OPERATING SYSTEMS - Spring 2025

Project Title: AirControlX - Automated Air Traffic Control System

Total Marks: 350

Due Dates:

- Module 1: 20 April-2025
- Module 2: 30-April-2025
- Module 3: 06-May-2025
- Demos: 07–10 May -2025

Instructions

- This project is required to be completed in a group. A Group of 2 students are allowed
- No Cross-section groups are allowed.
- Zero marks will be awarded to the students involved in plagiarism.
- Project will be submitted in modules on given deadlines; however, evaluation will be done collectively in demonstrations.
- Failure to submit on the module deadline will result in deduction of -20% in each phase of project.
- You have to submit a single zip folder of .c/.cpp files.
- Folder name should be your name and Roll No.
- Also submit self-assessment form in final submission. Bring self-assessment form in hard form for demo.
- Be prepared for viva and demos after the submission of the project.

Project Statement

AirControlX is a simulation of an Automated Air Traffic Control System (ATCS) at a multi-runway international airport. This system will be developed in **C/C++** using **core Operating Systems concepts** such as process Management, Threading, Synchronization (mutexes, semaphores, Scheduling algorithms and Inter-process communication (IPC).The project includes live flight management, real-time analytics, airspace regulation enforcement, airline violation processing, and simulated payment. A visual simulation of runway operations and flight activity will also be implemented.

Module 1: System Rules & Restrictions [100 Marks]

Restrictions

1. Airlines

There will be six airlines simulated in the system as given in the following table:

Airline	Type	Aircrafts	Flights
PIA	Commercial	6	4
AirBlue	Commercial	4	4

FedEx	Cargo	3	2
Pakistan Airforce	Military	2	1
Blue Dart	Cargo	2	2
AghaKhan Air Ambulance	Medical	2	1

Flights are the aircrafts in operation.

2. There are three Aircraft Types

- Commercial Flight: Regular passenger aircraft.
- Cargo Flight: Heavy freight aircraft operating under restricted windows.
- Emergency Flight: MedEvac, military, or diverted flights with top priority.

3. There are three runways available at the airport:

- RWY-A: North-South alignment (arrivals)
- RWY-B: East-West alignment (departures)
- RWY-C: Flexible for cargo/emergency/overflow

Only one aircraft may use a runway at any given time.

4. Flight Arrivals and Dispatching

Direction	**Flight Type**	**Time**	**Emergency Probability**
North	International Arrivals	Every 3 minutes	10% (low fuel/diversion)
South	Domestic Arrivals	Every 2 minutes	5% (air ambulance)
East	International Departures	Every 2.5 minutes	15% (military/priority)
West	Domestic Departures	Every 4 minutes	20% (VIP or urgent cargo)

Flights are assigned runways based on direction:
RWY-A: North/South arrivals
RWY-B: East/West departures
RWY-C: Backup/emergency/cargo use

5. Cargo Flight Restrictions

RWY-C is exclusively used for cargo.
In each simulation, there must be 1 cargo flight.
Flights violating this restriction are denied or rescheduled unless marked high priority.

6. Flight Speed Monitoring

There are three Aircraft speed is based on the flight phase and altitude. Radar thread continuously monitors speed and phase, enforcing realistic restrictions.

Arrival Simulation: Initial Speed

For arrivals, the initial speed will be set similarly based on the aircraft's approach to landing:

1. Holding Phase (in holding pattern before landing):

o Initial Speed: Set between 400–600 km/h, based on the aircraft's approach and ATC instructions.

2. Approach Phase (final descent toward runway):

o Initial Speed: Set between 240–290 km/h. This will be adjusted as the aircraft gets closer to the runway.

o Violation Detection: If the aircraft's speed is below 240 km/h or above 290 km/h, a violation will occur.

3. Landing Phase:

o Initial Speed: Set to 240 km/h at the start of the landing phase.

o Deceleration: The aircraft will gradually decelerate to 30 km/h or lower.

o Violation Detection: If the aircraft fails to decelerate below 30 km/h, a violation is triggered.

4. Taxi Phase (after landing):

o Initial Speed: Randomly set between 15 km/h and 30 km/h (within allowed taxi range).

5. At Gate (Initial Speed):

o Initial Speed: 0 km/h (aircraft is stationary).

