Module: ReadyQueue

Uses: ALL SCREENS

Extends: Activity

INTERFACE:

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Meaning |
| waiting | int | The number of threads waiting in the Queue |
| threadQueue | Thread[5] | The queue of processes waiting to be executed. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Method Name | Input | Output | Description | State Transitions | Dependencies |
| ReadyQueue | void | ReadyQueue | Constructor | waiting=0  threadQueue=new Process[5] |  |
| isFull() |  | boolean | Returns a boolean representing whether the queue is full or not |  |  |
| enqueue(Process) | A Process representing the new thread to be put into the threadQueue | void | blah a | threadQueue[n]=Process, where n is the first entry in the queue that is null |  |
| dequeue() | void | void | Removes a Process from the Queue | for (int i=0;i<threadQueue.length-1;i++){  threadQueue[i]=threadQueue[i+1]  }  threadQueue[threadQueue.length-1]=null |  |
| backInQueue | void | void | Puts the first item in the threadQueue in the last available spot | Process placeHolder=threadQueue[0];  dequeue();  enqueue(placeHolder) | dequeue()  enqueue() |

IMPLEMENTATION

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Meaning |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Method Name | Input | Output | Description | State Transitions | Dependencies |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Module: Process

Uses: ALL SCREENS

Extends: Thread

INTERFACE:

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Meaning |
| executionTime | long | represents how many clock cycles of work it has left |
| id | int | Represents the id number of the Process |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Method Name | Input | Output | Description | State Transitions | Dependencies |
|  |  |  |  |  |  |
| @Override  run() | void | void | Waits until the execution time has passed or until notified/ |  | wait(integer long) |
| Process | int ID | Process | Constructor  Generates a random number for executionTime and returns the ProcessObject | id=ID  executionTime=Random.nextLong | java.util.Random |
| updateTime(int Long l) | A long integer representing how long the Process ran for (CPU maxRunTime) | integer Long  how long the thread ran |  | int newTime=executionTime-l  int r = executionTime;  if (newTime>0){  executionTime=newTime;  return newTime  }  else{  executionTime=0;  return r;  } |  |
| checkTime() | void | integer Long  returns the executionTime |  |  |  |

IMPLEMENTATION

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Meaning |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Method Name | Input | Output | State Transitions | Dependencies |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Module: Generator

Uses: ALL SCREENS

Extends:

INTERFACE:

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Meaning |
| readyQ | ReadyQueue | represents the ReadyQueue Process |
| scheduler | RRScheduler |  |
| waiting | int | represents how many queues are in the readyQueue |
| processCount | int | represents how many processes have been generated |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Method Name | Input | Output | Description | State Transitions | Dependencies |
| Generator | ReadyQueue readyQ |  | Constructor | this.readyQ=RRscheduler.readyQ  waiting=readyQ.threadQueue.length |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

IMPLEMENTATION

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Meaning |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Method Name | Input | Output | State Transitions | Dependencies |
| enqueue() | void | void | Creates a new Process and adds it to the readyQueue | t=new Process(processCount)  readQ.enqueue(t)  processCount++ | Process constructor |
| checkQueue | void | void | checks the size of the readyQ | waiting=readyQ.readyQueue.length |  |
| generateMessage() |  | none | Generates a message of the form  "Process # loaded into ready Queue) |  |  |

Module: Dispatcher

Uses: ALL SCREENS

Extends:

INTERFACE:

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Meaning |
| readyQ | ReadyQueue | represents the ReadyQueue Process |
| cpu | CPU | represents the CPU process |
|  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Method Name | Input | Output | Description | State Transitions | Dependencies |
| Dispatcher | ReadyQueue readyQ, CPU cpu | DispatcherC | Constructor | this.readyQ=readyQ  this.cpu=cpu |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

IMPLEMENTATION

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Meaning |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Method Name | Input | Output | Description | State Transitions | Dependencies |
| select | void | void | take a Process from the readyQueue and give it to the CPU for processing | q=readyQ.readyQueue[0]  dequeue()  cpu.load(q) | ReadyQueue.dequeue(); |
| generateMessage() |  | none | Generates a message of the form  "Process # loaded into CPU) |  |  |
|  |  |  |  |  |  |

Module: CPU

Uses: ALL SCREENS

Extends:

INTERFACE:

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Meaning |
| maxExecutionTime | int long | represents the max executionTime a Process may run |
| reaper | GrimReaper | represents GrimReaper process |
| loadedThread | Process | The current thread on the CPU |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Method Name | Input | Output | Description | State Transitions | Dependencies |
| CPU | (int long maxExecution, GrimReaper reaper) |  |  | maxExecutionTime=maxExectution  this.reaper=reaper  loadedThread=null; |  |
| load(Process t) | Process representing the thread that was just loaded on the CPU | void | Runs the Process, stopping it if it runs for too long, then updates the remainig Execution time | loadedThread=t;  execute() | this.execute() |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Method Name | Input | Output | Description | State Transitions | Dependencies |
| unload() | Represents a Process being unloaded from the CPU |  | Passes the Process object to the GrimReaper | reaper.checkTime(loadedThread)  loadedThread=null; |  |
| execute() |  |  |  | loadedThread.run()  wait(maxExecutionTime)  loadedThread.notify()  loadedThread(maxExecutionTime) |  |
| generateMessage() |  | none | Generates a message of the form  "Process # executed for x seconds" |  |  |

Module: GrimReaper

Uses: ALL SCREENS

Extends:

INTERFACE:

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Meaning |
| currentThread | Process | Current thread to have its fate decided |
| readyQ | ReadyQueue |  |
|  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Method Name | Input | Output | Description | State Transitions | Dependencies |
| GrimReaper | (ReadyQueue readyQ) | GrimReaper | Constructor | this.readyQ=readyQ  currentThread=null; |  |
| checkTime(Process) |  |  |  | currentThread=Process  judge(); |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Method Name | Input | Output | Description | State Transitions | Dependencies |
| judge() |  |  |  | int timeLeft=currentThread.getExecutionTime()  if(timeLeft>0){  backInQueue(currentThread)  else{  dequeue();  currentThread=null;  } |  |
| OVERLOADED  generateMessage(id,x)  generateMessage(id) | int id  int x  x represents how many seconds a process with id number id has executed  id represents a processID | none | Generates a message of the form either  "Process id finished executing"  "Process id recycled into ready queue with x seconds remaining" |  | Generates a message and prints it  Integer.toString |
|  |  |  |  |  |  |

Module: RRScheduler

Uses: ALL SCREENS

Extends:

INTERFACE:

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Meaning |
| readyQueue | ReadyQueue |  |
| generator | Generator |  |
| dispatcher | Dispatcher |  |
| cpu | CPU |  |
| grimReaper | GrimReaper |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Method Name | Input | Output | Description | State Transitions | Dependencies |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

IMPLEMENTATION

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Meaning |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Method Name | Input | Output | Description | State Transitions | Dependencies |
| run() |  | Runs all the processes! | Runs all the processes! |  | Generator  GrimReaper  Dispatcher  ReadyQueue  CPU |
|  |  |  |  |  |  |
|  |  |  |  |  |  |