平成 17 年度秋期

基本情報技術者 試験資料

目　次

第 1 章　情報の基礎理論

№1　データの表現と単位　　★☆☆☆☆

①2 進数と基本単位

1) **2 進数**(binary numeral)

基数 2 で表した数

1 桁を表すのに 0 と 1 を用いる

・桁上げ(carry)　… ある桁の 1 に 1 を足すと桁上がりが生じる

・桁借り(borrow)　… ある桁の 0 から 1 を減算するとき、上位の桁から桁借りをする

2) **ビット**(bit , binary bit)

2 進数の 1 桁をビットと呼ぶ

コンピュータ内部の情報表現の最小単位

n ビット → 2^n の表現

ビット数	2 進数
1	2 (=2^1)
2	4 (=2^2)
3	8 (=2^3)
4	16 (=2^4)
8	256 (=2^8)
16	65,536 (=2^{16})
24	16,777,216 (=2^{24})

3) **バイト** (byte)

8 ビットを 1 バイトとした単位

1 バイト = 2^8 = 256 種類

コンピュータによるデータ処理の基本単位

4) **ワード** (word)

目的に合わせた数バイトを 1 単位とするもの

ⅰ. 16 ビット(2 バイト) = ワード

32 ビット(4 バイト) = ロング・ワード

ⅱ. コンピュータが一度に処理できるデータのビット数 = ワード

8, 16, 32, 64 ビットが単位

②**16 進数** (hexadecimal numeral)

0～15 を 1 桁で表現

0～9，A～F を使用

③補助単位

	単位記号	大きさ	指数表現
大	T (テラ)	×1 兆	= 10^{12}　≒2^{40}
↑	G (ギガ)	×10 億	= 10^{9}　≒2^{30}
	M (メガ)	×100 万	= 10^{6}　≒2^{20}
	k (キロ)	×1000	= 10^{3}　≒2^{10}
	m (ミリ)	÷1000	= 10^{-3}
	μ (ﾏｲｸﾛ)	÷100 万	= 10^{-6}
↓	n (ナノ)	÷10 億	= 10^{-9}
小	p (ピコ)	÷1 兆	= 10^{-12}

※基本情報技術者試験において、1 キロバイト=1000 バイトが暗黙のルール

№2　基数変換　　★★★★★

①基数と重み

1) **重み** (weight)

各数字位置の係数

各位置の数字にこの係数を掛けたものを足し算すると値が得られる

例) 4 桁の 10 進数 1234

重み	1000 =10^3	100 =10^2	10 =10^1	1 =10^0
10 進数	1	2	3	4

2) **基数記数法** (radix system)

ある桁とその 1 つ下位の桁との重みの比を正の整数にしたもの

3) **基数** (radix)

各桁の重みの基本となる数

②10 進数から n 進数への変換

ⅰ．まず基数で割り、
ⅱ．商と余りを求めていく。
ⅲ．求めた余りを、求めた順の逆に読む。

1) 10 進数→(2 進数→)16 進数

ⅰ．10 進数を 2 進数に基数変換する
ⅱ．右から 4 桁ずつ区切る
ⅲ．各区切りごとに基数変換する

2) 10 進小数からの変換

ⅰ．10 進小数を基数倍し、整数部を取り出す
ⅱ．小数部が 0 でなければⅰを行う
ⅲ．小数部が 0 になるまでⅰを行う
ⅳ．取り出した整数部を順に並べる

例) 10 進数の 0.625 を 2 進数に変換

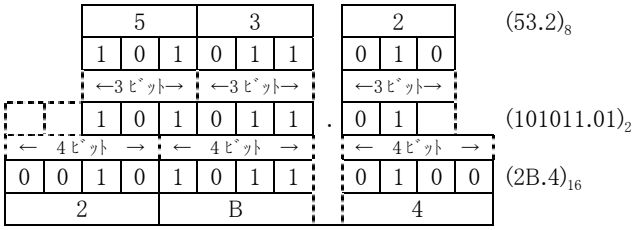
$$0.625 \times 2 = 1.25 \rightarrow 0.25 \times 2 = 0.5 \rightarrow 0.5 \times 2 = 1$$

↓ 1　↓ 0　↓ 1　→ 0.101

③*n* 進数から 10 進数への変換

各桁の値に重みを乗じた値を求めて合計する

④2 進数，8 進数，10 進数の変換



№3　負の整数の表現　　★★★☆☆

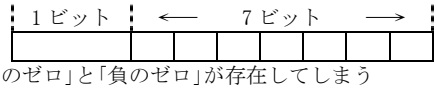
①符号ビットによる方法

2 進数の先頭の 1 桁を**符号ビット**(sign bit)とする

0 → 正
1 → 負

先頭を除く残りの桁 → 絶対値

例) 8 ビット表現の場合



※「正のゼロ」と「負のゼロ」が存在してしまう

②補数による表現

1) **補数** (complement)

与えられた数を、ある決められた数から引くことによって得られる数

- ⅰ．基数の補数
  n 進数の *n* に対する補数
- ⅱ．減基数の補数(基数−1 の補数)
  n 進数の *n*−1 に対する補数

※コンピュータのデータ表現は負の数を表現するのに補数を用いるのが一般的

2) 10 進数の補数

ⅰ．**9 の補数**
ある数値の各桁を 9 から引いた値

ⅱ．**10 の補数**
9 の補数＋1

例) 3 桁の 10 進数 123
9 の補数 → 876
10 の補数 → 877

3) 2 進数の補数

ⅰ．**1 の補数**
ある数値の各桁の値から 1 引いた値
＝ すべてのビットを反転する

ii . **2 の補数**
1 の補数＋1

例) 2 進数$(00101110)_2$
1 の補数 → $(11010001)_2$
2 の補数 → $(11010010)_2$

4) 補数による負の整数の表現

負の整数 → 2 の補数

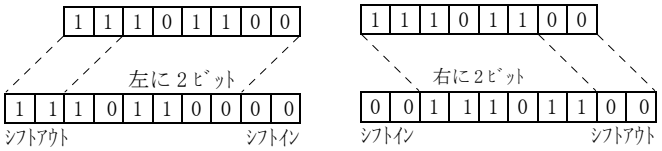
補数表示の固定小数点数 範囲 $-(2^{n-1}) \leqq x \leqq (2^{n-1}-1)$

№4　シフト演算　　★★★☆☆

①**シフト演算** (shift operation)
2 進数のビット全体を左右にずらす演算
2 のべき乗倍の乗除算に利用可能

1) **論理シフト** (logical shift)
ビット全体を単純に左右にシフトする演算
・実行規則
ｉ. 全ビットをシフトする
ii. シフトの結果、はみ出したビットは切り捨てる（**シフトアウト**）
iii. シフトの結果、空いたビット位置に 0 を格納　（**シフトイン**）

例)



2) **算術シフト** (arithmetic shift)
数値演算に利用するために、負数を 2 の補数で表現する形式に対応したシフト演算
ｉ. 符号ビットはシフトしない
ii. シフトの結果、はみ出したビットは切り捨て
iii. シフトの結果、空いたビット位置には、次の値を格納
左シフト → 0
右シフト → 符号ビットと同じ値

