



**LINUX SHELL
SCRIPTING EĞİTİMİ
2 GÜN**



Digital Vizyon
Akademi

www.digitalvizyon.net



İçindekiler

Eğitim Hakkında.....	3
Neler Öğreneceksiniz?	3
Ön Koşullar	4
Kimler Katılmalı.....	4
Outline	4
Introduction.....	4
Review of Basics – optional	4
Shell Programming	5
Special Characters and Variables.....	6
Creating Menus	6
Selection	6
Iteration, Looping	7
Sub-scripts and Functions.....	7
Advanced Commands.....	7
Advanced Edit – vi & sed	7
Text File Processing – awk	8

Eđitim Hakkında

“Linux Kabuk Betiđi Programlama Eđitimi”, bir Linux sistemini kabuk betikleri ile yönetme ve otomatize etme yeteneđi kazandıran, uygulama odaklı bir eđitim programıdır. Bu beceri, bir sistem yöneticisi, veri bilimcisi, yazılım geliřtiricisi veya diđer teknoloji odaklı roller için son derece önemlidir. Linux kabuk betiđi, genellikle sistem otomasyonu, görevlerin otomatikleřtirilmesi, hızlı prototipleme ve veri işleme gibi senaryolarda kullanılmaktadır.

Bu eđitim programı, Linux sistemi kabuđunun temel prensiplerini, kabuk betikleme dilinin esaslarını ve bu dilin nasıl etkin bir şekilde kullanılacağını ayrıntılı bir şekilde öđretir. Katılımcılar, kabuk betiklerinin oluşturulması, kabuk deđişkenlerinin ve argümanların nasıl kullanılacağı, akış kontrolü, döngüler, işlemler, kabuk hata ayıklama ve kabuk betikleri için programlama yapılarının nasıl oluşturulacağını öğrenirler.

Eđitim ayrıca, kabuk araçlarını ve uygulamalarını sistem yönetimi, veri yönetimi, ađ yönetimi ve güvenlik gibi çeřitli gerçek dünya senaryolarında nasıl kullanacağını öđretir. Bunun yanı sıra, Linux kabuk betiđi programlama dilinin diđer programlama dilleri ile nasıl entegre olacağını ve işbirliđi yapacağını da öğrenirsiniz.

Linux kabuk betiđi programlama becerisi, çeřitli Linux dağıtımlarında ve Unix tabanlı sistemlerde kullanılabilir. Bu nedenle, bu eđitim, katılımcıların geniş bir yelpazede sistemler üzerinde kabuk betiđi kullanma becerisi kazanmalarını sağlar. Eđitim sonunda, katılımcılar, Linux kabuk betiđi kullanarak karmařık görevleri otomatize etme ve çeřitli sistem ve uygulama yönetim işlemlerini gerçekleştirme konusunda sağlam bir bilgi ve beceri sahibi olacaklardır. Bu beceriler, her tür teknoloji odaklı pozisyon için büyük bir avantaj sağlar ve katılımcıların işlerinde daha üretken ve etkili olmalarına yardımcı olur.

Neler Öğreneceksiniz?

Linux Shell Scripting eđitiminde řunları öğrenebilirsiniz:

- Shell programlama dillerinin temel kavramları
- Shell betiklerinde kullanılan komutlar ve dosya yapılandırması
- Veri yönetme ve işleme teknikleri
- Fonksiyonlar ve döngülerin kullanımı
- Sistem verilerine erişme
- Otomatik işlemler yapma
- Hata ayıklama ve hata mesajlarını okuma
- Shell betiklerinin test ve debug etme teknikleri

Bu eđitim, Linux ve Unix benzeri işletim sistemlerinde shell betikleri yazmayı öğrenmek isteyen herkes için uygun bir eđitimidir ve shell programlama becerilerini geliřtirmek isteyen katılımcılar için faydalı olacaktır.

Ön Koşullar

Linux Shell Scripting eğitiminin ön koşulları şunlardır:

- Linux veya Unix benzeri bir işletim sistemi hakkında temel bilgi: Eğitim, Linux veya Unix benzeri bir işletim sisteminde verilmektedir ve bu sistemler hakkında temel bilgiye sahip olmak önemlidir.
- Komut satırına erişim: Eğitim sırasında shell betiklerini yazmak için komut satırına erişim gerekir.
- Temel dosya sistemi ve dosya yapılandırması bilgisi: Eğitim, dosya sistemi ve dosya yapılandırması hakkında temel bilgi gerektirir.

Programlama dilleri hakkında temel bilgi: Eğitim, programlama dilleri hakkında temel bilgi gerektirir ve bu dillerin temel kavramlarını anlamak önemlidir.

