VIM 编辑器

1. Vim简介与模式

Vim是一个高效文本编辑器，核心在于**模式切换**：

- **普通模式（Normal Mode）**：移动光标、执行命令（启动时默认模式）。
- **插入模式（Insert Mode）**：输入/编辑文本。
- **命令行模式（Command-line Mode）**：保存、退出、搜索等操作。

---

2. 启动与退出

- **启动Vim**：终端输入 `vim 文件名`（文件不存在则新建）。
- 退出Vim：
  - 普通模式下按 `:` 进入命令行模式。
  - `:q` 退出（未修改时）。
  - `:q!` 强制退出（不保存修改）。
  - `:wq` 保存并退出。

---

3. 模式切换

- 普通模式 → 插入模式：
  - `i`：光标前插入。
  - `a`：光标后插入。
  - `o`：下一行插入。
- **插入模式 → 普通模式**：
  - 按 `Esc` 或 `Ctrl+[`。

---

4. 基础移动（普通模式）

- **方向键**：`h`（左）、`j`（下）、`k`（上）、`l`（右）。
- 单词移动：
  - `w`：跳到下一个单词开头。
  - `b`：跳到上一个单词开头。
- 行内移动：
  - `0`：行首，`^`：第一个非空字符。
  - `$`：行尾。
- 全文移动：
  - `gg`：文件开头。
  - `G`：文件末尾。
  - `50G`：跳转到第50行。

---

5. 编辑文本

- 删除：
  - `x`：删除光标处字符。
  - `dw`：删除一个单词。
  - `dd`：删除整行。
- 撤销与重做：
  - `u`：撤销操作。
  - `Ctrl + r`：重做撤销的操作。
- 复制与粘贴：
  - `yy`：复制当前行。
  - `p`：粘贴到光标后，`P`：粘贴到光标前。

---

6. 查找与替换

- 查找：
  - 普通模式下按 `/`，输入关键词后回车（如 `/hello`）。
  - `n` 跳转到下一个匹配，`N` 上一个。
- 替换：
  - 替换当前行第一个匹配：`:s/old/new`
  - 替换当前行所有匹配：`:s/old/new/g`
  - 全文替换：`:%s/old/new/g`（加 `c` 确认每次替换，如 `:%s/old/new/gc`）。

---

7. 可视模式（Visual Mode）

- `v`：进入字符选择模式（按字符选择）。
- `V`：进入行选择模式（按行选择）。
- `Ctrl + v`：进入块选择模式（垂直选择）。
- **操作**：选中后按 `y` 复制，`d` 删除，`p` 粘贴。

---

8. 配置Vim（.vimrc）

- 创建配置文件：`vim ~/.vimrc`
- 常用配置示例：

  vim

  ```
  set number          " 显示行号
  syntax on           " 语法高亮
  set tabstop=4       " Tab缩进4空格
  set expandtab       " 用空格代替Tab
  ```

VI/VIM基础操作练习

初始文件准备

bash

```
vim /home/student/file.txt  # 创建并打开文件
```

文件初始内容（手动输入）

text

```
apple
banana
orange
grape
pear
```

分步练习任务

1. **打开文件并进入插入模式，在末尾添加一行** `pineapple`

   vim

   ```
   G       # 跳转到文件末尾
   i       # 进入插入模式
   （输入pineapple后按ESC）
   ```

2. **复制第2行（banana）并粘贴到第4行下方**

   vim

   ```
   2G      # 跳转到第2行
   yy      # 复制当前行
   4G      # 跳转到第4行
   p       # 粘贴
   ```

3. **删除第3行（orange）**

   vim

   ```
   3G      # 跳转到第3行
   dd      # 删除当前行
   ```

4. **显示行号并跳转到第5行（原pear行）**

   vim

   ```
   :set nu  # 显示行号
   5G       # 跳转到第5行
   ```

5. **将光标移动到行首，插入`watermelon`**

   vim

   ```
   0       # 跳转到行首
   i       # 进入插入模式
   （输入watermelon后按ESC）
   ```

6. **查找所有`apple`并替换为`mango`（仅替换前2行）**

   vim

   ```
   :1,2s/apple/mango/g  # 直接指定行号范围替换
   ```

7. **撤销最后一次替换操作**

   vim

   ```
   u       # 撤销操作
   ```

8. **强制退出不保存（模拟误操作）**

   vim

   ```
   :q!     # 强制退出
   ```

9. **重新打开文件并另存为`backup.txt`**

   bash

   ```
   vim /home/student/file.txt  # 重新打开
   :w /home/student/backup.txt # 另存为备份
   ```

