

systemdの基本

概要

systemdは、システム設定とサービスを管理する場合に使用します。systemdは、タスクを「ユニット」と呼ばれるコンポーネントに編成し、ユニットのグループを「ターゲット」に編成します。

目的

systemdの基本について習得します。これには、サービス管理、依存関係の追跡、ログ記録、リソース管理、ソケットの有効化、システム制御などの重要な機能が含まれます。

所要時間

理解に20分ほどを要します。

要件

- Linuxコマンドの基本的な理解
- Linuxのプロセス、デーモン、およびコントロールグループの基本的な理解

発行日: 11/12/2025

目次

- 1 systemdとは 3
- 2 systemdのブートプロセスについて 3

3	ユニットファイルの構造	7
4	ユニットファイルのタイプ	8
5	ユニットの依存関係と順序	10
6	ログ記録	11
7	systemdのターゲット	12
8	通常ユーザとしてのsystemdの使用	13
9	systemctl コマンド	13
10	systemdのトラブルシューティング	15
11	systemdのベストプラクティス	17
12	法的事項	18
A	GNUフリー文書利用許諾契約書(GFDL)	19

1 systemdとは

`systemd`は、Linuxオペレーティングシステムのシステムおよびサービスマネージャです。主要なLinuxディストリビューションでデフォルトの初期化システムです。`systemd`は、ユーザが直接開始するのではなく、`/sbin/init`を介してインストールされ、初期ブート中に起動されます。`systemd`は、ブート時に最初のプロセス(PID 1)として実行されたときにユーザスペースサービスを起動および維持するinitシステムとして機能します。PID 1は`init`と呼ばれ、最初に作成されるLinuxユーザモードプロセスです。その実行はシステムのシャットダウンまで続きます。

`systemd`はPID 1を所有し、カーネルによって直接起動されます。他のすべてのプログラムは、`systemd`またはそのいずれかの子プロセスによって直接起動します。`systemd`は、ホストのファイルシステムをマウントし、一時ファイルを管理します。SysV initスクリプトとの下位互換性があります。SysVは、`systemd`よりも以前の初期化システムです。

`systemd`では、ユニットとは、システムが操作および管理方法を認識しているリソースのことです。これは、`systemd`ツールで使用されるプライマリオブジェクトです。これらのリソースは、設定ファイルであるユニットファイルで定義します。

`systemctl`は、initシステムを制御する中央管理ツールです。`systemd`システムおよびサービスマネージャの状態を調べ、制御するために使用されます。

`systemd`でいうターゲットとは、システムのブート中に同期ポイントとして機能する関連ユニットのグループです。ターゲットのユニットファイルの拡張子は`.target`です。ターゲットのユニットは、依存関係の連鎖を通じてさまざまな`systemd`ユニットをまとめたものです。

トラブルシューティングには`journalctl`を使用できます。これを使用して、`systemd`ジャーナルからのログメッセージを問い合わせたり、表示したりします。

`systemd`の詳細については、<https://systemd.io> および `man 1 systemd`を参照してください。

2 systemdのブートプロセスについて

ブートプロセスの最初の手順は、LinuxオペレーティングシステムのメインコンポーネントであるLinuxカーネルをロードすることです。ロードされると、カーネルはハードウェアを初期化し、システムで実行される最初のプロセスである`systemd`プロセスを開始します。

2.1 Linuxのブートプロセス

Linuxのブートプロセスは、オペレーティングシステムの起動の初期ステージです。これは、オペレーティングシステムがメモリをロードし、コンポーネントを初期化してユーザアプリケーションの実行を準備するプロセスです。

Linuxのブートプロセスは次の4つの主要なステージに分けられます。

ステージ1: BIOS

コンピュータの電源をオンにすると、コンピュータはBIOS (Basic Input/Output System)を起動し、POST (Power On Self Test)を実行します。POSTは、ハードディスク、SSD、キーボード、RAM、USBポートなどのコンポーネントやその他のハードウェアの機能を検査する整合性チェックです。ハードウェアが期待どおりに動作する場合、ブートプロセスは次のステージに進みます。

ステージ2: ブートローダ

POSTが完了すると、BIOSはMBR (マスターブートレコード)に保存されているブートローダプログラムを検索してロードします。MBRは512バイトのコードで、通常はハードドライブのアーキテクチャに応じて `/dev/sda` または `/dev/hda` に配置されています。MBRは、LinuxのライブUSBまたはDVDインストール上に配置することもできます。BIOSは、このMBRコードをロードして実行します。

Linuxには、主要なブートローダとしてLILO、GRUB、およびGRUB2の3つがあります。GRUB2 (Grand Unified Bootloader)ブートローダは、最新のLinuxディストリビューションにおける最新のプライマリブートローダです。GRUB2の設定ファイルは `/boot/grub2/grub2.cfg` にあります。BIOSは、GRUB2ブートローダを見つけると、GRUB2を実行してメインメモリ(RAM)にロードします。

ステージ3: Linuxカーネルの初期化

Linuxカーネルは、オペレーティングシステムの心臓部です。Linuxシステムでは、カーネルはハードウェアとのインタフェースとして機能し、メモリ管理を制御し、プロセスを管理します。ブートローダは、選択したLinuxカーネルをロードします。カーネルは、圧縮されたバージョンから自己抽出され、ルートファイルシステムをマウントします。その後、 `/sbin/init` プログラムを実行します。

ステージ4: `systemd`

カーネルは、Linuxオペレーティングシステムのシステムおよびサービスマネージャである `systemd` をロードします。続いて、 `systemd` が他のすべての初期化プロセスを実行します。

2.2 systemdによるブートプロセス

カーネルがsystemdをロードすると、systemdが処理を引き継ぎ、システムを稼働させるために必要な他のシステムサービスを開始します。これには、ネットワーキングサービス、ログインマネージャなどのサービスが含まれます。

