The Linux Kernel API

This documentation is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

For more details see the file COPYING in the source distribution of Linux.

Table of Contents

1. Driver Basics

1.1. Driver Entry and Exit points

module_init -- driver initialization entry point
module_exit -- driver exit entry point

1.2. Atomic and pointer manipulation

atomic_read -- read atomic variable
atomic_set -- set atomic variable
atomic_add -- add integer to atomic variable
atomic_sub -- subtract the atomic variable
atomic_sub_and_test -- subtract value from variable and test result
atomic_inc -- increment atomic variable
atomic_dec -- decrement atomic variable
atomic_dec_and_test -- decrement and test
atomic_inc_and_test -- increment and test
atomic_add_negative -- add and test if negative
get_unaligned -- get value from possibly mis-aligned location
put_unaligned -- put value to a possibly mis-aligned location

2. Data Types

2.1. Doubly Linked Lists

list_add -- add a new entry
list_add_tail -- add a new entry
list_add_rcu -- add a new entry to rcu-protected list
list_add_tail_rcu -- add a new entry to rcu-protected list
list_del -- deletes entry from list.
list_del_rcu -- deletes entry from list without re-initialization
list_del_init -- deletes entry from list and reinitialize it.
list_move -- delete from one list and add as another's head
list_move_tail -- delete from one list and add as another's tail
list_empty -- tests whether a list is empty
list_splice -- join two lists
list_splice_init -- join two lists and reinitialise the emptied list.
list_entry -- get the struct for this entry
list_for_each -- iterate over a list

__list_for_each -- iterate over a list
list_for_each_prev -- iterate over a list backwards
list_for_each_safe -- iterate over a list safe against removal of list entry
list_for_each_entry -- iterate over list of given type
list_for_each_entry_reverse -- iterate backwards over list of given type.
list_for_each_entry_safe -- iterate over list of given type safe against removal of list entry
list_for_each_rcu -- iterate over an rcu-protected list
list_for_each_safe_rcu -- iterate over an rcu-protected list safe
list_for_each_entry_rcu -- iterate over rcu list of given type
list_for_each_continue_rcu -- iterate over an rcu-protected list
hlist_del_rcu -- deletes entry from hash list without re-initialization
hlist_for_each_entry -- iterate over list of given type
hlist_for_each_entry_continue -- iterate over a hlist continuing after existing point
hlist_for_each_entry_from -- iterate over a hlist continuing from existing point
hlist_for_each_entry_safe -- iterate over list of given type safe against removal of list entry

3. Basic C Library Functions

3.1. String Conversions

simple_strtoll -- convert a string to a signed long long
simple_strtoul -- convert a string to an unsigned long
simple_strtol -- convert a string to a signed long
simple_strtoull -- convert a string to an unsigned long long
vsnprintf -- Format a string and place it in a buffer
snprintf -- Format a string and place it in a buffer
vsprintf -- Format a string and place it in a buffer
sprintf -- Format a string and place it in a buffer
vsscanf -- Unformat a buffer into a list of arguments
sscanf -- Unformat a buffer into a list of arguments

3.2. String Manipulation

strcpy -- Copy a `NUL` terminated string
strncpy -- Copy a length-limited, `NUL-terminated` string
strcat -- Append one `NUL-terminated` string to another
strncat -- Append a length-limited, `NUL-terminated` string to another
strcmp -- Compare two strings
strncmp -- Compare two length-limited strings
strchr -- Find the first occurrence of a character in a string
strrchr -- Find the last occurrence of a character in a string
strlen -- Find the length of a string
strnlen -- Find the length of a length-limited string
strcspn -- Calculate the length of the initial substring of *s* which does
strpbrk -- Find the first occurrence of a set of characters
memset -- Fill a region of memory with the given value
bcopy -- Copy one area of memory to another
memcpy -- Copy one area of memory to another
memmove -- Copy one area of memory to another
memcmp -- Compare two areas of memory
memscan -- Find a character in an area of memory.
strstr -- Find the first substring in a `NUL` terminated string
memchr -- Find a character in an area of memory.
strnicmp -- Case insensitive, length-limited string comparison
strlcpy -- Copy a `NUL` terminated string into a sized buffer
strlcat -- Append a length-limited, `NUL-terminated` string to another
strspn -- Calculate the length of the initial substring of *s* which only
strsep -- Split a string into tokens

