

Lab 2. I/O Monitoring (Windows)

01 - Objectives

- How to determine the root sources of having intensive disk usage/RAM/CPU and intensive networking using **Task Manager** (Processes, Performance, App history, Start-up, Users, Details, Services).
- Analyzing performance issues due to intensive disk use using **Windows Performance Recorder, Process Monitor and Process Explorer**.
- Monitor the disk activity, identify who is generating it and how to figure out the issue by looking at the pdbs and the code.

02 - I/O Monitoring – Windows

Introduction

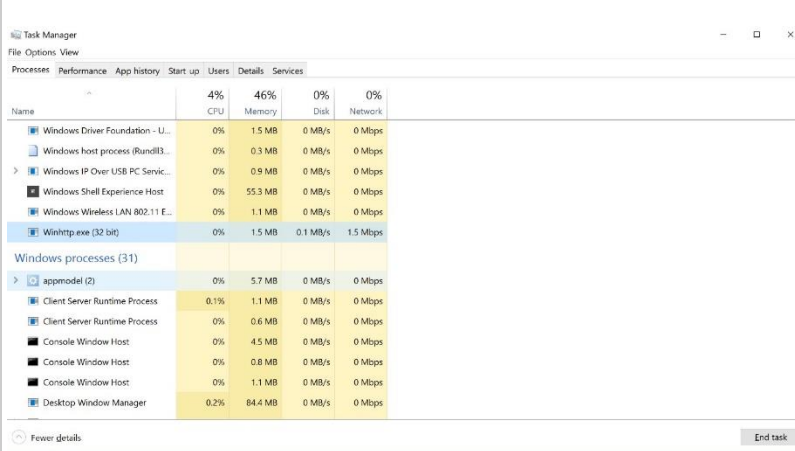
- Since computers started to surface, for many people it was a mystery what was happening behind the screen and it seemed magical when it wasn't working and even more magical when it was working. Since Linux is open-source, all sorts of tools appeared over time to analyse problems when they came up. On Windows on the other hand, the system being closed made it harder for tools to appear.
- The first tools were provided by Sysinternals. These were written by Mike Rusinovich, who chose to make public tools such as “File monitor” and “Registry monitor”, which were later combined into “Process monitor”. The tools were so good that even Microsoft's support teams were using them. Seeing their usefulness and appreciating the know-how of their operating system, Microsoft decided to buy Sysinternals, so now the original website redirects to <https://technet.microsoft.com/en-us/sysinternals> (outside Romania it probably redirects to a different link due to localization reasons that consider the language of the country where redirection is made).
- On this website can be found some of the tools that will be used in this tutorial - Process Monitor, Process Explorer, VMMap, Autoruns. Starting with Windows 7, Microsoft has begun to invest more and more in the performance of the system and in ways to monitor the system's performance. Some tools already existed since Windows 2000, but they were only used internally.

Task Manager

Shows the process name responsible for constant disk thrashing either by reads or writes. To start Task manager use the shortcut: **Ctrl + Shift + Esc**.

A. Task Manager - Processes tab

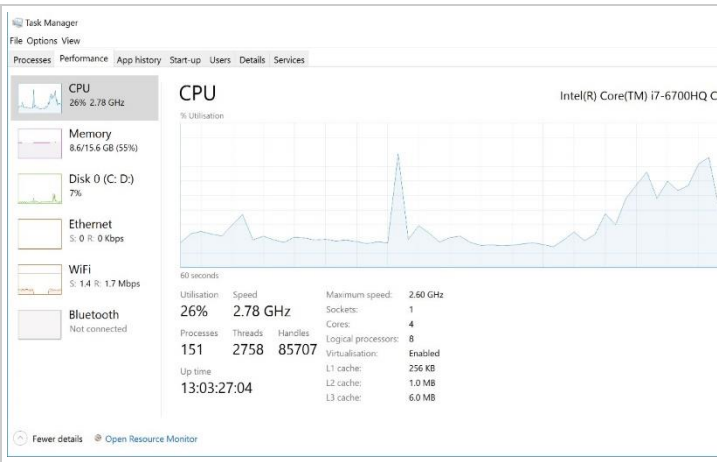
Processes tab shows all the running processes and their current resource usage in terms of CPU, Memory, Disk and Network.



