NAME

queue − implementations of linked lists and queues

DESCRIPTION

The <*sys/queue.h*> header file provides a set of macros that define and operate on the following data structures:

SLIST singly linked lists

LIST doubly linked lists

STAILQ
singly linked tail queues

TAILQ doubly linked tail queues

CIRCLEQ
doubly linked circular queues

All structures support the following functionality:

- Insertion of a new entry at the head of the list.
- Insertion of a new entry after any element in the list.
- O(1) removal of an entry from the head of the list.
- Forward traversal through the list.

Code size and execution time depend on the complexity of the data structure being used, so programmers should take care to choose the appropriate one.

Singly linked lists (SLIST)

Singly linked lists are the simplest and support only the above functionality. Singly linked lists are ideal for applications with large datasets and few or no removals, or for implementing a LIFO queue. Singly linked lists add the following functionality:

- O(n) removal of any entry in the list.

Singly linked tail queues (STAILQ)

Singly linked tail queues add the following functionality:

- Entries can be added at the end of a list.
- O(n) removal of any entry in the list.
- They may be concatenated.

However:

- All list insertions must specify the head of the list.
- Each head entry requires two pointers rather than one.

Singly linked tail queues are ideal for applications with large datasets and few or no removals, or for implementing a FIFO queue.

Doubly linked data structures

All doubly linked types of data structures (lists and tail queues) additionally allow:

- Insertion of a new entry before any element in the list.
- O(1) removal of any entry in the list.

However:

- Each element requires two pointers rather than one.

Doubly linked lists (LIST)

Linked lists are the simplest of the doubly linked data structures. They add the following functionality over the above:

- They may be traversed backwards.

However:

- To traverse backwards, an entry to begin the traversal and the list in which it is contained must be specified.

Doubly linked tail queues (TAILQ)

Tail queues add the following functionality:

- Entries can be added at the end of a list.
- They may be traversed backwards, from tail to head.
- They may be concatenated.

However:

- All list insertions and removals must specify the head of the list.
- Each head entry requires two pointers rather than one.

Doubly linked circular queues (CIRCLEQ)

Circular queues add the following functionality over the above:

- The first and last entries are connected.

However:

- The termination condition for traversal is more complex.

STANDARDS

Not in POSIX.1, POSIX.1-2001, or POSIX.1-2008. Present on the BSDs. <*sys/queue.h*> macros first appeared in 4.4BSD.

NOTES

Some BSDs provide SIMPLEQ instead of STAILQ. They are identical, but for historical reasons they were named differently on different BSDs. STAILQ originated on FreeBSD, and SIMPLEQ originated on NetBSD. For compatibility reasons, some systems provide both sets of macros. glibc provides both STAILQ and SIMPLEQ, which are identical except for a missing SIMPLEQ equivalent to **STAILQ_CONCAT**().

SEE ALSO

circleq(3), **insque**(3), **list**(3), **slist**(3), **stailq**(3), **tailq**(3)