Step 1: Create, Extract, Compress, and Manage tar Backup Archives

1. Command to **extract** the TarDocs.tar archive to the current directory:
   tar xvvf TarDocs.tar
   a. Running 'ls' shows "TarDocs TarDocs.tar"
   b. ls -l shows TarDocs as a directory
2. Command to **create** the Javaless_Doc.tar archive from the TarDocs/ directory, while excluding the TarDocs/Documents/Java directory:
   a. Command: tar cvwf Javaless_Docs.tar --exclude='./TarDocs/Documents/Javaa/*' TarDocs/
      i. --exclude in 'man tar' states it will exclude files matching a pattern.

3. Command to ensure Java/ is not in the new Javaless_Docs.tar archive:
   a. tar tvvf Javaless_Docs.tar | grep Java

Bonus

- Command to create an incremental archive called logs_backup_tar.gz with only changed files to snapshot.file for the /var/log directory:
  - sudo tar czvvwf logs_backup.tar.gz --listed-incremental=logs_backup.snar --level=0 /var/log

Critical Analysis Question

- Why wouldn't you use the options -x and -c at the same time with tar?
  - Because the -c is to create and the -x is to extract. It would be redundant to create a tar archive then extract it immediately. We need to use -c to create then when the time is right we will use -x to extract it.

Step 2: Create, Manage, and Automate Cron Jobs

1. Cron job for backing up the /var/log/auth.log file: See next page for screenshot.
   a. 0 6 * * 3 tar -zcf /var/backups/auth_backup.tgz /var/log/auth.log
      i. 0 6 * * 3 is "At 06:00 on Wednesday"
      ii. tar -zcf creates a .tgz backup of /var/log/auth.log and saves in /var/backups/auth_backup.tgz
   b. To create a sha-256, you would run sha256sum /var/backups/auth_backup.tgz

```
sysadmin@UbuntuDesktop: ~
File  Edit  View  Search  Terminal  Help
  GNU nano 2.9.3                /tmp/crontab.AfL5Q0/crontab                Modified

# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/

#Create archive of /var/log/auth.log
0 6 * * 3 tar -zcf /var/backups/auth_backup.tgz /var/log/auth.log

# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Spell    ^_ Go To Line
```

crontab guru

The quick and simple editor for cron schedule expressions by Cronitor

"At 06:00 on Wednesday."

next at 2021-08-04 06:00:00 random

0 6 * * 3

minute	hour	day (month)	month	day (week)

*	any value
,	value list separator
-	range of values
/	step values
@yearly	(non-standard)
@annually	(non-standard)
@monthly	(non-standard)
@weekly	(non-standard)
@daily	(non-standard)
@hourly	(non-standard)
@reboot	(non-standard)

Step 3: Write Basic Bash Scripts

1. Brace expansion command to create the four subdirectories:
    a. First ran mkdir backups to create the backups file. Then, using brace expansion I created the four directories using: mkdir ~/backups/{freemem,disuse,openlist,freedisk}

Paste your system.sh script edits below:

```
#!/bin/bash
# INSTRUCTIONS: Edit the following placeholder command and output filepaths
# For example: cpu_usage_tool > ~/backups/cpuuse/cpu_usage.txt
# The cpu_usage_tool is the command and ~/backups/cpuuse/cpu_usage.txt is the filepath
# In the above example, the `cpu_usage_tool` command will output CPU usage information into
a `cpu_u$
# Do not forget to use the -h option for free memory, disk usage, and free disk space

# Free memory output to a free_mem.txt file
free > ~/backups/freemem/free_mem.txt

# Disk usage output to a disk_usage.txt file
df --output=source,used > ~/backups/diskuse/disk_usage.txt

# List open files to a open_list.txt file
lsof > ~/backups/openlist/open_list.txt

# Free disk space to a free_disk.txt file
df --output=source,avail > ~/backups/freedisk/free_disk.txt
```

    b. Sources for the commands:
        i. lsof: https://www.tecmint.com/find-out-who-is-using-a-file-in-linux/
        ii. df options were from 'man df'
        iii. free command was taught in class.