ブートプロセスは、特定のターゲットユニットの実行順に並列化されます。systemdは、`/etc/systemd/system/default.target`ファイルを使用して、Linuxシステムをブートするターゲットを判断します。このファイルは、グラフィカルログインマネージャをブートする`graphical.target`へのリンクになっています。systemdは、`default.target`の依存関係であるすべてのターゲットユニットを有効にし、さらにこれらの依存関係の依存関係すべてを再帰的に有効にします。すべてのサービスが開始されると、システムの使用準備が整い、ログインマネージャが表示されます。これで、システムにログインして、システムの使用を開始できるようになります。

2.3 systemd-analyzeコマンドによるシステムブートプロセスのパフォーマンスの分析

`systemd-analyze`コマンドを使用して、システムブートプロセスのパフォーマンスを分析します。また、システムおよびサービスマネージャから他の状態情報やトレース情報を取得することもできます。このコマンドを使用して、ユニットファイルが正しいことを確認したり、システムマネージャの高度なデバッグに役立つ特別な機能にアクセスしたりします。

次に例を示します。

システムのブートにかかる時間の表示

```
> systemd-analyze time
Startup finished in 3.404s (kernel) + 2.415s (initrd) + 13.125s (userspace) =
 18.945s
graphical.target reached after 13.117s in userspace
```

ブートプロセスの概要の取得(開始されているサービスや、各サービスの開始にかかる時間など)

```
> systemd-analyze critical-chain
The time when unit became active or started is printed after the "@" character.
The time the unit took to start is printed after the "+" character.

graphical.target @13.117s
├─multi-user.target @13.117s
│   └─getty.target @13.117s
│       └─getty@tty1.service @13.116s
```

```

└─plymouth-quit-wait.service @10.775s +2.338s
  └─systemd-user-sessions.service @10.769s +3ms
    └─remote-fs.target @10.764s
      └─iscsi.service @10.747s +16ms
        └─network-online.target @10.744s
          └─NetworkManager-wait-online.service @1.547s +9.197s
            └─NetworkManager.service @1.507s +37ms
              └─network-pre.target @1.504s
                └─wpa_supplicant.service @2.341s +5ms
                  └─dbus.service @1.042s
                    └─basic.target @1.036s
                      └─sockets.target @1.036s
                        └─snapd.socket @1.035s +590us
                          └─sysinit.target @1.030s
                            └─systemd-update-utmp.service @1.025s +5ms
                              └─auditd.service @976ms +47ms
                                └─systemd-tmpfiles-setup.service @964ms +9ms
                                  └─local-fs.target @962ms
                                    └─snapd.mounts.target @961ms
                                      └─snap-core18-2796.mount @417ms +543ms
                                        └─dev-loop9.device @961ms +628us

```

このコマンドは、指定した各ユニットまたはデフォルトのターゲットについて、タイムクリティカルなユニットのツリーを出力します。サービスの初期化は、ソケットの有効化とユニットの並列実行に依存する場合があります。**blame**コマンドと同様に、ユニットの有効化にかかる時間を表示します。これは、アクティブ状態に直接移行するデバイスユニットなどのユニットでは定義されていません。

ブートプロセス中に開始されたサービスのリストの表示(各サービスにかかった時間順に表示される)

```

> systemd-analyze blame
9.197s NetworkManager-wait-online.service
4.002s fwupd.service
2.338s plymouth-quit-wait.service
1.282s dracut-pre-udev.service
1.062s sys-devices-platform-serial8250-tty-ttyS0.device
1.062s dev-ttyS0.device
1.061s dev-ttyS1.device
1.061s sys-devices-platform-serial8250-tty-ttyS1.device
1.060s dev-ttyS11.device
1.060s sys-devices-platform-serial8250-tty-ttyS11.device
1.059s sys-devices-platform-serial8250-tty-ttyS13.device
1.059s dev-ttyS13.device
1.059s sys-devices-platform-serial8250-tty-ttyS10.device
1.059s dev-ttyS10.device
1.058s sys-devices-platform-serial8250-tty-ttyS14.device
1.058s dev-ttyS14.device
1.058s dev-ttyS12.device

```

```
1.058s sys-devices-platform-serial8250-tty-ttyS12.device
1.056s sys-devices-platform-serial8250-tty-ttyS17.device
```

他のサービスの初期化が完了するまで待つために、あるサービスの初期化が遅くなる場合があります。このコマンドを使用して、ユニットの有効化にかかる時間を表示します。これは、アクティブ状態に直接移行するデバイスユニットなどのユニットでは定義されていません。Type=simpleであるサービスの結果は表示されません。systemdは、これらのサービスは直ちに開始されると見なすので、初期化の遅延を分析できないためです。

ブートプロセス中に発生するイベントを表示するベクタグラフィックファイルの生成

```
> systemd-analyze plot > /temp/sample.svg
```

このコマンドは、tempディレクトリにSVGファイルを作成します。SVGファイルは、一連のグラフィックベクタを定義したテキストファイルで、LibreOffice Drawなどのアプリケーションでグラフを生成するために使用されます。

3 ユニットファイルの構造

systemdでは、ユニットとは、システムが操作および管理方法を認識しているリソースを指します。これは、systemdツールで使用されるプライマリオブジェクトです。これらのリソースは、設定ファイルであるユニットファイルを使用して定義します。systemdを使用する作業では、ユニットファイルを理解していると管理が容易になります。ユニットファイルではシンプルな宣言型の構文を使用するので、有効化したときにユニットの目的と効果を簡単に確認できます。ユニットファイルには、次のようにディレクティブを記述したセクションがあります。

```
[Section]
Directive1=value
Directive2=value
. . .
```

各タイプのユニットファイルには以下のセクションがあります。

[Unit]

ほとんどのユニットファイルの先頭には[Unit]セクションがあります。このセクションを使用して、ユニットファイルのメタデータを定義し、ユニットファイルの他のユニットファイルとの関係を設定します。このセクションは、ユニットファイルの概要を記述していることから通常は先頭に配置されます。

[Automount] / [Mount] / [Path] / [Service] / [Slice] / [Socket] / [Swap] /
[Timer]

ユニットファイルのタイプごとに固有のディレクティブを記述したセクションです。使用可能なすべてのユニットファイルのタイプについては4項「[ユニットファイルのタイプ](#)」を参照してください。タイプ `device`、`target`、`snapshot`、および `scope` には、タイプ固有のセクションがないことに注意してください。

