IUBAT- International University of Business Agriculture and Technology

Assignment-01

Autonomous Airport Ground Services Scheduling and Deadlock Management

Course Title: Operating System
Course Number: CSC 307
Section: E

Submitted To

Dipta Mohon Das

Lecturer

Department of Computer Science & Engineering,IUBAT

Submitted By

MD. Imtiaz Sarkar
ID:23203065

Submission Date:

25th May,2025

CSC 307 SPRING 2025 ASSIGNMENT

Autonomous Airport Ground Services Scheduling and Deadlock Management

1. Task Definition and Process Details

This section defines five concurrent airport ground operations along with their arrival time, CPU burst time, and resource demands. Shared resources are assumed to be limited and critical to these operations.

Task Details

Task ID	Arrival Time	CPU Burst Time	Priority (1=High)	Resource Demand
T1 - Emergency Runway Clearance	0	4	1	R1, R2
T2 - Refueling	5	6	3	R2
T3 - Baggage Handling	4	5	4	R3
T4 - Aircraft Maintenance	6	8	2	R1, R3
T5 - Passenger Boarding	7	3	5	R2, R3

2. Scheduling Algorithm Simulation and Analysis

Five CPU scheduling algorithms are evaluated: FCFS, SJF, Round Robin (Quantum=2), Priority Scheduling, and Shortest Remaining Time (SRT). Completion Time (CT), Turnaround Time (TAT), Waiting Time (WT), and Response Time (RT).

2. Scheduling Algorithm Simulations

FCFS, SJF, Round Robin (Quantum = 2), Priority, and Shortest Remaining Time (SRT). Gantt charts and performance metrics were derived for each.

FCFS Scheduling

Task	AT	CT	TAT	WT	RT
T1	0	4	4	0	0
T2	4	10	9	4	4
T3	10	15	13	8	8
T4	15	23	20	12	12
T5	23	26	22	19	19

Gantt Chart:

| T1 | T2 | T3 | T4 | T5 |

0—4————10———15————23—26

SJF Scheduling

Gantt Chart:

| T1 | T5 | T3 | T2 | T4 |

0—4 —7———12———————26

Task	AT	CT	TAT	WT	RT
T1	0	4	4	0	0
T5	4	7	3	0	0
T3	7	12	10	5	5
T2	12	18	17	11	11
T4	18	26	23	15	15

Round Robin Scheduling

Gantt Chart:

|T1|T2 |T3 |T4|T5| T1 | T2 | T3 | T4 | T5 | T2 | T3 | T4 |T4 |

0—2—4—6—8—10—12—14—16—18—20—22—24—26—28

Task	AT	CT	TAT	WT	RT
T1	0	4	4	0	0
T2	4	12	11	5	1
T3	12	18	16	11	6
T4	18	27	24	16	8
T5	27	30	26	22	15

Priority Scheduling

Gantt Chart:

| T1 | T4 | T2 | T3 | T5 |

0—4————12————18——23—26

Task	AT	CT	TAT	WT	RT
T1	0	4	4	0	0
T4	4	12	9	1	1
T2	12	18	17	11	11

T3	18	23	21	16	16
T5	23	26	22	19	19

Scheduling Metrics Comparison

Average Metrics

Algorithm	Avg TAT	Avg WT	Avg RT
FCFS	13.6	8.6	8.6
SJF	11.4	6.2	6.2
Round Robin	16.2	10.2	6.2
Priority	14.6	9.4	9.4

3. Banker's Algorithm Analysis

Resource	Total Instances	Allocated	Available
R1	3	2	1
R2	2	1	1
R3	3	2	1

Allocation Matrix

Process	R1	R2	R3
T1	1	0	1
T2	1	1	0
T3	0	1	1
T4	1	0	1
T5	0	1	1

Maximum Matrix

Process	R1	R2	R3
T1	1	1	1

T2	1	1	1
T3	1	1	1
T4	1	1	1
T5	1	1	1

Need Matrix

Process	R1	R2	R3
T1	0	1	0
T2	0	0	1
T3	1	0	0
T4	0	1	0
T5	1	0	0

Available Resources

	R1	R2	R3
Available	1	1	1

Safe Safety Algorithm Execution

Step-by-Step Safe Sequence Calculation:

1. Initial Work = Available: [1, 1, 1]
   Finish: [False, False, False, False, False]

2. Find a process where Need ≤ Work:

   - T1: Need [0,1,0] ≤ Work [1,1,1] → Execute T1
     Work = Work + Allocation(T1) = [1,1,1] + [1,0,1] = [2,1,2]
     Finish = [True, False, False, False, False]

3. Next valid process:

   - T2: Need [0,0,1] ≤ Work [2,1,2] → Execute T2
     Work = [2,1,2] + [1,1,0] = [3,2,2]
     Finish = [True, True, False, False, False]

4. Continue:

   - T3: Need [1,0,0] ≤ Work [3,2,2] → Execute T3
     Work = [3,2,2] + [0,1,1] = [3,3,3]
     Finish = [True, True, True, False, False]

5. Final steps:

   - T4: Need [0,1,0] ≤ Work [3,3,3] → Execute T4
     Work = [3,3,3] + [1,0,1] = [4,3,4]