3.3. Bit Operations

set_bit -- Atomically set a bit in memory
__set_bit -- Set a bit in memory
clear_bit -- Clears a bit in memory
__change_bit -- Toggle a bit in memory
change_bit -- Toggle a bit in memory
test_and_set_bit -- Set a bit and return its old value
__test_and_set_bit -- Set a bit and return its old value
test_and_clear_bit -- Clear a bit and return its old value
__test_and_clear_bit -- Clear a bit and return its old value
test_and_change_bit -- Change a bit and return its new value
test_bit -- Determine whether a bit is set
find_first_zero_bit -- find the first zero bit in a memory region
find_first_bit -- find the first set bit in a memory region
find_next_zero_bit -- find the first zero bit in a memory region
find_next_bit -- find the first set bit in a memory region
ffz -- find first zero in word.
__ffs -- find first bit in word.
ffs -- find first bit set
hweight32 -- returns the hamming weight of a N-bit word

4. Memory Management in Linux

4.1. The Slab Cache

kmem_cache_create -- Create a cache.
kmem_cache_shrink -- Shrink a cache.
kmem_cache_destroy -- delete a cache
kmem_cache_alloc -- Allocate an object
__kmalloc -- allocate memory
__alloc_percpu -- allocate one copy of the object for every present
kmem_cache_free -- Deallocate an object
kfree -- free previously allocated memory
free_percpu -- free previously allocated percpu memory

4.2. User Space Memory Access

access_ok -- Checks if a user space pointer is valid
verify_area -- Obsolete, use `access_ok`
get_user -- Get a simple variable from user space.
put_user -- Write a simple value into user space.
__get_user -- Get a simple variable from user space, with less checking.
__put_user -- Write a simple value into user space, with less checking.
__copy_to_user -- Copy a block of data into user space, with less checking.
__copy_from_user -- Copy a block of data from user space, with less checking.
copy_to_user -- Copy a block of data into user space.
copy_from_user -- Copy a block of data from user space.
strlen_user -- Get the size of a string in user space.
__strncpy_from_user -- Copy a NUL terminated string from userspace, with less checking.
strncpy_from_user -- Copy a NUL terminated string from userspace.
clear_user -- Zero a block of memory in user space.
__clear_user -- Zero a block of memory in user space, with less checking.
strnlen_user -- Get the size of a string in user space.

5. The proc filesystem

5.1. sysctl interface

register_sysctl_table -- register a sysctl hierarchy
unregister_sysctl_table -- unregister a sysctl table hierarchy
proc_dostring -- read a string sysctl
proc_dointvec -- read a vector of integers
proc_dointvec_minmax -- read a vector of integers with min/max values
proc_doulongvec_minmax -- read a vector of long integers with min/max values
proc_doulongvec_ms_jiffies_minmax -- read a vector of millisecond values with min/max values
proc_dointvec_jiffies -- read a vector of integers as seconds

6. The Linux VFS

6.1. The Directory Cache

d_invalidate -- invalidate a dentry
d_find_alias -- grab a hashed alias of inode
shrink_dcache_sb -- shrink dcache for a superblock
have_submounts -- check for mounts over a dentry
shrink_dcache_parent -- prune dcache
shrink_dcache_anon -- further prune the cache
d_alloc -- allocate a dcache entry
d_instantiate -- fill in inode information for a dentry
d_alloc_root -- allocate root dentry
d_alloc_anon -- allocate an anonymous dentry
d_splice_alias -- splice a disconnected dentry into the tree if one exists
d_lookup -- search for a dentry
d_validate -- verify dentry provided from insecure source
d_delete -- delete a dentry
d_rehash -- add an entry back to the hash
d_move -- move a dentry
is_subdir -- is new dentry a subdirectory of old_dentry
find_inode_number -- check for dentry with name
__d_drop -- drop a dentry
d_add -- add dentry to hash queues
dget -- get a reference to a dentry
d_unhashed -- is dentry hashed

6.2. Inode Handling

clear_inode -- clear an inode
invalidate_inodes -- discard the inodes on a device
new_inode -- obtain an inode
iunique -- get a unique inode number
ilookup5 -- search for an inode in the inode cache
ilookup -- search for an inode in the inode cache
iget5_locked -- obtain an inode from a mounted file system
iget_locked -- obtain an inode from a mounted file system
__insert_inode_hash -- hash an inode
remove_inode_hash -- remove an inode from the hash
iput -- put an inode
bmap -- find a block number in a file
update_atime -- update the access time
inode_update_time -- update mtime and ctime time
make_bad_inode -- mark an inode bad due to an I/O error
is_bad_inode -- is an inode errored

6.3. Registration and Superblocks

deactivate_super -- drop an active reference to superblock
generic_shutdown_super -- common helper for ->`kill_sb`
sget -- find or create a superblock
get_super -- get the superblock of a device

