

Simple daemon with start-stop-daemon and runit

Arne Babenhauserheide

April 15, 2015

Creating a daemon with almost zero effort.

Contents

start-stop-daemon

The example with the start-stop-daemon uses Gentoo OpenRC as root.

The simplest daemon we can create is a while loop:

```
echo '#!/bin/sh' > whiledaemon.sh
echo 'while true; do true; done' >> whiledaemon.sh
chmod +x whiledaemon.sh
```

Now we start it as daemon

```
start-stop-daemon --pidfile whiledaemon.pid \
--make-pidfile --background ./whiledaemon.sh
```

Top shows that it is running:

```
top | grep whiledaemon.sh
```

We stop it using the pidfile:

```
start-stop-daemon --pidfile whiledaemon.pid \
--stop ./whiledaemon.sh
```

That's it.

If you want to ensure that the daemon keeps running without checking a PID file (which might in some corner cases fail because a new process claims the same PID), we can use `runsvdir` from `runit`.

daemon with runit

Minimal examples for runit daemons - first as unprivileged user, then as root.

runit as simple user

Create a script which dies

```
echo '#!/usr/bin/env python\nfor i in range(100): a = i*i' >/tmp/foo.py  
chmod +x /tmp/foo.py
```

Create the daemon folder

```
mkdir -p ~/.local/run/runit_services/python  
ln -sf /tmp/foo.py ~/.local/run/runit_services/python/run
```

Run the daemon via `runsvdir`

```
runsvdir ~/.local/run/runit_services
```

Manage it with `sv` (part of `runit`)

```
# stop the running daemon  
SVDIR=~/.local/run/runit_services/ sv stop python  
# start the service (it shows as 'run' in top)  
SVDIR=~/.local/run/runit_services/ sv start python
```

runit as root

Minimal working example for setting up runit as root - like a sysadmin might do it.

```
echo '#!/usr/bin/env python\nfor i in range(100): a = i*i' >/tmp/foo.py &&  
chmod +x /tmp/foo.py &&  
mkdir -p /run/arne_service/python &&  
printf '#!/bin/sh\nexec /tmp/foo.py' >/run/arne_service/python/run &&  
chmod +x /run/arne_service/python/run &&  
chown -R arne /run/arne_service &&  
su - arne -c 'runsvdir /run/arne_service'
```

Or without bash indirection (giving up some flexibility we don't need here)

```
echo '#!/usr/bin/env python\nfor i in range(100): a = i*i' >/tmp/foo.py &&  
chmod +x /tmp/foo.py &&  
mkdir -p /run/arne_service/python &&  
ln -s /tmp/foo.py /run/arne_service/python/run &&  
chown -R arne /run/arne_service &&  
su - arne -c 'runsvdir /run/arne_service'
```