PSMonitor Script Documentation

Abraham Reines

February 19, 2024

1 Script Requirements

The script `psmonitor.sh` is designed to scan the system process table and display all processes running in the system at specified intervals for a specified number of iterations. It takes two optional command-line arguments to customize 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 Sample Output

A sample output of the script includes the current date and time followed by the list of current processes. The output concludes with an attribution to the script's author and a statement of honor code compliance.

4 Error Checking

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

5 Handling Interrupts

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

6 Script Listing

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

tseconds=1
count=5

# Function to show 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
}
```

```
19
20 # Trap for SIGINT (Ctrl-C)
21 trap handle_interrupt SIGINT
22
23 # Parsing command-line options
24 while getopts ":t:n:" opt; do
25     case ${opt} in
26         t )
27             tseconds=$OPTARG
28             ;;
29         n )
30             count=$OPTARG
31             ;;
32         \? )
33             echo "Invalid Option: -$OPTARG" 1>&2
34             show_usage
35             ;;
36         : )
37             echo "Option -$OPTARG requires an argument." 1>&2
38             show_usage
39             ;;
40     esac
41 done
42
43 # Check for non-numeric values
44 if ! [[ $tseconds =~ ^[0-9]+$ ]] || ! [[ $count =~ ^[0-9]+$ ]]; then
45     echo "Error: tseconds and count must be positive and integers."
46     exit 3
47 fi
48
49 # Main loop
50 for (( i=0; i<$count; i++ )); do
51     echo $(date)
52     ps -ef
53     sleep $tseconds
54 done
55
56 echo
57 echo
58 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..."
```

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
```

or use the default values by not providing any arguments.