

Perl

АНТОН ДЕДИКОВ

Увод: **Perl** е универсален, интерпретативен език за програмиране, създаден от Лари Уол през 1987 г. Лари е трябвало да създава отчети за системата, която е поддържал тогава и не е имало подходящ инструмент в момента (програмата `awk` не е можела да отваря и затваря файлове въз основа на информацията в тях), а не му се е искало да създава специализиран инструмент. Така на бял свят се е появил нов език.

Име

Първоначално името на езика е било `Pearl` (произнася се по същия начин - „Perl“). Още преди да излезе първата версия на езика, Лари установява, че съществува език за програмиране на име `PEARL`. С цел да се избегнат конфликти, а и повлиян от характерната за философията на `Unix` тенденция към кратки имена, Лари променя името на езика на `Perl`, без да променя произношението му.

Впоследствие се появяват интерпретации на името като съкращение, най-популярната от които е **Practical Extraction and Report Language** (в превод: практичен език за извличане и отчети). Съществуват и много други интерпретации, включително и такива от самия Лари Уол, например шеговитото **Pathologically Eclectic Rubbish Lister** (в превод: патологично многостранен изборител на глупости). Всички те обаче са бекроними. Името на езика не е съкращение от нищо, поради което се пише само с една главна буква - `Perl`, а не `PERL` (за разлика от споменатия език `PEARL`, чието име наистина е съкращение).

Съществува също така тънка разлика в значението в зависимост от това дали името е изписано с главна буква в началото или изцяло с малки букви. `Perl` е името на езика за програмиране, което, както (почти) всяко друго име, се пише с главна буква. Под `perl` се има предвид интерпретатора на езика `Perl`, т.е. програмата, която се

стартира, за да се изпълни даден код на `Perl`. Името на интерпретатора спазва неписаното правило имената на команди под `Unix` да са изцяло с малки букви.

Свойства на Perl

В `Perl` са заимствани концепции от доста езици - `C`, `awk`, `sed`, `Lisp` и др. Най-силните му черти са регулярните изрази (англ. `regular expression`, често съкращавано на `regex`), вградените сложни структури от данни (обикновени и асоциативни масиви) и един от най - големите в света набори от свободно достъпни модули `CPAN`. На `Perl` може да се пише процедурно, обектно - ориентирано и функционално (поддържа обвивки, една от най - мощните абстракции в компютърната наука). `Perl` е слабо типизиран език. Той е интерпретиран, като програмата първо се компилира до машиннонезависими инструкции (байткод), които се изпълняват от интерпретатора. За разлика от `Java` обаче, байткодът не е лесно достъпен, благодарение на което са избегнати редица проблеми с обратната съвместимост. Интерпретаторът на `Perl` е написан на `C` и е преносим на огромен брой платформи и операционни системи. Програмата `perl2exe` и модулет `pp` могат да бъдат използвани за произвеждане (генериране) на изпълними програми от скриптове на `Perl`.



Предимства:

Обработка на текст

Най - силната страна на `Perl` е обработката на текст. Това го прави изключително популярен сред системните администратори. Операции като отваряне, четене, синтактичен разбор (известен още като „парсване“), търсене, заместване на текстови файлове са кратки и лесни за програмиране, имплементирани са много ефективно и се изпълняват по - бързо отколкото в повечето други езици.

Динамични уеб страници

Разработчиците на динамични уеб страници (страниците, чиито съдържание се произвежда на момента на тяхното извикване - т. е. динамично) често използват Perl поради големия брой безплатни скриптове, както и общността на разработчици предлагащи помощ на своите колеги. Богатата библиотека с модули, споменатата мощна текстообработка, която често се налага при генерирането на HTML страници, както и фактът, че е слабо типизиран и интерпретиран език, правят възможна бързата разработка на приложения в много области, включително и настолни графични приложения.

Фактът, че Perl се интерпретира в момента на изпълнение, налага уеб сървърът да стартира на интерпретатора perl за всяка уеб страница, която трябва да се генерира. При по - натоварени сайтове това може да доведе до значително натоварване на системата, както откъм памет, така и откъм процесорно време. За да се избегне това, специално за Perl е създаден модула **mod_perl** за сървъра **Apache**, който на практика представлява интерпретатор за Perl, вграден в кода на Apache. Това позволява на уеб сървъра да изпълнява код на Perl, без да се налага да стартира нов процес за всяка заявена уеб страница. Това води до значително увеличение на производителността при намалени ресурси. Mod_perl отива дори още по-надалеч и дава на скриптовите на Perl достъп до всички „вътрешности“ на самия Apache сървър, като им позволява да се намесват във всеки един етап от доставянето на HTML страницата, както и да използват конструкции на Perl в конфигурационните файлове на Apache.

Общността CPAN

CPAN (Comprehensive Perl Archive Network, в превод Изчерпателна мрежа от архиви на Perl) представлява огромна колекция от софтуер написан изцяло или отчасти на Perl. Основната градивна единица на този софтуер е модулет. Съществуват незадаждателни, но препоръчителни (и често спазвани на практика) указания за писането на модули за Perl, с необходимата им документация, която често с простотата и стандартността си превъзхожда документацията на много комерсиални продукти. Разработчиците на Perl създават програмни модули, разрешаващи разнообразни проблеми за почти всички операционни системи и дори програмни езици.

