Examples of exercises
Use case: Sleepduring

ARCOS

Operating Systems Design

Degree in Computer Engineering

University Carlos III of Madrid

Exercise
statement (1/2)

There is a HW system that includes a clock device. That clock generates an interruption on each tick.

An multi-process OS with a non-preemptive kernel needs to implement a new system call:

```
int sleepLater (int waitfor_s, int sleep_s);
```

This system call allows the calling process to continue executing (if not get blocked for any other reason) for *waitfor_s* seconds, and afterwards, it will sleep for *sleep_s* seconds.

Exercise
statement (2/2)

You are asked to:

a) Implement using pseudocode the required kernel feature.
   You will have to include in your answer:
   - All data structures required or modified
   - Interfaces with implemented functions
   - Events used

Exercise
solution

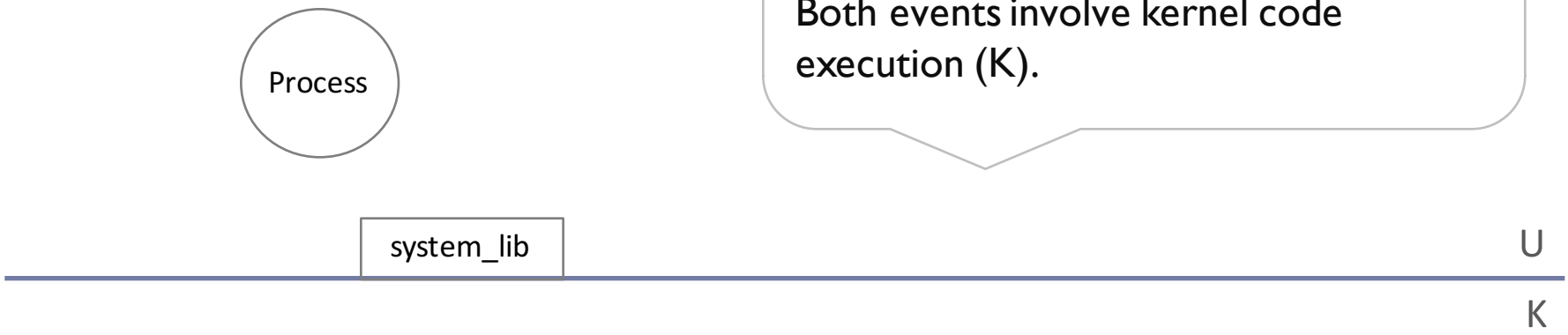
1. Initial approach:
    1. Draw a diagram of initial system state
    2. Modify the diagram to incorporate the exercise requirements
2. Answer the proposed questions
3. Review the answers

Exercise
solution

1. Initial approach:
    1. Draw a diagram of initial system state
    2. Modify the diagram to incorporate the exercise requirements
2. Answer the proposed questions
3. Review the answers

