Politecnico di Torino

Group 20

Sannia Gabriele - S331385
Sabella Mattia Luigi - S285363
Pittalis Domenico - S283602
Nobili Luca - S331461

Academic Year 2023/2024

COMPUTER ARCHITECTURE &
OPERATING SYSTEMS

HacklOSsim Project

Politecnico di Torino

FreeRTOS - Introduction

- Real-Time Operating System for microcontrollers
- Robustness, scalability
- Includes a kernel and a set of libraries
- Minimal ROM, RAM and processing overhead

Politecnico
di Torino

Tasks

- Real-time applications in FreeRTOS consist of independent tasks
- The Real-Time Scheduler decides which task runs based on priority
- Lower is the value, lower is the priority
- Tasks have no knowledge of the scheduler activity
  -> Correct Context Switch is only under the responsibility of scheduler
- Time-Sliced Round-Robin Scheduling may be used for tasks with identical priorities
- We created two simple demos for both schedulers.

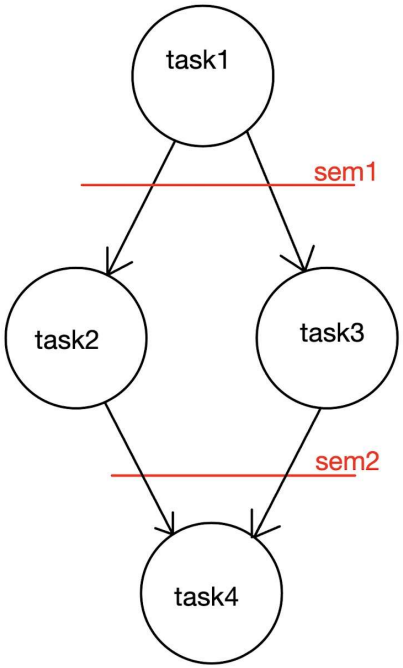
Politecnico
di Torino

Semaphores

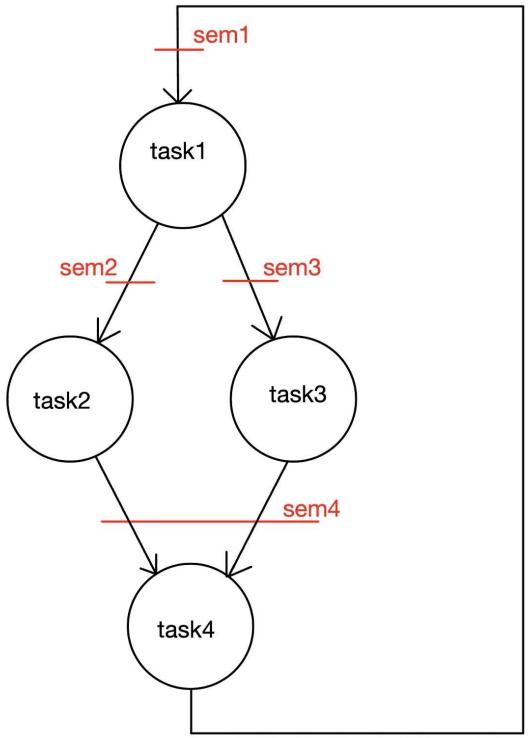
- Binary Semaphores - synchronization tasks.
  - Mutexes – simple mutual exclusion tasks.
  - Only difference: priority inheritance mechanism.

- Counting Semaphores are like queues with lengths greater than one.
- They are used for:
  - Event counting.
  - Resource management.

Semaphores Examples



Basic tasks
"semaphoresAcyclic.c"

