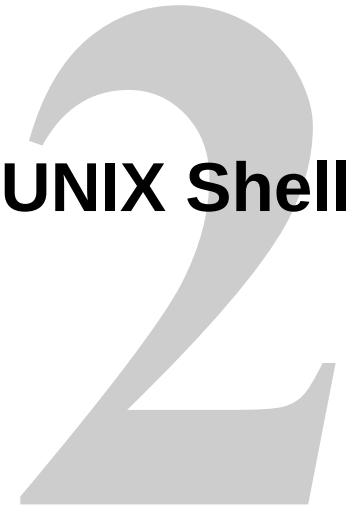
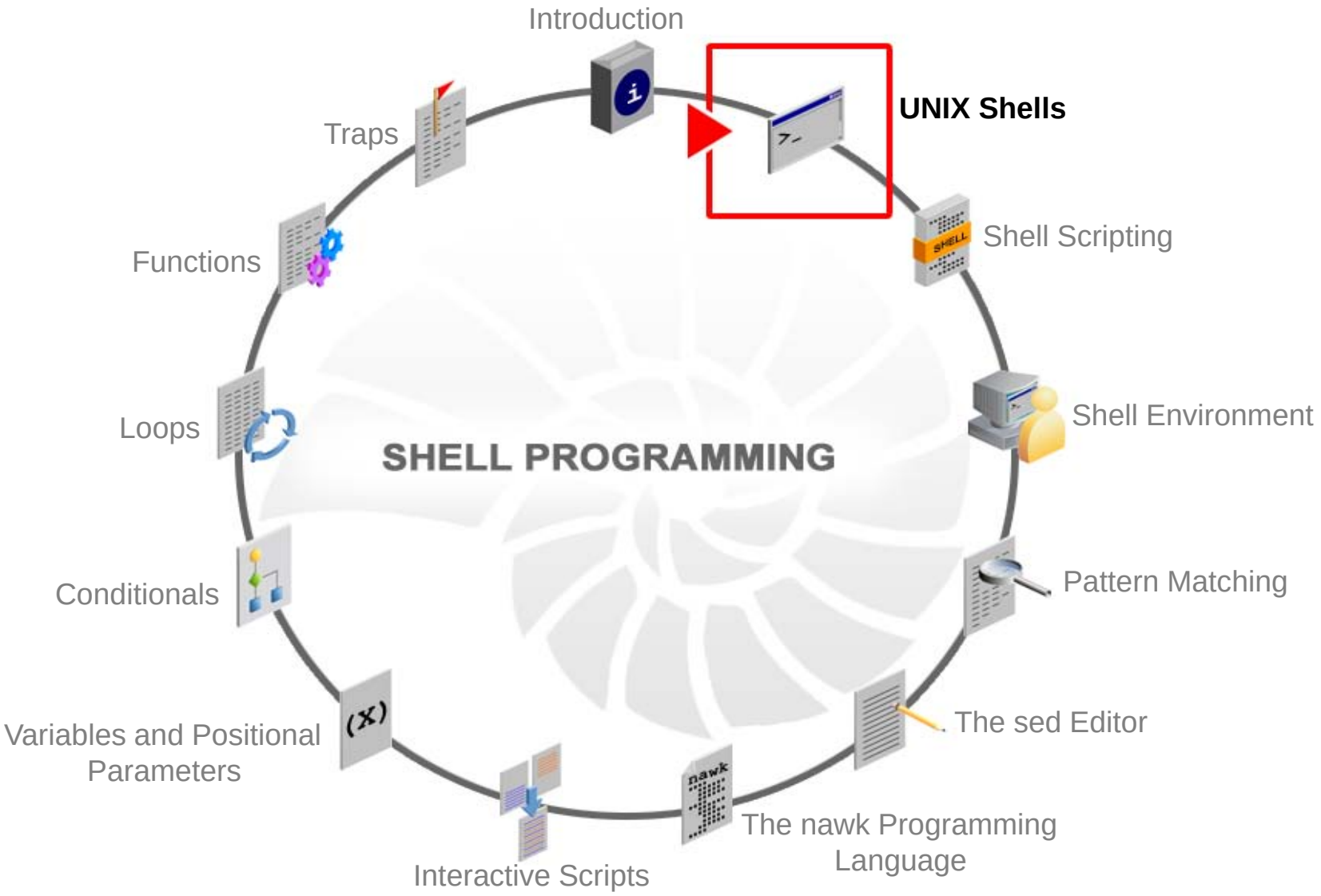


UNIX Shells



Objectives

After completing this lesson, you should be able to:

- Describe the role of a shell in the UNIX environment
- Describe the various UNIX shells

Agenda

- Describing the role of a shell in the UNIX environment
- Describing various UNIX shells

What Is a Shell?

