2

Introduction to the UNIX and Linux Environments

Objectives

After completing this lesson, you should be able to:

- Describe the UNIX and Linux operating systems
- Execute commands from the command line

Lesson Agenda

- Describing the UNIX and Linux operating systems
- Executing commands from the command line

Multiuser, Multitasking, Time-Sharing Operating Systems

- In 1965, Multics was being developed by MIT, Bell Labs (AT&T) and GE.
- In 1969, Bell Labs withdrew from the Multics Project and began work on its own UNIX OS.
- In 1977, programmers at the University of California, Berkeley made significant enhancements, including networking capability resulting in Berkeley Software Distribution (BSD UNIX).
- In 1983, Richard Stallman, formally of MIT's AI Lab, began the GNU Project to develop a free UNIX-like OS called GNU (GNU's Not UNIX).

Multiuser, Multitasking, Time-Sharing Operating Systems

- In 1987, AT&T UNIX, BSD UNIX, and other UNIX OSes were folded into System V Release 4 (SVR4) UNIX, that became an industry standard for the UNIX OS.
- In 1991,while the GNU OS was still in need of a kernel, Linus Torvalds, a Finnish Computer Science student, offered to create a kernel for the GNU OS.
  - In September, the Linux Kernel and the GNU userspace programs were released as the Linux OS.

UNIX and Linux OS Structure

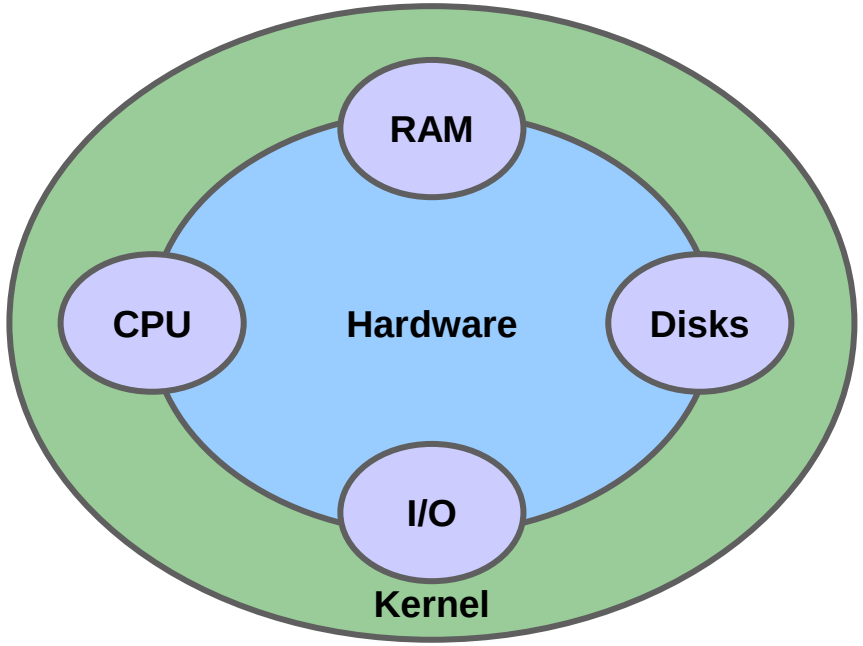
The UNIX and Linux OSes are structured around the following parts:

- Kernel
- File system
- Processes
- Shell

UNIX and Linux OS Structure: The Kernel

The kernel is the core of the OS and manages all the physical resources of the hardware, including:

- Memory management
- Device and storage management
- File system I/O
- Process management or Central Processing Unit (CPU) functions



UNIX and Linux OS Structure: The File System

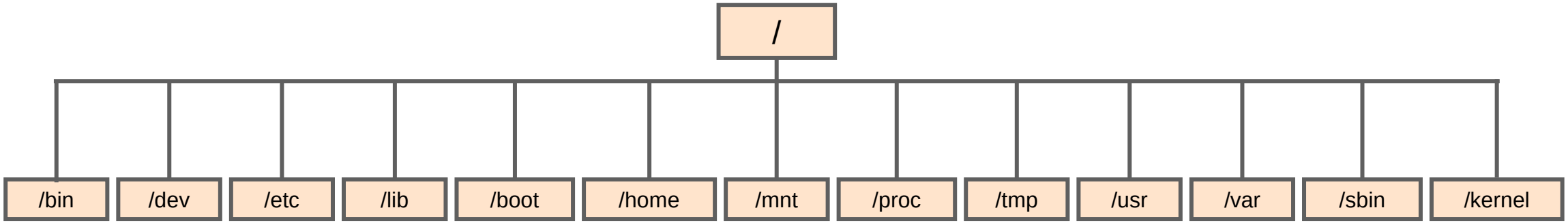
- All data in UNIX and Linux is organized into files.
- All files are organized into directories.
- These directories are organized into a file system.

The File System

- A file system is a logical collection of files on a partition, slice or disk.
- The UNIX and Linux file systems are a collection of files and directories that has the following properties:
  - It has a root directory (`/`) that contains other files and directories.
  - It has a boot directory (`/boot`) short for `bootstrap` which is used to boot up the operating system.
  - Each file or directory is identified by its name, the directory in that it resides, and a unique identifier, called an inode.
  - Each file system is self contained, in that there are no dependencies between one file system and another.

The `root` Directory "/"

- UNIX uses a hierarchical file system structure, much like an upside-down tree, with the root (/) at the base of the file system and all other directories spreading from there.
- The directories have specific purposes and generally hold the same types of information.
- The following directories exist on both UNIX and Linux versions that are POSIX compliant:



Note: On first login, you are taken to your home directory.

