Perl 入门和提高　　Lesson 6

周晓方

courses@xfzhou.homeftp.org

Perlfunc 1 --- Help file, ( ), Category

- Where to find perlfunc information:
  - (console window) perldoc -f func_name
  - (perl install dir) **<u>c:\perl</u>**\html\index.htm *click “perlfunc”*
- Careful
  - `print 1+2+4;   # 7`
  - `print (1+2+4); # 7`
  - `print (1+2)+4; # 3, not 7, print (1+2) is a func-call.`
- Category (altogether, about 200 pre-defined functions)
  - Scalar/string, Reg-exp, Number, Arrays,
  - List data, Hashs, In/output, Record,
  - Filehandle/Dir, Control, Scoping, Miscelleous,
  - Process, Perl-module, Class/OO,
  - Socket, Uid/Gid, Network, Time, etc.

Perlfunc 2 --- String and numeric

- `$/` the input record separator, 默认的`$/`是`"\n"`
- `chomp $V; chomp @L;` remove trailing $/ only
- `chop $V; chop @L;` chop of the last char
- `chr(65)` returns "A" <=> `ord "A" == 65`
- `crypt $plantext, $salt (one way)`
- `hex("0xFA1E")` returns 64030 / 八进制用oct "175036"
- 二进制怎么办? `0b101010` ➔ `?` `0xABCD` ➔ `?`
  其实oct函数是百搭! `$v = oct $v if $v =~ /^0/;`
- Case convert: `uc/lc/ucfirst/lcfirst $str`
  结果在函数返回值里面，函数本身并不改变*`$str`*的值
- `index STR, SUBSTR, POS; rindex`
- `substr $str, $offset, $len, $replace`
- `sprintf format, LIST`
- `abs, cos, exp, int, sin, cos, etc......`
  若是数值运算量的程序，可用`perl-XS`接口调用`C`子程序；
  若需要科学计算，可以安装和学用`PDL`模块

Perlfunc 3 --- quick review

- `rand` *`$e`* #random fraction `[0,$e)`, default `[0,1)`
- `srand` 种子 #automatic called after `perl 5.004`
  - 随机整数? `int rand 100` 返回`0..99`的随机整数

- Array: `push pop shift unshift splice`
- List: `grep join map reverse sort`
- Hash: `delete each exists keys values`
  - Remove a pair from hash: `delete $hash{$key};`
  - Check existence of a pair: `exists $hash{$key};`
- Misc: `defined scalar undef wantarray`
- File: `binmode open close die warn print printf unlink rename read seek tell sysread syswrite` *`etc`*......

Perlfunc 4——File and Directory

- 新建目录mkdir/删除目录rmdir/切换目录chdir

- 打开目录返回目录句柄*opendir* HDIR, "dir"
- 读取目录*readdir* HDIR
  - 标量环境每次返回一个目录项，最终返回undef
  - 列表环境返回全部目录项。注意：*readdir*返回的是相对*opendir*的路径，不是相对当前目录的路径
- 目录指针回到目录的开始 *rewinddir* HDIR
- 返回当前目录指针的位置*telldir* HDIR
- 设置当前目录指针*seekdir* HDIR, POS
  - POS必须是某次*telldir* HDIR返回的值
- 关闭目录句柄 *closedir* HDIR

Perlfunc 5 --- Fun stuffs

- prototype: the prototype of a func查看函数原形
    - prototype "CORE::push"　　　返回"\@@"
- caller: trace bach the caller's stack frames
    - ($package, $filename, $line) = caller;
    - ($package, $filename, $line, $subroutine, $hasargs, $wantarray, $evaltext, $is_require, $hints, $bitmask) = caller($i);
- 改文件时间utime $Tacc, $Tmod, @files
- 获得文件属性

  ($dev, $ino, $mode, $nlink, $uid, $gid, $rdev, $size, $atime, $mtime, $ctime, $blksize, $blocks) = stat($filename);

  $size = (stat($finename))[7];
- 进程执行秒: ($user, $sys, $cuser, $csys) = times;
- time(): seconds since 0:0:0 UTC, Jan 1, 1970
    - ($sec,$min,$hour,$mday,$mon,$year,$wday,$yday) = gmtime(time) 或者localtime(time)
- die "…"和warn "…"

Perlfunc 6 --- powerful *eval*

- `eval $Code; eval {code};` # execute piece of perl code and returen error message in `$@`

```
$z = 0; $y = 1;
$x = $y / $z        # fatal run-time error!!!
```

