

Locking and Unlocking Memory

You can use the **LockMemory** function to make a portion of the address space immovable in physical memory and ineligible for paging. The Operating System may move the contents of the specified range of logical addresses to a more convenient location in physical memory during the locking operation, but on completion, the contents of the specified range of logical addresses are resident and do not move in physical memory.

LockMemory is used by drivers and other code when hardware other than the Macintosh CPU is transferring data to or from user buffers, such as any NuBus master peripheral card or DMA hardware. This function prevents both paging and physical relocation of a specified memory area and enables the physical addresses of a memory area to be exported to the non-CPU hardware. Typically, you would use this service for the duration of a single I/O request. However, you could use this service to lock data structures that are permanently shared between driver or other code and a NuBus master.

Note: Do not confuse locking addresses in RAM (using **LockMemory**) with locking a handle (using **HLock**). A locked handle can still be paged out.

The main reason to disable movement of pages in physical memory is to translate virtual addresses to physical addresses. This translation is needed by bus masters, which must write to memory in the physical address space. To avoid stale data, the memory locked in RAM is marked noncacheable in the CPU cache.

You can lock a range of memory into contiguous physical memory by using the **LockMemoryContiguous** function. This function is used by driver and NuBus master or driver and DMA hardware combinations when a non-CPU device accessing memory is unable to handle physically discontinuous data transfers. You can also use this service when it is possible to handle physically discontinuous data transfers, but doing so causes performance degradation. However, the call to **LockMemoryContiguous** may be expensive because entire pages might need to be copied in order to make a range contiguous.

Note: It might not be possible to make a range physically contiguous if any of the pages contained in the range are already locked. Because a call to **LockMemoryContiguous** is not guaranteed to return the desired results, all code that uses **LockMemoryContiguous** must have an alternate method for locking the necessary ranges of memory.

To unlock a range of previously locked pages, use the **UnlockMemory** function. This function reverses the effects of **LockMemory** or **LockMemoryContiguous**. Unlocked pages are marked as cacheable.

Locking, contiguous locking, and unlocking operations are applied to ranges of the logical address space. If necessary to force the ranges onto page boundaries, the Operating System performs rounding of addresses and sizes, as described in **Holding and Releasing Memory**.