文本处理
（作者：程序员良许）

文件内容处理

- wc
  - 作用：统计文本信息
  - 语法：wc [OPTIONS]... [FILE]...
  - 参数
    - -w 统计字数。或 --words，只显示字数。一个字被定义为由空白、跳格或换行字符分隔的字符串
    - -c 统计字节数。或 --bytes或 --chars：只显示Bytes数
    - -l 统计行数。或 --lines：只显示行数
    - -m 统计字符数
    - -L 打印最长行的长度（不包含不可见字符）
  - 案例
    - wc file.txt 统计 file.txt 文件的行数、字数、以及字节数
    - wc -w file.txt 统计 file.txt 文件的字数
    - wc -m file.txt 统计 file.txt 文件的字符数
    - wc -c file.txt 统计 file.txt 文件的字节数
    - wc -l file.txt 统计 file.txt 文件的行数
    - wc -L file.txt 打印 file.txt 文件最长行的长度
    - cat file.txt | wc -l 使用管道统计文本行数
- split
  - 作用：文件分割
  - 语法：split [OPTION]... [FILE [PREFIX]]
  - 参数
    - 指定每多少行切成一个小文件
      - -l行数
      - -l行数
    - 指定每多少字节切成一个小文件
      - -b字节
      - -b字节
    - -d 使用数字作为后缀
    - -a 指定后缀长度（默认为2）
  - 案例
    - 假如文件名 split 会覆写前面文件的前缀文件名，split会自动在前面文件名后再加上编号
    - split -l 2 file1.txt 将file1.txt每2行切割成一个小文件
    - split -b 10k file2.txt 将file2.txt每10kb切割成一个小文件
    - split -b 10k -d -a 3 file2.txt 以数字作为后缀，并指定后缀宽度为3
    - split -b 10k -d -a 3 file2.txt split_file 为分割后的文件指定文件名的前缀
- cut
  - 作用：从文件中提取文本的一部分
  - 语法：cut OPTION... [FILE]...
  - 参数
    - -b 以字节为单位进行分割. 仅显示行中指定直接范围的内容
    - -c 以字符为单位进行分割. 仅显示行中指定范围的字符
    - -d 自定义分隔符，默认为制表符"TAB"
    - -f 显示指定字段的内容. 与 -d一起使用
    - -n 取消分割多字节字符
    - --complement 补足被选择的字节、字符或字段
    - --out-delimiter 指定输出内容里的字段分割符
  - 案例
    - cut -f 2 file1.txt 提取file1.txt第2列的内容
    - cut -f2 file1.txt 提取file1.txt的第2列
    - cut -f2 --complement file1.txt 提取file1.txt除第2列外的其它内容
    - cut -f2 -d" file2.txt 使用 -d 选项指定字段分隔符
    - cut -b 2-4,6 file3.txt 提取第2，3，4，6个字节
    - cut -c1-3 file3.txt 提取第1个到第3个字符
    - cut -c2 -f file3.txt 提取第2个字符
    - cut -c4- file3.txt 提取第4个之后的字符
- paste
  - 作用：合并两个或多个文件
  - 语法：paste [OPTION]... [FILE]...
