

IT 資格 道 場

Linux Essentials 010-160 学習用テキスト

予備知識ゼロから合格ラインへ —— 読んで理解する1冊目

無料サンプル(第1章まで)

LPI「Linux Essentials Exam 010 Objectives」Version 1.6(試験コード 010-160・LPI
公式・2026-10-13 取得)準拠
2026年10月版

ご利用にあたって

- このテキストは「IT資格道場」(www.shikaku-doujou.com)の購入者限定テキストのうち、学習用テキストの第1章までを収録した**無料サンプル**です。どなたでもご覧いただけます。
- 無料サンプルも著作物です。転載・改変しての再配布・二次販売はできません。
- AI モデルの学習データとしてのご利用(学習目的の入力・データセット化を含む)はできません。
- 製品版(学習用テキスト・頻出度別ノート・暗記用ノートの3点)は、ご購入者を識別できる透かし入りのPDFとして提供されます。
- 本PDFはスマートフォン・タブレットでの閲覧に合わせたA5版面・文字サイズで組版しています。

目次

この教材の使い方

試験の基本情報

このテキストの構成(全 5 章)

第1章 The Linux Community and a Career in Open Source [1.1/1.2/1.3/1.4]

Linux の成り立ちと主なディストリビューション [1.1]

主なオープンソースのアプリケーションとパッケージ管理 [1.2]

オープンソースの考え方とライセンス [1.3]

ICT スキルと Linux での作業 [1.4]

サンプルはここまでです

3冊(学習用・頻出度別・暗記用)の使い方は、ダウンロード欄の案内を参照してください。

SAMPLE

この教材の使い方

このテキストの目的は、Linux Professional Institute(LPI)が実施する **Linux Essentials(試験コード 010-160、現行の版は 1.6)** への合格。

Linux Essentials は、Linux を初めて学ぶ人が最初に受ける資格です。受験資格は不要で、一度合格すれば有効期間もありません。

出題の柱は 4 つ。

- 1. Linux の周辺知識(ディストリビューション、ライセンス、仕事での使われ方)
- 2. コマンドラインの基本操作
- 3. OS の中身
- 4. ユーザーと権限

この試験の性格を一言で言えば、**コマンドと出力を「読んで判断する」ことを問う試験**。`ls -l` の 1 行や `/etc/passwd` の 1 行が示され、何が起きているかを答える形。そこで本文は、コマンドの説明に加えて出力を載せ、読み方を示します。手元の端末で打って、自分の目で出力を読んでください。

試験の基本情報

項目	内容
試験名	Linux Essentials(010-160 v1.6)
運営	Linux Professional Institute(LPI)
問題数・時間	40 問・60 分
出題形式	選択問題と記述問題(端末に打つ文字列を入力)

項目	内容
合格点	公式の試験ページには記載なし
受験料	13,000 円・税抜(日本。2026 年 10 月時点)
試験言語	日本語を含む 6 言語
受験資格・有効期間	受験資格なし・有効期間なし

版の予定は次のとおり。

- LPI は 2026 年 10 月に Version 2.0 を公開する予定と告知している(細目の番号は 011.1～015.4 に変わる)
- 1.6 は 2027 年 4 月に終了する予定[^c0-1]
- 19 の細目の配点は、1.6 と 2.0 で同じ

試験の版や受験料は変わることがあります。申し込み前に、公式サイトで最新情報を確認してください。

このテキストの構成(全 5 章)

試験は 5 つの主題、19 の細目で構成され、配点の合計は 40 です。章はこの主題の順に並べています。

章	主題	細目	配点
第1章	The Linux Community and a Career in Open Source	1.1～1.4	7
第2章	Finding Your Way on a Linux System	2.1～2.4	9

章	主題	細目	配点
第3章	The Power of the Command Line	3.1～3.3	9
第4章	The Linux Operating System	4.1～4.4	8
第5章	Security and File Permissions	5.1～5.4	7

合計は 40 点分。

第1章 The Linux Community and a Career in Open Source [1.1/1.2/1.3/1.4]

Linux を取り巻く世界の地図を作る章です。ディストリビューションの種類、動くアプリ、配られるときのライセンス、仕事での使われ方を順に見ます。コマンドはほとんど出てきません。名前と役割と違いを結び付けて覚えます。

Linux の成り立ちと主なディストリビューション [1.1]

Linux は、厳密にはオペレーティングシステム (OS) の中心部分であるカーネルの名前です。カーネルは CPU・メモリ・ディスクなどのハードウェアを管理し、アプリに配る役目を持ちます。Unix の考え方を写し、Linus Torvalds がーから書きました。ライセンスは GPL-2.0 only で、システムコールの境目には例外が付いています。この例外のおかげで、Linux 上で動くプログラムは GPL にする必要があります。[^c1-1]

カーネルだけでは使えません。シェルやコマンド群 (多くは GNU プロジェクトの部品)、ライブラリ、パッケージ管理、インストーラを束ねて配るものが**ディストリビューション**です。[^c1-2] 同じカーネルを使っていても、束ね方が違えば別のディストリビューションになります。

