Scheduling Internals

Scheduling is performed by two functions: \texttt{OSSched()} and \texttt{OSIntExit()}. \texttt{OSSched()} is called by task level code while \texttt{OSIntExit()} is called by ISRs. Both functions are found in \texttt{os_core.c}.

The figure below illustrates the two sets of data structures that the scheduler uses; the priority ready bitmap and the ready list as described in The Ready List.

![Figure - Priority ready bitmap and Ready list](image)

**OS\_SchedRoundRobin()**

When the time quanta for a task expires and there are multiple tasks at the same priority, \mu\text{C}/OS-III will select and run the next task that is ready-to-run at the current priority. \texttt{OS\_SchedRoundRobin()} is the code used to perform this operation. \texttt{OS\_SchedRoundRobin()} is either called by \texttt{OSTimeTick()} or \texttt{OS\_IntQ\_Task()}. \texttt{OS\_SchedRoundRobin()} is found in \texttt{os_core.c}.

\texttt{OS\_SchedRoundRobin()} is called by \texttt{OSTimeTick()} if you selected the Direct Method of posting (see Interrupt Management). \texttt{OS\_SchedRoundRobin()} is called by \texttt{OS\_IntQ\_Task()} if you selected the Deferred Post Method of posting, described in Chapter 8.
The pseudo code for the round-robin scheduler is shown in Listing 7-3.

```c
void OS_SchedRoundRobin (void)
{
    if (OSSchedRoundRobinEn != TRUE) {
        return; (1)
    }
    if (Time quanta counter > 0) {
        Decrement time quanta counter;
    }
    if (Time quanta counter > 0) {
        return;
    }
    if (Number of OS_TCB at current priority level < 2) {
        return; (3)
    }
    if (OSSchedLockNestingCtr > 0) { (4)
        return;
    }
    Move OS_TCB from head of list to tail of list; (5)
    Reload time quanta for current task; (6)
}
```

Listing - OS_SchedRoundRobin() pseudocode

(1) OS_SchedRoundRobin() starts by making sure that round robin scheduling is enabled. Recall that to enable round robin scheduling, your code must call OSSchedRoundRobinCfg().

(2) The time quanta counter, which resides inside the OS_TCB of the running task, is decremented. If the value is still non-zero then OS_SchedRoundRobin() returns.

(3) Once the time quanta counter reaches zero, OS_SchedRoundRobin() checks to see that there are other ready-to-run tasks at the current priority. If there are none, the function returns. Round robin scheduling only applies when there are multiple tasks at the same priority and the task doesn’t completes its work within its time quanta.

(4) OS_SchedRoundRobin() also returns if the scheduler is locked.

(5) Next, OS_SchedRoundRobin() moves the OS_TCB of the current task from the head of the ready list to the end.

(6) The time quanta for the task at the head of the list is loaded. Each task may specify its own time quanta when the task is created or through OSTaskTimeQuantaSet(). If you set the task time quanta to 0 then µC/OS-III assumes the default time quanta, which
corresponds to the value in the variable OSSchedRoundRobinDfltTimeQuanta.