3) 2 のべき乗の乗除算
2 のべき乗以外の掛け算
途中の状況を保存しておき、加算する
例) 10 倍 → 2 倍 + 8 倍

4) **回転シフト (循環シフト)**
あふれたビット列は反対側の空いた部分に送られる。

№5　固定小数点と浮動小数点　　★★★☆☆

①**固定小数点数** (fixed-point number)
・小数点位置を固定した記数法
・ワードごとに、**最下位ビット**(**LSB** : Least Significant Bit)の右側を小数点位置として整数を表現
・負数は 2 の補数表現を使うことが多い
1 ワード=16 ビットの場合、**最上位ビット**(**MSB** : Most Significant Bit)は符号ビット、15 ビットで数値部を表現
・最上位ビットの右側を小数点位置として小数の表現も可能

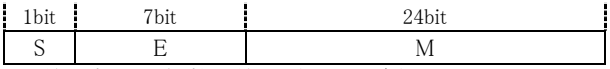
②**浮動小数点数** (floating-point number)
・実数型データを表現するときに使用
・科学技術計算など高精度の計算を要する場合の表現方法
・数値を符号、**仮数**、**底**、**指数**で正規化した表現で 2 進数に割り当てて表現

123 → 0.123 (仮数 mantissa) × 10 (底 base) ^{3} (指数 exponent)

1) **正規化**
仮数の値がある一定範囲に収まるように調整し、指数で実際の大きさを調整する記数法

2) 汎用コンピュータの浮動小数点数の形式

例) 32bit 表現の例



S：符号部 … 仮数の符号（0:正 1:負）

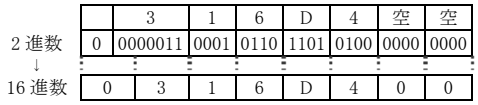
E：指数部 … 16 を基数とした 2 の補数

M：仮数部 … 1 以下の表現。16 進数の 0.XXX の小数部

$365.25 \xrightarrow[\text{基数変換}]{\text{16進数}} (16D.4)_{16}$

↓ 正規化

$(0.16D4)_{16} \times 10^3$



3) 指数部の表現

指数部 7bit

→ 指数≦ $2^7-1 = 127_{10}$

・**バイアス** (bias)

+64 (40_{16})を実指数に加える ＝「バイアスをかける」

・**イクセス表現** (excess expression)

バイアスをかけ、0～63 を負、64～127 を正として符号なしで正負の指数を示す

//

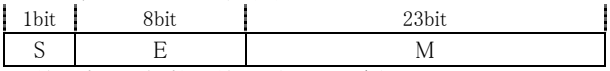
実指数 ＝ 指数部-64

例) 指数部 67 → 指数 = $+3_{(乗)}$

指数部 61 → 指数 = $-3_{(乗)}$

4) IEEE の浮動小数点数の形式

※IEEE…米国電気電子技術者協会



S：符号部 …仮数の符号（0:正 1:負）

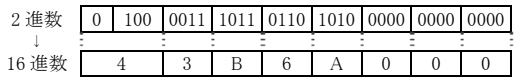
E：指数部 … 2 を基数として+127 する（バイアス 127）

M：仮数部 …（仮数-1)の 2 進小数（1.XXX…の小数部）

$365.25 \xrightarrow[\text{基数変換}]{\text{2進数}} (101101101.01)_2$

↓ 正規化

$(1.0110110101)_2 \times 2^8$



№6 精度と誤差 ★★★★☆

①有効数字

・真の値の近似値として信頼しうる範囲の数字

・推定される誤差がその最下位桁の範囲内にとどまる数字

○**絶対誤差** = |真値 － 近似値|

○**相対誤差** = |絶対誤差 / 真値|

②近似値表現

$(0.1)_{10}$

↓

$(0.0001100110011\cdots)_2$ 循環小数

2 進数で 10 進数を表現する以上、その仕組みの上で誤差がある

→ **近似値** (approximation)

1) **丸め誤差** (rounding error)
「丸め」により生じる誤差
※丸め…決められた表現範囲に収まるように、最下位桁からあふれた数を削除すること。四捨五入，切捨て，切り上げなど。

2) **打切り誤差** (truncation error)
計算処理が完全に終わる前に、規則にしたがって打ち切ることによって発　生する誤差

③浮動小数点の演算精度
1) **桁落ち**
ほぼ等しい値の浮動小数点同士の減算を行うと、演算結果の有効桁数が極端に減少する現象

2) **情報落ち**
浮動小数点の演算は、2 値の指数部の指数の大きい方の値にそろえる
↓
絶対値の大きさに極端に差がある 2 値の演算を行うと、小さいほうの値では仮数部の値が大きく右にずれる
↓
小さいほうの値の表現すべきビット情報が欠落してしまう

③**算術あふれ** (arithmetic overflow)
限られたビット数で表現する以上、限界がある

1) **オーバーフロー** (**overflow**)
非常に大きな値、または非常に小さい値同士の掛け算を行ったとき指数部の表現範囲の上限を超えてしまう現象

2) **アンダーフロー** (**underflow**)
0 に非常に近い、つまり絶対値が非常に近い値同士の掛け算を行ったとき、指数部の表現範囲の下限を超えてしまう現象

№7　10 進数と数値表現　　★★☆☆☆

①**BCD コード (Binary Coded Decimal code)**
・10 進数の 0～9 に対応した 4bit の 2 進数で 10 進数の各桁を表現する表現形式
・**2 進化 10 進コード**ともいう
・10 進数の桁数にあわせて 2 進数の桁も変化 (=可変長形式)

10 進数	2 進数	BCD コード
0	0000	0000
1	0001	0001
2	0010	0010
3	0011	0011
4	0100	0100
5	0101	0101
6	0110	0110
7	0111	0111
8	1000	1000
9	1001	1001
10	1010	0001 0000
11	1011	0001 0001
12	1100	0001 0010

②**ゾーン 10 進数**
10 進数の 1 桁を 1 バイト(8bit)で表現
┌上位 4bit → **ゾーン部**
└下位 4bit → 数値部 (BCD コードと同じ表現)

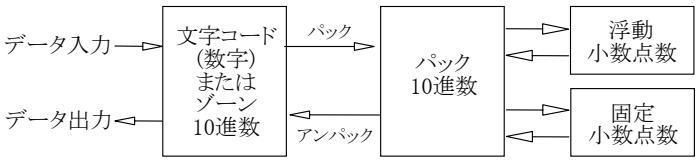
1) 上位 4bit
→ ゾーン部
JIS コード形式 = $(0011)_2$
EBCDIC コード形式 = $(1111)_2$
最下位桁のゾーン部 → 符号部
┌正 … $(1100)_2$
└負 … $(1101)_2$

10 進数	JIS コード形式	EBDIC 形式
123	0011 0001 0011 0010 1100 0011 ｿﾞｰﾝ 1 ｿﾞｰﾝ 2 符号 3 (31 32 C3)$_{16}$	1111 0001 1111 0010 1100 0011 ｿﾞｰﾝ 1 ｿﾞｰﾝ 2 符号 3 (F1 F2 C3)$_{16}$
−123	0011 0001 0011 0010 1101 0011 ｿﾞｰﾝ 1 ｿﾞｰﾝ 2 符号 3 (31 32 D3)$_{16}$	1111 0001 1111 0010 1101 0011 ｿﾞｰﾝ 1 ｿﾞｰﾝ 2 符号 3 (F1 F2 D3)$_{16}$

