FreeRTOS is designed to be small and simple. The kernel itself consists of only three C files. To make the code readable, easy to port, and maintainable, it is written mostly in C

Free RTOS: Real Time Operating System:

The kernel: consists of only three C files -> to make the code readable, easy to port, and maintainable, it is also written C, but there are a few assembly functions includedwhere needed.

Free RTOS provides methods for multiple threads or tasks, mutexes, semaphores and software timers. A tick-less mode -> provided for low power applications. Thread priorities are supported. Free RTOS provides methods for multiple threads or tasks, mutexes, semaphores and software timers. A tick-less mode -> provided for low power applications. Thread priorities are supported.

|  |
| --- |
| * allocate only; * allocate and free with a very simple, fast, algorithm; * a more complex but fast allocate and free algorithm with [memory coalescence](https://en.wikipedia.org/wiki/Coalescence_(computer_science)); * an alternative to the more complex scheme that includes memory coalescence that allows a heap to be broken across multiple memory areas. * and C library allocate and free with some mutual exclusion protection. |
|  |

FreeRTOS implements multiple threads by having the host program call a thread tick method at regular short intervals.

The thread tick method switches tasks depending on priority and a round-robin scheduling scheme. The usual interval is 1/1000 of a second to 1/100 of a second, via an interrupt from a hardware timer, but this interval is often changed to suit a particular application.

|  |
| --- |
|  |