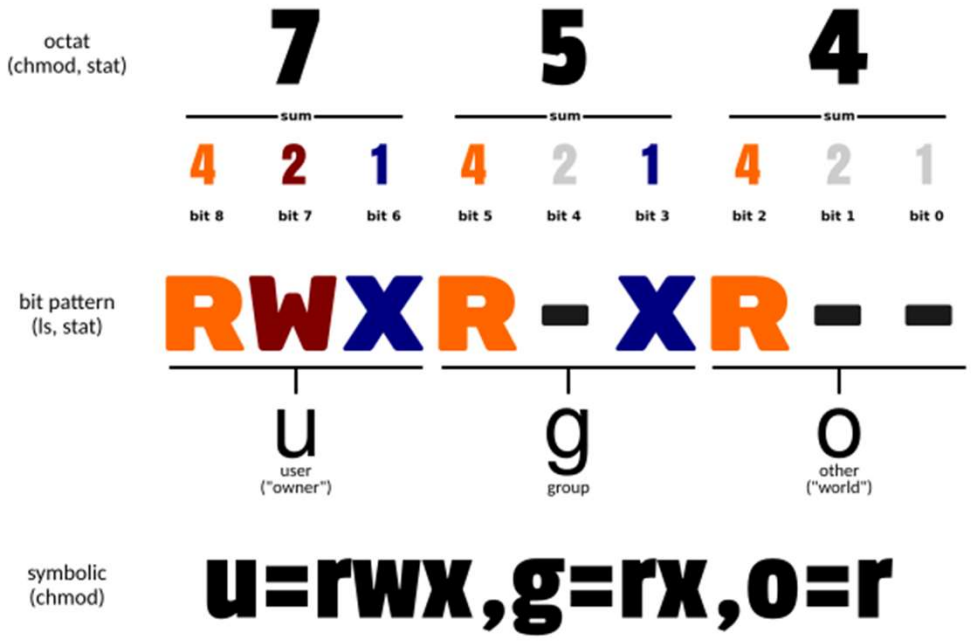
- Linux文件权限分为三级:文件所有者(owner)、用户组(group)、其他用户(other users)

- 如何查看权限: ls -l或者ll



7.文件权限

- 首位表示文件类型
  - d:文件夹
  - -:文件
- 接下来每三位一组，分别代表所有者、用户组、其他用户的权限
- 每一组可以使用八进制数表示，也可以使用形如g=rx表示
- u(user) g(group) o(others) a(all)



7.文件权限

- chmod(change mode)是用来控制用户对文件的权限的命令
- 注意:只有文件所有者和超级管理员才可以修改权限
- 如何修改权限:

  1.使用字符:

  ```
  chmod a+r filename   (等价于  chmod ugo+r filename)
  chmod u+w filename
  chmod ug+x,o-w filename
  chmod ug=rw,o=rx filename
  ```

  2.使用八进制数:

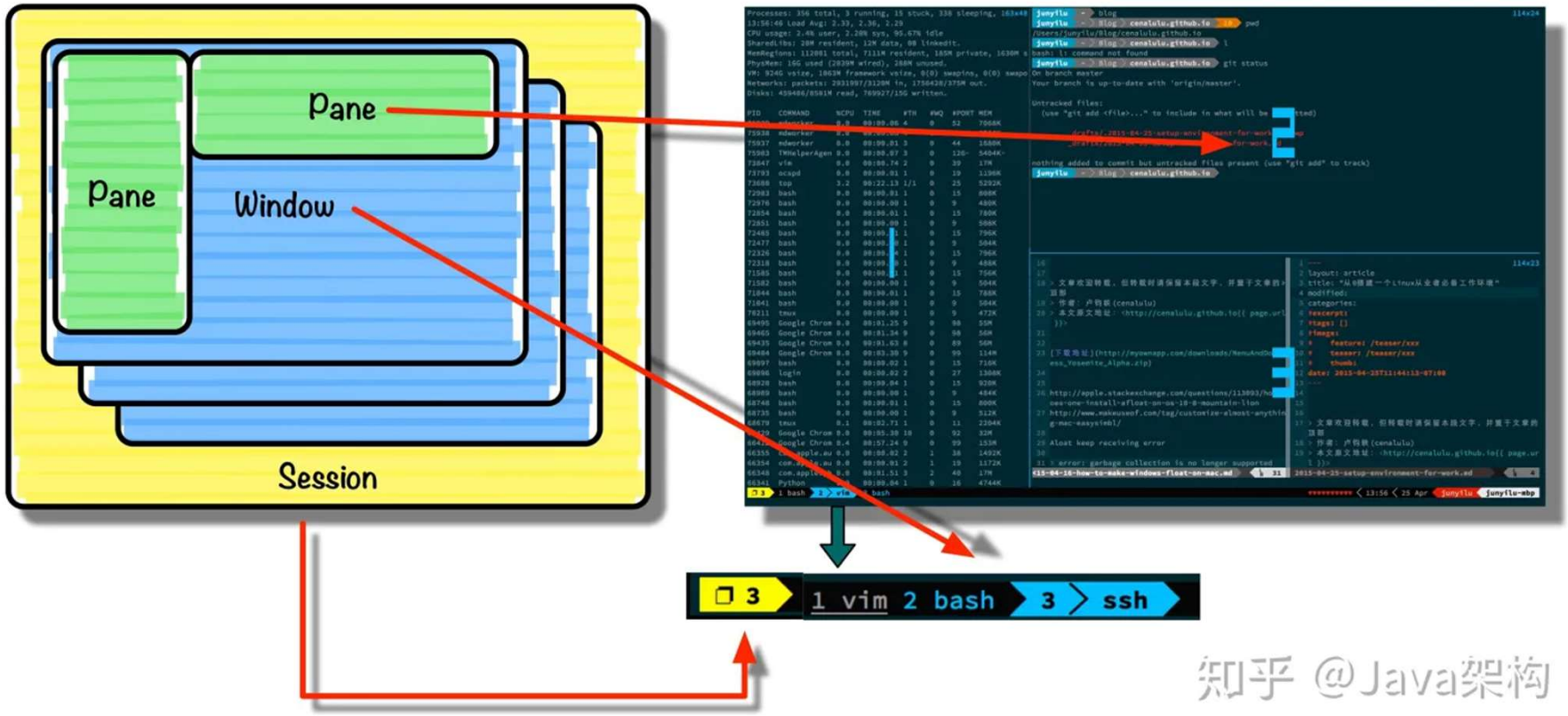
  ```
  chmod 777 filename   (等价于  chmod a=rwx filename)
  chmod 700 filename   (等价于  chmod u=rwx,go= filename)
  ```

