Linux kernel v.5.15.32

struct [**super\_block**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/super_block) {

struct [**list\_head**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/list_head) [**s\_list**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_list); */\* Keep this first \*/*

[**dev\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dev_t) [**s\_dev**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_dev); */\* search index; \_not\_ kdev\_t \*/*

unsigned char [**s\_blocksize\_bits**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_blocksize_bits);

unsigned long [**s\_blocksize**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_blocksize);

[**loff\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/loff_t) [**s\_maxbytes**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_maxbytes); */\* Max file size \*/*

struct [**file\_system\_type**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file_system_type) \*[**s\_type**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_type);

const struct [**super\_operations**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/super_operations) \*[**s\_op**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_op);

const struct [**dquot\_operations**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dquot_operations) \*[**dq\_op**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dq_op);

const struct [**quotactl\_ops**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/quotactl_ops) \*[**s\_qcop**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_qcop);

const struct [**export\_operations**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/export_operations) \*[**s\_export\_op**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_export_op);

unsigned long [**s\_flags**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_flags);

unsigned long [**s\_iflags**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_iflags); */\* internal SB\_I\_\* flags \*/*

unsigned long [**s\_magic**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_magic);

struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) \*[**s\_root**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_root);

struct [**rw\_semaphore**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/rw_semaphore) [**s\_umount**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_umount);

int [**s\_count**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_count);

[**atomic\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/atomic_t) [**s\_active**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_active);

#ifdef [**CONFIG\_SECURITY**](https://elixir.bootlin.com/linux/v5.15.32/K/ident/CONFIG_SECURITY)

void \*[**s\_security**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_security);

#endif

const struct [**xattr\_handler**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/xattr_handler) \*\*[**s\_xattr**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_xattr);

#ifdef [**CONFIG\_FS\_ENCRYPTION**](https://elixir.bootlin.com/linux/v5.15.32/K/ident/CONFIG_FS_ENCRYPTION)

const struct [**fscrypt\_operations**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/fscrypt_operations) \*[**s\_cop**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_cop);

struct [**key**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/key) \*[**s\_master\_keys**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_master_keys); */\* master crypto keys in use \*/*

#endif

#ifdef [**CONFIG\_FS\_VERITY**](https://elixir.bootlin.com/linux/v5.15.32/K/ident/CONFIG_FS_VERITY)

const struct [**fsverity\_operations**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/fsverity_operations) \*[**s\_vop**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_vop);

#endif

#ifdef [**CONFIG\_UNICODE**](https://elixir.bootlin.com/linux/v5.15.32/K/ident/CONFIG_UNICODE)

struct [**unicode\_map**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/unicode_map) \*[**s\_encoding**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_encoding);

[**\_\_u16**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/__u16) [**s\_encoding\_flags**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_encoding_flags);

#endif

struct [**hlist\_bl\_head**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/hlist_bl_head) [**s\_roots**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_roots); */\* alternate root dentries for NFS \*/*

struct [**list\_head**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/list_head) [**s\_mounts**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_mounts); */\* list of mounts; \_not\_ for fs use \*/*

struct [**block\_device**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/block_device) \*[**s\_bdev**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_bdev);

struct [**backing\_dev\_info**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/backing_dev_info) \*[**s\_bdi**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_bdi);

struct [**mtd\_info**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/mtd_info) \*[**s\_mtd**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_mtd);

struct [**hlist\_node**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/hlist_node) [**s\_instances**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_instances);

unsigned int [**s\_quota\_types**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_quota_types); */\* Bitmask of supported quota types \*/*

struct [**quota\_info**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/quota_info) [**s\_dquot**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_dquot); */\* Diskquota specific options \*/*

struct [**sb\_writers**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/sb_writers) [**s\_writers**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_writers);

*/\**

*\* Keep s\_fs\_info, s\_time\_gran, s\_fsnotify\_mask, and*

*\* s\_fsnotify\_marks together for cache efficiency. They are frequently*

*\* accessed and rarely modified.*

*\*/*

void \*[**s\_fs\_info**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_fs_info); */\* Filesystem private info \*/*

*/\* Granularity of c/m/atime in ns (cannot be worse than a second) \*/*

[**u32**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/u32) [**s\_time\_gran**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_time_gran);

*/\* Time limits for c/m/atime in seconds \*/*

[**time64\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/time64_t) [**s\_time\_min**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_time_min);

[**time64\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/time64_t) [**s\_time\_max**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_time_max);

#ifdef [**CONFIG\_FSNOTIFY**](https://elixir.bootlin.com/linux/v5.15.32/K/ident/CONFIG_FSNOTIFY)

[**\_\_u32**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/__u32) [**s\_fsnotify\_mask**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_fsnotify_mask);

struct [**fsnotify\_mark\_connector**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/fsnotify_mark_connector) [**\_\_rcu**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/__rcu) \*[**s\_fsnotify\_marks**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_fsnotify_marks);

#endif

char [**s\_id**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_id)[32]; */\* Informational name \*/*

[**uuid\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/uuid_t) [**s\_uuid**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_uuid); */\* UUID \*/*

unsigned int [**s\_max\_links**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_max_links);

[**fmode\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/fmode_t) [**s\_mode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_mode);

*/\**

*\* The next field is for VFS \*only\*. No filesystems have any business*

*\* even looking at it. You had been warned.*

*\*/*

struct [**mutex**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/mutex) [**s\_vfs\_rename\_mutex**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_vfs_rename_mutex); */\* Kludge \*/*

*/\**

*\* Filesystem subtype. If non-empty the filesystem type field*

*\* in /proc/mounts will be "type.subtype"*

*\*/*

const char \*[**s\_subtype**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_subtype);

const struct [**dentry\_operations**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry_operations) \*[**s\_d\_op**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_d_op); */\* default d\_op for dentries \*/*

*/\**

*\* Saved pool identifier for cleancache (-1 means none)*

*\*/*

int [**cleancache\_poolid**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/cleancache_poolid);

struct [**shrinker**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/shrinker) [**s\_shrink**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_shrink); */\* per-sb shrinker handle \*/*

*/\* Number of inodes with nlink == 0 but still referenced \*/*

[**atomic\_long\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/atomic_long_t) [**s\_remove\_count**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_remove_count);

*/\**

*\* Number of inode/mount/sb objects that are being watched, note that*

*\* inodes objects are currently double-accounted.*

*\*/*

[**atomic\_long\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/atomic_long_t) [**s\_fsnotify\_connectors**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_fsnotify_connectors);

*/\* Being remounted read-only \*/*

int [**s\_readonly\_remount**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_readonly_remount);

*/\* per-sb errseq\_t for reporting writeback errors via syncfs \*/*

[**errseq\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/errseq_t) [**s\_wb\_err**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_wb_err);

*/\* AIO completions deferred from interrupt context \*/*

struct [**workqueue\_struct**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/workqueue_struct) \*[**s\_dio\_done\_wq**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_dio_done_wq);

struct [**hlist\_head**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/hlist_head) [**s\_pins**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_pins);

*/\**

*\* Owning user namespace and default context in which to*

*\* interpret filesystem uids, gids, quotas, device nodes,*

*\* xattrs and security labels.*

*\*/*

struct [**user\_namespace**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/user_namespace) \*[**s\_user\_ns**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_user_ns);