  - 参数
    - -d 默认情况分隔符是空格或tab键，设置新的域分隔符
    - -s 将每个文件粘贴成一行
    - - 从标准输入中读取数据
  - 案例
    - paste file1.txt file2.txt 将file1.txt和file2.txt粘贴成一个新的文件
    - paste file2.txt file1.txt 顺序不一样，结果不一样
    - paste file1.txt file2.txt file3.txt 多文件拼接
    - paste -d" -s file1.txt file2.txt 设置域分隔符为，粘贴成新的文件
    - paste -d" - -s file1.txt file2.txt 将每个文件粘贴成一行
    - ls | paste -d" " - - - - 从标准输入中读取数据，每行显示4个文件名
- sort
  - 作用：对文件内容进行排序
  - 语法：sort [参数] 文件
  - 参数
    - -n 依照数值的大小排序
    - -t <分隔字符> 指定排序时所用的栏位分隔字符
    - -k 指定需要排序的栏位
    - -r 以相反的顺序来排序
  - 案例
    - sort file1.txt 按字母顺序排序
    - sort -r file1.txt 反向排序
    - sort -n file2.txt 按数字大小排序
    - sort -t : -k 3 -n file3.txt 以冒号 为间隔符，对指定的文件内容按照数字大小的第3列进行排序
- uniq
  - 作用：去除文件中的重复行
  - 语法：uniq [OPTIONS]... [INPUT [OUTPUT]]
  - 参数
    - -c 打印每行在文本中重复出现的次数
    - -d 只显示有重复的记录，每个重复记录只出现一次
    - -u 只显示没有重复的记录
  - 案例
    - uniq file.txt 删除连续文件中连续的重复行
    - uniq -c file.txt 打印每行在文件中出现重复的次数
    - uniq -d file.txt 只显示有重复的记录，且每个记录只出现一次
    - uniq -u file.txt 只显示没有重复的记录
- diff/patch
  - 作用：比较文件的差异
  - 语法：diff [OPTIONS]... FILES
  - 参数
    - y 以并列的方式显示文件的异同之处
    - c 显示全部内文，并标出不同之处
    - u 以合并的方式来显示文件内容的不同
    - -W 设置宽度
  - 显示提示
    - a - add
    - c - change
    - d - delete
    - | 前面2个文件内容有不同
    - < 后面文件比前面文件多了行的内容
    - > 后面文件比前面文件多了行的内容
    - + 比较的文件的前者比前者多一行
    - - 比较的文件的前者比前者少一行
    - ! 比较的文件两者有差别的行
  - 案例
    - diff file1.txt file2.txt 比较两个文件
    - diff -y -W 50 file1.txt file2.txt 并排格式输出
    - diff -c file1.txt file2.txt 上下文格式输出
    - diff -u file1.txt file2.txt 统一格式输出
    - diff file1.txt file2.txt > file.patch 生成补丁
    - patch file1.txt file.patch 打补丁
- join
  - 作用：连接两个文件
  - 语法：join [OPTIONS]... FILE1 FILE2
  - 参数
    - -a1或-a2 除了显示共同域的记录之外，-a1显示第一个文件没有共同域的记录，-a2显示第二个文件没有共同域的记录
    - -o 设置结果显示的格式
    - -t 改变域的分隔符
    - -v1或-v2 不显示共同域的记录之外，-v1显示第一个文件没有共同域的记录，-v2显示第二个文件中没有共同域的记录
    - -j 指定一个域作为匹配字段
  - 案例
    - join file1.txt file2.txt 默认以第一列作为连接字段
    - join -a1 file1.txt file2.txt 显示左边文件中的所有记录（右边文件中没有匹配的不显示）
    - join -a2 file1.txt file2.txt 显示右边文件中的所有记录（左边文件中没有匹配的不显示）
    - join -a1 -a2 file1.txt file2.txt 全连接（显示左边和右边所有记录）
    - join -o 1.2 1.3 2.3 file1.txt file2.txt 指定输出字段
    - join -t ":" file1.txt file2.txt 指定分隔符
    - join -v 1 file1.txt file2.txt 只显示第1个文件中没有相同校验的行
    - join -j 2 file1.txt file2.txt 指定了以两个文件要第2列做匹配字段
- tr
  - 作用：转换或删除文件中的字符
  - 语法：tr [OPTIONS]... SET1 [SET2]