8.tmux

- tmux(terminal multiplexer)是一个终端复用器。
- 会话(session):每次打开终端窗口(window)，输入命令。这种用户与计算机临时的交互，称作一次“会话”。每个会话都是一个独立工作区，包含一到多个窗口。
- 会话的一个特点是，窗口与其中启动的进程绑定。
- 如何解绑?

8.tmux

- Session Window Pane



8.tmux

- 安装tmux
  - sudo apt install tmux
- 打开tmux
  - tmux
- 退出tmux界面
  - Ctrl+b后再输入d
  - Ctrl+d
  - 输入exit
- 前缀键
  - 输入Ctrl+b后，再输入的指令才会生效

8.tmux

- 新建会话
  - tmux new -s   <session_name>
- 分离会话
  - tmux detach
  - Ctrl+b d
- 查看所有会话
  - tmux ls
- 接入会话
  - tmux attach –t <session_name>

8.tmux

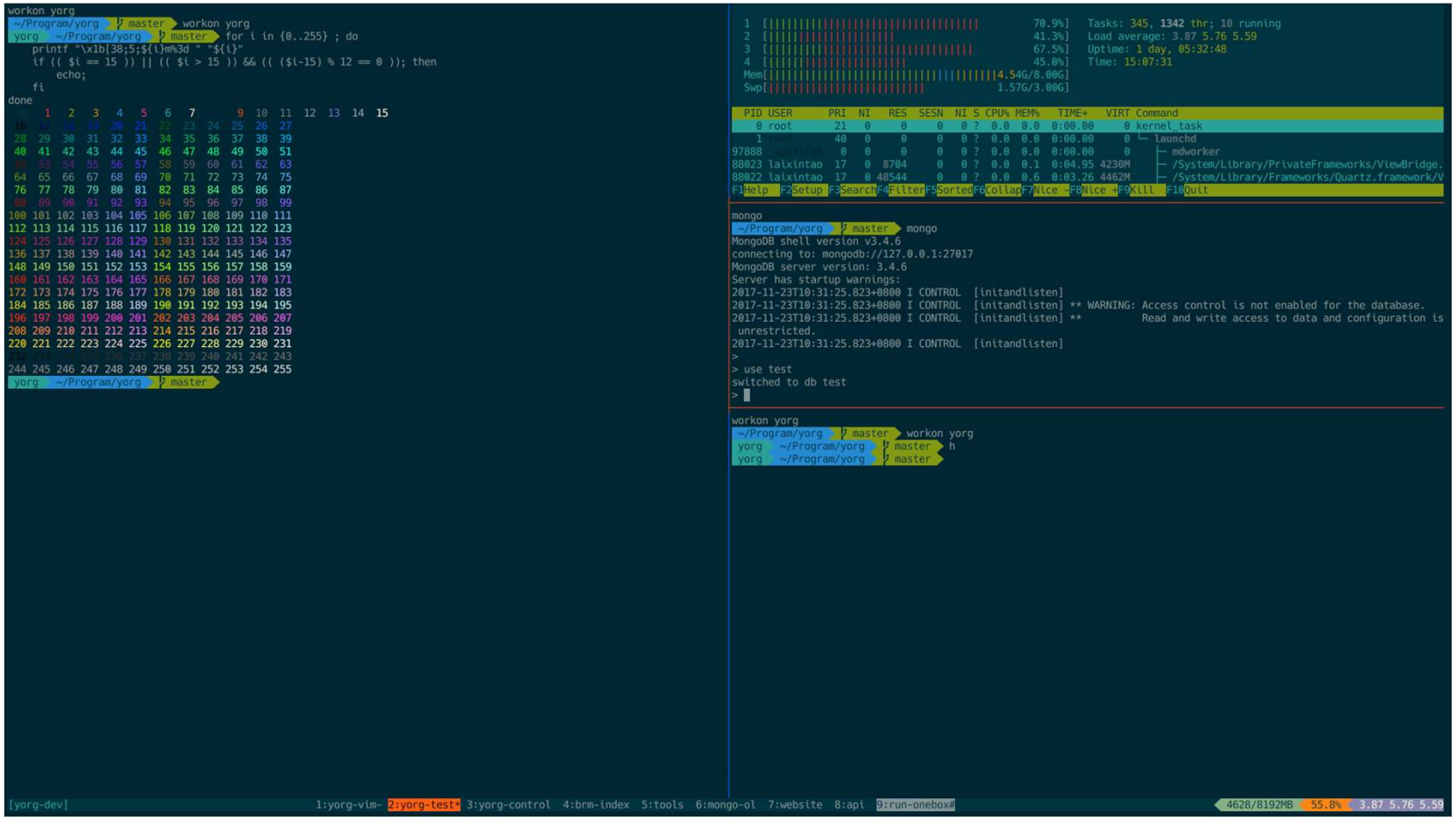
- 杀死会话
  - tmux kill-session -t <session_name>
- 切换会话
  - tmux switch -t <session_name>
- 重命名会话
  - tmux rename-session -t 0 <new_name>
  - Ctrl+b $ 重命名当前会话

