IT 基礎

TS011-15

サンプル

テクノファイブ株式会社

サンプル

本テキストはテクノファイブ株式会社が著作権および使用権を有しています。著作権者による事前の許諾なしに、いかなる手段による全部または一部の複写、再利用、第三者への提供もしくは使用の許諾、その他の処分をすることはできません。

Copyright 2009－2026 Technofive Co., Ltd. All Rights Reserved.

目　　次

はじめに ..... **V**

コースの目的 vi

■ コースの目標 vi

第 1 章　IT とは **1**

1 - 1　IT とは 2

■ IT の目的 2

■ 価値の側面 4

■ 道具としての価値、サービスとしての価値 5

■ AI とは 6

■ AI と IT 7

第 2 章　情報の表現 **9**

2 - 1　コンピュータで扱う情報 10

■ 情報の単位 10

2 - 2　数値の表現のしくみ 11

■ 10 進数と 2 進数 11

■ 10 進数から 2 進数への変換 13

■ 2 進数と 16 進数 14

■ 2 進数の表現範囲 15

■ 2 進数における負の整数の表現 16

■ 2 進数における小数の表現 17

■ 補助単位 21

2 - 3　数値の演算 22

■ 論理演算 22

2 - 4　文字の表現方法 25

第 3 章　ハードウエア **29**

3 - 1　コンピュータの構成要素 30

■ コンピュータの 5 大要素 30

3 - 2　コンピュータを構成するハードウエア 31

■ 入力装置 31

■ 出力装置 33

■ 中央処理装置 35

■ 主記憶装置 38

■ 補助記憶装置 40

■ その他の装置 45
■ バス（伝送路） 46

第 4 章　ソフトウエア 49

4 - 1　ソフトウエアとは 50
■ ハードウエアとソフトウエア 50
4 - 2　基本ソフトウエア 51
■ 基本ソフトウエアの種類 51
■ オペレーティング・システム 52
■ ユーティリティ 68
■ 開発環境 68
4 - 3　ソフトウエア開発 69
■ ソフトウエア開発の工程 69
■ 開発プロセス 71
■ プログラミング言語 74
■ コンパイラとインタプリタ 75
■ プログラミング手法 76
■ ライセンス 80
4 - 4　ミドルウエア 81
4 - 5　応用ソフトウエア 82
■ アプリケーションの役割 82
4 - 6　仮想化ソフトウエア 83
■ 仮想化ソフトウエアとは 83
■ 仮想化ソフトウエアの種類 84

第 5 章　IT とコンピュータ 89

5 - 1　IT におけるコンピュータの役割 90
■ インターネットのビジネスの活用 90
■ 基幹系システム 92
■ 情報系システム 93
5 - 2　デジタルトランスフォーメーション 101
■ 近代の DX 101
■ （参考）DX とデジタル化との違い 103
5 - 3　コンピュータ・システムにおける「保証」 105
■ システムの性能面の指標 105
■ システムの障害耐性 106
■ システムの障害耐性の指標 107
■ 信頼性を高めるための設計 108
■ データのバックアップ 111

第 6 章　ネットワークとセキュリティ（付録）.....113

6 - 1　ネットワークの重要性.....114

■ LAN と WAN.....115

■ インターネットとプロトコル.....116

■ インターネット上のアプリケーション.....117

6 - 2　ネットワークの安全性.....119

■ 暗号化.....120

■ 電子署名.....121

■ 認証.....121

サンプル

（空白ページ）

サンプル

はじめに

サンプル

コースの目的

このコースでは、IT（Information Technology）分野の基礎知識を修得することを目的とします。IT とは何かを理解し、IT 化を実現するために利用するコンピュータ・システム（ハードウエア・ソフトウエア）の基礎知識、ビジネスにおけるコンピュータの活用まで幅広く学習します。

■ コースの目標

このコースを受講することにより、次のことが理解できるようになります。

- IT とは何か、IT とコンピュータの関係を理解する。
- コンピュータ・システムで扱う情報とその表現方法を理解する。
- ハードウエアとソフトウエアの違いを理解する。
- それぞれのソフトウエアの機能と役割を理解する。
- コンピュータがビジネスにどのように活用されているかを理解する。

サンプル

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

第1章 IT とは

サンプル

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

1 - 1 IT とは

情報化社会と言われて久しい現在、社会における活動にはコンピュータ・システムが欠かせません。そこでは情報処理と通信技術を基盤として「IT（Information Technology）」という概念が存在しています。

「IT」という言葉の明確な定義はありませんが、例えば次のように定義されています。

「情報を取得、加工、保存、伝送するための科学技術」（e-Words[1]）



もっと簡単に訳せば、IT とは「**情報を効率よく処理する技術**＝情報技術」だといえます。

■ IT の目的

現在、IT はさまざまなシーンで利用されています。



自宅・オフィス



公共交通機関の制御

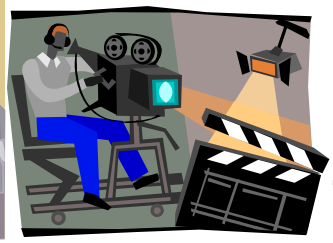




金融機関



医療機関



特殊撮影、編集



スーパーのレジ



郵便、宅配

[1] https://e-words.jp/w/IT.html

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

「IT」という言葉の持つ意味

「IT」という言葉の意味をそのまま受け取ると、**「情報を扱う（＝処理する）ための技術」**と考えられます。
IT を理解するためには、データ（Data）と情報（Information）との違いを理解しなければなりません。
データとは「何かを文字や符号、数値などのまとまりとして表現したもの[2]」です。
一方、情報とは「人が知覚したときに何らかの意味を想起させ、思考や行動に影響を与えるもの[3]」です。

例えば、お店の売上データや鉄道会社におけるある駅での入出場記録はデータです。データは単なる数値や文字などの羅列です。データは、ただ単に売れた、人が入場した、出場した、という定量的な表現をしているに過ぎません。

一方、売上を集計したり、駅の利用状況を分析したりすることによって、どんな商品に人気があるか、駅の利用者数が増えるのはどのような状況の時か、というようなことがわかります。これは、データを適切に処理することによって、データから情報を作り出していることになります。情報になれば、人はその情報から意志決定に必要な知見を得ることができます。

この、**データを情報化にする処理**過程において必要な技術が、IT です。



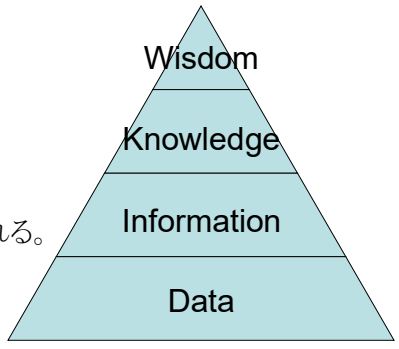
IT の意味するところをまとめると、次のような言葉で表現できます。

「データ」を
《大量・高速・正確に》《記録・集計・分析・通信》
することで、意志決定に役立つ「情報」に変化させ、<u>価値を産み出す</u>こと

DIKW モデル

近年では、情報を解釈するためのフレームワークとして、DIKW モデルという概念が使われています。

- Data　　：整理されていない「生のデータ」。
  例：○月○日の○時○分に商品××が△個売れた。
- Information　：Data を何らかの基準で分類・整理したもの。
  例：商品××の○月の売上高は□□円である。
- Knowledge　：Information を分析し、導き出される規則性や傾向。
  例：商品××は春・夏よりも秋・冬、特に○月によく売れる。
- Wisdom　　：Knowledge を活用して意思決定する力。
  例：商品××は、来月は△△個仕入れよう。



近年では、意思決定（Wisdom・知恵）に **AI**（Artificial Intelligence）の力を借りることが増えてきました。
意思決定を人間がするにしても AI がするにしても、社会生活を営むためには IT は欠かせません。

[2] https://e-words.jp/w/%E3%83%87%E3%83%BC%E3%82%BF.html
[3] https://e-words.jp/w/%E6%83%85%E5%A0%B1.html

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

■　価値の側面

前ページでは、「IT は仕事や日常生活において意思決定に必要な情報を人間に提示し、価値をもたらすこと」であると説明しました。

「価値」を辞書で引くと、「物事の役に立つ性質・程度（広辞苑）」とあります。役に立つ度合いが高ければ高いほど価値がある、ということです。

価値には、2 つの側面があります。

有用性

「目的適合性」ともいいます。製品やサービスを使うことによって目的を達成できる度合いのことです。今までできなかったことが、その製品やサービスを使うことによってできるようになったり、従来に比べてより効率的・効果的にできるようになったりすれば、その製品やサービスには有用性があるといいます。

例えば、チャット機能は「文章やスタンプなどを用いて、遠隔地にいる人と容易にコミュニケーションがとれる」という有用性をもたらします。手紙や電子メールにも同様の機能がありますが、チャットのリアルタイム性や利便性にはかないません。また、携帯電話やスマートフォン（以下、スマホ）を用いたチャット機能は、PC を用いたチャットに比べて「外出先でも自由にタイムリーなコミュニケーションがとれる」という有用性を提供します。

保証

上記の有用性が確実に提供される度合いのことです。

スマホを用いたチャットは高い有用性を提供しますが、スマホの電波が圏外になったり、スマホのバッテリーがなくなったり、スマホがウイルスに冒されていて使えなくなったりすると、利用者はチャットの有用性を確実に受け取ることができません。

逆に、いつでもどこでも使いたい時にスマホのチャット機能が本来の有用性を発揮してくれるのであれば、それは「有用性が保証されている度合いが高い」といいます。

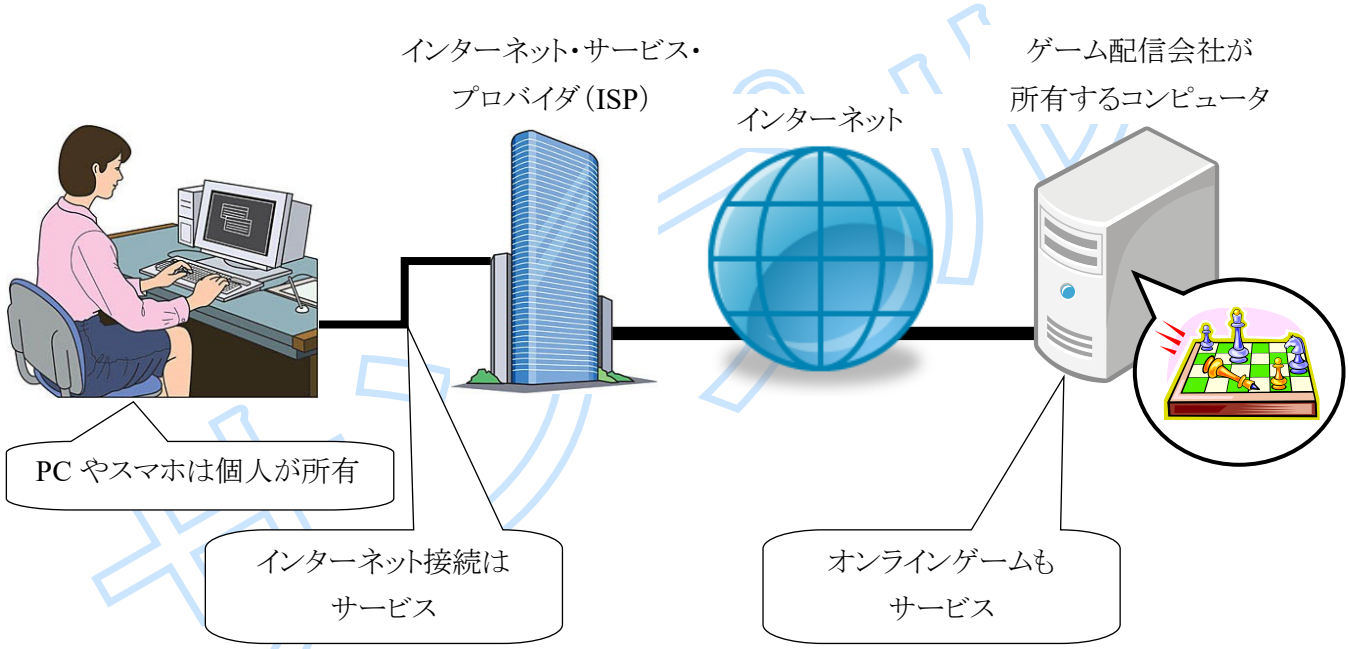
実際には、**価値は　有用性　と　保証　のどちらも満たされていなければなりません**。有用性が高いのに保証が低かったり、逆に保証が高いのに有用性が低かったりすれば、その製品やサービスは価値があるとは言えない（つまり、役にたつとはいえない）のです。

■ 道具としての価値、サービスとしての価値

IT には、道具としての価値を提供する側面と、サービスとしての価値を提供する側面とがあります。

例えば、自宅の PC やスマホからインターネットに接続し、オンラインゲームをして遊ぶ、という形態を考えてみましょう。

自宅にある PC は、ゲームで遊ぶ人自身が所有しています。スマホもゲームで遊ぶ人が所有しています。しかし、インターネット回線は通信事業者が所有し、ゲームで遊ぶ人は必要な時にインターネット接続サービスを受けているに過ぎません。また、ゲームのプログラムが格納されているコンピュータはゲーム配信業者が所有しており、これもまた必要な時にだけ接続して「ゲームで遊ぶ」というサービスだけを受けていることになります。



このように IT の世界では、個人が所有し、道具としての価値を提供しているという形態と、別の企業や団体が所有し、サービスとしての価値を受けているという形態があります。

コンピュータ・システムを、使用者自身が所有して管理することを**オンプレミス**、他者が所有し、提供しているコンピュータ・システムのサービスを必要に応じて利用することを**オンデマンド**といいます。現在の IT では、このオンプレミスとオンデマンドがさまざまな形態で融合した状態で存在しています。

さらに **IoT**（Internet of Things、モノのインターネット）という用語もあります。「コンピュータだけでなく、ありとあらゆるモノがプロセッサや通信機能を持ち、インターネットにつながる」という形態のことです。ありとあらゆるものがインターネットにつながることによって、必要な情報がすべてインターネット上にあり、インターネットが情報基盤の中心を担う、という状態になりました。

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

■ AI とは

AI（Artificial Intelligence）とは、「人間の知的な活動をコンピュータで模倣・再現する技術」であるといえます。人間の知的な活動とは、例えば次のようなものを指します。



- 見る、読む、聞く、認識する、理解する
- 考える、判断する
- 学ぶ、失敗しづらくなる、成長する
- 予測・推測する

AI は、これらをコンピュータで実現しようとする技術です。

AI の特徴

近年の AI の最大の特徴は、次の 3 点です。

- 機械学習
  明示的にプログラミングすることなく、データから規則性を学習することをいいます。
  例えば、大量の犬の画像を「これは犬です」と教えながら読み込ませ、次に大量の猫の画像を「これは猫です」と教えながら読み込ませます。すると AI は、犬の共通の特徴、猫の共通の特徴を学習します。すると、犬の画像と猫の画像を見分けることができるようになります。

- 深層学習（ディープ・ラーニング）
  機械学習を拡張し、規則性を何層にも重ねて学習することをいいます。
  例えば上記の犬や猫の画像を学習させる際、深層学習では犬の顔を「目、耳、鼻、口」などの部品レベル（第 1 層）で学習すると共に、それらの位置関係やバランスをもう一つ下の層（第 2 層）として学習します。細かい特徴を何層も重ねて学習できるので、機械学習よりも複雑な理解、推論が可能です。

- 強化学習
  報酬を最大化する行動方策を学習することをいいます。
  例えば、勝つとポイントがもらえ、負けるとポイントを失うゲームを AI が何度も試行します。すると、「こうするとより多くのポイントが得られる」という方法を学習し、AI がだんだん強くなっていきます。「良い／悪いはわかるが、どうすればより良くなるかがわからない」、というものを AI に学ばせる際に使用します。

AI の活用例

AI は、次のような分野で用いられます。

- 認識
  - 画像認識
  - 音声認識
  - 文字認識
  - 異常検知　など
- 理解・分析
  - 自然言語処理（文章の理解）
  - 意図の推定
  - データ分析
  - 因果関係の推定　など
- 予測
  - 需要予測
  - 売上予測
  - リスク予測
  - 行動予測　など
- 生成
  - 文章生成
  - 画像・動画生成
  - 音楽・音声生成
  - コード生成　など
- 判断・意志決定
  - 与信判断
  - 不正検知判断
  - 医療支援
  - おすすめの提案　など
- 最適化
  - 配送ルートの最適化
  - 在庫の最適化
  - 広告配信の最適化
  - スケジュールの最適化
  - ゲームや試合の勝率最大化　など
- 制御
  - 自動運転
  - ロボットの制御
  - ドローンの制御
  - 工場機械の精度　など

■ AI と IT

厳密には、AI は IT の一種です。しかしあえて違うものとして説明すると、下記のようにまとめられます。

観点	ＩＴ	ＡＩ
目的	情報処理	知的判断
処理の性質	ルール通りに動く	データから学ぶ
不確実性への対応	苦手	得意
数学的基盤	アルゴリズムを設計する	統計・確率・最適化を基にする

（空白ページ）

サンプル

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

第2章 情報の表現

サンプル

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

2 - 1 コンピュータで扱う情報

コンピュータは、すべての情報をオン(1)とオフ(0)の 2 つの状態で表現します。実際には、コンピュータ内部の電気信号の有無や電圧の高低により、1 と 0 の 2 つの状態が表現されます。



電荷があれば 1



電荷がなければ 0

この 1 と 0 との 2 種類の数値で物を数えたり、情報を表したりするやりかたのことを「2 進数」といいます。

しかし 1 と 0 だけでは人間にとっては非常に扱いづらいため、その情報を以下のようにまとめて扱います。コンピュータの内部では、数値データも文字データも、すべて 2 進数である 0 と 1 によって表現されているのです。

■ 情報の単位

すべてが 0 と 1 で表現されるといっても、0 と 1 の羅列ではどこからどこまでが意味のあるデータか分かりません。そのため、コンピュータが扱う情報は、以下の単位で処理されます。

ビット(binary digit)

2 進数の 1 ケタを「ビット」といいます。binary は 2 進数、digit はケタの意味です。

バイト(Byte)

一般的には、2 進数 1 ケタ(ビット)が 8 つ、つまり 8 ビットをひとかたまりとして「バイト(Byte)」といいます。元々は複数ビットをひとまとめにしたものを指した用語ですが、現在では 8 ビット以外の意味で使われることはありません。1 バイトで表現できるのは 10 進数で 0 から 255 までです。

ワード(Word)

コンピュータで最も一般的な処理サイズを「ワード(Word)」といいます。1 ワードが何ビットになるかは、コンピュータの種類(厳密には CPU の種類)によって異なります。32 ビット、64 ビットが一般的です。

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

2 - 2 数値の表現のしくみ

私たちが物を数えるときは「1、2、3、・・・ 9、10、11、・・・」と、自然に 10 進数で考えています。10 進数とは、0 から 9 までの 10 種類の数字で数えるやりかたです。



対して、コンピュータは 2 進数で情報を表現します。2 進数のままでは人間が理解しづらいので、なんらかの形で 2 進数から 10 進数に変換するしくみが必要です。また、コンピュータは 10 進数のままではその数を理解することはできませんから、同様に 10 進数から 2 進数またその逆を行うしくみが必要です。

さらに必要に応じて 8 進数や 16 進数など、その他の進数に置き換える必要があります。

■ 10 進数と 2 進数

コンピュータはどのようにして、10 進数を 2 進数として扱っているのでしょうか。普段私たちが使っている 10 進数と、コンピュータで扱われる 2 進数を元に考えてみましょう。

10 進数

10 進数の規則は以下のものがあります。

- 0 から 9 までの 10 種類の数字を使用する。
- 9 の次はケタが 1 つ繰り上がり、「10」という 2 ケタの数字で数値を表現する。

例えば、10 進数で 256 という数値を考えた場合、以下のようになります[4]。

$$256 = \underline{10^2} \times 2 + \underline{10^1} \times 5 + \underline{10^0} \times 6$$

各ケタに対して、10 のべき乗に各ケタの数を掛け合わせたものをすべて合算した数と考えられます。

10 進数では、この「10」を基数といい、各ケタに乗じている 10^n（n は各ケタを表す数）を「重み」と呼んでいます。どのような進数であっても、この考え方が基本となります。

[4] すべての数の 0 乗は1です。

2 進数

10 進数と同じように考えると、2 進数の規則は以下のようになります。

・0 と 1 の 2 種類の数字を使用する。

・1 の次はケタが 1 つ繰り上がり、「10」という 2 ケタの数字で数値を表現する。

例えば、1010 という数値を 2 進数で考えた場合、以下のようになります。

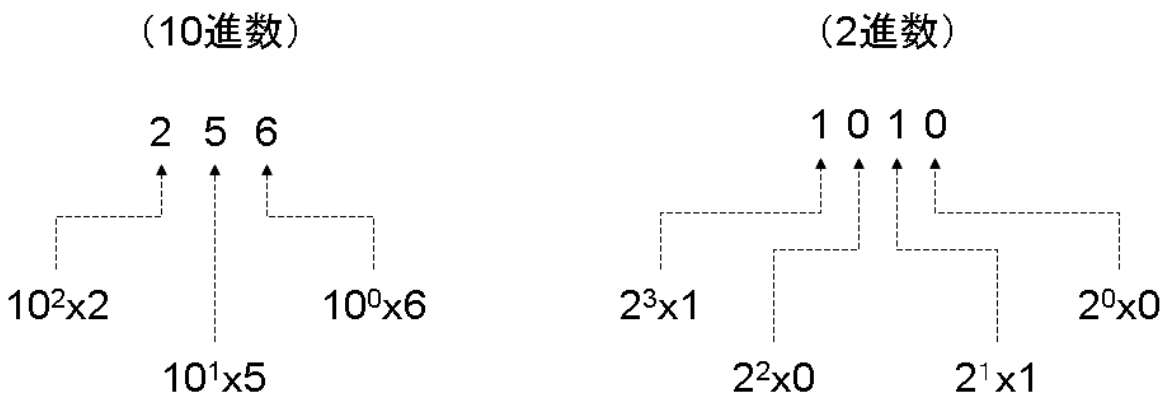
$$1010 = \underline{2}^3\times1+\underline{2}^2\times0+\underline{2}^1\times1+\underline{2}^0\times0 \rightarrow 8 + 0 + 2 + 0 = 10$$

これが何を意味するかというと、「2 進数の 1010 を 10 進数に変換すると 10 である」ということです。
このように、ある進数から別の進数に変換することを、基数変換といいます。

別の表現方法として、10 進数と 2 進数を区別するために上記の例は以下のように表現されます。

$$(1010)_2 = \underline{2}^3\times1+\underline{2}^2\times0+\underline{2}^1\times1+\underline{2}^0\times0 \rightarrow 8 + 0 + 2 + 0 = (10)_{10}$$

基数と重み

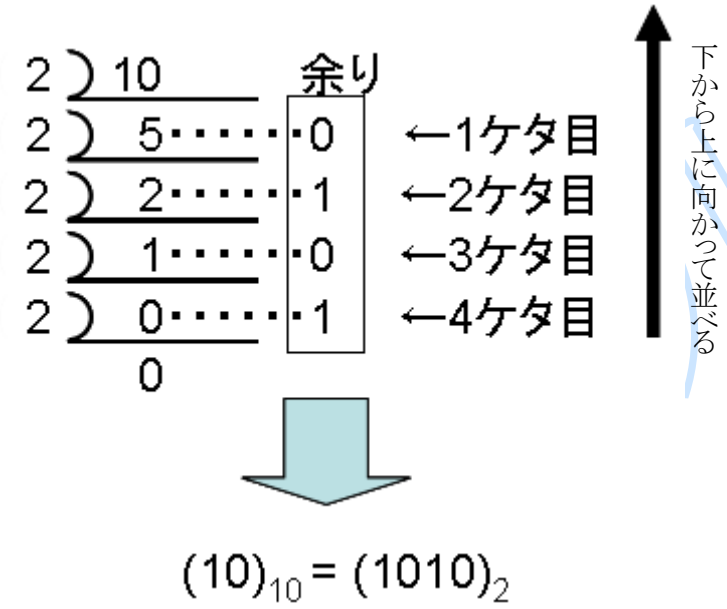


Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

■　10 進数から 2 進数への変換

10 進数から 2 進数への変換は、単純な方法で可能です。それは、10 進数の数値を 2 で割っていき、余りを求めることです。

例えば 10 を 2 で割った余り 0 が最下位（1 ケタ目）の値となります。次に、5 を 2 で割った商 2 をもう一度 2 で割ると、その余り 1 は次の位（2 ケタ目）の値となります。これを商が 0 になるまで繰り返します。出てきた余りを下から並べると 1010 となります。



$(10)_{10} = (1010)_2$

この方法は 2 進数だけでなく、何進数でも使えます。後述する 16 進数なら、16 で割っていけばよいのです。

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

■ 2 進数と 16 進数

2 進数は、0 と 1 の種類で情報を表現する半面、扱う数値が大きくなると、桁数が非常に大きくなってしまいます。そこで 16 進数を使うことが考えられました。

16 進数とは、0 から 9 の 10 文字に加え、A から F の 6 文字を数字として使って数値を表現する方法です。

10 進数と 2 進数、16 進数の関係

10 進数	2 進数	16 進数
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F
16	10000	10
17	10001	11
18	10010	12
19	10011	13
20	10100	14
⋮	⋮	⋮

10 進数と異なり、16 進数は 2 進数との相互変換が非常に簡単です。16 進数 1 ケタで表現できる最大の数は $(F)_{16}$ で、これは 2 進数の 1111 です。つまり、2 進数 4 ケタを 16 進数 1 ケタで表すことができます。

1 バイトは 8 ビットなので、16 進数 2 ケタでちょうど過不足なく表現できます。そのため、現在では 16 進数を使った表現が多く使われます。

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

■　2 進数の表現範囲

2 進数では扱うビット数が大きくなる（ケタが増える）ごとに 2 倍、4 倍、16 倍・・・と べき乗 になります。

桁数（ビット数）	表現できる情報量（10進数）	指数表現
1	2	2^1
2	4	2^2
3	8	2^3
4	16	2^4
5	32	2^5
6	64	2^6
7	128	2^7
8	256	2^8
16	65,536	2^{16}
24	16,777,216	2^{24}
32	4,294,967,296	2^{32}
64	18,446,744,073,709,551,616	2^{64}

コンピュータ（厳密には CPU）が一度に扱える情報の単位はワードなので、そのコンピュータが 1 ワード 32 ビットなのか 64 ビットなのかによって、一度に扱える情報量は変わってきます。

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

■ 2 進数における負の整数の表現

2 進数において負の整数を扱うためには、少々工夫が必要です。

ここではわかりやすくするために、4bit で考えます。4bit あれば 16 通りの整数を表現できます。

2 進数	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
10 進数	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

しかしこれでは負の整数を表現することができません。そこで、下記のように考え方を少し工夫します。

2 進数	1000	1001	1010	1011	1100	1101	1110	1111	0000	0001	0010	0011	0100	0101	0110	0111
10 進数	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7

これは「$(1111)_2$ に 1 を加算したら $(0000)_2$ になる」という考え方です。実際は $(1111)_2$ に 1 を加算したら $(10000)_2$ ですが、4bit しかないために桁あふれを起こしてしまって $(0000)_2$ になっている、と考えます。
このように考えれば、4bit あれば -8 から 7 までの 16 通りの整数を表現できます。

同様に、8bit あれば -128～127、16bit あれば -32768～32767、32bit あれば -2147483648～2147483647 の範囲の整数を表現することができます。

2 の補数

ここで、正の整数と負の整数との関係に注目します。

2 進数		0111	0110	0101	0100	0011	0010	0001	0000
10 進数		7	6	5	4	3	2	1	0
2 進数	1000	1001	1010	1011	1100	1101	1110	1111	
10 進数	-8	-7	-6	-5	-4	-3	-2	-1	

例えば、10 進数の 1 は 2 進数では 0001、10 進数の -1 は 2 進数では 1111 です。
この両者の関係は、「**0 と 1 を逆転させて 1 を加える**」と表現することができます。

$(1)_{10}$ の 2 進数表現 $(0001)_2$ の 0 と 1 を反転させて $(1110)_2$ 、1 を加えると $(1111)_2$ すなわち $(-1)_{10}$
$(-1)_{10}$ の 2 進数表現 $(1111)_2$ の 0 と 1 を反転させて $(0000)_2$ 、1 を加えると $(0001)_2$ すなわち $(1)_{10}$

この両者の関係を、**2 の補数**といいます。正の整数と負の整数とを変換する際の重要な考え方です。
また、正の整数は最上位ビットが 0、負の整数は最上位ビットが 1 である、ともいえます。

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

■　2 進数における小数の表現

2 進数で小数を扱う場合、数学的な考え方と実際にコンピュータで使われる方法は異なります。数学的には、整数部が倍々に増えて行ったように、小数部は半分ずつになると考えます。

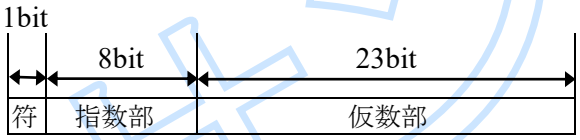
例えば、2 進数の 0.101 を 10 進数に変換する場合は、以下のように考えます。

$$(0.101)_2 = 1\times2^{-1} + 0\times2^{-2} + 1\times2^{-3} = 1\times1/2 + 0\times1/4 + 1\times1/8 = 5/8 = (0.625)_{10}$$

浮動小数点数

科学技術計算などにおいて非常に小さな値や非常に大きな値を扱う場合、8.854×10^{-42} や 6.02×10^{23} のように記述します。2 進数でも同じように、値を表す仮数部とケタを表す指数部に分けて表現します。これを**浮動小数点数**といいます。10 進数の小数を 2 進数の浮動小数点数に変換するにはいくつかの方法がありますが、最も一般的なのは IEEE754 で規定している方法です。

IEEE754 では、小数を符号部、指数部、仮数部に分割して格納します。32bit 形式の場合、それぞれの大きさは次の通りです。



符号部：正の数なら 0、負の数なら 1 を格納する
指数部：指数を 2 進数に変換し、その値に 01111111 を加えた値を格納する
仮数部：2 進数表現の小数点以下を仮数部に格納する

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

具体的な例として、 10 進数の 5.75 を 2 進数の浮動小数点数で表現することを考えます。

1. 10 進数を 2 進数に変換する

$$(5.75)_{10} = 4 + 0 + 1 + 0.5 + 0.25$$
$$= 1\times2^{2} + 0\times2^{1} + 1\times2^{0} + 1\times2^{-1} + 1\times2^{-2} = (101.11)_{2}$$

2.「1.」で始まるように小数点の位置をずらし、2^{n} を掛けて調整する（これを正規化という）

$$(101.11)_{2} = (1.0111)_{2} \times 2^{2}$$

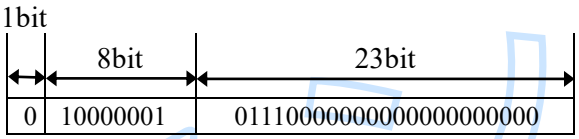
3. データの領域を、符号部、指数部、仮数部に分けて、次のように格納する。

・正の数なら符号部を 0 に、負の数なら符号部を 1 にする （今回の例では 0）

・2 の指数（べき乗）部分を指数部に格納する

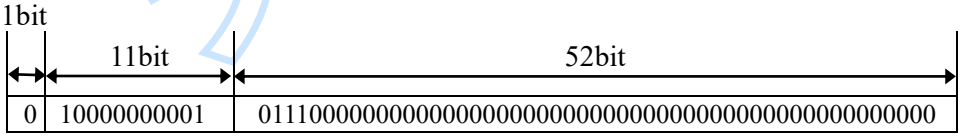
（今回の例では $(2)_{10}$ を 2 進数に変換した $(10)_{2}$ に $(01111111)_{2}$ を加えた $(10000001)_{2}$ ）

・2 進数表現の小数点以下を仮数部に格納する （今回の例では $(0111)_{2}$ ）



（参考） 64bit 形式の場合、指数部が 11bit、仮数部が 52bit になります。

指数部に加える値は $(01111111111)_{2}$ になります。



丸め誤差

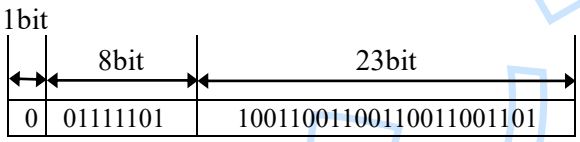
10 進数の小数を 2 進数の浮動小数点数に変換する場合、2^{-n} の組み合わせで表現することのできる値と表現できない値が出てきてしまいます。例えば、前述の 10 進数 5.75 の小数部分 0.75 は以下のように表現できるので、正確に 2 進数で表すことができます。

$$\begin{aligned} 0.75 &= 0.5 + 0.25 \\ &= 1\times2^{-1} + 1\times2^{-2} \end{aligned}$$

しかし、例えば 10 進数の 0.4 は 2^{-n} の組み合わせで正確に表現することはできません。上手に工夫して近似値で表現することしかできないのです。

$$\begin{aligned} 0.4 &\fallingdotseq 0 + 0.25 + 0.125 + 0 + 0 + 0.015620 + 0.0078125 + \cdots \\ &= 0\times2^{-1} + 1\times2^{-2} + 1\times2^{-2} + 0\times2^{-2} + 0\times2^{-2} + 1\times2^{-2} + 1\times2^{-2} + \cdots \end{aligned}$$

これを 32bit 形式の浮動小数点数で表現すると、以下のようになります。



これは非常に 0.4 に近いものですが、仮数部は 23bit しかないため、完全に正確な 0.4 ではありません。64bit 形式だと仮数部が 52bit になりますのでより 0.4 に近づけることができますが、それでも完全に正確な 0.4 であるとはいえません。このような、表現できる桁数に制限があるために誤差が出てしまうことを**丸め誤差**といいます。丸め誤差は、10 進数の小数を 2 進数の浮動小数点数で表現しようとする場合、どうしても出てきてしまう誤差です。

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

コラム:プログラムで 0.1 を 10 回足しても 1 にならない？

Java というプログラミング言語を用いて、以下のようなプログラムを作成します。

```
double num = 0.1, result = 0;
for(int i = 1; i <= 10; i++){
    result += num;
}
System.out.println(result);
```

このプログラムを簡単に説明すると、次の通りです。

浮動小数点数を格納することのできる num という名前の箱（変数）に 10 進数の 0.1 を格納した上で、
その num を 10 回足した結果を result という変数に格納し、
変数 result を画面に表示する

0.1 を 10 回足しているわけですから、画面に表示される結果は 1 になるはずです。
しかしこのプログラムを実行すると、次のように表示されます。

```
0.9999999999999999
```

これは、浮動小数点数 num に格納されている値は、丸め誤差によって 0.1 よりほんのわずかに小さな値になっていることが原因です。ほんのわずかな誤差でも、10 回繰り返すと目に見える差になって表れます。

実際にプログラミング言語を用いてプログラムを作成する際には、こうした誤差をどう扱うか、ということを考慮に入れる必要が出てくる場合があります。

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

■ 補助単位

大きな数や小さな数を表現するときは「補助単位」を使います。一般的な補助単位は以下のとおりです。コンピュータの世界では、記憶装置の単位に大きな補助単位を使い、動作時間の単位に小さな単位を使うのが一般的です。

補助単位(大)

補助単位	大きさ	意味
K(キロ)	10^3	千倍
M(メガ)	10^6	百万倍
G(ギガ)	10^9	十億倍
T(テラ)	10^{12}	一兆倍
P(ペタ)	10^{15}	千兆倍
E(エクサ)	10^{18}	百京倍

補助単位(小)

補助単位	大きさ	意味
m(ミリ)	10^{-3}	千分の一
μ(マイクロ)	10^{-6}	百万分の一
n(ナノ)	10^{-9}	十億分の一
p(ピコ)	10^{-12}	一兆分の一

なお、便宜上 10 のべき乗で大きさを示しましたが、実際には 2 進数を基本としているため、例えば、K の場合は 1,000 倍ではなく 1,024 倍になるので注意が必要です。

コラム:1K は、1,000 ? それとも 1,024 ?