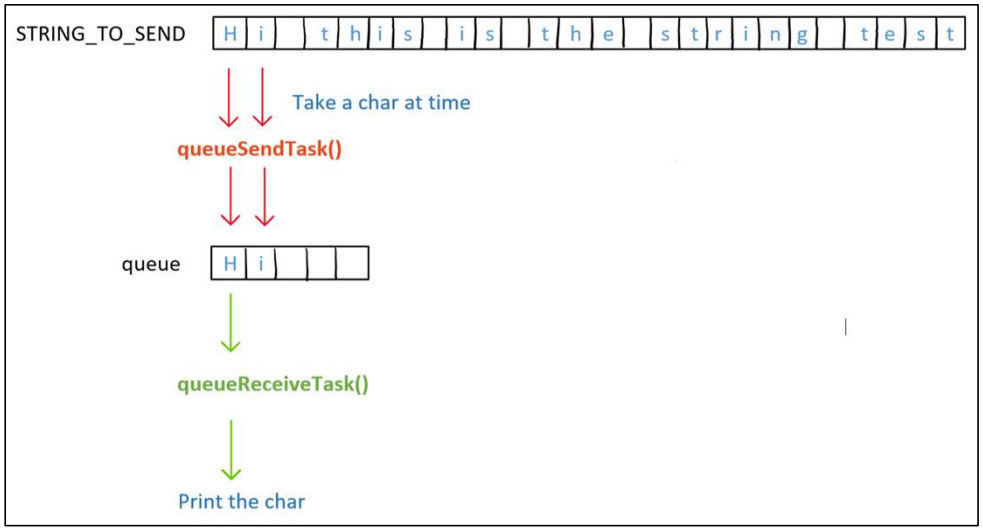


Extended tasks
"semaphoresCyclic.c"

Politecnico di Torino

Queues

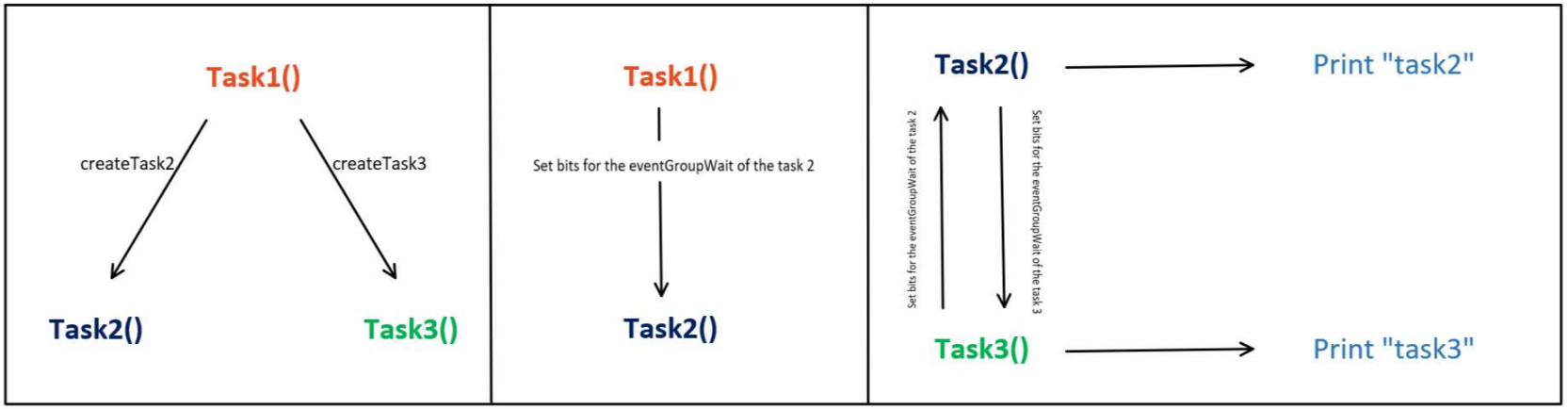
- Queues are essential for task communication in a multitasking system.
- They serve as channels through which tasks exchange messages, facilitating coordination and data sharing.
- Queues operate on a FIFO basis, where the data sent earlier is received first.
- Queues can block tasks when they are full or empty, ensuring efficient resource utilization.



Politecnico di Torino

EventGroup

- EventBits are individual bits used to represent different event states, such as message reception or task readiness.
- Event groups are collections of these event bits grouped together for easier management and task synchronization.
- Used for synchronization between tasks and managing multiple event states.



Politecnico di Torino

Earliest Deadline First with Background Scheduling

Politecnico di Torino

Earliest Deadline First Scheduling

- Dynamic priority scheduler
- Preemptive
- At each instant, the task with earliest deadline will receive highest priority
- handles periodic tasks

Backgroung Scheduling

- Handles aperiodic tasks
- Aperiodic tasks are scheduled in background, when no periodic instance is ready

How to implement?