File Types

- Everything in UNIX is considered to be a file, including physical devices, such as DVD-ROMs, USB devices, and external disks.
- In UNIX, there are three basic types of files:
  - Ordinary files (-)
  - Directories (d)
  - Special files
    - Symbolic links (l)
    - Character device (c)
    - Block device (b)
    - Socket (s)
    - Pipe (p)
    - Door (D)

UNIX and Linux OS Structure: Processes

- Every program you run in the UNIX or Linux OSes creates a process.
  - When you log in and start the shell, you start a process.
  - When you run a command or when you open an application, you start a process.
- The system starts processes called daemons.
- Daemons are processes that run in the background and provide services.
  - For instance, the desktop login daemon provides a graphical prompt for logging in to the OS.

UNIX and Linux OS Structure: The Shell

- The shell is primarily a command interpreter and serves as an interface between the user and the kernel.
- The shell accepts the commands that a user enters, interprets these commands, and passes them to the kernel, which executes the commands.
- The GNU Bourne Again Shell (`bash`) is the default shell in both Oracle Solaris and Oracle Linux.

Shells, a User Command Interface to the Kernel

The default shell in Oracle Linux and Oracle Solaris is:

- Bourne Again Shell (`bash`): Brian Fox (GNU Project 1989).

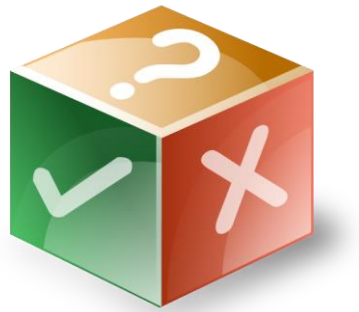
Other shells:

- Bourne (`sh`) - Stephen Bourne (Bell Labs 1977)
- C (`csh`) - Bill Joy (University of California, Berkeley 1978)
- TC (`tcsh`) - Ken Greer (Carnegie Mellon University 1981) an improved version of `csh`
- Korn (`ksh`) - David Korn (Bell Labs 1983)
- Z (`zsh`) - Paul Falstad (Princeton University 1990)
- Plus additional shells

Quiz

Select the default shells in Oracle Linux and Oracle Solaris OSes:

a. TC shell
b. Z shell
c. Korn shell
d. C shell
e. Bash shell
f. Bourne shell



User Accounts

- A `user` account is a login account.
  - Regular users can log in and use the system, but cannot administer the system.
- A `group` is a collection of users who can share files and other system resources.
  - Users working on the same project could be formed into a group.
  - Each group must have a name, a group identification (GID) number, and a list of usernames that belong to the group. The GID number identifies the group internally to the system.

Components of a User Account

The components of a user account are:

- Username
- Password
- User identification (UID) number
- Group identification (GID) number
  - Primary group
  - Secondary group
- Comment (GECOS)
- User's home directory
- User's login shell
  - Prompts for bash shell

Changing User Account Login Credentials

- As a matter of course a user's password expires on a regular basis.
- The `passwd` command allows you to update your password when it expires or on an as-needed basis.
- When changing a password, there can be new password requirements:
  - On the number of characters, numbers and symbols that can be used
  - On the minimum length of the password
  - And password reuse-ability

Changing User Account Login Credentials: Examples

- An Oracle Linux example:

```
$ passwd
Changing password for <username>
Changing password <username>
(current) UNIX password:
New Password:
Retype new Password:
passwd: all authentication tokens updated successfully
```

- An Oracle Solaris example:

```
$ passwd
passwd: Changing password for <username>
New Password:
Re-enter new Password:
passwd: password successfully changed for <username>
```

Roles and Rights Profile

- Unlike a user account, that is a *login* account, a `role` is not a login account.
- Role-based access control (RBAC) is a security feature for controlling user access to tasks that would normally be restricted to the `root` role.
- Each `role` is associated with a rights profile.
- Rights profiles are convenient collections of authorizations and other security attributes, commands with security attributes, and supplementary rights profiles.
- Once a user has signed on, if they have been granted the root rights profile, they can use the `su - root` to assume the `root` role.
- A user can assume only the roles that are assigned to the user's login account.

Note: Currently RBAC is only supported on **Oracle Solaris**. Although, Red Hat's Fedora Project has been testing RBAC as part of Security Enhanced Linux (SELinux) for several years.

Changing Ownership of a Login Session to Another User/Role

- You can use the `su` command to change the ownership of the current login session to another user of role.
- The `su` command is commonly used to switch to the `root` user (Linux) or assume the `root` role (Solaris).

```
$ su [-] [username]
```

- The – (dash) is used to the instantiate the home directory and environment variables of the switched-to user.
- If the `[username]` attribute is omitted then the `root` username is the default.

```
$ su -
```

User's Home Directory

- When you first log in, you are taken to your home directory.
- The home directory is where you create and organize all your files and subdirectories.
- To go to your home directory, use the following commands:

```
$ cd
or
$ cd ~
```

- To go to the home directory of another user, use the following command:

```
$ cd ~username
```

- The ~ (tilde) is a pathname expansion of a user's home directory.

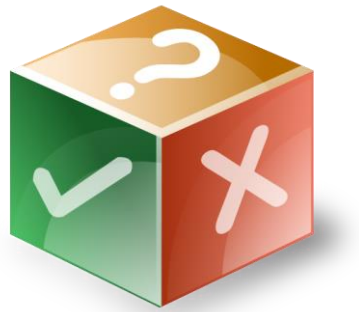
Note: Security settings within an organization might prevent users from accessing/viewing the home directory of another user.

Quiz

A role in Oracle Solaris is a login account just like a user account.

a. True

b. False



UNIX Variants

Several variants evolved out of UNIX. The following are some of the popular UNIX variants:

- BSD (Berkeley Software Distribution)
- SunOS (BSD - Predecessor of Solaris)
- Oracle Solaris (AT&T's Bell Labs - System V)
- AIX (IBM - System V)
- Mac OS X (Apple - BSD)
- HP-UX (Hewlett Packard - System V)
- UNIXWare (Novell - System V)

Oracle Solaris

- Sun’s original version of the UNIX Operating System was known as SunOS, based on BSD UNIX version 4.2.
- At that time, AT&T’s version of the UNIX environment was known as System V.
- AT&T, BSD, Sun and others created SVR4.
- SVR4 with multiprocessor capability became Oracle Solaris.

