#### **LAB REPORT ON**

#### **air crafts avionics using priority scheduling**

#### A Dissertation submitted to JNTU Hyderabad in partial fulfillment of the academic requirements for the award of the degree.

**Bachelor of Technology**

**in**

**Computer Science and Engineering (Data Science)**

Submitted by

K.KAUSHIK – 22H51A66F9

K.SUSHANTH – 22H5A66G0

K.ADITI -22H51A66G1

Under the esteemed guidance of

**Mrs. VIJETHA**

(Assistant Professor)



**Department of Computer Science and Engineering (Data Science)**

### CMR COLLEGE OF ENGINEERING & TECHNOLOGY

(UGC Autonomous)

\*Approved by AICTE \*Affiliated to JNTUH \*NAAC Accredited with A+ Grade

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD - 501401.

### 2022- 2026

**CMR COLLEGE OF ENGINEERING & TECHNOLOGY**

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD – 501401

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)**



### CERTIFICATE

This is to certify that the Lab Project report entitled **"AIR CRAFT AVIONICS USING PRIORITY SCHEDULING"** being submitted by K.Kaushik srinivas-22H51A66F9,K.sushanth-22H51A66G0,K.Aditi-22H51A66G1 in partial fulfillment for the award of **Bachelor of Technology in Computer Science and Engineering (artificial intelligence and machine learning)** is a record of bonafide work carried out his/her under my guidance and supervision.

#### The results embodies in this project report have not been submitted to any other University or Institute for the award of any Degree.

**Mrs. U. Hema Latha Dr. L. Chandra Sekhar Reddy**

**Assistant Professor Associate Professor and HOD**

**Dept. of CSE(DS) Dept. of CSE(DS)**

**External Examiner**

### ACKNOWLEDGEMENT

With great pleasure we want to take this opportunity to express my heartfelt gratitude to all the people who helped in making this project work a grand success.

We are grateful to **Mrs.vijetha**, **Assistant Professor**, Department of Computer Science and Engineering(Data Science) for his valuable technical suggestions and guidance during the execution of this projectwork.

We would like to thank **Dr. L. Chandra Sekhar Reddy**, Head of the Department of ComputerScience and Engineering(Data Science), CMR College of Engineering and Technology, who is the major driving forces to complete my project work successfully.

We are highly indebted to **Major Dr. V A Narayana,** Principal, CMR College of Engineering and Technology, for giving permission to carry out this project in a successful and fruitful way.

We would like to thank the **Teaching & Non- teaching** staff of Department of Computer Science and Engineering(Data Science) for their co-operation

We express our sincere thanks to **Shri. Ch. Gopal Reddy**, Secretary, CMR Group of Institutions, for his continuous care.

Finally, We extend thanks to our parents who stood behind us at different stages of this Project. We sincerely acknowledge and thank all those who gave support directly and indirectly in completion of this project work.

K.KAUSHIK SRINIVAS – 22H51A66F9

K.SUSHANTH – 22H5A66G0

K.ADITI -22H51A66G1

|  |  |  |
| --- | --- | --- |
| **TABLE OF CONTENTS** | | |
| **CHAPTER**  **NO.** | **TITLE** | **PAGE NO.** |
|  |  |  |
|  | ABSTRACT | I |
| **1** | **INTRODUCTION** |  |
|  | 1.1 Problem Statement | 2 |
|  | 1.2 Objective | 2 |
|  | 1.3 Project Scope and Limitations | 3 |
| **2** | **PROPOSED SYSTEM** |  |
|  | 2.1. Advantages of Proposed System | 5 |
|  | 2.2 Implementation   * 1. Design   2. Code and output | 6  7  7-9 |
| **3** | **RESULTS AND DISCUSSION** |  |
|  | 3.1 Result And Discussion | 11 |
| **4** | **CONCLUSION**  4.1 Conclusion and Future Enhancement |  |
|  | 13 |
|  | **REFERENCES** | 15 |

**ABSTRACT**

The following C code demonstrates a priority-based scheduling algorithm for processes, where each process represents an aircraft task in the context of avionics. In the avionics domain, tasks are characterized by their execution times (burst times) and priorities. The objective is to efficiently schedule these tasks to minimize waiting times and improve overall system performance.

To adapt this scheduling algorithm for aircraft avionics, the burst time and priority inputs can be related to the estimated execution time and criticality of each avionic task, respectively. The program employs a selection sort algorithm to arrange tasks in ascending order based on priority. The waiting time and turnaround time are then calculated for each task, reflecting the time spent waiting in the queue and the total time taken from task submission to completion.

The adapted algorithm serves as a basis for optimizing task scheduling within the avionics system, ensuring that critical tasks with higher priorities are executed promptly. The output provides insights into the performance of the scheduling algorithm, including average waiting time and average turnaround time, metrics crucial for assessing the efficiency of the avionics system.

