

13.1 同时等待多个内核对象

UCOSIII 中一个任务可以同时等待任意数量的信号量或者消息队列，当只要等到其中的任意一个的时候就会导致该任务进入就绪态，如图 13.1.1 所示。

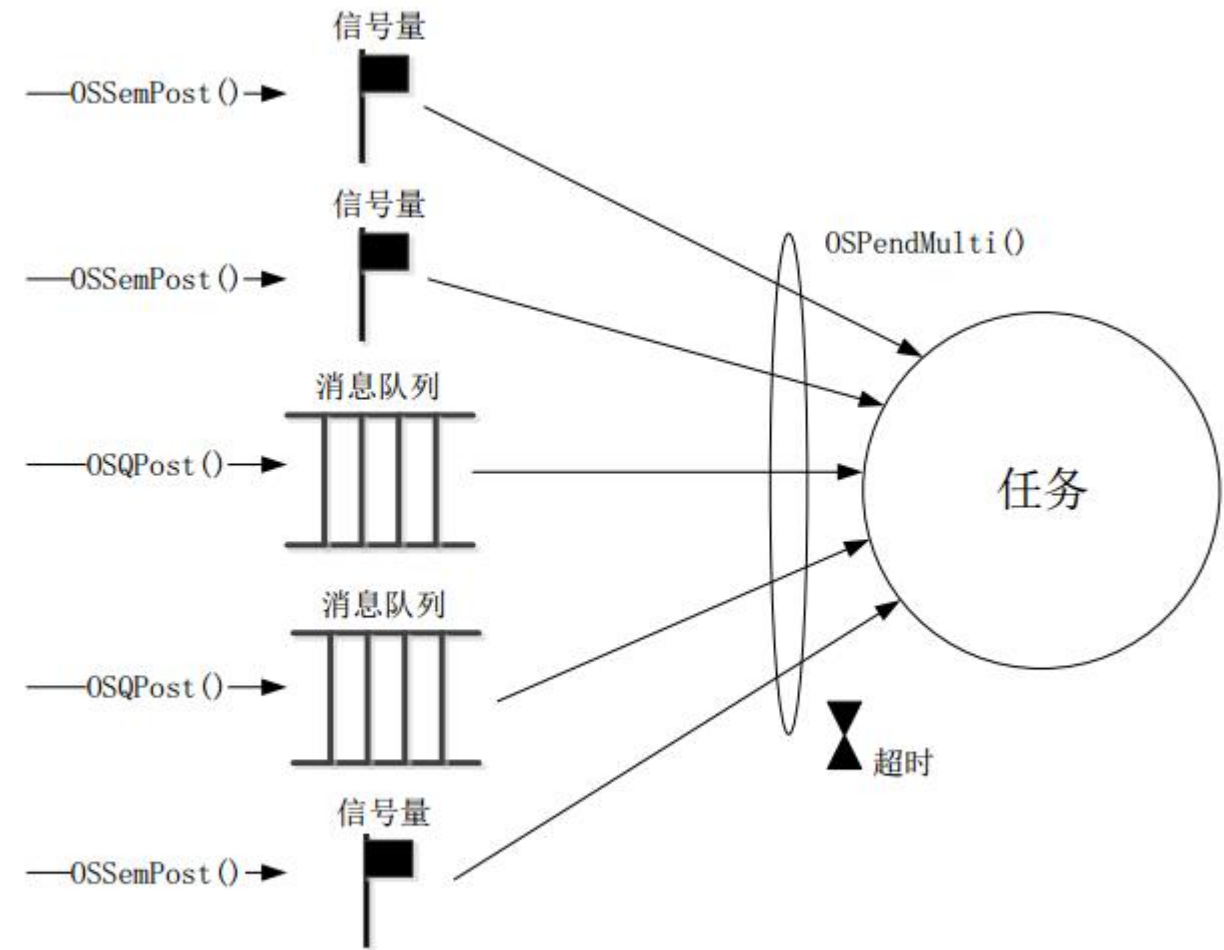


图 13.1.1 任务等待多个内核对象

在图 13.1.1 中任务通过调用函数 OSPendMulti()来等待多个内核对象，我们可以设定一个等待超时值，如果在指定的时间内没有一个内核对象被发布，那么将返回一个错误码，表示等待超时。

13.2 OSPendMulti()函数

函数 OSPendMulti()用来等待多个内核对象，调用 OSPendMulti()时，如果这些对象中有多个可用，则所有可用的信号量和消息都将返回给调用者，如果没有任何对象可用，则 OSPendMulti()将挂起当前任务，直到以下任一情况发生：

- 1、对象变为可用。
- 2、到达设定的超时时间。
- 3、一个或多个任务被删除或被终止。
- 4、一个或多个对象被删除。

如果一个对象变为可用，并且有多个任务在等待这个对象，则 UCOSIII 将恢复优先级最高的那个任务，函数 OSPendMulti()原型如下：

```
OS_OBJ_QTY OSPendMulti ( OS_PEND_DATA  *p_pend_data_tbl,
                        OS_OBJ_QTY      tbl_size,
                        OS_TICK          timeout,
                        OS_OPT           opt,
                        OS_ERR           *p_err)
```

p_pend_data_tbl: 指向 OS_PEND_DATA 表的指针，调用者通过该表来查询函数的调用结果。调用该函数的时候首先必须初始化 OS_PEND_DATA 表中的每个元素的 PendObjPtr，使得各个指针指向被等待的对象。

tbl_size: 表 p_pend_data_tbl 的大小，也就是所等待的内核对象数量。

timeout: 设定一个等待超时值(时钟节拍数)，用来设置任务等待对象发送的时间，如果为 0，表示这个任务将一直等待下去，直到对象被发送。

opt: 来选择是否使用阻塞模式，有两个选项可以选择。

OS_OPT_PEND_BLOCKING 如果没有任何消息存在的话就阻塞任务，一直等待，直到接收到消息。

OS_OPT_PEND_NON_BLOCKING 如果消息队列没有任何消息的话任务就直接返回。

p_err: 用来保存调用此函数后返回的错误码。