  - 参数
    - -c 反选设定字符。也就是符合 SET1 的部份不做处理，不符合的剩余部份才进行转换
    - -d 删除字集合出现的所有字符
    - -s 缩减连续重复的字符成指定的单个字符
    - 可以使用的字符类
  - 案例
    - 大小写转换
      - tr "a-z" "A-Z" "a-z" "A-Z" = file.txt
      - cat file.txt | tr a-z A-Z
      - echo "hello world" | tr [:lower:] [:upper:]
      - echo "Hello World!" | tr "[A-Z]" "[a-z]" （大小写互换）
      - echo 'abcAbdefCD' | tr -d "a-z" 删除小写字母
      - echo "abc123def6ddhello" | tr -d 0-9 删除数字
      - echo " Hello World ! " | tr -d " [:alpha:]" 删除空格和alpha字
      - echo -e "abc123\n def6ddd" | tr -d -c '0-9 \n' 删除不在指定集合中的字符
      - echo -e "fjrfoiZjrfjrfoiZr" | tr -s 'fr' 压缩重复的字符
      - echo "Helloooo Linuxxxxxx" | tr -s "l eo" 删除字符集中的重复字符
      - echo "2022 05 16" |tr -s " " " " 将多个连续空格合并为一个空格，并将空格替换为破折号
      - echo -e "hello\tworld" | tr "\t" " " 将制表符转换为空格
- sed
  - 作用：批量编辑文本文件
  - 语法：sed [选项] [动作] 文件
  - 参数
    - -n 仅显示script处理后的结果
    - -e 以选项中指定的script来处理输入的文本文件
    - -f 此选项会直接修改源文件，要慎用
  - 动作
    - a：新增
    - c：取代
    - d：删除
    - i：插入
    - p：打印
    - s：取代
  - 案例
    - 输出文件内容
      - sed -n '2p' file.txt 仅输出第2行
      - sed -n '3,5p' file.txt 输出3-5行
    - 删除文件内容
      - sed -n '/an/p' file.txt 搜索含有an关键字的行
      - sed '2,4d' file.txt 删除第2-4的数据
      - sed '/an/d' file.txt 删除含有an关键字的行
      - sed '2a nice' file.txt 在第2行后追加nice
      - sed '2i 123456789' file.txt 在第2行插入一行数据
    - 替换数据
      - sed '2c aaghjh' file.txt 替换第2行
      - sed '2,5c data changed' file.txt
      - sed 's/liangxu/alvin/g' file.txt 全局替换，把liangxu改为alvin
      - sed 3s/liangxu/alvin/g file.txt 只替换第3行
      - sed -e 's/liangxu/jg/g;s/alvin/jj/g' file.txt 多个条件
      - sed -i 3s/liangxu/alvin/g' file.txt 将操作写入文件
- awk
  - 作用：文本和数据进行处理的编程语言
  - 语法：awk 条件1 {动作1} 条件2 {动作2} / file.txt
  - 参数
    - -F 指定输入时用到的字段分隔符
    - -v 自定义变量
    - -f 从脚本中读取awk命令
    - -m 对val最设置内存限制
  - 内置变量
    - ARGC 命令行参数个数
    - ARGV 命令行参数数列
    - ENVIRON 支持队列中系统环境变量的使用
    - FILENAME awk浏览的文件名
    - FNR 浏览文件的记录数
    - FS 设置输入域分隔符，等价于命令行 -F选项
    - NF 浏览记录的域的个数
    - NR 已读的记录数
    - OFS 输出域分隔符
    - ORS 输出记录分隔符
    - RS 控制记录分隔符
  - 案例
    - df -h | awk '{print $1 "\t" $3}' 显示第1和第3列的内容
    - awk -F : '{print $1 "\t" $3}' /etc/passwd 指定冒号为分隔符，显示第1和第3列的内容
    - awk -F : '$3==500' /etc/passwd 指定冒号为分隔符，显示系统中所有UID等于500的用户信息（第3列）
    - awk -F: /'liangxu'/ /etc/passwd 搜索/etc/passwd有liangxu关键字的所有行
    - awk -F: /'liangxu'/{print $7} /etc/passwd 搜索/etc/passwd有liangxu关键字的所有行，并显示对应的shell