カーネルとディストリビューションは、別の階層です。カーネルを新しい版に入れ替えても、ディストリビューションの名前や系統は変わりません。逆に、Ubuntu と Debian は別のディストリビューションですが、どちらも Linux カーネルを使います。「Linux を入れた」という言い方は、実際にはディストリビューションを選んで入れたことを指します。

系統の見分け方

ディストリビューションは、土台にしたものとパッケージの形式で系統が分かります。この表の見方を覚えないと、場面問題が解けません。

系統	代表	パッケージ形式	主なパッケージ道具
Debian 系	Debian・Ubuntu・Linux Mint・Raspberry Pi OS (旧 Raspbian)	.deb	dpkg・apt-get
Red Hat 系	Red Hat Enterprise Linux (RHEL)・CentOS・Fedora・Scientific Linux	.rpm	rpm・yum (dnf)
SUSE 系	SUSE Linux Enterprise・openSUSE	.rpm	rpm・zypper
独立系	Arch Linux	独自	pacman

SUSE 系も rpm 形式ですが、依存を解決する道具は yum ではなく zypper です。Arch の pacman と合わせて、名前を知っていれば、細かい使い方まで覚える必要はありません。

名前ごとの要点

名前	成り立ち	要点
Debian	共同体(コミュニティ)が作る老舗。Ubuntu や Linux Mint の土台	リリースは stable・testing・unstable の 3 段で、unstable の別名は sid。新しいパッケージはまず unstable に入り、testing を経て、次の stable になる。サーバーで使うのは stable。保守は 5 年で、前半 3 年が通常、後半 2 年が長期サポート (LTS) [^c1-3]
Ubuntu	Debian をもとに Canonical 社が開発	6 か月ごとに新版が出る。そのうち 2 年ごと (偶数年の 4 月) に出る版が LTS (Long Term Support) で、標準の保守は 5 年。24.04 LTS (2024 年 4 月) の標準の保守は 2029 年 5 月まで。LTS でない版 (interim release) の保守は 9 か月。本番のサーバーは LTS を選ぶ[^c1-4]
Linux Mint	Ubuntu (と Debian) をもとにしたデスクトップ向け	モダンで使いやすいデスクトップを掲げる[^c1-5]
Red Hat Enterprise Linux (RHEL)	Red Hat 社の有償の製品	サブスクリプションを買い、サポートと更新を受けられる。主な版ごとに 10 年の保守期間があり、Full Support・Maintenance Support・Extended Life の段に分かれる[^c1-6]

名前	成り立ち	要点
Fedora	Red Hat が支える共同体の版	RHEL に入る新機能の先行役。新版は約 6 か月ごと、各版の保守は約 13 か月[^c1-7]
CentOS	RHEL のソースから再構築した無償の互換版 (CentOS Linux) だった	CentOS Linux 8 は 2021 年 12 月 31 日に、CentOS Linux 7 は 2024 年 6 月 30 日に保守を終えた[^c1-8]。今の CentOS Stream は RHEL の開発のすぐ先を追う版で、RHEL の再構築ではない[^c1-9]
Scientific Linux	米国 Fermilab が支えた RHEL の再構築版 (Red Hat 系)	2024 年 6 月 30 日に終了[^c1-10]
SUSE	ドイツ発の企業向け製品 SUSE Linux Enterprise を出す会社の名前でもある	有償の長期保守の製品
openSUSE	SUSE の共同体の版	2 本ある。Leap は SUSE Linux Enterprise と同じソースとバイナリを使う安定版。Tumbleweed は常に最新を取り込むローリングリリース[^c1-11]
Arch Linux	独立系	ローリングリリースで、簡素さ (simplicity) を掲げ、余計な追加や改変をしない。版という区切りがなく、更新を続けるだけで常に最新になる[^c1-12]

Debian の 3 段は場面で効きます。本番のサーバーに入れるのは、動作が確かめられた stable です。testing は次の stable の候補で、unstable (sid) は開発の最前線です。新しいパッケージを試すだけなら unstable ですが、止められないサーバーには向きません。

「CentOS は RHEL と同じ中身の無償版」という昔の説明は、試験の選択肢に残っていても今は成り立ちません。CentOS Linux はもう更新が出ないからです。Debian・Ubuntu・Linux Mint が同じ系統で、RHEL・CentOS・Fedora・Scientific Linux が同じ系統、と整理して覚えます。

保守期間の読み方

保守期間を知らないと、どの版を選ぶかの場面問題には答えられません。

種類	例	特徴
長期サポート (LTS)	Ubuntu LTS・Debian stable	数年間、セキュリティ修正が出る。サーバー向け
商用の長期保守	RHEL (10 年)・SUSE Linux Enterprise	有償契約で長期のサポート
短期の版	Ubuntu の interim (9 か月)・Fedora (約 13 か月)	新機能が早い。更新の手間が多い
ローリングリリース	Arch Linux・openSUSE Tumbleweed	版を持たず、常に最新へ更新を重ねる