Name	4% CPU	46% Memory	0% Disk	0% Network
Windows Driver Foundation - U...	0%	1.5 MB	0 MB/s	0 Mbps
Windows host process (RandE3...	0%	0.3 MB	0 MB/s	0 Mbps
Windows IP Over USB PC Servic...	0%	0.9 MB	0 MB/s	0 Mbps
Windows Shell Experience Host	0%	55.3 MB	0 MB/s	0 Mbps
Windows Wireless LAN 802.11 E...	0%	1.1 MB	0 MB/s	0 Mbps
Winhttp.exe (32 bit)	0%	1.5 MB	0.1 MB/s	1.5 Mbps
Windows processes (31)				
appmodel (2)	0%	5.7 MB	0 MB/s	0 Mbps
Client Server Runtime Process	0.1%	1.1 MB	0 MB/s	0 Mbps
Client Server Runtime Process	0%	0.6 MB	0 MB/s	0 Mbps
Console Window Host	0%	4.5 MB	0 MB/s	0 Mbps
Console Window Host	0%	0.8 MB	0 MB/s	0 Mbps
Console Window Host	0%	1.1 MB	0 MB/s	0 Mbps
Desktop Window Manager	0.2%	84.4 MB	0 MB/s	0 Mbps

B. Task Manager - Performance tab

Performance tab shows the usage level of the computer's main resources in the last 60 seconds.



C. Task Manager - App history tab

The **App history tab** was first added to Windows 8, and it shows the resource consumption of metro applications. Metro applications are touch-screen-friendly applications written especially for Microsoft's WinRT programming interfaces.

The screenshot shows the Windows Task Manager App history tab. It displays a table of resource usage since 10/27/2016 for the current user account. The table has columns for Name, CPU time, Network, Metered network, and Tile updates.

Name	CPU time	Network	Metered network	Tile updates
3D Builder	0:00:00	0 MB	0 MB	0 MB
Adobe Photoshop Express	0:00:00	0 MB	0 MB	0 MB
Alarms & Clock	0:00:02	0 MB	0 MB	0 MB
Calculator	0:00:43	0 MB	0 MB	0 MB
Camera	0:00:00	0 MB	0 MB	0 MB
Connect	0:00:00	0 MB	0 MB	0 MB
Contact Support	0:00:00	0 MB	0 MB	0 MB
Cortana	0:03:28	3.4 MB	0 MB	0 MB
Duolingo	0:00:00	0 MB	0 MB	0 MB
Eclipse Manager	0:00:00	0 MB	0 MB	0 MB
Feedback Hub	0:00:00	0 MB	0 MB	0 MB
Flipboard	0:00:00	0 MB	0 MB	0 MB
Fresh Paint	0:00:00	0 MB	0 MB	0 MB

D. Task Manager - Start-up tab

The **Start-up tab** shows all the applications that start at start-up, (or at least in Microsoft's vision - this will be further

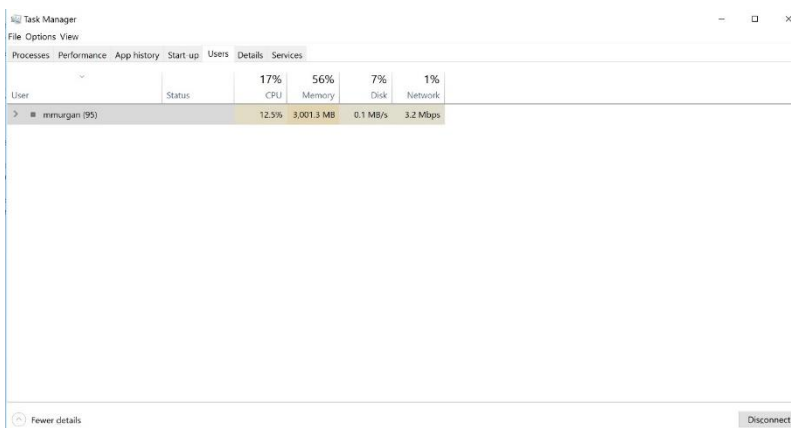
The screenshot shows the Windows Task Manager Start-up tab. It displays a table of applications that start at boot-up, including their Name, Publisher, Status, and Start-up impact.

Name	Publisher	Status	Start-up impact
Bluetooth Tray Application (2)	Broadcom Corporation	Enabled	Low
Cisco AnyConnect User Inte...	Cisco Systems, Inc.	Enabled	High
Dell System Detect	Dell	Enabled	Not measured
DPM Client (2)	Microsoft Corporation	Enabled	Medium
HD Audio Background Proc...	Realtek Semiconductor	Enabled	Medium
NVIDIA Update Backend	NVIDIA Corporation	Enabled	Low
Realtek HD Audio Manager	Realtek Semiconductor	Enabled	Low
Skype (2)	Skype Technologies S.A.	Enabled	High
Update.exe (9)		Enabled	High
Waves MaxxAudio Service A...	Waves Audio Ltd.	Enabled	Low

detailed in the Autoruns section), and their impact on the boot time. It is helpful to check this tab in case your computer takes a long to start up.