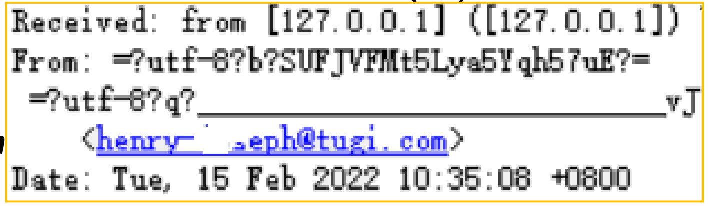
```
$z = 0; $y = 1;
eval {$x = $y / $z};     # just check $@
print $@;
# Illegal division by zero at - line 2.
print “Any how, I\'m still running...\n"
```

```
$z = 0; $y = 1;
eval '$x = $y / $z';     # same, but less effective
print $@;
# Illegal division by zero at (eval 1) line 1.
print "Any how, I\'m still running...\n"
```

```
eval {$x = $y / };       # compile (syntax) error
eval '$x = $y / ';       # run-time error
```

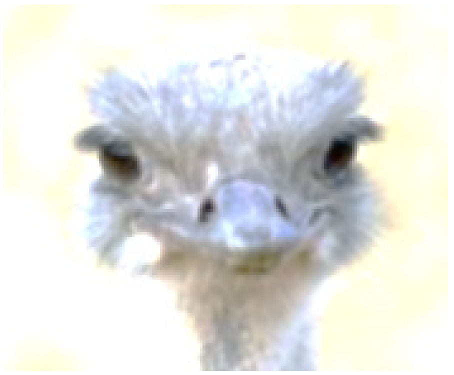
挽救Mail::POP3Client

- HW-03捕捉From邮件地址，会遗漏部分邮件
- 遇到bug有两个办法：
  1. 我是鸵鸟…看不见…看不见…
  2. 我是Sherlock Holmes，查查谁在捣蛋
- 现在诸位已经熟悉了**Perl**，让我们来着手分析问题
  仔细看邮件头信息和Mail::POP3Client的Head(·)：
- 邮件头信息：
  – 头信息每段都是顶格的"名称:数据"
  – 但数据可以是多行的，续行不顶格，白字符开头
- Mail::POP3Client的Head方法
  – Head($i)返回的单位是"行"，而不是头信息的有效段落
- 这就是能匹配到'From:'却匹配不到@地址的原因，'From:'和@地址不在同一行



```
Received: from [127.0.0.1] ([127.0.0.1])
From: =?utf-8?b?SUFJVFMt5Lya5Yqh57uE?=
 =?utf-8?q?________________________vJ
    <henry-  .eph@tugi.com>
Date: Tue, 15 Feb 2022 10:35:08 +0800
```

随手打补丁



```
{ # 该片断未优化，可能有bug，仅供参考。
    my $previous = -1;
    my @allheads;
    sub popHead {
        my $i = shift;
        my (@lines, $line);
        unless ($previous == $i) {
            my @lines = $pop->Head($i);
            @allheads = ();
            while (@lines) {
                $line = shift @lines;
                $line .= "\n" . shift @lines
                    while @lines and $lines[0] =~ /^\s/;
                push @allheads, $line;
            }
            $previous = $i;
        }
        return wantarray ? @allheads : shift @allheads;
    }
} # 使用了闭包，未用匿名函数，只能处理单路邮件，未融入原模块的OO机制
```

Big-endian v.s. Little-endian

- 多字节数据的内部存贮方式:
  - 例如双字123456==0x00_01_E2_40

字节顺序	地址0	地址1	地址2	地址3	典型处理器
高位在前 Big-endian	00	01	E2	40	Moto, Sparc, Cray, Cell, (MIPS)
低位在前 Little-endian	40	E2	01	00	Intel, VAX, (MIPS)

  - 以big-endian格式存贮123,456到数据文件上，在little-end处理器上读取后得到的是1,088,553,216
  - 有些处理器可以用通过特殊寄存器选择大端模式或者小端模式，请留意软件工具和编译器选项
  - 不同系统不同处理器之间交换数据要特别小心
  - 网络格式(TCP/IP包的字段等) 是指Big-endian
  - 设计软硬件系统时，还会有类似的比特顺序问题

