Glossary

ABI

Application Binary Interface, a defined interface that includes an *API*, but adds the further promise that a compiled object file will be portable; no recompilation will be required to move to any supported platform.

API

Application Programming Interface, a defined interface through which services can be obtained. A typical API is implemented as a set of callable functions and header files that define the data structures and specific values that the functions accept or return. The promise behind an API is that a program that compiles and works correctly will continue to compile and work correctly in any supported environment (however, recompilation may be required when porting or changing versions). See *ABI*.

big-endian

The hardware design in which the most significant bits of a multi-byte integer are stored in the byte with the lowest address. Big-endian is the default storage order in MIPS processors. Opposed to *little-endian*.

block

As a verb, to suspend execution of a process. *See* sleep.

block device

A device such as magnetic tape or a disk drive, that naturally transfers data in blocks of fixed size. Opposed to *character device*.

block device driver

Driver for a block device. A block device's driver is not allowed to support the ioctl(), read() or write() entry points, but does have a strategy() entry point. *See* character device driver.

bus-watching cache

A *cache memory* that is aware of bus activity and, when the I/O system performs a DMA write into physical memory or another CPU in a multiprocessor system modifies *virtual memory*, automatically invalidates any copy of the same data then in the cache. This hardware function eliminates the need for explicit data cache write back or invalidation by software.

cache coherency

The problem of ensuring that all cached copies of data are true reflections of the data in memory. The usual solution is to ensure that, when one copy is changed, all other copies are automatically marked as invalid so that they will not be used.

cache line

The unit of data when data is loaded into a *cache memory*. Typically 128 bytes in current CPU models.

cache memory

High-speed memory closely attached to a CPU, containing a copy of the most recently used memory data. When the CPU's request for instructions or data can be satisfied from the cache, the CPU can run at full rated speed. In a multiprocessor or when DMA is allowed, a *bus-watching cache* is needed.

character device

A device such as a terminal or printer that transfers data as a stream of bytes, or a device that can be treated in this way under some circumstances. For example, a disk (normally a *block device*) can be treated as a character device for purposes of reading diagnostic information.

character device driver

The kernel-level device driver for a *character device* transfers data in bytes between the device and the user program. A *STREAMS driver* works with a character driver. Note that a *block device* such as magnetic tape or disk drives can also support character access through a character driver. Each disk device, for example, is represented as two different device special files, one managed by a *block device driver* and one by a character device driver.

close

Relinquish access to a resource. The user process invokes the **close**() system call when it is finished with a device, but the system does not necessarily execute your *drv***close**() entry point for that device.

data structure

Contiguous memory used to hold an ordered collection of fields of different types. Any *API* usually defines several data structures. The most common data structure in the *DDI/DKI* is the *buf_t*.

DDI/DKI

Device Driver Interface/Device Kernel Interface; the formal API that defines the services provided to a device driver by the kernel, and the rules for using those services. DDI/DKI is the term used in the UNIX System V documentation. The IRIX version of the DDI/DKI is close to, but not perfectly compatible with, the System V interface.

deadlock

The condition in which two or more processes are blocked, each waiting for a lock held by the other. Deadlock is prevented by the rule that a driver upper-half entry point is not allowed to hold a lock while sleeping.

devflag

A public global flag word that characterizes the abilities of a device driver, including the flags D_MP, D_WBACK and D_OLD.

device driver

A software module that manages access to a hardware device, taking the device in and out of service, setting hardware parameters, transmitting data between memory and the device, sometimes scheduling multiple uses of the device on behalf of multiple processes, and handling I/O errors.

direct memory access

When a device reads or writes in memory, asynchronously and without specific intervention by a CPU. In order to perform DMA, the device or its attachment must have some means of storing a memory address and incrementing it, usually through *mapping registers*. The device writes to physical memory and in so doing can invalidate *cache memory*; a *bus-watching cache* compensates.

device number

Each *device special file* is identified by a pair of numbers: the *major device number* identifies the device driver that manages the device, and the *minor device number* identifies the device to the driver.

device special file

A filename in the */dev* directory that represents a hardware device. A device special file does not specify data on disk, but rather identifies a particular hardware unit and the device driver that handles it. The *inode* of the file contains the *device number* as well as permissions and ownership data.

downstream

The direction of STREAMS messages flowing through a write queue from the user process to the driver.

