NAME

session-keyring − session shared process keyring

DESCRIPTION

The session keyring is a keyring used to anchor keys on behalf of a process. It is typically created by **pam_keyinit**(8) when a user logs in and a link will be added that refers to the **user−keyring**(7). Optionally, PAM may revoke the session keyring on logout. (In typical configurations, PAM does do this revocation.) The session keyring has the name (description) *_ses*.

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

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

A process's session keyring is inherited across **clone**(2), **fork**(2), and **vfork**(2). The session keyring is preserved across **execve**(2), even when the executable is set-user-ID or set-group-ID or has capabilities. The session keyring is destroyed when the last process that refers to it exits.

If a process doesn't have a session keyring when it is accessed, then, under certain circumstances, the **user−session−keyring**(7) will be attached as the session keyring and under others a new session keyring will be created. (See **user−session−keyring**(7) for further details.)

Special operations

The *keyutils* library provides the following special operations for manipulating session keyrings:

keyctl_join_session_keyring(3)

This operation allows the caller to change the session keyring that it subscribes to. The caller can join an existing keyring with a specified name (description), create a new keyring with a given name, or ask the kernel to create a new "anonymous" session keyring with the name "_ses". (This function is an interface to the **keyctl**(2) **KEYCTL_JOIN_SESSION_KEYRING** operation.)

keyctl_session_to_parent(3)

This operation allows the caller to make the parent process's session keyring to the same as its own. For this to succeed, the parent process must have identical security attributes and must be single threaded. (This function is an interface to the **keyctl**(2) **KEYCTL_SESSION_TO_PARENT** operation.)

These operations are also exposed through the **keyctl**(1) utility as:

```
keyctl session
keyctl session − [<prog> <arg1> <arg2> ...]
keyctl session <name> [<prog> <arg1> <arg2> ...]
```

and:

```
keyctl new_session
```

SEE ALSO

keyctl(1), **keyctl**(3), **keyctl_join_session_keyring**(3), **keyctl_session_to_parent**(3), **keyrings**(7), **PAM**(7), **persistent−keyring**(7), **process−keyring**(7), **thread−keyring**(7), **user−keyring**(7), **user−session−keyring**(7), **pam_keyinit**(8)