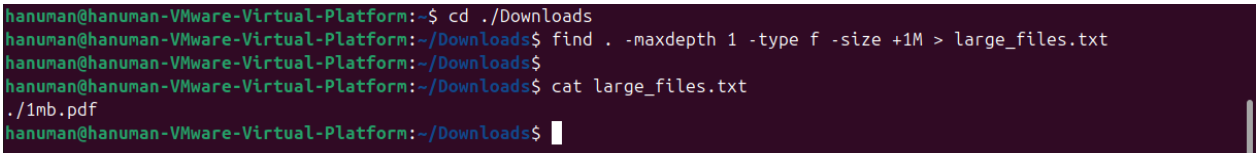
1) Use the appropriate command to list all files larger than 1 MB in the current directory and save the output to a file.



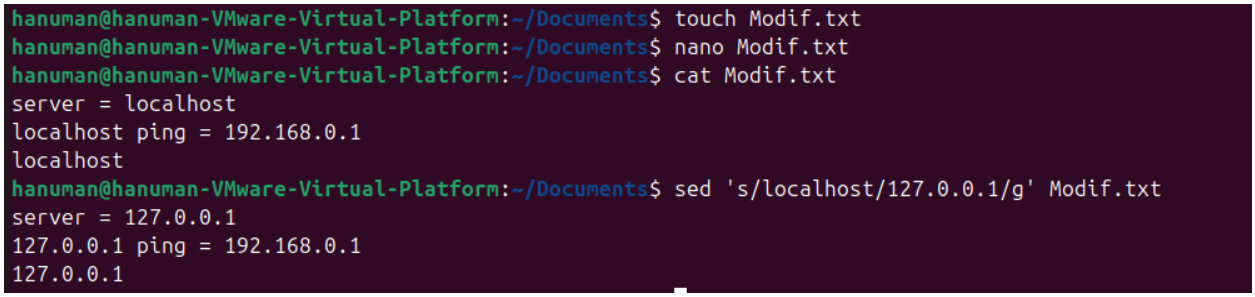
Command: `find . -maxdepth 1 -type f -size +1M > large_files.txt`

- **`find .`**: Tells Linux to start looking in the current directory (represented by the dot).
- **`-maxdepth 1`**: This ensures it only looks in the current folder and doesn't go digging into subfolders.
- **`-type f`**: Restricts the search to **files** only (excluding directories).
- **`-size +1M`**: The + sign means "greater than." You can also use `k` for Kilobytes or `G` for Gigabytes.
- **`>`**: This is a **redirection operator**. Instead of printing the names on your screen, it "pushes" them into the file `large_files.txt`.

Verification:

- command: cat large_files, this command will display the files larger than 1MB. In my VM I have only one file named 1mb.pdf(A sample pdf downloaded from google) which is of 1MB.

2) Replace all occurrences of "localhost" with "127.0.0.1" in a configuration file named config.txt, and save the updated file as updated_config.txt.



Command: `sed 's/localhost/127.0.0.1/g' config.txt > updated_config.txt`

- **`sed`**: Stands for "Stream Editor." It processes text line-by-line.
- **`s/old/new/`**: The standard syntax for substitution.

- **`/g` (Global)**: Without this, `sed` only replaces the *first* "localhost" it finds on each line. The `g` tells it to replace every single one.
- **Original vs. New**: Note that we use > to save a **new** file. If you wanted to change the file "in-place" without creating a new one, you would use the `-i` flag (`sed -i ...`).

