文本编辑器及文本处理

HUAWEI

前言

- 文本处理是操作系统对文件管理的基础操作，文本编辑器是计算机软件中的一种，主要用来编写和查看文本文件。不同的文件编辑器有不同的辅助功能。本章将介绍几种常见的文本编辑器以及文本处理基础操作。

目标

- 学完本课程后，您将能够：
  - 了解Linux常见的文本编辑器
  - 熟悉vi与vim的三种主要模式
  - 掌握vim文本编辑器的常用操作
  - 熟悉vim文本编辑器的快捷操作

目录

1. **Linux常用文本编辑器介绍**
2. 使用vim编辑器
3. 文本处理

Linux文本编辑器介绍

- 文本编辑器是操作系统基础的功能软件之一。根据使用环境的不同，Linux的文本编辑器有很多类型。
- 常见的Linux文本编辑器有：
  - emacs
  - nano
  - gedit
  - kedit
  - vi
  - vim

Linux文本编辑器 - emacs

- emacs是一款功能强大的编辑器，与其说是一款编辑器，它更像一个操作系统。emacs带有内置的网络浏览器、IRC客户端、计算器，甚至是俄罗斯方块。当然，emacs需要在图形化界面的Linux中使用。
- 优点：
  - 可定制，可扩展
  - 功能强大
  - 可以与许多自由软件编程工具集成
- 缺点：
  - 入门难度高，对普通用户不友好

Linux文本编辑器 - nano

- nano是命令行界面下一个相对简单的文本编辑器，它是为了代替闭源的Pico文本编辑器而开发的，1999年以GPL协议发布第一个版本，是一个自由软件，同时也是GNU计划的一个组成部分。nano有很多人性化的功能设计，如语法高亮、正则表达式搜索和替换、平滑滚动、多个缓冲区、自定义快捷键、撤销或重复编辑。
- 优点：
  - 易于使用，操作简单，适用于简单文本编辑。
- 缺点：
  - 对复杂的文本编辑比较耗时，无强大的命令功能进行复杂操作，不支持如宏、一次编辑多个文件、窗口分割、垂直块/矩形选择/编辑、自动完成等高级功能。

Linux文本编辑器 - gedit

- gedit是一个GNOME桌面环境下兼容UTF-8的文本编辑器。它简单易用，有良好的语法高亮，对中文支持很好，支持包括GB2312、GBK在内的多种字符编码。gedit是一款自由软件。gedit包含语法高亮和标签编辑多个文件的功能。利用GNOME VFS库，它还可以编辑远程文件。它支持完整的恢复和重做系统以及查找和替换。
- 优点：
  - 图形化界面，易上手，操作习惯与Windows类似，包括常用的快捷键如复制粘贴等。
- 缺点：
  - 需要安装图形化桌面才能使用。

Linux文本编辑器 - kedit

- 与gedit类似，kedit是KDE图形化桌面中常用的一种文本编辑器。kedit是一个非常小的编辑器，特别适用于浏览文本和各种配置文件。
- 优点：
  - 图形化界面，易上手，操作习惯与Windows类似，包括常用的快捷键如复制粘贴等。
- 缺点：
  - 需要安装图形化桌面才能使用。

Linux文本编辑器 - vi

- vi是标准的Unix文本编辑器，也是最古老的文本编辑器、最通用的文本编辑器。所有的Linux、Unix都默认带有vi文本编辑器。虽然vi的操作方式与其他常用的文本编辑器（如gedit）很不相同 ，但是由于其运行于字符界面，并可用于所有unix/linux环境，仍被经常使用。
- vi的三种命令模式：
  - Command：命令模式，用于输入命令；
  - Insert：插入模式，用于插入文本；
  - Visual：可视模式，用于浏览文本。
- 优点：通用，几乎所有的Unix、Linux都自带vi。
- 缺点：功能简单，显示效果单一。

Linux文本编辑器 - vim

- Vim是从vi发展出来的一个文本编辑器。其代码补完、编译及错误跳转等方便编程的功能特别丰富，在程序员中被广泛使用。和Emacs并列成为类Unix系统用户最喜欢的编辑器。
- Vim的第一个版本由布莱姆·米勒在1991年发布。最初的简称是Vi IMitation，随着功能的不断增加，正式名称改成了Vi IMproved。现在是在开放源代码方式下发行的自由软件。
- 从vi派生出来的Vim具有多种模式：
  - 基本模式：普通模式、插入模式、可视模式、选择模式、命令行模式、Ex模式
  - 派生模式：操作符等待模式、插入普通模式、插入可视模式、插入选择模式、替换模式
  - 其他：Evim
- openEuler 20.03 LTS系统安装后默认没有安装vim，需要手动安装vim