*/\**

*\* The list\_lru structure is essentially just a pointer to a table*

*\* of per-node lru lists, each of which has its own spinlock.*

*\* There is no need to put them into separate cachelines.*

*\*/*

struct [**list\_lru**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/list_lru) [**s\_dentry\_lru**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_dentry_lru);

struct [**list\_lru**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/list_lru) [**s\_inode\_lru**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_inode_lru);

struct [**rcu\_head**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/rcu_head) [**rcu**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/rcu);

struct [**work\_struct**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/work_struct) [**destroy\_work**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/destroy_work);

struct [**mutex**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/mutex) [**s\_sync\_lock**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_sync_lock); */\* sync serialisation lock \*/*

*/\**

*\* Indicates how deep in a filesystem stack this SB is*

*\*/*

int [**s\_stack\_depth**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_stack_depth);

*/\* s\_inode\_list\_lock protects s\_inodes \*/*

[**spinlock\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/spinlock_t) s\_inode\_list\_lock [**\_\_\_\_cacheline\_aligned\_in\_smp**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/____cacheline_aligned_in_smp);

struct [**list\_head**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/list_head) [**s\_inodes**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_inodes); */\* all inodes \*/*

[**spinlock\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/spinlock_t) [**s\_inode\_wblist\_lock**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_inode_wblist_lock);

struct [**list\_head**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/list_head) [**s\_inodes\_wb**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_inodes_wb); */\* writeback inodes \*/*

} [**\_\_randomize\_layout**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/__randomize_layout);

struct [**super\_operations**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/super_operations) {

struct [**inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/inode) \*(\*[**alloc\_inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/alloc_inode))(struct [**super\_block**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/super_block) \*sb);

void (\*[**destroy\_inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/destroy_inode))(struct [**inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/inode) \*);

void (\*[**free\_inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/free_inode))(struct [**inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/inode) \*);

void (\*[**dirty\_inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dirty_inode)) (struct [**inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/inode) \*, int flags);

int (\*[**write\_inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/write_inode)) (struct [**inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/inode) \*, struct [**writeback\_control**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/writeback_control) \*[**wbc**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/wbc));

int (\*[**drop\_inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/drop_inode)) (struct [**inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/inode) \*);

void (\*[**evict\_inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/evict_inode)) (struct [**inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/inode) \*);

void (\*[**put\_super**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/put_super)) (struct [**super\_block**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/super_block) \*);

int (\*[**sync\_fs**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/sync_fs))(struct [**super\_block**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/super_block) \*sb, int [**wait**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/wait));

int (\*[**freeze\_super**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/freeze_super)) (struct [**super\_block**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/super_block) \*);

int (\*[**freeze\_fs**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/freeze_fs)) (struct [**super\_block**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/super_block) \*);

int (\*[**thaw\_super**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/thaw_super)) (struct [**super\_block**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/super_block) \*);

int (\*[**unfreeze\_fs**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/unfreeze_fs)) (struct [**super\_block**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/super_block) \*);

int (\*[**statfs**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/statfs)) (struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) \*, struct [**kstatfs**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/kstatfs) \*);

int (\*[**remount\_fs**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/remount_fs)) (struct [**super\_block**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/super_block) \*, int \*, char \*);

void (\*[**umount\_begin**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/umount_begin)) (struct [**super\_block**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/super_block) \*);

int (\*[**show\_options**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/show_options))(struct [**seq\_file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/seq_file) \*, struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) \*);

int (\*[**show\_devname**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/show_devname))(struct [**seq\_file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/seq_file) \*, struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) \*);

int (\*[**show\_path**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/show_path))(struct [**seq\_file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/seq_file) \*, struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) \*);

int (\*[**show\_stats**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/show_stats))(struct [**seq\_file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/seq_file) \*, struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) \*);

#ifdef [**CONFIG\_QUOTA**](https://elixir.bootlin.com/linux/v5.15.32/K/ident/CONFIG_QUOTA)

[**ssize\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/ssize_t) (\*[**quota\_read**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/quota_read))(struct [**super\_block**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/super_block) \*, int, char \*, [**size\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/size_t), [**loff\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/loff_t));

[**ssize\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/ssize_t) (\*[**quota\_write**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/quota_write))(struct [**super\_block**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/super_block) \*, int, const char \*, [**size\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/size_t), [**loff\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/loff_t));

struct [**dquot**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dquot) \*\*(\*[**get\_dquots**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/get_dquots))(struct [**inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/inode) \*);

#endif

long (\*[**nr\_cached\_objects**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/nr_cached_objects))(struct [**super\_block**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/super_block) \*,

struct [**shrink\_control**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/shrink_control) \*);

long (\*[**free\_cached\_objects**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/free_cached_objects))(struct [**super\_block**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/super_block) \*,

struct [**shrink\_control**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/shrink_control) \*);

};

struct [**file\_system\_type**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file_system_type) {

const char \*name;

int [**fs\_flags**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/fs_flags);

#define [**FS\_REQUIRES\_DEV**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/FS_REQUIRES_DEV) 1

#define [**FS\_BINARY\_MOUNTDATA**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/FS_BINARY_MOUNTDATA) 2

#define [**FS\_HAS\_SUBTYPE**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/FS_HAS_SUBTYPE) 4

#define [**FS\_USERNS\_MOUNT**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/FS_USERNS_MOUNT) 8 */\* Can be mounted by userns root \*/*

#define [**FS\_DISALLOW\_NOTIFY\_PERM**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/FS_DISALLOW_NOTIFY_PERM) 16 */\* Disable fanotify permission events \*/*

#define [**FS\_ALLOW\_IDMAP**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/FS_ALLOW_IDMAP) 32 */\* FS has been updated to handle vfs idmappings. \*/*

#define [**FS\_THP\_SUPPORT**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/FS_THP_SUPPORT) 8192 */\* Remove once all fs converted \*/*

#define [**FS\_RENAME\_DOES\_D\_MOVE**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/FS_RENAME_DOES_D_MOVE) 32768 */\* FS will handle d\_move() during rename() internally. \*/*

int (\*[**init\_fs\_context**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/init_fs_context))(struct [**fs\_context**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/fs_context) \*);

const struct [**fs\_parameter\_spec**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/fs_parameter_spec) \*[**parameters**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/parameters);

struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) \*(\*[**mount**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/mount)) (struct [**file\_system\_type**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file_system_type) \*, int,

const char \*, void \*);

void (\*[**kill\_sb**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/kill_sb)) (struct [**super\_block**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/super_block) \*);

struct [**module**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/module) \*[**owner**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/owner);

struct [**file\_system\_type**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file_system_type) \* next;

struct [**hlist\_head**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/hlist_head) [**fs\_supers**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/fs_supers);

struct [**lock\_class\_key**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/lock_class_key) [**s\_lock\_key**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_lock_key);

struct [**lock\_class\_key**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/lock_class_key) [**s\_umount\_key**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_umount_key);

struct [**lock\_class\_key**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/lock_class_key) [**s\_vfs\_rename\_key**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_vfs_rename_key);

struct [**lock\_class\_key**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/lock_class_key) [**s\_writers\_key**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/s_writers_key)[[**SB\_FREEZE\_LEVELS**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/SB_FREEZE_LEVELS)];

struct [**lock\_class\_key**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/lock_class_key) [**i\_lock\_key**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_lock_key);

struct [**lock\_class\_key**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/lock_class_key) [**i\_mutex\_key**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_mutex_key);

struct [**lock\_class\_key**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/lock_class_key) [**invalidate\_lock\_key**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/invalidate_lock_key);

struct [**lock\_class\_key**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/lock_class_key) [**i\_mutex\_dir\_key**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_mutex_dir_key);

};

struct [**vfsmount**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/vfsmount) {

struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) \*[**mnt\_root**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/mnt_root); */\* root of the mounted tree \*/*

struct [**super\_block**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/super_block) \*[**mnt\_sb**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/mnt_sb); */\* pointer to superblock \*/*

int [**mnt\_flags**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/mnt_flags);

struct [**user\_namespace**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/user_namespace) \*[**mnt\_userns**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/mnt_userns);

