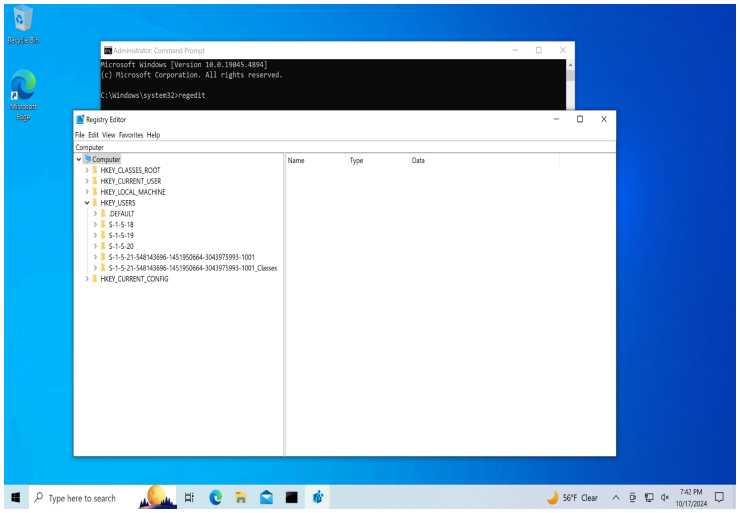
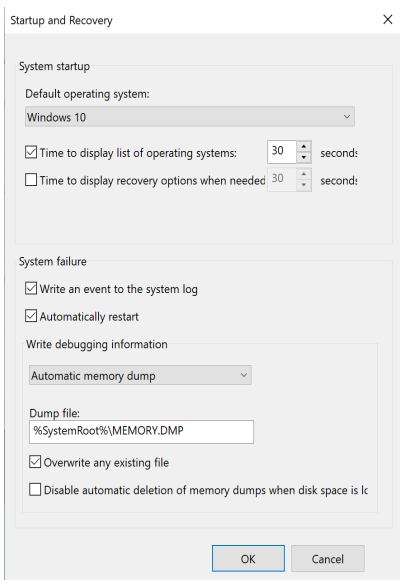
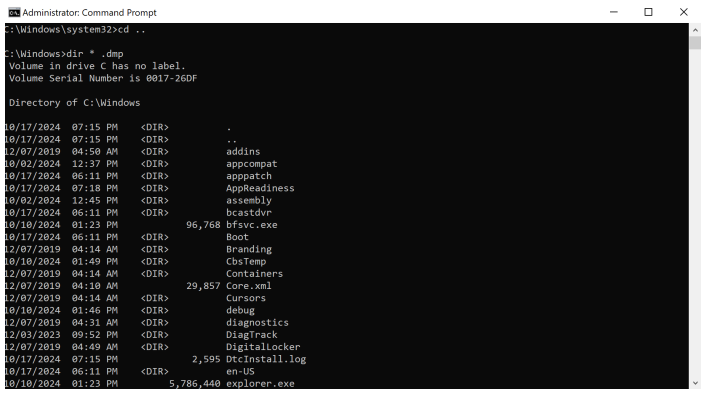


I opened the Windows Registry Editor and focused on the HKEY_USERS section. I used the regedit command from the command prompt to get there.

I looked at the HKEY_USERS hive to find user-specific data like preferences, application settings, and other stuff tied to each user's Security Identifier (SID). Forensic investigators usually check this part of the registry to see traces of user activity, the apps they used, and how they were connected to networks. This helps us understand how the system was used by different accounts.

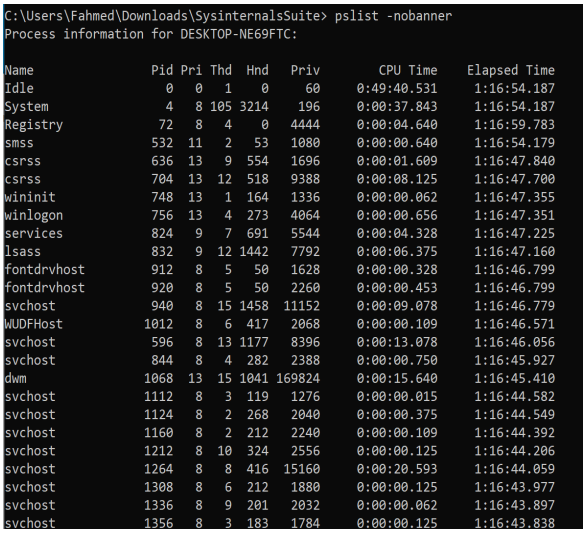
I accessed the Startup and Recovery settings on a Windows system, where I could configure how the system handles failures and crash dumps. Here, I set the system to create an automatic memory dump if it crashes and save it to the file %SystemRoot%\MEMORY.DMP. I also made sure that the system writes an event to the system log and automatically restarts after a crash. The checkbox for Overwrite any existing file is also selected so that the dump file doesn't accumulate and take up unnecessary space. This configuration helps us ensure that if the system ever crashes, we have a memory dump ready for analysis, which could be crucial in figuring out what went wrong. This setup is to prepare the system to record useful crash data, making it easier for us to analyze system failures and perform forensic analysis on what might have caused the crash.





