Glossary

activity

When using the Frame Scheduler, the basic design unit: a piece of work that can be done by one process without interruption. You partition the real-time program into activities, and use the Frame Scheduler to invoke them in sequence within each frame interval.

address space

The set of memory addresses that a process may legally access. The potential address space in IRIX is either 2^{32} (IRIX 5.3) or 2^{64} (IRIX 6.0); however only addresses that have been mapped by the kernel are legally accessible.

affinity scheduling

The IRIX kernel attempts to run a process on the same CPU where it most recently ran, in the hope that some of the process's data will still remain in the cache of that CPU. The process is said to have "cache affinity" for that CPU. ("Affinity" means "a natural relationship or attraction.")

arena

A segment of memory used as a pool for allocation of objects of a particular type. Usually the shared memory segment allocated by **usinit()**.

asynchronous I/O

I/O performed in a separate process, so that the process requesting the I/O is not blocked waiting for the I/O to complete.

average data rate

The rate at which data arrives at a data collection system, averaged over a given period of time (seconds or minutes, depending on the application). The system must be able to write data at the average rate, and it must have enough memory to buffer bursts at the *peak data rate*.

backing store

The disk location that contains the contents of a memory page. The contents of the page are retrieved from the backing store when the page is needed in memory. The backing

store for executable code is the program or library file. The backing store for modifiable pages is the swap disk. The backing store for a memory-mapped file is the file itself.

barrier

A memory object that represents a point of rendezvous or synchronization between multiple processes. The processes come to the barrier asynchronously, and block there until all have arrived. When all have arrived, all resume execution together.

context switch time

The time required for IRIX to set aside the context, or execution state, of one process and to enter the context of another; for example, the time to leave a process and enter a device driver, or to leave a device driver and resume execution of an interrupted process.

deadline scheduling

A process scheduling discipline supported by IRIX version 5.3. A process may require that it receive a specified amount of execution time over a specified interval, for instance 70ms in every 100ms. IRIX adjusts the process's priority up and down as required to ensure that it gets the required execution time.

deadlock

A situation in which two (or more) processes are blocked because each is waiting for a resource held by the other.

device driver

Code that operates a specific hardware device and handles interrupts from that device. Refer to the *IRIX Device Driver Programmer's Guide*, part number 007-0911-060.

device numbers

Each I/O device is represented by a name in the */dev* file system hierarchy. When these "special device files" are created (see the makedev(1) and install(1) reference pages) they are given major and minor device numbers. The major number is the index of a *device driver* in the kernel. The minor number is specific to the device, and encodes such information as its unit number, density, VME bus address space, or similar hardware-dependent information.

device service time

The amount of time spent executing the code of a *device driver* in servicing one interrupt. One of the three main components of *interrupt response time*.

device special file

The symbolic name of a device that appears as a filename in the */dev* directory hierarchy. The file entry contains the *device numbers* that associate the name with a *device driver*.

direct memory access (DMA)

Independent hardware that transfers data between memory and an I/O device without program involvement. Challenge/Onyx systems have a DMA engine for the VME bus.

file descriptor

A number returned by **open()** and other system functions to represent the state of an open file. The number is used with system calls such as **read()** to access the opened file or device.

frame rate

The frequency with which a simulator updates its display, in cycles per second (Hz). Typical frame rates range from 15 to 60 Hz.

frame interval

The inverse of *frame rate*, that is, the amount of time that a program has to prepare the next display frame. A frame rate of 60 Hz equals a frame time of 16.67 milliseconds.

frs control process

The process that creates a Frame Scheduler. Its process ID is used to identify the Frame Scheduler internally, so a process can only be frs control to one scheduler.

gang scheduling

A process scheduling discipline supported by IRIX. The processes of a *share group* can request to be scheduled as a gang; that is, IRIX attempts to schedule all of them concurrently when it schedules any of them—provided there are enough CPUs. When processes coordinate using locks, gang scheduling helps to ensure that one does not spend its whole time slice spinning on a lock held by another that is not running.

guaranteed rate

A rate of data transfer, in bytes per second, that definitely is available through a particular file descriptor.

hard guarantee

A type of *guaranteed rate* that is met even if data integrity has to be sacrificed to meat it.

heap

The *segment* of the *address space* devoted to static data and dynamically-allocated objects. Created by calls to the system function **brk()**.

interrupt

A hardware signal from an I/O device that causes the computer to divert execution to a device driver.

interrupt group

In the Challenge/Onyx hardware, each CPU has a register containing an interrupt group mask. Each interrupt source can be directed to a specific CPU or to an interrupt group number. When the interrupt destination is a group, all CPUs that have enabled that group receive the interrupt. The Frame Scheduler creates an interrupt group in order to synchronize minor frames among multiple synchronized CPUs.

interrupt latency

The amount of time that elapses between the arrival of an interrupt signal and the entry to the device driver that handles the interrupt.

interrupt response time

The total time from the arrival of an interrupt until the user process is executing again. Its three main components are *interrupt latency*, *device service time*, and *context switch time*.

locality of reference

The degree to which a program keeps memory references confined to a small number of locations over any short span of time. The better the locality of reference, the more likely a program will execute entirely from fast cache memory. The more scattered are a program's memory references, the higher is the chance that it will access main memory or, worse, load a page from swap.