} [**\_\_randomize\_layout**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/__randomize_layout);

struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) {

*/\* RCU lookup touched fields \*/*

unsigned int [**d\_flags**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/d_flags); */\* protected by d\_lock \*/*

seqcount\_spinlock\_t [**d\_seq**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/d_seq); */\* per dentry seqlock \*/*

struct [**hlist\_bl\_node**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/hlist_bl_node) [**d\_hash**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/d_hash); */\* lookup hash list \*/*

struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) \*[**d\_parent**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/d_parent); */\* parent directory \*/*

struct [**qstr**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/qstr) [**d\_name**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/d_name);

struct [**inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/inode) \*[**d\_inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/d_inode); */\* Where the name belongs to - NULL is*

*\* negative \*/*

unsigned char [**d\_iname**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/d_iname)[[**DNAME\_INLINE\_LEN**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/DNAME_INLINE_LEN)]; */\* small names \*/*

*/\* Ref lookup also touches following \*/*

struct [**lockref**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/lockref) [**d\_lockref**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/d_lockref); */\* per-dentry lock and refcount \*/*

const struct [**dentry\_operations**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry_operations) \*[**d\_op**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/d_op);

struct [**super\_block**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/super_block) \*[**d\_sb**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/d_sb); */\* The root of the dentry tree \*/*

unsigned long [**d\_time**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/d_time); */\* used by d\_revalidate \*/*

void \*[**d\_fsdata**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/d_fsdata); */\* fs-specific data \*/*

union {

struct [**list\_head**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/list_head) [**d\_lru**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/d_lru); */\* LRU list \*/*

[**wait\_queue\_head\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/wait_queue_head_t) \*[**d\_wait**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/d_wait); */\* in-lookup ones only \*/*

};

struct [**list\_head**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/list_head) [**d\_child**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/d_child); */\* child of parent list \*/*

struct [**list\_head**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/list_head) [**d\_subdirs**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/d_subdirs); */\* our children \*/*

*/\**

*\* d\_alias and d\_rcu can share memory*

*\*/*

union {

struct [**hlist\_node**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/hlist_node) [**d\_alias**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/d_alias); */\* inode alias list \*/*

struct [**hlist\_bl\_node**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/hlist_bl_node) [**d\_in\_lookup\_hash**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/d_in_lookup_hash); */\* only for in-lookup ones \*/*

struct [**rcu\_head**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/rcu_head) [**d\_rcu**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/d_rcu);

} [**d\_u**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/d_u);

} [**\_\_randomize\_layout**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/__randomize_layout);

struct [**dentry\_operations**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry_operations) {

int (\*[**d\_revalidate**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/d_revalidate))(struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) \*, unsigned int);

int (\*[**d\_weak\_revalidate**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/d_weak_revalidate))(struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) \*, unsigned int);

int (\*[**d\_hash**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/d_hash))(const struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) \*, struct [**qstr**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/qstr) \*);

int (\*[**d\_compare**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/d_compare))(const struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) \*,

unsigned int, const char \*, const struct [**qstr**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/qstr) \*);

int (\*[**d\_delete**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/d_delete))(const struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) \*);

int (\*[**d\_init**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/d_init))(struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) \*);

void (\*[**d\_release**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/d_release))(struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) \*);

void (\*[**d\_prune**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/d_prune))(struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) \*);

void (\*[**d\_iput**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/d_iput))(struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) \*, struct [**inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/inode) \*);

char \*(\*[**d\_dname**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/d_dname))(struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) \*, char \*, int);

struct [**vfsmount**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/vfsmount) \*(\*[**d\_automount**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/d_automount))(struct [**path**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/path) \*);

int (\*[**d\_manage**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/d_manage))(const struct [**path**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/path) \*, [**bool**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/bool));

struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) \*(\*[**d\_real**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/d_real))(struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) \*, const struct [**inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/inode) \*);

} [**\_\_\_\_cacheline\_aligned**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/____cacheline_aligned);

struct [**dentry\_stat\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry_stat_t) {

long [**nr\_dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/nr_dentry);

long [**nr\_unused**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/nr_unused);

long [**age\_limit**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/age_limit); */\* age in seconds \*/*

long [**want\_pages**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/want_pages); */\* pages requested by system \*/*

long [**nr\_negative**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/nr_negative); */\* # of unused negative dentries \*/*

long [**dummy**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dummy); */\* Reserved for future use \*/*

};

extern struct [**dentry\_stat\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry_stat_t) [**dentry\_stat**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry_stat);

*/\**

*\* Keep mostly read-only and often accessed (especially for*

*\* the RCU path lookup and 'stat' data) fields at the beginning*

*\* of the 'struct inode'*

*\*/*

struct [**inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/inode) {

[**umode\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/umode_t) [**i\_mode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_mode);

unsigned [**short**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/short) [**i\_opflags**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_opflags);

[**kuid\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/kuid_t) [**i\_uid**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_uid);

[**kgid\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/kgid_t) [**i\_gid**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_gid);

unsigned int [**i\_flags**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_flags);

#ifdef [**CONFIG\_FS\_POSIX\_ACL**](https://elixir.bootlin.com/linux/v5.15.32/K/ident/CONFIG_FS_POSIX_ACL)

struct [**posix\_acl**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/posix_acl) \*[**i\_acl**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_acl);

struct [**posix\_acl**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/posix_acl) \*[**i\_default\_acl**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_default_acl);

#endif

const struct [**inode\_operations**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/inode_operations) \*[**i\_op**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_op);

struct [**super\_block**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/super_block) \*[**i\_sb**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_sb);

struct [**address\_space**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/address_space) \*[**i\_mapping**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_mapping);

#ifdef [**CONFIG\_SECURITY**](https://elixir.bootlin.com/linux/v5.15.32/K/ident/CONFIG_SECURITY)

void \*[**i\_security**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_security);

#endif

*/\* Stat data, not accessed from path walking \*/*

unsigned long [**i\_ino**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_ino);

*/\**

*\* Filesystems may only read i\_nlink directly. They shall use the*

*\* following functions for modification:*

*\**

*\* (set|clear|inc|drop)\_nlink*

*\* inode\_(inc|dec)\_link\_count*

*\*/*

union {

const unsigned int [**i\_nlink**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_nlink);

unsigned int [**\_\_i\_nlink**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/__i_nlink);

};

[**dev\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dev_t) [**i\_rdev**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_rdev);

[**loff\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/loff_t) [**i\_size**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_size);

struct [**timespec64**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/timespec64) [**i\_atime**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_atime);

struct [**timespec64**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/timespec64) [**i\_mtime**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_mtime);

struct [**timespec64**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/timespec64) [**i\_ctime**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_ctime);

[**spinlock\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/spinlock_t) [**i\_lock**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_lock); */\* i\_blocks, i\_bytes, maybe i\_size \*/*

unsigned [**short**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/short) [**i\_bytes**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_bytes);

[**u8**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/u8) [**i\_blkbits**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_blkbits);

[**u8**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/u8) [**i\_write\_hint**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_write_hint);

[**blkcnt\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/blkcnt_t) [**i\_blocks**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_blocks);

#ifdef [**\_\_NEED\_I\_SIZE\_ORDERED**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/__NEED_I_SIZE_ORDERED)

[**seqcount\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/seqcount_t) [**i\_size\_seqcount**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_size_seqcount);