数年間は動かし続けたい業務サーバーなら、LTS か RHEL を選びます。新しい技術をいち早く試したい個人の端末なら、Fedora やローリングリリースが合います。更新のたびに全体を入れ直さなくてよいのは、ローリングリリースの利点です。ただし常に最新なので、動作が変わる更新が突然入ることも少なくありません。

具体的な場面に当てはめます。5 年間は保守され、本番で止められない Web サーバーを建てたいなら、Ubuntu の LTS か Debian の stable が候補です。サポート窓口に問い合わせる契約で責任を持ってもらいたいなら、RHEL か SUSE Linux Enterprise です。CentOS Linux 7 を使っていた環境は、保守が切れているので別の物へ移します。移す先は有償の RHEL でも、無償の CentOS Stream でも構いません。ただし CentOS Stream は RHEL の再構築ではないので、同じ版の挙動を保証する関係にはなりません。

組み込みシステムとモバイル

組み込みシステム (Embedded Systems) は、機器の中に埋め込まれた専用のコンピュータです。ルーター、テレビ、家電、車載機器などが当たります。こうした機器は、パソコンのような大きなディスクを前提にできません。部品を削って小さくでき、ソースも手に入る Linux は、その条件に合います。

Raspberry Pi は名刺より少し大きい程度の小型コンピュータで、専用の OS が **Raspberry Pi OS** (旧名 **Raspbian**) です。Debian を土台にした OS なので、パッケージは Debian 系の apt で扱います。^[^c1-13] OS のイメージを SD カードに書き込み、そのカードから起動する使い方が一般的です。試験で「Raspbian はどの系統か」と問われたときの答えは「Debian 系」。

Android は、Linux カーネルを土台にした携帯端末向けの OS で、Android 用に手を加えた Android Common Kernel (ACK) を使います。^[^c1-14] カーネルが Linux なので、Android を「Linux の仲間」と数えます。ただし Debian 系や Red Hat 系のような、ディストリビューションの系統には入りません。

クラウドの Linux

クラウドは、インターネット越しに計算資源を借りて使う方式です。借りる範囲によって、サービスの形が 3 つに分かれます。^[^c1-15]

モデル	借りるもの	利用者が管理する 範囲	例の場面
IaaS (Infrastructure as a Service)	仮想マシン・ストレージ・ネットワークなどの基盤	OS・ミドルウェア・アプリ	Linux の仮想マシンを借りて Web サーバーを入れる
PaaS (Platform as a Service)	アプリを動かす実行環境	自分のアプリと設定	自作のプログラムを置くだけで動かす
SaaS (Software as a Service)	完成したアプリ	ほぼ無し (限られた設定のみ)	ブラウザで使う Web メール

IaaS では OS の選択と管理が利用者の仕事なので、Linux の知識が直接必要です。管理の範囲は IaaS が最も広く、SaaS が最も狭くなります。借りる側の自由度も同じ順に大きく、SaaS では OS を選べません。

場面で見分けます。仮想マシンに Linux を入れて、自分で Web サーバーを設定するなら IaaS です。自作のプログラムを渡すだけで OS の面倒は見ないなら PaaS で、ブラウザで使うメールや文書作成のサービスは SaaS です。OS の更新 (パッチ当て) が利用者の仕事に残るのは、IaaS だけです。

クラウドの置き方 (配置モデル) の 4 つの名前:

- プライベート: 単独の組織だけが使う
- パブリック: 一般に開かれている
- コミュニティ: 共通の関心を持つ特定の組織の集まりが使う
- ハイブリッド: 2 つ以上を組み合わせる

サービスモデル (IaaS・PaaS・SaaS) と配置モデルは別の軸で、組み合わせて言えます。たとえば「パブリッククラウドの IaaS」です。

間違いやすい点

- Ubuntu と Linux Mint は Debian 系です。rpm と yum は使いません。
- CentOS Linux はもう更新が出ません。CentOS Stream は RHEL の再構築ではありません。
- Raspbian は Raspberry Pi OS の旧名で、Debian ベースです
- Android は Linux カーネルを使います。Debian 系や Red Hat 系の仲間ではありません。
- Fedora は先行版で、RHEL そのものではありません

主なオープンソースのアプリケーションとパッケージ管理 [1.2]

アプリは、使う場所で 3 つに分けて覚えます。デスクトップ、サーバー、開発言語です。そのあとで、どのシステムの道具でインストールするかを見ます。

デスクトップのアプリ

アプリ	役割	覚えること
LibreOffice	オフィススイート	文書の Writer・表計算の Calc・発表の Impress・図形の Draw・データベースの Base・数式の Math。標準の保存形式は ODF (.odt・ods・.odp)
OpenOffice.org	オフィススイート (LibreOffice の元)	LibreOffice は 2010 年に OpenOffice.org から分かれた
Thunderbird	メールクライアント	Mozilla 系
Firefox	Web ブラウザ	Mozilla 系