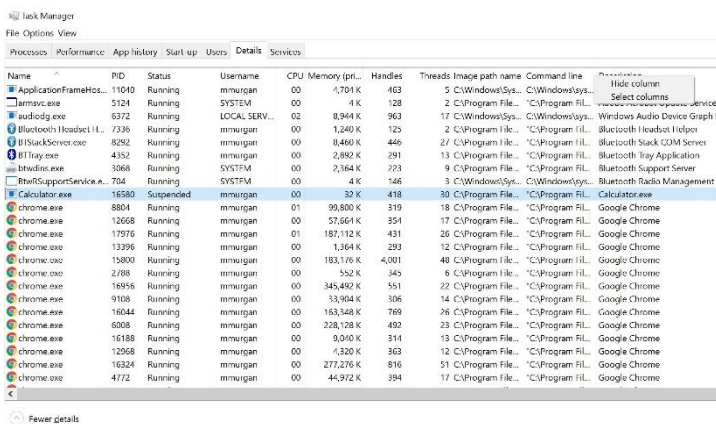
E. Task Manager - Users tab

Users tab shows the resource consumption of every logged in user. The screenshot below shows that there is only one user logged in.



F. Task Manager - Details tab

Details tab shows details for each process - pid, status, the user under which it runs. Right-clicking the column headers bar, offers the possibility

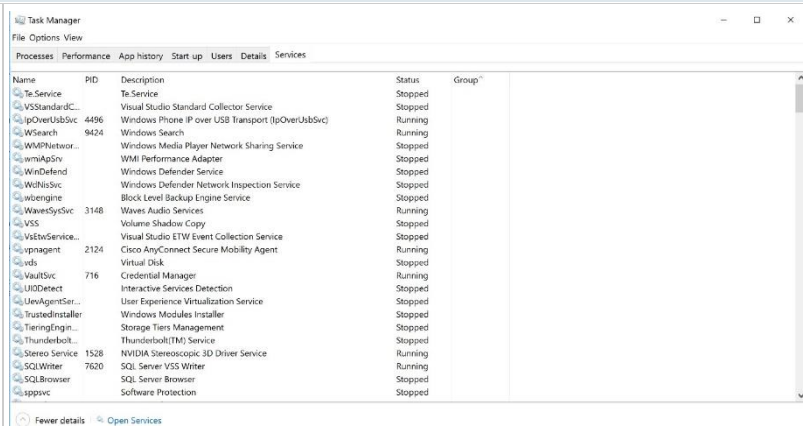


to add or remove columns. In the screenshot presented below the following columns were added: Handles, Threads, Image Path Name and Command Line. These new columns are very useful: the first one (Handles) when investigating a handle leak, the second one (Threads) in the case of investigating processes that create too many threads, the third one (Image Path Name) to find out the path from where the process was started, and the last one (Command

Line) to find out the parameters with which it was started.

G. Task Manager - Services tab

Services tab shows the service status. A Windows service can be considered similar to a Linux daemon: a process without a visual interface, offering services to user-created processes.



Conclusion:

- Task Manager can be used to identify which process uses a lot of RAM, CPU, accesses the disk many times or generates a lot of traffic on the network at a certain moment (Services tab). However, it does not offer information if in the long run, that same process is the one that generated the slowdown of the system. It does offer some information for longer periods of time, in the Start-up tab, which shows what process had higher impact at start-up, but does not specify the area that was impacted (disk space, RAM, CPU).
- You can sort by I/O read or I/O Writes, but no option to sort the results by Total I/O (combined Read & Write).

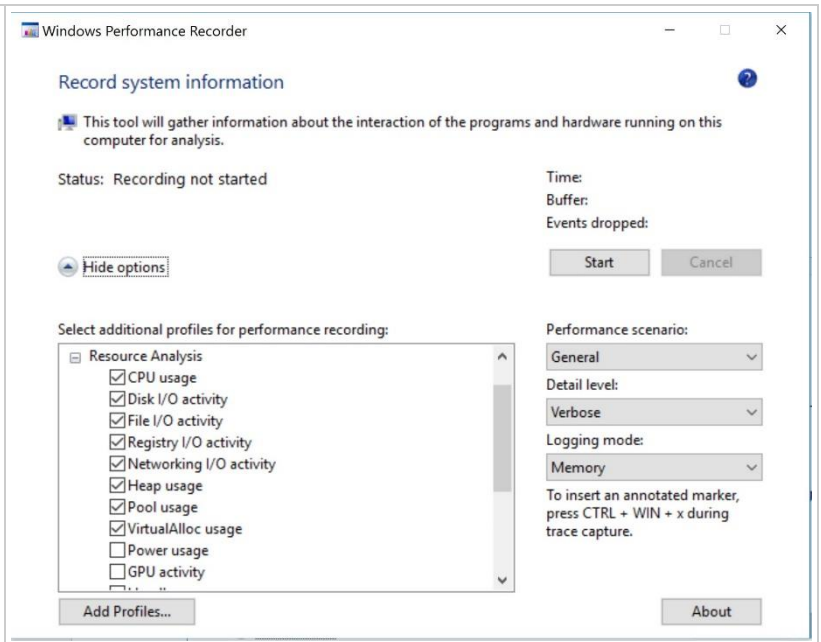
To overcome Task manager's limitation, and to perform a thorough analysis, use the excellent Resource Monitor (Resmon) utility, which is built-in to Windows.

