BATKY HOWELL

Perl Programming on UNIX

Perl began as a text-processing language, an extension to the popular but limited awk language. Perl evolved into a general-purpose programming language popular with web developers, database developers, and many other types of programmers. Perl is very strong at processing large amounts of data, including manipulation, analysis, validation, conversion, formatting, and reporting. It offers complete libraries for database access, web development, graphics programming, and other environmental requirements.

Batky-Howell's Perl Programming course for Unix/Linux teaches students the foundations of using Perl effectively in many application environments. In addition to teaching the basics, such as data types, operators, flow control, and subroutines, the course goes into great detail on using arrays and hashes for complex data manipulation, regular expressions for advanced text processing, and Perl's object-oriented features for modern OO programming practices. Students write many complete Perl programs in this course, which ensures that when they return to work they can become productive immediately.

Course Objectives:

- Program using all basic elements of Perl– data types, variables, operators, flow control, I/O.
- Select and design the most appropriate data structures (array, hash, etc.) for Perl applications.
- Take advantage of Perl's powerful quoting and interpolation mechanisms to write more effective code.
- Use regular expression pattern matching to search and manipulate large amounts of complex data.
- Improve program design and modularity with subroutines.
- Implement complex data structures through the use of references.
- Use packages and modules to create libraries, and to use thousands of existing libraries.
- Design and write object-oriented programs using Perl's extensive OO abilities.
- Read and write binary files for interchange with foreign systems or programs.

Audience: Programmers and system administrators.

Prerequisites: *Fundamentals of UNIX.* Experience in a high-level programming language, such as C, C++, or Java, is strongly recommended.

Number of Days: 5 days

1. **Course Introduction**
   Course Objectives
   Overview
   Suggested References
2. **Overview of Perl**
   What is Perl?
   Running Perl Programs
   Sample Program
   Another Sample Program
   Yet Another Example
3. **Perl Variables**
   Three Data Types
   Variable Names and Syntax
   Variable Naming
   Lists
   Scalar and List Contexts
   The Repetition Operator
4. **Arrays and Hashes**
   Arrays
   Array Functions

The foreach Loop
The @ARGV Array
The grep Function
Array Slices
Hashes
Hash Functions
Scalar and List Contexts Revisited

5. Quoting and Interpolation
String Literals
Interpolation
Array Substitution and Join
Backslashes and Single Quotes
Quotation Operators
Command Substitution
Here Documents

6. Operators
Perl Operators
Operators, Functions and Precedence
File Test Operators
Assignment Operator Notations
The Range Operator

7. Flow Control
Simple Statements
Simple Statement Modifiers
Compound Statements
The next, last, and redo Statements
The for Loop
The foreach Loop

8. I/O: Input Operations and File I/O
Overview of File I/O
The open Function
The Input Operator < >
Default Input Operator
The print Function
Reading Directories

9. Regular Expressions
Pattern Matching Overview
The Substitution Operator
Regular Expressions
Special Characters
Quantifiers (*, +, ?, {})
Assertions (^, $, \b, \B)

10. Advanced Regular Expressions
Substrings
Substrings in List Context
RE Special Variables
RE Options
Multi-line REs
Substituting with an Expression
Perl RE Extensions

11. Subroutines
Overview of Subroutines
Passing Arguments
Private Variables
Returning Values

12. References
References
Creating References
Using References
Passing References as Arguments to Subroutines
Anonymous Composers
The Symbol Table

13. Complex Data Structures
Two-dimensional Arrays in Perl
Anonymous Arrays and Anonymous Hashes
Arrays of Arrays
Arrays of References
A Hash of Arrays
A Hash of Hashes
And So On...

14. Packages and Modules
Packages
BEGIN and END Routines
require vs. use
Modules
The bless Function

15. Introduction to Object-Oriented Programming in Perl
What is Object-Oriented?
Why Use Object-Oriented Programming?
Classes, Objects, and Methods in Perl
Inheritance, the "is-a" Relationship
Containment, the "has-a" Relationship
Overloaded Operators
Destructors

16. **Binary Data Structures**
Variable-Length (Delimited) Fields
Variable vs. Fixed
Handling Binary Data
The pack Function
The unpack Function
The read Function
C Data Structures
17. **Multitasking with Perl**
What are Single and Multitasking?
UNIX Multitasking Concepts
Process Creation with fork
Program Loading with exec
File Descriptor Inheritance
How UNIX Opens Files
One-Way Data Flow - Pipes
Final Result - Page Viewing
18. **Sockets Programming in Perl**
Clients and Servers
Ports and Services
Berkeley Sockets
Data Structures of the Sockets API
Socket System Calls
Generic Client/Server Models
A Little Web Server