アプリ	役割	覚えること
GIMP	ラスター画像(写真など)の編集	画素の集まりを加工する
Inkscape	ベクター画像(SVG)の編集	線と面の数式で描く

LibreOffice は、Oracle 社の下に入った OpenOffice.org の開発者たちが 2010 年 9 月に The Document Foundation を作って始めました。[^c1-16] 名前が似ていても、いま積極的に開発されているのは LibreOffice の側です。

GIMP と Inkscape は取り違いやすい組です。写真を修整するなら GIMP、拡大しても荒れないロゴや図を描くなら Inkscape です。

サーバーのアプリ

サーバーのアプリ(ネットワーク越しにほかの機械へサービスを出すもの):

アプリ	役割	覚えること
Apache HTTPD	Web サーバー	既定で Listen 80(HTTP)。HTTPS は 443 番[^c1-17]
NGINX	Web サーバー	リバースプロキシ・負荷分散にも使う
MariaDB	リレーショナルデータベース	MySQL の開発者たちが作った、MySQL の互換版(フォーク) [^c1-18]
MySQL	リレーショナルデータベース	MariaDB の元
NFS	ファイル共有	Unix 系どうしの共有。NFSv4 は RFC 7530 で定められている[^c1-46]

アプリ	役割	覚えること
Samba	ファイル・プリンタ共有	SMB/CIFS で Windows と共有する[^c1-19]
Nextcloud・ownCloud	オンラインストレージ	自前のサーバーに置く、ファイルの同期と共有。 ownCloud から分かれたのが Nextcloud
Postfix・Dovecot	メール	Postfix がメールの転送 (MTA)、Dovecot が IMAP/POP3 の受け口

NFS と Samba の使い分けは、最もよく問われる組です。Linux どうし (Unix 系どうし) なら NFS、Windows が混ざるなら Samba を選びます。Windows の端末から Linux のフォルダを開けない、という場面なら、足りないのは Samba です。Windows の標準の共有の仕組みは、NFS ではありません。

やりたいことから選ぶ対応:

やりたいこと	使うアプリ
Web ページを配信する	Apache HTTPD・NGINX
データを表の形で蓄える	MariaDB・MySQL
Linux どうしでフォルダを共有する	NFS
Windows とフォルダやプリンタを共有する	Samba
ファイルを自前のサーバーで同期・共有する	Nextcloud・ownCloud
メールを転送する・受け取る	Postfix・Dovecot

Web サーバーとデータベースは、同じ機械で組み合わせて使います。たとえば Apache か NGINX で Web ページを出し、その裏で MariaDB か MySQL

にデータを入れ、PHP で処理を書く構成が典型です。

開発言語

言語	実行の仕方・用途
C	コンパイルして実行ファイルを作る。Linux カーネル自身の言語 (Rust も一部で使える) [^c1-20]
Java	コンパイルしてから仮想マシンの上で動かし、業務システム向け
JavaScript	ブラウザで動く言語として知られる
Perl	インタプリタで動く。テキスト処理に強い
shell	コマンドを並べたスクリプト (bash など)
Python	インタプリタで動く [^c1-21]
PHP	インタプリタで動く。Web ページの生成に使う

Python・Perl・PHP・シェルは、**インタプリタ**が 1 行ずつ読んで実行する言語で、事前にコンパイルしなくても動きます。C はコンパイラで機械語に変換してから実行するので、試験で「コンパイルが要る言語はどれか」と聞かれたら C が代表です。Java は、C のように機械語へ直接は変換しません。仮想マシン用の形式に変換して、仮想マシンの上で動かします。

パッケージとリポジトリ

パッケージは、プログラム本体・設定・「何に依存するか」という情報を 1 つにまとめた配布の単位です。**リポジトリ**は、パッケージを置いてある場所 (サーバー) です。パッケージ管理の道具は、この 2 つを使って、インストール・更新・削

除をします。形式は系統で違い、.deb を rpm で、.rpm を dpkg で入れることはできません。[^c1-22]

道具は 2 階層に分かれます。下の階層はパッケージ 1 個のファイルを直接扱い、上の階層はリポジトリから探して、依存関係ごと自動で入れます。

系統	下の階層 (ファイルを直接)	上の階層 (リポジトリ・依存を自動)
Debian 系	dpkg	apt-get (apt)
Red Hat 系	rpm	yum (dnf)

依存関係とは、「パッケージ A を動かすには B が必要」という関係です。dpkg や rpm は、足りない依存があると入れられません。apt-get や yum は、足りない依存も一緒に取ってきて入れます。

Debian 系: dpkg と apt-get
dpkg の主なオプション:

オプション	動き
-i	.deb ファイルを入れる
-r	消す (設定ファイルは残る)
-l	入っているパッケージの一覧
-L	そのパッケージが置いたファイルの一覧

`-i` は入れる相手が `.deb` ファイルです。パッケージ名を渡すのではありません。実際の出力は次のようになります (Debian 12。 `dpkg -l` の先頭に出る見出しの 5 行は省いています)。

```
$ dpkg -l bash
ii  bash                    5.2.15-2+b13 arm64      GNU Bourne Again Shell
$ dpkg -L bash | head -4
/.
/bin
/bin/bash
/etc
$ dpkg -S /bin/ls
coreutils: /bin/ls
```

