Lab:

Memory Usage and Management

Image Generated by Dalle-E 2 from chat GPT

Prepared for

Rajani Phadtare

Prepared by Group 5

Valentine Jingwa, Jean-Pierre Nde-Forgwang,

Operating Systems, Cohort D

School of Advance Digital Technology

SAIT

17 March 2024

Lab: Memory Usage and Management

Part A: Virtual Memory Management and Performance

Let's break down the calculations for each part mathematically, including the page fault calculations for the FIFO and LRU algorithms and the effective access time calculation.

a. FIFO with 3 frames:

Given the sequence: 1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6

- Start with empty frames.
- Follow the sequence, loading each page into a frame if it's not already there.
- If a page needs to be loaded but all frames are full, replace the oldest page (FIFO principle).
- I for time saving sake b, c, and d will be done on paper and the answers transferred over here.

Calculation:

1. Load 1: fault
2. Load 2: fault
3. Load 3: fault
4. Load 4 (replace 1): fault
5. Access 2 (already in memory): no fault
6. Load 1 (replace 3): fault
7. Load 5 (replace 4): fault
8. Load 6 (replace 2): fault
9. Access 2 (replace 1): fault
10. Load 1 (replace 5): fault
11. Access 2 (already in memory): no fault
12. Load 3 (replace 6): fault
13. Load 7 (replace 2): fault
14. Load 6 (replace 1): fault
15. Access 3 (replace 7): fault
16. Access 2 (replace 6): fault
17. Load 1 (replace 3): fault
18. Access 2 (already in memory): no fault
19. Load 3 (replace 1): fault
20. Load 6 (replace 2): fault

Total Page Faults = 16

b. LRU with 3 frames:

Calculation:
The key difference in LRU is that we replace the least recently used page instead of the oldest page.

Total Page Faults (Calculated) = 15

c. LRU with 5 frames:

With more frames, fewer replacements are needed.

Total Page Faults (Calculated) = 8

d. Minimum Page Faults:

Calculation:

The minimum number of page faults occurs when each page is loaded once without any need for replacement.

Given the sequence has 7 unique pages, and assuming unlimited frames, the minimum page faults = 7 (once for each unique page).

2. Effective Access Time Calculation

Given:
- TLB access time = 20ns
- Cache access time = 30ns
- Main memory access time = 100ns
- TLB hit rate = 60%
- Cache hit rate (not in TLB) = 10%
- Memory found in cache rate = 80%

Calculation
Formula for this case

Effective Access Time= (TLB hit rate X TLB access time) + (Cache hit rate, not in TLB X (TLB access time + Cache access time)) + ((1 - TLB hit rate - Cache hit rate) X Memory found in cache rate X (TLB access time + Main memory access time)) + ((1 - TLB hit rate - Cache hit rate) X (1 - Memory found in cache rate) X (TLB access time + Main memory access time))

= (0.6 X 20) + (0.1 X (20 + 30)) + ((1 - 0.6 - 0.1) X 0.8 X (20 + 100)) + ((1 - 0.6 - 0.1) X (1 - 0.8) X (20 + 100))

= 12 + 5 + (0.3 X 0.8 X 120) + (0.3 X 0.2 X 120)