Arrival Flight Speed Rules

Aircraft State	Expected Speed (km/h)	Speed Range	Violation Criteria
Holding	400–600	400–600	Speed exceeds 600 km/h → **Violation** (too fast to approach). The aircraft is expected to hold 1 KM around the airport.
Approach	240–290	240–290	Speed below 240 km/h or above 290 km/h → **Violation**
Landing	240 → 30	240 → 30	Speed exceeds 240 km/h or fails to slow below 30 km/h → **Violation**
Taxi	15–30	15–30	Speed exceeds 30 km/h → **Violation** (not appropriate for taxi)
At Gate	0–5	0–5	Speed exceeds 10 km/h → **Violation** (should be stationary)

Departure Simulation: Initial Speed

For departures, the initial speed will be set based on the phase the aircraft is in:

1. At Gate (Initial Speed):

o Initial Speed: 0 km/h (aircraft is stationary at the gate)

o Transition to Taxi: When the aircraft begins taxiing, its speed will increase to match the Taxi phase speed range (15–30 km/h).

2. Taxi Phase (after Gate):

o Initial Speed: Set to a random value between 15 km/h and 30 km/h (within the allowed taxiing speed range).

o Transition to Takeoff Roll: Once at the runway, the speed will increase based on the Takeoff Roll phase speed.

3. Takeoff Roll (on the runway):

o Initial Speed: 0 km/h (starts from standstill).

o Speed at end of Takeoff Roll: Will increase towards the takeoff speed, up to 290 km/h.
o Violation Detection: If the speed exceeds 290 km/h before lift-off, a violation will be triggered.

4. Climb (after takeoff):
o Initial Speed: Set between 250–463 km/h depending on the aircraft's altitude.
o Transition to Cruise: After reaching 10,000 ft, the aircraft will switch to a cruise speed range of 800–900 km/h.

Departure Flight Speed Rules

Aircraft State	Expected Speed (km/h)	Speed Range	Violation Criteria
At Gate	0–5	0–5	Speed exceeds 10 km/h → **Violation** (should be stationary at gate)
Taxi	15–30	15–30	Speed exceeds 30 km/h → **Violation** (exceeds taxi limit)
Takeoff Roll	0–290	0–290	Speed exceeds 290 km/h → **Violation** (accelerating too fast)
Climb	250–463	250–463	Speed exceeds 463 km/h → **Violation** (max 250 knots below 10,000 ft)
Departure (Cruise)	800–900	800–900	Speed below 800 km/h or above 900 km/h → **Violation** (outside cruise limits)

Speed must comply with the aircraft's current phase and ATC rules. Violations will trigger AVNs.

7. Airspace Violation Notice (AVN)

Every aircraft starts with Inactive AVN status.
If a flight exceeds speed limits, an **AVN is activated** (like a challan).
Violations activate an AVN, billed to the airline

8. Simulation Time

Each simulation run is 5 minutes long. That is, you are expected to show a timer, demonstrating the various flights that land or depart from the airport.

8. Fault Handling (Ground Only)

Ground faults (e.g., brake failure, hydraulic leak) may occur during taxi/gate operations.
Faulty aircraft are towed and removed from queues.

Deliverable:

1. A code implementing the aforementioned restrictions via C/C++ structures and classes. Please note that this code will serve as the foundation for the next modules in the project.

If the code is not submitted for module 1, there will be **a severe penalty of 20% marks deducted.**

Module 02 ATCS Core Functionalities [130 Marks]

1 Flight Entry & Scheduling [15]
FR1.1: The system shall allow entry of arrival and departure flight data including flight number, airline, aircraft type, direction, priority, and scheduled time.
FR1.2: The system shall generate a schedule queue for both incoming and outgoing flights.
FR1.3: The system shall handle conflicts by automatically rescheduling or rerouting aircraft based on priority and runway availability.

2. Runway Allocation [15]
FR2.1: The system shall allocate runways based on flight direction:
RWY-A for arrivals (North/South)
RWY-B for departures (East/West)
RWY-C for cargo, emergency, and overflow
FR2.2: Only one aircraft may use a runway at a time; this must be enforced with thread synchronization.
FR2.3: In case of multiple high-priority arrivals/departures, RWY-C must be used as backup.

3. Arrival & Departure Flow Management [60]
FR3.1: The system shall manage aircraft progress through phases:
Holding → Approach → Land → Taxi → Gate (for arrivals),
Gate → Taxi → Takeoff → Departure (for departures).
FR3.2: Aircraft movements shall follow real-world timing and phase transitions controlled by simulation logic.
FR3.3: Arrival priority shall be based on fuel status, emergency level, or type (e.g., Emergency > VIP > Cargo > Commercial).

