OPERATING SYSTEMS – COMP 3318 – FALL 2016

Homework #3

Due on November 02, 11:59pm

Geraldo Braho

Please follow the instructions below. Answer the questions with supporting documentation Q1-4 and submit your response through moodle.

(1) In the compiler window, enter these two following source codes:

```
program DeadlockP1
        resource(1, allocate)
        wait(3)
        resource(2, allocate)
        for n = 1 to 3
        next
end
```

```
program DeadlockP2
        resource(2, allocate)
        wait(3)
        resource(1, allocate)
        for n = 1 to 3
        next
end
```

(2) Compile each one of the source codes then load into the main memory
(3) Now switch to the OS simulator.

(4) Create a single process of DeadlockP1 first and then DeadlockP2. You can do this by double-clicking on each of the program names in the **PROGRAM LIST** frame under the **Program Name** column.
(5) In the **SCHEDULER** frame select **FCFS** scheduling policy in the **Policies** tab.
(6) Back in the **OS Control** tab use the **START** button to start the OS scheduler and observe the changing process states for few seconds.

Q1- Have you got a deadlock condition above? Explain using the tools provided by the simulator (add screenshots).

Answer:

Yes. R1 and R2 caused the deadlock because it is allocated to P1, and R2 to P2. Thus, the further request from P1 > R2 and P2 > R1 create a circular wait.

(7) If you have a deadlock condition then click on the **SHOW DEADLOCKED PROCESSES…** button in the **System Resources** window (Views tab, view resources). Observe the resource allocation graph.
(8) Now go back to the compiler window. And enter the following code

```
program DeadlockP3
        resource(2, allocate)
        resource(1, allocate)
        wait(3)
        for n = 1 to 3
        next
end
```

(9) Compile each one of the source codes then load into the main memory
(10) Now switch to the OS simulator.
(11) Create a single process of DeadlockP1 first and then DeadlockP3.
(12) Back in the **OS Control** tab use the **START** button to start the OS scheduler and observe the changing process states for few seconds.

Q2 - Have you got a deadlock condition this time? Explain using the tools provided by the simulator (add screenshots).

Answer :

Deadlock occurred because of P3 was allocated to R2 and P1 was allocated to R1. Thus, A circular wait was created when P3 > R1 and P1 > R2.

(13) Now Create a single process of DeadlockP3 first and then DeadlockP1.

Q3 - Have you got a deadlock condition this time? Explain why? Use tools of the simulator to support your observation. You can find some useful tools under the "Views" tab of the OS simulator. Use the screenshots and support your outcomes.

Answer:

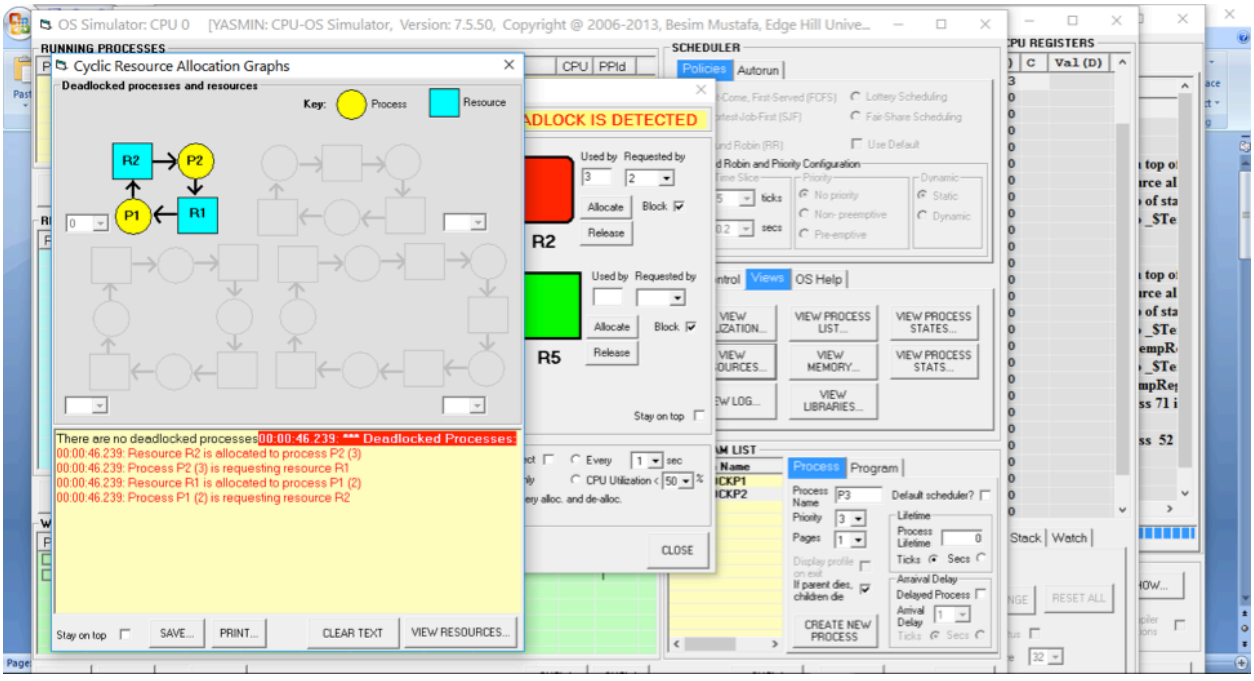
No deadlock in this case.