6.4. File Locks

posix_lock_file -- Apply a POSIX-style lock to a file
locks_mandatory_area -- Check for a conflicting lock
__break_lease -- revoke all outstanding leases on file
lease_get_mtime --
posix_block_lock -- blocks waiting for a file lock
posix_unblock_lock -- stop waiting for a file lock
lock_may_read -- checks that the region is free of locks
lock_may_write -- checks that the region is free of locks
locks_mandatory_locked -- Check for an active lock
fcntl_getlease -- Enquire what lease is currently active
fcntl_setlease -- sets a lease on an open file
sys_flock -- `flock` system call.
get_locks_status -- reports lock usage in /proc/locks

7. Linux Networking

7.1. Socket Buffer Functions

struct sk_buff -- socket buffer
skb_queue_empty -- check if a queue is empty
skb_get -- reference buffer
kfree_skb -- free an sk_buff
skb_cloned -- is the buffer a clone
skb_shared -- is the buffer shared
skb_share_check -- check if buffer is shared and if so clone it
skb_unshare -- make a copy of a shared buffer
skb_peek --
skb_peek_tail --
skb_queue_len -- get queue length
__skb_queue_head -- queue a buffer at the list head
skb_queue_head -- queue a buffer at the list head
__skb_queue_tail -- queue a buffer at the list tail
skb_queue_tail -- queue a buffer at the list tail
__skb_dequeue -- remove from the head of the queue
skb_dequeue -- remove from the head of the queue
skb_insert -- insert a buffer
skb_append -- append a buffer
skb_unlink -- remove a buffer from a list
__skb_dequeue_tail -- remove from the tail of the queue
skb_dequeue_tail -- remove from the head of the queue
skb_put -- add data to a buffer
skb_push -- add data to the start of a buffer
skb_pull -- remove data from the start of a buffer
skb_headroom -- bytes at buffer head
skb_tailroom -- bytes at buffer end
skb_reserve -- adjust headroom
skb_trim -- remove end from a buffer
skb_orphan -- orphan a buffer
skb_queue_purge -- empty a list
__skb_queue_purge -- empty a list
__dev_alloc_skb -- allocate an skbuff for sending

dev_alloc_skb -- allocate an skbuff for sending
skb_cow -- copy header of skb when it is required
skb_padto -- pad an skbuff up to a minimal size
skb_over_panic -- private function
skb_under_panic -- private function
alloc_skb -- allocate a network buffer
__kfree_skb -- private function
skb_clone -- duplicate an sk_buff
skb_copy -- create private copy of an sk_buff
pskb_copy -- create copy of an sk_buff with private head.
pskb_expand_head -- reallocate header of `sk_buff`
skb_copy_expand -- copy and expand sk_buff
skb_pad -- zero pad the tail of an skb
__pskb_pull_tail -- advance tail of skb header

7.2. Socket Filter

sk_run_filter -- run a filter on a socket
sk_chk_filter -- verify socket filter code

8. Network device support

8.1. Driver Support

init_etherdev -- Register ethernet device
alloc_etherdev -- Allocates and sets up an ethernet device
alloc_fddidev -- Register FDDI device
alloc_hippi_dev -- Register HIPPI device
alloc_trdev -- Register token ring device
alloc_fcdev -- Register fibre channel device
dev_add_pack -- add packet handler
__dev_remove_pack -- remove packet handler
dev_remove_pack -- remove packet handler
netdev_boot_setup_check -- check boot time settings
__dev_get_by_name -- find a device by its name
dev_get_by_name -- find a device by its name
__dev_get -- test if a device exists
__dev_get_by_index -- find a device by its ifindex
dev_get_by_index -- find a device by its ifindex
dev_getbyhwaddr -- find a device by its hardware address
dev_get_by_flags -- find any device with given flags
__dev_get_by_flags -- find any device with given flags
dev_alloc_name -- allocate a name for a device
dev_alloc -- allocate a network device and name
netdev_state_change -- device changes state
dev_load -- load a network module
dev_open -- prepare an interface for use.
dev_close -- shutdown an interface.
register_netdevice_notifier -- register a network notifier block
unregister_netdevice_notifier -- unregister a network notifier block
call_netdevice_notifiers -- call all network notifier blocks
dev_queue_xmit -- transmit a buffer
netif_rx -- post buffer to the network code
register_gifconf -- register a SIOCGIF handler
netdev_set_master -- set up master/slave pair
dev_set_promiscuity -- update promiscuity count on a device
dev_set_allmulti -- update allmulti count on a device
dev_ioctl -- network device ioctl

dev_new_index -- allocate an ifindex
register_netdevice -- register a network device
free_netdev -- free network device
unregister_netdevice -- remove device from the kernel

