Michael Flanagan

CIS 140U Lab 7

Chemeketa CC Online

Summer 2017

**1. Run the top command and explain each of the columns of information that is shown as well as describe in detail the first 5 lines of the output of the top command (you may have to do some research).**

PID = Process ID. This can be used to send a signal by mentioning the PID and the signal number

User = User ID. Fairly self-explanatory shows the userid that called that process.

PR = Priority. Lets you know the place that the process gets in the priority queue. Lower numbers get higher priority. This helps with preventing system crashes, CPU intensive items will get incremented to be lower priority, while less CPU intensive processes will get higher priority this helps balance out system processes so that CPU intensive items don’t slow the system to a halt.

NI = Nice Level = Another priority indicator, however this is a static priority assigned to an ask at the time its initialized.

VIRT = Virtual memory used by process = Virtual memory is an essentially fake memory that is stored on the hard disk but is treated by the kernel as random-access memory this helps prevent issues with memory overflow from standard memory in the computer system.

*https://www.lifewire.com/linux-top-command-2201163*

*https://www.linux.com/learn/uncover-meaning-tops-statistics*

**2. What command(s) would show you the same information as outputted on the first line from the top command?**

uptime

**3. Provide the process behind “killing” a process that is being outputted using the top command. Include in your answer as to who has the rights to kill processes.**

A non-privileged user can only influence their own processes. Users with higher level privileges such as system admins may kill lower users processes according to my interpretation of section 4.3.5.7 in the online book. When you kill a process, you stop it in its tracks, this might be needed if you have some badly written software that is causing a memory leak, or maybe some buggy software that is infinitely looping, or possibly a virus that is designed to crash the system by hogging system resources. If things aren’t working right and you have processes that is not familiar it would be good to try killing that process instead of just restarting the whole system which may have other users logged in and working at the time.

**4. Which option can be used with the top command to ensure that no processes can be “killed” during that commands execution?**

-s

**5. Run the ps command. Describe what this is a list of, and the process of “killing” a process within this list.**

ps displays information about a selection of the active processes.

I used man ps to get this information.

You can type kill <PID>.

Example:

/home/mflanagan ->ps

PID TTY TIME CMD

6403 pts/5 00:00:00 bash

6545 pts/5 00:00:00 ps

Kill 6403 would than kill bash. The system won’t actually let us kill bash because it’s not a process or job ID but if it was a process or job ID we could than kill it with that command.

**6. What command can be issued to “kill” a process that is not responding to the appropriate command? Explain in detail why you must issue the command you came up with to accomplish this task.**

**SIGKILL – 9 - <PID>**

The SIGKILL signal cannot be ignored by a process and the termination is handled outside of the process itself. This signal is useful for when an application has stopped responding.

*http://bencane.com/2014/04/01/understanding-the-kill-command-and-how-to-terminate-processes-in-linux/*

**7. Run the command vi& ( yes that is an ampersand ). What just occurred? Where is the vi program? What command will allow you to start using the vi program that is currently in the background? Explain in detail what just occurred.**

We started vi as a process that is now running in the background.

We can run it by saying screen vi&

**8. Run the command vi& ps. Write down the result of this command. Now issue the command kill PID (for the PID put in the process id number shown in the ps output for the vi program). Now run the ps command again, write down the results. What occurred? Was it what you expected? What option may be needed with kill to ensure that the process is terminated?**

It showed vi running. Than when I try to kill using the PID it wouldn’t allow it. This was not what I expected, I had to look online to figure out a surefire way to kill the process. I was able to kill it using kill -SIGKILL <PID>

*http://bencane.com/2014/04/01/understanding-the-kill-command-and-how-to-terminate-processes-in-linux/*

**9. Run the ifconfig command. What does this show? How many network adapters are shown? What is the internet address of the network adapter** named **eth0?**

**1 – 192.168.32.128**

**10. Run the ping command on the IP address that you recorded from above, along with the option –w 5. Write the command and output and explain the purpose of the ping command, the option used, and what our test tells us about our server’s network connection (read the man on ping).**

Ping = ping uses the ICMP protocol's mandatory ECHO\_REQUEST datagram to elicit

an ICMP ECHO\_RESPONSE from a host or gateway. ECHO\_REQUEST datagrams

(``pings'') have an IP and ICMP header, followed by a struct timeval

and then an arbitrary number of ``pad'' bytes used to fill out the

packet.

-w5 = -w deadline

Specify a timeout, in seconds, before ping exits regardless of

how many packets have been sent or received. In this case ping

does not stop after count packet are sent, it waits either for

deadline expire or until count probes are answered or for some

error notification from network.