Windows Performance Recorder

Installing Windows ADK will install Windows Performance Recorder. Check by clicking the windows button and typing "windows performance recorder".

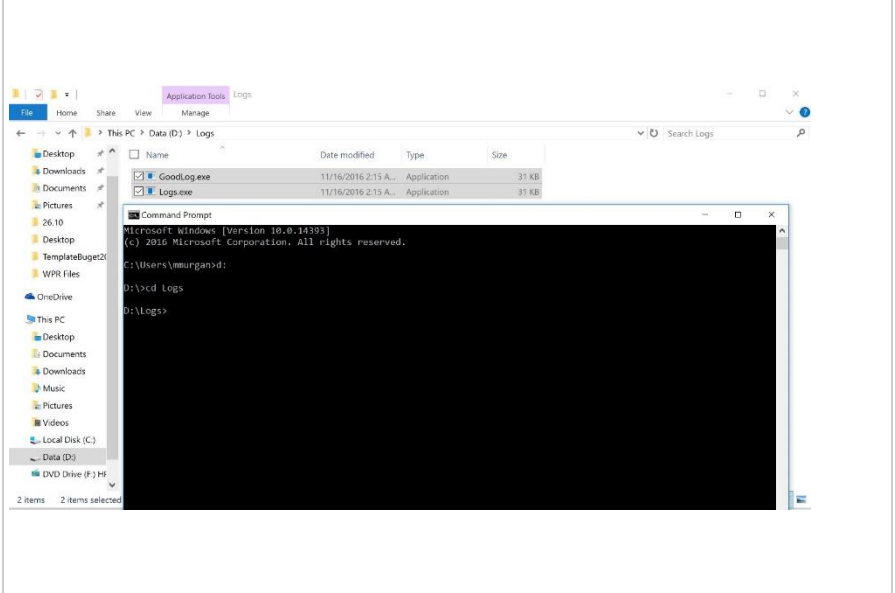


Click the **More options** button to get the list shown in the screenshot right below.



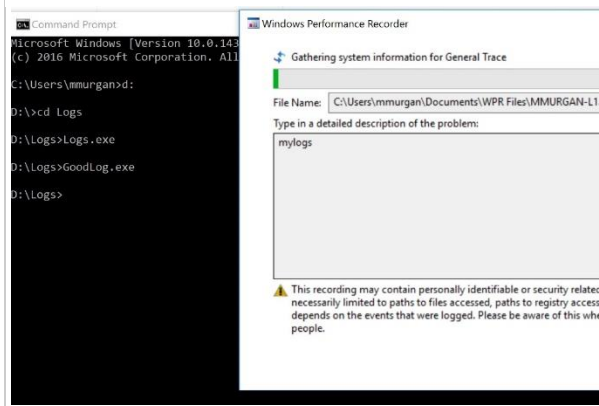
Datafile: Logs.exe and GoodLog.exe

Make sure that you select the same check boxes as in the screenshot, but do not click start just yet. Create a new directory and copy the **Logs.exe** and **GoodLog.exe** files into this directory. The behaviour of these two executables is similar to logging applications that write logs to the disk. Open a terminal and change the path to the directory where you copied the files.



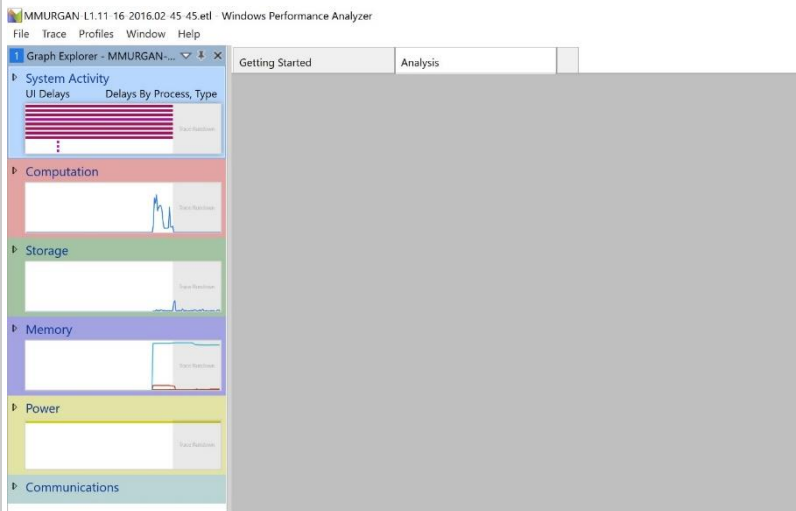
Windows Performance Recorder

Start **Windows Performance Recorder** and right after run **GoodLog.exe** and then **Logs.exe**. Once the two applications finish running, click the **Save** button in **Windows Performance Recorder**.

