CS 333 Operating System Lab

Autumn 2017

Department of Computer Science & Engineering
Indian Institute of Technology Bombay

Lab 1 : Linux tools and the proc file system

Goal: The goal of this lab is to get familiar with Linux tools and files used for system and process behaviour information and monitoring.

Important instructions:

1. Login as `labuser` for this assignment on the SL2 machines
2. This file is part the lab1.tar.gz archive which contains multiple directories with programs associated with exercise questions to follow.

Tools: Following are some basic Linux tools. The first step of this lab is to get familiar with the usage and capabilities of these tools.

To know more about them use: `man <command>`. Start with `man man`!

- **`top`**
  The `top` program provides a dynamic real-time view of a running system. It can display system summary information, as well as a list of processes or threads currently being managed by the kernel. The types of system summary information shown and the types, order and size of information displayed for tasks are all user-configurable.

- **`ps`**
  The `ps` command is used to view the processes running on a system. It provides a snapshot of the current processes along with detailed information like user id, cpu usage, memory usage, command name etc. It does not display data in real time like `top` or `htop` commands, but even though being simpler in features and output it is still an essential process management/monitoring tool that every linux newbie should know about and learn well.

- **`iostat`**
  `iostat` is a command used for monitoring input/output device usage by observing the time the devices are active in relation to their average transfer rates. `iostat` creates reports that can be used to change system configuration for better balance the input/output between physical disks.

- **`strace`**
  `strace` is a diagnostic, debugging and instructional userspace utility for Linux. It is used to monitor interactions between processes and the Linux kernel, which include system calls, signal deliveries, and changes of process state.

- **`lsof`**
  `lsof` is a tool used to list open files based on user, process, commands, network services, etc.

- **`lsblk`**
  `lsblk` is a tool used to list information about all available block devices such as hard disk, flash drives, CD-ROM…

- Also look up the following commands: `pstree`, `lshw, lspci, lscpu, dig, netstat, df, du.` (note that some of these may need root privileges)

The `/proc` file system

The `/proc` file system is a mechanism provided by Linux, so that kernel can send information to processes and also report information about the system and processes to users. This is a file interface provided to the user, to interact with the kernel and get the required information about processes running on the system. The `/proc` file system is nicely documented in the proc man page. You can access this document by running the command `man proc` on a Linux system. Understand systemwide proc files such as `meminfo`, `cpuinfo`, etc and process related files such as `status`, `stat`, `limits`, `maps` etc.

Exercises

1. Collect the following basic information about your machine using the `/proc` file system and answer the following questions:
   a. How many CPU and cores does the machine have?
   b. What is the frequency of each CPU ?
   c. How much memory does your system have?
   d. How much of it is free and available? What is the difference between them?
   e. What is total number of user-level processes in the system?
   f. How many context switches has the system performed since bootup?

2. Run all programs in the subdirectory `memory` and identify memory usage of each program. What is the meaning of the parameters `VmSize` and `VmRSS`?

   Compare the memory usage of these programs in terms of `VmSize` & `VmRSS` and justify your results based on code.

3. Run the executable `subprocesses` provided in the subdirectory `subprocess` and provide your roll number as input argument. Find the number of sub processes created by this program. Describe how you obtained the answer.

4. Run `strace` along with the binary program of `empty.c` given in subdirectory `strace`. What do you think the output of strace indicates in this case? How many different system call functions do you see?

   Next, use `strace` along with another binary program of `hello.c` (which is in the same directory).

   Compare the two `strace` outputs
   a. Which part of the `strace` output is common, and which part has to do with the specific program?
   b. List all unique system calls along with input and output parameters and overall functionality of each system call ?

5. Run the executable `openfiles` in subdirectory `files` and list the files which are opened by this program. Describe how you obtained the answer.

6. Find all the block devices on your system, their mount points and file systems present on them. Describe how you obtained the answer.

7. Create 5000 files starting from `foo1.pdf` to `foo5000.pdf using script.sh` given in the subdirectory `disk`.

   Run the programs `disk1` and `disk2` and identify the average disk utilization. Justify your answer with the help of code.

Submission details

- All submissions via moodle. Name your submissions as: `<rollno-lab1.tar.gz>`

- The tar should contain the following …
  - lab1.pdf which states the answer for each question and provides a justification for the same based on outputs of tools used. Make sure to provide snapshots or other details from tools used to justify.
  - In addition, for each question place relevant in the corresponding subdirectories. Filenames should relate to question number.
    For example: output-1a.txt for output of question 1a. Use further suffixes in case of multiple outputs.

- **Deadline: 22nd July 11.59 A.M.**
  Expected time for completion of lab: 2-3 hours