- ❑ Task Control Block modification
- ❑ New task create function
- ❑ New ready list
- ❑ Deadline & Priority updates
- ❑ Preemption management
- ❑ Context Switching management
- ❑ IDLE task management
- ❑ Aperiodic Task Management

Politecnico
di Torino

Task Control Block modification

- A variable to store the deadline si added to the TCB.

```
TickType_t xTaskDeadline
```

New task create function

- A new task creation function, **xTaskCreateEDF** , has been introduced to accommodate the deadline parameter

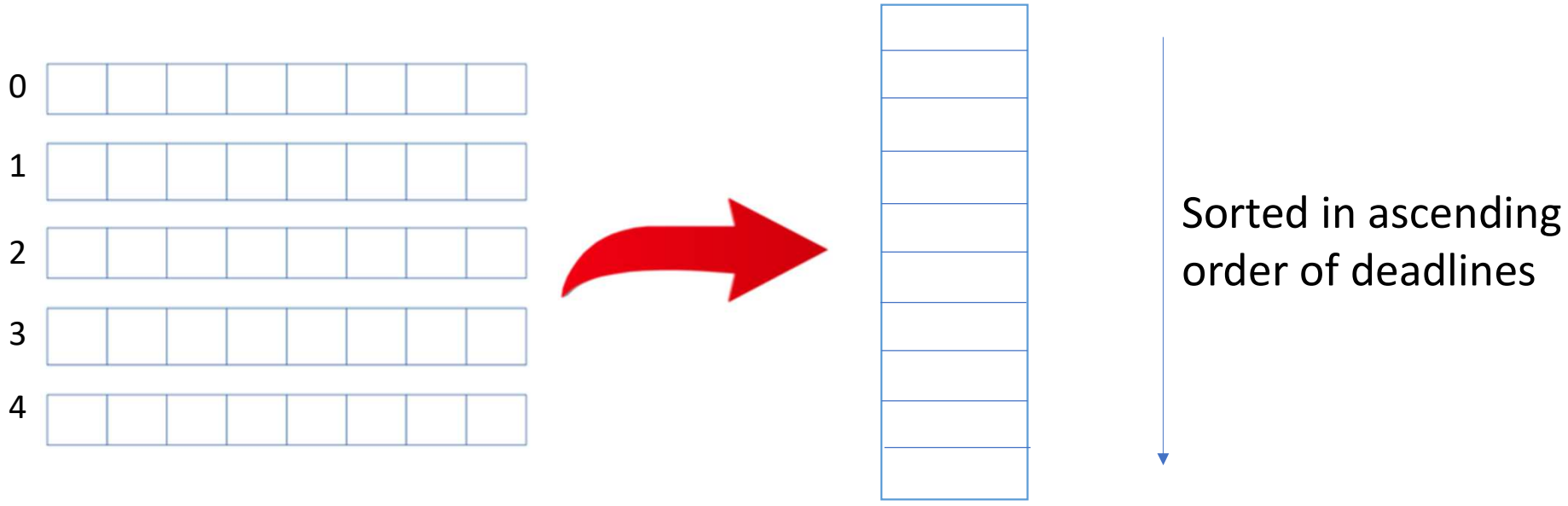
```
xTaskCreateEDF( TaskFunction_t pxTaskCode,…,
                TickType_t deadline1)
```

- Inside it, the value passed to the function is added to the TCB.

```
pxNewTCB->xTaskDeadline = deadline1;
```

