PSMonitor Script Documentation

Abraham Reines

February 20, 2024

1 Script Requirements

The `psmonitor.sh` script scans the system process table; it displays the process running on the system. A specific amount of intervals for a specified number of iterations is acheived. It takes two command-line arguments (CLAs) to set the time between scans and the number of scans.

2 Usage

The script is invoked with the following syntax:

```
./psmonitor.sh [-t tseconds] [-n count]
```

where `-t tseconds` specifies the time interval between each scan, and `-n count` specifies the number of times the scan is performed.

3 Error Checking

The script includes error checking for invalid options, missing argument values, and non-numeric input for time intervals and count.

4 Handling Interrupts

Interrupts, such as Ctrl-C, are gracefully handled by the script to provide a user-friendly exit message.

5 Script Listing

```
#!/bin/bash
# Author: Abraham Reines
# Date: 14-02-2024 09:35:07

tseconds=1
count=5

# Function for usage
show_usage() {
    echo "Usage: $0 [-t tseconds] [-n count]"
    exit 1
}

# Function to handle interrupts
handle_interrupt() {
    echo "Interruption occured. Exiting with grace..."
    exit 2
}

# SIGINT Trap (Ctrl-C)
trap handle_interrupt SIGINT

# Parsing
while getopts ":t:n:" opt; do
```

```
    case ${opt} in
        t )
            tseconds=$OPTARG
            ;;
        n )
            count=$OPTARG
            ;;
        \? )
            echo "Invalid Option: -$OPTARG" 1>&2
            show_usage
            ;;
        : )
            echo "Option -$OPTARG requires an argument." 1>&2
            show_usage
            ;;
    esac
done

# Check for integers
if ! [[ $tseconds =~ ^[0-9]+$ ]] || ! [[ $count =~ ^[0-9]+$ ]]; then
    echo "Error: tseconds and count must be positive and integers."
    exit 3
fi

# Main loop
for (( i=0; i<$count; i++ )); do
    echo $(date)
    ps -ef
    sleep $tseconds
done

echo
echo
echo "Program is written by Abraham Reines. This work complies with the JMU honor code. I did not give or receive unauthorized help on this assignment. Exiting..."
```

6 Sample Output Listing

For the first 10 lines:

```
Tue Feb 13 10:59:20 PST 2024
  UID   PID  PPID   C STIME   TTY           TIME CMD
    0     1     0   0  9:42AM ??         1:37.34 /sbin/launchd
    0   329     1   0  9:43AM ??         0:35.02 /usr/libexec/logd
    0   330     1   0  9:43AM ??         0:00.04 /usr/libexec/smd
    0   331     1   0  9:43AM ??         0:01.45 /usr/libexec/UserEventAgent (System)
    0   333     1   0  9:43AM ??         0:00.16 /System/Library/PrivateFrameworks/Uninstall.framework/Resources/uninstalld
    0   334     1   0  9:43AM ??         0:15.89 /System/Library/Frameworks/CoreServices.framework/Versions/A/Frameworks/FSEvents.framework/Versions/A/Support/fseventsd
    0   335     1   0  9:43AM ??         0:00.83 /System/Library/PrivateFrameworks/MediaRemote.framework/Support/mediaremoted
    0   338     1   0  9:43AM ??         0:00.90 /usr/sbin/systemstats --daemon
```

For the last 10 lines:

```
    0   856   671   0  9:43AM ttys001    0:00.01 login -pf abrahamreines
  501   861   856   0  9:43AM ttys001    0:00.10 -zsh
    0   860   671   0  9:43AM ttys002    0:00.01 login -pfl abrahamreines /bin/bash -c exec -la zsh /bin/zsh
  501   864   860   0  9:43AM ttys002    0:00.10 -zsh
    0   862   671   0  9:43AM ttys003    0:00.01 login -pfl abrahamreines /bin/bash -c exec -la zsh /bin/zsh
  501   886   862   0  9:43AM ttys003    0:00.10 -zsh
  501  8615  3718   0 10:55AM ttys009    0:00.12 /bin/zsh -il
  501 10168  8615   0 10:59AM ttys009    0:00.02 /bin/bash ./psmonitor.sh -t 1 -n 10
    0 10240 10168   0 10:59AM ttys009    0:00.01 ps -ef
```

```
Program is written by Abraham Reines. This work complies with the JMU honor code. I did not give or receive unauthorized help on this assignment. Exiting...
```

7 Executing the Script

To execute the script, first ensure it has the appropriate permissions set:

```
chmod +x psmonitor.sh
```

Then run the script by providing the desired arguments for time interval and count, e.g.:

```
./psmonitor.sh -t 1 -n 10 > psmonitor_output.txt
```

or use the default values by not providing any arguments.

Academic Integrity Pledge

"This work complies with the JMU honor code. I did not give or receive unauthorized help on this assignment."