③**パック 10 進数**
・1 バイトで 2 桁の 10 進数を表現
・符号は最下位桁の下位 4bit
正 … $(1100)_2$　　負 … $(1101)_2$
・ゾーン 10 進数に比べ使用するバイトが少ない
・ゾーン部がないため演算が簡単

偶数桁 n 桁の場合 $\frac{n}{2}+1$バイト
奇数桁 n 桁の場合 $\frac{(n+1)}{2}$バイト

④ゾーン 10 進数とパック 10 進数の使い分け



・データの入出力には、文字コードと同じ 1 バイト単位のゾーン 10 進数が便利

№8　文字データの表現　　★★★☆☆

①文字コード

サイズ	コード名称	概要
1byte	**ASCII コード**	英数字,記号,制御コードで 7bit コード
	ISO コード	ASCII をもとにした国際規格の 7bit コード
	JIS コード	ISO コードを基にした半角文字コード
		JIS7 単位符号：ISO コードとほぼ同じコード
		JIS8 単位符号：半角カタカナを追加
	EBCDIC	汎用ｺﾝﾋﾟｭｰﾀのデファクトスタンダードである 8bit ｺｰﾄﾞ
2byte	**JIS 漢字コード**	英数字,記号,ひらがな,カタカナ,漢字を表す 16bit コード
	ｼﾌﾄ JIS コード	JIS 漢字コードの表現領域をずらした 16bit コード
	Unidoce	世界中の文字を 2 バイトで表す国際規格 16bit コード
ﾏﾙﾁﾊﾞｲﾄ	**日本語 EUC**	UNIX で使用される 1～3 バイトのマルチバイトコード

②EBCDIC (Extended Binary Code Decimal Interchange Code)

拡張 2 進化 10 進コード
IBM 社が自社用汎用コンピュータ用に設計
↓
事実上の標準 (de fact standard：デファクトスタンダード)
世界標準の文字コード

③ASCII コード

ASCII：American national Standard Code for Information Interchange
1962 年 ANSI(American National Standards Institute：米国規格協会)が制定
7bit＋**パリティビット**(1bit) = 8bit
→誤りを検査するビット

④JIS コード

JIS：Japanese Industrial Standards（日本工業規格）
ISO コードをもとにした日本の国内規格
7bit … ローマ字用 7 単位符号
8bit … ローマ字,カタカナ用 8 単位符号
1 バイトコード

・JIS 漢字コード
2 バイト表現でひらがな・漢字をカバー

・区点コード
1 バイト目を「区」、2 バイト目を「点」とし 94 区 94 点のマトリクスに文字を収めた表現形式

⑤シフト JIS コード

・Microsoft 社などによって考案された文字コード
・コードの重複がないため、1 バイトコードと混在しても区別可能
・日本標準コード

⑥日本語 EUC

EUC：Extended Unix Code (拡張 UNIX コード)
UNIX で日本語文字を扱うためのコード
4 つの文字コードセットからなる

1byte 目	2byte 目	3byte 目	文字コード
21_H～$7E_H$	—	—	JIS X0201 (英数記号)
$A0_H$～FF_H	$A0_H$～FF_H	—	JIS X0208 (1/2byte 目ともに 80_H を減じる)
$8E_H$	21_H～$7E_H$	—	2byte 目が JIS7 単位の半角カタカナ
$8E_H$	$A0_H$～FF_H	$A0_H$～FF_H	2/3byte 目が JIS X0212 (補助漢字)

⑦**Unicode**

全世界の文字を 2byte で表示
1995 年に JIS 規格(JIS X0221)になった

⑧メタ文字（metacharacter）

特定の表現機能を持った文字

メタ文字	意味
.	任意の 1 文字を表す
(文字列)	“(”と”)”で囲まれた文字列を一つのパターンとして扱う
+	直前の文字またはパターンの 1 回以上の繰返しを表す
*	直前の文字またはパターンの 0 回以上の繰返しを表す
?	直前の文字またはパターンが 0 回または 1 回現れることを表す
¥	後続する文字をメタ文字ではなく、文字そのものとして扱う
|	文字またはパターンの選択を表す
[m-n]	m から n までの連続した文字の群から任意の 1 文字の選択を表す

⑨画像データ

・**ラスタ表現**

画像をマトリックス状に区切り(標本化)、それを画像(ピクセル)の集まりとして捉え(量子化)、これを符号化して表現する方法

・**ベクタ表現**

画像を、直線･円などの幾何学的な要素の集まりとして表現する。図形を縮小･拡大や変形しても画像が荒れることがない

№9　論理演算と論理回路　　★★★★☆

①論理演算と論理回路

1) **論理演算 (logical operation)**

2 進数が表現できる 1 と 0 を”真”と”偽”にあてはめ、真偽の値で行う演算のこと。

2) **論理回路 (logical circuit , logical device)**

論理演算を行う回路

3) 論理演算の基本

論理演算	演算記号
論理積（AND）	∧　・
論理和（OR）	∨　＋
否定（NOT）	－　¬
排他的論理輪（EOR, XOR）	∀　⊕
否定論理積（NDAND）	―
否定論理和（NOR）	―

・優先順位
高　NOT
　　AND
低　OR

・表現

ⅰ. **真理値表**（truth table）
入力値と出力値の組み合わせの表

ⅱ. **ベン図**（Venn diagram）
演算結果を集合(set)の相互関係で図示したもの

ⅲ. **MIL 記号**
米国軍用規格(Military standard)の電子回路の記号

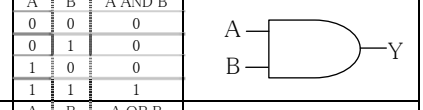
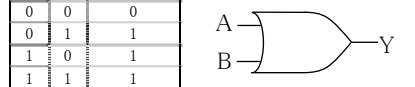
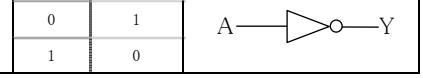
②基本回路

論理積，論理和，否定の演算回路

1) **AND 回路 (論理積)**
入力値 A, B が共に真である場合のみ、真

2) **OR 回路 (論理和)**
入力値 A, B のいずれかが真であれば、真

3) **NOT 回路 (否定)**
入力値の逆を出力

	真理値表			MIL 記号
AND 回路	A	B	A AND B	
	0	0	0	
	0	1	0	
	1	0	0	
	1	1	1	
OR 回路	A	B	A OR B	
	0	0	0	
	0	1	1	
	1	0	1	
	1	1	1	
NOT 回路	A		NOT A	
	0		1	
	1		0	

③基本回路の組み合わせ

1) **EOR 回路**

排他的論理和

入力値 A, B の値が不一致の場合に、真

2) **NOR 回路**

否定論理和

OR 回路の出力側に NOT 回路を組み合わせる

3) **NAND 回路**

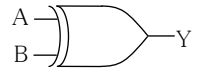
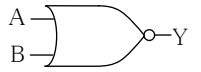
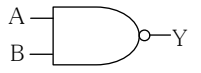
否定論理積

