Mastering BASH Scripts

Course 9887 - 24 Hours

Overview

This course deals comprehensively with the capabilities of BASH and the utilities commonly used in the development of scripts. The training concentrates on a common approach to BASH on variants of UNIX, and currently covers versions of BASH up to 4.1.

On Completion, Delegates will be able to

- Use all significant features of the BASH language
- Implement advanced variable and array manipulation
- Develop shell scripts that contain advanced decision making constructs
- Use the tracing features to speed the development of shell scripts
- Implement advanced data stream handling by using process substitution
- Incorporate functions in script design
- Integrate BASH scripts with those written using other shells
- Write reliable, maintainable and efficient shell scripts

Who Should Attend

The course is aimed at programmers and analysts operating in UNIX environments using BASH, as well as UNIX system and network managers. This course is particularly suitable for those companies, which take a multi-vendor open systems view, and is applicable to all flavours of UNIX including Linux, HP-UX, Solaris, AIX, and so on.

Prerequisites

- Delegates should have a good working knowledge of Linux or UNIX systems, which includes a working knowledge of UNIX commands and utilities. This can be gained by attending our UNIX Fundamentals or Linux Essentials courses
- Delegates should also have a practical understanding of programming concepts such as variables, and decision-making constructs.

JOHN BRYCE
תלמד הי-טק. זה עובד!
matrix company

matrix

Course Contents

Day 1

Introduction and Revision

- Command-line structure
- Shell interpretation
- Filename expansion
- Variable and command expansions
- Quoting
- Redirection
- Here documents
- Pipes

Using BASH

- The Bash story (in brief)
- Bash personalities
- Shell options
- Bash shell options
- Start-up files
- Some shell variables
- Aliases and functions
- Command-line recall and editing
- Navigation for the impatient
- echo and help

Shell Scripts

- Command types
- Creating shell scripts
- Creating processes
- The exec command
- Exit status
- Using # in shell scripts
- Running shell scripts
- BASH_ENV
- Command groups and sub-shells

Handling Variables

- Legal variable names
- Creating variables
- The read command
- Exporting variables
- Special characters
- Checking the environment
- eval

Parameters and Friends

- Positional parameters
- Access to positional parameters
- $*, $@, and $#
- Changing positional aparmeters
- $* versus $@
- Other special variables

Behaviour Therapy

- Sending signals,
- Receiving signals and the trap command
- Shell tracing methods
- Tracing shell expansions
- DEBUG and ERR traps
- Dry run of a script
- The script command
- Error messages are your friends

Day 2

Flow Control

- Testing commands
- 'if' statements
- 'while' and 'until' loops
- Bourne shell test command
- Korn shell test command
- Bourne and Korn syntax compared
- Numeric tests

Variable Expansion

- Extracting a sub-string
- Deleting a sub-string
- Editing sub-strings
- Test and set/exit expansions
- The colon command

Lofty Variables

- Variable attributes
- Compound assignments
- Arithmetic conditionals
- Working with floating-point

More Loops

- List processing - 'for' loop
- Processing positional parameters
- 'break' and 'continue'
- Infinite loops
- The 'select' loop
- Reading in a loop from a file
- Reading in a loop from a program (process substitution)

Sophisticated Tests

- Statement short-cuts
- I/O redirection with if-then-fi
- The 'case' statement
- case flow control - Bash 4
- Shell pattern matching
- Extended pattern operators
- Locales and character classes

Using Functions

- Functions vs. scripts
- Creating functions
- Listing and removing functions
- Function parameters and return values
- Local and global variables
- Where to store functions
- Exported functions
- Using external function libraries

Day 3

Professional Scripts

- Parsing script options with getopts
- Handling errors
- Formatting output - printf
- Tips for shell programmers: programming for change
- Tips for shell programmers: programming for performance

Handling Arrays

- Arrays: the concepts
- Bash array syntax
- Array syntax gotchas
- Initialising arrays
- Appending to an array
- Finding array limits
- Using read with an array
- Using mapfile/readarray
- Associative arrays

Advanced IO

- The concepts
- Using file descriptors
- Block redirection
- Named pipes (FIFOs)
- Process substitution

Basic sed

- Running sed
- Basic regular expressions
- sed line addressing

Basic awk

- The awk programming model
- awk variables
- awk opeators
- Comparing awk with shell constructs