目录

1. Linux常用文本编辑器介绍
2. **使用vim编辑器**
3. 文本处理

vim编辑器命令格式

- 命令格式：

vim [options] [file]...	编辑指定文件
vim [options] -	从标准输入 (stdin) 读取文本
vim [options] -t tag	编辑tag定义出的文件
vim [options] -q [errorfile]	编辑第一个出错处的文件

- 常见参数：
  - -c： 打开文件前线执行指定的命令
  - -R： 以只读方式打开，但是可以强制保存
  - -M： 以只读方式打开，不可以强制保存
  - -r： 回复崩溃的会话
  - +*num*： 从第*num*行开始

vim基础操作 - 打开文件

```
[root@openEuler ~]# vim  filename
```

- 如果*filename* 文件存在，则会打开文件并显示文件内容；
- 如果*filename* 文件不存在，vim会在下面提示 [New File]，并且会在第一次保存时创建该文件。

```
[root@openEuler ~]# vim  test.txt
~
~
~
~
~
“test.txt” [New File]
```

vim基础操作 - 移动光标

- 快速移动光标：
  - 上下左右键或k、j、h、l键上下左右移动光标
  - 0 移动到行首
  - g0 移到光标所在屏幕行行首
  - :n 移动到第n行。
  - gg: 到文件头部。
  - G: 到文件尾部。
- 数据操作：
  - yy or Y: 复制整行文本。
  - y[n]w: 复制一(n)个词。
  - d[n]w: 删除（剪切）1(n)个单词
  - [n] dd: 删除（剪切）1(n)行。

vim基础操作 - 数据操作

- 复制：
  - yy or Y: 复制整行文本。
  - y[n]w: 复制一(n)个词。
- 粘贴：
  - 面向行的数据：
    - p 放置数据在当前行的下面
    - P 放置数据在当前行的上面
  - 面向字符的数据：
    - p 放置数据在光标的后面
    - P 放置数据在光标前
- 删除：
  - d[n]w: 删除（剪切）1(n)个单词
  - [n] dd: 删除（剪切）1(n)行。

vim基础操作 - 行号显示与取消

- 显示行号
  - :set nu

```
   1  hello
   2  openEuler
~
~
~
~
~
:set nu
```

- 取消显示行号
  - :set nonu

vim基础操作 - 查找与替换

- 查找
  - :/*word*　在光标之后查找一个字符串word，按n向后继续搜索，shift+n向上搜索。
  - :?*word*　在光标之前查找一个字符串word，按n向后继续搜索。
- 替换
  - :1,5s/*word1*/*word2*/g　将文档中1-5行的word1替换为word2，不加g则只替换每行的第一个word1。
  - %s/*word1*/*word2*/gi　将文档所有的word1替换为word2，不区分大小写。

vim基础操作 - 设置搜索高亮

- 临时设置时，在命令模式下输入：
  - :set hlsearch

```
hell
openEuler
hell
world
~
~
:set nu
```

- 永久设置，需要在/etc/vimrc中配置，增加一行set hlsearch，然后更新变量即可。

vim基础操作 - 修改文件

- 使用vim filename打开文件后，进入的是普通模式。当想要修改文件时，可以按i键进入插入模式。进入插入模式时，会在最下面提示当前模式是Insert。按ecs可以退出插入模式，回到普通模式。

```
[root@openEuler ~]# vim  test.txt
#按i进入插入模式
~
~
~
~
~
~
~
-- INSERT --
```

vim基础操作 - 撤销或重做

- u 撤销最近的改变
- U 撤销当前行自从光标定位在上面开始的所有改变
- Ctrl+r 重做最后一次“撤销”改变

vim基础操作 - 保存文件并退出

- 退出插入模式：
  - 在插入模式下按ecs键退出插入模式
- 常用的保存/退出的命令：
  - :w　保存
  - :q　退出
  - :wq　保存并退出
  - :q!　强制退出
  - :wq!　强制保存并退出

