struct inode {

[**umode\_t**](https://elixir.bootlin.com/linux/latest/ident/umode_t) i\_mode; //Droits + type de fichier

unsigned short i\_opflags;

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

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

unsigned int i\_flags;

#ifdef CONFIG\_FS\_POSIX\_ACL

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

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

#endif

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

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

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

#ifdef CONFIG\_SECURITY

void \*i\_security;

#endif

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

unsigned long 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;

unsigned int \_\_i\_nlink;

};

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

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

struct [**timespec**](https://elixir.bootlin.com/linux/latest/ident/timespec) i\_atime;

struct [**timespec**](https://elixir.bootlin.com/linux/latest/ident/timespec) i\_mtime;

struct [**timespec**](https://elixir.bootlin.com/linux/latest/ident/timespec) i\_ctime;

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

unsigned short i\_bytes;

unsigned int i\_blkbits;

enum [**rw\_hint**](https://elixir.bootlin.com/linux/latest/ident/rw_hint) i\_write\_hint;

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

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

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

#endif

*/\* Misc \*/*

unsigned long i\_state;

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

unsigned long dirtied\_when; */\* jiffies of first dirtying \*/*

unsigned long dirtied\_time\_when;

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

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

#ifdef CONFIG\_CGROUP\_WRITEBACK

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

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

int i\_wb\_frn\_winner;

u16 i\_wb\_frn\_avg\_time;

u16 i\_wb\_frn\_history;

#endif

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

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

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

union {

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

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

};

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

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

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

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

#ifdef CONFIG\_IMA

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

#endif

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

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

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

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

union {

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

struct [**block\_device**](https://elixir.bootlin.com/linux/latest/ident/block_device) \*i\_bdev;

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

char \*i\_link;

unsigned i\_dir\_seq;

};

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

#ifdef CONFIG\_FSNOTIFY

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

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

#endif

#if [**IS\_ENABLED**](https://elixir.bootlin.com/linux/latest/ident/IS_ENABLED)(CONFIG\_FS\_ENCRYPTION)

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

#endif

void \*i\_private; */\* fs or device private pointer \*/*

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