AND 回路の出力側に NOT 回路を組み合わせる

4) **NXOR**

一致回路

入力値 A, B の値が等しい場合に、真

	真理値表			MIL 記号
EOR 回路	A	B	A EOR B	
	0	0	0	
	0	1	1	
	1	0	1	
	1	1	0	
NOR 回路	A	B	A NOR B	
	0	0	1	
	0	1	0	
	1	0	0	
	1	1	0	
NAND 回路	A	B	A NOR B	
	0	0	1	
	0	1	1	
	1	0	1	
	1	1	0	
NXOR 回路	A	B	NXOR	
	0	0	1	
	0	1	0	
	1	0	0	
	1	1	1	

④**ド・モルガンの法則**

1) 論理積の否定は、それぞれの否定の論理和に等しい

A	B	A･B	$\overline{A \cdot B}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

$\overline{A}$	$\overline{B}$	$\overline{A}+\overline{B}$
1	1	1
1	0	1
0	1	1
0	0	0

2) 論理和の否定は、それぞれの否定の論理積に等しい

A	B	A+B	$\overline{A+B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

$\overline{A}$	$\overline{B}$	$\overline{A}\cdot\overline{B}$
1	1	1
1	0	0
0	1	0
0	0	0

⑤**加算回路**

・**加算器** (adder)

1 桁の 2 進数の加算を行う加算回路

半加算器 (half adder)

下位からの桁上がりを意識しない

全加算器 (full adder)

下位からの桁上がりを意識したもの

1) **半加算器**

1 桁の 2 進数の加算パターン

0 + 0 = 0

0 + 1 = 1

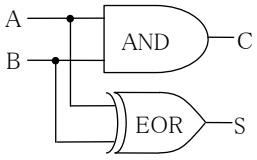
1 + 0 = 1

1 + 1 = 10

C：Carry (桁上げ)

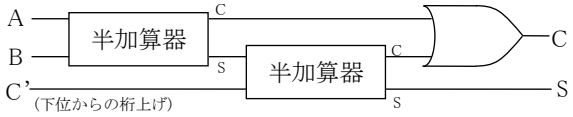
S：Sum (演算結果)

A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0



2) **全加算器**

下位からの桁上がりを演算結果 S に加算する



⑥**順序回路** (sequential circuit)

過去の入力信号を記憶しておき、過去の入力値と現在の入力値によって出力値が決まる回路。

※組み合わせ回路 (combinational circuit)

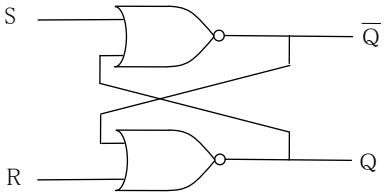
…出力値が入力値によって定まる回路

1) **フリップフロップ回路 (flip-flop circuit)**

順序回路の代表的なもので、記憶装置で利用。

例) RS フリップフロップ回路

S	R	Q	意味
0	0	0 or 1	直前の入力状態を保持
0	1	0	リセット
1	0	1	セット
1	1	−	禁止



№10　状態遷移と形式言語　　★★★★★

①**状態遷移 (state transition)**

…入力とそのときの内部状態に依存して状態が変化すること

1) **状態遷移図**

状態遷移の様子を有効グラフで図示したもの

※有効グラフ (directed graph)

接点の集合と、接点を結ぶ向きのある辺(矢印)の集合からなる図

2) **有限オートマトン (finite automaton)**

※オートマトン(automaton) … コンピュータなどを、数学的な観点からモデル化したもの

・有限状態部，入力テープ，読み取りヘッダからなり、出力はない単純なモデル
・受理状態と非受理状態の 2 つに区分される

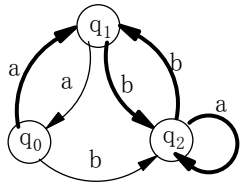
↓

〈遷移状態図で表現〉

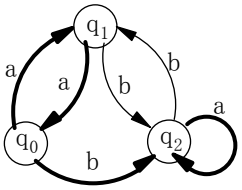
受理状態は◎印の接点で図示する

受理の判断は、初期状態から入力をたどり、受理状態で停止するか否かで行う。

q_0 が始点、q_2 が終点(受理状態の場合)



abab は受理されない



aaba は受理される

②**形式言語**

言語の文字記号のみが対象

音や意味内容には関与しない

1) **BNF 記法**

・プログラム言語 ALGOL60 の構文記述用に開発された記法

別名：**バッカス記法** (Backus normal form)

バッカス･ナウア記法 (Backus Naur form)

・文字列などの長さを決められない要素を、**メタ記号**(metasymbol)で書き換え規則を表現

【メタ記号】

〈○○〉 ：概念○○を表す

::= ：左辺の記号が、右辺の記号で定義されることを表す

｜ ：「または」を表す

例) 英数字という概念

〈英数字〉::=〈英字〉|〈数字〉 …英数字は英字または数字

〈英字〉::=a|b|c|…|x|y|z …英字は a～z のいずれかである

〈数字〉::=0|1|2|…|7|8|9 …数字は 0～9 のいずれかである

2) 構文図法

BNF 記法に対応する視覚的表現

・概念　　… 四角
・文字記号 … 丸
・つながりの規則 …矢印
・その他　　a の 0 回以上の繰り返し　… {a}
　　オプション　　… [b]

№11　確率と統計の基礎　　★★★☆☆

①順列と組合せ

1) **順列 (permutation)**

n 個の中から r 個を選ぶとき、並び番に並べたものの総数

順列　$_nP_r = \dfrac{n!}{(n-r)!} = n\times(n-1)\times\ldots\times(n-r+1)$

2) **組合せ (combination)**

n 個の中から r 個を選ぶとき、並び番を考えないものの総数

組合せ　$_nC_r = \dfrac{n!}{(n-r)!r!} = \dfrac{{}_nP_r}{r!}$

②**確率 (probability)**

ある事象が起こりうる可能性の割合

・起こりうるすべての場合の数 n
・事象 A が起こる場合の数　　a

事象Aが起こる確率　$P = \dfrac{a}{n}$

(起こる確率)+(起こらない確率) $= 1$

例題) サイコロを 2 回振って、少なくとも 1 回は 1 の目が出る確率

解答) 2 回とも 1 でない確率は、$\dfrac{5}{6}\times\dfrac{5}{6} = \dfrac{25}{36}$

全体(1)から減算すると、　$1 - \dfrac{25}{36} = \dfrac{11}{36}$

よって、1 回は 1 の目が出る確率は、$\dfrac{11}{36}$

※マルコフ過程

次の状態への推移確率が現在の状態によってのみ定まり、過去の状態に依存しない過程のこと

③データの中心を表す代表値

代表値		意　味
平均値	(mean)	すべてのデータの合計を、データ個数で除した値
中央値	**メジアン** (median)	データを昇順(降順)に整列したとき、中央に位置する値
最頻値	**モード** (mode)	データの出現頻度(度数)が最も多い値