目录

1. Linux常用文本编辑器介绍
2. 使用vim编辑器
3. **文本处理**

查看文件 - cat (1)

- cat 是一个文本文件查看和连接工具。cat有如下功能：
  - 显示文件内容，cat filename
  - 编辑一个文件，cat > filename。
  - 将几个文件合并为一个文件，cat file1 file2 > file3
- cat常用选项有：
  - -n：从1开始对所有行编号并显示在每行开头
  - -b：从1开始对非空行编号并显示在每行开头
  - -s：当有多个空行在一起时只输出一个空行
  - -E：在每行结尾增加$
  - --help：显示帮助信息

查看文件 - cat (2)

- cat查看文件用法举例

```
[root@openEuler ~]# cat /etc/profile                    #查看/etc/profile文件内容
[root@openEuler ~]# cat -b /etc/profile   #查看/etc/profile文件内容，并对非空白行进行编号，编号从1开始
[root@openEuler ~]# cat -n /etc/profile   #查看/etc/profile文件内容，并在每行前面显示行号。
[root@openEuler ~]# cat -E /etc/profile    #查看/etc/profile文件内容，并且在每行的结尾处附加$符号
[root@openEuler ~]# cat -s /etc/profile                  #查看/etc/profile文件内容，但是不输出多行空行，当有多个空行在一起时，只输出一个空行
```

查看文件 - more (1)

- more可以一次查看文件或者标准输入的一页，与cat不同的是more可以按页来查看文件的内容，还支持直接跳转行等功能。
- 命令格式：more [options] <file>...
- more 常用的选项有：
  - +n：从第n行开始显示
  - -n：定义屏幕大小为n行
  - -c：从顶部清屏，然后显示
  - -s：把连续的多个空行显示为一行

查看文件 - more (2)

- more的常用操作：
  - Enter：默认向下滚动1行
  - Ctrl+F：向下滚动一屏
  - 空格键：向下滚动一屏
  - Ctrl+B：向上滚动一屏
  - b：向上滚动一屏
  - =：输出当前行号
  - :f ： 输出文件名和当前行号
  - q：退出more

查看文件 - less (1)

- less 可以一次查看文件或者标准输入的一页，less 的用法比起 more 更加的有弹性。
- 命令格式： less [option] 文件
- less常用的选项有：
  - -f：强制打开特殊文件，例如外围设备代号、目录和二进制文件
  - -g：只标志最后搜索到的关键字
  - -i：忽略搜索时的大小写
  - -N：显示每行的行号
  - -s：当有多个空行在一起时只输出一个空行
  - -o <文件名> ：将less输出的内容保存到指定文件

查看文件 - less (2)

- less常用的操作：
  - b：向上翻一页
  - d：向下翻半页
  - h：显示帮助界面
  - q：退出less
  - u：向上翻半页
  - y：向上翻一行
  - 空格键：向下翻一行
  - Enter：向下翻一页
  - 上下键：向上/下翻一行

文件摘选 - head

- head用来显示文件的开头至标准输出中，默认head命令可以显示文件的前10行
- 命令格式：head [option]... [文件]...
- head常用的选项有：
  - -q：输出时隐藏文件名，head默认不显示文件名
  - -v：输出时显示文件名
  - -c *num*：显示前*num* 个字节
  - -n *num*：显示前*num* 行

文件摘选 - tail

- tail用来显示文件的末尾至标准输出中，默认tail命令可以显示文件的后10行
- 命令格式：tail [option]... [file]...
- tail常用的选项有：
  - -f：循环读取，对于日志文件的监控非常有用
  - -q：不显示文件名，tail默认不显示文件名
  - -v显示文件名
  - -c *num*：显示文件最后*num* 个字节
  - -n *num*：显示文件最后*num* 行

提取列或字段 - cut

- cut用于显示文件或者标准输入的特定列，如：

```
[root@openEuler ~]# cut  -d: -f1 /etc/passwd                    #显示/etc/passwd文件以 : 间隔的第一列
```

- 命令格式：cut [option]... [文件]
- cut常用的选项有：
  - -b [*范围*] ： 仅显示行中指定直接范围的内容
  - -c[*范围*] ： 仅显示行中指定范围的字符
  - -d：指定字段的分隔符，默认的字段分隔符为“TAB”
  - -f [*范围*]：显示指定第*num*个字段的内容，可以用逗号隔开显示多个字段

