

Linux Installation, Configuration, and Support

This is the Red Hat variant specific version of the course.

LESSON 1 - GETTING STARTED

There are no exercises for this chapter.

LESSON 2 - UNIX INSTALLATION

- Perform a standard installation on a server with a CD/DVD as the installation media.
- Confirm that the installation has been successfully completed.

LESSON 3 - SYSTEM MANAGEMENT TOOLS

- Configure the existing network card using the command line.

The following configuration steps need to be performed:

- The driver needs to be uncompressed.
- The driver then needs to be installed into the kernel.
- TCP/IP parameters to be set.
- hostname files created.
- network card is tested.

LESSON 4 - INSTALLATION TECHNIQUES

- Perform a complete installation on a server using a network drive as the installation media.
- Confirm that the installation has been successfully completed.

LESSON 5 - STARTUP AND SHUTDOWN

- Create customized scripts at boot time.
- Boot the system into single user mode and list the system daemons that are running.
- Boot the system into multi-user mode and compare the system daemons that are running to single user mode.

LESSON 6 - SYSTEM ADMINISTRATION TOOLS

- Use an administration GUI tool to create new users and groups.
- Use an administration GUI tool to configure the network connection.

LESSON 7 - XWINDOW SYSTEM

- Modify the system to automatically start in XWindows mode.
- Configure the XWindows application to run on a remote system using a X11 server.

LESSON 8 - RAID: REDUNDANT ARRAY OF INDEPENDENT DISKS

- Configure RAID for fault tolerance.
- Confirm that the RAID system is operation satisfactorily.

LESSON 9 - FILESYSTEMS

- Run fsck to confirm the operational status of a disk drive.
- Turn on the quota system to limit user disk usage.

LESSON 10 - KERNEL COMPILATION AND CONFIGURATION

- Modify kernel parameters, reboot the system and determine what changes has been made.

LESSON 11 - USER-LEVEL ADMINISTRATION AND SECURITY

- Add new users and groups using the command line and coding scripts.
- Use the Pluggable Authentication Module in order to allow a system administrator to add new authentication methods and to modify authentication policies by editing configuration files.

LESSON 12 - NETWORKING

- Update the httpd.conf file to modify the configuration of Apache.
- Test and confirm the changes.

UNIX System Administration

LESSON 1 - GETTING STARTED

There are no exercises for this chapter.

LESSON 2 - USER AND GROUP ADMINISTRATION

- Add new users and groups using the command line and programmatically using scripts.
- Turn on the quota system to limit user disk usage.
- Modify the user's environment to set each individual user session and have new environmental variables and a modified PATH.

LESSON 3 - ADMINISTERING LOGINS AND PERMISSIONS

- Search all files for a specified user and modify its ownership.
- Execute write and wall commands for contacting users.
- Create a special file (pipe) and determine its characteristics.
- Contact a user at login time.

LESSON 4 - UNIX FILE SYSTEM

- Determine the filesystems that are automatically mounted.
- Change the characteristics and options for a specified filesystem.

LESSON 5 - PHYSICAL DISKS

- Create a new partition, and format a new filesystem.
- Run the fsck utility to repair a filesystem.

LESSON 6 - DISK ADMINISTRATION

- Use df and du for determining file usage and problem areas.
- Create a logical volume using LVM.

LESSON 7 - OPERATIONAL STATES

- Create customized scripts at boot time.
- Boot the system into single user mode and list the system daemons that are running.
- Boot the system into multi-user mode and compare the system daemons that are running to single user mode.
- Modify kernel parameters, reboot the system and determine what changes have been made.

LESSON 8 - VIRTUAL MEMORY

- Use the vmstat utility to identify a memory problem area.

LESSON 9 - NETWORK FILE SYSTEMS

- Setup a NFS Servers and clients.
- Start and stop NFS.

LESSON 10 - PROCESS MANAGEMENT AND SCHEDULING

- Determine the processing which is using the most resources.
- Determine the resources that are being used.

LESSON 11 - BACKUP AND RECOVERY

- Execute a full backup of the system.
- Run an incremental backup of the system.
- Restore the system to a specific point.
- Backup all user home directories.