#endif

*/\* Misc \*/*

unsigned long [**i\_state**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_state);

struct [**rw\_semaphore**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/rw_semaphore) [**i\_rwsem**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_rwsem);

unsigned long [**dirtied\_when**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dirtied_when); */\* jiffies of first dirtying \*/*

unsigned long [**dirtied\_time\_when**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dirtied_time_when);

struct [**hlist\_node**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/hlist_node) [**i\_hash**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_hash);

struct [**list\_head**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/list_head) [**i\_io\_list**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_io_list); */\* backing dev IO list \*/*

#ifdef [**CONFIG\_CGROUP\_WRITEBACK**](https://elixir.bootlin.com/linux/v5.15.32/K/ident/CONFIG_CGROUP_WRITEBACK)

struct [**bdi\_writeback**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/bdi_writeback) \*[**i\_wb**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_wb); */\* the associated cgroup wb \*/*

*/\* foreign inode detection, see wbc\_detach\_inode() \*/*

int [**i\_wb\_frn\_winner**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_wb_frn_winner);

[**u16**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/u16) [**i\_wb\_frn\_avg\_time**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_wb_frn_avg_time);

[**u16**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/u16) [**i\_wb\_frn\_history**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_wb_frn_history);

#endif

struct [**list\_head**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/list_head) [**i\_lru**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_lru); */\* inode LRU list \*/*

struct [**list\_head**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/list_head) [**i\_sb\_list**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_sb_list);

struct [**list\_head**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/list_head) [**i\_wb\_list**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_wb_list); */\* backing dev writeback list \*/*

union {

struct [**hlist\_head**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/hlist_head) [**i\_dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_dentry);

struct [**rcu\_head**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/rcu_head) [**i\_rcu**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_rcu);

};

[**atomic64\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/atomic64_t) [**i\_version**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_version);

[**atomic64\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/atomic64_t) [**i\_sequence**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_sequence); */\* see futex \*/*

[**atomic\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/atomic_t) [**i\_count**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_count);

[**atomic\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/atomic_t) [**i\_dio\_count**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_dio_count);

[**atomic\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/atomic_t) [**i\_writecount**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_writecount);

#if [**defined**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/defined)([**CONFIG\_IMA**](https://elixir.bootlin.com/linux/v5.15.32/K/ident/CONFIG_IMA)) || [**defined**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/defined)([**CONFIG\_FILE\_LOCKING**](https://elixir.bootlin.com/linux/v5.15.32/K/ident/CONFIG_FILE_LOCKING))

[**atomic\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/atomic_t) [**i\_readcount**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_readcount); */\* struct files open RO \*/*

#endif

union {

const struct [**file\_operations**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file_operations) \*[**i\_fop**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_fop); */\* former ->i\_op->default\_file\_ops \*/*

void (\*[**free\_inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/free_inode))(struct [**inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/inode) \*);

};

struct [**file\_lock\_context**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file_lock_context) \*[**i\_flctx**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_flctx);

struct [**address\_space**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/address_space) [**i\_data**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_data);

struct [**list\_head**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/list_head) [**i\_devices**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_devices);

union {

struct [**pipe\_inode\_info**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/pipe_inode_info) \*[**i\_pipe**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_pipe);

struct [**cdev**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/cdev) \*[**i\_cdev**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_cdev);

char \*[**i\_link**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_link);

unsigned [**i\_dir\_seq**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_dir_seq);

};

[**\_\_u32**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/__u32) [**i\_generation**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_generation);

#ifdef [**CONFIG\_FSNOTIFY**](https://elixir.bootlin.com/linux/v5.15.32/K/ident/CONFIG_FSNOTIFY)

[**\_\_u32**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/__u32) [**i\_fsnotify\_mask**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_fsnotify_mask); */\* all events this inode cares about \*/*

struct [**fsnotify\_mark\_connector**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/fsnotify_mark_connector) [**\_\_rcu**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/__rcu) \*[**i\_fsnotify\_marks**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_fsnotify_marks);

#endif

#ifdef [**CONFIG\_FS\_ENCRYPTION**](https://elixir.bootlin.com/linux/v5.15.32/K/ident/CONFIG_FS_ENCRYPTION)

struct [**fscrypt\_info**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/fscrypt_info) \*[**i\_crypt\_info**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_crypt_info);

#endif

#ifdef [**CONFIG\_FS\_VERITY**](https://elixir.bootlin.com/linux/v5.15.32/K/ident/CONFIG_FS_VERITY)

struct [**fsverity\_info**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/fsverity_info) \*[**i\_verity\_info**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_verity_info);

#endif

void \*[**i\_private**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_private); */\* fs or device private pointer \*/*

} [**\_\_randomize\_layout**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/__randomize_layout);

struct [**inode\_operations**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/inode_operations) {

struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) \* (\*[**lookup**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/lookup)) (struct [**inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/inode) \*,struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) \*, unsigned int);

const char \* (\*[**get\_link**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/get_link)) (struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) \*, struct [**inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/inode) \*, struct [**delayed\_call**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/delayed_call) \*);

int (\*[**permission**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/permission)) (struct [**user\_namespace**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/user_namespace) \*, struct [**inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/inode) \*, int);

struct [**posix\_acl**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/posix_acl) \* (\*[**get\_acl**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/get_acl))(struct [**inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/inode) \*, int, [**bool**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/bool));

int (\*[**readlink**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/readlink)) (struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) \*, char [**\_\_user**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/__user) \*,int);

int (\*[**create**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/create)) (struct [**user\_namespace**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/user_namespace) \*, struct [**inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/inode) \*,struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) \*,

[**umode\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/umode_t), [**bool**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/bool));

int (\*[**link**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/link)) (struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) \*,struct [**inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/inode) \*,struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) \*);

int (\*[**unlink**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/unlink)) (struct [**inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/inode) \*,struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) \*);

int (\*[**symlink**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/symlink)) (struct [**user\_namespace**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/user_namespace) \*, struct [**inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/inode) \*,struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) \*,

const char \*);

int (\*[**mkdir**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/mkdir)) (struct [**user\_namespace**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/user_namespace) \*, struct [**inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/inode) \*,struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) \*,

[**umode\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/umode_t));

int (\*[**rmdir**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/rmdir)) (struct [**inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/inode) \*,struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) \*);

int (\*[**mknod**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/mknod)) (struct [**user\_namespace**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/user_namespace) \*, struct [**inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/inode) \*,struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) \*,

[**umode\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/umode_t),[**dev\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dev_t));

int (\*[**rename**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/rename)) (struct [**user\_namespace**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/user_namespace) \*, struct [**inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/inode) \*, struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) \*,

struct [**inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/inode) \*, struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) \*, unsigned int);

int (\*[**setattr**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/setattr)) (struct [**user\_namespace**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/user_namespace) \*, struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) \*,

struct [**iattr**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/iattr) \*);

int (\*[**getattr**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/getattr)) (struct [**user\_namespace**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/user_namespace) \*, const struct [**path**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/path) \*,

struct [**kstat**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/kstat) \*, [**u32**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/u32), unsigned int);

[**ssize\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/ssize_t) (\*[**listxattr**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/listxattr)) (struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) \*, char \*, [**size\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/size_t));

int (\*[**fiemap**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/fiemap))(struct [**inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/inode) \*, struct [**fiemap\_extent\_info**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/fiemap_extent_info) \*, [**u64**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/u64) start,

[**u64**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/u64) len);

int (\*[**update\_time**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/update_time))(struct [**inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/inode) \*, struct [**timespec64**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/timespec64) \*, int);

int (\*[**atomic\_open**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/atomic_open))(struct [**inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/inode) \*, struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) \*,

struct [**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file) \*, unsigned [**open\_flag**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/open_flag),

