

PROJECT TITLE :

Power Consumption and Activity Monitor: Develop an application which is able to identify the tasks and activities consuming battery resources. The app should be able to log the historical monitoring data and make it available for the end-user to review it.

TEAM MEMBERS :

S. Abishek - AP19110010087

E. Dhanush - AP19110010053

B. Nikhila - AP19110010074

S. Chinmaya Datta - AP19110010136

SUMMARY :

One of the major characteristics of any computerized device is that most applications run in full screen mode in order to provide better interaction between the end user and devices. This makes it challenging for software developers to monitor the system and application statistical data in real time. Power Monitoring Tool helps to close the gap that provides real-time monitoring of key power-related system/app data.

Activity Monitoring Tool maintains a non-intrusive way to collect a minimum set of process related data in the background.

Power Monitoring Tool provides the feature to collect the set of power-related data in a periodic manner. In such a way, the system and application power-related states are recorded in a log file. In this report we completed our project for windows.

OBJECTIVE :

Monitoring operating system processes enables us to monitor and display process activity in real-time. To create a Power and Activity(Process) monitor for windows operating system using python - A command line version of task manager.

OVERVIEW OF PROJECT:

1. Importing dependencies

We need to import necessary libraries for building this project.

```
import psutil
from datetime import datetime
import pandas as pd
import time
import os
```

- **Psutil Library:**

psutil (process and system utilities) is a cross-platform library for retrieving information on **running processes** and **system utilization** (CPU, memory, disks, network, sensors) in Python. It is useful mainly for **system monitoring**, and **management of running processes**. Here it is used for retrieving running processes information in Python.

- **Pandas:**

Pandas is used after retrieving process information, we need to sort by columns and print in a tabular way.

2. Getting Process information

We need to store all the processes in a list of dictionaries, so later it can be easily converted to a dataframe.

Then the **psutil.process_iter()** function which returns a generator yielding a process instance for all running processes in the operating system.

oneshot():

process.oneshot() helps us retrieve process information efficiently (faster way), Utility context manager which considerably speeds up the retrieval of multiple process information at the same time. Internally different process info (e.g. name(), ppid(), uids(), create_time(), ...)

PID of 0 is eliminated which is the System Idle Process that is for Windows NT, it has no useful information.

3. Finding Different Attributes For Activity Monitor:

- Retrieving the time when the process was created in `timestamp`, as a result, we'll convert to a proper Python `datetime` object
- Finding CPU usage as well as the number of cores that can execute this process using `cpu_affinity()`
- `process.cpu_percent()` method returns a float representing the current process CPU utilization as a percentage. So if it is the first time called it will return 0.0.
- *Process Status*: Status of the process, whether it is running, sleeping, etc
- *Process priority*:
Nice is used to get the process priority here. `nice(value=None)` Get or set process niceness (priority). On UNIX this is a number which usually goes from -20 to 20. The higher the nice value, the lower the priority of the process.
- *Memory usage*:
`memory_full_info()` is used to get the memory usage of the process
- Total written and read bytes by this process
- Total threads spawned
- Finally, the user that spawned that process with username

4. Creating a Dataframe for Process monitor:

- Add all the attribute information to our list with the help of append and get out of the loop and return it
- Convert process list into a dataframe with help of Pandas
- set the process id as index of a process
- sort rows by the column passed as argument
- pretty printing bytes with the help of `getsize()`
- It sorts the rows by the column `sort_by` that will be passed as command-line arguments
- It also formats the `create_time` as a readable date.

5. Command-line argument parsing:

Python's `argparse` built-in module allows us to easily parse arguments passed from the command line (i.e terminal).

We added a bunch of arguments, such as `columns` to show, `sort_by` column that we'll sort by in the dataframe, `number` of processes to show, and `live_update` which is responsible for whether you want to keep the program running and continuously updating and printing process information each time (like `top` command in Linux).