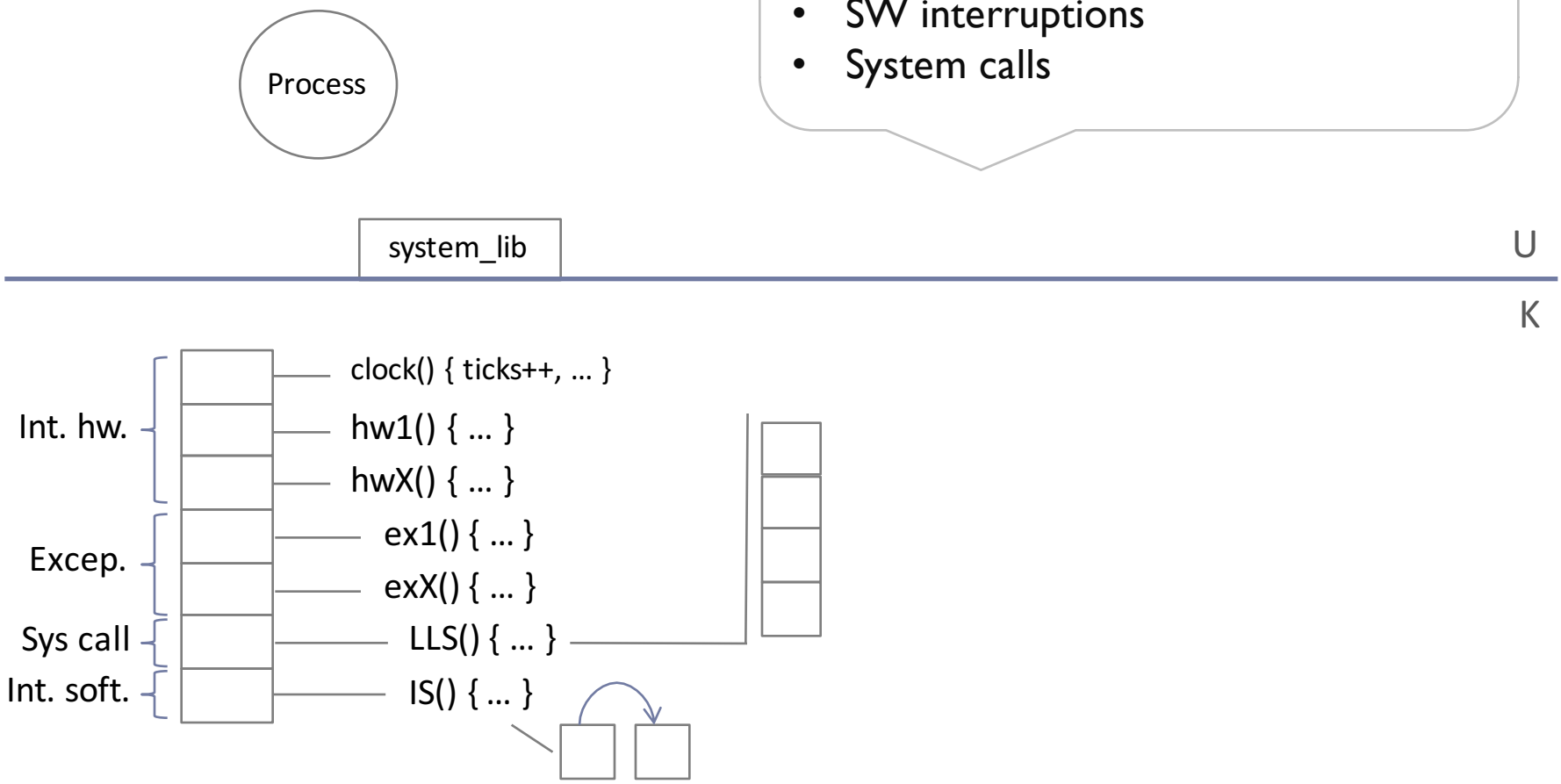
Exercise
solution

In user space (U) processes perform system calls through *system_lib* or provoke exceptions.
Both events involve kernel code execution (K).



