

UNIT V

EMBEDDED SYSTEM APPLICATION DEVELOPMENT

5.4 Memory Allocation Related Functions

OS Mem Create

OS Mem Put (*mem CBPointer,

*memErr)OSMemGet (*memCB Pointer,

memErr) OS Mem Query(mem CB Pointer,*
mem Data)

Semaphore Functions

Provides for using same semaphore functions as an event signaling flag or mute or counting semaphore.

OS Sem Create (semVal)

OS SemP end (*event Pointer, time Out,*Sem Err

Pointer) OS Sem Accept (*eventPointer)

OS Sem Post(*event Pointer)

OS Sem (*event Pointer)

Mail box Functions

- Used to communicate a pointer for information.

- OS- II permits one message – pointer per mail box.
- At the pointer there can be a string or data structure of no size limit.
- Assume an event pointer to the mail box=`* mbox msg` ,
- Pointer to the message,`* Msg Pointer` (for retrieving the message itself).

OS Mbox Create(`* mbox Msg`)

To create a mail box message pointer ECB of a mail box message.

OS Mbox Pend(`*mbox Msg, timeout,*Mbox Err`)

To check if mail box message not pending (available) then read `* mbox Msg` is and empty mail box[`*mbox Msg=NULL` again].If message is not available [`*mbox Msg` points to NULL],then wait ,suspend the task (block further running) till `*mbox Msg` not Null or timeout.

OSMboxAccept(`*mboxMsg`)

To check if mail box message at the`*Msg Pointer`, is available at `*mboxMsg`.

Un like OSM box Pend function, it does not block (suspend) the task if message is not available .If available, it returns the pointer.

OSM box Post(`* mbox Msg,*Msg Pointer`)

– Sends a message of task at address `Msg Pointer` by posting the address pointer to them box `Msg`.

– If box is already full(`*mbox Msg` not Null),then the message is not placed and error status sent.

OSM box Query(`*mbox Msg,*mbox Data`)

– To get mail box error information

– Pointer Null or Not Null,

Queue Functions

- The message pointers post into a queue by the tasks either at the back as in a *queue* or at the front as in a *stack*.
- A task can thus insert a given message for deleting either in the *first in first out* (FIFO) mode or in *priority* mode for priority message.
- Assume pointer, `** Q top`, to a queue of pointers for the messages and
- Assume two pointers,`*Q front Pointer` and `*Q back Pointer` to insert (post) and delete (retrieve) respectively the pointer of the message.

Functions

OSQ Create (`**QTop, qSize`) OSQ

Post(`*QMsg Pointer,*QMsg`)

OSQPostFront (`*QMsgPointer, *QMsg`)OSQPend (`*QMsgPointer, timeOut, *Qerr`)OSQFlush(`*QMsgPointer`)

OSQQuery(`*QMsgPointer,*QData`)

IPC Objects

- For which there is waiting process or thread
- A process (thread or scheduler, task or ISR) generates some information by or value and sends event or semaphore or message into queue or a single IPC or multiple objects as output so that it lets another process waiting for that object in order to take note or use the object.
- A process waits for an IPC or object (s)in order to take note or use the object(s)

IPC Object Functions

Wait For Single Object

Wait For Multiple