ICMP PACKET DETAILS

An IP header without options is 20 bytes. An ICMP ECHO\_REQUEST packet

contains an additional 8 bytes worth of ICMP header followed by an

arbitrary amount of data. When a packetsize is given, this indicated

the size of this extra piece of data (the default is 56). Thus the

amount of data received inside of an IP packet of type ICMP ECHO\_REPLY

will always be 8 bytes more than the requested data space (the ICMP

header).

If the data space is at least of size of struct timeval ping uses the

beginning bytes of this space to include a timestamp which it uses in

the computation of round trip times. If the data space is shorter, no

round trip times are given.

TTL DETAILS

The TTL value of an IP packet represents the maximum number of IP

routers that the packet can go through before being thrown away. In

current practice you can expect each router in the Internet to decre‐

ment the TTL field by exactly one.

The TCP/IP specification states that the TTL field for TCP packets

should be set to 60, but many systems use smaller values (4.3 BSD uses

30, 4.2 used 15).

The maximum possible value of this field is 255, and most Unix systems

set the TTL field of ICMP ECHO\_REQUEST packets to 255. This is why you

will find you can ``ping'' some hosts, but not reach them with tel‐

net(1) or ftp(1).

TIME DETAILS

The amount of time it took to send a message to that ip address and receive a response back.

In layman’s terms, we sent a message to 192.168.32.128, we expect to receive a response back from that computer ping will give us the milliseconds that it took to reach that destination. The w5 part tells it to ping that location 5 times, if we don’t specify the w5 than the terminal will just keep pinging forever.

**11. Run the command netstat –lnt. Write down the output. What does this command do? Explain each of the options ( -lnt ). After reading the manual about the netstat command, please provide a brief summary of the purpose and capabilities of this command.**

Kernel Interface table

Iface MTU Met RX-OK RX-ERR RX-DRP RX-OVR TX-OK TX-ERR TX-DRP TX-OVR Flg

ens33 1500 0 224 0 0 0 133 0 0 0 BMRU

lo 65536 0 307 0 0 0 307 0 0 0 LRU

--verbose , -v

Tell the user what is going on by being verbose. Especially print some useful information about unconfigured address families.

--wide , -W

Do not truncate IP addresses by using output as wide as needed. This is optional for now to not break existing scripts.

--numeric , -n

Show numerical addresses instead of trying to determine symbolic host, port or user names.

--numeric-hosts

shows numerical host addresses but does not affect the resolution of port or user names.

--numeric-ports

shows numerical port numbers but does not affect the resolution of host or user names.

--numeric-users

shows numerical user IDs but does not affect the resolution of host or port names.

--protocol=family , -A

Specifies the address families (perhaps better described as low level protocols) for which connections are to be shown. family is a comma (',') separated list of

address family keywords like inet, unix, ipx, ax25, netrom, and ddp. This has the same effect as using the --inet, --unix (-x), --ipx, --ax25, --netrom, and --ddp

options.

The address family inet includes raw, udp and tcp protocol sockets.

-c, --continuous

This will cause netstat to print the selected information every second continuously.

-e, --extend

Display additional information. Use this option twice for maximum detail.

-o, --timers

Include information related to networking timers.

-p, --program

Show the PID and name of the program to which each socket belongs.

-l, --listening

Show only listening sockets. (These are omitted by default.)

-a, --all

Show both listening and non-listening sockets. With the --interfaces option, show interfaces that are not up

-F

Print routing information from the FIB. (This is the default.)

-C

Print routing information from the route cache.

**12. Run the command whois google.com. What does this command do? What information is available that may be useful to a system administrator?**

The whois command displays ownership information for the domain that is called by the whois command. It shows registrar information and ownership information which may be useful in locating the owner of a specific URL to do updates so that sites are compatible with current Operating systems, browsers etc…

**13. Run the command traceroute google.com. What does this command do? What information is available that may be useful to a system administrator?**

It gives the number of hops to get to that specific ipaddress (hops are the internet backbone servers located between your current physical location and the physical location of the server containing the URL address, and/or IP address of the location your trying to get the traceroute for). The traceroute will also give the time it took to hit each backbone server in milliseconds. This can be useful in determining which internet backbone is having an outage if users are unable to pull up a website. IF it just gets stuck on a hop forever that means an internet backbone server is down and there was no other route for the ping to go.

**4. How to turn in. Please use a word processor to compose your answers\*. After you have completed your lab, submit it as an attachment to the Lab 7 assignment turn in folder on Blackboard.**

**\*Check your work to be sure that it is professionally formatted, and free**