Name: Loc Le
CS333 Project 3

Main.h

```
1 header Main
2
3   uses System, Thread, Synch
4
5   functions
6     main ()
7     SleepingBarber ()
8     GetHairCut (p: int)
9     GamblingParlor ()
10     LetsPlay(p: int)
11
12   class BarberMonitor
13     superclass Object
14     fields
15       customers: Semaphore -- Semaphore for the customer
16       barbers: Semaphore  -- Semaphore for the barber
17       mutex: Mutex -- Mutex lock
18
19       waiting: int  -- Amount of customers waiting
20       chairs: int   -- Amount of chairs in the shop (5 chairs)
21       barberStatus: int -- Status for barber
22       status: array [10] of int      -- Array of statuses for the 10 customers
23
24     methods
25       Init()  -- Initialize and create all variables
26       Barber(p: int) -- steps of barber at hairshop
27       Customer(p: int)       -- steps of customers at hairshop
28       PrintCustomerStatus(p: int)
29       PrintBarberStatus()
30       PrintChairs()
31   endClass
32
33      class GameMonitor
34     superclass Object
35     fields
36       mutex2: Mutex -- Mutex to lock
37       front: Condition -- First group in the line
38       restOfLine: Condition -- Rest of the groups in line
39       diceAvailable: int  -- Tells how many dice are available
40       groupsWaiting: int -- Tells how many groups are waiting
41     methods
42       Init()
43       Request (numberNeeded: int)  -- A group requests a the amount of dice they need for
their game
44       Return (numberReturned: int) -- A group returns the dice after they have finished th
eir game
45       Print (str: String, count: int) -- Print statement for the program
46   endClass
47
48 endHeader
```

Main.c

```
1 code Main
2
3   -- OS Class: Project 3
4   --
    -- <PUT YOUR NAME HERE>
5   -- <Loc Le>
6   --
7
8 -----------------------------  Main  ---------------------------------
9
10   function main ()
       FatalError ("Need to implement")
11       print ("Thread-based Programs Starts...\n")
12
13       InitializeScheduler ()
```

```
14
15        -----  Uncomment any one of the following to perform the desired test  -----
16
17          SleepingBarber ()
18         -- GamblingParlor()
19
20        ThreadFinish ()
21      endFunction
22
23
24  ----------------------  Sleeping Barber  --------------------------------
25
26    enum BLANK, ENTER, SITTING, BEGIN, FINISH, LEAVE, START, END
27    var
28      monitor: BarberMonitor
29      customer: array [10] of Thread = new array of Thread {10 of new Thread}
30
31    function SleepingBarber ()
32      var
33        i: int
34        print ("       ")
35        print ("Barber    ")
36        print ("     1")
37        print ("     2")
38        print ("     3")
39        print ("     4")
40        print ("     5")
41        print ("     6")
42        print ("     7")
43        print ("     8")
44        print ("     9")
45        print ("     10")
46        nl()
47
48        monitor = new BarberMonitor
49        monitor.Init()
50
51        for i = 0 to 9
52          customer[i].Init("i")
53          customer[i].Fork(GetHairCut, i)
54        endFor
55    endFunction
56
57    function GetHairCut (p: int)
58      monitor.Customer(p)
59      monitor.Barber(p)
60    endFunction
61
62    behavior BarberMonitor
63
64    method Init ()
65      var
66        p: int
67      -- Create and initialize variables
68      waiting = 0
69      chairs = 5
70      barberStatus = BLANK
71      status = new array of int { 10 of 0}
72      mutex = new Mutex
73      mutex.Init()
74
75      customers = new Semaphore
76      customers.Init(0)
77
78      barbers = new Semaphore
79      barbers.Init(0)
80
81      for p = 0 to 9
82        status[p] = BLANK  -- Customers havenot entered to the hairshop
83      endFor
84    endMethod
85
86    method Barber (p: int)
87      var
88        i: int
89      while true
90        customers.Down() -- Go to sleep if the amount of customers is 0
91        mutex.Lock()  -- Lock the mutex
92        waiting = waiting - 1  -- Decrement the amount of waiting customers
```

```
93
94        barberStatus = START -- barber starts to do the haircut
95        monitor.PrintBarberStatus() -- print status for barber
96        status[p] = BEGIN -- Customers haircut begins
97        monitor.PrintCustomerStatus (p) -- Print the new status change
98
99        -- Barber is performing the haircut
100        for i = 0 to 100      -- Busy loop for haircut
101          currentThread.Yield()
102        endFor
103
104        status[p] = FINISH  -- Customer haircut finishes
