

Real-Time Operating system

(RTOS)

Real-time

Embedded systems often have real-time computing constraints

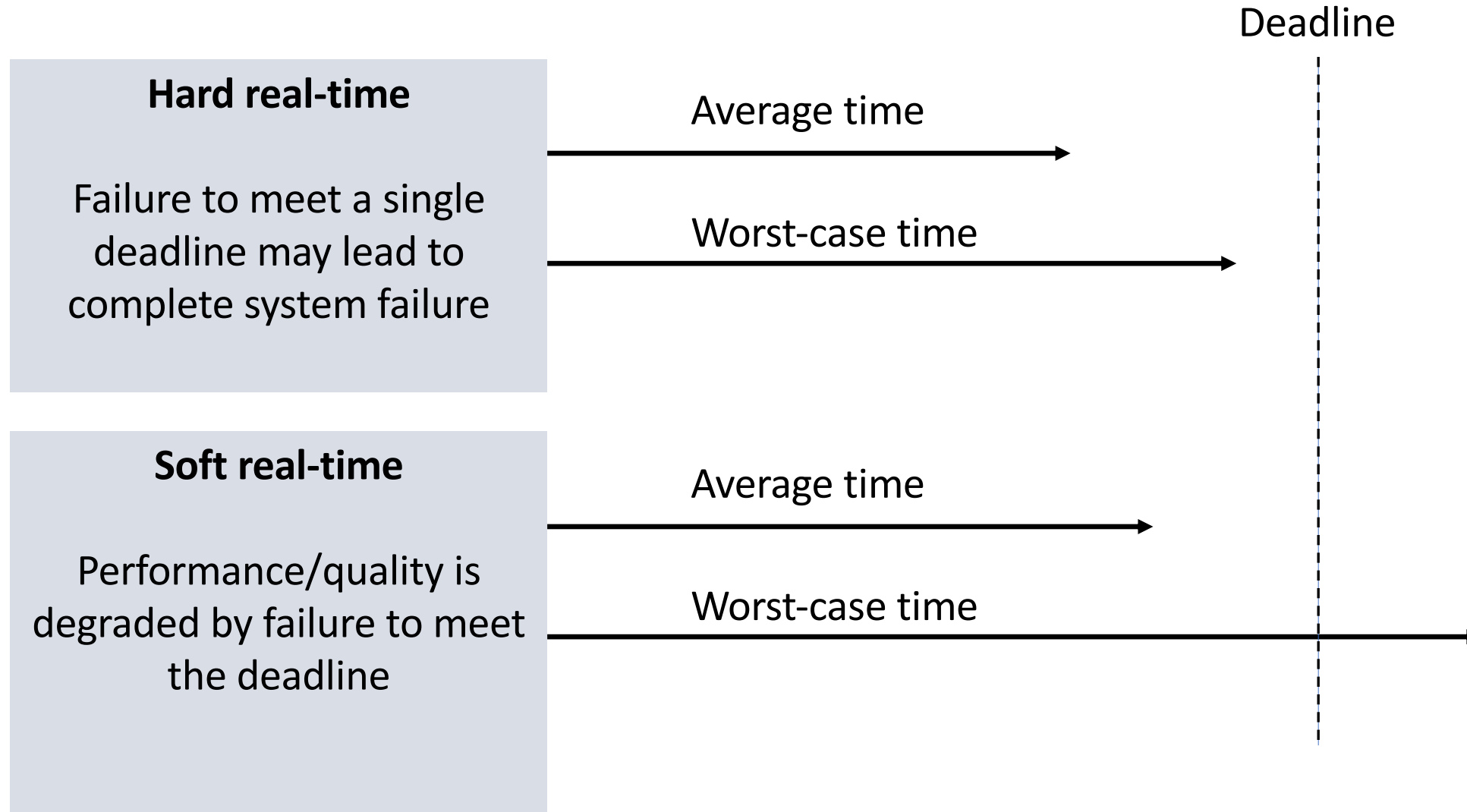


(Temporal) Determinism

Correctness of system depends not only on the logical result of the computation, but also on the time at which the result is generated