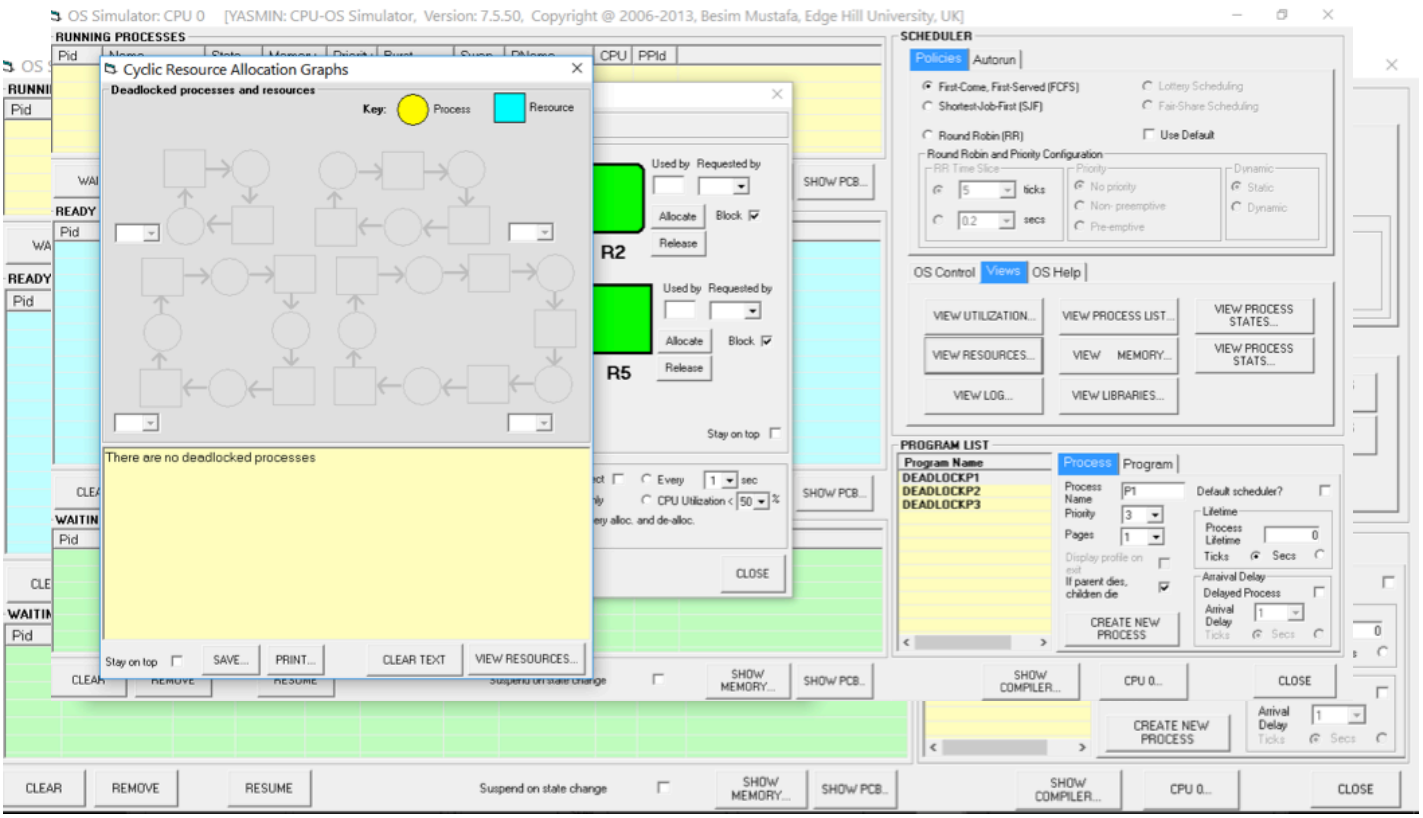
Q4 –Create a deadlock case involving 3 processors and 3 resources? Use the screenshots and support your outcomes.

