

目次

1 章 コンピュータ開発の歴史	3
1.1 コンピュータ開発の歴史	4
1.2 商用コンピュータの歴史	6
1.3 マイクロプロセッサの歴史	8
1.4 パソコンの歴史	10
1.5 ネットワークの歴史	12
1.6 家庭用ゲーム機の歴史	14
第 2 章 コンピュータの基本	17
2.1 コンピュータの仕事の流れ	18
2.2 ハードウェアとソフトウェア	20
2.3 パソコンの外観と内部構造	22
2.4 コンピュータの基本構成	24
2.5 中央処理装置と記憶装置	26
2.6 入力装置と出力装置	28
第 3 章 制御と演算	31
3.1 命令語とデータ語	32
3.2 命令の種類	34
3.3 命令セットアーキテクチャの分類	36
3.4 2 進数と 10 進数	38
3.5 データ形式 (1)	40
3.6 データ形式 (2)	42
第 4 章 2 進演算と論理演算	45
4.1 2 進演算 (1)	46
4.2 2 進演算 (2)	48
4.3 ブール代数とブール関数	50
4.4 論理素子と回路	52
4.5 論理設計 (1)	54
4.6 論理設計 (2)	56
第 5 章 パソコンのハードウェア (1)	59
5.1 CPU (1)	60

5.2 CPU (2)	62	第 10 章 情報処理システム	129
5.3 主メモリ	64	10.1 システムの信頼性	130
5.4 キャッシュメモリ	66	10.2 システムの構成形態	132
5.5 仮想メモリ	68	10.3 システムの性能評価	134
5.6 バス / 入出力インタフェース	70	10.4 システムの高信頼化	136
 第 6 章 パソコンのハードウェア (2)	73	10.5 システムの処理形態 (1)	138
6.1 HDD	74	10.6 システムの処理形態 (2)	140
6.2 光ディスク	76	 第 11 章 情報セキュリティの基礎	143
6.3 フラッシュメモリ	78	11.1 暗号の歴史とアルゴリズムの分類	144
6.4 SSD	80	11.2 秘密鍵暗号系と公開鍵暗号系	146
6.5 USB	82	11.3 秘密鍵暗号	148
6.6 NAS	84	11.4 公開鍵暗号	150
 第 7 章 オペレーティングシステム	87	11.5 デジタル署名と認証	152
7.1 OS の歴史	88	11.6 公開鍵証明書と認証局	154
7.2 OS の役割と機能	90	 第 12 章 インターネットのセキュリティ対策	157
7.3 パソコンの OS	92	12.1 ネットワークにおける不正	158
7.4 プロセス管理	94	12.2 コンピュータウイルス	160
7.5 メモリ管理 (1)	96	12.3 ネットワークセキュリティ (1)	162
7.6 メモリ管理 (2)	98	12.4 ネットワークセキュリティ (2)	164
 第 8 章 ソフトウェアとプログラミング	101	12.5 SSL	166
8.1 ソフトウェアの位置づけ	102	12.6 電子政府	168
8.2 アプリケーションプログラム	104	 第 13 章 マルチメディア端末としてのコンピュータ	171
8.3 アルゴリズム	106	13.1 アナログとデジタル	172
8.4 プログラム	108	13.2 マルチメディア圧縮技術	174
8.5 プログラミング言語	110	13.3 音楽配信の仕組み	176
8.6 インタプリタとコンパイラ	112	13.4 動画配信・共有の仕組み	178
 第 9 章 ネットワーク技術とインターネット	115	13.5 動画再生の仕組み	180
9.1 情報通信技術	116	13.6 テレビとパソコン	182
9.2 OSI (基本) 参照モデル	118	 第 14 章 モバイル技術	185
9.3 LAN と WAN	120	14.1 スマートフォン	186
9.4 インターネット	122	14.2 タブレット端末	188
9.5 TCP/IP	124	14.3 IC/RFID	190
9.6 ルータとサーバの機能	126	14.4 NFC	192

14.5 電子財布	194
14.6 GPS	196
第 15 章 コンピュータとネットワークの最新技術	199
15.1 クラウドコンピューティング	200
15.2 クライアント仮想化技術	202
15.3 SNS	204
15.4 HTML5	206
15.5 電子書籍	208
15.6 スーパーコンピュータ京	210

第 1 章 コンピュータ開発の歴史

コンピュータはわずか 70 年足らずで素晴らしい発展を遂げました。勿論これで成長が終わったわけではなく、これからも形を変え益々進展し普及していくでしょう。本章では、どのような経緯をたどって今日のコンピュータが出来上がったかを見ていきます。

- 1.1 コンピュータ開発の歴史
- 1.2 商用コンピュータの歴史
- 1.3 マイクロプロセッサの歴史
- 1.4 パソコンの歴史
- 1.5 ネットワークの歴史
- 1.6 家庭用ゲーム機の歴史

1.1 コンピュータ開発の歴史

開発期のコンピュータ

コンピュータの歴史における開発期とは、研究成果を工学的・技術的興味から将来の要求を満たすような実際の機械を作り評価するために、お金に糸目を付けず財力に任せ投資した時期です。

B. Pascal が最初の実用機を完成させました。これは 10 進で桁上げの機構を歯車によって行い、加算と減算のみの機能でした。この種の計算機は W. Leibniz によって改良され、乗算もできるようになりました。

C. Babbage は計算順序およびデータをカードに穴をあけて機械に入力する完全に自動化した計算機を考えました。この着想が現在のコンピュータの基礎になっています。

1889 年に H. Hollerith が国勢調査の統計処理を行うための統計機として、パンチカードシステムを開発実用化しました。

1940 年代に入り、米国では大砲の弾道計算などに大型の自動計算機が必要でした。MIT の V. Bush は N. Wiener の助けを借りて自動計算機の研究を行っていました。その後、1942 年にはベル電

話研究所において全リレー式の計算機のモデルが開発され、また 1944 年にはハーバード大学の H. H. Aiken が IBM 社と共同でリレー式の大型自動計算機を完成させました。これはリレー式による Babbage の機械の実証と言えます。さらに、1948 年には IBM 研究所においてマーク II が、1950 年にベル研究所においてマーク III が、また 1952 年にはハーバード大学においてマーク IV が完成しました。

真空管式計算機

他方、ペンシルバニア大学の J. P. Eckert と J. W. Mauchly がパルス技術を用いて電子管式の計算機 ENIAC (Electronic Numerical Integrator And Computer) の製作を 1943 年に開始し、1945 年に完成させています。これが世界最初の電子計算機で、18,800 本もの真空管を用い、消費電力約 120 キロワット、重量約 30 トン、24 馬力の冷却装置を必要としていました。また、これは 10 進で計算を行う方式であり、今日のコンピュータが数秒で実行する計算を 20 時間も必要としたそうです。Eckert は主として回路設計を、Mauchly は主



ENIAC と EDSAC に注目して下さい。両者はともに電子化（真空管式）されており電子計算機と呼んで良いでしょう。しかし、前者のプログラムは外部プログラム方式と言って配線によっていたのです。これに対し、後者は内部プログラム方式と言って、プログラムを高速の主メモリに蓄え、この内容を逐次読み取って高速に計算を進めていきます。この点が大いに異なっています。プログラムやソフトウェアについては後に述べますが、前者が現在の電卓のようなイメージ、後者が現在のコンピュータと言えば両者の違いは分かるでしょう。

表 1.1 コンピュータ開発の歴史

年	研究者	試作開発機
1642	B. Pascal(仏)	機械(歯車)式加減機の発明
1670	W. Leibniz(独)	機械式四則演算機の構想
1832	C. Babbage(英)	自動逐次式計算機Analytical Engineの開発
1889	H. Hollerith(米)	パンチカードシステムの開発
1930	G. R. Stibitz(独)	リレー式計算機の開発
1944	H. H. Aiken(米)	自動逐次式リレー計算機Harvard Mark Iの開発
1945	J. P. Eckert(米) J. W. Mauchly(米)	最初の真空管式計算機ENIACの開発 (ペンシルベニア大学)
1945	J. von Neumann (ハンガリー)	プログラム内蔵方式の提案
1949	M. V. Wilkes(英)	最初のプログラム内蔵型真空管式計算機EDSACの完成 (ケンブリッジ大学)

としてアルゴリズムの開発を行ったと言われています。

プログラム内蔵方式

さて、コンピュータに実行させたい具体的な仕事の内容とその手順を機械が理解できる言葉で指示したものをプログラムと言います。Neumann により

プログラミング技術の基礎が確立され、1949 年英国における EDSAC (Electronic Delay Storage Automatic Calculator)、1951 年 2 進法を採用した米国の EDVAC (Electronic Discrete Variable Automatic Computer)、さらに 1952 年 Neumann による JOHNNIAC などが開発されました。

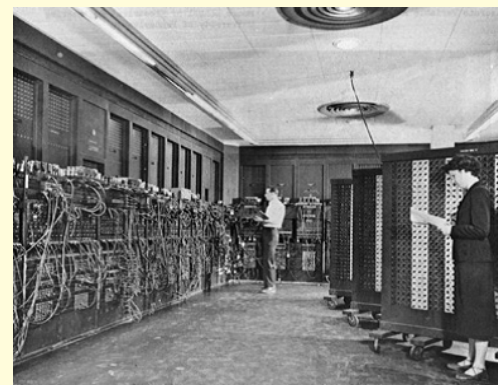


写真 1.1 ENIAC
出典：フリー百科事典『ウィキペディア (Wikipedia)』



写真 1.2 EDSAC
出典：フリー百科事典『ウィキペディア (Wikipedia)』
Computer Laboratory,
University of Cambridge

1.2 商用コンピュータの歴史

商用コンピュータの発展

コンピュータの歴史における商用期とは、開発期に得られた成果を基に、コストを考えた実用化のための試練を経て市場に投入されたもので、当然製造販売して利潤が得られなければなりません。

商用コンピュータの発展の歴史を表1.2にまとめました。第1世代は、真空管、

第2世代ではトランジスタ、さらに第3世代では個々の半導体素子などを組み合わせた集積回路(IC)が開発・実用化されました。IC実装技術の向上に支えられて、第3.5世代にはさらに大規模集積回路(LSI)、第4世代には超大規模集積回路(VLSI)へと次々進歩して現在に至ります。この頃の特筆すべきことは、後述の通りマイクロプロセッサの出現で

表 1.2 商用コンピュータの発展

	世代	使用素子	素子の動作速度	性能比	言語	主な特徴
1950	第1世代	真空管	10^{-3} 秒		機械語	蓄積プログラム方式など基礎技術の確立期
1955						
1960	第2世代	トランジスタ	10^{-6} 秒	1	アセンブラ	本格的実用化期、オペレーティングシステム(OS)の出現、バッチ、リモートバッチ方式
1965	第3世代	IC (集積回路)	10^{-9} 秒	10^3	高水準言語	オンライン実用化期、TSS、リアルタイムシステム
1970	第3.5世代	LSI (大規模集積回路)	10^{-10} 秒	10^4		分散処理システム実用化期、マルチプログラミング、仮想メモリ方式
1975	第4世代	VLSI (超大規模集積回路)		10^5	構造化言語	コンピュータネットワーク、データベース実用化期、人工知能の研究開発期
1980						
1985						
1990	第5世代	ULSI (超々大規模集積回路)	10^{-10} 秒以下	10^6 以上	オブジェクト指向言語	柔軟な情報処理、クライアントサーバシステム、インターネットとIT(情報通信技術)
1995						
2000						
2005						

す。

現在のコンピュータ

現在のコンピュータは第5世代に分類されますが、第1～4世代が主として使用素子の発展によって世代をめぐらえてきた程明確な区分はありません。また現在ではこのような区分はほとんど議論されません。コンピュータの商用期の歴史として知っておけばよいでしょう。第4世代以降はハードウェアではなく、むしろ利用形態・応用分野などがコンピュータの姿を変えてきたと言う方が正しいです。図1.1に情報基盤(インフラ、プラットフォーム)の大きな流れを示します。

1950年代から始まったメインフレームは集中処理を得意としました。この技術は処理能力・コストの分散と高性能化などを目指してオフィスコンピュータ・ミニコンピュータを生みました。一方当初、電卓用に開発されたマイクロ

プロセッサの出現は組み込みシステムとして多くの機器に利用されると同時に、マイコン・パソコン・ワークステーションなどのコンピュータの分野にダウンサイジングや低コスト化をもたらした大変大きなインパクトを与えました。その結果1990年代、クライアントサーバを中心とした分散処理が普及しました。

現在、ネットワーク上に存在するサーバが提供するサービスを、ユーザが低コストで利用するというクラウドコンピューティングの考え方に向かっています。これはコンピュータとネットワークが、低コストになって初めて可能なネットワークコンピューティングの一つの形態です。自社システムを作るより、他社が提供するサービスを利用した方がコストを下げられる場合が多いのです。このような処理と通信の統合こそITの真髄です。これにモバイルが加わり、大きく変わろうとしています。

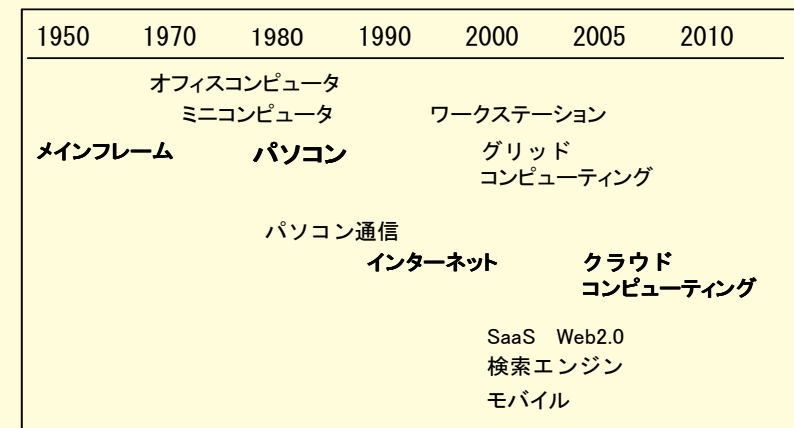


図 1.1 情報基盤の変遷

1.3 マイクロプロセッサの歴史

卓上計算機の歴史

表 1.3 に示した通り、機械式計算機(歯車式)が手動式から電動式へと進歩し、さらにトランジスタ化されました。

1964 年に電子式卓上計算機(電卓)がシャープ(株)から発表され、電卓のブームが始まります。その後、1967 年には IC 化された電卓が作られました。しかし、低コスト化のために IC チップ数を少なくする必要から機種ごとの IC を開発することになり、仕様の変更が大きな負担になりました。

この問題を解決するために、日本の電卓メーカー ビジコン社が設立間もないインテル社に LSI の設計を依頼しました。日米共同で新構想の電卓用 LSI の設計を行い電卓にマイクロプログラム制御方式(後述)をとり入れたのです。

次に、メモリ部分を演算部分とは別の LSI とし蓄積プログラム方式を導入しています。この電卓は読み出し専用メモリに、機種ごとに異なるプログラムを実装

していました。この新構想の電卓用 LSI は小さいけれども汎用性のある、後にマイクロプロセッサと呼ばれるワンチップ CPU です(これは、日本人 嶋正利らの手で開発されたものです)

インテルのマイクロプロセッサ

その後、1971 年にインテル社が自由度のある 4 ビット用チップ 4004 型(これが世界最初のマイクロプロセッサです。)を(写真 1.3 参照)、続いて 1972 年に 8 ビット用チップ 8008 型(これも日本のセイコー社が時計用に開発依頼したものです。)を発表し、マイクロコンピュータの時代が始まりました。

すなわち、電卓から出発したマイクロプロセッサを演算素子・メモリ素子などの別々の LSI を組み合わせ、入出力機器を加えてコンピュータを構成するので

ムーアの法則

1965 年インテル社創始者の一人

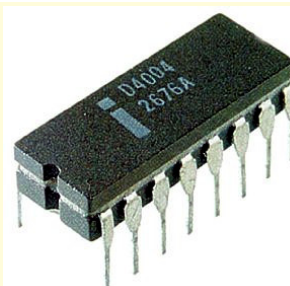
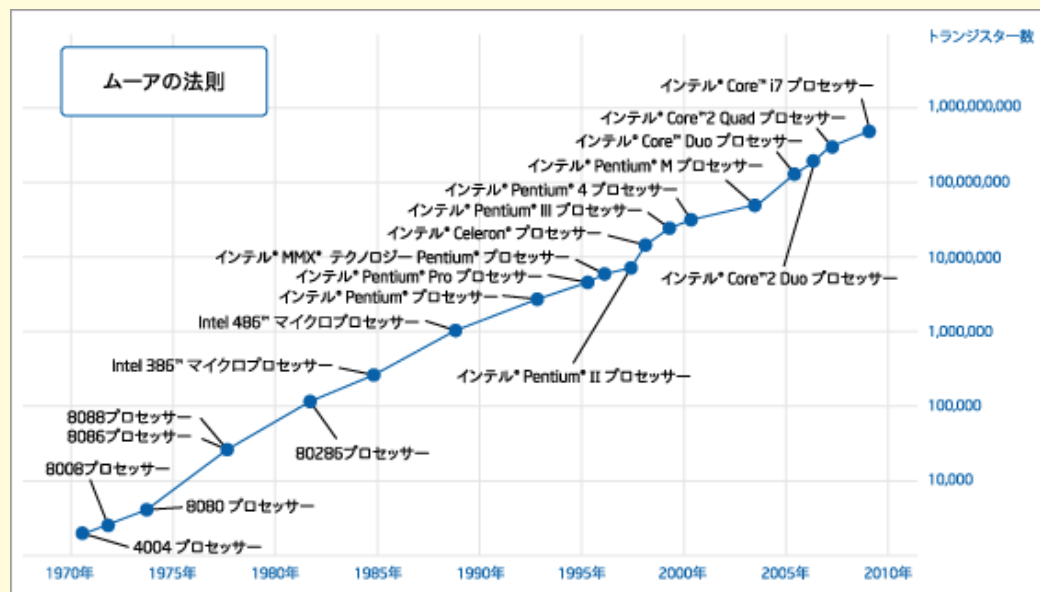


写真 1.3 インテル社 4004 型マイクロプロセッサの外観

表 1.3 機械式・電気式卓上計算機の歴史

年	機種
1820年	アリスモータ計算機(手動式)
1923年	タイガー計算機(手動式)
1930年代	モンロー計算機 (電動式・キーボード式)
1964年	シャープ電卓(トランジスタ式)
1966年	シャープ電卓(IC化)



インテルミュージアム マイクロプロセッサを支える最新テクノロジーの世界
<http://www.intel.co.jp/jp/intel/museum/processor/index.htm> (2012.8.24)

図 1.2 マイクロプロセッサとムーアの法則

Gordon Moore の提唱によるムーアの法則「集積回路におけるトランジスタの集積度は 2 倍/2 年(4 倍/3 年とする場合もある。)である」がほぼ成り立つことを確認することが出来ます(図 1.2 参照)。

ただし、この経験則は半導体の微細加工技術の発展を根拠としており、最近この傾向は鈍化しています。それでも「集積度」を「性能」と読み替えれば、依然として成り立つとされています。

LSI の量産技術の進歩によりマイクロプロセッサはますます低コストになり、きわめて広く使われるようになります。マイクロプロセッサは自動車・カメラ・家庭電気製品のような民生品にまでたくさん用いられています。一方、マイクロ

コンピュータも次第に高性能化され、パソコン(PC)・ワークステーション(WS)などとして爆発的に普及しています。



ひらさわ教授のコラム 1

1970 年代当初、私は企業でテレメータ・テレコントロールシステム用機器の開発に携わっていました。従来はシステムをミニコンで実現していましたが、これを複数のマイクロプロセッサによる分散処理をし、拡張性・信頼性・機能性・経済性などを図りました。単純な機能の集まりなので、ハードウェア・ソフトウェアともモジュール化しており、まさに今日で言う組み込みシステムです。最初は 8008 型を導入しましたが開発が間に合わず、8080 型から実用機として活躍しました。応用プログラムの開発は、汎用大型機による PL/I クロスコンパイラ(貧弱な能力しかないコンピュータ(マイコン)でコンパイルするのではなく、強力な能力をもつコンピュータ(大型機など)でコンパイルしてもらう方法。エミュレータ。)を用いており、大規模なシステムのコンパイルには苦労しました。懐かしい思い出です。

次のコラムへ

1.4 パソコンの歴史

マイクロプロセッサをCPUとして用いたコンピュータ(当初はマイクロコンピュータと呼んでいました。海外ではホームコンピュータやホビーパーソナルコンピュータと言う言葉もあります。)であるパソコンの歴史を振り返ります。表1.4に特筆すべき出来事を示します。表1.3を参照しながら見て下さい。

マイクロコンピュータセット「MCS-4」

1971年のインテル社4004型の開発はコンピュータの心臓部をワンチップ(LSI)化したもので、私たちが個人的にコンピュータを持てると言うマイクロコンピュータ(マイコン)の出現を期待させるものです。インテル社は同時に、ROM(4001型)、RAM(4002型)、シフトレジスタ(4003型)など周辺LSIを組み合わせてMCS-4として売り出しました。

キット型マイコン「アルティア 8800」

1975年ミッツ社はインテル社8080型CPUを用いた組み立てマイコンキットを発売しました。これが世界初のパソコンと言われています。1974年にデジタルリサーチ社が発表した8080用OS CP/Mを使用しています。このキットはアメリカ国中のパソコンブームに火をつけたと言われています。わずか256[バイト:B]の主メモリでしたが拡張ボード(4K[B])も用意され、また

BASIC (Beginner's All-purpose Symbolic Instruction Code) でプログラムできるという優れたものでした。

完成型マイコン「アップルII」

1977年のアップルコンピュータ社アップルIIはロックウェル社のMCS6502(8ビット型)を使い、キーボード・ディスプレイを持つ完成品でした。主メモリは4K[B]でしたが、BASICもROMで内蔵されていました。カセットテープも接続出来るので、後にゲーム

表 1.4 パソコンの歴史

年	パソコンの開発・製品化
1971	インテル社4004型を用いたマイクロコンピュータセット:MCS-4発売
1975	インテル社8080型を用いたキット型マイコン:ミッツ社 アルティア 8800 発売
1976	組み立てキット型マイコン(トレーニング用):NEC社TK-80発売
1977	完成型マイコン:アップルコンピュータ社 アップルII 発売 コモドール社 PET 2001 発売 タンディ社 TRS-80 発売 精工舎 SEIKO-5700 発売
1978	シャープ社 MZ-80K 発売
1979	NEC社 PC-8001 発売
1981	IBM社 IBM PC 発売
1982	NEC社 PC-9801 発売
1983	アップル社 マッキントッシュ 発売
1985	IBM社 IBM PC/AT 発売

カセットテープの音を聞く

などのソフトを買って遊ぶ人にも普及していきます。当時フレキシブルディスク装置が開発され、その高信頼性からビジネス用に用いられました。

16ビット型パソコン「IBM PC」

メインフレームで絶対的な地位にあったIBM社が遂にパソコンを開発しました。1981年、MS-DOSと言うOS(CP/Mの機能を完全に含む)を搭載したIBM PC(PCはPersonal Computerの略)です。CPUはインテル社8088型で16ビット型、主メモリは最大64KB、5インチのフレキシブルディスク装置、モノクロまたは4色カラーディスプレイを実装していました。IBM社は引き続き1985年にIBM PC/AT(ATはAdvanced Technologyに略)を発売します。IBM PCと同様オープンアーキテクチャを採用しました。その結果、この機種の様子が事実上のパソコンの業界標準となりました。



ひらさわ教授のコラム2

1977年アップルIIと同時期に発売されたコモドール社の「PET2001」(PETはPersonal Electronic Transactorの略)の後の機種はカセットテープで配布されるBASICを使用するパソコンです。1981年、私が企業から大学に移った際、神戸のソフトウェア開発会社社長の友人が研究室の学生の遊び道具としてPETを寄贈してくれました。学生が神戸から新幹線で運び、研究室に初めてのパソコンとして設置しました。早速学生の一人がBASICを使って野球ゲームを作りました。ボールが投げられるとバットが動き、乱数によりヒットになったり三振したりします。まだゲームらしきものが身近ない時代で、息子などが休日に研究室で遊んでいたものです。



写真 1.4 ミッツ社「アルティア」の外観
http://commons.wikimedia.org/wiki/File:Altair_8800_Computer.jpg



写真 1.5 精工舎社「SEIKO-5700」の外観
<http://www.itoi.co.jp/attic/time12.html>(Time Machine, 2012.9.2)



写真 1.6 コモドール社「PET2001」の外観
<http://www4.airnet.ne.jp/mit/Museum/COMODORE/PET2001.html>(2012.9.2)

1.5 ネットワークの歴史

ネットワークの発展

ネットワークは、まず 1870～80 年代に電話網として整備され始めました。しかし 1970 年代までは、コンピュータ（処理）とネットワーク（通信）は別々でした。これが日本では 1985 年、電気通信市場解放が実施され一気に 1980 年代後半からパソコン通信が普及していきます。

1990 年代にはインターネットが、2000 年代にはテレビネットワークとアプリケーションが手を組み IP 電話・IP テレビ（放送）などが普及していきます。

さらに 2003 年に、次世代ネットワーク (Next Generation Network: NGN) 構想が発表され、電話網とインターネットの

境がなくなっていくます。

図 1.3 にネットワークの発展の流れを示します。図中、ネットワークインフラ系は従来の電話系です。インターネット系は ARPANET による TCP/IP (Transmission Control Protocol/Internet Protocol) の実績により実用化されています。

コンピュータネットワークは大別して、オフィス内・工場内・大学内などの狭域コンピュータネットワーク：LAN (Local Area Network) と広域コンピュータネットワーク：WAN (Wide Area Network) があります。前者はクライアントサーバシステムや研究室内の PC 用ネットワークなど、後者は主として電話網・インターネットなどを指しています。

1950	1960	1970	1980	1990	2000	2010
[ネットワークインフラ系]						
	57 電話網全国展開		79 全国自動化	90 光アクセス導入		
			87 携帯電話	99 ADSL		
			88 ISDN	97 ATM		
[インターネット系]						
		69 ARPANET		83 TCP/IP採用		
				90 インターネット(米)		
			81 N1(日)	93 インターネット(日)		
			84 JUNET(日)			
					03 NGN	

図 1.3 ネットワークの発展

表 1.5 コンピュータネットワークの歴史

年	出来事
1969 年	ARPANET 実験開始（1989 年運用停止）
1974 年	IBM SNA(System Network Architecture) 発表
1977 年	国際標準化機構（International Organization for Standardization: ISO）作業開始
1983 年	開放型システム相互接続（Open System Interconnection: OSI）参照モデル発表
1984 年	JUNET（日本）サービス開始（1994 年運用停止）
1989 年	WWW (World Wide Web) 開発
1990 年	商用インターネットのサービス開始
1993 年	WWW ブラウザ発表

ネットワークを用いる目的は離れている端末を結ぶだけでなく、分散させてリスク管理を行うことです。すなわち、機能分散・負荷分散により性能を向上し、仕事を分散させています。

コンピュータネットワークの歴史

表 1.5 にコンピュータネットワークの歴史を示します。ARPANET から、インターネットに至る出来事をまとめています。

1961 年に米国ユタ州の電話局が爆破されたことが分散型によるネットワーク構造を促進し、リスク分散型の ARPANET につながっていると言われています。そうして、ALPANET は、1969 年米国の実験用のコンピュータネットワークとしてスタートしました。1983 年に軍用部分が切り離され研究教育用として発展し、これが 1990 年代当初から商用インターネットに移行しました。イ