Тези модули обикновено са достъпни за използване под същия лиценз, както и самия Perl (т. нар. Artistic License), което допринася за по - нататъчното разрастване на общността. Всички тези модули могат да бъдат разледани и свалени от сайта на CPAN

Използване на командния интерпретатор

Името на главният изпълним файл (т.е. интерпретатора на езика, който превръща текстовия код на Perl в изпълними инструкции за процесора) е същото като името на програмния език, но обикновено се пише с малка буква - **perl** в *nix системи или **perl.exe** в Windows системи. Този изпълним файл може да се използва за бързи изпълнения от командния ред на компютъра, който биха отнели много редове (и вероятно скорост) ако биха били изпълнени на други програмни езици. Следият списък илюстрира тези примери.

- #отвори всички файлове от типа *.c и промени foo на barperl -p -i.bak -e 's/bfoo\b/bar/g' *.c
- # принтирай първите 50 страници евтино perl -pe 'exit if \$. > 50' f1 f2 f3 ...
- # изтрий първите 10 реда на файла foo.txtperl -i.old -ne 'print unless 1 .. 10' foo.txt
- # изтрий всички други редове освен тези между START и END perl -i.old -ne 'print unless /^START\$/ .. ^/END\$/' foo.txt

Недостатъци на езика

Противниците на Perl твърдят, че той има много неясен и дори откровено грозен синтаксис. Редицата синтактични съкращения, целящи пестене на време при писане на код, водят до безкраен брой от сложни за запомняне символи и структури. Мотото на езика „Има повече от един начин да го направиш“ се счита за основата на хаос от течения, които елиминират логичността и догматичността, които би трябвало да определят развитието на един програмен език. Комбинирано със слабата типизация (която по принцип е една от силните страни на Perl), всичко това позволява да се пише код, който по - късно е неразбираем дори за самия автор. Поради това някои наричат Perl език „само за писане“ (write-only).

Един от начините да се избегне това, е да се спазва някакъв основен набор от правила и конвенции при писането на код, които да са общоприети от всички. Фактът, че Perl до голяма степен не налага такива правила, обаче води до възникването на много стилови „течения“ сред програмистите на Perl, като за хората следващи даден стил е трудно да разберат код, писан в друг стил.

Основни типове данни

Най-използваните типове данни на Perl са:

Скаларна променлива - Единична стойност, която може да бъде число, низ или препратка. За по-кратко често се нарича само "скалар".

Масив - Списък от скалари, означавани (т.е. номерирани) последователно, започвайки от 0. Поредният номер на даден елемент от масива се нарича индекс. Достъпът до отделни елементи от масива става единствено чрез индекси.

Хеш - Хешът (наричан още хеш - таблица или асоциативен масив) е множество от подредени двойки скалари. Обърнете внимание, че двойките помежду си нямат подредба или номерация, т. е. образуват математическо множество. Елементите на една конкретна двойка обаче са подредени, т. е. има първи и втори: Първият се нарича ключ (key), а вторият стойност (value). И двата скалара могат да бъдат произволни и няма нужда да са от един и същи тип. Достъпът до отделни стойности от хеша става чрез ключовете. В известен смисъл ключовете при хеша съответстват на индексите при масива. В това отношение хешът може да се разглежда като по - обща форма на масив, при която индексите не са ограничени до числа, а могат да бъдат произволни скалари. Тъй като ключовете се използват за идентификация на стойностите в хеша, те трябва да са уникални, т. е. да няма повторения. За разлика от тях, стойностите могат да се повтарят - на различни ключове могат да съответстват едни и същи стойности.

Файлов манипулатор - Наричан още файлов дескриптор (file descriptor или file handle), този тип променливи предоставя достъп до файл или друго поточно

устройство, като например мрежова връзка, тръба (pipe), и т. н. Файловият дескриптор може да бъде отворен за писане, четене или и за двете.

Подпрограма - Подпрограмата (subroutine) е парче код, на който могат да се предават аргументи, той извършва някакво действие с тях и/или пресмята резултат, и накрая евентуално връща стойност (обикновено пресметнатата стойност, или нищо, ако няма такава).

Инсталация

Perl е част от всяка Unix или Unix - подобна система. Това може да бъде проверено с изпълнението на командата perl - V в шела. За Windows програмната среда може да бъде безплатно свалена от сайта на ActiveState или сайта на Strawberry Perl - дистрибуция съдържаща dmake, която улеснява инсталацията на модули изискващи make под Unix. Всяка стандартна инсталация на програмната среда може да бъде допълвана с най - различни модули от сайта на CPAN, голяма част от които функционират на различни операционни системи.

Документация

Документацията на Perl изобилства от шеги и игри на думи, за разлика от сухия език на повечето компютърна литература, а годишните обръщения на Лари („State of the Onion“, каламбур с обръщението на президента на САЩ, „State of the Union“) са ненадминати образци на духовитост. Въпреки шеговитият си характер, документацията на езика е изключително пълна и най - важното общодостъпна. С командата perldoc <име на модул> се извежда помощта за съответния модул. Онлайн документацията изобилства с примери и указания дори и за начинаещи в езика.

Заклучение: Богатата библиотека с модули на Perl, мощната текстообработка, която често се налага при генерирането на HTML страници, както и фактът, че е слабо типизиран и интерпретиран език, правят възможна бързата разработка на приложения в много области, включително и настолни графични приложения. Недостатъкът му е, че няма точни правила и конвенции за писане и това затруднява програмистите.