

Skriptaus

Riku Saikkonen

28. helmikuuta 2006

Sisältö

Mitä skripteillä tehdään ja mitä ei?

Skriptikieli ohjelmointikielenä

Perinteiset skriptikielet + esimerkki (shell/AWK)

Säännölliset lausekkeet

Uudempia skriptikieliä

Tietoa Perlistä + esimerkkejä

Mistä saa tietoa Perlistä?

Lisäys työkaluluentoon

Yksi tärkeä työkalupari unohtui: Jäsentimien tekemiseen käytetään hyvin yleisesti näitä:

- lex tai GNU flex: tekee konfiguraation perusteella selaimen (lekserin) eli tuottaa C-ohjelman, joka lukee merkkijonon ja jakaa sen osiin
- yacc tai GNU bison: tekee annetusta kieliopista jäsentimen (parserin) eli tuottaa C-ohjelman, joka lukee selaimen tulostetta ja jäsentää sen jäsenyspuuksi (tai vastaavaksi)
- useimmille muillekin kielille kuin C:lle on vastaavat työkalut

Mitä skripteillä tehdään? 1/2

Esimerkiksi:

- tekstimuotoisen datan käsittelyä
 - järjestämistä, laskuja, tilastoja (kuten taulukkolaskennassa)
 - muutetaan muodosta toiseen esim. tietyn ohjelman syötteeksi
 - esim. ”lue ohjelman tuloste ja tee siitä HTML-sivu”
- kertakäyttöisiä ohjelmia
- komentojen automatisointia
 - ylläpitotyökaluja, varmuuskopiointia, ...
 - esim. kuvatiedostojen automaattista muokkausta

Mitä skripteillä tehdään? 2/2

- isojen ohjelmien konfigurointia tai laajennusta
 - esim. Emacs, Make, CVS, Gimp
- WWW-sivuja tai niiden osia
- uusia komentorivityökaluja
- joskus isojakin ohjelmia (ei ehkä kannata)
- useammin isojen ohjelmien yhteenliimausta

Skriptikieli ohjelmointikielenä

Skriptikielissä on yleensä:

- joustavat tietotyypit (esim. luku ja merkkijono samassa)
- ei rajoituksia tietotyyppien koolle
- lyhyt koodi (lyhennysmerkintöjä, ei pakollisia määrittelyjä)
- säännöllisiä lausekkeita käsittelevää syntaksia
- joustava syntaksi, paljon vaihtoehtoista syntaksia
- mahdollisuus muiden ohjelmien ajamiseen kätevästi
- paljon valmiita kirjastofunktioita

Skriptikieli on *kätevä*, mutta siinä on helpompi tehdä virheitä.

Mitä skripteillä ei kannata tehdä?

- isoja, monimutkaisia ohjelmia
- paljon muutoksia vaativia ohjelmia
- (yleensä) useamman ihmisen ylläpitämiä ohjelmia
- nopeutta vaativaa laskentaa

Perinteiset skriptikielet

- shell eli sh-skriptit: Unixin komentotulkki ja Unix-työkalut
- Unixiin kuuluu työkaluajattelu: ”ohjelman on parempi tehdä yksi asia hyvin kuin monta asiaa kohtalaisesti”
 - esim. ispell-ohjelma tekee pelkästään tekstin oikolukua, ja tekstieditorit käyttävät sitä eivätkä toteuta omaansa
 - shell-skripteissä yhdistellään kymmeniä pieniä ohjelmia hyvinkin isoiksi kokonaisuuksiksi
 - työkaluajattelu on hyvä pitää mielessä myös omia skriptejä tehdessä: jaa skripti useampaan osaan
- sed: vanha ja omituinen skriptikieli tekstidatan käsittelyyn
- AWK: pieni skriptikieli C-tyyppisellä syntaksilla
 - tehty sarakkeisen tekstin käsittelyyn
 - käytetään usein shell-skriptien jatkeena

Säännölliset lausekkeet

- käytetään kaikissa skriptikielissä ja usein muuallakin
- tavoite: määrittelee hahmon, jota etsitään tekstistä
- perussyntaksi: kirjaimet ym. vastaavat itseään
 - `.` vastaa mitä tahansa merkkiä, `\.` vastaa pistettä
 - `[a-z]` vastaa yhtä näistä, `[^a-z]` mitä tahansa muuta
 - `*` sallii edellisen toistamisen $0..∞$ kertaa (sulut ryhmittelevät)
 - `?` sallii edellisen toistamisen 0 tai 1 kertaa
 - `foo|bar` on tai-operaattori (syntaksi vaihtelee)
 - `^` rivin alku, `$` rivin loppu
- esimerkkejä: `foo` `^foo$` `^[0-9]*:` `[A-Z][a-z]*`
`[+-]?[0-9][0-9]*(\.[0-9][0-9]*)?`
- eri kielissä on usein hieman eri syntaksi ja eri lisäominaisuudet