通常、コンピュータで扱う情報量の単位は

1K ＝ 1,024 （$=2^{10}$）

が基本ですが、特に記載が無い場合、1K ＝ 1,000 と考えてよいとされています。

また、販売されている携帯端末や補助記憶装置によっては、1K ＝ 1,000 で計算された記憶容量が記載されているものもあります。そのような場合、パソコンなどに接続して実際に使ってみると、1K ＝ 1,024 で計算されるので、容量が少なく見えます。

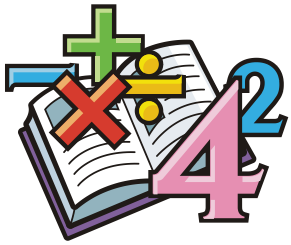
Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

2 - 3 数値の演算

コンピュータではすべての情報が 2 進数で表現されますが、単に数値同士を足したり引いたりするのか、それとも、ある条件式の真偽を判定するのかによって、演算の仕方が異なります。

数値演算を行う場合、大きく以下の 2 つに分かれます。

論理演算・・・演算対象となる数値の関係を表します。
算術演算・・・和・差・積・商に代表される四則演算を行います。



■ 論理演算

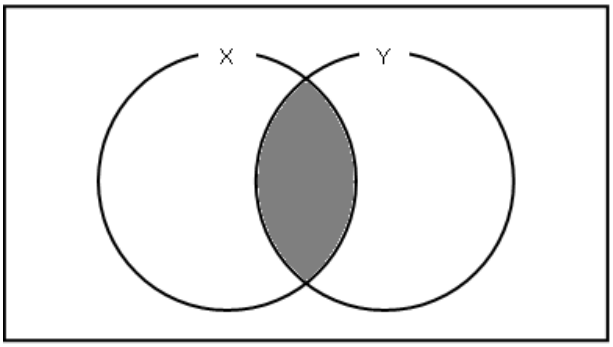
基本的な論理演算には、論理積、論理和、否定があります。論理演算は条件の真偽が演算対象となるため、足し算や掛け算の結果と異なる点に注意してください。

例えば、2 つの条件をそれぞれ、X、Y とし、条件が真のときを 1、偽のときを 0 とした場合、各演算の結果は以下のようになります。

論理積（AND）

2 つの条件がともに真（1）のときだけ、演算結果を真（1）とする演算です。

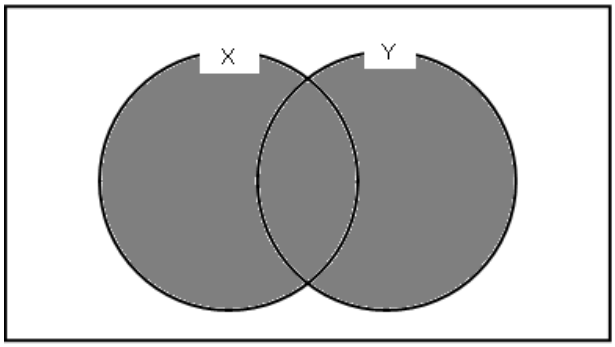
X	Y	X and Y
0	0	0
0	1	0
1	0	0
1	1	1



論理和（OR）

2 つの条件のうち 1 つでも真（1）のとき、演算結果を真（1）とする演算です。

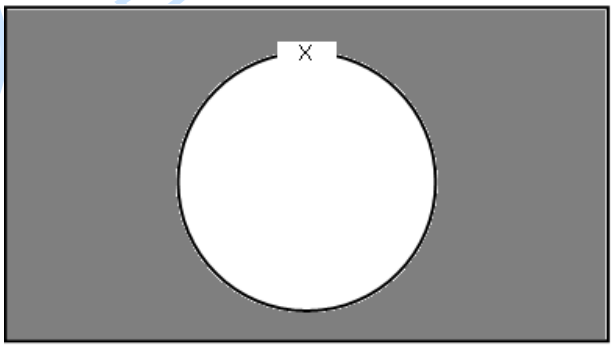
X	Y	X or Y
0	0	0
0	1	1
1	0	1
1	1	1



否定（NOT）

条件を反転させる演算を否定と呼びます。NOT を 2 回適用すると元に戻ります。

X	not X
0	1
1	0

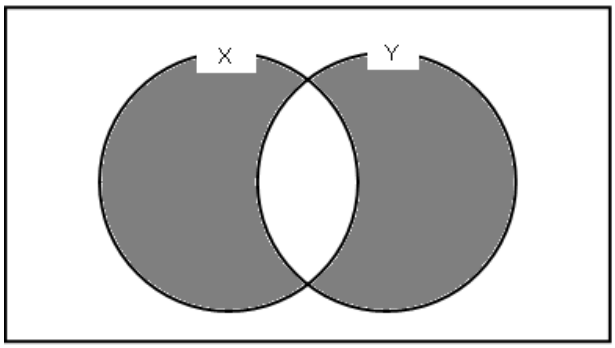


Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

排他的論理和（XOR）

2 つの条件の真（1）、偽（0）が異なるとき、演算結果を真（1）とする演算です。

X	Y	X xor Y
0	0	0
0	1	1
1	0	1
1	1	0



2 - 4 文字の表現方法

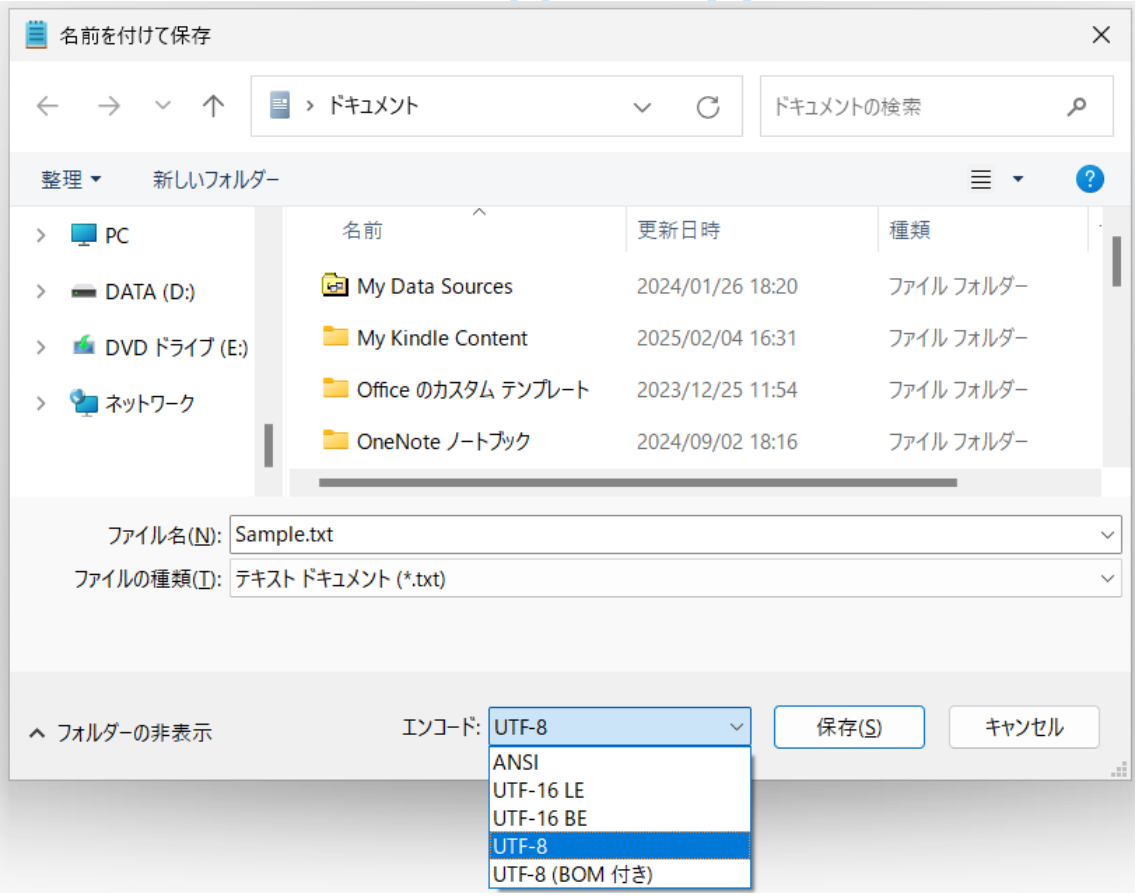
コンピュータで文字をデータとして扱うには、文字を数値化する必要があります。文字データは 1 文字をあらかじめ決めたビット数の数値で表現します。通常、1 文字は 8～32 ビットで表現しますが、この 0 と 1 の組み合わせをコード（符号）といいます。

また、文字を表現するためには、以下のことを考慮しなければなりません。

文字セット	どの文字を扱うかを決定します。アルファベットだけなのか、漢字を含めるのか、どの記号を含めてどの記号を含めないか等を決定します。
文字コード	どの文字にどのような数値を割り当てるかを決定します。文字セットによって、1 文字を 1 バイトで表現する場合と、複数バイトを必要とする場合とがあります。
エンコード	文字コードの表現方法を決定します。例えば、すべての文字を同じバイト長で表現するのか、文字ごとに異なる長さを認めるのか等を決定します。

これら 3 つは、文字をどのように決めているかという規則に相当するので、コード体系と呼ばれます。コード体系には JIS コードをはじめ、ASCII コード、Unicode など、さまざまな種類が存在します。

Windows のメモ帳でも、ファイルを保存する際に文字コードが選択できます。



Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

ASCII コード

「ASCII（American Standard Code for Information Interchange）コード」は米国の規格協会（ANSI：American National Standards Institute）が制定したもので、7 ビットで 128 種類の文字を表します。ASCII コードは、実質的には世界的な標準コードとなっており、国際標準機関である ISO（International Organization for Standardization）が制定した「ISO 646-1991」も ASCII コードに基づいています。

現在「半角英数字」と呼ばれている文字は、この ASCII コードです。

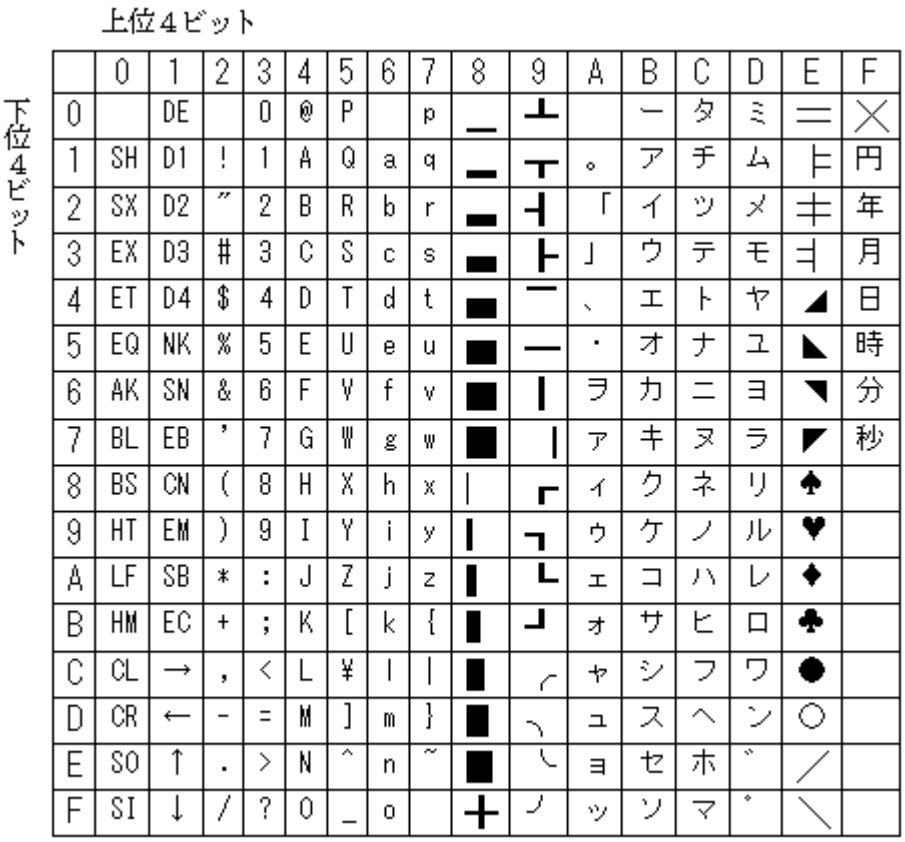
上位３ビット

下位4ビット	0	1	2	3	4	5	6	7
0	NUL	DLE	SP	0	@	P	`	p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	|
D	CR	GS	-	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	DEL

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

JIS コード（JIS X 0201）

カナ文字を扱うため、ASCII コードの 128 文字分の空き領域にカナ文字を追加した JIS コードが標準化されました。「半角カタカナ」を含む文字コードです。



下位4ビット ＼ 上位4ビット	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		DE		0	@	P		p	▁	┴		ー	タ	ミ	═	╳
1	SH	D1	!	1	A	Q	a	q	▂	┬	。	ア	チ	ム	╞	円
2	SX	D2	″	2	B	R	b	r	▃	┤	「	イ	ツ	メ	╪	年
3	EX	D3	#	3	C	S	c	s	▄	├	」	ウ	テ	モ	╡	月
4	ET	D4	$	4	D	T	d	t	▅	─	、	エ	ト	ヤ	◢	日
5	EQ	NK	%	5	E	U	e	u	▆	┼	・	オ	ナ	ユ	◣	時
6	AK	SN	&	6	F	V	f	v	▇	╎	ヲ	カ	ニ	ヨ	◥	分
7	BL	EB	’	7	G	W	g	w	▇	┃	ァ	キ	ヌ	ラ	◤	秒
8	BS	CN	(	8	H	X	h	x	▏	┌	ィ	ク	ネ	リ	♠	
9	HT	EM	)	9	I	Y	i	y	▎	┐	ゥ	ケ	ノ	ル	♥	
A	LF	SB	*	:	J	Z	j	z	▍	└	ェ	コ	ハ	レ	♦	
B	HM	EC	+	;	K	[	k	{	▋	┘	ォ	サ	ヒ	ロ	♣	
C	CL	→	,	<	L	¥	l	|	▉	╭	ャ	シ	フ	ワ	●	
D	CR	←	-	=	M	]	m	}	▉	╮	ュ	ス	ヘ	ン	○	
E	SO	↑	.	>	N	^	n	~	▉	╰	ョ	セ	ホ	゛	／	
F	SI	↓	/	?	O	_	o		╋	╯	ッ	ソ	マ	゜	＼	

JIS 漢字コード（JIS X 0213）

JIS コードでは漢字やひらがなを扱うことができませんでした。また 1 バイトでは最大でも 256 種類の情報を表すことしかできなかったため、事実上漢字を表すことは不可能でした。そこで、2 バイトで漢字を表す規格が策定されました。それが JIS 漢字コード（JIS X 0213）です。インターネットで情報をやりとりする際には、「ISO-2022-JP」という名称で表現されることもあります。

JIS コードと JIS 漢字コードをより簡単に扱えるように考案された、シフト JIS コードというものもあります。ただし、JIS 漢字コードとシフト JIS コードとは、コード体系に互換性がありません。

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

EUC コード

EUC は、UNIX システムで世界各国の文字を統一的に扱うためのコードです。EUC は AT&T 社が制定した文字コードで、日本語版の他、中国語版などがあります。日本語版の EUC コードは、「EUC-JP」と呼ばれます。JIS 漢字コードをベースにして作られていますが、JIS 漢字コードと完全な互換性があるわけではありません。

Unicode

中国や日本のように漢字を使用している言語圏では、2 バイトでも自国の文字をすべて表現することはできません。(2 バイト＝16 ビットは、情報量として 2^{16}＝65,536 までしか扱えません。) また、国際化、インターネット化が進んだ現代では、自国以外の文字も共通の文字コードで表す必要が出てきました。

そこで、Unicode と呼ばれる文字コードが提案されました。Unicode は Apple 社、IBM 社、Microsoft 社など米国の情報関連企業が中心となって提唱し、1993 年に ISO-10646 と統合、標準化されました。まったく新たに策定された規格なので、JIS 漢字コード、シフト JIS コード、EUC コードなどとは互換性がありません。

Unicode のエンコード方法を「UTF (Unicode または UCS Transformation Format)」と呼びます。エンコードとはデータを一定の法則で変換することを意味し、変換の法則をエンコード方式といいます。

Unicode という 1 つの文字コードに対して複数のエンコード方式があります。よく使用されるのは **UTF-8** と UTF-16 です。UTF-8 は、1 バイト文字は ASCII と互換性があり、それ以外の文字は最大 4 バイトで表現するもので、インターネットにおいて最もよく使用されています。UTF-16 は 1 文字を 2 バイト、または 4 バイトで表現するもので、バイト単位では ASCII と互換性がありません。「Unicode テキスト」というと、UTF-16 のことを指します。

コラム:パリティ

データ通信を行っている最中に、電気的なノイズや通信ケーブルの品質が悪いことなどによって、本来 1 であるビットが 0 になってしまったり、またその逆になったりすることがあります。ASCII コードでは間違いを検出するための方法として「パリティ」を使うことがあります。パリティは 2 進数データ中で 1 の値を持つケタの数を数えることでエラーを検出します。

パリティは、本来のデータに「パリティビット」と呼ばれる 1 ケタ (1 ビット) 追加します。1 になっているビット数が、1 文字全体で奇数個になるようにパリティビットを設定するものを「奇数パリティ」といい、逆に偶数個になるように設定するものを「偶数パリティ」といいます。

パリティビットを使ってデータ転送の誤りを検出することを「パリティチェック」といいます。ただし、パリティでは、どのビットが間違っているかは判別できないため、エラーが検出された場合は、データを再転送する必要があります。また、2 ビット以上の値が間違っている場合、偶数個のエラーを検出できません。そのため、現在では文字コードの検査にはほとんど使われていません。

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

第3章 ハードウエア

サンプル

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

3 - 1 コンピュータの構成要素

パーソナル・コンピュータ（パソコン）などのように個人が利用するコンピュータをはじめとして、コンピュータにはさまざまな種類が存在しています。大型で超高速なスーパーコンピュータから家電製品を制御するための組み込みのマイクロチップまで多岐に亘ります。しかし、それらは共通の構造を持っています。

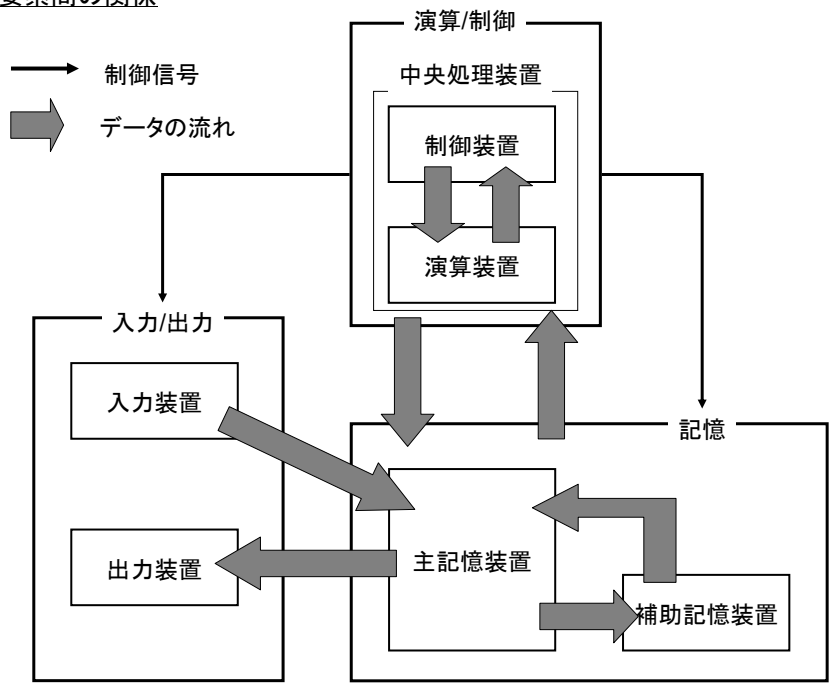
■　コンピュータの 5 大要素

コンピュータを構成する要素には以下の 5 つがあり、これらをコンピュータの 5 大要素と呼んでいます。

要素	代表的な装置	説明
入力	入力装置	プログラムやデータをコンピュータで処理できる表現に変換
出力	出力装置	コンピュータ外部に表示または印刷
演算	中央処理装置（CPU）	コンピュータの中枢部。主記憶装置からプログラムやデータを取り出して計算や条件判断を実行
制御	中央処理装置（CPU）	各装置が総合的に動作するように操作・調整
記憶	主記憶装置（メモリ）	実行中のプログラムやデータを記憶
	補助記憶装置	プログラムやデータを記憶

この 5 つの要素が文字通り、中央処理装置を中心に構成されることで、コンピュータとして機能します。

5大要素間の関係



Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

3 - 2 コンピュータを構成するハードウエア

コンピュータのさまざまな種類の部品から構成されていますが、それらは前述した 5 大要素に分類されます。



それでは、各要素を実現する代表的な装置を順に見ていきましょう。

■　入力装置

現在のコンピュータは「プログラム内蔵方式」を採用しているので、プログラムやデータをコンピュータに入力しなければなりません。このためのハードウエアが「入力装置」です。

キーボードやマウス、スキャナやデジタルカメラなどがこの入力装置にあたります。



入力装置は以下の機能を持ちます。

1. コンピュータ外部からの入力信号をコンピュータ本体へ送る。
2. 入力された情報（文字や数値など）をコンピュータで扱える形式に変換する。

代表的な入力装置には以下のものがあります。

名称	説明
キーボード	代表的な入力装置。タイプライタ形式の装置で、文字や数値入力に使用する。
マウス	画面上に表示されるカーソルを操作するための入力装置。近年では、タッチパッドやトラックポイントなど、目的が同じで形状の異なる装置も数多く存在する。
タッチパネル（タッチスクリーン）	表示用のディスプレイが入力装置を兼ねているもの。銀行の ATM や携帯ゲーム機、携帯電話やスマートフォン、タブレット、一部の PC などに利用されている。画面がタッチされたことを圧力で検知する感圧式、静電容量の変化で検知する静電容量式などがある。
イメージスキャナ	写真や絵などの紙面を画像データとして取り込む装置。コピー機と同様の原理で、紙面に光を当て、その反射光で原稿の濃淡を読み取る。
カメラ	CCD や CMOS センサなどを利用して写真や動画などを撮影する。 QR コードを読み取る装置も、カメラである。
マイク	音を電気信号に変換し、音声や音楽などを録音する。

入力装置の多くは人間が操作します。そのため、入力処理はコンピュータの処理速度よりもはるかに低速です。そこで、効率を向上させるためにさまざまな工夫がなされています。

バッファ

バッファは、入力データを一時的に蓄えておくメモリです。多くの場合、入力が完全に完了してからコンピュータの処理が始まります。そのため、いったん入力情報を蓄えておき、データがそろったらまとめて処理を行います。また、CPU がほかの作業をしていても、バッファに格納するようにしておけばデータを取りこぼすことがありません。

割り込み

CPU は接続されているすべての周辺機器を制御しなければなりませんが、その一方で、すべての機器を監視しているわけではありません。例えば入力装置からの入力は、CPU が常に監視しているわけではないのです。そのため、何らかの入力が発生したことを、入力装置から CPU に伝える必要があります。また、実行中のプログラムに何らかの異常が発生したときは、その異常を CPU に伝えなければなりません。

これらの、なんらかのきっかけで CPU に働きかけを行う処理を「割り込み」といいます。

キーボードやマウスの場合は頻繁に割り込みが発生します。しかしキーボードやマウスは CPU の処理速度から見るとはるかに低速なので、CPU に大きな負担がかかるわけではありません。入力装置によってはバッファが満杯になりそうになったときだけ割り込みが発生するものもあります。

この割り込みには、ハードウエアによる「外部割り込み」と、プログラムによる「内部割り込み」があります。

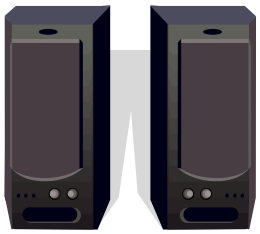
割り込みの種類		発生のきっかけ
外部割り込み		実行中のプログラムとは関係なく発生
	機械チェック割り込み	・電源異常、主記憶装置の障害、周辺機器の障害で発生
	入出力割り込み	・入出力装置や入出力チャネルの動作の中断、完了で発生
	タイマ割り込み	・タスクの終了に割り当てられた時間が経過したとき
	コンソール割り込み	・ユーザが割り込み用の操作をしたとき
内部割り込み		実行中のプログラムが原因で発生
	プログラム割り込み	・0 除算、桁あふれ、実行中のプログラムがエラーを起こしたとき ・仮想記憶のページフォールトなど
	スーパバイザコール割り込み	・タスクなどの切り替えを行ったとき ・モードの切り替え命令を行ったとき

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

■　出力装置

コンピュータによって処理された結果を、コンピュータの外部へ出力するための装置です。実際には、2進数信号を、文字や数字、音声といった人間が理解できる形に変換する作業も行います。

ビデオモニタ、スピーカ、プリンタなどはこの出力装置にあたります。


表示には、ディスプレイ画面のような一時的な表示である「ソフトコピー」と、印刷物のように半永久的な表示である「ハードコピー」とがあります。ソフトコピーを出力する装置の代表はディスプレイで、ハードコピーを出力する装置の代表はプリンタです。

そのほか広義には、エラーがあったことを告知するランプも出力装置の一種です。さらに、ブザーや合成音声装置、携帯電話のバイブレーター（振動発生装置）なども出力装置といえます。

ここでは、代表的な出力装置である、ディスプレイとプリンタについて説明します。

名称	説明
液晶ディスプレイ	液晶（液体と結晶の両方の性質を持つ物質で、電圧をかけると分子の配列が変わり、それによって光の反射率や透過率が変わる性質を持つ）を使用したビデオモニタ。液晶自体には発光能力が無いためバックライトを利用して光を表現している。薄くて消費電力が小さく、結果として発熱も小さい。
有機 EL ディスプレイ	有機物を使った電子発光体（エレクトロルミネッセンス）を用いたビデオモニタ。電圧をかけると、自ら発光する素子を使用している。液晶に比べ高速・高コントラスト・高画質・低消費電力。液晶ディスプレイに変わる、次世代のディスプレイとして注目されている。
プロジェクタ	液晶や DLP（Digital Light Processing）、LCOS（Liquid crystal on silicon）などの技術を使い、画像や映像を大型スクリーンに投影する形式で表示するビデオ出力デバイス。通常プロジェクタをディスプレイと表現することはまずないが、プロジェクタもディスプレイ装置の一種である。明るさの単位としてルーメン（lm）を用いることが多い。ルーメンとは光源から放たれる光の強さの単位であり、実際のスクリーン上の明るさのことではない。プロジェクタがスクリーンから離れるほど大画面になるが、そのぶん画像や映像は暗く見える。

名称	説明
ドット・インパクト・プリンタ	細いピンを並べた印字ヘッドをインクリボンにたたきつけて印字する。 複写式の伝票や銀行の通帳などへの印字に用いられる。複写紙に印字できる長所がある反面、印字音が大きく、解像度も低いという欠点がある。
レーザ・プリンタ	レーザを使って帯電ドラムに照射し、トナーを用紙に付着させ、その後熱をかけて定着させる。ビジネス用途では最もよく用いられているプリンタ。
インクジェット・プリンタ	インクを紙に噴きつけて印字する。レーザ・プリンタと比べると印刷速度や性能が低いが、カラーで印字しても安価で、写真印字に向いている。
サーマルプリンタ	熱に反応する特殊な用紙を用い、その用紙に熱を加えることによって印字する。トナーやインクリボンを用いる必要がないので、プリンタを小型化することができる。レジやタクシーのレシートを印字する時などに用いられる。
3D プリンタ	立体物を表すデータをもとに空間に樹脂などを何層にも積み重ね、立体の造形物を作成する。実際にはプリント（印刷）しているのではないが、便宜的にプリンタという用語を用いる。

出力装置にも以下のような効率を向上させる工夫があります。

スプール

出力装置へデータを送信する場合は、CPU がメモリから出力装置へ転送しますが、メモリ→CPU 間とCPU→出力装置間ではデータの転送速度に大きな差があるため（CPU→出力装置間のほうがはるかに遅い）、一度に大量のデータを転送しては、出力装置がデータを取りこぼしてしまいます。また、出力装置がすべてのデータを受け取るまで CPU が待つのでは、全体の処理速度が出力装置のスピードに引っ張られて遅くなってしまいます。

そのため、出力データを一時的にハードディスクにためるしくみが必要となります。それがスプールです。スプールは、主としてプリンタのために使われます。

スプールは後述するキャッシュと似ていますが、スプールは「後で処理をするために今は一時的にデータをためておく」という目的で使われるものをさします。

CPU は、スプールへの書き込みを完了した時点で印刷終了とみなし、次の作業を行います。実際に印刷をするのは「スプーラ」というソフトウエアです。

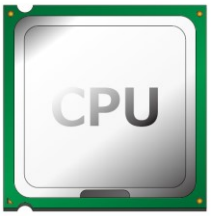
ポーリング

プログラムが一定の周期で周辺機器を監視することを「ポーリング」といいます。例えばプリンタは、紙切れを起こすと「印刷不可」の状態になります。前述のスプーラは一定間隔でプリンタを監視し（ポーリングを行い）、印刷可能になった時点であらためてプリンタに印刷データを送ります。