[Install]

これは多くの場合、ユニットファイルの最後のセクションであり、オプションです。このセクションを使用して、ユニットファイルが有効または無効なときのその動作を定義します。ユニットファイルを有効にすると、そのファイルはブート時に自動的に起動します。特定のユニットでは、適切に動作するために他の関連ユニットとの依存関係を持っていることがあります。たとえば、`chrony` にはディレクティブ `After`、`Wants`、および `Before` が必要で、これらはすべて `chrony` が動作するための依存関係です。

例 1: `systemd` のサービスファイル

```
[Unit]
Description=usbguard ❶

[Service]
ExecStart=/usr/sbin/usb-daemon ❷

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

- ❶ サービスファイルの目的の簡単でわかりやすい説明。
- ❷ サービスの開始時に実行するプログラムを指定します。
- ❸ ネットワーキングを備えたマルチユーザシステムを起動します。グラフィカルな環境はありません。このディレクティブで依存関係を指定できます。

4 ユニットファイルのタイプ

ファイル拡張子でユニットのタイプを判断できます。`systemd` では、記述されたりソースのタイプによってユニットが分類されています。

`systemd` で使用可能なユニットファイルのタイプは次のとおりです。

.service

サービスやアプリケーションをどのように管理するかを記述します。これには、サービスの開始または停止方法、設定ファイルの再ロード方法(該当する場合)、サービスを自動的に開始する条件、関連するユニットファイルの依存関係または階層情報が含まれます。

.scope

このユニットファイルは、`systemd`によって、D-Busインタフェースから受信した情報から自動的に作成され、外部で作成された一連のシステムプロセスを管理するために使用されます。

.path

パスを使用した有効化で使用するパスを定義します。デフォルトでは、ファイル名のベース名が同じ `.service` ユニットファイルが有効になります。`inotify`は、ファイルの変更に関する通知を受け取る必要があるプログラムで使用するカーネルAPIです。

.snapshot

`systemctl snapshot` コマンドによって、`.snapshot` ユニットファイルが自動的に作成されます。このコマンドでは、システムの現在の状態を表す一時的なスナップショットが作成されます。システムを変更するとその現在の状態を変更できます。一時的な状態のロールバックにはスナップショットを使用します。

.timer

`systemd`で管理するタイマを定義します。このタイマは、遅延または実行スケジュールを設定した有効化で使用するcronジョブに似ています。このタイマが設定値に達すると、ベース名が同じでファイル拡張子が `.service` であるユニットファイルが起動します。

.slice

Linuxのコントロールグループのノードを関連付けます。これにより、スライスに関連付けたプロセスへのリソース割り当てやリソース制限ができます。この名前は、コントロールグループツリーにおける階層を示します。ユニットは、そのタイプに応じてデフォルトでスライスに配置されます。

.target

ブート時または状態の変更時に他のユニットと同期状態にするか、システムを新しい状態にします。他のユニットでは、ターゲットの操作と同期するためにターゲットとの関係を指定します。

.socket

ソケットベースの有効化でsystemdが使用するネットワーク、IPCソケット、またはFIFOバッファを記述します。このユニットで定義したソケット上でアクティビティが確認されると起動する.serviceファイルがあります。

.device

ファイルシステムudevまたはsysfsでsystemdによる管理が指示されているデバイスを定義します。すべてのデバイスに.deviceファイルが存在するわけではありません。デバイスの順序指定、マウント、またはアクセスで、このユニットファイルが必要です。

.swap

システムのスワップ空間を定義します。ユニットファイルの名前は、この空間のデバイスまたはファイルパスを反映している必要があります。

.mount

systemdで管理するシステムのマウントポイントを定義します。このファイルには、スラッシュをダッシュに変更したマウントパスに基づく名前が割り当てられます。/etc/fstabのエントリには、自動的に作成されたユニットを指定できます。

.automount

自動的にマウントされるマウントポイントを定義します。参照するマウントポイントに基づくファイル名をファイルに割り当てます。マウントの詳細を定義するには、ベース名が同じ.mountユニットファイルが必要です。

5 ユニットの依存関係と順序

systemdには2種類の依存関係があります。要件の依存関係と順序の依存関係です。要件の依存関係では、ユニットを有効にするときに他のどのユニットを開始または停止するかを指定します。順序の依存関係では、ユニットを開始する順序を指定します。

ユニットの依存関係

ユニットファイルには依存関係の機能があります。ユニットを実行するために、他のユニットが1つ以上必要な場合があります。これらの依存関係は、ディレクティブWantsとRequiresを使用してユニットファイルに設定します。

Wants

たとえば、ユニットAにWants=unit Bを記述している場合、ユニットAを実行するとユニットBも実行されます。ただし、ユニットBが正常に開始しなくても、正常に実行されているユニットAには何の影響もありません。

Requires

ユニットAにRequires=unit Bを記述している場合、両方のユニットが実行されますが、ユニットBが正常に実行されないとユニットAは無効になります。ユニットAのプロセスが正常に実行されているかどうかは関係ありません。

ユニットの順序

適切な指示がない場合、systemdではグループに属するユニットを同時に実行できます。正しい順序でサービスを開始することは、Linuxシステムが適切に機能するうえで重要です。ユニットファイルディレクティブBeforeとAfterを使用して順序を設定できます。

Before

たとえば、ユニットAにBefore=unit Bを記述している場合、両方のユニットを実行すると、ユニットAが正常な実行状態になってからユニットBが実行されます。

After

ユニットAにAfter=unit Bを記述している場合、両方のユニットを実行すると、ユニットBが正常な実行状態になってからユニットAが実行されます。

6 ログ記録

ログファイルとジャーナルは、システム管理者にとって重要です。これらはシステムに関する詳細な情報を提供し、トラブルシューティングと監査において非常に重要です。ログファイルには、カーネル、アプリケーション、およびシステムにログインするユーザによって生成されたイベントとメッセージが含まれています。journalctlコマンドを使用してジャーナルに問い合わせることができます。このコマンドによって、systemdで収集されたログが表示されます。systemd-journaldサービスはsystemdのログ収集を扱います。systemd-journaldは、イベントとメッセージをバイナリ形式で保存します。

7 systemdのターゲット

systemdは、ユニットとターゲットを使用します。systemdユニットは、システム上のサービスまたはアクションを定義し、名前、タイプ、および設定ファイルで構成されます。systemdターゲットは、複数のユニットを組み合わせ、ターゲットを達成するにはどのサービスを開始する必要があるかを定義します。たとえば、サーバの場合、これはネットワークが実行されていて複数のユーザがログインできる状態です。これらのファイルは、サフィックス `.target` で識別できます。

ユニットファイルと同様に、さまざまなターゲットを依存関係でネストできます。たとえば、`multi-user.target` では、(特に)ログインおよびユーザセッションサービスを設定するターゲットが必要です。

systemdの一般的なターゲット

`default.target`

デフォルトでブートします。`default.target` ファイルは、デスクトップワークステーションの `graphical.target` など、真のターゲットファイルへのシンボリックリンクです。サーバの場合、通常は `graphical.target` です。

`poweroff.target`

システムをシャットダウンして、電源をオフにします。

`rescue.target`

ベースシステムをプルして、レスキューシェルセッションを開始するターゲットユニット。

`multi-user.target`

グラフィカルではない(コンソール)マルチユーザシステムをセットアップします。

`graphical.target`

ネットワークサービスを備えたグラフィカルマルチユーザシステムを使用します。

`reboot.target`

システムをシャットダウンして再起動します。

systemdターゲットの詳細については、**man 5 systemd.target** および **man 7 systemd.special** を参照してください。

8 通常のユーザとしてのsystemdの使用

セキュリティを強化する場合や、rootユーザ特権がない場合、systemdを通常のユーザとして使用できます。非特権サービスを実行するには、userサービスを作成します。

ユーザサービスを作成および使用する際には、次の点を考慮してください。

- ユーザサービスセッションは、ユーザのセッションが終了すると終了します。loginctl enable-linger USERNAMEコマンドを使用することで、この設定を無効にすることができます。
- ユーザサービスファイルは、/etc/systemd/userまたは $\$HOME/.config/systemd/user/$ にあります。
- systemctl --userコマンドを使用してユーザサービスを制御できます。

9 systemctlコマンド

systemctlコマンドは、systemdおよびサービスマネージャの状態を調べ、制御するために使用されます。

次の一般的なsystemctlコマンドを使用できます。**man systemctl**のページを参照してください。

9.1 systemdの情報の表示

systemdコンポーネントに関する情報を表示するには、次のコマンドを使用できます。

systemctl list-units

systemdユニットを一覧表示します。オプションの引数`--state=running`を使用するとアクティブなユニットを表示し、`--type=service`を使用すると、終了したアクティブなユニットを表示できます。

systemctl list-unit-files

systemdのユニットとステータスを一覧にします。ステータスには、静的、生成済み、無効、エイリアス、マスク、有効などがあります。

systemctl list-dependencies

依存関係ツリーを一覧表示します。

systemctl list-dependencies UNIT_FILE

ユニットファイルの依存関係を一覧表示します。

9.2 systemdサービスの管理

systemctl コマンドでは、サービスに関する次のタスクを実行できます。

systemctl status SERVICE

特定のサービスの状態を確認します。

systemctl show SERVICE

サービス情報を表示します。

systemctl start SERVICE

サービスを手動で開始する代わりに **start** コマンドを使用します。設定ファイルの記述を変更した場合は、関連するサービスを開始する必要があります。

systemctl stop SERVICE

実行している特定のサービスを停止します。

systemctl restart SERVICE

サービスを手動で再開する代わりに **restart** コマンドを使用します。設定ファイルの記述を変更した場合は、関連するサービスを再開する必要があります。

systemctl enable SERVICE

ブート時にサービスを有効にします。

systemctl disable SERVICE

ブート時にサービスを無効にします。

systemctl reload-or-restart SERVICE

再ロードできるサービスであれば再ロードし、再ロードできないサービスであれば再開します。サービスが実行されていない場合は再開します。

systemctl mask SERVICE

サービスがマスクされている場合、ユニットファイルが /dev/null にシンボリックリンクされています。マスクされたサービスのシンボリックリンクは、/dev/null を指すように /etc/systemd/system から作成されます。その結果、サービスを別の有効なサー

ビスが必要としていても、そのサービスはロードできません。このようなサービスは手動で停止する必要があります。停止しないとバックグラウンドで実行が継続します。`--runtime` オプションを使用すると、システムの次回再起動までサービスを一時的にマスクできます。