EISA bus

Enhanced Industry Standard Architecture, a bus interface supported by certain Silicon Graphics systems.

EISA Product Identifier (ID)

The four-byte product identifier returned by an EISA expansion board.

file handle

An integer returned by the **open**() kernel function to represent the state of an open file. When the file handle is passed in subsequent kernel services, the kernel can retrieve information about the file, for example, when the file is a *device special file*, the file handle can be associated with the major and minor *device number*.

gigabyte

See kilobyte.

GIO bus

Graphics I/O bus, a bus interface used on Indigo, Indigo^2, and Indy workstations.

I/O operations

Services that provide access to shared input/output devices and to the global data structures that describe their status. I/O operations open and close files and devices, read data from and write data to devices, set the state of devices, and read and write system data structures.

inode

The UNIX and IRIX disk object that represents the existence of a file. The inode records the owner and group IDs and permissions. For regular disk files, the inode distinguishes files from directories and has other data that can be set with chmod. For device special files, the inode contains the major and minor device numbers and distinguishes block from character files.

inter-process communication

System calls that allow a process to send information to another process. There are several ways of sending information to another process: signals, pipes, shared memory, message queues, semaphores, streams, or sockets.

interrupt

A hardware signal that causes a CPU to set aside normal processing and begin execution of an interrupt handler. An interrupt is parameterized by the type of bus and the *interrupt level*, and possibly with an *interrupt vector* number. The kernel uses this information to select the interrupt handler for that device.

interrupt level

A number that characterizes the source of an *interrupt*. The VME bus provides for seven interrupt levels. Other buses have different schemes.

interrupt priority level

The relative priority at which a bus or device requests that the CPU call an interrupt process. Interrupts at a higher level are taken first. The interrupt handler for an interrupt can only be preempted on its CPU by an interrupt handler for an interrupt of higher level.

interrupt vector

A number that characterizes the specific device that caused an *interrupt*. Most VME bus devices have a specific vector number set by hardware, but some can have their vector set by software.

ioctl

Control a character device. Character device drivers may include a "special function" entry point, *pfx***ioct**().

IRQ

Interrupt Request Input, a hardware signal that initiates an interrupt.

k0

Virtual address range within the kernel address space that is cached but not mapped by translation look-aside buffers. Also referred to as kseg0.

k1

Virtual address range within the kernel address space that is neither cached nor mapped. Also called kseg1.

k2

Virtual address range within the kernel address space that can be both cached and mapped by translation look-aside buffers. Also called kseg2.

kernel level

The level of privilege at which code in the IRIX kernel runs. The kernel has a private address space, not acceptable to processes at *user-level*, and has sole access to physical memory.

kilobyte (KB)

1,024 bytes, a unit chosen because it is both an integer power of 2 (2^{10}) and close to 1,000, the basic scale multiple of engineering quantities. Thus 1,024 KB, 2^{20}, is 1 megabyte (MB) and close to 1e6; 1,024 MB, 2^{30}, is 1 gigabyte (GB) and close to 1e9; 1,024 GB, 2^{40}, is 1 terabyte (TB) and close to 1e12. In the MIPS architecture using 32-bit addressing, the user segment spans 2 GB. Using 64-bit addressing, both the user segment and the range of physical addresses span 1 TB.

kseg*n*

See k0, k1, k2.

little-endian

The hardware design in which the least significant bits of a multi-byte integer are stored in the byte with the lowest address. Little-endian order is the normal order in Intel processors, and optional in MIPS processors. Opposed to *big-endian*. (These terms are from Swift's *Gulliver's Travels*, in which the citizens of Lilliput and Blefescu are divided by the burning question of whether one's breakfast egg should be opened at the little or the big end.)

