Mittuniversitetet
MID SWEDEN UNIVERSITY

DT149G Administration of UNIX-like systems

Laboratory Assignment— System Administration I

Lennart Franked*

October 25, 2024

Contents

1 Introduction

This lab will cover process management, scheduling and log files.

2 Aim

After completion of this assignment you will have:

- Become familiar with process handling, priorities and scheduling.
- Knowledge how logging works in a UNIX-like system.

*E-post: lennart.franked@miun.se.

3 Reading instructions

Before starting this assignment you should have read Nemeth et al., 2017, chapters 4, 10, 29

4 Tasks

Perform the following tasks and document all the steps taken to complete the tasks.

4.1 Process management

Before starting this task you should become familiar with the following commands: `ps (1)`, `nice (1)`, `renice (1)`, `nohup (1)` and `kill (1)`
You should also create a simple script that contains an infinite loop that you can use to run the commands against e.g.

```
#!/bin/bash
while true
do
  echo a > /dev/null
done
```

Give the script executable rights and start it in a new terminal

- With the help of `ps (1)` identify what process ID your newly created script got.
- Write down the priority and nice value of the process,
- restart the script with a lower priority, check what priority and nice value the process have now.
- now increase the scripts priority without restarting it:
  - without using sudo, how highly will you be able to prioritize the process?
  - with sudo, how highly will you be able to prioritize the process?
- now stop your script using `kill (1)` and `pkill (1)`
- start the script with the `nohup (1)` command after which you close the terminal. Using another terminal check to see if the script is still running.
- kill the script and rerun it using the `&` character, e.g.

  ```
  firefox&
  ```

  What happens?
- Once more rerun the script and suspend it using `<CTRL>-Z`. Now with the help of the commands `jobs (1)`, `bg (1)` and `fg (1)`, start the script in the foreground, then once again suspend the process and start the script in the background.

To answer in your report For this task, answer the following questions in your report:

- Explain the relationship between nice-value and priority
- What is the highest and lowest priority a user can set on a process?
- With a few sentances, desribe the different ways of running a script in the background, bringing it to the front.
- Give a practical use-case of the `nohup (1)` command.

4.2 Scheduling

Before starting this section you should get acquainted with `crontab (1)`, `at (1)` and `find (1)`

1. Create a script that removes all files in `/tmp` that haven't been access in the last two days. If you do not have a file that meet this criteria, you can change the timestamp with the help of `touch (1)`.
2. Configure `crontab (1)` to run your script every evening at 23:50.
3. Run the script 21:30 this evening.

To answer in your report For this task, answer the following questions in your report:

- What is the difference between running something with `Crontab (1)` and with `at (1)`?
- Include a screenshot of your modified crontab configuration file.

4.3 System statistics

Before starting this task you should see the man pages for `vmstat (8)`, `top (1)`, `last (1)`, `w (1)` and `uptime (1)`.

1. Try `vmstat (8)`, What can you find?
2. Try `top (1)`, what information are you able to retrieve?
3. How much memory and swap is used / configured on your machine?
4. Name the 10 users that last logged in on your machine.
5. Use `w (1)`, who is currently logged in on your machine?
6. What information are you able to retrieve with the `uptime (1)` command.

To answer in your report For this task, answer the following questions in your report:

- What is the difference between vmstat and top? What information can be found using both tools, and what information is unique for each of the tools?
- What is the difference between the three load-average values that are given with the uptime-command?
- Add a screenshot showing the last 10 users logged in on your system.

4.4 Log files

The following man pages should be read before starting this task, `dmesg (1)`, `logger (1)` and `rsyslogd (8)`

1. What information can be retrieved using `dmesg (1)`
2. Enter `/var/log/` and go through the different log files, see `rsyslog.conf (5)` for more information about the different log files.
3. Use `logger (1)` send a message to syslog at a facility.level for example `lpr.notice`, verify that your message is visible in the syslog.

To answer in your report For this task, answer the following questions in your report:

- What is the difference between syslog and dmesg?
- Add a screenshot showing the logmessage that you sent to syslog with `logger (1)`.

5 Examination

Hand in a report containing all your solutions to the questions in Section 4. *Remember that you must include references to the given reading instructions, alternatively to the laboratory work you have done*

References

Nemeth, E., Snyder, G., Hein, T. R., Whaley, B., & Mackin, D. (2017). *Unix and linux system administration handbook* (Fifth edition.). Addison-Wesley/Pearson.