8.2. 8390 Based Network Cards

ei_open -- Open/initialize the board.
ei_close -- shut down network device
ei_tx_timeout -- handle transmit time out condition
ei_interrupt -- handle the interrupts from an 8390
ethdev_init -- init rest of 8390 device struct
alloc_ei_netdev -- alloc_etherdev counterpart for 8390
NS8390_init -- initialize 8390 hardware

8.3. Synchronous PPP

sppp_input -- receive and process a WAN PPP frame
sppp_close -- close down a synchronous PPP or Cisco HDLC link
sppp_open -- open a synchronous PPP or Cisco HDLC link
sppp_reopen -- notify of physical link loss
sppp_change_mtu -- Change the link MTU
sppp_do_ioctl -- Ioctl handler for ppp/hdlc
sppp_attach -- attach synchronous PPP/HDLC to a device
sppp_detach -- release PPP resources from a device

9. Module Support

9.1. Module Loading

request_module -- try to load a kernel module
call_usermodehelper -- start a usermode application

9.2. Inter Module support

10. Hardware Interfaces

10.1. Interrupt Handling

disable_irq_nosync -- disable an irq without waiting
disable_irq -- disable an irq and wait for completion
enable_irq -- enable handling of an irq
probe_irq_mask -- scan a bitmap of interrupt lines

10.2. MTRR Handling

mtrr_add -- Add a memory type region
mtrr_del -- delete a memory type region

10.3. PCI Support Library

pci_bus_max_busnr -- returns maximum PCI bus number of given bus' children
pci_max_busnr -- returns maximum PCI bus number
pci_find_capability -- query for devices' capabilities
pci_bus_find_capability -- query for devices' capabilities
pci_find_parent_resource -- return resource region of parent bus of given region
pci_set_power_state -- Set the power state of a PCI device
pci_save_state -- save the PCI configuration space of a device before suspending
pci_restore_state -- Restore the saved state of a PCI device

pci_enable_device_bars -- Initialize some of a device for use
pci_enable_device -- Initialize device before it's used by a driver.
pci_disable_device -- Disable PCI device after use
pci_enable_wake -- enable device to generate PME# when suspended
pci_release_region -- Release a PCI bar
pci_request_region -- Reserved PCI I/O and memory resource
pci_release_regions -- Release reserved PCI I/O and memory resources
pci_request_regions -- Reserved PCI I/O and memory resources
pci_set_master -- enables bus-mastering for device dev
pci_set_mwi -- enables memory-write-invalidate PCI transaction
pci_clear_mwi -- disables Memory-Write-Invalidate for device dev

10.4. PCI Hotplug Support Library

pci_hp_register -- register a hotplug_slot with the PCI hotplug subsystem
pci_hp_deregister -- deregister a hotplug_slot with the PCI hotplug subsystem
pci_hp_change_slot_info -- changes the slot's information structure in the core

10.5. MCA Architecture

10.5.1. MCA Device Functions
10.5.2. MCA Bus DMA

11. The Device File System

devfs_put -- Put (release) a reference to a devfs entry.
devfs_mk_symlink --
devfs_mk_dir -- Create a directory in the devfs namespace.

12. Security Framework

register_security -- registers a security framework with the kernel
unregister_security -- unregisters a security framework with the kernel
mod_reg_security -- allows security modules to be "stacked"
mod_unreg_security -- allows a security module registered with `mod_reg_security` to be unloaded
capable -- calls the currently loaded security module's `capable` function with the specified capability

13. Power Management

pm_register -- register a device with power management
pm_unregister -- unregister a device with power management
pm_unregister_all -- unregister all devices with matching callback
pm_send -- send request to a single device
pm_send_all -- send request to all managed devices
pm_find -- find a device