10. **保存并退出**

    vim

    ```
    :wq     # 保存退出
    ```

最终文件内容验证

text

```
apple
banana
grape
banana
watermelonpear
```

---

1. **基础操作**
   - `i`进入插入模式、`G`跳转行尾
   - `yy`复制、`p`粘贴

- `dd` 删除行

2. **行号与跳转**
   - `:set nu` 显示行号、`nG` 跳转行号（视频中 `58G` 操作）
3. **替换与撤销**
   - `:1,2s/apple/mango/g` 行号范围替换
   - `u` 撤销
4. **文件管理**
   - `:wq` 保存退出、`:q!` 强制退出
   - `:w 新文件名` 另存为

文件类型

1、学习目标

1. 掌握 `ls -l` 命令查看文件类型
2. 理解不同文件类型的符号标识
3. 认识颜色显示与文件类型的关联性
4. 学会准确判断文件类型的方法

2、核心命令：`ls -l`

命令格式

bash

```
ls -l [文件名/目录名]
```

输出示例

bashuser

```
-rw-r--r-- 1 user group 4096 Aug 1 10:00 example.txt
drwxr-xr-x 2 user group 4096 Aug 1 10:01 my_folder
```

3、文件类型符号说明（第一列第一个字符）

符号	文件类型	典型示例	可视化特征
`-`	普通文件	.txt/.sh/.png	黑色文本/蓝色可执行文件
`d`	目录文件	/home /var/log	蓝色文件夹图标
`b`	块设备文件	/dev/sda	特殊设备标识
`c`	字符设备文件	/dev/tty	终端设备标识
`l`	符号链接文件	/usr/bin/python -> python3	浅蓝色带箭头图标

4、操作演示

1. 查看普通文件类型

bash

```
ls -l = ll  /etc/passwd
# 输出示例：-rw-r--r-- 1 root root 1234 Aug 1 10:00 passwd
```

2. 查看目录文件类型

bash

```
ls -ld /var/log  # 注意加-d参数查看目录本身
# 输出示例：drwxr-xr-x 10 root root 4096 Aug 1 09:00 log
```

3. 查看设备文件类型

bash

```
ls -l /dev/sda
# 输出示例：brw-rw---- 1 root disk 8, 0 Aug 1 08:00 sda
```

5、颜色显示的不可靠性

1. **环境差异**：不同Linux发行版默认颜色方案不同
2. **配置影响**：用户可能自定义`LS_COLORS`环境变量
3. **终端限制**：远程SSH连接可能丢失颜色信息
4. **关键原则**：`文件颜色≠类型标识`，必须以`ls -l`输出为准

6、综合练习

练习1：类型判断

bash

```
ls -l /bin/ls /home /dev/tty
```

1. 指出每个文件的类型符号
2. 说明对应的文件类型
3. 验证文件实际类型是否符号显示

用户管理

1.用户/组的基本概念

1.1 概念

系统中的每个进程（运行中的程序）都以特定用户身份运行。每个文件都归属于特定用户所有。对文件和目录的访问权限受用户身份限制。进程关联的用户身份决定了该进程可访问的文件和目录范围。

1.2 用户的作用

- 查看当前登录的用户

```
[root@xnha ~]# id
uid=0(root) gid=0(root) 组=0(root)
```

- 查看文件的owner

```
[root@xnha ~]# ll /home   #ls -l = ll
总用量 8
drwx------. 17 abc   abc    4096 3月   4 20:04 abc
drwx------. 15 test  test   4096 11月  6 03:58 test
drwx------.  3 user1 user1    78 10月 28 21:15 user1
drwx------.  4 user2 user2   129 10月 27 22:43 user2
drwx------.  3 user3 user3    78 10月 27 21:13 user3
drwx------.  3 user4 group1   78 10月 27 21:47 user4
drwx------.  4 user5 user5   113 10月 27 22:23 user5
drwx------.  3 user6 user6    78 10月 27 21:48 user6
```

