

CENG 328 OPERATING SYSTEM Lab Quiz3

Name Surname:
Student ID:
Section: 1

input0.dat

```
C 10 // compute for 10 milliseconds
R 0 // request resource 0
C 10 // compute for 10 milliseconds
R 1 // request resource 1
C 10 // compute for 10 milliseconds
R 2 // request resource 2
C 5 // compute for 5 milliseconds
F 2 // free resource 1
F 1 // free resource 1
F 0 // free resource 0
H // halt process
```

input1.dat

```
C 10 // compute for 10 milliseconds
R 1 // request resource 1
C 5 // compute for 5 milliseconds
F 1 // free resource 1
H // halt process
```

input2.dat

```
C 10 // compute for 10 milliseconds
R 2 // request resource 2
C 20 // compute for 20 milliseconds
R 1 // request resource 1
C 10 // compute for 10 milliseconds
F 1 // free resource 1
F 2 // free resource 2
H // halt process
```

Question: Explain what happen if we run the following command in MOSS deadlock simulator? Is there any deadlock?

```
$java deadlock input 3 2 1 2
```

Answer:

According to the Moss Deadlock simulator, here we have 3 processes and 3 resources with 2, 1 and 2 instance respectively. Processes schedule files are input0, input1 and input2 respectively.

There is no DEADLOCK.

Time	Process 1	Process 2	Process 3	Available Resources
0-10 s	executed	executed	executed	R0,R0,R1,R2,R2
10. s	request resource 0 and take it	request resource 1 and take it	request resource 2 and take it	R0,R2
15. s	executed with resource 0	Free resource 1 and halted	Executed with resource 2	R0,R1,R2
20. s	request resource 1 and take it		Executed with resource 2	R0,R2
30. s	request resource 2 and take it		request resource 1 and since R1 is not available blocked	R0
35.s	Free resource 2,1,0 and halted		Take resource 1	R0,R0,R2,R2
40. s			Free resource 1,2 and halted.	R0,R0,R1,R2,R2

CENG 328 OPERATING SYSTEM Lab Quiz3

Name Surname:
Student ID:
Section: 2

input0.dat

```
C 10 // compute for 10 milliseconds
R 0 // request resource 0
C 10 // compute for 10 milliseconds
R 1 // request resource 1
C 10 // compute for 10 milliseconds
R 2 // request resource 2
C 5 // compute for 5 milliseconds
F 2 // free resource 2
F 1 // free resource 1
F 0 // free resource 0
H // halt process
```

input1.dat

```
C 10 // compute for 10 milliseconds
R 1 // request resource 1
C 10 // compute for 10 milliseconds
F 1 // free resource 1
H // halt process
```

input2.dat

```
C 10 // compute for 10 milliseconds
R 2 // request resource 2
C 20 // compute for 20 milliseconds
R 1 // request resource 1
C 5 // compute for 5 milliseconds
F 1 // free resource 1
F 2 // free resource 2
H // halt process
```

Question: Explain what happen if we run the following command in MOSS deadlock simulator? Is there any deadlock?

```
$java deadlock input 3 2 1 1
```

Answer:

According to the Moss Deadlock simulator, here we have 3 processes and 3 resources with 2, 1 and 1 instance respectively. Processes schedule files are input0, input1 and input2 respectively.

There is DEADLOCK.

Time	Process 1	Process 2	Process 3	Available Resources
0-10 s	executed	executed	executed	R0,R0,R1,R2
10. s	request resource 0 and take it	request resource 1 and take it	request resource 2 and take it	R0
20. s	request resource 1 and take it	Free resource 1 and halted	Executed with resource 2	R0
30. s	request resource 2 and since R2 is not available blocked		request resource 1 and since R1 is not available blocked	R0

CENG 328 OPERATING SYSTEM Lab Quiz3

Name Surname:
Student ID:
Section: 3

input0.dat

```
C 10 // compute for 10 milliseconds
R 0 // request resource 0
C 10 // compute for 10 milliseconds
R 1 // request resource 1
C 10 // compute for 10 milliseconds
R 2 // request resource 2
C 5 // compute for 5 milliseconds
F 2 // free resource 2
F 1 // free resource 1
F 0 // free resource 0
H // halt process
```

input1.dat

```
C 5 // compute for 5 milliseconds
R 0 // request resource 0
C 30 // compute for 30 milliseconds
F 0 // free resource 0
H // halt process
```

input2.dat

```
C 10 // compute for 10 milliseconds
R 2 // request resource 2
C 20 // compute for 20 milliseconds
R 1 // request resource 1
C 5 // compute for 5 milliseconds
F 1 // free resource 1
F 2 // free resource 2
H // halt process
```

Question: Explain what happen if we run the following command in MOSS deadlock simulator? Is there any deadlock?

```
$java deadlock input 3 1 2 2
```

Answer:

According to the Moss Deadlock simulator, here we have 3 processes and 3 resources with 1, 2 and 2 instance respectively. Processes schedule files are input0, input1 and input2 respectively.