14. Block Devices

blk_queue_prep_rq -- set a prepare_request function for queue
blk_queue_merge_bvec -- set a merge_bvec function for queue
blk_queue_make_request -- define an alternate make_request function for a device
blk_queue_bounce_limit -- set bounce buffer limit for queue
blk_queue_max_sectors -- set max sectors for a request for this queue
blk_queue_max_phys_segments -- set max phys segments for a request for this queue
blk_queue_max_hw_segments -- set max hw segments for a request for this queue
blk_queue_max_segment_size -- set max segment size for blk_rq_map_sg
blk_queue_hardsect_size -- set hardware sector size for the queue
blk_queue_stack_limits -- inherit underlying queue limits for stacked drivers
blk_queue_segment_boundary -- set boundary rules for segment merging

blk_queue_dma_alignment -- set dma length and memory alignment
blk_queue_find_tag -- find a request by its tag and queue
blk_queue_free_tags -- release tag maintenance info
blk_queue_init_tags -- initialize the queue tag info
blk_queue_end_tag -- end tag operations for a request
blk_queue_start_tag -- find a free tag and assign it
blk_queue_invalidate_tags -- invalidate all pending tags
generic_unplug_device -- fire a request queue
blk_start_queue -- restart a previously stopped queue
blk_stop_queue -- stop a queue
blk_run_queue -- run a single device queue
blk_cleanup_queue -- release a `request_queue_t` when it is no longer needed
blk_init_queue -- prepare a request queue for use with a block device
blk_requeue_request -- put a request back on queue
blk_insert_request -- insert a special request in to a request queue
blk_congestion_wait -- wait for a queue to become uncongested
blk_attempt_remerge -- attempt to remerge active head with next request
generic_make_request --
submit_bio --
process_that_request_first -- process partial request submission
end_that_request_first -- end I/O on a request
end_that_request_chunk -- end I/O on a request

15. Miscellaneous Devices

misc_register -- register a miscellaneous device
misc_deregister -- unregister a miscellaneous device

16. Video4Linux

video_register_device -- register video4linux devices
video_unregister_device -- unregister a video4linux device

17. Sound Devices

register_sound_special -- register a special sound node
register_sound_mixer -- register a mixer device
register_sound_midi -- register a midi device
register_sound_dsp -- register a DSP device
register_sound_synth -- register a synth device
unregister_sound_special -- unregister a special sound device
unregister_sound_mixer -- unregister a mixer
unregister_sound_midi -- unregister a midi device
unregister_sound_dsp -- unregister a DSP device
unregister_sound_synth -- unregister a synth device

18. 16x50 UART Driver

uart_update_timeout -- update per-port FIFO timeout.
uart_get_baud_rate -- return baud rate for a particular port
uart_get_divisor -- return uart clock divisor
uart_register_driver -- register a driver with the uart core layer
uart_unregister_driver -- remove a driver from the uart core layer
uart_add_one_port -- attach a driver-defined port structure
uart_remove_one_port -- detach a driver defined port structure
uart_register_port --
uart_unregister_port -- de-allocate a port
register_serial -- configure a 16x50 serial port at runtime
unregister_serial -- remove a 16x50 serial port at runtime

serial8250_suspend_port -- suspend one serial port
serial8250_resume_port -- resume one serial port

19. Z85230 Support Library

z8530_interrupt -- Handle an interrupt from a Z8530
z8530_sync_open -- Open a Z8530 channel for PIO
z8530_sync_close -- Close a PIO Z8530 channel
z8530_sync_dma_open -- Open a Z8530 for DMA I/O
z8530_sync_dma_close -- Close down DMA I/O
z8530_sync_txdma_open -- Open a Z8530 for TX driven DMA
z8530_sync_txdma_close -- Close down a TX driven DMA channel
z8530_describe -- Uniformly describe a Z8530 port
z8530_init -- Initialise a Z8530 device
z8530_shutdown -- Shutdown a Z8530 device
z8530_channel_load -- Load channel data
z8530_null_rx -- Discard a packet
z8530_queue_xmit -- Queue a packet
z8530_get_stats -- Get network statistics

20. Frame Buffer Library

20.1. Frame Buffer Memory

register_framebuffer -- registers a frame buffer device
unregister_framebuffer -- releases a frame buffer device

20.2. Frame Buffer Console

drivers/video/console/fbcon.c -- Document generation inconsistency

20.3. Frame Buffer Colormap

fb_alloc_cmap -- allocate a colormap
fb_dealloc_cmap -- deallocate a colormap
fb_copy_cmap -- copy a colormap
fb_set_cmap -- set the colormap
fb_default_cmap -- get default colormap
fb_invert_cmaps -- invert all defaults colormaps

20.4. Frame Buffer Video Mode Database

fb_find_mode -- finds a valid video mode
__fb_try_mode -- test a video mode

20.5. Frame Buffer Macintosh Video Mode Database

mac_vmode_to_var -- converts vmode/cmode pair to var structure
mac_var_to_vmode -- convert var structure to MacOS vmode/cmode pair
mac_map_monitor_sense -- Convert monitor sense to vmode
mac_find_mode -- find a video mode

20.6. Frame Buffer Fonts

Next
Driver Basics