- 查看运行进程的username：

```
[root@xnha ~]# ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.7 175744 13924 ?        Ss   08:55   0:02
/usr/lib/syst
root           2  0.0  0.0      0     0 ?        S    08:55   0:00
[kthreadd]
root           3  0.0  0.0      0     0 ?        I<   08:55   0:00 [rcu_gp]
root           4  0.0  0.0      0     0 ?        I<   08:55   0:00
[rcu_par_gp]
root           5  0.0  0.0      0     0 ?        I<   08:55   0:00
[slub_flushwq
root           6  0.0  0.0      0     0 ?        I    08:55   0:02
```

1.3 用户组存储的信息

- 用户基本信息文件 /etc/passwd

```
[root@xnha ~]# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:65534:65534:Kernel Overflow User:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
systemd-coredump:x:999:997:systemd Core Dumper:/:/sbin/nologin
systemd-resolve:x:193:193:systemd Resolver:/:/sbin/nologin
tss:x:59:59:Account used for TPM access:/:/sbin/nologin
```

文件结构解析

```
root:x:0:0:root:/root:/bin/bash
用户名:x:uid:gid:描述:HOME:shell
```

字段详细说明

1. **用户名**
   - 登录系统的名字（如 `root`）。
2. **密码占位符**
   - 实际密码存储在 `/etc/shadow`，此处为 `x`。
3. **UID（用户身份证号）**
   - 系统约定（RHEL7）：
     - `0`：特权用户（root）
     - `1~999`：系统用户
     - `1000+`：普通用户
   - Root 用户特性：
     - uid 为 `0`，拥有最高权限。
     - 可覆盖文件系统权限、管理软件和系统文件。
     - 控制大多数硬件设备。
4. **GID（组号）**
   - 默认规则：
     - 每创建一个用户，系统会自动创建同名组。
5. **描述**
   - 用户描述信息（+如职位、部门），默认与用户名一致。
6. **家目录（HOME）**
   - 用户登录后的默认目录（如 `/root`）。
7. **登录 Shell**
   - 用户的命令解释器（如 `/bin/bash`）。
   - 若设置为 `/sbin/nologin`，用户无法登录系统。

- 用户密码信息文件 /etc/shadow

```
root:$6$FQ38PLJfnBOidGC5$a33TV.S8uHAoJ3kDqbaW.m2AJ4x/jpeVgbBVcwJoMmJ0eK.94mm3REO0fLxjBMohzUEPv0LveX7fIkANKB.8i.::0:99999:7:::
```

字段说明（9列，冒号分隔）

1. **登录名**
   - 与 `/etc/passwd` 中的用户名一致（如 `root`）。
2. **加密口令**
   - 格式：

     ```
     $加密算法$salt$加密结果
     ```

     - `$1`：MD5
     - `$5`：SHA-256
     - `$6`：SHA-512
     - `$2`：Blowfish
   - 特殊标记：
     - `*`：账号被锁定（禁止登录）。
     - `!!`：密码已过期。
     - 空：无密码（可直接登录）。
3. **最后一次修改时间**
   - 从 **1970年1月1日**（UNIX纪元）到密码最后一次修改的天数（如 `15636`）。
4. **最小时间间隔**
   - 密码修改后，必须经过该天数才能再次修改（`0` 表示无限制）。
5. **最大有效天数**
   - 密码有效期（如 `99999` 表示永久有效）。
6. **警告天数**
   - 密码到期前 **提前多少天** 发出警告（如 `7` 天）。
7. **不活动天数**
   - 密码过期后，允许用户 **不修改密码** 继续登录的天数（超过则锁定账号）。
8. **失效时间**
   - 账号的绝对过期时间（从 1970年1月1日起的天数，到期后账号禁用）。
9. **保留字段**
   - 未使用，留空。

示例解析

```
root:$1$MYG2NDG6$a1wtyr5GDM2esAPjugOYP0:15636:0:99999:7:: :
```

- **加密算法**：`$1`（MD5）。
- **salt 值**：`MYG2NDG6`。
- **加密结果**：`a1wtyr5GDM2esAPjugOYP0`。
- 密码策略：

- 最后一次修改时间：1970年1月1日后的第 `15636` 天。
- 密码永久有效（`99999` 天）。
- 密码到期前 `7` 天发出警告。

注意事项

- **密码字段为空**：用户无需密码即可登录（高风险，不推荐）。
- **锁定账号**：在密码字段前添加 `*` 或 `!!`。
- **时间计算基准**：不同系统可能以不同时间为起点（如 SCOLinux 使用 1970年1月1日）。

