## Holding and Releasing Memory

You can use the **HoldMemory** function to make a portion of the address space resident in physical memory and ineligible for paging. This function is intended primarily for use by drivers that access user data buffers at interrupt level, whether transferring data to or from them. Calling **HoldMemory** on the appropriate memory ranges thus prevents them from causing page faults at interrupt level. The contents of the specified range of virtual addresses can move in physical memory, but they are guaranteed always to be in physical memory when accessed.

> **Note:** If you use the device-level **PBRead** and **PBWrite** functions when doing data transfers, the Operating System automatically ensures that the data buffers are held down before the transfer of data.

The following sample code instructs the Operating System to hold in RAM the range of memory that starts at address 0x32500 and that is 8192 bytes long.

myAddress = 0x32500; myLength = 8192; myErr = **HoldMemory** (myAddress, myLength);

Note that holding is applied to whole pages of the virtual address space, regardless of the starting address and length parameters you supply. If the starting address parameter supplied to the **HoldMemory** function is not on a page boundary, then it is rounded down to the nearest page boundary. Similarly, if the specified range does not end on a page boundary, the length parameter is rounded up so that the entire range of memory is held. This rounding might result in the holding of several pages of physical memory, even if the specified range is less than a page.

To release memory that was held down using **HoldMemory**, you must use the **UnholdMemory** function, which simply reverses the effects of the HoldMemory function. For example, the pages held in memory in the previous example can be released as follows:

myErr = **UnholdMemory** (myAddress, myLength);

As with holding, letting go is applied to whole pages of the virtual address space. Similar rounding of the address and length parameters is performed as required to keep the range on page boundaries.

> **Note:** The system heap is always held in memory and is never paged out.