```
Created symlink /etc/systemd/system/FOSSLinux.service → /dev/null.
```

systemctl unmask SERVICE

サービスのマスクを解除します。システムを手動で開始または再開する場合に効果的です。

9.3 システムの状態の管理

`systemctl` コマンドでは、再起動、シャットダウンなど、システム上で次に示すような電源管理プロセスを実行できます。

systemctl reboot

システムを再起動します(`reboot.target`)。

systemctl poweroff

システムの電源をオフにします(`poweroff.target`)。

systemctl emergency

緊急モードへ移行します(`emergency.target`)。

systemctl default

デフォルトのターゲットへ戻ります(`multi-user.target`)。

10 systemdのトラブルシューティング

次のトラブルシューティングのヒントを使用すると、`systemd` サービスの問題を特定して解決し、システムの円滑な動作を保証できます。

systemd-analyze verify SERVICE を使用して `systemd` ユニットファイルの構文を確認

`systemd` サービスを開始または有効化する前に、ユニットファイルの構文にエラーがないことを確認します。次に例を示します。

```
> sudo systemd-analyze verify /etc/systemd/system/my-custom-service.service
```

このコマンドは、ユニットファイルを分析して、構文エラー、不足しているファイルなどの問題を報告します。サービスを有効にして開始する前に、報告された問題を修正する必要があります。

journalctl -u SERVICE コマンドを使用してサービスのログを確認

`systemd` サービスで問題が発生した場合は、そのサービスのログを確認します。次に例を示します。

