Síťujeme v Perlu

Milan Šorm
sorm@fi.muni.cz
http://www.fi.muni.cz/~sorm

24. dubna 1999

Zapojil jsem telefon do místa, kde dříve býval mixér. Někomu jsem zavolal. Ozvalo se „Aáááááá...“. (Steven Wright)

1. Perl

```
$skalár                $skalár = 'něco';
@pole   $pole[0]       @pole = (1,'něco',undef);
%hash   $hash{key}     %hash = (a => 1, b => 'c');

if     (podmínka) { něco; } else { jinak; }
unless (podmínka) { něco; } else { jinak; }
něco if podmínka;

while (podmínka) { něco; }    # until
něco while podmínka;

for (my $i = 0; $i < $max; $i++) { něco; }
for (@seznam) { něco; }

open FILE,"něco" or die "open: $!";
while (<FILE>) {
   chomp;
   @items = split /:/;
   if ($items[0] =~ /^item\d{2}/) {
      print "$items[0] => ",join ',',@items;
      print "\n";
   }
}
close FILE;
```

```
sub callback { print "Ahoj\n"; }

$odkaz = \$proměnná;
$odkaz = \@pole;
$odkaz = [ 1, 2, 3 ];
$odkaz = { a => 1, b => 2 };
$odkaz = \&callback;
$odkaz = sub { print "Ahoj\n"; }

print @$odkaz;

$objekt = new Objekt @parametry;
$objekt->metoda(@parametry);
metoda $objekt @parametry;

use Balík;
Balík::metoda(@parametry);
print $Balík::proměnná;
```

- `perl program`
- `perl -e 'kód'`
- `#! /usr/bin/perl`
- `-w`, `-T`, `-n`, `-p`, `-i`, `-d`, `-d:ptkdb`
- `use strict;`

2. CPAN

- Comprehensive Perl Archive Network
- stovky modulů, 16 kategorií
- `http://www.perl.com/CPAN`
- `ftp://ftp.fi.muni.cz/pub/perl`
- moduly ve formě `*.tar.gz`
- jednoduše (`CPAN.pm`):

```
perl -MCPAN -e shell
d /Modul/
install Modul
```

- ručně:

```
tar xfvz Modul.tar.gz ; cd Modul
perl Makefile.PL
make
make test
make install
cd .. ; rm -rf Modul/
```

- POD – Plain Old Documentation
- `man Modul` (`man Net::FTP`)

3. BSD sockety

- implementace BSD socketů na nejnižší úrovni (voláním C rozhrání) – `accept`, `bind`, `connect`, `getpeername`, `getsockname`, `getsockopt`, `listen`, `recv`, `send`, `setsockopt`, `shutdown`, `socket`, `socketpair`
- v případě chyby vrací `undef`
- modul `Socket` – další funkce pro práci se sockety (`socket.h`) – `sockaddr_in`, `sockaddr_un`, `inet_aton`, `inet_ntoa` a různé konstanty
- klient pro TCP
- server pro TCP (váže se na všechny rozhrání – `INADDR_ANY`, může se však vázat i na konkrétní – firewally, gatewaye)
- na straně serveru doporučen mód nakažení (taint–mode) – pomocí `perl -T`
- UDP – umožňuje např. současný broadcast nebo multicast na více počítačů (pro srovnání TCP i UDP verze sledovátka časů na několika strojích)
- modul `IO::Socket` pro zjednodušení práce
- pozor na blokující operace (vícevláknový server, multiplexování, neblokující POSIX)

4. TCP klient

```
use Socket;

my $remote = shift or 'localhost';
my $port = shift or 2345;
if ($port =~ /\D/) {      # název služby
   $port = getservbyname($port,'tcp');
}
die "Port neexistuje" unless $port;

my $iaddr = inet_aton($remote)  # IP adresa
   or die "Host $remote neexistuje.";
my $paddr = sockaddr_in($port,$iaddr);
my $proto = getprotobyname('tcp');  # č.prot.

socket(SOCK, PF_INET, SOCK_STREAM, $proto)
   or die "socket: $!";

connect(SOCK, $paddr) or die "connect: $!";

while (my $line = <SOCK>) {
   print $line;
}

close(SOCK) or die "close: $!";
```