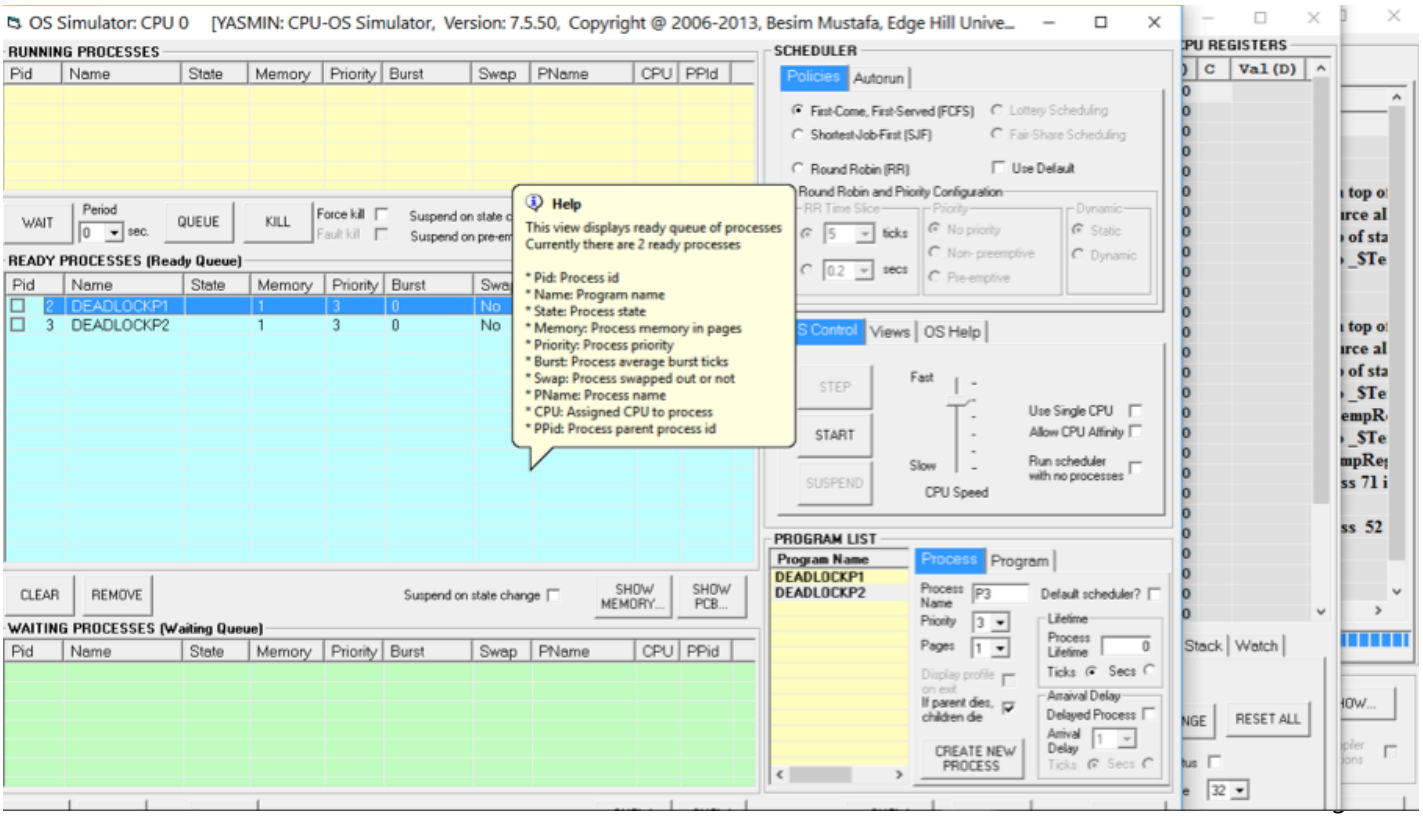
Answer:

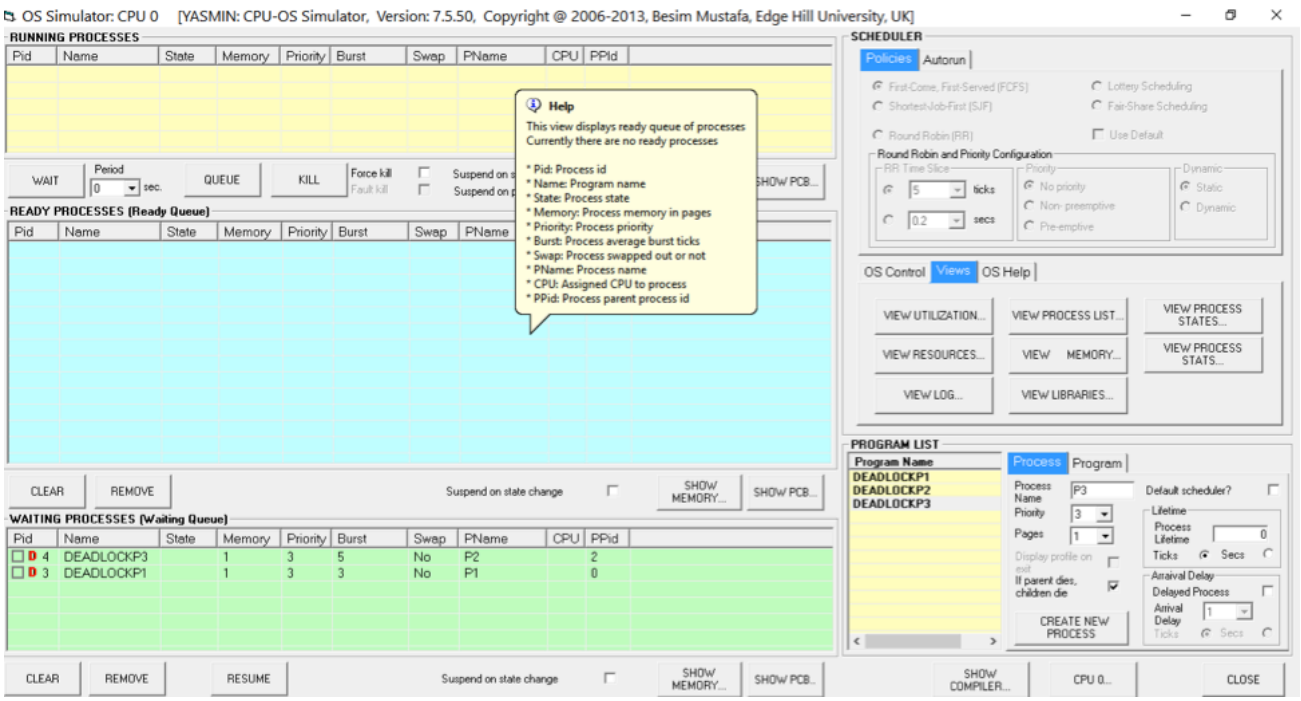
```
program DeadlockP1
        resource(1, allocate)
        wait(3)
        resource(2, allocate)
        for n = 1 to 3
        next
end


program DeadlockP2
        resource(2, allocate)
        wait(3)
        resource(3, allocate)
        for n = 1 to 3
        next
end

program DeadlockP3
        resource(3, allocate)
        wait(3)
        resource(1, allocate)
        for n = 1 to 3
        next
end
```