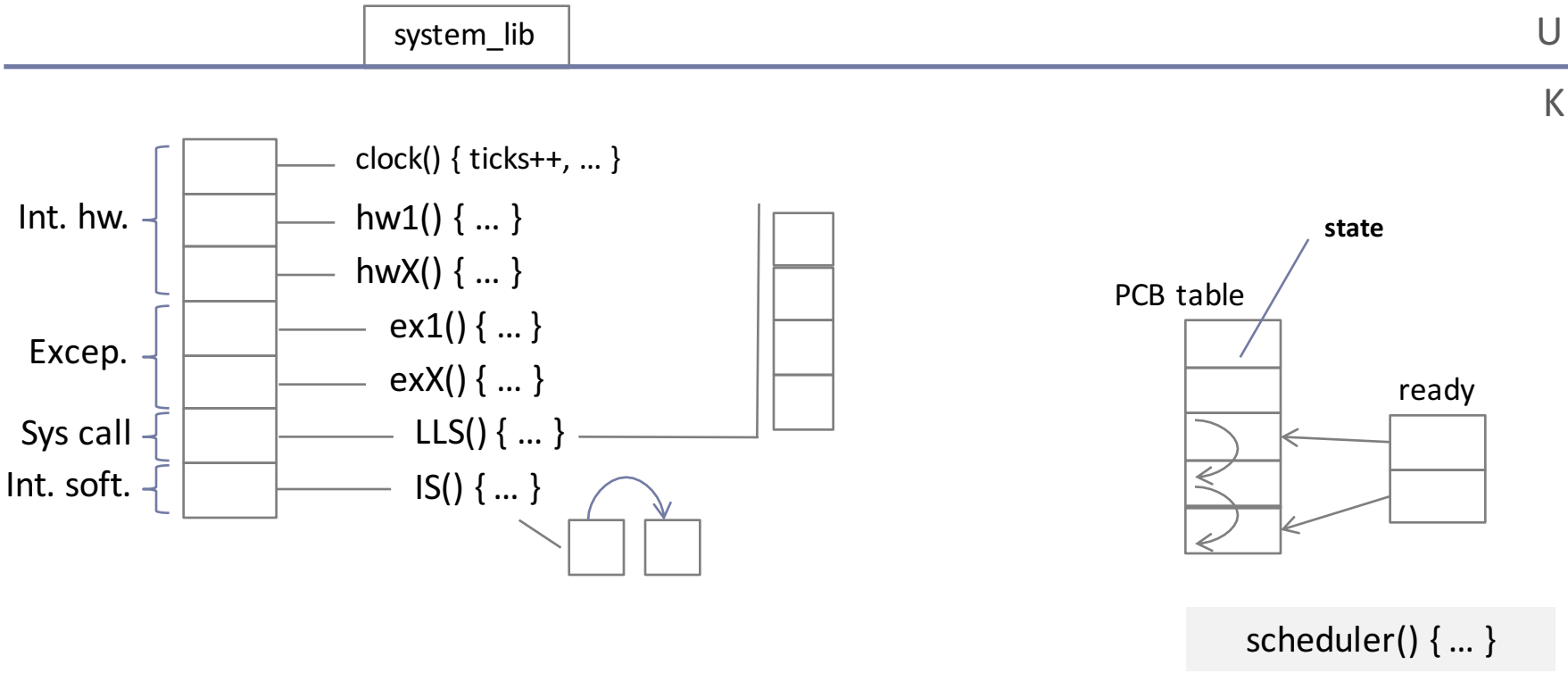
Exercise
solution

Lesson 2: operating system working
- HW interruptions
- Exceptions
- SW interruptions
- System calls



