

Корисні функції Perl

regex-url <https://perldoc.perl.org/perluniprops>

I remember seeing this awhile ago on this list and saved it:

```
#bin2dec:
$out = unpack("N", pack("B32", substr("0" x 32, $in, -32)));
#bin2hex:
$out = unpack("H8", pack("B32", substr("0" x 32, $in, -32)));
#bin2oct:
$out = sprintf "%o", unpack("N", pack("B32", substr("0" x 32, $in, -32)));
#dec2bin:
$out = unpack("B*", pack("N", $in));
#dec2hex:
$out = unpack("H8", pack("N", $in));
#dec2oct:
$out = sprintf "%o", $in;
#hex2bin:
$out = unpack("B32", pack("N", hex $in));
#hex2dec:
$out = hex $in;
#hex2oct:
$out = sprintf "%o", hex $in;
#oct2bin:
$out = unpack("B32", pack("N", oct $in));
#oct2dec:
$out = oct $in;
#oct2hex:
$out = unpack("H8", pack("N", oct $in));
#hex2ansii
$out = pack('H*', $in);
#is ipv4
```

```
my $ok = '10.10.255.1' =~ '\b((25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]? )(\.|\$)){4}\b'? 1: 0;

sub ip2bin {
    my $ip = shift;
    my $bin = join(' ', map substr(unpack("B32",pack("N",$_)),-8), split(/\./,$ip));
    return $bin;
    return unpack('B32', pack('C4C4C4C4', split(/\./, $ip)));
}

sub bin2ip {
    my $bin = shift;
    return join '.', unpack('C4C4C4C4', pack('B32', $bin));
}

sub ip2num {
    my $ip = shift;
    return unpack("N",pack("C4",split(/\./,$ip)));
    return unpack('N',pack('C4',split(/\./,$_[0])));
}

sub num2ip {
    my $num = shift;
    my $res = (($num>>24) & 255) ".". (($num>>16) & 255) ".". (($num>>8) & 255) ".". ($num & 255);
    return $res;
}

# decode utf like \x{42f}
print Dumper(%mydata) =~ s/\\x\{([0-9a-f]{2,})\}/chr hex $1/ger;
#https://stackoverflow.com/a/50511147/14644785

# regex ip
^( (25[0-5]|2[0-4]\d|1\d{2}|0?[0-9]?\d) (\.|\$) ){4}\b

# regex vlan
```

```
^(409[0-5]|40[0-8]\d|[1-3]\d{3}|[1-9]\d{0,2}?)\b
```

```
find /home/user -type f | xargs perl -p -i -e "s/\t/ /gmi"
```

Обидва оператори `i ||` можуть використовуватися для визначення значення за замовчуванням в Perl, але вони мають різні особливості. (defined-or): Цей оператор перевіряє, чи змінна визначена (defined). Якщо вона не визначена (undef), то вона буде замінена значенням справа від `.` `perl Copy code my $zminna = $potentsiyno_ne_znahodzhena_zminna 'значення_за_замовчуванням';`

(logical-or): Цей оператор перевіряє істинність значення ліворуч від

`.` Якщо це значення "істинне" (true), то воно повертається, інакше повертається значення справа від

`perl Copy code my $zminna = $potentsiyno_ne_znahodzhena_zminna || 'значення_за_замовчуванням';` Якщо вам важливо перевірити, чи змінна визначена (defined), та використовувати значення за замовчуванням тільки в разі її відсутності, то краще використовувати `.` *Якщо вам просто потрібно отримати значення за замовчуванням, якщо ліва частина "неправда" (false), то можна використовувати `||`.*

From:
<https://ndp.pp.ua/> - my NoDeny Wiki

Permanent link:
<https://ndp.pp.ua/doku.php/perl/helpfull>

Last update: **08/12/2023 13:28**