= 12 + 5 + 28.8 + 7.2

= 53 ns

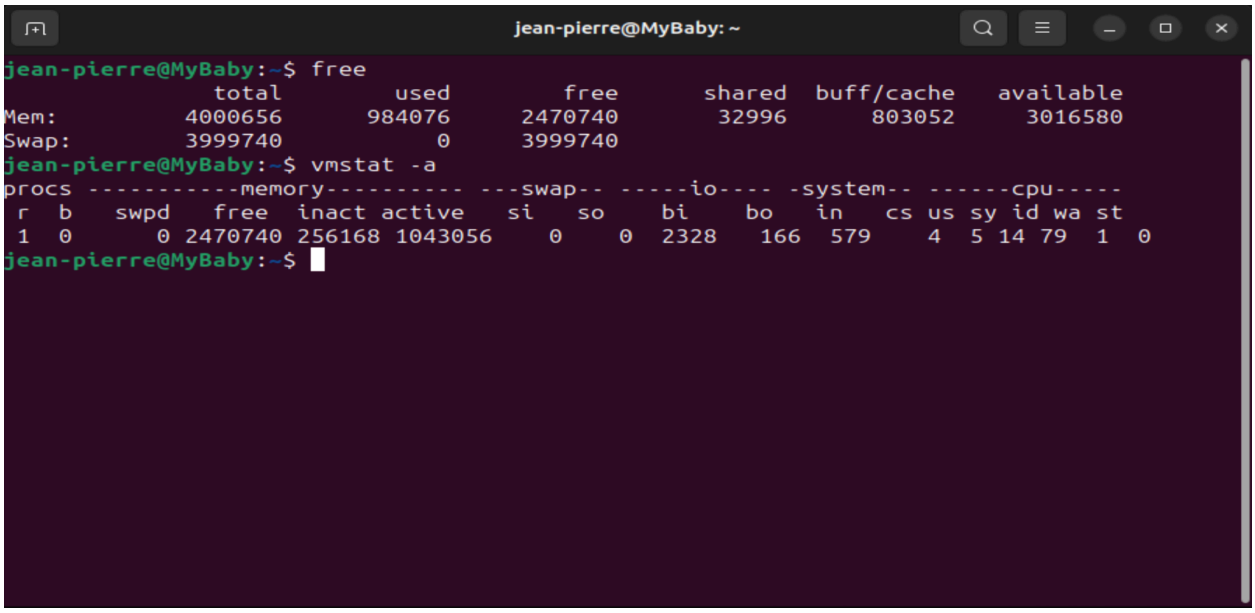
3. Purpose For TLB

The TLB (Translation Lookaside Buffer) is used to reduce the time taken to translate virtual addresses to physical addresses by storing recent translations. It's a cache for the page table, making memory access faster.

Part B: Linux Memory Management

1. Complete the table using the Linux tools `free` and `vmstat -a` to find the memory usage values. Remember to include units. (3 marks)

Physical Memory	Virtual Memory
Total: 4000656kb	Free: 3999740 kB
Used: 984076kb	Inactive: 256168 kB
Available: 3016580kb	Active: 1043056 kB
Buffers: 803052kb	



2. Open a terminal window, start `top` and study the output. Record the value of RES for the `top` process (including units). What does this value represent? (1 mark)

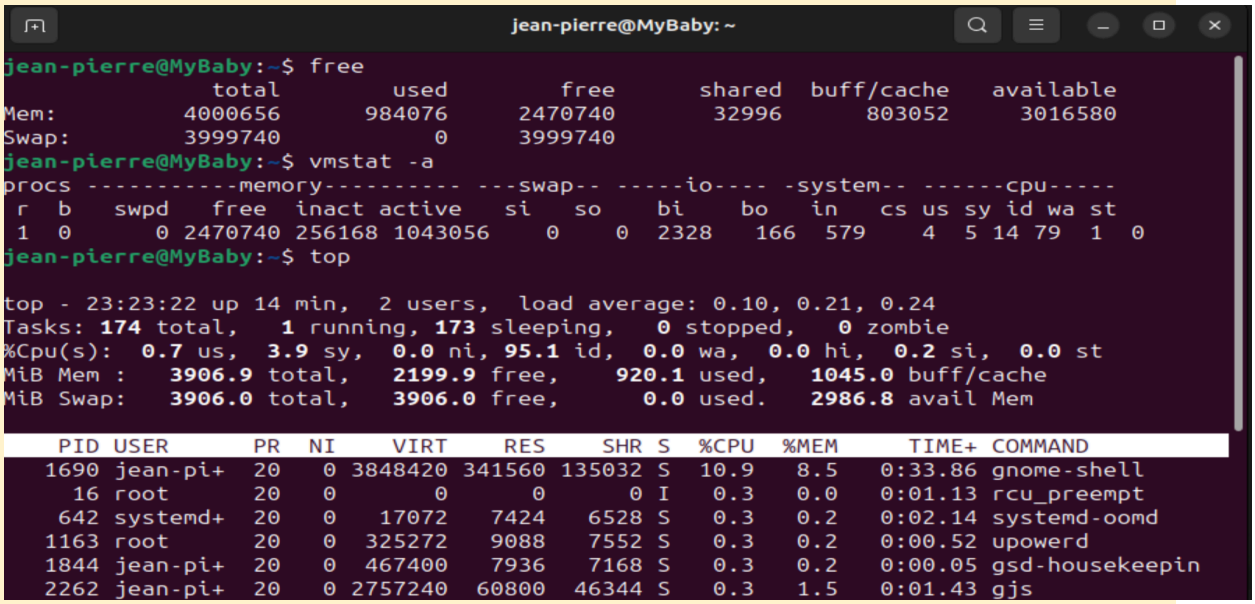
Answer: 341560kb

3. Conducting any necessary research, briefly describe how to specify sorting on virtual memory used in `top`. (1 mark)

Answer: You can press ‘F’, then choose the field representing VIRT (virtual memory), and press s to sort by this field