2. Command to make the system.sh script executable:
    a. chmod +x system.sh

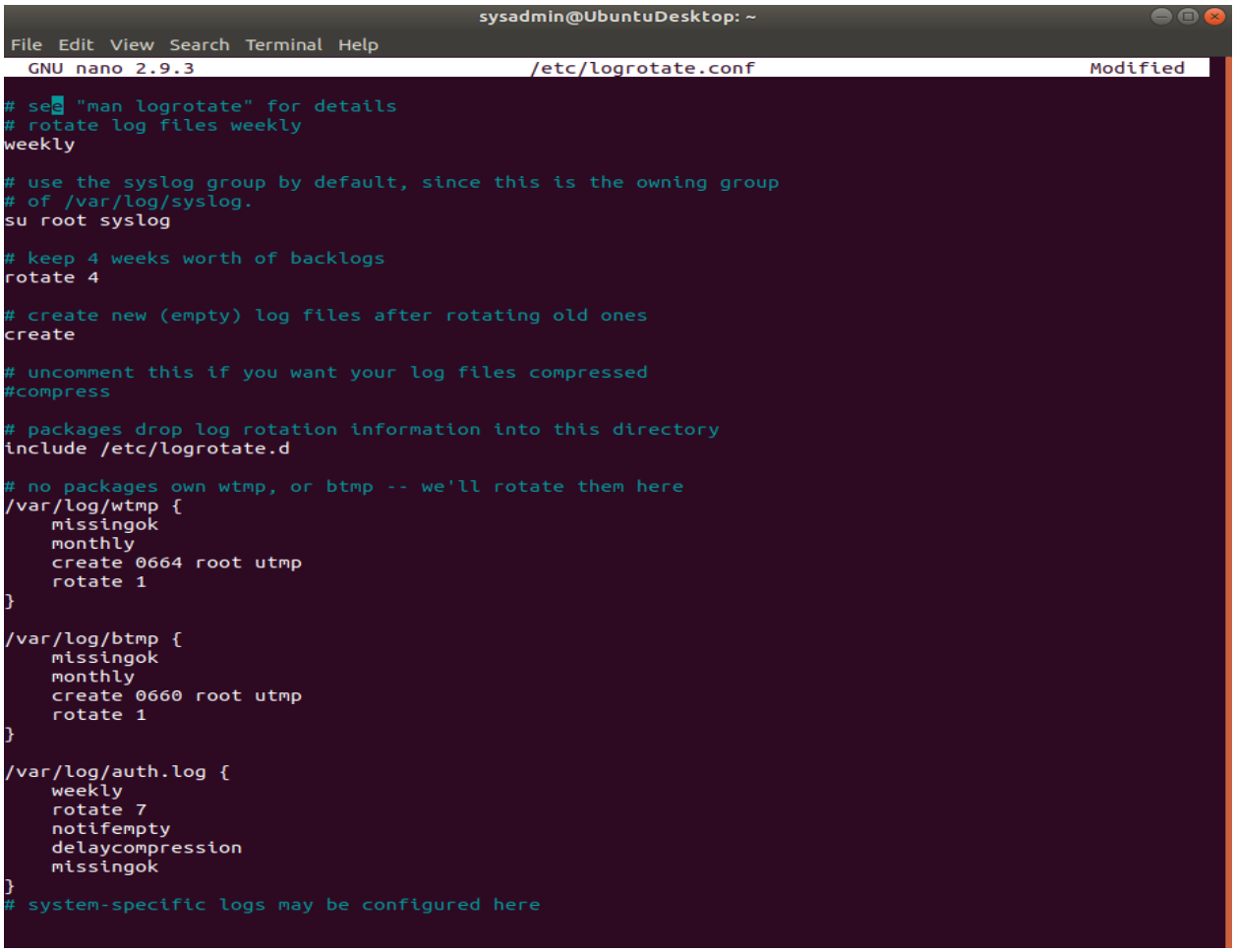
Optional

- Commands to test the script and confirm its execution:
    - ./system.sh to run the scripts. Then:
    - cat ~/backups/freemem/free_mem.txt
    - cat ~/backups/diskuse/disk_usage.txt
    - cat ~/backups/openlist/open_list.txt
    - cat ~/backups/freedisk/free_disk.txt

Step 4. Manage Log File Sizes

1. Run sudo nano /etc/logrotate.conf to edit the logrotate configuration file.
2. Configure a log rotation scheme that backs up authentication messages to the /var/log/auth.log.

```
/var/log/auth.log {
        weekly
        rotate 7
        notifempty
        delaycompression
        missingok
}
```



Bonus (Research Activity): Perform Various Log Filtering Techniques

1. Command to return journalctl messages with priorities from emergency to error:

    a. journalctl --p “emerg”..”err”

2. Command to check the disk usage of the system journal unit since the most recent boot:

    a. journalctl --disk-usage

3. Comand to remove all archived journal files except the most recent two:

    a. journalctl --rotate

    b. journalctl --vacuum-time=1s

        i. https://www.linuxuprising.com/2019/10/how-to-clean-up-systemd-journal-logs.html

4. Command to filter all log messages with priority levels between zero and two, and save output to /home/sysadmin/Priority_High.txt:

    a. journalctl -p “emerg”..”crit” > /home/sysadmin/Priority_High.txt

5. Command to automate the last command in a daily cronjob. Add the edits made to the crontab file below:

    a. Since there was no given date, I pick 3 a.m. on Friday of every week.

        0 3 * * 5 journalctl -p “emerg”..”crit” > /home/sysadmin/Priority_High.txt