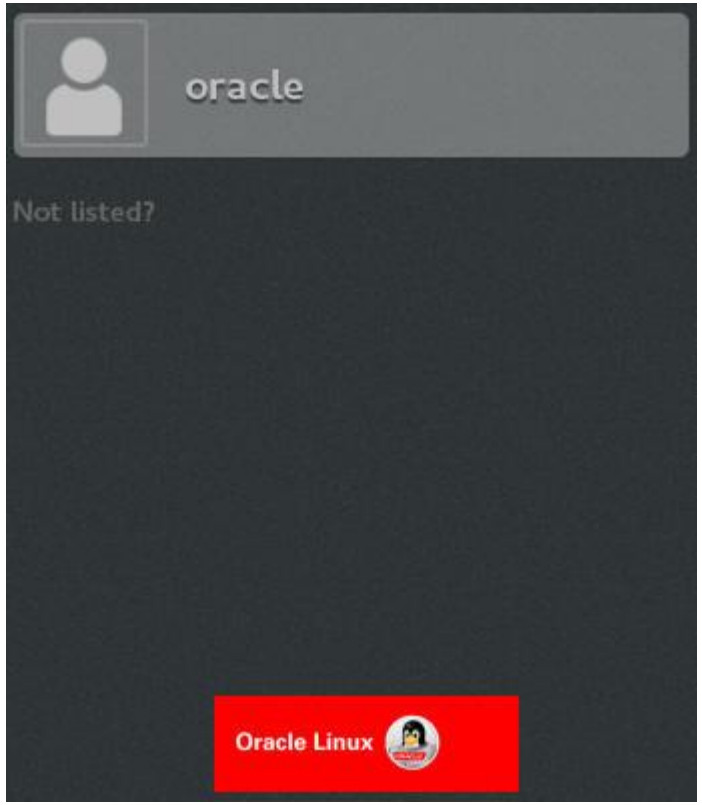
Linux Distributions

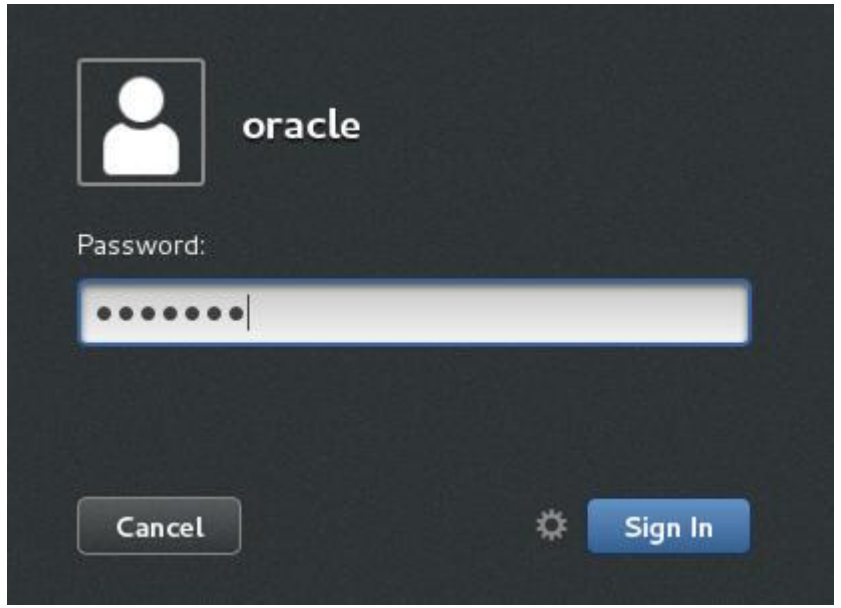
- Linux is packaged and distributed as a Linux distribution (distro) for desktop and server.
- There are over 600 Linux distros (see DistroWatch.com).
- Linux is a UNIX-like OS assembled and developed by Richard Stallman's GNU Project supplying the userspace programs and the Linux kernel from Linus Torvalds under the open source software development and distribution model.
- Some popular Linux distributions include:
  - Red Hat Enterprise Linux (RHEL) including Fedora, CentOS, and others
  - Oracle Linux (OL) based on RHEL with Oracle's Unbreakable Enterprise Kernel (UEK)
  - Debian GNU/Linux including Ubuntu, Linux Mint, and others
  - Slackware
  - openSUSE (SUSE)
  - Gentoo

The Desktop Environment

- The desktop environment is a graphical user interface (GUI) that allows you to perform a range of activities, such as:
  - Securing and selecting sessions
  - Adding and deleting workspaces
  - Changing backgrounds
  - Managing files
- The look and feel of Oracle Solaris and Oracle Linux desktop environments are displayed in the subsequent slides.