Perlfunc 7 --- ***pack*** and ***unpack***

- list of readable data ➔pack(template, list)➔data chunk
- list of readable data🡐unpack(template, list)🡐data chunk

- Sample: fixed-length data file

–stduent id	8	5500123
–name	16	Bill Gates
–email address	24	..@..com

- Insert the record to database:

```
print MYDATA pack("A8 A16 A24",
     "5500123", "Bill Gates", "..\@..com");
 012345678901234567890123456789012345678901234567
"5500123 Bill Gates      ..@..com              "
```

- Retrieve the record from database:

```
seek(MYDATA, somewhere, 0);
read(MYDATA, $buf, 48, 0);
($sid, $sname, $semail) = unpack("A8 A16 A24", $buf);
```

Perlfunc 8 --- ***pack*** and ***unpack***

- stduent id是多字节整数的情况会怎样？看pack函数手册：
  - s, S, i, I, l, and L are inherently non-portable between processors and operating systems
  - n   An unsigned short in "network" (big-endian) order. (short here are _exactly_ 16 bits)
  - N   An unsigned long in "network" (big-endian) order. (long, _exactly_ 32 bits)
  - v   An unsigned short in "VAX" (little-endian) order.
  - V   An unsigned long in "VAX" (little-endian) order.
- Sample: fixed-length data file

–stduent id	(long long int)	8	07300723579 (0x00000001_B328337B)
–name	(char [ ]) 16		Bill Gates
–email address	(char [ ]) 24		..@..com

- Insert the record to database:

```
$id_low = 0xb328337b; $id_high = 0x00000001; #或用64位版本的perl
print pack("N2 A16 A24",
     $id_high, $id_low, "Bill Gates", "..\@..com");
 012345678901234567890123456789012345678901234567
"   ☺|(3{Bill Gates      ..@..com                "
```

- Retrieve the record from database:

```
seek(MYDATA, somewhere, 0);
read(MYDATA, $buf, 48, 0);
($id_h, $id_l, $sname, $semail) = unpack("N2 A16 A24", $buf);
```

Perlfunc 9 --- ***pack*** and ***unpack***

- Template, type+size, size is integer, '*', etc

  "A" An ASCII string, will be space padded. "Z" for ASCIIZ string
  pack "C*", (65..90); # ABCD..Z, **C** for unsigned char
  pack "B32","01010000011001010111001001101100";#"Perl",High bit 1st bin str
  pack "H8","5065726c"; # both produce "Perl",High nybble 1st hex str
  C无符号字符 c带符号字符 B高位在右比特 b高位在左 H/h
  S无符号16位 s有符号16位 L无符号32位 l有符号32位(see endian)
  N/n 32/16位无符号网络顺序(big-endian)，V/v 32/16位VAX顺序

- unpack(template, $scalar); the reverse of pack()

  unpack "C*", "ABCD…Z"; # (65..90)
  unpack "B8","A";   # "01000001" i.e. 0x41, decimal 65, or "A"
  pack "H8","AaBb"; # hex string "41614262"
  unpack "H8", pack "CCCC", 202, 120, 224, 10 # "ca78e00a"

Perlfun 10 – vec()

- vec STRING, OFFSET, BITS
  将STRING看作二进制数据，访问其中的一段

BITS >= 8: (8, 16, 32,..), big-endian order

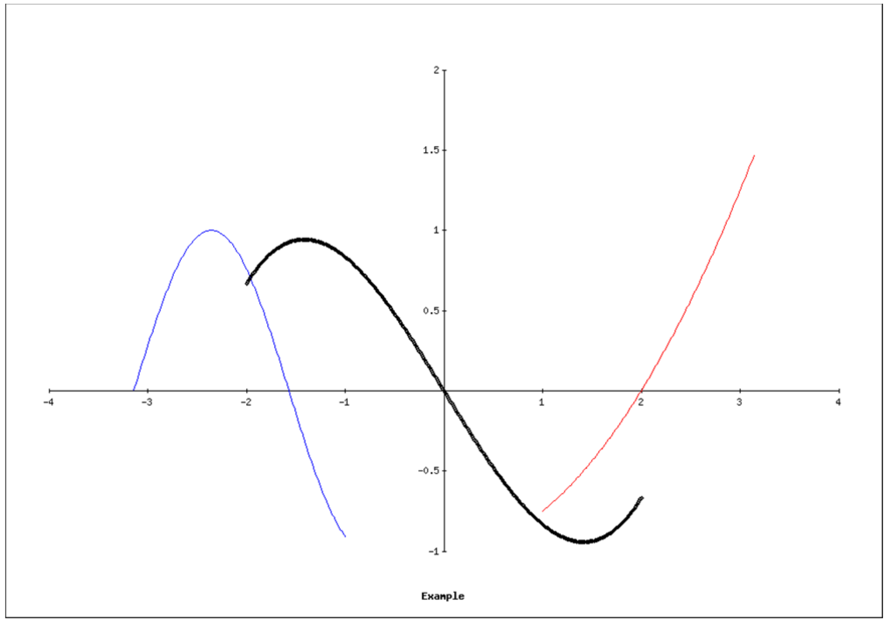
```
vec($foo,  0, 32) = 0x5065726C; # Perl
```

