NAME

sf_buf - manage temporary kernel address space mapping for memory pages

SYNOPSIS

#include <sys/sf_buf.h>

*struct sf_buf **
sf_buf_alloc(*struct vm_page *m, int flags*);

void
sf_buf_free(*struct sf_buf *sf*);

vm_offset_t
sf_buf_kva(*struct sf_buf *sf*);

*struct vm_page **
sf_buf_page(*struct sf_buf *sf*);

DESCRIPTION

The **sf_buf** interface, historically the sendfile(2) buffer interface, allows kernel subsystems to manage temporary kernel address space mappings for physical memory pages. On systems with a direct memory map region (allowing all physical pages to be visible in the kernel address space at all times), the *struct sf_buf* will point to an address in the direct map region; on systems without a direct memory map region, the *struct sf_buf* will manage a temporary kernel address space mapping valid for the lifetime of the *struct sf_buf*.

Call **sf_buf_alloc**() to allocate a *struct sf_buf* for a physical memory page. **sf_buf_alloc**() is not responsible for arranging for the page to be present in physical memory; the caller should already have arranged for the page to be wired, i.e., by calling vm_page_wire(9). Several flags may be passed to **sf_buf_alloc**():

SFB_CATCH — Cause **sf_buf_alloc**() to abort and return NULL if a signal is received waiting for a *struct sf_buf* to become available.

SFB_NOWAIT — Cause **sf_buf_alloc**() to return NULL rather than sleeping if a *struct sf_buf* is not immediately available.

SFB_CPUPRIVATE — Cause **sf_buf_alloc**() to only arrange that the temporary mapping be valid on the current CPU, avoiding unnecessary TLB shootdowns for mappings that will only be accessed on a single CPU at a time. The caller must ensure that accesses to the

virtual address occur only on the CPU from which **sf_buf_alloc**() was invoked, perhaps by using **sched_pin**().

Call **sf_buf_kva**() to return a kernel mapped address for the page.

Call **sf_buf_page**() to return a pointer to the page originally passed into **sf_buf_alloc**().

Call **sf_buf_free**() to release the *struct sf_buf* reference. The caller is responsible for releasing any wiring they have previously acquired on the physical page; **sf_buf_free**() releases only the temporary kernel address space mapping, not the page itself.

Uses of this interface include managing mappings of borrowed pages from user memory, such as in zero-copy socket I/O, or pages of memory from the buffer cache referenced by mbuf external storage for sendfile(2).

SEE ALSO

sendfile(2), vm_page_wire(9)

AUTHORS

The *struct sf_buf* API was designed and implemented by Alan L. Cox. This manual page was written by Robert N. M. Watson.