Kimler Katılmalı

Linux Shell Scripting eğitimine:

- Linux veya Unix benzeri bir işletim sistemi kullanan kullanıcılar
- Komut satırına erişimi olan ve shell betikleri yazmak isteyen kullanıcılar
- Dosya sistemi ve dosya yapılandırması hakkında temel bilgi sahibi olan kullanıcılar
- Programlama dilleri hakkında temel bilgiye sahip olan kullanıcılar
- Otomatik işlemler yapmak isteyen sistem yöneticileri veya network operatörleri
- Linux shell betikleri hakkında daha fazla bilgi ve beceri kazanmak isteyen kullanıcılar

Bu eğitim, shell programlama becerilerini geliştirmek isteyen ve Linux veya Unix benzeri bir işletim sistemi kullanan herkes için faydalı olabilir.

Outline

Introduction

- Objectives
- Course Schedule
- Course Introductions
- Pre-requisites
- Bibliography
- Delegate Responsibilities
- Course Summary

Review of Basics – optional

- Files
- Directories and Processes



- Displaying Directories – ls
- Getting Help – man
- Displaying Files – cat, more
- Simple Commands
- Piping – using | symbol
- Redirection – using > symbol
- Redirection – using >> symbol
- Redirecting Input – using < symbol
- Redirecting Errors
- Using /dev/null
- Foreground Jobs
- Background Jobs – &
- Stopping Jobs – kill
- Special Files – . and ..
- The Command Prompt – PS1
- Creating Aliases
- Using Aliases
- Command history
- Command Line Edit – using Arrow Keys
- Command Line Edit – using vi
- Customising the Shell
- The Home Directory – HOME
- The Working Directory – pwd
- Changing Directories – cd
- Make Directory – mkdir
- Remove Directory – rmdir
- Interactive Copy – cp -i
- Interactive Move – mv -i
- Interactive Remove – rm -i
- Removing Directories – rm -r
- Wild Cards – File Name Generation
- Word count – wc
- Finding Files – find
- Finding Text in Files – grep, sort, cut, uniq, tr
- Changing Permissions – chmod

Shell Programming

- Creating Shell Scripts
- Testing the Command
- Creating the Script File
- Adding Execute Permissions
- Testing the Script
- Running the Script – time
- Running the Script – dot
- Running the Script – exec
- Running the Script – ksh
- Debugging the Script
- Passing Parameters



- The read Command
- Using read in Scripts
- Choosing the Shell
- Tidying The Output
- Suppressing Input – stty

Special Characters and Variables

- Quotation Marks
- Quoted Strings – “, ‘
- Escape Character – *
- Quoted Strings – `
- Using \$(...)
- System Variables – set
- Other Special Variables
- User Variables
- User Variables – unset
- Variables in Scripts
- Typeset Command
- Integer Variables
- Special Constructs

Creating Menus

- Conditions – case
- Creating a Menu
- Iteration – while, until
- Holding the Output – sleep
- Holding the Output – dummy read
- Ending the Script
- Ending the Loop
- Hidden Option – forking a shell
- Trapping Interrupts – trap
- More on Interrupts
- Highlighting Text- tput
- Positioning the Cursor – backtab \b
- Korn Shell Menu

Selection

- Conditional Testing
- Testing File Attributes
- Testing File Permissions
- Multiple Tests – Nested ifs
- Multiple Tests – elseif
- The NOT Condition
- Multiple Conditions in One Test
- Testing Strings
- Using Square Brackets
- Avoiding Null Input
- Testing Numbers
- More on case



- Displaying Variables
- Placing Spaces
- Testing Commands

Iteration, Looping

- Computations – let
- Iteration – while, until
- Infinite Loop
- Iteration – for
- More than 9 Parameters – shift

Sub-scripts and Functions

- Local and Global Variables
- Sub-scripts
- Export Command
- Exit Status
- Aliases
- Functions
- Useful Functions
- Return Status
- Manipulating Functions
- Functions in Scripts

Advanced Commands

- Debugging – set
- Setting – set
- Merging Files – join
- Merging Files – paste, basename and dirname
- Job Control – jobs
- Regular Expressions
- Global Regular Expression Printer – grep
- Expression grep – egrep
- Fast grep – fgrep
- Computations – expr
- Computations – bc, eval Command,
- Argument Processing
- getopt Command
- Using Arrays

Advanced Edit – vi & sed

- vi – Mapping Key Combinations
- Mapping Control Keys
- Using Abbreviations
- Editing Many Files
- Regular Expressions
- Global Insert
- Here document,



- Batch Stream Editor – sed, sed delete, -d, sed print, -p, sed suppress, -n, sed quit, -q, sed file, -f, sed Multiple Input, sed
- Find and Replace
- sed Example

Text File Processing – awk

- The awk Programming Tool
- Pattern Matching
- Regular Expressions
- Comparison Patterns
- Compound Patterns
- Range Patterns
- Special Patterns
- Built-in Functions
- Troubleshooting awk Scripts
- Passing Parameters
- Using Contains
- printf Function
- Computations
- Functions
- substr
- length
- Action Statements
- Control Flow Statements
- Arrays
- Arrays with for Loops
- awk Examples