locks

Memory objects that represent the exclusive right to use a shared resource. A process that wants to use the resource requests the lock that (by agreement) stands for that resource. The process releases the lock when it is finished using the resource. See *semaphore*.

major frame

The basic frame rate of a program running under the Frame Scheduler.

minor frame

The scheduling unit of the Frame Scheduler, the period of time in which any scheduled process must do its work.

overrun

When incoming data arrives faster than a data collection system can accept it, so that data is lost, an overrun has occurred.

overrun exception

When a process scheduled by the Frame Scheduler should have yielded before the end of the minor frame and did not, an overrun exception is signalled.

pages

The units of real memory managed by the kernel. Memory is always allocated in page units on page-boundary addresses. Virtual memory is read and written from the swap device in page units.

page fault

The hardware event that results when a process attempts to access a page of virtual memory that is not present in physical memory.

peak data rate

The instantaneous maximum rate of input to a data collection system. The system must be able to accept data at this rate to avoid *overrun*. See *average data rate*.

process

The entity that executes instructions in a UNIX system. A process has access to an *address space* containing its instructions and data. The state of a process includes its set of machine register values, as well as many *process attributes*.

process attributes

Variable information about the state of a process. Every process has a number of attributes, including such things as its process ID, user and group IDs, working directory, open file handles, scheduler class, environment variables, and so on. See the fork(2) reference page for a list.

process group

See *share group*.

processor sets

Groups of one or more CPUs designated using the *pset* command.

programmed I/O (PIO)

Transfer of data between memory and an I/O device in byte or word units, using program instructions for each unit. Under IRIX, I/O to memory-mapped VME devices is done with PIO. See DMA.

race condition

Any situation in which two or more processes update a shared resource in an uncoordinated way. For example, if one process sets a word of shared memory to 1, and the other sets it to 2, the final result depends on the "race" between the two to see which can update memory last. Race conditions are prevented by use of *semaphores* or *locks*.

resident set size

The aggregate size of the valid (that is, memory-resident) pages in the address space of a process. Reported by *ps* under the heading RSS. See *virtual size* and the ps(1) reference page.

scheduling discipine

The rules under which an activity process is dispatched by a Frame Scheduler, including whether or not the process is allowed to cause overrun or underrun exceptions.

segment

Any contiguous range of memory addresses. Segments as allocated by IRIX always start on a page boundary and contain an integral number of pages.

semaphore

A memory object that represents the availability of a shared resource. A process that needs the resource executes a "p" operation on the semaphore to reserve the resource, blocking if necessary until the resource is free. The resource is released by a "v" operation on the semaphore. See *locks*.

share group

A group of two or more processes created with **sproc()**, including the original parent process. Processes in a share group share a common *address space* and can be scheduled as a gang (see *gang scheduling*). Also called a *process group*.

signal latency

The time that elapses from the moment when a signal is generated until the signal-handling function begins to execute. Signal latency is longer, and much less predictable, than *interrupt latency*.

soft guarantee

A type of *guaranteed rate* that XFS may fail to meet in order to retry device errors.

spraying interrupts

In order to equalize workload across all CPUs, the Challenge/Onyx systems direct each I/O interrupt to a different CPU chosen in rotation. In order to protect a real-time program from unpredictable interrupts, you can isolate specified CPUs from sprayed interrupts, or you can assign interrupts to specific CPUs.

striped volume

A logical disk volume comprising multiple disk drives, in which segments of data that are logically in sequence ("stripes") are physically located on each drive in turn. As many processes as there are drives in the volume can read concurrently at the maximum rate.

translation lookaside buffer (TLB)

An on-chip cache of recently-used virtual-memory page addresses, with their physical-memory translations. The CPU uses the TLB to translate virtual addresses to physical ones at high speed. When the IRIX kernel alters the in-memory page translation tables, it broadcasts an interrupt to all CPUs, telling them to purge their TLBs. You can isolate a CPU from these unpredictable interrupts, under certain conditions.

transport delay

The time it takes for a simulator to reflect a control input in its output display. Too long a transport delay makes the simulation inaccurate or unpleasant to use.

underrun exception

When a process scheduled by the Frame Scheduler should have started in a given minor frame but did not (owing to being blocked), an underrun exception is signalled. See *overrun exception*.

VERSA-Model Eurocard (VME) bus

A hardware interface and device protocol for attaching I/O devices to a computer. The VME bus is an ANSI standard. Many third-party manufacturers make VME-compatible devices. The Silicon Graphics Challenge/Onyx and Crimson computer lines support the VME bus.

video on demand (VOD)

In general, producing video data at video frame rates. Specific to *guaranteed rate*, a disk organization that places data across the drives of a *striped volume* so that multiple processes can achieve the same guaranteed rate while reading sequentially.

virtual size

The aggregate size of all the pages that are defined in the address space of a process. The virtual size of a process is reported by *ps* under the heading SZ. The sum of all virtual sizes cannot exceed the size of the swap space. See *resident set size* and the ps(1) reference page.

virtual address space

The set of numbers that a process can validly use as memory addresses.