ンターネットは技術的にも歴史的にも ALPANET に負うところが大きいのです。

1989 年にはティム・バーナーズリーにより、CERN 欧州原子核研究機構での情報管理システムとして世界中のコンピュータに蓄積されている文書をリンクして表示する WWW(World Wide Web) システムが提案されました。その後、WWW を閲覧するシステムとして、モザイク、ネッスケープ、MS インターネット・エクスプローラーなどが開発されてきました。この時期が、インターネットの大衆化と商業利用の始まりとなりました。

そして現在、コンピュータネットワークを活用し、ネットワーク上に存在するサーバが提供するサービスを、ユーザが低コストで利用するというクラウドコンピューティングの考え方に向かっています。

1.6 家庭用ゲーム機の歴史

代表的な家庭用ゲーム機の歴史

1983年に発売された任天堂のファミリーコンピュータ以降の日本における代表的な家庭用ゲーム機を表1.6に示します。

本表では、発売年とともに各ゲーム機のCPUのビット数を示しました。しかし、最近のゲーム機は、3DCGにおける座標変換を専門的に行うジオメトリエンジンやレンダリングエンジンなどを複数持ち、分業によってゲームが処理されています。よってCPUの性能だけでゲーム機のパフォーマンスは決まりません。

携帯型ゲーム機

また、携帯型ゲーム機は、列挙するにとどめるが、1990年前後のゲームボーイ、ゲームギア、PCエンジンGT、1990年代中期のゲームボーイポケット、ゲームボーイライト、1990年代後期のゲームボーイカラー、ネオジオポケット、ワンダースワン、2000年代初期のゲームボーイアドバンス、ゲームボーイマイクロ、スワンクリスタル、2000年代中期のニンテンドーDS、DS Lite、DSi、DSi LL、PSP、PSPGo、2010年代のニンテンドー3DS、PS Vitaなどが発売されている。

表 1.6 家庭用ゲーム機の歴史

年代	代表的な家庭用ゲーム機
1980年代前半 ～ 1980年代中盤	- ファミリーコンピュータ、(任天堂・1983年) - スーパーカセットビジョン (エポック社・1984年) - セガ・マークIII(セガ・1985年)
1980年代後半 ～ 1990年代前半	- PCエンジン、(NECホームエレクトロニクス・1987年)(8bit) - メガドライブ、メガCD(セガ・1988年,1991年)(16bit) - スーパーファミコン(任天堂・1990年)(16bit)
1990年代中盤 ～ 1990年代後半	- セガサターン(セガ・1994年)(32bit) - プレイステーション(SCE・1994年)(32bit) - NINTENDO64(任天堂・1996年)(64bit)
1990年代末 ～ 2000年代初頭	- ドリームキャスト(セガ・1998年)(32bit) - プレイステーション2(SCE・2000年)(128bit) - ニンテンドーゲームキューブ(任天堂・2001年)(32bit) - Xbox(マイクロソフト・2001年)(32bit)
2000年代中盤 ～ 2000年代末	- Xbox 360(マイクロソフト・2005年)(128bit) - プレイステーション3(SCE・2006年)(128bit) - Wii(任天堂・2006年)(64bit+128bit)



写真 1.7 ファミリーコンピュータ
(1983年7月 任天堂)



写真 1.8 メガドライブ
(1988年10月 セガ)



写真 1.9 スーパーファミコン
(1990年11月 任天堂)



写真 1.10 セガサターン
(1994年11月 セガ)



写真 1.11 PlayStation
(1994年12月 SCE)



写真 1.12 Nintendo64
(1996年6月 任天堂)



写真 1.13
ドリームキャスト
(1998年11月セガ)



写真 1.14
Playstation2
(2000年3月 SCE)



写真 1.15
ゲームキューブ
(2001年9月 任天堂)



写真 1.16
Xbox
(2002年2月 Microsoft)



写真 1.17
Xbox360
(2005年12月 Microsoft)



写真 1.18
Playstation3
(2006年11月 SCE)



写真 1.19
Wii
(2006年12月 任天堂)

演習問題

1. 次の文の空欄 [a][b] に入る適切な語の組み合わせを, (1)-(4)の中から選びなさい。

IT は [a] と訳されている。しかし, ICT の方が正確で普及しつつある。いずれも [b] を含む技術である。

- (1) [a] 情報技術 [b] インターネット
- (2) [a] 情報通信技術 [b] コミュニケーション
- (3) [a] 情報技術 [b] ネットワーク
- (4) [a] 情報理論 [b] コミュニケーション

2. 次の文章で間違っているものを選びなさい。

- (1) マイクロプロセッサは CPU を LSI 化したものである。
- (2) パソコンはパーソナルコンピュータの略で, 日本で用いられている名称である。
- (3) マイクロプロセッサはワンチップ CPU と呼ばれている。
- (4) パソコンはマイクロプロセッサを用いたマイコンとは全く異なる。

3. 次の文の空欄 [a][b] に入る適切な語の組み合わせを, (1)-(4)の中から選びなさい。

[a] は [b] から 2 年遅れて完成した。いずれもノイマン型のコンピュータである

- (1) [a] ENIAC [b] EDSAC
- (2) [a] ENIAC [b] EDVAC
- (3) [a] EDSAC [b] ENIAC
- (4) [a] EDVAC [b] EDSAC

4. 次の文章で間違っているものを全て選びなさい。

- ☐ (1) ENIAC は 18,800 本の真空管式で大きな電力を必要とした。
- ☐ (2) コンピュータの研究開発に機械式計算機の考えはほとんど役立っていない。
- ☐ (3) ENIAC のプログラムは配線で出来ていた。
- ☐ (4) 今日のコンピュータの大半はノイマン型と言われるプログラム内蔵方式である。

5. 次の空欄 [a][b][c] に入る適切な語の組み合わせを, (1)-(4)の中から選びなさい。

コンピュータの商用化の歴史は, 1950 年当初から 1970 年代位まで使用素子によって [a] から [b] へ, そして [c] へと世代が塗り替えられた。

- (1) [a] 真空管 [b] トランジスタ [c] IC
- (2) [a] 真空管 [b] トランジスタ [c] LSI
- (3) [a] トランジスタ [b] IC [c] VLSI
- (4) [a] IC [b] VLSI [c] ULSI

6. 次の文章で間違っているものを全て選びなさい。

- ☐ (1) コンピュータ商用期の初期には, UNIVAC I や IBM 7090 などのメインフレームが出現した。
- ☐ (2) マイクロプロセッサは組み込みシステムとして利用されているがその数は少なく, ほとんどはパソコンの CPU として活躍している。
- ☐ (3) マイクロプロセッサは商用期の歴史で, 第 3.5 世代に開発されている。
- ☐ (4) 情報基盤の大きな流れはメインフレーム・パソコン・インターネット・ネットワークコンピューティングと続いていると考えられている。

7. 次の空欄 [a][b][c] に入る適切な語の組み合わせを, (1)-(4)の中から選びなさい。

世界で最初のマイクロプロセッサは [a] で [b] 年に開発された。日本からインテル社に [c] 用に開発が依頼されたものである (注: ixxxx はインテル社の型名を表す)。

- (1) [a] i4004 [b] 1971 [c] 電卓
- (2) [a] i8008 [b] 1972 [c] パソコン
- (3) [a] i8080 [b] 1974 [c] パソコン
- (4) [a] Pentium [b] 1993 [c] パソコン

8. 次の文章で間違っているものを全て選びなさい。

- ☐ (1) ムーアの法則で有名なムーアはトランジスタの発明者の一人ショックレーが設立した研究所を退社し, フェアチャイルド社を経てインテル社を設立した。
- ☐ (2) 設立当初からインテル社は CPU を手がけており, i4004 を生む十分な技術レベルがあった。
- ☐ (3) マイクロプロセッサ i4004 は 8×2 のデュアルインラインというパッケージに実装されている。
- ☐ (4) マイクロプロセッサ i4004 は 4 ビット CPU である。

回答を送信



解説



第 2 章 コンピュータの基本

テレビ・電話・電車・車・飛行機・船などとコンピュータは何が違うのでしょうか。コンピュータという機械と, それ以外の機械の違いは, 後者はハードウェアだけの機械であり, 前者はハードウェアとソフトウェア (後述) という両輪で動く機械なのです。機械工学や電気工学の 200 年以上の歴史の中でソフトウェアと言う概念を持った機械はコンピュータを除いて実在しないのです。

- 2.1 コンピュータの仕事の流れ
- 2.2 ハードウェアとソフトウェア
- 2.3 パソコンの外観と内部構造
- 2.4 コンピュータの基本構成
- 2.5 演算・制御装置と記憶装置
- 2.6 入力装置と出力装置

2.1 コンピュータの仕事の流れ

新しいタイプの機械

現在のほとんどの電機製品にはマイクロプロセッサが組み込まれ、プログラムで制御されています（それはマイクロプロセッサの出現で、必要な機能を安価で実現しやすい組み込みシステムを利用したのであって、原理的にマイクロプロセッサを使わない電子回路やリレー回路で実現することは出来るはずです.）。

しかし、電話の本来の機能は遠く離れた2点間で私たちの会話が出来ることであり、車の本来の機能は離れた場所に歩くより早く行くことです。したがって、電話や車にソフトウェアはない、強いて電話や車のソフトウェアは何かと言えば、これらを使う方法や使う技術と言うことになってしまいます。その結果、コンピュータは人類が今までに持っていなかった新しいタイプの機械であると言えます。

また、従来の機械・道具は人間の肉体的・物理的能力を強化・増幅・支援するものです。例えば、自動車は人間の足を、望遠鏡は人間の目を、電話は人間の耳をそれぞれ強化・増幅・支援するものと考えられます。この点からみると、コンピュータは人間の精神的・頭脳的（活動）能力を支援する従来にはない全く新しい機械と言うことが出来ます。

コンピュータとは

もともと、コンピュータの誕生と同時にソフトウェアという新しい概念が生まれたのです。したがって古い英語の辞書にはソフトウェアと言う言葉はありません（最近の辞書には次のように書かれています）。

(1) [コンピュータ] ソフトウェア（プログラムやその作成技術の総称；cf. hardware）。

(2)（機械・設備の）利用法，利用技術；視聴覚教育の教材」．[ジーニアス英和辞典より]）。

なお、前述の本・新聞などは紙という媒体（ハードウェア）に情報（広い意味のソフトウェア）が乗っていると考えることが出来ます。

では、コンピュータとは一体何物で、どう理解すればよいのでしょうか。コンピュータは明らかに情報（データ）を処理する機械あるいは道具です。前述の通りコンピュータはハードウェアとソフトウェアという2つの構成要素から成り、これらは互いに補い合うような役割分担を持っています。

すなわち、極度に共通化・汎用化されたハードウェアとこれを利用目的に合わせ専用化・適合理化させるソフトウェアからできている機械です。

マイクロプロセッサもそうですが、コ

ンピュータのハードウェアそのものは特殊な場合を除き、どんなシステムにも、どんな場面にも利用できる汎用性を持つよう設計されています。勿論パソコンもそうです。特殊な用途（例えば、「みどりの窓口」の端末機、バンキング用端末機など）以外、そのままパソコンが使用出来

ます。これをある目的に適した機能を持たせる役割がソフトウェアなのです。パソコンもソフトウェアがなければただの箱です。ハードウェアだけでは全く何もしてくれません。ソフトウェアの比重が機能的にもコスト的にも極めて高いのです。これも従来の機械・道具とは本質的に異なる新しい構成概念と言えます。

コンピュータの仕事

さて、先ほどコンピュータは情報を処理する機械であると言いました。

コンピュータはソフトウェアをとりかえることにより、給与計算用にも銀行の預金計算用にも、列車の座席予約用にも、あるいは火力発電制御用にも使えます。しかし、このようにコンピュータの仕事は多岐にわたっているようでも共通的にみれば、図1.4のように示すことができます。例えば、給与計算では入力とは個人の本給や作業時間などのデータであり、出力は給与明細書です。火力発電制御では入力は炉の温度や電圧・周波数などのセンサーの計測値であり、出力は炉の

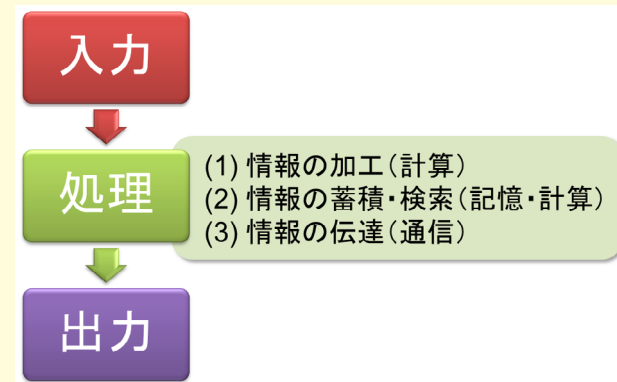


図 1.4 コンピュータの仕事の流れ

バーナのアクチュエータ値や制御盤の指示計の値です。

図1.4で実行される処理（プロセス）は
(1) 情報の加工（計算）
(2) 情報の蓄積・検索（記憶・計算）
(3) 情報の伝達（通信）
です。

ここで、(1) はコンピュータの計算（演算）機能です。例えば、クラス全員の採点結果を入力し平均点を出力するなどの処理を指します。その結果、統計処理によりクラスの成績に関する新しい知識が得られます。

(2) は大量のデータ（データベース）を記憶しその中から必要なものだけを取り出す処理によりデータベースの価値を生みます。

(3) は遠方のデータを必要なときにとり出す処理です。

例えば、WWW（World Wide Web）上の情報にアクセスし、これを見る操作（これには検索機能も含まれている）でコンピュータの重要な処理の一つです。

2.2 ハードウェアとソフトウェア

ハードウェアとソフトウェアの定義

コンピュータはハードウェアとソフトウェアから構成されています。ここまで用語を大変あいまいに使ってきましたが、そろそろはっきり定義しましょう。

ハードウェアは「金物」であり、電子回路・電子部品・機構部品などで構成された装置そのものです。装置の機能・機構を指すこともあります。

ソフトウェアは広義には問題解決に必要な処理手順をすべて含む総合的概念です。狭義にはコンピュータのハードウェア能力を効果的に利用するためのプログラム・データの集合体でありコンピュータと共に生れた人工語で、もともと英語辞書にはありません。

プログラム

ここで、プログラムとは利用者がコンピュータの分かる言葉(プログラム言語、例えばC言語などで書いたもので、最後はマシン語(機械語)に翻訳される。)で記述された問題を解く手順です。

私たちはプログラムを作成する前に解きたい問題に対し、コンピュータが得意とする解き方(算法)を考えます。この問題を解く手順は、自然言語(日本語・英語など)、あるいはその箇条書き(ステップ法など)で記述されたり、フローチャートとよばれる処理の流れ図で表さ

れます。これをアルゴリズム(算法)と言い、プログラム言語に依存したプログラム(アルゴリズムを特定の言語で記述したもの)とは区別して下さい。

トレードオフの関係

コンピュータで実現したい機能を、ハードウェアで実現するか、ソフトウェアで実現するかを考えます。両者はコンピュータの両輪ですから、どちらか一方だけというのはあり得ませんが、どれくらいのウエイトかと言うのは設計の際、選ぶことが出来ます。

例えば、2桁の短いソロバンと中間結果を記憶するメモを使って、これを5回繰り返せば10桁の長いソロバンで出来る計算を実行することが出来ます。この5回の計算を(時間的に直列に)プログラムで実行すればよいのです。したがって前者(2桁のソロバン)はソフトウェアに重心があり、後者(10桁のソロバン)はハードウェアに重心があります。

助け合いの関係

このような両者の重心をどこに置くかと言う問題は、コンピュータアーキテクチャの問題であり、設計者が選択することが出来ます。これをハードウェアとソフトウェアのトレードオフと言い、この様子を図1.5に示します。ハードウェア

ソフトウェアのウェイト

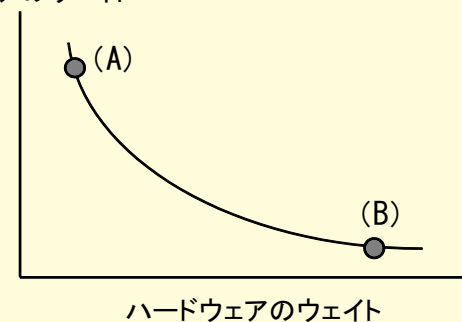


図1.5 ハードウェアとソフトウェアのトレードオフ

を極力少なくしソフトウェアに重心をおく領域(A)では、融通性は増しますが処理速度が低下します。逆に、ソフトウェアを少なくし、ハードウェアに重心をおく領域(B)では、処理速度は高くなりますが変更などに対する融通性は望めません。一度ハードウェアで実現すると修正は難しくなります。

いままで見てきたように、ハードウェアとソフトウェアは互いに補い合う関係にあります。

ノイマンアーキテクチャ(後述)のハードウェアは低コスト化が図られ、同時に極度に汎用的な構成法をとっています。一方、汎用的なハードウェアに解き

たい問題ごとに違った専用のソフトウェア(ただし、問題ごとに違った部分はごく一部で、大半は既製のソフトウェアの中から選択したもの)がインストールされ要求に合ったシステムを実現しています。ここで、既製のソフトウェアの主なものオペレーティングシステム(OS)とよばれるもので、パソコンの場合WindowsやMac OSなどがあります。

なお、コンピュータ開発にかかるコストは1970年代を境にハードウェアコストとソフトウェアコストが逆転し、現在ソフトウェア開発費の全体に占める割合は90%以上とされています。



図1.6に10進1桁のコンピュータを使って、 $718+464=1182$ を計算する例を示します。最初に加えたい2つの数(718,464)の最終桁(8,4)を加え結果(1,2)を得ます。(1)は最終桁から次の桁への桁上げで、これをメモリに記憶しておきます。次に、記憶した(1)をメモリから読み出しこれと2つの数の最終桁から2桁目の(1,6)を加え結果(0,8)を得ます。今回は桁上げがありませんので、メモリには(0)を記憶しておきます。このような計算を必要な桁数だけ繰り返せば、10進数1桁のプロセッサで任意の長さの桁の足し算が出来ます。ただし、足し算を繰り返し実行しているので時間がかかり、処理速度は低下することに注意して下さい。

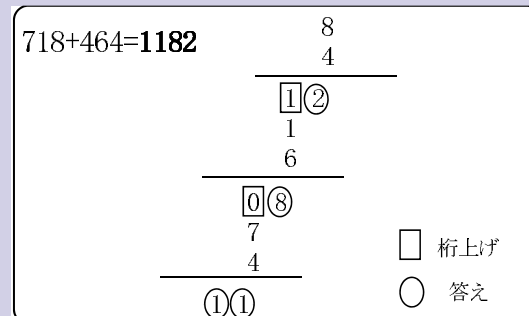


図1.6 10進1桁のプロセッサを用いた計算例

2.3 パソコンの外観と内部構造

パソコンの分類

パソコンの外観・内部機器構成などについて全体を把握し、パソコンの仕様を決める要素を理解します。パソコンの機能・性能を決める評価基準を明確にすることが大切です。写真 1.20 を見ながら、外観からパソコンを分類します。

(1) デスクトップ型パソコンには、タワー型とコンパクト型があります。タワー型デスクトップパソコンは、フルタワー型とマイクロタワー型に分けられます。いずれも立てて使うタイプです。

(2) ノート型パソコンには、A4 型・B5 型(CD・DVD ドライブ装置は外付け)・ミニノートなどが開発され実用化されています。

(3) 超小型パソコンは、5～6 型のワイド液晶を用いた 0.5kg 位の PC です。

モバイル型 PC とも呼ばれています。

パソコンの構成要素 (例)

図 1.7 はよく見るデスクトップ型のパソコンです。パソコンの構成要素を考えます。

(1) 本体：

本体には、マザーボード (CPU・主メモリ・ビデオメモリなどが実装されている)・電源ユニット・ファン・ハードディスクドライブ装置 (HDD) が含まれます。その他、CD・DVD ドライブ装置・PC カードスロット・USB ポート・拡張スロット・音声入力ポート・音声出力ポート・無線 LAN などが付いています。

(2) 入力装置：

入力装置には、キーボード・マウス・音声入力・WEB カメラが付けられます。

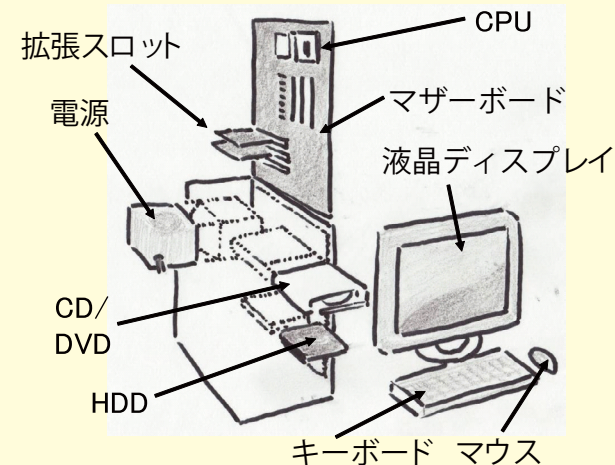


図 1.7 各装置の外観

(3) 出力装置：

出力装置には、液晶ディスプレイ・音声出力があり、プリンタをつなげるコネクタが付いています。

メモリ・HDD

・MIPS (Million Instruction Per Second)
・ベンチマークテスト・・・使用環境に近い状態で評価

パソコンの機能と性能

パソコンの機能・性能について考えます。多くの応用システムでは、使用する推奨環境があらかじめ明らかにされています。この中に、要求される機能と性能があります。機能は有無、性能は定量的に決められます。

通常は、予算に合わせ多機能、高性能を実現させることになります。その場合、コンピュータの性能評価方法は、何で測ればよいのでしょうか。次のように、いくつかの評価尺度が知られています。

- ・CPU クロック周波数
- ・データ転送速度 (内部)
- ・メモリ容量・・・主メモリ・キャッシュ

ベンチマークテストは良く用いられる方法で、ベンチマークデータと言う典型的な問題を解き、性能を評価します。また、事務計算や科学技術計算における命令の使用頻度に応じた実行速度を測り評価することもあります。

以上をまとめると、次の通りです。

- ・パソコンの内部構成機器について、本体・入出力装置に分けて考える。
- ・パソコンの仕様を決める要素には機能と性能に分ける必要がある。要求に合わせて選択しなければならない。
- ・パソコンの性能を決める評価基準にはクロック周波数・メモリ容量などの他、ベンチマークテストが用いられる。



写真 1.20 パソコンの外観
出典：フリー百科事典「ウィキペディア (Wikipedia)」

2.4 コンピュータの基本構成

コンピュータの基本的な構成を図 1.8 に示します。この 5 つの装置（または機能、これらを 5 大装置・5 大機能と言います）がそろって始めてコンピュータと名乗ってよいのです。まず、その構成要素について簡単に説明しましょう。

(1) 入出力装置

処理させようとするデータやプログラムをコンピュータに与えたり、処理された結果を取り出したりするための装置です。最近では、その用途に応じてさまざまな入出力装置が次々と開発され実用化されています。

(2) メモリ装置

プログラムや処理すべきデータの格納、中間および最終結果の保存を行うための装置であり、主メモリ（装置）と補

助メモリ（装置）とに大別されます。

通常、前者は高速小容量、後者は低速大容量のメモリ素子を用います。

ある処理を実行するときには、そのプログラム・データは主メモリの中に置かれていなければなりません。

主メモリは語（word）とよばれる適当な情報の単位ごとに通し番号の番地（address）が与えられています。

主メモリと補助メモリの速度と容量の違いをソフトウェアによりうまく使い分け、補い合う工夫がなされています。

(3) 演算装置

プログラムによって記述された命令に従って、指定されたデータに対して算術演算・論理演算などの各種演算処理を実行する装置であり、演算操作を実行するための算術論理演算装置（Arithmetic

Logic Unit: ALU）と、演算の途中結果や最終結果を一時格納するための累算器（アキュムレータ）やレジスタなどより構成されています。最近のマイクロプロセッサの集積度が上がっている理由の一つに、キャッシュメモリ（後述）の増加があります。なお、キャッシュメモリは勿論メモリですが、汎用レジスタと同様演算装置に含めることがあります。

(4) 制御装置

コンピュータを構成する各部分の機能をプログラムによって指示された命令通りに働かせるために全体を統括する装置です。5 つの装置の中で最も分かりにくい部分です。一連の命令は主メモリの中に命令語として格納されています。制御装置は主メモリに格納されている命令

を一つずつ次々と取り出して内容を解読し、その結果に基づいてコンピュータ各部に制御信号を送ったり、他の装置からの信号を受けとり命令を実行します。

CPU と（命令語・データ語が入っている）主メモリのやりとりを理解することにより、コンピュータの動きが分かります。

まず最初、制御装置と主メモリの間で命令を読み出・解読・実行をします。制御装置で 1 命令ごとに実行する動作を命令サイクルと言います。実行の段階で必要ならば演算装置を使います。（加算などの）命令の内容に従って、主メモリからデータを読み出し演算装置を使って（加算などの）演算をし、その結果を主メモリに書き込むことにより（加算などの）演算は完了します。



以上で 5 大装置を説明しましたが、さらに、次のような名称も使われています。

(5) 中央処理装置と周辺装置

上に述べたように、制御装置・演算装置は互いに密接な関係にあり、通常これらをひとまとめにして一つの装置として実現し、中央処理装置（Central Processing Unit: CPU と呼んでいます。図 2.4 に細い点線で示した部分が CPU です。マイクロプロセッサはこの部分を LSI 化したものです。主メモリの一部を制御・演算装置に含めて考える場合は、キャッシュメモリなど高速アクセスメモリを制御・演算装置のすぐそばに配置するため、LSI 上に一体化されています。

また、入出力装置と補助メモリ装置を周辺装置（peripheral unit）と呼ぶことがあります。フレキシブルディスクメモリ・USB メモリ・フラッシュメモリ・SSD（Solid State Drive）メモリ・HDD（Hard Disc Drive）メモリなど機能は補助記憶装置ですが、入出力装置に分類されることもあります。特に、SSD メモリ・HDD メモリなどは通常補助記憶装置ですが、持ち運びするなど使い方によっては入出力装置と考えてよいでしょう。

(6) 命令・データと制御信号

おおまかに言って、命令・データはまず入出力装置から主メモリに読み込まれます。主メモリのデータは、一連の命令で構成されたプログラムの指示に従って演算装置で処理（加工）され、結果は主メモリに蓄えられます。主メモリの結果を出力装置で取り出します。制御装置は命令ごとに異なった指示をします。これは制御信号となって各装置に配られます。また加算命令なら、主メモリから該当するデータを制御装置のレジスタ・アキュムレータに転送を指示する制御信号を出し、該当するデータがそろったら演算装置の加算回路を使って計算し結果を主メモリに転送するよう指示する制御信号を出すことにより実行されるという訳です。細かいステップを踏んではじめて一つの命令が実行されるのです。1 本のプログラムは、さらにこの命令を何千ステップも実行しなければならないのです。

コンピュータの中では命令とデータは区別されず、主メモリに蓄えられます。両者を区別できるのは、制御装置が命令サイクルが実行サイクルかによります。前者であれば命令、後者であればデータと認識するのです（したがって、命令語をデータ語と見て書き換えれば、プログラムが変更されたことになります。すなわち、自分で自分の処理内容（プログラム）を変えることが出来るのです。これは少し難しい説明ですが、ノイマン型の大きな特徴です。).