105        monitor.PrintCustomerStatus (p)   -- Print the new status change
106
107        barberStatus = END  -- Barber finishes the haircut
108        monitor.PrintBarberStatus() -- Print out barber status
109
110        barbers.Up()  -- Barber is available again
111        mutex.Unlock() -- Unlock the mutex so customers can leaves
112
113        mutex.Lock() --  Lock mutex for print status
114        status[p] = LEAVE -- Customer's happy, and leaves the shop
115        monitor.PrintCustomerStatus(p)    -- Print new status for customers
116        mutex.Unlock()        -- Unlock the mutex
117
118      endWhile
119    endMethod
120
121    method Customer (p: int)
122      mutex.Lock()
123      status[p] = ENTER  -- Customer enters the barbershop
124      monitor.PrintCustomerStatus (p)                 -- Print the new status change
125      if waiting < chairs     -- Check to see if there are open chairs
126        waiting = waiting + 1 -- Increment the counter of waiting customers
127        status[p] = SITTING  -- Customer sits in the waiting chair
128        monitor.PrintCustomerStatus (p)       -- Print the new status change
129        customers.Up()  -- Customer sit and being served / Wakes up the barber if necessary
130        mutex.Unlock()  -- Unlock the mutex
131
132        if waiting == 1       -- First customer, wake up the barber
133          barbers.Down()  -- Keeps tracks of number of free barbers / sleeps if no customers
134        endIf
135
136      else
137        status[p] = LEAVE                                  -- Customer leaves because waiting chairs are full
138        monitor.PrintCustomerStatus (p)       -- Print the new status change
139        mutex.Unlock()                                                          -- Unlock the mutex
140      endIf
141     endMethod
142
143    -- Print chairs status in the shop. Each chairs taken print x, otherwise print
144    method PrintChairs()
145      var
146        i: int
147        if waiting > 0
148          for i = 1 to waiting
149            print("x")
150          endFor
151          for i = 1 to (5 - waiting)
152            print("-")
153          endFor
154        else
155          print("-----")
156        endIf
157    endMethod
158
159      method PrintCustomerStatus (p: int)
160          var
161            p1: int
162        monitor.PrintChairs()
163        print("                 ")
164          for p1 = 1 to p
165          print("      ")
166        endFor
167        switch status [p]
168           case BLANK:
169             print ("      ")
170             break
171                           case ENTER:
```

```
            print ("E     ")
                                break
          case SITTING:
            print ("S     ")
                                break
          case BEGIN:
            print ("B     ")
                                break
          case FINISH:
            print ("F     ")
                                break
          case LEAVE:
            print ("L     ")
                                break
        endSwitch
          nl()
        endMethod

      method PrintBarberStatus ()
          monitor.PrintChairs()
          switch barberStatus
            case BLANK:
              print ("        ")
              break
            case START:
              print ("  start     ")
              break
            case END:
              print ("  end     ")
              break
          endSwitch
        nl()
      endMethod
      endBehavior

------------------------ Gambling Parlor --------------------------------
var
      monitor2: GameMonitor
      ThreadArray: array [8] of Thread = new array of Thread {8 of new Thread}

    function GamblingParlor ()

        monitor2 = new GameMonitor
        monitor2.Init()

        ThreadArray[0].Init ("A")
        ThreadArray[0].Fork (LetsPlay, 4)

        ThreadArray[1].Init ("B")
        ThreadArray[1].Fork (LetsPlay, 4)

        ThreadArray[2].Init ("C")
        ThreadArray[2].Fork (LetsPlay, 5)

        ThreadArray[3].Init ("D")
        ThreadArray[3].Fork (LetsPlay, 5)

        ThreadArray[4].Init ("E")
        ThreadArray[4].Fork (LetsPlay, 2)

        ThreadArray[5].Init ("F")
        ThreadArray[5].Fork (LetsPlay, 2)

        ThreadArray[6].Init ("G")
        ThreadArray[6].Fork (LetsPlay, 1)

        ThreadArray[7].Init ("H")
        ThreadArray[7].Fork (LetsPlay, 1)
    endFunction

    function LetsPlay(p: int)
      var
        i: int
        for i = 0 to 4
          monitor2.Request(p)
          currentThread.Yield()