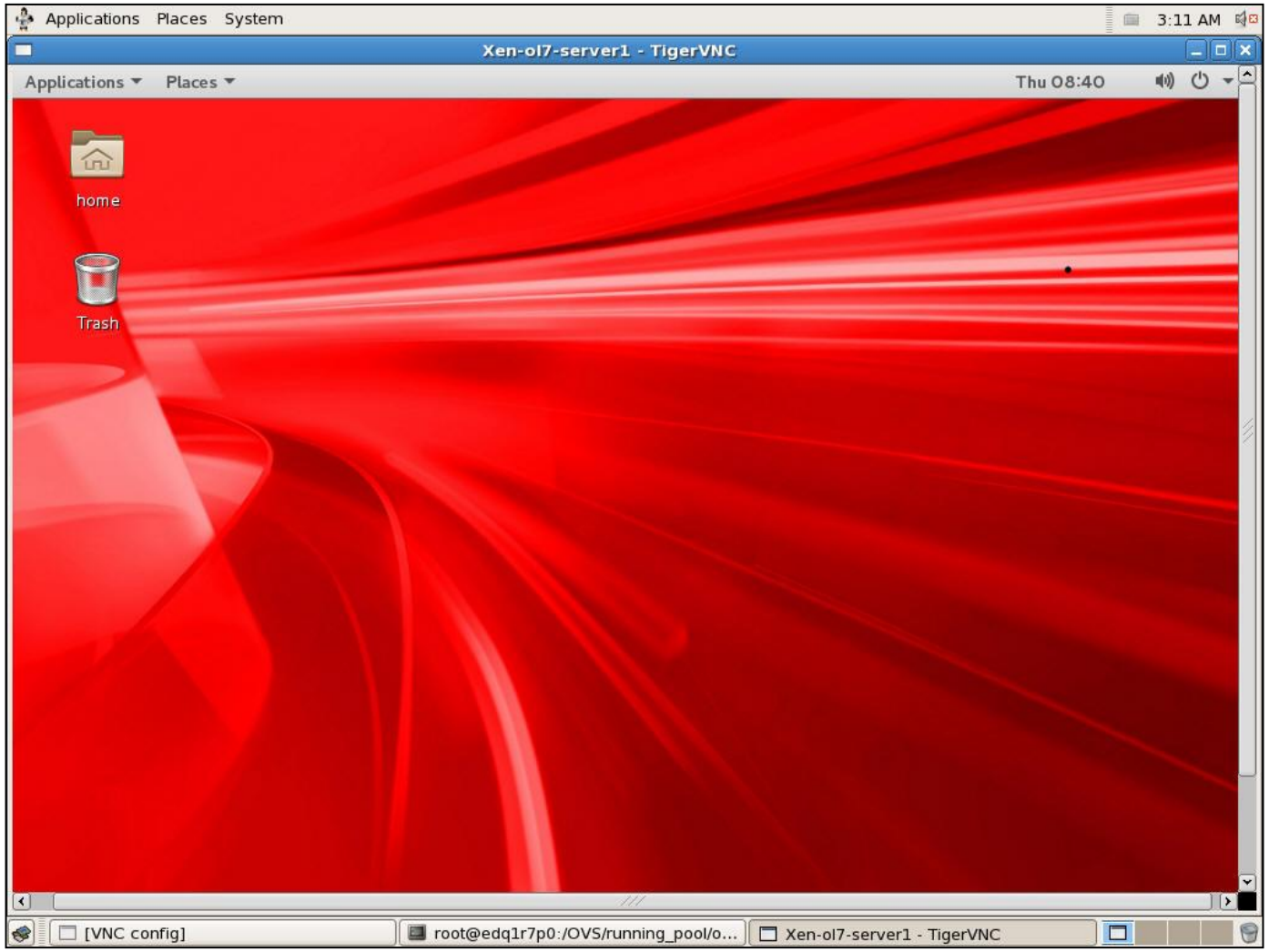
Logging In to Oracle Linux Using the Desktop Login Screen