8.tmux

- 新建窗口
  - Ctrl+b c
- 切换窗口
  - Ctrl+b num
  - Ctrl+b p
  - Ctrl+b n
- 窗口划分
  - Ctrl+b “ 上下
  - Ctrl+b % 左右
  - Ctrl+b +上下左右 切换pane

8. tmux

8. tmux


workon yorg
~/Program/yorg master workon yorg
yorg ~/Program/yorg master for i in {0..255} ; do
printf "\x1b[38;5;${i}m%3d " "${i}"
if (($i == 15)) || (($i > 15)) && ((($i-15) % 12 == 0)); then
echo;
fi
done
Tasks: 345, 1342 thr; 10 running
Load average: 3.87 5.76 5.59
Uptime: 1 day, 05:32:48
Time: 15:07:31
PID USER PRI NI RES SESN NI S CPU% MEM% TIME+ VIRT Command
0 root launchd
mongo
~/Program/yorg master mongo
MongoDB shell version v3.4.6
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.4.6
Server has startup warnings:
2017-11-23T10:31:25.823+0800 I CONTROL [initandlisten]
2017-11-23T10:31:25.823+0800 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2017-11-23T10:31:25.823+0800 I CONTROL [initandlisten] ** Read and write access to data and configuration is unrestricted.
> use test
switched to db test
workon yorg
[yorg-dev] 1:yorg-vim- 2:yorg-test* 3:yorg-control 4:brm-index 5:tools 6:mongo-ol 7:website 8:api 9:run-onebox#

9.拓展资料

- https://missing-semester-cn.github.io

9.拓展资料——环境变量

➢问题引入

- 在命令行中执行命令时，例如`cp a.cpp ASTA_test/` 系统会在当前文件夹下寻找`a.cpp`
- 但是例如g++编译时，当前文件夹下并没有g++，系统去哪里寻找g++呢?
- 系统除了在当前文件夹下寻找，还会到PATH环境变量中寻找

9.拓展资料——环境变量

➢环境变量

- 指的是在操作系统中用来指定操作系统运行环境的一些参数，如系统文件夹的位置
- 当系统执行一个程序而没有告知程序所在的完整路径时，系统除了在当前文件夹下寻找外，还会前往PATH指定路径中寻找。

➢Linux下环境变量

- echo $PATH可以查看当前PATH环境变量
- 环境变量存放在~/.bashrc中
- source ~/.bashrc可以立即启用
- 每次打开shell都会执行~/.bashrc

9.拓展资料——内置命令和外置命令

- **内部命令 /bin**
  - Shell自带的命令，例如cd/echo/kill
  - 效率比较高
  - 内置有说明文档，help cd
- **外部命令**
  - 自己写的shell脚本，以及find等命令
- **如何判断?**
  - type cd
  - type find

9.拓展资料——挂起、恢复进程

- **Ctrl+z**
  - 可以将在前台运行的命令放到后台，并暂停

- **jobs**
  - 可以查看当前在后台运行的命令

9.拓展资料——挂起、恢复进程

- **fg %num**
  - 将后台命令调至前台

- **bg %num**
  - 将后台暂停的进程在后台执行

- **kill %num**
  - 终止命令编号为num的进程