# 查看命令文档

type 查看命令类型（是否是shell内建命令）

help 查看内建命令文档

man 查看非内建命令文档

type ls

help cd

man ls

# sh -c的作用

`sudo echo "hahah" >> test.csv`

-bash: test.asc: Permission denied

第一种是利用 "sh -c" 命令，它可以让 bash 将一个字串作为完整的命令来执行，这样就可以将 sudo 的影响范围扩展到整条命令。具体用法如下：

$ sudo /bin/sh -c 'echo "hahah" >> test.asc'

# 常用命令

## 常用符号

$\* 为"1 2 3" （一起被引号包住）

$@ 为"1" "2" "3" （分别被引号包住）

$# 为3 (表示参数的数量）

$$ Shell本身的PID（ProcessID，即脚本运行的当前 进程ID号）

$! Shell最后运行的后台Process的PID(后台运行的最后一个进程的 进程ID号)

$? 最后运行的命令的结束代码（返回值）即执行上一个指令的返回值 (显示最后命令的退出状态。0表示没有错误，其他任何值表明有错误)

${1} 第一个变量

$()和 ` `

在 bash shell 中，$( ) 与` ` (反引号) 将内容变成可执行命令

${ }

${ }用于表示变量。一般情况下，$var 与${var} 并没有什么不一样，但是用 ${ } 会比较精确的界定变量名称的范围。

$[] $(())

$[]和$(())是一样的，都是进行数学运算的。支持+ - \* / %（“加、减、乘、除、取模”）。但是注意，bash只能作整数运算，对于浮点数是当作字符串处理的。

(( ))及[[ ]]

分别是[ ]的针对数学比较表达式和字符串表达式的加强版。

<https://blog.csdn.net/qq_40907977/article/details/110949516>

## 四则运算

s=$((60000/$taskNum))

echo "ssssss: $s"

a=0

((a=60000/$taskNum))

echo "aaaaaa: $a"