NAME

request_key − Request a key from the kernel's key management facility

SYNOPSIS

#include <keyutils.h>

key_serial_t request_key(const char ****type*, const char ****description***,**
const char ****callout_info*, key_serial_t** *keyring***);**

DESCRIPTION

request_key() asks the kernel to find a key of the given *type* that matches the specified *description* and, if successful, to attach it to the nominated *keyring* and to return its serial number.

request_key() first recursively searches all the keyrings attached to the calling process in the order thread-specific keyring, process-specific keyring and then session keyring for a matching key.

If **request_key**() is called from a program invoked by **request_key**() on behalf of some other process to generate a key, then the keyrings of that other process will be searched next, using that other process's UID, GID, groups and security context to control access.

The keys in each keyring searched are checked for a match before any child keyrings are recursed into. Only keys that are **searchable** for the caller may be found, and only **searchable** keyrings may be searched.

If the key is not found then, if *callout_info* is set, this function will attempt to look further afield. In such a case, the *callout_info* is passed to a userspace service such as **/sbin/request−key** to generate the key.

If that is unsuccessful also, then an error will be returned, and a temporary negative key will be installed in the nominated *keyring*. This will expire after a few seconds, but will cause subsequent calls to **request_key**() to fail until it does.

The *keyring* serial number may be that of a valid keyring to which the caller has write permission, or it may be a special keyring ID:

KEY_SPEC_THREAD_KEYRING
This specifies the caller's thread-specific keyring.

KEY_SPEC_PROCESS_KEYRING
This specifies the caller's process-specific keyring.

KEY_SPEC_SESSION_KEYRING
This specifies the caller's session-specific keyring.

KEY_SPEC_USER_KEYRING
This specifies the caller's UID-specific keyring.

KEY_SPEC_USER_SESSION_KEYRING
This specifies the caller's UID-session keyring.

If a key is created, no matter whether it's a valid key or a negative key, it will displace any other key of the same type and description from the destination *keyring*.

RETURN VALUE

On success **request_key**() returns the serial number of the key it found. On error, the value **-1** will be returned and errno will have been set to an appropriate error.

ERRORS

ENOKEY
No matching key was found.

EKEYEXPIRED
An expired key was found, but no replacement could be obtained.

EKEYREVOKED
A revoked key was found, but no replacement could be obtained.

EKEYREJECTED
The attempt to generate a new key was rejected.

ENOMEM
Insufficient memory to create a key.

EINTR
The request was interrupted by a signal.

EDQUOT
The key quota for this user would be exceeded by creating this key or linking it to the keyring.

EACCES
The keyring wasn't available for modification by the user.

LINKING

Although this is a Linux system call, it is not present in *libc* but can be found rather in *libkeyutils*. When linking, **-lkeyutils** should be specified to the linker.

SEE ALSO

keyctl(1), **add_key**(2), **keyctl**(2), **request-key**(8)