Verification:

- The modif.txt is now updated. The word localhost has been replaced with 127.0.0.1

3) Use the appropriate command to search for lines containing the word "ERROR" in a log file but exclude lines containing "DEBUG". Save the results to a file named filtered_log.txt.

log:

DEBUG: Starting the application initialization.
ERROR: Unable to connect to the database.
INFO: User 'admin' logged in successfully.
DEBUG: Fetching configuration settings from the server.
ERROR: DEBUG - Configuration settings could not be applied.
INFO: Scheduled job 'backup' completed successfully.
DEBUG: Connection to server timed out. Retrying...
ERROR: Failed to fetch data from API endpoint '/users'.
INFO: Maintenance mode activated.
ERROR: DEBUG - Query execution failed due to a syntax error.
DEBUG: Reloading application modules.
ERROR: Missing required parameter in the request.
INFO: Shutting down the system gracefully.
DEBUG: Closing unused network connections.
ERROR: DEBUG - Unexpected server response received.

```
hanuman@hanuman-VMware-Virtual-Platform:~/Documents$ echo -e "DEBUG: Starting the application initialization.\nERROR: Un
able to connect to the database.\nINFO: User 'admin' logged in successfully.\nDEBUG: Fetching configuration settings fro
m the server.\nERROR: DEBUG - Configuration settings could not be applied.\nINFO: Scheduled job 'backup' completed succe
ssfully.\nDEBUG: Connection to server timed out. Retrying...\nERROR: Failed to fetch data from API endpoint '/users'.\nI
NFO: Maintenance mode activated.\nERROR: DEBUG - Query execution failed due to a syntax error.\nDEBUG: Reloading applica
tion modules.\nERROR: Missing required parameter in the request.\nINFO: Shutting down the system gracefully.\nDEBUG: Clo
sing unused network connections.\nERROR: DEBUG - Unexpected server response received." > logfile.log
hanuman@hanuman-VMware-Virtual-Platform:~/Documents$
hanuman@hanuman-VMware-Virtual-Platform:~/Documents$ awk '/ERROR/ && !/DEBUG/' logfile.log > filtered_log.txt
hanuman@hanuman-VMware-Virtual-Platform:~/Documents$
hanuman@hanuman-VMware-Virtual-Platform:~/Documents$ cat filtered_log.txt
ERROR: Unable to connect to the database.
ERROR: Failed to fetch data from API endpoint '/users'.
ERROR: Missing required parameter in the request.
```

Command: `grep "ERROR" log.txt | grep -v "DEBUG" > filtered_log.txt`

- **`grep "ERROR"`**: Searches `log.txt` and pulls out every line containing that word.

- **The Pipe (`|`)**: This is one of the most powerful tools in Linux. It takes the output of the first command and feeds it as the input to the next.
- **`grep -v "DEBUG"`**: The `-v` flag stands for **invert-match**. It tells the system: "Keep everything *except* the lines that contain 'DEBUG'."
    - In the log file, there is a line that says `ERROR: DEBUG - by` using the second grep, we successfully filter that out as requested.

Verification:

- In the output file, filtered_log.txt, the word 'ERROR' has been searched but the lines with the word 'DEBUG' has been neglected.