```
> sudo journalctl -u my-custom-service.service
```

このコマンドは指定されたサービスのログを表示します。このログにはエラーメッセージや警告などの関連情報が記録されています。これらのログを使用して、サービスの問題を特定して修正できます。

systemd-analyze plot コマンドを使用してブートプロセスを視覚化

ブートプロセス中にサービスで問題が発生している場合は、**systemd-analyze plot command** を使用してブートプロセスを視覚化し、問題を特定できます。次に例を示します。

```
> sudo systemd-analyze plot > boot-plot.svg
```

このコマンドでは、ブートプロセスと潜在的な問題のグラフィカル表現が含まれる `boot-plot.svg` という SVG ファイルが作成されます。各サービスの開始時刻と停止時刻も含まれます。このファイルを SVG 互換の画像ビューアまたは Web ブラウザで開き、ブートプロセスで発生する問題の原因となっているサービスを分析できます。

障害が発生したサービスのトラブルシューティング

障害が発生したサービスを確認し、ログ出力を検査するには、次のコマンドを実行します。

```
> sudo systemctl --state=failed
```

実行しているサービスの状態を確認

実行しているサービスの現在の状態を確認するには次のコマンドを実行します。

```
> sudo systemctl status SERVICE
```

長時間を要するシャットダウンや再起動

シャットダウンや再起動に長時間を要する場合は、終了していないサービスがあることが考えられます。`systemd` は、各サービスの強制終了を試みる前に、その終了を一定時間待機します。多く見られる問題は、保留状態のサービスやシャットダウンの停止です。これを確認するには次のコマンドを使用します。

```
> sudo systemctl poweroff
Failed to power off system via logind: There's already a shutdown or sleep
operation in progress
```

```
> sudo systemctl list-jobs
```

実行中および待機中のジョブをキャンセルし、再度シャットダウンまたは再起動することができます。

```
> sudo systemctl cancel
```

```
> sudo systemctl stop systemd-suspend.service
```

11 systemdのベストプラクティス

いくつかのベストプラクティスに従うことで、さまざまな状況に対応できる効率的なsystemdサービスを実現できます。

実行しているサービスの状態を確認

実行しているサービスの現在の状態を確認するには次のコマンドを実行します。

```
> sudo systemctl status SERVICE
```

systemdユニットファイルでは絶対パスを使用

設定ファイルや、systemdユニットファイルのスクリプトなど、実行可能ファイルや必須ファイルには絶対パスを使用します。systemdでは、`$PATH`などのユーザの環境変数がファイルの検索に使用されません。

ExecReloadディレクティブの使用

`systemctl reload`コマンドでサービスを再ロードするときに実行する特定のコマンドを定義するには、`[SERVICE]`セクションでExecReloadディレクティブを使用します。再起動を伴わずに設定を動的に再ロードできるサービスで、このディレクティブが効果的です。

```
[Service]
ExecStart=PATH_TO_EXECUTABLE
ExecReload=PATH_TO_RELOAD_SCRIPT
```

RestartSecディレクティブの使用

障害が発生した後でサービスを再起動するまでの遅延時間を秒の単位で定義するには、`[SERVICE]`セクションでRestartSecディレクティブを使用します。リソースを解放するために指定の時間を必要とするサービスや、システムに対する過大な負荷の原因となりかねない急速な再起動ループを防止するサービスで、このディレクティブが効果的です。

```
[Service]
ExecStart=PATH_TO_EXECUTABLE
Restart=on-failure
RestartSec=5
```

リモートコンピュータでの緊急モードの無効化

リモートコンピュータでの緊急モードを無効化できます。このようなコンピュータとして、Google Cloudでホストされている仮想マシンなどがあります。このモードが有効な場合、そのコンピュータはネットワークへの接続をブロックされます。次に例を示します。