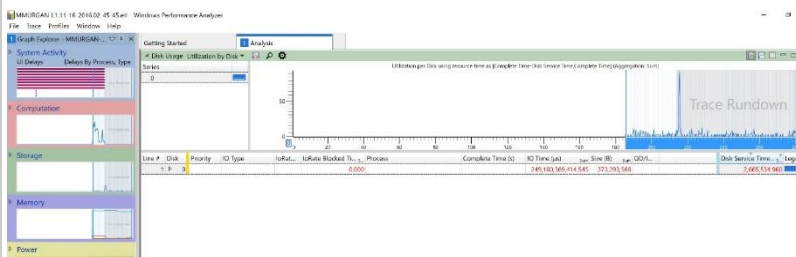


Windows Performance Analyzer

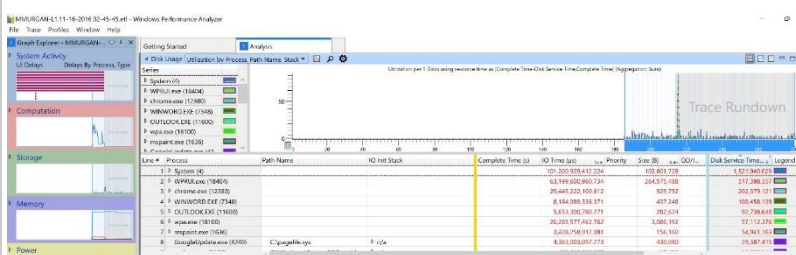
After the capture is saved, the Open option will become available in Windows Performance Analyzer. When clicking the Open button it should open a window such as the one below.



Double clicking on Storage should display the following window. Analyse the resources.

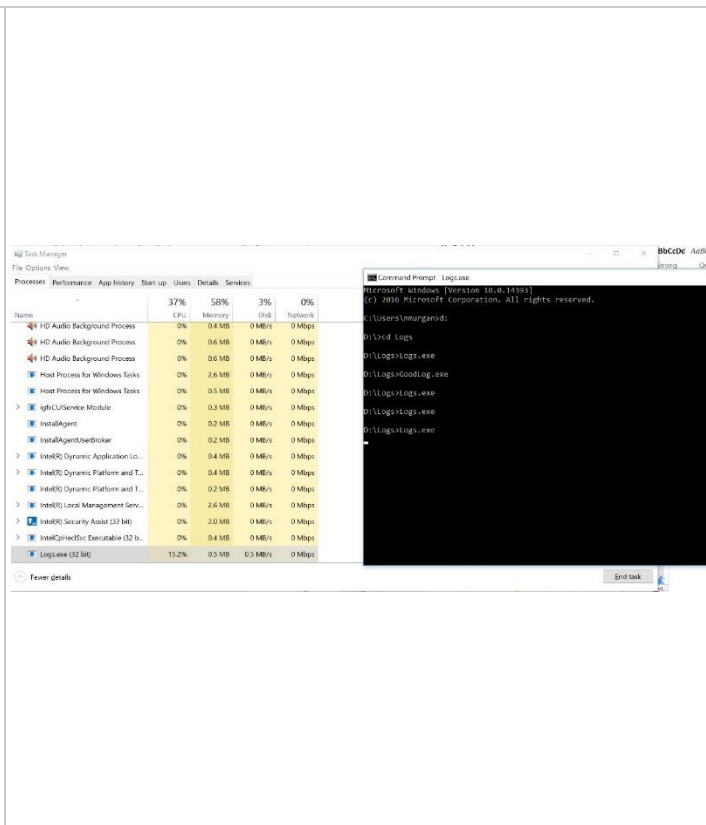


In the upper-left corner of the newly opened window it can select Disk Usage, Utilization by Disk. Click on Utilization by Disk and select: Utilization by Process, Path Name, and Stack. This will generate the following output.



Task Manager

The graph looks interesting. Processes can be selected for observing their activity on the disk. It can be noticed that our processes are not shown. Run Logs.exe again while keeping Task Manager on.



Conclusions:

- This shows that there is activity on the disk. The question is why doesn't Windows Performance Analyzer show it. The way Windows Performance Recorder records activity is based on events generated by the Windows kernel. It registers to track the events, listens to them, and during the recording period it constantly samples which process uses which resource at the time of sampling.
- It sums up the number of time that a process was caught doing something. In our case, the two processes want to write to the disk, but they are not the ones that get to do the actual writing. They tell the system that they want to write, and the System process schedules the writing. The reason for this is targeting a more efficient disk writing, as the System process is trying to minimise the impact to the disk. This is why our process's writing is passed over to the System process.

Process Monitor

Process Monitor is an excellent troubleshooting tool from Windows Sysinternals that displays the files and registry keys that applications access in real-time. The results can be saved to a log file, which you can send it to an expert for analyzing a problem and troubleshooting it.

How to Use Process Monitor to Track Registry and File System Changes?