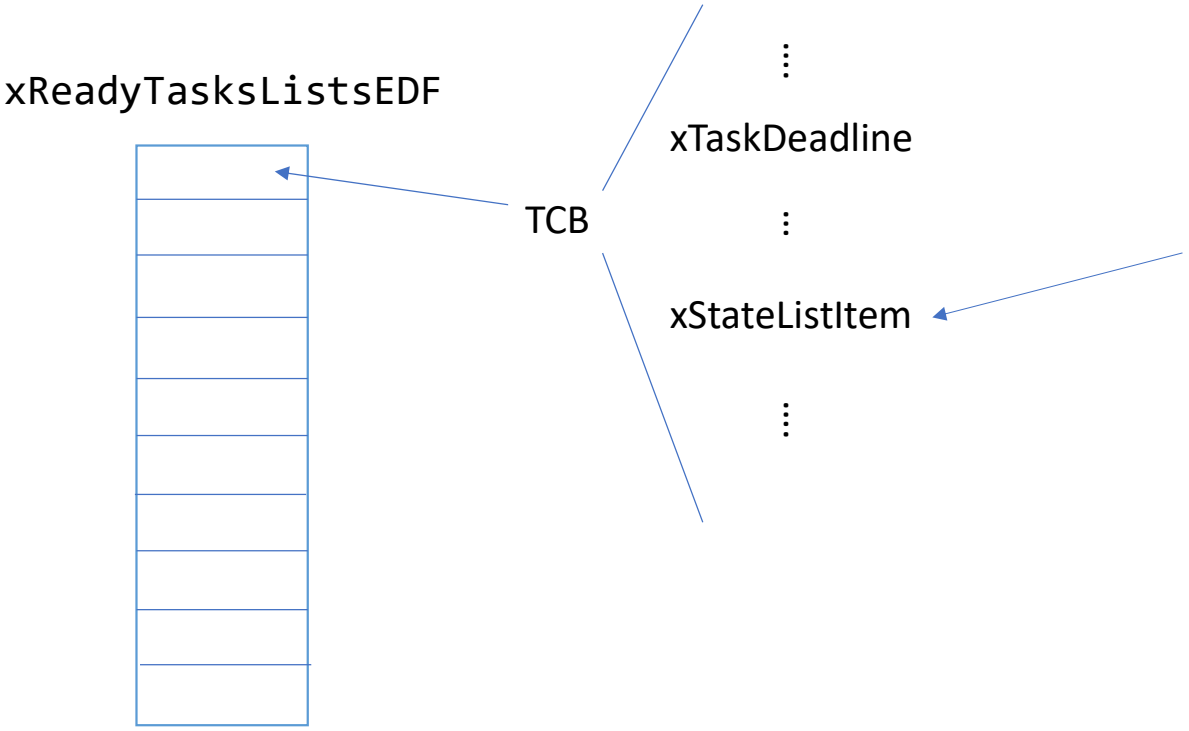
Politecnico di Torino

New Ready List



Politecnico di Torino

How to use the EDF ready list



List used internally by FreeRTOS to manage task state information. Contains an item, that is used as deadline in our scheduler.

Politecnico di Torino

Deadline & Priority updates

- The prvAddTaskToReadyList function is modified to update task deadlines before the insertion into the ready list.
- The deadline is recalculated as the sum of the previous deadline and the declared deadline value in the TCB.

```
( pxTCB )->xStateListItem.xItemValue += ( pxTCB )->xTaskDeadline;
```

Preemption management

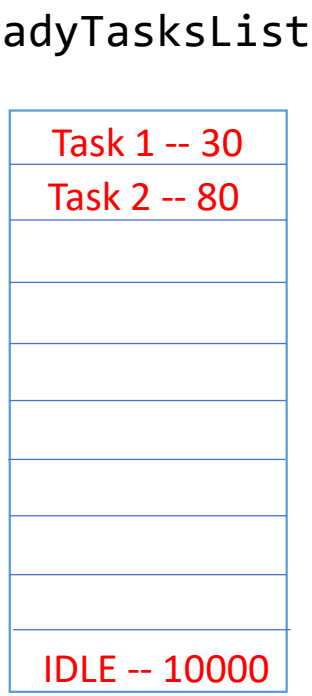
- Within the `xTaskIncrementTick` function, a new preemption policy is introduced.
- When a task scheduled for insertion into the ready list has an earlier deadline than the currently running task, preemptive action is taken.
- To reduce context switching overhead, if both tasks share the same deadline, preemptive action is not executed.

Context Switching management

- The task to be executed is positioned at the head of this list.
- The function listGET_OWNER_OF_HEAD_ENTRY is used to obtain the task in the head of the list.

