CS553 Homework #1

Linux & Computer Systems

Instructions:

- *Assigned date: Wednesday August 20th, 2025*
- *Due date: 11:59PM on Friday August 29th, 2025*
- *Maximum Points: 50*
- *This lab can be done in groups of 2 students*
- *Please post questions on Canvas*
- *Only a softcopy submission is required through GIT: push changes to GIT repository, and email confirmation will be sent to your HAWK email address at the deadline*
- *Late submission will be penalized at 20%; late submissions beyond 24 hours will receive 0 points*

1 Your Assignment

This project aims to teach you the basics of Linux and Virtual Machines, as well as some basic Q&A about computer systems. Any programming you do in this assignment will be limited to BASH scripting and/or Python. You can use any computer for this part of the assignment (e.g. your laptop running any OS).

Here is your assignment. Collect evidence of your work through screen shots and log files. You will write a report at the end that outlines what you have done to complete this assignment.

1. (10 points) Setup VM, Linux, and basic testing – must take screen shots at each step to receive points
    a. Read Oracle VirtualBox White Paper (https://www.oracle.com/a/ocom/docs/dc/em/oracle-vm-virtualbox-overview-2981353.pdf)
    b. Download Oracle VirtualBox 7.1.4 (https://www.virtualbox.org/wiki/Downloads)
    c. Install VirtualBox
    d. Download Ubuntu Server 24.04.1 LTS Linux (https://ubuntu.com/download/server) ISO image – no need to install the Desktop version
    e. Create Virtual Machine (VM), to support Linux, Ubuntu, 64-bit, 4GB RAM, Virtual Disk 25GB, VDI image, dynamically allocated, 2-core, and a network interface (1GbE or WiFi) with NAT support
    f. Install Linux from the ISO image
    g. Create a user id and password
    h. Turn on Firewall and block all ports
    i. Enable SSH access to your new Linux installation; open SSH port in firewall
    j. Repeat steps e through i, and create another VM with the same specifications as the first one
    k. Create private/public keys and install them properly in both of your new VMs
    l. Test that you can connect remotely to your VMs with your keys, from one VM to the other VM

2. (7 points) Show an example of using the following commands (hint: you can use man to find more information about each one); take screen shots of your commands; make sure to clear the screen between each command; explain in your own words what these commands do:

a. ssh	n. watch	aa. chmod	nn. mount
b. ssh-keygen	o. ps	bb. chown	oo. passwd
c. scp	p. top	cc. useradd	pp. uname
d. history	q. htop	dd. man	qq. whereis
e. sudo	r. gcc	ee. locate	rr. whatis
f. ip	s. tail	ff. find	ss. su
g. dd	t. grep	gg. sed	tt. ping
h. fdisk	u. kill	hh. awk	uu. traceroute
i. apt	v. killall	ii. diff	vv. date
j. vi	w. du	jj. sort	ww. time
k. time	x. df	kk. export	xx. wget
l. tar	y. screen	ll. pwd	yy. wc
m. cat	z. vim	mm. crontab	zz. pwgen

3. (10 points) Write bash scripts to do the following:
   a. Write a script called "generate-dataset.sh <filename> <num_records>" with two command line arguments specifying the file name to output and the number of records, where each record is separated by new line character, and each has the following format: <integer> <integer> <ASCII_string>. The integer should be any random number up to a 32-bit integer. The ASCII_string should be any string using ASCII of exactly 100 bytes long. Use the "time" command to show how long the benchmark took to complete. The benchmark should run for at least 10 seconds, and it should complete even if the ssh (or bash) session is terminated. How many records does your file contain after running it? You must write this script entirely with existing Linux commands (which you can install if they don't exist on your system), without using any other programming languages like C, C++, Java, or Python.
   b. Write a script called "sort-data.sh" that takes input a file from part (a) above and sorts the file based on the first column data (make sure to only sort based on the first column data, and not on the entire line of data; also make sure you are treating the data in column 1 as numbers and not text). Use the "time" command to show how long the sort script took to complete.
   c. Use the script in part (a) and generate 3 data files with different number of records (1000, 100000, 10000000); measure time taken to generate these records. Sort the data with your script from part (b) and measure the time. Write a Python matplotlib script to generate a graph for the time taken to generate the data and the time taken to sort the data at the 3 different scales. The graph should automatically adjust to the number of entries, and the scale of the data.

