Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'queue.7' command

$ man queue.7

QUEUE(7)   Linux Programmer's Manual   QUEUE(7)

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:

    * singly linked lists (SLIST)

    * doubly linked lists (LIST)

    * singly linked tail queues (STAILQ)

    * doubly linked tail queues (TAILQ)

    * doubly linked circular queues (CIRCLEQ)

    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 ap?

    propriate one.

  Singly linked lists (SLIST)

    Singly linked lists are the simplest and support only the above func?

    tionality.   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) ad?

ditionally 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.

CONFORMING TO

      Not in POSIX.1, POSIX.1-2001 or POSIX.1-2008. Present on the BSDs.

      <sys/queue.h> macros first appeared in 4.4BSD.

SEE ALSO

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

COLOPHON

      This page is part of release 5.10 of the Linux man-pages project.   A

      description of the project, information about reporting bugs, and the

      latest   version   of   this   page,   can   be   found   at

      https://www.kernel.org/doc/man-pages/.

GNU   2020-11-16   QUEUE(7)