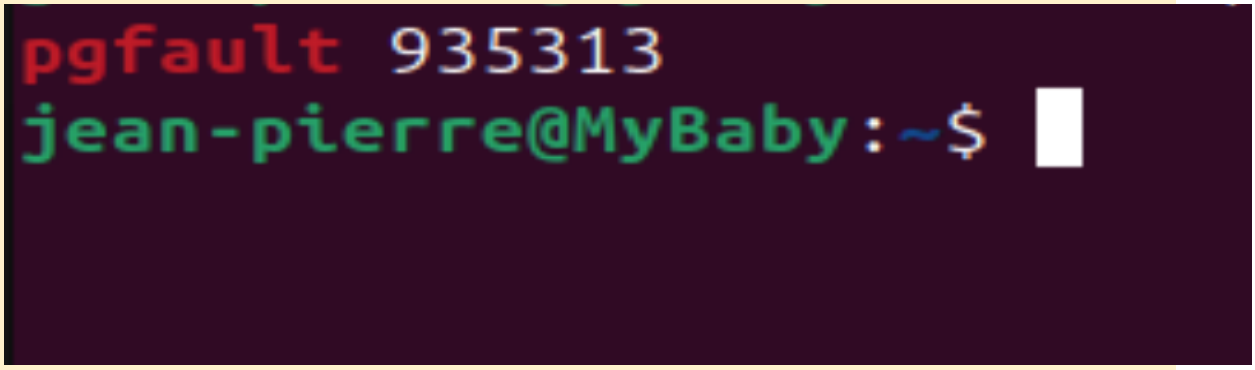
4. Analyzing the output of `top`, what program needs the most virtual memory? (1 mark)

Answer: gnome-shell



5. Explore the `/proc/vmstat` pseudo-file, and then record the `pagefaults`. (1 mark)

Answer: 935313



6. What's the difference between `pgfault` and `pgmajfault`? Why is it important to know these values? (2 marks)

Answer: pgfault includes all minor page faults and major page faults, while pgmajfault only counts major ones.

7. Briefly describe the following parameters within `/proc/meminfo`. Conduct additional research as necessary.

   a. Dirty (1 mark)

   Answer: Memory waiting to be written back to the disk.

   b. Mapped (1 mark)

   Answer: Memory that has been mapped into the address space of a process.

   c. PageTables (1 mark)

   Answer: Memory used to store the page tables for virtual memory.

   d. Hugepagesize (1 mark)

   Answer: Size of huge pages.