    - head /etc/passwd | awk -F : 'BEGIN{print " name\tuid"}{print $1 "\t" $3}END{print " from the /etc/passwd"}' BEGIN到END的使用

查找文本内容

- grep
  - 作用：文本搜索工具
  - 语法：grep [参数] 文件
  - 参数
    - -i 忽略大小写
    - -c 只输出匹配行的数量
    - -l 只列出符合匹配的文件名，不列出具体的匹配行
    - -n 列出所有的匹配行，显示行号
    - -h 查询多文件时不显示文件名
    - -s 不显示不存在、没有匹配文本的错误信息
    - -v 显示不包含匹配文本的所有行
    - -w 匹配整词
    - -x 匹配整行
    - -r 递归搜索
    - -q 禁止输出任何结果，已退出状态表示搜索是否成功
    - -b 打印匹配行前文件头部的偏移量，以字节为单位
    - -o 与 -b结合使用，打印匹配的词语文件头部的偏移量，以字节为单位
    - -F 匹配固定字符串的内容
    - -E 支持扩展的正则表达式
  - 案例
    - 搜索某个文件中，包含某个关键词的内容 grep liangxu /etc/passwd
    - 搜索多个文件中，包含某个关键词的内容 grep liangxu /etc/passwd /etc/shadow
    - 搜索多个文件中，包含某个关键词的内容，不显示文件名称 grep -h liangxu /etc/passwd /etc/shadow
    - 递归搜索，不仅搜索指定目录，还搜索其所子目录内是否有关键词文件 grep -r hello *
    - 输出在某个文件中，包含某个关键词的数量 grep -c root /etc/passwd /etc/shadow
    - 忽略大小写 grep -i LIANGXU /etc/passwd
    - 在文件中查找要搜索的内容，并显示行号 grep -n liangxu /etc/passwd
    - 反向查找 grep -v liangxu /etc/passwd
    - 只列出符合匹配的文件名，不列出具体的匹配行 grep -l hello *
    - 搜索某个文件中，精准匹配到某个关键词的内容 grep -x hello file1.txt
    - grep -x hello file3.txt
    - 判断某个文件中，是否包含某个关键词，通过返回状态值输出结果（0为包含，1为不包含），方便在Shell脚本中判断和调用 grep -q hello file1.txt
    - grep -q hello file2.txt
- egrep
  - 作用：查找指定的字符串
  - 语法：egrep [参数] [文件]
  - 参数：同grep
  - 案例
    - 查找包含一个或一个以上"e"的内容 egrep 'e+' file.txt
    - 查找包含linux或666的内容 egrep 'linux|666' file.txt
    - 查找包含linux整单词的内容 egrep '\linux\' file.txt
    - 查找包含一个或多个linux整体的内容 egrep '\linux+\' file.txt
    - 查找以e开头的内容 egrep '^e' file.txt
    - 查找以linux结尾的内容 egrep 'linux$' file.txt
    - 查找连续出现3次ab的内容 egrep 'ab{3}' file.txt
    - 查找包括abc或ab的内容 egrep 'abc|ab' file.txt

查看文本内容

- cat
  - 作用：在终端上显示文件内容
  - 语法：cat [OPTIONS]... [FILE]...
  - 参数
    - -n 显示行数（空行也编号）
    - -s 连续重复的空行
    - -b 显示行数（空行不编号）
    - -E 每行结束处显示$符号
    - -T 将TAB字符显示为 ^I符号
  - 案例
    - 查看 file.txt 文件内容 cat file.txt
    - 查看多个文件内容 cat file1.txt file2.txt
    - 查看 file.txt 文件内容，并显示行数编号 cat -n file.txt
    - 去除重复的空行 cat -s file.txt
    - 重定向文件内容
      - cat file.txt > file1.txt # 如果file1.txt不存在，则创建，并且内容会被覆盖
      - cat file.txt >> file1.txt # 如果file1.txt不存在，则创建，并且内容会被追加到末尾
    - 将 file1.txt 和 file2.txt 合并为一个文件 cat file1.txt file2.txt > combinedfile.txt
    - 使用 cat 创建文件（按键 Enter 入文本，完成后按 CRTL-D 保存文件） cat > file.txt
- more
  - 作用：分页显示文本文件内容
  - 语法：more [options] file...
