

```

1 class Process:
2     def __init__(self, chanID, procID, procIDSet):
3         self.chan.join(procID)
4         self.procID = int(procID)
5         self.otherProcs.remove(self.procID)
6         self.queue = [] # The request queue
7         self.clock = 0 # The current logical clock
8
9     def requestToEnter(self):
10         self.clock = self.clock + 1 # Increment clock value
11         self.queue.append((self.clock, self.procID, ENTER)) # Append request to q
12         self.cleanupQ() # Sort the queue
13         self.chan.sendTo(self.otherProcs, (self.clock, self.procID, ENTER)) # Send request
14
15     def ackToEnter(self, requester):
16         self.clock = self.clock + 1 # Increment clock value
17         self.chan.sendTo(requester, (self.clock, self.procID, ACK)) # Permit other
18
19     def release(self):
20         tmp = [r for r in self.queue[1:] if r[2] == ENTER] # Remove all ACKs
21         self.queue = tmp # and copy to new queue
22         self.clock = self.clock + 1 # Increment clock value
23         self.chan.sendTo(self.otherProcs, (self.clock, self.procID, RELEASE)) # Release
24
25     def allowedToEnter(self):
26         commProcs = set([req[1] for req in self.queue[1:]]) # See who has sent a message
27         return (self.queue[0][1] == self.procID and len(self.otherProcs) == len(commProcs))

```