Perl one-liners

Rev. 20060114


20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42

Overview

`-e`	`BEGIN/END`
`-n`	
`-p`	
`-M`	
`-i`	
`-l`	
`-w`	
`-c`	
`-a`	
`-F`	

Build-in variables

`$_` current input line

„`print;`” == „`print $_;`”

„`/wyraz/`” == „`$_ =~ /wyraz/`”

`$.` current input line number

„Hello world”

```
~$ perl -e 'print "hello
  world!\n"'
hello world!
```

Example

```
~$ ls -lAF | perl -e 'while (<>) { next if
  /^d/; $sum += (split)[4] } print "$sum\n"'
34914
```

Example (decomposed)

```
~$ ls -lAF
total 100
-rw-r--r-- 1 bfilipow bfilipow 12288 Oct 21 18:53 .09LST16.pl.swp
-rw-r--r-- 1 bfilipow bfilipow 12288 Oct 22 06:29 .report.pl.swp
-rw-r--r-- 1 bfilipow bfilipow   299 Oct 21 18:50 09LST16.pl
drwxr-xr-x 4 bfilipow bfilipow  4096 Nov  6 20:59 Ansicolor.pl/
-rw-r--r-- 1 bfilipow bfilipow   126 Nov  3 20:52 ansicolor.pl
(…)

~$ ls -lAF | perl -e 'while (<>) { print $_ }'
total 100
-rw-r--r-- 1 bfilipow bfilipow 12288 Oct 21 18:53 .09LST16.pl.swp
-rw-r--r-- 1 bfilipow bfilipow 12288 Oct 22 06:29 .report.pl.swp
-rw-r--r-- 1 bfilipow bfilipow   299 Oct 21 18:50 09LST16.pl
drwxr-xr-x 4 bfilipow bfilipow  4096 Nov  6 20:59 Ansicolor.pl/
-rw-r--r-- 1 bfilipow bfilipow   126 Nov  3 20:52 ansicolor.pl
(…)
```

Example (decomposed)

```
~$ ls -lAF | perl -e 'while (<>) { next if /^d/;
  print $_; }'
#skip non-directories

~$ ls -lAF | perl -e 'while (<>) { next if /^d/;
  print +(split)[4], "\n" } '
# print the 4th group

~$ ls -lAF | perl -e 'while (<>) { next if /^d/;
  $sum += (split)[4] } print "$sum\n"'
#the final version
34914
```

Input

Standard input

```
~$ cat afile | perl -e 'while (<>) { print unless /\s+#/ }
   '

~$ perl -e 'while (<>) { print unless /\s+#/ }' < afile

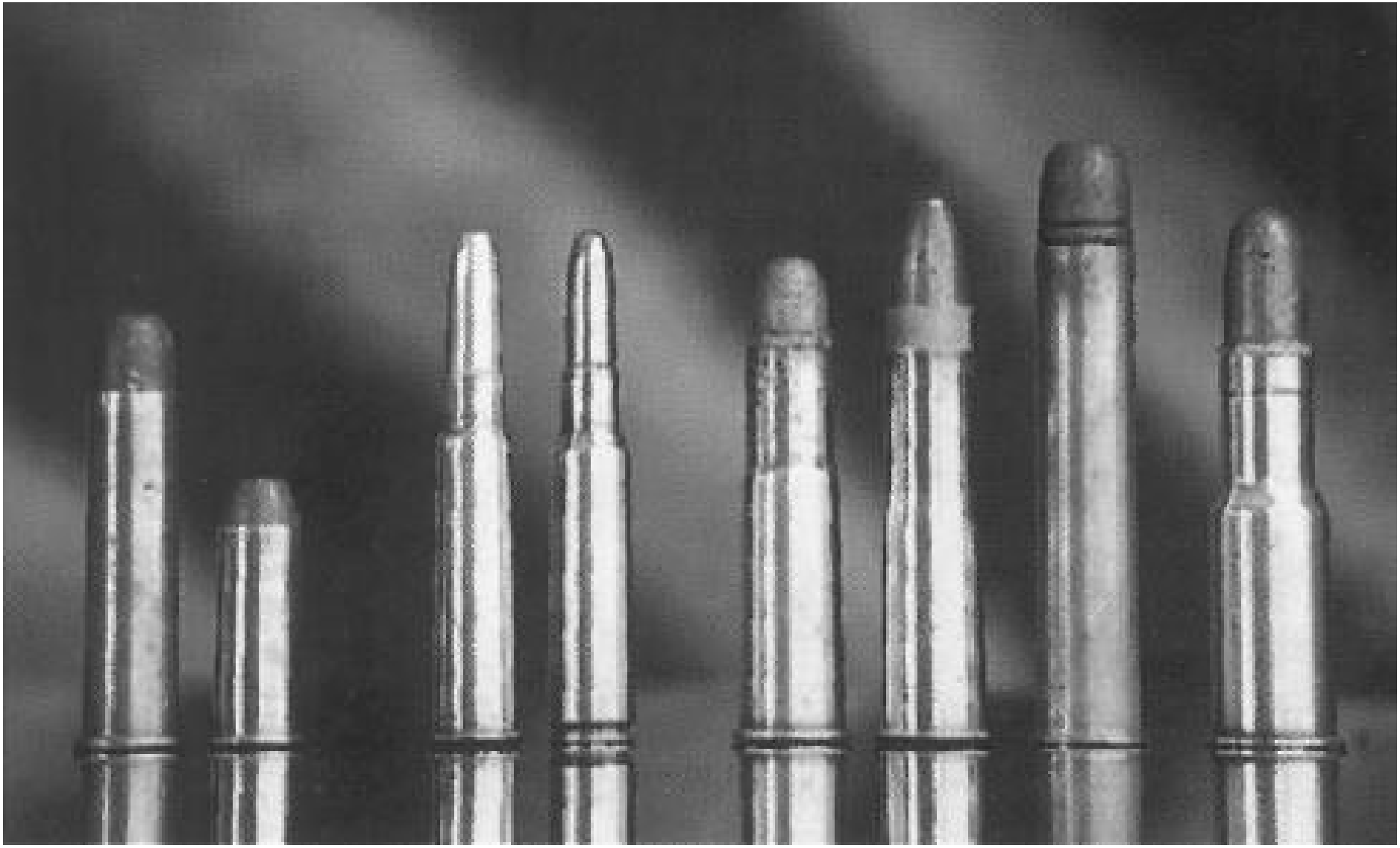
~$ perl -e 'while (<>) { print unless /\s+#/ }' afile
```