- 组信息文件 /etc/group

  `/etc/group` 文件结构解析

  ```
  root:x:0:
  组名:组密码:组ID:组成员
  ```

  字段说明（4列，冒号分隔）

  1. **组名**
     - 组的名称（如 `root`）。
  2. **组密码**
     - 一般为 `x`，实际组密码存储在 `/etc/shadow`（通常不设置）。
  3. **组ID（GID）**
     - 组的唯一标识符（如 `0`）。
  4. **组成员**
     - 属于该组的用户列表（多个用户用逗号分隔）。
     - **默认值**：空（仅包含同名用户，但默认不显示）。

  示例解析

  ```
  root:x:0:
  ```

  - **组名**：`root`
  - **组密码**：`x`（无实际密码）。
  - **组ID**：`0`（特权组）。
  - **组成员**：空（未显式添加用户时仅包含同名用户 `root`）。

  注意事项

  - **基本组**：用户创建时自动生成同名组（如用户 `user01` 的组 `user01`）。
  - **附加组**：用户可手动加入其他组（如将用户加入 `hr` 组）。

用户/组的管理

用户管理

1. 用户创建

1.1 默认创建（自动生成同名基本组）

bash

```
[root@localhost ~]# useradd user01
[root@localhost ~]# id user01
uid=1001(user01) gid=1001(user01) 组=1001(user01)
```

- 说明：
  - 用户编号（UID）：唯一用户标识（如 `1001`）。
  - 组编号（GID）：同名基本组的标识（如 `1001`）。
  - **规则**：未指定选项时，系统自动创建同名组作为用户的主组（Primary Group）。

1.2 指定选项创建

bash

```
# 指定 UID
[root@localhost ~]# useradd user02 -u 1503
# 制定备注
[root@xnha ~]# tail -3 /etc/passwd
user1:x:1001:1001::/home/user1:/bin/bash
user2:x:1503:1503::/home/user2:/bin/bash
user3:x:1504:1504:第三个用户:/home/user3:/bin/bash
# 指定家目录
[root@localhost ~]# useradd user03 -d /user03    /home/user03
```

- 验证：

  bash

  ```
  grep user03 /etc/passwd  # 查看家目录是否修改
  ```

---

2. 用户删除

bash

```
[root@localhost ~]# userdel -r user02  # 彻底删除用户及家目录
```

- 注意事项：
  - `-r` 选项会删除用户家目录和邮箱文件（`/var/spool/mail`）。
  - 未加 `-r` 会导致残留文件，可能影响后续创建同名用户。

---

3. 用户密码管理

方法一：root 修改其他用户密码

bash

```
[root@localhost ~]# passwd alice  # 输入两次新密码
```

方法二：用户自行修改密码

bash

```
[zhuzhu@localhost ~]passwd  # 需输入原密码
```

---

4. 用户属性修改

修改登录 Shell

bash

```
[root@localhost ~]# usermod -s /sbin/nologin user02  # 禁用用户登录
```

- 验证：

  bash

  ```
  grep user02 /etc/passwd  # 查看登录Shell字段是否为/sbin/nologin
  ```

---

5. 组成员管理

将用户追加到附加组

bash

```
# 语法
usermod -aG 组名 用户名
wheel
usermod -aG wheel user02
# 示例
[root@localhost ~]# usermod -aG hr user02

# 验证
[root@localhost ~]# id user02
uid=1002(user02) gid=1002(user02) 组=1002(user02),1005(hr)
```

从组中移除用户

bash

```
# 语法
gpasswd -d 用户名 组名

# 示例
[root@localhost ~]# gpasswd -d user02 hr
```

用户组管理

---

1. 组基础操作

创建组

bash

```
# 默认创建
[root@localhost ~]# groupadd hr

# 指定 GID
[root@localhost ~]# groupadd net01 -g 1007

# 验证
[root@localhost ~]# grep 'net01' /etc/group
net01:x:1007:
```

删除组

bash

```
[root@localhost ~]# groupdel user01
                    userdel -r user01
```

---

2. 组分类

1. **基本组**
   - 随用户自动创建的同名组（如 `user01` 组）。
   - 用户文件默认属组为基本组。
2. **附加组**
   - 用户手动加入的其他组（如 `hr` 组）。
   - 用于扩展权限管理。

---

3. 组成员管理示例

1. **创建共享目录**

   bash

   ```
   mkdir /shared
   chgrp hr /shared  # 设置属组为hr
   chmod 770 /shared  # 赋予组读写执行权限
   ```

