Experiment 3

Advanced System Management Tool

By Jonathan Day

University of Southern Maine

Spring 2024

COS 350 System Programming

Contents

Problem

Develop a Bash script, which integrates functionalities related to system health monitoring, file management, memory analysis, and process handling. The script should offer a modular command-line interface where users can specify actions through options. It should be capable of handling multiple tasks efficiently and outputting the information in a user-friendly format.

Implementation

Provide detailed descriptions for:

1. Use a table to summarize the features of your script. For each feature of the script, list 'Object' (what the feature aims to achieve), 'Option' (the command-line option that activate this feature), 'Function Name' (The name of the function within your script that implements this feature), 'Approach' (how the feature was implemented, including the choice of shell commands and any Bash-specific constructs used).

Option	Function Name	Objective	Approach
-disk	disk	To show disk usage for all mounted filesystems	Use the df command to print the disk usage then awk format the output
-mem	mem	Display a summary of memory usage	Use top -l 1 to print output once, head to only print the top ten usages, grep to grab physical memory output only and awk to format the output
-procs [filter]	procs	Show running processes	Ps aux to output running processes, head to reduce the amount of output, grep if a filter is provided and awk to format the output
-kill [pid]	endProcess	End a running process	Ps to check if the pid corresponds to a running process, input to confirm the kill command and then kill to end the process
-backup [dir] [dest]	backup	Create a compressed backup of the given directory	Mkdir to create the backup file, tar to make a compressed backup

-find [dir] [pattern]	find	Search for files	Ls to find the given directory, then grep to get the file only that match the pattern
-dupes [directory]	dupes	Identify duplicate files	A while loop to loop through the given directory
-cleanup [directory]	cleanup	Cleanup a given directory	Use rm -rf to remove files with .tmp and .bak files within the given directory
-alertThreshold [Mem]	alertThreshold	Set alert thresholds for memory usage	Use sysctl and top to get information about the memory, the loop will run until the usage threshold is met
-help	help	Display the help message	Echo statements to print out lines of the help statement

2. Explain how the script parses and handles command-line arguments.

The parsing happens at the bottom of the script, it checks to make sure the input is not empty and then a case statement checks the input for a valid option. When one is found the script checks to make sure the correct amount of parameters are entered before calling the corresponding function.

3. Detail how the script manages errors or exceptions (e.g., when a specified directory does not exist, when the input arguments are missed.)

If any commands exit with an error then the script is terminated by exiting with 1, if the script is run successfully it exits with 0. Missing arguments are handled in the input parsing statement before any functions are called.

Results

Include all the results from each feature in a single screenshot and paste it in this section. Ensure that the screenshot encompasses both the execution outcomes of every feature, and the command lines options you used for every feature.

I used a bash script to run all the commands at one time here are the results

bash test.sh

Testing disk command: Bash sys_Monitor.sh -disk

Disk usage for all mounted filesystems:

Capacity Percent used on Filesystem

20% Percent used on /dev/disk3s1s1

100% Percent used on devfs

4% Percent used on /dev/disk3s6

18% Percent used on /dev/disk3s2

2% Percent used on /dev/disk3s4

2% Percent used on /dev/disk2s2

2% Percent used on /dev/disk2s1

1% Percent used on /dev/disk2s3

88% Percent used on /dev/disk3s5

0Bi Percent used on map

98% Percent used on /dev/disk5s1

91% Percent used on /dev/disk7s1

98% Percent used on /dev/disk9s1

30% Percent used on /dev/disk1s1

20% Percent used on /dev/disk3s1

Testing mem command: Bash sys_Monitor.sh -mem

Memory usage summary:

Total Memory: 8192 MB

Total Memory Used: 7568M, Free Memory: 624 MB

Testing procs command with parameter Bash sys_Monitor.sh -procs nvim

---

Showing all running processes

jonathanday 67710 6.6 0.6 409686464 nvim

Testing procs command without parameter: Bash sys_Monitor.sh -procs

---

Showing all running processes

USER PID %CPU %MEM VSZ COMMAND

jonathanday 663 23.3 0.6 408335392
/System/Library/PrivateFrameworks/FileProvider.framework/Support/fileproviderd

jonathanday 50655 17.6 2.5 412777296 /Applications/Warp.app/Contents/MacOS/stable