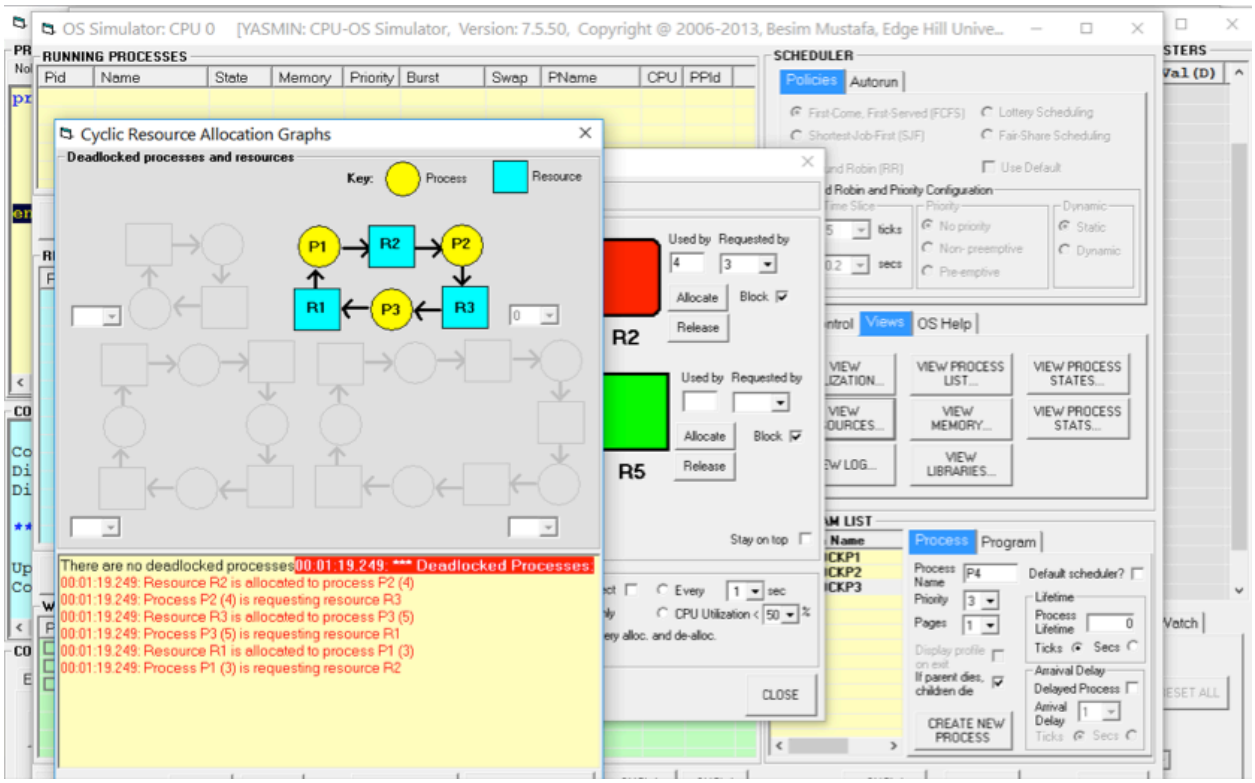
Screenshots

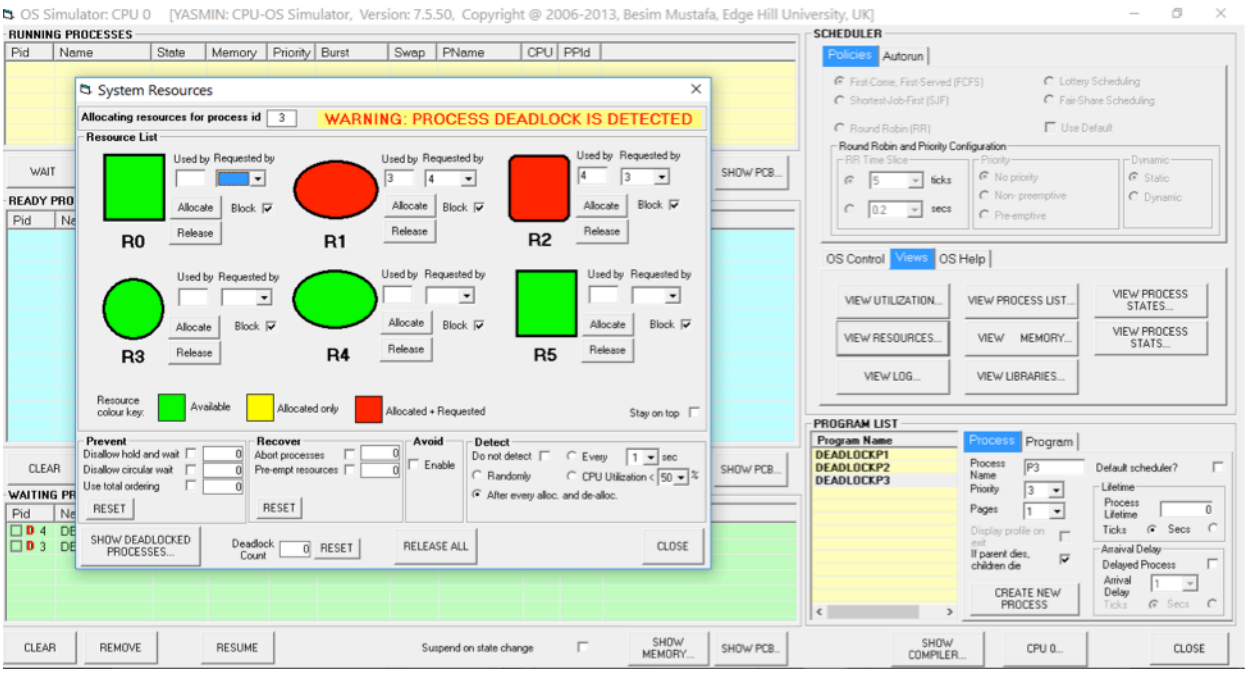



OS Simulator: CPU 0 [YASMIN: CPU-OS Simulator, Version: 7.5.50, Copyright @ 2006-2013, Besim Mustafa, Edge Hill University, UK]
RUNNING PROCESSES
Cyclic Resource Allocation Graphs
Deadlocked processes and resources
Key: Process Resource
There are no deadlocked processes
Stay on top
SAVE...
PRINT...
CLEAR TEXT
VIEW RESOURCES...
SCHEDULER
Policies
Autorun
First-Come, First-Served (FCFS)
Shortest-Job-First (SJF)
Round Robin (RR)
Lottery Scheduling
Fair-Share Scheduling
Use Default
Round Robin and Priority Configuration
OS Control
Views
OS Help
VIEW UTILIZATION...
VIEW PROCESS LIST...
VIEW PROCESS STATES...
VIEW RESOURCES...
VIEW MEMORY...
VIEW PROCESS STATS...
VIEW LOG...
VIEW LIBRARIES...
PROGRAM LIST
Program Name
DEADLOCKP1
DEADLOCKP2
DEADLOCKP3
Process
Program
CREATE NEW PROCESS
R2
R5
Allocate
Release
Block
Used by
Requested by
SHOW PCB...
SHOW MEMORY...
SHOW COMPILER...
CPU 0...
CLOSE
CLEAR
REMOVE
RESUME
Suspend on state change