4) Write a code to identify the process with the highest memory usage and then terminate it.

```
hanuman@hanuman-VMware-Virtual-Platform:~$ ps aux --sort=-%mem --columns=150 | head -n 1
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
hanuman@hanuman-VMware-Virtual-Platform:~$ top

top - 22:25:16 up  4:44,  1 user,  load average: 0.52, 0.71, 0.53
Tasks: 319 total,   1 running, 318 sleeping,   0 stopped,   0 zombie
%Cpu(s):  1.2 us,  1.9 sy,  0.0 ni, 96.8 id,  0.2 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   1919.4 total,     93.7 free,   1418.4 used,    423.1 buff/cache
MiB Swap:   2048.0 total,   1024.1 free,   1023.9 used.    501.0 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   2068 hanuman   20   0 3961164 146012  71528 S   4.6   7.4   7:51.41 gnome-shell
   8000 hanuman   20   0  629112  46456  39788 S   1.7   2.4   0:06.60 gnome-terminal-
   5764 hanuman   20   0   11.5g 369940 113828 S   0.7  18.8   5:29.95 firefox
   8214 hanuman   20   0   14516   5844   3612 R   0.7   0.3   0:00.12 top
    314 root      20   0       0      0      0 S   0.3   0.0   0:02.00 jbd2/sda2-8
    504 systemd+  20   0   17564   5420   5268 S   0.3   0.3   0:11.15 systemd-oomd
    685 root     -51   0       0      0      0 S   0.3   0.0   0:05.02 irq/16-vmwgfx
   4130 hanuman   20   0 1301616 106012  57576 S   0.3   5.4   1:19.42 soffice.bin
   6342 hanuman   20   0 2893756 117140  47660 S   0.3   6.0   4:23.04 Isolated Web Co
   6864 root      20   0       0      0      0 I   0.3   0.0   0:03.99 kworker/0:2-mm_percpu_wq
   7837 root      20   0       0      0      0 I   0.3   0.0   0:00.66 kworker/u513:3-writeback
      1 root      20   0   23256   8912   6508 S   0.0   0.5   0:03.59 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.08 kthreadd
      3 root      20   0       0      0      0 S   0.0   0.0   0:00.00 pool_workqueue_release
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-rcu_gp
      5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-sync_wq
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-kvfree_rcu_reclaim
      7 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-slub_flushwq
      8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-netns
```

```
  6342 hanuman   20   0 2893756 117140  47660 S   0.3   6.0   4:23.04 Isolated Web Co
  6864 root      20   0       0      0      0 I   0.3   0.0   0:03.99 kworker/0:2-mm_percpu_wq
  7837 root      20   0       0      0      0 I   0.3   0.0   0:00.66 kworker/u513:3-writeback
     1 root      20   0   23256   8912   6508 S   0.0   0.5   0:03.59 systemd
     2 root      20   0       0      0      0 S   0.0   0.0   0:00.08 kthreadd
     3 root      20   0       0      0      0 S   0.0   0.0   0:00.00 pool_workqueue_release
     4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-rcu_gp
     5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-sync_wq
     6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-kvfree_rcu_reclaim
     7 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-slub_flushwq
     8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-netns
    11 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-events_highpri
    12 root      20   0       0      0      0 I   0.0   0.0   0:00.00 kworker/u512:0-ipv6_addrconf
    13 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-mm_percpu_wq
    14 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_kthread
    15 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_rude_kthread
    16 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_trace_kthread
hanuman@hanuman-VMware-Virtual-Platform:~$ kill 5764
hanuman@hanuman-VMware-Virtual-Platform:~$
```

Command: `ps aux --sort=-%mem | head -n 2`

- **`ps aux`**: Displays every process running on the system.
- **`--sort=-%mem`**: The minus sign – indicates a descending sort (highest to lowest). `%mem` targets Memory usage.
- **`head -n 2`**: Shows only the top 2 lines (the header and the top process).
- **`kill [PID]`**: Once you identify the PID (Process ID) from the list, `kill` sends a signal to the OS to shut that process down.

Verification:

- `The top command has given the highest memory usage and I have killed that process. For me the process was FireFox I killed it using kill <PID of FireFox>`

5) Use the networking tool command and print all the gateway available in a sorted manner

```
hanuman@hanuman-VMware-Virtual-Platform:~/Documents$ ip route
default via 192.168.23.2 dev ens33 proto dhcp src 192.168.23.129 metric 100
192.168.23.0/24 dev ens33 proto kernel scope link src 192.168.23.129 metric 100
hanuman@hanuman-VMware-Virtual-Platform:~/Documents$ netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         192.168.23.2    0.0.0.0         UG        0 0          0 ens33
192.168.23.0    0.0.0.0         255.255.255.0   U         0 0          0 ens33
hanuman@hanuman-VMware-Virtual-Platform:~/Documents$ ip route | grep default | awk '{print $3}' | sort
192.168.23.2
```

Command: `ip route show | grep default | awk '{print $3}'`

- **`ip route`**: The modern replacement for the old `route` or `netstat` commands. It shows how data packets travel out of your computer.

- **`grep default`**: The "default" route is typically your gateway (your router's IP).
- **`awk '{print $3}'`**: `awk` is a text-processing tool that views a line as a series of columns. In the output of `ip route`, the gateway IP is usually the **3rd** word on the line.
- **`sort -u`**: Ensures that if there are multiple identical paths, you only see a clean, unique list.

Verification:

- `ip route displays all the routing network and I have displayed the sorted output.`