[**umode\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/umode_t) [**create\_mode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/create_mode));

int (\*[**tmpfile**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/tmpfile)) (struct [**user\_namespace**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/user_namespace) \*, struct [**inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/inode) \*,

struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) \*, [**umode\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/umode_t));

int (\*[**set\_acl**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/set_acl))(struct [**user\_namespace**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/user_namespace) \*, struct [**inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/inode) \*,

struct [**posix\_acl**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/posix_acl) \*, int);

int (\*[**fileattr\_set**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/fileattr_set))(struct [**user\_namespace**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/user_namespace) \*[**mnt\_userns**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/mnt_userns),

struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) \*[**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry), struct [**fileattr**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/fileattr) \*[**fa**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/fa));

int (\*[**fileattr\_get**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/fileattr_get))(struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) \*[**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry), struct [**fileattr**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/fileattr) \*[**fa**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/fa));

} [**\_\_\_\_cacheline\_aligned**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/____cacheline_aligned);

*/\**

*\* Inode flags - they have no relation to superblock flags now*

*\*/*

#define [**S\_SYNC**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/S_SYNC) (1 << 0) */\* Writes are synced at once \*/*

#define [**S\_NOATIME**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/S_NOATIME) (1 << 1) */\* Do not update access times \*/*

#define [**S\_APPEND**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/S_APPEND) (1 << 2) */\* Append-only file \*/*

#define [**S\_IMMUTABLE**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/S_IMMUTABLE) (1 << 3) */\* Immutable file \*/*

#define [**S\_DEAD**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/S_DEAD) (1 << 4) */\* removed, but still open directory \*/*

#define [**S\_NOQUOTA**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/S_NOQUOTA) (1 << 5) */\* Inode is not counted to quota \*/*

#define [**S\_DIRSYNC**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/S_DIRSYNC) (1 << 6) */\* Directory modifications are synchronous \*/*

#define [**S\_NOCMTIME**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/S_NOCMTIME) (1 << 7) */\* Do not update file c/mtime \*/*

#define [**S\_SWAPFILE**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/S_SWAPFILE) (1 << 8) */\* Do not truncate: swapon got its bmaps \*/*

#define [**S\_PRIVATE**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/S_PRIVATE) (1 << 9) */\* Inode is fs-internal \*/*

#define [**S\_IMA**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/S_IMA) (1 << 10) */\* Inode has an associated IMA struct \*/*

#define [**S\_AUTOMOUNT**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/S_AUTOMOUNT) (1 << 11) */\* Automount/referral quasi-directory \*/*

#define [**S\_NOSEC**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/S_NOSEC) (1 << 12) */\* no suid or xattr security attributes \*/*

#ifdef [**CONFIG\_FS\_DAX**](https://elixir.bootlin.com/linux/v5.15.32/K/ident/CONFIG_FS_DAX)

#define [**S\_DAX**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/S_DAX) (1 << 13) */\* Direct Access, avoiding the page cache \*/*

#else

#define [**S\_DAX**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/S_DAX) 0 */\* Make all the DAX code disappear \*/*

#endif

#define [**S\_ENCRYPTED**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/S_ENCRYPTED) (1 << 14) */\* Encrypted file (using fs/crypto/) \*/*

#define [**S\_CASEFOLD**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/S_CASEFOLD) (1 << 15) */\* Casefolded file \*/*

#define [**S\_VERITY**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/S_VERITY) (1 << 16) */\* Verity file (using fs/verity/) \*/*

*/\*\**

*\* struct address\_space - Contents of a cacheable, mappable object.*

*\* @host: Owner, either the inode or the block\_device.*

*\* @i\_pages: Cached pages.*

*\* @invalidate\_lock: Guards coherency between page cache contents and*

*\* file offset->disk block mappings in the filesystem during invalidates.*

*\* It is also used to block modification of page cache contents through*

*\* memory mappings.*

*\* @gfp\_mask: Memory allocation flags to use for allocating pages.*

*\* @i\_mmap\_writable: Number of VM\_SHARED mappings.*

*\* @nr\_thps: Number of THPs in the pagecache (non-shmem only).*

*\* @i\_mmap: Tree of private and shared mappings.*

*\* @i\_mmap\_rwsem: Protects @i\_mmap and @i\_mmap\_writable.*

*\* @nrpages: Number of page entries, protected by the i\_pages lock.*

*\* @writeback\_index: Writeback starts here.*

*\* @a\_ops: Methods.*

*\* @flags: Error bits and flags (AS\_\*).*

*\* @wb\_err: The most recent error which has occurred.*

*\* @private\_lock: For use by the owner of the address\_space.*

*\* @private\_list: For use by the owner of the address\_space.*

*\* @private\_data: For use by the owner of the address\_space.*

*\*/*

struct [**address\_space**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/address_space) {

struct [**inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/inode) \*host;

struct [**xarray**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/xarray) [**i\_pages**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_pages);

struct [**rw\_semaphore**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/rw_semaphore) [**invalidate\_lock**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/invalidate_lock);

[**gfp\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/gfp_t) [**gfp\_mask**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/gfp_mask);

[**atomic\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/atomic_t) [**i\_mmap\_writable**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_mmap_writable);

#ifdef [**CONFIG\_READ\_ONLY\_THP\_FOR\_FS**](https://elixir.bootlin.com/linux/v5.15.32/K/ident/CONFIG_READ_ONLY_THP_FOR_FS)

*/\* number of thp, only for non-shmem files \*/*

[**atomic\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/atomic_t) [**nr\_thps**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/nr_thps);

#endif

struct [**rb\_root\_cached**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/rb_root_cached) [**i\_mmap**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_mmap);

struct [**rw\_semaphore**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/rw_semaphore) [**i\_mmap\_rwsem**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/i_mmap_rwsem);

unsigned long [**nrpages**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/nrpages);

[**pgoff\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/pgoff_t) [**writeback\_index**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/writeback_index);

const struct [**address\_space\_operations**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/address_space_operations) \*[**a\_ops**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/a_ops);

unsigned long flags;

[**errseq\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/errseq_t) [**wb\_err**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/wb_err);

[**spinlock\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/spinlock_t) [**private\_lock**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/private_lock);

struct [**list\_head**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/list_head) [**private\_list**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/private_list);

void \*[**private\_data**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/private_data);

} [**\_\_attribute\_\_**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/__attribute__)(([**aligned**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/aligned)(sizeof(long)))) [**\_\_randomize\_layout**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/__randomize_layout);

*/\**

*\* On most architectures that alignment is already the case; but*

*\* must be enforced here for CRIS, to let the least significant bit*

*\* of struct page's "mapping" pointer be used for PAGE\_MAPPING\_ANON.*

*\*/*

struct [**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file) {

union {

struct [**llist\_node**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/llist_node) [**fu\_llist**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/fu_llist);

struct [**rcu\_head**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/rcu_head) [**fu\_rcuhead**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/fu_rcuhead);

} [**f\_u**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/f_u);

struct [**path**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/path) [**f\_path**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/f_path);

struct [**inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/inode) \*[**f\_inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/f_inode); */\* cached value \*/*

const struct [**file\_operations**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file_operations) \*[**f\_op**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/f_op);

*/\**

*\* Protects f\_ep, f\_flags.*

*\* Must not be taken from IRQ context.*

*\*/*

[**spinlock\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/spinlock_t) [**f\_lock**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/f_lock);

enum [**rw\_hint**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/rw_hint) [**f\_write\_hint**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/f_write_hint);

