IMAT5122 – Computer Systems and Networks

COURSEWORK 2021/2022

FIRSTNAME LASTNAME

AFFILIATION

COURSE NAME

INSTRUCTOR NAME

DUE DATE

**SysMonitor.sh**

Bash script that automatically and regularly writes to a log file SysMonitor.log with timestamped detailed information about significant changes to

1) The Current users logged IN

2) Current Processes

3) Top 5 CPU Utilizing processes

4) Devices Plugged in

5) Disk Usage

6) Network Interfaces and their state

This script should be capable of running in any OS having a Bash environment.

**Installation**

SysMonitor is a self-contained script. Simply copy SysMonitor.sh and SysMonitor.conf to a directory of your choosing. Don’t forget to make it executable:

|  |
| --- |
| chmod +x SysMonitor.sh |

**Usage**

|  |
| --- |
| Bash script that automatically and regularly writes to a log file SysMonitor.log with timestamped detailed information about significant changes to  1) The Current users logged IN  2) Current Processes  3) Top 5 CPU Utilizing processes  4) Devices Plugged in  5) Disk Usage  6) Network Interfaces and their state  Usage: ./SysMonitor.sh [Option 1[2[3...]]]"  -h Shows this help page  -d Runs the script in DEBUG mode  Example: ./SysMonitor.sh |

**Design**

The script uses pre-installed command line tools and doesn’t rely on additional programs / tools / libraries / etc. Some of the tools used are

* who – it displays information about all users currently logged on the local system.
* ps command – short for Process Status, used to display or view information related to the processes running in a Linux system.
* top – it provides a dynamic real-time view of the running processes.
* lsusb – used to display the information about USB busses and the devices connected to them.
* nmcli – used for controlling NetworkManager. It also displays network device status, create, edit, activate/deactivate, and delete network connections.

At its core the script works by comparing the outputs of the aforementioned commands every 5 seconds by temporarily created output. The output of commands are stripped and only key-details is extracted for comparison. sed [1] and grep [2] command line tools were used for stripping and managing relevant information. cmp [3] and diff [4] command line tools were used for comparison.

By editing the SysMonitor.conf file variable the script can be made to run at regular interval, by default the script runs every 5 seconds, giving the value of 0 will run the script without an interval.

The logging format is:

TIMESTAMP LOGLEVEL MESSAGE

To get rid of the temporarily files created by the script, the script has a function that is triggered when a keyboard interrupt [5] occurs (CTRL + C) which sends a termination signal to the script, effectively deleting temporary files. The script also have a DEBUG mode which can be activated by the –d option.