提取列或字段 - awk

- awk是一个强大的文本分析工具，简单来说awk就是把文件或者标准输入逐行读入，以空格为默认分隔符将每行切片，切开的部分再进行各种分析处理。

```
[root@openEuler ~]# last -n 5 | awk '{print $1}'      #显示最近登录系统的5个账号
```

- awk基本命令格式：awk 动作 文件名，如：awk '{print $0}' test.txt

提取关键字 - grep

- grep命令是一种强大的文本搜索工具，它能使用正则表达式搜索文本，并把匹配的行打印出来。grep在一个或多个文件中搜索字符串模板。如果模板包括空格，则必须被引用，模板后的所有字符串被看作文件名。搜索的结果被送到标准输出，不影响原文件内容。
- 命令格式：grep [option] [file]...
- grep常用的选项有：
  - -c：统计符合样式的行数
  - -i：忽略大小写
  - -w：只显示全子符合的行
  - -x：只显示全行符合的行

文本统计 - wc

- wc命令用于计算字数。利用wc指令我们可以计算文件的字节数、字数、或是列数，若不指定文件名称、或是所给予的文件名为"-"，则wc指令会从标准输入设备读取数据。
- 命令格式：wc [option]... [file]...
- wc常用选项有：
  - -c或--bytes或--chars：只显示字节数
  - -l或--lines：只显示行数
  - -w或--words：只显示字数

文本排序 - sort

- sort命令可以将文件进行排序，并将排序结果标准输出。sort命令既可以从特定的文件，也可以从stdin中获取输入。
- 命令格式：sort [option]... [file]...
- sort常用的选项有：
  - -b：忽略每行前面开始的空格字符
  - -c：检查文件是否已经按照顺序排序
  - -d：排序时，处理英文字母、数字及空格字符外，忽略其他字符
  - -f：排序时，将小写字母视为大写字母
  - -n：依照数值的大小排序
  - -r：以相反的顺序排序
  - -o <文件>：将排序后的结果存入指定的文件
  - -u：忽略相同行

文本比较 - diff

- diff以逐行的方式，比较文本文件的异同处。如果指定要比较目录，则diff会比较目录中相同文件名的文件，但不会比较其中子目录。
- 命令格式：diff [option]... file
- diff常用的选项有：
  - -B：不检查空白行
  - -c：显示全部内文，并标出不同之处
  - -i：忽略大小写的不同
  - -r：比较子目录中的文件
  - -w：忽略全部的空格字符

文本操作工具 - tr

- tr 指令从标准输入设备读取数据，经过字符串转译后，将结果输出到标准输出设备，常用于转换或删除文件中的字符。
- 命令格式：tr [option]... set1 [set2]

```
[root@openEuler ~]# cat text.txt | tr a-z A-Z        #将小写转换为大写输出
```

- tr常用的选项有：
  - -c：反选设定字符，也就是符合set1的部分不做处理，不符合的剩余部分才进行转换
  - -d：删除字符
  - -s缩减连续重复的字符成指定的单个字符
  - -t：削减set1指定范围，使之与set2设定长度相等

文本操作工具 - sed

- 相比较tr，sed可以修改字符串。sed是一种在线编辑器，可以对来自文件、以及标准输入的文本进行编辑。执行时，sed会从文件或者标准输入中读取一行，将其复制到缓冲区，对文本编辑完成之后，读取下一行直到所有的文本行都编辑完毕。所以sed命令处理时只会改变缓冲区中文本的副本，如果想要直接编辑原文件，可以使用-i选项或者将结果重定向到新的文件中。
- 命令格式：sed [option]... [option] {script-only-if-no-other-script} [input-file]...
- sed常用的选项：
  - -n：取消默认输出
  - -e：多点编辑，可以执行多个子命令
  - -f：从脚本文件中读取命令
  - -i：直接编辑原文件
  - -l：指定行的长度
  - -r：在脚本中使用扩展表达式

本章总结

- 本章主要介绍了Linux操作系统的发展，以及openEuler操作系统简介、安装和快捷操作。

更多信息

- 更多Linux快捷键信息请参考：https://linuxtoy.org/archives/bash-shortcuts.html

Thank you.

把数字世界带入每个人、每个家庭、
每个组织，构建万物互联的智能世界。

Bring digital to every person, home, and organization for a fully connected, intelligent world.

HUAWEI