user@.service

- Description
- Controlling resources for logged-in users
- Examples
- See Also

systemd

user@.service

5

user@.service

user-runtime-dir@.service

systemd-user-runtime-dir

System units to start the user manager

`user@UID.service`

`user-runtime-dir@UID.service`

`/usr/lib/systemd/systemd-user-runtime-dir`

`user-UID.slice`

Description

The `systemd(1)` system manager (PID 1) starts user manager instances as `user@UID.service`, with the user's numerical UID used as the instance identifier. These instances use the same executable as the system manager, but running in a mode where it starts a different set of units. Each `systemd --user` instance manages a hierarchy of units specific to that user. See `systemd(1)` for a discussion of units and `systemd.special(7)` for a list of units that form the basis of the unit hierarchies of system and user units.

`user@UID.service` is accompanied by the system unit `user-runtime-dir@UID.service`, which creates the user's runtime directory `/run/user/UID` when started, and removes it when it is stopped. It also might apply runtime quota settings on `/tmp/` and/or `/dev/shm/` for the user. `user-runtime-dir@UID.service` executes the `systemd-user-runtime-dir` binary to do the actual work.

User processes may be started by the `user@.service` instance, in which case they will be part of that unit in the system hierarchy. They may also be started elsewhere, for example by `sshd(8)` or a display manager like `gdm`, in which case they form a .scope unit (see `systemd.scope(5)`). Both `user@UID.service` and the scope units are collected under the `user-UID.slice`.

Individual `user-UID.slice` slices are collected under `user.slice`, see `systemd.special(7)`.

Controlling resources for logged-in users

Options that control resources available to logged-in users can be configured at a few different levels. As described in the previous section, `user.slice` contains processes of all users, so any resource limits on that slice apply to all users together. The usual way to configure them would be through drop-ins, e.g. `/etc/systemd/system/user.slice.d/resources.conf`.

The processes of a single user are collected under `user-UID.slice`. Resource limits for that user can be configured through drop-ins for that unit, e.g. `/etc/systemd/system/user-1000.slice.d/resources.conf`. If the limits should apply to all users instead, they may be configured through drop-ins for the truncated unit name, `user-.slice`. For example, configuration in `/etc/systemd/system/user-.slice.d/resources.conf` is included in all `user-UID.slice` units, see `systemd.unit(5)` for a discussion of the drop-in mechanism.

When a user logs in and a .scope unit is created for the session (see previous section), the creation of the scope may be managed through `pam_systemd(8)`. This PAM module communicates with `systemd-logind(8)` to create the session scope and provide access to hardware resources. Resource limits for the scope may be configured through the PAM module configuration, see `pam_systemd(8)`. Configuring them through the normal unit configuration is also possible, but since the name of the slice unit is generally unpredictable, this is less useful.

In general any resources that apply to units may be set for `user@UID.service` and the slice units discussed above, see `systemd.resource-control(5)` for an overview.

Examples

Hierarchy of control groups with two logged in users

```
$ systemd-cgls
Control group /:
-.slice
├─user.slice
│ ├─user-1000.slice
│ │ ├─user@1000.service
│ │ │ ├─pulseaudio.service
│ │ │ │ └─2386 /usr/bin/pulseaudio --daemonize=no
│ │ │ └─gnome-terminal-server.service
│ │ │   └─init.scope
│ │ │     ├─ 4127 /usr/libexec/gnome-terminal-server
│ │ │     └─ 4198 zsh
│ │ …
│ │ └─session-4.scope
│ │   ├─ 1264 gdm-session-worker [pam/gdm-password]
│ │   ├─ 2339 /usr/bin/gnome-shell
│ │   …
│ │ ├─session-19.scope
│ │   ├─6497 sshd: zbyszek [priv]
│ │   ├─6502 sshd: zbyszek@pts/6
│ │   ├─6509 -zsh
│ │   └─6602 systemd-cgls --no-pager
│ …
│ └─user-1001.slice
│   ├─session-20.scope
│   │ ├─6675 sshd: guest [priv]
│   │ ├─6708 sshd: guest@pts/6
│   │ └─6717 -bash
│   └─user@1001.service
│     ├─init.scope
│     │ ├─6680 /usr/lib/systemd/systemd --user
│     │ └─6688 (sd-pam)
│     └─sleep.service
│       └─6706 /usr/bin/sleep 30
…
```

User with UID 1000 is logged in using `gdm` (`session-4.scope`) and `ssh(1)` (`session-19.scope`), and also has a user manager instance running (`user@1000.service`). User with UID 1001 is logged in

using `ssh` (`session-20.scope`) and also has a user manager instance running (`user@1001.service`). Those are all (leaf) system units, and form part of the slice hierarchy, with `user-1000.slice` and `user-1001.slice` below `user.slice`. User units are visible below the `user@.service` instances (`pulseaudio.service`, `gnome-terminal-server.service`, `init.scope`, `sleep.service`).

Default user resource limits

```
$ systemctl cat user-1000.slice
# /usr/lib/systemd/system/user-.slice.d/10-defaults.conf
# …
[Unit]
Description=User Slice of UID %j
After=systemd-user-sessions.service

[Slice]
TasksMax=33%
```

The `user-UID.slice` units by default do not have a unit file. The resource limits are set through a drop-in, which can be easily replaced or extended following standard drop-in mechanisms discussed in the first section.

See Also

`systemd(1)`, `systemd.service(5)`, `systemd.slice(5)`, `systemd.resource-control(5)`, `systemd.exec(5)`, `systemd.special(7)`, `capsule@.service(5)`, `pam(8)`

□□□ #2
□ □□□ □□□ 2026-10-31 15:08:39 UTC
□ □□□ □□□ 2026-10-31 15:08:40 UTC