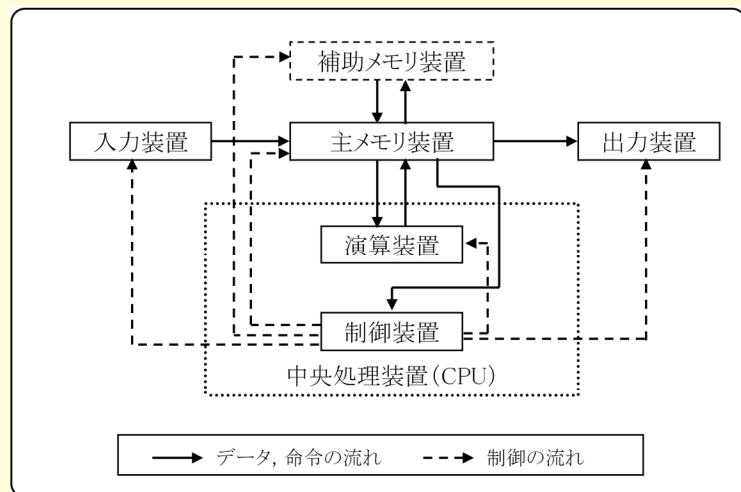


図 1.8 コンピュータの基本構成

2.5 中央処理装置と記憶装置

中央処理装置 (CPU)

前頁で述べたように、制御装置・演算装置は互いに密接な関係にあり、通常これらをひとまとめにして一つの装置として実現し、中央処理装置 (Central Processing Unit: CPU と呼んでいます。マイクロプロセッサはこの部分を LSI 化したものです。

コンピュータで行う高度な処理を考えると、CPU も相当複雑な演算を行っていると思ってしまうのですが、実は CPU 内で行われていることを分解すると、スイッチのオンとオフという単純な作業を行っているだけです。このスイッチのオンとオフを上手に制御することで複雑な処理を実現しています。

CPU 内部で、このオン／オフの作業は超高速に行われていて、その速度を動作周波数 (単位は Hz (ヘルツ)) と呼びます。たとえば、1MHz (メガヘルツ)

なら、1 秒間に 100 万回のオン／オフを行っていることになります。

主メモリ装置

主メモリ装置は、情報を記憶する装置の中で、すぐに CPU で利用される情報を記憶しておく装置です。ハードディスクなどの補助メモリ装置に記憶されている情報も一旦、主メモリに読みだしてから CPU で利用されます。

主メモリ装置はランダムアクセスメモリ (Random Access Memory: RAM) と呼ばれることもあり、格納された情報は任意にアクセスできるため読み出しや書き込みがしやすく、すぐに利用したい情報の一時的な格納場所として最適です。

CPU にとって、すぐに利用できる状態の情報が多ければ多いほど、ハードディスクなどから情報を読み出す回数が減るのでコンピュータ全体の処理速度を向上させることができます。



写真 1.21 CPU (Intel Core 2 Duo E6600)
出典：フリー百科事典
「ウィキペディア (Wikipedia)」



写真 1.22 メモリー
出典：フリー百科事典
「ウィキペディア (Wikipedia)」

補助メモリ装置

ハードディスクドライブは、磁気ディスクの表面を磁化することで、情報を長期間記憶しておく装置です。回転する磁気ディスクに対して同心円状に磁気ヘッドを移動させることで、必要な情報を読み出ししたり、空いているスペースに情報を書き込んだりします。ハードディスクドライブには複数枚の磁気ディスクが収められていて、記憶できる情報量を増やしています。また、ハードディスクドライブの読み出しや書き込み速度は、磁気ヘッドに移動速度や磁気ディスクの回転数で決まります。

光ディスクドライブは、CD や DVD、ブルーレイディスク (BD) などの光ディスクを利用して情報を読み込んだり書き込んだりする装置です。CD から DVD、さらに BD と光ディスクの進化に伴い、1 枚のディスクに記憶できる容量も飛躍的に増加しました。ただし、記憶速度に関しては、大容量の情報を記憶するにはそれなりの時間がかかります。光ディス

クドライブでは、「〇倍速」という表現で情報を記憶する速度を表しています。

ソリッドステートドライブ (solid state drive、SSD) は記憶装置として半導体素子メモリを用いたディスクドライブです。使用するメモリの種類により RAM を使う RAM ディスク、フラッシュメモリを使う Flash SSD などがあります。ハードディスクドライブと比較すると、シークタイムがないためランダムアクセス性能に優れ、物理的な稼働箇所がないため省電力、静か、振動・衝撃に強いといった特徴があります。逆に、容量単位の価格がハードディスクドライブより高価、フラッシュメモリを用いたものでは素子が劣化などの欠点もあります。



写真 1.24 光ディスクドライブ
出典：フリー百科事典
「ウィキペディア (Wikipedia)」



写真 1.23 ハードディスクドライブ
出典：フリー百科事典
「ウィキペディア (Wikipedia)」



写真 1.25 ソリッドステートドライブ (SSD)
出典：フリー百科事典
「ウィキペディア (Wikipedia)」

2.6 入力装置と出力装置

入力装置

コンピュータに操作の指示を与える入力装置で、もっとも基本的なものが、キーボードとマウスです。またノートパソコンの場合、その形状の制約からマウスの代わりにキーボードの手前にタッチパッドが備わっています。さらに、タブレット型のコンピュータの場合には、画面そのものがタッチパネルになっています。

キーボードは、おもに文字入力を行うための入力装置です。アルファベットや数字、記号、ならびに Ctrl キーなどの動作用のキーで構成されています。文字入力以外にも動作用のキーとアルファベットキーなどを組み合わせることでファイルに保存や検索など、マウスを使った操作を代用することもできます。

マウスは、パソコンの画面上に表示される矢印（ポインタ）やアイコンを操作するポインティングデバイスと呼ばれる

装置の一つです。ポインティングデバイスには、ほかにノートパソコンで主流のタッチパッドや、丸いボールを転がしてポインタを移動させるトラップボールなどがある。

キーボードとマウスをコンピュータに接続するには、現在では USB を用いた接続が一般的ですが、利便性を考慮して無線通信 (Bluetooth) など使うことも増えてきています。



写真 1.27 マウス
出典：フリー百科事典
「ウィキペディア (Wikipedia)」



写真 1.26 キーボード
出典：フリー百科事典
「ウィキペディア (Wikipedia)」



写真 1.28 タッチパッド
出典：フリー百科事典
「ウィキペディア (Wikipedia)」

出力装置

ディスプレイは、コンピュータの操作画面や処理結果を表示する装置です。出力装置の一つです。かつてはブラウン管を用いた CRT ディスプレイが主流でしたが、現在では液晶ディスプレイが一般的になりました。映像信号の伝送が RGB アナログ信号の場合は VGA 端子、デジタル信号の場合は、DVI 端子や HDMI 端子などでコンピュータと接続されます。画面を構成する素子（画素）の数によって視覚的な画面の緻密さが表現されます。このことを解像度と呼び、例えば WUXGA という規格では 1920 × 1200 ドット、つまり横一辺に 1920 個の素子（ドット）、縦一辺に 1200 個の素子があることを示しています。

プリンタは、ディスプレイと同様にコンピュータの主要な出力装置の一つです。コンピュータから送られてきた情報を紙などの媒体に印刷します。インクジェットプリンタは、主に液状のインクを微粒子化し、加圧や加熱などにより微

細な孔から射出させる方式で印刷します。レーザープリンタは、帯電させた感光体にレーザー光などを照射し顔料粉末（トナー）を付着させ、用紙に転写した上で熱や圧力をかけて定着させる方式です。プリンタとコンピュータとは、USB 端子を用いた接続が主流となっているが、近年では複数のコンピュータから印刷したり、コンピュータ以外から印刷する要求が増えているため、無線通信でコンピュータと接続できるものも増えてきています。オフィス環境では、ネットワークに接続したすべてのコンピュータから印刷できるネットワークプリンタが主流です。



写真 1.30 インクジェットプリンタ
出典：フリー百科事典
「ウィキペディア (Wikipedia)」

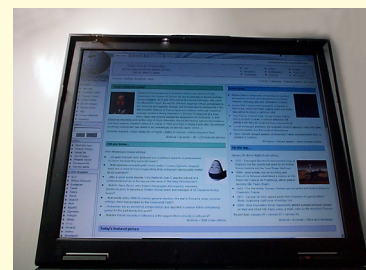


写真 1.29 ディスプレイ
出典：フリー百科事典
「ウィキペディア (Wikipedia)」



写真 1.31 レーザープリンタ
出典：フリー百科事典
「ウィキペディア (Wikipedia)」

演習問題

- 次の空欄 [a][b] に入る適切な語の組み合わせを、(1)-(4)の中から選びなさい。
コンピュータのハードウェアは [a] を持つように設計され、ソフトウェアはそれに [b] を持たせる役割を持っている。
(1) [a] 専用性 [b] 汎用性
(2) [a] 基本性 [b] 応用性
(3) [a] 汎用性 [b] 専用性
(4) [a] 処理性 [b] 専用性
- 次の文章で間違っているものを選びなさい。
(1) コンピュータは必ずハードウェアとソフトウェアを持っている。
(2) 組み込みシステムはマイクロプロセッサによって、特定の機能を実現したものである。
(3) 組み込みシステムは特定の機能を持つもので、したがってソフトウェアの要素はない。
(4) コンピュータにおける処理とは、計算・記憶・通信の各機能を指す。
- 次の空欄 [a][b] に入る適切な語の組み合わせを、(1)-(4)の中から選びなさい。
ハードウェアは [a] を言い、ソフトウェアは狭義にはハードウェア能力を効果的に利用するための [b] を言う。
(1) [a] 金物 [b] プログラム・データの集合体
(2) [a] 叩けば音がするもの [b] プログラム
(3) [a] 電子回路 [b] プログラムの集合体
(4) [a] 金物 [b] 総合的概念
- 次の文章で間違っているものを選びなさい。
(1) ソフトウェアは無体材である。
(2) ソフトウェアが売買されるとき、対象は成果物である。
(3) ハードウェアとソフトウェアには機能を実現する際、トレードオフがある。ハードウェアのウエイトが大きければ高速化が望める。
(4) アルゴリズムはコンピュータが得意とする方法で問題を解く手順で、実行できる言語で記述されていなければならない。
- 次の空欄 [a][b][c][d][e] に入る適切な語の組み合わせを、(1)-(4)の中から選びなさい。

コンピュータの基本機能は [a][b][c][d][e] の各機能で、中央処理装置 (CPU) の機能は [a][b] を指す。
(1) [a] 入力 [b] 記憶 [c] 演算 [d] 制御 [e] 出力
(2) [a] 入力 [b] 出力 [c] 演算 [d] 記憶 [e] 制御

- (3) [a] 演算 [b] 記憶 [c] 制御 [d] 入力 [e] 出力
(4) [a] 演算 [b] 制御 [c] 入力 [d] 記憶 [e] 出力
- 制御装置に関する次の文章で間違っているものを選びなさい。
(1) 命令サイクルを持ち、1 命令ごと実行する。
(2) 命令語は解釈サイクルで、データ語は実行サイクルでそれぞれ読み出される。
(3) 一つの命令サイクルの中で割り込みの有無がチェックされる。
(4) 実行サイクルでは、プログラムによって指示された演算などの内容が実行される。
- 次の空欄 [a][b][c] に入る適切な語の組み合わせを、(1)-(4)の中から選びなさい。
世界最初の完成型パソコンは [a] 年 [b] 社が発表した [c] とされている。したがって、[a] 年はパソコンの幕開けの年と言われている。
(1) [a] 1971 [b] インテル [c] MCS-4
(2) [a] 1976 [b] NEC 社 [c] TK-80
(3) [a] 1975 [b] ミッツ [c] アルティア 8800
(4) [a] 1977 [b] アップルコンピュータ [c] アップル II
- 次の文章で間違っているものを選びなさい。
(1) インテル社製 4004 型 LSI は 4 ビットのパソコンでマイクロプログラム制御方式を採用している。
(2) ミッツ社製のアルティア 8800 は S-100 バスを採用し、互換性を持つ機種開発に都合が良かった。
(3) アップルコンピュータ社製アップル II は技術情報公開を行ったため、爆発的に販売台数を伸ばした。
(4) IBM 社製 IBM PC は、16 ビット型パソコンでフレキシブルディスク装置を実装し、ビジネス用を狙った。

第 3 章 制御と演算

ほげほげ

3.1 命令語とデータ語

3.2 命令の種類

3.3 命令セットアーキテクチャの分類

3.4 2 進数と 10 進数

3.5 データ形式 (1)

3.6 データ形式 (2)

3.1 命令語とデータ語

3.2 命令の種類

3.3 命令セットアーキテクチャの分類

3.4 2進数と 10進数

3.5 データ形式 (1)

3.6 データ形式 (2)

第 4 章 2 進演算と論理演算

ほげほげ

4.1 2 進演算 (1)

4.2 2 進演算 (2)

4.3 ブール代数とブール関数

4.4 論理素子と回路

4.5 論理設計 (1)

4.6 論理設計 (2)

4.1 2 進演算 (1)

4.2 2 進演算 (2)

4.3 ブール代数とブール関数

4.4 論理素子と回路

4.5 論理設計 (1)

4.6 論理設計 (2)

第5章 パソコンのハードウェア (1)

ほげほげ

5.1 CPU (1)

5.2 CPU (2)

5.3 主メモリ

5.4 キャッシュメモリ

5.5 仮想メモリ

5.6 バス / 入出力インタフェース

5.1 CPU (1)

5.2 CPU (2)

5.3 主メモリ

5.4 キャッシュメモリ

5.5 仮想メモリ

5.6 バス / 入出インタフェース

第6章 パソコンのハードウェア (2)

ほげほげ

6.1 HDD

6.2 光ディスク

6.3 フラッシュメモリ

6.4 SSD

6.5 USB

6.6 NAS

6.1 HDD

6.2 光ディスク

6.3 フラッシュメモリ

6.4 SSD

6.5 USB

6.6 NAS

第7章 オペレーティングシステム

ハードウェアの話が終わったので、ソフトウェアの話に入ります。ここでは、ソフトウェアと言ってもプログラミングではありません。まず、オペレーティングシステム (Operating System: OS) がなぜ必要であったか、その歴史を振り返ります。次に、パソコンのオペレーティングシステムについて Windows OS 系と Mac OS 系の歴史・機能などを把握します。特に、基本ソフトウェアの重要な管理機能についてその概要を説明します。

7.1 OS の歴史

7.2 OS の役割と機能

7.3 パソコンの基本ソフトウェア

7.4 プロセス管理

7.5 メモリ管理 (1)

7.6 メモリ管理 (2)

7.1 OS の歴史

なぜ、どのようにしてオペレーティングシステムが作られたか、OS の歴史について学びます。ここではパソコンの OS (例えば Windows) に限定しません。パソコンの OS は突然登場した訳ではなく、メインフレーム・ミニコンなどの OS の長い歴史を受け継いでいることが分かります。したがって、用語・構成法・機能・技術・実現法などはメインフレームの OS から引き継いでいます。

OS 変遷の歴史

図 1.9 に OS 変遷の歴史を示します。OS の原型は図中、1950 年代前半にモニタとして誕生します。マルチプログラミング可能になり、やがて汎用 OS に変遷します。一方で、OS が巨大になり過ぎたと言う反省から UNIX が出現します。パソコン用の MS-DOS はシングルタスク・シングルウィンドウであったものが、Windows OS 系になりマルチタスク・マルチウィンドウに変わってきました。

OS 誕生以前

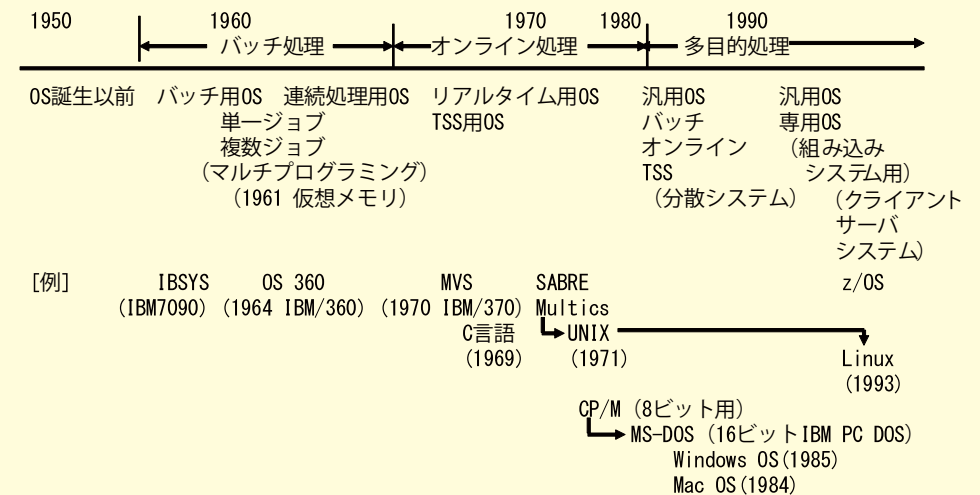
OS 誕生以前はブートストラップローダというコンピュータにプログラムやデータを読み込ませるのに必要な最低限のプログラムと、計算開始前のテストプログラムを手で使っていました。前者は当時多くは紙テープで出来ており、コ

ンピュータのコンソールパネルのスイッチを用いてインストールしました。後に、これがモニタ・IOCS(Input Output Control System) となり、少しだけ使いやすくなります。この時代はプログラマ自身がオペレータを務めるオープンジョブ制でした。これは、現在専有して使うパソコンを使う状況とほぼ同じです。

バッチ処理用 OS

バッチ処理はリアルタイム処理と対比される処理形態です。後者はデータが発生する度に処理をするのに比べ、前者はデータをまとめて一括して処理をします。

スループットを向上させるため、次々と、(a) ジョブの連続処理、(b) 複数ジョブの連続処理、(c) マルチプログラミングによるジョブの連続処理を行う技術が開発されました。その結果、(b) はコンピュータの操作を専用のオペレータに任せるクローズドジョブ制が採用されるようになりました。プログラマとオペレータは切り離され、プログラマはコンピュータルームに入らなくなります。(c) でマルチプログラミング技術が開発され、複数のジョブがあたかも並列的に処理されるようになり、リモートバッチ処理や SPOOL(Simultaneous Peripheral Operation On Line) が実用化されます。SPOOL によりプリンタの並列動作 (異



CP/M (Control Program for Microcomputer): 8080用
MS-DOS (Microsoft Disk Operating System) :16ビット用 シングルタスク・シングルウィンドウ

図 1.9 OS の変遷

なるジョブの出力) が可能になります。ここで、ジョブとはユーザから見た仕事の単位 (例えば、給与計算、日計表の作成など) で、スループットは単位時間当たりの (実効) 処理能力を示します。

リアルタイム処理用 OS

ハードウェアが高価な時代にコンピュータを遊ばせない方法として、さらにバッチ処理ジョブの応答時間の短縮・スループットの向上が図られます。応答時間の短縮のためには割り込み処理・優先処理が、さらにネットワークと端末を用いたオンラインリアルタイムシステムも実用化されました。

タイムシェアリング処理用 OS

ネットワークと端末を用い、大型コンピュータを多数の人が同時に利用する形

態として、TSS(Time Shearing System) がアメリカを中心に盛んに用いられた。MIT の Multics は有名です。TSS はユーザにとって、コンピュータと会話形式で利用できるという特徴があります。

OS の汎用化と統合化

OS はさらに進化して、リアルタイム処理とバッチ処理の並行処理が可能となります。リアルタイム処理をしながら、バックグラウンドジョブとしてバッチ処理をします。これにはデータベース管理システムの普及があります。OS の統合化により、いろんな使い方を可能にする方法として、ハードウェア機能をソフトウェアで模倣する仮想コンピュータ (存在しないコンピュータを実現する) の考え方が生まれます。

7.2 OS の役割と機能

OS の役割

次に、OS の役割について考えます。
次のようなものがあります。

● (1) ハードウェアの管理と有効利用
資源（CPU・主メモリ・外部メモリ・入出力機器）管理，すなわち資源の奪い合いの調整を行い，マルチプログラミング環境下における CPU の有効利用を図ります。コンピュータ自身が自分を最適な状態に保つためです。

● (2) プログラムの負荷軽減
ハードウェアの仮想化・独立化によりプログラムの負荷を軽減します。主として，仮想メモリ技術により，主メモリの容量を仮想的に十分大きくしプログラムをメモリ容量制限から開放し，また言語プロセッサにより高級言語でプログラミングを楽にするなどによります。

● (3) オペレータの負荷軽減
コンピュータの動作を分析し，起こり得る事態をあらかじめ想定し，誤り発生時や異常処理に備えます。その結果，正常な処理はコンピュータ自身で行い，オペレータは異常時のみ対応すればよいようにします。また，現在パソコンではマルチウインドウ・アイコンによる GUI(Graphical User Interface) が採用さ

れていますが，これもオープンショッパ制の負荷低減の一つです。

● (4) ネットワークとの結合
OS にはネットワークアーキテクチャが提供されており，ネットワークを簡単に利用することが出来ます。

● (5) その他
OS には異常診断・復旧処理・システム再構成・データの機密保護・破壊保護・課金・統計データの蓄積・(オペレーションの) ミスによるダウンの防止・変更に伴う機能の追加・拡張・修正・消滅など様々な機能があります。

OS の効果

次に，OS を用いた効果について説明します。OS がいない場合に比較すれば明らかですが，

- (1) 処理能力の向上
- (2) 応答時間の短縮
- (3) 使い易さ（ユーザインタフェース）の向上
- (4) 拡張性・利用性の向上
- (5) RAS 機能・セキュリティ機能の向上
- (6) 共同作業の支援などが期待できます。

ここで，RAS は Reliability Availability Serviceability の略で，信頼性・可用性・保守性を意味し

ています。信頼性はハードウェアやソフトウェアが故障する頻度が小さく故障している期間が短いこと，可用性は利用者が安定して使用でき，一定の期間内で機能を維持している割合が高いことで，通常稼働率で評価されます。保守性は保守しやすいことを指しています。さらに，Integrity（保全性）と Security（安全性）を RAS に加えた RASIS という指標もあります。

OS の構成と機能

最後に OS の構成と機能をまとめておきます。簡単に言えば，OS とはハードウェア（CPU・メモリ・入出力機器など）を動かすためのソフトウェアで，ハードウェアに最も近い下位のソフトウェアに位置づけられます。もう少し付け加えると，ハードウェアの抽象化，すなわち応用ソフトウェア開発のための（ハード

ウェアの）インタフェース統一機能をもつものと言えます。なお，多くのソフトウェアが共通して利用する機能は，OS（ミドルウェア）などの形で提供されています。その結果，OS の構成を図 7.2 のように示すことが出来ます。ここで，応用ソフトウェアはアプリケーションプログラム（Application Program: AP）とも言います。以上をまとめると，下記の通りです。

- ・OS はメインフレームの時代から長い歴史がある。
- ・OS はハードウェアに最も近いソフトウェアである。基本ソフトウェア（カーネル）と応用ソフトウェアに分けられる。
- ・OS の重要な機能は ①ハードウェアを有効利用するための制御・管理と ②ユーザに使いやすい環境を提供することである。

応用ソフトウェア

基本ソフトウェア

ハードウェア

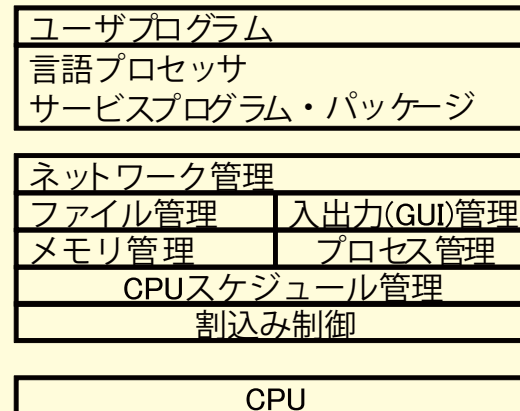


図 1.10 OS の変遷

7.3 パソコンの OS

ここでは、パソコンの OS、特に Windows OS 系と Mac OS 系の歴史・特徴を取り上げます。

パソコンの OS

パソコンの OS として、Windows OS 系と Mac OS 系を取り上げます。これ以外にワークステーションでは、Unix OS 系、Linux OS 系があります。な

お、組み込みシステム用 OS としては、ITRON (Industrial TRON)、MES(Micro Embedded System) などがあります。まず、表 1.7 に簡単に Windows OS と Mac OS の比較結果を示します。

OS の機能

OS の機能は外部から見て、パソコンの起動・終了処理、アプリケーションの

表 1.7 Windows OS と Mac OS の比較

	Windows	Mac
OS 開発会社	マイクロソフト社	アップル社
搭載ハードウェアメーカー	インテル社中心の全メーカー (アップル社を除く)	アップル社
世界的シェア	90% 以上	10% 以下
得意分野	汎用	音楽・画像・映像・組版など多色画像処理
特徴	汎用	サウンド系・アート系
機能	豊富	多くない
応用ソフトウェア	多い	少ない

参考文献：石田晴久，新パソコン入門，岩波書店，p.153, 2000.11.