5. TCP server

```
use Socket;

my $port = shift or 2345;
my $proto = getprotobyname('tcp');

socket(SERVER, PF_INET, SOCK_STREAM, $proto)
   or die "socket: $!";
setsockopt(SERVER, SOL_SOCKET, SO_REUSEADDR,
   pack("l",1)) or die "setsockopt: $!";

bind(SERVER,sockaddr_in($port, INADDR_ANY))
   or die "bind: $!";
listen(SERVER, SOMAXCONN) or die "listen: $!";

for (; my $paddr = accept(CLIENT, SERVER);
       close CLIENT) {
   my ($port, $iaddr) = sockaddr_in($paddr);
   my $name = gethostbyaddr($iaddr,AF_INET);

   print CLIENT "Ahoj $name, nyní je ",
      scalar localtime, "\n";
}
```

- vylepšení – více vláken (`fork()`)

6. TCP sledování času

```
use Socket;

my $iaddr = gethostbyname('localhost');
my $proto = getprotobyname('tcp');
my $port = getservbyname('time','tcp');
my $paddr = sockaddr_in(0,$iaddr);

$| = 1;

for my $host (@ARGV) {
   printf "%-24s ",$host;
   my $hisiaddr = inet_aton($host);
   my $hispaddr = sockaddr_in($port,$hisiaddr);
   socket(SOCKET, PF_INET, SOCK_STREAM, $proto)
      or die "socket: $!";
   connect(SOCKET,$hispaddr)
      or die "connect: $!";
   my $rtime = ' ';
   read(SOCKET,$rtime,4);
   close(SOCKET);
   my $histime = unpack("N",$rtime)
      - 2208988800;     # my máme méně o 70 let
   printf "%8d\n",$histime - time;
}
```

7. UDP sledování času

```
use Socket;       $| = 1;
my $iaddr = gethostbyname(hostname());
my $proto = getprotobyname('udp');
my $port = getservbyname('time','udp');
my $paddr = sockaddr_in(0,$iaddr);

socket(SOCKET, PF_INET, SOCK_DGRAM, $proto);
bind(SOCKET,$paddr);

my $count = 0;
for my $host (@ARGV) {
   my $hisiaddr = inet_aton($host);
   my $hispaddr = sockaddr_in($port,$hisiaddr);
   send(SOCKET,0,0,$hispaddr);   ++$count;
}

my $rin = "";  vec($rin,fileno(SOCKET),1) = 1;
while ($count and select(my $rout = $rin,
      undef, undef, 10)) {
   my $rtime = ' ';  --$count;
   my ($hispaddr = recv(SOCKET,$rtime,4,0));
   ($port,$hisiaddr) = sockaddr_in($hispaddr);
   my $host = gethostbyaddr($hisiaddr,AF_INET);
}
```


8. IO::Socket

```
use IO::Socket;      # server

my $sock = new IO::Socket::INET(
   LocalHost => 'localhost', LocalPort => 2345,
   Proto => 'tcp', Listen => 5, Reuse => 1 );
die "Socket nevytvořen: $!" unless $sock;

while (my $new_sock = $sock->accept()) {
   while (defined (my $buf = <$new_sock>)) {
      print $buf;
   }
}
close $sock;


use IO::Socket;      # klient

my $sock = new IO::Socket::INET(Proto => 'tcp',
   PeerAddr => 'server', PeerPort => 2345 );

for my $i (1 .. 10) {
   print $sock "Zpráva č.$i\n"; $sock->flush();
}
close $sock;
```

9. Vícevláknový server

```
use IO::Socket;

$SIG{CHLD} = sub { wait() };

my $main_sock = new IO::Socket::INET (
   LocalHost => 'localhost', LocalPort => 5432,
   Proto => 'tcp', Listen => 5, Reuse => 1 );

while ($new_sock = $main_sock->accept()) {
   $pid = fork();

   unless ($pid) {        # synovský proces
      while (defined ($buf = <$new_sock>)) {
         # něco se děje s $buf
         print $new_sock "Přijato: $buf\n";
      }
      exit(0);            # konec syna
   }

   # rodič se vrací na accept()
}

close $main_sock;
```


10. Multiplexování pomocí select

```
use IO::Socket;  use IO::Select;

# stvoříme $main_sock jako vždy
my $readable_handles = new IO::Select();
$readable_handles->add($main_sock);

while (1) {
   my ($new_readable) = IO::Select->select(
      $readable_handles, undef, undef, undef);
   for my $sock (@$new_readable) {
      if ($sock == $main_sock) {
         my $new_sock = $sock->accept();
         $readable_handles->add($new_sock);
      } else {
         $buf = <$sock>;
         if ($buf) {
            # nějaká akce s $buf
         } else {
            $readable_handles->remove($sock);
            close $sock;
         }
      }
   }
}
```

