

systemd and Linux Watchdog

Run a program at...
login? = .profile file
boot? = systemd

What to do if software locks up?



systemd

- **systemd** used by most Linux distros as first user-space application to be run by the kernel.
 - 'd' means daemon:
 - ...
 - Use systemd to run programs at boot (and many other things).

Jack of All Trades

systemd Utilities

systemctl journalctl notify analyze cglc cgtop loginctl nspawn

systemd Daemons

systemd
journald networkd
logind user session

systemd Targets

bootmode basic multi-user graphical user-session
dbus telephony user-session display service
shutdown reboot dlog logind user-session tizen service

systemd Core

manager unit login namespace log
systemd service timer mount target multiseat inhibit
snapshot path socket swap session pam cgroup dbus

systemd Libraries

dbus-1 libpam libcap libcryptsetup tcpwrapper libaudit libnotify

Linux Kernel

cgroups autofs kdbus

- **Replaces old “init” system:**
 - Manages dependencies and allows concurrency when starting up applications
 - Does many things: login, networking, mounting, etc
- **Controversy**
 - Violates usual *nix philosophy of do one thing well.
<http://www.zdnet.com/article/linus-torvalds-and-others-on-linuxs-systemd/>
 - Some lead developers are said to have a bad attitude towards fixing “their” bugs.
- It's installed on the Beaglebone, so we'll use it!
 - Copy your code to BBG's eMMC (vs run over NFS).

Create a systemd service

- Setup .service file:

```
(bbg)$ cd /lib/systemd/system  
(bbg)$ sudo nano foo.service
```

Assume 11-HttpsProcTimer
example installed to /opt/

Use
absolute
paths

```
[Unit]  
Description=HTTPS server to view /proc on port 8042  
  
[Service]  
User=root  
WorkingDirectory=/opt/10-HttpsProcTimer-copy/  
ExecStart=/usr/bin/node /opt/10-HttpsProcTimer-copy/server.js  
SyslogIdentifier=HttpsProcServer  
  
[Install]  
WantedBy=multi-user.target
```

Controlling a Service

- **Configure to run at startup**
(bbg)\$ `systemctl enable foo.service`
- **Manually Starting/Stopping**
(bbg)\$ `systemctl start foo.service`
 - Can replace start with stop or restart
- **Status**
(bbg)\$ `systemctl status foo.service`
(bbg)\$ `journalctl -u foo.service`
(bbg)\$ `systemctl | grep HTTPS`

Demo: Browse to <https://192.168.7.2:3042> after reboot

Startup Script Suggestions

- If your app needs some startup steps, try a script:
 - copy app to file system (not running via NFS)
 - add 10s delay at startup
 - I have found that some hardware configuration commands can fail if done too soon.
 - add 1s delay between commands
 - Allows the system time for the previous setup action to complete.
 - change to the required directory

Linux Watchdog

Watchdog

- **Watchdog Timer**

Guards against system locking-up:

- Working software periodically "hits" (*pets!*) the WD.
- ..

- **Applications**

- Most embedded systems use a watchdog: allow it to recover from faults (SW and some HW).
- Critical in applications where humans cannot reboot.

Watchdog Usage

- Usage
 - Opening "file" `/dev/watchdog`; starts timer
 - Writing anything to it resets timeout
If timeout expires it resets board.
 - What should happen when the program exits?..
- Console demo
 - `(bbg)$ cat > /dev/watchdog`
- Can compile kernel to disallow turning off watchdog:
 - `CONFIG_WATCHDOG_NOWAYOUT`