[**atomic\_long\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/atomic_long_t) [**f\_count**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/f_count);

unsigned int [**f\_flags**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/f_flags);

[**fmode\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/fmode_t) [**f\_mode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/f_mode);

struct [**mutex**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/mutex) [**f\_pos\_lock**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/f_pos_lock);

[**loff\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/loff_t) [**f\_pos**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/f_pos);

struct [**fown\_struct**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/fown_struct) [**f\_owner**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/f_owner);

const struct [**cred**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/cred) \*[**f\_cred**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/f_cred);

struct [**file\_ra\_state**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file_ra_state) [**f\_ra**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/f_ra);

[**u64**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/u64) [**f\_version**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/f_version);

#ifdef [**CONFIG\_SECURITY**](https://elixir.bootlin.com/linux/v5.15.32/K/ident/CONFIG_SECURITY)

void \*[**f\_security**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/f_security);

#endif

*/\* needed for tty driver, and maybe others \*/*

void \*[**private\_data**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/private_data);

#ifdef [**CONFIG\_EPOLL**](https://elixir.bootlin.com/linux/v5.15.32/K/ident/CONFIG_EPOLL)

*/\* Used by fs/eventpoll.c to link all the hooks to this file \*/*

struct [**hlist\_head**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/hlist_head) \*[**f\_ep**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/f_ep);

#endif */\* #ifdef CONFIG\_EPOLL \*/*

struct [**address\_space**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/address_space) \*[**f\_mapping**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/f_mapping);

[**errseq\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/errseq_t) [**f\_wb\_err**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/f_wb_err);

[**errseq\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/errseq_t) [**f\_sb\_err**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/f_sb_err); */\* for syncfs \*/*

} [**\_\_randomize\_layout**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/__randomize_layout)

[**\_\_attribute\_\_**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/__attribute__)(([**aligned**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/aligned)(4))); */\* lest something weird decides that 2 is OK \*/*

struct [**file\_operations**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file_operations) {

struct [**module**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/module) \*[**owner**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/owner);

[**loff\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/loff_t) (\*[**llseek**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/llseek)) (struct [**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file) \*, [**loff\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/loff_t), int);

[**ssize\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/ssize_t) (\*[**read**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/read)) (struct [**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file) \*, char [**\_\_user**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/__user) \*, [**size\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/size_t), [**loff\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/loff_t) \*);

[**ssize\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/ssize_t) (\*[**write**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/write)) (struct [**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file) \*, const char [**\_\_user**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/__user) \*, [**size\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/size_t), [**loff\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/loff_t) \*);

[**ssize\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/ssize_t) (\*[**read\_iter**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/read_iter)) (struct [**kiocb**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/kiocb) \*, struct [**iov\_iter**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/iov_iter) \*);

[**ssize\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/ssize_t) (\*[**write\_iter**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/write_iter)) (struct [**kiocb**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/kiocb) \*, struct [**iov\_iter**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/iov_iter) \*);

int (\*[**iopoll**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/iopoll))(struct [**kiocb**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/kiocb) \*[**kiocb**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/kiocb), [**bool**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/bool) [**spin**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/spin));

int (\*[**iterate**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/iterate)) (struct [**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file) \*, struct [**dir\_context**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dir_context) \*);

int (\*[**iterate\_shared**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/iterate_shared)) (struct [**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file) \*, struct [**dir\_context**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dir_context) \*);

[**\_\_poll\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/__poll_t) (\*[**poll**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/poll)) (struct [**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file) \*, struct [**poll\_table\_struct**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/poll_table_struct) \*);

long (\*[**unlocked\_ioctl**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/unlocked_ioctl)) (struct [**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file) \*, unsigned int, unsigned long);

long (\*[**compat\_ioctl**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/compat_ioctl)) (struct [**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file) \*, unsigned int, unsigned long);

int (\*[**mmap**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/mmap)) (struct [**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file) \*, struct [**vm\_area\_struct**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/vm_area_struct) \*);

unsigned long [**mmap\_supported\_flags**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/mmap_supported_flags);

int (\*[**open**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/open)) (struct [**inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/inode) \*, struct [**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file) \*);

int (\*[**flush**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/flush)) (struct [**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file) \*, [**fl\_owner\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/fl_owner_t) id);

int (\*[**release**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/release)) (struct [**inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/inode) \*, struct [**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file) \*);

int (\*[**fsync**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/fsync)) (struct [**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file) \*, [**loff\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/loff_t), [**loff\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/loff_t), int [**datasync**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/datasync));

int (\*[**fasync**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/fasync)) (int, struct [**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file) \*, int);

int (\*lock) (struct [**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file) \*, int, struct [**file\_lock**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file_lock) \*);

[**ssize\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/ssize_t) (\*[**sendpage**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/sendpage)) (struct [**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file) \*, struct [**page**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/page) \*, int, [**size\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/size_t), [**loff\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/loff_t) \*, int);

unsigned long (\*[**get\_unmapped\_area**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/get_unmapped_area))(struct [**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file) \*, unsigned long, unsigned long, unsigned long, unsigned long);

int (\*[**check\_flags**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/check_flags))(int);

int (\*[**flock**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/flock)) (struct [**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file) \*, int, struct [**file\_lock**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file_lock) \*);

[**ssize\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/ssize_t) (\*[**splice\_write**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/splice_write))(struct [**pipe\_inode\_info**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/pipe_inode_info) \*, struct [**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file) \*, [**loff\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/loff_t) \*, [**size\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/size_t), unsigned int);

[**ssize\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/ssize_t) (\*[**splice\_read**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/splice_read))(struct [**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file) \*, [**loff\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/loff_t) \*, struct [**pipe\_inode\_info**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/pipe_inode_info) \*, [**size\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/size_t), unsigned int);

int (\*[**setlease**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/setlease))(struct [**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file) \*, long, struct [**file\_lock**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file_lock) \*\*, void \*\*);

long (\*[**fallocate**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/fallocate))(struct [**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file) \*[**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file), int mode, [**loff\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/loff_t) offset,

[**loff\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/loff_t) len);

void (\*[**show\_fdinfo**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/show_fdinfo))(struct [**seq\_file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/seq_file) \*m, struct [**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file) \*f);

#ifndef [**CONFIG\_MMU**](https://elixir.bootlin.com/linux/v5.15.32/K/ident/CONFIG_MMU)

unsigned (\*[**mmap\_capabilities**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/mmap_capabilities))(struct [**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file) \*);

#endif

[**ssize\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/ssize_t) (\*[**copy\_file\_range**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/copy_file_range))(struct [**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file) \*, [**loff\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/loff_t), struct [**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file) \*,

[**loff\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/loff_t), [**size\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/size_t), unsigned int);

[**loff\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/loff_t) (\*[**remap\_file\_range**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/remap_file_range))(struct [**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file) \*[**file\_in**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file_in), [**loff\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/loff_t) [**pos\_in**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/pos_in),

struct [**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file) \*[**file\_out**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file_out), [**loff\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/loff_t) [**pos\_out**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/pos_out),

[**loff\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/loff_t) len, unsigned int remap\_flags);

int (\*[**fadvise**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/fadvise))(struct [**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file) \*, [**loff\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/loff_t), [**loff\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/loff_t), int);

} [**\_\_randomize\_layout**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/__randomize_layout);

struct [**proc\_ops**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/proc_ops) {

unsigned int [**proc\_flags**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/proc_flags);

int (\*[**proc\_open**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/proc_open))(struct [**inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/inode) \*, struct [**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file) \*);