Real-time



Timing in embedded system

- Sleep / delay()

```
int main(){
    while(1){
        doImportantStuff();
        doStuff();
        usleep(10000);
    }
}
```

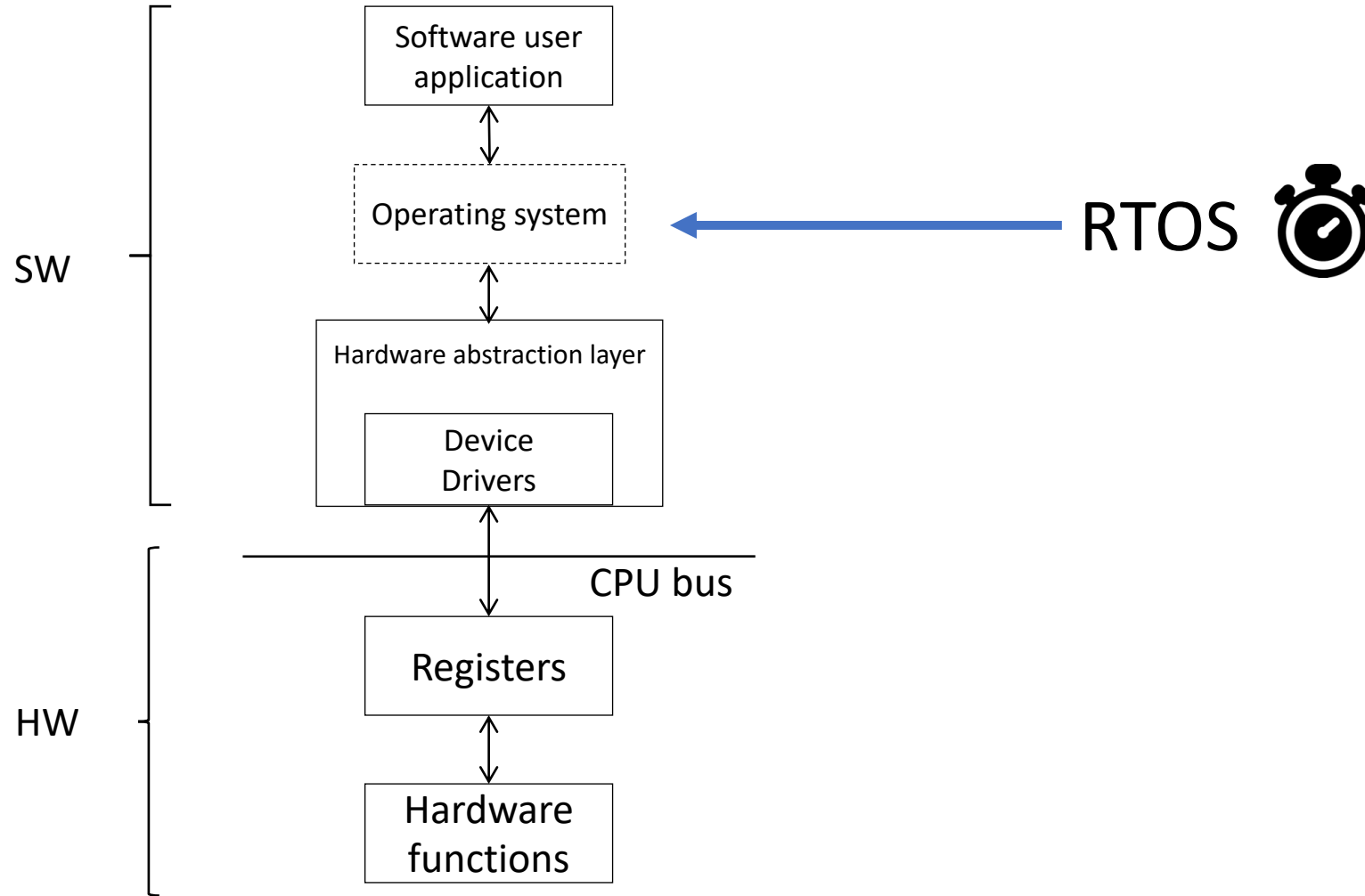
super loop

- Timer interrupts

```
int main(){
    setupTimerWithInterrupt();
    while(1){
        doStuff();
    }
}

void handleInterrupt(TIMER_INT){
    doImportantstuff();
}
```

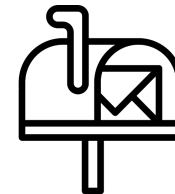
Abstraction levels



Real-Time Operating System (RTOS)

A structure that primarily provides:

- Time management
- Task management
- Inter-task communication



What does an RTOS do?

