**CIS 657 Lab 2**

Submit two files:

* Create your Lab 2 document (doc or pdf) including the **disclosure form**, **screenshot(s) of your output**, **your solution code (filename and your own code, not Nachos original code) with description (you need to explain your solution)**  to the **Turnitin link** on the blackboard (under Assignments).
* Do a “make clean” in code/build.linux, then compress your Nachos folder on the VM, download the compressed file into your local machine and name the compressed file **Lab2\_yourNetId.zip (Note: Zip only!!)**. Submit the zipped file to the **Assignment Code Submission link** of Blackboard.

Due: Sep. 21 (Friday, end of the day)

Late submission: you have total 72 hours throughout the semester.

*Follow the Lab1 instruction and create a new fresh Nachos folder.*

**Overview**

We will practice C++ in Nachos by **implementing Laundry-room simulations of 20 people** doing their laundry. A **sequence number 1 to 20** will be assigned (generated) to each person when he/she comes. The laundry room allows **one person at a time**, so the next person has to wait in the queue until the current one is done with all his/her laundry. *After finishing, a person leaves and the simulation continues until everyone leaves*.

There are two simulations:

1. People come and line up outside a laundry room based on their **priorities (0 to 9), which are randomly generated**. Then they start entering the laundry room, one by one according to their positions in the queue, to do their laundries.
2. People come and line up outside the laundry room one by one based on **their sequence numbers** (the sequence of their arrival). Then they start entering the room one by one **according to their priorities (randomly generated 0 to 9)**.

Each person has a certain amount of clothes to wash, which should take a certain amount of time. The **time should be randomly generated**.

**Output:**

For each simulation, you should print out the followings:

* Sequence number and priority information of each person in the queue after they line up outside the laundry room.
* After people start entering the room,
  + Order # (starting from 1)
  + Who is entering (person’s sequence # and priority)
  + Who is leaving and how long the leaving person has used the laundry room (if order # is 1, no leaving person exists)
  + Total laundry time

**Implementation Requirements:**

* Define a class for Person, which contains all the information you need for a person, and use it to create 20 people (Person objects).
  + Implement the Person class in the new files (declaration in .h file and definition in .cc file)
  + The member variables have to be private. Implement public getter/setter functions for accessing these variables
  + Change code/build.linux/Makefile to include the new files to the NachOS compilation process
* Call rand() function to randomly generate a number.
* Implement the waiting queue using the functions defined in code/lib/list.h and list.cc
  + Read all the codes and comments in list.h, list.cc, and code/threads/threadtest.cc
  + You need to sort 20 people differently for each simulation. For queue manipulations, the existing functions in list.h/cc should be enough.
  + Hint: ThreadTest() function is called by main()
* Use List::Apply() function for printing the information (sequence # and priority) of everyone lining up outside the laundry room
  + You need to define your own print function in threadtest.cc, for example:  
    void PrintWaiting(Person waiting);
  + List::Apply() will apply the function PrintWaiting to every element in the list.
* The entire simulation process should be done in threadtest.cc.
* Your implementation must be incorporated and compiled with NachOS under thread.
* Your Lab2 document must include your code from your class declaration, definition, and implementation of two simulations.

**Testing:**

We will build and run your Nachos on the VM.

**Grading:**

* Syntax/Runtime/Logic Errors with proper Makefile [-50, -15]
* Class declaration/definition respectively [-20, -10]
* Satisfy all implementation requirements [-50, -5]
* Output(by Students)/Test(by TAs) [-50]