

# 管道

该命令 `ls | head` 运行 `ls` 和 `head` 并将 `ls` 的标准输出通过管道传输到 `head` 的标准输入中。

以下 shell 命令在管道中特别有用：

- `cat [FILENAME [FILENAME...]]` 将一个或多个文件的内容写入标准输出。这是启动管道的好方法。如果省略所有文件名，`cat` 只会读取其标准输入并将其写入标准输出。
- `head [-n N]` 读取其标准输入并仅将前 `N` 行（如果关闭该选项，则默认值为 10）写入标准输出。您还可以在参数前添加一个减号，例如 `head -n -2` 跳过最后两行并写入其余所有行。
- `tail [-n N]` 与 `head` 类似，只是它写入最后 `N` 行（带有减号，它会跳过前 `N` 行）。
- `sort` 将其所有标准输入读入内存缓冲区，然后对行进行排序并将它们全部写入标准输出。
- `uniq` 读取标准输入并写入标准输出，但会跳过紧接着的重复行，例如，如果有三行 `A`、`A`、`B`，那么它只会写入 `A`、`B`，但如果它得到 `A`、`B`、`A`，则会写入所有三个。删除重复行的常用方法是 `... | sort | uniq | ...`。
- `grep [-iv] EXPRESSION` 读取标准输入并仅打印与标准输出的正则表达式匹配的行。它 `-i` 不区分大小写，并且 `-v` 仅打印与表达式不匹配的行。
- `sed -e COMMAND` 从标准输入读取行，根据命令转换它们并将结果写入标准输出。`sed` 有自己的命令语言，但最常见的一种是 `s/SOURCE/DEST/` 将与源正则表达式匹配的子字符串更改为目标正则表达式。
- `wc [-l]` 代表字数统计，但用 `-l` 它来计算行数。`wc -l` 如果您只想知道特定命令或管道产生多少个结果（假设每行一个结果），则将放在管道的最末端非常有用。

所有这些命令实际上都采用可选的额外文件名作为参数，在这种情况下，它们从该文件中读取作为输入。例如，要显示名为 `Readme.txt` 的文件的前 10 行，您可以执行 `cat Readme.txt | head` 或 `head Readme.txt`。

## 单词列表练习 - 管道和正则表达式

大多数 Linux 发行版（包括 Debian）都附带一个字典文件 `/usr/share/dict/words`，其中包含按字母顺序排列的英语单词列表，用于拼写检查程序。该列表包括一系列专有名词，例如国家和城市。如果您想在没有它的系统上查看它，您可以使用以下命令下载它：

```
wget https://users.cs.duke.edu/~ola/ap/linuxwords -O words
```

`wget` 是用于下载文件的两个实用程序之一，另一个是 `curl`。请注意，输出文件名的选项是大写 `O`，而不是小写 `o` 或零。

找到一行命令（可能带有管道），将以下内容打印到终端。您可以 `cat /usr/share/dict/words | ...` 通过提供单词文件作为管道中第一个命令的参数来使用 `cat` 启动每个命令，也可以不使用 `cat` 来执行每个命令。

如果英语不是您的母语，请忽略猜测部分 - 它不会被评估。

- 文件中的第一个单词。你能猜到会是什么吗？
- 文件中的最后一个字。你能猜出这个吗？
- 单词文件中的单词数 - 每行一个单词。
- 文件中的第 6171 个单词。你能读懂我的心思并直接猜出这个词吗？
- 所有包含字母 Q 的单词均大写。（包含一个或多个字母的字符串的正则表达式与包含该表达式作为子字符串的所有字符串相匹配。）
- 所有以字母 X 开头的单词。正则表达式 `x` 将匹配单词中任何位置的 X，但 `^x` 仅匹配字符串开头的 X。
- 所有以 j 结尾的单词。（该表达式 `'j$'` 仅在字符串末尾匹配 aj，但您必须用单引号将其括起来以阻止 shell 解释美元符号）。
- 包含字母 Q 的单词数，忽略大小写（例如是否大写）。
- 包含字母序列的前五个单词 `cl`。
- 所有包含序列“kp”但不包含“ckp”的单词。你能猜出其中任何一个吗？
- 恰好两个字母的最后 15 个单词。表达式 `.`（句点）匹配单个字符，例如，`'^...$'` 将匹配格式为 *string* 的开始和结束之间正好三个字符的所有字符串。由于美元符号，您需要引用它。
- 列表中前 100 个单词中包含字母 y 的所有单词。
- 列表中最后 100 个单词中包含字母 y（无论是否大写）的前 5 个单词。
- 所有不带元音的三字母单词 (aeiou)。正则表达式 `'[aeiou]'` 匹配包含括号字符之一的任何字符串；您需要引号来阻止 shell 解释括号。请记住也要排除带有大写元音的单词。其中有 343 个。
- 所有单词均由 7 个字母组成，其中第三个字母是 e，单词结尾为“-ded”。这种搜索对于填字游戏来说非常有用。这种形式的单词有 14 个，你能猜出来吗？

额外的正则表达式问题：

- 查找所有以 P 开头（无论是否大写）且至少包含四个字母 a 的单词。\* 在正则表达式中将 a 放在某个内容后面会搜索任意数量的重复项，包括 0，因此例如 `'a*'` 会查找具有任意数量的字母 a 的单词，包括 0（这不是您想要的）。您需要单引号来阻止 shell 扩展 \*。你能猜出这些词吗？解决方案中有 14 个匹配项，但基本上有 5 个单词：两个恶魔词和三个非专有名词的名词，全部具有所有格和复数形式（除了其中一个本身是复数形式）。