What is anchor?
==========

A USB driver needs to support some callbacks requiring
a driver to cease all IO to an interface. To do so, a
driver has to keep track of the URBs it has submitted
to know they've all completed or to call usb_kill_urb
for them. The anchor is a data structure takes care of
keeping track of URBs and provides methods to deal with
multiple URBs.

Allocation and Initialisation
==========

There's no API to allocate an anchor. It is simply declared
as struct usb_anchor. init_usb_anchor() must be called to
initialise the data structure.

Deallocation
==========

Once it has no more URBs associated with it, the anchor can be
freed with normal memory management operations.

Association and disassociation of URBs with anchors
==========

An association of URBs to an anchor is made by an explicit
call to usb_anchor_urb(). The association is maintained until
an URB is finished by (successful) completion. Thus disassociation
is automatic. A function is provided to forcibly finish (kill)
all URBs associated with an anchor.
Furthermore, disassociation can be made with usb_unanchor_urb()

Operations on multitudes of URBs
==========

usb_kill_anchored_urbs()

This function kills all URBs associated with an anchor. The URBs
are called in the reverse temporal order they were submitted.
This way no data can be reordered.

usb_unlink_anchored_urbs()

This function unlinks all URBs associated with an anchor. The URBs
are processed in the reverse temporal order they were submitted.
This is similar to usb_kill_anchored_urbs(), but it will not sleep.
Therefore no guarantee is made that the URBs have been unlinked when
the call returns. They may be unlinked later but will be unlinked in
finite time.

usb_scuttle_anchored_urbs()

All URBs of an anchor are unanchored en masse.

usb_wait_anchor_empty_timeout()

This function waits for all URBs associated with an anchor to finish
or a timeout, whichever comes first. Its return value will tell you
whether the timeout was reached.

usb_anchor_empty()

Returns true if no URBs are associated with an anchor. Locking
is the caller's responsibility.

usb_get_from_anchor()

Returns the oldest anchored URB of an anchor. The URB is unanchored
and returned with a reference. As you may mix URBs to several
destinations in one anchor you have no guarantee the chronologically
first submitted URB is returned.