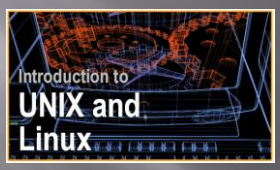


CST 126 – LESSON 7

Using Utilities to Accomplish Complex Tasks

Chapter 6

Overview

Shell scripts can be used to accomplish a variety of tasks on Unix systems. Some of these include:

- Creating and executing a script to list user information.
- Listing directories and files separately.
- Identifying changes made to files in a directory.
- Changing environment variables.
- Complex scripting.

Creating and Executing a Script

- Creating a script.
  - Why? A series of commands can be executed repeatedly/automatically by placing them in a script file.
  - Why? Placing commands in scripts helps avoid errors and save time.
  - How? Use an Editor.
- Executing a script.
  - Instructions can be given to the current shell to read the file (source) and execute all the commands.
  - The script file can be made executable, and a child shell can be started to read the script file and execute the commands.

Executing a Script

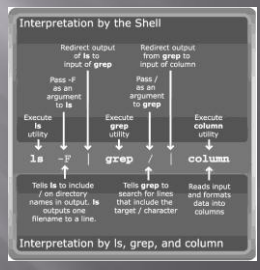
- The command used to make the shell executable is "chmod +x filename" or "chmod 755 filename"
- By default, the output of the script is redirected to the screen.
- The output of the script can be redirected to a file.
- In ksh enter the name of the script to execute at the prompt. (ex. $ workday)

Scripting Example: Listing Directories and Files Separately

Assume we want to select and display only directories / files. We might use the following commands:

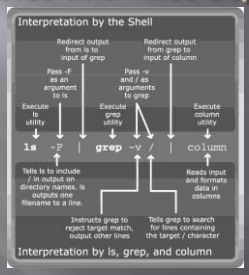
- The "ls –F | grep /" command is used to select only lines that contain directory names.
- The "ls –F | grep / | column" command can be used to put the output into columns.
- The "ls –F | grep –v / | column" command is used to select only files.

Scripting Example: Listing Directories and Files Separately



Selecting Directories Using Multiple Utilities

Scripting Example: Listing Directories and Files Separately

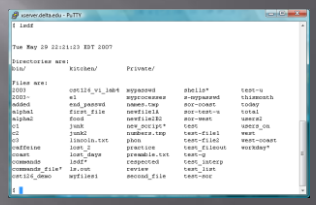


Interpretation by grep Using –v option

Scripting Example: Listing Directories and Files Separately

Could look something like this:

```
echo
echo
date
echo
echo 'Directories are:'
ls –F | grep / | column
echo
echo 'Files are:'
ls –F | grep –v / | column
echo
```



Adding Comments to Scripts

Adding comments to scripts:

- The # (pound) sign is used to add comments to scripts.
- Ensure that the # sign is placed at the beginning of each line.
- The pound (#) sign cannot be used inside a long command line.

Creating Complex Scripts

- By employing several utilities connected together using pipes in a script, we can accomplish complex tasks.
- Process involves:
  - Define the problem to be solved.
  - Break the problem into meaningful pieces that can be accomplished separately.
  - Write a basic script that accomplishes some task.
  - Make sure it works.
  - Include additional code to increase the scripts functionality.
  - Keep debugging as you develop the script.

Summary

- A series of commands placed in a script file can be repeatedly executed, avoiding errors and saving time.
- The # sign is used to add comments to script files.
- The pipeline feature of UNIX is very useful and is central to manipulating data effectively with UNIX utilities.
- Complex scripts need to be developed incrementally and tested as they are written.