Please note that when implementing this in an aircraft avionics system, it is essential to consider real-time constraints, safety requirements, and the specific characteristics of avionic tasks.

# CHAPTER 1

## INTRODUCTION

### CHAPTER 1 INTRODUCTION

##### Problem Statement

The Memory Matching Game project aims to address the need for a visually engaging and interactive memory-based gaming experience. The problem at hand involves designing an efficient and enjoyable memory matching game that not only tests and improves users' memory capabilities but also provides pleasing and user-friendly experience. The incorporation of randomized tile placements, a countdown timer, and real-time score updates aims to make the game more dynamic and challenging. By addressing these challenges, the project endeavors to deliver a well-rounded solution that combines the cognitive benefits of a memory game with an engaging and visually appealing user interface.

##### Research Objective

##### The research objectives for the Memory Matching Game project encompass assessing the cognitive impact of the game on memory, concentration, and pattern recognition skills through empirical studies. The study aims to analyze user engagement with the graphical user interface created using Python's Turtle graphics library, evaluating interaction patterns and overall satisfaction. Additionally, the effectiveness of the randomization strategy for tile placements will be explored to understand its role in sustaining player interest and challenging memory skills. The project also investigates the influence of temporal pressure introduced by a countdown timer on player performance, aiming to find the optimal balance between challenge and enjoyment. Ultimately, the research seeks to contribute to the formulation of design guidelines for educational memory games, providing insights into creating effective and engaging learning experiences.

* 1. **Scope**

1.Flight Control Systems:

* Prioritize control algorithms for ailerons, elevators, and rudders to ensure real-time response and stability.
* Allocate resources based on the criticality of flight control tasks during different flight phases.

2.Navigation Systems:

* Schedule GPS and inertial navigation system updates with high priority to maintain accurate position information.
* Ensure timely execution of route planning and trajectory calculations.

3.Communication Systems:

* Prioritize air-to-ground and inter-aircraft communication for critical updates and coordination.

Optimize data transmission scheduling for reliable and timely exchange of information.

##### Limitation

1. Complexity: Aircraft avionics systems can be incredibly complex, with numerous tasks and processes running simultaneously. Assigning priorities to these tasks can be challenging and may require extensive analysis and testing.

2. Task Interdependencies: Avionics tasks often have interdependencies, meaning that the completion of one task may be dependent on the successful execution of another. Prioritizing tasks in such scenarios can be tricky and may require careful consideration.

3. Dynamic Environment: The aviation industry operates in a dynamic environment, where conditions can change rapidly. This dynamic nature can affect the priority of tasks, as certain tasks may become more critical or time-sensitive based on real-time factors such as weather conditions or air traffic.

4. Resource Constraints: Avionics systems have limited resources, such as processing power, memory, and bandwidth. Allocating these resources efficiently while considering task priorities is crucial for optimal system performance.

5. Trade-offs: Priority scheduling involves making trade-offs between different tasks and their associated priorities. Sometimes, prioritizing one task may result in delays or compromises for other tasks. Striking the right balance is essential but can be challenging.

# CHAPTER 2

## PROPOSED

## SYSTEM

### CHAPTER 2

**PROPOSED SOLUTION**

##### Advantages of proposed System

1.Enhanced Safety and Reliability:

* Prioritize critical avionics tasks to ensure the safety of the aircraft.
* Reduce the risk of system failures by allocating resources based on priority.

2.Optimized System Performance:

* Improve overall avionics system efficiency by scheduling tasks based on their importance and urgency.
* Ensure that essential functions receive sufficient resources to operate smoothly.

3.Real-time Responsiveness:

* Achieve low-latency responses for time-sensitive avionics applications.
* Guarantee timely execution of critical tasks, such as navigation and collision avoidance.

4.Resource Allocation Management:

* Allocate computational resources judiciously to prevent bottlenecks and optimize system performance.
* Manage processor, memory, and communication bandwidth efficiently for various avionics subsystems.

##### Implementation:

1.Input:

* Read the total number of flights n.
* For each flight i from 1 to n:
* Read and store the arrival time, departure time, and priority of the flight.

2.Sorting Flights Based on Priority:

* Use the selection sort algorithm to sort the flights based on priority.
* For each flight i from 1 to n:
* Find the flight with the minimum priority among the remaining flights.
* Swap the current flight with the flight having the minimum priority.

3.Print Order of Flights:

* For each flight i from 1 to n:
* If it is the first flight, print "Flight i goes first."
* Otherwise, print " -> Flight i goes next."

4.Output Sorted Flights:

* Print the table of sorted flights with columns for Flight ID, Arrival Time, Departure Time, and Priority.