LESSON 12 - PRINTING

- Install a printer on a system.
- Schedule print jobs and diagnose print problems.

LESSON 13 - NETWORK APPLICATIONS

- Startup the FTP server.
- Determine which users can use the server and which are denied.
- Set up the view of the individual users.

UNIX Network Administration

LESSON 1 - GETTING STARTED

There are no exercises for this chapter.

LESSON 2 - ETHERNET IEEE IDENTIFIERS

- Given the client specific configuration of the network, determine the IP address and the netmasks.

LESSON 3 - VIRTUAL LAN

- A VLAN: Virtual LAN is a group of hosts with a common set of requirements that communicate as if they were attached to the same physical network.
- Perform a network reconfiguration and modification to the subnet.

LESSON 4 - ROUTING

- Create multiple networks connected via static routing.
- Configure the routers and the host.

LESSON 5 - TRANSPORT

- Install a program on two machines that communicates via Remote Procedure Calls.

LESSON 6 - COMMON SERVICES

- Startup the FTP server.
- Determine which users can use the server and which are denied.
- Setup up the view of the individual users.
- Remove the capability to use rsh and rlogin to improve security.

LESSON 7 - NETWORK TIME

- Set up the NCP: Network Time Protocol in order for synchronization to occur at all the computers on the network.

LESSON 8 - DHCP: DYNAMIC HOST CONFIGURATION PROTOCOL

- Create and configure a DHCP server.
- Set the IP address of the server's Ethernet card.
 - The address must be a static IP address.
 - Establish the subnet mask.

LESSON 9 - DNS: DOMAIN NAME SYSTEM

- Configure a DNS server.
- The DNS server will have to refer to itself for all DNS queries by configuring the `/etc/resolv.conf` file to reference localhost only.
 - The `/etc/named.conf` file contains the main DNS configuration and informs BIND where to find the configuration or zone files for each domain it owns.
 - Two zone areas will need be updated by the student:
 - Forward zone file definitions list files to map domains to IP addresses.
 - Reverse zone file definitions list files to map IP addresses to domains.

LESSON 10 - SNMP PROTOCOL

An SNMP-managed network consists of three key components: managed devices, agents, and NMSs - network-management systems.

- A managed device is a network node that contains an SNMP agent and that resides on a managed network.
- An agent is a network-management software module that resides in a managed device.
- An NMS executes applications that monitor and control managed devices.
 - NMSs provide the bulk of the processing and memory resources required for network management.
 - Each of the three components will need to be configured.

LESSON 11 - NFS: NETWORK FILESYSTEM

- Install the Windows Services for UNIX's NFS components in order to connect MS Windows to an NFS system.

LESSON 12 - TROUBLESHOOTING

- Based upon classroom specific instructions, browse the logs in order to determine problems.
- Use the `sar`, `vmstat`, `iostat`, and `dtrace` tools for determining the solution.

UNIX: Shell Programming

LESSON 1 - ENVIRONMENT

- Display and change the values in an environmental variable.
- Display the values in all environmental variables.
- Create a file (script) which supplied commands and run the script.

LESSON 2 - SHELL SCRIPT ELEMENTS

- Code a simple mathematical express.
- Code a basic loop and a conditional termination of the program.
- Special or builtin variables are defined by the shell for use at any time; the workshop will use two of the special variables.

LESSON 3 - CONDITIONAL PROGRAM, EXECUTION

- Code and test a compound conditional if statement.
- Code and test a nested if statement.
- Determine if a user is a superuser.

LESSON 4 - PROGRAM LOOPS AND ITERATION

- Code logic using a while statement and than convert it to a for statement.
- Develop techniques for breaking out of a loop.

LESSON 5 - STRING AND PATTERNS

- Code for a basic equal string comparison condition.
- Use regular expressions in a string comparison to locate files with a string containing a pattern.

LESSON 6 - STORING AND ACCESSING DATA

- Using sed, code a script to create an extract of a file.
- Code a script to read a file, and display the file with sequence numbers at the end of the line.
- Code a script to read a file, and updated specific records in the file.