I used the command prompt to search for .dmp files in the Windows directory by running the dir *.dmp command. The command is looking for any memory dump files stored in the C:\Window folder.

This is to check if the system has created any memory dump files that we can analyze later. These dump files store important information about the system's memory and the state it was in at the time of a crash, which helps us investigate the cause of the failure. Being able to locate and analyze these dumps is crucial in the forensic process since they hold valuable data that can be used to trace the source of an issue or even identify potential security threats.

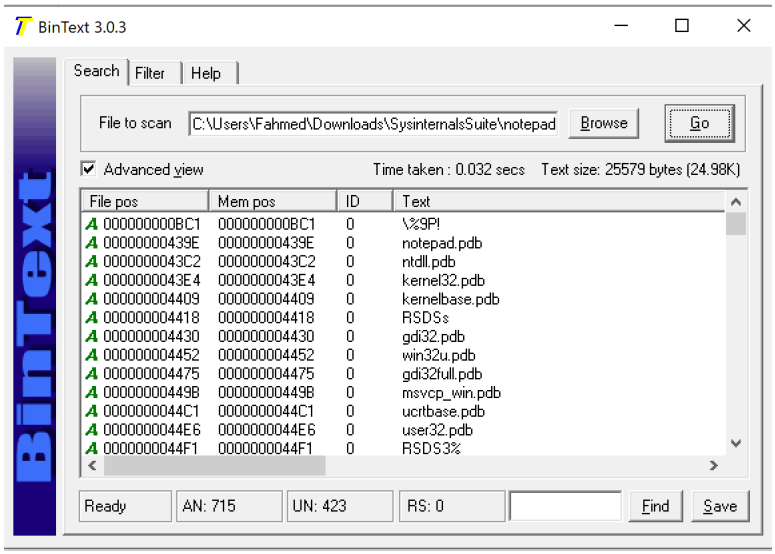


```
C:\Users\Fahmed\Downloads\SysinternalsSuite> pslist -nobanner
Process information for DESKTOP-NE69FTC:

Name                Pid Pri Thd  Hnd   Priv        CPU Time    Elapsed Time
Idle                  0   0   1    0     60     0:49:40.531     1:16:54.187
System                4   8 105 3214    196     0:00:37.843     1:16:54.187
Registry             72   8   4    0   4444     0:00:04.640     1:16:59.783
smss                532  11   2   53   1080     0:00:00.640     1:16:54.179
csrss               636  13   9  554   1696     0:00:01.609     1:16:47.840
csrss               704  13  12  518   9388     0:00:08.125     1:16:47.700
wininit             748  13   1  164   1336     0:00:00.062     1:16:47.355
winlogon            756  13   4  273   4064     0:00:00.656     1:16:47.351
services            824   9   7  691   5544     0:00:04.328     1:16:47.225
lsass               832   9  12 1442   7792     0:00:06.375     1:16:47.160
fontdrvhost         912   8   5   50   1628     0:00:00.328     1:16:46.799
fontdrvhost         920   8   5   50   2260     0:00:00.453     1:16:46.799
svchost             940   8  15 1458  11152     0:00:09.078     1:16:46.779
WUDFHost           1012   8   6  417   2068     0:00:00.109     1:16:46.571
svchost             596   8  13 1177   8396     0:00:13.078     1:16:46.056
svchost             844   8   4  282   2388     0:00:00.750     1:16:45.927
dwm                1068  13  15 1041 169824     0:00:15.640     1:16:45.410
svchost            1112   8   3  119   1276     0:00:00.015     1:16:44.582
svchost            1124   8   2  268   2040     0:00:00.375     1:16:44.549
svchost            1160   8   2  212   2240     0:00:00.109     1:16:44.392
svchost            1212   8  10  324   2556     0:00:00.125     1:16:44.206
svchost            1264   8   8  416  15160     0:00:20.593     1:16:44.059
svchost            1308   8   6  212   1880     0:00:00.125     1:16:43.977
svchost            1336   8   9  201   2032     0:00:00.062     1:16:43.897
svchost            1356   8   3  183   1784     0:00:00.125     1:16:43.838
```

I ran the pslist -nobanner command to display a list of running processes on my system. This is aligned with Step 4 from the lab guide. The output shows details such as the Process ID (PID), priority (Pri), number of threads (Thd), number of handles (Hnd), private memory usage (Priv), CPU time, and the total elapsed time that each process has been running.