表 1.8 OS の管理機能（基本ソフトウェア）

BIOS*	周辺機器（HDD・キーボード・ビデオカードなど）を制御管理するプログラム群、マザーボード上に不揮発性メモリで装着
プロセス（タスク）管理	マルチタスクの実現（プログラム実行の管理）・・・タスクの切り替え・CPU の配分
メモリ管理	仮想メモリの実現（メモリの階層構造と領域の管理）・・・メモリの割り当て・解放・スワップ
ファイル（データ）管理	ディレクトリによる階層構造管理・・・ファイルの共有と排他制御・障害対応（電源断時など）
入出力管理	入出力機器・周辺機器（デバイスドライバ）の管理
GUI**	マルチウィンドウ・アイコン・メニュー・マウスによるインタフェース管理

*BIOS：Basic Input Output System（基本入出力システム）

**GUI：Graphical User Interface（1970 年代，ゼロックス社 Alto が原型）

CUI：Character based User Interface（MS-DOS のようにテキストベースでコマンドを入力する。Windows のもアクセサリ「コマンドプロンプト」がある）

注：表 1.8 のような分類・記述は 1964 年 IBM 社から発表された IBM/360 に準じている。

表 1.9 Windows OS の流れ

年	Windows	役割・特徴
1985	1.0	最初の Windows OS 試作
1992	3.0	GUI 対応
1993	NT	企業向け
1995	95	OS として独立
1998	98	低マシンスペックで動作
2000	2000	NT の改良版，サーバ向け UNIX に対抗
2001	XP	個人向けとサーバ向け
2007	Vista	ファイル管理の軽減，起動時間短縮
2009	7	検索機能，性能向上
2012	8	7 の後継，パソコン・タブレット端末用

インストールなどの機能があります。内部的には、仮想メモリ、アプリケーションの共通機能 API（入出力・画面表示・プリント・保存など）、マルチプロセス（複数作業の並行処理）、マルチユーザ（ユーザごとのプライバシー保護）、ユーティリティの提供（インターネットブラウザ・メールソフト・セキュリティ対策など）があります。ここで、API は Application Program Interface の略称です。表 1.8 に OS の基本ソフトウェアに含まれる主な管理機能を示します。

パソコンの OS の歴史

(1) Windows OS 系

表 1.9 に Windows OS の歴史とそれぞれの特徴をまとめておきます。Windows 1.0 は最初の試作ですが、余り売れなかったようです。実用化されたのは Windows 3.1 からです。特に、NT はオフィス向けに普及しています。そして、95 で定着し不動の地位を得ています。2000 と Me はサーバ向けと個人向けに分けたものですが、不評に終わった

表 1.10 Mac OS の流れ

年	Windows	役割・特徴
1984	System 1	Macintosh 発売時の OS
1986	漢字Talk 1.0	System 3.1 をベースに日本語対応
1996	System 7.5.5	System シリーズ最後のバージョン
1997	Mac OS 7.6	統一 OS
1998	Mac OS 8.5	Power PC 専用，フォント環境改善
2001	Mac OS X v10.0	UNIX 技術を取り入れる
2009	Mac OS X v10.6 (Snow Leopard)	Intel プロセッサ専用
2012	OS X v10.8 (Mountain Lion)	iOS 5 で提供されたアプリケーションや新機能が Mac でも利用可能

と言われています。

(2) Mac OS 系

次に、同様に Mac OS 系も表 1.10 に示します。Mac OS は 1986 年までを第 1 期，1998 年までを第 2 期，1999 年までを第 3 期，それ以降を第 4 期と区別されています。第 2 期の途中で漢字系がなくなっています。1997 年の統一 OS が正式な Mac OS と言われています。

以上をまとめると、下記の通りです。

- ・最初の Windows OS 系・Mac OS 系は、いずれも 1980 年代半ばに開発・発表された。

- ・パソコンの基本ソフトウェアには、プロセス・メモリ・ファイル・入出力のほか GUI 管理機能（パソコン独特のもの）がある。

- ・Windows OS 系は圧倒的なシェアを持ち、企業向け汎用 OS として利用されて来た。

- ・Mac OS 系はその開発の歴史から、サウンド系・アート系を得意分野とする独自の歴史がある。

7.4 プロセス管理

プロセス管理

プロセス管理（タスク管理）機能は、主にプロセスへの資源（CPU・主メモリ・入出力機器など）の割り当てと、複数プロセスの並行処理です。前者は、優先処理とスケジューリング、後者は、割込み制御機構とマルチプログラミング技術によっています。

ここで、プロセスとは、コンピュータから見た仕事の単位です（ジョブは人間から見た仕事の単位です）。プロセスはジョブを細かく分けて、これ以上分けられない仕事の単位です。TSS でもそうですが、CPU があるユーザの仕事のみに関わると、他のユーザが長く待たされることになります。リアルタイムシステムならば、回線上のデータが失われたり、処理の実時間性が保てなくなります。したがって、あるプロセスを次の実行まで

待たせてよい時間内に、他のプロセスに切り替え、外から見て 1 台の CPU で並行して複数のプロセスを走らせる訳です。

プロセスの状態遷移

図 1.11 にプロセスの状態遷移図を示します。図中左端からプロセス生成されると、まず「実行可能状態」になりディスパッチ（運ぶ）されると「実行状態」になります。実行状態のプロセスはある程度実行されると（タイムスライス完了）、他のプロセスの実行のため再び実行可能状態になります。通常は実行可能状態と実行状態を往復します。しかし、入出力待ちなどこれ以上続けられないときは「待ち状態」になり、入出力などが終了すれば再び実行可能状態になります。このようにして、プロセスがすべて実行されると「プロセス完了状態」になります。

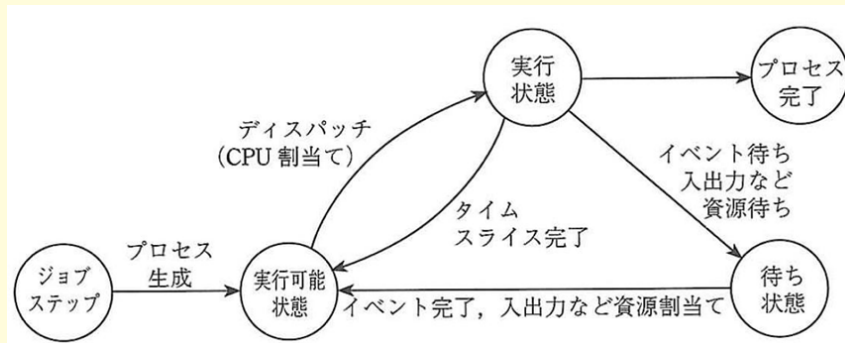


図 1.11 プロセスの状態遷移

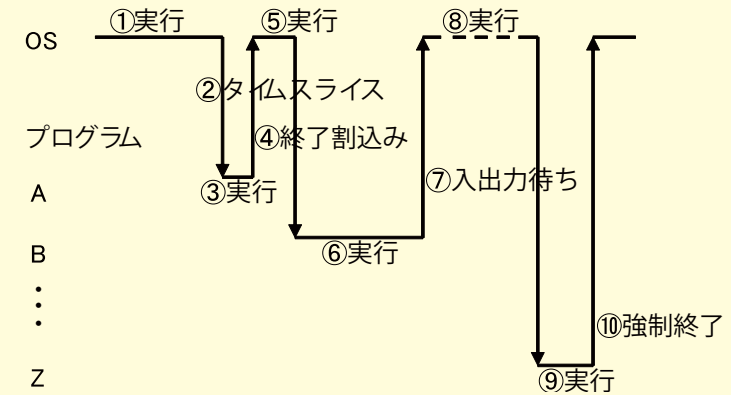


図 1.12 割込みとマルチプログラミングの例

割り込みとマルチプログラミング

図 1.12 に OS とプログラム A, B, C, ..., Z の実行が切り替えられる状態の例を示します。横軸は時間を示します。図中① OS を実行中、② タイムスライスでディスパッチされ③ プログラム A のプロセスに実行が移り、④ タイムスライス（時間割り当て）などで再び⑤ OS に実行が移ります。また、⑥ プログラム B が一連の処理の最後に入出力命令を実行する場合、プリンタなどにデータを渡した

後、⑦ 入出力命令の実行待ちとなり、再び⑧ OS に戻ります。

スケジューリングアルゴリズム

プロセス管理の最後に、表 7.4 にプロセススケジューリングのアルゴリズムを示します。FIFO 法は銀行の窓口処理のように、待ち行列に並んだ順に処理される最も一般的な方法です。優先順位法はオンラインリアルタイムシステムによく用いられます。ラウンドロビン法は TSS に用いられた方法です。

表 1.11 プロセススケジューリング法

方式名称	アルゴリズム	特徴
FIFO (First In First Out) 法	実行可能になったプロセスの順に CPU を割りつける方法	最も単純であるが、あるプロセスの CPU 専有時間が長いと他のプロセスの待ち時間が大
優先順位 (priority) 法	あらかじめ決められたジョブの優先順位に応じてプロセス実行優先順位をきめる方法（ジョブの優先順位はパラメータで入力する）	緊急度の高いジョブを実行する場合には有効
ラウンドロビン (round robin) 法	CPU 占有時間として一定のタイムスライスをきめる。時間内に処理が完了しない場合には、プロセスを実行可能状態とし、その待ち行列の最後に置く方法	TSS のように多数のユーザが同時に計算機を共有する場合に、CPU の占有防止ができ、均等化が可能
ダイナミックディスパッチング法	実行状態が待ち状態になるまでの時間が長いほど実行優先順位を低くする方法	並列処理の度合を高め、入出力の多いジョブのスループット向上が可能
デッドラインスケジューリング法	終了目標時間を定め、これに近くなるに従い処理優先順位を高くする方法	TSS の応答時間やバッチ処理時間に対して、目標時間内に処理を完了することが目的

7.5 メモリ管理 (1)

次の重要な OS に役割はメモリ管理機能です。ここでは、仮想メモリを取り上げます。以前、CPU の説明の時にキャッシュメモリについて説明しました。キャッシュメモリは主メモリより高速に動作する必要があるため大半をハードウェアの力を借りていますが、仮想メモリはソフトウェアの力が活躍します。

仮想メモリ

仮想メモリは、①ハードウェアとしてはメモリ階層・動的アドレス変換を用います。②ソフトウェアとしてはページ置換アルゴリズム、またはプログラムの論理構造による置換が大きな役割を果たします。ここで、キャッシュメモリで述べた通り、プログラムとデータの局所参照性を利用します。局所参照性には、
(a) 時間的局所性
(b) 空間的局所性
があります。

(a) は一度参照されたページは近い将来、再び参照される可能性が大きい、(b) は一度参照された命令語・データ語の近傍は、再び参照される可能性が大きいという性質です。

ここで、空間とは場所を意味しています。

次に、図 1.13 に仮想メモリの位置づけを示します。実在するメモリは高速ではあるが小容量か、低速だが大容量のいずれかです。高速大容量のメモリが欲しいのですが、なかなかそうはいきません。そこで、主としてソフトウェアにより（ハードウェアとしては）存在しない仮想的 (Virtual) なメモリを作ります。

そのために、図 1.14 のように高速小容量のメモリを上、低速大容量のメモリを下にし、全体として階層構造を持ったシステムを考えます。これをメモリ階層と言います。このようにして、急ぎ

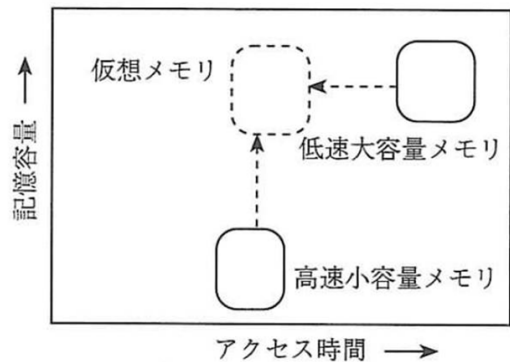


図 1.13 仮想メモリの位置づけ

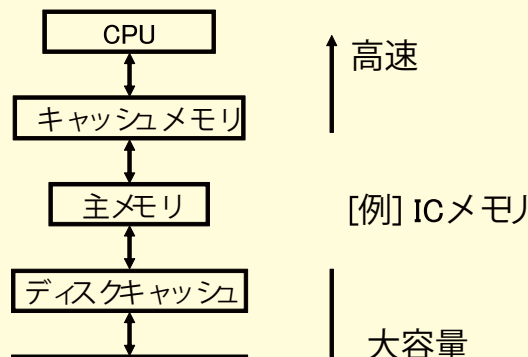


図 1.14 メモリ階層の例

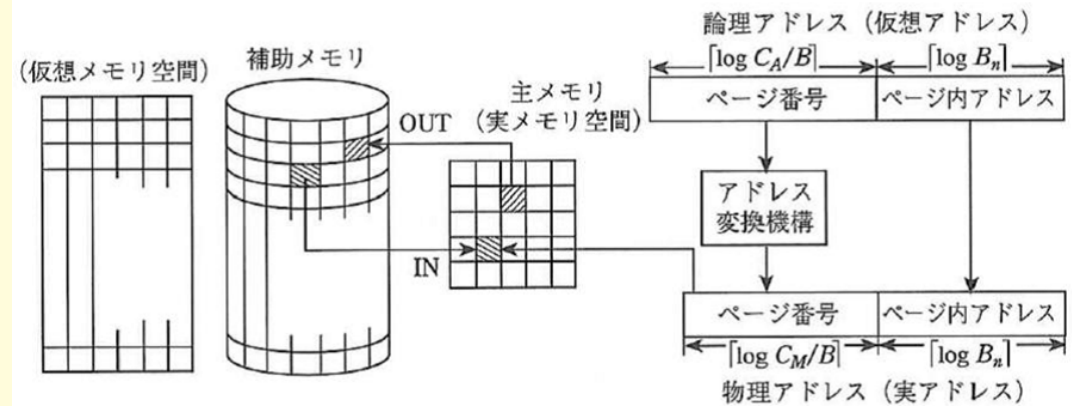


図 1.15 仮想メモリの実理

のもの（命令語・データ語）は上に、急ぎでないものは下に記憶するようにすれば、CPU にはいつも必要なものが上位に存在し、参照される訳です。常にそう出来ればいいのですが、当然うまくいかない場合もあります。その場合は下にあるものを取りに行く訳です

が、その割合（ミス率）が例えば 10～20% あれば、十分であることが知られています。

スケジューリングアルゴリズム

図 1.15 に仮想メモリの実理を示します。仮想メモリ空間は大きく、主メモリ空間は相対的に小さいとします。前者は低速大容量のディスクなど、後者は高速小容量の IC メモリなどで実現します。大きさの異なるメモリ空間のアドレス変

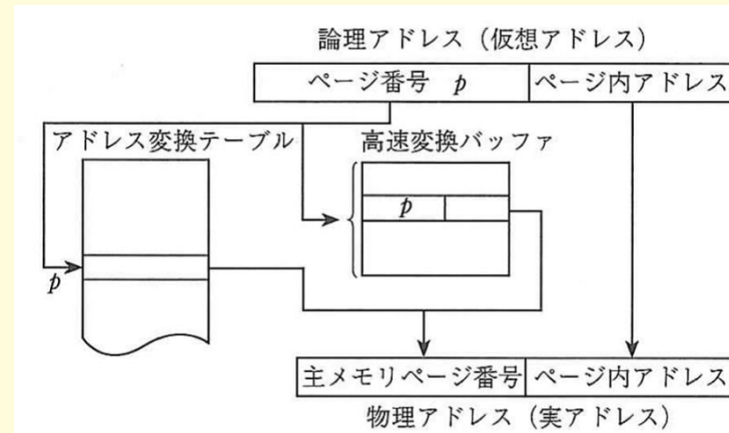


図 1.16 ページングにおけるアドレス変換

換を効率良くするために、仮想メモリに必要なもう一つのハードウェアが動的アドレス変換機構です。

この部分を図 1.16 に示します。高速バッファは、この中に対応表があれば高速に（外部メモリ上の）論理（仮想）アドレスのページ番号を（主メモリ上の）物理（実）アドレスのページ番号に変換します。

7.6 メモリ管理 (2)

ページ置換アルゴリズム

前節でみたように、統計的な性質である局所参照性を用いても、当然欲しいものがない確率（ミス率）が存在します。その場合には、メモリ階層の下にあるメモリから取り出さなければなりません。メモリ容量は一定ですから、不要と思われるページを吐き出す（ページアウト）しなければ必要なページが入りません。表 7.6 にページングにおけるページ置換アルゴリズムを示します。

このアルゴリズムの動きは、図 7.5.5 を見れば理解できます。真ん中の長方形は主メモリのページの並び（待ち行列状）を示します。FIFO アルゴリズムは主メモリに入って最も古いものを吐き出す方法です。LRU アルゴリズムは主メモリのページが参照されるたびにそのページを待ち行列の最後尾に移す必要がありますが、参照後常に最も古いものが最前列に来ます。このアルゴリズムは現在の主流です。

表 1.12 ページングにおけるページ置換アルゴリズム

方式名称	アルゴリズム
FIFO (First In First Out) アルゴリズム	ページインしてから、最長時間経過したページをページアウトする方法
LRU (Least Recent Used) アルゴリズム	最後に参照されてから、最長時間経過したページをページアウトする方法

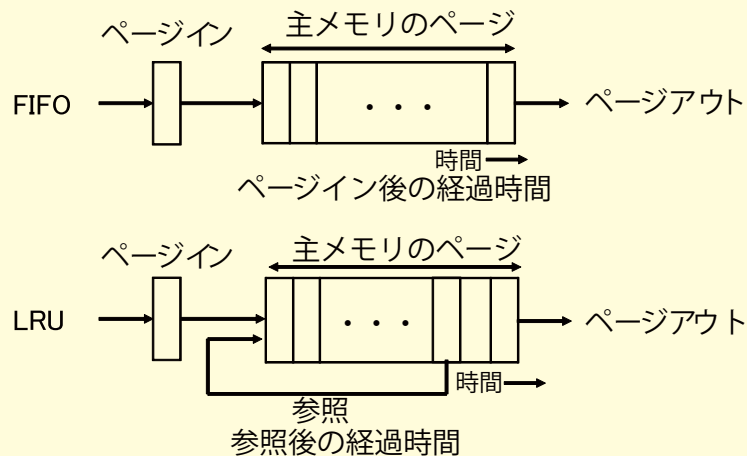
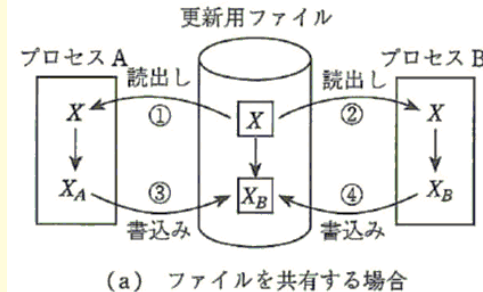


図 1.17 FIFO と LRU 待ち行列

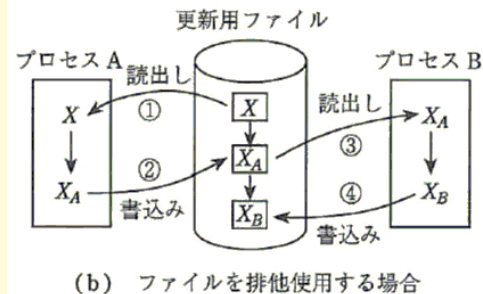
ファイル管理

ファイル管理（データ管理とも言う）機能は外部メモリ、入出力ユニットのデータセット（ファイル）を IOCS を用いて簡単かつ効率よく利用できるようにします。ファイル編成法や入出力スケジューリング機能を実現します。ここでは、オンラインリアルタイムシステムなどで問題となるファイルの共用と排他制御について説明します。

図 7.6.2 ファイルの更新例を示します。(a) は複数ユーザでファイルを共用する場合です。③プロセス A で書き込まれたデータは、④プロセス B により上書きされ、XA は消去されます。(b) は



(a) ファイルを共有する場合



(b) ファイルを排他使用する場合

処理順序 ①→②→③→④

図 1.18 ファイルの更新

ファイルを排他的に使用する場合があります。②プロセス A で書き込まれたデータ XA は読み出しのみで書き込みはブロック（禁止）されます。③でプロセス B が読み出し処理をした後、⑤で書きこまれ、元の XA は XB に更新されます。みどりの窓口などで発券し、残席数を更新するような場合を考えればよいと思います。



ひらさわ教授のコラム 3

今回は、昔コンピュータに初めて出会ったときの印象を述べます。それは、学生時代の 1962 年の夏です。大学が初めてコンピュータ LGP-30 を導入し、これを用いて学生に夏期講習を実施し、その演習を受けたときです。仕様はほとんど記憶も資料もないので、Wikipedia で調べました。おおよそ、次のようなコンピュータです。1956 年、Royal Precision Electronic Computer Company によって製造され、価格は約 \$40,000。2 進法を用いた 32[ビット]/[ワード] で主メモリが 4096[ワード] の磁気ドラム。素子はダイオードと真空管と言ったところが主な仕様らしいことが分かりました。紙テープリーダとタイプライタが付いていたように記憶しています。丸いブラウン管オシロスコープも付いていました。磁気ドラムは 64 トラックでそれぞれ 64 セクタ（要するに $64 \times 64 = 4096$ のワードが円筒状に配置されている）を持っています。プログラムを記憶させるのに、プログラムステップが 1 次元状に順々に並んでいた訳ではなく、命令の実行時間を考慮して飛び飛びに配置し、プログラムの主メモリアクセス時間を最適化しています。初めて蓄積プログラム方式を勉強したとき、配線に代わり手順を主メモリに記憶するということは理解できました。

しかし、プログラム自身が自分を書き換えることはなかなか理解できませんでした。クイズのような話ですが、主メモリに記憶したプログラムを用いて、主メモリを全てクリア（ゼロにする）することが可能なのです。プログラムが書かれている部分も最後にはクリアされます。プログラムの部分を残し、他の全てをクリアするのはイメージが湧きますが、クリアするプログラム自身も最後にはゼロにならなければなりません。これを高級言語（例えば、FORTRAN）で書くことは不可能です。

これを可能にするのが、LGP-30 に用意された「Y 命令」というものでした。それは、自分のプログラムの中の特定の番地の命令語を他の命令語で置き換えることにより実行されます。これを実際に使ったとき、蓄積プログラム方式でもややもやとしていたことが、晴れたのを憶えています。

次のコラムへ

演習問題

1. 次の空欄 [a][b][c] に入る適切な語の組み合わせを、(1)-(4)の中から選びなさい。

オンラインリアルタイムシステムとタイムシェアリングシステムはいずれも [a] と端末を持つ点で共通している。しかし、前者は利用目的が [b] であり、ファイルを [c] 点が後者と異なる。

- (1) [a] ネットワーク [b] 専用的 [c] 共有しない
- (2) [a] ネットワーク [b] 専用的 [c] 共有する
- (3) [a] ネットワーク [b] 汎用的 [c] 共有する
- (4) [a] メインフレーム [b] 汎用的 [c] 共有する

2. 次の文章で間違っているものを選びなさい。

- (1) OS の重要な役割にハードウェア資源の管理がある。
- (2) スループットとは単位時間当たりの処理能力で、オーバヘッド処理を含む。
- (3) オープンショップ制というのはプログラマとオペレータが同じであり、現在のパソコンの利用形態に近い。
- (4) OS はハードウェアを動かすソフトウェアで、ハードウェアに最も近いソフトウェアである。

3. 次の空欄 [a][b][c] に入る適切な語の組み合わせを、(1)-(4)の中から選びなさい。

パソコンの OS（基本ソフトウェア）の 4 大機能は [a]・[b]・入出力管理機能・[c] である。

- (1) [a] タスク管理機能 [b] メモリ管理機能 [c] プロセス管理機能
- (2) [a] プロセス管理機能 [b] メモリ管理機能 [c] ファイル管理機能
- (3) [a] プロセス管理機能 [b] メモリ管理機能 [c] GUI 管理機能
- (4) [a] データ管理機能 [b] タスク管理機能 [c] ファイル管理機能

4. 次の文章で間違っているものを選びなさい。

- (1) Windows OS と Mac OS のシェアは前者が 90% を占める。しかし、Mac OS はアート系に強みがあり独特の特徴を持っている。
- (2) 組み込みシステム用には ITRON など独自の OS がある。
- (3) BIOS はパソコンの周辺機器を制御・管理するプログラムで原型は 1970 年代にゼロックス社で開発された。
- (4) 以前広く用いられていた「コマンドプロンプト」による CUI は、現在でも根強い人気がある。

5. 次の空欄 [a][b][c] に入る適切な語の組み合わせを、(1)-(4)の中から選びなさい。

OS のプロセス管理機能は最初 [a] からプロセスを生成し、競合しなければ [b] を割り当て [c] にする。

- (1)[a] ジョブステップ [b] 主メモリ [c] 実行状態
- (2)[a] ジョブステップ [b] CPU [c] 実行状態
- (3)[a] タスク [b] CPU [c] 実行状態
- (4)[a] ジョブステップ [b] CPU [c] 待ち状態

6. 次の文章で間違っているものを選びなさい。

- (1) マルチプログラミング技術は OS のプロセス管理による複数プロセスの並行処理によるものである。
- (2) 仮想メモリのページ置換アルゴリズムには、主として RLU が用いられている。
- (3) 複数のプロセスが同一のメモリ領域にアクセスするとき、先行プロセスがアクセスを終了するまで後続プロセスを待たせるのを排他制御と言う。
- (4) 仮想メモリはハードウェアを用いて大容量・高速アクセスメモリを仮想的に実現する。

7. 次の空欄 [a][b][c] に入る適切な語の組み合わせを、(1)-(4)の中から選びなさい。

ソフトウェアには無料で配布する [a]、[b] などのように試行期間は無料の [c] がある。

- (1) [a] オープンソースソフト [b] メールソフト [c] シェアウェア
- (2) [a] フリーウェア [b] アンチウィルス [c] オープンソースソフト
- (3) [a] オープンソースソフト [b] ワープロ [c] シェアウェア
- (4) [a] フリーウェア [b] アンチウィルス [c] シェアウェア

8. 次の文章で間違っているものを選びなさい。

- (1) フリーウェアは無料のソフトであるが、全てがソースコードを公開している訳ではない。
- (2) SPSS, SAS などは統計用パッケージで有料ソフトである。
- (3) Outlook Express はマイクロソフトが提供するブラウザで、Windows のユーティリティの一つである。
- (4) オープンソースソフトはソースコードを公開しているので、ユーザが手を加えることが可能である。

第 8 章 ソフトウェアとプログラミング

ほげほげ

- 8.1 ソフトウェアの位置づけ
- 8.2 アプリケーションプログラム
- 8.3 アルゴリズム
- 8.4 プログラム
- 8.5 プログラミング言語
- 8.6 インタプリタとコンパイラ

8.1 ソフトウェアの位置づけ

8.2 アプリケーションプログラム

8.3 アルゴリズム

8.4 プログラム

8.5 プログラミング言語

8.6 インタプリタとコンパイラ

第 9 章 ネットワーク技術とインターネット

ほげほげ

9.1 情報通信技術

9.2 OSI（基本）参照モデル

9.3 LAN と WAN

9.4 インターネット

9.5 TCP/IP

9.6 ルータとサーバの機能

9.1 情報通信技術

9.2 OSI（基本）参照モデル

9.3 LAN と WAN

9.4 インターネット

9.5 TCP/IP

9.6 ルータとサーバの機能

第 10 章 情報処理システム

いままで、ハードウェア・ソフトウェア・ハードウェアについて見てきました。これらの3種類の技術を用いてデータの収集、記録、加工、配布にかかわる一連の仕組みの総称を情報処理システムと呼びます。

10.1 システムの信頼性

10.2 システムの構成形態

10.3 システムの性能評価

10.4 システムの高信頼化

10.5 システムの処理形態 (1)

10.6 システムの処理形態 (2)

10.1 システムの信頼性

情報処理システム

情報処理 (information processing) システムとは、コンピュータの物理的構造と論理的な機構から構成されるコンピュータシステムに、ある業務を想定してそのための応用ソフトウェアを盛り込んだシステムのことを指します。

データの収集・記録・加工・配布に関わる一連の仕組みの総称を情報処理システムと呼びます。

図 1.19 に示すように、いままで見てきたソフトウェア、ハードウェア、ネットワークの集合体と考えてください。部品の集合ではなく、機能の集合体、もしくは技術の集合体と考えれば良いです。いずれを外しても正確に理解することはできません。

信頼性の尺度 (RASIS)

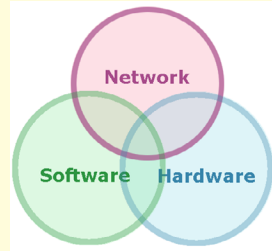


図 1.19 情報処理システム

コンピュータシステムが期待された機能・性能を維持しているかの検証のために RASIS という評価項目が知られています。故障や誤動作がいかに少ないかを表す指標が信頼性 (Reliability) です。システムの稼働率 (ユーザがどのくらいコンピュータを利用できるか) を示す指標を可用性 (Availability) と呼びます。システムが故障したとき、どれだけ早く、かつ容易に修理できるかを表す指標が、保守性 (Serviceability) です。システムの柔軟性 (データの破壊・喪失等

表 1.13 信頼性の尺度 (RASIS)

信頼性の尺度	説明
信頼性 (Reliability)	故障や誤動作がいかに少ないかを表す指標
可用性 (Availability)	システムの稼働率 (ユーザがどのくらいコンピュータを利用できるか) を示す指標
保守性 (Serviceability)	システムが故障したとき、どれだけ早く、かつ容易に修理できるかを表す指標
完全 (保全) 性 (Integrity)	システムの柔軟性 (データの破壊・喪失等への強度) を表す指標
安全 (機密) 性 (Security)	不当・不正な利用に対する強度を表す指標

故障



図 1.20 システムの稼働時間

への強度) を表す指標が完全 (保全) 性 (Integrity) であり、不当・不正な利用に対する強度を表す指標を安全 (機密) 性 (Security) と呼びます。表 10.1 にこれらの 5 つの評価項目をまとめています。RASIS のうち、RAS (信頼性、可用性、保守性) は数値化することができ、客観的に捉えやすいという特徴があります。

信頼性を示す指標として、隣接した故障と故障との間の時間の平均値を平均故障間隔 (MTBF: Mean Time Between Failures) と呼びます。言い換えると連続稼働できる時間の平均値です。つまり、MTBF の数値が大きいほど信頼性の高いシステムであるといえます。