There is no DEADLOCK.

Time	Process 1	Process 2	Process 3	Available Resources
0-5 s	executed	executed	executed	R0,R1,R1,R2,R2
5. s	executed	request resource 0 and take it	executed	R1,R1,R2,R2
10. s	request resource 0 and since R0 is not available blocked	Executed with resource 0	request resource 2 and take it	R1,R1,R2
30. s	Blocked	Executed with resource 0	request resource 1 and take it	R1,R2
35. s	take resource 0	Free resource 0 and halted	Free resource 1,2 and halted.	R1,R1,R2,R2
45. s	request resource 1 and take it			R1,R2,R2
55. s	request resource 2 and take it			R1,R2
60. s	Free resource 2,1,0 and halted.			R0,R0,R1,R2,R2

CENG 328 OPERATING SYSTEM Lab Quiz3

Name Surname:
Student ID:
Section: 4

input0.dat

```
C 10 // compute for 10 milliseconds
R 0 // request resource 0
C 10 // compute for 10 milliseconds
R 1 // request resource 1
C 10 // compute for 10 milliseconds
R 2 // request resource 2
C 5 // compute for 5 milliseconds
F 2 // free resource 2
F 1 // free resource 1
F 0 // free resource 0
H // halt process
```

input1.dat

```
C 10 // compute for 10 milliseconds
R 1 // request resource 1
C 5 // compute for 5 milliseconds
F 1 // free resource 1
H // halt process
```

input2.dat

```
C 10 // compute for 10 milliseconds
R 2 // request resource 2
C 20 // compute for 20 milliseconds
R 1 // request resource 1
C 10 // compute for 10 milliseconds
F 1 // free resource 1
F 2 // free resource 2
H // halt process
```

Question: Explain what happen if we run the following command in MOSS deadlock simulator? Is there any deadlock?

```
$java deadlock input 3 1 1 1
```

Answer:

According to the Moss Deadlock simulator, here we have 3 processes and 3 resources with 1 instance. Processes schedule files are input0, input1 and input2 respectively.

There is DEADLOCK.

Time	Process 1	Process 2	Process 3	Available Resources
0-10 s	executed	executed	executed	R0,R1,R2
10. s	request resource 0 and take it	request resource 1 and take it	request resource 2 and take it	
15. s	execute with resource 0	Free resource 1 and halted	execute with resource 2	R1
20. s	request resource 1 and take it		execute with resource 2	
30. s	request resource 2 and since it is not available, blocked		request resource 1 and since it is not available, blocked	R1,R2,R2

CENG 328 OPERATING SYSTEM Lab Quiz3

Name Surname:
Student ID:
Section: 5

input0.dat

```
C 10 // compute for 10 milliseconds
R 0 // request resource 0
C 10 // compute for 10 milliseconds
R 1 // request resource 1
C 10 // compute for 10 milliseconds
R 2 // request resource 2
C 5 // compute for 5 milliseconds
F 2 // free resource 2
F 1 // free resource 1
F 0 // free resource 0
H // halt process
```

input1.dat

```
C 10 // compute for 10 milliseconds
R 1 // request resource 1
C 10 // compute for 10 milliseconds
F 1 // free resource 1
H // halt process
```

input2.dat

```
C 10 // compute for 10 milliseconds
R 2 // request resource 2
C 10 // compute for 10 milliseconds
R 1 // request resource 1
C 5 // compute for 5 milliseconds
F 1 // free resource 1
F 2 // free resource 2
H // halt process
```

Question: Explain what happen if we run the following command in MOSS deadlock simulator? Is there any deadlock?

```
$java deadlock input 3 1 2 1
```

Answer:

According to the Moss Deadlock simulator, here we have 3 processes and 3 resources with 1,2 and 1 instance respectively. Processes schedule files are input0, input1 and input2 respectively.

There is no DEADLOCK.

Time	Process 1	Process 2	Process 3	Available Resources
0-10 s	executed	executed	executed	R0,R1,R1,R2
10. s	request resource 0 and take it	request resource 1 and take it	request resource 2 and take it	R1
20. s	request resource 1 and take it	Free resource 1 and halted	request resource 1 and take it	
25. s	execute with resource 0 and 1		Free resource 1,2 and halted	R1,R2
30. s	request resource 2 and take it			R1
35. s	Free resource 2,1,0 and halted			R0,R1,R1,R2