          monitor2.Return(p)
          currentThread.Yield()
        endFor
```

```
  endFunction

  behavior GameMonitor
    -- Creates and initializes all the variables used
    method Init()

      -- Create and initialize variables
      mutex2 = new Mutex
      mutex2.Init()

      front = new Condition
      front.Init()

      restOfLine = new Condition
      restOfLine.Init()

      diceAvailable = 8
      groupsWaiting = 0
    endMethod

    method Request(numberNeeded: int)
      mutex2.Lock()  -- Lock the mutex
      self.Print ("requests", numberNeeded) -- Print the amount of dice needed
      groupsWaiting = groupsWaiting + 1 -- Increase number of group waiting for dice

      if groupsWaiting > 1   -- There is a line, wait in the line
        restOfLine.Wait (&mutex2)  -- Unlock mutex and wait for signal
      endIf
      while diceAvailable < numberNeeded -- at front of list, wait for dice
        front.Wait(&mutex2)  -- Unlock mutex and wait for signal
      endWhile

      diceAvailable = diceAvailable - numberNeeded
      groupsWaiting = groupsWaiting - 1
      restOfLine.Signal (&mutex2)
      self.Print("proceeds with", numberNeeded) -- print out how many lines needed
      mutex2.Unlock ()  -- Unlock the mutex
    endMethod

    method Return (numberReturned: int)
      mutex2.Lock ()  -- Lock the mutex
      diceAvailable = diceAvailable + numberReturned
      self.Print ("releases and adds back", numberReturned) -- amount of dice available
      front.Signal (&mutex2)  -- Wakeup the first group in line, if they exist
      mutex2.Unlock()
    endMethod

    method Print (str: String, count: int)
      print (currentThread.name)
      print ("  ")
      print (str)
      print ("  ")
      printInt (count)
      nl ()
      print ("------------------------------Number of dice now avail = ")
      printInt (diceAvailable)
      nl ()
    endMethod
  endBehavior
endCode
```

sleeping barber.txt

```
Beginning execution...
====================  KPL PROGRAM STARTING  ====================
Thread-based Programs Starts...
Initializing Thread Scheduler...
       Barber         1     2     3     4     5     6     7     8     9     10
-----                 E
x----                 S
x----                       E
xx---                       S
xx---                             E
xxx--                             S
xx---  start
xx---                        B
xx---                        F
xx---  end
```

```
16 xx---                                  E
17 xxx--                                  S
18 xxx--                                        E
19 xxxx-                                        S
20 xxxx-                                              E
21 xxxxx                                              S
22 xxxxx                                                    E
23 xxxxx                                                    L
24 xxxxx                                                          E
25 xxxxx                                                          L
26 xxxxx                                                                E
27 xxxxx                                                                L
28 xxxxx                                                                      E
29 xxxxx                                                                      L
30 xxxx-  start
31 xxxx-                              B
32 xxxx-                              F
33 xxxx-  end
34 xxxx-                        L
35 xxx--  start
36 xxx--                                  B
37 xxx--                                  F
38 xxx--  end
39 xx---  start
40 xx---                                        B
41 xx---                                        F
42 xx---  end
43 x----  start
44 x----                B
45 x----                F
46 x----  end
47 -----  start
48 -----                                              B
49 -----                                              F
50 -----  end
51 -----                              L
52 -----                                    L
53 -----                                          L
54 -----                B
55 -----                                              L
56
57 *****  A 'wait' instruction was executed and no more interrupts are scheduled... halting e
mulation!  *****
58
59 Done!  The next instruction to execute will be:
60 000EC8: 09000000       ret
61 Number of Disk Reads    = 0
62 Number of Disk Writes   = 0
63 Instructions Executed   = 1399952
64 Time Spent Sleeping     = 0
65     Total Elapsed Time  = 1399952
```

gaming parlor.txt