また、保守性を示す指標として、故障からの復旧までの時間の平均値を平均修理時間 (MTTR: Mean Time To Repair) と呼びます。

可用性を示す指標として、稼働率があります。前述の平均故障間隔 (MTBF) と平均修理時間 (MTTR) をつかって下記の式で稼働率を求めることができます。

$$\text{稼働率} = \text{MTBF} / (\text{MTBF} + \text{MTTR})$$

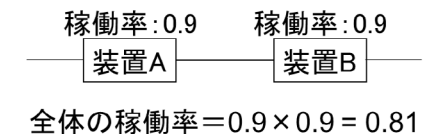
稼働率が高いほど、そのシステ

ムは正常に動作するといえます。このように、MTBF が大きく MTTR が小さいシステムほど可用性が高く、総合的な信頼性が高いシステムであるといえます。

個々の装置が直列に並んだ直列システムでは、装置が 1 つでも故障するとシステム全体がダウンしてしまうので、システム全体としての稼働率はそれぞれの装置の稼働率の積で求めることができます。

また、並列システムでは、2 つの装置の両方が故障したときのみシステムがダウンします。よって並列システム全体の稼働率は両方の装置が非稼働となる確率を求め、その確率を 1 から引くことで計算できます。

■直列システム



■並列システム

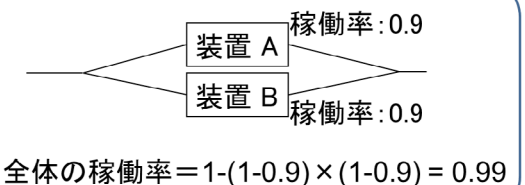


図 1.21 システムの稼働率の計算

10.2 システムの構成形態

コンピュータシステムの構成の主な形態を学びます。

シンプレックスシステム

シンプレックスシステムは、機器やデータベースなどを含めて、1系列のシステムだけで構成されるシステムです。最も単純で安価なシステム構成だが、システムを構成している要素の1つでも故障するとシステム全体がダウンしてしまうため、信頼性はあまり高くありません。

デュプレックスシステム

デュプレックスシステムは、オンラインの業務を処理する動作系システムと、平常時にはオフラインの緊急性の低い再実行可能な業務を処理する待機系システムの2系列のシステムで構成されるシステムです。動作系に何らかの障害が発生した場合、待機系に切り替えてオフライン業務を再開することで円滑な運用を確保する仕組みです。通常は、待機系をバッチ処理などに利用することで経済的なシステム構成が実現できます。ただし、障害発生の際の切り替え時には、動作系の状態を待機系に引き継がせる必要があるなど、複雑な操作が要求されます。

デュアルシステム

デュアルシステムは、業務を完全に遂行できるシステムを二重に装備したシス

テムです。それぞれの系統が常時同一の処理を実行し、処理した結果を周期的に相互チェックしながら運用します。障害を検知した場合には、障害を起こした系統を瞬時に切り離して正常なシステムのみで運転を継続することができます。

マルチプロセッサシステム

マルチプロセッサシステム（多重プロセッサシステム）は、複数のCPUが主記憶装置や外部記憶装置を共有し、処理を並行して行うシステムです。CPUに故障が発生した場合、これを切り離して処理を続行することができます。

タンデムシステム

タンデムシステムは、複数のCPUを直列に配置し、主記憶装置を共有するシステムです。主となるCPUの前で通信制御などを行うCPUであるFEP (Front-End Processor) と、外部記憶装置の制御を行うCPUであるBEP (Back-End Processor) を備えています。タンデムとは「タテ並びの馬」という意味でCPUが直列に並んでいることを表しています。

ロードシェアシステム

ロードシェアシステムは、複数のCPUの間に共用ディスクを配置し、負荷を分散させることで、システム全体の

処理能率の向上を図るシステムです。構成的にはデュプレックスシステムと同じく複数の系統を持ちます。電子交換機などにより入力データを各処理系に振り分け、システム全体に対する負荷をバ

ランスよく配分し、負荷の状況に応じてCPU相互間で負荷分担を行います。1台のCPUがダウンしてしまった場合は、他のCPUで処理を代替することができます。

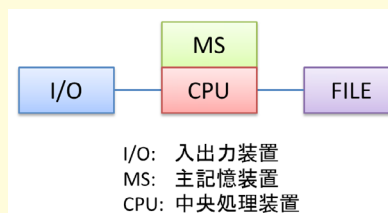


図 1.22 シンプレックスシステム

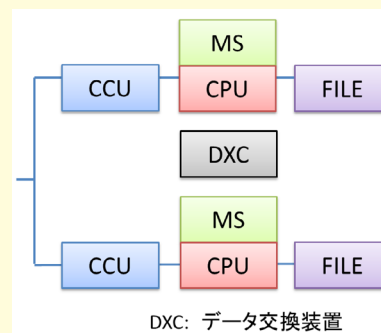


図 1.24 デュアルシステム

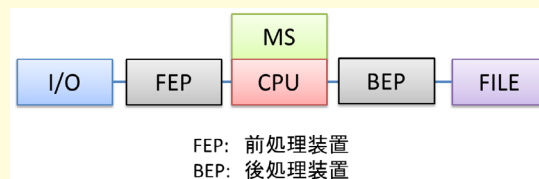


図 1.26 タンデムシステム

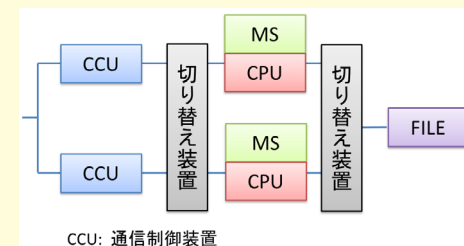


図 1.23 デュプレックスシステム

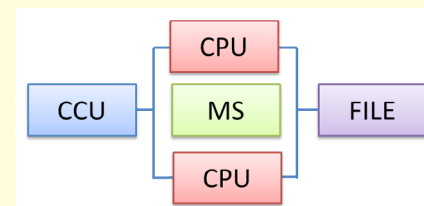


図 1.25 マルチプロセッサシステム

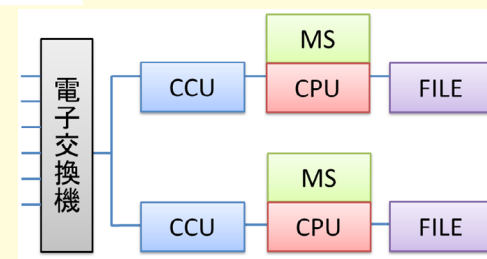


図 1.27 ロードシェアシステム

10.3 システムの性能評価

本節では、コンピュータシステムの性能評価について学びます。

プロセッサの性能指標

プロセッサの性能評価の1つの指標として、MIPS(Million Instructions Per Second)が知られています。MIPSは、1秒間に何百万命令を実行できるかを示す指標です。例えば、1秒間に700万命令実行できるプロセッサは、7MIPSの性能であるといえます。ただし、MIPS値は、命令の種類が定義されていないので、異なるコンピュータアーキテクチャの比較材料とするのは難しいといえます。さらに、FLOPS(Floating-point Operations Per Second)という指標もあります。これは、1秒間に実行できる浮動小数点命令の回数を表します。例えば、1秒間に700万回の浮動小数点演算を実行できるプロセッサは、7MFLOPS(メガフロップス)の能力を持っているといえます。FLOPSは、大型コンピュータやスーパーコンピュータの性能指標に使われます。独立行政法人理化学研究所は、2006年に1PFLOPS(ペタフロップス)のスーパーコンピュータを開発しました。また、2011年には、独立行政法人理化学研究所(理研)と富士通株式会社は、スーパーコンピュータ「京」が、10PFLOPS(毎秒1.051京回=10,510兆回の浮動小数点演算数)を実現したと

発表しました。

ベンチマークテスト

プロセッサやシステムの性能評価を行う際に、いくつかの標準的なテスト用のプログラムを用意しておき、さまざまな状況で実行して実際の処理時間を測定することにより、システムの性能を計ります。これをベンチマークテストと呼びます。ベンチマークテストは、単純にMIPS値やクロック数の比較だけでは難しい、異機種間の性能比較に適しています。ベンチマークテストの代表例に以下のようなものがあります。

・SPECint

SPEC(Standard Performance Evaluation Corporation)という団体が作成した、整数演算の性能を測定するためのベンチマークテスト

・SPECfp

SPECが作成した浮動小数点演算の性能を測定するためのベンチマークテスト

・TPC

TPC(Transaction Processing Performance Council)という団体が作成した、トランザクション処理の性能を測定するためのベンチマークテスト

システムの性能

●スループット(throughput)

単位時間あたりにコンピュータシステ

ムが処理できる仕事量をスループット(throughput)と呼びます。スループットはシステムの総合的な性能を表わします。色々な要因が関係するので、この値だけでシステムの性能を比較することはできませんが、スループットを最大にすることはシステム設計における大きな目標の一つです。

●応答時間(response time)

要求の終わり(入力の完了)から応答の始まり(画面などへの出力の開始)までの時間を応答時間(response time)といいます。オンライントランザクション処理システム、リアルタイム制御処理システム、タイムシェアリング処理システムの性能評価の指標に用いられます。例えば、コールセンタなどにおける電話注文登録処理など、コンピュータからの速い応答が求められるシステムは応答時間が短いことが要求されます。

●ターンアラウンドタイム(turnaround time)

ジョブを提出してから、完全な出力結果がユーザのもとへ返されるまでの時間が、ターンアラウンドタイム(turnaround time)です。バッチ処理システムの性能評価の指標に用いられます。例えば、月次の売上集計にかかる時間など、とにかく処理が早く完全に終わることが求められる処理では、短いターンアラウンドタイムが要求されます。

ワンポイント
オーバーヘッド(Overhead)とは、コンピュータで1つの作業を実行しようとするときに、応用プログラムの本来の処理以外のOSが必要とする時間(もしくは容量)のことです。コンピュータ利用者のプログラムに直接関係のない処理のことを指します。一般的には、無駄や時間待ち合わせなどによって、計算上よりも時間や容量が余分にかかることを言います。OSが高度な処理をするようになると、データの退避やリカバリ、OS稼働時のプログラムモジュールのダイナミックリンクなど、ユーザから見えない部分での仕事が多くなりオーバーヘッドは増加してしまいます。GUIなどのグラフィック処理では、オーバーヘッドが大きくなる傾向があります。

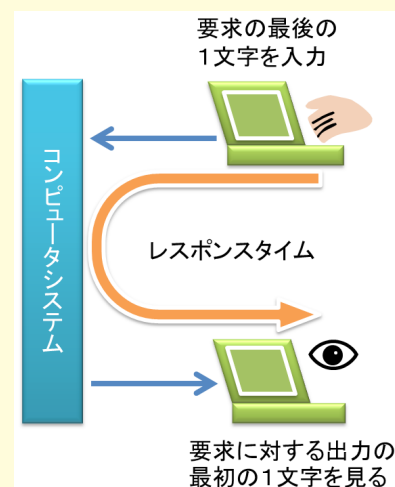


図 1.28 レスポンスタイム

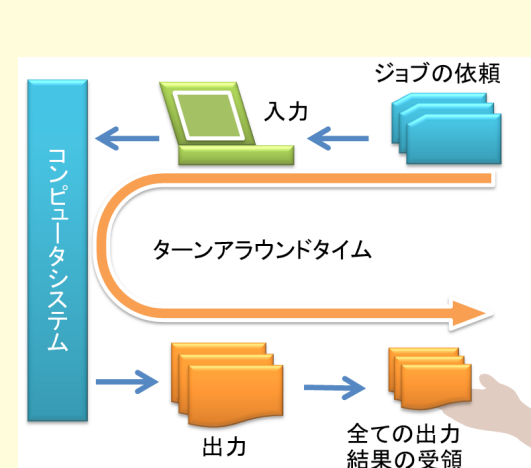


図 1.29 ターンアラウンドタイム

10.4 システムの高信頼化

本節では、コンピュータシステムの高信頼化の方法について学びます。

高信頼化

システムの信頼性を高めるための主な考え方にフルプルーフ、フェイルソフト、フェイルセーフ、フォールトトレラントなどがあります。

●フルプルーフ

不注意による操作誤りを起こさないように、確認手順などを配慮した設計をするという考え方をフルプルーフといいます。意図しない使われ方をしても、故障しないように設計することを指します。「フルプルーフ」とは「間違えようのない、極めて単純な」という意味です。

●フェイルソフト

ある装置にトラブルが発生したとき、そのトラブルが原因でシステム全体がダウンしないように、トラブルの影響範囲を最低限にするという考え方をフェイルソフトといいます。トラブルが発生した場合には、システム全体の構成から該当箇所を切り離して、性能もしくは機能が低下しても残りの正常な部分だけで処理を実行させ、サービス自体は継続させます。システムの一部に障害が発生したときに、多少の機能低下は避けられないとしても、その障害が原因となる二次的障害の影響ができるだけ小さくなるように

設計することを指します。

●フェイルセーフ

ある装置にトラブルが発生しても、予備の機器に切り替えて、システムの性能を低下させずにリカバリーさせるといふ考え方をフェイルセーフといいます。フェイルソフトより一歩進んだ安全性を目指しています。具体的には、システムを二重化、三重化してシステムの信頼性を増すという方法を取ります。これにより装置の一方が故障しても機能を低下させずに残りの装置でシステムを運転し続けられるように設計するします。「フェイルセーフ」とは「安全側に作動する、安全装置のついた」という意味です。

●フォールトトレラント

システム構成内にトラブルが発生しても、常にシステムを稼働状態に維持させるといふ考え方をフォールトトレラントといいます。具体的には、すべての機器を完全二重化するなどして、フェイルソフトやフェイルセーフよりもさらなる信頼性の向上を目指します。広義には、フェイルソフトやフェイルセーフの考え方も含まれます。

システム全体の信頼性を向上させるために、ある機能を実現するのに必要な最小限の装置のほかに、予備の装置も用意し、現用の装置に障害が発生した場合に、予備装置で代替できるようにシステムを構成することを、冗長構成といいます。

フォールトトレラントの考え方の具体的な実現方式です。

冗長方式には、二重化方式と N+1 冗長方式の 2 種類があり、また冗長構成されたシステムの運転方式としては、代表的なものにスタンバイ方式と同期運転方式があります

●二重化方式

同一の装置を 2 台設置し 1 台が障害になった場合にも、正常な装置で運転を続行する方式です。この方式は、信頼性は高くなりますが、コストが高くなる傾向があります。

●N+1 冗長方式

N 台の装置に対して 1 台の装置を共通の予備として設置する方式。N 台中 1 台までの障害に対しては有効ですが、2 台以上が同時に壊れる二重障害に対しては十分ではありません。ただし単純な 2 重化（多重化）方式より装置のコストは低く抑えられます。

●スタンバイ方式

ある装置を現用装置の予備として待機

させる方式です。待機状態によって、ホットスタンバイ、コールド・スタンバイなどの方式があります。ホットスタンバイは、片方（主系・本番系）を作動させ、もう片方（待機系・予備系）は同じ動作を行いながら待機状態にしておく方式です。待機系は主系と常に同じ状態を保っておき、主系に障害が発生すると即座に待機系に処理が引き継がれます。主系と待機系の同期を行わずに、主系に障害が発生してから待機系を作動させる方式をコールドスタンバイと呼びます。ホットスタンバイはコールドスタンバイに比べて主系と待機系の同期に余計なコストがかかりますが、その分信頼性が高くなります。

●同期運転方式

片方（待機系・予備系）をもう片方（主系・本番系）と同期して並列運転する方式です。本番系と予備系の処理結果を常時照合することで、障害の発見や迅速な対処が可能な方式です。



身近な所でも冗長性を持った機器が使われています。複数の HDD を使って「冗長性」を確保する事を目的とした仕組みが RAID です。RAID には、冗長性と共に容量の小さな HDD をまとめて一つの大きな HDD として扱えるようにする目的があります。現在よく使われているのは、RAID1 と RAID5 と呼ばれる方式です。

RAID1 は、2 台以上の HDD を組み合わせて同一データを複数の HDD に書き込み、HDD のコピーを用意するというシンプルな方法で耐障害性を高めたものです。上で説明した「二重化方式」に相当します。1 台の HDD に障害が発生しても、他の HDD でデータを処理できるため動作不能に陥ることはありません。ただし、同じ内容が複数台の HDD に保存される為、データ保存領域として利用できる容量は、構成する HDD の合計容量の 1/2 以下となります。

RAID5 は、データは各 HDD に分割して書き込まれるのに加えて、同時に「パリティ」と呼ばれる冗長コードが生成されて書き込まれる方式です。これにより、構成 HDD のうち 1 台に障害が発生しても、残りのデータとパリティを元に欠損したデータを算出でき、完全な状態のデータを生成することができます。パリティは各ドライブに分散して書き込まれるため、特定のパリティドライブに対する負荷が減り高速化が図れます。（例えば 3 台構成の RAID5 アレイでどの HDD に障害が発生しても、残りの 2 台構成の状態でも稼働可能です。欠損したデータは毎回演算によって算出されます。）最低 3 台の HDD が必要で、構成 HDD のうち 2 台以上に障害が発生した場合には動作不能となります。

10.5 システムの処理形態 (1)

本節と次節ではコンピュータシステム
の処理形態について学びます。

バッチ処理

バッチ処理は、処理すべきデータを蓄積しておいて、一定の時間に一括して処理を行う方式です。一括処理とも言います。コンピュータに指示を与えると、後はコンピュータが自動的に複数のジョブを処理します。処理内容が定型的であり、その処理に関して人的介入の必要がないような場合に有効です。ジョブを処理する順序をうまく調整したり、処理装置が使用されていない時間を有効に活用することができます。このように、バッチ処理は、コンピュータ資源の有効利用、人的操作ミスの排除、スループットの向上などの効果があります。日・週・月・年ごとなど定期的にデータの集計をする場合に有効である。具体例としては、学力

試験の結果集計、企業の給与計算、水道・電気・ガス・電話代・クレジット利用料金の精算・処理請求、売上集計処理などがバッチ処理に向いています。

対話型処理

対話型処理は、人間とコンピュータが対話形式で処理を進めていく方式です。バッチ処理が、ハードウェア資源が有効利用される処理方式であるのに対して、対話型処理は、主に人的資源（ヒューマンリソース）が有効利用される方式です。また、バッチ処理が、処理のアルゴリズムが明確でない場合に処理の中断などが発生するのにに対して、対話型処理は、処理途中で人的介入が可能なので、処理のアルゴリズムが 100% 確定していなくても、訂正しながら処理を進めることができます。具体例としては、Windows の操作、文書作成時のかな漢字変換など



対話型処理の生産性は、ユーザインタフェースと応答時間によって左右されます。対話型処理システムにおいては、利用者とコンピュータとの会話をとりもつユーザインタフェースの良し悪しが重要です。ユーザインタフェースが悪いと、コンピュータの処理時間が短くても、人間の処理時間がボトルネック（足を引っ張る要素）になってしまいます。人間の判断をできるだけ容易にするように、また入力動作を簡単に便利にするように、よりよい GUI などを提供する必要があります。また、対話型処理では、コンピュータシステムに対する要求の終わりから応答の始まりまでの時間（応答時間）が短くなければなりません。また、対話型処理の OS には、1 人のユーザが、1 つのプログラムを使って、1 つの仕事を行うシングルユーザ・シングルタスク（例：MS - DOS）と、1 人のユーザが、複数のプログラムを使って、複数の仕事を行うシングルユーザ・マルチタスク（例：Windows）、さらに、複数のユーザが、複数のコンピュータやプログラムを使って、複数の仕事を行うマルチユーザ・マルチタスク（例：UNIX）に分類できます。

を挙げることができます。

オンライントランザクション処理

オンライントランザクション処理（OLTP：On-Line Transaction Processing）は、トランザクションが発生するたびに、オンラインで一定の時間内にそのトランザクションに対する処理結果を返す方式です。オンライン処理とデータベースシステムが融合した処理方式となります。ここで、トランザクションとは、アプリケーションから見た意味のある一連の処理のまとまりのこと（一般的には、処理・取引という意味）を指します。例えば、「銀行口座 A から別口座 B に n 円を送金する」場合、「A と B の両口座を変更すること」が意味のある

処理のまとまり（トランザクション）となります。

OLTP システムの具体例としては、飛行機や列車の座席予約システム、JR の「みどりの窓口」の発券システム、金融業務システム、銀行における預金オンラインシステム（ATM: 自動現金預け払い機）、保険業界の契約管理システムや保険総合口座システムなどが挙げられます。

OLTP システムは、データベースと密接に関係しているため、時間が経過し、データ量が増えていくにつれ、システムが肥大化していく傾向があります。即時性と信頼性にある程度の厳しい要求があり、ハードウェア・ソフトウェアともに高価なものになります。

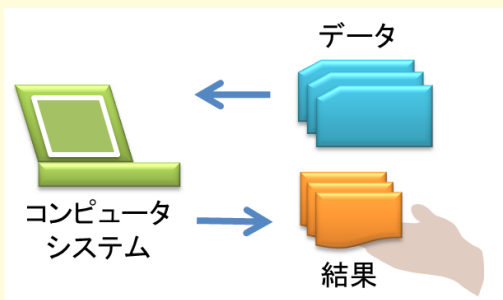


図 1.30 バッチ処理

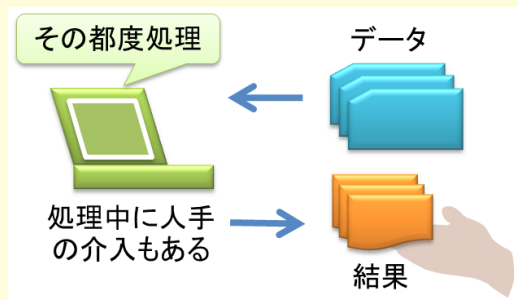


図 1.31 対話型処理

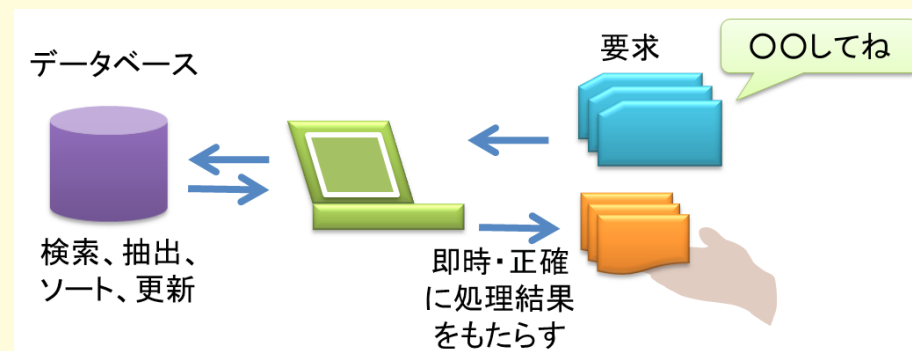


図 1.32 オンライントランザクション処理

10.6 システムの処理形態 (2)

トランザクションの特性

トランザクションが正常に動作するために、持っていなければならない特性として、Atomicity（原子性）、Consistency（一貫性）、Isolation（独立性）、Durability（耐久性）の頭文字を取って ACID 特性といいます。

● Atomicity（原子性）

トランザクションは「正常に全て終了する」か「何も行われていない」かのどちらかでなければならないという特性を原子性といいます。トランザクションは回復の単位であるともいえます。

● Consistency（一貫性）

トランザクションの処理によって、矛盾が発生しない状態にしなければなりません。データに矛盾がなく一貫していることを一貫性といいます。

● Isolation（独立性）

複数のトランザクションを同時に実行しても、お互いに干渉せず、直列動作性を保たなければなりません。この特性を独立性といいます。隔離性や分離性とも呼ばれます。

● Durability（耐久性）

トランザクションが正常に完了した後に障害が発生してもトランザクションの状態は維持されなければなりません。この特性を耐久性といいます。また、永続性とも呼ばれます。

命令に対して即時に処理結果を出す方式をリアルタイム制御処理といいます。刻々と変化する対象の状態を常に監視し、制御する目的で用いられます。データのやり取りがオンラインで行われる場合は、オンラインリアルタイム処理とも呼ばれます。オンライントランザクション処理よりも即時性が要求され、機器の制御に観点を移す場合に使われます。オンライントランザクション処理は即時処理方式という意味で、リアルタイム処理ではありますが、「トランザクション」に対するリアルタイム処理と捉えてください。

リアルタイム制御処理

リアルタイム制御処理の具体例としては、航空管制システム、列車運行制御システム、化学プラントの制御システム、原子力発電所の制御システム、自動車の燃料系の電子制御システムなどが挙げられます。また、マイコン内蔵と言われる家電製品（エアコンや電子炊飯器など）のような、小規模のコンピュータ（マイコン）によって制御されるものを含む場合もあります。

複数の端末（ターミナル）を1台のホストコンピュータ（大きな処理能力のある汎用コンピュータ）に接続して集中的にデータ処理を行う、ホスト・ターミナル型のコンピュータシステムに見られる

処理形態を集中処理システムといいます。TSS（タイムシェアリングシステム）は集中処理の1つです。

リアルタイム制御処理

データ処理を1台のホストコンピュータやサーバに集中させるのではなく、

ネットワークで接続された複数のコンピュータで処理する形態を分散処理システムといいます。C/Sシステム（クライアントサーバシステム）が分散処理の代表例です。図10.6に集中処理と分散処理のシステム構成と表10.6に比較を示します。

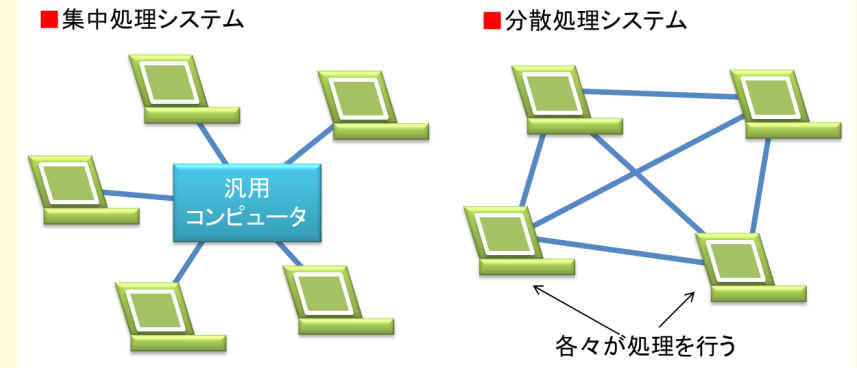


図 1.33 集中処理と分散処理

表 1.14 集中処理と分散処理の比較

	集中処理	分散処理
長所	● データの一貫性が保てる ● 資源の一括管理が可能 ● システム全体が把握しやすい ● メンテナンスが容易 ● 安全対策を集中して実行可能	● 一部のシステムダウンがシステム全体に波及しない ● 上位コンピュータへの通信が減るので通信コストが低減する ● 分散して入力や処理が行われるので上位コンピュータの負荷が軽減される ● 下位コンピュータでエラーチェックなどの処理が行われるため、上位コンピュータにエラーデータなどが蓄積されにくくなる
短所	× ハードウェア、ソフトウェアの追加、変更に制約が発生する × 集中管理のためシステムダウンによる影響が大きい	× システム構築が複雑になり、構築者には高度なスキルが要求される。特に異機種混合システムの場合は、問題が起きやすくなる。 × システム全体の保守、運用コストが増大する × セキュリティに関して、下位コンピュータになればなるほど低くなる

演習問題

1. コンピュータシステムの応答時間の定義として、適切なものはどれか。

- (a) コンピュータシステムに対する問合せの終わりを指示してから応答が完全に終わるまでの経過時間
- (b) コンピュータシステムに対する問合せの終わりを指示してから応答が始まるまでの経過時間
- (c) コンピュータシステムに対する問合せの始まりを指示してから応答が完全に終わるまでの経過時間
- (d) コンピュータシステムに対する問合せの始まりを指示してから応答が始まるまでの経過時間

2. あるジョブのターンアラウンドタイムを解析したところ、1,350 秒のうち CPU 時間が 2/3 であり、残りは入出力時間であった。1 年後に、CPU 時間はデータ量の増加を考慮しても、性能改善によって当年比 80% に、入出力時間はデータ量の増加によって当年比 120% になることが予想されるとき、このジョブのターンアラウンドタイムは何秒になるか。ここで、待ち時間、オーバヘッドなどは無視する。

