NAME

kstat_queue, kstat_waitq_enter, kstat_waitq_exit, kstat_runq_enter, kstat_runq_exit, kstat_waitq_to_runq, kstat_runq_back_to_waitq – update I/O kstat statistics

SYNOPSIS

#include <sys/types.h>
#include <sys/kstat.h>

void kstat_waitq_enter(kstat_io_t **kiop***);**

void kstat_waitq_exit(kstat_io_t **kiop***);**

void kstat_runq_enter(kstat_io_t **kiop***);**

void kstat_runq_exit(kstat_io_t **kiop***);**

void kstat_waitq_to_runq(kstat_io_t **kiop***);**

void kstat_runq_back_to_waitq(kstat_io_t **kiop***);**

INTERFACE LEVEL

Solaris DDI specific (Solaris DDI)

ARGUMENTS

kiop Pointer to a **kstat_io**(9S) structure.

DESCRIPTION

A large number of I/O subsystems have at least two basic "lists" (or queues) of transactions they manage: one for transactions that have been accepted for processing but for which processing has yet to begin, and one for transactions which are actively being processed (but not done). For this reason, two cumulative time statistics are kept: wait (pre-service) time, and run (service) time.

The **kstat_queue( )** family of functions manage these times based on the transitions between the driver wait queue and run queue.

kstat_waitq_enter( )

kstat_waitq_enter( ) should be called when a request arrives and is placed into a pre-service state (such as just prior to calling **disksort**(9F)).

kstat_waitq_exit( )

kstat_waitq_exit( ) should be used when a request is removed from its pre-service state. (such as just prior to calling the driver's **start** routine).

kstat_runq_enter( )

kstat_runq_enter( ) is also called when a request is placed in its service state (just prior to calling the driver's start routine, but after **kstat_waitq_exit( )**).

kstat_runq_exit( )

kstat_runq_exit( ) is used when a request is removed from its service state (just prior to calling **biodone**(9F)).

kstat_waitq_to_runq( )

kstat_waitq_to_runq( ) transitions a request from the wait queue to the run queue. This is useful wherever the driver would have normally done a **kstat_waitq_exit( )** followed by a call to **kstat_runq_enter( )**.

kstat_runq_back_to_waitq( )

kstat_runq_back_to_waitq( ) transitions a request from the run queue back to the wait queue. This may be necessary in some cases (write throttling is an example).

RETURN VALUES

None.

CONTEXT

kstat_create( ) can be called from user or kernel context.

WARNINGS

These transitions must be protected by holding the kstat's **ks_lock**, and must be completely accurate (all transitions are recorded). Forgetting a transition may, for example, make an idle disk appear 100% busy.

SEE ALSO

kstat_create(9F), **kstat_delete**(9F), **kstat_named_init**(9F), **kstat**(9S), **kstat_io**(9S)

Writing Device Drivers