6. Finally call our functions we made and show the dataframe:

- We use `psutil.sensors_battery()` to find the battery status and percentage.
- Printing all the processes with the help of functions used and the dataframe is visible for the end user to review it.
- We use the `head()` method here which prints the first n rows.

RESULT OF PROJECT:

Run the project in the command line and with help of various arguments we can sort the monitoring of process in the dataframe visible for the end user.

Run the following command in the terminal to view the output

`D:/anaconda3/python.exe d:/activity.py`

```
(base) D:\>D:/anaconda3/python.exe d:/activity.py
=====
POWER CONSUMPTION
Battery percentage : 100
Power plugged in : True
=====
ACTIVITY MONITOR
=====
pid      name      cpu_usage memory_usage read_bytes write_bytes status      create_time  nice  n_threads  cores
4         System    0.0      0.00B      110.30MB  662.18MB  running  1970-01-01 05:30:00  0     226        0
3952      svchost.exe 0.0      0.00B      3.39MB   96.72KB   running  2021-11-14 15:23:11  0     7          0
3944      svchost.exe 0.0      0.00B      0.00B    0.00B     running  2021-11-14 15:23:11  0     2          0
3936      ICESoundService64.exe 0.0      0.00B      516.00B  0.00B     running  2021-11-14 15:23:11  0     6          0
3928      RtkBtManServ.exe 0.0      0.00B      0.00B    83.65KB   running  2021-11-14 15:23:11  0     2          0
3920      svchost.exe 0.0      0.00B      2.85MB   448.39KB   running  2021-11-14 15:23:11  0     12         0
3912      svchost.exe 0.0      0.00B      6.51MB   55.73MB   running  2021-11-14 15:23:11  0     5          0
3904      svchost.exe 0.0      0.00B      15.62MB  12.00MB    running  2021-11-14 15:23:11  0     16         0
3772      SearchIndexer.exe 0.0      0.00B      23.52MB  2.00MB     running  2021-11-14 22:49:35  0     18         0
3688      svchost.exe 0.0      0.00B      98.05KB  178.49KB   running  2021-11-14 15:23:08  0     4          0
9668      svchost.exe 0.0      0.00B      7.11KB   0.00B      running  2021-11-14 15:25:15  0     3          0
3532      svchost.exe 0.0      0.00B      116.00B  32.16KB    running  2021-11-14 15:23:08  0     13         0
3496      spoolsv.exe 0.0      0.00B      53.01KB  41.74KB    running  2021-11-14 15:23:08  0     7          0
3440      svchost.exe 0.0      0.00B      9.64KB   18.36KB    running  2021-11-14 17:26:29  0     3          0
3388      svchost.exe 0.0      0.00B      0.00B    0.00B      running  2021-11-14 15:23:08  0     5          0
3332      svchost.exe 0.0      0.00B      202.77KB 160.00B    running  2021-11-14 15:23:07  0     13         0
9680      jhi_service.exe 0.0      0.00B      3.57KB   0.00B      running  2021-11-14 15:25:15  0     2          0
3128      svchost.exe 0.0      0.00B      442.73KB 164.00KB   running  2021-11-14 15:24:11  0     2          0
3092      svchost.exe 0.0      0.00B      0.00B    0.00B      running  2021-11-14 15:23:07  0     11         0
3084      svchost.exe 0.0      0.00B      0.00B    0.00B      running  2021-11-14 15:23:07  0     4          0
3076      svchost.exe 0.0      0.00B      1.61KB   0.00B      running  2021-11-14 15:23:07  0     13         0
3020      svchost.exe 0.0      0.00B      0.00B    0.00B      running  2021-11-14 15:23:07  0     12         0
10008     LMS.exe     0.0      0.00B      931.78KB 178.85KB   running  2021-11-14 15:25:16  0     4          0
2936      svchost.exe 0.0      0.00B      0.00B    0.00B      running  2021-11-14 15:23:07  0     8          0
3960      svchost.exe 0.0      0.00B      22.50KB  0.00B      running  2021-11-14 15:23:11  0     3          0
```