2. **验证权限**

   bash

   ```
   su - user02
   touch /shared/test.txt  # 成功则权限生效
   ```

注意事项

1. 修改 UID/GID 前备份：

   bash

   ```
   find / -user 旧UID -exec chown 新UID {} \;  # 批量修改文件所有者
   ```

2. 组权限生效条件：
   - 用户需重新登录或使用 `newgrp` 命令。
3. **删除用户必用** `-r`：避免残留文件导致后续操作失败。

Linux 用户与组管理综合实验

实验目标

1. 掌握用户和组的创建、删除及属性修改。
2. 理解用户配置文件（`/etc/passwd`、`/etc/shadow`、`/etc/group`）的结构。
3. 学会设置用户密码、组成员权限及共享目录访问控制。

实验步骤

---

1. 创建用户与验证基本信息

1. **默认创建用户** `user01`：

   bash

   ```
   useradd user01
   id user01  # 验证 UID/GID 和同名基本组
   tail -n 1 /etc/passwd  # 查看用户信息
   ```

2. **指定选项创建用户**：
   - 创建用户`user02`，指定 UID 为 `1503`：

     bash

     ```
     useradd user02 -u 1503
     id user02  # 验证 UID 是否生效
     ```

   - 创建用户 `user03`，指定家目录为 `/user03`：

     bash

```
useradd user03 -d /user03
ls -ld /user03  # 验证家目录是否存在
```

---

2. 用户密码管理

1. **Root 为用户 `user01` 设置密码**：

   bash

   ```
   passwd user01  # 输入两次密码
   grep user01 /etc/shadow  # 查看加密后的密码字段
   ```

2. **用户 `user01` 自行修改密码**：

   bash

   ```
   su - user01  # 切换到 user01
   passwd       # 输入原密码后修改新密码
   exit         # 返回 root
   ```

---

3. 修改用户属性

1. **禁用用户 `user02` 的登录 Shell**：

   bash

   ```
   usermod -s /sbin/nologin user02
   grep user02 /etc/passwd  # 验证登录 Shell 是否修改
   ```

2. **尝试登录 `user02`（应失败）**：

   bash

   ```
   su - user02  # 预期提示“此账户不可用”
   ```

---

4. 用户组管理

1. **创建组 `hr` 和 `dev`**：

   bash

   ```
   groupadd hr
   groupadd dev -g 2000  # 指定 GID 为 2000
   tail -n 2 /etc/group  # 查看组信息
   ```

2. **将用户加入附加组**：

   bash

```
usermod -aG hr user01  # 将 user01 加入 hr 组
usermod -aG dev user02  # 将 user02 加入 dev 组
id user01  # 验证组成员（应显示 hr 组）
id user02  # 验证组成员（应显示 dev 组）
```

3. **从组中移除用户**：

   bash

```
gpasswd -d user01 hr  # 将 user01 移出 hr 组
id user01  # 验证组成员是否移除
```

---

5. 共享目录权限验证

1. **创建共享目录并设置权限**：

   bash

```
mkdir /shared
chgrp dev /shared   # 设置属组为 dev
chmod 770 /shared   # 赋予组读写执行权限
ls -ld /shared      # 验证权限
```

2. **验证组成员权限**：
   - 用户 `user02`（属于 `dev` 组）：

     bash

```
su - user02
touch /shared/test.txt  # 应成功创建文件
exit
```

   - 用户 `user01`（不在 `dev` 组）：

     bash

```
su - user01
touch /shared/test.txt  # 应提示“权限被拒绝”
exit
```

---

6. 清理实验环境

1. **删除用户及其家目录**：

   bash

```
userdel -r user01
userdel -r user02
userdel -r user03
```

2. **删除组**：

   bash

```
groupdel hr
groupdel dev
```

3. **删除共享目录**:

   bash

   ```
   rm -rf /shared
   ```

实验总结

核心知识点

1. **用户管理**:
   - `useradd`、`usermod`、`userdel` 命令的使用。
   - 用户密码策略通过 `/etc/shadow` 控制。
   - 禁用登录 Shell 可限制用户访问。
2. **组管理**:
   - `groupadd`、`groupdel`、`gpasswd` 命令的使用。
   - 基本组（自动创建）与附加组（手动加入）的区别。
   - 通过组权限实现共享目录的访问控制。
3. **权限验证**:
   - 使用 `id` 查看用户组关系。
   - 通过 `chgrp` 和 `chmod` 设置目录权限。