LESSON 7 - SHELL VARIABLES

- Code a routine to determine if a SHELL variable exists.
- Code an expression and assign the results to a variable.
- Write a script that will echo the third parameter; if the parameter is present.
- Write a script that will recognize whether a word entered from the keyboard starts with an upper or lower case character or a digit.

LESSON 8 - PROCESSING DATA

- Search the system for larger files and run a script on each one of the files.
- Code a script to determine if the disk space is running low and if the memory usage is high.
- If either is true, send a message to the system administrator.
- Create a script that copies the file /etc/passwd file into the home directory.
 - Append the date and time to the last line of the backup file.
- From an administrator's perspective, it would be useful to know all usernames on the system and which shell each is using. Code a script to display the information.
- Given a file of numbers (one per line), write a script that will find the lowest and highest number.
- Write a script that will first verify if a file exists.

LESSON 9 - MODULAR PROGRAMMING WITH FUNCTIONS

- Create a function to display uptime.
- Code and test the home_space function will display the amount of space each user is using in his/her home directory.

UNIX: Programming

LESSON 1 - GETTING STARTED

- Open the IDE, code a simple program.

LESSON 2 - UNIX FACILITIES

- Using system calls, get the following information: the real user-ID, group-ID, effective user-ID, effective group-ID, process-ID, and the parent process-ID.

LESSON 3 - EXTERNAL INTERFACES

- External libraries are groups of executable C or C++ functions that can be dynamically accessed by applications.
- Invoke the two routines provide by the instructor.

LESSON 4 - SOCKETS

- Create a TCP socket stream and bind to the stream.
 - Handle the listen and accept functions.
- Create a client and server program that passing some information to the server, which performs simple calculations and returns the results.

LESSON 5 - STREAMS

- Using streams and the Standard I/O library, copy one file to another.

LESSON 6 - CLIENT/SERVER PROGRAMMING

- Interprocess communication between processes on the same machine or on different machines through TCP/IP sockets is a mode of point-to-point asynchronous communication.
- Code a server that converts all the strings into Proper Case strings.

LESSON 7 - PIPES

- Develop the exercise from lesson 6; however, instead of a TCP stream, use named pipes.
- Display a file by piping into a PAGER program.

LESSON 8 - PROCESSES

- Print current process resource limits.
- Invoke a process and determine if it is still running.
 - Kill the process and check if it is running.

LESSON 9 - SYSTEM V IPC

- Code a subsystem that would share application data with the UNIX System V IPC mechanisms.

LESSON 10 - SIGNALS

- Print the signal mask for current process.
- Call a blocking read with a timeout using longjmp.

LESSON 11 - SEMAPHORES

- Code a program to create a semaphore by using semget.
- Another task will query the value of the semaphore and take actions based on the value.
- Since semaphores have a count associated with them; they can be utilized when multiple threads need to achieve an objective cooperatively.

LESSON 12 - MESSAGE QUEUES

- Create and connect to a message queue.
- Create a key and send some data to the queue.
- Code a routine to receive the data.
- Code a routine to destroy the queue.

LESSON 13 - SHARED MEMORY

- A shared memory segment is an area of memory that can be allocated and attached to an address space.
- Processes that have this memory segment attached can have access to it.
- Code a client and server program that passes data back and forth.
- Handle the race condition.

LESSON 14 - I/O

- Code a routine to perform large non-blocking writes.

LESSON 15 - FILES AND DEVICES

- Use the C function ``ioctl'` (I/O control) for sending special control commands to devices such as the disk and the network interface.

LESSON 16 - MEMORY MANAGEMENT

- Create a file using memory mapped I/O.
- Read and write to the file.

LESSON 17 - FILE AND RECORD LOCKING

- Create a program which determines the presence of a deadlock situation by testing if a file region is locked by another process.
- The program should also resolve the deadlock problem.

Solaris Fundamentals

LESSON 1 - INTRODUCTION

There are no exercises for this chapter.

LESSON 2 - COMMANDS AND UTILITIES

- Based upon specifications providing by the instructor, code and run a simple script.
- Execute the following commands:
 - Passing arguments to the script coded previously.
 - Configure the environment for creating an alias.