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

■　中央処理装置

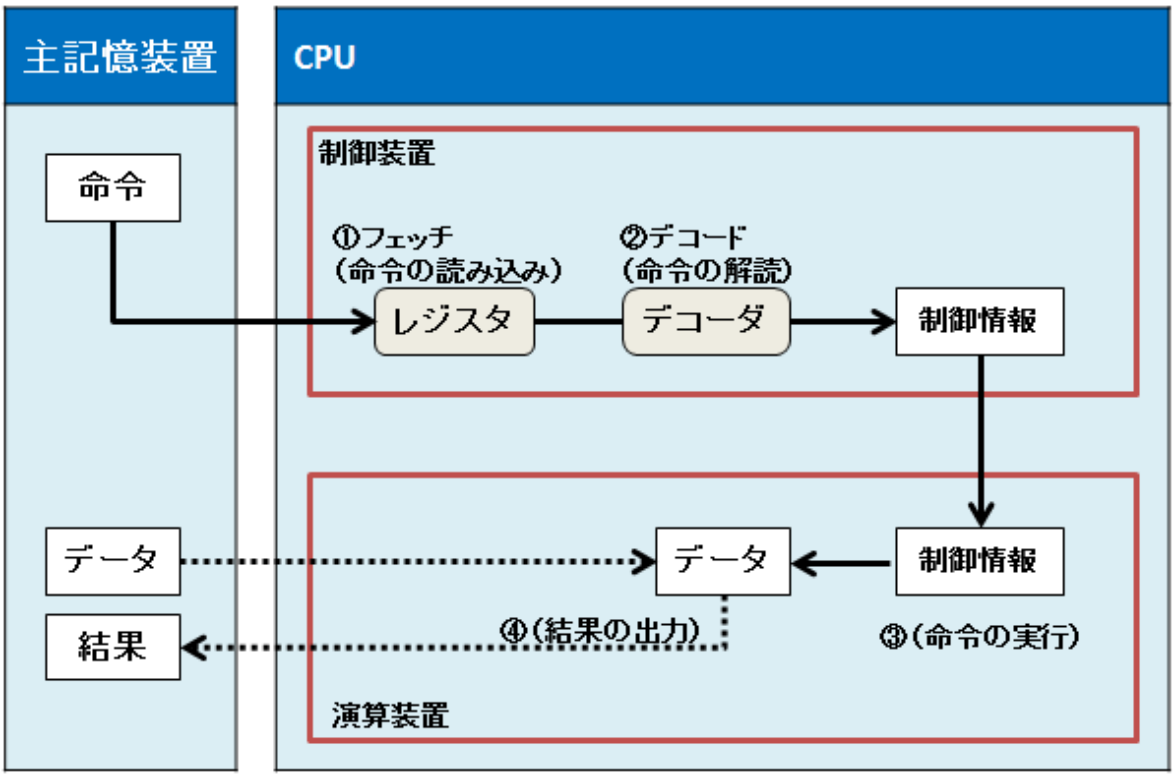
中央処理装置（CPU：Central Processing Unit）は、文字通り、コンピュータの中心的機能である演算と制御を担当します。



CPU のしくみ

CPU の構成要素とその実行の流れを下図で説明します。

※実際にはもう少し複雑ですが、図と説明を簡素化しています。



制御装置にあるレジスタ（プログラムカウンタ）には、これから実行される命令が格納されています。制御装置はまず、主記憶装置からその命令を読み取って命令レジスタに格納します。命令レジスタとは CPU 内部にある小さなメモリのことで、具体的な命令や演算を示す「命令部」と、処理対象となるデータの主記憶上の場所（アドレス）を示す「アドレス部」からなります。

続いて解読器が命令部を解読し、内容を演算装置に送ります。 次にアドレスレジスタがアドレス部を読み取り、演算の対象となるデータを主記憶装置から読み取ります。

読み取られたデータは汎用レジスタに格納され、ALU（演算回路）によって命令が実行（演算）されます。最後に演算結果が主記憶装置に戻されます。

CPU の主な機能は以下の通りです。

機能	説明
フェッチ（fetch）	主記憶装置から命令を 1 つ取り出し、レジスタに格納する
デコード（decode）	命令の種類を解釈し、その命令の具体的な指示内容を解読する
実行（execute）	命令を実行し、各装置や回路が動作する （命令には算術演算や論理演算、条件判断などがある）
ライトバック（write back）	命令の実行結果として得たデータを、主記憶装置に書き戻す

CPU では、上記のような一連の動作が命令の実行単位となっています。これを「基本サイクル」と呼びます。CPU には、この基本サイクルを動作させるための回路が用意されています。CPU は、この基本サイクルを繰り返すことにより、処理を進めていきます。

CPU 内部の高速化

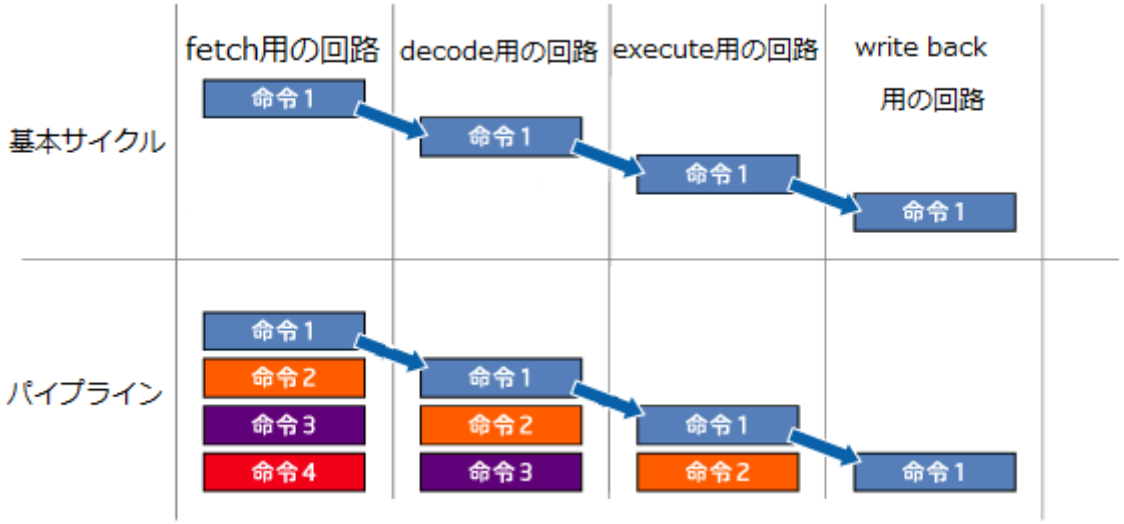
CPU は基本サイクル 1 回を 1 つの命令として実行しますが、その基本サイクルを高速化するための手法がいくつかあります。

■ 基本サイクルを短くする

CPU は「クロック周波数」と呼ばれる一定間隔の信号と同期して動作します。以前はクロックを速くすれば速度が向上するということができましたが、現在は一概にそうは言えなくなりました。これは、CPU の性能向上のために複数の技術を組み合わせているためです。

■ 流れ作業的に基本サイクルを実行する

基本サイクルを構成する動作を流れ作業的に動作させても速度が向上します。このようなしくみを「パイプライン」といいます。パイプラインを効果的に使うには、重なり合った命令同士がお互いに影響しないようにしなければなりません。



（インテル・ミュージアム：http://www.intel.co.jp/jp/intel/museum/mpuworks/page_2.htm の図を元に作成）

■ 同時にいくつかの基本サイクルを実行する

同時に複数の基本サイクルを実行すれば速度が向上します。例えば、2 つのサイクルを同時に実行できれば 2 倍の速度になります。このようなしくみを「スーパースカラー」といいます。

さらには、1 つの CPU にフェッチ、デコード、実行、ライトバックを行う回路を 2 つ以上設けるような技術もあります。そのような技術を用いて作られた CPU のことを、マルチコア CPU とよびます。1 つの CPU に上記の回路を 2 セット組み込んだものをデュアルコア、3 セット組み込んだものをトリプルコア、4 セット組み込んだものをクアッドコア、8 セット組み込んだものをオクタコアと呼んでいます。

CPU の高速化を助ける技術

CPU は、その他の装置と比べると、その動作速度が非常に高速です。逆に CPU 以外の装置は、CPU に比べてはるかに低速に動作します。

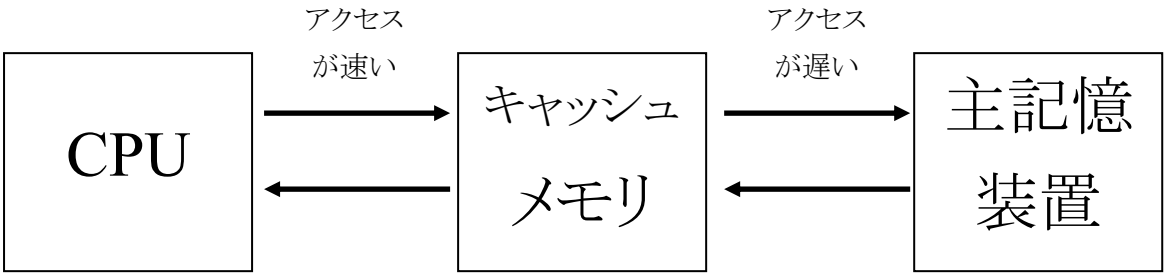
CPU が最も頻繁にやりとりを行う装置が、後述する主記憶装置です。CPU は、自分と比べると低速に動作する主記憶装置によって実行速度を下げてしまわないよう、さまざまな工夫をこらしています。

■ キャッシュメモリ

CPU と主記憶装置との間に、主記憶装置よりも高速にデータを読み書きできる記憶装置を用意します。その記憶装置のことを、キャッシュメモリといいます。キャッシュメモリは、主記憶装置から読み込んだ命令やデータを一時的に記憶しておきます。

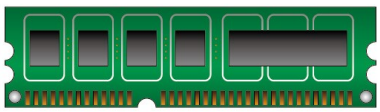
CPU が同じ命令やデータを主記憶装置から読み出す場合、わざわざ主記憶装置にそのデータを取りにいかなくても、キャッシュメモリから読み出すことができるため、高速にデータの読み込みが可能です。

また、演算結果をライトバックする際も、キャッシュメモリにそのデータが書き込まれた段階でライトバックが終わったとみなします。キャッシュメモリから主記憶装置への書き出しは後からゆっくり行われます。こうしてライトバック時にもキャッシュメモリを使用し、CPU の動作を妨げないようにしています。



■ 主記憶装置

情報を利用する(アクセスする)場合、必ず主記憶装置に格納されていなければいけません。いいかえれば、CPU が直接アクセスできるのは主記憶装置だけです。一般に、主記憶装置のことを**メインメモリ**と呼びます。



通常、主記憶装置に格納できる最低単位はバイトです。主記憶装置は、1 バイトごとに「番地(アドレス)」と呼ばれる通し番号で管理されます。CPU は、番地を指定して主記憶装置のデータを読み書きします。

ただし、多くの CPU はバイト単位で主記憶装置にアクセスすることはできません。
CPU によって異なりますが、一般的にはワードが最低単位です(多くは 32 ビットまたは 64 ビット)。そのため、1 バイト(8 ビット)だけを使う場合も 4 バイト(32 ビット)を一度に使う場合も速度は変わりません。
このワード単位でデータのアクセスを行うことから、CPU の性能を表す言葉として、32 ビット CPU や 64 ビット CPU という用語を使用することがあります。

メモリアクセスの高速化

前述の通り、主記憶装置は CPU に比べて大変動作速度の遅い装置です。そのため少しでもメインメモリへのアクセスの速度を向上させるために、いくつかの技術が確立されました。

■ 基本サイクルを短くする

CPU 同様、メインメモリも「クロック」と呼ばれる一定間隔の信号と同期して動作します。クロックを速くすれば、速度が向上します。

■ プリフェッチ機能を持つ

プリフェッチとは、CPU が命令を取り出すフェッチ動作を判断して、CPU が実際にフェッチ動作を行う前にメインメモリの内容を先読みしておく機能です。通常、CPU がフェッチしたメモリ領域の 2～8 バイト後ろを同時に読み込むことで、メモリから CPU にデータを渡す処理を高速化しています。

■ メモリインターリーブ

主記憶装置を独立にアクセス可能な「バンク」と呼ばれる単位に分割して、各バンクを別々のバスで CPU とつなげ、それぞれのバンクを順次並行してアクセスできるようにして高速化を図るしくみです。通常、同じサイズの主記憶装置を 2 枚セットで装着することで実現しています。

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

メモリの種類

主記憶装置は「半導体メモリ」を使って実現します。半導体メモリは、主記憶装置に必要なランダムアクセス性があり、自由に読み書きできるからです。

主記憶装置はその性質によって、揮発性（電源を落とすとデータが消えてしまう）ものと、不揮発性（電源を落としてもデータが消えない）のものに大別されます。

RAM（Random Access Memory）

データの読み書きが可能ですが、コンピュータの電源を切ると記憶していた内容がすべて消えてしまう記憶装置です。RAM は揮発性なのでデータの永続記憶ができません。

RAM はさらに DRAM（Dynamic RAM）と SRAM（Static RAM）に大別されます。

違い	DRAM	SRAM
動作	遅い	速い
リフレッシュ	必要	不要
集積度	高い	低い
価格	安い	高い
利用部分	メインメモリ	キャッシュメモリ

ROM（Read Only Memory）

もともとは、製造時にあらかじめデータやプログラムを書き込んでおき、読み取り専用として利用する記憶装置としての呼称でした。現在では、RAM と対比して「電源を切っても記憶内容が消えないメモリ」という意味で用いられています。ROM にはいくつか種類があり、ROM ライタという装置を使って、内容を一度だけ記憶させることができる PROM（Programmable ROM）と、紫外線照射によって何度でも書き換え可能な EPROM（Erasable PROM）、これを電気的に行う EEPROM（Electrically EPROM）があります。補助記憶装置として使用するフラッシュメモリ（SD カードや USB メモリなど）も EEPROM の一種です。

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

■　補助記憶装置

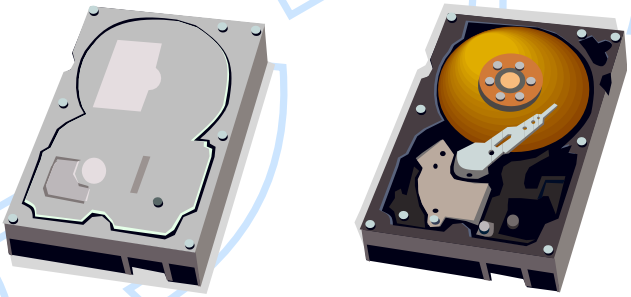
主記憶装置として用いられている DRAM は、コンピュータの電源を切るとその内容を保持することができません。また、扱うすべての命令やデータを主記憶装置に格納しておくには容量が足りません。そこで、大容量の命令やデータを永続的に記録しておくために補助記憶装置が必要となります。一般に、補助記憶装置のことを**ストレージ**と呼びます。

補助記憶装置は、情報を自由に読み書きできるものと、書き換え不可能なものに分かれます。どちらの場合も、コンピュータの電源を切ってもデータを永続的に記録しておくことが可能です。

補助記憶装置は、プログラムやデータを記憶するための記憶媒体（メディア）と、その内容を読み書きするための装置（ドライブ）から構成されます。これは、CD と CD プレーヤー、DVD と DVD ビデオレコーダなどの関係にあたります。

代表的な補助記憶装置には、以下のようなものがあります。

ハードディスクドライブ



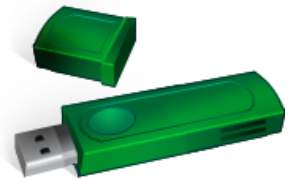
ハードディスクドライブはハードディスク（HDD：Hard Disk Drive）ともいいます。ハードディスクはかつて、パーソナル・コンピュータから大型コンピュータまで、全てのコンピュータで主要な作業用補助記憶装置として幅広く普及していました。ハードディスクは、磁気を帯びた磁性体でコーティングされた金属製、またはガラス製の円盤（ディスク）を複数枚重ね合わせたもので、ディスク表面に塗られた磁性体の磁気の向きによってデータを表現します。

ディスクに記録された磁気情報は、ヘッドと呼ばれる小型の装置で読み取ります。ヘッドは、高速に回転するディスクの風圧を利用してわずかに浮いています。浮いている間隔は髪の毛やタバコの煙 1 粒よりも狭いため、わずかな振動でもヘッドがディスクに接触し、壊れてしまいます。したがって、ハードディスクは動作中に振動させることに大変弱い、という欠点があります。普通、「ハードディスク」および「ハードディスクドライブ」という呼び方はどちらも同じ意味で使われます（英語では「Hard Drive」といいます）。

ハードディスクの長所は、安価、高速、高記憶容量です。大量のデータをアーカイブ（長期保存）するのに向いています。記憶容量はさまざまですが、通常、数百ギガバイトから数テラバイト程度です。

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

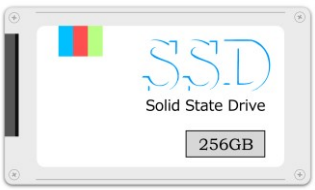
フラッシュメモリ

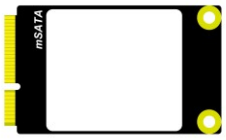




フラッシュメモリは前述の EEPROM の一種で、電源を切っても記録内容が消えないメモリです。手軽な補助記憶装置として、デジカメや携帯電話などにも利用されています。その形状の違いから、SD カード、CFexpress などの種類があります。

フラッシュメモリのうち、USB 規格に従った入出力機能を備えた製品を特に「USB メモリ」と呼びます。USB メモリは、低容量で着脱が極めて簡単なハードディスク装置の代わりとして扱えます。







SSD

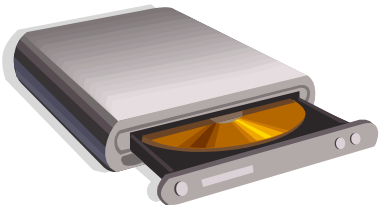
フラッシュメモリの中で、容量をさらに数百ギガバイトから数テラバイト程度まで増やし、ハードディスク装置と同じインターフェースで利用可能にしたものを SSD（Solid State Disk または Solid State Drive）と呼びます。「Solid State」とは半導体の意味です。ハードディスクよりも高価ですが、高速に動作すること、ハードディスクよりも消費電力が低いことなどから、急速に普及が進んでいます。

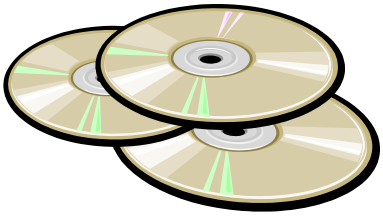
ハードディスクドライブと比較した、SSD の利点と欠点は以下の通りです。ただし技術革新により、SSD の欠点は徐々に改善しつつあります。

	ハードディスクドライブ	SSD
利点	・大容量のものが安価で入手できる	・ハードディスクドライブに比べて高速 ・消費電力が低い ・衝撃に強い（駆動部分がないため）
欠点	・SSD に比べて低速 ・消費電力が高い ・衝撃に弱い	・ハードディスクドライブよりもやや高価 ・書き換え可能回数がハードディスクドライブに比べて少ない

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

CD、DVD、BD などの光ディスクドライブ





合成樹脂の円盤にレーザ光線を当て、その反射の違いでデータを読み取る装置を「光ディスク」といいます。光ディスクは、読み取り専用、1 回のみ書き込み可能、読み書き可能なものと、さまざまな種類があります。

光ディスクの代表的なものは、CD、DVD、そして Blu-ray Disc（BD）です。これらのメディアは透明なプラスチック版にアルミの蒸着面を挟んであり、書き込まれた情報はピットと呼ばれるアルミ面の凹凸で表現されています。ドライブはこの凹凸にレーザ光を当て、その反射を検知することでデータを読み取ります。ピットの列のことをトラックといいます。トラックのピッチの細かさの違いと、データを圧縮して格納する技術の違いなどで、実際の記録できるデータの容量が変わります。

メディアは次の 3 つに分かれます。

・読み取り専用のもの
・追記のみ可能なもの
・書き換え可能なもの

読み取り専用のものは、出荷時に機械的にプレスすることにより、凹凸をつけています。
追記のみ可能なものは、金属に直接凹凸をつけるのではなく、記録したい部分の色素を焼き切ることでデータを記録します。CD や DVD を「焼く」という言葉はここからきています。
書き換え可能なメディアは、レーザ光を当てて加熱することで変質させたり、元に戻したりすることのできる特殊な金属を記録面に塗っています。また、記録するためには対応したドライブと、専用のソフトウエアが必要です。

その他、それぞれの容量は、次ページの通りです。

BD の正式名称は「Blue-ray」ではなく、「Blu-ray」です。これは「Blue-ray Disc」とすると、英語圏の国々で「青色光（で読み取る）ディスク」を意味する一般名詞と解釈される可能性があることと、商標として登録が認められない可能性があるためとで、わざと「Blu」という名称を使っています。

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

	CD	DVD	Blu-ray Disc	Ultra HD Blu-ray
容量	8cm 180MB 12cm 650MB または 700MB	8cm 1.4GB （片面 1 層） 2.6GB （片面 2 層） 12cm 4.7GB （片面 1 層） 8.5GB （片面 2 層）	8cm 7.5GB（1 層） 15GB（2 層） 12cm 25GB（1 層） 50GB（2 層） 100GB（3 層） 128GB（4 層）	12cm 50GB（2 層） 66GB（2 層） 100GB（3 層）
メディア	読み取りのみ CD-ROM 追記のみ CD-R 書き換え可能 CD-RW	読み取りのみ DVD-ROM 追記のみ DVD-R DVD+R 書き換え可能 DVD-RW DVD+RW DVD-RAM	読み取りのみ BD-ROM 追記のみ BD-R 書き換え可能 BD-RE	（Blu-ray Disc と同じ） ※ Ultra HD Blu-ray の最大の特徴は、4K 放送に対応したこと

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

磁気テープドライブ



原理的には音楽用のオーディオカセットと同じく磁気テープを使ってデータを記録する装置です。主として移送用、バックアップ用、長期保存用に使われます。磁気テープの長所は、非常に低コストで大量のデータを記憶できることですが、短所は、データを取り出す時にテープの最初から順に読まなければならないことです。

コンピュータ用として用いられている規格には、DAT（Digital Audio Tape）、DDS（Digital Data Storage）、DLT（Digital Linear Tape）、LTO（Linear Tape-Open）などがあります。いずれもテープ 1 本あたり、数百メガバイトから数百ギガバイトまでの容量を持ちます。

磁気テープ装置は、データのバックアップ用途としてはハードディスクにその座を明け渡しつつあります。しかし、法律や内部統制などの理由で財務データや顧客取引データなどを長期保存する目的として、引き続き利用されています。そのような用途では、一度記録したデータを二度と消去したり書き換えたりできないようにして、物理的に改ざんできないような状態でデータを長期保存することが求められます。そのような、一度記録したデータを二度と消去したり書き換えたりできないようなメディアのことを **WORM**（Write Once Read Many）といいます。

コラム：ゼロ・スピンドル PC

「スピンドル（spindle）」とは、回転軸のことです。モーターを用いて何かを回転させているもの全般を指します。具体的にはハードディスクドライブ、CD や DVD などの光ディスクドライブ、CPU や PC 内部を冷却するためのファンなどが、スピンドルを持つ機器です。スピンドルを持つ機器は、物理的に回転する場所の故障率が高い、省電力設計に向かない、振動や騒音が発生する、といった欠点があります。

小型のノート PC やタブレットには、こうしたスピンドルを持つ機器が用いられなくなりました。まず光ディスクドライブが、次に CPU の省電力化、低発熱化が進んで冷却ファンがなくなりました。昨今では、ハードディスクドライブの代わりに SSD が搭載されている PC も多く見受けられるようになりました。こうして、スピンドルを持つ部品をまったく持たない PC（ゼロ・スピンドル PC、またはスピンドルレス PC）が作られるようになりました。ゼロ・スピンドル PC の技術が発達した現代では、小型ノート PC とタブレットとの区別は難しいものになっています。

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

■　その他の装置

コンピュータ・システムを構成するその他の装置を紹介します。

有線 LAN コントローラ

コンピュータ・システムを有線 LAN に接続するための装置です。有線 LAN コントローラは、LAN ケーブルを通ってやってきた信号を受信すると、まず自分宛の信号であるかどうかをチェックします。自分宛でない信号は破棄し、自分宛の信号であれば、その信号を CPU に送ります。逆に CPU からの命令を受け、データを有線 LAN に乗せることのできる電気信号に変換して、対象となるコンピュータ・システムに伝送するという役割も果たします。

無線 LAN コントローラ

コンピュータ・システムを無線 LAN に接続するための装置です。コンピュータ・システムに内蔵するタイプの他、後述する USB を用いて、無線 LAN 機能のないコンピュータ・システムに後から増設するタイプのものもあります。

コンピュータ・システムに後から増設するタイプのものは、無線 LAN アダプタといいます。



GPU

Graphics Processing Unit の略で、CPU から画像描画の命令を受け、その処理を専門に司るプロセッサです。大量のグラフィックデータを並列に演算処理し、その結果をディスプレイに伝送します。画面描画という比較的負荷のかかる処理を CPU から解放する目的で作られました。近年では、大量のデータを並列処理できるという特性から、AI が機械学習を行うために使われるようになりました。



TPU

Tensor Processing Unit の略です。Google 社が開発した、大規模な機械学習や深層学習を高速に行うために設計された専用のプロセッサです。AI は、学習や推論のために大量の演算を何億回も繰り返します。それを CPU や GPU に比べて効率的に処理できるように作られています。また、ネットワーク上で何千個もの TPU を連携させることで、巨大な AI の仕組みを構築することも可能です。Google 社の AI モデルである Gemini は、TPU 上で動作していると考えて差し支えありません。

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

■　バス（伝送路）

マザーボード上で CPU は、多くの周辺装置と通信を行います。この通信を効率よく行うため、コンピュータ・システムでは周辺機器同士で共通の通信回路を利用します。これを「バス」といいます。

コンピュータ・システムは CPU を中心に、その他の周辺装置が「バス」で接続されます。「バス」はその目的と中を流れるデータの量や種類などにより、いくつかの種類が用意されています。

最も高速に動作するのが、CPU とメモリとをつないでいるバスです。このバスを「CPU バス」と呼びます。CPU バスにはメモリ以外にも、特に高速な動作が必要な周辺装置が接続されています。

ストレージを接続するバス

規格名	特徴
Serial ATA	SATA という場合もあります。パーソナル・コンピュータにおいて、ハードディスクや SSD、CD/DVD などを接続するバスです。保証帯域幅は最大 6Gbit/s。
Serial Attached SCSI	SAS という場合もあります。サーバ機において、ハードディスクや SSD などを接続するために用いられています。SATA よりも高速に動作します。保証帯域幅は最大 24Gbit/s。
InfiniBand	高信頼性が求められる、データセンターのサーバで用いられます。複数のレーン（最大 12 レーン）を束ねてさらに高速な帯域を確保することができます。保証帯域幅は 1 レーンあたり最大 100Gbit/s。

内蔵型周辺機器を接続するバス

規格名	特徴
PCI Express	コンピュータ・システムに内蔵しているグラフィック・コントローラよりもさらに高速なグラフィック・コントローラを接続したり、有線 LAN や USB などを後から増設したりするために用いるバスです。
Mini Serial ATA	mSATA という場合もあります。SATA 規格の一種ですが、通常の SATA よりも小型で、コンピュータのマザーボードに SSD などを直接接続する形で使います。最大帯域幅は 6Gbit/s。
M.2	エムドットツーと読みます。他のバスに比べて非常に小さいのが特徴で、小型ノートパソコンやタブレットなどに無線 LAN や SSD などを搭載することが可能です。NVMe 規格と SATA 規格とがあり、最大帯域幅は 40Gbit/s（NVMe）。

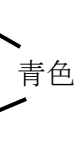
また、追加の周辺装置を接続するためのバスもあります。このバスを「入出力バス」と呼びます。
現在主流の入出力バスには、次のようなものがあります。

USB（Universal Serial Bus）

規格名（旧名称）	伝送速度	最大伝送距離
USB 1.0、1.1	12 Mbit/s	5m
USB 2.0	480 Mbit/s	5m（Type-C は 4m）
USB 3.2 Gen 1（USB 3.0、USB 3.1 Gen 1）	5 Gbit/s	2 ～ 3m
USB 3.2 Gen 2（USB 3.1 Gen 2）	10 Gbit/s	1m
USB 3.2 Gen 2×2	20 Gbit/s	1m
USB4 Version 1.0	最大 40 Gbit/s	40Gbps の場合 0.8m
USB4 Version 2.0	最大 80 Gbit/s	0.8m
USB4 Version 2.0 （拡張モード）	最大 80Gbit/s（双方向） 120Gbit/s（片方向）＋ 40Gbit/s（逆方向）	0.8m

USB のプラグ

規格名	形状	
	USB 2.0	USB 3.x
Type－A		
Type－B		
Type Micro B（Micro-B）		
Type－C		



※USB4 のプラグは Type-C を使用することが必須となっている。

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

Bluetooth

規格名	特徴
Ver.3.0 以前	古いバージョン。現行ではほぼ利用されない。
Ver.4.0	従来のバージョン《BR（Basic Rate）/EDR（Enhanced Data Rate）》に加えて、従来のバージョンとは異なる方式で大幅な省電力化を実現する低消費電力モード BLE（Bluetooth Low Energy）に対応。通信速度は 1Mbps。家電製品同士の通信など、小さなデータを頻繁に通信するのに向いている。ホスト側は BR/EDR と BLE の両方に対応するデュアルモードを実装できる。
Ver.4.1	Ver.4.0 に、モバイル端末向け通信サービスの電波との干渉を抑える技術、 データ転送の効率化、ホストにもクライアントにもなれる機能、などを追加した。
Ver.4.2	BLE の通信効率を向上し、IPv6 でインターネット接続できるようにした。
Ver.5.0	BLE の通信速度として 2Mbps、1Mbps、500Kbps、125kbps をサポート。
Ver.5.1	Bluetooth 機器の方向を探知する機能を追加した。
Ver.5.2	より低消費電力かつ高品質かつマルチストリーム対応の LE Audio 規格を追加した。
Ver.5.3	接続制御を効率化するとともに、消費電力の低減化を実現した。
Ver.5.4	店舗において、棚札の価格表示変更が簡単に行える 電子棚札（Electric Shelf Label：ESL）に対応した。

また、Bluetooth は、Class やプロファイルなどの情報も比較、検討する必要があります。Class は電波の強度を示す概念で、Class1 は 100 メートル、Class2 は 10m、Class3 は 1m の電波到達距離があります。一方プロファイルは機器の種類ごとに策定されているプロトコル（通信規約）のことで、マウスやキーボード用の HID、マイクやスピーカ用の HSP、ハンズフリー通話用の HFP などのプロファイルがあります。

NFC（Near Field Communication）



NFC とは、近距離の無線通信技術を統一化した規格の名称です。
非接触 IC カードに用いられたり、物流の商品タグに利用されたりするほか、Bluetooth 機器のハンドオーバー（機器の切り替え）や無線 LAN アクセスポイントに最初に接続する際などにも用いられます。
キャラクターフィギュアをゲーム機にタッチすることで、ゲームとキャラクターフィギュアとが連動する仕組みなどにも応用されています。

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

第4章 ソフトウエア

サンプル

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

4 - 1 ソフトウエアとは

JIS 規格（日本工業規格：Japanese Industrial Standards）によれば、「ソフトウエア」とは、



「データ処理システムを機能させるための、プログラム、手順、規則、関連文書などを含む知的な創作」

であると定義しています。

つまり、広い意味でのソフトウエアとは、コンピュータに与えるプログラムだけではなく、プログラムの使い方やマニュアル、その利用規約等までを含むと考えられています。

しかし、ゲームソフトやワードプロセッサといったような日常的な使い方では、「ソフトウエア」は「プログラム」と同じ意味で使われています。ここでは、「ソフトウエア」を「プログラム」と同義と考え、以下の 3 つに分類します。

- 基本ソフトウエア
- ミドルウエア
- 応用ソフトウエア

■　ハードウエアとソフトウエア

プログラムはデータと同様、コンピュータが扱う情報として記憶されます。それは電子的な情報に過ぎず、実体を持ちません。

コンピュータ本体やその周辺機器など、実体を持つものをハードウエアと呼びます。それに対してプログラムやデータのことをソフトウエアと呼びます[5]。ソフトウエアは、ハードウエアと対比させるために後から作られた造語です。

[5] ハードウエア製造時に、ROM を使ってハードウエア内部にあらかじめ組み込まれているソフトウエアもあります。そのようなソフトウエアを、特にファームウェアと呼ぶこともあります。

