runit

runit is a suite of tools which provides an init (PID 1) as well as daemontools-compatible process supervision framework, along with utilites which streamline creation and maintenance of services.

Contents

Installation
Installation of services
Programs
Files
Basic usage
Runlevel
Service directory structure
Service dependencies
See also

Installation

Install the `runit` package.

Installation of services

runit service packages are named *`package_name`*`-runit` and, when installed, will be available in `/etc/runit/sv`.

Programs

Runit has several programs, but usually you will only interact directly with one program only.

- `sv` - used for controlling services, getting status of services, and dependency checking.
- `chpst` - control of a process environment, including memory caps, limits on cores, data segments, environments, user/group privileges, and more.

- `runsv` - supervises a process, and optionally a log service for that process.
- `svlogd` - a simple but powerful logger, includes auto-rotation based on different methods (time, size, etc), post-processing, pattern matching, and socket (remote logging) options. Say goodbye to logrotate and the need to stop your services to rotate logs.
- `runsvchdir` - changes service levels (runlevels, see below)
- `runsvdir` - starts a supervision tree
- `runit-init` - PID 1, does almost nothing besides being the init

Files

There are several files that will be installed by `runit`.

- `/etc/runit/1` - stage 1, system's one-time initialization tasks
- `/etc/runit/2` - stage 2, Normally runs `runsvdir`, should not return until the system is going to halt or reboot.
- `/etc/runit/3` - stage 3, system's shutdown tasks
- `/etc/runit/ctrlaltdel` - Runit will execute this when receiving a `SIGINT` signal
- `/etc/runit/runsvdir/*` - Runlevels
- `/etc/runit/sv/*` - directory containing subdirectories of available service files
- `/run/runit/service` - always symlinked to active runlevel, `sv` will search for running service here

However, since `runit` itself depends on `runit-rc`, there will be several extra rc files installed, most contained in `/etc/rc` and `/usr/lib/rc`.

Basic usage

Unlike other distros using runit, Artix doesn't store its service directory in `/var/service` or `/service`, but in `/run/runit/service` instead.

- Enable service (in runlevel *default*) `# ln -s /etc/runit/sv/service_name /run/runit/service`
- Disable service `# unlink /run/runit/service/service_name`
- Stop immediately `# sv down service_name` or `# sv stop service_name`
- Start (if not running) `# sv up service_name` or `# sv restart service_name`
- Restart `# sv restart service_name`
- Reload `# sv restart service_name`
- Status check `# sv status service_name`
- Switch runlevels (this will stop all services that are currently running and will start all

services in the new runlevel) `# runsvchdir runlevel`

Runlevel

By default, runit has 2 runlevels, `default` and `single`. You can make your own runlevels just by creating a new folder in `/etc/runit/runsvdir/` and symlinking your desired service to that runlevel.

```
ln -s /etc/runit/sv/service /etc/runit/runsvdir/runlevel@@
```

Service directory structure

This is a tree of a complete service directory structure (aka `/etc/runit/sv/servicedir`), in some run scripts, typically only `run` will be available as usually it's the only file needed.

```
servicedir
├── run (755)
├── check (755)
├── conf (644)
├── finish (755)
└── log (directory)
    ├── config (644)
    └── run (755)
```

A runit (or any daemontools-compatible) run script service directory usually contains only one executable file, `run`, which runs process in the **foreground**. Processes that run in the background cannot be supervised by runit.

If a service directory contains another directory named `log`, the output of the `run` process in the service directory will be piped to the input of the `run` process in the log directory. If the log service uses `svlogd`, it may be configured by using the file `config`. How `svlogd` can be configured is explained in the `svlogd(1)` manpage.

A run script may also contain executables like `finish` and `check`. `finish` will be executed when a service is stopped, and `check` will be executed (if exists) by `sv check` or `sv status`.

A run script may also contain a `conf` file (which is not executable) that modifies the variables available to the script.

Service dependencies

Some services may depend on other services. For example, `NetworkManager` depends on `dbus`. To ensure that required dependencies are satisfied, check the service's `run` file. For example, for NetworkManager:

```
# /etc/runit/sv/NetworkManager/run
sv check dbus >/dev/null || exit 1
```

This means you have to enable `dbus` for NetworkManager to start.

See also

- http://smarden.org/runit - Official runit documentation
- https://docs.voidlinux.org/config/services/index.html - Void Linux handbook on runit

---