```
> sudo systemctl mask emergency.service
```

```
> sudo systemctl mask emergency.target
```

12 法的事項

Copyright © 2006–2025 SUSE LLC and contributors. All rights reserved.

この文書は、GNUフリー文書ライセンスのバージョン1.2または(オプションとして)バージョン1.3の条項に従って、複製、頒布、および/または改変が許可されています。ただし、この著作権表示およびライセンスは変更せずに記載すること。ライセンスバージョン1.2のコピーは、「GNUフリー文書ライセンス」セクションに含まれています。

SUSEの商標については、<https://www.suse.com/company/legal/>を参照してください。その他の第三者のすべての商標は、各社の所有に帰属します。商標記号(®、™など)は、SUSEおよび関連会社の商標を示します。アスタリスク(*)は、第三者の商標を示します。

本書のすべての情報は、細心の注意を払って編集されています。しかし、このことは正確性を完全に保証するものではありません。SUSE LLC、その関係者、著者、翻訳者のいずれも誤りまたはその結果に対して一切責任を負いかねます。

A GNUフリー文書利用許諾契約書(GFDL)

Copyright (C) 2000, 2001, 2002 Free Software Foundation, Inc. 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA. この使用許諾書を一字一句そのままの複製および頒布することは許可されますが、変更は許可されません。

0. 序文

この利用許諾契約書の目的は、マニュアル、テキストブック、またはその他の機能的で有用な文書を、自由という意味で「フリー」にすることです。つまり、そのような文書を、変更の有無や商用非商用に関わらず、コピーまたは再配布する実効的な自由をすべての人々に保証することです。第二に、本利用許諾契約書は、作者または発行者が他者によって行われた変更について責任を負わないとともに、その著作物の功績が確保されるように意図されています。

本利用許諾契約書は、「コピーレフト」(著作物を自由に複製および改変できるようにすること)の一種であり、文書の派生著作物は、それ自体が同じ意味においてフリーでなければなりません。フリーソフトウェア向けに考慮されたコピーレフト利用許諾であるGNU一般公衆利用許諾契約書(GPL)を補足するものです。

弊社は、この利用許諾契約書をフリーソフトウェアのマニュアルに使用するために設計しました。それは、フリーソフトウェアにはフリーマニュアルが必要であるためです。つまり、フリープログラムには、そのソフトウェアと同じ自由を提供するマニュアルが付属しなければなりません。ただし、本利用許諾契約書は、ソフトウェアマニュアルに制限されるものではありません。主題であるか否か、または印刷された本として発行されるか否かに関わらず、任意のテキスト著作物に使用することができます。本利用許諾契約書は、その目的が指示または参照に置かれている著作物に主に使用することを推奨します。

1. 適用範囲と定義

本利用許諾契約書は、この利用許諾の条項に従って頒布できることを定めた著作権者の通告が記載されている任意のメディアにおけるマニュアルまたは他の著作物に適用されます。そのような通告は、その著作物をここに記載されている条件に従って使用するための世界的な無償の利用許諾を無期限で付与します。次に示す「文書」は、そのような任意のマニュアルまたは著作物を指します。その公衆ユーザはいずれも被許諾者であり、「利用者」と呼ばれます。利用者は、著作権法に従った許可が必要になるような方法で著作物を複製、変更または頒布する場合に、利用許諾を受け入れます。

文書の「変更された版」とは、そのまま複製されるか、変更または別の言語に翻訳された(またはその両方)文書あるいはその一部を含んだ著作物のことです。

「二次セクション」は、文書の発行者または作者と文書の全体的な主題(または関連事項)との関係のみを示す文書の名前付き付録または前付け部分です。総体的な主題に直接関わる内容は含まれていません。(したがって、文書が部分的に数学のテキストブックになっている場合、二次セクションでは数学について説明されない場合があります)。関係には、主題または関連事項との歴史的なつながり、あるいはそれらに関する法的、商的、哲学的、倫理的、政治的位置付けが含まれる場合があります。

「不変セクション」は、文書が本利用許諾契約書の条件の下でリリースされる旨を述べている通告において、そのタイトルが不変セクションのものとして指定されている、ある特定の二次セクションです。セクションが、すでに説明した二次セクションの定義に一致しない場合は、不変として指定することはできません。文書には、不変セクションが含まれない場合があります。文書で不変セクションを特定しない場合、不変セクションは含まれません。

「カバーテキスト」とは、文書が本利用許諾契約書の条件の下でリリースされる旨を述べている通告において、表カバーテキストまたは裏カバーテキストとして列挙されている、ある一定の短い文章のことです。表カバーテキストは、最大で5語、裏カバーテキストは、最大で25語によって構成できます。

文書の「透過的な複製」とは、その仕様が一般の利用者にとって入手可能で、一般的なテキストエディタまたは一般的な描画プログラム(画素で構成される画像用)、あるいは広く使用されている図面エディタ(図面用)で文書を直接改訂するのに適した形式で表される機械可読の複製のことです。テキストフォーマッタへの入力またはテキストフォーマッタへの入力に適したさまざまな形式への変換に適していることも前提になります。読者による以後の変更を阻止または妨げるようにマークアップまたはマークアップのない状態が調整されている、他の点では透過的なファイル形式で行われた複製は、透過的な複製ではありません。イメージ形式は、相当量のテキストに使用されている場合、透過的ではありません。「透過的」ではない複製は、「不透明」と呼ばれます。

透過的な複製に適した形式として、マークアップのないプレーンなASCII、Texinfo入力形式、LaTeX入力形式、一般に取得可能なDTDを使用するSGMLまたはXML、標準に準拠したHTML、人為的変更用のPostScriptまたはPDFがあります。透過的なイメージ形式には、PNG、XCF、JPGが含まれます。不透明な形式には、独自のワードプロセッサのみで読み取りおよび編集を行える独自の形式、DTDまたは処理(またはその両方)ツールを一般に取得できないSGMLまたはXML、機械生成HTML、出力のみを目的として一部のワードプロセッサによって作成されるPostScriptまたはPDFが含まれます。

「タイトルページ」とは、印刷された本の場合、タイトルページ自体、および本利用許諾契約書でタイトルページに表示することが要求される材料を読みやすいように保持するために必要な以降のページのことを指します。そのようなタイトルページがない形式の著作物の場合、「タイトルページ」は、本文の開始部分に先行する、著作物のタイトルを最も顕著に表している部分の近くにあるテキストのことを指します。

「XYZという表題の付いた」セクションとは、そのタイトルが正確にXYZになっているか、またはXYZを別の言語に翻訳しているテキストに続いてカッコ付きのXYZが含まれている文書の名前付きサブユニットのことです。(ここで、XYZは、次に示すように、「謝辞」、「献辞」、「推薦」、「履歴」などの特定のセクション名を表します)。文書を変更するときに、そのようなセクションの「タイトルを保存する」とは、この定義に従って「XYZという表題の付いた」セクションが残されることを表します。

