List of semaphores:

Initial values for all semaphores are 0

start:

This semaphore is only used in the main function. Main() waits on start every time it creates a new thread and each thread signals start after its creation message. This semaphore keeps creation messages from overlapping and keeps threads from losing their thread id’s when threads are created too fast.

All of the following semaphores I split into two identical counterparts because I was unable to solve problems with guests choosing the same employee/bellhop when getting out of line around the same time. My solution was to split the interactions so guests only had a chance to interact with one of the employees/bellhops.

deskLine1/deskLine2:

Guests wait on this semaphore if there is not an available employee at the check in desk. Once a desk employee has finished with a customer it signals the deskLine.

employeeSem1/employeeSem2:

This semaphore is used by an employee while interacting with a customer. It waits when the guest needs to give information to the employee or while waiting on the guest to take information from the employee. The guest signals this when it has given/taken its information.

deskGuestSem1/deskGuestSem2:

This semaphore is used by the guest while interacting with the desk employee. It waits when the employee needs to give information to the guest or while waiting on the employee to take information from the guest. The employee signals this when it has given/taken its information.

bellhopLine1/bellhopLine2:

Guests wait on this semaphore if there is not an available bellhop. Once a bellhop has finished delivering bags it signals the bellhopLine.

bellhopSem1/bellhopSem2:

This semaphore is used by a bellhop while interacting with a customer. It waits when the guest needs to give information/bags to the bellhop or while waiting on the guest to take bags from the bellhop. The guest signals this when it has given/taken its information/bags.

hopGuestSem1/hopGuestSem2:

This semaphore is used by a guest while interacting with a bellhop. It waits when the bellhop needs to give bags to the guest or while waiting on the bellhop to take information/bags from the guest. The bellhop signals this when it has given/taken its information/bags.

Psuedocode:

frontDesk() {

signal(start)

start helping guests at the front desk

if is 0 {

while (true) {

call on next guest in line

signal(deskLine1)

wait(employeeSem1)

take guest number

assign guest to a room and give them their key

wait for guest to take key and leave

signal(deskGuestSem1)

wait(employeeSem1)

}

}

else {

while (true) {

call on next guest in line

signal(deskLine2)

wait(employeeSem2)

take guest number

assign guest to the next empty room and give them their key

wait for guest to take key and leave

signal(deskGuestSem2)

wait(employeeSem2)

}

}

}

bellhop() {

signal(start)

start helping guests with bags

if id is 0 {

while (true) {

call on next guest in line

signal(bellhopLine1)

wait(bellhopSem1)

get guest number and bags from guest

go to guest’s room

signal(hopGuestSem1)

wait(bellhopSem1)

give bags to guest

signal(hopGuestSem1)

wait(bellhopSem1)

return to lobby

}

}

else {

while (true) {

call on next guest in line

signal(bellhopLine2)

wait(bellhopSem2)

get guest number and bags from guest

go to guest’s room

signal(hopGuestSem2)

wait(bellhopSem2)

give bags to guest

signal(hopGuestSem2)

wait(bellhopSem2)

return to lobby

}

}

}

guest() {

signal(start)

enter hotel and wait in a line for the front desk

if id is even

wait(deskLine1)

go to desk and give guest number to employee

signal(employeeSem1)

wait(deskGuestSem1)

take room key from employee

signal(employeeSem1)

}

else

wait(deskLine2)

go to desk and give guest number to employee

signal(employeeSem2)

wait(deskGuestSem2)

take room key from employee

signal(employeeSem2)

}

If more than 2 bags {

wait in line for a bellhop

If id is even {

wait(bellhopLine1)

go to bellhop and give bellhop guest number

wait for bellhop to take bags

signal(bellhopSem1)

wait(hopGuestSem1)

}

else {

wait(bellhopLine2)

go to bellhop and give bellhop guest number

wait for bellhop to take bags

signal(bellhopSem2)

wait(hopGuestSem2)

}

}

go to room

If more than 2 bags {

If id is even {

wait for bellhop

signal(bellhopSem1)

wait(hopGuestSem1)

take bags from bellhop and give them a tip

signal(bellhopSem1)

}

else {

wait for bellhop

signal(bellhopSem2)

wait(hopGuestSem2)

take bags from bellhop and give them a tip

signal(bellhopSem2)

}

}

retire for the evening

tell main() that this thread is ready to join