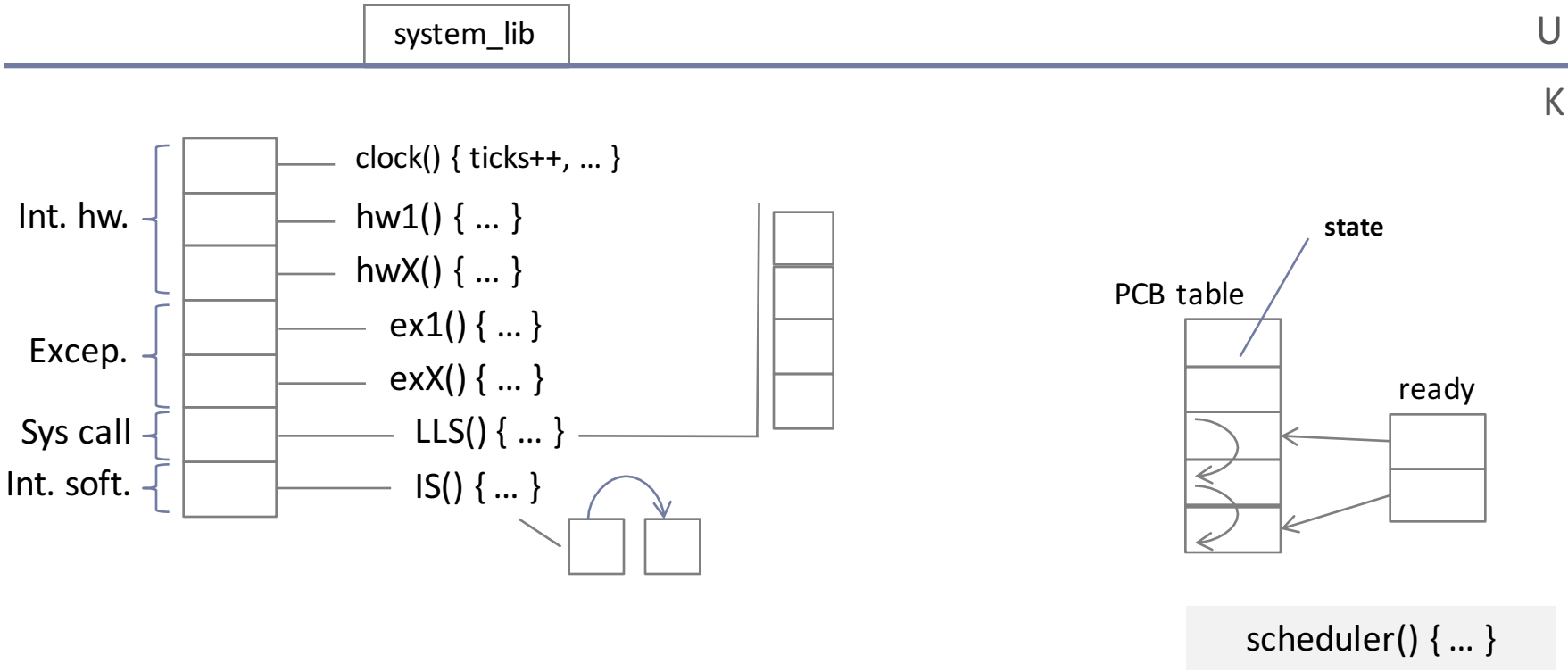
Exercise
solution

Lesson 3b: process management
- PCB table
- Ready-state queues
- scheduler



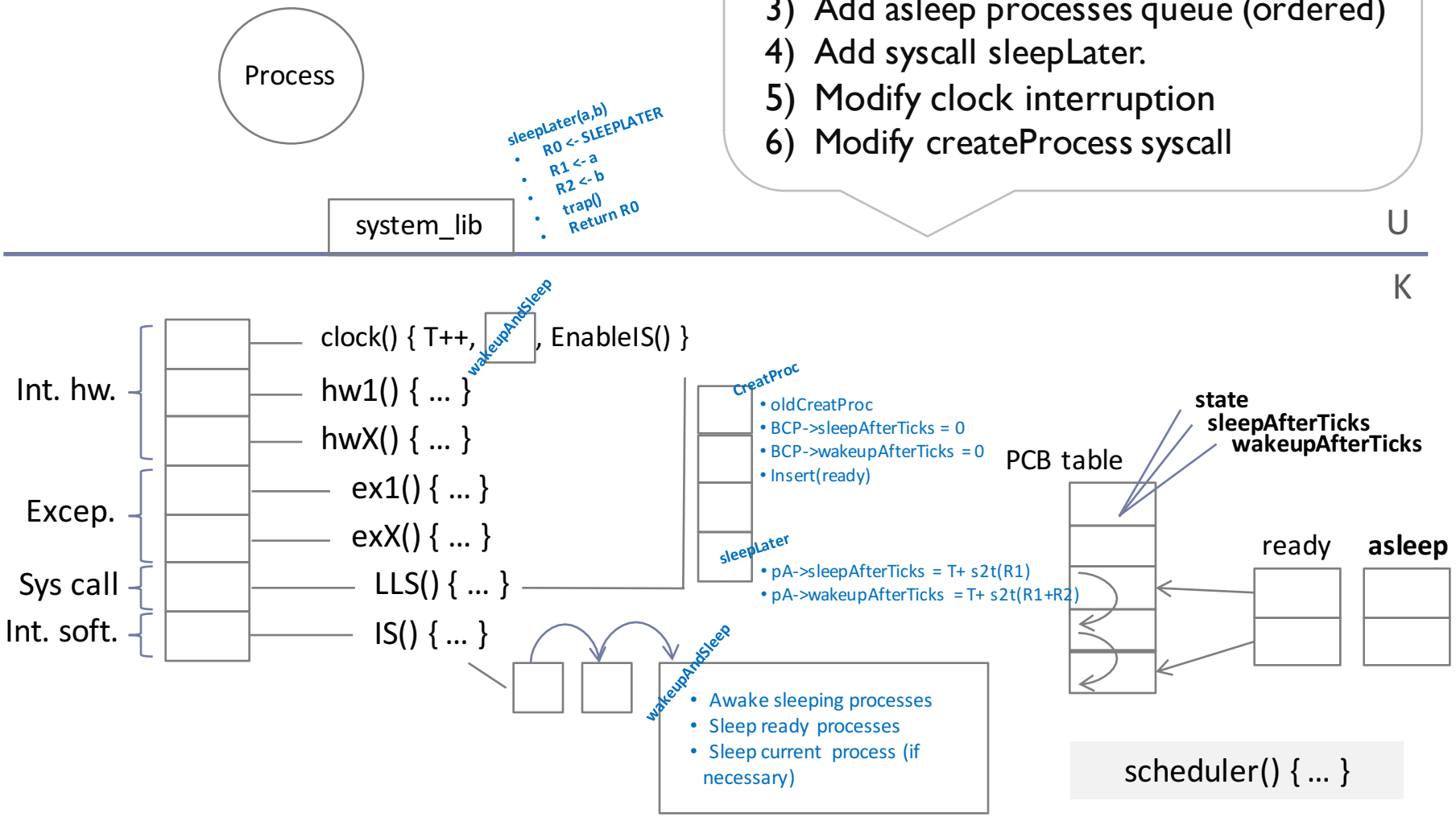
Exercise
solution

Initial structure completed



Exercise
solution

Adding sleepLater to the kernel:
1) Add 'sleepAfterTicks' field to PCB
2) Add 'wakeupAfterTicks' field to PCB
3) Add asleep processes queue (ordered)
4) Add syscall sleepLater.
5) Modify clock interruption
6) Modify createProcess syscall



Exercise
solution

1. Initial approach:
    1. Draw a diagram of initial system state
    2. Modify the diagram to incorporate the exercise requirements
2. Answer the proposed questions
3. Review the answers

Exercise
solution

Based on the proposed approach, answer the questions

Data structures:

- PCB:
  - sleepAfterTicks
  - wakeupAfterTicks
  - State: add a new state ASLEEP
- Add a new queue:
  - Sleeping processes queue

Exercise
solution

Funcions:

int createProcess(…)

- Set sleepAfterTicks and wakeupAfterTicks to null values
- return createProcess_base(…)

int sleepAfter (int waitfor_s, int sleep_s)

- current->wakeupAfterTicks = Ticks + SecondsToTicks(waitfor_s+sleep_s);
- current->sleepAfterTicks = Ticks + SecondsToTicks(waitfor_s);

clock_hw_interruption()

- Ticks = Ticks + 1;
- Insert_Software_Interruption(IntSwWakeupAndSleep)
- Software_Interruption();

Exercise
solution

IntSwWakeupAndSleep()

- proc = ExtractFirstProcess (asleep_processes_queue);