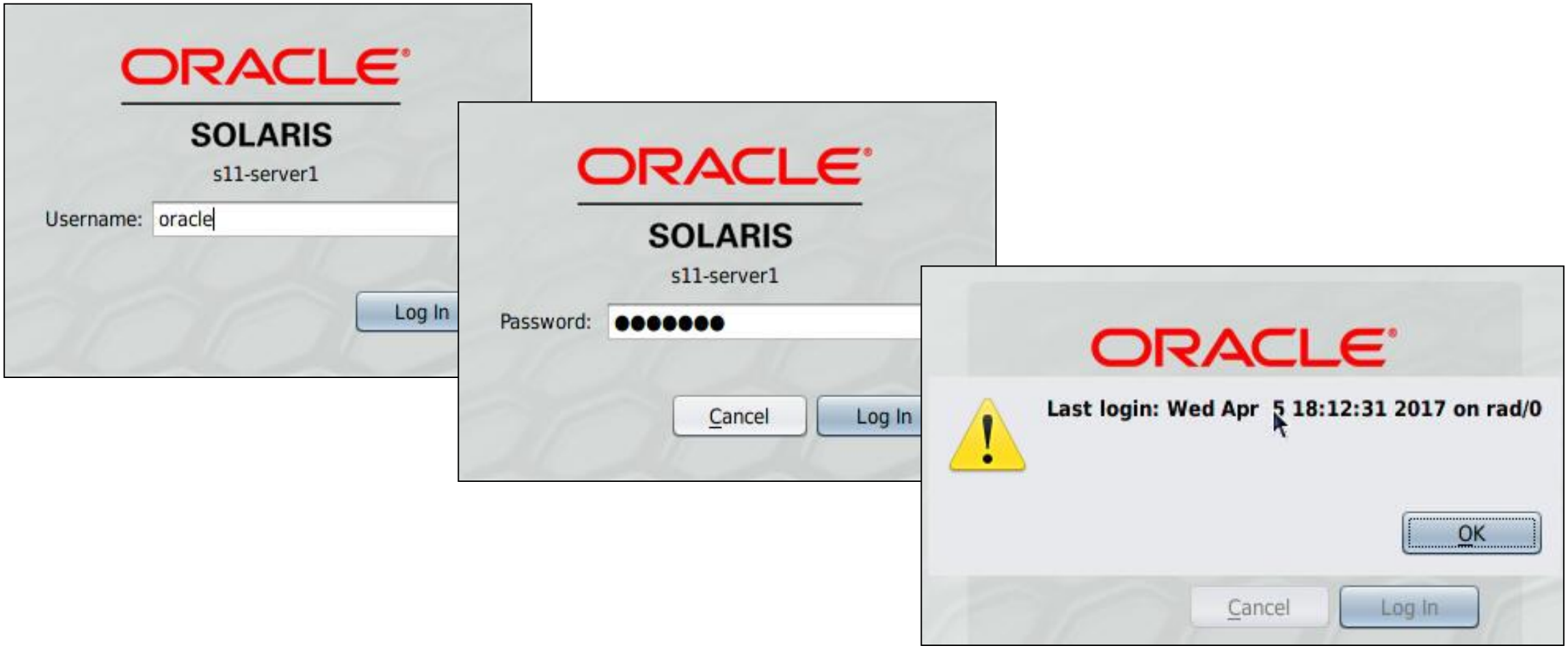


Oracle Linux Desktop

ORACLE
LINUX



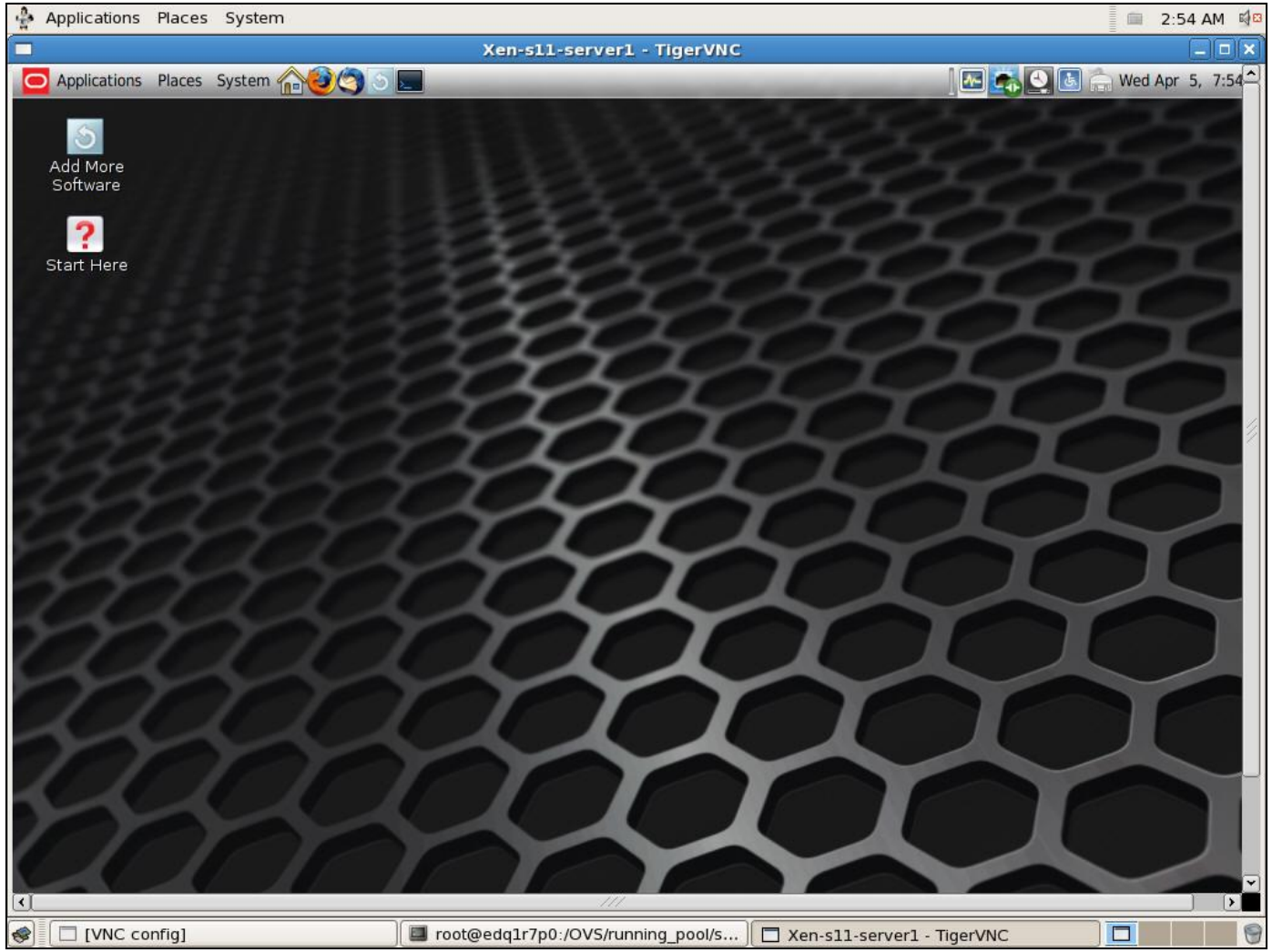
Logging In to Oracle Solaris Using the Desktop Login Screen