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

4 - 2 基本ソフトウエア

コンピュータを利用する場合、その目的に応じたソフトウエアが不可欠です。しかし、多くのソフトウエアで似たような機能が必要となります。例えばデータをファイルとして補助記憶装置に保存する機能や、ファイルとして保存しているデータを読み取る機能、そのファイルをコピーしたり削除したりする機能、あるいはデータを印刷する機能などは、その代表です。これらの機能をソフトウエアごとに作成していると非効率的です。また、そのソフトウエアをコンピュータで実行するためには、キーボードやマウス、ディスプレイ、ハードディスクなどのさまざまなハードウエアを制御する必要があります。

ハードウエアを制御し、かつコンピュータ上で実行するすべてのソフトウエアに必要な機能を提供するためのソフトウエアを、特に「基本ソフトウエア」といいます。

■　基本ソフトウエアの種類

基本ソフトウエアは、大きく以下の 3 種類に分けられます。

- オペレーティング・システム
- ユーティリティ
- 開発環境

■ オペレーティング・システム

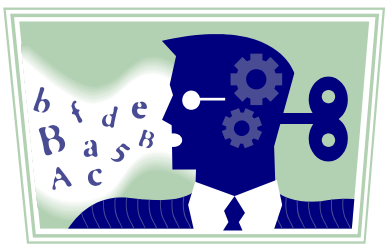
コンピュータが動作するための基本的な機能を提供するソフトウエアのことを、オペレーティング・システムといいます。一般には、省略して OS と呼びます。OS はコンピュータの電源を入れた時に最初に主記憶装置に読み込まれ、コンピュータが停止するまで実行を続けます。

オペレーティング・システムの役割は、コンピュータを効率よく利用するための機能を提供することです。

OS の主な目的は、以下の通りです。

役割	説明
タスク管理	複数のアプリケーションを同時に実行するためのしくみを提供する
メモリ管理	利用可能なメモリをプログラムに割り当てたり、使用が終わったメモリを回収したりする
資源管理	プログラムが必要とする各周辺装置を資源として割り当て、それらの制御を行う
ファイル管理	基本ソフトウエアや応用ソフトウエアが使用するファイルを、補助記憶装置のしくみや物理的な格納領域を意識することなく使用できるようにする

サンプル

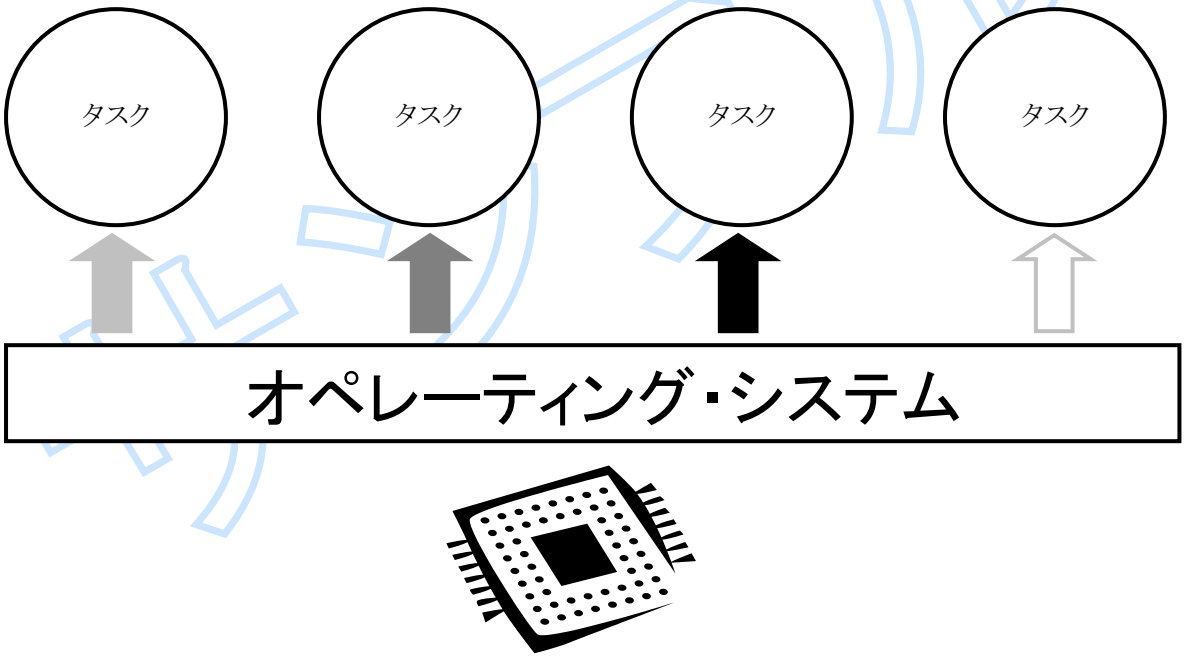


タスク管理

「タスク」とは、コンピュータから見たプログラムの動作単位のことです。人間の立場からみたひとかたまりの動作単位のことを「ジョブ」と呼ぶ場合もあります。一度に 1 つしかタスクを起動できないような OS を「シングルタスク OS」といい、複数のタスクを同時に起動できる OS を「マルチタスク OS」といいます。

マルチタスク OS の場合、それぞれのタスクは「CPU を利用する権利」を OS から与えられて動作します。ある一瞬に、CPU 利用権を持つタスクは 1 つだけです。OS から CPU を利用する権利を与えられたタスクだけがプログラムを動作させられます。CPU を利用する権利はやがて、別のタスクに移ります。次に CPU 利用権を持ったタスクが、プログラムを実行できるようになります。

このタスク間の CPU 利用権の切り替えは、人間が判断できないほどの早い時間で行われます。そのため人間からすると、複数のプログラムが同時に動作しているように見えるのです。

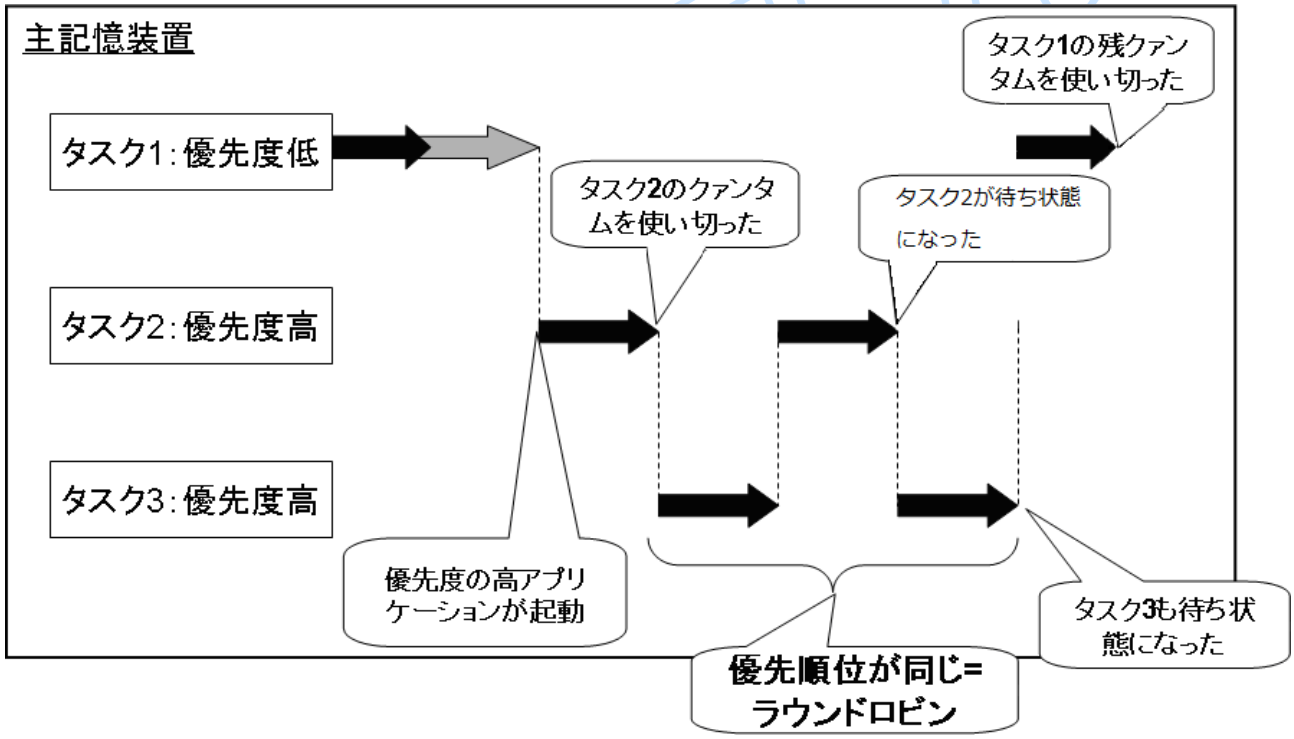


マルチタスクは、次のようなしくみで実現しています。

まず、あらかじめ OS がそれぞれのタスクに「優先度」を割り当てます。優先度の高いタスクほど先に実行されます。また、タスクごとに「CPU を利用できる時間」が割り当てられます。このタスクごとに割り当てられる CPU 利用可能時間のことをクァンタムといいます。一般に、この時間は数ミリ秒から数百ミリ秒です。

各タスクは、自分に与えられたクァンタムの分だけ CPU を占有し、プログラムを実行します。与えられたクァンタムを使い切ると、OS はそのタスクから CPU 利用権を奪い取り、同じ優先度を持つ別のタスクに渡します。優先度が同じタスクは、クァンタムごとに順に CPU を占有して動作します。この、順繰りに動作するという動きのことを「ラウンドロビン」と呼びます。

現在 CPU の利用権を持っているタスクよりも優先度の高いタスクが起動すると、OS は優先度のより高いタスクに強制的に CPU 利用権を移します。
また、タスクが何らかの待ち状態になった場合にも CPU 利用権を切り替えます。



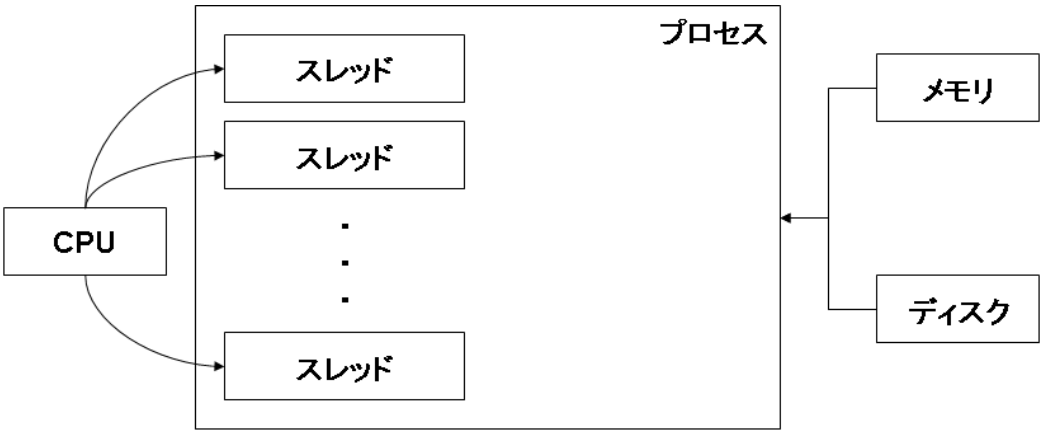
このように OS が各タスクへの CPU の利用権を管理する方式を「プリエンプティブ・マルチタスク」といいます。

プロセスとスレッド

マルチタスクを実現する場合、考慮しなければならないのがプロセスとスレッドという概念です。

プロセスとは、コンピュータ・システム内で動作しているプログラムと、そのプログラムを動作させるのに必要なメモリ領域一式を指します。通常、単一のアプリケーションは単一のプロセスで動作しています。

またスレッドとは、そのプロセス内に存在するプログラムの実行箇所です。

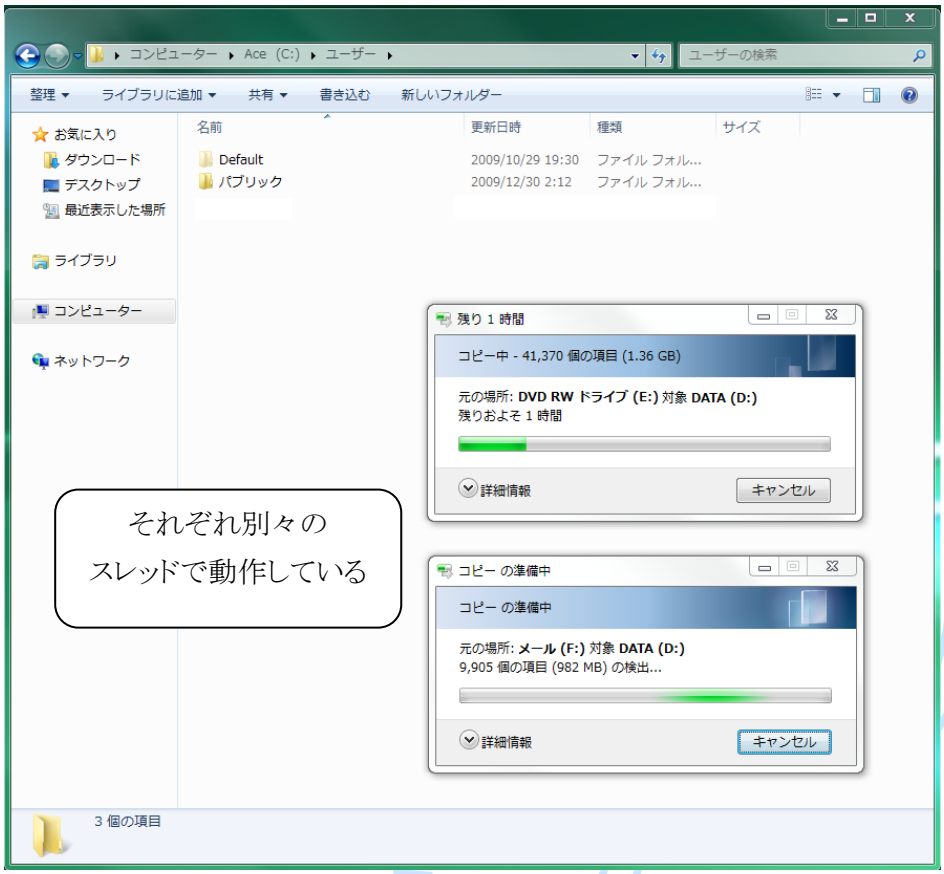


CPUはスレッド単位で割り当てられる
メモリやディスクはプロセス単位で割り当てられる

例えば、Windows にはファイルの操作をするための「エクスプローラ」というソフトウエアがあります。エクスプローラを使用して複数のフォルダを開いても、OS 上で動作している「エクスプローラ」というソフトウエアは 1 つです。OS 上では、エクスプローラを動作させるための「プロセス」が 1 つだけ存在しているのです。

一方、エクスプローラのそれぞれのウィンドウは独立に動きます。あるフォルダにあるファイルをすべて別のフォルダにコピーするという操作を実行しても、その操作が完了するまで待たされるということはありません。あるフォルダのファイルをコピーしている間に、別のフォルダのファイルをコピーすることも可能です。これは、1 つのプロセスに別々のスレッドが存在していて、スレッド単位でマルチタスクを実行しているのです。

OS において、マルチタスクの最小単位はスレッドです。1 つのプロセスには、必ず 1 つ以上のスレッドが存在しています。プロセス上にいくつのスレッドを作ることができるかについては、プログラムを開発した人が決めます。



エクスプローラは、全体が 1 つのプロセスで動作している

●プロセス

プロセスは、以下の要素で構成されています。

- スレッドとその状態
- そのプロセスを動作させたユーザと、そのユーザが持つ権利情報
- メモリを含めた資源の割り当て情報（どの資源をどのように割り当てるかは OS が決める）
- 実行プログラムそのもの

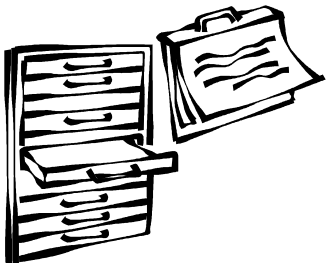
多くの OS では、プロセスごとに内容が保護されています。あるプロセスから別のプロセスの中身を覗いたり、勝手に操作したりすることはできません。プロセス間はあらかじめ決められた方法に従って、あらかじめ設定された操作だけが行えるようになっています。

●スレッド

スレッドはプロセスごとに 1 つ以上存在します。各スレッドは、プログラムのどの部分を実行しているか、という状態を持ちます。これをハードウエアコンテキストといいます。一方メモリなどの資源はプロセスに割り当てられるため、1 つのプロセス内のすべてのスレッドは、同じ資源を共有して使用することになります。

メモリ管理

主記憶装置であるメモリは、1 本の長い帯のような状態であると考えることができます。その長い帯にプログラムやデータを記憶しています。



OS 上では、前述の通り複数のプロセスが動作しています。各プロセスにメモリを割り当てること、各プロセスが使用しているメモリが混乱しないよう、以下のようにきちんと管理することも、OS の重要な仕事です。

メモリ・フラグメンテーション

メモリ・フラグメンテーションとは、分裂、破砕、断片化したものという意味です。

長時間コンピュータを起動してさまざまなプログラムをメモリ領域に読み込んだり、開放したりしていると、連続した大きな空き領域が次第に少なくなり、データが小さな断片に分割されて新たに読み込まれるようになります。これを「メモリ・フラグメンテーション」といいます。パフォーマンスに影響を与えることがあります。

メモリ・コンパクション

OS などがメモリの占有領域の再配置などを行うことによって、メモリ・フラグメンテーションを解消することをいいます。

オーバーレイ

実行するプログラムがメモリ領域に入りきらないほど大きいときに使われる方法です。これは、プログラムの実行にはプログラム全体をメモリ領域に読み込む必要はなく、これから実行しようとする部分だけを読み込めばいいという考え方に基づいています。プログラムを細かく分割しておいて、必要な部分だけをメモリ領域に読み込みます。

仮想記憶

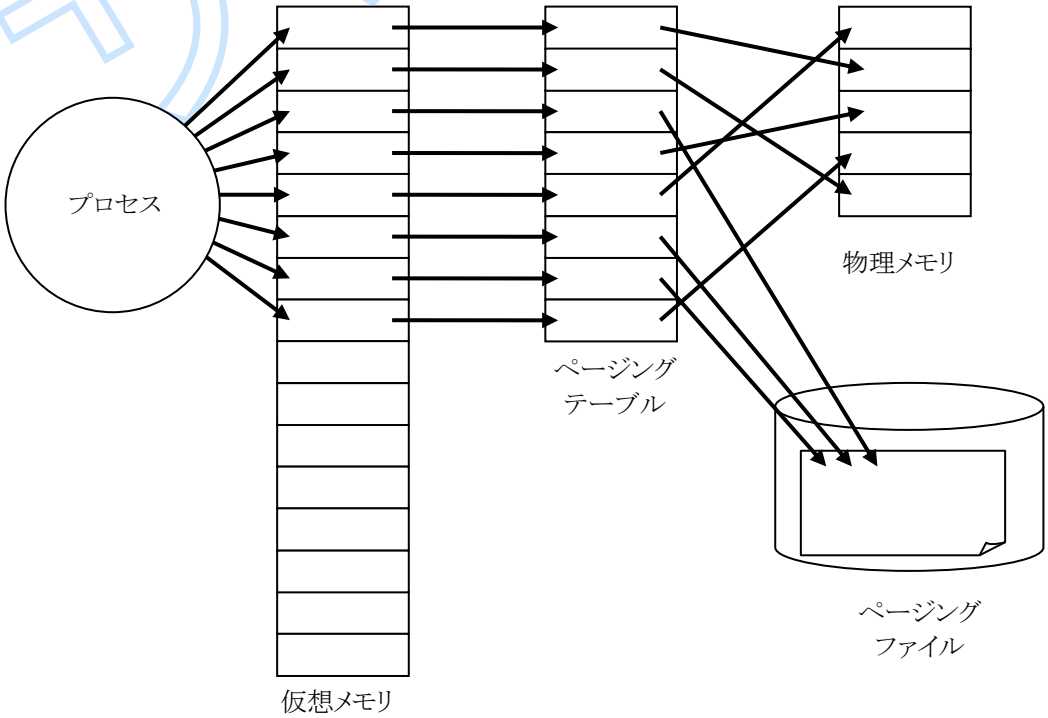
実際は、ほとんどの OS では、コンピュータに物理的に存在しているメモリをそのままプロセスに割り当てているわけではありません。OS が行うメモリ管理では、仮想記憶という概念を使います。「仮想記憶」は限られたメモリを有効に利用するしくみです。「仮想」は「Virtual」の翻訳ですが、ここでは「見せかけ上存在している」というような意味で使っています。

仮想記憶の原理は、主記憶装置上にある「現在使っていないプログラムやデータ」を一時的にハードディスクなどに退避させ、必要になったら主記憶装置に呼び戻すことです。この作業は OS が自動的に行います。そのため、OS 上で動作しているプログラムは、OS が「存在しているかのように見せかけている」大量のメモリを使用しているかのように動作します。アプリケーションが利用する仮想的なメモリを「仮想メモリ」または「論理メモリ」と呼び、実際に存在する主記憶装置を「物理メモリ」と呼びます。

現在、ほとんどの OS で用いられている仮想記憶のしくみは、「デマンドページング方式」と呼ばれるものです。単に「ページング方式」と呼ぶこともあります。

デマンドページング方式では、メモリを固定長のサイズに分割して管理します。分割されたメモリの断片を「ページ」と呼びます。1 ページのサイズは 2 キロバイトから 4 キロバイトが一般的です。

プロセスが OS に対してメモリを要求すると、OS はそのプロセスに対して固定長のページ単位でメモリを割り当てます。このとき OS がプロセスに見せるのは、OS 上で構築した仮想的なメモリ空間です。OS は仮想メモリと物理メモリの対応表を作り、アプリケーションから物理メモリを隠します。この表を「ページングテーブル」といいます。

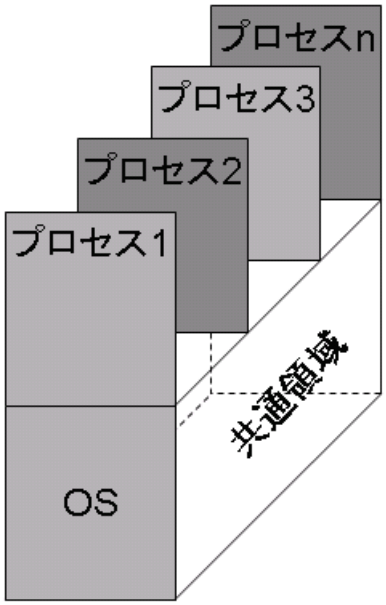


物理メモリが足りなくなると、OS はしばらくの間使っていないプログラムやデータが格納されている物理メモリ空間を「ページングファイル」または「ページファイル」と呼ばれるファイルに記録し、そのメモリ空間を空けます。この作業を「ページアウト」といいます。また、ページアウトしたメモリ空間が必要になると、ページングファイルからその情報を読み取って物理メモリに戻します。その作業を「ページイン」といいます。OS はこのような方法で、あたかも大量のメモリが存在しているかのように見せかけるのです。

仮想メモリ空間の例

現在、一般に使われている仮想記憶空間（仮想メモリ空間）は「多重仮想記憶（MVS：Multiple Virtual Storage）」と呼ばれるレイアウトになっています。MVS は米国 IBM 社が開発した OS の名称ですが、これが一般名称にもなっています。

多重仮想記憶では、プロセスごとに独立したメモリ空間を持ちます。そのため、同じ番地（仮想記憶上の番地）でも、プロセスが異なれば異なる値を指します。



仮想メモリのしくみは OS によって異なりますが、例えば 64 ビット OS である Windows では図のようなレイアウトになっています。

64 ビット OS は、理論的にはプロセスごとに 16 エクサバイト（2 の 64 乗）のメモリ空間を用意できます。しかし実際にはそれだけ広大なメモリ空間を管理するにはオーバーヘッドがかかるため、各プロセスには最大 16 テラバイトのメモリ空間を割り当てるよう制限されています。そのうちの下位 8 テラバイトは共通で、上位 8 テラバイトがプロセスごとに独立に用意されます。もし 4 つのプロセスが同時に動作しているのであれば、OS は合計で 40 テラバイトものメモリ空間をプロセスに割り当てていることになります。

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

スラッシング

仮想記憶において物理メモリの容量が十分でない場合、多くのプログラムを読み込むとページング処理が多発します。そのため、ページング処理に CPU の処理能力の大部分が当てられ、外部からの他の処理を受け付けられない状態になることを「スラッシング」といいます。スラッシングが発生すると、システムの反応速度が極端に遅くなります。

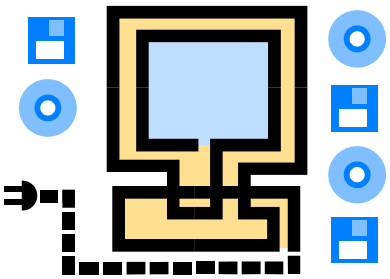
カーネルモード と ユーザーモード

ほとんどの CPU は、2 種類のプログラム実行モードを持っています。1 つは「カーネルモード」または「スーパーバイザーモード」と呼ばれるもので、もう 1 つは「ユーザーモード」と呼ばれるものです。

カーネルモードは、CPU が持つすべての機能を使えるモードです。例えば、メモリの割り当てや周辺装置の利用はカーネルモードでしか行えません。OS がメモリの確保や周辺装置を扱うためのプログラムはカーネルモードで動作します。一方、ユーザーモードは限定した機能しか使えません。OS 上で動作する大部分のプログラムはユーザーモードで動作します。

OS 上で動作しているプロセスがメモリを確保したり、周辺装置を使いたくなったりした場合は、OS に依頼を出します。OS はプログラムの依頼に基づいて、可能ならその作業をカーネルモードで実行し、結果をプロセスに戻します。もし他のアプリケーションに悪影響があるなどの問題がある場合は依頼を拒否します。

周辺装置は新しいものがどんどん開発されています。新しい周辺装置にはたいてい新しい機能が追加されています。その周辺装置の機能を利用するには、OS がその周辺装置が持つ新しい機能を知っていなければなりません。具体的には、周辺装置が持つ機能を有効に活用できるプログラムが OS に備わっていなければなりません。しかし、新しい周辺装置が発売になるたびに OS を作り変えるのは事実上不可能です。そこで、OS そのものと周辺装置を制御するプログラムとを切り離し、周辺装置を制御するプログラム部分を簡単に追加したり削除したりできるようになっています。この、周辺装置（Device）を制御（Drive）するためのプログラムのことを「デバイスドライバ」といいます。デバイスドライバは OS の一部としてカーネルモードで動作します。



Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

資源管理

「資源管理」も OS の重要な仕事です。例えば、複数のプログラムが同時にプリンタに印刷を始めても、実際に同時に印字して印字結果が混じり合うことは許されません。また、あるプログラムがファイルを読み込んでいる最中に、別のプログラムがそのファイルを書き換えてしまうと、予期せぬ事態が起こってしまいます。このような制御を行うのが資源管理の目的です。

資源管理には以下の機能があります。

資源割り当て

プロセスからの周辺装置の利用要求に対する動作が「資源割り当て」です。「資源」には、メモリやハードディスク、プリンタなどの周辺装置のほか、OS が保存しているさまざまな情報も含まれます。例えば、プロセスの実行順序を制御するための情報も資源の 1 つです。

プロセスが資源を要求した場合、OS は以下のいずれかの返事を返します。

1. 資源を割り当てる（成功したことを通知）
2. 資源を割り当てない（失敗したこととその原因を通知）

失敗の原因にはさまざまな原因があります。要求した資源が見つからない、その資源が別のプロセスに割り当てられている、要求した資源がそのプロセスの権利で操作することができない、などです。OS はその原因をプロセスに伝えます。例えば、OS によって印刷を禁止されているプロセスがプリンタ資源を要求すると、「印刷の権限が不足している」というエラーを返します。

排他制御

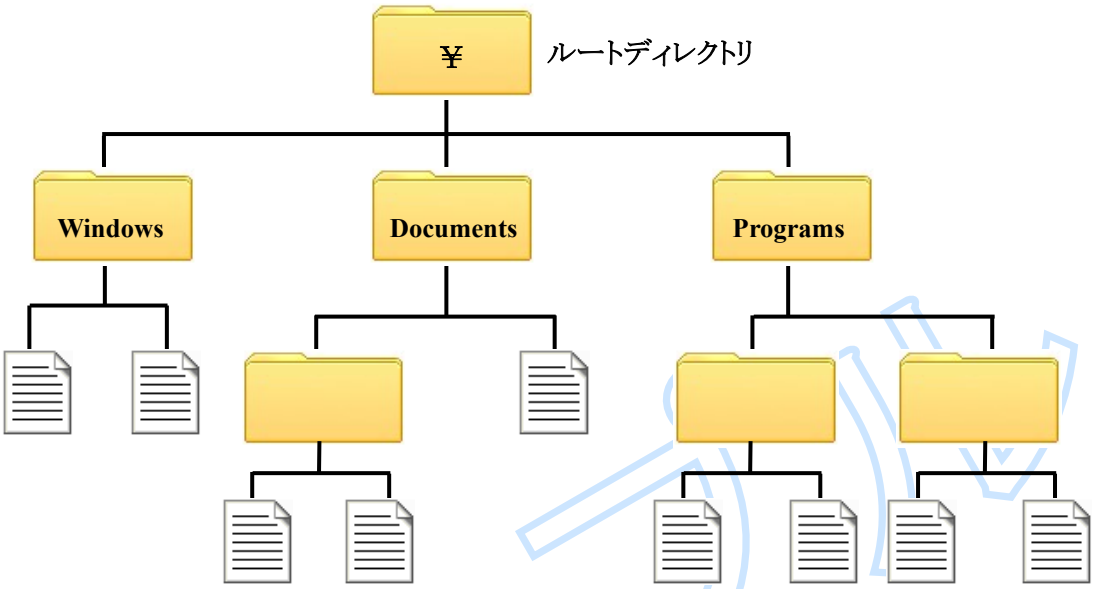
複数のプロセスが同時に同じ資源を要求する場合もあります。資源の種類によっては、それが可能なものとそうでないものがあります。読み取り専用の情報（例えば OS の種類やバージョン、プリンタ資源が存在しているかどうか）であれば、異なるプロセスが同時に同じ資源にアクセスすることができます。

しかし、1 度に 1 つのプロセスからしか利用できない資源も多く存在します。最もわかりやすい例は、前述のプリンタです。この場合、OS は、あとから要求したプロセスに対して「使用中」という返事を返します。同時に利用できるプロセスを最大 1 に制限する機能が「排他制御」です。

排他制御により資源が利用できなかった場合の動作は、プロセス上で動作しているプログラムによって異なります。例えばプログラムは OS に対して「資源が空いたら通知してほしい」という要求を設定したり、「資源が空いたら即座にこの処理を実行してほしい」と依頼内容を OS に登録したりします。

ファイル管理

OS 上で何らかの作業を行い、その作業の結果を補助記憶装置に保存する時には、ファイルと呼ばれる単位で保存します。ファイルは通常、ディレクトリ、またはフォルダに格納される形式でディスク上に記録されています。



ユーザは、OS を経由してファイル単位でデータを読み出したり、記録したりします。ファイルが補助記憶装置のどの場所に、どのような形式で保存されているかということは、ユーザは意識する必要がありません。それは、OS がファイルの読み出しや記録に関する管理を行っているからです。

補助記憶装置にファイルを保存するためには、OS が理解できる形式のファイル・システムを補助記憶装置上に作成しておく必要があります。ファイル・システムとは、補助記憶装置上にファイルを格納するためのしくみのことです。ファイル・システムがなければ、OS は補助記憶装置にファイルを格納することができません。

補助記憶装置上にファイル・システムを作成する作業のことを、初期化、またはフォーマットといいます[6]。

絶対パスと相対パス

階層化されたディレクトリ構造の場合、最上位のディレクトリをルートディレクトリといいます。また、ユーザが選択したいずれかのディレクトリを、カレントディレクトリといいます。

ファイルの場所を指定する際に、ルートディレクトリからの経路を示す「絶対パス」と、カレントディレクトリからの経路を示す「相対パス」で表す方法の 2 種類があります。

[6] 家電量販店などにある SD カードや USB メモリ、ハードディスクなどは、あらかじめフォーマットされ、ファイル・システムが作成された状態で販売されています。そのため、通常は購入後あらためてフォーマットする必要はありません。しかし、そのファイル・システムが使用中の OS にとって理解できない形式のものであれば、購入後あらためて使用している OS を用いてフォーマットしなければなりません。

