Project 2: Threads and Semaphores Design

|  |  |  |
| --- | --- | --- |
| Semaphore | Initial Value | Purpose |
| customer\_total | 50 | Number of customers is 50 |
| customer\_capacity | 10 | Only 10 customers can be inside the Post Office at a time. |
| worker | 3 | 3 postal workers, one thread each, check if any postal worker is available |
| mutex | 1 | Enqueue and Dequeue customer and task number |
| scale | 1 | Define the only one scale in postal office |
| cust\_ready | 0 | Boolean value, |
| leave\_PostOffice | 0 | Boolean value, |
| finish[50] | 0 | To indicate which customer’s task has been finished |

Void Worker( )

{

int cust\_num

int cust\_order;

while(true)

{

Semwait(customer\_ready);

Semwait(mutex);

dequeue1[cust\_num];

dequeue2[cust\_order];

Semsignal(mutex);

serve task;

if(mail a package)

{

wait(scale)

mail a package

signal(scale)

}

Semsignal(finish[cust\_num]);

Semwait(leave\_PostOffice);

signal(worker);

}

}

Void Customer( )

{

int cust\_number;

int cust\_order;

Semwait(max\_capacity);

Enter\_PostOffice;

Semwait(worker);

Semwait(mutex);

enqueue1(cust\_number);

enqueue2(cust\_order);

Semsignal(customer\_ready);

Semsignal(mutex);

wait(finish[customer]);

signal(leave\_PostOffice);

leave post office;

signal(customer\_capacity);

}

Void main()

{

creat PostOffice constructor;

run PostOffice

}