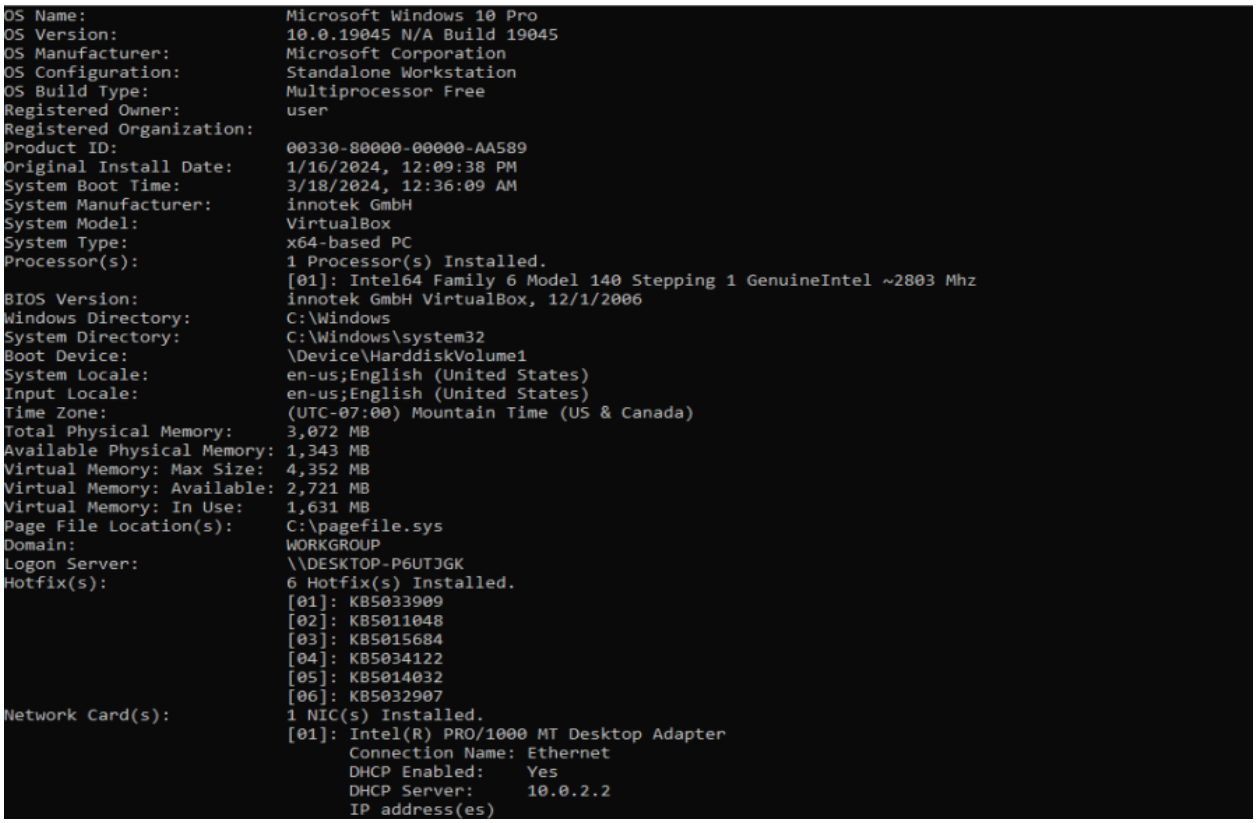
8. Run `pmap -x pid` (replace `pid` with the respective `pid` of the gnome-terminal process). What does this command show you? (1 mark)

   Answer: Shows memory map of a process, including the memory used by the gnome-terminal process.

9. Explain the output as shown with the `-x` option. Why is this useful? (1 mark)

   Answer: It provides detailed information about the memory usage of each part of the process, including shared libraries, stack, and heap usage

Part C: Windows Memory Management



```
OS Name:                   Microsoft Windows 10 Pro
OS Version:                10.0.19045 N/A Build 19045
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Standalone Workstation
OS Build Type:             Multiprocessor Free
Registered Owner:          user
Registered Organization:
Product ID:                00330-80000-00000-AA589
Original Install Date:     1/16/2024, 12:09:38 PM
System Boot Time:          3/18/2024, 12:36:09 AM
System Manufacturer:       innotek GmbH
System Model:              VirtualBox
System Type:               x64-based PC
Processor(s):              1 Processor(s) Installed.
                           [01]: Intel64 Family 6 Model 140 Stepping 1 GenuineIntel ~2803 Mhz
BIOS Version:              innotek GmbH VirtualBox, 12/1/2006
Windows Directory:         C:\Windows
System Directory:          C:\Windows\system32
Boot Device:               \Device\HarddiskVolume1
System Locale:             en-us;English (United States)
Input Locale:              en-us;English (United States)
Time Zone:                 (UTC-07:00) Mountain Time (US & Canada)
Total Physical Memory:     3,072 MB
Available Physical Memory: 1,343 MB
Virtual Memory: Max Size:  4,352 MB
Virtual Memory: Available: 2,721 MB
Virtual Memory: In Use:    1,631 MB
Page File Location(s):     C:\pagefile.sys
Domain:                    WORKGROUP
Logon Server:              \\DESKTOP-P6UTJGK
Hotfix(s):                 6 Hotfix(s) Installed.
                           [01]: KB5033909
                           [02]: KB5011048
                           [03]: KB5015684
                           [04]: KB5034122
                           [05]: KB5014032
                           [06]: KB5032907
Network Card(s):           1 NIC(s) Installed.
                           [01]: Intel(R) PRO/1000 MT Desktop Adapter
                                 Connection Name: Ethernet
                                 DHCP Enabled:    Yes
                                 DHCP Server:     10.0.2.2
                                 IP address(es)
```