Command-line arguments

```
~$ perl -e 'print "@ARGV\n"' Foo Bar Buzz
Foo Bar Buzz
```

Useful command line switches

```
~$ perldoc perlrun
~$ perl --help
```



The `-e` switch

```
~$ perl -e 'print "Hello ";' -e
  'print "world\n";'
Hello world
```

The -n switch

```
while (<>) {
  <-e argument>
  <-e argument>
}
```

```
~$ perl -ne 'print $_' afile
#cat afile
```

The `-n` switch

```
# 1. just lines 15 to 17
~$ perl -ne 'print if 15 .. 17' test.file
```

The -n switch

```
# 2. just lines NOT between line
  10 and 20
~$ perl -ne 'print unless 10 ..
  20‘ test.file
```

The `-n` switch

```
# 3. lines between START and END
~$ perl -ne 'print if /START/ .. /END/' test.file
```

The -n switch

```
# 4. lines NOT between START and
  END
~$ perl -ne 'print unless /START/
  .. /END/' test.file
```

The -n switch

```
# just lines 15 to 17, efficiently
~$ perl -ne 'print if $. >= 15; exit if $.
  >= 17;‘
```

The -p switch

```
while (<>) {
  <-e argument>
  <-e argument>
  print;
}
```

```
~$ perl -pe '' afile
#cat afile
```

The -p switch

```
# remove first column from <ls -l> output
~$ ls -l | perl -pe 's/\S+ //'

 1 bfilipow bfilipow  299 Oct 21 18:50 09LST16.pl
 4 bfilipow bfilipow 4096 Nov  6 20:59 Ansicolor.pl
 1 bfilipow bfilipow  126 Nov  3 20:52 ansicolor.pl
(…)
```

The `-i` switch

```
~$ cat test
XXX
XXX
test
~$ perl -pi -e 's/XXX/YYY/' test
~$ cat test
YYY
YYY
test
~$
```

The -i switch

```
# 1. in-place edit of *.c files changing all
  foo to bar
~$ perl -p -i.bak -e 's/\bfoo\b/bar/g' *.c

# 2. delete first 10 lines
~$ perl -i.old -ne 'print unless 1 .. 10'
  foo.txt

# 3. Insert line numbers in a file
~$ perl -pi -e '$_ = sprintf "%04d %s", $.,
  $_' test
```

The `-l` switch

```
~$ perl -e
  '$date=localtime(time); print
  $date,"\n";`
Mon Nov 21 12:58:25 2005

~$ perl -l -e
  '$date=localtime(time); print
  $date;`
Mon Nov 21 12:59:15 2005
```

The `-w` switch

```
~$ perl -e 'qwerty'
~$ perl -w -e 'qwerty'
Unquoted string "qwerty" may clash
  with future reserved word at -e line
  1.
Useless use of a constant in void
  context at -e line 1.
~$
```