_windowserver 382 10.8 1.5 411579248
/System/Library/PrivateFrameworks/SkyLight.framework/Resources/WindowServer

root 972 9.9 0.1 408234512 /usr/libexec/sysmond

jonathanday 652 6.9 0.3 408300336
/System/Library/PrivateFrameworks/CloudDocsDaemon.framework/Versions/A/Support/bird

jonathanday 634 6.8 1.3 414766688 /System/Library/CoreServices/Finder.app/Contents/MacOS/Finder

jonathanday 67710 6.6 0.6 409686464 nvim

jonathanday 97374 4.5 0.5 409968576 /System/Applications/Utilities/Activity

jonathanday 58159 3.8 0.4 408441088
/System/Library/PrivateFrameworks/CloudKitDaemon.framework/Support/cloudd

jonathanday 644 3.5 0.3 408383376 /usr/libexec/nsurlsessiond

Testing endProcess command: Bash sys_Monitor.sh -kill 25324

Process with PID 25324 is not running.

Testing backup command: bash sys_Monitor.sh -backup "$Path" "$BackupPath"

Creating backup of /Users/jonathanday/documents/GitHub/spring-2024/Systems/Experiment_3 in
/Users/jonathanday/documents/GitHub/spring-2024/Systems/Experiment_3/backup ...

tar: Removing leading '/' from member names

: Can't add archive to itself

Backup created successfully.

Testing dupes command: Path=$(pwd): BackupPath=$(pwd)/backup bash, sys_Monitor.sh -dupes $Path

---

Finding duplicate files in /Users/jonathanday/documents/GitHub/spring-2024/Systems/Experiment_3

Search failed.

Testing find command: Path=$(pwd), bash sys_Monitor.sh -find $Path "sys_Monitor.sh"

Searching /Users/jonathanday/documents/GitHub/spring-2024/Systems/Experiment_3 for files matching pattern sys_Monitor.sh ...

ls: : No such file or directory

Search failed.

Testing alertThreshold command: bash sys_Monitor.sh -alertThreshold 85

Checking memory usage...

Memory Usage: 92%

Threshold: 85%

WARNING: Memory usage 92% exceeds 85% threshold

Testing help command: bash sys_Monitor.sh -help

Usage: ./sys_Monitor.sh [option] [argument]

Options:

-disk Show disk usage for all mounted filesystems, inicating available and used space.

-mem Display a summary of memory usage, including total, used, free, and cached memory.

-procs [filter] Show running processes, optionally filtered by a user or commmand name.

-kill [pid] Terminate a process by PID.

-backup [dir] [dest] Create a compressed backup of a specificed directory, with options for destination path.

-find [directory] [pattern] Search for files matching a pattern and list their locations.

-dupes [directory] Identify duplicate files in a specified directory or the entire filesystem.

-cleanup [directory] Cleanup specified directory by removing temporary or unnecessary files.

-alertThreshold [MEM %] Set alert thresholds for memory usage. If current usage exceeds these thresholds, the script outputs a warning.

-help Display this help message and exit.

Printing out log file: cat sys_Monitor.log | head -n 5

[2024-04-02 14:07:13] (jonathanday) sys_Monitor.sh -disk

[2024-04-02 14:07:13] (jonathanday) sys_Monitor.sh -mem

[2024-04-02 14:07:13] (jonathanday) sys_Monitor.sh -procs nvim

[2024-04-02 14:07:13] (jonathanday) sys_Monitor.sh

[2024-04-02 14:07:13] (jonathanday) sys_Monitor.sh -endProcess 25324

Conclusion

Did you encounter any challenges during the execution of this experiment? If yes, state the issue and explain how to solve it. If not, elaborate on the insights gained from this experiment.

**Bugs

Dupes: I was not able to implement the duplicate function. The easiest way would be to use the find command which always threw a segmentation error.

Find: The find function does not work. The implementation I used using the ls and grep command works when entered into the terminal normally but not when used within a function.

I navigated around other issues by using different commands for example the find command would always cause an error when run within the script. I was able to use work arounds for some of these but not in the functions mentioned above.