

Perl单行命令: 提高数据处理效率

1. perl 命令行接口

CentOS 8.2 默认安装的版本是 v5.26.3

```
1  $ perl -h
2
3  Usage: perl [switches] [--] [programfile] [arguments]
4      -O[octal]          specify record separator (\0, if no argument)
5      -a                autosplit mode with -n or -p (splits $_ into @F)
6      -C[number/list]   enables the listed Unicode features
7      -c                check syntax only (runs BEGIN and CHECK blocks)
8      -d[:debugger]     run program under debugger
9      -D[number/list]   set debugging flags (argument is a bit mask or
    alphabets)
10     -e program         one line of program (several -e's allowed, omit
    programfile)
11     -E program        like -e, but enables all optional features
12     -f                don't do $sitelib/sitecustomize.pl at startup
13     -F/pattern/        split() pattern for -a switch (//'s are optional)
14     -i[extension]     edit <> files in place (makes backup if extension
    supplied)
15     -Idirectory       specify @INC/#include directory (several -I's allowed)
16     -l[octal]         enable line ending processing, specifies line terminator
17     -[mM][-]module    execute "use/no module..." before executing program
18     -n                assume "while (<>) { ... }" loop around program
19     -p                assume loop like -n but print line also, like sed
20     -s                enable rudimentary parsing for switches after
    programfile
21     -S                look for programfile using PATH environment variable
22     -t                enable tainting warnings
23     -T                enable tainting checks
24     -u                dump core after parsing program
25     -U                allow unsafe operations
26     -v                print version, patchlevel and license
27     -V[:variable]     print configuration summary (or a single Config.pm
    variable)
28     -w                enable many useful warnings
29     -W                enable all warnings
30     -x[directory]     ignore text before #!perl line (optionally cd to
    directory)
31     -X                disable all warnings
32
33  Run 'perldoc perl' for more help with Perl.
```

常用的命令参数功能描述:

1	<code>-0<octal></code>	(8进制表示)指定记录分隔符(\$/变量)，默认为换行。
2	<code>-a</code>	自动分隔模式，用空格分隔\$_并保存到@F中。相当于 <code>@F = split ' '</code> 。分隔符可以使用 <code>-F`</code> 参数指定。
3	<code>-F</code>	指定 <code>-a`</code> 的分隔符，可以使用正则表达式 <code>/pattern/</code> 。
4	<code>-e</code>	执行指定代码块。
5	<code>-i<extension></code>	原地替换文件，并将旧文件用指定的扩展名备份。不指定扩展名则不备份。
6	<code>-l</code>	对输入内容自动 <code>chomp</code> ，对输出内容自动添加换行。
7	<code>-n</code>	自动按行循环，相当于 <code>while(<>) { # your code goes here }</code>
8	<code>-p</code>	自动按行循环以及输出，相当于 <code>while(<>) { # your code goes here; print; }</code>

2. 常用的 one-line 命令行

- 替换一个文件行尾所有的空白

```
1 | perl -lpe 's/\s*$//'
```

注意事项：`-l` 参数保留了每行的`\n` 分隔符。

- 原位编辑

```
1 | perl -i.bak -pne 's/\t/,/g' zotu_table.txt
```

- 给每一行添加行号. `$.` 是行号符.

```
1 | perl -ne 'print "$.\t$_"' zotu_table.txt
```

- 自动分割, 默认按照`\t` 制表符分割

```
1 | perl -lane 'print "$F[0]\t$F[1]"' zotu_table.txt
```

注意事项: 使用 `-F` 参数可以设定自动分隔符

```
1 | perl -F: -lane 'print $F[0]' /etc/passwd
2 | perl -F:/ -lane 'print $F[0]' /etc/passwd
3 | perl -F'\t' -lne 'print $F[0]' zotu_table.txt
```

- BEGIN 和 END 可以在循环前初始化参数，接受打印结果

```
1 | grep -v "#" zotu_table.txt | perl -lane 'BEGIN {$t = 0}; $t += $F[1]; END { print $t }'
```

- 在线计算器

```
1 | perl -l -e 'print 2**8'
```

3.Reference

[The top 10 tricks of Perl one-liners](#)