CSCI402 Lecture September 9, 2010

Analysis for Customer-OrderTaker Interaction)

- 1 line for Customers waiting to give order
- Multiple Customers
- 1 or more OrderTakers
- Cust wait in line if no available OrderTakers

Data

○ Need monitor variable to track state of OrderTakers

```
Condition* custWaitingCV int custLineLength //monitor
variable OT can check to see if he has to signal next
thread)
Lock* custLineLock //prompt user for maxNumOrderTakers
(don't want to make this constant)
int orderTakerStatus[maxNumOrderTakers] //1st monitor
variable customer checks when he enters the restaurant
- initialize to "BUSY" (1=BUSY, 0=FREE) (or use an enum**)
int myOrderTaker = -1; //need to index your orderTaker
– initialize all variablefirst, THEN make threads
```

Customer Pseudocode

```
custLineLock->Acquire();
for(maxNumOrderTakers) {
  if(orderTakerStatus[i]==0) {
    OrderTakerStatus[i]=1;
    myOrderTaker=i;
   }
}
if(myOrderTaker == -1) { //no one's available, get in line
   custLineLength++;
   custWaitingCV->Wait(custLineLock); //wait for OT to be
available
}
/*potential problem: if OT changes status to available, I
go to end of ready queue, while a customer who just started
might search, find available OT, and take him without
waiting in line→  not fair! We need another status*/
//get a "waiting" OT w/ different value, i.e. 2
for(maxNumOrderTakers) {
   if(orderTakerStatus[i]==2) { //note: if this condition
    fails, we know it must be a manager doing the order
    taking
    myOrderTaker = i;
   }
   orderTakerStatus[i] = 1;
}
//custLineLength--; //no longer decrementing here
//@ this point, customer is DONE waiting in line and is
```

```
ready to give order
custLineLock->Release(); //time to play some tennis
OrderTakerLock[myOrderTaker]->Acquire(); //give ordertaker
my order
OrderTakerCV[myOrderTaker]>Signal(orderTakerLock[myOrderTak
er]); //monitor variable for each OrderTaker, index
myOrderTaker
orderTakerCV[myOrderTaker]->Wait();
```

OrderTaker Pseudocode

```
void OrderTaker(int myId); //how we're keeping track of
diff OTs
while(true) {
   custLineLock->Acquire();
   if(custLineLength > 0) {
    custWaitingCV->Signal(custLineLock);
    custLineLength--; //now let the OrderTakers decrement
the line length
    orderTakerStatus[myId] = 2;
   else if(/*food to bag*/) {
    custLineLock->Release(); //bag 1 order at a time
continue;
   } else {
   //Nothing to do
    orderTakerStatus[myIndex]=0;
   }
}
//Now I need to wait for a customer
orderTakerLock[myId]->Acquire(); //array of locks –
guarantees I have control of the monitor variable before
the customer does
custLineLock->Release();
orderTakerCV[myId]->Wait(orderTakerLock[myId]);
//Now process the order..}
```