UNIX® PATTERN TOOLS

General:

This course presents the basic concepts and operation of UNIX commands which use pattern matching and regular expressions. Included in the course are explanations of the stream editor (`sed`), as well as the pattern scanning and processing language (`awk`). The course provides an understanding of how to use these powerful pattern matching tools for information retrieval, data manipulation, and report generation.

Objectives:

Upon successful completion of this course, the student will be able to:

- Use the stream editor `sed` as a filter within a UNIX shell pipeline.
- Describe and use special regular expression characters in `sed` addresses and substitutes.
- Specify the features of and uses for `awk`.
- List the components of an `awk` program.
- Define and code `awk` patterns and actions.
- Use `awk` built-in functions.
- Use `awk` arrays and associative memory.
- Code and execute a complete `awk` program which reads an input file and produces a simple report.
- Interface `awk` with other UNIX tools to perform data validation, extraction and reporting functions.

Audience:

Technical Users, Applications Programmers, and Systems Programmers.

Prerequisites:

Introduction to UNIX and Shell Programming courses or equivalent experience.

Duration:

Three (3) days including classroom lecture and lab sessions. When combined with the Shell Programming course, this course can be taught in 2 days for a total of 5 days.

PanTech
908-281-6759

UNIX PATTERN TOOLS
COURSE OUTLINE

I. **STREAM EDITOR SED**

- A. What is SED
- B. Commands
  1. General Syntax
  2. Specific Commands
  3. Substitute Command
- C. CUT Command
- D. Regular Expressions

II. **AWK INTRODUCTION**

- A. AWK Features
- B. Records and Fields
- C. Input Files
- D. Command Line
- E. Patterns and Actions
- F. Comments
- G. Simple `print` Statement

III. **SIMPLE ACTIONS AND PATTERNS**

- A. User Defined Variables/Constants
- B. Special Variables
- C. Operators
  1. Numeric
  2. Alternative
  3. Relational and Logical
- D. Regular Expression
  1. Operators
  2. Metacharacters
- E. Patterns
  1. Ranges
  2. Special Pattern `BEGIN`
  3. Special Pattern `END`
- F. Changing Field Separators

IV. **COMPLEX ACTIONS**

- A. Formatted Printing with `printf`
  1. Format Specification
  2. Output Format Control
- B. Operator Review
  1. Binary
  2. Assignment
  3. Unary
  4. Concatenation
- C. Variables and Fields
- D. Flow of Control Statements
  1. `if`
  2. `while`
  3. `for`
  4. `break`
  5. `continue`

V. **BUILT-IN FUNCTIONS**

- A. Function Format
- B. String Functions
  1. length
  2. index
  3. substr
  4. sprintf
- C. Mathematical Functions

VI. **ARRAYS AND SHELL**

- A. Arrays and Array Use
- B. `for` Statement and Arrays
- C. Associative Memory
- D. Current `FILENAME`
- E. `split` Function
- F. Additional Statements
  1. `next`
  2. `getline`
  3. `exit`
- G. NULL Record Separator
- H. Output Redirection
- I. Pipes and Pipelines
  1. Filters
  2. Internal Pipes
- J. Command Line Parameters
- K. Interaction with Shell

VII. **AWK CONCLUSIONS**

- A. Advantages
- B. Disadvantages
- C. Where AWK fits

PanTech
908-281-6759