[**ssize\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/ssize_t) (\*[**proc\_read**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/proc_read))(struct [**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file) \*, char [**\_\_user**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/__user) \*, [**size\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/size_t), [**loff\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/loff_t) \*);

[**ssize\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/ssize_t) (\*[**proc\_read\_iter**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/proc_read_iter))(struct [**kiocb**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/kiocb) \*, struct [**iov\_iter**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/iov_iter) \*);

[**ssize\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/ssize_t) (\*[**proc\_write**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/proc_write))(struct [**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file) \*, const char [**\_\_user**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/__user) \*, [**size\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/size_t), [**loff\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/loff_t) \*);

*/\* mandatory unless nonseekable\_open() or equivalent is used \*/*

[**loff\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/loff_t) (\*[**proc\_lseek**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/proc_lseek))(struct [**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file) \*, [**loff\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/loff_t), int);

int (\*[**proc\_release**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/proc_release))(struct [**inode**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/inode) \*, struct [**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file) \*);

[**\_\_poll\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/__poll_t) (\*[**proc\_poll**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/proc_poll))(struct [**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file) \*, struct [**poll\_table\_struct**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/poll_table_struct) \*);

long (\*[**proc\_ioctl**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/proc_ioctl))(struct [**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file) \*, unsigned int, unsigned long);

#ifdef [**CONFIG\_COMPAT**](https://elixir.bootlin.com/linux/v5.15.32/K/ident/CONFIG_COMPAT)

long (\*[**proc\_compat\_ioctl**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/proc_compat_ioctl))(struct [**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file) \*, unsigned int, unsigned long);

#endif

int (\*[**proc\_mmap**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/proc_mmap))(struct [**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file) \*, struct [**vm\_area\_struct**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/vm_area_struct) \*);

unsigned long (\*[**proc\_get\_unmapped\_area**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/proc_get_unmapped_area))(struct [**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file) \*, unsigned long, unsigned long, unsigned long, unsigned long);

} [**\_\_randomize\_layout**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/__randomize_layout);

struct [**path**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/path) {

struct [**vfsmount**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/vfsmount) \*[**mnt**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/mnt);

struct [**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry) \*[**dentry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry);

} [**\_\_randomize\_layout**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/__randomize_layout);

struct [**softirq\_action**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/softirq_action)

{

void (\*[**action**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/action))(struct [**softirq\_action**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/softirq_action) \*);

};

struct [**proc\_dir\_entry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/proc_dir_entry) {

*/\**

*\* number of callers into module in progress;*

*\* negative -> it's going away RSN*

*\*/*

[**atomic\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/atomic_t) [**in\_use**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/in_use);

[**refcount\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/refcount_t) [**refcnt**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/refcnt);

struct [**list\_head**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/list_head) [**pde\_openers**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/pde_openers); */\* who did ->open, but not ->release \*/*

*/\* protects ->pde\_openers and all struct pde\_opener instances \*/*

[**spinlock\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/spinlock_t) [**pde\_unload\_lock**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/pde_unload_lock);

struct [**completion**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/completion) \*[**pde\_unload\_completion**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/pde_unload_completion);

const struct [**inode\_operations**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/inode_operations) \*[**proc\_iops**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/proc_iops);

union {

const struct [**proc\_ops**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/proc_ops) \*[**proc\_ops**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/proc_ops);

const struct [**file\_operations**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file_operations) \*[**proc\_dir\_ops**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/proc_dir_ops);

};

const struct [**dentry\_operations**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dentry_operations) \*[**proc\_dops**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/proc_dops);

union {

const struct [**seq\_operations**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/seq_operations) \*[**seq\_ops**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/seq_ops);

int (\*[**single\_show**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/single_show))(struct [**seq\_file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/seq_file) \*, void \*);

};

[**proc\_write\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/proc_write_t) [**write**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/write);

void \*data;

unsigned int [**state\_size**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/state_size);

unsigned int [**low\_ino**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/low_ino);

[**nlink\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/nlink_t) [**nlink**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/nlink);

[**kuid\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/kuid_t) [**uid**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/uid);

[**kgid\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/kgid_t) [**gid**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/gid);

[**loff\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/loff_t) size;

struct [**proc\_dir\_entry**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/proc_dir_entry) \*[**parent**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/parent);

struct [**rb\_root**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/rb_root) [**subdir**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/subdir);

struct [**rb\_node**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/rb_node) [**subdir\_node**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/subdir_node);

char \*name;

[**umode\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/umode_t) mode;

[**u8**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/u8) flags;

[**u8**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/u8) [**namelen**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/namelen);

char [**inline\_name**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/inline_name)[];

} [**\_\_randomize\_layout**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/__randomize_layout);

struct [**tasklet\_struct**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/tasklet_struct)

{

struct [**tasklet\_struct**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/tasklet_struct) \*next;

unsigned long state;

[**atomic\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/atomic_t) count;

[**bool**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/bool) [**use\_callback**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/use_callback);

union {

void (\*[**func**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/func))(unsigned long data);

void (\*[**callback**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/callback))(struct [**tasklet\_struct**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/tasklet_struct) \*t);

};

unsigned long data;

};

struct [**work\_struct**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/work_struct) {

[**atomic\_long\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/atomic_long_t) data;

struct [**list\_head**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/list_head) entry;

[**work\_func\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/work_func_t) [**func**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/func);

#ifdef [**CONFIG\_LOCKDEP**](https://elixir.bootlin.com/linux/v5.15.32/K/ident/CONFIG_LOCKDEP)

struct [**lockdep\_map**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/lockdep_map) [**lockdep\_map**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/lockdep_map);

#endif

};

struct [**delayed\_work**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/delayed_work) {

struct [**work\_struct**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/work_struct) [**work**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/work);

struct [**timer\_list**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/timer_list) [**timer**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/timer);

*/\* target workqueue and CPU ->timer uses to queue ->work \*/*

struct [**workqueue\_struct**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/workqueue_struct) \*[**wq**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/wq);

int cpu;

};

enum {

[**WQ\_UNBOUND**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/WQ_UNBOUND) = 1 << 1, */\* not bound to any cpu \*/*

[**WQ\_FREEZABLE**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/WQ_FREEZABLE) = 1 << 2, */\* freeze during suspend \*/*

[**WQ\_MEM\_RECLAIM**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/WQ_MEM_RECLAIM) = 1 << 3, */\* may be used for memory reclaim \*/*

[**WQ\_HIGHPRI**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/WQ_HIGHPRI) = 1 << 4, */\* high priority \*/*

[**WQ\_CPU\_INTENSIVE**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/WQ_CPU_INTENSIVE) = 1 << 5, */\* cpu intensive workqueue \*/*

[**WQ\_SYSFS**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/WQ_SYSFS) = 1 << 6, */\* visible in sysfs, see workqueue\_sysfs\_register() \*/*

*/\**

*\* Per-cpu workqueues are generally preferred because they tend to*

*\* show better performance thanks to cache locality. Per-cpu*

*\* workqueues exclude the scheduler from choosing the CPU to*

*\* execute the worker threads, which has an unfortunate side effect*

*\* of increasing power consumption.*

*\**

*\* The scheduler considers a CPU idle if it doesn't have any task*

*\* to execute and tries to keep idle cores idle to conserve power;*

*\* however, for example, a per-cpu work item scheduled from an*

*\* interrupt handler on an idle CPU will force the scheduler to*

*\* execute the work item on that CPU breaking the idleness, which in*

*\* turn may lead to more scheduling choices which are sub-optimal*

*\* in terms of power consumption.*

*\**

*\* Workqueues marked with WQ\_POWER\_EFFICIENT are per-cpu by default*

*\* but become unbound if workqueue.power\_efficient kernel param is*

*\* specified. Per-cpu workqueues which are identified to*

*\* contribute significantly to power-consumption are identified and*

*\* marked with this flag and enabling the power\_efficient mode*

*\* leads to noticeable power saving at the cost of small*

*\* performance disadvantage.*

*\**

*\* http://thread.gmane.org/gmane.linux.kernel/1480396*

*\*/*

[**WQ\_POWER\_EFFICIENT**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/WQ_POWER_EFFICIENT) = 1 << 7,