The `-c` switch

```
~$ perl -c ansicolor.pl
ansicolor.pl syntax OK
~$ perl -c prompt.pl
Can't locate IO/Prompt.pm in @INC
  (@INC contains: /etc/perl
  /usr/local/lib/perl/5.8.7 (…)
  /usr/local/lib/site_perl .) at
  prompt.pl line 1.
BEGIN failed--compilation aborted at
  prompt.pl line 1.
~$
```

The -a switch

```
~$ ls -lA | perl -a -lne 'print
  $F[4] . "\t" . $F[8];'
# size + name

~$ ls -lA | perl -a -lpe '$_ =
  $F[4] . "\t" . $F[8];'
# size + name
```

The -M switch

```
~$ cat afile | perl -w -Mstrict
  -e 'my $var = 17; print $var'
17
```

Other switches

`-F/pattern/` split /pattern/

(...)

```
~$ perl --help
```

BEGIN/END

END

```
~$ ls -lAF | perl -ne 'next if /^d/;
  $sum += (split)[4]; END{ print
  "$sum\n" }'
34914

~$ ls -lAF | perl -alne 'next if
  /^d/; $s+=$F[4]; END{ print $s; };`
#remember Example?
```

BEGIN

```
~$ ls -lAF | perl -ne
  'BEGIN{$sum=1024} next if /^d/;
  $sum += (split)[4]; END{ print
  "$sum\n" }'
35938
```

Summary

Perl one-liners
list of switches

`-c` check syntax only (runs BEGIN and CHECK blocks)

`-e 'command'` one line of program (several -e's allowed, omit program file)

`-i` edit files in place

`-l` enable line ending processing

`-Mmodule` execute `use module...' before executing program

`-n` assume 'while (<>) { ... }' loop around program

`-p` assume loop like -n but print line also

`-w` enable many useful warnings (RECOMMENDED)

Conclusion

- one-liners are not-so-easy, but handy
- throwaways, not pyramids
- check twice
- have a lot of fun