④データのばらつきを表す特性値

分散度	計算方法
偏差	変量 － 平均値
偏差平方和	全てのデータの偏差の 2 乗の合計
分散	偏差平方和÷データ個数
標準偏差	$\sqrt{}$分散

・分散 (bariance)
　… データのばらつきを表す
・**範囲 (range：レンジ)**
　… 最大値と最小値の差

⑤正規分布と標準化

1) **正規分布 (normal distribution)**

連続する値をとる確率分布

特徴：平均値が大きく、平均値に対し左右対称で釣鐘形状をとる
　　平均と分散が決まれば、確率に関する特性を規定できる

i．**確率変数** (probability variable)
変数 x がある値をとる確率が決まっているときの x をいう

ii．**確率分布** (probability distribution)
変数 x に対する確率の関係

2) 標準正規分布表の利用

・標準化

確率変数 x を変換して、平均 0，分散 1 の確率変数 U をつくる

$U = \dfrac{(x-\mu)}{\sigma} = \dfrac{(\text{データ値} - \text{平均値})}{\text{標準偏差}}$

第 2 章　ハードウェア

№1　コンピュータの構成要素　　★☆☆☆☆

①コンピュータの五大装置

コンピュータシステム ┬ **ハードウェア (hardware)**
　　└ **ソフトウェア (software)**

1) **入出力装置 (input-output unit)**
　入力装置(input unit)と出力装置(output unit)を合わせたもの

2) **中央処理装置 (Central Processing Unit：CPU)**
　演算装置と制御装置を合わせたもの

　ⅰ. **演算装置**
　　＝ **算術論理演算装置** (Arithmetic and Logic Unit：**ALU**)
　　　├ **算術演算装置** (Arithmetic Unit) … 数学的な計算を行う
　　　└ **論理演算装置** (Logic Unit)　　… 論理的な演算を行う

　ⅱ. **制御装置** (control unit)
　　コンピュータを動作させるプログラムを解析し、他の 4 つの装置に制御信号を出す

　　※命令デコーダ
　　…・命令を選択制御するために利用されるデコーダ(decoder：復号器)
　　・命令がどのような仕事を指示しているかを解読し、作業を実行するための信号を各制御装置に送る。
　　・制御装置に含まれる

3) **記憶装置 (storage)**
　├ **主記憶装置** (main storage)
　　メモリ(memory)と呼ばれる
　└ **補助記憶装置** (auxiliary storage)
　　主記憶装置の容量不足を補ったり、データを保存したりする

②コンピュータの種類

1) **パーソナルコンピュータ** (personal computer)
　パソコンと呼ばれる個人用コンピュータ
　デスクトップ，ノートブック，パームトップなど多種多様

2) **ワークステーション** (workstation)
　主に研究開発分野で使用されるコンピュータ
　EWS(Engineering WorkStation)とも呼ばれる

3) **汎用コンピュータ** (general purpose computer)
　主に大規模な基幹システムで使用されるコンピュータ
　メインフレーム(mainframe)とも呼ばれる

4) **オフィスコンピュータ** (office computer)
　企業などの事務処理を目的としたコンピュータ
　パソコンの高性能化で存在価値は薄れている

5) **スーパーコンピュータ** (supercomputer)
　最先端の技術を集大成した超高性能コンピュータ
　コストも大きく、利用目的は限定されている

6) **マイクロコンピュータ** (microcomputer)
　マイコンと呼ばれる制御用の組込みコンピュータ

№2　コンピュータの動作原理　　★★☆☆☆

①プロセッサのアーキテクチャ

　※アーキテクチャ … 設計方針，基本構造

プロセッサは記憶装置から命令を 1 つずつ順番に取り出して解読･実行する

　＝ **ノイマン型コンピュータ**

1) **プログラム記憶方式**

プログラム内蔵方式，**ストアードプログラム方式**(stored program)ともいう

実行するプログラムを、データとしてあらかじめ主記憶装置に記憶させておく方式。

↓

ハードウェアからプログラムが独立し、ソフトウェアによる汎用性が実現する

2) **逐次制御方式 (serial control)**

記憶させたプログラムの命令を 1 つずつ取り出して解読･実行する

②メモリアクセス

1) **アドレス (address)**

主記憶装置の記憶領域は、0 番地から順番にアドレスが付けられている

2) **バス (bus)**

装置間でデータを転送する機構

ⅰ. **アドレスバス** (address bus)　：アドレス番号を送るバス
ⅱ. **コントロールバス** (control bus)　：読書きの制御信号を送るバス
ⅲ. **データバス** (data bus)　：データを双方向に転送するバス

③**レジスタ (register)**

プロセッサの中で一時的な記憶に使う、ごく小さなサイズの記憶装置

1) **プログラムカウンタ**（**PC**：Program Counter）

命令アドレスレジスタ(instruction address register)ともいう

次に実行すべき命令のアドレスを保持する専用レジスタ

2) **命令レジスタ**（**IR**：Instruction Register）

主記憶装置から読み出した命令を解釈する為に保持する専用レジスタ

3) **汎用レジスタ**（**GR**：General Register）

専用の用途が決められていないレジスタ。

演算途中の値の一時的な保持や、指標レジスタ，アキュムレータなどに代用されることが多い。

4) **指標レジスタ**（index register）

インデックスレジスタとも言う

命令のアドレス指定でアドレス部に対して加算する値を保持する

5) **ベースレジスタ**（base register）

基底レジスタまたはベースアドレスレジスタ(base address register)

プログラムの先頭アドレスを保持するレジスタ

6) **アキュムレータ**（累算器：Accumulator）

演算の際の被演算数を保持し、演算後は演算結果が格納される

7) **フラグレジスタ**（**FR**：Flag Register）

命令の実行中に起こるある状態を保持する専用レジスタ

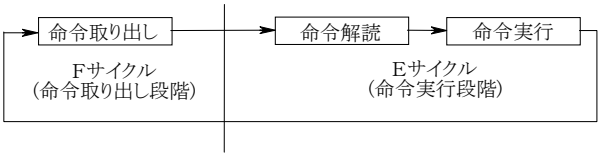
演算結果の正負やゼロ，桁上げの有無，オーバーフローの有無などを示す専用のビットで構成

8) **スタックポインタ**（**SP**：Stack Pointer）

後入れ先出し(Last-In First-Out:LIFO)のデータ構造であるスタックを制御するレジスタ

④**マシンサイクル**（machine cycle）

処理装置が 1 つの命令を取り出して実行する流れ



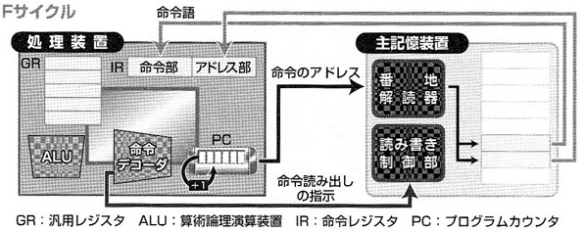
1) **命令取り出し段階 (instruction fetch cycle)：F サイクル**

次に実行する命令を主記憶装置から取り出す段階

ⅰ. **フェッチ**

プログラムカウンタの指すアドレスを元に、主記憶装置に記憶された命令を 1 語、命令レジスタに取り出す。

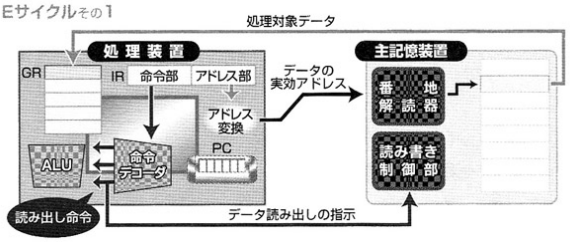
ⅱ．プログラムカウンタの更新
命令を 1 語取り出すたびに、プログラムカウンタの値を 1 増加させる。プログラムカウンタは、常に次取り出す命令のアドレスを示す。