lock

A data object that represents the exclusive right to use a resource. A lock can be implemented as a *semaphore* (q.v.) with a count of 1, but because of the frequency of use of locks, they have been given distinct software support (see LOCK(D3)).

major device number

A number that specifies which device driver manages the device represented by a *device special file*. In IRIX 6.2, a major number has at most 9 bits of precision (0-511). Numbers 60-79 are used for OEM drivers. See also *minor device number.*

map

In general, to translate from one set of symbols to another. Particularly, translate one range of memory addresses to the addresses for the corresponding space in another system. The *virtual memory* hardware maps the process address space onto pages of physical memory. The *mapping registers* in a *DMA* device map bus addresses to physical memory corresponding to a buffer. The mmap(2) system call maps part of process address space onto the contents of a file.

mapping registers

Registers in a DMA device or its bus attachment that store the address translation data so that the device can access a buffer in physical memory.

megabyte

See kilobyte.

minor device number

A number that, encoded in a *device special file*, identifies a single hardware unit among the units managed by one device driver. Sometimes used to encode device management options as well. In IRIX 6.2, a minor number may have up to 18 bits of precision. *See also* major device number.

mmapped device driver

A driver that supports mapping hardware registers into process address space, permitting a user process to access device data as if it were in memory.

module

A STREAMS module consists of two related queue structures, one for upstream messages and one for downstream messages. One or more modules may be pushed onto a stream between the stream head and the driver, usually to implement and isolate a communication protocol or a line discipline.

open

Gain access to a device. The kernel calls the *pfx***open**() entry when the user process issues an **open**() system call.

page

A block of virtual or physical memory, of a size set by the operating system and residing on a page-size address boundary. The page size is 4,096 (2^{12}) bytes when in 32-bit mode; the page size in 64-bit mode can range from 2^{12} to 2^{20} at the operating system's choice (see the getpagesize(2) reference page).

PIO

Programmed I/O, meaning access to a VME device by mapping device registers into process address space, and transferring data by storing and loading single bytes or words.

poll

Poll entry point for a non-stream character driver. A character device driver may include a *drv***poll**() entry point so that users can use **select**(2) or **poll**(2) to poll the file descriptors opened on such devices.

prefix

Driver prefix. The name of the driver must be the first characters of its standard entry point names; the combined names are used to dynamically link the driver into the kernel. Specified in the *master.d* file for the driver. Throughout this manual, the prefix *pfx* represents the name of the device driver, as in *pfx***open**(), *pfx***ioctl**().

primary cache

The *cache memory* most closely attached to the CPU execution unit, usually in the processor chip.

primitives

Fundamental operations from which more complex operations can be constructed.

priority inheritance

An implementation technique that prevents *priority inversion* when a process of lower priority holds a mutual exclusion *lock* and a process of higher priority is blocked waiting for the lock. The process holding the lock "inherits" or acquires the priority of the highest-priority waiting process in order to expedite its release of the lock. IRIX supports priority inheritance for mutual exclusion locks only.

priority inversion

The effect that occurs when a low-priority process holds a *lock* that a process of higher priority needs. The lower priority process runs and the higher priority process waits, inverting the intended priorities. *See* priority inheritance.

process control

System calls that allow a process to control its own execution. A process can allocate memory, lock itself in memory, set its scheduling priorities, wait for events, execute a new program, or create a new process.

protocol stack

A software subsystem that manages the flow of data on a communications channel according to the rules of a particular protocol, for example the TCP/IP protocol. Called a "stack" because it is typically designed as a hierarchy of layers, each supporting the one above and using the one below.

pseudo-device

Software that uses the facilites of the *DDI/DKI* to provide specialized access to data, without using any actual hardware device. Pseudo-devices can provide access to system data structures that are unavailable at the user-level. For example, the fsctl driver gives superuser access to filesystem data (see fsctl(7)) and the inode monitor pseudo-device allows access to file activity (see imon(7)).

read

Read data from a device. The kernel executes the *pfx***read**() entry point whenever a user process calls the **read**() system call.

scatter/gather

An I/O operation in which what to the device is a contiguous range of data is distributed across multiple pages that may not be contiguous in physical memory. On input to memory, the device scatters the data into the different pages; on output, the device gathers data from the pages.