4. (8 points) Answer the following questions about VMs:
   a. In the system configuration of the VM, explain how changing the number of processors changes the behavior of your VM. Explain a scenario where you want to set this to the minimum, and a scenario where you want to set it to the maximum. Why is setting it to the maximum potentially a bad idea?

b. In the system configuration of the VM, under the Acceleration Tab, explain the difference between the paravirtualization options: None, Legacy, Minimal, Hyper-V, and KVM. Explain which one would be best to use with Ubuntu Linux, and why.

c. In storage devices when configuring the VM, there are multiple types of storage controllers: explain the difference between the IDE, SATA, and NVMe controller. Give an example for each type of storage controller of a scenario where you may want to use this type of controller.

d. In the network configuration of the VM, there are multiple types of network adapters: explain the difference between NAT, Bridged Adapter, Internal Network, and Host-only Network. Give an example for each type of network of a scenario where you may want to use this type of network.

e. For the USB configuration of the VM, explain the difference between USB 1.1, 2.0, and 3.0 controllers.

5. (10 points) Answer the following questions about computer processors:
   a. Describe what a core and hardware thread is on a modern processor, and the difference between them?
   b. How many cores do the fastest processors from each manufacturer have? Give an example (specific model, specs, and price).
      (a) Intel CPU (x86)
      (b) AMD CPU (x86)
      (c) IBM CPU (Power9)
      (d) ThunderX CPU (ARM)
      (e) NVIDIA GPU
   c. Why do we not have processors running at 1THz today (as might have been predicted in the year 2000)?
   d. Describe Moore's Law. Is it going to go on forever? If not, when will it end? Justify your answer to why it will end and when.

6. (5 points) Answer the following questions about threading:
   a. Why is threading useful on a single-core processor?
   b. Do more threads always mean better performance?
   c. Is super-linear speedup possible? Explain why or why not.
   d. Why are locks needed in a multi-threaded program?
   e. Would it make sense to limit the number of threads in a server process?

2 Where you will submit

You will have to submit your solution to a private git repository created for you at https://classroom.github.com/a/xjbwHwc9. The repository is create through GitHub Classroom and you will need to accept the assignment before you can clone it, by accessing this invitation link: https://classroom.github.com/a/xjbwHwc9. The first time you access an invitation link the system will ask you to identify yourself. Please select the identifier that contains your school email address. After that you can clone the repository. Then you will have to add or update your source code, documentation, and report. All repositories will be collected automatically at the deadline. One final collection will be made 24 hours after the deadline. It is your responsibility to make sure your submission is correct and complete in the GIT repo. If you cannot access your repository contact the TAs. You can find a git cheat sheet here: https://www.git-tower.com/blog/git-cheat-sheet/

3 What you will submit

When you have finished implementing the complete assignment as described above, you should submit your solution to your private git repository. Each program must work correctly and be detailed in-line documented. You should hand in:

1. **Source code:** All of the source code, including proper documentation and formatting.
2. **Readme:** A detailed manual describing the structure of your files and directory organization. The manual should be able to instruct users how to run the program step by step. The manual should contain example commands. This should be included as readme.txt in the source code folder.
3. **Report:** A written document (typed, named hw1-report.pdf) describing the overall assignment completion, along with screen shots and text answering the key questions. Make sure to label answers with the appropriate question number from the homework writeup.
4. **AI Assistance History:** A PDF file that has the entire history of your LLM questions and answers that are relevant to this assignment. You are encouraged to use AI to help you figure out the material in this assignment, but you are not supposed to let the AI do the assignment for you. Use the AI assistant as a tutor or guide. If you did not use any AI assistance, please include a PDF file that states that you did not use any AI assistance. Identify what other sources of information you used in the completion of this assignment.

Submit step: code/report through GIT.

Grades for late programs will be lowered 20%; no submission will be accepted more than 24 hours late