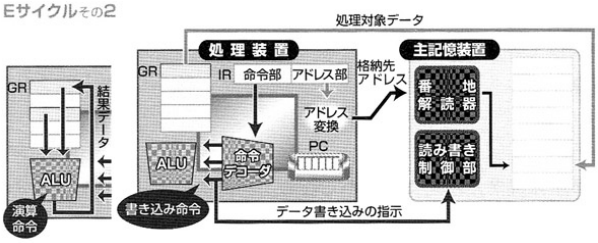
2) **命令実行段階 (execution cycle)：E サイクル**
命令レジスタに取り出した命令を実行する段階

ⅰ．命令の解読
取り出した命令の命令部は、命令解読器(**decoder**)に送られ解読される。解読の結果は制御信号に変換され、演算装置などに送られる。

ⅱ．命令の取り出し
アドレス部は、必要に応じてアドレス変換し、主記憶装置の指定アドレスからデータを取り出す。データは汎用レジスタ(GR)などに記憶される。



ⅲ．命令の実行
命令を実行する。
・演算命令 → 演算装置によって演算が実行され、その結果でいくつかのレジスタ内容が更新される
・書き込み命令 → 主記憶装置の指定アドレスとデータの転送が行われる。



№3　命令とアドレス指定　　★★★★☆

①命令と命令形式

1) **機械語 (machine language)**
コンピュータが唯一理解できるハードウェア固有の言語
プロセッサごとに異なる

命令部[オペコード：opecode] (operation part)
命令や演算を指示

アドレス部[オペランド：operand] (address part)
処理対象となるデータの主記憶装置上のアドレスを指定

2) **命令形式 (instruction format)**

0 アドレス命令	アドレス部を持たない形式
1 アドレス命令	アドレス部で処理対象を指定する形式
2 アドレス命令	2 つのアドレスで処理対象(1～2 個)を指定する形式
3 アドレス命令	処理対象や結果の格納先を 3 つのアドレスで指定する形式

②アドレス修飾と実効アドレス

1) **アドレス指定方式 (addressing)**

処理対象を示すアドレス部を、固定的な値ではなく、主記憶装置に記憶させてから計算によって求められるように指定する。

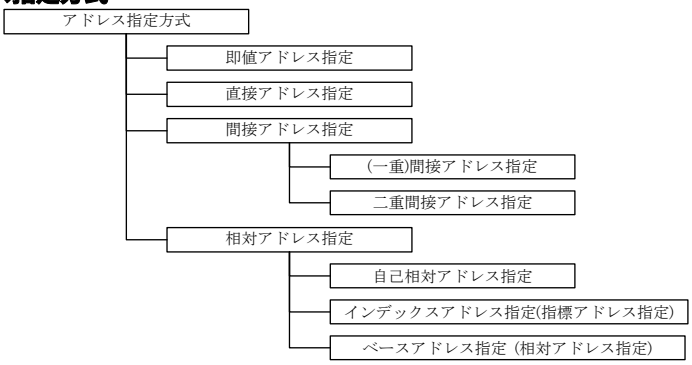
2) **アドレス修飾 (address modification)**

何らかの計算によってアドレスを求めること

↓

求めたアドレス値 = **実効アドレス**(effective address)(有効アドレス)

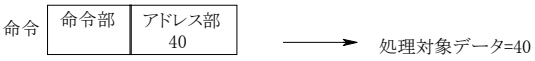
③アドレス指定方式



1) **即値アドレス指定方式** (immediate addressing)

・アドレス部の値がアドレス値ではなく、対象となるデータの値そのものである方式

・主記憶にアクセスする必要がない

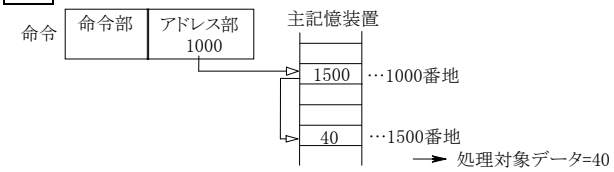


2) **間接アドレス指定方式** (indirect addressing)

・アドレス部の値が指定するアドレスに、処理対象データの記憶場所のアドレスが記憶されている方式

・主記憶に複数回アクセスするため、実行速度がやや低い

・利点：命令のアドレス部を固定したまま、処理対象アドレスを変更できる



3) **指標アドレス指定方式** (index addressing)

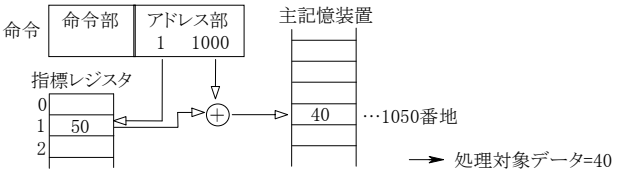
・**インデックスアドレス指定方式**とも言う

・アドレス部にアドレス値を指定する部分と指標レジスタ番号を指定する部分とがあり、

指標レジスタ内容＋アドレス値 = 実効アドレス

とする方式

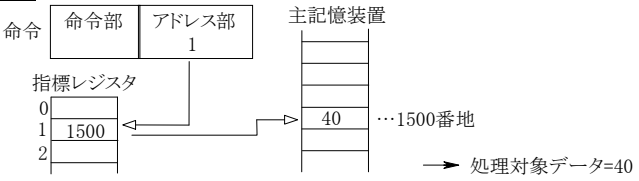
・利点：連続して記憶された配列型のデータを、指標レジスタの内容を増減して、同じ命令の繰り返しで処理できる



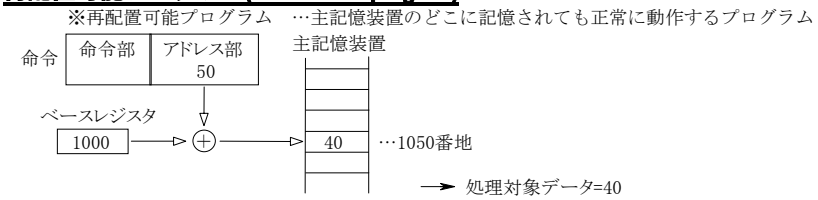
4) **レジスタアドレス指定方式** (register addressing)

・アドレス部の値が指標レジスタのレジスタ番号のみで、処理対象データを指定する方式

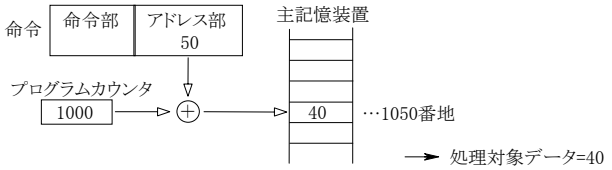
・利点：アドレス部がレジスタ番号だけなので命令語長が短い



5) **ベースアドレス指定方式** (base addressing)
・アドレス部の値にベースレジスタの内容を加算することで、実効アドレスとする方式
・ベースレジスタにはプログラムの先頭アドレスが記憶される
↓
プログラム内での相対位置を指定する
再配置可能プログラム(relocatable program)として実行可能



6) **相対アドレス指定方式** (relative addressing)
・プログラムカウンタの値をもとに、現在実行中の命令のアドレスをアドレス部の値に加算することで実効アドレスとする方式
・再配置可能プログラムを実行可能
・ベースレジスタや汎用レジスタを持たないプロセッサで使用