SCSI

Small Computer System Interface, the bus architecture commonly used to attach disk drives and other block devices.

SCSI driver interface

A collection of machine-independent input/output controls, functions, and data structures, that provides a standard interface for writing a SCSI driver.

semaphore

A data object that represents the right to use a limited resource, used for synchronization and communication between asynchronous processes. A semaphore contains a count that represents the quantity of available resource (typically 1). The **P** operation (mnemonic: de*P*lete) decrements the count and, if the count goes negative, causes the caller to wait (see psema(D3X), cpsema(D3X)). The **V** operation (mnemonic: re*V*ive) increments the count and releases any waiting process (see vsema(D3X), cvsema(D3X)). *See also* lock.

signals

Software interrupts used to communicate between processes. Specific signal numbers can be handled or blocked. Device drivers sometimes use signals to report events to user processes. Device drivers that can wait have to be sensitive to the possibility that a signal could arrive.

sleep

Suspend process execution pending occurrence of an event. The term "block" is also used.

socket

A software structure that represents one endpoint in a two-way communications link. Created by socket(2).

spl

Set priority level, a function that was formerly part of the *DDI/DKI*, and used to lock or allow interrupts on a processor. It is not possible to use spl effectively in a multiprocessor system, so it has been superceded by more sophisticated means of synchronization such as the *lock* and *semaphore*.

strategy

In general, the plan or policy for arbitrating between multiple, concurrent requests for the use of a device. Specifically in disk device drivers, the policy for scheduling multiple, concurrent disk block-read and block-write requests.

STREAM

A linked list of kernel data structures that provide a full-duplex data path between a user process and a device. Streams are supported by the STREAMS facilities in UNIX System V Release 3 and later.

STREAM head

The stream head, which is inserted by the STREAMS subsystem, processes STREAMS-related system calls and performs data transfers between user space and kernel space. It is the component of a stream closest to the user process. Every stream has a stream head.

STREAMS

A kernel subsystem used to build a stream, which is a modular, full-duplex data path between a device and a user process. In IRIX 5.x and later, the TCP/IP stack sits on top of the STREAMS stack. The Transport Layer Interface (TLI) is fully supported.

STREAMS driver

A software module that implements one stage of a STREAM. A STREAMS driver can be "pushed on" or "popped off" any *STREAM*.

TCP/IP

Transmission Control Protocol/Internet Protocol.

terabyte

See *kilobyte (KB)*.

TFP

The internal code name for the MIPS R8000 processor, used in some Silicon Graphics publications.

TLI

Transport Interface Layer.

user-level

The privilege level of the system at which user-initiated programs run. A user-level process can access the contents of one address space, and can access files and devices only by calling kernel functions. Contrast to *kernel level*.

unmap

Disconnect a memory-mapped device from user process space, breaking the association set by mapping it.

VME bus

VERSA Module Eurocard bus, a bus architecture supported by the Silicon Graphics Challenge and Onyx systems.

VME-bus adapter

A hardware conduit that translates host CPU operations to VME-bus operations and decodes some VME-bus operations to translate them to the host side.

virtual memory

Memory contents that appear to be in contiguous addresses, but are actually mapped to different physical memory locations by hardware action of the translation lookaside buffer (TLB) and page tables managed by the IRIX kernel. The kernel can exploit virtual memory to give each process its own address space, and to load many more processes than physical memory can support.

virtual page number

The most significant bits of a virtual address, which select a *page* of memory. The processor hardware looks for the VPN in the TLB; if the VPN is found, it is translated to a physical page address. If it is not found, the processor traps to an exception routine.

volatile

Subject to change. The volatile keyword informs the compiler that a variable could change value at any time (because it is mapped to a hardware register, or because it is shared with other, concurrent processes) and so should always be loaded before use.

wakeup

Resume suspended process execution.

write

Write data to a device. The kernel executes the *pfx***read**() or *pfx***write**() entry points whenever a user process calls the **read**() or **write**() system calls.