

After working on UNIX systems for a period of time, most users become aware of more advanced “power tools” that can greatly increase their productivity, and can also provide the ability to accomplish additional application tasks that would be unfeasible to automate without such tools. Programmers, administrators, and others who use UNIX on a daily basis will benefit from learning and applying additional skills with UNIX commands and utilities including advanced vi, regular expressions, sed, awk, Perl, and more advanced shell programming features.

In this course, students learn and practice advanced skills in using these tools. They increase their productivity in UNIX by learning how to create powerful Korn shell scripts for processing text, managing files, and performing other complex tasks; how to use advanced vi features for more effective editing; how to apply regular expressions to many kinds of text-processing applications; and how to use sed and awk to filter and manipulate large text files. The course also provides an introduction to the Perl programming language.

Course Objectives:

- Take advantage of advanced features of the vi editor to increase productivity when editing files and programs.
- Use regular expressions to find patterns in text files. This allows you to locate specific files or text, display desired lines in files, or edit text by adding to it, deleting from it, or modifying it.
- Edit files using the streaming editor, sed.
- Write awk programs to create reports and edit text files.
- Design and write sophisticated shell scripts to automate common tasks.
- Explain advantages of Perl and read simple Perl programs.

Audience: Programmers, system administrators, database administrators, testers, or anyone who wishes to develop more advanced skills in UNIX.

Prerequisites: *Fundamentals of UNIX*

Number of Days: 4 days

<p>1. Course Introduction Course Objectives Overview Suggested References</p>	<p>4. Shell Interaction - Extending vi File Name Shortcuts in vi Invoking Shell Commands - ex Reading the Output of a Command Invoking Filters from vi</p>
<p>2. ex and vi Options ex and vi – Two Editors in One ex and vi – Options How to Set Options Within vi How to Set Options in .exrc</p>	<p>5. vi Macros What are Macros and Why?</p>
<p>3. vi Buffers The Unnamed Buffer Named Buffers</p>	

- The :map Command
- The vi Quote Mechanism
- Markers
- Executing Commands from a Buffer
- 6. Regular Expressions**
- What is a Regular Expression?
- Literal Regular Expressions
- Regular Expressions: ^, \$, \, . [s]
- Regular Expressions: *, \{m,n\}
- Subexpressions
- 7. Shell Programming**
- Filename Generation
- Named Parameters
- Positional Parameters
- Special Parameters
- Parameter Substitution
- Here Documents
- Shell Commands
- Command List Separators
- Control Flow - Conditionals
- The case Construct
- Control Flow - Loops
- The trap Command
- 8. Korn Shell Features**
- Aliases
- Command History
- Functions
- The print and read Commands
- The set Command
- 9. Introduction to sed**
- About sed
- Why Use sed?
- Invoking sed
- How sed Works
- 10. Using sed**
- sed Addressing
- sed Addressing - Formats
- sed Functions
- 11. Introduction to awk**
- Introduction to awk
- How awk Programs Work
- Running awk Programs
- 12. Awk Patterns**
- Summary of Patterns
- BEGIN and END
- Expressions

13

- String-Matching Patterns
- Extended REs in awk
- Range Patterns
- Overview of Perl**
- What is Perl?
- Running Perl programs
- Sample Programs