BITS < 8: (4, 2, 1), byte -> little-endian bits

```
join "", map vec("AB", $_, 1) , reverse 0..15; #0100001001000001
$a = chr(0x9e); vec($a, 1, 2) = 0; print ord($a);
#得到0x92(146),相当于0x9e & 0xf3
```

- Use vec/pack/unpack with binmode/seek/sysread/syswrite()
- Homework:
  - ~~062eval.pl is unsafe. Write a safer calculating program based on 062eval.pl, accepts only integers, '+', '-', '*', and '/‘. Also dump the hex value of input string, *学号-06.pl*~~
  - ~~输入rm *， 输出726D202A, not safe~~
  - ~~输入1+2*30， 输出312B322A3330, 61~~
  - ~~输入0/0， 输出302F30， Illegal division by zero at (eval 1) line 1.~~

homework准备，介绍作图模块

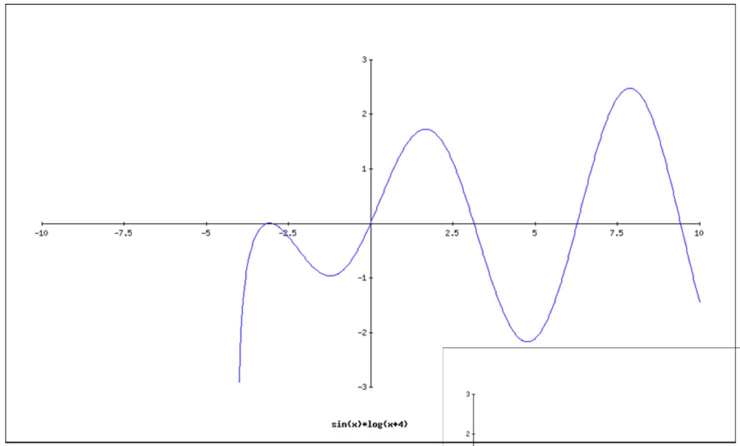


- cpan安装Chart::Plot，作图风格类似数学函数曲图
- 每次加入一组数据，分别是x和y数组的引用
- 一张图可放多段数据，自动设置纵横轴、标出
- 下例画出sin(2x) –pi~-1，
  x**2/2-1 1~pi，
  x**3/6-x -2~2

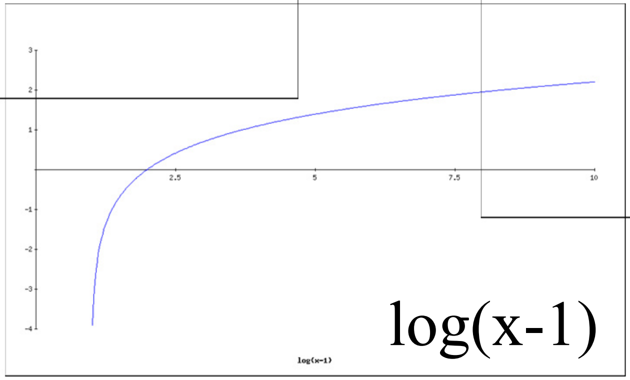
```
#!/usr/bin/perl -w
use strict;
use Chart::Plot;
my $fig = Chart::Plot
        ->new(1000, 700);
my @x = map $_/100, 100..314;
my @y = map $_**2/4-1, @x;
$fig->setData([@x], [@y],
        'Red SolidLine NoPoints');
@x = map -$_/100, 100..314;
@y = map sin(2*$_), @x;
$fig->setData([@x], [@y],
        'Blue Dashedline NoPoints');
@x = map $_/100, -200 .. 200;
@y = map $_**3/6-$_, @x;
$fig->setData([@x], [@y],
        'Black Noline Points');
$fig->setGraphOptions(
        'title' => 'Example',);
open F, '>Fig.png' or die;
binmode F;
print F $fig->draw('png');
close F;
1;
```

