

Team 16

# **Process Manager**

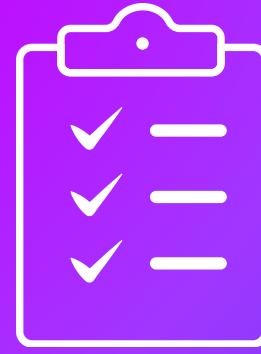
Presentated by Jessica H , Phillip O, Tyson L, Taoffek A, Vincent C

# Introduction

Identifying system and user processes is like checking who's doing what on a computer and how much they're using its power. It's important to know because some tasks can hog a lot of the computer's energy, like how some apps might make your phone battery drain faster.

Categorizing these tasks based on how much they use the computer's power (like its brainpower, memory, or how fast it reads and writes data) helps manage everything better. It's like making sure the most important things get what they need without slowing down everything else.

# Tasks



- 1 Tyson  
Responsible for creating the UI for the process manager, and assisting with implementing the manager and debugging it.
- 2 Phillip  
Responsible for creating a data structure to show CPU Utilization, and creating a data structure to show process name. Also assisted with implementing the process manager and debugging it.
- 3 Jessica  
Responsible for creating a data structure to see if the status is CPU Bound or I/O Bound.
- 4 Vincent  
Responsible for creating a data structure to show I/O utilization and creating a data structure to show memory utilization.
- 5 Taoffek  
Responsible for creating the data structure to pull Process ID.

# Motivation & Problem Formulation

## Motivation:

- Emphasizing the significance of categorizing processes for effective resource allocation and task prioritization.

## Problems

- Accuracy and Timeliness: Sometimes there are difficulties in real-time monitoring and assessment of process resource usage, impacting decision-making for resource allocation.

# Method

## **Step 1: Define the Process Manager Class**

Create a class to consolidate the functionalities and data structures needed for managing processes.

## **Step 2: Define Data Structures**

Create data structures based on the responsibilities assigned to each team member. These might include dictionaries, lists, or custom objects to store Process IDs, status (CPU Bound or I/O Bound), process names, CPU utilization, I/O utilization, memory utilization, and UI components.

# Method

## Step 3: Implement Methods for Each Task

- Process ID Retrieval (Taoffek's Task):
  - Create a method to retrieve Process IDs using the data structure designed by Taoffek.
- Status Checking (Jessica's Task):
  - Develop a method to determine if a process is CPU Bound or I/O Bound using Jessica's designed data structure.
- Process Information Retrieval (Vincent's and Phillip's Tasks):
  - Create a method to gather Process Names, CPU Utilization, I/O Utilization, and Memory Utilization using data structures created by Phillip and Vincent.
- UI Display (Tyson's and Phillip's Task):
  - Implement a method to showcase the UI components designed by Tyson for the process manager.

# Method

## Step 4: Test and Debug

- Test the methods to ensure they function as expected and handle various scenarios. Debug any issues encountered during testing.

## Step 5: Documentation

- Document the functionality of each method, outlining its purpose, and expected output.
- Explaining the project, input, output, and how to run the code.

# Results



Process Manager				
PID	Name	CPU Usage (%)	Memory Usage (bytes)	Bound Type
17467	java	4.9	186609664	I/O Bound
18549	Google	0.0	284856320	I/O Bound
18119	Google	0.0	233316352	I/O Bound
14198	Google	17.7	298401792	I/O Bound
9466	idea	0.0	778305536	I/O Bound
1399	Discord	4.1	166256640	I/O Bound
949	Google	0.3	217587712	I/O Bound

Process Manager with no  
Underlying Processes

# Results



Process Manager				
PID	Name	CPU Usage (%)	Memory Usage (bytes)	Bound Type
25209	java	23.6	184700928	I/O Bound
1399	Discord	4.6	173682688	I/O Bound
949	Google	0.2	241582080	I/O Bound
25111	Google	0.0	104271872	I/O Bound
24280	Google	0.0	226963456	I/O Bound
18549	Google	0.0	254672896	I/O Bound
14104	Google	0.0	104550400	I/O Bound
9466	idea	0.0	545075200	I/O Bound
35964	N/A	-1.0	1851392	I/O Bound
35943	N/A	-1.0	6782976	I/O Bound
35720	screencapture	1.6	12750848	I/O Bound
34889	N/A	-1.0	5480448	I/O Bound
34580	mdworker_shared	0.0	7438336	I/O Bound
33374	N/A	-1.0	1622016	I/O Bound
32459	mdworker_shared	0.0	5976064	I/O Bound
32458	mdworker_shared	0.0	5828608	I/O Bound
32275	replayd	0.0	5844992	I/O Bound
32029	triald	0.0	7258112	I/O Bound
32007	N/A	-1.0	5689344	I/O Bound
30833	N/A	-1.0	10125312	I/O Bound
30541	N/A	-1.0	11247616	I/O Bound
20000	N/A	-1.0	8032276	I/O Bound

Process Manager with all  
Processes

# References

---

 Stack Overflow

---

 GeeksforGeeks

---

 YouTube

---

 Knowledge and Collaborations from Team Members

---