```
1 Beginning execution...
2 ====================  KPL PROGRAM STARTING  ====================
3 Thread-based Programs Starts...
4 Initializing Thread Scheduler...
5 A  requests  4
6 ------------------------------Number of dice now avail = 8
7 A  proceeds with  4
8 ------------------------------Number of dice now avail = 4
9 B  requests  4
10 ------------------------------Number of dice now avail = 4
11 B  proceeds with  4
12 ------------------------------Number of dice now avail = 0
13 D  requests  5
14 ------------------------------Number of dice now avail = 0
15 A  releases and adds back  4
16 ------------------------------Number of dice now avail = 4
17 B  releases and adds back  4
18 ------------------------------Number of dice now avail = 8
19 E  requests  2
20 ------------------------------Number of dice now avail = 8
21 F  requests  2
22 ------------------------------Number of dice now avail = 8
23 G  requests  1
24 ------------------------------Number of dice now avail = 8
```

```
D  proceeds with  5
------------------------------Number of dice now avail = 3
A  requests  4
------------------------------Number of dice now avail = 3
C  requests  5
------------------------------Number of dice now avail = 3
E  proceeds with  2
------------------------------Number of dice now avail = 1
D  releases and adds back  5
------------------------------Number of dice now avail = 6
B  requests  4
------------------------------Number of dice now avail = 6
H  requests  1
------------------------------Number of dice now avail = 6
F  proceeds with  2
------------------------------Number of dice now avail = 4
D  requests  5
------------------------------Number of dice now avail = 4
E  releases and adds back  2
------------------------------Number of dice now avail = 6
G  proceeds with  1
------------------------------Number of dice now avail = 5
F  releases and adds back  2
------------------------------Number of dice now avail = 7
E  requests  2
------------------------------Number of dice now avail = 7
A  proceeds with  4
------------------------------Number of dice now avail = 3
G  releases and adds back  1
------------------------------Number of dice now avail = 4
F  requests  2
------------------------------Number of dice now avail = 4
A  releases and adds back  4
------------------------------Number of dice now avail = 8
G  requests  1
------------------------------Number of dice now avail = 8
C  proceeds with  5
------------------------------Number of dice now avail = 3
A  requests  4
------------------------------Number of dice now avail = 3
C  releases and adds back  5
------------------------------Number of dice now avail = 8
B  proceeds with  4
------------------------------Number of dice now avail = 4
C  requests  5
------------------------------Number of dice now avail = 4
H  proceeds with  1
------------------------------Number of dice now avail = 3
B  releases and adds back  4
------------------------------Number of dice now avail = 7
D  proceeds with  5
------------------------------Number of dice now avail = 2
H  releases and adds back  1
------------------------------Number of dice now avail = 3
B  requests  4
------------------------------Number of dice now avail = 3
E  proceeds with  2
------------------------------Number of dice now avail = 1
D  releases and adds back  5
------------------------------Number of dice now avail = 6
H  requests  1
------------------------------Number of dice now avail = 6
F  proceeds with  2
------------------------------Number of dice now avail = 4
E  releases and adds back  2
------------------------------Number of dice now avail = 6
D  requests  5
------------------------------Number of dice now avail = 6
G  proceeds with  1
------------------------------Number of dice now avail = 5
E  requests  2
------------------------------Number of dice now avail = 5
F  releases and adds back  2
------------------------------Number of dice now avail = 7
A  proceeds with  4
------------------------------Number of dice now avail = 3
G  releases and adds back  1
------------------------------Number of dice now avail = 4
F  requests  2
```

```
104 ------------------------------Number of dice now avail = 4
105 G  requests  1
106 ------------------------------Number of dice now avail = 4
107 A  releases and adds back  4
108 ------------------------------Number of dice now avail = 8
109 C  proceeds with  5
110 ------------------------------Number of dice now avail = 3
111 A  requests  4
112 ------------------------------Number of dice now avail = 3
113 C  releases and adds back  5
114 ------------------------------Number of dice now avail = 8
115 B  proceeds with  4
116 ------------------------------Number of dice now avail = 4
117 H  proceeds with  1
118 ------------------------------Number of dice now avail = 3
119 B  releases and adds back  4
120 ------------------------------Number of dice now avail = 7
121 D  proceeds with  5
122 ------------------------------Number of dice now avail = 2
123 H  releases and adds back  1
124 ------------------------------Number of dice now avail = 3
125 C  requests  5
126 ------------------------------Number of dice now avail = 3
127 E  proceeds with  2
128 ------------------------------Number of dice now avail = 1