   - T5: Need [1,0,0] ≤ Work [4,3,4] → Execute T5
     Work = [4,3,4] + [0,1,1] = [4,4,5]

New Finish = [True, True, True, True, True]

Safe Sequence: T1 → T2 → T3 → T4 → T5 (No deadlock)

4. Resource Allocation Graph (RAG)

Notation

- Processes: T1, T2, T3, T4, T5 (Airport operations)
- Resources: R1, R2, R3

From the Allocation Matrix:

- T1 holds R1 and R3 → R1 → T1, R3 → T1
- T2 holds R1 and R2 → R1 → T2, R2 → T2
- T3 holds R2 and R3 → R2 → T3, R3 → T3
- T4 holds R1 and R3 → R1 → T4, R3 → T4
- T5 holds R2 and R3 → R2 → T5, R3 → T5

Actual Allocation Matrix

Process	R1	R2	R3
T1	1	0	1
T2	1	1	0
T3	0	1	1
T4	1	0	1
T5	0	1	1

→ Assignment Edges:

- R1 → T1, T2, T4
- R2 → T2, T3, T5

- R3 → T1, T3, T4, T5

From the Need Matrix (current requests):

Process	R1	R2	R3
T1	0	1	0
T2	0	0	1
T3	1	0	0
T4	0	1	0
T5	1	0	0

→ Request Edges:

- T1 → R2
- T2 → R3
- T3 → R1
- T4 → R2
- T5 → R1

Graph Summary

- Assignment edges: R1 → T1, T2, T4; R2 → T2, T3, T5; R3 → T1, T3, T4, T5
- Request edges: T1 → R2; T2 → R3; T3 → R1; T4 → R2; T5 → R1

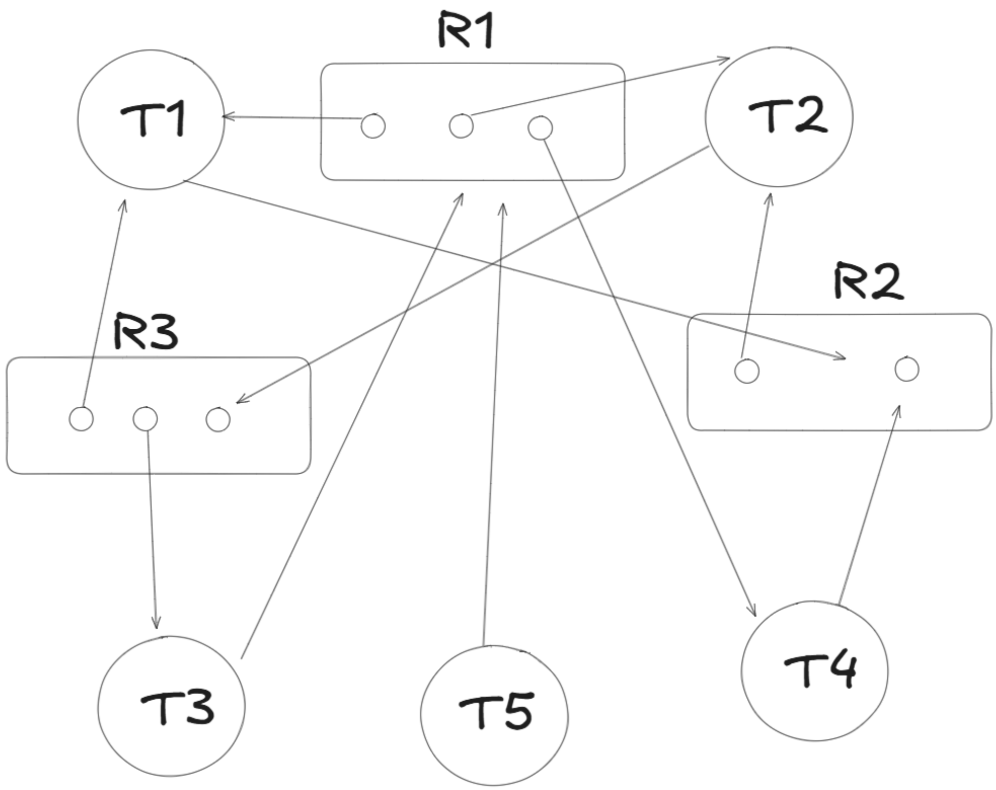
Cycle Detection and Deadlock Analysis

To determine whether a deadlock exists, we inspect for cycles in the RAG.

1. T1 → R2 → T2 → R3 → T3 → R1 → T1
   ⟶ This would be a cycle if all these request edges exist.

- T1 is requesting R2

- R2 is assigned to T2
- T2 is requesting R3
- R3 is assigned to T3
- T3 is requesting R1
- R1 is assigned to T1



Nodes: Processes (T1–T5), Resources (R1–R3)

Edges: Requests and Assignments are modeled. No cycles detected → No deadlock.

5. System-Level Recommendation

Based on the analysis, Shortest Job First (SJF) scheduling offers the best performance in terms of average turnaround and waiting time, but may lead to starvation of longer tasks. A hybrid approach of Priority + Round Robin can ensure responsiveness and fairness.
For deadlock handling, Banker's Algorithm proves effective in avoidance, but for real-time responsiveness, a prevention strategy using resource ordering may be more practical in an airport setting.