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

代表的なファイル・システム

代表的なファイル・システムには次のようなものがあります。

■ FAT ／ FAT32 ／ exFAT

FAT は、米 Microsoft 社が 1981 年に開発した MS-DOS という OS に採用され、その後 Windows でも用いられていたファイル・システムです。FAT はその後 FAT32、exFAT と拡張され、現在でも USB メモリや SD カードなどの標準的なファイル・システムとして採用されています。また、Windows 以外の OS でもこれらのファイル・システムを使用できる OS が多数存在します。

■ NTFS ／ ReFS

NTFS は、米 Microsoft 社が開発した Windows NT（Windows Server の前身）という OS に採用されたファイル・システムです。それまで用いられていた FAT に比べて、以下のような点で優れています。

－ファイル単位、フォルダ単位でアクセス許可の設定を行うことができる
－ファイル単位、フォルダ単位で圧縮して格納し、ディスク容量を節約できる
－ファイル単位、フォルダ単位で暗号化し、本人以外は参照できないようにできる
－ドライブ単位でユーザごとの最大使用可能容量を制限できる

以上のような特徴から、現在一般的に用いられている Windows OS では標準のファイル・システムとして用いられています。

Windows Server 2012 以降では、NTFS に比べてより耐障害性や拡張性を高めた ReFS というファイルシステムが利用可能になっています。

■ HFS ／ HFS＋ ／ APFS

HFS は、米 Apple 社が Mac 用に開発したファイル・システムです。Mac OS 標準フォーマットともいいます。その後、大容量のファイルを扱うために HFS+（Mac OS 拡張フォーマット）が開発されました。さらに、フラッシュメモリや SSD に最適化した APFS というファイル・システムが開発され、現行の Mac や iPhone、Aplle Watch などに用いられている OS の標準ファイル・システムは APFS です。

■ ext2 ／ ext3 ／ ext4

ext2 は、初期の Linux で用いられていたファイル・システムです。ファイル・システムが論理的に壊れてしまった場合の復旧作業が大変困難であり、後に論理的破壊から復旧できる ext3 が開発されました。また、ext3 をさらに拡張し、より広大なサイズのファイルと多数のファイルを管理できる ext4 が開発されています。ext4 は、現在の Linux における標準的なファイル・システムになっています。

オペレーティング・システムの種類

OS はシステムの処理形態によっていくつかの種類があります。

バッチ型 OS

即時性の要求されない大量の情報を、コンピュータがまとめて処理する形態のことを「バッチ処理」といいます。身近な例では、携帯電話や水道光熱費、クレジットカードなどの請求書の発行が挙げられます。これらの請求書は、月に一度回線契約者の手元に送られてきます。事業者は契約者の 1 カ月間の契約状況と使用状況をデータとして管理し、月に 1 度、全契約者に対して請求書を作成します。このように、バッチ処理とは一定期間に 1 度、大量のデータを一括して処理する場合に使います。

バッチ処理は、どのような処理結果がいつ必要かによって、行われるタイミングが異なります。毎日行われるバッチ処理のことを「日次バッチ」、1 カ月に 1 回行われるバッチ処理のことを「月次バッチ」といいます。

大量のデータが扱われるバッチ処理では、コンピュータに対してかかる負荷も大きくなるため、多くの場合コンピュータの利用頻度の高い日中を避けて夜間に行われます。夜間に行われるバッチ処理のことを「夜間バッチ」といいます。したがって、毎晩行われるバッチ処理は、「日次バッチ」であり、なおかつ「夜間バッチ」であるといえます。

また、コンピュータがネットワークにつながった状態で動作するバッチ処理を特に「オンラインバッチ処理」といいます。一方、ネットワークに接続しない状態で行うバッチ処理を「オフラインバッチ処理」といいます。

さて、このバッチ処理に適した OS が「バッチ型 OS」です。現在では、バッチ処理ができない OS はありません。

対話型 OS

コンピュータに通信回線で常時つながっている端末と呼ばれる装置から情報を入力すると、即座に必要な処理を行う形態のことを「オンラインリアルタイム処理」といいます。身近な例では、JR のみどりの窓口に設置してある列車の座席予約システムや、ショッピングサイトの商品購入システムがこの形態にあたります。

1 件 1 件の、座席検索や座席予約の単位を「トランザクション」といいます。そして、オンラインでトランザクションを処理する形態を「OLTP（Online Transaction Processing）」といいます。

さて、OLTP での利用に向いている OS が「対話型 OS」です。対話型 OS は、利用者がコンピュータと対話的（Interactive）に操作する OS で、現在最も広く使われています。対話型 OS にはコマンドをキーボードから入力する「CUI（Command User Interface または Character User Interface）型」と、マウスでアイコンやメニューを操作する「GUI（Graphical User Interface）型」があります。GUI 型の OS のほとんどは CUI も装備しています。また、ほとんどの対話型 OS は、バッチ処理の機能も有しています。

リアルタイム型 OS

周辺機器から割り込みがあった場合、その応答時間を厳密に予測できる OS が「リアルタイム型 OS」です。リアルタイム型 OS は、文字通り「リアルタイム」に処理を行う必要があるので、航空管制や発電所など、非常に応答性と正確性が厳密に要求される場面で使用されています。

最近は携帯電話や自動車に組み込まれたコンピュータの OS としても使われます。



Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

代表的なオペレーティング・システム

代表的な OS には以下のものがあります。

●Microsoft Windows

「Microsoft Windows」は、Microsoft 社が開発した PC 用の対話型 GUI OS です。Windows の利点は分かりやすいグラフィカルユーザーインターフェースです。

現在の Windows は、Windows NT という名称で 1993 年に登場しました。その後、デスクトップ用途の OS としては、Windows 2000、XP、Vista、7、8、8.1、10 と変遷し、現在は Windows 11 という名称になっています。

一方、サーバ用途の OS としては、Windows NT、2000、2003、2008、2012、2016、2019、2022 と変遷してきています。いずれもその時点での最新技術を導入しており、大企業の中心となるコンピュータ用 OS として利用されています。

●UNIX 系の OS

「UNIX」は、米 AT&T 社ベル研究所のケン・トンプソン（Ken Thompson）らが中心となって開発した対話型 OS で、1969 年に最初のバージョンが完成しました。

初期の UNIX は研究者向けに公開されていたので、さまざまな種類の UNIX が誕生しました。特に、カリフォルニア大学バークレイ校で開発された UNIX はネットワーク機能を標準で備え、「BSD（Berkley Software Distribution）」として広く普及しました。

バークレイ校で開発された UNIX は、現在無料のソフトウエア FreeBSD として流通しています。また、AT&T からライセンスを受けて、多くのコンピュータベンダが商用の UNIX を開発し販売しています。その代表は米国 Oracle 社の Solaris や米国 Hewlett Packard（HP）社の HP-UX、米国 IBM 社の AIX などです。

UNIX を参考にして、別のアイデアを基に最初から作り直したのが Linux です。Linux はヘルシンキ大学 の学生（当時）であったリーナス・トーバルズ（Linus Torvalds）が作成しました。無料で流通していますが、有料のサポート会社がいくつか存在します。

また、米 Apple 社の Mac で動作する macOS（旧 Mac OS X、OS X）も UNIX OS です。2007 年 10 月に発売された MacOS X 10.5 から正式に UNIX としての認証を受けました。バージョン 10.2 から macOS と表記するようになりました。

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

組み込み機器用の OS

携帯電話などの情報機器に内蔵されたコンピュータ（組み込みコンピュータ）にも OS（組み込み OS）が存在します。代表的なものは以下の通りです。

■ Android OS

米 Google 社が中心となって設立した OHA（Open Handset Alliance）という団体が開発している、携帯電話や情報携帯端末用の OS です。正確には、OS、電話帳や Web ブラウザ、各種ユーザーインターフェースなどをパッケージにしたものを「Andoroid」と呼び、Andoroid OS はその中の一部です。Linux をベースに開発されています。完全に無償の OS で、誰でも使用することができます。

■ iOS / iPadOS / Watch OS

米 Apple 社が開発した、スマートフォンの iPhone、タブレットコンピュータの iPad、セットトップボックスである Apple TV、スマートウォッチの Apple Watch などで用いられている OS です。これらの OS は同社のコンピュータ用 OS である macOS を元に作られていますが、各機器向けにカスタマイズされています。ハードウエアを企画・製作しているメーカーが提供する OS であるということもあって、同社のハードウエアと非常に高い親和性を持ちます。



Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

■　ユーティリティ

OS そのものや OS が制御している周辺装置、その OS 上で動作するソフトウエアのさまざまな設定・管理を行うためのソフトウエアです。コンピュータに OS を組み込む際に同時に導入されるものや、その OS ベンダから別途提供されるもの、商品として購入するものなど、その導入形態はまちまちです。OS と同時に導入されるユーティリティは、OS の一部とみなされる場合もあります。

■　開発環境

OS の主な目的は、アプリケーションプログラムを動作させることにあります。アプリケーションプログラムを作り上げるためのプログラム群を、開発環境といいます。

開発環境に関しては、次節で詳しく扱います。

サンプル

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

4 - 3 ソフトウエア開発

コンピュータに作業をさせるには、必ずそのためのソフトウエアが必要となりますが、その場合のソフトウエアの入手方法は大きく 2 つに分かれます。

1. 市販されているソフトウエアを購入して使う
2. ソフトウエアを最初から新規に開発する

標準的な業務であれば市販のパッケージソフトで十分対応することも可能です。市販されているソフトウエアを購入することのメリットは、(新規開発に比べて)価格が安い、市販の書籍やインターネット上に情報が多数存在している、利用者が多いので素早い不具合報告やバージョンアップが期待できる、などです。一方で、目的に応じたすべてのニーズに応えられるとは限らない、汎用的に作られているため独自の要求を満たさない、既知の不具合を利用しての盗聴や情報改ざんといった危険性がある、というデメリットもあります。

市販のパッケージソフトでは充分に要求を満たすことができない場合は、ソフトウエア開発事業者に依頼して専用のソフトウエアを新規に開発することになります。

■　ソフトウエア開発の工程

一般に、ソフトウエア開発は大きく「要件定義」、「設計」、「実装」、「テスト」の 4 つの工程に分かれています。

それぞれの工程で行うべき内容とその役割を説明します。

要件定義

ソフトウエア開発の最初の工程は、要件定義です。この工程では、開発しようとするソフトウエアに対して、顧客がどのような要求を持っているのかを明らかにします。そして新しいシステムに必要な機能を定義し、文書化します。開発するソフトウエアが最終的に満足しなければならない機能のことをソフトウエアの要求仕様と呼びます。

システムの現状と顧客の要求を分析し、最終的にソフトウエアの要求仕様を決めることが要件定義の目的です。要件定義を行う開発担当者のことをシステム・アナリストと呼びます。

要求仕様を決定するには、コンピュータ化しようとしている業務に関して深く理解している必要があります。そのため、利用者の代表とシステム・アナリストが協力しあって作成しなければなりません。また要求仕様は、コンピュータの技術的、専門的な言葉を用いるのではなく、利用者の立場に立った言葉を用いて記述します。

設計

ソフトウエア開発の 2 番目の工程は設計です。この工程では、要件定義で明らかにされたソフトウエアの要求仕様をどのように実現していくかを決めます。設計を行う技術者のことを「システムエンジニア」と呼びます。

■外部設計

要件定義の結果を受けて、システムのおおまかな機能設計を行います。この工程では、ハードウエアやソフトウエアなどは意識せず、利用者の立場から見た設計を行います。具体的には、フォーム／帳票設計、データ設計、コード設計を行います。

■内部設計

外部設計の結果を受けて、開発者からみたシステム設計を行います。システム上必要となる全ての機能をコンポーネントと呼ばれる個々のプログラムに分割し、プログラム間の流れ（インターフェース）を明確にします。具体的には、システムフロー設計、コンポーネント分割などを行います。

■プログラム設計

内部設計で分割した各コンポーネントに対して、実際にプログラムが組めるように詳細に設計します。コンポーネントを、さらに細かいモジュールと呼ばれる単位に分け、モジュール間のデータのやりとりや各モジュールがデータベース上のデータをどのように扱うか、ということを設計します。

実装（プログラミング）

実装段階では設計工程までで作成された文書に従って、「プログラミング言語」（後述）を用いて実際にプログラムを作成します。この作業のことを「プログラミング」または「コーディング」といい、プログラムを記述する技術者のことを「プログラマ」と呼びます。プログラミングやコーディングは、文章を書くときと同じように、他人にも分かりやすいものを書くことが大切です。

テスト

プログラムが完成してもソフトウエア開発が終わったわけではありません。テストの工程では、製造したソフトウエアを実際に実行させ、さまざまなテストデータを入力して仕様通りに動くかどうか検査し、間違いを取り除いていきます。厳密なテストを行うには高度な技術が必要です。



■デバッグ

プログラムは複雑なので、どうしても間違いが付きまといます。プログラム内の間違いのことを「バグ（虫）」といい、間違いを見つけて取り除く作業のことを「デバッグ」といいます。

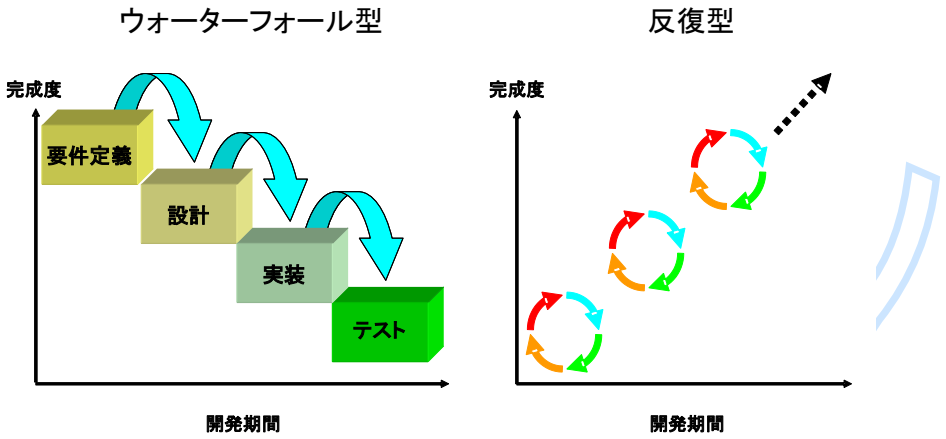
Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

■　開発プロセス

以上がシステム開発時のおおまかな流れですが、各工程をどのように進めていくかを定めているのが「開発プロセス」と呼ばれるものです。

代表的な開発プロセスには、「ウォーターフォール型」と「反復型」があります。

代表的な開発プロセス



ウォーターフォール型

代表的な開発プロセスの 1 つで、1970 年代から利用され始め、現在でも多くのプロジェクトで採用されています。ウォーターフォール型の開発プロセスは以下のような特徴があります。

前提	・ ユーザの要求は最初に固まる ・ 机上の検討だけで正しい設計が得られる ・ 前工程は間違っていない ・ 「要件定義」→「設計」→「実装」→「テスト」と直線的に作業が続く
利点	・ 長年の実績があり、プロジェクト管理の経験者が豊富 ・ 見積もり（コスト、ヒューマンリソース、スケジュール）に関する具体的な数値がわかりやすい
問題点	・ 後工程で変更が発生した場合、前工程からのやり直しに膨大な時間がかかる ・ 開発初期段階でユーザのフィードバックが得られない ・ 後工程において前工程の間違いを発見したとき、リスクが一気に露呈する

反復型（スパイラル型）

1 つのシステムを管理しやすい単位（ミニプロジェクト）に分割し、開発サイクルを反復しながら、継続的にシステムを洗練していくプロセスです。

反復型のプロセスの特徴は以下の通りです。

利点	・ 初期の開発サイクルでリスクを発見できる ・ 重要な部分から小さな単位で利用者に対して導入・公開することで、早期に利用者からのフィードバックを得られる ・ ある時点で要求、市場、使用する技術の変化が発生しても、次の反復で吸収しやすい
問題点	・ 管理が難しい ・ 開発プロジェクトの終了見極めが難しい ・ 要求が高まりすぎる傾向にある

うまく反復を管理することができれば、以下のような利点も得られます。

- 製品をスケジュールどおりに市場投入できないというリスクが軽減できる
- 短期間で明確な目標に向けて作業ができるので、開発者のモチベーションを維持できる

反復型（アジャイル型）

計画よりもフィードバックを重視した開発技法です。すべての工程を 1 日から 1 週間程度で終了するいくつもの工程に分割し、「不完全だがとりあえず動く」という段階のソフトウエアを早々に利用者に公開して、利用者からフィードバックを得ます。そのフィードバックを元に、徐々にソフトウエアを改良していくのです。

各工程では、設計書をはじめとするさまざまなドキュメント（文書）を作成します。ドキュメントは、開発したソフトウエアの改良や修正などの運用・保守をするために非常に重要です。

ほとんどのシステム開発プロジェクトでは、開発したアプリケーションと共に各種ドキュメントを顧客に納品・検収するのが一般的です。また、既存のアプリケーションに変更などが生じた場合、それらのドキュメントがあることで変更が容易になります。

大規模なソフトウエア開発では、各工程を管理するのは非常に困難です。特にアジャイルでは、各工程をきちんと管理しないとプロジェクト全体がどんどん増大し、コストが膨れ上がる可能性があります。そのため、プロジェクト管理のためのさまざまな技法が使用されます。

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

DevOps

立場、役割の違いから対立しがちな「開発（Development）」と運用「（Operations）」が協力してユーザの要求に対して、スピーディーに、そしてより柔軟に対応できるシステムを作り上げるための取り組みのことです。

旧来、新しいシステムを作り上げたり、既存のシステムを改良したりする役割の「開発」部門と、システムを安定的に稼働させ続ける役割の「運用」部門は、その役割や優先順位の違いに起因する衝突が少なからずありました。例えば開発部門は「利用者にとって使いやすい、レスポンスの速いシステムを作る」ということを最優先に考え、運用・保守のしやすさについてあまり考慮せずに（そのための時間があまりとられずに）システムを作ることがあります。また、少ない開発要員や限られた時間の中でシステムを作る必要に迫られることも多く、利用者が求めるすべての機能が搭載されないままリリースする（次の改訂で対応する）ということも少なからず存在します。一方運用部門は、利用者からの問い合わせや障害対応などを一手に担う部門です。利用者が求める機能が搭載されていない場合の問い合わせやクレームは運用部門に向けられます。また、効率の良い運用に必要な機能が、システムに搭載されていない場合もあります。

こうした開発部門と運用部門との衝突を防ぐには、部門間の情報の共有、目的を達成している度合いの測定、プロセスの自動化、協力体制、そして組織のカルチャーが不可欠です。特に、情報システムが担う役割や自組織が利用者やお客さまに対して提供できる価値を戦略的に考えること、お客さまから依頼されたシステムを、運用も含めて設計すること、開発の趣旨や方向性などを運用部門も理解しておくこと、現場でどのように運用されているかを開発部門も理解しておくこと、など、システムのライフサイクル全体を包括的に考えた「開発と運用のコラボレーション」ともいえるべきしくみが必要です。

DevOps は、これらのしくみを組織全体で作り上げ、開発部門と運用部門とが協力し合って、お客さまの要求に迅速に、正確に対応するためのしくみを提供します。DevOps という言葉が初めて登場したのは 2009 年であり、IT の歴史から見てもまだ新しい概念です。また、まだ具体的な定義は存在していません。しかし、今後開発部門と運用部門とが協力し合ってシステムを作り上げ、運用することが必要である、と認識され始めたことは間違いありません。

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

■　プログラミング言語

コンピュータの内部では、全ての情報が 2 進数で表現されていることはすでに説明しました。したがって、プログラムも全て 0 と 1 との組み合わせで記述されています。

このような、コンピュータが理解できる形式で書かれているプログラムを「機械語（マシンコード）」といいます。

しかし、機械語のプログラムは人間にとっては理解しづらく、作成や修正が困難です。そこで、プログラムの手順をもっと分かりやすく記述する方法が考えられました。これが「プログラミング言語」です。

最初に登場したプログラミング言語は「アセンブリ言語」です。これは、機械語と 1 対 1、あるいはそれに近い対応関係を持った命令を使って記述しました。高速なプログラムを作れるため、現在でも一部で使われています。

しかし、機械語は、1 命令あたりの機能が低いため、簡単な作業でも多くのステップが必要になります。そこで、もう少し人間が普段使用している言語や数学的表現に近い記述方法を使うことが検討されました。このようにして設計されたプログラミング言語を「高級言語」といいます。

高級言語は、最終的に実行される機械語と単純な対応をしていません。そのため、高級言語から機械語に翻訳をする必要があります。この翻訳を行うことを「コンパイル」といい、コンパイルするためのプログラムを「コンパイラ」といいます。変換前のプログラム（高級言語で書かれたプログラム）を、「元になるプログラム」という意味から「ソースプログラム」あるいは「ソースコード」と呼びます。



Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

■　コンパイラとインタプリタ

プログラミング言語の翻訳（コンパイル）を行う際、一度に完全な機械語を生成する場合と、1 行ずつ逐次翻訳をする場合があります。一度で翻訳するプログラムを「コンパイラ」、逐次翻訳をするプログラムを「インタプリタ（通訳）」といいます。

コンパイラ

一度翻訳を行えば、後は機械語のプログラムを実行するだけなので高速に実行できます。その反面、1 カ所変更するだけでも、ソースプログラムの修正と再コンパイルが必要になります。コンパイル作業は、小さなプログラムでは数秒ですが、大規模なアプリケーションの場合は数十時間かかることもあります。

インタプリタ

ソースプログラムを 1 文ずつ読み込んではその都度機械語に通訳し、実行するタイプのプログラムのことを指します。事前にコンパイルが不要なので、修正した内容は即再実行できます。しかし、毎回機械語に通訳しながら実行するので、コンパイルしたプログラムに比べて実行時の速度は遅くなってしまいます。

インタプリタをさらに手軽に使えるように改良したのが「スクリプト」です。プログラミング言語とスクリプト言語との明確な違いはありませんが、現在では、アプリケーションを開発するための言語をプログラミング言語、アプリケーションの制御を行うための言語をスクリプト言語と呼ぶ傾向があります。スクリプトは、単純な文法規則を持ち、専門のプログラマでなくても使えるように工夫されています。また最近のスクリプト言語は大規模なシステム開発に耐えうる機能性と堅牢性を持つものが増えています。



Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

■　プログラミング手法

一般にプログラムを作成する場合、事前に仕様書や設計書などの文書を作ります。それらがある場合とない場合では、その品質に大きな差が出ます。

仕様書や設計書は開発するプログラムのまさに「設計図」です。設計図が正しく作られていないと、その設計図を元にしたプログラムが正しく作られる可能性は低くなります。また、テストは設計図を元に行われることが多いので、設計図の品質が悪いとテストにも意味がなくなってしまいます。

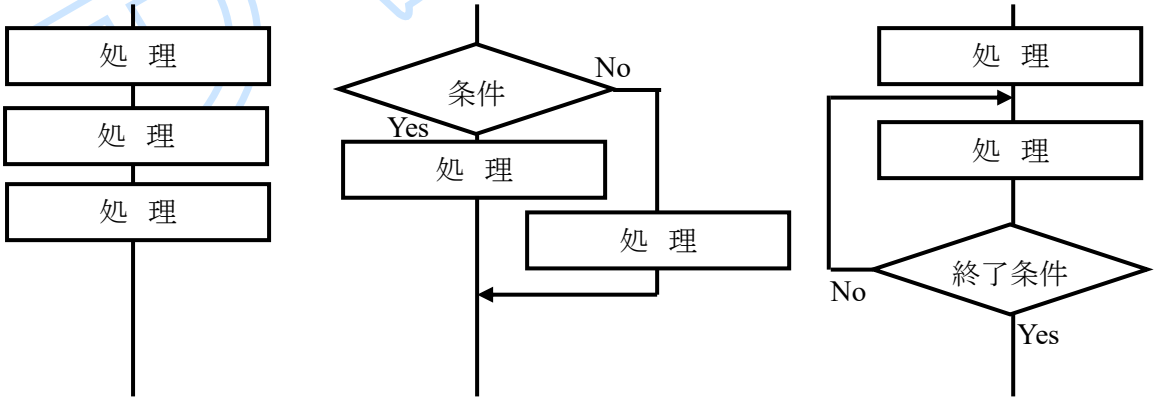
作成したプログラムを第三者に検査してもらうことも有効です。これを「コードレビュー」といいます。コードレビューをすると、確実にプログラムの間違いが減ります。

そのほか、プログラムの生産性を向上させるために、さまざまなプログラミング技法があります。中でも古典的なものが「構造化プログラミング」です。

構造化プログラミング

構造化プログラミングとは、「順次」、「分岐」、「繰り返し」の 3 つの組み合わせでプログラムの流れをつくるという考え方です。

構造化プログラミングは分かりやすいプログラムを書くために効果的です。さらに、プログラムを動作の手順である「アルゴリズム」とデータの扱い方である「データ構造」に分けて考え、データ構造にあったアルゴリズムを選ぶべきだという考え方に発展しました。



＜順次＞　＜分岐＞　＜繰り返し＞

またプログラムは、なるべく独立した部品を組み合わせ、各部品は最小限のデータのやり取りで済ませることが望ましいとされています。そうしておけば、ある部分を修正しても、その影響が及ぶ範囲を最低限に抑えることができます。このようなプログラム設計手法を「モジュール化」といい、部品化した各プログラムを「モジュール」といいます。モジュール化されたプログラムを組み合わせて作成したプログラムは、各モジュールの内部を理解する必要がないので、容易に理解できます。

プログラミングを行う場合、プログラミング言語のコンパイラまたはインタプリタと、プログラムを記述するためのツールが必要です。最近では、これにデバッグ用のテストツールも加えた「統合開発環境（IDE：Integrated Development Environment）」を使うことが増えてきました。

プログラミング言語の仕様は特許も著作権も存在しないので、誰でも自由にコンパイラやインタプリタを作成できます。ただし、実際のプログラムであるコンパイラやインタプリタには著作権などの知的所有権が発生します。

なお、略語すべて大文字で記述するのが一般的ですが、Basic のように英単語として発音する場合は先頭のみ大文字で記述することもあります。

サンプル



以下に、代表的なプログラミング言語を紹介します。

COBOL（コボル）

1960 年、米国国防省とコンピュータベンダ、ユーザ団体が協力して発足した CODASYL 委員会が開発した事務処理用の言語で、COmmon Business Oriented Language の略です。COBOL は、英語の文章に近い表現でプログラムが記述できるのが特徴です。現在でも金融機関を中心に広く利用されています。

Basic（ベーシック）

1963 年、米国のダートマス大学でトマス・カーツ（Thomas.E.Kurtz）とジョン・ケムニィ（Jhon.G.Kemeny）が教育用に開発した言語です。Basic は Beginners All-purpose Symbolic Instruction Code の略とされていますが、英語の basic をもじってこじつけたようです。文法がやさしく、初心者に分かり易いのが特徴です。Basic のインタプリタを標準で搭載する PC が出荷され始めたのがきっかけで、爆発的に普及した言語です。

C（シー）

1972 年、UNIX を作成することを目的として、米国の AT&T ベル研究所でデニス・リッチー（Dennis Ritchie）によって設計されたシステム記述言語です。UNIX のほとんどの部分は C 言語で記述されています。演算の種類が豊富なことや、他のコンピュータへの移植性が高いことなどが理由で、速度を要求されるアプリケーションに現在でも使われています。

C++（シープラスプラス）

1983 年、米国の AT&T ベル研究所のビヨーン・ストラウストラップ（Bjarnee Stroustrup）が作成した言語です。C 言語にオブジェクト指向という概念を取り入れた言語として設計されました。C 言語を元にしているので、C 言語との親和性が高いのが特徴です。

Java（ジャヴァ）

Java は 1995 年、米国 Sun Microsystems 社のジェームス・ゴスリング（James Gosling）が機器組み込み用のプログラムを作成する目的で開発した言語です。C++ 同様、オブジェクト指向という概念を取り入れています。

当時普及が進んでいた C++ に比べて仕様が単純なため、一般のプログラム言語として脚光を浴びました。当初はスクリプト言語の一種として扱われていましたが、コンパイラとインタプリタの両方の機能を兼ね備えているため、現在ではインターネット上で動作するアプリケーション、携帯電話用のアプリケーション、大規模なシステムアプリケーションを開発するための言語として広く使われています。

Visual Basic（ビジュアル ベーシック）

米 Microsoft 社により、Windows アプリケーション開発用に Basic 言語を大幅に機能拡張して作られた言語です。GUI を駆使した統合開発環境（IDE）で直感的なプログラム開発ができます。現在では Web ベースのアプリケーションなど、多様な形態のプログラム作成に使用されています。

C#（シーシャープ）

Microsoft 社のアンダース・ヘルスバーグ（Anders Helsberg）を中心とするチームが開発し、2000 年に発表された比較的新しい言語です。2001 年には国際的な標準化団体である ECMA International、2005 年には JIS の標準規格言語として認定されています。

C++ や Java の機能に Visual Basic の使い易さを加えたような仕様で、Windows や Web ベースのアプリケーションを始めとする多様な形態のプログラム開発が可能です。

Python（パイソン）

1990 年、オランダ人のグイド・ヴァンロッサム（Guido van Rossum）が開発したプログラミング言語です。文法が簡易で可読性に優れています。拡張機能が豊富で、Web アプリの制御や AI プログラミングなどに用いられています。

Perl（パール）

1987 年、米国のラリー・ウォール（Larry Wall）によって開発されたスクリプト言語です。テキスト（文字）ファイルから特定のテキストを検索・抽出したり、そのテキストを加工・出力したりするのに向いています。また、テキストファイルをデータベースのように扱うこともできます。

PHP（ピーエイチピー）

動的に Web ページを生成するのに向いているスクリプト言語です。Perl と同様にデータベースを扱うシステム開発にも向いています。

Ruby（ルビー）

1995 年、日本人である まつもとゆきひろ氏が開発したスクリプト言語です。Perl の代替として開発されました。手軽にオブジェクト指向が利用できることを特徴としています。

■ ライセンス

ここでいうライセンスとは、ソフトウエアライセンスのことです。一般的に市販のソフトウエアを購入して使用するには、使用開始時にソフトウエアの使用許諾書に書かれている規定事項に同意しなければなりません。ソフトウエアは著作物として保護されています。ソフトウエアを購入した場合でも、著作財産権を購入したことにはなりません。そのため、ソフトウエアの使用、改変、再配布、譲渡、ライセンスの有効期限などについて、適切に把握して管理しなければなりません。

プロプライエタリ・ソフトウエア

プロプライエタリ（Proprietary）には独占的な・所有者の、という意味する言葉です。代表的な OS である、Windows や Mac OS、また Word や Excel のようにソフトウエアメーカーがソースコードを独占しているソフトウエアはプロプライエタリ・ソフトウエアです。

OSS（Open Source Software）

ソースコードが公開されているソフトウエアのことをいいます。オープンソースということもあります。OSS には、以下のような定義があります。

1. 再頒布の自由	6. 利用する分野に対する差別の禁止
2. ソースコードの頒布の許可	7. ライセンスの分配
3. 派生ソフトウエアの作成・頒布の許可	8. 特定製品でのみ有効なライセンスの禁止
4. 作者のソースコードの完全性	9. 他のソフトウエアを制限するライセンスの禁止
5. 個人やグループに対する差別の禁止	10. ライセンスの技術的中立性

http://www.opensource.jp/osd/osd-japanese.html

コピーレフト

OSS のほとんどは、「コピーレフト」に分類することができます。コピーレフトとは、著作権を意味する「copyright（コピーライト）」からできた造語で、著作権者が著作権を持ち、配布先のユーザに対して自由に利用したり、再配布したりする権利を与えるものです。再配布する際には、配布先のユーザにも同様の権利を与えることを義務付けています。

GPL（GNU General Public License）

フリーソフトウエア財団（FSF：Free Software Foundation）から配布されているライセンスのことです。ソフトウエアをさまざまな制約から解放して、誰もが自由に扱えるということを目標にしています。ソースプログラムを公開することを条件として自由に改変したり、再配布したりすることなどを許しています。

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

4 - 4 ミドルウエア

さまざまなアプリケーションに対して特定の機能を提供するソフトウエアのことをミドルウエアといいます。OS とアプリケーションの間で動作し、処理を行うソフトウエアのことです。ミドルウエアだけで利用されることはありません。代表的なミドルウエアには、Web サーバ、アプリケーションサーバ、データベース管理サーバなどがあります。

Web サーバ

Web サーバは、クライアントコンピュータ上のウェブブラウザから送られたリクエストを受け、HTML や画像などの Web ページを構成するコンテンツを返します。代表的な製品に IIS（Internet Information Services）、Apache などがあります。

アプリケーションサーバ

ユーザからのリクエストを受け、データベースサーバに橋渡しをする機能を提供します。代表的な製品に ASP.NET、Tomcat などがあります。

データベースサーバ

データベースを保持し、このデータベースを管理し、ユーザからのリクエストに応じた処理を行います。代表的な製品に Microsoft SQL Server、ORACLE、MySQL、PostgreSQL などがあります。

