

NAME

user-session-keyring – per-user default session keyring

DESCRIPTION

The user session keyring is a keyring used to anchor keys on behalf of a user. Each UID the kernel deals with has its own user session keyring that is shared by all processes with that UID. The user session keyring has a name (description) of the form `_uid_ses.<UID>` where `<UID>` is the user ID of the corresponding user.

The user session keyring is associated with the record that the kernel maintains for the UID. It comes into existence upon the first attempt to access either the user session keyring, the **user-keyring(7)**, or the **session-keyring(7)**. The keyring remains pinned in existence so long as there are processes running with that real UID or files opened by those processes remain open. (The keyring can also be pinned indefinitely by linking it into another keyring.)

The user session keyring is created on demand when a thread requests it or when a thread asks for its **session-keyring(7)** and that keyring doesn't exist. In the latter case, a user session keyring will be created and, if the session keyring wasn't to be created, the user session keyring will be set as the process's actual session keyring.

The user session keyring is searched by **request_key(2)** if the actual session keyring does not exist and is ignored otherwise.

A special serial number value, **KEY_SPEC_USER_SESSION_KEYRING**, is defined that can be used in lieu of the actual serial number of the calling process's user session keyring.

From the **keyctl(1)** utility, '@us' can be used instead of a numeric key ID in much the same way.

User session keyrings are independent of **clone(2)**, **fork(2)**, **vfork(2)**, **execve(2)**, and **_exit(2)** excepting that the keyring is destroyed when the UID record is destroyed when the last process pinning it exits.

If a user session keyring does not exist when it is accessed, it will be created.

Rather than relying on the user session keyring, it is strongly recommended—especially if the process is running as root—that a **session-keyring(7)** be set explicitly, for example by **pam_keyinit(8)**.

NOTES

The user session keyring was added to support situations where a process doesn't have a session keyring, perhaps because it was created via a pathway that didn't involve PAM (e.g., perhaps it was a daemon started by **inetd(8)**). In such a scenario, the user session keyring acts as a substitute for the **session-keyring(7)**.

SEE ALSO

keyctl(1), **keyctl(3)**, **keyrings(7)**, **persistent-keyring(7)**, **process-keyring(7)**, **session-keyring(7)**, **thread-keyring(7)**, **user-keyring(7)**