Step 1: Running Process Monitor & Configuring Filters

1. Download Process Monitor from Windows Sysinternals site.
2. Extract the zip file contents to a folder of your choice.
3. Run the Process Monitor application.
4. Include the processes that you want to track the activity on. For this example, you want to include Notepad.exe in the (Include) Filters.
5. Click Add, and click OK.
6. From the Options menu, click Select Columns.
7. Under "Event Details", enable Sequence Number, and click OK.

You can add multiple entries as well, in case if you want to track few more processes along with Notepad.exe. To keep this example simpler, let's only track Notepad.exe. (You'll now see the Process Monitor main window tracking the list of registry and file accesses by processes real-time, as and when they occur.)

Step 2: Capturing Events

8. Open Notepad.

9. Switch to Process Monitor window.

10. Enable the "Capture" mode (if it's not already ON). You can see the status of the "Capture" mode via the Process Monitor toolbar.

11. The highlighted button above is the "Capture" button, which is currently disabled. You need to click that button (or use Ctrl + E key sequence) to enable capturing of events.

12. Cleanup the existing events list using Ctrl + X key sequence (Important) and start afresh.

13. Now switch to Notepad and try to reproduce the problem.

14. To reproduce the problem (for this example), try writing to HOSTS file (C:\Windows\System32\Drivers\Etc\HOSTS) and saving it. Windows offers to save the file (by showing the Save As dialog) with a different name, or in a different location. So, what happens under the hood when you save to HOSTS file? Process Monitor shows that exactly.

15. Switch to Process Monitor window, and turn off Capturing (Ctrl + E) as soon as you reproduce the problem. Important Note: Don't take much time to reproduce the problem after enabling capturing. Similarly turn off capturing as soon as you finish reproducing the problem. This is to prevent Process Monitor from recording other unneeded data (which makes analysis part more difficult). You need to do all that as quickly as you can.

The log file above tells us that Notepad encountered an ACCESS DENIED error when writing to the HOSTS file. The solution would be to simply run Notepad elevated (right-click and choose "Run as Administrator") to be able to write to HOSTS file successfully.

Step 3: Saving the Output

16. In the Process Monitor window, select the File menu and click Save.

17. Select Native Process Monitor Format (PML), mention the output file name and Path, save the file.

18. Right-click on the Logfile.PML file, click Send To, and choose Compressed (zipped) folder. This compresses the file by ~90%. Look at the graphic below. You certainly want to zip the log file before sending it to someone.

04 - Exercises

Exercise 01. [30p] Task Manager, Windows Performance Recorder and Process Monitor

- Go through the tutorials: Task Manager, Windows Performance Recorder and Process Monitor. Discuss the output and call the assistant to show him/her your progress.

Exercise 02. [10p] Task Manager

- Which program is constantly reading or writing to your hard disk?

How to:

- Open Task Manager, and select the Details tab.
- Right-click on the column header (Name, PID, Status etc) and click Select Columns.
- Enable the following checkboxes and click OK.

I/O read bytes is the number of bytes read in input/output operations generated by a process, including file, network, and device I/Os. Whereas I/O write bytes is the number of bytes written in input/output operations by a process, including file, network, and device I/Os. I/O Read Bytes & I/O Write Bytes directed to CONSOLE (console input object) handles are not counted.

- Next, sort the listings by I/O Read bytes and see which application is generating the maximum I/O (in bytes/sec). Similarly, sort by I/O Write bytes to see which program is writing to the hard disk continuously.

- Once you identify the program, decide if you need the program or not. Leave it as it is if the I/O operations are justified. Else, remove the program or consult its documentation to tweak the settings if any. For instance, one of your browser extensions may cause high disk or CPU usage. You need to isolate the extension, add-on or the browser's feature causing the trouble.

Exercise 03. [30p] Process Monitor

Download the archive logs-final.7z and check if you have "Process Monitor" installed (Windows 10).

logs-final.7z: parola

HandleLeak.7z: parola7

Task.7z: parola17

[10p] Task A - Checking logging file

Looking at the logs created by the two apps - bad.log, good.log - they are identical, but Logs.exe has a significantly longer running time compared to GoodLog.exe. Start Process Monitor.

If the 4 buttons in the black area on the upper part of the window are selected, Process Monitor will display the activity (in this order) for: registry, files, networking, process and thread activity. By unchecking them, the corresponding events will be no longer displayed. In the menu bar there is the Filter field. If selected, it will trigger a dropdown menu that contains another Filterfield. If this second Filter field is selected, it will open the window shown below. Replicate this on your computer.

