Project 2

CS 4348.002

Prof. Greg Ozbirn

Vineeth Soma

February 28th, 2015

**Design:**

The project will be divided into three files – Project2, Customer, PostalWorker.

**Project2:**

**Semaphores:**

postOfficeLimit – this limits number of customers that can enter postoffice. Value: 10

postalWorkerLimit – this limits the number of customers that can use postal worker assistance. Value: 3

**Functions:**

initializePostalWorkers(): initializes and starts 3 threads of PostalWorker class

initializeCustomers(): initializes and starts 50 threads of customer class

endPostalWorkers(): joins 3 threads of postal workers once they stop executing.

endCustomers(): joins 50 threads of customers once they start executing

**Psuedocode:**

InitializePostalWorkers()

InitializeCustomers()

endPostalWorkers()

endCustomers()

**Customers:**

**Semaphores:**

poLimit- passed from Project2 class, usage limits the number of customer to enter post office. Value: 10

pwLimit- passed from Project2 class, usage limits the number of customers interacting with postal workers. Value: 3

taskDone – to stop the thread execution of customer till postal worker finishes task. Value: 0

**Functions:**

enterPostOffice ()– acquires poLimit semaphore.

assignTask() – randomly assigns task to customer.

waitForIdlePostalWorker ()– after acquiring pwLimit semaphore, tries to acquire mutual exclusion on a postal worker, calls idlePostalWorkerFound() in postal worker class.

waitForTaskToBeDone()- waits till assisting postal worker finishes task , acquires taskDone semaphore.

exitPostOffice() – customer exists post office by releasing poLimit semaphore

taskDone()- used by postal worker to release pwLimit and taskDone semaphores. Also, calls postalWorkerIdle() in postal worker class to release mutual exclusion of the worker.

**Psuedocode:**

enterPostOffice()

assignTask()

waitForIdlePostalWorker()

waitForTaskToBeDone()

exitPostOffice()

**PostalWorker:**

**Semaphores:**

idlePostalWorker – used for mutual exclusion of postal worker by customer. Value: 1

waitingForCustomer- used to hold execution till customer is assigned to postal worker. Value: 0

Scale – use to emulate scale resource for mutual exclusion by postal worker. Value: 1

**Functions:**

waitingForCustomer() – waits till customer is assigned . Acquires waitingForCustomer

assistingCustomer(): starts assisting customer on task, calls doTask( int task ) to do specific task based on task variable defined in customer class.

doTask(int task) – contains switch function that calls specific task function based on task value

stopWaitingForCustomer() – called by customer to start postal worker thread flow. Releases waitingForCustomer.

idlePostalWorkerFound() – called by customer to find idle postal worker. Returns true when idlePostalWorker is acquired.

buyStamps() – makes thread sleep for task time defined for buying stamps

mailLetter() – makes thread sleep for task time defined for mailing a letter.

mailPackage() – acquires scale semaphore, makes thread sleep for task time defined for mailing a package and releases scale.

**Pseudocode:**

While(true)

waitingForCustomer()

assistingCurrentCustomer()