文書では、本利用許諾契約書が文書に適用される旨を述べている通告の付近に保証の放棄を含めることができます。保証の放棄条項は、本利用許諾契約書内の参照によって、保証の放棄に関してのみ組み込まれると見なされます。つまり、これらの保証の放棄条項がもつ可能性のある他のいかなる含意も無効であり、本利用許諾契約書の意味にまったく影響を与えません。

2. そのままの複製

利用者は、商用か否かを問わず、任意のメディアにおいて文書を複製または頒布することができます。その際に、本利用許諾契約書、著作権表示、および本利用許諾契約書が文書に適用される旨を述べる利用許諾通告をすべての複製で再生し、本利用許諾契約書の条件に他のいかなる条件も追加しないことが前提条件になります。利用者は、技術的手段によって、作成または頒布する複製の読み込みまたはさらなる複製を妨げたり、制御したりすることはできません。ただし、複製と引き換えに対価を受け取ることができます。十分に大量の複製を頒布する場合は、セクション3の条件に従う必要もあります。

すでに述べた同じ条件に従って複製を貸与したり、複製を公開したりすることもできます。

3. 大量の複製

発行する文書の印刷した複製(または、通常、印刷したカバーをもつメディアに含まれた複製)が100部を超え、文書の利用許諾通告でカバーテキストを必要とする場合は、すべてのカバーテキスト(表カバーの表カバーテキスト、裏カバーの裏カバーテキスト)を明瞭かつ読みやすく記載したカバーに文書の複製を同封する必要があります。また、両方のカバーでは、これらの複製の発行者として、利用者を読みやすい状態で明確に識別しなければなりません。表カバーには、フルタイトルを記述し、タイトルのすべての語が同等に目立つようにする必要があります。カバーには他のマテリアルを追加することもできます。カバーに限って変更を行った場合の複製は、文書のタイトルが保持されていて、これらの条件を満たしている限り、他の点に関してそのままの複製と見なすことができます。

いずれかのカバーで、必要なテキストが多すぎて、読みやすい状態に収まらない場合は、列挙されている最初の部分(問題なく収まる分)を実際のカバーに記載し、残りの部分を隣接ページに入れます。

文書の不透明な複製を100部以上公開または頒布する場合は、それぞれの不透明な複製とともに機械可読の透過的な複製を含めるか、それぞれの不透明な複製内あるいはその複製とともに、ネットワークの一般利用者が標準的な一般ネットワークプロトコルを使用して、追加マテリアルのない文書の完全な透過的複製をダウンロードするときにアクセスできるコンピュータネットワークの場所を明記する必要があります。後者のオプションを使用する場合は、不透明な複製の大量頒布を開始するときに十分慎重な手順を取り、この透過的な複製が、その版の不透明な複製を最後に一般頒布した後(直接またはエージェントや小売業者を通じて)少なくとも1年間、指定した場所で継続的にアクセス可能となるように配慮する必要があります。

大量の複製を再頒布する時点よりもかなり前に、文書の作者に連絡して、文書の更新版を提供する機会を与えることが要求されますが、必須ではありません。

4. 変更

文書の変更された版を、すでに述べた第2項および第3項の条件に従って複製および頒布することができます。その際は、本利用許諾契約書に確実に従って、変更された版をリリースし、変更された版が文書の役割を担うようにして、その複製を所要する任意の利用者に変更された版の頒布および変更の利用許諾を与えることが前提になります。また、変更された版で次のことを行う必要があります。

- A. タイトルページ(カバーがある場合はカバー上も含める)で、文書、および以前の版の文書(以前の版がある場合は、その旨、文書の履歴セクションに列挙する)と識別されるタイトルを使用します。前の版と同じタイトルは、その版の元の発行者が許可を与えた場合に、使用することができます。
- B. タイトルページ上に、この要件から解放されない限り、変更された版において変更の著者としての責任を担う1人以上の人またはエンティティとともに、文書の筆頭著者を少なくとも5人、作者として列挙します(5人に満たない場合は、その筆頭著者のすべて)。
- C. タイトルページ上に、変更された版の発行者の名前を、発行者として記載します。
- D. 文書のすべての著作権表示を保持します。
- E. 変更に関する適切な著作権表示を、他の著作権表示の隣に追加します。
- F. 著作権表示の直後に、本利用許諾契約書の条項に従って変更された版を利用するための許可を一般利用者にする利用許諾通告を、次の補遺に示す形式で含めます。
- G. その利用許諾通告に、不変セクションの全リスト、および文書の利用許諾通告で指定されている必須カバーテキストを保持します。
- H. 本利用許諾契約書の変更されていない複製を含めます。

- I. 「履歴」という表題のセクションを保持して、そのタイトルを保持し、タイトルページに記載されているとおりに、変更された版のタイトル、年度、新しい作者、発行者を少なくとも示す項目を追加します。文書に履歴というセクションがない場合は、そのタイトルページに記載されているとおりに文書のタイトル、年度、作者、発行者を示すセクションを作成し、前の文章に記載されているとおりに変更された版を示す項目を追加します。
- J. 文書の透過的な複製に一般利用者がアクセスできるように文書で指定されている場合は、そのネットワークの場所、およびその文書の基盤となった前の版に対応して文書で指定されているネットワークの場所を保持します。これらは、「履歴」セクションに配置することができます。文書自体よりも4年以上前に発行された著作物の場合、または参照されているその版の元の発行者が許可を与えている場合は、そのネットワークの場所を省略することができます。
- K. 「謝辞」または「献辞」という表題のセクションの場合は、そのセクションのタイトルを保持し、セクション内に、それぞれの貢献者謝辞またはその中の献辞(またはその両方)のすべての内容と意味合いを保持します。
- L. 文書のすべての不変セクションを保持し、そのテキストおよびタイトルを未変更のままにします。セクション番号またはそれと同等の要素は、セクションタイトルの一部と見なされません。
- M. 「推薦」という表題の任意のセクションを削除します。そのようなセクションは、変更された版に含めることはできません。
- N. 既存のセクションのタイトルを変更して、「推薦」という表題にしたり、タイトルが不変セクションと矛盾したりしないようにします。
- O. 保証の放棄を保持します。

変更された版に、二次セクションと見なされ、文書から複製された材料を含まない新しい前付けセクションまたは付録が含まれる場合は、これらの一部またはすべてを任意に「不変」として指定することができます。これを行うには、変更された版の利用許諾表示内で列挙されている不変セクションにそのタイトルを追加します。これらのタイトルは、他のすべてのセクションタイトルと異なっている必要があります。

「推薦」という表題のセクションを追加することができますが、その際は、変更された版のさまざまな当事者による推薦以外の要素が含まれていないことが前提になります。たとえば、校正者によるコメント、または文が標準的な信頼できる定義として組織によって承認されていることを示すという宣言文などが相当します。

表カバーテキストとしての最大5語の短い文、および裏カバーテキストとしての最大25語の短い文を、変更された版のカバーテキストのリストの終わりに追加できます。表カバーテキストの短い1文および裏カバーテキストの短い1文のみを、1つのエンティティが追加できます(また