Process Monitor

Time	Process Name	PID	Operation	Path	Result	Detail
3:23:01	dmv.exe	5680	ReadFile	C:\Windows\System32\dmv.exe	SUCCESS	Offset: 35,328 Len...
3:23:01	dmv.exe	5680	RegQueryValue	HKCU\SOFTWARE\Microsoft\Window...	SUCCESS	Type: REG_DW...
3:23:01	dmv.exe	5680	RegQueryValue	HKCU\SOFTWARE\Microsoft\Window...	SUCCESS	Type: REG_DW...
3:23:01	dmv.exe	5680	RegQueryValue	HKCU\SOFTWARE\Microsoft\Window...	SUCCESS	Type: REG_DW...
3:23:01	dmv.exe	5680	RegQueryValue	HKCU\SOFTWARE\Microsoft\Window...	SUCCESS	Type: REG_DW...
3:23:01	SearchIndexer...	9424	FileSystemControl	C:	SUCCESS	Control: FSCTL_R...
3:23:01	dmv.exe	5680	RegQueryValue	HKLM\SOFTWARE\Microsoft\Window...	SUCCESS	Type: REG_DW...
3:23:01	Explorer.EXE	9656	ReadFile	C:\Windows\explorer.exe	SUCCESS	Offset: 2,089,984...
3:23:01	Explorer.EXE	9656	ReadFile	C:\Windows\explorer.exe	SUCCESS	Offset: 2,130,944...
3:23:01	Explorer.EXE	9656	RegQueryKey	HKCU\Software\Classes	SUCCESS	Query: Name
3:23:01	Explorer.EXE	9656	RegQueryKey	HKCU\Software\Classes	SUCCESS	Query: Handle Tag...
3:23:01	Explorer.EXE	9656	RegOpenKey	HKCU\Software\Classes	SUCCESS	Query: Name
3:23:01	Explorer.EXE	9656	RegOpenKey	HKCR\Applications\Procmon64.exe	NAME NOT FOUND	Desired Access: R...
3:23:01	Explorer.EXE	9656	RegOpenKey	HKCU\Software\Classes	SUCCESS	Query: Name
3:23:01	Explorer.EXE	9656	RegOpenKey	HKCU\Software\Classes	SUCCESS	Query: Handle Tag...
3:23:01	Explorer.EXE	9656	RegOpenKey	HKCU\Software\Classes	SUCCESS	Query: Name
3:23:01	Explorer.EXE	9656	RegOpenKey	HKCU\Software\Classes\Applications\...	NAME NOT FOUND	Desired Access: R...
3:23:01	Explorer.EXE	9656	RegOpenKey	HKCR\Applications\Procmon64.exe	NAME NOT FOUND	Desired Access: R...
3:23:01	Explorer.EXE	9656	CreateFile	C:\Users\immurgan\AppData\Local\Te...	SUCCESS	Desired Access: R...
3:23:01	Explorer.EXE	9656	QueryBasicInfo	C:\Users\immurgan\AppData\Local\Te...	SUCCESS	Creation Time: 11/1...
3:23:01	Explorer.EXE	9656	CloseFile	C:\Users\immurgan\AppData\Local\Te...	SUCCESS	
3:23:01	Explorer.EXE	9656	RegQueryKey	HKLM	SUCCESS	Query: Handle Tag...
3:23:01	Explorer.EXE	9656	RegOpenKey	HKLM\Software\Microsoft\Windows\C...	SUCCESS	Query: Handle Tag...
3:23:01	Explorer.EXE	9656	RegQueryValue	HKLM\SOFTWARE\Microsoft\Window...	SUCCESS	Type: REG_DW...
3:23:01	Explorer.EXE	9656	RegCloseKey	HKLM\SOFTWARE\Microsoft\Window...	SUCCESS	
3:23:01	Explorer.EXE	9656	RegQueryKey	HKCU	SUCCESS	Query: Handle Tag...
3:23:01	Explorer.EXE	9656	RegOpenKey	HKCU\Software\Microsoft\Windows\C...	SUCCESS	Desired Access: Q...
3:23:01	Explorer.EXE	9656	RegQueryValue	HKCU\SOFTWARE\Microsoft\Window...	SUCCESS	Type: REG_DW...
3:23:01	Explorer.EXE	9656	RegCloseKey	HKCU\SOFTWARE\Microsoft\Window...	SUCCESS	
3:23:01	Explorer.EXE	9656	RegQueryKey	HKCU	SUCCESS	Query: Handle Tag...
3:23:01	Explorer.EXE	9656	RegOpenKey	HKCU\Software\Microsoft\Windows\C...	SUCCESS	Desired Access: Q...
3:23:01	Explorer.EXE	9656	RegQueryValue	HKCU\SOFTWARE\Microsoft\Window...	SUCCESS	Type: REG_DW...
3:23:01	Explorer.EXE	9656	RegCloseKey	HKCU\SOFTWARE\Microsoft\Window...	SUCCESS	
3:23:01	Explorer.EXE	9656	RegOpenKey	HKCU\SOFTWARE\Microsoft\Window...	NAME NOT FOUND	Desired Access: Q...
3:23:01	dmv.exe	5680	CreateFile	C:\Users\immurgan	NAME COLLISION	Desired Access: R...
3:23:01	dmv.exe	5680	RegQueryValue	HKCU\SOFTWARE\Microsoft\Window...	SUCCESS	Type: REG_DW...
3:23:01	dmv.exe	5680	RegQueryValue	HKCU\SOFTWARE\Microsoft\Window...	FAILURE	Type: REG_DW...