№4　プロセッサの性能と高速化技術　　★★★★★

①プロセッサの性能

1) **クロック周波数** (clock frequency)
デジタル回路で、動作のタイミングをとるための信号の周波数。メガヘルツ(MHz)の単位で表す

2) **MIPS (Million Instructions Per Second)**
1 秒間に実行できる命令数を百万命令単位で表したもの
1MIPS = 100 万命令/秒 = 1 μ 秒/命令

3) **CPI (Clock cycles Per Instruction)**
1 命令の実行に要するクロック数を表す

4) **命令ミックス** (instruction mix)
処理装置の命令の比率に重みをつけ、異なるコンピュータの性能を比較する手段

②逐次制御と先行制御

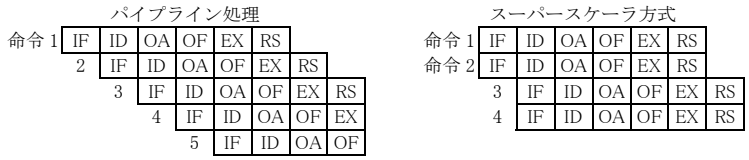
1) **逐次制御方式**
命令を 1 つずつ順番に実行する方式
この方法では、命令取り出し段階には、命令実行段階で動作する装　置が休止している状態がある
↓

2) **先行制御方式**
無駄な休止状態をなくすため、命令実行中に次の命令を取り出す

③**パイプライン処理 (pipeline processing)**
先行制御方式を発展させた方式
1 命令をさらに細分化し、同時実行させ、CPI 値を 1 に近づけること

ⅰ. [IF]　命令のフェッチ
ⅱ. [ID]　命令のデコード
ⅲ. [OA]　オペランドのアドレス計算
ⅳ. [OF]　オペランドのフェッチ
ⅴ. [EX]　命令の実行
ⅵ. [RS]　演算結果の格納

○**スーパースケーラ方式 (superscalar)**
パイプラインを複数本備えることで、CPI 値を 1 未満にする技術



※分岐予測

…あらかじめ命令が分岐するかどうかを予測した上で、パイプラインに分岐先の命令を流し込むことで処理の高速化を図る方法

④**並列処理 (parallel processing)**

複数のデータを並列に処理する方式

1) **SISD**（単一命令･単一データ：**シスド**：Single Instruction Single Data System）
1 命令で 1 データを処理。並列処理ではなく通常の処理

2) **SIMD**（単一命令･複数データ：**シムド**：Single Instruction Multiple Data System）
1 命令で複数のデータを同時処理。マルチメディア系での応用が顕著

3) **MISD**（複数命令･単一データ：Multiple Instruction Single Data System）
複数命令で 1 データを同時処理。分類上の概念（実機はない）

4) **MIMD**（複数命令･複数データ：**ミムド**：Multiple Instruction Multiple Data System）
独立した複数命令が別々のデータを処理。マルチプロセッサ

5) **マルチプロセッサ (multi-processor)**
複数のプロセッサを持つシステム
- **密結合マルチプロセッサシステム**
  1 つの OS で制御され、主記憶装置を共用しながら処理を行う
- **疎結合マルチプロセッサシステム**
  それぞれの OS に制御され、主記憶装置を個別に持つ

6) **アレイプロセッサ**（array processor）
- パイプライン処理の考え方を用い、主記憶装置付の多数の演算装置を格子状に配置したもの
- **ベクトルプロセッサ**(vector processor)ともいい、スーパーコンピュータで利用

⑤CISC と RISC

1) **CISC (Complex Instruction Set Computer)**
- **複合命令セットコンピュータ**
- プロセッサに高機能な命令を備えさせることで、機械語プログラミングの負荷軽減を狙う
- 回路そのものは単純化し、高機能な命令はプログラムとして記述したマイクロプログラム(microprogram)で実行

※マイクロプログラム

…・ファームウェア。ハードウェアとソフトウェアの中間的な存在
・CISC の内部で複雑な命令をプログラム的な手法で実現したもの
・一般のプログラムとは異なり低レベルの処理で実行されるプログラム

2) **RISC (Reduced Instruction Set Computer)**
- **縮小命令セットコンピュータ**
- プロセッサには単純な機械語命令だけを備え、複雑な命令はプログラムで実行
- 命令が単純なので、回路に組み込んでしまうワイヤードロジック(wired logic)で実装する

	CISC	RISC
命令の種類	多い	少ない
命令長	可変長	固定長
アドレス修飾の種類	多い	少ない
主記憶アクセス	多くの命令	ワード,ストアのみ
演算の対象	メモリ，レジスタ	レジスタ
汎用レジスタ数	少ない	多い
制御部の構成	マイクロコード制御	ワイヤードロジック制御

№5　半導体メモリと主記憶装置　　★★★★☆

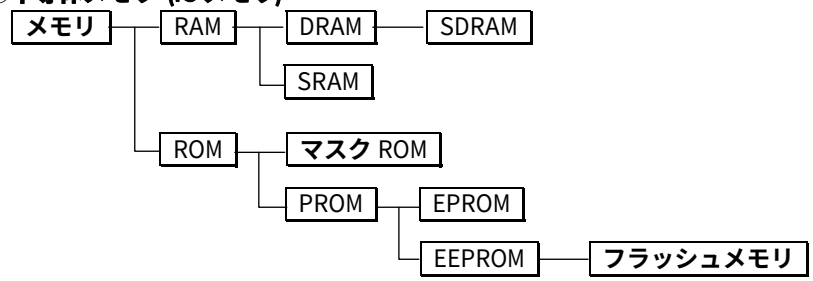
①**集積回路 (Integrated Circuit：IC)**
- 半導体を高度に集積したもの
- プロセッサや主記憶装置などに使われる
- 素子の種類によって分類

IC
- **バイポーラ型 (bipolar：双極性型)**
- **MOS 型 (Metal Oxide Semiconductor:金属酸化物半導体型)**

種類	速度	消費電力	コスト	用途
ﾊﾞｲﾎﾟｰﾗ型	高速	大	高	論理素子，高速記憶素子
MOS 型	低速	小	低	記憶素子

※MOS 型は、ユニポーラ(uni-polar：単極性)型ともいわれる
※MOS 型の一種で、**CMOS (Complementary MOS：相補性 MOS)**が最近の主流となっている

②半導体メモリ (IC メモリ)



1) **RAM (Random Access Memory)**
データの書き込み･読み出しが可能な半導体メモリ
揮発性メモリ(volatile storage)

比較項目	DRAM	SRAM
リフレッシュ	必要	不要
集積度	高い	低い
アクセス速度	低速（数百 ns）	高速（数 ns）
消費電力	多い	少ない
製造コスト	安い	高い
用途	主記憶装置	キャッシュメモリ
IC メモリ	MOS 型	CMOS 型、バイポーラ型

ⅰ. **DRAM (Dynamic RAM)**
MOS 型
コンデンサが蓄えた電荷量でビット情報を表現
ⅱ. **SRAM (Static RAM)**
バイポーラ型/MOS 型
フリップフロップ回路で、1 と 0 の状態を保持

2) **ROM (Read Only Memory)**
読み出し専用の半導体メモリ
不揮発性メモリ(non-volatile storage)

種類	書込み	消去	特徴
マスク ROM	×	×	製造時に書込まれたデータを使用
PROM	○	×	データの書込みは 1 回のみ可能
EPROM	○	○	紫外線によりデータの消去が可能
EEPROM	○	○	電気的にデータの消去が可能