`dpkg -l` の行頭の `ii` は「入っている」の意味です。`dpkg -S` は逆向きの問い合わせで、ファイルから所属するパッケージを調べます。

apt-get の主な動き:

動き	意味
update	リポジトリのパッケージ一覧を最新にする (パッケージ自体は更新しない)
upgrade	入っているパッケージを新しい版へ更新する
install	入れる。必要な依存も一緒に入る
remove	消す(設定ファイルは残る)
purge	設定ファイルごと消す
autoremove	不要になった依存を消す

間違えやすい点は `update` と `upgrade` の取り違いです。`update` が更新するのは「一覧」だけで、パッケージの中身は変わりません。「中身」を新しい版に

するのは upgrade です。通常は update のあとに upgrade か install を実行します。

リポジトリの設定は、`/etc/apt/sources.list` と `/etc/apt/sources.list.d/` の下に置きます。[^c1-23] Debian 12 のコンテナでは `/etc/apt/sources.list.d/debian.sources` に、URLs (取りに行く先) と Suites (bookworm など) が書かれています。

Red Hat 系: rpm と yum

rpm の主なオプション:

オプション	動き
-i	入れる
-U	更新(入っていなければ入れる)
-e	消す(erase)
-q	問い合わせ(パッケージ名を渡す)
-qa	入っている全パッケージ

実際の出力は次のようになります (Rocky Linux 9)。

```
$ rpm -q bash
bash-5.1.8-6.el9_1.aarch64
$ rpm -ql bash | head -3
/etc/skel/.bash_logout
/etc/skel/.bash_profile
/etc/skel/.bashrc
$ rpm -qf /bin/ls
coreutils-single-8.32-34.el9.aarch64
```

`-q` に名前を渡すと、入っているかどうか分かります。`-ql` は中身のファイル一覧で、`dpkg` の `-L` に当たります。`-qf` はファイルから所属パッケージを探し、`dpkg` の `-s` に当たります。

`yum` は、リポジトリから依存ごと入れる道具で、`yum install パッケージ名`、`yum remove パッケージ名`、`yum update` のように使います。RHEL 8 以降では、`yum` は `dnf` の別名にすぎません。Rocky Linux 9 で調べると、`yum` の実体は `/usr/bin/dnf-3` を指します。リポジトリの設定は、`/etc/yum.repos.d/*.repo` に置きます。[[^c1-24](#)]

削除と更新の違い

Debian 系では、`remove` と `purge` の違いが問われます。`remove` はプログラムを消しますが、設定ファイルは消しません。`purge` は設定ファイルまで消すので、まっさらに戻したいときに使います。`dpkg` の `-r` は `remove` と同じ動きで、設定ファイルを残します。

`rpm` では、`-i` と `-U` の違いが問われます。`-i` は新しく入れるだけで、更新はしません。`-U` は、入っていれば新しい版へ更新し、入っていなければ新しく入れます。手元の `.rpm` は、迷ったら `-U` で入れても支障ありません。

2 系統の対応表

目的	Debian 系	Red Hat 系
ファイルを直接入れる	dpkg -i	rpm -i (更新は -U)
入っているものの一覧	dpkg -l	rpm -qa
パッケージの中のファイル	dpkg -L 名前	rpm -ql 名前
ファイルが属すパッケージ	dpkg -S ファイル	rpm -qf ファイル
リポジトリから依存ごとに入れる	apt-get install	yum install
リポジトリの設定	/etc/apt/sources.list と sources.list.d/	/etc/yum.repos.d/

依存関係の扱いは、場面問題の中心です。dpkg や rpm は、必要な別のパッケージが足りないと、エラーを出して止まります。足りないものを自動で探して入れるのは、リポジトリを見に行く apt-get と yum (dnf) の仕事です。リポジトリが登録されていなければ、apt-get も yum も探しに行けません。

試験の場面問題は、たいてい次の形です。「Ubuntu のサーバーに Web サーバーを入れたい。依存パッケージも自動で入れたい」なら apt-get install です。dpkg -i では依存が解決されません。「Rocky Linux に、手元の .rpm を入れる」なら rpm -i か rpm -U です。逆に、Ubuntu で rpm、RHEL で apt-get を使う選択肢は成り立ちません。

オープンソースの考え方とライセンス [1.3]

ソフトウェアには、ソースコード (人が読める設計図) が付いて配られるものと、実行ファイルだけが配られるものがあります。この節では、ソースを公開する考え方と、使い方の条件を決めるライセンスを見ます。

Free Software と Open Source

Free Software (フリーソフトウェア) は、Richard Stallman が 1983 年に始めた GNU プロジェクトから生まれた考え方です。1985 年には **FSF** (Free Software Foundation) が作られました。[^c1-25] 「free」は無料でなく自由の意味で、利用者に次の 4 つの自由を保証するソフトウェアを指します。[^c1-26]

- 自由 0: 目的を問わず実行する自由
- 自由 1: 動作を調べ、必要に応じて改変する自由 (ソースコードの入手が前提)
- 自由 2: コピーを再配布する自由
- 自由 3: 改変した版を配布する自由

