Embedded Linux systemd

Norman McEntire

Introduction

- **systemd** is the “init” system in modern Linux distros
  - Starts as Process ID 1 (the init process)
- NOTE: Some older Embedded Linux System Still use “init” and a SystemV or custom startup system
  - If “systemd” is not in the processes then not using systemd
    - Or if “pstree” does not show systemd
- **systemctl** is key command for controlling systemd

Advantages of systemd (compared to original SysVInit)

- Units: Resources are organized into Units
- Parallel Startup: Services can startup in parallel, resulting in faster boot up
- Dependency Handling
- On-Demand Starting of Services
- Resource Control: Integrates with the Linux CGroups (Control Groups) to limit CPU and memory usage
- Service Monitoring and Restarting
- Integrated Logging with journald
- Configuration and Admin Tools: systemctl, journalctl
- Predictable Device Names
- User Session Management

/sbin/init

```
$ ls -l /sbin/init
lrwxrwxrwx 1 root root 20 May 13 03:36 /sbin/init -> /lib/systemd/systemd
```

```
$ ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.4  33928  8928 ?        Ss   08:33   0:03 /sbin/init splash
root           2  0.0  0.0      0     0 ?        S    08:33   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        I<   08:33   0:00 [rcu_gp]
. . .
_rpc         347  0.0  0.1   7604  3008 ?        Ss   08:33   0:00 /sbin/rpcbind -f -w
systemd+     363  0.0  0.1  22292  2576 ?        Ssl  08:33   0:00 /lib/systemd/systemd-timesyncd
avahi        421  0.0  0.1   6916  2636 ?        Ss   08:33   0:02 avahi-daemon: running [raspberrypi.local]
. . .
metaemb+    3265  0.0  0.2   8732  3952 pts/0    Ss   10:17   0:00 -bash
. . .
metaemb+    3591  0.0  0.1  10860  2516 pts/0    R+   10:31   0:00 ps aux
```

pstree -p

systemd is PID 1
(The first process in the system)

```
$ pstree -p
systemd(1)─┬─ModemManager(533)─┬─{ModemManager}(574)
           │                   └─{ModemManager}(576)
           ├─applet.py(902)
           ├─avahi-daemon(421)───avahi-daemon(439)
           ├─blkmapd(159)
           ├─bluetoothd(931)
           ├─colord(595)─┬─{colord}(611)
           │             └─{colord}(621)
           ├─cron(699)
           ├─cups-browsed(698)─┬─{cups-browsed}(726)
           │                   └─{cups-browsed}(727)
           ├─cupsd(528)─┬─dbus(684)
           │            └─dbus(1215)
           ├─dbus-daemon(433)
. . .      │
           ├─ssh-agent(897)
           ├─sshd(583)───sshd(3249)───sshd(3264)───bash(3265)───pstree(3845)
. . .      │
           ├─systemd-journal(149)
           ├─systemd-logind(467)
           ├─systemd-network(202)
           ├─systemd-timesyn(363)───{systemd-timesyn}(602)
           ├─systemd-udevd(186)
           ├─thd(470)
           ├─udisksd(475)─┬─{udisksd}(539)
           │              ├─{udisksd}(546)
           │              ├─{udisksd}(587)
           │              └─{udisksd}(654)
           ├─wpa_supplicant(476)
           └─wpa_supplicant(628)
```

man systemd

SYSTEMD(1) systemd SYSTEMD(1)

NAME
systemd, init - systemd system and service manager

SYNOPSIS
/lib/systemd/systemd [OPTIONS...]

init [OPTIONS...] {COMMAND}

DESCRIPTION
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.

man systemctl

```
SYSTEMCTL(1)                                systemctl                                SYSTEMCTL(1)

NAME
       systemctl - Control the systemd system and service manager

SYNOPSIS
       systemctl [OPTIONS...] COMMAND [UNIT...]

DESCRIPTION
       systemctl may be used to introspect and control the state of the "systemd" system and service manager.
       Please refer to systemd(1) for an introduction into the basic concepts and functionality this tool
       manages.
```

/etc/systemd

```
$ ls /etc/systemd/
journald.conf  network        pstore.conf    sleep.conf  system.conf     user
logind.conf    networkd.conf  resolved.conf  system      timesyncd.conf  user.conf
```