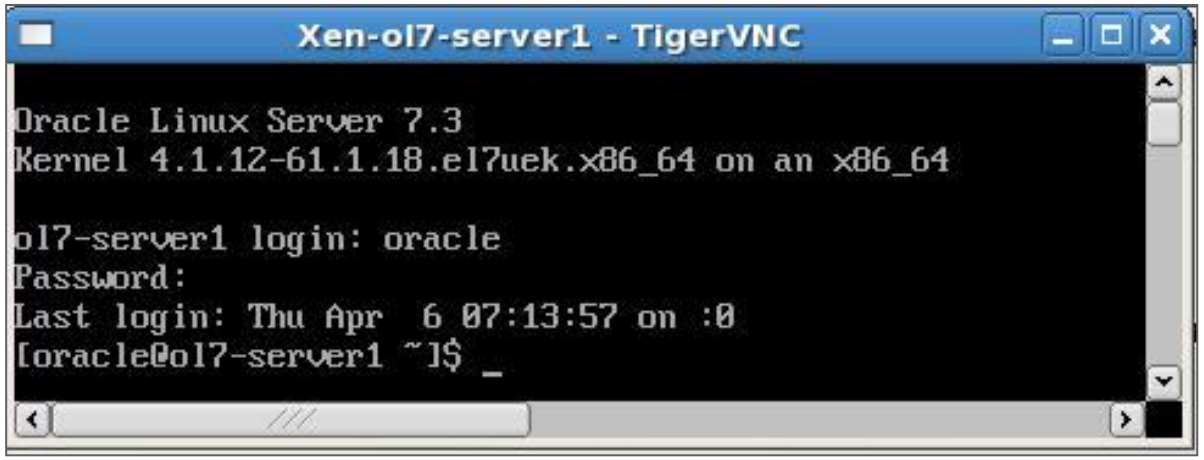
Oracle Solaris Desktop

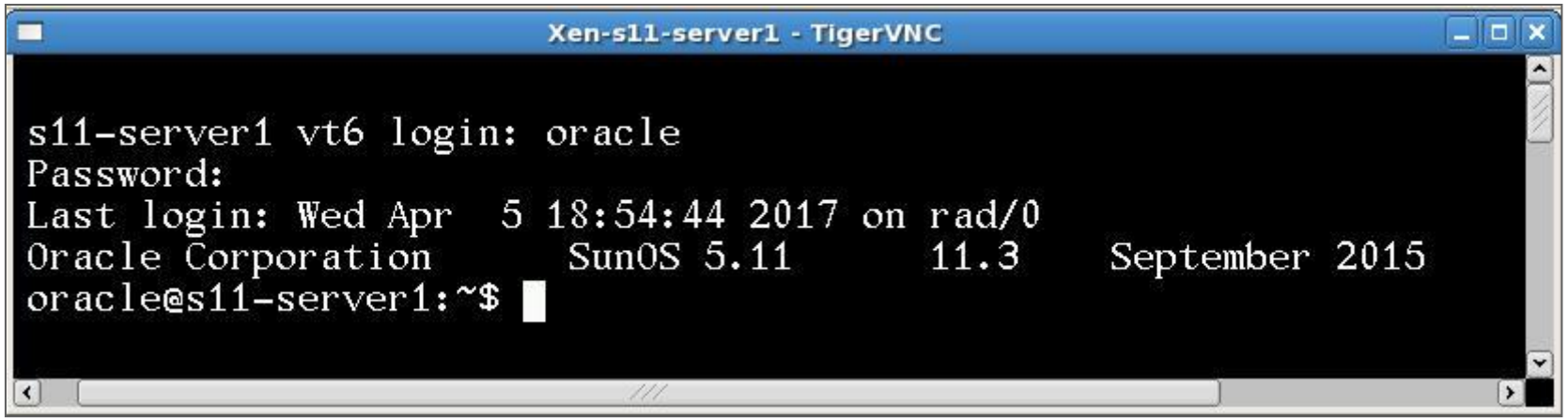
ORACLE

SOLARIS



Logging In Using the Command-line Option





Logging Out

- Depending on the interface, you use different commands or steps to log out.
- To log out of the graphical user interface:
  1. In Solaris on the desktop window, click **System**. In Linux, click the down arrow on the far right-hand side of the top task bar.
  2. Next, in Solaris click **Log Out <*username*>,** or in Linux just click the **<username>** and then click **Log Out**. A logout confirmation window appears.
  3. Click **OK** or **Log Out** or just press **Enter** to log out.
- To log out of the command-line interface:
  Type `exit`. This causes your shell to exit, or stop running.

Note: Some shells, depending on your configuration, also log you out if you type the EOF (End-Of-File) character, **Ctrl + D**.

Practice 2: Overview

This practice covers only the highlighted topics:

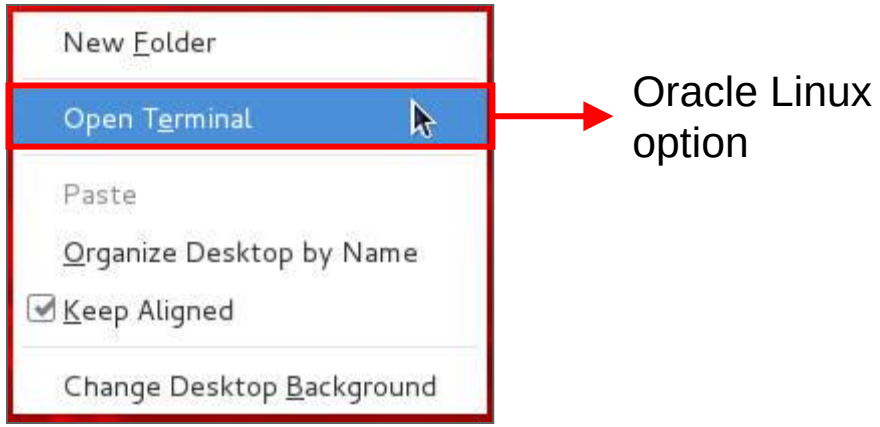
- 2-1: Logging in to the system and changing your user login password
- 2-3: Using the man pages
- 2-2: Displaying system information using the command line

Lesson Agenda

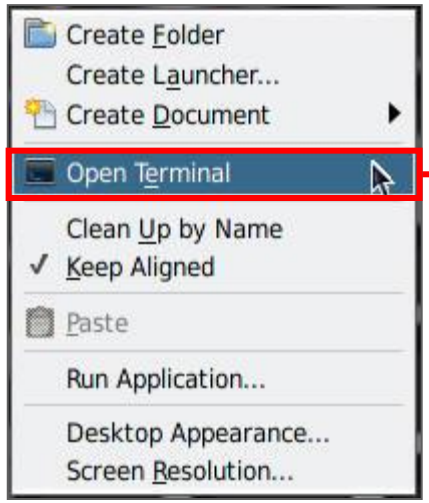
- Describing the UNIX and Linux operating system
- Executing commands from the command line

Executing Commands from the Command Line

- You can use system commands on the command line to instruct the system to perform specific tasks.
- The commands are entered into a terminal window.
- Right-clicking the Desktop causes a popup window to appear. Selecting *Open Terminal* will launch the terminal window.



