Functions

This section describes three types of functions and macros:

- General-purpose functionsÐto perform basic tasks
- Driver Debugging Module (DDM) functionsÐto help all drivers keep debugging information
- Miscellaneous functionsÐsuch as DMA alignment macros, functions that work only in the kernel, and functions specific to a particular machine architecture.

Unless noted otherwise, all of the functions described in this section work in both user-level and kernel-level drivers.

Other Functions Available to Drivers

Almost all Mach functions are available to kernel-level device drivers. If you don't find the appropriate functionality in a method or function, you can use a Mach function. For example, **port_allocate()** and **msg_send()** are used by many drivers.

Note: Instead of including the header file **mach/mach.h**, you must include **mach/mach_user_internal.h** and **mach/mach_interface.h**.

The **host_priv_self()** Mach function does *not* work in the kernel. You should use **IOHostPrivSelf()** instead.

General-Purpose Functions

The general-purpose functions, defined in the header file **driverkit/generalFuncs.h**, provide a consistent interface for device drivers that may have to run in kernel space at one time (or in one configuration) and in user space at another time. Using these functions minimizes the work or porting between the two environments. All the Driver Kit classes, as well all NeXT kernel-level drivers that use the Driver Kit, were written using these functions so that they have one set of source files with minimal kernel and user mode differences.

Warning: Before using any of the general-purpose functions, each user-level driver must call **IOInitGeneralFuncs()**. (Kernel-level drivers don't need to call it.)

Thread Functions

These functions provide the functionality of the C-thread functions in a uniform way in both user and kernel space.

IOForkThread()
IOSuspendThread()
IOResumeThread()
IOExitThread()

Timer Functions

IOSleep()
IODelay()
IOScheduleFunc()
IOUnscheduleFunc()
IOGetTimestamp()

Memory Allocation and Copying Functions

IOCopyMemory()
IOMalloc()
IOFree()

Miscellaneous General-Purpose Functions

IOInitGeneralFuncs()
IOFindNameForValue()
IOFindValueForName()
IOLog()
IOPanic()

Driver Debugging Module (DDM) Functions

See the ªAdding Debugging Codeº section in Chapter 2 for information on using the DDM.

IOAddDDMEntry()
IOClearDDM()
IOCopyString()
IODEBUG()
IOGetDDMEntry()
IOGetDDMMask()
IOInitDDM()
IONsTimeFromDDMMsg()
IOSetDDMMask()

Miscellaneous Functions

Kernel-Only Functions

The function **IOConvertPort()** is necessary for some kernel-level driversĐand not for user-level driversĐbecause kernel-level drivers can execute in more than one task. The first thread of a kernel-level driver executes in the loadable kernel server's task, any threads that the driver creates execute in the kernel I/O task, and network drivers and drivers with UNIX entry points (at some stage) can execute in the context of an unknown task.

IOGetObjectForDeviceName() provides to kernel-level drivers some of the functionality provided to user-level programs by IODeviceMaster. Similarly, **IOHostPrivSelf()** is used by some kernel-level drivers that need the information normally returned by **host_priv_self()** (which is one of the few Mach functions that doesn't work in the kernel).

The function **IOVmTaskSelf()** supplies a **vm_task_t** for Mach function calls that expect one for the kernel; this is necessary because **vm_task_t** and **task_t** aren't the same in the kernel (as they are at user level). **IOVmTaskCurrent()** supplies a **vm_task_t** that's needed by some UNIX-style drivers. Finally, **IOVmTaskForBuf()** supplies a **vm_task_t** for the unknown task that is requesting UNIX-style I/O.

IOConvertPort()
IOGetObjectForDeviceName()
IOHostPrivSelf()
IOPhysicalFromVirtual()
IOSetUNIXError()
IOVmTaskCurrent()
IOVmTaskForBuf()
IOVmTaskSelf()

DMA Alignment Macros

IOAlign()
IOIsAligned()

Architecture-Specific Functions

The following functions are used by some Intel drivers to read and write I/O ports:

inb()
inw()
inl()
outb()
outw()
outl()

Some Intel drivers use the following function to help handle interrupts:

IODisableInterrupt()
IOEnableInterrupt()
IOSendInterrupt()

Some Intel devices require memory in the low 16 MB:

IOMallocLow()

Intel display drivers often use the following functions to read and write VGA registers:

IOReadRegister()
IOReadModifyWriteRegister()
IOWriteRegister()