はエンティティによって行われた調整を通じて)。文書に、利用者または利用者が関わる同じエンティティによって行われた調整を通じて前に追加された同じカバーのカバーテキストがすでに含まれている場合は、別のものを追加することはできませんが、古いものを置き換えることができます。ただし、その古いものを追加した前の発行者から明示的な許可を得る必要があります。

文書の作者および発行者は、本利用許諾契約書により、その名前を得るために使用したり、変更された版の推薦を主張または暗示したりする許可を与えるものではありません。

5. 文書の結合

文書は、すでに述べた変更された版に関するセクション4の条件に従って、本利用許諾契約書の下でリリースされた他の文書と結合することができます。その際は、その組み合わせの中に、元の全文書のすべての不変セクションを未変更のまま含めて、そのすべてを結合された著作物の不変セクションとしてその利用許諾表示に列挙し、そのすべての保証の放棄を保持することが前提となります。

結合された著作物には、本利用許諾契約書の複製を1つのみ含める必要があります。複数の同一の不変セクションは、単一の複製で置き換えることができます。同じ名前だが内容の異なる複数の不変セクションがある場合は、そのような各セクションのタイトルを固有なものにします。その際は、その終わりに、カッコ付きで、そのセクションの元の作者または発行者の名前(既知の場合)、あるいは固有の番号を追加します。不変セクションのリスト内のセクションタイトルには、結合された著作物の利用許諾表示の場合と同じ調整を加えます。

組み合わせでは、さまざまな元の文書の「履歴」という表題のセクションを結合して、1つの「履歴」というセクションを構築する必要があります。同じように、「謝辞」という表題のセクション、および「献辞」という表題のセクションも結合します。「推薦」という表題のすべてのセクションを削除する必要があります。

6. 文書のコレクション

文書および本利用許諾契約書の下でリリースされた他の文書から成るコレクションを作成して、さまざまな文書に含まれる本利用許諾契約書の個々の複製を、コレクションに含まれる単一の複製で置き換えることができますが、他のすべての点での各文書のそのままの複製に関する本利用許諾契約書の規則に従うことが前提になります。

そのようなコレクションから単一の文書を抽出して、その文書を本利用許諾契約書に従って個々に頒布することができますが、その際は、本利用許諾契約書の複製を抽出した文書に挿入して、その文書のそのままの複製に関するその他のすべての点で、本利用許諾契約書に従う必要があります。

7. 独立した著作物の集積

文書またはその派生物を他の個別および独立した文書または著作物とともに、ストレージまたは頒布メディア内またはそのボリューム上に蓄積することを「集積」と呼びます。その場合は、個々の著作物の許可を超えてその蓄積の利用者の法的権限を制限することに、蓄積による著作権を使用しないことが前提になります。文書が集積に含まれる場合、本利用許諾契約書は、それ自体が文書の派生著作物ではない集積内の他の著作物に適用されません。

セクション3のカバーテキスト要件が文書のこれらの複製に適用可能であり、文書が集積全体の半分に満たない場合は、文書のカバーテキストを、集積内の文書のカバー、または文書が電子形式の場合は、電子的な同等のカバーに配置することができます。それ以外の場合は、集積全体の印刷されたカバー上に表示する必要があります。

8. 翻訳

翻訳は一種の変更と見なされるため、セクション4の条件に従って文書の翻訳を配布することができます。不変セクションを翻訳で置き換えるには、著作権者からの特別な許可が必要ですが、これらの不変セクションの元のバージョンのほかに、一部またはすべての不変セクションの翻訳を含めることができます。本利用許諾契約書、文書内のすべての利用許諾表示、および保証の放棄の翻訳を含めることができますが、その際は、本利用許諾契約書の元の英語版、およびそれらの利用許諾表示と保証の放棄の元の版も含めることが前提になります。本利用許諾契約書、利用許諾表示、または保証の放棄の翻訳と元の版との間に不一致がある場合は、元の版が優先されます。

文書内のセクションに、「謝辞」、「献辞」、または「履歴」という表題が付いている場合は、そのタイトルを保持する(セクション1)ための要件(セクション4)により、通常は実際のタイトルを変更する必要があります。

9. 終了

利用者は、本利用許諾契約書に明示的に記載されている形態を除き、文書を複製、改変、二次利用許諾、および頒布してはなりません。別の方法で文書を複製、改変、二次利用許諾、または頒布しようとするのは無効であり、本利用許諾契約書の下で利用者の権利は自動的に消滅します。ただし、本使用許諾の下で利用者から複製物または権利を受領した関係者は、条項を遵守している限り、権利が消滅することはありません。

10. 本利用許諾契約書の将来の改訂

フリーソフトウェア財団は、GNUフリー文書利用許諾契約書(GFDL)の新版または改訂版を随時公表することがあります。そのような新版は、性格的には現行版と似たものになりますが、新たな問題や懸案事項に対応するために細部が異なる可能性があります。<https://www.gnu.org/copyleft/>を参照してください。

本利用許諾契約書の各版には、区別するための版番号が設定されます。文書に、それに適用される本利用許諾契約書の特定の版番号と「後継版」が指定されている場合、利用者は、選択によって、その指定された版の条項またはフリーソフトウェア財団から公開される後継版の条項(ドラフトではない)に従うことになります。文書に、本利用許諾契約書の版番号が指定されていない場合、利用者は、フリーソフトウェア財団からこれまでに公開された任意の版(ドラフトではない)を選択することができます。

補遺: 本利用許諾書をご使用の文書に使用する方法

```
Copyright (c) YEAR YOUR NAME.  
Permission is granted to copy, distribute and/or modify this document  
under the terms of the GNU Free Documentation License, Version 1.2  
or any later version published by the Free Software Foundation;  
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.  
A copy of the license is included in the section entitled "GNU  
Free Documentation License".
```

不変セクション、表カバーテキスト、および裏カバーテキストがある場合は、「with...Texts」の行を次のように置き換えます。

```
with the Invariant Sections being LIST THEIR TITLES, with the  
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

カバーテキストのない不変セクションが含まれている場合、またはこの3つの他の組み合わせの場合は、その2つの代替要素をマージして状況に合わせます。

文書にプログラムコードの重要な例が含まれている場合は、GNU一般公衆利用許諾契約書(GPL)などの選択したフリーソフトウェアの利用許諾に従って、これらの例を平行してリリースし、フリーソフトウェアでのその利用を許可することを推奨します。