Clock

You need to implement a mechanical clock, the clock has 4 buttons A, B, C, D, and 3 states Alarm, Timer, and Stopper.
A - changes the state. The clock state changes in the following order:
Alarm→Timer→Stopper and go back to Alarm
B, C, D - Will do some functionality based on the clock state

Please implement the following class:

```
Class Clock{
doA{}
doB{}
doC{}
doD{}
}
```

Note: we don’t care about the actual functionality of B, C, and D in each state.

OS

You need to design and develop the sleep function of the OS.

You should implement the function sleep(Job job, long milliSeconds). When the OS calls this function with a job, it is added to a dedicated sleeping jobs queue (you need to design and implement it as well). At the right time, the OS should bring the job back into the running jobs queue from which the OS will take and execute it (you don't need to implement that part)

You should also implement the function awake(). This function is called by the OS every 1ms and is responsible for moving the ready to execute sleeping jobs to the running queue.

A. When you think about the design, think about performance issues. Which function should be more efficient and why?
B. Space complexity of the program should not exceed O(k), where ‘k’ is the amount of the jobs.
C. The OS does not have a clock function. You don't have access to the current time.
D. You cannot use a global static variable or any other counter method to simulate current time.

Good Luck!