Trong Zephyr RTOS, một thread (luồng) là một đơn vị cơ bản của việc thực thi, tương ứng với một tác vụ cụ thể. Mỗi thread có một ưu tiên (priority) để quyết định thứ tự thực thi trong hệ thống. Ưu tiên càng cao, thread sẽ được ưu tiên thực thi trước. Điều này đảm bảo rằng các tác vụ quan trọng hoặc yêu cầu thời gian thực sẽ nhận được sự ưu tiên cao hơn.

Scheduler (lập lịch) trong Zephyr RTOS là thành phần quan trọng quản lý việc lựa chọn thread nào được thực thi và khi nào. Zephyr hỗ trợ hai phương pháp lập lịch chính: Cooperative Scheduling (lập lịch hợp tác) và Preemptive Scheduling (lập lịch ưu tiên). Lập lịch hợp tác cho phép các thread tự ý "nhường" CPU cho các thread khác, trong khi lập lịch ưu tiên cho phép hệ điều hành gián đoạn (preempt) thread hiện tại và chuyển đổi sang thread ưu tiên cao hơn.

Semaphore (ngăn chặn) là một cơ chế trong Zephyr RTOS được sử dụng để đồng bộ hóa quyền truy cập vào tài nguyên chia sẻ. Semaphore có thể được xem như một cờ hiệu đơn giản, cho phép một thread giữ hoặc giải phóng tài nguyên để các thread khác có thể sử dụng. Semaphore cung cấp cơ chế an toàn và đồng bộ trong việc truy cập tài nguyên chia sẻ.

Mutex (khóa) là một loại semaphore đặc biệt trong Zephyr RTOS. Mutex giúp đảm bảo rằng chỉ có một thread được phép truy cập vào tài nguyên chia sẻ vào một thời điểm nhất định. Khi một thread đã lấy được mutex, các thread khác phải chờ đợi cho đến khi mutex được giải phóng trước khi có thể truy cập tài nguyên.

Với sự hỗ trợ của thread, priority, scheduler, semaphore và mutex, Zephyr RTOS cung cấp một môi trường lập trình mạnh mẽ để phát triển các ứng dụng nhúng thời gian thực. Bằng cách sử dụng các khái niệm này, lập trình viên có thể xây dựng các hệ thống nhúng đáp ứng yêu cầu cao về độ tin cậy và thời gian thực.