LinuC

# LinuC レベル1 Version10.0
# 技術解説無料セミナー
## 2020/7/18 開催

| | |
|---|---|
| 主題 | 1.01<br>Linuxのインストールと仮想マシン・コンテナの利用 |
| 副題 | 1.01.3<br>ブートプロセスとsystemd |


本日の講師

三澤 康巨


LPI-JAPAN

# 三澤 康巨

- KDDI株式会社で、電話等のネットワークサービス設備のエンジニアリングなど様々な業務を担当しました。
- 2017年10月から2年半、KDDIグループ内のサーバ研修講師を務めました。
- サーバ研修受講者の中から、200名を超える LinuC レベル1 合格者を出しました。
- 2020年3月、KDDIを定年退職しました。

## ■LinuCとは

クラウド時代の即戦力エンジニアであることを証明するLinux技術者認定資格

### ✓現場で「今」求められている新しい技術要素に対応

- オンプレミス／仮想化を問わず様々な環境下でのサーバー構築
- 他社とのコラボレーションの前提となるオープンソースへの理解
- システムの多様化に対応できるアーキテクチャへの知見

### ✓全面的に見直した、今、身につけておくべき技術範囲を網羅

今となっては使わない技術やコマンドの削除、アップデート、新領域の取り込み

### ✓Linuxの範疇だけにとどまらない領域までカバー

セキュリティや監視など、ITエンジニアであれば必須の領域もカバー

# ■Version10.0と従来の出題範囲の比較

| | テーマ | Version 10.0 | 従来 |
|---|---|---|---|
| LinuC-1 | 仮想化技術 | ・仮想マシン／コンテナの概念<br>・クラウドセキュリティの基礎 | ←<br>（Version10.0で新設） |
| | オープンソースの文化 | ・オープンソースの定義や特徴<br>・コミュニティやエコシステムへの貢献 | ←<br>（Version10.0で新設） |
| | その他 | →<br>（Version10.0で削除） | アクセシビリティ、ディスククォータ、プリンタの管理、SQLデータ管理、他 |
| LinuC-2 | 仮想化技術 | ・仮想マシンの実行と管理（KVM）<br>・コンテナの仕組みとDockerの導入 | ←<br>（Version10.0で新設） |
| | システムアーキテクチャ | ・クラウドサービス上のシステム構成<br>・高可用システム、スケーラビリティ、他 | ←<br>（Version10.0で新設） |
| | その他 | ・統合監視ツール（zabbix）<br>・自動化ツール（Ansible） | ←<br>（Version10.0で出題範囲に追加） |
| | | →<br>（Version10.0で削除） | RAID、記憶装置へのアクセス方、FTPサーバーの保護、他 |

# ブートプロセスとsystemd

1. はじめに
2. ブートプロセス
3. systemdのユニットとターゲット
4. デフォルトターゲットの変更、稼働中ターゲットの変更
5. サービスの制御
6. システムの停止・再起動

# 1. はじめに



■本セミナーでは、Linuxシステムが起動する仕組みと、システムを統括しているsystemdの取り扱いについて学習します。

■学習効果を高めるため、実行例の出てくる部分では、ご自分でも実行してみることをお勧めします。

■Linuxには多数のディストリビューションが存在しますが、本セミナーの実行例では、CentOS 7 を使用します。

- ビジネス用サーバーの多くで稼働している Red Hat Enterprise Linux 7 (RHEL7) と互換性があります。
- RHEL7は有料ですが、CentOS 7 は無料で利用できます。

# 1. はじめに



■CentOS 7 に基づく学習環境の構築方法を、LPI-Japanのサイトでご紹介しています。

LinuC レベル１／レベル２ Version 10.0 学習環境構築ガイド

https://linuc.org/docs/v10/guide_text.pdf

■学習環境構築ガイドでは、2種類の環境の構築方法を紹介しています。

【環境A】

- 用意したコンピュータの内蔵ストレージを上書きして、Linux専用コンピュータを構築します。
- WindowsやMacOS等の既存OSは使えなくなります。
- 不要になった古いPC等がある場合に、それを使ってください。

【環境B】

- WindowsやMacOS等の既存OSを壊すことなく、外付けSSDにLinuxをインストールします。
- これによって、既存OSとLinuxとの間を切り替えて利用することができます。
- 但し、既存OSとLinuxとを同時に利用することはできません。

■Linuxシステムの電源をONすると、以下のプロセスでシステムが起動します。

# 2.1 BIOS/UEFI



- 電源ONすると、BIOSまたはUEFIが起動します。

- BIOS（Basic Input/Output System）はコンピュータ内のROMに書き込まれた最も原始的なプログラム（ファームウェア）です。
- BIOSはブートデバイス（HDD、SSD等）を選択して、そこに格納されているブートローダを起動します。