4 - 5 応用ソフトウエア



OS は、コンピュータを利用するための最低限の機能を提供しますが、それだけでは利用者の目的にあった機能は実現できません。例えば、「家計簿をつける」、「年賀状を書く」、「デジカメで撮った写真を印刷する」、といった目的は、OS だけでは果たせません。

利用者の目的に合った機能を実現するソフトウエアを「応用ソフトウエア」あるいは「アプリケーションソフトウエア（アプリケーション）」といいます。通常「コンピュータを使う」というと、アプリケーションを使うことを指します。

多くの人が必要とするアプリケーションは、パッケージソフトウエアとして市販されています。またフリーウエアという形で、インターネット経由で無償入手できるアプリケーションもあります。企業向けの特殊な用途の場合は、その 1 社のためだけにアプリケーションを作成することもあります。

■ アプリケーションの役割

アプリケーションの役割は利用者がやりたいことを実現することです。コンピュータで何か特定のことをするには、必ずそのためのアプリケーションが必要になります。

利用者は、自分が必要とするアプリケーションを選択して調達することになります。アプリケーションを調達する際は、どの OS に対応しているか、そのアプリケーションが目的に合っているか、使い勝手は適当か、といったことを十分に確認する必要があります。

例外もありますが、アプリケーションは通常、OS ごとに用意されます。別の OS 用のアプリケーションを購入しても、ほとんどのプログラムは動作しません。例えば Windows 用のアプリケーションは、macOS や Linux では動作しません。

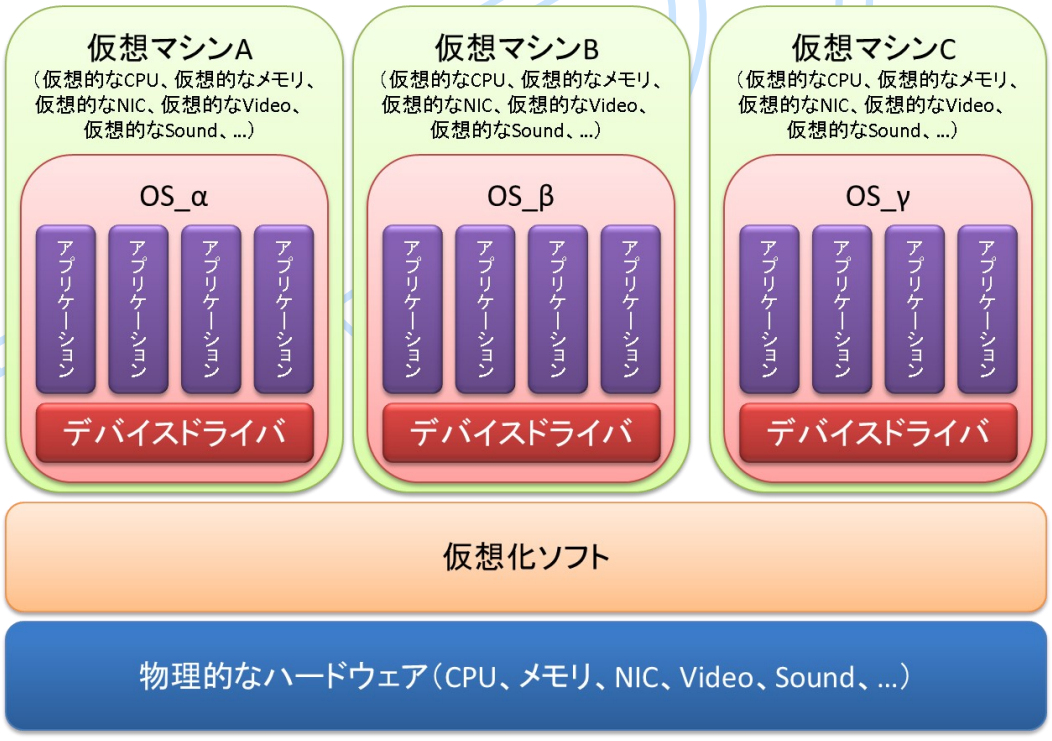
4 - 6 仮想化ソフトウエア

ソフトウエアの特殊な概念として、仮想化ソフトウエアと呼ばれるものがあります。

■ 仮想化ソフトウエアとは

仮想化とは、「資源の物理的特性を、その資源と相互作用する他のシステムやアプリケーション、利用者から隠蔽する技法」と定義されています。1 台の物理コンピュータ・システムをあたかも複数台のコンピュータ・システムであるかのように見せかけたり、逆に複数台のシステムをあたかも 1 台のシステムに見せかけたりする技術です[7]。

仮想化ソフトウエアとは、ソフトウエア的にコンピュータのハードウエア一式を存在しているかのように見せかけるソフトウエアのことです。ソフトウエア的に実現した仮想的なコンピュータ・システムのことを「**仮想マシン**」といいます。1 台の仮想マシンは、あたかも 1 台の物理的なコンピュータ・システムであるかのように振る舞います。当然、その仮想マシン上に OS をインストールすることもできますし、アプリケーションを動作させることもできます。仮想マシンにインストールする OS のことをゲスト OS といいます。



それぞれの仮想マシンは互いに完全に独立していますので、それぞれの仮想マシンの仮想的なハードウエア構成が違っていてもかまいませんし、それぞれの仮想マシンにインストールするゲスト OS がまったく異なる OS であってもかまいません。

[7]仮想化といってもその技術は様々である。例えば、第 4 章で紹介した仮想記憶も仮想化の一種であるし、1 台の Web サーバに複数のドメイン名を割り当て、あたかも複数台の Web サーバであるかのように見せかけるというのも仮想化であると言える。。しかし本書では、最も一般的な「仮想化ソフトを用いた仮想マシンの作成」に焦点を絞って説明する。

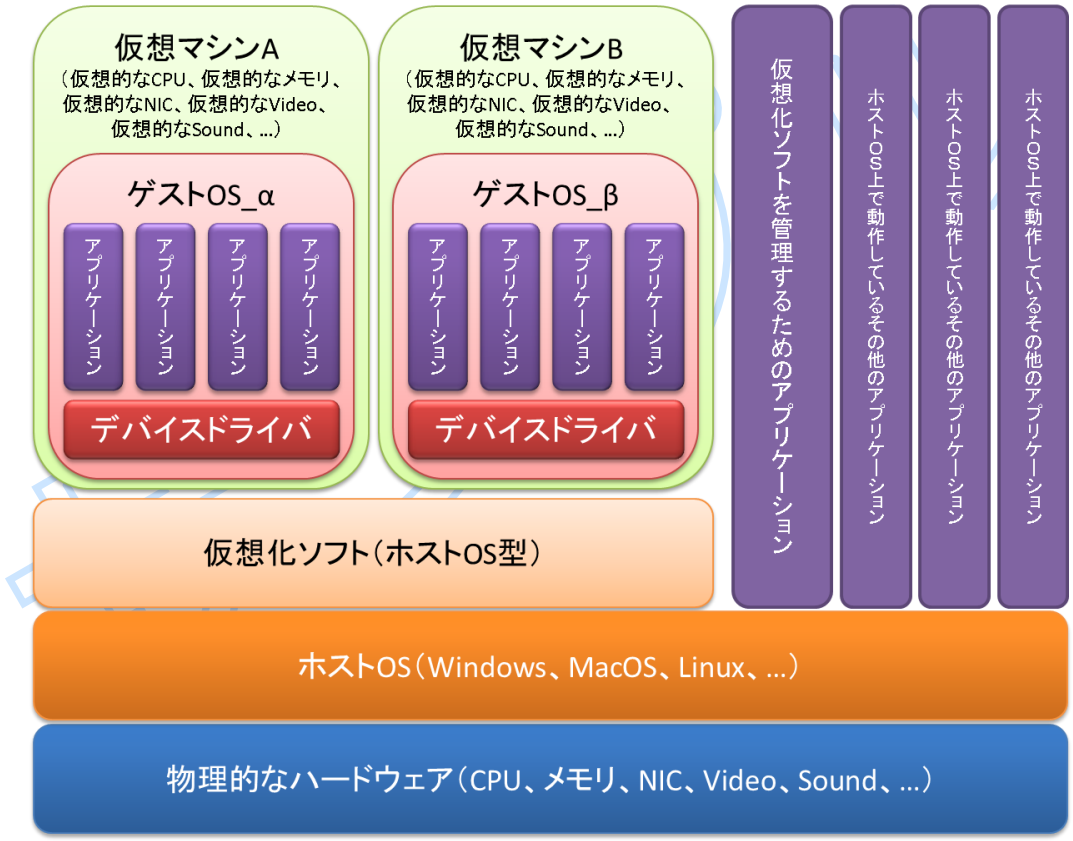
Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

■　仮想化ソフトウエアの種類

仮想化ソフトウエアは、大きく 2 種類あります。

ホスト OS 型

ホスト OS 型は、物理マシン上に Windows や MacOS、Linux といった通常の OS（ホスト OS）をインストールし、その上に仮想化ソフトをアプリケーションとしてインストールするタイプのものを指します。主な製品としては、VMware Workstation/Player/Fusion、Microsoft Virtual PC、Oracle VirtualBox、Parallels Desktop などがあります。



ホスト OS 型仮想化ソフトウエアは、専用の物理マシンを必要としません。普段使っている PC にホスト OS 型仮想化ソフトウエアをインストールして、手軽に仮想化環境を構築することが可能です。その一方で、仮想化ソフトウエアの真のメリットが活かせないという特徴があります。

ハイパーバイザ型

ハイパーバイザ型は、ホスト OS を使わず、物理マシンに直接仮想化ソフトをインストールするタイプのものを指します。物理マシンには、あたかも OS であるかのようにハイパーバイザ型仮想マシンを直接インストールします。仮想化ソフトそのものの管理は、仮想化ソフト上に存在している管理専用の仮想マシンから行うか、または管理用の別のマシンを準備してそこに管理用ソフトウエアをインストールすることによって行います。仮想マシン上に管理用ソフトウエアをインストールすることも可能です。主な製品としては、VMware vSphere、Microsoft Hyper-V、Citrix XenServer などがあります。

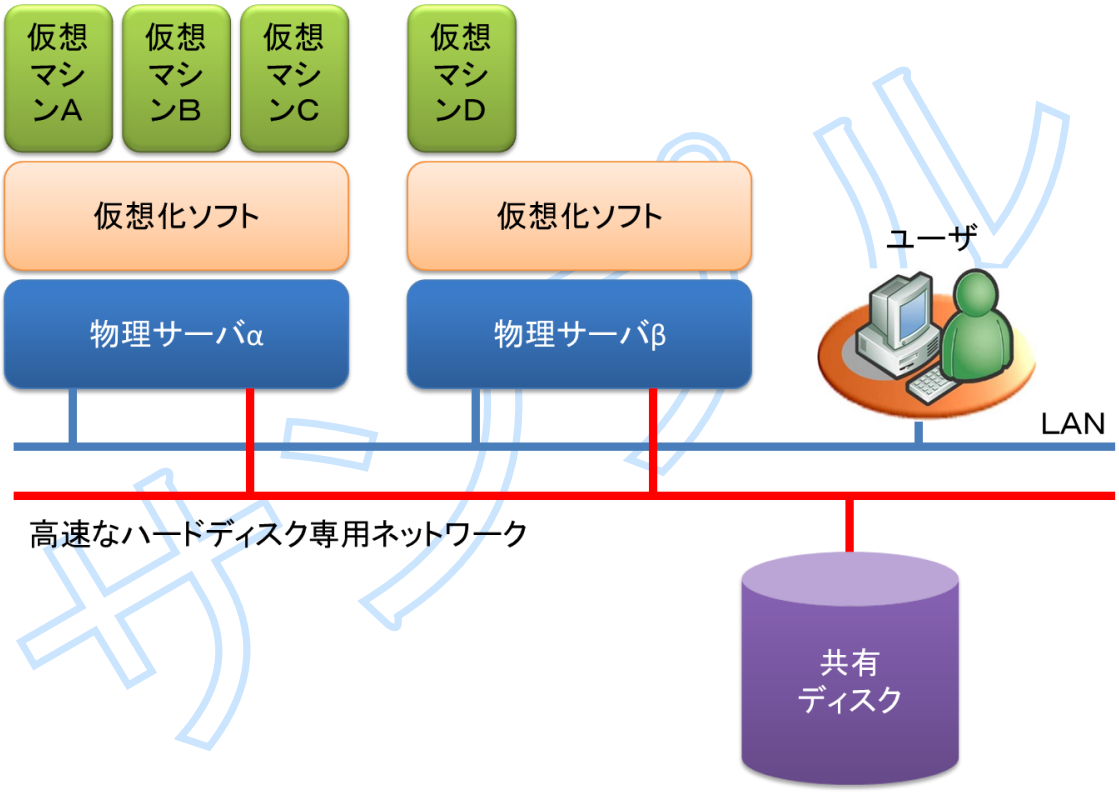


さらに、ハイパーバイザ型仮想化ソフトは、モノリシック型とマイクロカーネル型に分けられます。この 2 つの大きな違いは、デバイスドライバの構成です。モノリシック型は物理マシンをコントロールするためのデバイスドライバを自前で持つのに対し、マイクロカーネル型は管理専用の仮想マシンにインストールされるゲスト OS が持っているデバイスドライバを借りて物理マシンをコントロールします。VMware 製品はモノリシック型、Hyper-V や XenServer はマイクロカーネル型です。

ハイパーバイザ型の仮想化ソフトウエアの最大のメリットは、ライブマイグレーションが可能であるという点です。

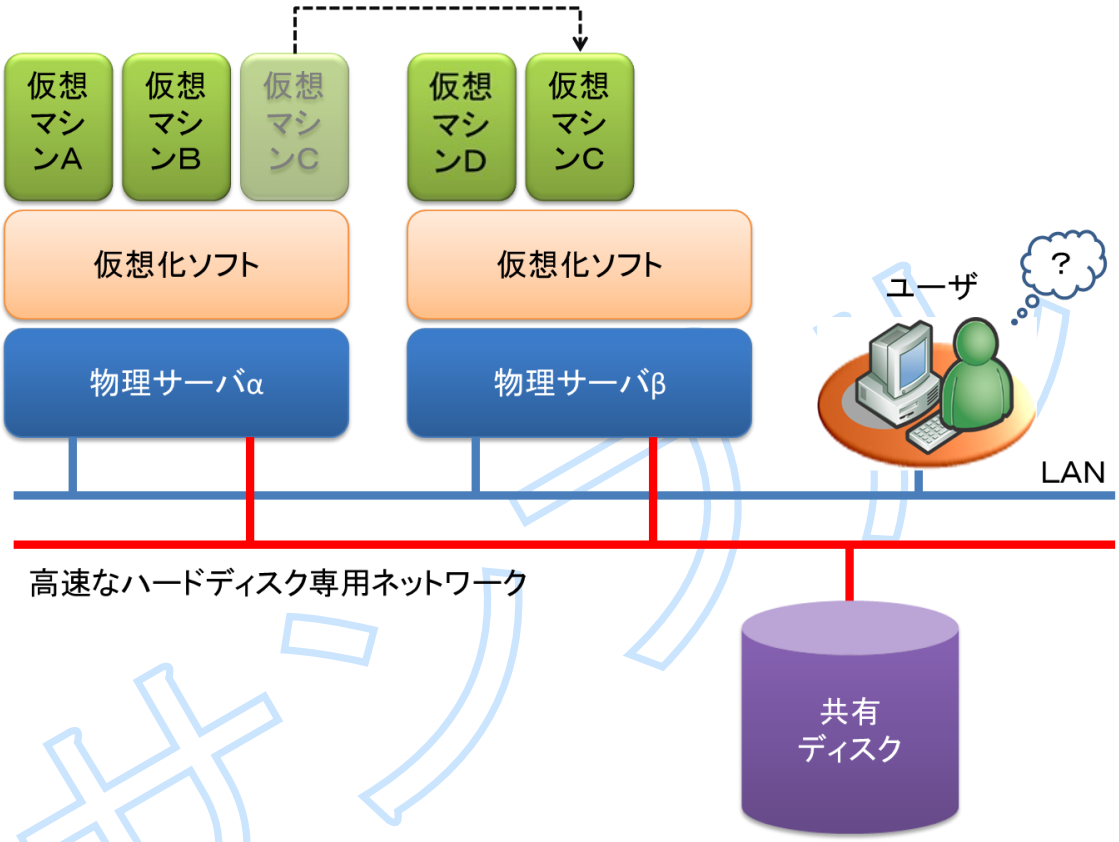
ライブマイグレーションとは、「仮想マシンを動作させたまま、他の物理サーバに移動する」技術です。ここでは技術的な詳しい説明は避けますが、代表的な仮想化ソフトウエアの多くはライブマイグレーションの機能を有しています。

準備としては、図 1 のようにハイパーバイザ型の仮想化ソフトを導入した物理サーバ 2 台(以上)と、両方の物理サーバからローカルディスクのように参照することのできる共有ディスクが必要です。



上図では、物理サーバ α 上で仮想マシンA、B、C が、物理サーバ β 上で仮想マシンDが動作しています。ネットワーク経由でサーバを利用するユーザからは、おそらく異なる 4 台のサーバ(A、B、C、D)が動作しているように見えることでしょう。

ライブマイグレーションは、これらの仮想マシンを「動作させたまま」物理マシン間を移動させてしまいます。下図では、仮想マシンCが、物理マシンαから物理マシンβに移動しています。簡単に言えば、物理マシンα上で動作している仮想マシンCのメモリの状態をかたっぱしから物理マシンβにコピーし（その間、物理マシンβ上の仮想マシンCはスリープ状態になっています）、最後に物理マシンα上の仮想マシンCをスリープ状態にして、即座に物理マシンβ上の仮想マシンを復帰させる、というような技を使っています。



ネットワーク経由でサーバCを利用しているユーザからは、ほんのわずかな間、サーバCのレスポンスが遅くなったと感じることでしょう。しかしそれはすぐに復帰しますので、ユーザはサーバCが一瞬スリープした（ましてや動作している物理サーバが移動した）とは気付きません。

ライブマイグレーション機能は、仮想化ソフトウエアを導入する最大のメリットです。ライブマイグレーションの機能は、具体的には次のようなメリットを生み出します。

- ●遊休資産を論理的に減らすことができ、リソースを効率的に使える
- ●ハードウエアメンテナンスのためにサービスが停止することを防げる
- ●耐障害弾力性の高い環境を構築できる
- ●拡張性の高いサーバ環境を構築することができる

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

（空白ページ）

サンプル

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

第5章 IT とコンピュータ

サンプル

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

5 - 1 IT におけるコンピュータの役割

これまでの章で見てきたように、IT 化においてコンピュータの役割は非常に重要になっています。



「IT＝パソコン（コンピュータ）」ではないとしても、IT 化を実現する手段としてコンピュータは不可欠です。

また現代はネットワーク抜きでコンピュータを語ることはできません。IT を実現するためにはネットワークおよびセキュリティは無視できません。

この章では、IT を支える基盤技術としてのコンピュータが現在のビジネスにどのように活用されているかを説明します。

■　インターネットのビジネスの活用

インターネット創生期には、インターネットは商用利用が禁止されていました。これは、初期のインターネットが、研究目的にのみ公開されていたからです。

その後、ISP（インターネット サービス プロバイダ）が共同でネットワーク管理を行うようになったり、商用利用の制限がなくなったりして、インターネットは急速に普及しました。

現在インターネット上で行われている代表的なビジネスには以下のものがあります。

B2C（Business to Consumer ：B to C）

企業（ビジネス）対個人（コンシューマ）の取引を「B2C」あるいは「BtoC」と呼びます。個人である一般利用者が Web ページにアクセスして商品を選び、クレジットカードや代金引換配達で支払いを行うような利用形態を総称した呼び方です。



商品は時計やカバンといった物理的なものから、音楽や映画、ゲームなどソフトウエア的なもの、クーポンの配布や電車、飛行機、ホテルの予約といったサービス的なものまで、さまざまなものがあります。音楽や映画、ゲームなどの、内容に主な価値があるものを「コンテンツ（中身）」と呼びます。また、コンテンツを記録したメディアを「コンテナ（入れ物）」と呼びます。例えば音楽がコンテンツなら、そのコンテナは CD です。また、放送も一種のコンテナです。

従来、コンテンツとコンテナは一体とされてきました。しかし、ネットワークを使うと、コンテンツとコンテナを切り離し、コンテンツだけを売買できます。その結果生まれたのが「コンテンツビジネス」です。

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

B2B（Business to Business ：B to B）

企業対企業の取引を「B2B」あるいは「BtoB」と呼びます。おおまかな利用方法は B2C と変わりません。しかし対個人との取引と異なり、企業間取引は発注量や金額が多くなったり、納期に厳しい制限があったりなど、対個人とは異なる要求が出てきます。また、注文処理や請求処理を自動化したいというニーズもあります。



多くの企業では経理処理にコンピュータを使っています。また、Web を使って注文する場合もコンピュータを使います。しかし、ソフトウエアの開発者が異なるとデータの表現形式が異なるため、ほとんどの場合、送られてきた注文情報をそのまま自社のコンピュータに取り込むことができません。担当者は Web での注文情報を、いちいちコンピュータに手で入力しなければならなかったのです。

そこで、コンピュータで情報を交換するときの共通フォーマットを決めようという動きが出てきました。そこで採用されたのが「XML（Extensible Markup Language）」というデータの意味や構造を記述するためのマークアップ言語です。

XML は、HTML と同様の形式ですが、拡張性に優れ、あらゆるデータとその意味を表現できます。企業間で XML の利用形式を決めておけば、発注に使ったデータをそのまま経理用として処理できます。現在は、XML そのものの標準化と並行して、業種ごとの意味情報の標準化も進められています。

XML を用いて複数の Web アプリケーションを連携して使用するしくみを実現するのが「XML Web サービス」（または単に「Web サービス」）です。例えば検索サービスと地図サービスを連携させ、検索したキーワードと同じ店名を持つ店舗を地図上に表示したり、その店舗の予約サービスを連携させて、地図上でその店舗を予約できたりするサービスが実現します。サービスを提供する側にとっては、インターネット上の便利なサービスを組み合わせて簡単にシステムを構築していくことができますし、利用者にとっては、1 つのサービスにアクセスするだけで自動的に別のサービスの恩恵が受けられるというメリットがあります。



Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

■　基幹系システム

「基幹業務」とは、その業務が止まってしまうと企業活動に支障をきたすような、ビジネス上不可欠な業務のことをいいます。

具体的には、商品やサービスを生産するために必要な素材などを購入する「仕入・購買」、実際に商品やサービスを作り出す「生産」、これらを市場に提供して対価を受取る「営業・販売」、企業の収益と資産を計算して税務署などに報告するための「財務会計」と「在庫管理」、さらにこれら企業活動を支える人材を管理する「人事」などの各業務のことです。
「基幹系システム」とは、これらの基幹業務をサポートすることを目的としてコンピュータ・システムの総称です。実際には、業務ごとに必要なハードウエアや、機能が異なるソフトウエアが用意されます。

基幹系システムの役割は、発注書や納品書の作成、在庫や顧客情報などの検索のほか、顧客や仕入先のデータや、その相手先と行った取引などのビジネスデータを管理することです。

基幹系システムは企業のコンピュータ・システムの基盤であり、ここで企業が戦略的に活用していくための基礎となるデータが作成・管理されているのです。

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

■　情報系システム

「情報系システム」とは、基幹業務から得られた情報を元にさまざまな分析を行い、今後の会社の方針を立案したり、競合他社に対する優位性を保つための方策を練ったりするために必要な機能を提供するためのコンピュータ・システムのことを表します。

情報系システムは、情報分析に基づく知的業務をサポートします。こうした業務は、決められた計算式に当てはめればよいというものではありません。これが基幹系システムとの違いです。

多くの企業には、本社があり「営業本部」や「商品本部」などの本部組織と事業ごとに組織化した事業部性があります。例えば本部組織は企業が持つ情報を分析し、次のような業務を行っています。

①営業本部では、基幹系システムや、独自の調査を行って売上データを入手する
②そのデータから顧客層（年齢、性別など）に応じた商品の売上動向を分析する
③その情報を分析し、顧客層別の販売キャンペーンを立案し実施する

また、商品本部では、次のような業務が行われています。

①同様に売上データを入手する
②顧客層別ではなく、商品別の売上動向を分析し、売れ筋商品を見極める
③得られた情報を使って、商品の改良や新商品の開発に活用する

部署によってデータの活用方法やその手段はさまざまです。いずれの場合も、最初のステップは基幹系システムから情報を得ます。しかし、2 番目以降のステップはその都度異なります。情報系システムは 2 番目のステップである、情報を分析し、ビジネス戦略の計画と実行を行う知的業務をサポートします。



Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

ERP（Enterprise Resource Planning）

このように、基幹系システムと情報系システムは密接な連携が必要です。そこで、最初から両者を統合したシステムを使おうというアイデアが生まれました。これが ERP（Enterprise Resource Planning: 経営資源利用計画）です。ERP は、会社がその活動の全てをコンピュータ化するために導入します。一般に、ERP の機能を活かすには、業務形態も ERP に合わせて変更する必要があります。

情報系システムは、基幹系システムと異なり、業務組織ごとにそのアプリケーション機能が切り分けられません。企業の戦略を立てるためには、製品ラインや部門ごとの分析ではなく、トータルな分析を必要とするからです。

通常、情報系システムは、それ自体で新しく情報を生み出すことはありません。情報系システムは、基幹系システムが作り出した原始データをさまざまな角度から集計してユーザに報告するインフォメーションシステムなのです。

1970 年代にコンピュータが各企業に導入されたときは「EDP（Electronic Data Processing）」と呼ばれていました。これは、コンピュータの用途が基幹業務に偏っていて、売り上げなどのデータを単純に集計することが主目的だったからです。

1980 年代に入ると、情報系システムの重要性が認識されるようになりました。基幹系システムから得られたデータを総合的に分析する必要が生まれたのです。このような情報分析を行うコンピュータ・システムを「IS（Information System）」といいます。

例えば、小売業者が自社商品の商品別購買者層を分析したいと思った場合は、販売管理システムに売上を計上する際に、購入者の年齢、性別、単身者か家族構成はどうなっているのか、といった顧客の情報が入力されデータとして保有されていなければなりません。これらのデータが記録されていないと、購買者の分析を行うことはできません。つまり、基幹系システムの在り方が情報系システムの限界を決定します。これは情報系システムが基幹系システムの次のステップであるからです。

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

情報系システムにおける分散処理

情報系システムを利用するユーザは、試行錯誤をしながら、さまざまな切り口からデータを分析します。例えば、ある月の企業全体の売上が極端に落ち込んだとします。営業本部のスタッフはこの原因が特定の商品にあるのか、営業所にあるのか、あるいは市場全体の動向にあるかといった具合に分析を繰り返していきます。このような業務をサポートするために、情報系システムはオンラインリアルタイム処理をベースに構築されています。

情報系システムは、基幹系システムに保管してあるデータを必要に応じて分析し、何らかの意思決定を行う際の参考にするためのシステムです。情報系システムは基幹系システムとは異なり、処理要求が多種多様で、顧客、商品、場所といった多次元のビジネス的な切り口での分析を必要とします。基幹系システムの OLTP に対して、このような処理形態を「OLAP（On-Line Analytical Processing）」といいます。

情報系システムは複数のデータベースを参照する必要があり、また部門やニーズによって処理内容が多彩に変化するため、集中型を採用することはあまりありません。多くの場合「分散処理システム」を採用します。分散処理システムでは、複数のコンピュータが適切な機能を分担し、協調して機能します。

分散処理システムは、以下の 2 つの要素から構成されます。

■ サーバ

利用者が共通して利用するシステム機能やデータを管理し提供するコンピュータ、またはソフトウエアのことです。クライアントからの要求を受け付けて、処理の結果をクライアントに返します。多くのクライアントから利用されるため、高い性能が必要です。

■ クライアント

クライアントとは、システムの利用者が直接操作するコンピュータ、またはソフトウエアのことです。情報系システムの利用者は、必ずしもコンピュータの専門家ではないので、簡単に操作できることが重要です。クライアントはサーバに処理要求を出します。実際の処理はサーバが行います。クライアントはその処理結果を受け取り、利用者にわかりやすい形式で表示します。最近では、クライアントとして Web ブラウザを使い、クライアント側の処理をほとんど行わないか、あるいは表示だけに専門化したシステムもあります。

1 台のサーバに複数のクライアントが接続して利用するような形態を、2 層クライアント／サーバ・モデルといいます。

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

3 層クライアント／サーバ・モデル

サーバをプレゼンテーション層、アプリケーション層、データ層の 3 つの役割に分離する場合もあります。これを、「3 層クライアント／サーバ・モデル（3 Tier Client Server Model）」といいます。現在インターネットで用いられているクライアント／サーバ・モデルのほとんどは、この 3 層クライアント／サーバ・モデルです。また、最終的なクライアントに Web ブラウザを使用するモデルのことを、特に Web3 階層クライアント／サーバ・モデルといいます。ここでは、現在一般的な Web3 階層クライアント／サーバ・モデルについて解説します。

Web3 層クライアント／サーバ・モデルは、以下のような役割分担を行います。

■プレゼンテーション層

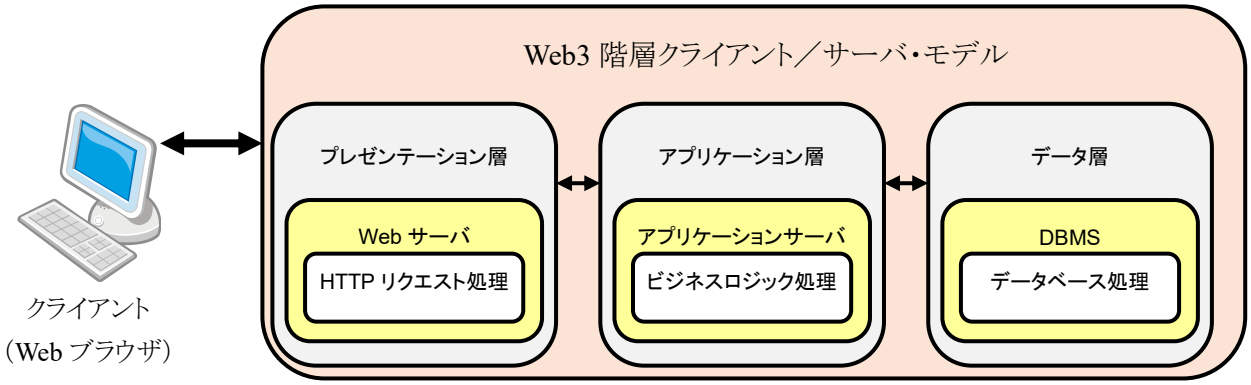
クライアントである Web ブラウザからの HTTP リクエストを受け付け、必要に応じてアプリケーション層に処理を依頼する部分です。また、アプリケーション層の処理結果を受け取り、その情報を元にページを作成して Web ブラウザに渡すという処理も行います。

■アプリケーション層

一般に WAS（Web Application Server）といいます。目的ごとに必要なプログラムを用意する部分です。例えば、同じ売上データを使う場合でも、支払い処理と売り上げ予測では用意すべきプログラムが違います。ビジネスロジックは互いに独立性を高めた状態で作成します。後から新たな機能（分析機能など）を追加するような場合、従来のシステムに影響が出ないようにするためです。
多くの場合、ビジネスロジックの処理に必要なデータをデータ層にアクセスし、データの追加・検索・更新・削除を行います。

■データ層

大量のデータを保持し、必要に応じてビジネスロジックに提供する部分です。データ層は、単なるデータの入れ物として振る舞います。業務形態が大きく変わらない限り、データベースを変更する必要はありません。現場で必要とされる形式への加工はビジネスロジックで行います。



現在では、3 層クライアント／サーバ・モデルのほとんどは、クライアントに Web ブラウザを使用した Web 3 層クライアント／サーバ・モデルです。Web 3 層を中心としたクライアント／サーバ・モデルには、集中処理型システムにはないメリットがあります。

■低価格

サーバは高価ですが、クライアントは一般的な PC でよいため、安価に構築できます。特に Web 3 層モデルの場合は、クライアントに Web ブラウザを使用するため、クライアント用のアプリケーションを開発する必要がない、サーバ側のプログラムを変更してもクライアントの変更は必要ないという大きな利点があります。

■操作性

クライアントには操作性に優れた GUI を採用します。Web 3 層モデルの場合、クライアント側で用いるプログラムは Web ブラウザです。コンピュータ・システムであれば標準で搭載されており、多くの人がその操作方法を知っているため、導入コストだけでなく教育コストも削減できます。