番号は 0 から始まります。自由 1 と自由 3 は、ソースコードが手に入らないと実現できません。無料で配られる実行ファイルだけのソフト (フリーウェア) は、ソースがないので自由 1 を満たさず、Free Software ではありません。

Open Source Software は、1998 年に作られた **OSI** (Open Source Initiative) が掲げる考え方です。[^c1-27] OSI は Open Source Definition (OSD) という 10 項目の定義を持っています。自由な再配布、ソースコードの提供、派生物の許可、個人や集団への差別の禁止、特定の製品に依存しないことなどが入っています。[^c1-28]

2 つの違いは立場にあります。Free Software は「利用者の自由」という倫理を、Open Source は「公開で開発すると品質が上がる」という実利を前面に出します。ただし、片方にしか当てはまらないソフトウェアばかりとは限りません。多くのソフトウェアは、両方の定義を満たしています。

両方をまとめた呼び名が 2 つあります。

- FOSS: Free and Open Source Software の略
- FLOSS: Free/Libre and Open Source Software の略 (libre は「自由」を表し、free が無料と取り違えられるのを避ける)

どちらも、片方の立場に偏らない言い方として使われます。[^c1-29]

団体	設立	掲げるもの
FSF (Free Software Foundation)	1985 年	Free Software・4 つの自由
OSI (Open Source Initiative)	1998 年	Open Source Software・OSD の 10 項目

ライセンスの分類: コピーレフトとパーミッシブ

ライセンスは、改変版や配布のときに何を求めるかで 2 系統に分かれます。

コピーレフト (Copyleft) は、改変版や派生物にも同じ自由 (同じライセンス) を求めます。自由が配布のたびに失われないように、鎖でつなぐ仕組みです。

パーミッシブ (Permissive) は条件が緩やかで、派生物を別の (閉じた) ライセンスで配ってもよいと認めます。

ライセンス	系統	要点
GPL (GPLv2・GPLv3)	強いコピーレフト	派生物を配るなら、同じ GPL でソースも渡す[^c1-30]
LGPL	弱いコピーレフト	ライブラリ向け。つなぐ側のプログラムは別のライセンスでもよい[^c1-31]

ライセンス	系統	要点
BSD (2 条項・3 条項)	パーミッシブ	著作権表示と無保証の表示を残す。3 条項は、名前を宣伝に使うことも禁じる[^c1-32]
MIT	パーミッシブ	著作権表示とライセンス文を残す。無保証[^c1-33]
Creative Commons	作品向け	文章・画像・音楽などの作品に使う。ソフトウェア向けではない[^c1-34]

Creative Commons の条件 (4 つの記号の組み合わせ):

記号	意味
BY (表示)	作者のクレジットを表示する
SA (継承)	改変した作品は、同じ条件で公開する (コピーレフトに近い)
NC (非営利)	営利目的の利用を認めない
ND (改変禁止)	改変や翻案を認めない

GPL の義務が動くのは、他人に配ったときです。GPL のソフトを入手して社内だけで使うなら、配布していないので、ソースを公開する義務は生じません。社外の顧客に改変版を渡すと、顧客にも同じ GPL でソースを渡す義務が生まれます。

ライセンス	改変版を配るとき、ソースの公開を求められるか	閉じた製品に組み込めるか
GPL	求められる(同じ GPL で)	組み込んだ製品を配るなら不可
LGPL	ライブラリ自身の改変部分は求められる	利用する側は可
BSD・MIT	求められない	可(表示は残す)

場面で選びます。自社の製品に取り込んで、ソースは公開したくないなら、BSD か MIT のコードを使います。GPL のコードを入れた製品は、配るなら GPL にしなければなりません。ライブラリだけを借りたいなら、LGPL のものを選べば、自社のプログラムは別のライセンスのままで済みます。BSD と MIT の違いは小さく、どちらも著作権表示を残す以外の制約を課しません。BSD の 3 条項は、そのうえで作者の名前を宣伝に使うことを認めません。

写真や文章を載せるときは、Creative Commons の条件を読みます。「改変しない」と「非営利に限る」ことは別の記号 (ND と NC) で、組み合わせて使えます。たとえば BY-NC-SA は、表示を守り、営利に使わず、改変したら同じ条件で公開する、という意味になります。

オープンソースのビジネスモデル

ソースが無償で手に入っても収益を得る方法があり、代表的なのは、ソフトウェア本体でなく周りのサービスを売ることです。[^c1-35]

- サポートと保守のサブスクリプション (RHEL のように、更新とサポートを契約で売る)
- 導入・設定・研修などのサービス
- 有償の拡張版を併売する (無償のコミュニティ版と、機能の多い企業版)

Red Hat が RHEL で行っているのが 1 つ目で、ソースは公開されたまま、お金は「使う権利」でなく「サポートや更新」に払われます。Linux が無償で使えるのに、企業向けの有償ディストリビューションが成り立つのは、この仕組みがあるからです。ソースの公開と収益は、矛盾しません。