5.End

##### DESIGN:

##### 1. Task Classification: Start by classifying tasks based on their criticality and time sensitivity. Identify tasks that are essential for flight safety and those that can be delayed or deprioritized if necessary.

##### 2. Priority Assignment: Assign priorities to tasks based on their classification. Critical tasks, such as flight control systems or collision avoidance, should have higher priorities, while less critical tasks, like system diagnostics or data logging, can have lower priorities.

##### 3. Resource Allocation: Take into account the available system resources, such as processing power, memory, and bandwidth. Allocate resources to tasks based on their priorities, ensuring that critical tasks have sufficient resources for timely execution.

##### 4. Task Dependencies: Consider the dependencies between tasks. If a task relies on the completion of another task, ensure that the dependent task has a higher priority or is executed before the dependent task.

##### 5. Dynamic Adjustments: Build flexibility into the system to handle dynamic changes in task priorities. Real-time factors like weather conditions, air traffic, or emergency situations may require adjustments to task priorities on the fly.

##### 6. Testing and Validation: Thoroughly test and validate the system to ensure that the priority scheduling mechanism functions as intended. Simulate various scenarios and stress test the system to identify any potential issues or conflicts.

**2.4 Code :**

#include <stdio.h>

// Structure to represent a flight

struct Flight {

int id; // Flight ID

int arrivalTime; // Arrival time

int departureTime; // Departure time

int priority; // Priority based on some criteria (you can customize this)

};

int main() {

struct Flight flights[20];

int n, i, j, total = 0, pos, temp;

// Input the number of flights

printf("Enter the total number of flights: ");

scanf("%d", &n);

// Input details for each flight

printf("\nEnter Arrival Time, Departure Time, and Priority for each flight\n");

for (i = 0; i < n; i++) {

printf("\nFlight %d\n", i + 1);

flights[i].id = i + 1; // Flight ID

printf("Arrival Time: ");

scanf("%d", &flights[i].arrivalTime);

printf("Departure Time: ");

scanf("%d", &flights[i].departureTime);

printf("Priority: ");

scanf("%d", &flights[i].priority);

}

// Sorting flights based on priority using selection sort

for (i = 0; i < n; i++) {

pos = i;

for (j = i + 1; j < n; j++) {

if (flights[j].priority < flights[pos].priority)

pos = j;

}

temp = flights[i].priority;

flights[i].priority = flights[pos].priority;

flights[pos].priority = temp;

temp = flights[i].arrivalTime;

flights[i].arrivalTime = flights[pos].arrivalTime;

flights[pos].arrivalTime = temp;

temp = flights[i].departureTime;

flights[i].departureTime = flights[pos].departureTime;

flights[pos].departureTime = temp;

temp = flights[i].id;

flights[i].id = flights[pos].id;

flights[pos].id = temp;

// Print the order of flights

if (i == 0) {

printf("\nFlight %d goes first", flights[i].id);

} else {

printf(" -> Flight %d goes next", flights[i].id);

}

}

// Output the sorted flights based on priority

printf("\n\nFlight\tArrival Time\tDeparture Time\tPriority");

for (i = 0; i < n; i++) {

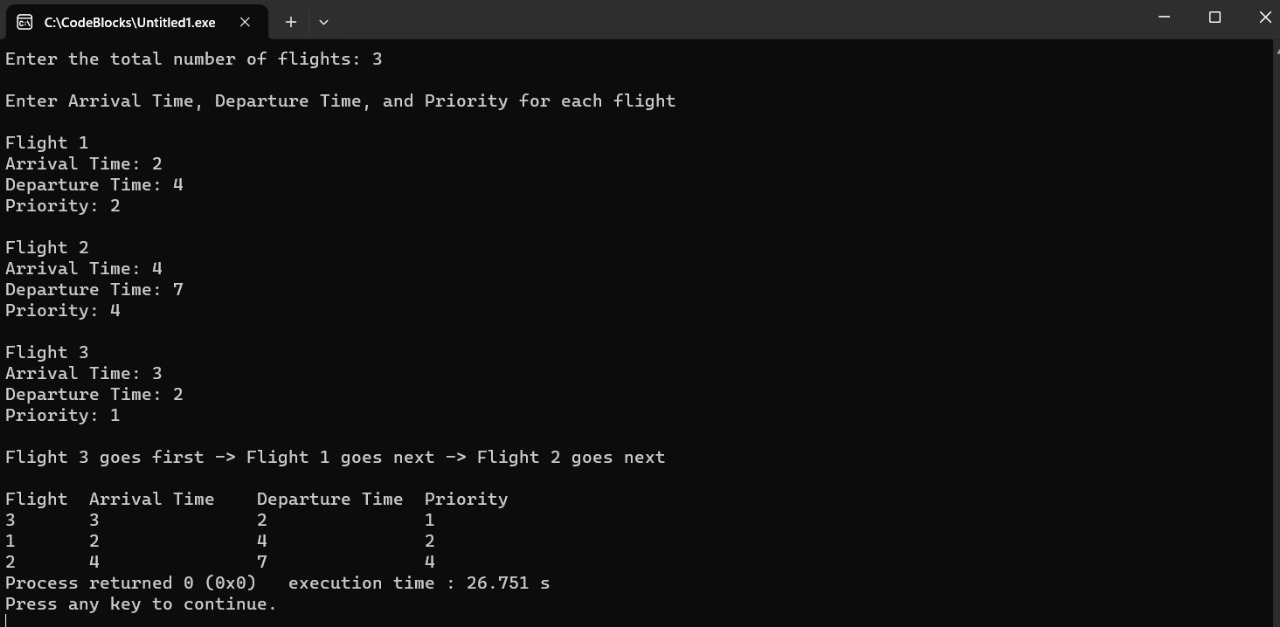
printf("\n%d\t%d\t\t%d\t\t%d", flights[i].id, flights[i].arrivalTime,