/etc/systemd/system.conf

NOTE: When systemd runs as system instance
(Compared to user instance)
this system.conf config file is read
And also the
/etc/systemd/system directory

```
$ cat /etc/systemd/system.conf
#  This file is part of systemd.
#
#  systemd is free software; you can redistribute it and/or modify it
#  under the terms of the GNU Lesser General Public License as published by
#  the Free Software Foundation; either version 2.1 of the License, or
#  (at your option) any later version.
#
# Entries in this file show the compile time defaults.
# You can change settings by editing this file.
# Defaults can be restored by simply deleting this file.
#
# See systemd-system.conf(5) for details.

[Manager]
#LogLevel=info
#LogTarget=journal-or-kmsg
#LogColor=yes
#LogLocation=no
#LogTime=no
#DumpCore=yes
. . .
```

/etc/systemd/system/

```
ls /etc/systemd/system
bluetooth.target.wants                          dhcpcd.service.d              printer.target.wants
dbus-fi.w1.wpa_supplicant1.service              display-manager.service       rc-local.service.d
dbus-org.bluez.service                          getty.target.wants            reboot.target.wants
dbus-org.freedesktop.Avahi.service              getty@tty1.service.d          remote-fs.target.wants
dbus-org.freedesktop.ModemManager1.service      graphical.target.wants        sockets.target.wants
dbus-org.freedesktop.network1.service           halt.target.wants             sshd.service
dbus-org.freedesktop.timesync1.service          multi-user.target.wants       sysinit.target.wants
default.target                                  network-online.target.wants   syslog.service
default.target.wants                            pigpiod.service.d             timers.target.wants
dev-serial1.device.wants                        poweroff.target.wants
```

systemd Concepts

- Starts as process ID 1 (/sbin/init is symlink to systemd)
- Provides a dependency system between “units”
  - 11 different Units
    - (See next slide)
  - Units may be
    - Active
    - Inactive
    - Failed

systemd Units

- Service Units
- Socket Units
- Target Units
- Device Units
- Mount Units
- Automount Units
- Timer Units
- Swap Units
- Path Units
- Slice Units
- Scope Units

systemd Units - Overview - Part 1

man systemd.unit

- A unit file is…
  - A plain text file
  - In ini-style
  - Encodes info about a unit
    - .service, .socket, .device, .mount, .automount, .swap, .target, .path, .timer , .slice, .scope
- Many common configuration options

Ini Sections

man systemd.unit

- [Unit]
  - Generic info such as
    - Description=“Human readable name goes here”
    - Documentation=
    - Wants=
- [Install]
  - Installation info used by “enable” and “disable” commands

systemd Service Units (.service)

- A unit configuration file that ends .service contains info about a process controlled and supervised by systemd
- man 5 systemd.service
- NOTE! If a service is requested under a certain name but no unit config file found, systemd looks for SysV init script
- [Service] section in config file
  - Type=
  - ExecStart=
  - Environment=
  - Etc.

systemd Target Units (.target)

- Used to group other units
  - Provide well known sync points during boot
  - man 5 systemd.target

systemd Device Units (.device)

- Expose Kernel Devices
  - Implement device-based activation
  - Work together with udev (dynamic device management)
    - The udev daemon receives device events from kernel as device added/removed
  - man 5 systemd.device

```
$ ps aux | grep udev
root       184  0.0  0.1  20484  2608 ?        Ss   Aug26   0:06 /lib/systemd/systemd-udevd
```

systemd Mount Units (.mount)

- Control filesystem mount points
  - man 5 systemd.mount

systemd Automount Units (.automount)

- Provide auto mount
  - man 5 systemd.automount

systemd Timer Units (.timer)

- Provide for triggering activation of other units based on timers
- For each timer file, a matching unit file must exist describing the unit to activate
  - man 5 systemd.timer

systemd Swap Units (.swap)

- Similar to mount units but focused on swap partitions
  - man 5 systemd.swap

systemd Path Units (.path)

- Path units are used to activate other services when file system objects change
- For each path file, a matching unit file must exist, describing the unit to activate when path changes
- Internally, the inotify(7) API used to monitor the filesystem
  - man 5 systemd.path

systemd Slice Units (.slice)

- Slice units group units which manage system processes in a hierarchical tree
  - The management is performed by creating a node in Linux Control Group (cgroup)
  - man 5 systemd.slice

systemd Scope Units (.scope)