Homework作业7

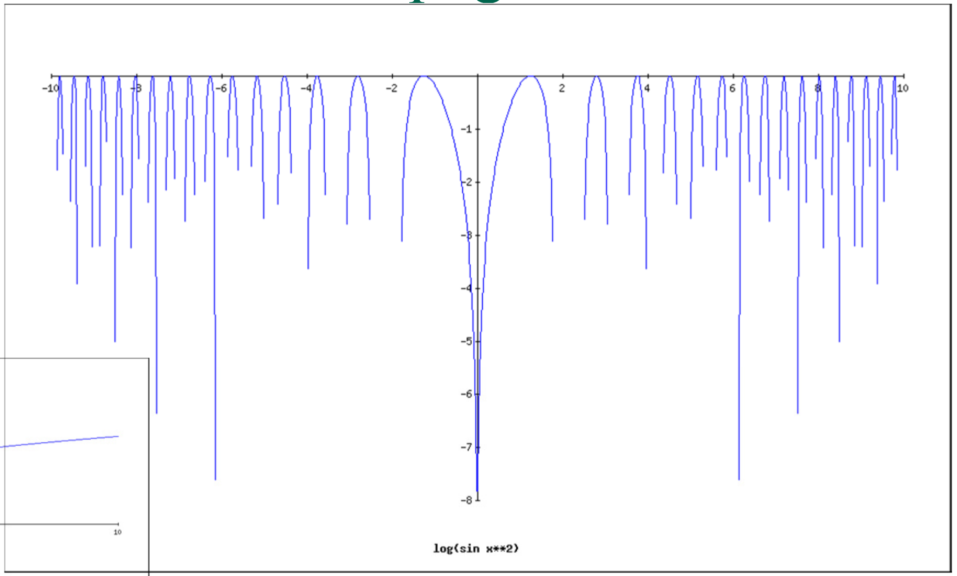
- 从<>读入数学表达式，单变量x，如log(sin(2*x))，+-*/()指数**浮点数、函数sin cos log exp abs sqrt
- 用适当的方法过滤输入，发现非法输入则报错
- 用eval在区间[-10,10]求值表达式，步长0.01，并作图
- 无效区间不作图，如log(x)，在x<=0的区间没有曲线
- 如果函数在整个[-10,10]上都无效，则报错
- 递交 **学号-07.pl**，结果文件存入 学号-07.png



sin(x)*log(x+4)



log(x-1)



log(sin x**2)

Special variable 1-- `perlvar`

- 都是些奇怪的符号或全大写的名字
- $_ Default parameter for many funcs and RE
- In map( ) and grep( ) function: $_

```
@upper_case_list = map(uc($_), @list);
```

- In sort( ) function: $a, $b

```
# same thing, but with explicit sort routine
@articles = sort {$a cmp $b} @files;

# now case-insensitively
@articles = sort {uc($a) cmp uc($b)} @files;

# same thing in reversed order
@articles = sort {$b cmp $a} @files;
```

@_	parameter list passed to sub routine
$$	Pid
$(	Group id
$]	Perl版本号＋小数点patch level 5.006001

Special Variables 2

- $. Current line number of the file handle last read
- $ARGV current file name when read from < >
- $" separator for print "@array";
- $! Error number or error string (see also $^E)
- $@ Error string of last eval( )
- $^E Extented OS Error information
- $? Status code return by child process, closed pipe, `` etc
- Pattern memory: $1, $2, …, $+ (last bracket matches)
- "ABCDEFG" =~ /CD/; print " **$` - $& - $'** ";

Special Variables 3

- $0 Program name
- Command line: @ARGV
- Environments: %ENV

```
 foreach $key (keys(%ENV))
   {print "$key\t$ENV{$key}\n";}
```

- Signal handling: %SIG

```
     $SIG{"HUP"} = "IGNORE";
     $SIG{"INT"} = "DEFAULT";
     $SIG{"QUIT"}= \&My_handler;
     $SIG{__WARNING__} = sub {
          die $_[0];
          }
```

- $^O OS Name when perl was built
- **perldoc perlvar**