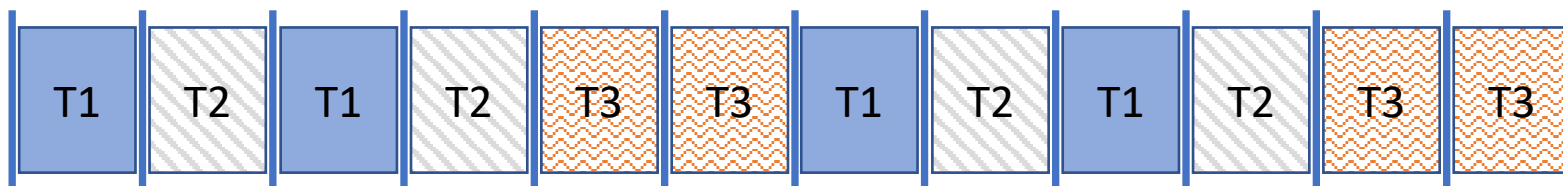
- Break time into uniform chunks (*ticks*)



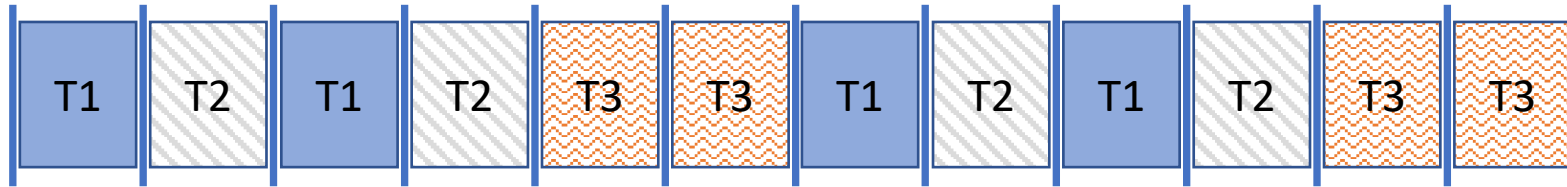
- Split a problem into smaller manageable units / subtask. (*tasks*)



- Assign time slots to these subtasks (*scheduling*)



Scheduler

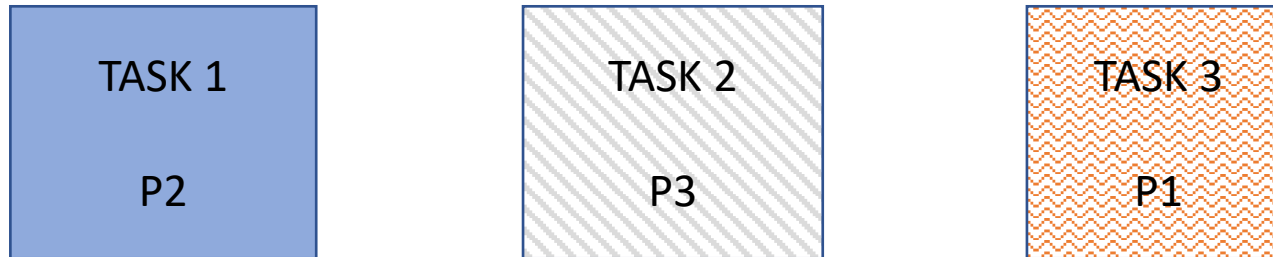


The part of the OS or kernel that is responsible for determining which task to run next