- Scope units group are similar to service units but manage foreign processes instead of starting them as well
  - The main use of scope units is grouping worker processes for a system service
  - man 5 systemd.scope

systemctl (default is list-units)

```
  UNIT                                                                                LOAD   ACTIV>
  proc-sys-fs-binfmt_misc.automount                                                   loaded activ>
  sys-devices-platform-emmc2bus-fe340000.mmc-mmc_host-mmc0-mmc0:5048-block-mmcblk0-mmcblk0p1.device loaded activ>
  sys-devices-platform-emmc2bus-fe340000.mmc-mmc_host-mmc0-mmc0:5048-block-mmcblk0-mmcblk0p2.device loaded activ>
  sys-devices-platform-emmc2bus-fe340000.mmc-mmc_host-mmc0-mmc0:5048-block-mmcblk0.device           loaded activ>
  sys-devices-platform-scb-fd580000.ethernet-net-eth0.device                          loaded activ>
  sys-devices-platform-soc-fe00b840.mailbox-bcm2835_audio-sound-card0-controlC0.device loaded activ>
  sys-devices-platform-soc-fe201000.serial-tty-ttyAMA0-hci0.device                    loaded activ>
  sys-devices-platform-soc-fe201000.serial-tty-ttyAMA0.device                         loaded activ>
  -.mount                                                                             loaded activ>
  boot.mount                                                                          loaded activ>
  dev-mqueue.mount                                                                    loaded activ>
  proc-fs-nfsd.mount                                                                  loaded activ>
  run-rpc_pipefs.mount                                                                loaded activ>
  cups.path                                                                           loaded activ>
  systemd-ask-password-plymouth.path                                                  loaded activ>
  systemd-ask-password-wall.path                                                      loaded activ>
  init.scope                                                                          loaded activ>
  session-1.scope                                                                     loaded activ>
  session-3.scope                                                                     loaded activ>
  session-6.scope                                                                     loaded activ>
  alsa-restore.service                                                                loaded activ>
  avahi-daemon.service                                                                loaded activ>
```

systemctl status

```
CGroup: /
        ├─user.slice
        │ └─user-1000.slice
        │   ├─session-3.scope
        │   │ ├─1225 /bin/login -f
        │   │ └─1299 -bash
        │   ├─session-6.scope
        │   │ ├─3249 sshd: metaembedded [priv]
        │   │ ├─3264 sshd: metaembedded@pts/0
        │   │ ├─3265 -bash


        ├─init.scope
        │ └─1 /sbin/init splash
        └─system.slice
          ├─lightdm.service
          │ ├─708 /usr/sbin/lightdm
          │ └─735 /usr/lib/xorg/Xorg :0 -seat seat0 -auth /var/run/lightdm/root/
          ├─nfs-idmapd.service
          │ └─329 /usr/sbin/rpc.idmapd
          ├─nfs-mountd.service
          │ └─559 /usr/sbin/rpc.mountd --manage-gids
```

systemctl status bluetooth

```
metaembedded@raspberrypi:~ $ systemctl status bluetooth
● bluetooth.service - Bluetooth service
     Loaded: loaded (/lib/systemd/system/bluetooth.service; enabled; vendor preset: enabled)
     Active: active (running) since Sat 2023-08-19 03:17:13 PDT; 8h ago
       Docs: man:bluetoothd(8)
   Main PID: 931 (bluetoothd)
     Status: "Running"
      Tasks: 1 (limit: 3258)
        CPU: 69ms
     CGroup: /system.slice/bluetooth.service
             └─931 /usr/libexec/bluetooth/bluetoothd

Aug 19 03:17:13 raspberrypi systemd[1]: Starting Bluetooth service...
Aug 19 03:17:13 raspberrypi bluetoothd[931]: Bluetooth daemon 5.55
Aug 19 03:17:13 raspberrypi systemd[1]: Started Bluetooth service.
```

systemctl show (show properties of Units)

```
Version=247.3-7+rpi1+deb11u2
Features=+PAM +AUDIT +SELINUX +IMA +APPARMOR +SMACK +SYSVINIT +UTMP +LIBCRYPTSETUP
Architecture=arm
FirmwareTimestampMonotonic=0
LoaderTimestampMonotonic=0
KernelTimestamp=Wed 1969-12-31 16:00:00 PST
KernelTimestampMonotonic=0
InitRDTimestampMonotonic=0
UserspaceTimestamp=Wed 1969-12-31 16:00:02 PST
UserspaceTimestampMonotonic=2651711
FinishTimestamp=Sat 2023-08-19 03:17:52 PDT
FinishTimestampMonotonic=55496974
```