IDLE task management

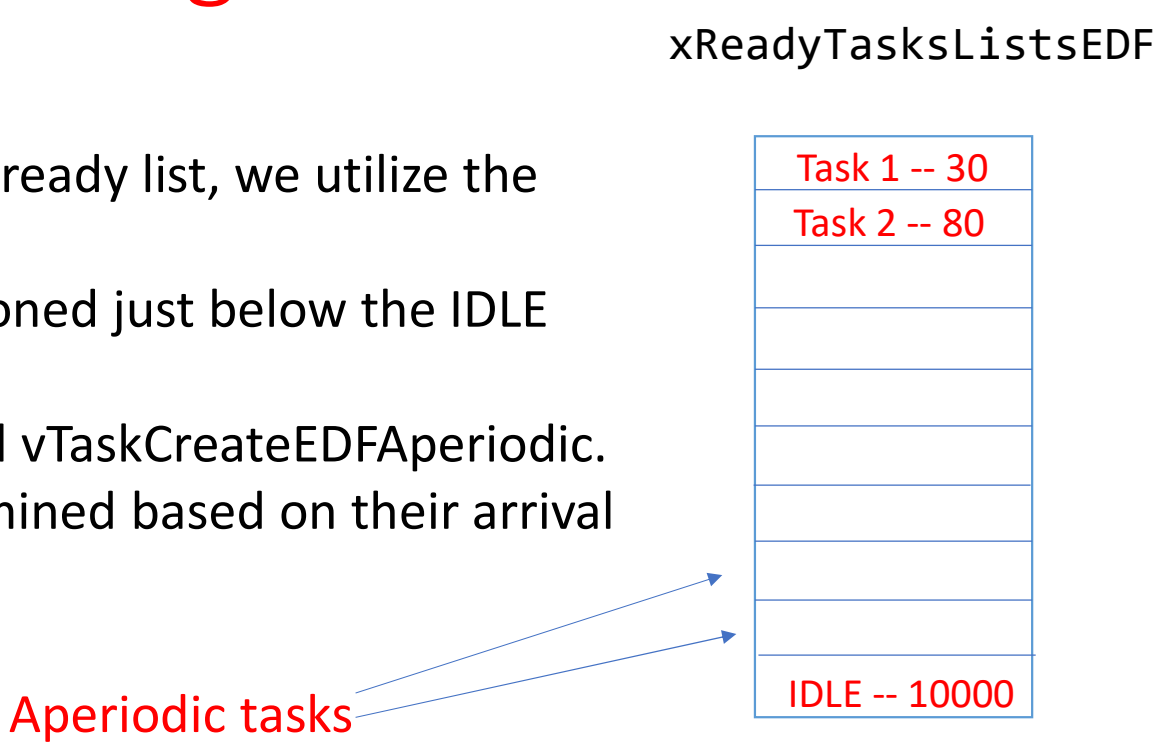
- IDLE task is initialized with vTaskCreateEDF
- It simulates lowest priority by assigning farthest deadline, e.g., 10000.



Politecnico di Torino

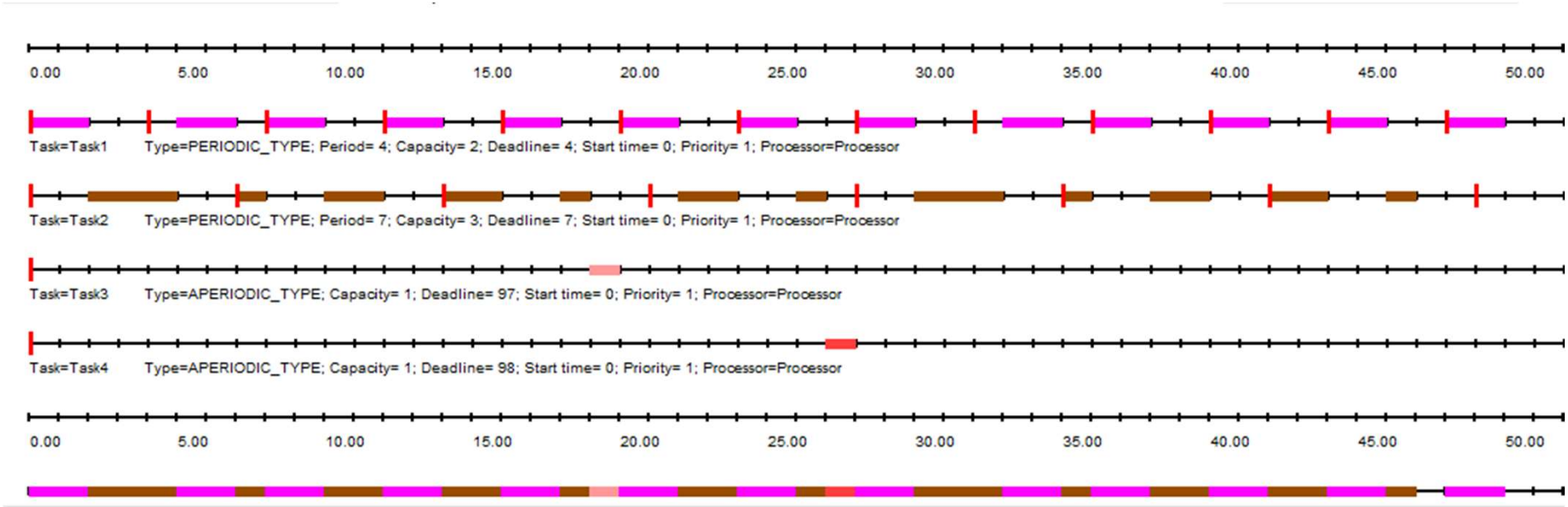
Aperiodic task management

- Instead of creating a new ready list, we utilize the xReadyTasksListsEDF.
- Aperiodic tasks are positioned just below the IDLE task's deadline.
- We've devised a dedicated vTaskCreateEDFAperiodic.
- Their deadlines are determined based on their arrival order.



