Guest:

* 25 guest visit hotel
* Each guest has rand num of bags [0-5]
* Each must check in front desk
* Gets room number from front desk employee
* If guest has > 2 bags, give to bellhop
* Guest enters assigned room
* Receives bag from bellhop and give tip
* Retires for evening

Front desk:

* Two employees at the front desk (1 thread each)
* Check in guest
* Finds available room
* Gives room number to guest

Bellhop:

* 2 bellhops
* Gets bag from guest
* Same bellhop that took bag delivers bag **after** guest is in room
* Takes tip

Semaphores needed:

* Max\_guest = 25
  + Max guest
* Front\_desk = 2
  + Num of employees at front
* Bellhops = 2
  + Num of bellhops
* Rooms = 25
  + Max number of rooms given max number of guest
* checkIn = 2
  + Max number of guest that can check in at front desk is 2 bc of front desk employees
* Bag\_and\_tip = 2
  + Checks for everytime the bellhop needs to take bags to the guest room and then will receive money

Threads:

* Guest[1-25]
* Front\_desk[1-2]
* Bellhop[1-2]

Front\_desk function:

Be created

Takes one guest per front\_desk that are available (Max = 2)

Assigns a room to the guest to be sent to

Checks if the guest has > 2 bags, if so call the bellhop to take it, otherwise go to room

Bellhop:

Be created

Waits to be called by guest to take bags

When called by guest, takes bag to the room

Waits for guest to receive bag and also take tip

Guest:

Be created

Enters with # of bags (rand)

Waits to go to front desk to register for a room

Assigned a room

If bags are > 2, then request for help from bellhop

Go to room

If bags > 2, get bags from bellhop and tip

Retire for evening

| Initialize semaphores:  Front\_desk\_employee = 2  Bellhop = 2  guestSemaphore = 0  maxGuest = 25 |  |  |
| --- | --- | --- |
| **Void** guest (){  wait(maxGuest);  guestCreated();  FrontDeskCreated();  bellhopCreated();  guestEnter();  wait(room);  goToRoom();  if(bags > 2){ wait(bellhopArrive);  signal(tip);  rest();  signal(maxGuest)  } | **Void** frontDesk(){  while(true){  wait(guestEnter);  signal(frontDesk);  assignRoom();  signal(room);  If (bags > 2){signal(bellHop);}  }  } | **Void** bellHop(){  while(true){  wait(bags);  findsGuest();  signal(bellhopArrive();)  wait(tip);  }  } |