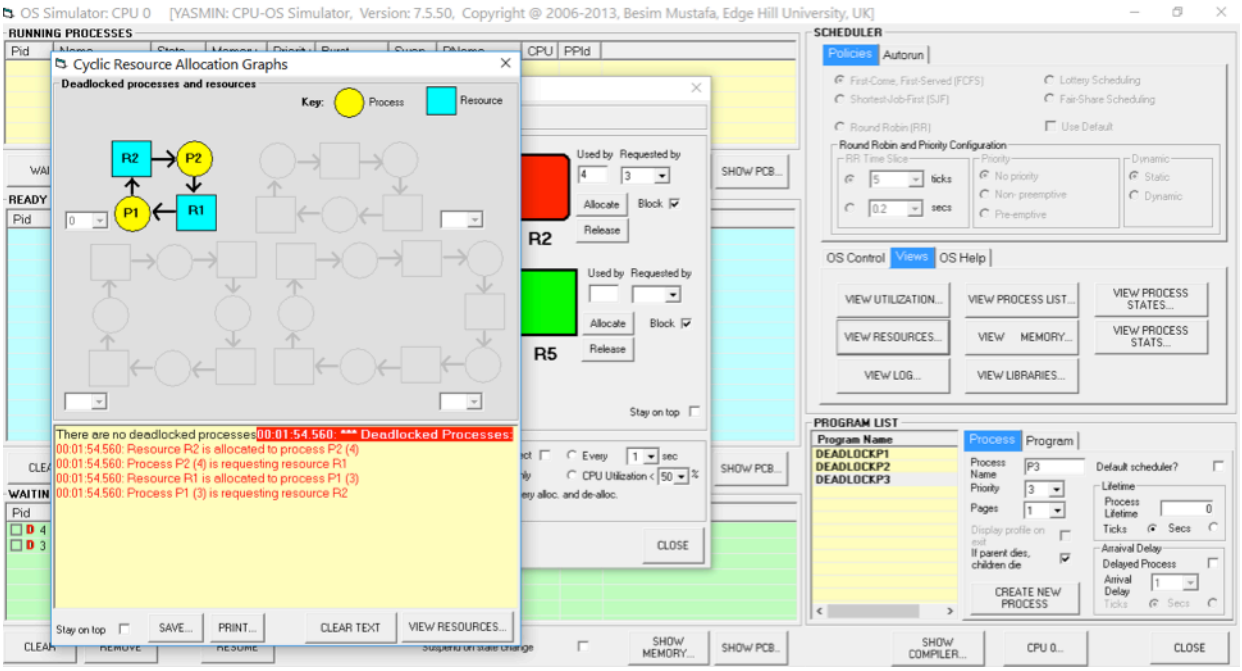

OS Simulator: CPU 0 [YASMIN: CPU-OS Simulator, Version: 7.5.50, Copyright @ 2006-2013, Besim Mustafa, Edge Hill Unive...
RUNNING PROCESSES
Pid
Name
State
Memory
Priority
Burst
Swap
PName
CPU
PPId
WAIT
Period
QUEUE
KILL
Force kill
Fault kill
Suspend on state c
Suspend on pre-en
READY PROCESSES (Ready Queue)
2 DEADLOCKP1 1 3 0 No
3 DEADLOCKP2 1 3 0 No
Help
This view displays ready queue of processes
Currently there are 2 ready processes
* Pid: Process id
* Name: Program name
* State: Process state
* Memory: Process memory in pages
* Priority: Process priority
* Burst: Process average burst ticks
* Swap: Process swapped out or not
* PName: Process name
* CPU: Assigned CPU to process
* PPid: Process parent process id
CLEAR
REMOVE
Suspend on state change
SHOW MEMORY...
SHOW PCB...
WAITING PROCESSES (Waiting Queue)
SCHEDULER
Policies
Autorun
First-Come, First-Served (FCFS)
Shortest-Job-First (SJF)
Round Robin (RR)
Lottery Scheduling
Fair-Share Scheduling
Use Default
Round Robin and Priority Configuration
STEP
START
SUSPEND
Fast
Slow
CPU Speed
Use Single CPU
Allow CPU Affinity
Run scheduler with no processes
PROGRAM LIST
Program Name
DEADLOCKP1
DEADLOCKP2
Process
Program
CREATE NEW PROCESS
CPU REGISTERS
Val (D)
Stack
Watch
RESET ALL