systemctl show bluetooth

```
Type=dbus
Restart=no
NotifyAccess=main
RestartUSec=100ms
TimeoutStartUSec=1min 30s
TimeoutStopUSec=1min 30s
TimeoutAbortUSec=1min 30s
TimeoutStartFailureMode=terminate
TimeoutStopFailureMode=terminate
RuntimeMaxUSec=infinity
WatchdogUSec=0
WatchdogTimestampMonotonic=0
RootDirectoryStartOnly=no
RemainAfterExit=no
GuessMainPID=yes
MainPID=931
ControlPID=0
BusName=org.bluez
FileDescriptorStoreMax=0
NFileDescriptorStore=0
StatusText=Running
StatusErrno=0
Result=success
ReloadResult=success
CleanResult=success
UID=[not set]
```

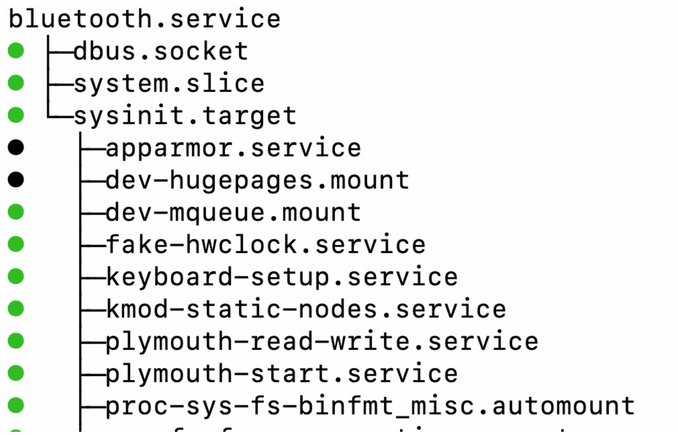
systemctl cat bluetooth (show backing files)

```
# /lib/systemd/system/bluetooth.service
[Unit]
Description=Bluetooth service
Documentation=man:bluetoothd(8)
ConditionPathIsDirectory=/sys/class/bluetooth

[Service]
Type=dbus
BusName=org.bluez
ExecStart=/usr/libexec/bluetooth/bluetoothd
NotifyAccess=main
#WatchdogSec=10
#Restart=on-failure
CapabilityBoundingSet=CAP_NET_ADMIN CAP_NET_BIND_SERVICE
LimitNPROC=1
ProtectHome=true
ProtectSystem=full

[Install]
WantedBy=bluetooth.target
Alias=dbus-org.bluez.service
```

systemctl list-dependencies bluetooth



systemctl start/stop/reload bluetooth

```
metaembedded@raspberrypi:~ $ sudo systemctl stop bluetooth
metaembedded@raspberrypi:~ $ sudo systemctl start bluetooth
metaembedded@raspberrypi:~ $ sudo systemctl restart bluetooth
```

systemctl set-property UNIT property=value

Example: **systemctl set-property foobar.service CPUWeight=200**

systemctl enable/disable bluetooth

```
$ sudo systemctl enable bluetooth
Synchronizing state of bluetooth.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable bluetooth
```

```
$ sudo systemctl disable bluetooth
Synchronizing state of bluetooth.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install disable bluetooth
Removed /etc/systemd/system/bluetooth.target.wants/bluetooth.service.
Removed /etc/systemd/system/dbus-org.bluez.service.
```

systemctl list-timers

```
metaembedded@raspberrypi:~ $ systemctl list-timers
NEXT                         LEFT          LAST                         PASSED       UNIT                         >
Sat 2023-08-19 17:01:04 PDT 4h 55min left  Fri 2023-08-18 20:13:11 PDT 15h ago      apt-daily.timer              >
Sun 2023-08-20 00:00:00 PDT 11h left       Sat 2023-08-19 00:00:18 PDT 12h ago      logrotate.timer              >
Sun 2023-08-20 00:00:00 PDT 11h left       Sat 2023-08-19 00:00:18 PDT 12h ago      man-db.timer                 >
Sun 2023-08-20 03:10:18 PDT 15h left       Sun 2023-08-13 03:10:28 PDT 6 days ago   e2scrub_all.timer            >
Sun 2023-08-20 06:50:11 PDT 18h left       Sat 2023-08-19 09:29:25 PDT 2h 35min ago apt-daily-upgrade.timer      >
Sun 2023-08-20 08:48:38 PDT 20h left       Sat 2023-08-19 08:48:38 PDT 3h 16min ago systemd-tmpfiles-clean.timer>
Mon 2023-08-21 00:39:37 PDT 1 day 12h left Mon 2023-08-14 00:13:28 PDT 5 days ago   fstrim.timer                 >

7 timers listed.
Pass --all to see loaded but inactive timers, too.
```