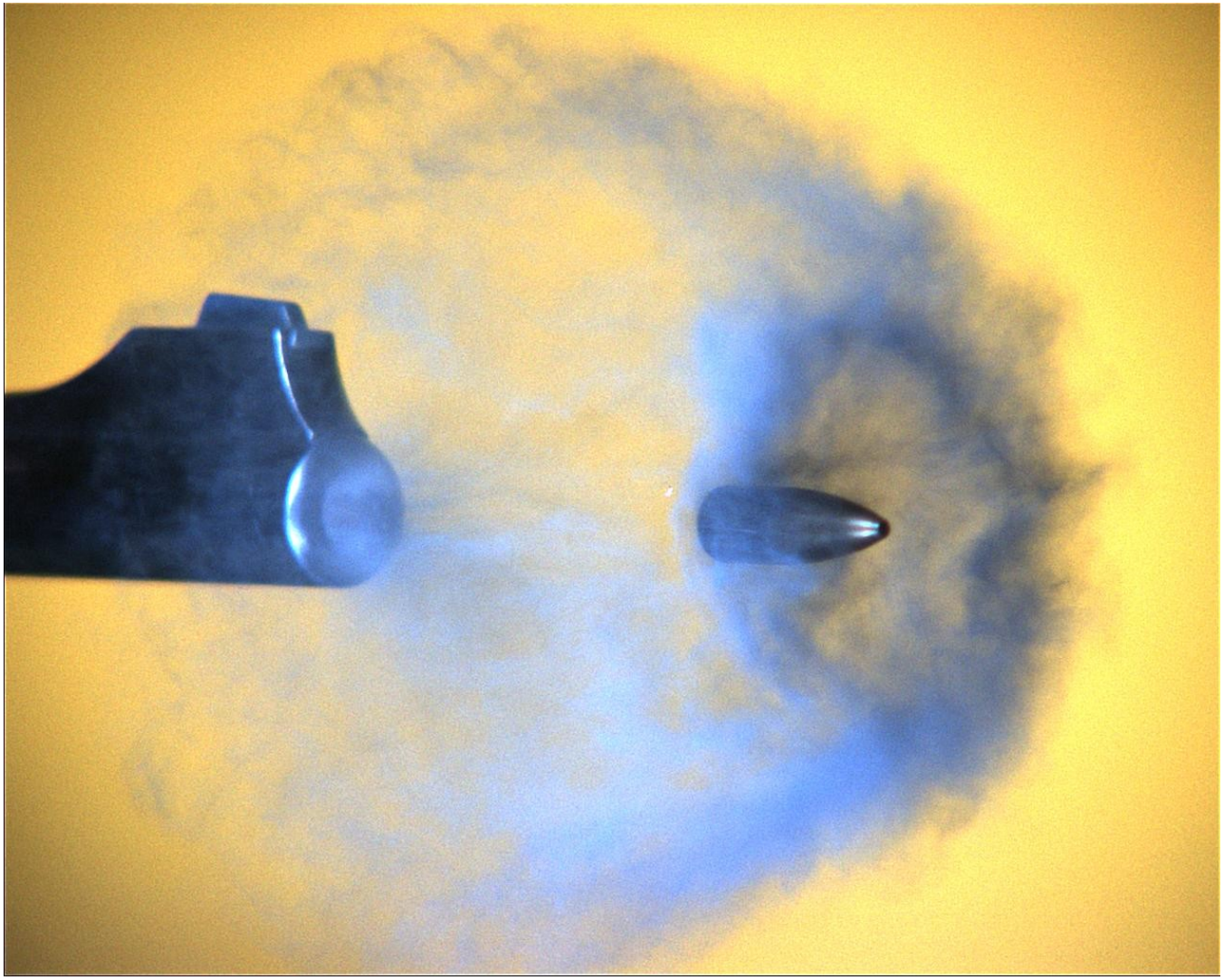
References

- „*Perl One-liners*”, Jeff Bay, http://www.theperlreview.com/Issues/
- *„Perl as a command-line utility”,* Teodor Zlatanov *http://www-128.ibm.com/developerworks/linux/library/l-p101/*
- „*More one-line Perl scripts*”, Teodor Zlatanov http://www-128.ibm.com/developerworks/linux/library/l-p1
- `~$ perldoc perlrun`
- www.google.com

Q&A

Workshop

One-liners applied



Exercise a

- Create 24 files a.txt .. z.txt

```
~$ perl -MShell -le 'for
  ('a'..'z')
  { touch($_.".txt"); }`
```

OR

```
~$ perl -le 'for (`a'..`z')
  { `touch $_.txt;` }'
```

Exercise b

- Every second column from output?

from

```
drwxr-xr-x   19 root     root         4096 May  4  2005 usr
drwxr-xr-x   23 root     root         4096 Nov 18 13:48 var
```

to

```
d w r x -     9 r o       o t        4 9   a   4   0 5 u r
d w r x -     3 r o       o t        4 9   o   8 1 : 8 v r
```

```
~$ ls -Al / | perl -aF// -nle '$i=1;@F=map{$i++%2?$_:" "}@F;print
   @F;`
```

OR

```
~$ ls -Al / | perl -pe `s/(.)./$1 /g;`
```

Exercise c

- Print every 3rd row from input

```
ll | perl -ne 'print unless $. %
  3;'
```

Example d

- increment all numbers found in files
- `perl -i -pe 's/(\d+)/ 1 + $1 /ge' file1 file2`

Example e

- delete all but lines between START and END

```
perl -i.old -ne 'print unless
  /^START$/ .. /^END$/' foo.txt
```

Example f

- command-line that reverses the whole input by lines (printing each line in reverse order)

```
perl -e 'print reverse <>' file1
  file2 file3 ....
```

Example g

- Create file with 30 lines each containing „XYZ”

```
perl -le 'for (1..30) { print
  „XYZ”;}' > file1
```

- replace string XYZ with a random number less than 611 in these files

```
perl -i.bak -pe "s/XYZ/int
  rand(611)/e" file1
```

Example h

- **Insert line numbers in a file**
- `perl -pi -e '$_=sprintf "%04d %s", $., $_' test`

Example i

- Appending data to existing files is easy. So is inserting data into arbitrary locations in a file, such as prepending a new first line to a set of files. In the following case, #!/usr/bin/perl will be added as the first line of all *.pl files in the current directory.
- `$ perl -i -ple 'print q{#!/usr/bin/perl} if $. == 1; close ARGV if eof' *.pl`

Example j

- Replace the second line of a file with some text, but only if that line is blank.
- ```
  $ perl -ple '$_ = "some text"
  if $. == 2 and m/^$/;` file1
  ```
  ```

Example k

- Add first and penultimate columns
- `perl -lane 'print $F[0] + $F[-2]'`

Example I

- **Remove empty lines from a file**
- `perl -ni.bak -e'/\S/ && print' file1 file2`

Fortune

```
if [ -e /usr/games/fortune ];
  then
     /usr/games/fortune -n 200 -s
else
perl -ne '$a .= $_; END{ @b =
  split/%\n/, $a; print $b[ int
  rand @b ]; }' ~/fortunes/*
fi
```

Example …

- Google for more

Back-up

Obfuscation

```
~$ perl -e
  '$_=sprintf("%08b"x12,map
  hex,"5EE62C8938D2813268509E9B"=
  ~/../g);print sort{(1+chop)*2-
  3}split"","Puht r arJtel\ncerae
  on,khs";'
```