[**\_\_WQ\_DRAINING**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/__WQ_DRAINING) = 1 << 16, */\* internal: workqueue is draining \*/*

[**\_\_WQ\_ORDERED**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/__WQ_ORDERED) = 1 << 17, */\* internal: workqueue is ordered \*/*

[**\_\_WQ\_LEGACY**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/__WQ_LEGACY) = 1 << 18, */\* internal: create\*\_workqueue() \*/*

[**\_\_WQ\_ORDERED\_EXPLICIT**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/__WQ_ORDERED_EXPLICIT) = 1 << 19, */\* internal: alloc\_ordered\_workqueue() \*/*

[**WQ\_MAX\_ACTIVE**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/WQ_MAX_ACTIVE) = 512, */\* I like 512, better ideas? \*/*

[**WQ\_MAX\_UNBOUND\_PER\_CPU**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/WQ_MAX_UNBOUND_PER_CPU) = 4, */\* 4 \* #cpus for unbound wq \*/*

[**WQ\_DFL\_ACTIVE**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/WQ_DFL_ACTIVE) = [**WQ\_MAX\_ACTIVE**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/WQ_MAX_ACTIVE) / 2,

};

struct [**workqueue\_struct**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/workqueue_struct) {

struct [**list\_head**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/list_head) [**pwqs**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/pwqs); */\* WR: all pwqs of this wq \*/*

struct [**list\_head**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/list_head) list; */\* PR: list of all workqueues \*/*

struct [**mutex**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/mutex) [**mutex**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/mutex); */\* protects this wq \*/*

int [**work\_color**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/work_color); */\* WQ: current work color \*/*

int [**flush\_color**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/flush_color); */\* WQ: current flush color \*/*

[**atomic\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/atomic_t) [**nr\_pwqs\_to\_flush**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/nr_pwqs_to_flush); */\* flush in progress \*/*

struct [**wq\_flusher**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/wq_flusher) \*[**first\_flusher**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/first_flusher); */\* WQ: first flusher \*/*

struct [**list\_head**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/list_head) [**flusher\_queue**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/flusher_queue); */\* WQ: flush waiters \*/*

struct [**list\_head**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/list_head) [**flusher\_overflow**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/flusher_overflow); */\* WQ: flush overflow list \*/*

struct [**list\_head**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/list_head) [**maydays**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/maydays); */\* MD: pwqs requesting rescue \*/*

struct [**worker**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/worker) \*[**rescuer**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/rescuer); */\* MD: rescue worker \*/*

int [**nr\_drainers**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/nr_drainers); */\* WQ: drain in progress \*/*

int [**saved\_max\_active**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/saved_max_active); */\* WQ: saved pwq max\_active \*/*

struct [**workqueue\_attrs**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/workqueue_attrs) \*[**unbound\_attrs**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/unbound_attrs); */\* PW: only for unbound wqs \*/*

struct [**pool\_workqueue**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/pool_workqueue) \*[**dfl\_pwq**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/dfl_pwq); */\* PW: only for unbound wqs \*/*

#ifdef [**CONFIG\_SYSFS**](https://elixir.bootlin.com/linux/v5.15.32/K/ident/CONFIG_SYSFS)

struct [**wq\_device**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/wq_device) \*[**wq\_dev**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/wq_dev); */\* I: for sysfs interface \*/*

#endif

#ifdef [**CONFIG\_LOCKDEP**](https://elixir.bootlin.com/linux/v5.15.32/K/ident/CONFIG_LOCKDEP)

char \*[**lock\_name**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/lock_name);

struct [**lock\_class\_key**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/lock_class_key) [**key**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/key);

struct [**lockdep\_map**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/lockdep_map) [**lockdep\_map**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/lockdep_map);

#endif

char name[[**WQ\_NAME\_LEN**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/WQ_NAME_LEN)]; */\* I: workqueue name \*/*

*/\**

*\* Destruction of workqueue\_struct is RCU protected to allow walking*

*\* the workqueues list without grabbing wq\_pool\_mutex.*

*\* This is used to dump all workqueues from sysrq.*

*\*/*

struct [**rcu\_head**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/rcu_head) [**rcu**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/rcu);

*/\* hot fields used during command issue, aligned to cacheline \*/*

unsigned int flags [**\_\_\_\_cacheline\_aligned**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/____cacheline_aligned); */\* WQ: WQ\_\* flags \*/*

struct [**pool\_workqueue**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/pool_workqueue) [**\_\_percpu**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/__percpu) \*[**cpu\_pwqs**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/cpu_pwqs); */\* I: per-cpu pwqs \*/*

struct [**pool\_workqueue**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/pool_workqueue) [**\_\_rcu**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/__rcu) \*[**numa\_pwq\_tbl**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/numa_pwq_tbl)[]; */\* PWR: unbound pwqs indexed by node \*/*

};

*/\*\**

*\* struct socket - general BSD socket*

*\* @state: socket state (%SS\_CONNECTED, etc)*

*\* @type: socket type (%SOCK\_STREAM, etc)*

*\* @flags: socket flags (%SOCK\_NOSPACE, etc)*

*\* @ops: protocol specific socket operations*

*\* @file: File back pointer for gc*

*\* @sk: internal networking protocol agnostic socket representation*

*\* @wq: wait queue for several uses*

*\*/*

struct [**socket**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/socket) {

[**socket\_state**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/socket_state) state;

[**short**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/short) type;

unsigned long flags;

struct [**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file) \*[**file**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/file);

struct [**sock**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/sock) \*sk;

const struct [**proto\_ops**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/proto_ops) \*ops;

struct [**socket\_wq**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/socket_wq) [**wq**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/wq);

};

struct [**sockaddr**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/sockaddr) {

[**sa\_family\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/sa_family_t) [**sa\_family**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/sa_family); */\* address family, AF\_xxx \*/*

char [**sa\_data**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/sa_data)[14]; */\* 14 bytes of protocol address \*/*

};

*/\* Structure describing an Internet (IP) socket address. \*/*

#if [**\_\_UAPI\_DEF\_SOCKADDR\_IN**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/__UAPI_DEF_SOCKADDR_IN)

#define [**\_\_SOCK\_SIZE\_\_**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/__SOCK_SIZE__) 16 */\* sizeof(struct sockaddr) \*/*

struct [**sockaddr\_in**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/sockaddr_in) {

[**\_\_kernel\_sa\_family\_t**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/__kernel_sa_family_t) [**sin\_family**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/sin_family); */\* Address family \*/*

[**\_\_be16**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/__be16) [**sin\_port**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/sin_port); */\* Port number \*/*

struct [**in\_addr**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/in_addr) [**sin\_addr**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/sin_addr); */\* Internet address \*/*

*/\* Pad to size of `struct sockaddr'. \*/*

unsigned char [**\_\_pad**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/__pad)[[**\_\_SOCK\_SIZE\_\_**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/__SOCK_SIZE__) - sizeof([**short**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/short) int) -

sizeof(unsigned [**short**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/short) int) - sizeof(struct [**in\_addr**](https://elixir.bootlin.com/linux/v5.15.32/C/ident/in_addr))];

};