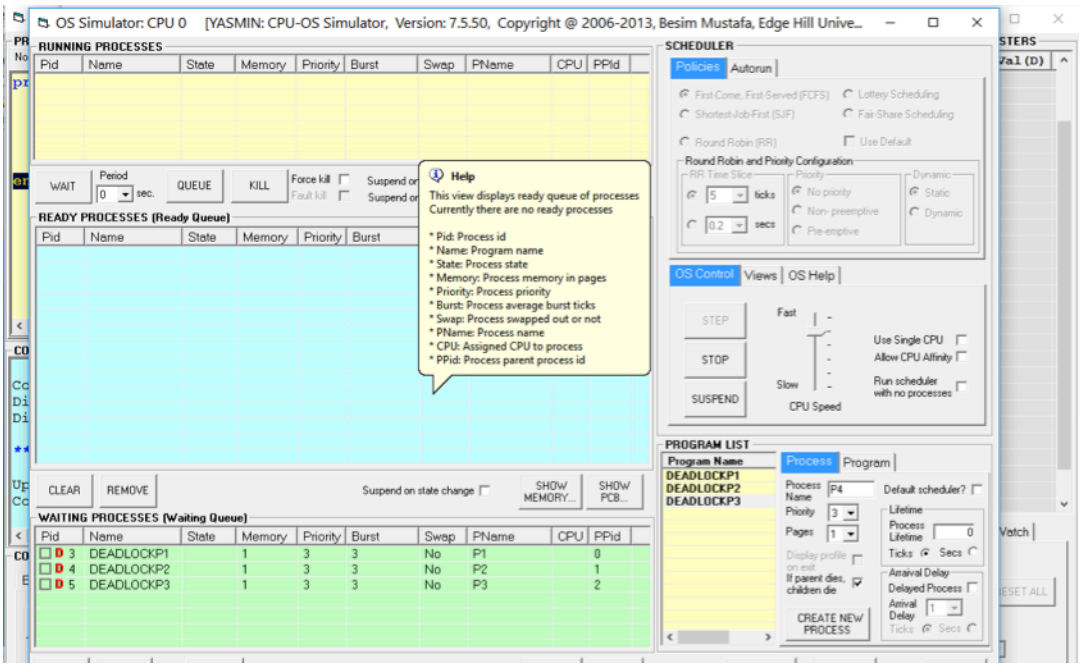

OS Simulator: CPU 0 [YASMIN: CPU-OS Simulator, Version: 7.5.50, Copyright @ 2006-2013, Besim Mustafa, Edge Hill University, UK]
RUNNING PROCESSES
Pid Name State Memory Priority Burst Swap PName CPU PPId
Help
This view displays ready queue of processes
Currently there are no ready processes
* Pid: Process id
* Name: Program name
* State: Process state
* Memory: Process memory in pages
* Priority: Process priority
* Burst: Process average burst ticks
* Swap: Process swapped out or not
* PName: Process name
* CPU: Assigned CPU to process
* PPid: Process parent process id
WAIT
Period
QUEUE
KILL
READY PROCESSES (Ready Queue)
CLEAR
REMOVE
Suspend on state change
SHOW MEMORY...
SHOW PCB...
WAITING PROCESSES (Waiting Queue)
D 4 DEADLOCKP3 1 3 5 No P2 2
D 3 DEADLOCKP1 1 3 3 No P1 0
RESUME
SCHEDULER
Policies
Autorun
First-Come, First-Served (FCFS)
Shortest-Job-First (SJF)
Round Robin (RR)
Lottery Scheduling
Fair-Share Scheduling
Use Default
Round Robin and Priority Configuration
OS Control
Views
OS Help
VIEW UTILIZATION...
VIEW PROCESS LIST...
VIEW PROCESS STATES...
VIEW RESOURCES...
VIEW MEMORY...
VIEW PROCESS STATS...
VIEW LOG...
VIEW LIBRARIES...
PROGRAM LIST
Program Name
DEADLOCKP1
DEADLOCKP2
DEADLOCKP3
Process
Program
Process Name
Priority
Pages
Default scheduler?
Lifetime
Process Lifetime
Ticks
Secs
Arrival Delay
Delayed Process
Arrival Delay
If parent dies, children die
CREATE NEW PROCESS
SHOW COMPILER...
CPU 0...
CLOSE