systemctl list-sockets

```
metaembedded@raspberrypi:~ $ systemctl list-sockets
LISTEN                          UNIT                                ACTIVATES
/dev/rfkill                     systemd-rfkill.socket               systemd-rfkill.service
/run/avahi-daemon/socket        avahi-daemon.socket                 avahi-daemon.service
/run/cups/cups.sock             cups.socket                         cups.service
/run/dbus/system_bus_socket     dbus.socket                         dbus.service
/run/initctl                    systemd-initctl.socket              systemd-initctl.service
/run/rpcbind.sock               rpcbind.socket                      rpcbind.service
/run/systemd/fsck.progress      systemd-fsckd.socket                systemd-fsckd.service
/run/systemd/journal/dev-log    systemd-journald-dev-log.socket     systemd-journald.service
/run/systemd/journal/socket     systemd-journald.socket             systemd-journald.service
/run/systemd/journal/stdout     systemd-journald.socket             systemd-journald.service
/run/systemd/journal/syslog     syslog.socket                       rsyslog.service
/run/thd.socket                 triggerhappy.socket                 triggerhappy.service
/run/udev/control               systemd-udevd-control.socket        systemd-udevd.service
/run/uuidd/request              uuidd.socket                        uuidd.service
0.0.0.0:111                     rpcbind.socket                      rpcbind.service
0.0.0.0:111                     rpcbind.socket                      rpcbind.service
[::]:111                        rpcbind.socket                      rpcbind.service
[::]:111                        rpcbind.socket                      rpcbind.service
audit 1                         systemd-journald-audit.socket       systemd-journald.service
kobject-uevent 1                systemd-udevd-kernel.socket         systemd-udevd.service
route 1361                      systemd-networkd.socket             systemd-networkd.service

21 sockets listed.
Pass --all to see loaded but inactive sockets, too.
```

man journalctl

```
JOURNALCTL(1)                                journalctl                                JOURNALCTL(1)

NAME
       journalctl - Query the systemd journal

SYNOPSIS
       journalctl [OPTIONS...] [MATCHES...]

DESCRIPTION
       journalctl may be used to query the contents of the systemd(1) journal as written by systemd-
       journald.service(8).

       If called without parameters, it will show the full contents of the journal, starting with the oldest
       entry collected.
```

journalctl

```
-- Journal begins at Mon 2023-02-20 17:34:05 PST, ends at Sat 2023-08-19 17:33:47 PDT. --
Feb 20 17:34:05 raspberrypi kernel: Booting Linux on physical CPU 0x0
Feb 20 17:34:05 raspberrypi kernel: Linux version 5.15.84-v7l+ (dom@buildbot) (arm-linux-gnueabihf-gcc-8 (Ubuntu
Feb 20 17:34:05 raspberrypi kernel: CPU: ARMv7 Processor [410fd083] revision 3 (ARMv7), cr=30c5383d
Feb 20 17:34:05 raspberrypi kernel: CPU: div instructions available: patching division code
Feb 20 17:34:05 raspberrypi kernel: CPU: PIPT / VIPT nonaliasing data cache, PIPT instruction cache
Feb 20 17:34:05 raspberrypi kernel: OF: fdt: Machine model: Raspberry Pi 4 Model B Rev 1.4
Feb 20 17:34:05 raspberrypi kernel: random: crng init done
Feb 20 17:34:05 raspberrypi kernel: Memory policy: Data cache writealloc
Feb 20 17:34:05 raspberrypi kernel: Reserved memory: created CMA memory pool at 0x000000001ac00000, size 320 MiB
Feb 20 17:34:05 raspberrypi kernel: OF: reserved mem: initialized node linux,cma, compatible id shared-dma-pool
Feb 20 17:34:05 raspberrypi kernel: Zone ranges:
Feb 20 17:34:05 raspberrypi kernel:   DMA      [mem 0x0000000000000000-0x000000002fffffff]
Feb 20 17:34:05 raspberrypi kernel:   Normal   empty
Feb 20 17:34:05 raspberrypi kernel:   HighMem  [mem 0x0000000030000000-0x000000007fffffff]
```