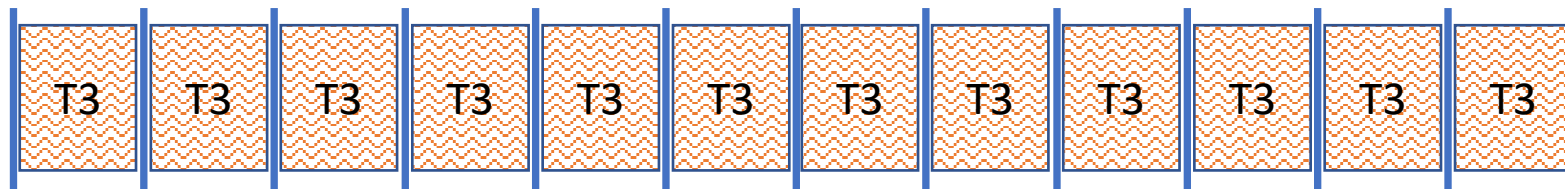
But how does it decide?

Priorities

Most RTOSs assigns an explicit priority to each task



The OS then runs the task with the highest priority



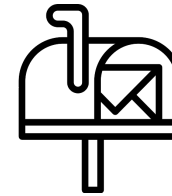
Blocking

delay(t)



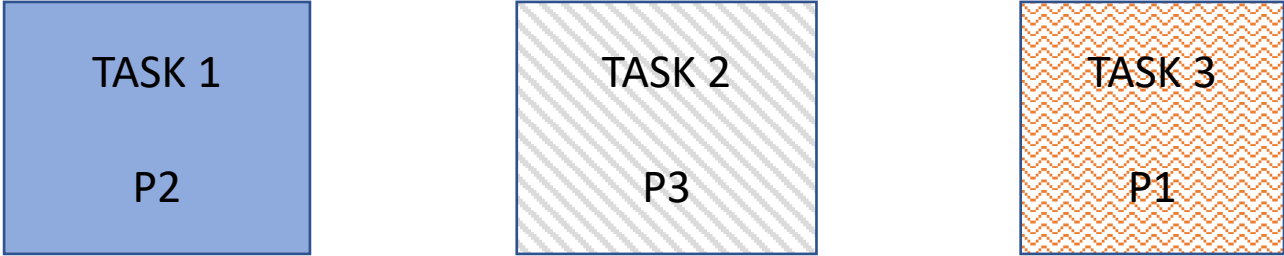
```
void task1(){  
    while(1){  
        doStuff();  
        delay(100);  
    }  
}
```

Inter-task communication

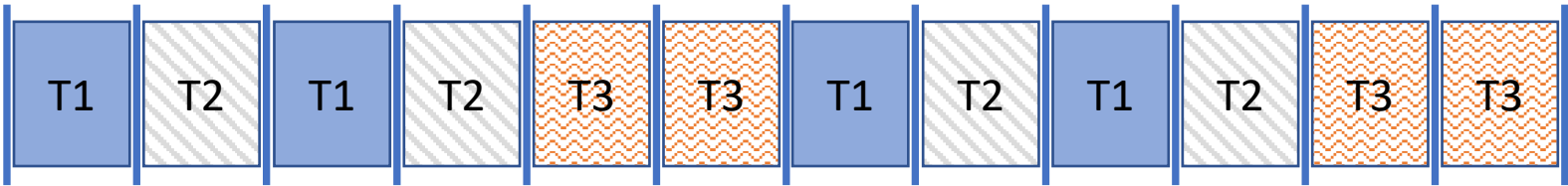


```
void task2(){  
    while(1){  
        waitForEvent(timeout);  
        doStuff();  
    }  
}
```

Priorities & blocking



The scheduler evaluates which task to run at every timer tick



The highest priority unblocked tasks always runs

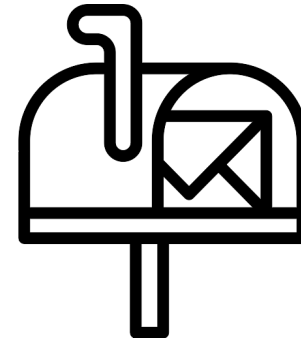
Inter-task communication

Semaphores



“Key” to control access, by multiple tasks, to a common resource

Messages (*mailbox, queues*)



Can be used to communicate data between tasks

Both can be used to synchronize the execution of tasks

Example from FYS4220 project