129 H  requests  1
130 ------------------------------Number of dice now avail = 1
131 B  requests  4
132 ------------------------------Number of dice now avail = 1
133 E  releases and adds back  2
134 ------------------------------Number of dice now avail = 3
135 D  releases and adds back  5
136 ------------------------------Number of dice now avail = 8
137 F  proceeds with  2
138 ------------------------------Number of dice now avail = 6
139 E  requests  2
140 ------------------------------Number of dice now avail = 6
141 D  requests  5
142 ------------------------------Number of dice now avail = 6
143 G  proceeds with  1
144 ------------------------------Number of dice now avail = 5
145 F  releases and adds back  2
146 ------------------------------Number of dice now avail = 7
147 A  proceeds with  4
148 ------------------------------Number of dice now avail = 3
149 G  releases and adds back  1
150 ------------------------------Number of dice now avail = 4
151 F  requests  2
152 ------------------------------Number of dice now avail = 4
153 A  releases and adds back  4
154 ------------------------------Number of dice now avail = 8
155 G  requests  1
156 ------------------------------Number of dice now avail = 8
157 C  proceeds with  5
158 ------------------------------Number of dice now avail = 3
159 A  requests  4
160 ------------------------------Number of dice now avail = 3
161 H  proceeds with  1
162 ------------------------------Number of dice now avail = 2
163 C  releases and adds back  5
164 ------------------------------Number of dice now avail = 7
165 B  proceeds with  4
166 ------------------------------Number of dice now avail = 3
167 H  releases and adds back  1
168 ------------------------------Number of dice now avail = 4
169 C  requests  5
170 ------------------------------Number of dice now avail = 4
171 E  proceeds with  2
172 ------------------------------Number of dice now avail = 2
173 B  releases and adds back  4
174 ------------------------------Number of dice now avail = 6
175 H  requests  1
176 ------------------------------Number of dice now avail = 6
177 D  proceeds with  5
178 ------------------------------Number of dice now avail = 1
179 E  releases and adds back  2
180 ------------------------------Number of dice now avail = 3
181 B  requests  4
182 ------------------------------Number of dice now avail = 3
```

```
F  proceeds with  2
------------------------------Number of dice now avail = 1
E  requests  2
------------------------------Number of dice now avail = 1
D  releases and adds back  5
------------------------------Number of dice now avail = 6
G  proceeds with  1
------------------------------Number of dice now avail = 5
F  releases and adds back  2
------------------------------Number of dice now avail = 7
D  requests  5
------------------------------Number of dice now avail = 7
A  proceeds with  4
------------------------------Number of dice now avail = 3
F  requests  2
------------------------------Number of dice now avail = 3
G  releases and adds back  1
------------------------------Number of dice now avail = 4
A  releases and adds back  4
------------------------------Number of dice now avail = 8
G  requests  1
------------------------------Number of dice now avail = 8
C  proceeds with  5
------------------------------Number of dice now avail = 3
H  proceeds with  1
------------------------------Number of dice now avail = 2
C  releases and adds back  5
------------------------------Number of dice now avail = 7
B  proceeds with  4
------------------------------Number of dice now avail = 3
H  releases and adds back  1
------------------------------Number of dice now avail = 4
E  proceeds with  2
------------------------------Number of dice now avail = 2
B  releases and adds back  4
------------------------------Number of dice now avail = 6
C  requests  5
------------------------------Number of dice now avail = 6
H  requests  1
------------------------------Number of dice now avail = 6
D  proceeds with  5
------------------------------Number of dice now avail = 1
E  releases and adds back  2
------------------------------Number of dice now avail = 3
F  proceeds with  2
------------------------------Number of dice now avail = 1
D  releases and adds back  5
------------------------------Number of dice now avail = 6
G  proceeds with  1
------------------------------Number of dice now avail = 5
F  releases and adds back  2
------------------------------Number of dice now avail = 7
C  proceeds with  5
------------------------------Number of dice now avail = 2
G  releases and adds back  1
------------------------------Number of dice now avail = 3
H  proceeds with  1
------------------------------Number of dice now avail = 2
C  releases and adds back  5
------------------------------Number of dice now avail = 7
H  releases and adds back  1
------------------------------Number of dice now avail = 8

*****  A 'wait' instruction was executed and no more interrupts are scheduled... halting e
mulation!  *****

Done!  The next instruction to execute will be:
000EC8: 09000000       ret
Number of Disk Reads    = 0
Number of Disk Writes   = 0
Instructions Executed   = 522830
Time Spent Sleeping     = 0
    Total Elapsed Time  = 522830
```