Users can sort upto which values they need and print them according with the sort-by argument.

```
d:/activity.py --sort-by memory_usage --descending
```

```
(base) D:\>D:/anaconda3/python.exe d:/activity.py --sort-by memory_usage --descending
```

POWER CONSUMPTION										
Battery percentage : 100										
Power plugged in : True										
ACTIVITY MONITOR										
pid	name	cpu_usage	memory_usage	read_bytes	write_bytes	status	create_time	nice	n_threads	cores
11032	Code.exe	0.0	577.78MB	688.67MB	65.52MB	running	2021-11-16 07:04:42	32768	14	8
9412	firefox.exe	0.0	330.77MB	4.93GB	97.18MB	running	2021-11-16 06:41:41	32	63	8
960	firefox.exe	0.0	251.21MB	10.62GB	6.94GB	running	2021-11-16 06:41:39	32	94	8
4480	firefox.exe	0.0	229.06MB	163.41MB	1.46GB	running	2021-11-16 07:26:28	64	30	8
11844	firefox.exe	0.0	206.21MB	3.22GB	146.19MB	running	2021-11-16 06:41:47	32	29	8
3284	firefox.exe	0.0	135.79MB	91.98MB	521.90MB	running	2021-11-16 06:43:11	32	39	8
664	Code.exe	0.0	105.78MB	754.81MB	758.26MB	running	2021-11-16 07:05:44	32	30	8
16480	Code.exe	0.0	97.73MB	37.08MB	2.13MB	running	2021-11-16 07:06:13	32	17	8
16592	Code.exe	0.0	88.26MB	88.43MB	11.22MB	running	2021-11-16 07:05:45	32	15	8
4988	firefox.exe	0.0	59.58MB	7.88MB	279.07MB	running	2021-11-16 08:13:48	64	28	8
8560	firefox.exe	0.0	53.02MB	9.48MB	14.40MB	running	2021-11-16 09:45:08	32	33	8
8280	Zoom.exe	0.0	40.25MB	31.78MB	386.56KB	running	2021-11-16 08:01:43	32	48	8
16264	explorer.exe	0.0	37.95MB	27.83MB	44.25KB	running	2021-11-16 06:41:15	32	77	8
16000	python.exe	0.0	35.25MB	8.08MB	0.00B	running	2021-11-16 09:47:10	32	4	8
13320	firefox.exe	0.0	26.14MB	14.17MB	121.33MB	running	2021-11-16 07:16:20	64	29	8
13656	Code.exe	0.0	23.29MB	97.23MB	46.85MB	running	2021-11-16 07:04:49	32	22	8
11648	firefox.exe	0.0	22.44MB	10.49MB	194.97MB	running	2021-11-16 06:42:16	64	29	8
16616	firefox.exe	0.0	22.22MB	7.17MB	5.14MB	running	2021-11-16 07:54:43	32	28	8
16284	firefox.exe	0.0	20.85MB	12.48MB	46.42MB	running	2021-11-16 06:46:33	64	28	8
16096	firefox.exe	0.0	18.60MB	6.75MB	16.78MB	running	2021-11-16 07:54:49	64	28	8
6476	Code.exe	0.0	18.55MB	1.89MB	1.29MB	running	2021-11-16 07:04:54	32	16	8
2740	firefox.exe	0.0	18.14MB	8.94MB	3.74MB	running	2021-11-16 06:41:48	32	28	8
16268	Code.exe	0.0	17.96MB	100.71MB	85.64MB	running	2021-11-16 07:04:41	32	30	8
5104	firefox.exe	0.0	17.65MB	7.18MB	59.23MB	running	2021-11-16 07:30:36	64	28	8
14732	Zoom.exe	0.0	7.44MB	33.35MB	302.55KB	running	2021-11-16 08:01:34	32	28	8