- A shell is a program that provides an interface between a user and an operating system (OS) kernel.
- An OS starts a shell for each user when the user logs in or opens a terminal or console window.
- The shell's primary function is to read commands that are typed into a console or terminal window and then execute them.
- All OSs have shells:

OS	Shell
DOS	command.com
Oracle Solaris	Bash
Oracle Linux	Bash
MAC OS X	Bash
MS Windows	cmd.exe

Functions of a Shell

The main functions of a shell include the following:

- Command-line interpreter
- Programming language
- User environment

Command-Line Interpreter

When you type a command on the command line, the shell:

1. Interprets the command by parsing the command line and addressing metacharacters, redirection, and control.
2. Searches for and executes the command.
3. Analyzes each command and initiates execution of the requested program.

Programming Language

- You can type the commands directly into the shell at the prompt, or the shell can read commands from a file.
- A shell can contain two types of commands:
  - Built-in: `cd`, `exit`, `break`
  - Binary: `cat`, `cp`, `ls`
- A file containing shell commands is called a shell program or a shell script.
- Apart from commands, a shell script can contain programming constructs.
- A shell script is a text file that is interpreted, not compiled.
- The shell reads a line in the shell script and processes all statements found on that line before reading the next line.

User Environment

- The shell also provides a user environment that you can customize by using initialization files.
- These files contain settings for user environment characteristics, such as:
  - Search paths for finding commands
  - Default permissions on new files
  - Values for variables that other programs use
  - Values that you can customize

Agenda

- Describing the role of a shell in a UNIX environment
- Describing various UNIX shells

Types of Shells

UNIX-like OSs support a variety of shells:

Shell	Program File Name	File Path	Non-root default prompt	Root default prompt
Bash	`bash`	`/bin/bash`	`$`	`#`
Bourne	`sh`	`/bin/sh`	`$`	`#`
Korn	`ksh`	`/bin/ksh`	`$`	`#`
Z	`zsh`	`/bin/zsh`	`$`	`#`
Note: Above shells are Bourne-compatible				
C	`csh`	`/bin/csh`	`%`	`#`
TC	`tcsh`	`/bin/tcsh`	`%`	`#`

The Bourne Shell

- The Bourne shell (`sh`):
  - Is the original UNIX shell written by Steve Bourne at AT&T Bell Labs
  - Is the preferred shell for shell programming because of its compactness and speed
- The command full-path name is `/bin/sh` and `/sbin/sh`.
- The default prompt:
  - For a non-root user is $
  - For a root user is #

The C Shell

- The C shell (`csh`):
  - Is a UNIX enhancement written by Bill Joy at the University of California at Berkeley
  - Has features for interactive use such as aliases and command history
  - Includes convenient programming features such as built-in arithmetic and a C-like expression syntax
- The command full-path name is `/bin/csh`.
- The default prompt:
  - For a non-root user is %
  - For a root user is #

The Korn Shell

- The Korn shell (`ksh`):
  - Is a superset of the Bourne shell written by David Korn at AT&T Bell Labs
  - Supports everything in the Bourne shell
  - Has interactive features comparable to those in the C shell
  - Includes convenient programming features such as built-in arithmetic and C-like arrays and functions
  - Is faster than the C shell
  - Runs scripts written for the Bourne shell
- The command full-path name is `/bin/ksh`.
- The default prompt:
  - For a non-root user is $
  - For a root user is #

The Bash Shell

- The Bash shell:
  - Is compatible with the Bourne shell
  - Includes useful features from the Korn and C shells
  - Provides arrow keys that are automatically mapped for command recall and editing
- The command full-path name is `/bin/bash`.
- The default prompt:
  - For a non-root user is `bash-x.xx$`
  - For a root user is `bash-x.xx#`

The Z Shell

- The Z shell (`zsh`):
  - Closely resembles the Korn shell
  - Includes built-in spelling correction and programmable command completion
- The command full-path name is `/bin/zsh`.
- The default prompt:
  - For a non-root user is %
  - For a root user is #

The Enhanced C Shell

- The Enhanced C shell (`tcsh`):
  - Is compatible with the C shell
  - Has command-line editor, programmable word completion, and spelling correction features
  - Has arrow keys that are automatically mapped for command recall and editing
- The command full-path name is `/bin/tcsh`.
- The default prompt:
  - For a non-root user is >
  - For a root user is #

Shell Features Comparison

Feature	Bash	Bourne	Korn	C	TC
Programming language	Yes	Yes	Yes	Yes	Yes
Shell variables	Yes	Yes	Yes	Yes	Yes
Command alias	Yes	No	Yes	Yes	Yes
Command history	Yes	No	Yes	Yes	Yes
File name completion	Yes	No	Yes*	Yes*	Yes
Command line editing	Yes	No	Yes*	No	Yes
Job control	Yes	No	Yes	Yes	Yes

Note: * means not a default setting for this shell.

Using a Different Shell

- To change your current shell, type in an alternative shell (command) at the prompt.
- For example, if you are in the bash shell, type `csh` at the prompt to change your shell to the C shell.
- To determine your current shell, execute the `ps` command.
- If you have opened a series of shells, use the `ps|sort` command to sort the shells in the order in which you invoked them.

Summary

In this lesson, you should have learned how to:

- Describe the role of shells in the UNIX environment
- Describe the standard shells

Practice 2 Overview: UNIX Shells

This practice covers the following topic:

- Reviewing Unix Shells