Oracle Linux option



Oracle Solaris option

Command-Line Syntax

- The command syntax is the structure and order of the command components: command name, options, and arguments.
- Command-line commands can exist with or without options or arguments.
- You can change the behavior of commands by using a combination of options and arguments.

Item	Description
Command	Specifies what the system does (an executable)
Option	Specifies how the command runs (a modifier). Options start with a dash (-) symbol.
Argument	Specifies what is affected (a file, a directory, or text)

Using UNIX Commands

- The philosophy of UNIX commands is to create small, efficient programs that do a specific task and then group (pipe) them together to accomplish larger activities.
- There are over 4500 UNIX commands that can be run at the command line.
- To display the operating system information:

```
$ uname
SunOS
```

- To display the date and time:

```
$ date
Fri Mar 24 16:01:47 UTC 2017
```

- To clear the terminal window:

```
$ clear
```

- While there are many command-line commands, one regularly only uses just a couple of dozens of those commands. Most of which will be covered in this course.

Using Commands with Options

- Adding options to a command alters the information displayed.
- You can use more than one option with a command.
- You can also list multiple options separately or they can be combined after a dash (-).
- Use of a dash (-) preceding an option is command specific. Also, options are command specific.
- To get help with any specific command, one can enter `<command> --help` as an option.

Using Commands with Options on Oracle Linux

- The given example shows the `uname` command with two options:
  - The `-i` option displays the name of the hardware platform.
  - The `-n` option prints the host name of the local system.

```
$ uname -i
x86_64
$ uname -n
ol7-server1
```

- The following example shows the `uname` command with two combined options.

```
$ uname -rs
Linux 4.1.12-61.1.18.el7uek.x86_64
```

Using Commands with Options in Oracle Solaris

- The given example shows the `uname` command with two options:
  - The `-i` option displays the name of the hardware platform.
  - The `-n` option prints the host name of the local system.

```
$ uname -i
i86PC
$ uname -n
s11-server1
```

- The following example shows the `uname` command with two combined options:

```
$ uname -rs
SunOS 5.11
```

Using Commands with Arguments

- Arguments enable you to additionally define the output from a command.
- The following example shows the `cal` command with two arguments, `12` and `2015`.
  - The first argument, `12`, specifies the month.
  - The second argument, `2015`, specifies the year.

```
$ cal 12 2015
December 2015
S  M  Tu W  Th  F  S
      1  2  3   4  5
6  7  8  9  10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
```

Using Commands with Options and Arguments

Using the `ls` command without an option, with an option, with an argument, and with an option and argument together.

```
$ ls
dante     dir3          file.2          file3       greetings
dante_1   dir4          file.3          file4        myvars
$ ls -l
total 94
-rw-r--r-- 1 oracle      1319 Feb 6 09:25 dante
-rw-r--r-- 1 Oracle       368 Feb 6 09:25 dante_1
...(output truncated)
$ ls dante
dante
$ ls -l dante
-rw-r--r-- 1 oracle      1319 Feb 6 09:25 dante
```

Using Multiple Commands on the Command Line

- You can enter multiple commands on a single command line by using a semicolon (;) to separate each command.

```
$ command [options] [argument]; command [options] [argument]
```

- The shell recognizes the semicolon (;) as a command separator.
- The following example shows two commands separated by a semicolon.

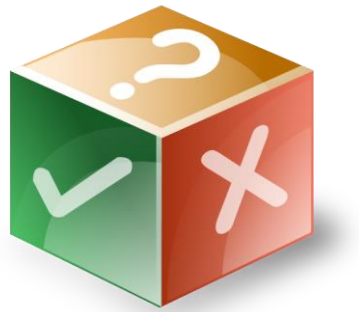
```
$ date; uname
Fri Mar 24 16:01:47 UTC 2017
SunOS
```

- The shell executes each command from left to right when you press Enter/Return.

Quiz

Given the `uname -a` command, which type of UNIX command component does `-a` represent?

a. Option
b. Parameter
c. Argument



Using the Man Pages

- The online technical reference manual (`man`) pages provide detailed syntax, descriptions, and usage of the commands.
- You can use the `man` *`command`* to display the man page entry that explains a given command.

```
$ man command
$ man [options] command
$ man [options] filename
```

- For more information about the `man` command options, see the `man` page.

Displaying the Man Pages

To display the man pages for the `uname` command using the `man` command.

```
$ man uname
Reformatting page.  Please Wait... done
User Commands                                     uname(1)
NAME
       uname - print name of current system
SYNOPSIS
       uname [ -aimnprsvX ]
       uname [ -S system_name ]
DESCRIPTION
The uname utility prints information about the current system
on the standard output. When options are specified, symbols
representing one or more system characteristics will be written
to the standard output. If no options are specified, uname
prints the current operating system's name. The options print
selected information returned by uname(2), sysinfo(2), or both.
...(output truncated)
```

Scrolling Through the Man Pages

The following table lists the keyboard commands for scrolling through the man pages.

Keyboard Command	Action
h	Provides a description (help) of all scrolling capabilities
Space bar	Displays the next screen of a man page
Return / Enter	Displays the next line of a man page
b	Moves back one full screen
/pattern	Searches forward for the *pattern* (regular expression)
n	Finds the next occurrence of the *pattern*
N	Changes the direction of the search
q	Quits the man command and returns to the shell prompt

Note: The `man` command uses the `less` command to scroll through the man pages. More about the `less` command in the lesson titled “Working with Files and Directories.”

