Project
AirControlX

CS2006

Operating Systems

Submitted by: Taha Ahmed, Zara Tahir
Roll number: 23I-0532, 23I-0802
Date: 5/7/2025

Project Report: AirControlX — Airport Simulation System

Overview

AirControlX is a comprehensive simulation system designed to emulate real-world airport control operations. The system manages flights, runway allocations, and aviation violations, modeling essential processes such as flight scheduling, runway assignments, flight phase transitions, and violation handling.

Core Components and System Logic

1. Flight Generation and Management

Design Logic:

- Flights are generated concurrently in four directions—north, south, east, and west—utilizing multithreading to simulate real-time operations.
- Each flight is assigned randomly selected attributes, including airline, type (Commercial, Cargo, or Military), and emergency status.
- To reflect realistic airline operations, the system imposes constraints on the number of flights per airline.
- Flights undergo a defined sequence of phases: gate → taxi → takeoff → cruise → approach → landing → taxi → completion.
- Emergency and cargo flights are prioritized for handling and scheduling.

2. Flight Scheduling

Design Logic:

- Flights are categorized into either arrival or departure queues based on their current operational phase.
- These queues are implemented as priority queues, prioritizing flights by urgency and scheduled time. A First-Come-First-Serve (FCFS) approach is employed within the same priority level.
- The scheduler dynamically reorders the queues and estimates wait times accordingly.
- Emergency flights are given top priority and may preempt others in the queue.

3. Runway Management

Design Logic:

- The simulation incorporates three runways: A (arrival), B (departure), and C (flexible/emergency use).
- Runway allocation is determined based on flight type and operational phase:
  - Runway A handles arrivals.
  - Runway B handles departures.
  - Runway C serves emergencies, cargo, or overflow traffic.
- Mutexes ensure thread-safe runway assignments and releases.
- The system continually evaluates availability and allocates the most suitable runway, with special provisions for emergency and cargo flights.

4. Flight Controller

Design Logic:

- The controller oversees flight progression through operational phases.
- Flight speed is dynamically adjusted based on phase, with random fault simulations (e.g., brake failure) introduced to increase realism.

- The system detects and flags speed violations, forwarding them to the Aviation Violation Notice (AVN) subsystem.
- It handles phase transitions and updates flight status (completed, faulty, etc.) accordingly.

5. **Aviation Violation Notice (AVN) Management**

Design Logic:

- Tracks and records operational violations such as phase-specific speed infractions.
- Penalties are calculated based on violation severity, flight phase, and aircraft type.
- Generates detailed violation reports for each airline along with system-wide statistics.
- Supports data import/export functionalities to maintain persistence and enable post-simulation analysis.

6. **Multithreading and Simulation Control**

Design Logic:

- The system utilizes POSIX threads to concurrently manage flight generation, control, and AVN processing.
- Signal handling is incorporated for graceful termination of the simulation.
- The simulation operates for a predefined period (e.g., 2 minutes during testing) and ensures all threads are properly synchronized and terminated upon completion.

Key Algorithms and Design Decisions

- **Priority Queuing:**
  Used to efficiently manage flight scheduling by prioritizing emergency and critical flights.

- **Thread Synchronization:**
  Implemented using mutexes to avoid race conditions in shared resource management (e.g., runways).
- **Randomization**:
  Introduced in airline selection, emergency occurrence, and fault injection to enhance simulation variability and realism.
- **Phase-Based State Machine:**
  Each flight operates as a state machine, transitioning logically between defined operational phases.
- **Violation Detection:**
  Enforces phase-specific constraints (e.g., speed limits), logging violations for subsequent AVN processing.
- **Resource Constraints:**
  Imposes limits on flight counts per airline and mandates the inclusion of at least one cargo flight to mirror real-world limitations.

Example of Core Logic (Pseudocode)

```
// Flight Generation
for each direction in [north, south, east, west]:
    while simulation_running and within time limit:
        create new Flight with random airline, type, emergency
        if airline flight limit not reached:
            add to scheduler
            if runway available:
                assign runway
                start flight thread

// Flight Movement
while flight not completed:
    update speed based on phase
    check for violations
    if phase complete:
        transition to next phase
    sleep for simulation interval

// Runway Assignment
if flight is emergency or cargo:
    try assign flex runway (C), else A or B
else if departure:
    assign B, else A, else C if overflow
```

Conclusion

The AirControlX system offers a robust and modular simulation of airport operations, encompassing flight scheduling, runway assignment, violation tracking, and multithreaded execution. With its realistic constraints, priority-based algorithms, and dynamic behavior modeling, the project demonstrates the feasibility and complexity of managing modern airport environments through software simulation.