CS241 Lawrence Angrave L17 – Deadlock II, Dining Philosophers

Three gardeners visit the garden shed pick up their desired tools for the day. There is a potential for deadlock. Fortunately they know about the Coffman conditions! Find four ways to solve the problem (break one Coffman condition each time). Name which condition you break in each case.

Remember Mergesort? How can you implement parallel Mergesort? Explain what synchronization calls you will use and when.

What is a context switch?

When does it happen?

Mapping Virtual Memory to RAM

Design a Single-level paging system for a 32 bit system, faster than you can say "Hardware people! Why is the hardware always late? :-)"

|  |  |
| --- | --- |
| 20 bits | 12 bits |

Virtual MemoryAddress:

|  |
| --- |
| My $$ RAM |
| 4096 bytes |
| 4096 bytes |
| 4096 bytes |
| 4096 bytes |
| 4096 bytes |
| 4096 bytes |
| 4096 bytes |
| 4096 bytes |
| 4096 bytes |
| 4096 bytes |
| .... |

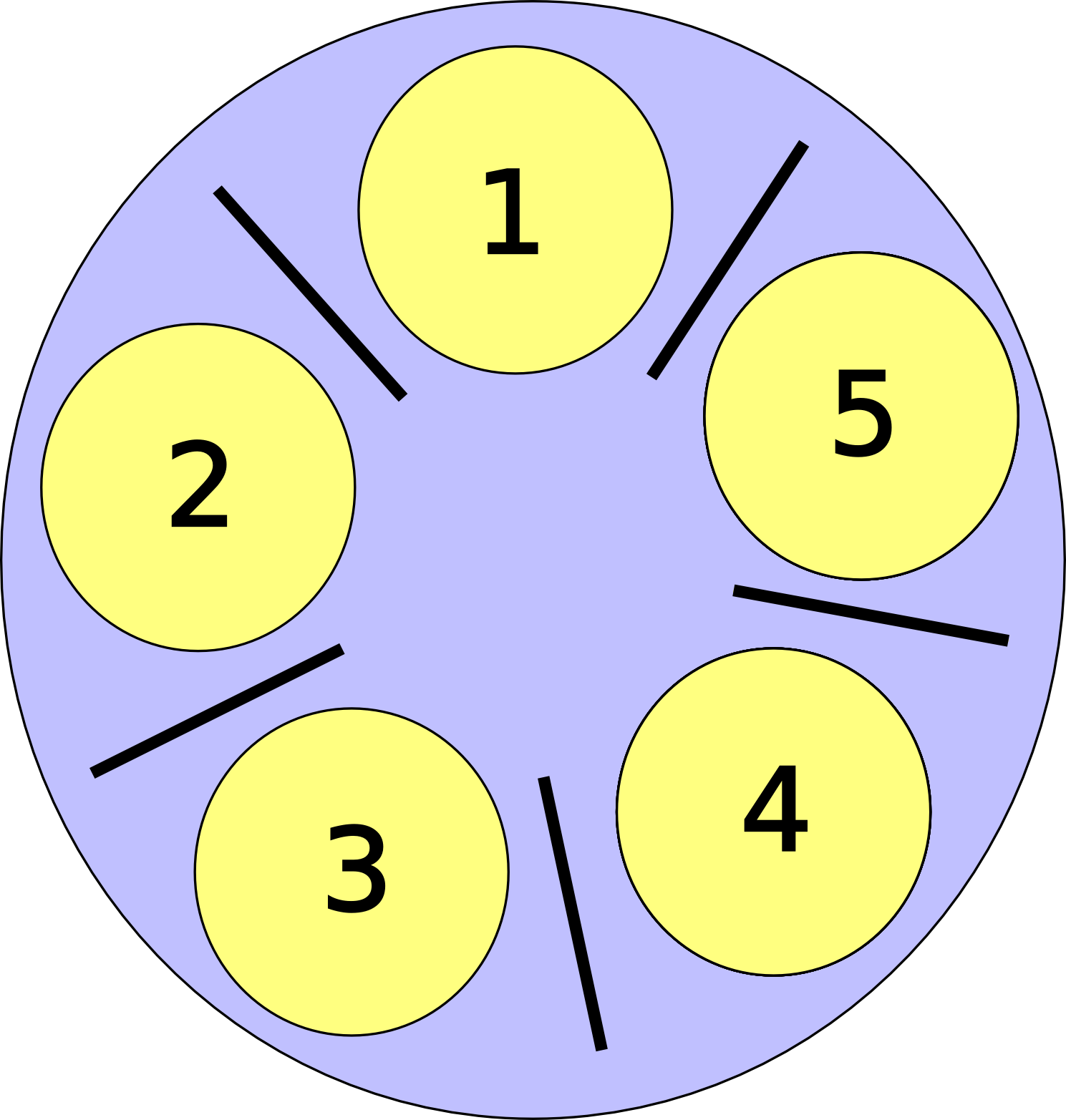
Page table:

(per process? or global?)

Can your process

write to the page table ?

(what would happen if it could?)

What is the Dining Philosophers problem?

Candidate Solutions:

1. "Pick up left chopstick. Pickup right chopstick. Eat. Release both."

2. "Pick up right. Pick up left. Eat. Release both"

3. "Eat when I tell you"

4. "Pick up left chopstick. Try to pickup right chopstick (Fail? release both and restart). Eat. Release both."

5?

Monday Quiz: Implement a barrier using Condition Variables. The 3 steps of cond\_wait. Locking mutual exclusion. Implement a multi-threaded (blocking) Ring Buffer using semaphores or condition variables. Spotting synchronization errors in multi-threaded code that uses semaphores/mutexes/condition variables- race conditions, mutual exclusion, bounded wait, starvation, progress. The producer consumer. The reader writer problem. "Writers preference solution" to the R/W problem.

Deadlock. Coffman conditions. Applying Coffman conditions. Dining Philosophers.