1. Complete the table using Windows System Information tool. Remember to include units. (2 marks)

Physical Memory	Virtual Memory
Total: ____3072__	Total: ___4352_____
Available: ____1343_____	Available: ____2721_____

2. Use **procexp > Select Columns** to add all columns related to memory settings. Briefly describe the following columns:

a. Working Set Size:

This represents the set of pages in the virtual memory that are currently resident in physical memory and can be accessed without causing a page fault. It includes both shared and private data. The size of the working set reflects the memory demand of the process.

b. WS Private Bytes:

This measures the amount of memory that a process is using that cannot be shared with other processes. These are pages in memory that are only accessible by the process. It's

an indicator of the actual memory a process is using in a private manner, excluding any memory shared with other processes.

c. WS Shareable Bytes:

This represents the portion of the working set that could potentially be shared with other processes. Shareable bytes include memory that contains code (such as DLLs) or data that can be accessed by more than one process. However, whether this memory is shared in practice depends on whether other processes actually access it.

3. Difference Between Private and Shareable Bytes:

The main difference between private bytes and shareable bytes lies in their accessibility and exclusivity to a process. Private bytes are exclusively used by a single process, and no other process can access this memory. On the other hand, shareable bytes are part of the working set that could be accessed by other processes, implying that this memory can potentially be shared among multiple processes.

4. Largest Virtual Memory Size:

To determine which process has the largest virtual memory size, you would use Process Explorer to sort the processes by the "Virtual Size" column. This information requires real-time data from your system's Process Explorer.

5. Highest Number of Page Faults:

Similar to finding the largest virtual memory size, identifying the process with the highest number of page faults involves sorting the processes in Process Explorer by the "Page Faults" column. This will show you which process has generated the most page faults over its execution period.

6. Go to **procexp > System Information** and briefly describe the following values that appear on the **Memory** tab. Conduct additional research as necessary.

    a. Zeroed (1 mark)

    Answer: This is memory that has been cleared, or filled with zeros, by the system and is ready to be allocated to a process without any additional preparation

b. Free (1 mark)

Answer: Memory that is not currently being used or reserved, and can be allocated to processes or the operating system as needed

c. Modified (1 mark)

Answer: This represents memory that has been modified (or written to) and needs to be written to disk before it can be used for another purpose

d. ModifiedNoWrite (1 mark)

Answer: A category of memory that contains modified data that doesn't need to be written to disk.

e. Standby (1 mark)

Answer: Memory that contains cached data and code that is not actively being used but can be quickly reactivated

7. What's the difference between Modified and ModifiedNoWrite? (1 mark)

Answer: Modified memory contains data that must be written to disk before it can be used for something else, while ModifiedNoWrite memory contains data that has been changed but doesn't need to be saved to disk, and so it can be repurposed more quickly.

8. In the same tab, there is a section for kernel memory. What's the difference between Paged and Nonpaged? (1 mark)

Answer: Paged kernel memory is the portion of the kernel memory that can be paged out to the system's page file on disk when not in use, freeing up physical memory for other

tasks. Nonpaged kernel memory must stay in physical memory at all times and cannot be paged out, as it contains code and data necessary for handling low-level system operations that must be ready to execute at any time without the delay of paging in.

9. Briefly describe the purpose of Windows pagefile.sys. Conduct additional research as necessary. (1 mark)

Answer: Windows system page file used to store pages of memory that do not currently fit in physical memory.

10. How big is the pagefile? How might this affect the performance of the system? (2 marks)

Answer: The size of pagefile.sys can vary and is usually managed by Windows automatically based on the amount of physical memory. However, users can manually configure its size. The size of the pagefile can affect system performance; too small of a pagefile might cause the system to run out of virtual memory, leading to crashes or performance issues, while a pagefile that is too large can take up unnecessary disk space.

Part D: Windows Memory, Registry Settings and Prefetch

1. Briefly describe the effect of setting the registry value **HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management\ClearPageFileAtShutdown** to **1**. What would this be used for? Conduct additional research as necessary. (1 mark)

When set to 1, ensures that the pagefile is cleared upon system shutdown, which can be used for security purposes to prevent sensitive data from being recovered.

