

```

1 void scheduleArrival(){
2     // add a process to the process list
3     process* pIt = pHead;
4     while( pIt->pNext !=0 ){
5         pIt = pIt->pNext;
6     }
7     pIt->pNext = new process;
8     pIt->pNext->arrivalTime = pIt->arrivalTime + genExp((float)lambda);
9     pIt->pNext->startTime = 0.0;
10    pIt->pNext->reStartTime = 0.0;
11    pIt->pNext->finishTime = 0.0;
12    pIt->pNext->serviceTime = genExp(mu);
13    pIt->pNext->remainingTime = pIt->pNext->serviceTime;
14    pIt->pNext->pNext = 0;
15
16    // create a corresponding arrival event
17    eventQueue* nuArrival = new eventQueue;
18    nuArrival->time = pIt->pNext->arrivalTime;
19    nuArrival->type = 1;
20    nuArrival->pLink = pIt->pNext;
21    nuArrival->eNext = 0;
22
23    // insert into eventQ in asc time order
24    insertIntoEventQ(nuArrival);
25 }
26 void scheduleAllocation(){
27     // create a new event queue node
28     eventQueue* nuAllocation = new eventQueue;
29
30     // identify the next process to be allocated to the cpu:
31     process* nextProc;
32     if( schedulerType == 1 ) nextProc = rHead->pLink; // FCFS
33     else if( schedulerType == 2 ){ // SRTF
34         if( cpuHead->clock > rHead->pLink->arrivalTime ){
35             nextProc = getSRTFProcess();
36         }
37         else{
38             nextProc = rHead->pLink;
39         }
40     }
41     else if( schedulerType == 3 ){ // HRRN
42         nextProc = getHRRNProcess();
43     }
44
45     // set the time of the allocation event
46     if( cpuHead->clock < nextProc->arrivalTime ){
47         nuAllocation->time = nextProc->arrivalTime;
48     }
49     else{
50         nuAllocation->time = cpuHead->clock;
51     }
52
53     // set the values for type, next, and pLink
54     nuAllocation->type = 3;
55     nuAllocation->eNext = 0;
56     nuAllocation->pLink = nextProc;
57
58     // insert new event into eventQ
59     insertIntoEventQ( nuAllocation );
60 }
61 void scheduleDeparture(){
62     // create a new event node for the departure event
63     eventQueue* nuDeparture = new eventQueue;
64     nuDeparture->type = 2;
65     nuDeparture->eNext = 0;
66     nuDeparture->pLink = cpuHead->pLink;

```

```

67
68 // set the departure time for the event
69 if( schedulerType == 1 || // FCFS
70     schedulerType == 3 ){ // HRRN
71     nuDeparture->time =
72         cpuHead->pLink->startTime +
73         cpuHead->pLink->remainingTime;
74 }
75 else if( schedulerType == 2 ){ // SRTF
76     if( cpuHead->pLink->reStartTime == 0 ){
77         nuDeparture->time =
78             cpuHead->pLink->startTime +
79             cpuHead->pLink->remainingTime;
80     }
81     else{
82         nuDeparture->time =
83             cpuHead->pLink->reStartTime +
84             cpuHead->pLink->remainingTime;
85     }
86 }
87
88 // insert the new event into eventQ in asc time order
89 insertIntoEventQ(nuDeparture);
90 }
91
92 void schedulePreemption(){
93     eventQueue* nuPreemption = new eventQueue;
94     nuPreemption->time = eHead->time;
95     nuPreemption->type = 4;
96     nuPreemption->eNext = 0;
97     nuPreemption->pLink = eHead->pLink;
98
99     // pop the arrival event from eHead so that
100     // it can be replaced by the preemption event
101     popEventQHead();
102
103     // insert new event into eventQ
104     insertIntoEventQ( nuPreemption );
105 }
106 void insertIntoEventQ( eventQueue* nuEvent ){
107     // put the new event in the readyQ, sorted by time
108     if( eHead == 0 ) eHead = nuEvent;
109     else if( eHead->time > nuEvent->time ){
110         nuEvent->eNext = eHead;
111         eHead = nuEvent;
112     }
113     else{
114         eventQueue* eIt = eHead;
115         while( eIt != 0 ){
116             if( (eIt->time < nuEvent->time) && (eIt->eNext == 0) ){
117                 eIt->eNext = nuEvent;
118                 break;
119             }
120             else if( (eIt->time < nuEvent->time) &&
121                     (eIt->eNext->time > nuEvent->time)){
122                 nuEvent->eNext = eIt->eNext;
123                 eIt->eNext = nuEvent;
124                 break;
125             }
126             else{
127                 eIt = eIt->eNext;
128             }
129         }
130     }
131 }

```