SystemD

systemd is a system and service manager for Linux operating systems. When run as first process on boot (as PID 1), it acts as init system that brings up and maintains userspace services. Separate instances are started for logged-in users to start their services. **systemd** is usually not invoked directly by the user, but is installed as the /sbin/init symlink and started during early boot. The user manager instances are started automatically through the user@.service(5) service. For compatibility with SysV, if the binary is called as **init** and is not the first process on the machine (PID is not 1), it will execute **telinit** and pass all command line arguments unmodified. That means **init** and **telinit** are mostly equivalent when invoked from normal login sessions. See telinit(8) for more information.

When run as a system instance, systemd interprets the configuration file system.conf and the files in system.conf.d directories; when run as a user instance, systemd interprets the configuration file user.conf and the files in user.conf.d directories. See systemd-system.conf(5) for more information.

systemd provides a dependency system between various entities called "units" of 11 different types. Units encapsulate various objects that are relevant for system boot-up and maintenance. The

majority of units are configured in unit configuration files, whose syntax and basic set of options is described in systemd.unit(5), however some are created automatically from other configuration files, dynamically from system state or programmatically at runtime. Units may be "active" (meaning started, bound, plugged in, ..., depending on the unit type, see below), or "inactive" (meaning stopped, unbound, unplugged, ...), as well as in the process of being activated or deactivated, i.e. between the two states (these states are called "activating", "deactivating"). A special "failed" state is available as well, which is very similar to "inactive" and is entered when the service failed in some way (process returned error code on exit, or crashed, an operation timed out, or after too many restarts). If this state is entered, the cause will be logged, for later reference. Note that the various unit types may have a number of additional substates, which are mapped to the five generalized unit states described here.