A) 1,095 B) 1,260 C) 1,500 D) 1,665

3. システムの構成に関する次の記述に対応するシステム構成名を (a) ~ (f) から選べ。

- (1) 一方の中央処理装置がオンライン処理、もう一方の中央処理装置がオフライン処理を行っており、オンライン系のシステムに故障が発生したときにオフライン系のシステムに切り替えて利用する。
- (2) 端末装置を専用に制御する処理装置と、ファイルを専用に制御する処理装置がメインの中央処理装置に直結され、それらが一体となって処理を行う。
- (3) 処理に必要な最小限の装置が装備され、単一システムとも呼ばれるシステム構成である。
- (4) 複数の処理装置が主記憶装置を共有し、いくつかの処理を分担して並列処理を行うことができる。
- (5) 各処理装置を二重化させ、信頼性を高めるために一定時間ごとにクロスチェックを行う。
- (6) 複数の系統の処理系をもち、システム全体に対する負荷をそれぞれの処理系にバランスよく配分することで、処理効率の向上を図ったシステム構成である。

- (a) シンプレックスシステム
- (b) デュプレックスシステム
- (c) デュアルシステム
- (d) マルチプロセッサシステム
- (e) タンデムシステム (f) ロードシェアシステム

4. 次の記述 a ~ d に対応する、システムの信頼性を高めるための考え方の名称を (a) ~ (d) から選択せよ。

- (1) すべての機器を完全二重化するなどして、システム内で故障が発生しても、常にシステムを正常な稼動状態を維持できるように設計すること
- (2) 意図しない使われ方をしても、故障しないように設計すること
- (3) システムの一部に障害が発生したときに、多少の機能低下は避けられないとしても、その障害が原因となる二次的障害の影響ができるだけ小さくなるように設計すること
- (4) 装置を二重化しておき、一方が故障しても機能を低下させずに残りの装置でシステムを運転し続けられるように設計すること

- (a) フールプルーフ (b) フェイルソフト
- (c) フェイルセーフ (d) フォールトトレラント

5. OLTP におけるアプリケーションに要求される 4 つの特性は、それぞれの頭文字をとって ACID 特性と呼ばれている。この ACID 特性に関する説明のうち、適切なものはどれか。

- (a) トランザクション処理終了後は、その内容は失われることはないという特性で、一貫性 (Consistency) という。
- (b) トランザクションの処理によって、システム内の関係するすべてについて矛盾を生じないという特性で、耐久性 (Durability) という。
- (c) 一つのトランザクションの途中でトラブルが起これども、正常に処理された部分は生かされるという特性で、原子性 (Atomicity) という。
- (d) 複数のトランザクションが並行して処理されていても、互いに干渉されことなく順序を保って実行されるという特性で、独立性 (Isolation) という。

第 11 章 情報セキュリティの基礎

コンピュータの周辺技術として、情報セキュリティ技術を取り上げます。情報セキュリティの基礎としての暗号技術、認証、署名、電子商取引に應用する公開鍵基盤 (Public Key Infrastructure: PKI) および PKI に必須の認証局について述べます。

11.1 暗号の歴史とアルゴリズムの分類

11.2 秘密鍵暗号系と公開鍵暗号系

11.3 秘密鍵暗号

11.4 公開鍵暗号

11.5 デジタル署名と認証

11.6 公開鍵証明書と認証局

11.1 暗号の歴史とアルゴリズムの分類

本節では暗号の歴史を見ます。また、暗号方式を分類し、現代暗号と言われている商用向き暗号の特徴を学びます。

秘匿と認証

暗号の機能には情報の秘匿と認証があります。情報の秘匿は情報が露呈しても（鍵がなければ）情報の意味が分からないようにし、盗聴を防ぎ機密を守ることです。

一方、認証とはメッセージの改ざん、送信者の成りすまし、メッセージを受取っているにも拘らず否認、といったことを防ぐことです。例えば、エンティティ認証ではエンティティ A がエンティティ B に安全に情報を伝えようとしたとします。このとき、A が B に自分が A であることを認めさせることはできますが、A とは違うエンティティ C が A に成り

すまし、B に自分は A であることを認めさせることが出来ないための技術です。このような署名の技術は契約・送金などビジネス上では欠かせないものです。

暗号の歴史

暗号の歴史を振り返ると、主に外交や軍事用の用途で用いられてきた近代暗号以前と、商用に利用されるようになった現代暗号に大別されます。近代暗号以前の暗号ではアルゴリズムや鍵は非公開でした。それに対して現代暗号では鍵だけ非公開（一部の鍵は公開）にしてアルゴリズムは公開されています。暗号の歴史を表 1.15 と表 1.16 に示します。

暗号アルゴリズムの分類

暗号は大別して (1) 秘密鍵暗号（共通鍵暗号・慣用暗号・対称暗号とも言う）

表 1.15 暗号の歴史（古代～中世）

古代				中世	
BC19世紀	BC5世紀	BC2世紀	BC1世紀	16世紀	19世紀
[エジプト時代] ヒエログリフ	[バビロニア] アトバッシュ	[ギリシャ] ポリュビオス	[ローマ] シーザー	[日本] 上杉謙信	[アメリカ] ビール
	[スパルタ] スキュタレー			[ドイツ] トリデミウス	
				[フランス] ビジネル	
				ルイ14世	

表 1.16 暗号の歴史（近代～現代）

近代暗号		現代暗号	
1900	1950	2000	
[ドイツ] (1917) ツインマーマン (1918) ADFGX	[日本] (1937) パープル (1926) エニグマ	(1976) DH(Diffie-Hellman)公開鍵暗号 (1977) DES(Data Encryption Standard) (1977) RSA (Rivest-Shamir-Adleman) (2000) AES (Advanced Encryption Standard)	

と (2) 公開鍵暗号（非対称暗号とも言う）があります。

秘密鍵暗号には、例えばシーザー暗号・スキュタレー暗号・エニグマ暗号・パープル暗号（九七式暗号）・DES(Data Encryption Standard)・AES(Advanced Encryption Standard) などがあります。暗号化と復号に同じ鍵を用いるのが特徴です。

一方、公開鍵暗号には、例えば RSA(Rivest-Shamir-Adelman) 暗号・ElGamal

暗号・McEliece 暗号などがあります。暗号化と復号には異なる鍵を用い、暗号化鍵を公開するのが特徴です。

表 1.17 暗号アルゴリズムの分類

アルゴリズム	鍵	暗号の種類	暗号の名前(例)
秘匿	一部秘匿	ストリーム暗号	NSFR
		ブロック暗号	MULTITX
	公開鍵 (一部公開)	ストリーム暗号	バーナム暗号, RC4
		ブロック暗号	DES, FEAL, MISTY, AES
公開	公開鍵 (一部公開)	メッセージ復元	RSA, ElGamal, Cramer-Shoup
		メッセージ非復元 (主としてデジタル署名用)	DH, PSS, DSA



写真 1.32 エニグマ暗号機
出典: フリー百科事典『ウィキペディア (Wikipedia)』

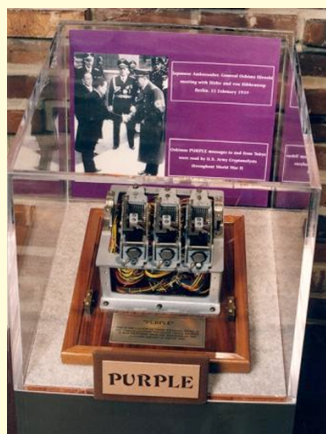


写真 1.33 パープル暗号機
出典: フリー百科事典『ウィキペディア (Wikipedia)』

11.2 秘密鍵暗号系と公開鍵暗号系

秘密鍵暗号系

図 1.35 に秘密鍵暗号系のモデルを示します。秘密鍵暗号系では、暗号化鍵と復号鍵を同じ鍵 K を使います。次式は暗号化と復号の計算式を示しています。

$$\text{暗号化 } C: c = C_K(m)$$

$$\text{復号 } D: \hat{m} = D_K(c)$$

次式はある文 m を暗号化し復号すれば、元の文 m に戻ること示しています。

$$m = D_K(C_K(m))$$

公開鍵暗号系

公開鍵暗号では暗号化鍵 K_E と復号鍵 K_D を一対ずつ ($K_E \neq K_D$) 作り、暗号化鍵を公開し復号鍵を秘密にします。送信者 A は公開リスト（電話帳のようなもの）上の受信者 B の暗号化鍵 K_E を用いて平文から暗号文を作成し、受信者 B は自分だけが知っている復号鍵 K_D を用いて暗号文を平文にします。ここで、暗号文を作った A でさえ、暗号文から平文にすることが出来ないことに注意して下さい。

$$\text{暗号化 } C: c = C_{K_E}(m)$$

$$\text{復号 } D: \hat{m} = D_{K_D}(c)$$

$$m = D_{K_D}(C_{K_E}(m))$$

公開鍵暗号では、秘密鍵暗号のようにあらかじめ鍵を配送する必要はありません。また、 N 人で相互に通信するとき安全に管理しなければならない鍵の数は N でよく、秘密鍵暗号ではその数が $N(N-1)/2$ となることから有利となります。

しかし、公開鍵暗号のアルゴリズムは通常整数のべき乗演算と剰余演算を実行しなければならず、計算速度が落ちるといふ欠点があります。なお、公開鍵暗号の安全性は、暗号化鍵 K_E から復号鍵 K_D が容易に求まらないという仕組みにあります。そのために方向性関数が用いられます。

- 暗号化鍵(K_E) ≠ 復号鍵(K_D)

K_E : 公開

K_D : 秘密



- 一方向性関数: $y = f(x)$

$x \rightarrow y$ 簡単

$y \rightarrow x$ 困難

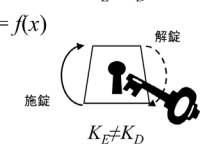


図 1.34 鍵と一方向性関数

盗聴と改ざん

さて、前に述べた通り暗号の役割には秘匿（盗聴の防止）と認証（改ざん・成りすまし・否認の防止）があります。ここで、盗聴とは不正な第三者により暗号文が解読されること、改ざんとは不正な第三者により暗号文を変更すること、成

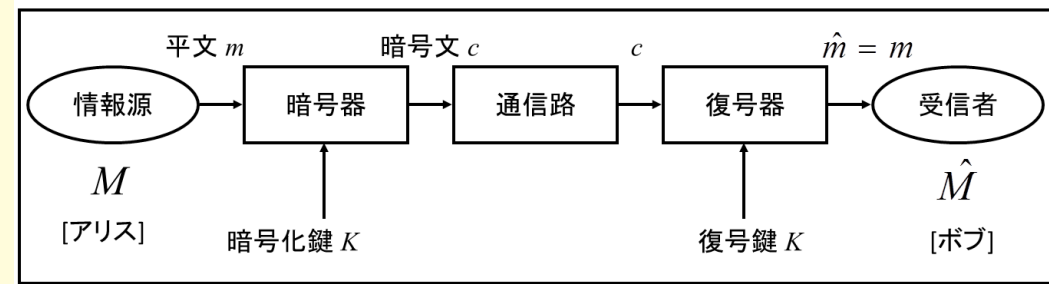


図 1.35 秘密鍵暗号系

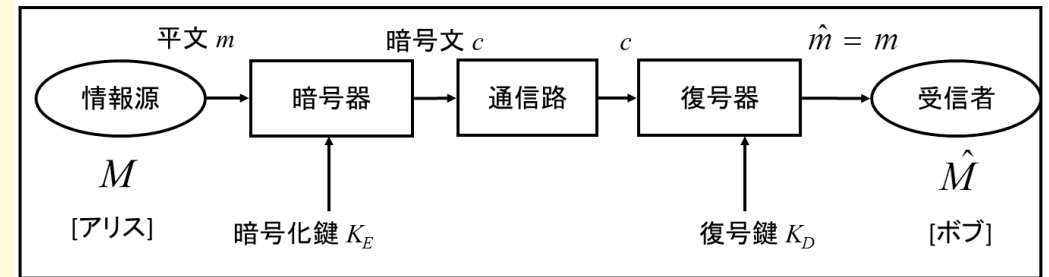
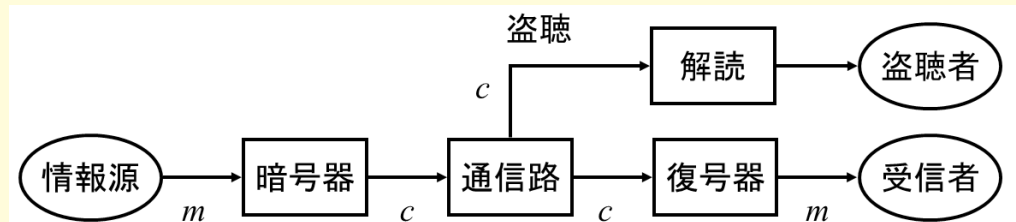
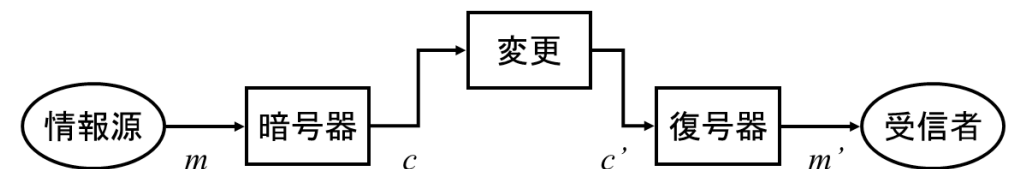


図 1.36 公開鍵暗号系

りすましとは不正な第三者が正当な送信者と主張すること、否認とは送信者（受信者）が平文を送信（受信）していないと主張すること、を言います。このように不正は商取引上困ります。図 1.37 に盗聴と改ざんの概念を示します。



(a) 盗聴



(b) 改ざん

図 1.37 盗聴と改ざん

11.3 秘密鍵暗号

秘密鍵暗号のアルゴリズムには、大別して (a) 換字法と (b) 転字法があります。前者は平文の各文字を他の文字に置き換える方法で、シーザー暗号はこれに相当します。一方、後者は平文の各文字の順序を入れ替える方法で、例えばスキュタレー暗号があります。

換字法と転字法

図 1.38 に換字法の例を示します。アルファベットの小文字を別のアルファベットの大文字に変換する変換テーブルが鍵 K になります。復号には逆変換表を用いて変換します。

また、図 1.39 に転字法の例を示し

す。転字法は位置を置き換える方法です。図 1.39 の例では 1 番目の文字を 4 番目に、2 番目の文字を 3 番目に、というように置き換えていきます。

DES と AES

DES (Data Encryption Standard) は、換字と転字を繰り返す方法です。DES のアルゴリズムを図 1.40 に示します。DES は、アルゴリズムが簡単で高速化および小型化が可能という特徴があります。より安全性を高めるために、DES を三重に適用するようにした Triple DES(3DES) 方式があります。また、2000 年には、アメリカ合衆国の新暗号

規格 (Advanced Encryption Standard) として AES 暗号が規格化されました。

表 1.18 に DES と AES の安全性の比較を示します。表 11.3 に DES と AES の安全性の比較を示します。解読するまでに要する時間を総当たりによる解読時間としています。解読にかかる平均総時間 T は、 $T=(1/2)|K|t$ [秒] で求められます。ここで、 t は鍵 1 つあたりの解読時間、 $|K|$ は鍵集合の個数です。

表 1.18 解読にかかる平均総時間

アルゴリズム	鍵	暗号の種類	暗号の名前(例)
秘匿	一部秘匿	ストリーム暗号 ブロック暗号	NSFR MULTIX
公開	共通鍵 (秘匿)	ストリーム暗号 ブロック暗号	バーナム暗号, RC4 DES, FEAL, MISTY, AES
	公開鍵 (一部公開)	メッセージ復元 メッセージ非復元 (主としてデジタル署名用)	RSA, ElGamal, Cramer-Shoup DH, PSS, DSA

変換テーブルK

秘密鍵(共通鍵):K

鍵の数 $|K|=26! \div 4.0 \times 10^{26}$

平文:小文字
暗号文:大文字

暗号化の鍵Kの対応表

a	b	c	d	e	f	g	h	i	j	k	l	m
X	N	Y	A	H	P	O	G	Z	Q	W	B	T
n	o	p	q	r	s	t	u	v	w	x	y	z
S	F	L	R	C	V	M	U	E	K	J	D	I

復号化の鍵Kの対応表

A	B	C	D	E	F	G	H	I	J	K	L	M
d	l	r	y	v	o	h	e	z	x	w	p	t
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
b	g	f	j	q	n	m	u	s	k	a	c	i

平文 $m = \text{"encryption"} \rightarrow$ 暗号文 $c = \text{"HSYCDLMZFS"}$

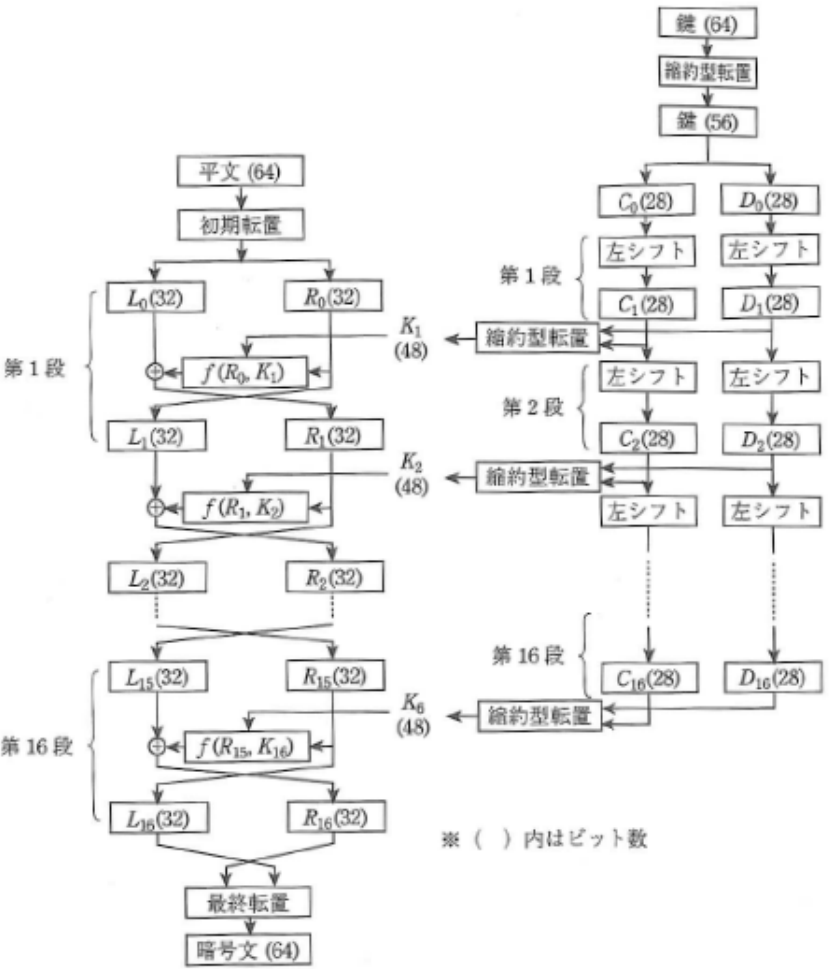
図 1.38 換字法の例

変換テーブルK(鍵)
(ブロック長:5)

$$K = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 4 & 3 & 1 & 5 & 2 \end{pmatrix}$$

平文 $m = \text{"cyber"}$
 $\rightarrow$ 暗号文 $c = \text{"bryce"}$

図 1.39 転字法の例



※ () 内はビット数

図 1.40 DES アルゴリズム

11.4 公開鍵暗号

公開鍵暗号

公開鍵暗号は、数論を用いた計算を行うことで実行されます。一方向性関数による数論の難問を解く困難さにより安全性を保っています。公開鍵暗号には、素因数分解問題の難問性に基づく RSA 暗号、離散対数問題に基づく ElGamal 暗号、ナップザック問題に基づく MH 暗号、線形符号の復号問題に基づく McEliece 暗号、楕円曲線上の離散対数問題を安全性の根拠とする楕円曲線暗号などがあります。

RSA 暗号のアルゴリズム

●準備

(1) 2つの大きな素数 p, q を選び、合成数 $n=pq$ を求める。 p, q は秘密にしておく。

(2) オイラー関数 $\phi(n) = (p-1)(q-1)$ を計算し、

$$\text{GCD}(d, \phi(n)) = 1$$

$$ed = 1 \pmod{\phi(n)}$$

を満たす対 (e, d) を求める。すなわち、 $\phi(n)$ と互いに素な整数 d を求め、 $\phi(n)$ を法とする演算の下に d の逆数を e とする。

(3) ここで

KE (暗号化鍵): (e, n)

KD (復号鍵): (d, n)

とし、KE を公開し、KD を秘密に保つ。

●暗号化と復号

(1) 送信者 A はメッセージ (平文) m を $0 \leq m \leq n-1$ の整数で表現し、受信者 B の公開鍵 KE を用いて暗号文 c を

$$\text{暗号化: } c = m^e \pmod{n}$$

とする。もちろん暗号文 c は $0 \leq c \leq n-1$ となる整数である。

(2) 受信者 B は自分だけ知っている秘密の鍵 KD を用いて

$$\text{復号: } m = c^d \pmod{n}$$

を計算し、平文 m を得る。

ただし、 $\phi(n) = (p-1)(q-1)$ は p, q がともに素数のときに成り立つ。また、 $\text{GCD}(a, b)$ は a と b の最大公約数を示す。

RSA 暗号の計算例

では、具体的な数値を当てはめて実際に計算してみましょう。

まず [準備] として (1) $p=7, q=5$ とします。 $n=35$ です。 $p=7, q=5$ は秘密にします。 (2) $\phi(n) = (7-1)(5-1) = 24$ です。 d と 24 の最大公約数が 1 (つまり $\text{GCD}(d, 24)=1$) となる d を $d=7$ とします。ユークリッドの互除法を用いて、 $7e=1 \pmod{24}$ を解くと $e=31$ となります。 (3) 暗号鍵 KE=(31,35)、復号鍵 DE=(7,35) とまります。

次に [暗号化と復号] です。 (1) 暗号化:

平文 $0 \leq m \leq n-1=34$ となる $m=8$ とします。 $c=m^e=8^{31}=84 \cdot 84 \cdot 84 \cdot 84 \cdot 84 \cdot 84 \cdot 84 \cdot 83=512 \pmod{35}=22 \pmod{35}$ 。 (2) 復号: $m=c^d=22^7=8 \pmod{35}$ となり平文が復元されることがわかります ($84=1 \pmod{35}$, $224=1 \pmod{35}$, $223=8 \pmod{35}$ を用いました)。

RSA 暗号の特徴

RSA 暗号は、秘密鍵暗号比べて、アルゴリズムが複雑で計算量が大きいという特徴があります。ただし鍵管理に関しては、 N 人が相互に通信するためには N この秘密鍵を管理すればよく、鍵管理は秘密鍵暗号比べて容易になります。

また、RSA 暗号は素因数分解の困難さに安全性の根拠を置いており、 n を素因数分解し p と q を求めるためには 1 ステップあたり 1μ 秒のコンピュータを用いた場合、表 11.4 のような計算時間がかかることになります。

表 1.19 解読にかかる平均総時間

n の桁数 (10進)	処理回数	処理時間
50	1.4×10^{10}	3.9時間
100	2.3×10^{15}	74年
200	1.2×10^{23}	3.8×10^9 年
300	1.5×10^{29}	4.9×10^{15} 年

ハイブリット型暗号

秘密鍵暗号と公開鍵暗号の比較を表 1.20 に示します。ここで示したように、秘密鍵暗号は鍵配送が必要で鍵管理のコストも高くなっています。ただし処理速

度は小さくなっています。これに対して公開鍵暗号は鍵配送が不要で鍵管理のコストも少ない代わりに処理速度が大きくなります。

このように長所と短所があるため、実際には、両方の暗号を組み合わせられています。つまり公開鍵暗号を使ってその後用いる秘密鍵 (高々 100 ~ 300 ビット) を配送・共有して、実際に送りたいデータ (大量の場合もある) の暗号化には先ほど共有した秘密鍵を用いて暗号化する。

表 1.20 秘密鍵暗号と公開鍵暗号の比較

暗号方式	鍵配送	鍵管理	処理速度
秘密鍵暗号	必要	$n(n-1)/2$	小
公開鍵暗号	不要	n	大

鍵 K を公開鍵暗号で暗号化し、メッセージ m を鍵 K を使って送る

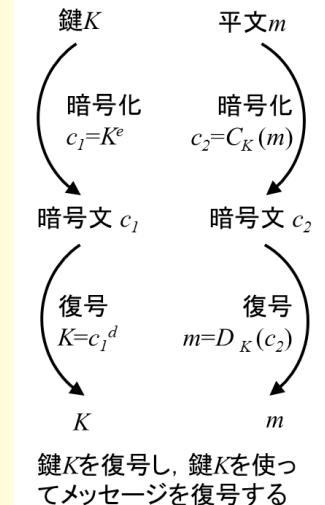


図 1.41 ハイブリット型暗号

11.5 デジタル署名と認証

本節では、公開鍵暗号の優れた性質がデジタル署名や認証に生かされていることを理解します。

暗号系とデジタル署名系

図 1.42 に暗号系とデジタル署名系の比較を示します。まず、暗号系として A が B に暗号文を送ることを考えます。この場合、A が B の公開鍵を用いて平文を暗号化し、暗号文を B に送ります。B は送られてきた暗号文を自身で持っている復号鍵（秘密鍵）を用いて復号します。次に、B が A にデジタル署名を送ることを考えます。ここで RSA 暗号では公開鍵と秘密鍵を入れ替えても暗号機能は成り立つという性質を用います。B が B の秘密鍵を用いて検証文を暗号化し（こ

れが署名）、A に暗号文（署名文）を送ります。A は B の公開鍵を用いて署名文を復号（検証）します。復号が成功すれば送られてきた検証文は確かに B さんから送られたものであり、信路の途中で改ざんされていないということが確認できたことになります。

エンティティ認証

エンティティ認証（相手認証）は、本当に通信相手が B であるということを A が検証することです。他人による B への「なりすまし」を防止します。図 1.43 にエンティティ認証の概要を示します。A は B に乱数を送ります。B は送られてきた乱数を B の秘密鍵で暗号化し、暗号文を A に送り返します。A は送られて

きた暗号文を B の公開鍵で復号し、復号結果と最初に B に送信した乱数を比較します。一致した場合には認証成功であり、一致しない場合には否認します。

メッセージ認証

メッセージ認証は、B が A に送るメッセージに署名（印鑑のようなもの）を付けて送り、A がこれを検証します。図 1.44 にメッセージ認証の概要を示します。B は送りたいメッセージを自分の秘密鍵で暗号化します。さらに、B はメッセージ

のハッシュ（メッセージを一方方向性ハッシュ関数で処理した結果（メッセージダイジェストともいう））を自分の秘密鍵で暗号化（署名）します。これらを A に送ります。A は、暗号文と署名を B の公開鍵を用いて復号します。復号した平文のハッシュを計算し、署名の復号結果と比較します。一致した場合には署名が正当であり、一致しない場合には不正と判断します。