  - 基本操作
    - 回车：下滚一行
    - 空格：下翻一页
    - num 指定每屏显示的行数
    - +num 从第+num行开始显示
  - 参数
    - -p 清除屏幕再显示文本文件的剩余内容
    - -c 与 -p相似，不滚屏，先显示内容再清除旧内容
    - -s 多个空行压缩成一行显示
    - +/pattern 分页显示指定的文本内容
    - more -2 /bashrc
    - 先进行搜索操作，随后以每次20行内容的格式显示指定的文本文件内容
    - more -c -10 -2 /bashrc
    - 分页显示指定的文本文件内容，清屏后结束行以上空行的情况，到0一行空行显示
    - more -s +/bashrc
    - 从第+10行开始，分页显示指定的文本文件内容
    - more +10 -s /bashrc
- less
  - 作用：分页显示文本内容
  - 语法：less [参数] 文件
  - 快捷操作
    - 空格键 下翻一页
    - 回车键 下滚一行
    - j 下滚一行
    - k 上滚一行
    - b 回翻一页
    - f 下翻一页
    - q 退出
    - /word 搜索word关键词
  - 案例
    - 查看文件 less -/bashrc
    - 查看多个文件 less -/bashrc -/bash_history
      - n：浏览下一个文件
      - p：浏览前一个文件
    - 查看历史命令的更 less 命令分页显示 history | less
- head
  - 作用：显示文件开头的内容
  - 语法：head [OPTIONS]... [FILE]...
  - 参数
    - -n <数字> 定义显示行数
    - -c <数字> 指定显示头部内容的字符数
  - 案例
    - 显示文件的前10行内容（默认） head -/bashrc
    - 显示文件的前5行内容 head -n 5 -/bashrc
    - 显示文件除了最后4行的全部内容 head -n -4 -/bashrc
    - 显示文件的前20个字节 head -c 20 -/bashrc
    - 显示文件除了最后30个字节的全部内容 head -c -30 -/bashrc
- tail
  - 作用：查看文件尾部内容
  - 语法：tail [OPTIONS]... [FILE]...
  - 参数
    - -c N 输出文件尾部的 N（N为数字）个字节内容
    - -f 显示文件最新追加的内容
    - -n N 输出文件的尾部N（N位数字）行内容
    - 显示 file.txt 文件的最后 10 行内容 tail file.txt
    - 显示 file.txt 文件的最后 20 行内容 tail -n 20 file.txt
    - 显示 file.txt 文件从第 20 行至文件末尾 tail +20 file.txt
    - 显示 file.txt 文件的最后 10 个字符 tail -c 10 file.txt
    - 动态显示文件的最后 10 行内容 tail -f file.txt
- tac
  - 作用：反向显示文件内容
  - 语法：tac [参数] [文件]
  - 案例：tac file.txt
- nl
  - 作用：添加行号
  - 语法：nl [OPTIONS]... [FILE]...
  - 参数
    - -b a 也给空行添加行号（类似 cat -n）
    - -b t 空行不显示行号
    - -n ln 行号在屏幕的最左方显示
    - -n rn 行号在自己行位的最右方显示，且不加0
    - -n rz 行号在自己行位的最右方显示，且加0
    - -w 行号栏位的占用的位数
    - -p 不在逻辑页重新计数
  - 案例
    - 用nl列出file.txt的内容，空行也加上行号 nl -b a file.txt
    - 行号自己行位的最右方显示，且加0对齐格式 nl -b a -n rz file.txt
    - 行号宽度设置为5 nl -b a -n rz -w 5 file.txt
    - 空行不显示行号 nl -b t file.txt