ICT スキルと Linux での作業 [1.4]

Linux を使う人が身に付けておく基本的な ICT の技能として、ブラウザの使い方、コマンドラインへの入り方、パスワード、プライバシー、仕事での Linux の使われ方を扱います。

ブラウザとプライバシー

cookie は、Web サイトが端末に置く小さなデータです。ログイン状態を保つ役目がある一方で、サイトをまたいだ追跡広告にも使われます。cookie を無効にすると、サイトを開いたときの自動ログインが効きません。[^c1-36]

プライベートブラウジングは、閉じると閲覧の履歴・cookie・入力内容を残さない機能です。ダウンロードしたファイルとブックマークは残ります。匿名にはなりません。接続先のサイト、プロバイダー (ISP)、勤務先のネットワークの管理者は、通信の記録を見られます。[^c1-37]「プライベートで開けば会社閲覧が分からない」は誤りです。

Web を検索して情報を探し、保存する作業の注意点は次のとおりです。

- 検索結果には広告と本当の結果が混ざるので、URL を見てから開く
- アドレスが `https://` で始まるサイトは、TLS で通信を暗号化している (HTTPS、既定の番号は 443 番) [^c1-38]
- ページの保存、ブックマーク、履歴は、ブラウザの設定画面で管理する
- 履歴・cookie・キャッシュの削除は、設定画面から行える

アドレスが `http://` で始まるサイトでは、パスワードなどが暗号化されずに流れます。ログイン画面や支払いの画面では、`https://` かどうかを必ず見ます。

HTTPS が暗号化するのは通信路で、サイトの中身が安全であることまでは保証しません。プライバシーを守る道具には、ブラウザの設定 (cookie の拒否や削除、履歴の削除)、プライベートブラウジング、広告や追跡を遮る拡張機能、パスワード管理ツールがあります。どれも、見られる範囲を狭めるだけで、通信の相手や管理者から完全に隠れる道具ではありません。

コマンドラインへの入り方

Linux の画面操作は 2 通りあります。**GUI** (グラフィカルユーザーインターフェース) は、アイコンやウィンドウをマウスで操作する方式です。**CLI** (コマンドラインインターフェース) は、文字を打ち込んで、文字で結果を受け取る方式です。Linux の GUI は、X Window System (現在は Wayland も使う) の上で、GNOME や KDE などのデスクトップ環境が動いて実現されています。[^c1-39]

ターミナルとコンソールは、似ていますが別の言葉です。

用語	意味	例
コンソール	機械に直接つながる端末。 仮想コンソールがある	/dev/tty1・/dev/tty2 など
ターミナル (端末エミュレータ)	デスクトップの画面の中に開く、コマンド入力用のウィンドウ	GNOME Terminal など

デスクトップを使っても、Ctrl+Alt+F1～F6 のような組み合わせを押すと、文字だけの仮想コンソールに切り替わります。[^c1-40] 画面が固まったとき

の逃げ道や、GUI の入っていないサーバーで使い、コマンドで切り替える道具は `chvt` です。

端末の種類は `tty` コマンドで分かり、ターミナルのウィンドウでは `/dev/pts/0` のような擬似端末の名前が、仮想コンソールでは `/dev/tty3` のような名前が表示されます。Debian 12 のコンテナの端末で実行すると、実際に `/dev/pts/0` と表示されました。`pts` はターミナルの窓、`tty` は仮想コンソール、と見分けます。

プロンプトの記号は、実行している人を示します。一般ユーザーは `$` で終わり、root(管理者)は `#` で終わります。

パスワードの問題

パスワードの決め方で押さえる点は、次の3つです。

- 長さを重く見る。短くて複雑な文字列より、長い文字列のほうが破られにくい
- ほかのサービスと使い回さない
- 漏れた疑いがあれば、すぐに変える

NIST のガイドライン (SP 800-63B-4、2025 年 7 月) は、パスワードだけでログインする場合は最低 15 文字、多要素認証の一部として使う場合は最低 8 文字とし、長さを重く見るよう求めています。文字種の組み合わせの強制と、定期的な変更の強制は禁じています。漏れた証拠があるときは変更させます。
[^c1-41]

Linux では、パスワードの長さや文字の決まりを PAM のモジュール `pam_pwquality` が検査し、設定は `/etc/security/pwquality.conf` に書きま
す (`minlen` が最小の長さ)。[^c1-42] Rocky Linux 9 のこのファイルには、
`minlen = 8` がコメントアウトされた既定値として書かれています。

場面で考えます。社員が「パスワードは 8 文字以上の英数字」という規則に従い、全サービスで同じ文字列を使い回していたとします。1 つのサービスから漏れると、ほかのサービスにも入られます。規則の文字数を満たしていても、使い回しは避けなければなりません。管理者が「3 か月ごとに強制変更」を決めても、利用者が末尾の数字だけを変えるようになり、守りは強くなりません。長い文字列にして、使い回さず、漏れたときに変える運用が筋です。