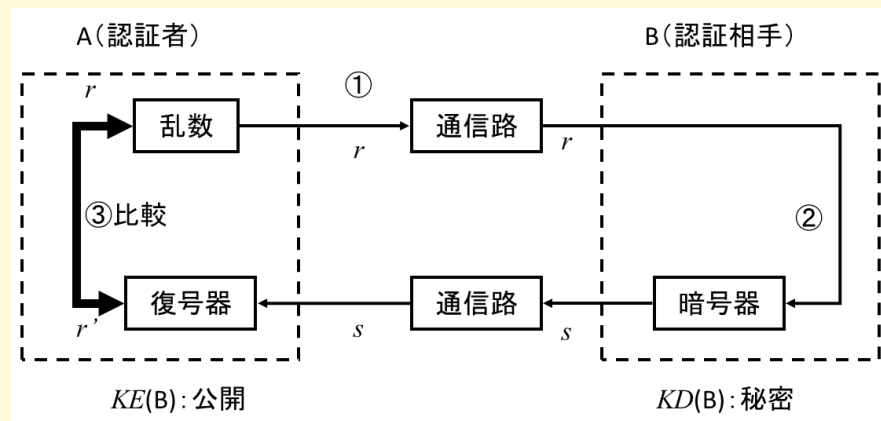


図 1.43 エンティティ認証（相手認証）

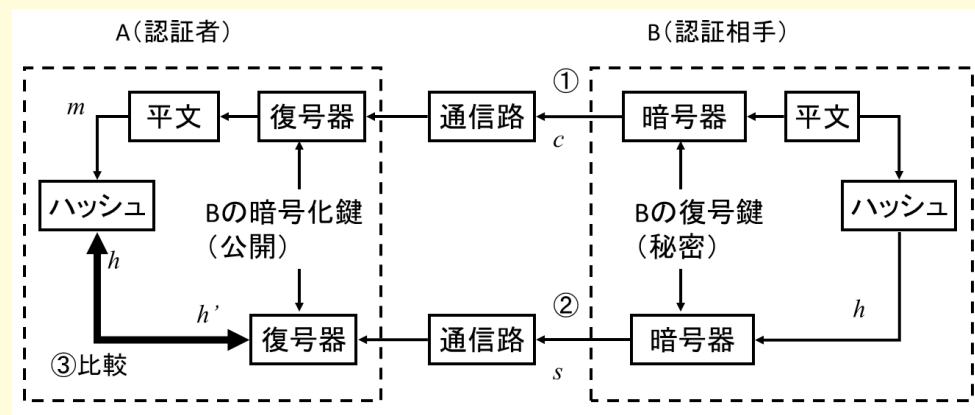


図 1.44 メッセージ認証

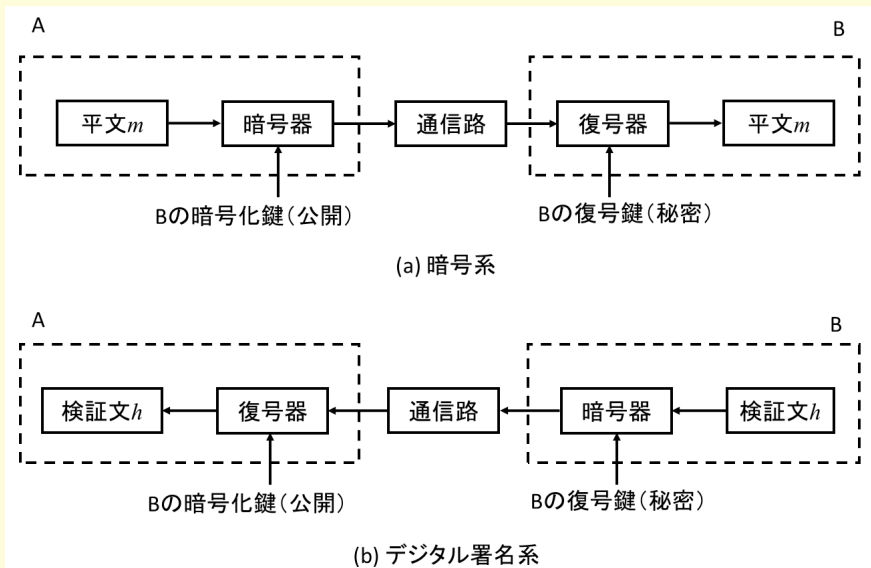


図 1.42 暗号系とデジタル署名系の比較

11.6 公開鍵証明書と認証局

前節までで公開鍵暗号の仕組みと公開鍵暗号を使ったデジタル署名と認証について見てきました。しかし、公開鍵は数字の羅列であるため、公開鍵だけを見てその公開鍵が誰のものであるかを特定できません。つまり、だれがデジタル署名を生成したのかということを公開鍵だけを見て判断できません。これらの問題を解決するための仕組みが公開鍵証明書と認証局です。

公開鍵証明書

公開鍵証明書は、公開鍵にその所有者の情報などを付加した情報です。そしてその情報が正しいことを保証するために、公開鍵と所有者の組み合わせを保証する機関（これを認証局と呼ぶ）が、自身の秘密鍵でデジタル署名を付加したものが公開鍵証明書です。デジタル署名を認証局の公開鍵を使って検証することにより、公開鍵とその所有者の情報に正しいことが確認できます。公開鍵証明書のフォーマットを図 1.45 に示します。

証明書のバージョン
シリアル番号
署名アルゴリズム
発行者の名前
有効期限
主体者（公開鍵の持ち主）の名前
公開鍵アルゴリズム
署名アルゴリズム（3段目と同一）
署名値

図 1.45 X.509 公開鍵証明書のフォーマット

いことが確認できます。公開鍵証明書のフォーマットを図 1.45 に示します。

認証局

いままで見てきたように、公開鍵とその所有者の対応関係を保証する機関が認証局（Certification Authority: CA）です。認証局は信頼できる第三者機関という位置づけになります。公開鍵証明書に付与された認証局のデジタル署名値を検証することができますが、その検証には認証局の公開鍵を使います。では、その公開鍵は本当に認証局の持ち物でしょうか。認証局の公開鍵にも公開鍵証明書が必要となります。認証局の公開鍵に対して、さらに上位の認証局が公開鍵証明書を発行します。このように認証局は階層構造を取ります。認証局の階層構造を図 1.46 に示します。

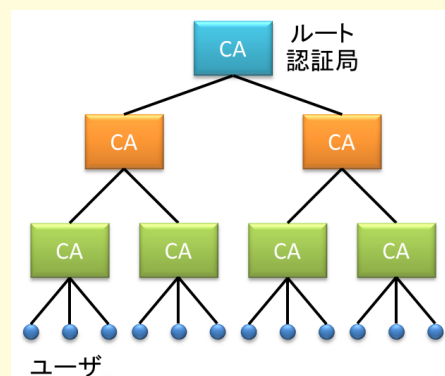


図 1.46 認証局の階層構造

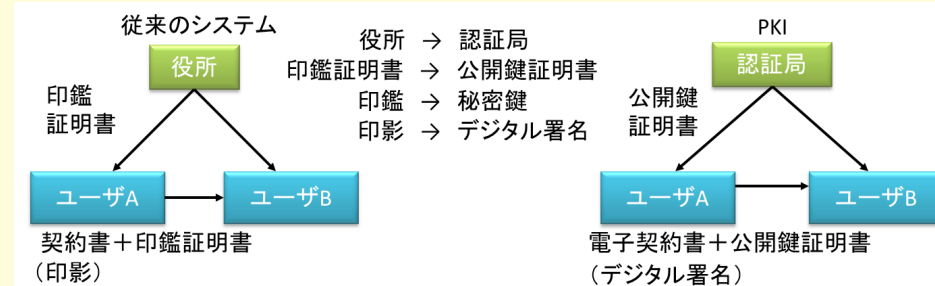


図 1.47 従来システムと PKI の比較

PKI

このように、公開鍵暗号技術、および電子証明書、認証局などの技術を利用したセキュリティのインフラ（基盤）のことを公開鍵暗号基盤（Public Key Infrastructure: PKI）と呼びます。最後に、従来のシステム（印鑑登録と印鑑証明書）と PKI の対応を図 1.47 に示します。図に示すように役所の代わりが認証局、印鑑証明書の代わりが公開鍵証明書、印鑑の代わりが秘密鍵、印影の代わりがデジタル署名になります。

その他の電子認証システム

その他の電子認証システムに下記のようシステムがあります。

- ① 時刻証明：当該情報が当該時刻に存在したことを証明する（タイムスタンプ機能）。
- ② 内容証明：当該内容の情報が通信されたことを証明する。
- ③ 配達証明：受信者が当該情報を受け取ったことを証明する。
- ④ 原本保存：情報（原本）を保存し、要求に応じ謄本を発行する。

ひらさわ教授のコラム 4

最近、暗号の研究を少しは始めましたが、余り進んでいません。暗号は好きではないのです。何故ならば、暗号研究で最も価値のあるのは、誰かが提案した暗号のアルゴリズムを解読すること、すなわちその研究成果を潰すことだからです。他の多くの分野の研究分野では、新しい方法・方式・アルゴリズムなどを提案して、それが従来知られているものと比較して優れていれば評価されます。しかし、暗号はそれが実用化されるまでにその脆弱性を潰さなければ、社会に大きな悪影響を及ぼします。こうして生き残ったものだけが実用的価値がある訳です。逆に言えば、暗号を解読する方法を明らかにすることが、新しい暗号を発見することより大切なことです。

現在までに、そのような画期的な研究があります。DES（Data Encryption Standard の略。元は IBM 社の特許。アメリカ国家標準局が DES として標準化した。）に関し、一発解読ではありませんが、かなり有効な方法が発表されました。日本の M 社の研究グループが PC を何千台も使用して、実用の範囲で解読できる攻撃法を発見しました。これを契機に、DES は改良され 3 重 DES、AES を経て新しい暗号に変わってきています。公開鍵暗号も同じ運命でしょう。コンピュータの性能が上がるにつれて、2 つの素数の積が 100 桁から 200 桁に増え、今後さらに桁数が増えて行くでしょう。素因数分解の効率的なアルゴリズムが発見されないとも限りません。このように一発解読でなくても、暗号解読には危うさが潜んでいるのです。暗号の研究は、新しい暗号を発見することより既存の暗号を解読することが大切で、弱点を突く研究が求められており、評価されるのです。この点が、どうも私には暗号研究が向かない、意欲が出ない理由です。

演習問題

1. 次の空欄 [a][b][c] に入る適切な語の組み合わせを、(1)-(4)の中から選びなさい。

公開鍵暗号は [a] を公開し、[b] を秘密にする。秘匿を目的とする場合、平文を [a] で暗号化する。[c] では [b] で暗号化し [a] で復号する。

- (1) [a] 暗号化鍵 [b] 復号鍵 [c] デジタル署名
- (2) [a] 復号鍵 [b] 暗号化鍵 [c] デジタル署名
- (3) [a] 暗号化鍵 [b] 復号鍵 [c] ユーザ認証
- (4) [a] 復号鍵 [b] 暗号化鍵 [c] エンティティ認証

2. 次の文章で間違っているものを選びなさい。

(1) 暗号の役割には情報の秘匿の他に認証がある。認証が必要な理由は改ざん・なりすまし・否認に対処するためである。

(2) 秘密鍵暗号では暗号化鍵と秘密鍵が同じで、いずれも秘密にする。公開鍵暗号では暗号化鍵を秘密にし、復号鍵を公開する。

(3) シーザー暗号は換字法を用いた秘密鍵暗号の一種である。英文字の場合、鍵の数は 26 である。

(4) 指紋を用いたユーザ認証は個人の身体的特徴を捉えたもので実用化されている。

3. 次の空欄 [a][b][c] に入る適切な語の組み合わせを、(1)-(4)の中から選びなさい。

不正アクセスの対策にはファイアウォールによる [a]、ユーザを特定する [b]、送信者の押印に相当する [c] がある。

- (1) [a] アクセス制御 [b] メッセージ認証 [c] デジタル署名
- (2) [a] パケットフィルタリング [b] メッセージ認証 [c] 電子署名
- (3) [a] アクセス制御 [b] 個人認証 [c] 電子署名
- (4) [a] ゲートウェイ制御 [b] 端末認証 [c] デジタル署名

4. 次の文章で間違っているものを選びなさい。

(1) インターネットに接続されたパソコンのデータを盗むウィルスはスパイウェアと言う。

(2) スпамフィルタはスパムメールで使われる単語の出現頻度を求め、統計的手法で判断する。

(3) インターネットの脆弱性は善意のユーザによるコミュニティで使われてきた歴史的な背景に起因している。

(4) パケットフィルタ型ファイアウォールは TCP 層でアドレス情報を用いて選別する。

5. 次の空欄 [a][b][c] に入る適切な語の組み合わせを、(1)-(4)の中から選びなさい。

セキュリティの 3 要素には、情報を見せない [a]、情報が破壊・改ざんされない [b]、情報を利用できる [c] がある。

- (1) [a] 秘匿性 [b] 完全性 [c] 原本性
- (2) [a] 機密性 [b] 証明性 [c] 可用性
- (3) [a] 証拠性 [b] 完全性 [c] 有用性
- (4) [a] 機密性 [b] 完全性 [c] 可用性

6. 次の文章で間違っているものを選びなさい。

(1) ファイアウォールは IP フィルタリングにより、外部ネットワークからの不正アクセスを防止するが、内部ネットワークからの不正流出は対象としない

(2) PKI は公開鍵暗号を用いてデジタル署名を可能とし、第三者機関による認証局によって従来の印鑑証明機能を持つ。

(3) 否認と言うのは行為の後、その行為を否定するもので認証機能により防止できる。

(4) VPN は公衆回線の特定ルータ間を暗号化し、特定ユーザが排他的に回線を利用する仮想的私設網である。

7. 次の空欄 [a][b][c] に入る適切な語の組み合わせを、(1)-(4)の中から選びなさい。

電子政府システムを従来のシステムと比較すると、認証局が [a] に、公開鍵証明書が [b] に、デジタル署名が [c] に対応すると言うことが出来る。

- (1) [a] 役所 [b] 印鑑証明書 [c] 印鑑
- (2) [a] 役所 [b] 電子契約書 [c] 印影
- (3) [a] 政府 [b] 契約書 [c] 印鑑
- (4) [a] 省庁 [b] 印鑑証明書 [c] 署名

8. 次の文章で間違っているものを選びなさい。

(1) PKI として仕組みが確立するためには、ユーザとその公開鍵の関係が明確にされねばならない。

(2) 電子署名法が施行されたのは平成 13 年である。

(3) 電子政府の構想には電子商取引の推進がある。

(4) PKI の仕組みの最重点項目は契約書などの文書の暗号化にある。

第 12 章 インターネットのセキュリティ対策

インターネットは非常に便利ですが、その特徴としてセキュリティに対し脆弱であることが知られています。ここでは、具体的にその危険性について考察し、対策を考えます。

12.1 ネットワークにおける不正

12.2 コンピュータウィルス

12.3 ネットワークセキュリティ (1)

12.4 ネットワークセキュリティ (2)

12.5 SSL

12.6 電子政府

12.1 ネットワークにおける不正

ネットワークにおける不正

インターネットはもともと、研究者コミュニティの善意のユーザから誕生し、したがって悪意による混乱ということに非常に脆弱です。現在でもボランティアによる運営がされており、オープン・自由を標榜しています。これらのことから、インターネットを安全に使うためにはセキュリティ対策が必要です。最近、IPv6（プロトコルの一種）にはセキュリティの強化が行われています。

一方、このような技術的な対策は一定の効果はありますが、限界があります。他の方法、法律・道徳・倫理・哲学などの側面から啓蒙活動が不可欠と言われていいます。特に、倫理は、年代・場所などにより異なる公序良俗を啓蒙する活動で、学会などでも取り組まれています。

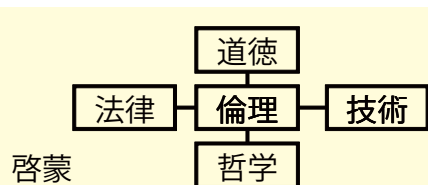


図 1.48 必要不可欠な啓蒙活動

インターネットがセキュリティに対し脆弱であることが、ユーザに危険性をもたらすことを考えて見ましょう。クラッカー（悪意をもって他人のコンピュータのデータやプログラムを盗み見たり、改ざん・破壊などを行う者。一方コンピュー

タネットワークに不正に侵入したりする者のことを一般に「ハッカー」と呼び、悪意を持った「クラッカー」と区別する場合がある。）の出現などの直接的要因と、規制が少ないという自由な環境で自己責任が問われるという間接的要因があります。さらに、高速・広域ネットワークであることが、電気的スピードで瞬時にして世界中に伝わるという怖さがあります。

その波及問題としては、アダルトサイト・出会い系サイト・ネットオークション・ホームバンキング・ネットゲームなどさまざまあり、不正・不純が生じる可能性が指摘されています。

具体的問題としては、クラッカー（悪意のある不正使用者）の出現であり、ここでは不正アクセス・ウィルス・スパムメールを取り上げます。その他に、著作権保護・個人情報保護などがあります。

不正アクセス対策

権限のないユーザがコンピュータに不正にアクセスし、データの破壊・改ざん・不正使用・盗用する行為を不正アクセスと言います。これには、下記に示す (i)(ii) があります（「不正アクセス行為の禁止等に関する法律（不正アクセス禁止法）」（2000 年施行）より）。

(i) 覗き見・成りすましなどによる他人の利用者識別情報 (ID) の無許可の利用

(ii) 盗聴・改ざん・破壊などによる、利用者識別機能の代替

また、これらのネットワーク上の技術的対策としては、以下が挙げられます。

① アクセス制御・・・FW (Fire Wall)

② 認証・・・個人認証

③ 電子署名・・・PKI

ファイアウォール (FW) は、送信元 IP アドレス・送信先 IP アドレス・送信元ポート番号・送信先ポート番号などのアドレス情報と ACL (Access Control List) を対比させ、不正アドレスを検出します。これにはさらに以下の①～③のタイプがあります。

① IP 層におけるパケットフィルタ型

② TCP 層におけるサーキットレベルゲートウェイ型

③ アプリケーション層におけるアプリケーションゲートウェイ型

①は単純・高速、②は設定は簡単ですが、クライアント側ソフトの対応が必要です。③はアプリケーション固有の制御が必要で、要求されたセキュリティレベルにあわせ選択する必要があります。上記 3 つのデータ転送レイヤを図 1.49 に示します。図中 AP はユーザシステムに近いアプリケーション層、DL はネットワークハードウェアに近いデータリンク層です。

最後にネットワークにおけるファイアウォールの位置を図 1.50 に示します。FW は不正アクセスを禁止するために通すもの・通さないものをフィルタリングします。ここで、インターネットから Web サーバ・メールサーバは見る事が出来ます。

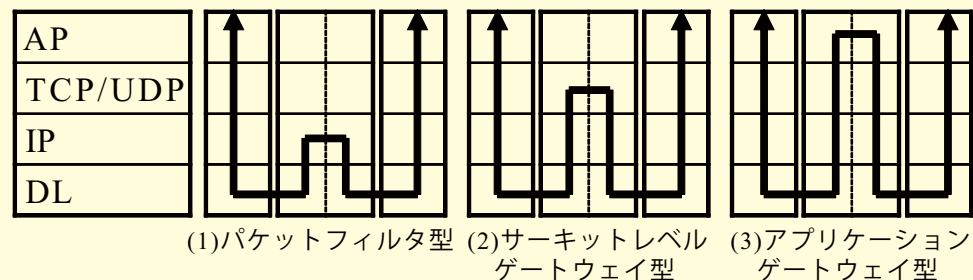


図 1.49 FW のデータ転送レイヤ

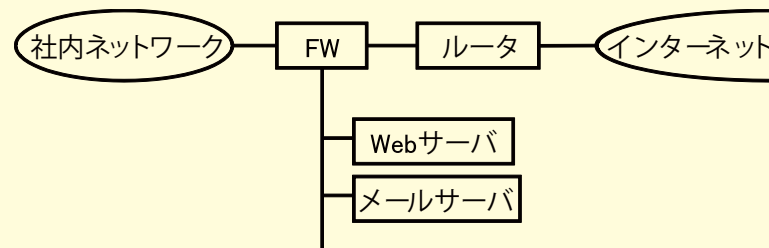


図 1.50 FW の位置

12.2 コンピュータウイルス

ウイルス

ウイルスとは、第三者のプログラムやデータベースに対し、意図的に被害を及ぼすようなプログラムです。次の項目、自己伝染機能・潜伏機能・発病機能の一つ以上有するものを言います（「コンピュータウイルス対策基準」（2000年通産省告示最終改定）より）。

- ・自己伝染機能：他のプログラムに自らをコピー、あるいは自らを他のシステムにコピーする。
- ・潜伏機能：特定時刻・一定時間・処理回数などの条件により発病する（初期は発病を知らない）。
- ・発病機能：プログラム・データのファイルを破壊、或いは意図しない動作をする（実際に起こる）。

ウイルスの種類には、ワーム・マクロウイルス・トロイの木馬（木馬の中に兵士がたくさんいて、夜になると敵陣内で活動するというギリシャ時代の神話にな

ぞらえた術語。バックグラウンドで悪意にある動作をする 1987 年「PC-Write」が有名）・自己増殖型ウイルス・スパイウェア（インターネットに接続された PC の状態（頻度・傾向など）を盗む）などがあります。従来からのコンピュータウイルスの分類を図 12.2 に示します（「従来からの」というのは現在では必ずしもこの分類が適用されるとは限らないからです）。

これらに対して下記のような対策があります。

- ①ウイルス対策ソフト
- ②ウイルス予防
- ③ウイルス対策の運用管理

●ウイルス対策ソフト

①ウイルス対策ソフトには、例えばシマンテック社のものには次のような機能があります。

[例] シマンテック社ノートンインターネットセキュリティ

- ・ファイアウォール (FW)
- ・アンチウイルス（メール・ファイルの

保護）

- ・アンチスパム（スパムメール検知）
- ・広告ブロック
- ・不正侵入検知
- ・更新ファイルサービス（ウイルス定義ファイル・スパム定義ファイル・スパムメール特性ファイル・Web フィルタリングリスト等）

●ウイルス予防

②ウイルス予防には、常に次のような対策を立てる必要があります。

- ・FW によりアクセス可能な発信元・プロトコルを制限
 - ・サーバ・クライアントの脆弱性対策
 - ・不正侵入検知
 - ウイルス対策の運用管理
- 企業における③ウイルス対策の運用管理には、下記を制度化し、運用することが望まれています。
- ・早期発見・早期撃退
 - ・運用管理手順・規則の制定
 - ・行動基準の制定
 - ・ウイルス対策ソフトの評価

スパムメール

スパムメールとは、受信者の意図に反し、不要なメールを無差別・大量に送られてくる電子メールのことです。スパムメールは全てのメールの約 50 ～ 60% と言われています。このようなスパムメールは通常、広告用です。これに効果があるため、スパムメールが減らないといわれています。その対策には以下の対策が挙げられます。

①ブラックリスト法・・・スパムメールの送信元のブラックリストに従って、受信を拒否する。

[例] ブラックリスト

- ・MAPS (Mail Abuse Prevention System)
- ・RBL (Realtime Black List)

②スパムフィルタ法

- ・スパムで使われる単語（スパム用語）の辞書を作る。
- ・スパム用語の出現確率を求め統計的手法で判定する。

以上をまとめると、次の通りです。

・善意のユーザを前提に構成されてきたインターネットは、もともと危険性をはらんでいる。それが社会問題を誘起する原因になっている。

・悪意のあるクラッカーの手によれば多くの混乱を招くが、現状では不正アクセス対策・ウイルス対策・スパムメール対策などが講じられている。それなりの効果が期待出来るが十分ではない。



英国のコメディ番組において、食堂のメニューに「豚肉と煮豆とスパム」「スパムと卵とソーセージとスパム」「スパムとスパムとスパムと煮豆とスパムとスパムと…」などとなぜか「スパム」ばかり入っており、「スパム」が連発されるうちに、周りにいた客たちが「スパム、スパム、スパム…」と合唱を始め、食堂はわけのわからない状態になります。さらに、その後登場する歴史学者の話にも、さらに、番組の最後のクレジットにもあらゆるところに「SPAM」がちりばめられています。執拗に繰り返される「SPAM」が、迷惑メールなどを表す「スパム」の語源の一つとされています。

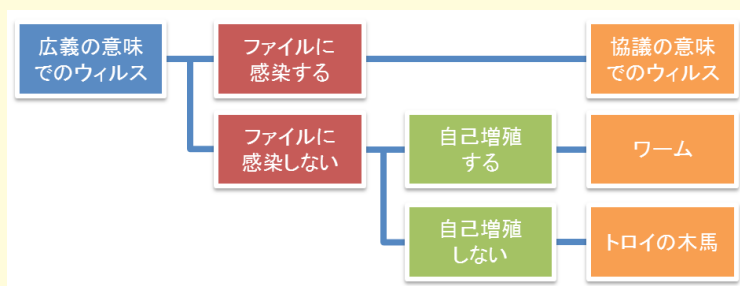


図 1.51 従来からのコンピュータウイルスの分類

12.3 ネットワークセキュリティ(1)

ここでは、インターネットの特性からセキュリティ対策を考えます。セキュリティ確保のためには暗号が活躍します。また、ネットワークにおけるセキュリティ技術の階層化と、これを実現するファイアウォール (FW)・仮想私設網 (VPN)・公開鍵基盤 (PKI) について概要を把握します。

インターネットの脅威と対策

セキュアなネットワークに要求される条件には、機密性・完全性・可用性・証拠性・原本性があります。このうち、機密性 (Confidentiality)・完全性 (Integrity)・可用性 (Availability) は特に重要で、セキュリティの3要素、略してCIAと言われています。それぞれの意味、喪失時の脅威、考えられるその対策を表1.21に示します。

表 1.21 インターネットの脅威とセキュリティ対策

条件	内容	喪失時	主な対策
機密性	情報を見せない	見られる・盗まれる	暗号化
完全性	情報が完全である	破壊・改ざんされる	アクセス管理
可用性	情報を利用出来る	情報が使えなくなる	アクセス管理
証拠性	事実を証明出来る	事実が改ざん・否認される	デジタル署名
原本性	情報がコピーされない	不正にコピーされる	デジタル署名

セキュリティの3要素: 機密性 (Confidentiality)・完全性 (Integrity)・可用性 (Availability), 略してCIA.