■性能

クライアント側は入力や表示、サーバとのデータのやりとりにのみ責任を持ちます。そのため高性能なシステムは必要ありません。サーバはさまざまなデータ処理を行う必要がありますが、逆に入力や表示といった仕事を行う必要がないので、その分だけ作業負荷が軽減されます。軽減できた負荷は、クライアントに対する処理速度の向上に回すことができます。作業を分担し、それぞれの作業に適したハードウエアを使用することで、性能の向上が期待できます。また費用対効果の向上も期待できます。

Web アプリケーション利用の実際

1990 年代後半から、必要なアプリケーションをサーバ側にのみインストールし、利用者はクライアントからネットワーク経由でそのサーバに接続してアプリケーションを利用する、という形態が研究されてきました。従来はネットワークのスピードの問題やサーバの動作速度の問題でなかなか普及しませんでしたが、2000 年代に入ってからネットワークスピードの向上、サーバの動作速度の向上、ソフトウエア開発技術の進歩などで、アプリケーションをサーバ側に置いてクライアントからそれを利用する、という形態が徐々に普及し始めました。

やがて、インターネット経由でアプリケーションを提供するアプリケーション・サービス・プロバイダ（ASP）というサービス業者が姿を現しました。ソフトウエアは ASP が提供するサーバに Web アプリケーションとしてインストールされており、利用者はクライアント上の Web ブラウザからそのサーバに接続してアプリケーションを利用します。そして利用した期間や量に従ってサービス料金を支払うという形態です。このようにアプリケーションプロバイダが提供しているソフトウエアサービスを Web ブラウザ経由で利用する形態のことを、SaaS（Software as a Service：サース）と呼びます。SaaS は必要なときに必要なサービスを、社内でそのシステムを構築したり、アプリケーションを購入・インストールしたりする手間なく利用できるというメリットがあります。

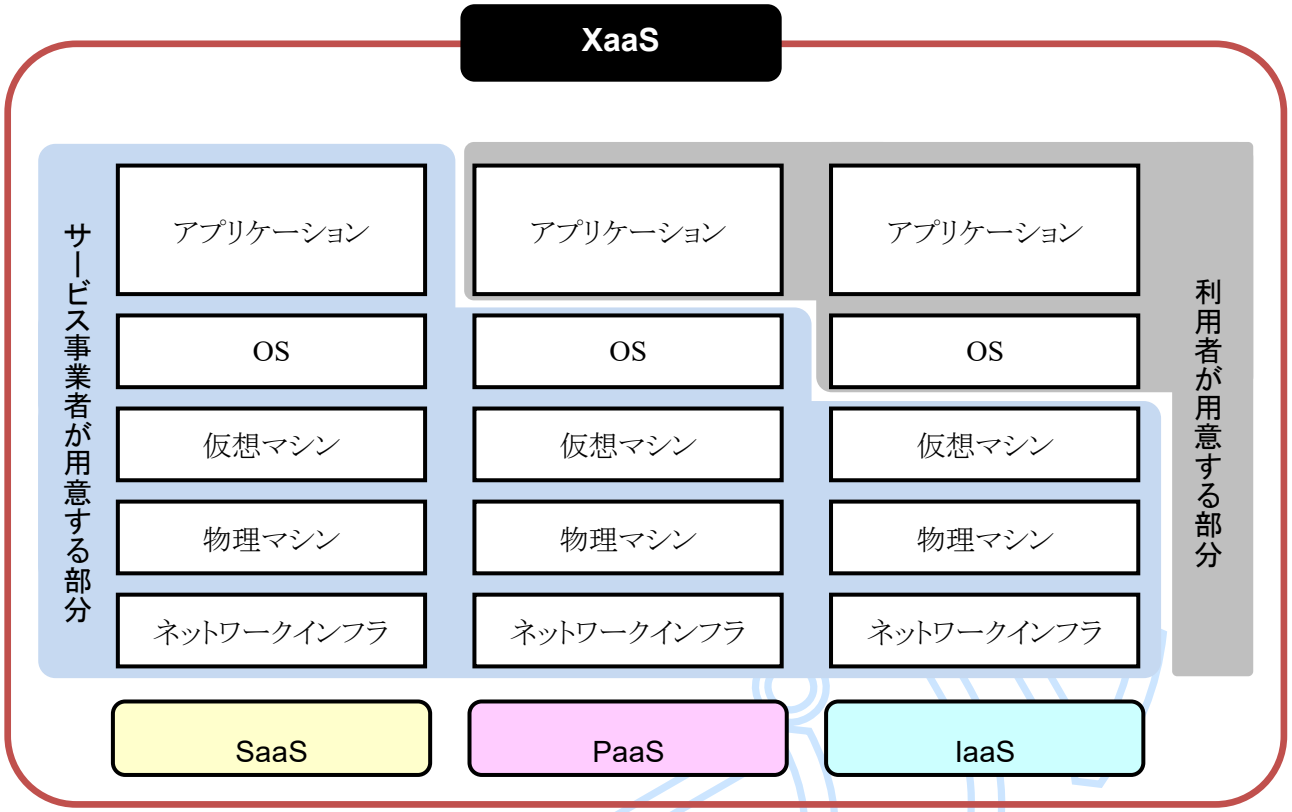
さらに最近では、「ネットワークの向こう側にあるサービスの提供を受ける」という形態のことを総称して「クラウド・コンピューティング」と呼ぶようになりました[8]。概念的には「ネットワーク・コンピューティング」と同義です。Web アプリケーションが提供しているさまざまなサービスをインターネット越しに利用する、という形態です。利用者は、インターネット上のどのサーバがどのようなサービスを提供しているか意識する必要はない、という特徴を持っています。前述の SaaS は、クラウド・コンピューティングの一種であるとみなされます。

最近では、SaaS の他に、PaaS（Platform as a Service）や IaaS（Infrastructure as a Service）と呼ばれるサービス形態も利用されるようになってきました。

SaaS はサービス事業者がプラットフォームやアプリケーション全体を用意し、利用者がそれを利用するというスタイルです。SaaS の場合、利用者は自分でアプリケーションを構築したりカスタマイズしたりすることができません。サービス事業者が用意したアプリケーションをそのまま使用するか、またはあらかじめサービス事業者と利用者との間で話し合った内容にカスタマイズしたアプリケーションを使用するしかありません。

[8] クラウド（雲）は、ネットワーク（通常はインターネット）を表します。古くからコンピュータ・システムのイメージ図では、ネットワークを雲のように表す場合が多く、それが由来であるといわれています。

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.



それに対し PaaS は、サービス事業者がアプリケーションを構築・運用する環境部分を提供するというスタイルです。サービス事業者は、IT インフラストラクチャ、OS、アプリケーション開発環境や運用環境などを提供します。利用者は PaaS の環境に Web ブラウザを用いて接続し、そこに用意されている開発環境や運用環境を用いてアプリケーションを開発したり、開発したアプリケーションの保守を行ったりします。SaaS に比べ、利用者が独自のアプリケーションを開発できるというメリットがあります。

また、IT インフラストラクチャや仮想マシンなどの基盤部分のみをサービス事業者が提供する IaaS と呼ばれる形態も存在します。利用者はその IT インフラストラクチャ上に、自分で用意した OS や開発環境、アプリケーション、データベースシステムなどを搭載して利用します。利用者側にも高い IT の知識や技術が必要な反面、IT インフラストラクチャそのものの管理をサービス事業者に任せることができ、また PaaS と違って自由なソフトウエア環境を選択することができるというメリットがあります。

図に中に示した XaaS（エクサース、ザース）は、インターネットを介してコンピューティングにおけるリソースを提供するサービスの総称と考えるといいでしょう。「X」という文字の部分は、IaaS ならば「I」に置き換わり、SaaS ならば「S」に置き換わるというように、未知のアルファベットが入ることを意味しています。

その他の動向

Web アプリケーションはビジネスロジックが重要です。しかし、Web の技術を使う上での決定的な欠点があります。それは、クライアントとサーバの通信に HTTP または HTTPS が使用されることに起因しています。

HTTP や HTTPS は、1 つの Web ページの表示が終わったら、その都度クライアントとサーバ間の接続を切断します。従来の Web アプリケーションでは、サーバにリクエストを送信後、レスポンスを新たに Web ページとして受け取り、はじめて画面遷移が発生します。例えば、情報を入力するための Web ページが表示されている段階では、一度接続が切れているのです。その後利用者が必要な情報を入力して「登録」などのボタンをクリックすると、あらためてサーバとの接続を確立します。そのため、どうしても非同期通信にならざるを得ません。

そこで Ajax（エイジャックス）と呼ばれる技術が登場しました[9]。Ajax は通信結果に応じてタイムリーに表示ページの一部を書き換えるというアプローチを取る技術です。情報を登録する画面と、その入力した情報を元に処理を実行した結果を表示する画面を 1 つの画面として Ajax を利用して作成すると、画面遷移を伴わない動的な Web アプリケーションの製作が実現可能になります。例えば Web 検索に応用することで、従来は入力確定後に行っていた検索を、ユーザがキー入力をする間にバックグラウンドで絞り込み検索を行うことによってリアルタイムに結果を表示していくといったことも可能です。

また、Web ブラウザのみで動作し（別途追加のソフトウエアや Web ブラウザの拡張を必要としない）既存の技術の組み合わせであることも特長です。 Ajax は、Web ブラウザ内で非同期通信とインターフェースの構築などを行う技術の総称です。XMLHttpRequest（HTTP 通信を行うための JavaScript 組み込みクラス）による非同期通信を利用し、ダイナミック HTML を使用した動的なページ変化を実現しています。

現在では、サーバ側からクライアント側に更新されたデータを push 送信する、リバース Ajax と呼ばれる技術も広く実装されています。本来 HTTP や HTTPS は、クライアントからのリクエストにサーバが応えるという方式のため、サーバ側から能動的にクライアントにデータを送ることができません。しかしリバース Ajax の機能を用いれば、サーバ側のデータが更新された時にタイムリーにそのデータをクライアントに送信し、画面上に常に最新の情報を表示させておくことが可能になります。

[9] Ajax は、Web ブラウザ内で非同期通信とインターフェースの構築などを行う技術の総称です。XMLHttpRequest（HTTP 通信を行うための JavaScript 組み込みクラス）による非同期通信を利用し、ダイナミック HTML を使用した動的なページ変化を実現しています。

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

5 - 2 デジタルトランスフォーメーション

デジタルトランスフォーメーション（Digital transformation： DX）は、スウェーデンのウメオ大学のエリック・ストルターマン教授が 2004 年に提出した論文「Information Technology and the Good Life」の中で初めて提唱した概念です。

ストルターマン教授が 2004 年に提唱した DX の概念と近代の DX の定義には差異があります。
ここでは、近代の DX について解説します。

■ 近代の DX

2010 年の半ば頃から、DX はストルターマン教授が提唱した概念を拡張する形で、さまざまな解釈が加えられ始めました。現時点では、DX に関する明確な定義は存在しません。
代表的な団体が DX をどのように定義しているかを紹介します。

IDC Japan

IDC Japan は、2018 年に DX を次のように定義しています。

> 企業が外部エコシステム（顧客、市場）の破壊的な変化に対応しつつ、内部エコシステム（組織、文化、従業員）の変革を牽引しながら、第 3 のプラットフォームを利用して、**新しい製品やサービス、新しいビジネスモデルを通して**、ネットとリアルの両面での**顧客エクスペリエンスの変革を図ること**で**価値を創出**し、**競争上の優位性を確立すること**を指す。
>
> （IDC Japan 株式会社、用語解説一覧 https://www.idc.com/jp/research/explain-word より引用）

ここに述べる「第 3 のプラットフォーム」とは、上記に述べたモビリティ（モバイル端末）、ビッグデータ／アナリティクス、クラウド、そしてソーシャル技術に加え、AR/VR、IoT、認知システム／AI、ロボティクス、3D プリンティングなども含めた技術を活用したコンピューティング環境のことを指します。

経済産業省

経済産業省は、2018 年に DX を次のように定義しています。

> 企業がビジネス環境の激しい変化に対応し、データとデジタル技術を活用して、顧客や社会のニーズを基に、製品やサービス、ビジネスモデルを変革するとともに、**業務そのものや、組織、プロセス、企業文化・風土を変革し、競争上の優位性を確立すること**。
>
> （デジタルトランスフォーメーションを推進するためのガイドライン Ver.1.0 2018 年 12 月 より引用）

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

情報処理推進機構

また、独理行政法人 情報処理推進機構（Information-technology Promotion Agency：IPA）は、2020 年に DX を次のように定義しています。

> デジタル技術で**ビジネスモデルや働き方を変えること**。
>
> （YouTube 「IPA Channel」 デジタルトランスフォーメーション（DX）ってなんだ より引用）

本書での DX の定義

上記の例を踏まえ、本書では DX を次のように定義します。

> 既存のデジタル技術や第 3 のプラットフォームなどを基盤とし、次に述べる目的を果たすことによって、企業の競争優位性を確立し、社会に新たな基盤を生み出すこと。
>
> - 新たな顧客体験（Customer Experience）の創出
> - 新たなビジネスチャンスの創出
> - 企業文化、風土、組織、ビジネスモデルの変革
> - 既存の枠組みにとらわれない、変化への柔軟な対応

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

■　（参考）DX とデジタル化との違い

DX 以前にも、業務データをデジタル化する動きはありました。デジタル化は、大きく 3 種類に分類することができます。

デジタイゼーション（Digitization）

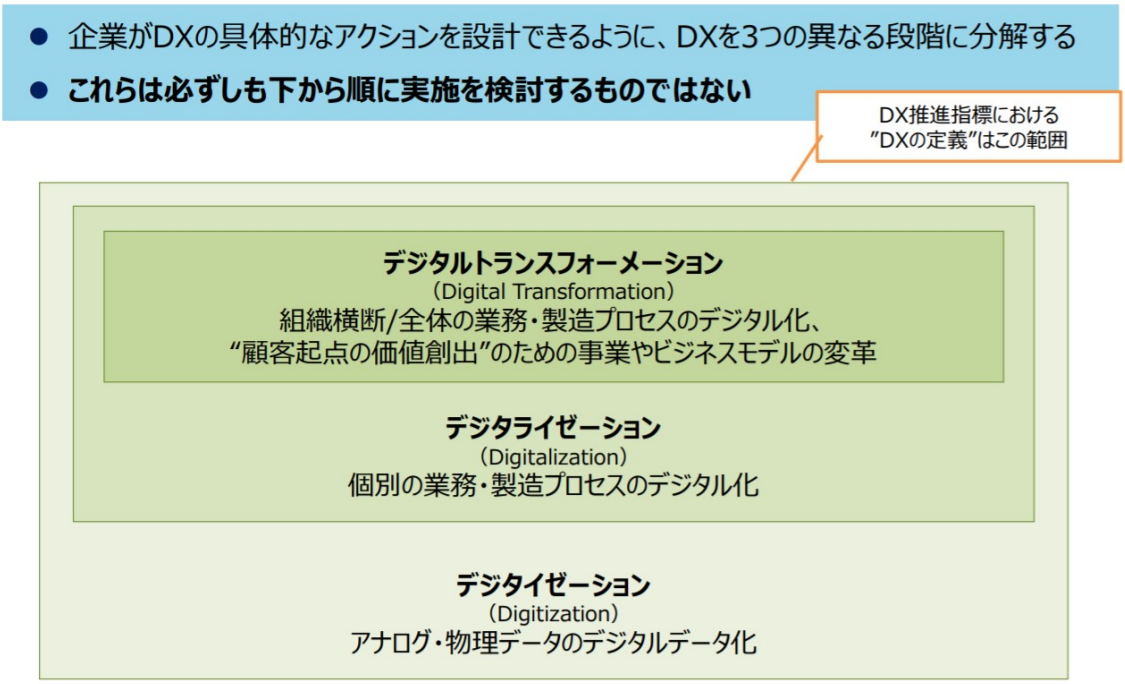
デジタル技術を活用することで、**ビジネス上のデータや人力での作業をデジタル化すること**です。たとえば、紙の書類をスキャンしてデジタル化したり、電話で受け付けていた注文を電子的なフォームに入力してもらってメール送信したりといったことなどがこれにあたります。デジタイゼーションは、手作業で行っていたことをデジタル化したり、現実世界の手作業をそのままデジタル世界に置き換えたりすることで、業務効率を高めたり、コスト削減を図ったりすることを狙うものです。

デジタライゼーション（Digitalization）

現実世界のビジネスプロセスをデジタル化し、**生産性を高めたり、ビジネスチャンスを広げたりすること**です。たとえば、ヘルプデスクにおける電話による電話問い合わせを Web から受け付けるようにし、受付と同時に担当者を割り当て、その担当者が問い合わせ業務に応じる、というような活動がこれにあたります。デジタイゼーションとの最大の違いは、業務プロセスがデジタル化されているという点です。担当者の対応内容はログとして記録に残り、データベース化され、類似の問い合わせの際に有効活用できるよう、検索性の高いシステムに蓄積されます。問い合わせ件数の多い事案は改善を必要とする項目として取り上げられ、業務改善が進みます。このような仕組みを作って業務全体の付加価値を高めるのがデジタライゼーションであるといえます。

デジタルトランスフォーメーション（Digital Transformation）

DX は、デジタル技術を活用することで、**組織や業界のビジネスモデルを変革し、デジタル化なしでは実現できなかった新たな価値を創造すること**です。たとえばヘルプデスクによる電話問い合わせを、AI を利用したチャットボットに置き換え、Web ベースで利用者からの軽微な問い合わせに応答します。問い合わせの頻度や回答はビッグデータとして取り扱い、チャットボットの回答改善に利用されます。チャットボットが対応できない問い合わせのみ、人間のサポートスタッフが対応します。ヘルプデスクの電話がつながりにくいという利用者の不満を解消するとともに、ヘルプデスク要員の労力を優先度の高いサポートに集中させることができる、というメリットも提供できます。DX は利用者に対する価値にとどまらず、DX を導入した企業にとっての経営上の価値にも影響を与えます。こうしたビジネスモデルそのものに変化を与えることで競争優位性を保つことが、DX が目指すものです。



（デジタルトランスフォーメーションレポート 2 中間とりまとめ 2020 年 12 月 28 日 より引用）

経済産業省では、DX が目指す「ビジネスモデルのデジタル化」は、プラットフォームのデジタル化、業務のデジタル化、製品／サービスのデジタル化の上に成り立つと説明しています。

	未着手	デジタイゼーション	デジタライゼーション	デジタルトランスフォーメーション
ビジネスモデルのデジタル化				ビジネスモデルのデジタル化
製品／サービスのデジタル化	非デジタル製品／サービス	デジタル製品	製品へのデジタルサービス付加	製品を基礎とするデジタルサービス デジタルサービス
業務のデジタル化	紙ベース・人手作業	業務／製造プロセスの電子化	業務／製造プロセスのデジタル化	顧客とのE2Eでのデジタル化
プラットフォームのデジタル化	システムなし	従来型ITプラットフォームの整備		デジタルプラットフォームの整備
DXを進める体制の整備	ジョブ型人事制度 リカレント教育	CIO/CDXOの強化 リモートワーク環境整備	内製化	

（デジタルトランスフォーメーションレポート 2 中間とりまとめ 2020 年 12 月 28 日 より引用）

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

5 - 3 コンピュータ・システムにおける「保証」

第 1 章でも述べたとおり、コンピュータ・システムの価値には有用性としての側面と保証としての側面があります。

有用性としての側面は、「そのコンピュータ・システムにはどのようなことができるのか、どのような機能があるのか」ということです。利用者のニーズに的確に応えられるシステムは、有用性が高い、といえます。一方、保証としての側面は、「コンピュータ・システムに備わっている有用性が、確実に利用者の元に届けられるか」ということです。反応速度が適切だったり、めったに故障が起きなかったり、故障が起きても迅速に復旧できたりするシステムは、保証されている、といえます。

コンピュータ・システムの「保証」としての側面には、次の 2 つの考え方があります。

- ●システムの性能面
- ●システムの障害耐性

■ システムの性能面の指標

システムの性能は、以下の 3 つの考え方によって測定します。

スループット（処理能力）

システムが単位時間あたりに処理する仕事の量をいいます。

レスポンスタイム（応答時間）

コンピュータに対する指示の入力が終わってから、画面に結果の 1 文字目が表示されるまでの時間をいいます。

ターンアラウンドタイム

コンピュータに対する指示の入力が始まってから、コンピュータがその処理を終え、結果の出力が終わるまでの経過時間をいいます。

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

■　システムの障害耐性

故障の頻度が多かったり、一度故障したらなかなか直らなかったりするのでは、そのコンピュータ・システムの障害耐性は低い、ということになります。

コンピュータ・システムの障害対策の度合いを評価する指標に、RASIS と呼ばれるものがあります。RASIS は、次に挙げる 5 つの項目の頭文字を組み合わせたもので、システムを設計する際に考慮すべき重要な項目です。

信頼性（Reliability）

故障が発生しにくい度合いを指します。これを測るための指標として平均故障間隔（MTBF：Mean Time Between Failure）があります。MTBF が長ければ長いほど、信頼性が高いといえます。

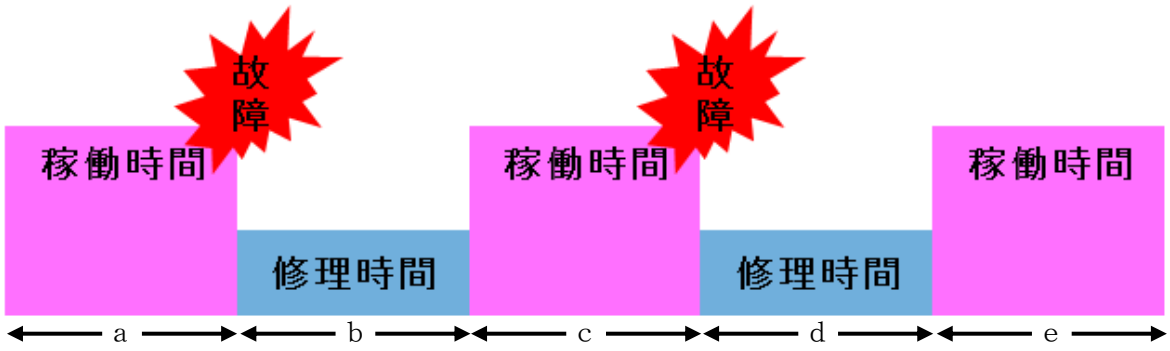
可用性（Availabirity）

システムが正常に動作している時間の割合を指します。これを測るための指標として、稼働率があります。例えば、本来 24 時間稼働しなければならなかったシステムが、実際には 21 時間しか稼働していなかった場合の稼働率は、21 ／ 24 ≒ 0.875（約 88%）ということになります。

また、稼働率を MTBF ／（MTBF ＋ MTTR）と表現する場合もあります。

保守性（Serviceability）

故障が発生した時の、復旧の迅速さの度合いを指します。これを測るための指標として平均修理時間（MTTR：Mean Time To Repair）があります。MTTR が短ければ短いほど、保守性が高いといえます。



MTBF：a、c、e の時間の平均

MTTR：b、d の時間の平均

稼働率：（ a ＋ c ＋ e ）÷（ a ＋ b ＋ c ＋ d ＋ e ）

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

保全性（Integrity）

破壊や過失によるデータ消失を未然に防いだり、故障や不具合があってもデータが矛盾を起こさず一貫性を保っていたり、データを消失しても容易に修復できたりする度合いを指します。

安全性（Security）

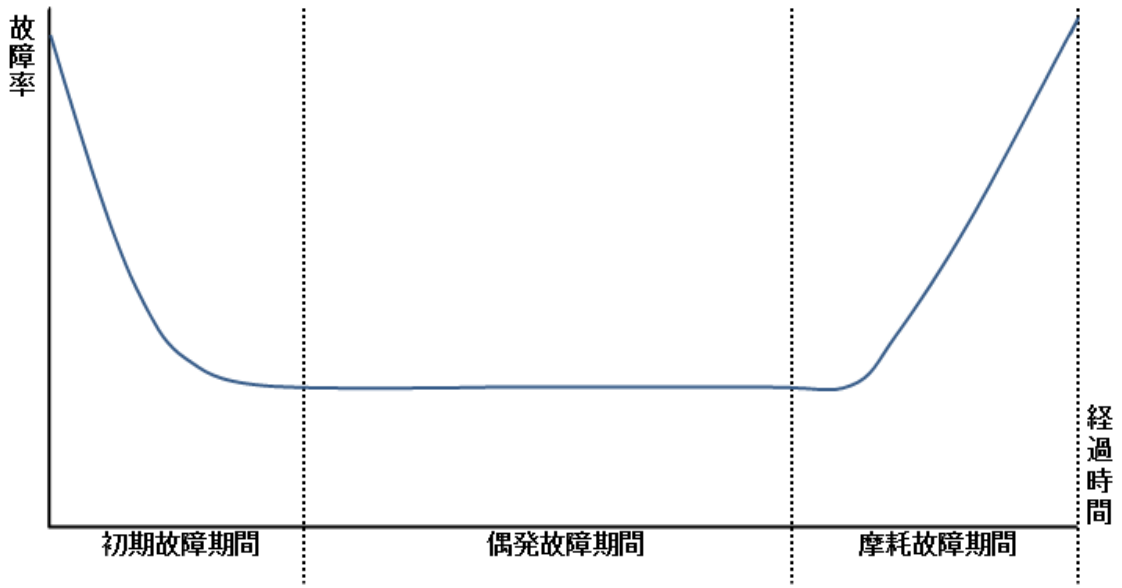
システムやデータに対する不正アクセスのされにくさの度合いを指します。

■　システムの障害耐性の指標

コンピュータや周辺機器などのハードウエアが、仕様で提示されている機能が実行できなくなったり、動かなくなったりすることを故障といいます。故障は、以下の 3 種類に分類できます。

1. 初期故障 ： 設計上の問題や製造過程の問題、ソフトウエアの不具合などのために起こる初期段階での故障
2. 偶発故障 ： 安定期に散発的に発生する故障
3. 摩耗故障 ： ハードウエアの寿命による故障

導入経過時間と故障率の推移をグラフに表すと、浴槽のような形になります。これをバスタブ曲線と呼んでいます。



Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

■　信頼性を高めるための設計

システムの一部に障害が発生して使用できなくなること、つまり障害は、避けて通ることはできません。そのため、障害が起きたとしてもシステムを使い続けることができるようにしなければなりません。高い信頼性を実現する方法には以下の 2 つがあります。

フォールト・アボイダンス

システムや周辺機器に障害が起きないようにすることです。十分なテスト、高品質のシステム構成を導入することによって、障害の発生頻度を最低限に抑えることを目指します。

フォールト・トレランス

工業製品であり人の作るものである限り、フォールト・アボイダンスには限界があります。そこで、システムに障害が発生しても、障害に耐えて稼働し続けるために、システム構成を多重化する方法です。サーバコンピュータなどの CPU やメインメモリ、HDD などを多重化して 1 年 365 日稼働し続けることを目指します。

現在は、このフォールト・トレランスが主流です。

以下に、このフォールト・トレランスの代表的な技術を紹介します。

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

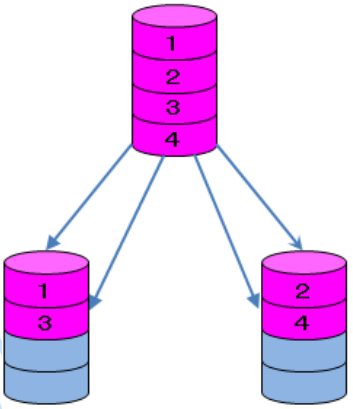
RAID（Redundant Array of Inexpensive（または Independent）Disks）

ハードディスクの多重化、つまり複数の磁気ディスク装置を組み合わせて、大容量の 1 つの磁気ディスクのように扱う技術です。その目的は、性能や信頼性の向上です。RAID のレベルは RAID0～RAID5、また、各ベンダが独自のレベルを設定しています。ここでは、ストライピング（RAID0）、ミラーリング（RAID1）、パリティつきストライピング（RAID 5）を紹介します。

●ストライピング（RAID 0）

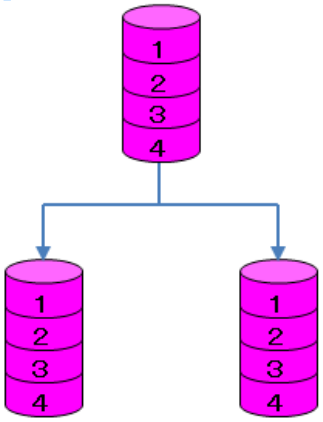
データをブロック単位に分割して、複数のディスクに分散して記録します。このように、複数のディスクにデータが帯状（ストライプ状）に配置されることから「ストライピング」といいます。

分散されたデータに同時に並行してアクセスできるため、アクセスが高速になります。ただし、単純にデータが分散しているだけなので、障害時にデータを再生成する機能はありません。このためディスクが 1 台でも障害を起こすとデータの読み書きができなくなります。



●ミラーリング（RAID 1）

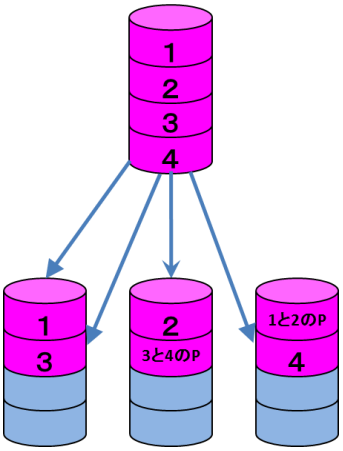
データをまったく同じように 2 台のディスクに書き込みます。2 台のディスクの内容が鏡に映したかのようにまったく同じになるので、「ミラーリング」といいます。アクセス速度の向上はありません。しかし、一方のディスクに障害が発生したときに、もう一方のディスクに自動的に切り替わってデータが処理されるため、ディスク障害によるデータの消失やシステムの停止が起こりません。一方、実際にはディスクの容量のうち半分しか使えないので、コストパフォーマンスは低くなります。



●パリティつきストライピング（RAID 5）

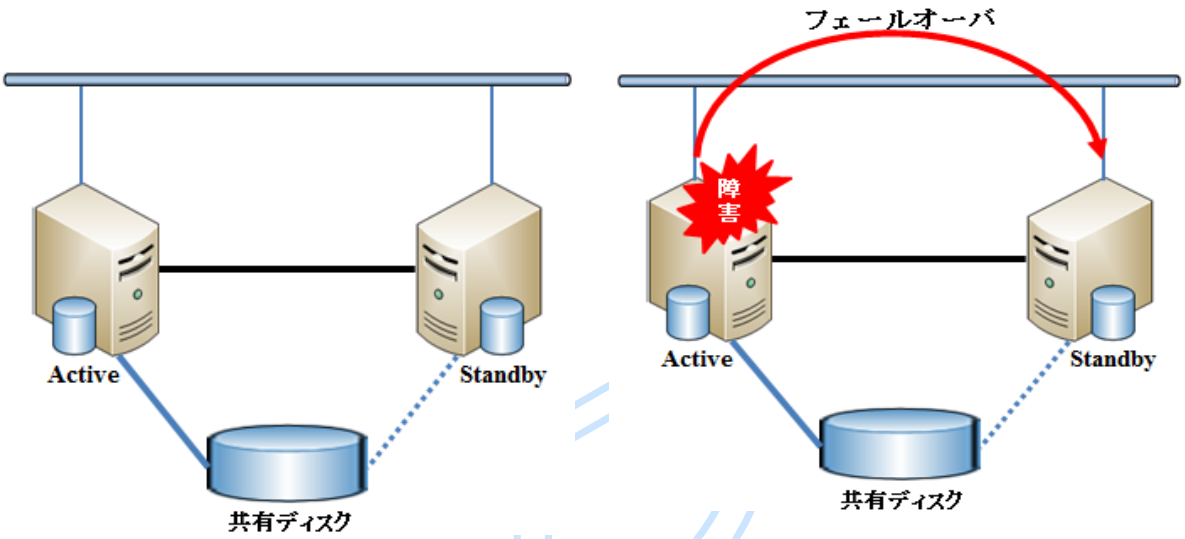
RAID 0 の技術に、ディスク装置が壊れた時のデータ復旧の技術を加えたものです。データをブロック単位に分割して、複数のディスクに分散して書き込みます。その際、2 つのブロックを用いてパリティを計算し、別のディスクに書き込みます。ディスク装置が壊れると、壊れていないディスク装置にあるデータを用いて、壊れたディスク上にあるデータを復旧することが可能です。

RAID 5 は障害対策になるだけでなく、障害が発生していない状態ではディスクの負荷分散（読み込み時）を行うので、読み取り速度は速くなります（データ書き込み時にはパリティ計算を行うので、パフォーマンスは低下する）。一方、障害が発生すると、パリティ情報から失われたデータを復元しながら読み込みを行う必要があるので、パフォーマンスは低下します。



