PSMonitor Script Documentation

Abraham Reines

February 25, 2024

1 Script Requirements

The `psmonitor.sh` script displays the list of all the processes running in the system, once every tseconds for count number of times. A specific amount of intervals for a specified number of iterations is acheived. It takes two command-line arguments (CLAs) to repeatedly scan the system process table, displaying the list of processes running in the system.

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
# Modified: 2024-02-24 10:58:51

tseconds=1
count=5

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

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

```

```
# SIGINT Trap (Ctrl-C)
trap Interrupt?_time_to_end SIGINT

# Parsing
while getopts ":t:n:" opt; do
    case ${opt} in
        t )
            tseconds=$OPTARG
            ;;
        n )
            count=$OPTARG
            ;;
        \? )
            echo "Invalid Option: -$OPTARG" 1>&2
            use_me
            exit 1
            ;;
        : )
            echo "Option -$OPTARG requires an argument." 1>&2
            use_me
            exit 1
            ;;
    esac
done

# no options were specified? time to complain
if [ $OPTIND -eq 1 ]; then
    echo "No options were specified."
    use_me
    exit 1
fi

# 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
    echo
    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:

```
1
2
3  Sat Feb 24 10:39:50 PST 2024
4    UID   PID  PPID   C STIME   TTY           TIME CMD
5      0     1     0   0  8:55AM ??         1:21.64 /sbin/launchd
6      0   330     1   0  9:22AM ??         0:35.30 /usr/libexec/logd
7      0   331     1   0  9:22AM ??         0:00.05 /usr/libexec/smd
8      0   332     1   0  9:22AM ??         0:01.45 /usr/libexec/UserEventAgent
       ↪  (System)
9      0   334     1   0  9:22AM ??         0:00.17 /System/Library/
       ↪ PrivateFrameworks/Uninstall.framework/Resources/uninstalld
10     0   335     1   0  9:22AM ??         0:17.74 /System/Library/Frameworks/
       ↪ CoreServices.framework/Versions/A/Frameworks/FSEvents.framework/
       ↪ Versions/A/Support/fseventsd
```

For the last 5 lines:

```
1    501 16399     1   0 10:39AM ??         0:00.07 /System/Library/Frameworks/
       ↪ CoreServices.framework/Frameworks/Metadata.framework/Versions/A/
       ↪ Support/mdworker_shared -s mdworker -c MDSImporterWorker -m com.
       ↪ apple.mdworker.shared
2     89 16446     1   0 10:39AM ??         0:00.09 /System/Library/Frameworks/
        ↪ CoreServices.framework/Frameworks/Metadata.framework/Versions/A/
        ↪ Support/mdworker_shared -s mdworker -c MDSImporterWorker -m com.
        ↪ apple.mdworker.shared
3    501 14623 14622   0 10:33AM ttys001    0:00.22 /bin/zsh -il
4    501 16431 14623   0 10:39AM ttys001    0:00.02 /bin/bash ./psmonitor.sh -t
       ↪  1 -n 10
5      0 16499 16431   0 10:40AM ttys001    0:00.00 ps -ef
6
7
8  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...
```

Full output too long to display in this PDF. The output is intended to be submitted as a .txt file alogn with `psmonitor.sh`

7 Executing the Script

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

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

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

```
1  ./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."