Perl

Practical Extraction and Report Language
or Pathologically Eclectic Rubbish Lister

Free, interpreted language maintained by the Perl Porters
- Open source software

Created and maintained by Larry Wall
- Created while working on the NSA's "Blacker" project
- Studied linguistics and then computers

Initially designed as a "glue" language for the UNIX operating system
- Duct-tape of the Internet

Perl Now and Then

History

- Perl 1.0 was released to usenet's alt.comp.sources in 1987
  - Included 10 pages of documentation
- Perl 5 was released in 1994
  - Included over 1000 pages of documentation

Modern Perl

- The current maintenance version of Perl is 5.10
- The current experimental version of Perl is 6.0 (Topaz)
- Most development is being done by the Perl Development team
  - Strongly supported by its users
  - Perl Porters – core development team
- Overall design philosophy: “Common things should be easy; advanced things should at least be possible”
- Guiding motto – "There's more than one way to do it"

Perl features

Designed for problems which are about 90% text and 10% something else

Works with third-party databases like MySQL, Oracle, Sybase, Postgres, and many others through a database interface called DBI

Can work with HTML, XML, and other mark-up languages

Has Unicode support

Supports both procedural and object-oriented programming

No memory management issues

No pointers

Built-in debugger

Uses for Perl

File and directory manipulation

Scanning large amounts of data

- Regular expressions (regex's)

Data reduction language

- Creating and obtaining data and creating formatted reports on that data

Server-side programming

- CGI scripts
- With and without database programming

Platforms

Now runs on numerous platforms (UNIX/Linux variations, MS-DOS, Windows, Mac, ...)

Highly portable

Developed into a general-purpose programming language

Freely available and redistributable

- Open source, released under the GNU Copyleft License

Many ideas are borrowed from natural language

Availability and Support

- Source code (and lots more) available at CPAN site
  - Comprehensive Perl Archive Network
  - www.cpan.org
- Full documentation comes with Perl
  - perldoc command
- Binaries available for just about every operating system
  - Macs, Windows 9x/NT/2000/XP, Linux/UNIX and tons of operating system most people have never heard of
  - Windows binaries available at ActiveState website
    - www.activestate.com

Perl compilation

- Two phases to compiling a Perl program
  - Compiler phase
    - Converts source code into Perl opcodes (language understood by the Perl interpreter)
  - Execution phase
    - Perl interpreter interprets one opcode at a time (runs the program)
- What are the advantages and disadvantages of interpreters and compilers?

Things of interest (or maybe not)

Just a couple of Larry Wall quotes

I'm reminded of the day my daughter came in, looked over my shoulder at some Perl 4 code, and said, "What is that, swearing?"

It won't be covered in the book. The source code has to be useful for something, after all... :-)

If you want to program in C, program in C. It's a nice language. I use it occasionally.

Down that path lies madness. On the other hand, the road to hell is paved with melting snowballs.

No, I'm not going to explain it. If you can't figure it out, you didn't want to know anyway... :-)

JAPH - Just Another Perl Hacker

Signatures people attach to their postings that look strange but are legal Perl programs

```
$_="cker,rl haer PeanothJust ";1 while s/(.{5})(.{5})?/$_{$2}=$1,$2/e;print while $_=$_{$_};
```

```
$x=25;print substr(',rekcah lreP rehtona tsuJ',$x,1) while --$x >= 0
```

Running Perl programs

Programs are referred to as scripts

- What's the difference between a script and a program?

Perl is flexible on how the program is supplied

Can be written completely on the command line or a Perl script can be supplied in a separate file and or just the script itself can be invoked (sort of)

Running Perl programs

Completely on command line

perl -e 'program in apostrophes'

Command line with seperate script file - perl scriptFile

The most convenient way to invoke a script is to let the operating system find the interpreter for you.

- "Shebang" notation - the first line of the file has a special meaning
- Tells the operating system which program to use to run the script
- #! followed immediately by the path and program to execute
- -#!/usr/bin/perl
- Only works on UNIX/Linux systems (more to the point, doesn't work on Windows systems)
- Script must be executable

Finding perl

To find the perl compiler on a UNIX/Linux system use the which command.

The which command locates a file and returns it's path

On Gandalf "which perl" returns /usr/bin/perl

Making scripts executable

To alter the access privileges use the chmod (change mode) command

chmod zzz file

Three z's are owner, group (all students on Agassiz) and public (or world) permissions

r (4) – read

w (2) – write

x (1) – execute

Add together for permission for each group

chmod 700 gives the owner permission to read, write and execute and no one else has any permissions

chmod 744 allows the owner to read, write and execute the file and every one else in the world can view the file

Making scripts executable

To change specific file modes use the following notation

- u user (owner) of the file
- g group
- o others (public)
- a all (owner, group and public)
- = assign a permission
- + add a permission
- - remove a permission

chmod u+x filename – gives the owner execute permission

chmod g-rw filename – gives the read/write permissions from the group members