2. Examine the prefetcher and then answer the following questions. Conduct additional research as necessary.

    a. Briefly describe the prefetcher's function. (1 mark)

    Answer: It's designed to speed up the application launch process and reduce boot time by pre-loading parts of frequently used applications and files into memory.

b. How can the prefetch utility improve system performance? (1 mark)

Answer: The prefetch utility in Windows improves system performance by preloading parts of frequently used applications into memory before they are needed

c. Access the folder **c:\Windows\Prefetch** and identify the file type in this directory. (1 mark)

Answer: These are prefetch files that are created by the system as it runs applications; they contain information about which files are accessed and are used to speed up the application launch process on subsequent uses.

d. Identify two files that relate to tasks you have often performed on this virtual machine. (1 mark)

Answer: Notepad++, PowerShell

3. Briefly describe the effect of setting the registry value **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management\PrefetchParameters\EnablePrefetcher** to the values 0, 1, 2 and 3. Conduct additional research as necessary. (2 marks)

Answer:
0: Disables prefetching.
1: Enables prefetching for application launch only.
2: Enables boot prefetching only.
3: Enables prefetching for both application launch and boot.

The SysInternals suite that you downloaded from a previous lab exercise has many tools to aid Windows system debugging. Process Explorer lets you observe processes in the system but it's a moving snapshot, whereas Process Monitor captures detailed information.

Procmon is an advanced logging tool that can tell you what low-level operations a process is performing and is essential to diagnostics. Procmon captures information about the registry,

files, processes and network activity, which can total millions of operations in just a few seconds. When you're using procmon, it's most important to know how to filter the capture to find the suspect component for analysis.

1. Start Process Monitor (procmon.exe) from the **Sysinternals** folder.

   The main window immediately begins displaying event capture data. Each row in the table represents one low-level event that occurred in the system.

2. Stop/disable event capture by selecting **File** on the menu and unchecking **Capture Events** (or click the magnifying glass icon on the toolbar).
3. Clear the event list using the **Edit** menu or the button on the toolbar (hover over the buttons to view the function and the shortcut key).
4. Let's capture the event of opening the Command Prompt. First re-enable event capture.
5. Start Command Prompt.
6. Disable event capture. You will see that many events were captured, not only the one related to the Command Prompt.
7. Now, consider a scenario where you're interested in the console window. You need to add a filter to procmon, which you can do in two ways:

   - Find (CTRL+F) **cmd.exe** under *Process Name*, and then right-click it and select **Include 'cmd.exe'**. OR
   - Select **Filter > Filter** from the menu. Choose **Process Name** from the leftmost dropdown box and type **cmd.exe** in the third dropdown box. Click **Add** and then click **OK**.

     Note: Procmon filters out events that are usually irrelevant by default (any event initiated in kernel mode and events associated with the Sysinternals tools themselves).

8. From the *Path* column, find a prefetch event in the list. Right-click it and include the path in the filter.

   Only events related to prefetching the CMD process should appear.

Promon contains important information regarding each event. They can be viewed via the Properties dialog box via the right-click context menu or simply double-clicking on a particular event. There are 3 tabs in this window:

- Event: information about the event, many of these are already in the Detail column but here in more readable format.
- Process: information about the process that generated the event, including a list of modules loaded into the process's address space at the time of the event.

- Stack: great for debugging! This shows you the call sequence leading up to this event in user and kernel mode.

9. Explain the four (or five if you're lucky!) events from the previous step. (4 marks)

Answer:
- File system events: These would typically include the opening, reading, or writing of files. In the context of starting the Command Prompt, you might see events related to reading from the cmd.exe executable file or related DLLs.

- Registry events: When cmd.exe starts, it might read from or write to certain registry keys. These could be configuration settings or other data relevant to how the Command Prompt operates.

- Network events: It's less common for cmd.exe to generate network events upon startup unless it's configured to run a network command immediately upon opening.

- Process/thread events: You might see the Command Prompt process being created, along with its primary thread starting. Additionally, any threads it creates as part of its startup process would also be logged.

Resources

Silberschatz, A., Galvin, P.B., & Gagne, G. (2018). *Operating system concepts* (10th ed.). John Wiley & Sons, Inc.

Erl, T., Mahmood, Z., & Puttini, R. (2013). *Cloud computing: Concepts, technology & architecture*. Prentice Hall.

Arpaci-Dusseau, A. C., & Arpaci-Dusseau, R. H. (2018). *Operating systems: Three easy pieces.* CreateSpace Independent Publishing Platform. https://pages.cs.wisc.edu/~remzi/OSTEP/