- UEFI（Unified Extensible Firmware Interface）はBIOSを改良したファームウェアです。システムを起動するための機能はBIOSと同じです。

# 2.2 ブートローダ

- ブートローダはBIOSまたはUEFIによって起動されます。
- ブートローダはカーネルを選択して起動するプログラムです。
- 多くのLinuxディストリビューションは、ブートローダとしてGRUB（Grand Unified Bootloader）を採用しています。GRUBには、古いGRUB Legacyと新しいGRUB2の2種類があります。（CentOS 7 はGRUB2を採用）
- ブートローダが起動するとカウントダウンが行われ、その間にカーネルを選択することが可能です。手動のカーネル選択が行われず、カウントダウンが終了すると、ブートローダはデフォルトのカーネルを自動で選択します。
- ブートローダは、選択されたカーネルをブートデバイスから読み出して、起動します。

# 2.3 カーネルとinitramfs



- カーネルはLinuxの中心部を構成するプログラムで、CPU、メモリ、ストレージ等のリソースを制御します。Linux Foundationによって、随時バージョンが更新されています。

- ブートローダによってカーネルが起動されると、カーネルはinitramfsを起動します。

- initramfsはシステム起動用の小規模ファイルシステムで、ルートファイルシステムをルートディレクトリにマウントします。

# 2.4 systemd



- initramfsがルートファイルシステムをマウントすると、カーネルは systemd を読み出して、起動します。

- systemd はシステム起動時に最初に生成されるプロセスで、デーモンとしてメモリに常駐し、システム全体を統括します。PID（プロセスID）が必ず1となります。
  （CentOS 6 以前、最初に生成するプロセスは init でした）

- システム起動時、systemd はシステム稼働に必要な子プロセスを順に生成していき、システムを初期化します。

**systemdと子プロセス**

```
# ps ax
  PID TTY      STAT   TIME COMMAND
    1 ?        Ss     0:03 /usr/lib/systemd/systemd --switched-root --system --deserializ
    2 ?        S      0:00 [kthreadd]
    4 ?        S<     0:00 [kworker/0:0H]
. . .
```

# 3. systemdのユニットとターゲット



- システム起動時、systemdは設定ファイルに従ってシステムの初期化とサービスの起動を行います。

- systemdの設定ファイルはユニット(unit)と呼ばれ、12タイプのユニットがあります。
  service、target、socket、device、mount、automount、swap、path、timer、snapshot、slice、scope
- serviceユニットには、サービスの起動と停止に関する設定が記述されています。
- target はユニットをグループ化したもので、システムの起動状態を制御する設定です。
- service と target の詳細は後述します。

- ユニットのファイル名は「ユニット名.ユニットタイプ」となっています。例えば、www（ウェブ）サービスの起動と停止に関する設定は「httpd.service」に記述されます。

（参考）ユニットファイルは /usr/lib/systemd/system ディレクトリと /etc/systemd/system ディレクトリに格納されています。

# 3. systemdのユニットとターゲット



- システム起動時、systemd は default.target の設定に従ってシステムを起動します。
- default.target は一般的に graphical.target または multi-user.target へのシンボリックリンクになっています。graphical.target へリンクしている場合は、システムがグラフィカルモードで起動します。multi-user.target へリンクしている場合は、マルチユーザのCUIモードで起動します。
- レスキューモードは、rootユーザのみが利用可能で、システムのメンテナンスに利用されます。

| ターゲット | 起動モード | システムの状態 | init のランレベル |
|---|---|---|---|
| graphical.target | グラフィカルモード | マルチユーザ＋GUI | 5 |
| multi-user.target | マルチユーザモード | マルチユーザ＋CUI | 3 |
| rescue.target | レスキューモード | シングルユーザ（rootのみ） | 1 |
| poweroff.target | 電源オフ | システム停止 | 0 |
| reboot.target | システム再起動 | | 6 |

■systemd を使ってシステムを管理するには systemctlコマンドを使用します。

**systemctl コマンド**

```
systemctl [オプション] サブコマンド [ユニットファイル名]
```

serviceユニットの場合は「ユニットファイル名」中の「.service」を省略できます。

■デフォルトターゲットを管理するには、次のサブコマンドを使用します。

| | |
|---|---|
| get-default | デフォルトターゲットの確認 |
| set-default | デフォルトターゲットの変更 |

## デフォルトターゲットの確認とマルチユーザモードへの変更

```
# systemctl get-default
graphical.target

# systemctl set-default multi-user.target
Removed symlink /etc/systemd/system/default.target.
Created symlink from /etc/systemd/system/default.target to /usr/lib/systemd/system/multi-user.target.

# systemctl get-default
multi-user.target

# ls -l /etc/systemd/system/default.target
lrwxrwxrwx. 1 root root 41  7月  6 14:43 /etc/systemd/system/default.target -> /usr/lib/systemd/system/multi-user.target
```