11. Hotové klientské moduly

- `Net::Ping` (ICMP, TCP, UDP, echo)
- `Net::FTP` (active, passive, dataconn)
- `Net::Telnet` (commands, buffer, pty)
- `Net::SMTP` (SMTP, ESMTP)
- `Net::Time` (time, daytime)
- `Net::NNTP` (reader, post, slave news server)
- `Net::Ident` (BSD socket lookup)
- `Net::POP3` (POP3, APOP3)
- `Net::SNPP` (paging protocol)
- `Net::Finger`
- `Net::DNS` (resolver, zone backup)
- `Net::TFTP`
- `Net::IMAP` (ve vývoji, málo dokumentace)
- `Net::SNMP` (get, set request)
- . . . a další

12. Net::Ping, Net::FTP

```
use Net::Ping;

my $ping = new Net::Ping 'icmp';
for my $host (@host_array) {
   print "$host is ";
   print "NOT " unless $ping->ping($host,2);
   print "reachable.\n";
   sleep(1);
}
$ping->close();
```

```
use Net::FTP;

my $ftp = new Net::FTP 'ftp.fi.muni.cz';
die "Nelze se připojit: $!" unless $ftp;

$ftp->login('anonymous','sorm@fi.muni.cz');
$ftp->cwd('/pub/perl');
$ftp->binary();
for my $file ($ftp->dir()) {
   $ftp->get($file,"mirrored_$file");
}
$ftp->quit();
```

13. Net::Telnet, Net::SMTP

```
use Net::Telnet;

my $telnet = new Net::Telnet (
   Timeout => 10, Prompt => '/bash\$ $/');

$telnet->open('aisa');
$telnet->login($login,$passwd);
@lines = $telnet->cmd('/usr/bin/whoami');
print @lines;
$telnet->close;


use Net::SMTP;

my $smtp = new Net::SMTP 'relay.muni.cz';
$smtp->mail($ENV{USER});
$smtp->to('postmaster');
$smtp->data();
$smtp->datasend('To: postmaster\n');
$smtp->datasend('Subject: Čau poštmistře\n');
$smtp->datasend('');
$smtp->datasend('Pozdrav z Prudké !\n");
$smtp->dataend();
$smtp->quit();
```


14. Net::Time, Net::NNTP

```
use Net::Time qw(inet_time inet_daytime);

print inet_time('time.fi.muni.cz','tcp');
print inet_daytime('time.fi.muni.cz');


use Net::NNTP;  use Mail::Internet;

my $nntp = new Net::NNTP 'news.fi.muni.cz';
$nntp->reader();
my $rlist = $nntp->newsgroups();
print join "\n",@$rlist;
print "-" x 80, "\n";

my $rart = $nntp->newnews('cz.muni.fi-unix',
   time() - 86400);       # 1 den dozadu
for (@$rart) {
   my $ref = $nntp->article();
   my $mail = new Mail::Internet($ref);
   my $telo = $mref->body();
   print join "\n",@$telo;
   print "-" x 80, "\n";
}
$nntp->quit();
```

15. Net::Ident, Net::POP3

```
use Net::Ident;

# práce s BSD sockety, chceme ověřit SOCKET
my $username = (SOCKET->ident_lookup(30)
                or "UNKNOWN");


use Net::POP3;  use Mail::Internet;

my $pop3 = new Net::POP3 'pop3.email.cz';
$pop3->login($login,$passwd);
my $href = $pop3->list();
for my $msgnum (keys %$href) {
   my $ref = $pop3->get($msgnum);

   my $mail = new Mail::Internet($ref);
   my $telo = $mref->body();
   print join "\n",@$telo;
   print "-" x 80, "\n";

   $pop3->delete($msgnum);
}
$pop3->quit();
```