This step provides an overview of the processes currently active on the system, along with important statistics about each process's resource usage. By analyzing these details, we can identify which processes are using a significant amount of system resources, which may indicate a performance issue or the presence of suspicious activity. This command is important for forensic investigators to monitor system processes and look for anything out of the ordinary, like unexpected background processes or elevated resource usage that might indicate a compromise.

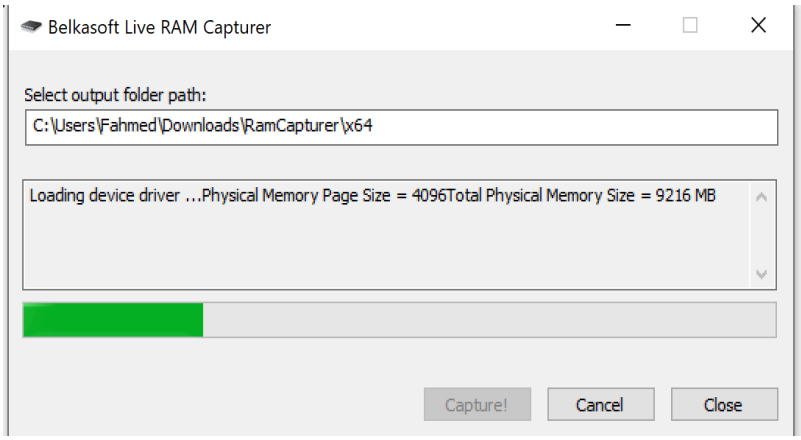


In this screenshot, I used the BinText tool to analyze the contents of a memory dump of the notepad.exe process. This action aligns with Step 6 of the lab guide, where we were instructed to use the BinText tool to open and view the contents of a process memory dump. BinText allows us to extract readable text strings from binary files, and here, it shows some .pdb

files associated with Notepad, like ntdll.pdb, kernel32.pdb, and msvcp_win.pdb. These files are debugging symbols that can provide useful information about the process's execution state. The memory positions (Mem pos) indicate where in memory the strings were located, while the Text column shows the extracted text. This step examines the content of the process memory dump for readable strings that may reveal valuable information, such as internal process states, error messages, or even potential sensitive data that may have been loaded into memory. This is especially useful in forensics when investigating suspicious or compromised processes.

```
C:\Users\Fahmed\Downloads\SysinternalsSuite>procdump -nobanner -mm 8060
[20:31:14] Dump 1 initiated: C:\Users\Fahmed\Downloads\SysinternalsSuite\notepad.exe_241017_203114.dmp
[20:31:14] Dump 1 complete: 1 MB written in 0.2 seconds
[20:31:14] Dump count reached.
```

The procdump command creates a memory dump of the Notepad process with a Process ID (PID) of 8060. This action aligns with Step 5 in the lab, where we were instructed to use procdump from Sysinternals tools to capture a memory dump of a specific process.
The command procdump -nobanner -mm 8060 creates a mini memory dump of the Notepad process. The output shows that the dump was successfully created and saved as notepad.exe_241017_203114.dmp. The dump file is 1 MB in size and was created in 0.2 seconds. The message Dump count reached confirms that the dump process was completed. This step captures the state of the process's memory at a particular moment in time. This dump file can later be analyzed using tools like BinText or other memory analyzers to extract useful forensic data. By doing this, we can investigate the memory of the process to look for any anomalies, security risks, or other relevant information that may provide insights into the process’s behavior or interactions.



In this screenshot, I’m using the Belkasoft Live RAM Capturer to capture a live memory dump of the system. This aligns with Step 9 in the lab, where we are instructed to perform RAM acquisition using a tool like Belkasoft Live RAM Capturer. The output folder is set to C:\Users\Fahmed\Downloads\RamCapturer\x64, and the tool is in the process of loading the device driver to begin the capture. The information at the bottom shows the physical memory

page size as 4096 and the total physical memory size as 9216 MB. The purpose of this step is to capture the entire contents of the system's RAM in a live state. RAM contains volatile data that will be lost when the system is shut down or rebooted, so capturing it live is critical for forensic investigations. By acquiring a memory dump, we can analyze the RAM later for active processes, network connections, encryption keys, and other transient data that could be crucial for our investigation.

Executive Summary:

In this lab, we worked on capturing and looking into memory and process info from a Windows system using different forensic tools to get important details. First, we set up the system to create crash dumps if it ever fails, making sure we have memory dumps ready to analyze after a crash. We then used pslist to list and check the processes running on the system, which helped us see how resources were being used and if there was anything unusual. After that, I used procdump to create a memory dump of the Notepad process. We opened that dump in BinText to pull out any readable data, which gave us a look at what was in memory, like program files and debugging info. This was useful to help track what the process was doing and check for anything strange. We also used Belkasoft Live RAM Capturer to grab a full dump of the system's RAM while it was running. This was important because RAM holds data that disappears when the computer is turned off, so capturing it live lets us save a snapshot of everything happening at that moment. This can include things like network connections or passwords that might be lost later.