From the two dropdown menus in the upper part of the context window, select "Process Name" instead of "Architecture" and "is" instead of "contains". In the text field add Logs.exe, click the Add button and then the OK button. Open the terminal and run Logs.exe. After the program is done running, save the Process Monitor capture. Use Ctrl + X to reset all the events captured in Process Monitor. Go to Filter → Filter area, double-click on the filter that was just added and change Logs.exe with GoodLog.exe, then click Add and Ok. Start GoodLog.exe and save the capture once the program finishes running. Scroll down in the two capture-logs until you notice the activity for bad.log respectively good.log.

The image displays two screenshots of Process Monitor. The left screenshot shows a dense list of system events, including file operations and process actions, with a high frequency of entries. The right screenshot shows a much sparser list of events, with large gaps between entries, indicating a significant slowdown in logging. A red highlight is visible in the right screenshot, corresponding to the text in the adjacent paragraph.

Notice the difference. On the left-hand side it is shown the faster logging process, and on the right-hand side the slower one. Look in the red highlighted area to see the difference. On the left-hand side the logging file is opened, followed by continuous writing, while on the right-hand side the file is opened and closed for every writing operation which explains the significant slowdown.

To recap, Task Manager shows what processes use the disk intensively at the current time, Windows Performance Recorder / Windows Performance Analyzer show who used the disc during a longer time period, although they were showing the activity as belonging to the System process instead of our process. Using Process Monitor we could identify our processes' entire activity and we could determine why one is slower than the other. But what if we could find out which line in the code causes the problem? Go back to Process Monitor. Use the window of the badly written logging program (Logs.exe). Go to Options → Configure

Configure Symbols

Process Monitor uses symbols to resolve function names when displaying thread stack locations on the Stack page of an event's properties dialog.

If you do not require that information you do not need to configure symbols.

DbgHelp.dll path (version 6.0 or later):

Symbol paths:

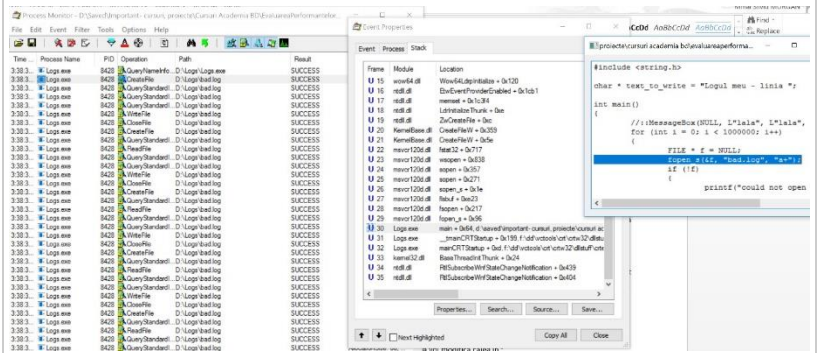
When displaying stack traces for modules for which you have both symbols and source code available Process Monitor can let you view the source associated with a stack frame.

Source code paths:

OK Cancel

Symbols, which will open the window shown below.

In the log (D:\Logs\bad.log) go to CreateFile. Double-click to open the Event Properties window. Choose the Stacktab, scroll down and you can notice that in the main function of main_bad_log.cpp, at line 12 the opening takes place. Click the "Source" button to view the source code containing the issue.



[20p] Task B - Investigating a handle leak

In logs-final.7z you have another example of two executables: **good.exe** and **bad.exe**. Both have the same outcome, the only difference being their running time (one of them is significantly slower). **Identify the problem.**

How to:

- A handle leak consists of a process that opens files and does not close them. On modern computers if this action is performed millions of times, the system may become unresponsive and will either experience an overall slowdown or the application that causes this will eventually crash. You may think that millions of handles are impossible to reach, so it is not worth paying attention to this problem. However, imagine that there are services running on servers for years. As an example, having a handle leak every 2 seconds amounts for over 10 million handle leaks in a year. How should such problems be investigated?

- Hint:** Open up a terminal and run HandleLeak.exe. Check out the "Details" tab in Task Manager after adding the "Handles" column.

Exercise 04. [30p] Process Explorer

- It can be noticed that the number of handles keeps growing. This is clearly a problem, but how do we investigate it?

How to:

- Run it as administrator. It is similar to Task Manager. Select the process that you are interested in, namely HandleLeak, and press "Ctrl + H".

- "Ctrl + H" opens a window under the "Process" section that displays all open handles along with information about them. Thus it will display file handles, registry handles, threads handles, and so on. There is another view (Ctrl + D) that displays all the loaded dlls.

- So it can be noticed that the leaks are on the following file: D:\Logs\HandleLeak\leak.txt. This is very useful information, but it would be better to find out who is responsible for the leak in code. Run Process Monitor with a filter on HandleLeak.exe and to notice the stack where the leakage is happens.