

Systemd について

systemd は Linux における init デーモンの代替であり、CentOS7 から導入された。

使い方

<http://blog.yuryu.jp/2014/07/systemd-quick-guide.html>

操作	SysV Init	Systemd
起動	/etc/init.d/sshhd start	systemctl start sshd
終了	/etc/init.d/sshhd stop	systemctl stop sshd
状態確認	/etc/init.d/sshhd status	systemctl status sshd
強制終了	PID 探して kill -9	systemctl kill -s 9 sshd
再起動	/etc/init.d/sshhd restart	systemctl restart sshd
設定反映	/etc/init.d/sshhd reload	systemctl reload sshd
状態取得	/etc/init.d/sshhd status	systemctl status sshd
自動起動を有効	chkconfig sshd on	systemctl enable sshd
自動起動を無効	chkconfig sshd off	systemctl disable sshd
自動起動の状態確認	chkconfig --list sshd	systemctl is-enabled sshd(status でも表示される)
自動起動の状態確認 2	chkconfig --list	systemctl list-unit-files
サービス一覧の表示	ls /etc/init.d	systemctl --type service

ページャ

systemctl の出力結果がページャに渡されるを防ぐには

```
systemctl --no-pager
```

のように、--no-pager オプションを使う

ランレベルについて

Systemd ではランレベルに似たものとしてターゲットを使っています。

基本的に各サービスのランレベルを変える必要はないはず・・・。

SysV ランレベル	systemd ターゲット	説明
0	runlevel0.target, poweroff.target	システムを停止。
1, s, single	runlevel1.target, rescue.target	シングルユーザーモード。
2, 4	runlevel2.target, runlevel4.target, multi-user.target	ユーザー定義・サイト指定ランレベル。デフォルトでは、3 と同一。

3	runlevel3.target, multi-user.target	マルチユーザー、非グラフィカル。一般的にマルチコンソールやネットワークを介してログインするのに使われます。
5	runlevel5.target, graphical.target	マルチユーザー、グラフィカル。通常、ランレベル3の全てのサービスにグラフィカルログインを付加。
6	runlevel6.target, reboot.target	再起動
emergency	emergency.target	緊急シェル

■ 各サービスのランレベルを変更する

各ターゲット内にサービスへのシンボリックリンクを作成したり消したりする・・・でいいのかな？

ただ、各ターゲット間で依存関係があったりするので注意。

■ OS の起動時のターゲット（ランレベル）を変更する

```
systemctl set-default multi-user.target
```

依存関係の一覧

<http://d.hatena.ne.jp/enakai00/20130914/1379146157>

```
systemctl list-dependencies
```

unit の一覧

```
systemctl list-unit-files
```

稼働している service の一覧

```
systemctl list-units
```

■ 設定ファイルのディレクトリ

<http://d.hatena.ne.jp/enakai00/20130914/1379146157>

ディレクトリ	説明
/usr/lib/systemd/system	システムデフォルトの設定。RPM パッケージが提供するデフォルト設定を配置する。
/etc/systemd/system	ユーザ設定。ここに同名のファイルを配置するとこちらのファイルの内容が優先される。

Unit の種類

Unit のタイプ	説明
-----------	----

mount	指定のファイルシステムをマウントする
automount	オートマウント処理を実施する（automountd の代替的な機能）
service	指定のバイナリを実行する（主にはデーモンの起動に使用する）
socket	systemd が Socket を Listen して、接続があるとプロセスに受け渡す（xinetd の代替的な機能）
path	指定のファイルが作成されると、指定されたサービスを起動する
device	udev から通知されたデバイスを表す
target	複数の Unit をまとめるために使用する

任意のシェルスクリプトを service として実行する

https://wiki.archlinux.org/index.php/Systemd_FAQ_%28%E6%9C%AC%E8%AA%9E%29#.E3.83.96.E3.83.BC.E3.83.88.E3.83.97.E3.83.AD.E3.82.BB.E3.82.B9.E3.81.AE.E9.96.93.E3.81.AB.E3.82.B9.E3.82.AF.E3.83.AA.E3.83.97.E3.83.88.E3.82.92.E8.B5.B7.E5.8B.95.E3.81.99.E3.82.8B.E3.81.AB.E3.81.AF.E3.81.A9.E3.81.86.E3.81.99.E3.82.8C.E3.81.B0.E3.81.84.E3.81.84.E3.81.A7.E3.81.99.E3.81.8B.EF.BC.9F

<http://qiita.com/masami256/items/ef0f23125cf8255e4857>

サンプル 1

シェルを実行するだけのサンプル。

デーモンとしてではなく、一回処理するだけならサンプル 2 のように

```
RemainAfterExit=yes
```

を設定したほうがいいかもしれない。

■ 任意のシェルスクリプトを準備する

```
vi /etc/systemd/system/myscript.service
```

```
[Unit]
Description=My script

[Service]
ExecStart=/usr/bin/my-script

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

■ 実行権限を付加

他の service ファイルと同じように

```
chmod 755 myscript.service
```

■ サービスを有効にする

```
systemctl enable myscript.service
```

サンプル2

```
[Unit]
Description=Load dump capture kernel
After=local-fs.target

[Service]
Type=oneshot
ExecStart=/opt/bin/run-kexec.sh
RemainAfterExit=yes

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

サンプル1との大きな違いは、

```
Type=oneshot
RemainAfterExit=yes
```

の2行。

デーモンではなく、一度だけのシェル実行等の場合は

```
Type=oneshot
```

とする。また、

```
RemainAfterExit=yes
```

を設定することで systemd はプロセスが終了した後もサービスがアクティブだとみなします。

サンプル3 (dice)

<http://blogs.yahoo.co.jp/oguri0821/63640997.html>

<https://ospry.wordpress.com/2013/12/09/fedora20-ddns-with-ieserver-dice/>

```
vi /usr/lib/systemd/system/dice.service
```

```
[Unit]
Description=The ddojp
After=network.target remote-fs.target nss-lookup.target

[Service]
Type=notify
EnvironmentFile=/usr/local/src/DiCE/diced
ExecStart=/usr/local/src/DiCE/diced -d -l
ExecReload=/usr/local/src/DiCE/diced $OPTIONS
ExecStop=/usr/local/src/DiCE/diced $OPTIONS
RemainAfterExit=yes
PrivateTmp=true

[Install]
WantedBy=runlevel3.target
```

任意のプログラムをデーモンとして実行する

サンプル1

<http://hakaikosen.hateblo.jp/entry/2014/07/15/215546>

```
[Unit]
Description=hoge server daemon
After=syslog.target network.target

[Service]
ExecStart=hoge オプション（実行したいプログラムを実行）
ExecReload=/bin/kill -HUP $MAINPID
ExecStop=/bin/kill $MAINPID

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

プログラムの終了は強制的に kill しているので注意