CSE 321 **Auto-Flight Controller System** **Fall2013**

Due date: 9/28/2013

Motivation:

Incident 1: "The near-miss involving three planes operated by US Airways happened on Tuesday in skies close to Reagan Washington National Airport….The FAA added that due to bad weather, air traffic controllers switched landing and departing operations and miscommunication "led to a loss of the required separation" between the jets. …The near-miss comes a year after two passenger planes were forced to land without help from the tower at the same airport, after the lone air traffic controller on duty fell asleep….That incident led to a review of staffing levels and worker fatigue at US airports.." August 2012.

Federal guidelines require craft to be separated by at least 1,000 vertical feet and 3.5 lateral miles.

Incident 2: "According to a statement issued by Hong Kong's Civil Aviation Department, a Cathay Pacific Airways flight that was bound for Hong Kong International Airport came within 1 nautical mile (2 km) of a Dragonair A330 airplane that was in a holding pattern for landing…The two jets, carrying more than 600 passengers and crew members, came within seconds of colliding, the newspaper reported, citing a former Hong Kong aviation official…." Sept 27, 2011.

Incident #3: "Airliners too close for comfort over New York", CNN, June 25, video, 2013.http://www.cnn.com/2013/06/21/travel/nyc-jfk-airliners-too-close

Problem Statement: Let us assume that Federal Aviation Administration (FAA) wants to address these and such other problems in due to traffic and lack of personnel using several means. One of the approaches we (CSE321ers) want to recommend is a combination of sensors on planes, collision detection system and a **real-time system** that automatically controls the flight landing, take-off and cruising. **Your task is to design, implement and test this realtime software** we will call "**Auto-Flight Controller" (AFLIC** not AFLAC**)**; assume that the sensors and collision detection systems are in place and we need to develop the software.

Study of existing systems: Of course, there is a system called Traffic Collision Avoidance System (TCAS) that is based on sensors and instrumentation placed on each airplane in service. However we feel it is local solution centered on the aircraft and does not provide a global view of the whole picture.http://en.wikipedia.org/wiki/Traffic_collision_avoidance_system

Requirements:

1. AFLIC is installed and operates at the airport (one system per airport) and monitors and controls the operation of the planes landing and taking of at the airport.
2. Once the flight leaves the area /vicinity of the airport and reaches a cruising altitude the monitoring the handed-over to an intermediate AFLIC or another AFLIC at an airport. So AFLIC is indeed a truly

distributed realtime monitoring system. (This handover from one AFLIC to another is similar how you cell phone is handed over from one cell tower to another.)

3. You need to worry about only one AFLIC. Just as the flight "goes out of range" of an AFLIC say that it is not anymore it is watch list and remove it from the monitoring list of this AFLIC.
4. AFLIC also keeps track of number of runways, whether they are active. A runway is said to be active if there is a flight on the runway either taking off or landing. For simplicity we will allow only one flight on any runway at any time. There are a finite number of runways at an airport. Assume that a runway is a mutually exclusive resource that can allow a take-off or a landing at any given time.
5. When a flight requests landing, if a runway is available it is allocated. If not, the flight keeps hovering around in a holding pattern. All the flights requesting landing are queued up and are served first come first serve basis.
6. When a flight taking off, it requests runway to takeoff, once allocated, it departs gate, and starts moving towards a runway. Otherwise it is queued up for an available runway.
7. Collision detection instrumentation (sensors, alarms etc.) will detect imminent mid-collision, run-way collision, and any collision within the airport when the flights are moving around. Appropriate action to be taken when collision detection happens.
8. Life cycle of a flight: (These are the states) AFLIC registers an active flight, it is at gate, gets clearance to take-off, moves towards a runway, requests runway, gets clearance for runway, goes into runway, takes-off, reaches cruising altitude , it is deleted (stored in background) from the AFLIC system of the airport. Exceptional situations are collision detection and the corrective operations that may happen at various points.
9. Assume other conditions and make sure you document your assumptions.
10. Design this realtime system, implement it and test it with a hypothetical airport handling, say, f (10) flights, r (2) runways, and single airport.

Design:

1. Start by designing the state diagram representing the life cycle of a flight. Also use a table-driven scheduling for the flights monitored by an AFLIC. You will have to deign these two before you can start implementing the code. You may have to revisit and revise your state diagram and the state transition table based on the needs.
2. Implement the AFLIC system using C/C++ language. Define a function for each of the operations. Assume collision detection as an asynchronous event. (How will you simulate an interrupt and handle it ?)
3. You can use multi-processing for simulation of several flights co-existing.
4. Test your system for various conditions using sample scripts.
5. For implementation of the operation you may print out the message and allow some predetermined elapsed time for the operation taking place.
6. On completion your program should AFLIC application will take sample scenarios as input and play out the scenarios and action taking place through a series of messages printed out.

What to submit? Please submit your design documents, well commented code online and clear simulation run outputs showing various conditions: submit cse-321 your files