flights[i].departureTime, flights[i].priority);

}

return 0; }

**OUTPUT:**



# CHAPTER 3

## RESULTS AND DISCUSSION

### CHAPTER3

### RESULTS AND DISCUSSION

1. Improved Efficiency: Priority scheduling in avionics systems helps improve overall system efficiency by ensuring critical tasks are given higher priority and executed in a timely manner. This can enhance the safety and performance of the aircraft.

2. Enhanced Safety: By assigning higher priorities to critical tasks like flight control systems, collision avoidance, and sensor monitoring, avionics systems can prioritize safety-critical operations. This helps in mitigating potential risks and ensuring the safety of passengers and crew.

3. Resource Optimization: Priority scheduling allows for efficient allocation of system resources, such as processing power, memory, and bandwidth. By assigning resources based on task priorities, avionics systems can optimize resource utilization and improve overall system performance.

4. Adaptability to Dynamic Environments: The use of priority scheduling enables avionics systems to adapt to dynamic environments. Real-time factors like weather conditions, air traffic, or emergency situations can influence task priorities, allowing the system to respond effectively to changing circumstances.

5. Trade-offs and Challenges: While priority scheduling offers numerous benefits, there are also trade-offs and challenges to consider. Balancing task priorities and resource allocation can be complex, and there may be situations where lower-priority tasks might experience delays or compromises.

Overall, the results and discussions on aircraft avionics using priority scheduling highlight its positive impact on efficiency, safety, and resource optimization. It's an ongoing area of research and development, aiming to further enhance avionics systems and their performance.

# CHAPTER 4

## CONCLUSION

### CHAPTER 4

### CONCLUSION

##### Conclusion and Future Enhancement:

* + 1. **Conclusion:**

In conclusion, aircraft avionics systems using priority scheduling have proven to be highly effective in improving efficiency, enhancing safety, and optimizing resource allocation. By assigning priorities to tasks based on their criticality, avionics systems can ensure that critical operations are given the necessary attention and resources. This helps in mitigating risks, adapting to dynamic environments, and optimizing overall system performance. However, it's important to carefully balance task priorities and consider trade-offs to ensure the smooth operation of the avionics system. The field of aircraft avionics continues to evolve, and priority scheduling remains a crucial aspect in ensuring the reliability and safety of aircraft operations.

##### Future Enhancement:

1. Advanced Machine Learning: Integrating machine learning algorithms into avionics systems can enhance priority scheduling by allowing the system to learn and adapt to different scenarios. This can improve the accuracy of task prioritization and optimize resource allocation in real-time.

2. Predictive Analytics: By leveraging data analytics and predictive models, avionics systems can anticipate potential issues or failures in advance. This proactive approach can help prioritize maintenance tasks and allocate resources accordingly, minimizing downtime and improving operational efficiency.

3. Dynamic Priority Adjustments: Developing algorithms that can dynamically adjust task priorities based on real-time conditions can further optimize avionics systems. This flexibility allows for efficient resource allocation and adaptability to changing flight conditions, air traffic, and other factors.

4. Integration with Next-Gen Technologies: As aircraft technology continues to advance, integrating priority scheduling with emerging technologies like artificial intelligence, Internet of Things (IoT), and cloud computing can unlock new possibilities. This can lead to more efficient and intelligent avionics systems.

5. Collaborative Decision Making: Enabling avionics systems to communicate and collaborate with other onboard systems, air traffic control, and ground operations can enhance priority scheduling. This can facilitate better coordination and decision-making, improving overall flight efficiency and safety.

## REFERENCES

### REFERENCES