16. Net::SNPP, Finger, DNS

```
use Net::SNPP;
my $snpp = new Net::SNPP 'snpp.operator.cz';
$snpp->send( Pager => $my_pager_number,
             Message => 'Oběd je na stole.',
             Alert => 1,
             Hold => time + 3600) # za hodinu
          or die $snpp->message();
$snpp->quit():


use Net::Finger;
my @lines = finger('sorm@aisa.fi.muni.cz',1);


use Net::DNS;

my $dns = new Net::DNS::Resolver;
my $query = $dns->search('fi.muni.cz');
if ($query) {
   for $rr ($query->answer) {
      next unless $rr->type eq 'A';
      print $rr->address, "\n";
   }
} else { print $dns->errorstring, "\n"; }
```

17. Net::TFTP, IMAP, SNMP

```
use Net::TFTP;
my $tftp = new Net::TFTP('pyrrha.fi.muni.cz',
                         BlockSize => 1024);
$tftp->ascii;  $fh = $tftp->get('boot.config');
while (<$fh>) { print; }
$tftp->close;


use Net::IMAP;
my $imap = new Net::IMAP 'aisa.fi.muni.cz';
$imap->login($login,$passwd);
print $imap->status('incoming');
$imap->quit();


use Net::SNMP;
my ($snmp,$error) = session Net::SNMP(
   Hostname => 'kyklop.fi.muni.cz',
   Port => 161, Community => 'public');
my $sysUpTime = '1.3.6.1.2.1.1.3.0';
my $response = $snmp->get_request($sysUpTime);
unless ($respone) { ... }
printf "Uptime: %s\n",$response->{$sysUpTime};
$snmp->close();
```


18. LWP

```
use HTTP::Request;
use LWP::RobotUA;
use HTML::SimpleParse;
use URI::URL;

my $ua = new LWP::UserAgent;
$ua->agent('WebCopy/0.1 ' . $ua->agent);
my @TODO = ('http://www.linux.cz');  my @WAS = ();

while (my $now_URL = pop @TODO) {
   my $req = new HTTP::Request GET => $now_URL;
   my $res = $ua->request($req);
   if ($res->is_success) {
      process($now_URL,$res->content);
      push @WAS,$now_URL;
   } else { ... }
}

sub process {
   my ($url, $content) = @_;

   # použití parseru na získání URL
   # z $content --> @TODO
}
```

19. HTML::SimpleParse

```
$parser = new HTML::SimpleParse $content;
for ($parser->tree) {
   my $cnt = $$_{content};
   my $odkaz = '';
   if ($$_{type} eq 'starttag') {
      if ($cnt =~ /^A .*HREF="(.*?)"/i) {
         $odkaz = $1;
      } elsif ($cnt =~ /^IMG .*SRC="(.*?)"/i) {
         $odkaz = $1;
      } elsif ($cnt =~
           /^BODY .*BACKGROUND="(.*?)"/i) {
         $odkaz = $1;
      }
   }
   if ($odkaz) {
      my $uri = new URI::URL $odkaz;
      $uri = $uri->base($current_url)->abs;
      $odkaz = $uri->as_string;  my $je = 0;
      for (@WAS) {
         if ($_ eq $odkaz) { ++$je;  last; }
      }
      push @TODO,$odkaz unless $je;
   }
}
```


20. Mail::Internet, Mail::Folder

```
use Mail::Internet;
use Mail::Folder;

my $folder = new Mail::Folder('mbox',
     'Mail/newsy',Create => 1);

for (@$rart) {
   my $ref = $nntp->article($_); # z dřívějška

   my $mref = new Mail::Internet($ref);
   my $telo = $mref->body();
   my $hlavicky = $mref->head();
   $hlavicky->replace('Cc','gate@muni.cz');
   $hlavicky->replace('X-Reader','Perl/Read');

   $mref = new Mail::Internet(
   Header => $hlavicky, Body => $telo );

   $folder->append_message($mref);
}

$folder->sync();
$folder->close();
```

21. Vhodná literatura a odkazy

- Larry Wall, Tim Christiansen, Randal L. Schwartz: Programování v jazyce Perl (CPress; ISBN 8085896958)
- Sriram Srinivasan: Programování v Perlu pro pokročilé (CPress; ISBN 8072260790)
- `man perlipc`
- knihy a články o programování v sítích v UNIXu obecně (např. Andrew S. Tanenbaum: Computer Networks; Prentice-Hall, 1996)
- CPAN, moduly v kategorii věnované sítím, adresář `Net` a README k těmto souborům, případně jejich manuálové stránky (dokumentace v POD)
- Povídání (tato přednáška), které vyjde již brzo v Linuxových novinách (`http://www.linux.cz/noviny`)