# 4. デフォルトターゲットの変更、稼働中ターゲットの変更



## デフォルトターゲットの確認とグラフィカルモードへの変更

```
# systemctl get-default
multi-user.target

# systemctl set-default graphical.target
Removed symlink /etc/systemd/system/default.target.
Created symlink from /etc/systemd/system/default.target to
/usr/lib/systemd/system/graphical.target.

# systemctl get-default
graphical.target

# ls -l /etc/systemd/system/default.target
lrwxrwxrwx. 1 root root 40  7月  6 14:47 /etc/systemd/system/default.target ->
/usr/lib/systemd/system/graphical.target
```

■デフォルトターゲットの変更は、次回のシステム起動時に反映されます。

# 4. デフォルトターゲットの変更、稼働中ターゲットの変更

■稼働中のターゲットを変更するには サブコマンド isolate を使用します。

**multi-user.target に変更する**

```
# systemctl isolate multi-user.target
login:       ※CUIのログインプロンプトが表示されます
```

**graphical.target に変更する**

```
# systemctl isolate graphical.target
             ※GUIのログイン画面が表示されます
```

**rescue.target に変更する**

```
# systemctl isolate rescue.target
#            ※rootでログインした状態になります
```

poweroff.target に変更　→ システム停止
reboot.target に変更　→ システム再起動

# 5. サービスの制御



- システム起動時、systemd はserviceユニット「サービス名.service」の設定に従って各種サービスを起動します。
- serviceユニットの起動設定が enabled または static ならば、systemd がそのサービスを自動的に起動します。disabled の場合は自動的に起動しません。

代表的な serviceユニット

| serviceユニット | サービス |
|---|---|
| httpd.service | www（ウェブ）サービス |
| sshd.service | SSH（リモートアクセス）サービス |
| firewalld.service | ファイアウォールサービス |
| crond.service | cron（ジョブスケジューリング）サービス |
| rsyslog.service | Syslogサービス |

■サービスの起動設定と現在の状態を確認するには、サブコマンド status を使用します。

**サービスの起動設定と現在の状態を確認する**

```
# systemctl status httpd.service      ※「.service」は省略可能
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset:
disabled)                                                ※disabled なので、自動的に起動しない。
   Active: inactive (dead)      ※現在は起動していない
     Docs: man:httpd(8)
           man:apachectl(8)
```

サービスの起動設定を確認するには、サブコマンド is-enabled も使用できます。

**サービスの起動設定を確認する**

```
# systemctl is-enabled httpd.service      ※「.service」は省略可能
disabled
```

■サービスの起動設定を変更するには、次のサブコマンドを使用します。

| | |
|---|---|
| enable | enabled（自動的に起動する設定）に変更する |
| disable | disabled（自動的に起動しない設定）に変更する |

**サービスを enabled（自動的に起動する設定）に変更する**

```
# systemctl enable httpd.service      ※「.service」は省略可能
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to
/usr/lib/systemd/system/httpd.service.

# systemctl status httpd.service
●httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset:
disabled)                                  ※enabled に変わり、次回システム起動時に自動的に起動する。
   Active: inactive (dead)      ※現在の inactive 状態は変わらない。
     Docs: man:httpd(8)
           man:apachectl(8)

# ls -l /etc/systemd/system/multi-user.target.wants/httpd.service
lrwxrwxrwx. 1 root root 37  7月  6 15:13 /etc/systemd/system/multi-
user.target.wants/httpd.service -> /usr/lib/systemd/system/httpd.service
     ※シンボリックリンクファイルhttpd.serviceが生成した。
```

## サービスを disabled（自動的に起動しない設定）に変更する

```
# systemctl disable httpd.service      ※「.service」は省略可能
Removed symlink /etc/systemd/system/multi-user.target.wants/httpd.service.

# systemctl status httpd.service
●httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset:
disabled)                                          ※disabled に戻った。
   Active: inactive (dead)
     Docs: man:httpd(8)
           man:apachectl(8)

# ls -l /etc/systemd/system/multi-user.target.wants/httpd.service
ls: /etc/systemd/system/multi-user.target.wants/httpd.service にアクセスできません: そのよ
うなファイルやディレクトリはありません
     ※シンボリックリンクファイルhttpd.serviceが削除された。
```

- enabled または disabled に変更しても、サービスの現在の状態は変わらず、次にシステムが起動する時に反映されます。

# 5. サービスの制御



■サービスの現在の状態を変更するには、次のサブコマンドを使用します。

| | |
|---|---|
| start | 起動する：active（起動している状態）に変更する |
| stop | 停止する：inactive（起動していない状態）に変更する |