LESSON 3 - FILE AND DIRECTORIES

- List the filesystems and determine the freespace and options.
- Determine the file permission of a group of files.
- Copy a file into a different directory.
 - Change the ownership of a file.
 - Create a list of files and redirect into a file.
- Use the grep utility for searching all files in a directory.
- Submit a background job.

LESSON 4 - UNIX COMMANDS

- Use the cut command for extracting the data from a file.
- Sort a file based on a specific field in the file.
 - Sort it in ascending and descending order.
- Compare two files and list the differences between the files.

LESSON 5 - vi TEXT EDITOR

- Using an existing file:
 - Change the data.
 - Insert lines; insert multiple lines.
 - Navigate the data by moving to the top, bottom, specific line or a search value.
 - Delete and change lines.
 - Move and update lines.
 - Save the file.
- Configure the vi environment.

LESSON 6 - EXECUTION ENVIRONMENT

- Change the profile file to set environment variables, aliases, and prompt for an individual user.
- Switch the users shells.
- Invoke the editing mode on the command line.

LESSON 7 - SHELL PROGRAMMING

- Code two scripts; a simple and more advanced script to perform calculations on passed arguments.
 - Use substitution to change the values in commands.
 - Execute statements base on a passed value and run loops based on a flag.
 - Run the script in background and confirm that the script executed properly.

LESSON 8 - CDE: COMMON DESKTOP ENVIRONMENT

- Invoke dtpad to edit files and use it as a substitute for vi.
- Invoke and use the Solaris Management Console to see the roles of users.

LESSON 9 - KORN SHELL PROGRAMMING

- Code and test several Korn Shell scripts to manipulate files and perform calculations.
- Write a script that will echo the third parameter, but only if it is present.
- Given a file of numbers (one per line), write a script that will find the lowest and highest number.
- Write a script that will recognize if a word entered from the keyboard starts with an upper or lower case character or a digit.
 - Use the "case" statement.
 - The script will output the word, followed by "upper case", "lower case", "digit", or "not upper, lower, or digit".

LESSON 10 - sed

- Execute the sed utilities for searching and modifying existing files.
- Code the utility using saved scripts.
- Convert a DOS (Windows) file to UNIX format.

LESSON 11 - awk PROGRAMMING

- Create a column delimited file with the following fields:
 - First Name
 - Last Name
 - Phone Number
 - Campaign Contributions

- The records will need to meet the conditions for performing the following exercises.
Enter the awk command that will print:
 - All the phone numbers.
 - Dan's phone number.
 - Susan's name and phone number.
 - All last names beginning with D.
 - All first names beginning with C or E.
 - All first names containing only four characters.
 - The first names of all those in the 916 area code.
 - Mike's campaign contributions preceded by a dollar sign (\$).
 - Last names followed by a comma followed by the first name.
 - awk will be easier than with sed.

Solaris System Administration

LESSON 1 - INSTALLATION OVERVIEW

There are no exercises for this chapter.

LESSON 2 - OPERATING SYSTEM CONCEPTS

- Invoke the Solaris Management Console and add users.
- Validate that the network configuration was successful and review user roles and privileges.

LESSON 3 - ACCESSING THE SYSTEM

- Write a script that lists all users and their associated IP address.

LESSON 4 - SYSTEM PROCESSES

- Run a background job that is in an endless loop.
 - Locate the PID and the PPID.
 - Attempt to kill the job.
 - If unsuccessful in killing the job, try another technique.
- List all the processes and determine which processes are using the most resources.

LESSON 5 - ARCHIVING USER DATA

- Tar and zip all the home directories.
- Determine which compress techniques create the smallest file.

LESSON 6 - USER ACCOUNT AND GROUP MANAGEMENT

- Create new groups and users.
- Modify users and delete users.
- Modify the shell for a user.

LESSON 7 - FILE TYPES AND DEVICES

- List the system devices and locate where they have been defined.

LESSON 8 - SOFTWARE PACKAGE ADMINISTRATION

- Add a new package to the system.
- List the information for the package.
- Delete the package.