4. Priority Handling [20]
FR4.1: The system shall assign emergency and high-priority flights immediate runway access, pre-empting lower-priority flights.
FR4.2: The system must re-order queues when an emergency flight is registered.

5. Queue and Delay Management[10]
FR5.1: Flights unable to be assigned a runway must be queued based on priority and FCFS within priority class.
FR5.2: The system shall calculate estimated wait time for every aircraft.

6. Status Tracking[10]
FR6.1: The system must track each aircraft's current status (e.g., Waiting, Taxiing, Taking Off, Cruising, Landing, At Gate).
FR6.2: Status must be updated in real-time based on simulation time and transition logic.

Deliverable:

1. A code implementing the functionalities to the code developed in module 1.

If the code is not submitted for module 1, there will be **an additional severe penalty of 20% marks deducted.**

Module 3: Subsystems & Airline Integration [120 Marks]

Functional requirements: There will be processes managing the ATC, AVN Generator, AirlinePortal and StripePayment

1. ATCS Controller Process [30]

AirControlX will have these components to monitor and manage air traffic violations

Violation Detection:

AirControlX will monitor the speed, altitude, and position of every aircraft in the airspace.If any aircraft exceeds the designated airspace boundary, altitude limit, or speed restriction, the system will notify the AVN Generator Process.

Violation Status Updates:

The system will update the violation status to "active" for any aircraft found violating airspace or flight parameters. This status will be tracked and available for the air traffic controllers to take appropriate actions.

2. ATC Analytics

AirControlX will also maintain detailed analytics about the air traffic within the controlled airspace.

2.1 Air Traffic Analytics:

AirControlX Dashboard display the following data:

- Number of Active Violations: It will keep track of the number of aircraft currently in violation of flight rules (e.g., altitude, speed, and airline name).
- Aircraft with Active Violations: The system will display the aircraft's ID and relevant violation status for air traffic controllers to assess and address the situation.
- The simulation shall visually represent aircraft in different travel states (taxiing, takeoff, etc.).
- Runway occupancy and flight movement must be animated using a C++ graphics library (e.g., SFML/OpenGL).

2. AVN Generator Process [30]

a. The AVN Generator creates a detailed Airspace Violation Notice including:

b. AVN ID, Airline name, Flight number, Aircraft type, Speed recorded vs. permissible, AVN issuance date and time, Total fine amount (including service fee), Payment status (default: unpaid),and Due date (3 days from issuance). The Detail of payment is calculated as follows:

i. PKR 500,000 (Commercial)

ii. PKR 700,000 (Cargo)

iii. A 15% administrative service fee is added to the total

c. This information is forwarded to the Airline Portal and StripePay process.

d. Upon receiving a successful payment confirmation from StripePay, the AVN Generator:

i. Updates the payment status to "paid"

ii. Sends confirmation to the Airline Portal

iii. Notifies the ATC controller that the airline has cleared the violation

3. Airline Portal [20]

i. The Airline Portal is accessible by airline representatives using the aircraft ID and AVN issue date/time.

ii. Airlines admin can view all active and historical AVNs, including:

AVN ID

Aircraft ID (Flight Number)
Aircraft type
Payment status (paid/unpaid/overdue)
Total fine amount including service charges

iii. On receiving a successful payment alert from the StripePay process, the Airline Portal updates the corresponding AVN entry to reflect the payment and displays all associated details.

4. StripePay Process [20]

i.Airline Admin can pay the challan by using another process called StripePayment.

ii..Stripepayment will receive AVNID, Aircraft ID (Flight Number) , airctaft type and amount to be paid.

iii.Airline Admin will enter the paid amount and the stripePayment process will send successful payment status to AVN Generator Process and Airline Portal

5. Simulation [20]

The system will simulate and visualize the Air Traffic Control flow at the airport. You may use libraries such as SFML or openGL or any graphic library in C++ to create graphical representations of the intersection, vehicles, and traffic light states. The graphical output will include:

- visual representation of Aircraft, runways and Air traffic Control (ATC) Tower.
- color coded runways and aircraft to be distinguished in simulation
- Animated movement of Aircraft.
- Arrival Flights with flight status
- departure Flights with flight status

Deliverable:

1. A Complete code implementing the additional functionalities in the code developed in module 1 and module 2.
2. If the code is not submitted for module 3, there will be **an additional severe penalty of 20% marks deducted.**
3. Final submission includes self-assessment form.
4. Self-assessment form will be uploaded later.

Happy Coding 😊