journalctl --lines 20

(Show most recent 20 lines)

```
metaembedded@raspberrypi:~ $ journalctl --lines 20
-- Journal begins at Mon 2023-02-20 17:34:05 PST, ends at Sat 2023-08-19 17:33:47 PDT. --
Aug 19 17:17:01 raspberrypi CRON[1618]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
Aug 19 17:17:01 raspberrypi CRON[1619]: (root) CMD (   cd / && run-parts --report /etc/cron.hourly)
Aug 19 17:17:01 raspberrypi CRON[1618]: pam_unix(cron:session): session closed for user root
Aug 19 17:23:10 raspberrypi systemd[1]: Starting Cleanup of Temporary Directories...
Aug 19 17:23:10 raspberrypi systemd[1]: systemd-tmpfiles-clean.service: Succeeded.
Aug 19 17:23:10 raspberrypi systemd[1]: Finished Cleanup of Temporary Directories.
Aug 19 17:25:01 raspberrypi dhcpcd[438]: wlan0: Router Advertisement from fe80::7ad6:d6ff:fe00:c4cd
Aug 19 17:25:02 raspberrypi dnsmasq[612]: reading /run/dnsmasq/resolv.conf
Aug 19 17:25:02 raspberrypi dnsmasq[612]: using nameserver 192.168.4.1#53
Aug 19 17:28:12 raspberrypi dhcpcd[438]: wlan0: Router Advertisement from fe80::7ad6:d6ff:fe00:c4cd
Aug 19 17:28:12 raspberrypi dnsmasq[612]: reading /run/dnsmasq/resolv.conf
Aug 19 17:28:12 raspberrypi dnsmasq[612]: using nameserver 192.168.4.1#53
Aug 19 17:28:12 raspberrypi dnsmasq[612]: using nameserver 2600:1700:6cf8:1120::1#53
Aug 19 17:33:27 raspberrypi dhcpcd[438]: wlan0: Router Advertisement from fe80::7ad6:d6ff:fe00:c4cd
Aug 19 17:33:28 raspberrypi dnsmasq[612]: reading /run/dnsmasq/resolv.conf
Aug 19 17:33:28 raspberrypi dnsmasq[612]: using nameserver 192.168.4.1#53
Aug 19 17:33:46 raspberrypi dhcpcd[438]: wlan0: Router Advertisement from fe80::7ad6:d6ff:fe00:c4cd
Aug 19 17:33:47 raspberrypi dnsmasq[612]: reading /run/dnsmasq/resolv.conf
Aug 19 17:33:47 raspberrypi dnsmasq[612]: using nameserver 192.168.4.1#53
Aug 19 17:33:47 raspberrypi dnsmasq[612]: using nameserver 2600:1700:6cf8:1120::1#53
```

journalctl --follow

```
metaembedded@raspberrypi:~ $ journalctl --follow
-- Journal begins at Mon 2023-02-20 17:34:05 PST. --
Aug 19 12:27:33 raspberrypi sshd[3249]: pam_unix(sshd:session): session closed for user metaembedded
Aug 19 12:27:33 raspberrypi systemd[1]: session-6.scope: Succeeded.
-- Boot 3182afde63b34166984183a2bda290eb --
Aug 19 17:10:54 raspberrypi systemd-timesyncd[333]: Initial synchronization to time server [2600:3c01::f03c:93f
fe5b:8a7d]:123 (2.debian.pool.ntp.org).
Aug 19 17:15:39 raspberrypi dhcpcd[438]: wlan0: Router Advertisement from fe80::7ad6:d6ff:fe00:c4cd
Aug 19 17:15:39 raspberrypi dnsmasq[612]: reading /run/dnsmasq/resolv.conf
Aug 19 17:15:39 raspberrypi dnsmasq[612]: using nameserver 192.168.4.1#53
Aug 19 17:15:39 raspberrypi dnsmasq[612]: using nameserver 2600:1700:6cf8:1120::1#53
Aug 19 17:17:01 raspberrypi CRON[1618]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
Aug 19 17:17:01 raspberrypi CRON[1619]: (root) CMD (   cd / && run-parts --report /etc/cron.hourly)
Aug 19 17:17:01 raspberrypi CRON[1618]: pam_unix(cron:session): session closed for user root
```