Searching the Man Pages

There are two ways to search for information in the man pages:

- Searching by section number
- Searching by keyword

Section	Solaris Description	Section	Linux Description
1	User Commands	**1**	User Commands
2	System Calls	**2**	System Calls
3	Library Functions	**3**	Library Functions
7	Devices and Special files	**4**	Devices and Special files
4	File Formats	**5**	File Formats
5	Miscellaneous	**7**	Miscellaneous
1M	System Administration	**8**	System Administration

Searching the Man Pages: By Section

- The online man page entries are organized into sections based on the type or usage of the command or file.
  - Section 1 contains user commands.
  - Section 4 in Oracle Solaris and Section 5 in Oracle Linux contains information about various file formats.
- In Oracle Linux to look up a specific section of the man page, use the `man` command with the section number, and the command or file name.

```
$ man sectionnumber command
or
$ man sectionnumber filename
```

- In Oracle Solaris to look up a specific section of the man page, use the `man` command with the `-s` option, followed by the section number, and the command or file name.

```
$ man -s sectionnumber command
or
$ man -s sectionnumber filename
```

Searching the Man Pages: By Keyword

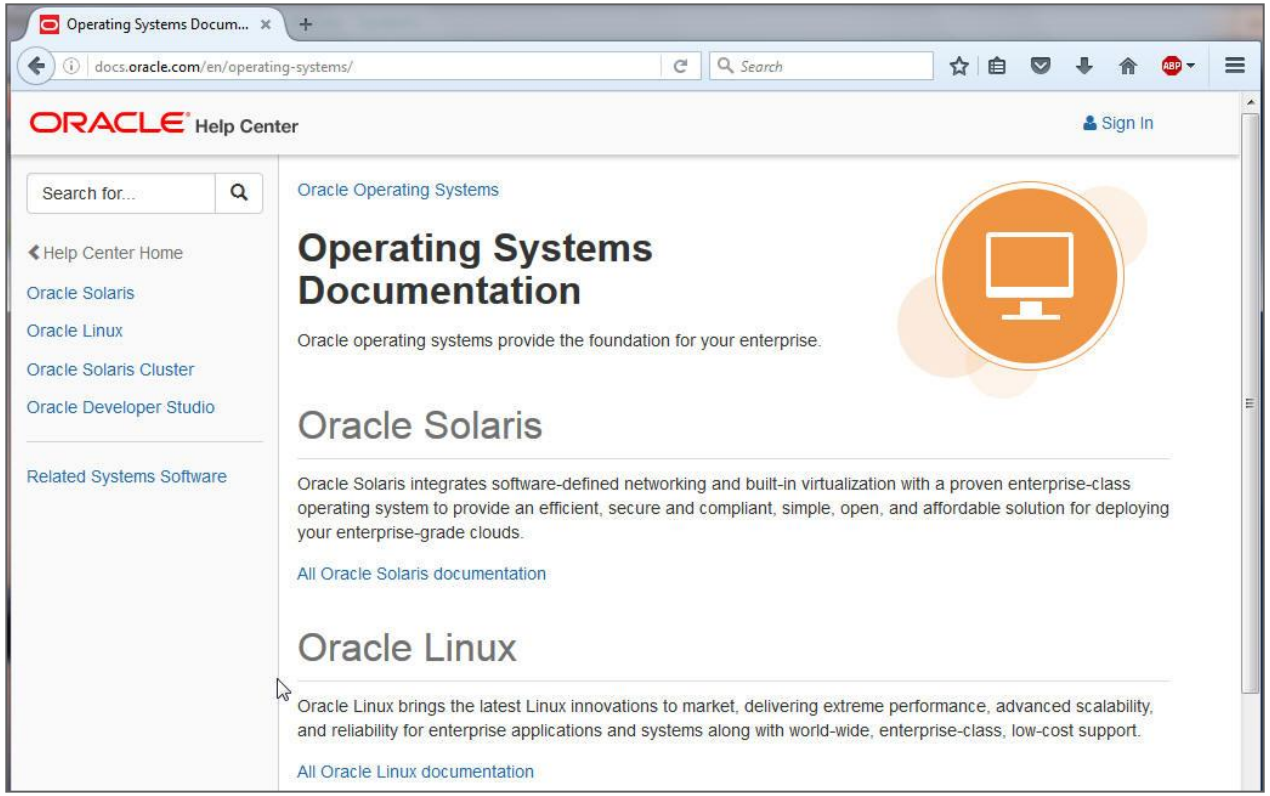
- When you are unsure of the name of a command, you can use the `man` command with the `-k` option and a keyword to search for matching man page entries.

```
$ man -k keyword
```

- The `man` command output provides a list of commands and descriptions that contain the specified keyword.

Accessing Online Product Documentation

For additional information about Oracle products, you can access the Oracle Technical Network (OTN) Documentation website.



Quiz

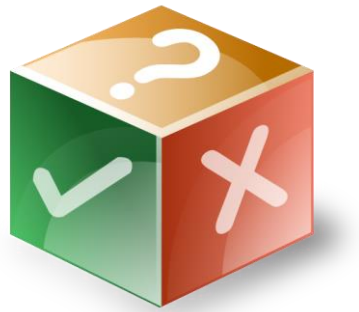
Which of the following `man` command options displays a specific section of the man page?

a. `-h`

b. `-q`

c. `-s`

d. `-n`



Summary

In this lesson, you should have learned how to:

- Describe the UNIX and Linux operating systems
- Execute commands from the command line

Practice 2: Overview

This practice covers only the highlighted topics:

- 2-1: Logging in to the system and changing your user login password
- 2-2: Displaying system information using the command line
- 2-3: Using the man pages