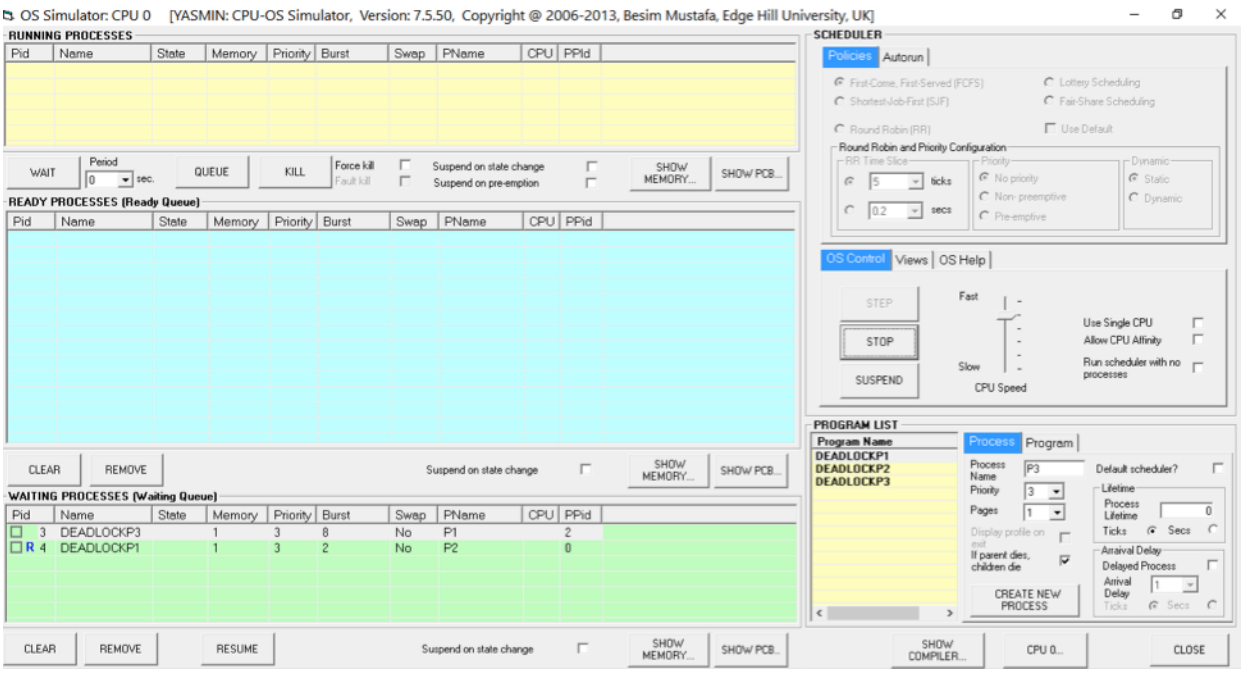

OS Simulator: CPU 0 [YASMIN: CPU-OS Simulator, Version: 7.5.50, Copyright @ 2006-2013, Besim Mustafa, Edge Hill Unive...
RUNNING PROCESSES
Cyclic Resource Allocation Graphs
Deadlocked processes and resources
Key: Process Resource
P1 R2 P2
R1 P3 R3
There are no deadlocked processes00:01:19.249: *** Deadlocked Processes:
00:01:19.249: Resource R2 is allocated to process P2 (4)
00:01:19.249: Process P2 (4) is requesting resource R3
00:01:19.249: Resource R3 is allocated to process P3 (5)
00:01:19.249: Process P3 (5) is requesting resource R1
00:01:19.249: Resource R1 is allocated to process P1 (3)
00:01:19.249: Process P1 (3) is requesting resource R2
Used by Requested by
Allocate
Block
Release
R2
R5
Stay on top
CLOSE
SCHEDULER
Policies
Autorun
Views
OS Help
VIEW PROCESS LIST...
VIEW PROCESS STATES...
VIEW MEMORY...
VIEW PROCESS STATS...
VIEW LIBRARIES...
Process
Program
Process Name
Priority
Pages
Default scheduler?
Lifetime
Process Lifetime
Ticks
Secs
Arrival Delay
Delayed Process
Arrival Delay
If parent dies, children die
CREATE NEW PROCESS


OS Simulator: CPU 0 [YASMIN: CPU-OS Simulator, Version: 7.5.50, Copyright @ 2006-2013, Besim Mustafa, Edge Hill University, UK]
RUNNING PROCESSES
Pid
Name
State
Memory
Priority
Burst
Swap
PName
CPU
PPId
System Resources
Allocating resources for process id
3
WARNING: PROCESS DEADLOCK IS DETECTED
Resource List
R0
R1
R2
R3
R4
R5
Used by
Requested by
Allocate
Block
Release
Resource colour key:
Available
Allocated only
Allocated + Requested
Stay on top
Prevent
Disallow hold and wait
Disallow circular wait
Use total ordering
Recover
Abort processes
Pre-empt resources
Avoid
Enable
Detect
Do not detect
Every
sec
Randomly
CPU Utilization <
After every alloc. and de-alloc.
RESET
SHOW DEADLOCKED PROCESSES...
Deadlock Count
RELEASE ALL
CLOSE
SCHEDULER
Policies
Autorun
First-Come, First-Served (FCFS)
Shortest-Job-First (SJF)
Round Robin (RR)
Lottery Scheduling
Fair-Share Scheduling
Use Default
Round Robin and Priority Configuration
RR Time Slice
ticks
secs
Priority
No priority
Non-preemptive
Pre-emptive
Dynamic
Static
OS Control
Views
OS Help
VIEW UTILIZATION...
VIEW PROCESS LIST...
VIEW PROCESS STATES...
VIEW RESOURCES...
VIEW MEMORY...
VIEW PROCESS STATS...
VIEW LOG...
VIEW LIBRARIES...
PROGRAM LIST
Program Name
DEADLOCKP1
DEADLOCKP2
DEADLOCKP3
Process
Program
Process Name
P3
Priority
Pages
Default scheduler?
Lifetime
Process Lifetime
Ticks
Secs
If parent dies, children die
Arrival Delay
Delayed Process
Arrival Delay
CREATE NEW PROCESS
WAIT
READY PRO
WAITING PR
CLEAR
REMOVE
RESUME
Suspend on state change
SHOW MEMORY...
SHOW PCB...
SHOW COMPILER...
CPU 0...
CLOSE


System Resources
WARNING: PROCESS DEADLOCK IS DETECTED
R0
R1
R2
R3
R4
R5
SHOW DEADLOCKED PROCESSES...
RELEASE ALL
CLOSE


Cyclic Resource Allocation Graphs
Deadlocked processes and resources
There are no deadlocked processes
00:01:54:560: *** Deadlocked Processes:
00:01:54:560: Resource R2 is allocated to process P2 (4)
00:01:54:560: Process P2 (4) is requesting resource R1
00:01:54:560: Resource R1 is allocated to process P1 (3)
00:01:54:560: Process P1 (3) is requesting resource R2
SAVE...
PRINT...
CLEAR TEXT
VIEW RESOURCES...





PS. If you cant read the screenshots from the documents do not hesitate to visit my Github Repository for this Class [Operating System Fall 2016]