Creating a “Hello World” Service - Part 1

```
$ sudo vi /usr/local/bin/hello-bash-service.sh

$ cat /usr/local/bin/hello-bash-service.sh
#!/bin/bash

COUNT=0
while true ; do
   COUNT=$((COUNT+1))
   echo "COUNT: $COUNT" > /tmp/hello-bash-service.txt
   sleep 10
done

exit 0

$ sudo chmod +x /usr/local/bin/hello-bash-service.sh


$ hello-bash-service.sh

$ cat /tmp/hello-bash-service.txt
COUNT: 6
```

Creating a “Hello World” Service - Part 2

```
$ ls /etc/systemd/system
bluetooth.target.wants                      dhcpcd.service.d              printer.target.wants
dbus-fi.w1.wpa_supplicant1.service          display-manager.service       rc-local.service.d
dbus-org.bluez.service                      getty.target.wants            reboot.target.wants
dbus-org.freedesktop.Avahi.service          getty@tty1.service.d          remote-fs.target.wants
dbus-org.freedesktop.ModemManager1.service  graphical.target.wants       sockets.target.wants
dbus-org.freedesktop.network1.service       halt.target.wants             sshd.service
dbus-org.freedesktop.timesync1.service      multi-user.target.wants       sysinit.target.wants
default.target                              network-online.target.wants   syslog.service
default.target.wants                        pigpiod.service.d             timers.target.wants
dev-serial1.device.wants                    poweroff.target.wants
```

```
$ sudo vim /etc/systemd/system/hello-bash-service.service

$ cat /etc/systemd/system/hello-bash-service.service
[Unit]
Description=Hello World Service

[Service]
Type=simple
ExecStart=/usr/local/bin/hello-bash-service.sh

[Install]
WantedBy=multi-user.target
```

Creating a “Hello World” Service - Part 3

```
$ sudo systemctl daemon-reload

$ sudo systemctl start hello-bash-service

$ $ systemctl status hello-bash-service
● hello-bash-service.service - Hello World Service
     Loaded: loaded (/etc/systemd/system/hello-bash-service.service; disabled; vendor preset: enabled)
     Active: active (running) since Sat 2023-08-19 18:54:19 PDT; 2h 27min ago
   Main PID: 3727 (hello-bash-serv)
      Tasks: 2 (limit: 3258)
        CPU: 6.064s
     CGroup: /system.slice/hello-bash-service.service
             ├─3727 /bin/bash /usr/local/bin/hello-bash-service.sh
             └─7315 sleep 10

Aug 19 18:54:19 raspberrypi systemd[1]: Started Hello World Service.
```

Creating a “Hello World” Service - Part 4

```
$ pstree
systemd─┬─ModemManager───2*[{ModemManager}]
        ├─applet.py
        ├─avahi-daemon───avahi-daemon
        ├─blkmapd
        ├─bluetoothd
. . .
        ├─hello-bash-serv───sleep
. . .

$ cat /tmp/hello-bash-service.txt
COUNT: 6797


$ sudo systemctl stop hello-bash-service


$ sudo systemctl status hello-bash-service
● hello-bash-service.service - Hello World Service
     Loaded: loaded (/etc/systemd/system/hello-bash-service.service; disabled; vendor preset: enabled)
     Active: inactive (dead)
. . .
. . .
Aug 20 13:48:40 raspberrypi systemd[1]: Stopping Hello World Service...
Aug 20 13:48:40 raspberrypi systemd[1]: hello-bash-service.service: Succeeded.
Aug 20 13:48:40 raspberrypi systemd[1]: Stopped Hello World Service.
Aug 20 13:48:40 raspberrypi systemd[1]: hello-bash-service.service: Consumed 46.558s CPU time.
```

Summary

- systemd is process ID 1 in modern embedded Linux systems
- /sbin/init is symbolic link
- /etc/systemd directory
  - /etc/systemd/system
- systemctl - control systemd and the service manager
  - systemctl COMMAND UNIT
- journalctl - query the systemd journal (log info)