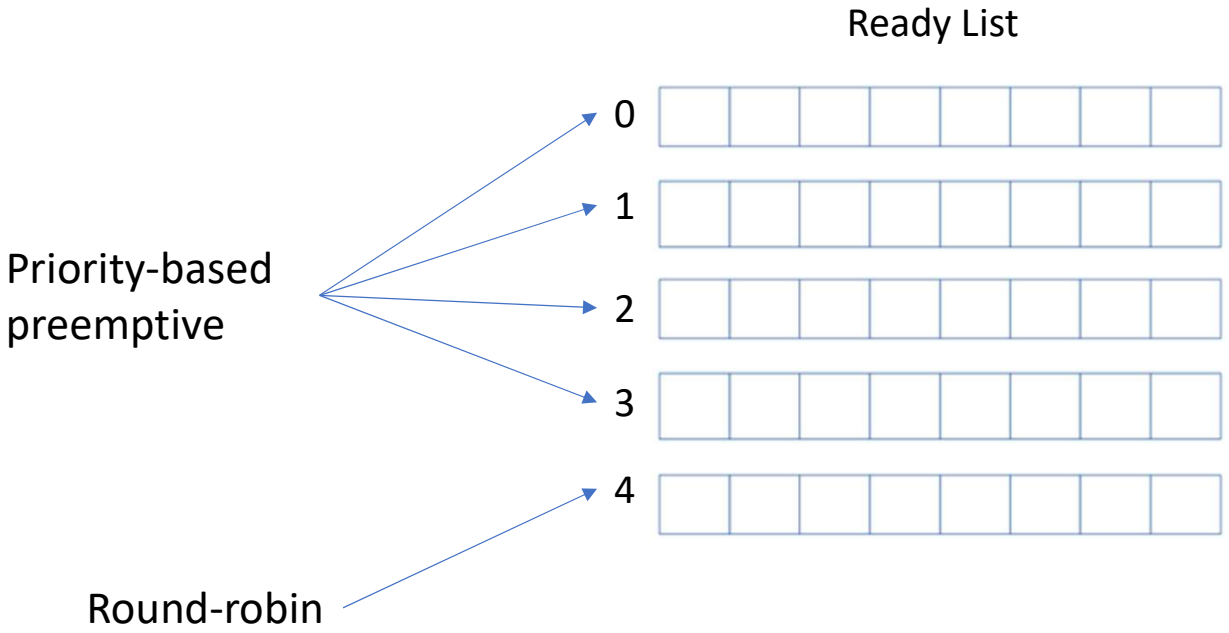
Politecnico di Torino

Demo



Politecnico di Torino

MultiLevel Queue Scheduler



Politecnico di Torino

How to implement?

- Task Control Block modification
- xTaskPrioritySet
- Priority upgrade
- Highest Priority List Round Robin

Task control block modification

- Addition of uxOldPriority Variable.

- Inclusion of cc Variable:
  - Reserved for internal purpose.

```
int cc
UBaseType_t uxOldPriority
```

Those values are then setted in the vTaskCreate function.

xTaskPrioritySet

- A smaller version version of the **vTaskPrioritySet** is added to properly modify the priority of a task.

Priority upgrade

- When system's ticks reach a multiple of the "update rate", a priority boost is applied.
- This boost affects all tasks in the ready list except the one currently running.
- To modify the priority, the `xTaskPrioritySet` is used.

Politecnico di Torino

Highest priority list Round-Robin

- The `xTaskIncrementTick` function containes the code for our Round-Robin scheduling.
- Activated if there are at least two tasks present in the highest priority list.
- This mechanism ensures fair execution among tasks sharing the same highest priority level.

```
UBaseType_t c = (configMAX_PRIORITIES - 1U);
if( listCURRENT_LIST_LENGTH( &( pxReadyTasksLists[ c ] )) > 1)
{
    xSwitchRequired = pdTRUE;
}
```

Demo