**サービスを起動する**

```
# systemctl start httpd.service      ※「.service」は省略可能

# systemctl status httpd.service
●httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset:
disabled)                                        ※disabled なので、次回システム起動時には停止する。
   Active: active (running) since 月 2020-07-06 15:23:17 JST; 17s ago
・・・      ※active に変わった。
 Main PID: 5433 (httpd)
・・・
```

## サービスを起動する（続き）

```
# ps ax | grep httpd
 5433 ?        Ss     0:00 /usr/sbin/httpd -DFOREGROUND      ※httpdのプロセス
 5436 ?        S      0:00 /usr/sbin/httpd -DFOREGROUND        （デーモン）が生成した。
 5437 ?        S      0:00 /usr/sbin/httpd -DFOREGROUND
 5438 ?        S      0:00 /usr/sbin/httpd -DFOREGROUND
 5439 ?        S      0:00 /usr/sbin/httpd -DFOREGROUND
 5440 ?        S      0:00 /usr/sbin/httpd -DFOREGROUND
 6839 pts/0    R+     0:00 grep --color=auto httpd
```

## サービスを停止する

```
# systemctl stop httpd.service      ※「.service」は省略可能

# systemctl status httpd.service
●httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset:
disabled)
   Active: inactive (dead)      ※inactive に戻った。
 . . .
# ps ax | grep httpd
 6916 pts/0    R+     0:00 grep --color=auto httpd      ※httpdデーモンが消滅した。
```

# 6. システムの停止・再起動



- graphical.target で稼働している時は、Windowsと同様にGUIメニューからシステムを停止・再起動させることができます。

- CUIでシステムを停止・再起動させるコマンドは何種類か存在します。

# 6. システムの停止・再起動



■shutdownコマンドは、指定した時刻にシステムを停止・再起動させます。

**システムを停止する**

```
shutdown -h 時刻
```

**システムを再起動する**

```
shutdown -r 時刻
```

| 時刻 | 説明 |
|---|---|
| now | すぐに停止・再起動 |
| +n または n | n分後に停止・再起動 |
| hh:mm | hh時mm分に停止・再起動 |

■shutdownコマンドで設定したシステム停止・再起動が実行される前に取消すには、オプション -c を使用します。

**システム停止・再起動を取消す**

```
shutdown -c
```

## shutdownコマンドの実行例

```
# shutdown -h 16
Shutdown scheduled for 月 2020-07-06 15:58:27 JST, use 'shutdown -c' to cancel.
#
Broadcast message from root@totsuka (Mon 2020-07-06 15:43:27 JST):
                                        ※警告メッセージがシャットダウン開始15分前に出力される。
The system is going down for power-off at Mon 2020-07-06 15:58:27 JST!

                ※シャットダウン開始5分前を過ぎると、一般ユーザの新規ログインは禁止される。
(ENTER)
# shutdown -c

Broadcast message from root@totsuka (Mon 2020-07-06 15:53:49 JST):

The system shutdown has been cancelled at Mon 2020-07-06 15:54:49 JST!
```

■システムをすぐに停止するには、shutdown -h now に加えて、以下のコマンドを使用することができます。

**システムをすぐに停止する**

```
poweroff
systemctl poweroff
systemctl isolate poweroff.target
```

halt コマンドを使ってもシステムがすぐに停止しますが、電源ONのままとなります。

■システムをすぐに再起動するには、shutdown -r now に加えて、以下のコマンドを使用することができます。

**システムをすぐに再起動する**

```
reboot
systemctl reboot
systemctl isolate reboot.target
```

# 本セミナーのまとめ



1. CentOS 7 に基づく学習環境を自分で構築してみましょう。
2. 電源ONすると、BIOS/UEFI→ブートローダ→カーネル→initramfs→systemdの順にシステムが起動します。systemdは最初に生成されるプロセスで、デーモンとして常駐し、システム全体を統括します。
3. systemdの設定ファイルはユニットと呼ばれます。serviceユニットにはサービスの起動設定が記述されています。targetはユニットをグループ化したもので、systemdは default.target の設定に従ってシステムを起動します。
4. デフォルトターゲットの設定を確認するには systemctl get-default 、設定を変更するには systemctl set-default を使用します。稼働中のターゲットを変更するには systemctl isolate を使用します。

# 本セミナーのまとめ



5. systemdはserviceユニット「サービス名.service」の設定に従って各種サービスを起動します。サービスの設定と状態を確認するには、systemctl status を使用します。サービスを起動する/しない設定に変更するには、systemctl enable/disable を使用します。サービスをすぐに起動する/停止するには、systemctl start/stop を使用します。
6. shutdownコマンドは、指定した時刻にシステムを停止・再起動させます。システムをすぐに停止するには poweroffコマンド、すぐに再起動するには rebootコマンドも使用できます。