情報セキュリティ技術の階層

図1.52に情報セキュリティ技術の階層を示します。図中の耐タンパー技術は、中をこじ開けようとするとう壊れてしまうような作り方を言います。一般的にはハードウェア・ソフトウェアによらず、容易に外部から解析できないよう不正な手段による機密データの読み取りを防ぐ能力を指します。図中、上位ほど総合的な対策を指しています。すなわち下位の技術を部品として組み込み、上位のシステムを構築します。

図1.53にネットワークセキュリティの強化を示します。右に進むほどセキュリティが強化されます。

FWは企業向けにイントラネットを構築します。イントラネットとは、ネットワーク上でアクセス出来るユーザが限定

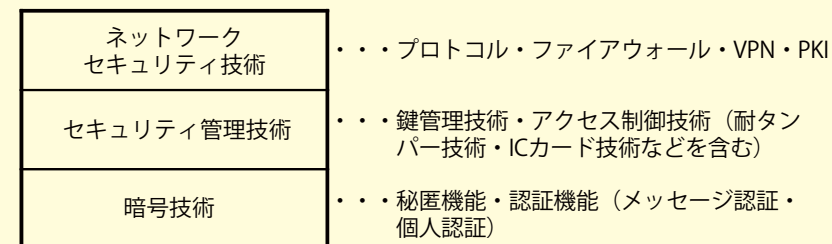


図 1.52 情報セキュリティ技術の階層

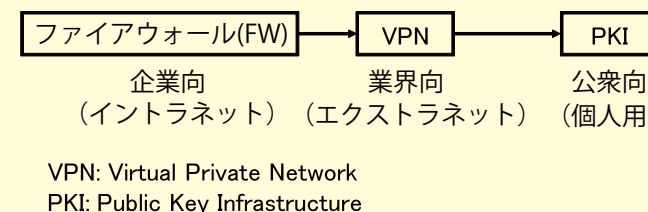


図 1.53 ネットワークセキュリティの強化

されたネットワークの領域を指します。

また、VPNは業界向けにエクストラネットを構築します。エクストラネットとは、イントラネットを拡張したネットワークで、多企業間でデータベースなどのリソースを共有するセキュアなネットワーク領域を指します。自動車業界のサプライチェーンマネジメントシステムなどが応用例として有名です。PKIは、電子商取引や電子データ交換を可能にします。

PKIは前章で述べたように、ユーザの身元について信頼できる第三者機関が審査・保証を行う仕組みを有し、公開鍵証明書に記載されたユーザの公開鍵を使用してユーザが相互に相手を正当であると認める認証機

能を持ち、ユーザ同士が直接、通信文を暗号化・復号することを可能とするセキュアなネットワークを指します。

ファイアウォール (FW)

FWはフィルタ機能を持つゲートウェイで、IPフィルタリングによるアクセス制御を行います。その結果、外部ネットワークからの不正アクセスの防止し、また外部ネットワークへの不正流出の防止することが出来ます。図1.54に、FWの位置を示します。

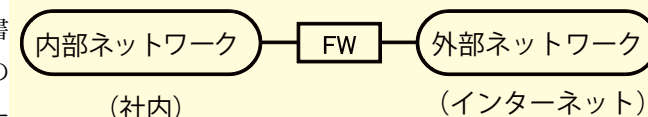


図 1.54 FWによる内部ネットワークの保護

12.4 ネットワークセキュリティ (2)

仮想私設網 (VPN)

VPN は公衆網に暗号技術を用いて仮想的に専用回線網を構築する方法です。仮想的な専用回線網とは、物理的にはつながっているが、論理的には別のネットワークになっている網です。特定ルータ間の通信を暗号化し、特定ユーザだけが排他的に利用する（他人が入られない）閉域ネットワークを論理的に作ります。具体的な実現方法には、下記の方式があります。

- ①プロバイダ・キャリアが自社回線を用いて IP 網を提供する IP-VPN 方式
- ②インターネットを使って自社で構成するインターネット VPN 方式

ここで用いる暗号化には、秘匿（盗聴防止）と認証（改ざん防止）が含まれます。

以上の結果、図 1.55 に示した通り、インターネット（公衆網）を介して仮想

的に私設専用線を用いたクローズドネットワーク通信が可能になります。

ファイアウォール (FW) 機能・仮想私設網 (VPN) 機能を別々に実現するのではなく、ウィルス対策機能・スパムメール対策機能・内容フィルタリング機能・不正侵入の検出・防御機能などを含む統合化装置があります。この構成例を図 1.56 に示します。この図で、インターネットからは統合化装置に繋がる Web サーバ・メールサーバは見えますが、社内ネットワークに繋がる Web サーバ・メールサーバは見えません。

公開鍵基盤 (PKI)

公開鍵基盤 (PKI) は、証明書の作成・管理・格納・配布・破棄などに必要な資源・ポリシー・プロトコルによってサービスされる基盤です。公開鍵暗号を用い、認証局 (Certification Authority: CA) が第三

者機関 (Trusted Third Party: TTP) によって運営されます。

不正対策とシステムの機能

以上で述べたことをまとめると、表 1.22 の通りです。ここで、○印は不正対策が可能であることを示します。改ざん・成りすまし・否認の対策には PKI が必要です。高度な対策が可能となりますが、コストは大きくなります。

セキュリティアーキテクチャ

表 1.23 に ISO7948 によるセキュリティアーキテクチャを示します。この中味は秘匿・アクセス制御の他、改ざん・成りすまし・否認などのネットワークにおける不正防止を全て含んでいます。また、表 1.24 に不正行為とその対策をまとめておきます。

ネットワークアーキテクチャとプロトコル

表 1.25 に TCP/IP アーキテクチャとプロトコルの対応を示します。コールバックとは、電話などでかけた相手から（一旦回線を切って）本当に本人かどうか再度かけてもらって確認する方法です。ネットワーク層における IP フィルタリングは FW の仕事です。

- 以上で重要なことは、次の通りです。
- ・セキュリティの 3 要素 (CIA) は機密性 (Confidentiality)・完全性 (Integrity)・可用性 (Availability) である。
 - ・ネットワークのセキュリティ対策にはウィルス対策・FW・VPN・PKI がある。FW・VPN を統合化したシステムも用いられている。

表 1.22 不正対策とシステムの機能

	盗聴	改ざん	成りすまし	否認	暗号化	認証
ウィルス対策		○				
FW	○					
VPN	○	○			○	
PKI	○	○	○	○	○	○

表 1.24 ネットワークにおける不正行為と対策

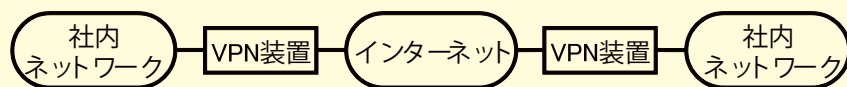
ネットワークへの攻撃・不正アクセス	セキュリティ対策
盗聴（解読）	秘匿（暗号化）
改ざん	メッセージ認証
成りすまし（偽造）	エンティティ認証 アクセス管理 ワンタイムパスワード
否認	エンティティ認証
その他	

表 1.23 ISO7948 のアーキテクチャ

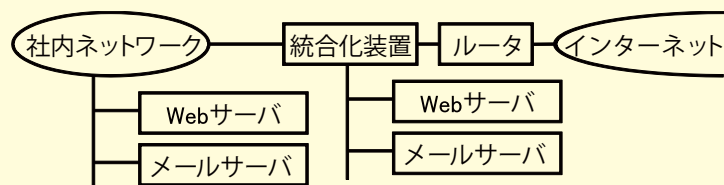
項目	内容	対策
相手承認	正当なユーザか	認証機能（なりすまし防止）
アクセス管理	アクセス権があるか	アクセス制御
秘匿	盗聴防止	秘匿機能
情報の完全性	改ざん・誤り防止	認証機能
否認防止	行為を後に否認	認証機能

表 1.25 TCP/IP アーキテクチャとプロトコル

レイヤ	プロトコル
7:アプリケーション層	相手認証・アクセス権限制御
4:トランスポート層	回線選択（盗聴・障害回線の回避）
3:ネットワーク層	IP フィルタリング (FW)
2:データリンク層	回線番号・コールバック・オフライン接続



暗号化・（復号） （暗号化）・復号
図 1.55 VPN を用いた社内ネットワークの網構成例



（社内用）
図 1.56 統合化装置を用いたネットワーク構成例

12.5 SSL

ここでは、インターネットでいろいろなサービスを受ける際の安全性を確保する SSL について見ていきます。

SSL の役割

SSL(Secure Socket Layer) とは、ウェブサイトに入力する個人情報やクレジットカード情報などを暗号化し、安全に送受信する技術です。SSL の役割を下記に示します。

- ・ブラウザとウェブサーバとの通信を暗号化することで個人情報を守ります。
- ・アクセス先のウェブサイトが安全なウェブサイトか否かを確認できます。

SSL の見た目

上述のように SSL は、インターネット上でのやり取りを安全に行える技術です。アクセス先のウェブサイトが SSL に対応しているかを確認することが大切です。図 1.57 に示すように SSL が導入されているウェブサイトでは、ブラウザのアドレスバーに表示される URL 「http://」にセキュア (Secure) を表す「s」が付き、「https://」になります。

SSL プロトコル

SSL(v3) での SSL Handshake Protocol を図 1.58 に示します。

まず、ClientHello メッセージにてクライアント側でサポートしている暗号化・圧縮アルゴリズムが送信されます。サーバ側でその中から選択したあと ServerHello メッセージで送信される。このとき双方でランダムに生成した値も交換されます。Certificate メッセージでサーバ自身の公開鍵証明書が送信されます。公開鍵証明書が存在しない場合には、ServerKeyExchange メッセージで RSA 公開鍵又は Diffie-Hellman 公開鍵が送信されます。SSL で使用される公開鍵証明書は X.509 v3 が採用されています。

CertificateRequest メッセージはクライアントに公開鍵証明書の提示を求める場合に用いられます。ServerHelloDone メッセージで ServerHello メッセージからの一連の送信が終えたことをクライアントに伝えられます。クライアントは CertificateRequest メッセージを受け取った場合に限り Certificate メッセージを送信します。クライアントが公開鍵

証明書を持っていない場合にはここで Alert を返却します。クライアントは 48 ビットの PreMasterSecret データを生成し ClientKeyExchange メッセージで送信します。CertificateVerify メッセージはクライアントの公開鍵証明書の検証を行う際に利用されます。Finished メッセージは SSL Handshake Protocol が終了するときに送信される。

ChangeCipherSpec メッセージは使用する暗号化・圧縮アルゴリズムを変更する際に送信されます。新規セッション確立時の SSL Handshake Protocol 中にも利用されます。

EV SSL 証明書

EV(Extended Validation) SSL 証明書とは、通常の SSL サーバ証明書と比較してさらに信頼性を高めたサーバ証明書です。

SSL で使われるサーバ証明書の信頼性が低下しているという状況を受けて、認証局と Web ブラウザのベンダで構成する業界団体「CA/Browser フォーラム」が、利用者の身元情報などをより厳格に審査をしたうえで証明書を発行する「EV (extended validation) 証明書ガイドライン」を策定しました。このガイドラインによる厳しい審査をクリアした企業だ

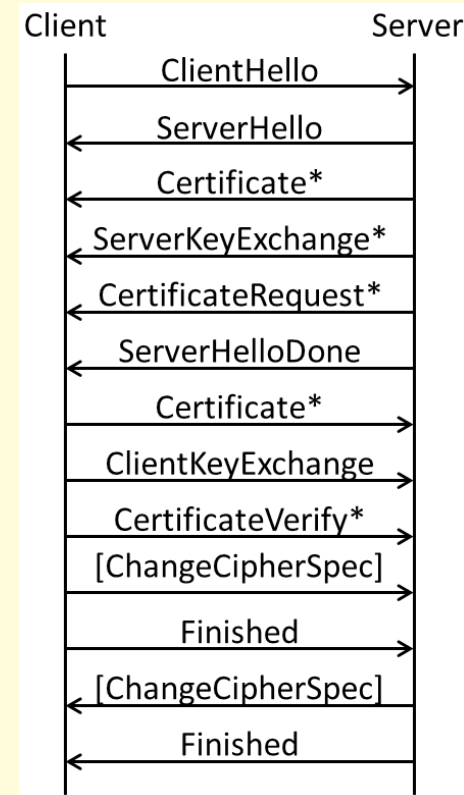


図 1.58 SSL Handshake Protocol

けに発行されるサーバ証明書が EV-SSL 証明書です。

発行された EV-SSL 証明書には、それが EV-SSL 証明書であることを示す認証局ごとの識別子情報が記述されます。Web ブラウザは、この識別子情報をチェックして、証明書が EV-SSL 証明書であることを認識します。



図 1.57 SSL 時のブラウザのアドレスバー

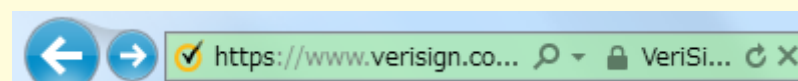


図 1.59 EV SSL サーバ証明書のサイト

12.6 電子政府

SSL の役割

電子政府（e-Japan）の構想における重点項目は、下記の通りです.

さらに、総務省はユビキタスネット社会の実現を目指した u-Japan 政策を通じて、e-Japan 戦略の今後に貢献していくことを宣言しています。(http://www.soumu.go.jp/menu_seisaku/ict/u-japan/index.html 参照).

u-Japan では、これまでの有線中心の

表 1.26 電子政府（e-Japan）の歴史

年	出来事
1994	政府内に高度情報通信社会推進本部設置
1999	行動計画発表
2001	e-Japan戦略発表 電子署名法施行
2002	e-Japan重点計画発表
2003	住民基本台帳ネットワーク稼動開始 住民基本ICカード発行開始 電子証明書の発行開始
2004	税金の電子申告開始

参考文献：石田晴久，インターネット安全活用術，岩波書店，2004.10.

⑤高度情報通信ネットワークの安全性・信頼性確保（高速大容量ネットワークの形成）

u-Japan では、これまでの有線中心の

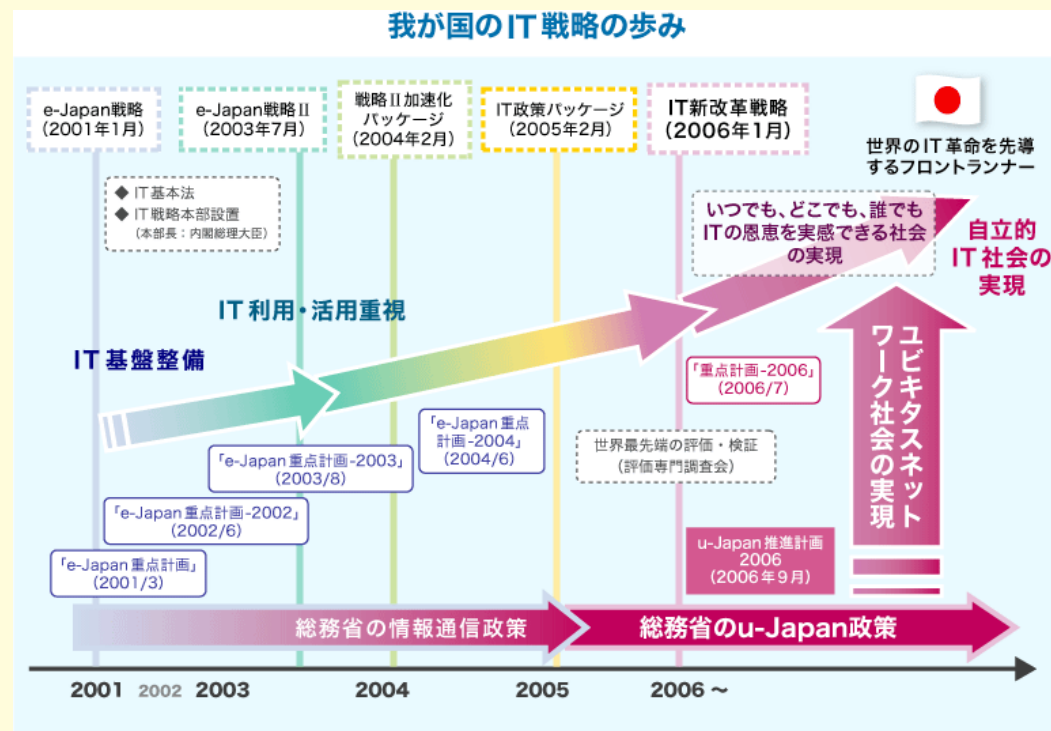


図 1.60 総務省「e-Japan 戦略」の今後の展開より

インフラ整備から、有線・無線の区別のないシームレスなユビキタスネットワーク環境への移行を目指しています。有線から無線、ネットワークから端末、認証やデータ交換等を含めた有機的な連携によって、あらゆる場面でネットワークにつながる環境の整備を掲げています。

ザ登録（印鑑登録に対応）が必要です.

では、何をやろうとしているのか。現在可能なことは、国税の申告・納税・届出（e-Tax）とパスポートのオンライン申請程度です。

電子政府システムに必要な住民基本台帳カードの例を示します。写真が入っているので官公庁が発行している公的な身分証明書になります。

電子政府システムは全面的に PKI，すなわち公開鍵暗号とデジタル署名を用いることを前提としています。その結果，文書の暗号化と盗聴・改ざん・成りすまし・否認の防止が出来，将来の電子商取引に繋がります。ただし，デジタル署名用の電子証明書(印鑑証明に対応)とユー



図 1.61 住民基本台帳カードの例

演習問題

1. 次の空欄 [a][b][c] に入る適切な語の組み合わせを、(1)-(4)の中から選びなさい。

公開鍵暗号は [a] を公開し、[b] を秘密にする。秘匿を目的とする場合、平文を [a] で暗号化する。[c] では [b] で暗号化し [a] で復号する。

- (1) [a] 暗号化鍵 [b] 復号鍵 [c] デジタル署名
- (2) [a] 復号鍵 [b] 暗号化鍵 [c] デジタル署名
- (3) [a] 暗号化鍵 [b] 復号鍵 [c] ユーザ認証
- (4) [a] 復号鍵 [b] 暗号化鍵 [c] エンティティ認証

2. 次の文章で間違っているものを選びなさい。

(1) 暗号の役割には情報の秘匿の他に認証がある。認証が必要な理由は改ざん・なりすまし・否認に対処するためである。

(2) 秘密鍵暗号では暗号化鍵と秘密鍵が同じで、いずれも秘密にする。公開鍵暗号では暗号化鍵を秘密にし、復号鍵を公開する。

(3) シーザー暗号は換字法を用いた秘密鍵暗号の一種である。英文字の場合、鍵の数は 26 である。

(4) 指紋を用いたユーザ認証は個人の身体的特徴を捉えたもので実用化されている。

3. 次の空欄 [a][b][c] に入る適切な語の組み合わせを、(1)-(4)の中から選びなさい。

不正アクセスの対策にはファイアウォールによる [a]、ユーザを特定する [b]、送信者の押印に相当する [c] がある。

- (1) [a] アクセス制御 [b] メッセージ認証 [c] デジタル署名
- (2) [a] パケットフィルタリング [b] メッセージ認証 [c] 電子署名
- (3) [a] アクセス制御 [b] 個人認証 [c] 電子署名
- (4) [a] ゲートウェイ制御 [b] 端末認証 [c] デジタル署名

4. 次の文章で間違っているものを選びなさい。

(1) インターネットに接続されたパソコンのデータを盗むウィルスはスパイウェアと言う。

(2) スпамフィルタはスパムメールで使われる単語の出現頻度を求め、統計的手法で判断する。

(3) インターネットの脆弱性は善意のユーザによるコミュニティで使われてきた歴史的な背景に起因している。

(4) パケットフィルタ型ファイアウォールは TCP 層でアドレス情報を用いて選別する。

5. 次の空欄 [a][b][c] に入る適切な語の組み合わせを、(1)-(4)の中から選びなさい。

セキュリティの 3 要素には、情報を見せない [a]、情報が破壊・改ざんされない [b]、情報を利用できる [c] がある。

- (1) [a] 秘匿性 [b] 完全性 [c] 原本性
- (2) [a] 機密性 [b] 証明性 [c] 可用性
- (3) [a] 証拠性 [b] 完全性 [c] 有用性
- (4) [a] 機密性 [b] 完全性 [c] 可用性

6. 次の文章で間違っているものを選びなさい。

(1) ファイアウォールは IP フィルタリングにより、外部ネットワークからの不正アクセスを防止するが、内部ネットワークからの不正流出は対象としない

(2) PKI は公開鍵暗号を用いてデジタル署名を可能とし、第三者機関による認証局によって従来の印鑑証明機能を持つ。

(3) 否認と言うのは行為の後、その行為を否定するもので認証機能により防止できる。

(4) VPN は公衆回線の特定ルータ間を暗号化し、特定ユーザが排他的に回線を利用する仮想的私設網である。

7. 次の空欄 [a][b][c] に入る適切な語の組み合わせを、(1)-(4)の中から選びなさい。

電子政府システムを従来のシステムと比較すると、認証局が [a] に、公開鍵証明書が [b] に、デジタル署名が [c] に対応すると言うことが出来る。

- (1) [a] 役所 [b] 印鑑証明書 [c] 印鑑
- (2) [a] 役所 [b] 電子契約書 [c] 印影
- (3) [a] 政府 [b] 契約書 [c] 印鑑
- (4) [a] 省庁 [b] 印鑑証明書 [c] 署名

8. 次の文章で間違っているものを選びなさい。

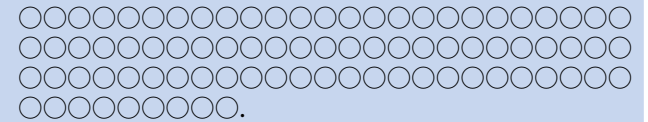
(1) PKI として仕組みが確立するためには、ユーザとその公開鍵の関係が明確にされねばならない。

(2) 電子署名法が施行されたのは平成 13 年である。

(3) 電子政府の構想には電子商取引の推進がある。

(4) PKI の仕組みの最重点項目は契約書などの文書の暗号化にある。

第 13 章 マルチメディア端末としてのコンピュータ



- 13.1 アナログとデジタル
- 13.2 マルチメディア圧縮技術
- 13.3 音楽配信の仕組み
- 13.4 動画配信・共有の仕組み
- 13.5 動画再生の仕組み
- 13.6 テレビとパソコン

13.1 アナログとデジタル

13.2 マルチメディア圧縮技術

13.3 音楽配信の仕組み

13.4 動画配信・共有の仕組み

13.5 動画再生の仕組み

13.6 テレビとパソコン

第 14 章 モバイル技術

ほげほげ

- 14.1 スマートフォン
- 14.2 タブレット端末
- 14.3 IC/RFID
- 14.4 NFC
- 14.5 電子財布
- 14.6 GPS

14.1 スマートフォン

ここでは、iPhone の発売以来、従来の携帯電話からのシフトが始まっているスマートフォンについて見ていきます。

スマートフォンとは

スマートフォンとは、多機能携帯電話のことを指します。従来から携帯情報端末 (PDA) というデバイスがありましたが、スマートフォンは携帯情報端末 (PDA) に携帯電話・通信機能を統合したものと表現されることもあります。表 1.27 に示すような汎用的な OS を搭載し、利用者が後からソフトウェアなどを追加できるようになっています。

スマートフォンは、インターネットとの親和性が高く、パソコンと同じような

ウェブブラウザによるウェブ閲覧や、電子メールの送受信などが可能です。

スマートフォン用の OS

現在は、iOS を搭載した iPhone と、Android を搭載した製品が主流となっています。今後は、Windows Phone 8 などでも登場してきます。それぞれのスマートフォン用の OS の特徴を表 x.x にまとめています。特に Android は OS のライセンスや開発環境が無償で提供されており一般の開発者が気軽にアプリケーションの開発が可能であるという特徴があります。

アプリケーション開発環境

Android OS 用のアプリケーションは

表 1.27 主なスマートフォンの OS

	解説
Android	Google 社が開発している Linux カーネルをベースとしたスマートフォン用の OS。OS のライセンス料金がからないという特徴がある。また、開発環境も無償で公開されており、一般の開発者が無料でアプリケーションの作成が可能。
iOS	Apple 社が開発した iPhone, iPad 用の OS。Mac OS X と共通するコンポーネントから構成されるが、バイナリ互換性はない。
Windows Phone	Microsoft 社が開発したスマートフォン用 OS。Windows Mobile プラットフォームを踏襲することなく、ゼロベースで開発がなされた。NT カーネル化、マルチコア CPU・SD カード・DirectX などのサポートが強化された。

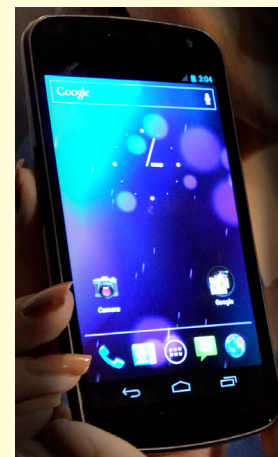


写真 1.34
Android OS4.0 搭載 Galaxy Nexus
出典：フリー百科事典
『ウィキペディア (Wikipedia)』



写真 1.35 iPhone 4S
出典：フリー百科事典
『ウィキペディア (Wikipedia)』



写真 1.36 iPhone 4S
出典：フリー百科事典
『ウィキペディア (Wikipedia)』

Java 言語で記述するため、開発に必要なのは Java の開発環境である JDK (Java Development Kit) です。また、開発を効率に行うため Eclipse という統合開発環境を用います。両者とも無償で提供されています。

iOS 用のアプリケーションは Objective-C という言語を用います。開発には Mac 上の Xcode という開発環境

を使います。iOS 用のアプリケーション開発には開発者登録が必要です。

また、Windows Phone 8 用のアプリケーションは、Windows 8 のアプリケーション開発と同じ Visual Studio などの環境で開発が行えます。

14.2 タブレット端末

ここでは、iPad の登場より一気に注目を集めるようになったタブレット端末について見ていきます。

タブレット端末とは

タブレット端末は、薄型の板状の本体にタッチパネル式のディスプレイ備えたコンピュータです。2010 年ごろから iPad が登場し、2011 年には様々なメーカーから Android を搭載した製品がリリースされるようになりました。

タブレット端末は、写真 1.37 や写真 1.38、写真 1.39 に示すように、前節で示したスマートフォンと同様に iOS を搭載する iPad と Android を搭載する製品が主流となっています。

タブレット端末の特徴

タブレット端末は、パソコンとスマートフォンの中間的な存在ですが、パソコンと比べた場合には、まず持ち運びやすいという点を挙げられます。タブレット端末の大きさは A5 サイズから A4 サイズ程度です。キーボードが具備されていないためネットブックなどの小型のパソコンと比べてもよりも薄くて軽く、気軽に持ち歩いて利用できます。

また、スマートフォンと比べた場合には、インターネット経由でアプリケーションをダウンロードして機能を拡張できる点はスマートフォンと同様です。タ

ブレット端末は、画面が大きいと文字が見やすく、画面内に表示されるボタンやキーボードが押しやすいという利点を挙げられます。そのため、Web サイトの閲覧が効率的に行えたり、ビジネス文書の確認・修正作業などを容易に行えるという特徴があります。

MDM と BYOD

モバイルデバイスマネジメント (MDM) は、スマートフォンやタブレットなどの端末を遠隔から管理する技術です。デバイス紛失などのデータ漏洩へつながる大きな脅威に対して、デバイス管理、デバイスロック、データを消去などを含めた運用管理を行うことが可能となります。

また、近年、企業の情報処理分野で大きな話題に BYOD (ブリング・ユア・OWN・デバイス) があります。BYOD とは、私物のスマートフォンやタブレット端末を業務で利用することを指しています。このような BYOD に対してある程度のセキュリティを確保するためにも必要となる機能が前述の MDM になります。

電子書籍専用タブレット端末

いままで見てきたものは、汎用性の高い一般的なタブレット端末ですが、電子書籍を読むことに特化した専用端末としてのタブレット端末もあります。電子書

籍用のタブレット端末としての特徴は、電子書籍を読むことに特化した画面表示にあります。汎用的なタブレット端末が液晶画面を採用しているのに対して、電子書籍用のタブレット端末は電子ペーパーを採用しています。バックライトを必要とせず、表示内容を変更するときだけ電気を使うので消費電力が少なくてす

み、液晶と違って画面が光らないので炎天下でも見やすく、目にもやさしいとされています。電子書籍に関しては、15.5 節に詳述しますが、専用端末の名前だけ挙げると、Amazon 社の「Kindle」、ソニー社の「Reader」、楽天の kobo などがあります。



写真 1.37 Android 搭載 GALAXY Tab
出典：フリー百科事典『ウィキペディア (Wikipedia)』



写真 1.38 iPad2
出典：フリー百科事典『ウィキペディア (Wikipedia)』



写真 1.39 Android 搭載 NTT docomo ARROWS Tab LTE F-01D
出典：フリー百科事典『ウィキペディア (Wikipedia)』

14.3 IC/RFID

ここでは、iPad の登場より一気に注目を集めるようになったタブレット端末について見ていきます。

タブレット端末とは

14.4 NFC

14.5 電子財布

14.6 GPS

第 15 章 コンピュータとネットワークの 最新技術

ほげほげ

15.1 クラウドコンピューティング

15.2 クライアント仮想化技術

15.3 SNS

15.4 HTML5

15.5 電子書籍

15.6 スーパーコンピュータ京

15.1 クラウドコンピューティング

15.2 クライアント仮想化技術

15.3 SNS

15.4 HTML5

15.5 電子書籍

15.6 スーパーコンピュータ京

演習問題

11Q 5H