クラスタ

クラスタとは、1 台のサーバを利用しているかのように動作する複数のサーバから構成されるサーバグループのことです。グループ内の 1 台のサーバに障害が発生した際に、別のサーバに処理を引き継いで、システム全体がダウンすることを避けるのが目的です。クラスタの方式にはすべての系統を常に稼働させる Active / Active 方式と、いくつかを待機状態にして、障害時に切り替えて処理を引き継ぐ Active / Standby 方式とがあります。また、共有ディスクを使用する場合もあります。



ホットサイト

災害を考慮したシステム構成で、災害を同時に受けないと考えられる遠隔地にバックアップセンタを設けて、常に待機させておきます。コンピュータセンタからシステム復旧に必要なデータを定期的に送信して災害に備えます。

コラム:フォールト・トレランスの考え方

フォールト・トレランスの実現方法として、耐障害性において 3 つの考え方があります。

- フェールセーフ

  システムの誤作動や障害発生した時に、その影響範囲を最小限にとどめること、安全に運用することを最優先する考え方。

- フェールソフト

  障害が発生した時に、システム性能の低下をある程度許容してでも、システム全体が停止しないように必要な機能を稼働させようとする考え方。

- フェールオーバ

  障害が発生した時に、他のシステムが処理やデータを自動的に引き継ぎ、稼働を続ける考え方。

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

■　データのバックアップ

バックアップとは、システム上のデータの消失に備えて、外部の記憶媒体にデータを複製しておくことです。データの消失は、システムの障害だけが原因ではありません。以下のようなケースに備え、しっかりと計画を立ててバックアップを取るようにします。

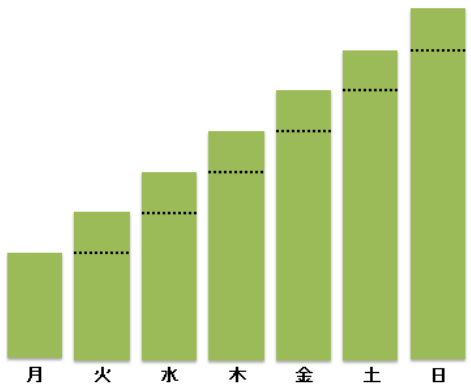
データ消失の例

データは、次のような事象で消失します。

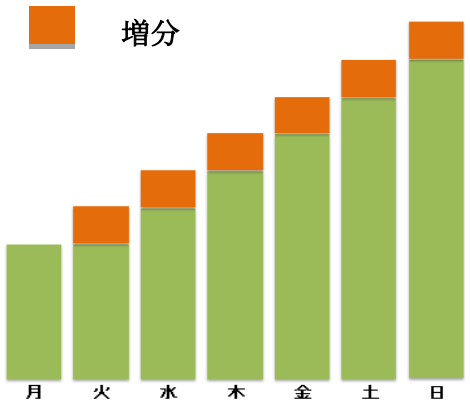
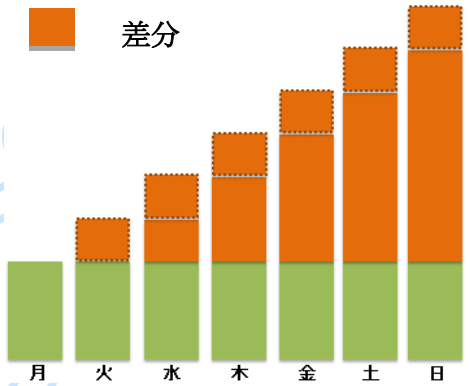
- ●人　為　的　なミス　:誤って削除、上書き、フォーマットしてしまう
- ●ハードウエア障害　:寿命による故障、破損
- ●ソフトウエア障害　:OS の不具合
- ●災　　害　:落雷、火災、洪水、地震
- ●犯　　罪　:コンピュータウイルス、サイバーテロ、盗難
- ●そ　の　他　:戦争、テロリズムなど。

バックアップの種類

バックアップ方法には、数種類あります。組織においてはその業務とバックアップ対象の重要性、コストなどを考慮して最適な方法を選択する必要があります。また、複数の方法を組み合わせることも考えます。

種類	特　徴	イメージ
フル・バックアップ	・毎回対象となるデータ全てをバックアップ ・バックアップに要する時間が長くなる ・復元作業は 1 回の読み出しで終了 ・バックアップファイルを上書きせず、 毎回のバックアップファイルを全て残す	

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

増分バックアップ	・初回にフルバックアップを取り、以降は変更部分だけをバックアップ ・フルバックアップを取る日以外は、変更分だけをバックアップするので、バックアップに要する時間は最少となる ・復元作業は、フルバックアップを取った部分と、その日までの増分を読みだすため、作業時間は最大になる	
差分バックアップ	・初回にフルバックアップを取り、以降はフルバックアップ以後に変更した部分をバックアップ ・復元作業は、フルバック部分と最新の差分バックアップを読み出す 2 つの工程で終了	

アーカイブ

使用頻度が低いデータや、一定期間の保存を義務付けられたデータを、外部記憶媒体に移し替えて長期に保存する目的で作成されたバックアップのことを、特に**アーカイブ**といいます。近年では、ネットワーク経由でやりとりする目的で、複数のファイルをひとまとめにしたものもアーカイブといいます。

一般にバックアップは、バックアップをとったファイルが（バックアップをとった時点で）すでに壊れていたり、ウイルスに感染していたりした時の用心として、数世代にわたって保存しておくほうが望ましいとされています。しかし、ある程度世代の古いバックアップは、新しいバックアップが存在することを機に削除することもあります。

一方アーカイブは原則として、一度作成したらそれを削除したり、別のアーカイブで上書きしたり、という運用はしません。複数の世代のアーカイブを残す場合は、保存先を変えたり、アーカイブのファイル名を変えたりして対応します。

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

第6章 ネットワークとセキュリティ（付録）

サンプル

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

6 - 1 ネットワークの重要性



コンピュータは単独で動作しているものもありますが、そのほとんどは何らかのネットワークに接続して稼動しています。携帯電話も一種のコンピュータ端末と考えれば、コンピュータとネットワークとの関連と重要性が理解しやすいでしょう。

1982 年に創業した Sun Microsystems 社は、「コンピュータはネットワークに接続されてはじめてその真価を発揮できる。ネットワークこそがコンピュータだ[10]。」という理念を提唱しました。現在のようにコンピュータがネットワークに接続されることが当たり前ではなかった 30 年前にこのような理念を掲げていたのは非常に先見性があったといえます。

コンピュータがネットワークに接続されていればデータ転送が容易に行えるため、より高度な機能を比較的簡単に実現できます。

ネットワークの目的は主として以下のものがあります。

■他のコンピュータとデータを交換する

電子メールやファイル転送はデータ交換の一例です。携帯電話も音声データを「交換」することで通信していると考えられます。また、インターネットで Web ページを閲覧するのもデータ交換と考えられます。

■他のコンピュータの機能を共有する

ネットワークを利用すれば、物理的に離れているコンピュータや周辺機器を利用できます。たとえば、ネットワークに接続しているプリンタを複数台のコンピュータで共有したり、インターネット上にあるコンピュータのハードディスクに格納されているデータをネットワーク経由で参照したりします。

今では、家庭用ゲーム機もネットワークに簡単に接続できるようになりました。家庭用ゲーム機同士で通信対戦を行ったり、インターネットを介してコミュニケーションを図ったりできるようになっています。

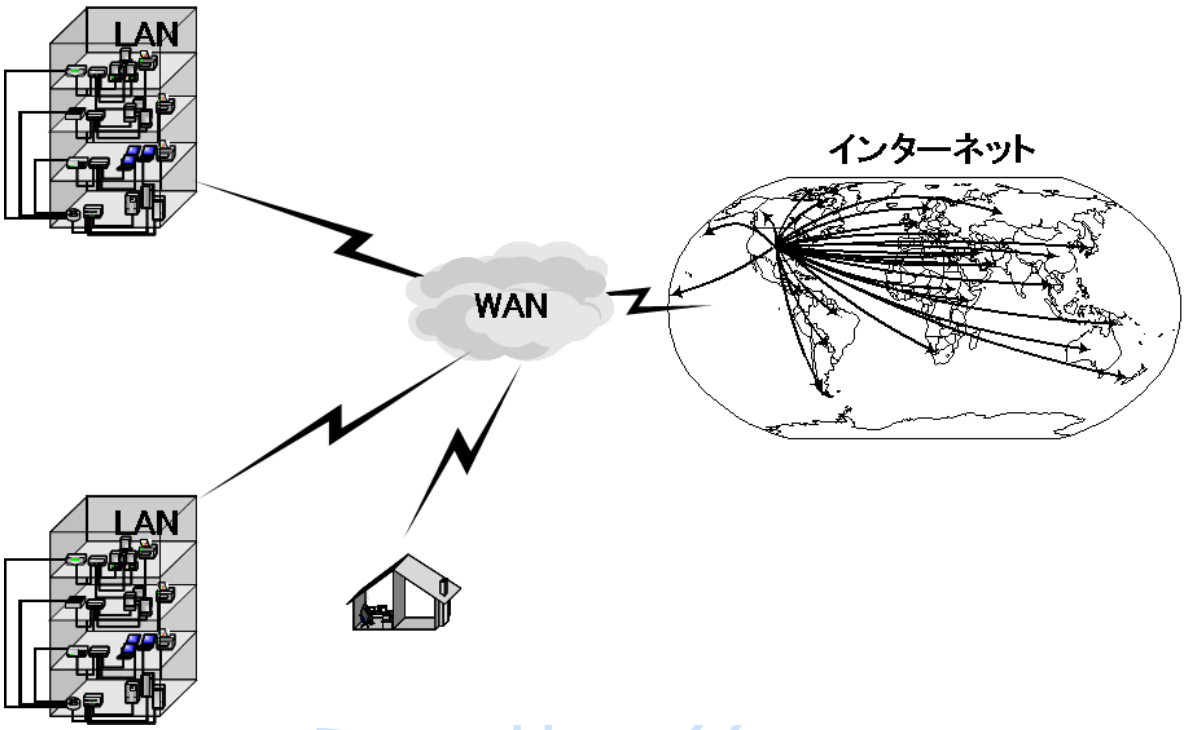
さらに最近ではグリッド・コンピューティング（複数のコンピュータの CPU の空き時間を利用することで、より高度な機能を実現すること）や、クラウド・コンピューティング（ネットワークをクラウド=”雲”のように捉えて不特定多数のコンピュータのまとまりとみなし、そこから提供されるサービスを実現すること）に代表されるネットワーク・コンピューティングによって、そのから受ける恩恵は計り知れないものとなっています。

[10] 原文は「The Network is The Computer.」

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

■ LAN と WAN

ネットワークは大きく LAN（Local Area Network）と WAN（Wide Area Network）に分類されます。



LAN（Local Area Network）

「構内通信網」と呼ばれており、オフィスビル内などの比較的狭い範囲での通信網のことを意味します。最近では、遠く離れた本社と支社間に仮想的な LAN を構築する場合もあります（これは「仮想 LAN」と呼ばれます）。

WAN（Wide Area Network）

「広域通信網」と呼ばれており、都市間などの遠距離通信を意味します。日本国内においては、「電気通信事業者（キャリア）の通信回線網を利用したネットワーク」を指します。遠隔地の LAN 同士を接続する目的で使用します。

LAN はケーブル工事などの初期コストがかかりますが、運用コストはほとんどかかりません。それに対して、WAN は初期コストに加えて、毎月の運用コストもかかります。

現在、ビジネスで使われているコンピュータは、ほとんどが LAN に接続され、ファイルやプリンタを共有しています。

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

■　インターネットとプロトコル



現在インターネット（The Internet）と呼ばれる通信網は 1960 年代に米国で実験が開始された ARPAnet と呼ばれるネットワークを原型としています。

ARPAnet は、米国国防総省がスポンサーとなった官産学協同のプロジェクトによって生まれたインターネットの原型となったネットワークです。

当時 ARPAnet では、ネットワークをより効率的に利用するためのプロトコルとして「TCP/IP」が開発されました。この TCP/IP が今でもインターネットで利用されています。

「プロトコル」とは、コンピュータ同士を接続してデータやメッセージをやり取りするための手順や約束事を決めたものです。通信手順、または通信規約とも呼ばれます。

しかし、ARPAnet は、誰でも接続できるわけではありませんでした。そこで、全米科学技術財団（NSF）が大学間を接続するネットワークの構築を開始しました。これを NSFnet と呼びます。NSFnet と ARPAnet の両方に接続された大学もあったので、そこを中継点として両者が結ばれました。

その後、ARPAnet の機能は NSFnet に移行し、さらに NSFnet の運営はいくつかの民間企業に分割委託されました。これが現在のインターネットです。インターネットへの接続サービスを行っている組織を「ISP（Internet Service Provider）」といいます。

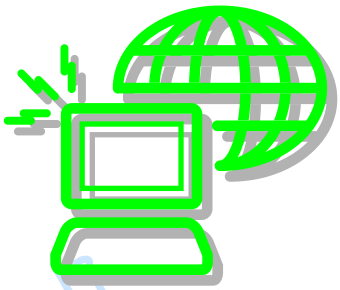
インターネットは、複数の企業が管理するネットワークの集合体です。さまざまなハードウェアベンダのさまざまな機器が接続されています。それでも自由に通信ができるのは、すべての機器が TCP/IP という共通のプロトコルを使っているからです。

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

■　インターネット上のアプリケーション

インターネット上ではさまざまなアプリケーションが利用されていますが、その中でも特に重要なものが WWW、電子メール、DNS です。

WWW（World Wide Web）



1989 年にスイスの「CERN（ヨーロッパ素粒子物理学研究所）」Tim Berners-Lee 氏が他の研究者の論文閲覧システムとして開発したのが始まりです。もともとは CERN の研究者が、他の研究者の論文や文献を簡単に閲覧する目的で開発されたと言われています。

WWW は相互に関連性のある文書を提供します。関連情報を参照するため機能を「リンク」と呼び、リンクを含む文書を「ハイパーテキスト」と呼びます。

Web は、サービスを提供するサーバ（Web サーバ）に「HTML（Hyper Text Markup Language）」と呼ばれる共通の言語で記述された情報を蓄積します。

HTML は、通常のテキストに「タグ」と呼ばれる書式属性やリンク情報を定義する文字列を埋め込んだものです。サービスを利用するクライアント側では「Web ブラウザ」と呼ばれるソフトウエアを使用します。Web ブラウザは HTML ファイルを解釈し、適当な形式に変換して表示します。

HTML は、単なる情報表示のほか、サーバ側やクライアント側でプログラムを実行する機能も備えています。このような機能を利用したシステムを「Web アプリケーション」といいます。不特定多数の人が自由に書き込みを行うことのできる「掲示板」、ショッピングサイトで購入したい商品を選択する際に利用する「ショッピングカート」、日記のように使える「ブログ」や「ツイッター」などは、すべて Web アプリケーションです。

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

電子メール

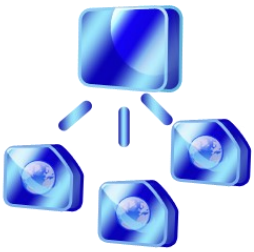


電子メールはインターネットの初期の段階から存在するアプリケーションです。個人または特定多数に対して、手紙に似たイメージでメッセージを送ることができます。元来電子メールには到達保障がありませんが、現在のネットワークではメールの紛失はほとんどありません。

電子メールは、メールの送信元から受け取り、送信先へ転送するサービスを提供する「メールサーバ」とよばれるソフトウエアと、メールを送受信するための「メールクライアント」と呼ばれるソフトウエアを使用します。

また、電子メールの仕組みを応用し、不特定多数の人でメッセージを共有できるようにしたものに「ネットワークニュース」があります。

DNS（Domain Name System）



DNS はコンピュータの名前を内部表現に変換するサービスです。通常はその存在を意識することはありませんが、現在のインターネットで最も重要なサービスです。

インターネット上に存在するすべてのコンピュータは、論理的なネットワークの単位である「ドメイン」に分割されます。

TCP/IP による通信では、各コンピュータを識別するために「IP アドレス」という 32 ビットの数値を用いています。一方その数値では人間が理解しづらいので、「ホスト名」という名前をつけています。DNS はドメイン上でその IP アドレスとホスト名の組み合わせ登録・管理するためのしくみを提供するものです。 特に電子メールと Web アプリケーションにはこの DNS は不可欠なサービスです。

www.yahoo.co.jp ⟷ 203.216.227.176

ホスト名　　IP アドレス

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

6 - 2 ネットワークの安全性

ネットワークを使う上で注意しなければならないのは「セキュリティ（安全性）」です。以下のようなリスク（危険性）に注意してください。なお、これらのリスクを減らす技術も実用化されていますが、すべてのネットワークに対して対策が施されているわけではありません。

盗聴・漏洩

ネットワーク上に流れるデータのほとんどは暗号化されていません。誰かが何らかの方法でネットワーク情報を盗み、内容が知られてしまう可能性があります。盗聴を防ぐためには「暗号化」を行います。

改ざん・破壊

ネットワーク上に流れるデータの一部を改ざんしたり破壊したりすることで業務を混乱させ、業務を継続できなくすることが可能です。これを防ぐためには「電子署名」と呼ばれる技術を使います。

なりすまし

別のコンピュータや利用者を装い、不正にネットワークを利用することが可能です。なりすましを防ぐためには「認証」という技術を使います。

事後否認

ネットワーク経由で契約や商品購入をした後で、「そんなことはやってない」と事実を否定することが可能です。事後否認を防ぐためには「電子署名」という技術を使います。



盗聴・漏洩



改ざん・破壊



なりすまし



事後否認

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

■ 暗号化

インターネットをはじめとするネットワーク上では、さまざまなデータが行き交い、不特定多数のサーバを経由してデータが送受信されています。

送信者から受信者へ安全にデータを届けるためには、送信者がデータの送信時に暗号化を行い、その暗号文を送るようにします。受信者は受け取ったデータを元に戻し、内容を読み取ります。暗号化されたデータを元に戻すことを復号といいます。データが暗号化されていれば、途中の経路でそのデータが盗聴されたとしても内容が容易に推測できないため、安全性や信頼性が高まります。

暗号化されていないデータを平文（ひらぶん）といいます。データの暗号化とは、平文（ひらぶん）を「暗号化アルゴリズム」と「暗号化鍵」を使って、容易に解読できない暗号文に変換することです。

暗号化方式は大きく以下の 2 つに分けられます。

共通鍵暗号方式（秘密鍵暗号方式）

暗号化と復号に共通の鍵を使用する暗号方式です。送信者は「共通の秘密鍵」で暗号化し、受信者は「共通の秘密鍵」でデータを復号します。

この方式の利点は、暗号化と復号に使う鍵が 1 つで済むこと、暗号化と復号の処理時間がかからないことです。

欠点としては、この秘密鍵を持っていれば誰でも暗号化・復号を行えるため、秘密鍵を第三者に知られないよう管理しなければならないことです。

公開鍵暗号方式

暗号化と復号にそれぞれ別々の鍵を使う暗号化方式です。データを受け取りたい受信者はあらかじめ秘密鍵と公開鍵のペアを作り、公開鍵を送信者に渡しておきます。送信者は「受信者の公開鍵」を使って暗号化し、受信者は自分しか知らない「受信者の秘密鍵」で復号します。

この方式では、公開鍵と秘密鍵をペアで生成します。公開鍵で暗号化した暗号文はペアである秘密鍵でしか復号できません。秘密鍵が漏洩しない限り、公開鍵は文字通り不特定多数に公開しても安心です。

欠点としては、公開鍵暗号方式は暗号化、復号の処理に時間がかかることです。

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

■ 電子署名

電子署名は公開鍵暗号方式を応用した「デジタル署名」というものを行って、データの改ざんを防ぐ方法です。

デジタル署名は「送信者の秘密鍵（署名鍵）」で署名を生成し、データに添付して送信します。受信者は、ペアとなる「送信者の公開鍵（検証鍵）」で署名の検証を行います。

署名の生成に使う秘密鍵は本人しか知らないので、その対である「送信者の公開鍵」で検証できたということは、送信者本人が「送信者の秘密鍵」で署名を生成したということに他ならないのです。

電子署名は、あくまで途中で改ざんされていないことを保証するだけです。データを暗号化する訳ではないので、盗聴されていないことを保証するものではありません。

■ 認証

コンピュータは、通常、ユーザ名とパスワードを使って本人の確認をします。パスワードは自分だけが知っている文字列です。どのようなセキュリティ対策をしても、ユーザ名とパスワードが漏れてしまえば、なりすましを防ぐ方法はほとんどありません。そのため、パスワードの管理には十分な注意が必要です。

パスワードは、覚えやすく、他人から類推されにくいものが望ましいとされています。また、紙に書いて保存する場合は、鍵のかかる安全な場所にしまうようにします。付箋紙に書いて PC の近くに貼ったり、机の上に放置したりしてはいけません。

パスワードに関する一般的な注意を紹介しておきましょう。

- 可能な限り長いパスワード（通常 8 文字以上が良いとされている）を使う
- 英字、数字、記号を混在させる。
- 単純な英単語を使わない。
- 家族、ペット、趣味などに関連する語は使わない。
- 複数のコンピュータ、あるいは複数のサイトで同じパスワードを使わない。
- 他人にパスワードを教えない。

また、他人がパスワードを入力しているときは手元を見ないことはエチケットのひとつです。最近では、指紋認証や静脈認証、顔認証など、その人固有の身体情報を使った認証（バイオメトリクス認証）も実用化されています。ただし誤認識の問題など、いくつかの課題は完全には解決していません。

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

索　引

1

10 進数 11
16 進数 14

2

2 の補数 16
2 進数 10, 12
2 層クライアント／サーバ・モデル 95

3

3 Tier Client Server Model 96
3D プリンタ 34
3 層クライアント／サーバ・モデル 96

A

Active / Active 110
Active / Standby 110
AI 3, 6
Ajax 100
ALU 35
AND 22
Android OS 67
ANSI 26
Apache 81
APFS 63
Artificial Intelligence 3, 6
ASCII コード 26
ASP 98
ASP.NET 81

B

B2B 91
B2C 90
Basic 78
binary 10
binary digit 10
Bluetooth 48
Blu-ray 42
Blu-ray Disc 42
BSD 66
BtoB 91
BtoC 90
Byte 10

C

C 78
C# 79
C++ 78
CD 42
Central Processing Unit 35
CFexpress 41
Class 48
COBOL 78
copyright 80
CPU 30, 35
CUI 65

D

DAT 44
DATA 3
decode 36
DevOps 73
digit 10
Digital Transformation 103
Digitalization 103
Digitization 103
DIKW モデル 3
DLT 44
DNS 118
DRAM 39
DVD 42
DX 101, 103
Dynamic RAM 39

E

EEPROM 39
Electrically EPROM 39
ERP 94
EUC-JP 28
EUC コード 28
execute 36
exFAT 63
ext2 63
ext3 63
ext4 63

F

FAT 63
FAT32 63
fetch 36
FreeBSD 66
FSF 80

G

GPL 80
GPU 45
GUI 65

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

H

Hard Disk Drive 40
HDD 40
HFP 48
HFS 63
HFS＋ 63
HID 48
HSP 48
HTTP 100
HTTPS 100
Hyper-V 85

I

IaaS 98, 99
IDE 77
IIS 81
InfiniBand 46
Information 3
Information Technology 2
Internet of Things 5
iOS 67
IoT 5
iPadOS 67
IS 94
ISO 26
ISO-2022-JP 27
ISP 90
IT 2, 3

J

Java 78
JIS X 0201 27
JIS X 0213 27
JIS コード 27
JIS 漢字コード 27

K

Knowledge 3

L

Linux 66
LTO 44

M

M.2 46
Mac OS 拡張フォーマット 63
Mac OS 標準フォーマット 63
Microsoft Windows 66
Mini Serial ATA 46
mSATA 46
MTBF 106
MTTR 106
Multiple Virtual Storage 59
MVS 59
MySQL 81

N

Near Field Communication 48
NFC 48
NOT 23
NTFS 63

O

OHA 67
OLAP 95
OLTP（Online Transaction Processing） 65
OR 23
ORACLE 81
OS 52
OSS（Open Source Software） 80

P

PaaS 98, 99
Parallels Desktop 84
PCI Express 46
Perl 79
PHP 79
PostgreSQL 81
Programmable ROM 39
PROM 39
Python 79

R

RAID 109
RAID 0 109
RAID 1 109
RAID 5 109
RAM 39
Random Access Memory 39
RASIS 106
Read Only Memory 39
ROM 39
Ruby 79

S

SaaS 98
SAS 46
SATA 46
SD カード 39, 41
Serial ATA 46
Serial Attached SCSI 46
Solid State 41
Solid State Disk 41
SQL Server 81
SRAM 39
SSD 41
Static RAM 39

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

T

Tomcat 81
TPU 45

U

Unicode 28
Unicode テキスト 28
Universal Serial Bus 47
UNIX 66
USB 47
USB メモリなど 39
UTF 28
UTF-16 28
UTF-8 28

V

Virtual 58
Virtual PC 84
VirtualBox 84
Visual Basic 79
VMware 84
vSphere 85

W

Watch OS 67
Web3 階層クライアント／サーバ・モデル 96
Web アプリケーション 98
Web サーバ 81
Web サービス 91
Wisdom 3
Word 10
WORM 44
write back 36

X

XaaS 99
XenServer 85
XML 91
XML Web サービス 91
XMLHttpRequest 100
XOR 24

あ

アーカイブ 112
アジャイル型 72
アセンブリ言語 74
アドレス 35, 38
アドレス部 35
アドレスレジスタ 35
アプリケーション・サービス・プロバイダ 98
アプリケーションサーバ 81
アプリケーション層 96
アプリケーションソフトウエア 82
アルゴリズム 76
安全性 107

い

イメージスキャナ 31
インクジェット・プリンタ 34
インターネット サービス プロバイダ 90
インタプリタ 75

う

ウォーターフォール型 71
運用 73

え

液晶ディスプレイ 33
エクサ 21
エクスプローラ 55
エムドットツー 46
エンコード 25
演算 30
演算回路 35

お

応用ソフトウエア 82
オーバーレイ 57
オープンソース 80
オクタコア 37
オフラインバッチ処理 64
オペレーティング・システム 52
重み 11
オンデマンド 5
オンプレミス 5
オンラインバッチ処理 64
オンラインリアルタイム処理 65, 95

か

カーネルモード 60
解読器 35
開発 73
開発環境 68
開発プロセス 71
外部設計 70
外部割り込み 32
仮数部 17
仮想 58
仮想化 83
仮想化ソフトウエア 83
仮想記憶 58
仮想記憶空間 59
仮想マシン 83
仮想メモリ 58
仮想メモリ空間 59
価値 3, 4, 105
稼働率 106
カメラ 31
可用性 106

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

カレントディレクトリ 62

き

キーボード 31
記憶 30
記憶媒体 40
ギガ 21
機械語 74
基幹業務 92
基幹系システム 92
基数 11
基数変換 12
揮発性 39
基本サイクル 36
基本ソフトウエア 51
キャッシュメモリ 37
キロ 21

く

クアッドコア 37
クァンタム 54
偶発故障 107
クライアント 95
クラウド・コンピューティング 98
クラスタ 110
繰り返し 76
クロック 38
クロック周波数 36

け

経営資源利用計画 94
ゲスト OS 83
月次バッチ 64

こ

高級言語 74
構造化プログラミング 76
コーディング 70
コード 25
コードレビュー 76
コピーライト 80
コピーレフト 80
コンテナ 90
コンテンツ 90
コンテンツビジネス 90
コンパイラ 74, 75
コンパイル 75
コンピュータの 5 大要素 30

さ

サーバ 95
サーマルプリンタ 34
差分バックアップ 112
算術演算 22

し

磁気テープドライブ 44
資源 61
資源管理 61
資源割り当て 61
指数部 17
システム・アナリスト 69
システムエンジニア 70
実行 36, 37
実装 70
シフト JIS コード 27
主記憶装置 30
出力 30
出力装置 30, 33
順次 76
情報系システム 93
初期故障 107
ジョブ 53
シングルタスク 53
信頼性 106

す

数値演算 22
スーパースカラー 37
スーパーバイザーモード 60
スクリプト 75
ストライピング 109
ストレージ 40
スパイラル型 72
スプーラ 34
スプール 34
スラッシング 60
スループット 105
スレッド 55, 56

せ

制御 30
設計 70
絶対パス 62

そ

相対パス 62
増分バックアップ 112
ソースコード 74
ソースプログラム 74
ソフトウエア 50
ソフトウエアライセンス 80
ソフトコピー 33

た

ターンアラウンドタイム 105
第 3 のプラットフォーム 101
対話型 OS 65
多重仮想記憶 59
タスク 53

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

タスク管理 53
タッチスクリーン 31
タッチパネル 31

ち

中央処理装置 30, 35

て

ディスプレイ 33
データ 3
データ構造 76
データ層 96
データベースサーバ 81
デコード 36, 37
デジタイゼーション 103
デジタライゼーション 103
デジタルトランスフォーメーション 101, 103
テスト 70
デバイスドライバ 60
デバッグ 70
デマンドページング方式 58
デュアルコア 37
テラ 21
伝送路 46

と

統合開発環境 77
ドット・インパクト・プリンタ 34
ドライブ 40
トランザクション 65
トリプルコア 37

な

内部設計 70
内部割り込み 32
ナノ 21

に

日次バッチ 64
入出力バス 47
入力 30
入力装置 30, 31

ね

ネットワーク・コンピューティング 98

は

ハードウエア 50
ハードウエアコンテキスト 56
ハードコピー 33
ハードディスク 40
ハードディスクドライブ 40
排他制御 61
排他的論理和 24
バイト 10
ハイパーバイザ型 85
パイプライン 36
バグ 70
バス 46
バスタブ曲線 107
バックアップ 111
パッケージソフト 69
バッチ処理 64
バッチ型 OS 64
バッファ 32
パリティつきストライピング 109
バンク 38
番地 38
半導体メモリ 39
反復型 72
汎用レジスタ 35

ひ

光ディスク 42
ピコ 21
ビット 10
否定 23

ふ

ファイル・システム 62
ファイル管理 62
フェッチ 36, 37
フォールト・アボイダンス 108
フォールト・トレランス 108
不揮発性 39
符号 25
符号部 17
物理メモリ 58
浮動小数点数 17
フラッシュメモリ 39, 41
プリエンプティブ 54
プリフェッチ機能 38
プリンタ 33
フル・バックアップ 111
プレゼンテーション層 96
プログラマ 70
プログラミング 70
プログラミング言語 74
プログラム 50
プログラムカウンタ 35
プログラム設計 70
プロジェクタ 33
プロセス 55, 56
プロファイル 48
プロプライエタリ 80
プロプライエタリ・ソフトウエア 80
分岐 76
分散処理システム 95

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

へ

平均故障間隔 106
ページアウト 59
ページイン 59
ページファイル 59
ページングテーブル 58
ページングファイル 59
ページング方式 58
ペタ 21

ほ

ポーリング 34
保守性 106
保証 4, 105
補助記憶装置 30, 40
補助単位 21
ホスト OS 84
ホスト OS 型 84
保全性 107
ホットサイト 110

ま

マイク 31
マイクロ 21
マイクロカーネル型 85
マウス 31
マシンコード 74
摩耗故障 107
マルチコア CPU 37
マルチタスク 53
丸め誤差 19

み

ミドルウエア 81
ミラーリング 109
ミリ 21

む

無線 LAN アダプタ 45
無線 LAN コントローラ 45

め

命令部 35
命令レジスタ 35
メインメモリ 38
メガ 21
メディア 40
メモリ 30
メモリ・コンパクション 57
メモリ・フラグメンテーション 57
メモリインターリーブ 38
メモリ管理 57

も

文字コード 25
文字セット 25
文字データ 25
モジュール 77
モジュール化 77
モノのインターネット 5
モノリシック型 85

や

焼く 42

ゆ

有機 EL ディスプレイ 33
ユーザーモード 60
有線 LAN コントローラ 45
優先度 54
ユーティリティ 68
有用性 4, 105

よ

要件定義 69

ら

ライセンス 80
ライトバック 36, 37
ライブマイグレーション 86
ラウンドロビン 54
ランダムアクセス 39

り

リアルタイム型 OS 65

る

ルートディレクトリ 62

れ

レーザ・プリンタ 34
レジスタ 35
レスポンスタイム 105

ろ

論理演算 22
論理積 22
論理メモリ 58
論理和 23

わ

ワード 10
割り込み 32

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

（空白ページ）

サンプル

Copyright 2009-2026 Technofive Co., Ltd. All Rights Reserved.

サンプル

IT 基礎

TS011-15

初版:2009 年 10 月 31 日
改訂:2026 年 02 月 04 日

著者:テクノファイブ株式会社
〒659-0054
兵庫県芦屋市浜芦屋町 9-14-205
TEL 0797-55-5577 FAX 0797-23-4670
URL http://www.techno5.co.jp