

Semaphores and VxWorks Events

Semaphores can send VxWorks events to a specified task when they become free. For more information, see [3.3.7 VxWorks Events](#), p. 121.

3

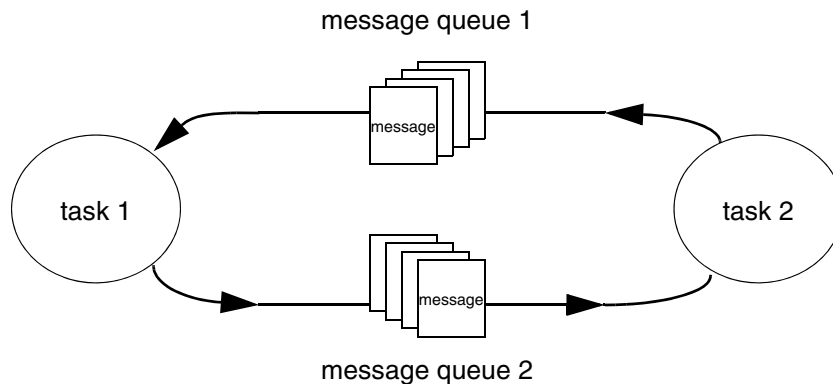
3.3.5 Message Queues

Modern real-time applications are constructed as a set of independent but cooperating tasks. While semaphores provide a high-speed mechanism for the synchronization and interlocking of tasks, often a higher-level mechanism is necessary to allow cooperating tasks to communicate with each other. In VxWorks, the primary intertask communication mechanism within a single CPU is *message queues*.

For information about socket-based message communication across memory spaces (kernel and processes), and between multiple nodes, see [3.3.8 Message Channels](#), p. 127.

Message queues allow a variable number of messages, each of variable length, to be queued. Tasks and ISRs can send messages to a message queue, and tasks can receive messages from a message queue.

Figure 3-14 Full Duplex Communication Using Message Queues



Multiple tasks can send to and receive from the same message queue. Full-duplex communication between two tasks generally requires two message queues, one for each direction; see [Figure 3-14](#).