- While (Proc->wakeupAfterTicks =< Ticks)
  - proc->state = READY; // Change state **from asleep to ready**.
  - Enqueue(ready_processes_queue, proc); // Enqueue in **ready state queue**.
  - sleepAfterTicks = wakeupAfterTicks = 0; // Set sleepAfterTicks and wakeupAfterTicks to null values
  - Proc = ExtractFirstProcess (asleep_processes_queue);
- Proc = Scheduler() ; // ExtractFirstProcess (ready_processes_queue);
- While((Proc->sleepAfterTicks =< Ticks) && (Proc->wakeupAfterTicks > Ticks))
  - Proc->state = ASLEEP; // Change state **from ready to asleep**.
  - Enqueue(asleep_processes_queue, proc); // Enqueue in **asleep processes queue (decreasing order)**.
  - Proc = Scheduler();
- if ((current > sleepAfterTicks =< Ticks) && (current-> wakeupAfterTicks > Ticks))
  - current->state = ASLEEP; // Change state **from ready to asleep**.
  - Enqueue(asleep_processes_queue, current); // Enqueue in **asleep processes queue (decreasing order)**.
  - old_current = current;
  - current = Scheduler(); // ExtractFirstProcess (ready_processes_queue);
  - current->state = EXECUTION
  - Swap_ontext(old_current, current); // Activator (dispatcher)

Exercise
solution

1. Initial approach:
    1. Draw a diagram of initial system state
    2. Modify the diagram to incorporate the exercise requirements
2. Answer the proposed questions
3. Review the answers

Examples of exercises
Use case: sleepduring

ARCOS

Operating Systems Design

Degree in Computer Engineering

University Carlos III of Madrid