一般ユーザーは、自分のパスワードを `passwd` コマンドで変えます。root は、ほかのユーザーのパスワードを、古いパスワードを知らなくても変えられます。[^c1-43] 利用者がパスワードを忘れたときは、管理者 (root) に再設定を頼むしかありません。保存されているのは暗号化した値だけなので、管理者でもパスワードそのものは見られません。再設定で新しい値に置き換えます。

Linux の業界での使われ方

Linux は、サーバー、クラウド、組み込み機器、スーパーコンピュータなど、幅広い現場で使われます。企業が選ぶ理由は、ライセンス費用を抑えやすいこと、ソースが読めること、設定を自動化しやすいことです。

クラウドと仮想化

クラウドの考え方は 1.1 で見ました。パブリッククラウドの利点は、機器を買わずに使った分だけ払えることと、台数をすぐ増減できることです。[^c1-44]

仮想化は、1 台の物理マシンの上に、複数の仮想的なマシンを作る技術です。

仮想マシンは、ハイパーバイザの上で OS ごと動かす方式で、Linux では KVM が代表です。[^c1-45] **コンテナ**は、OS のカーネルを共有したまま、アプリごとに隔離した環境を作ります。

方式	カーネル	起動	隔離の単位
仮想マシン	マシンごとに別のOS・カーネルを持つ	遅め	仮想的なハードウェアごと
コンテナ	ホストのカーネルを共有する	速い	アプリの実行環境

仮想マシンの中では Windows も動かしますが、コンテナは Linux カーネルを共有するので、Windows 用のアプリをそのままでは動かさせません。1 台の物理サーバーに多くのサービスを詰めたいときは、軽いコンテナが向きます。OS ごと別の環境が必要なら、仮想マシンを選びます。

発表や企画でのオープンソースのアプリ

業務で資料を作るときに使える、無償のオープンソースのアプリ:

目的	アプリ
発表資料を作る	LibreOffice Impress
報告書などの文書を書く	LibreOffice Writer
数値を集計し、グラフを作る	LibreOffice Calc
写真を修整する	GIMP
ロゴや図(ベクター)を描く	Inkscape
メールを読み書きする	Thunderbird
Web を見る	Firefox

LibreOffice の標準形式は ODF で、相手が別のソフトウェアを使っている場合は、保存形式を選んで渡します。ファイルを共有する場面では、オンラインストレージの Nextcloud や ownCloud も使えます。

複数の知識を組み合わせる場面

試験の難しい問題は、1つの節の知識では解けません。

1つ目は、クラウド上の Web サーバーです。パブリッククラウドの IaaS で Rocky Linux の仮想マシンを借り、Apache HTTPD を入れます。Rocky Linux は Red Hat 系なので、`yum install httpd` のように入れ、待ち受けは既定で Listen 80 です。Ubuntu の仮想マシンでは、yum は使えません。同じ作業は `apt-get install apache2` で行い、OS の更新は IaaS なので利用者の仕事です。

2つ目は、製品に組み込む部品の選定です。自社の製品にライブラリを取り込み、ソースは公開せずに配りたい場面を考えます。GPL のライブラリは、組み込むと製品も GPL にしなければなりません。LGPL か、BSD か MIT のライブラリなら、条件の表示を守れば取り込めます。

3つ目は、端末が使えなくなった場面です。デスクトップの画面が固まったら、Ctrl+Alt+F3 など仮想コンソールへ切り替え、ログインしてコマンドで直します。このとき開いているのは、ターミナルではなくコンソールです。

間違いやすい点

- ターミナルは画面の中のウィンドウで、コンソールは機械に直接つながる端末
- プライベートブラウジングは匿名ではなく、ダウンロードしたファイルは残る
- パスワードは、複雑さより長さを重く見る
- 仮想マシンはカーネルごと別物で、コンテナはホストのカーネルを共有する
- 管理者は、忘れたパスワードを調べるのではなく、新しい値に再設定する

SAMPLE

サンプルはここまでです

続き(第2章～巻末)は、Linux Essentials 問題集のご購入で、頻出度別ノート・暗記用ノートと合わせて3点すべてダウンロードできます。

SAMPLE

Linux Essentials 010-160 学習用テキスト(無料サンプル)

版

2026年10月版(LPI「Linux Essentials Exam 010 Objectives」Version 1.6(試験コード 010-160・LPI 公式・2026-10-13 取得)準拠)

発行

IT資格道場(<https://www.shikaku-doujou.com>)

お問い合わせ

info@shikaku-doujou.com

Linux は Linus Torvalds 氏の米国およびその他の国における登録商標です。Linux Essentials と LPI は、Linux Professional Institute の商標です。記載されているその他の会社名・製品名・サービス名は、各社の商標または登録商標です。

本教材は LPI が公開する出題範囲(Linux Essentials Exam 010 Objectives Version 1.6)の内容を基に作成したもので、Linux Professional Institute との提携・承認関係はありません。

本文・図表・設問はすべてオリジナルの著作物です。無断での転載・複製・再配布・二次販売はできません。お気づきの点は info@shikaku-doujou.com までご連絡ください。