**Output**

|  |
| --- |
| [Thu 25 Nov 2021 02:08:11 PM EST] [DEBUG] > SysMonitor SCRIPT\_STARTS\_HERE  [Thu 25 Nov 2021 02:08:11 PM EST] [DEBUG] Scanning for Users Currently Logged In...  < frosty  [Thu 25 Nov 2021 02:08:11 PM EST] [DEBUG] Showing current processes...  < frosty 1000 1306 ?  < frosty 1000 1380 ?  < frosty 1000 2706 pts/1  < frosty 1000 2879 ?  < frosty 1000 3445 ?  < root 0 869 tty7  < USER UID PID TT  [Thu 25 Nov 2021 02:08:11 PM EST] [DEBUG] Scanning Top 5 CPU Utilizing processes...  %CPU %MEM COMMAND  40.0 3.2 kazam  6.7 1.6 Xorg  6.7 0.0 top  0.0 0.1 systemd  0.0 0.0 kthreadd  [Thu 25 Nov 2021 02:08:11 PM EST] [DEBUG] Scanning for PluggedIn devices...  < Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub  < Bus 001 Device 002: ID 1bcf:0005 Sunplus Innovation Technology Inc. Optical Mouse  < Bus 001 Device 003: ID 8087:0aaa Intel Corp. Bluetooth 9460/9560 Jefferson Peak (JfP)  < Bus 001 Device 005: ID 2717:ff88 Xiaomi Inc. Mi/Redmi series (RNDIS + ADB)  < Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub  [Thu 25 Nov 2021 02:08:11 PM EST] [DEBUG] Disk Usage by frosty....  < 993M /home/frosty  [Thu 25 Nov 2021 02:08:11 PM EST] [DEBUG] Scanning for Network Interfaces....  < DEVICE TYPE STATE CONNECTION  < eth0 ethernet unavailable --  < lo loopback unmanaged --  < p2p-dev-wlan0 wifi-p2p disconnected --  < usb0 ethernet connected Wired connection 2  < wlan0 wifi connected Druid  [Thu 25 Nov 2021 02:08:21 PM EST] [DEBUG] Scanning Top 5 CPU Utilizing processes...  %CPU %MEM COMMAND  37.5 3.2 kazam  6.2 0.0 top  0.0 0.1 systemd  0.0 0.0 kthreadd  0.0 0.0 rcu\_gp  [Thu 25 Nov 2021 02:08:22 PM EST] [DEBUG] Disk Usage by frosty....  < 994M /home/frosty  ---  > 993M /home/frosty  [Thu 25 Nov 2021 02:09:19 PM EST] [DEBUG] Scanning for PluggedIn devices...  < Bus 001 Device 006: ID 8564:1000 Transcend Information, Inc. JetFlash  [Thu 25 Nov 2021 02:09:25 PM EST] [DEBUG] Showing current processes...  < frosty 1000 1539 ?  5a5  > root 0 4222 ?  [Thu 25 Nov 2021 02:09:35 PM EST] [DEBUG] Scanning for PluggedIn devices...  > Bus 001 Device 006: ID 8564:1000 Transcend Information, Inc. JetFlash  [Thu 25 Nov 2021 02:09:56 PM EST] [DEBUG] < SysMonitor SCRIPT\_EXITS\_HERE |

**Conclusion**

* The script automatically and regularly writes to the log file in easily readable format TIMESTAMP LOGLEVEL MESSAGE.
* During the first run it will scan for all the information and then only logs the changes that occurs to those set of information.
* The time interval for each scan is configurable via the SysMonitor.conf file.
* The script also have a DEBUG mode.
* The script can also run as a background process by adding an ampersand (&) at the end of script. Example: ./SysMonitor.sh &
* The script does not provide configurable log and level.
* It can’t export the generated log file into different format like JSON or HTML so it can be consumed by SIEM [6] software.
* The script might have performance bottleneck.

**Bibliography**

[1] Efficient Logging Mechanism [Available Online]: <https://www.cubicrace.com/2016/03/efficient-logging-mechnism-in-shell.html?m=1>

[2] Bash Shell redirect output and errors [Available Online]:

<https://www.google.com/amp/s/www.cyberciti.biz/faq/how-to-redirect-output-and-errors-to-devnull/>

[3] File Comparison and Difference commands [Available Online]:

<https://www.tecmint.com/best-linux-file-diff-tools-comparison/amp/>

[4] Shell Programming grep-sed-awk [Available Online]:

<https://www.google.com/amp/s/www.genuinecoder.com/unix-shell-programming-grep-sed-awk-html/%3famp>

[5] The Linux Documentation Project [Available Online]: <https://tldp.org>

[6] Linux Logs Explained [Available Online]: <https://www.google.com/amp/s/www.plesk.com/blog/featured/linux-logs-explained/amp/>

[7] Bash Built-in commands [Available Online]: <https://www.computerhope.com/unix/bash/index.htm>

**Appendix**

[1] sed – sed command stands for stream editor and it can perform lots of function on file like, searching, find and replace, insertion and much more.

[2] grep – grep command stand for globally search for regular expression and print out, the grep filter searches a file for a particular pattern of characters, and displays all lines that contains that pattern.

[3] cmp – cmp command is used to compare two files byte by byte. If a difference is found, it reports the byte and number where the difference is found. If no difference is found, by default, cmp returns no output.

[4] diff – diif command stands for difference. This command is used to display the difference in the files by comparing the files line by line.

[5] Keyboard Interrupt – Bash allows users to trap interrupts (generated by pressing

Ctrl-c) in the shell scripts.

[6] SIEM – (Security Information and Event Management) is a software solution that aggregates and analyses activity from many different resources across your entire IT infrastructure.