Shell/AWK-esimerkki

```
#!/bin/sh
# Mark diffs between files in a new file (diffpp clone)
# Syntax: mark-diffs from-file to-file | enscript -eξ ...
#       or enscript -eξ --filter='mark-diffs ../old/%s %s' ...
if [ -f "$1" ]; then
    diff -b -U 10000 "$1" "$2" |
    gawk '
NR <= 3 { next; }
/^+/ { print "§shade{0.8}" substr($0,2) "§shade{1.0}" ; next }
/^-/ { next }
/^ / { print substr($0,2); next }
{ print "error: $0"; exit 1 }'
else
    echo "§shade{0.8}      (this is a new file)      §shade{1.0}"
    cat "$2"
fi
```

Uudempiä skriptikieliä

- Perl
 - C/AWK/shell-tyyppinen syntaksi
 - isompi kieli: ei tarvitse ulkoisia työkaluja
 - hyvin paljon käytetty, monia lisäkirjastoja
- muita yleiskäyttöisiä skriptikieliä: Python, Ruby, ...
- erikoistuneita skriptikieliä: PHP, Javascript, Emacs Lisp, Gimpin Scheme, taulukkolaskentaohjelman makrot (?)
- muutamissa varsinaisissa ohjelmointikielissä on skriptiominaisuuksia: mm. Haskell, jotkut Schemet

Perl 1/2

- iso, paljon käytetty skriptikieli
- syntaksi yhdistelmä mm. C:stä, AWKista ja shell-skripteistä
 - ”kaiken voi tehdä monella tavalla”
 - joustava, yrittää sallia luonnollista kieltä muistuttavan ”lauserakenteen”
 - Perl-koodi on usein vaikealukuista, mutta selkeääkin voi tehdä
 - turhan helppo tehdä virheitä: hiukan väärin kirjoitettu koodi tekee usein jotain ihan muuta virheilmoituksen sijaan
- automaattinen muistinhallinta (tosin voi vuotaa muistia)
- tietotyyppejä: kokonaisluku, liukuluku, merkkijono, taulukko, hash-taulu, viittaukset (ym.)

Perl 2/2

- melko kokonainen kieli, ei tarvitse aputyökaluja
- myös moduulijärjestelmä ja oliot (käytetään harvoin)
- erittäin paljon lisäkirjastoja, esimerkiksi:
 - Unixin C-kirjastofunktiot
 - useita graafisia käyttöliittymäkirjastoja
 - HTML:n käsittelyä ja WWW-serveriliittymiä
 - tietokantarajapintoja
 - kuvankäsittelyä ja viivadiagrammien piirtoa
 - verkkoprotokollien toteutuksia

Pieni Perl-esimerkki

```
#!/usr/bin/perl -w
my $i = 0;
while ($rivi = <>) {
    my ($a, $b, $c) = split(/:/, $rivi);
    $i = $i + $b + $c;
    print "$a $i\n";
}
```

Pidempi Perl-esimerkki

```
#!/usr/bin/perl -w

open(LUKUJ, "lukuj.txt") or die "Can't open lukuj.txt: $!\n";

while ($rivi = <LUKUJ>) {
    my ($kurssi, $tyyppi, $vkp, $ajat, $sali) = split(/ +/, $rivi);
    if ($tyyppi =~ /^L/) {
        my ($alkua, $loppua) = split(/-/, $ajat);
        $luentoaika{$kurssi} += $loppua - $alkua;
    }
}

foreach my $kurssi (sort keys %luentoaika) {
    print "$kurssi $luentoaika{$kurssi}\n";
}
```

Perl-ohjeita

- kirjoja:
 - R. Schwartz et. al., *Learning Perl*, 2nd ed., O'Reilly 1997
 - L. Wall et. al., *Programming Perl*, 2nd/3rd ed., O'Reilly 1996/2000
- verkossa on useita Perl-oppaita, esim.
<http://www.ebb.org/PickingUpPerl>
http://learn.perl.org/library/beginning_perl/
eräs luettelo:
http://www.perl.com/cs/user/query/q/6?id_topic=74
- koneelta:
 - man perl ja sen alisivut on hyvä referenssi
 - perldoc vastaava, esim. `perldoc -f split`