CS550 Homework #1

Linux & Computer Systems

Instructions:

- *Assigned date: Tuesday September 5th, 2023*
- *Due date: 11:59PM on Tuesday September 19th, 2023*
- *Maximum Points: 100%*
- *This homework can be done in groups up to 3 students*
- *Please post your questions to BB*
- *Only a softcopy submission is required; it will automatically be collected through GIT after the deadline;*
- *Late submission will be penalized at 20% per day; an email to the TA with the subject "CS550: late homework submission" must be sent*

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. (30 points) Setup VM, Linux, and basic testing – must take screen shots at each step to receive points
    a. Read Oracle VirtualBox White Paper
    b. Download Oracle VirtualBox 7.0
    c. Install VirtualBox 7.0 (for M1/M2 Apple, use UTM)
    d. Download Ubuntu 22.04 Linux ISO image
    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 Bridged Adapter
    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 5 through 9, 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. (16 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	c.	scp	e.	sudo	g.	touch
b.	ssh-keygen	d.	history	f.	ip	h.	ls

i.	mkdir	w.	top	kk.	mv	yy.	whereis
j.	cd	x.	htop	ll.	man	zz.	whatis
k.	dd	y.	gcc	mm.	locate	aaa.	less
l.	fdisk	z.	tail	nn.	find	bbb.	su
m.	apt	aa.	grep	oo.	sed	ccc.	ping
n.	vi	bb.	kill	pp.	awk	ddd.	traceroute
o.	time	cc.	killall	qq.	diff	eee.	date
p.	tar	dd.	du	rr.	sort	fff.	time
q.	rm	ee.	df	ss.	export	ggg.	wget
r.	cat	ff.	screen	tt.	pwd	hhh.	wc
s.	bash	gg.	vim	uu.	crontab	iii.	clear
t.	more	hh.	chmod	vv.	mount	jjj.	exit
u.	watch	ii.	chown	ww.	passwd		
v.	ps	jj.	useradd	xx.	uname		

3. (15 points) Write bash scripts to do the following:
    a. Write a script called “disk-benchmark-background.sh” that uses the dd command to run a benchmark against the local disk in the background, that captures all the output (both standard out and error output) to a file “disk-benchmark-background-log.txt”. 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.
    b. Write a script called “network-test.sh” that takes input a file “network-test-machine-list.txt” with a list of DNS names (e.g. google.com, iit.edu, anl.gov), each name on a separate line, and runs the ping utility collecting 3 samples from each DNS name, and writing the RTT (round trip time) average latency into a file “network-test-latency.txt” where each line will have the DNS name and average RTT separated by a space. Make sure it works with at least 10 DNS names, but it should work for an unspecified number of DNS names.
    c. Write a Python matplotlib script to generate a graph of the “network-test-latency.txt” data. The graph should automatically adjust to the number of entries, and the scale of the data.
4. (15 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. (12 points) Answer the following questions about computer processors:
    a. Today's commodity processors have 1 to 96 cores, and specialized GPUs having 10000+ CUDA-cores. About how many cores/threads are expected to be in future commodity processors in the next five years?
    b. Describe what a core and hardware thread is on a modern processor, and the difference between them?
    c. Compare GPU and CPU chips in terms of their strength and weakness. In particular, discuss the tradeoffs between power efficiency, programmability and performance.
    d. Why do we not have processors running at 100GHz today (as might have been predicted in the year 2000)?
6. (12 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/9e-hLfci. You will have to firstly clone the repository. Then you will have to add or update your source code, documentation and report. Your solution will be collected automatically after the deadline. If you want to submit your homework later, you will have to push your final version to your GIT repository and you will have let the TA know of it through email. There is no need to submit anything on BB for this assignment. 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.

Submit code/report through GIT.

Grades for late programs will be lowered 20% per day late.