ⅰ. **マスク ROM** (MASK ROM)
ⅱ. **PROM** (Programmable ROM)
ⅲ. **EPROM** (Erasable PROM)
ⅳ. **EEPROM** (Electrically Erasable PROM)
※**フラッシュメモリ**…電気的に内容を一括消去でき、何度も書込める PROM

③キャッシュメモリ (cache memory)

・高速のプロセッサと低速の主記憶装置間の速度差を緩和する**緩衝記憶装置(buffer storage)**
・データの局所性*を利用し、アクセスする可能性のあるデータを選択的に読み込んで、アクセス効率を向上させる
*局所性…関連のある情報などが、一箇所にまとまる性質
・主記憶全域をランダムにアクセスするプログラムでは、効果は低い
・プロセッサに近いほうから、1 次キャッシュ(primary cache), 2 次キャッシュ(secondary cache)

1) 主記憶への書き込み
ⅰ. **ライトスルー方式** (write through)
・プロセッサがデータを書込むとき、キャッシュメモリと主記憶装置の両方に書込む
・書込み速度は高速化されない
・書き込みミスのリスクは低減

ⅱ. **ライトバック方式** (write back)
・プロセッサがデータを書込むときキャッシュメモリだけに書込。
・キャッシュメモリの内容が書き換えられる時に、更新された部分を主記憶装置に書込む
・書込み速度は高速化される
・書込みミスのリスクが増し、制御回路も複雑

2) アクセス速度の計算

・**ヒット率**…必要とするデータがキャッシュメモリに存在する確率
・**NFP**…(Not Found Probability)　〃　存在しない確率

ヒット率 ＋ NFP ＝ 1

キャッシュメモリを利用したデータアクセス時間 ＝ T_a

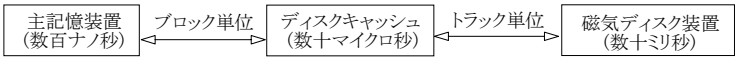
キャッシュメモリへのアクセス時間 ＝ T_c

主記憶装置へのアクセス時間 ＝ T_m

$T_a = T_c \times$ ヒット率 $+ T_m \times$ NFP

④**ディクスキャッシュ**（disk cache）

主記憶装置と磁気ディスク装置間の緩衝記憶装置



⑤**メモリインタリーブ**（memory interleave）

主記憶装置を複数の**バンク*(bank)**に分割し、並列的にアクセスすることで高速化を図る方式

*バンク …メモリ管理の単位となる、一定量のメモリの集合のこと

この方式はキャッシュメモリへのデータ転送や、パイプライン方式でのデータの先読みなどに効果的

○メモリリーク

動的に確保した主記憶装置が解放されず、主記憶中で利用できる領域が減少すること

○スラッシング

主記憶にないページが参照されるページフォルトが頻発し、ページの書き換えばかり時間がとられて処理が進まない状態

№6　補助記憶装置とその種類　　★★☆☆☆

①補助記憶装置の役割

1）記憶容量

OS の記憶管理などにより、不足する主記憶容量を補うように動作する

2）データの保存

作成したデータをファイルとして補助記憶装置の不揮発性媒体に書込んで保存する

3）データの配布

補助記憶装置は記憶媒体を切り離して持ち運べるものが多く、互換性もかなり保証されている

②補助記憶装置の分類

1）動作機構による分類

ⅰ. **直接アクセス (direct access)**

記憶内容の任意の場所を自由にアクセスできる

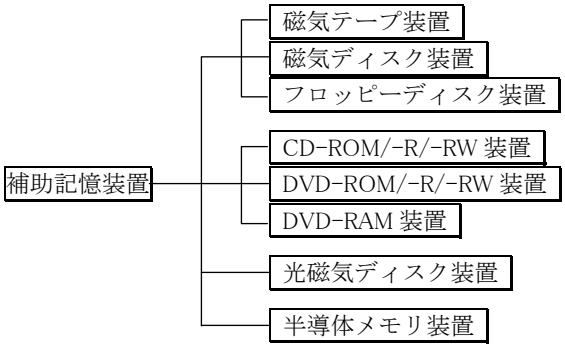
直接アクセス記憶装置(DASD：Direct Access Storage Device)

ⅱ. **順次アクセス (sequential access)**

先頭から順番にしかアクセスできない

DAT(Digital Audio Tape)を応用した **DDS(Digital Data Tape)**など

2）記憶媒体による分類



№7　磁気ディスク装置の構造と記憶装置　　★★★★★

①磁気ディスク装置の構造

1）**磁気ディスク装置 (magnetic disk drive)**

磁性体を塗布した円盤に、アクセスアーム(access arm)先端の磁気ヘッド(magnetic head)によりデータを記憶する

一般に**ハードディスク装置(HDD：Hard Disk Drive)**という

2) **トラック (track)**
・円盤上の記憶領域は、同心円状のトラックに分割され、データが保存される
・外周のトラック 0 から内側に順に番号がふられている
・トラック密度(tpi：Track Per Inch) …1inch あたりのトラック数

3) **シリンダ (cylinder)**
・磁気ディスクの同一半径のトラックを集めたもの
・同じシリンダのアクセスは、アクセスアームを移動する必要がない

4) **セクタ (sector)**
・トラックを放射状に等分した記憶単位
・セクタ容量は装置内で一定　　→ 中心部(高密度)，外周部(低密度)
※**ゾーンビット記憶方式 (zone bit recording)** …外側のゾーンほどセクタ数を多くする

②磁気ディスク装置の記憶容量
1) トラック容量が表示されている場合

記憶容量 = 1 トラックあたりの記憶容量
×1 シリンダあたりのトラック数×シリンダ数

例) シリンダ数(800)，1 シリンダあたりのトラック数(19)，1 トラックあたりの記憶容量(20,000byte)
20,000×19×800 = 304,000,000(byte) = 304M(byte)

2) セクタ容量が示されている場合

記憶容量 = 1 セクタあたりの記憶容量
×1 トラックあたりのセクタ数
×1 シリンダあたりのセクタ数×シリンダ数

例) シリンダ数(2,000)，1 シリンダあたりのトラック数(15)，1 トラックあたりのセクタ数(32)，1 セクタあたりの記憶容量(500)
500×32×15×2,000 = 480,000,000(byte) = 480M(byte)

③磁気ディスク装置の記憶方法
1) **バリアブル方式**
データの読み書きを**ブロック*(block)**単位で行う方式
※ブロック …複数の項目を 1 つにまとめたコードの集まり
↓
1 ブロックに含まれるレコード数 = **ブロック化因数**(blocking factor)

・IBG (Inter Block Gap)
ブロック間の無効領域
・IRB (Inter Record Gap)
ブロック化しない場合の
レコード間の無効領域

2) **セクタ方式**
データの読み書きをセクタ単位で行う方式
1 セクタに複数のデータは記憶できない

④保存できるレコード数の計算
バリアブル方式ではブロック化により記憶効率を上げる

1 シリンダあたりのトラック数	19 トラック
1 トラックのバイト数	20,000 バイト
IBG のバイト数	250 バイト
1 レコードのバイト数	600 バイト
レコード件数	15,000 件

◎ブロック化因数 = 6 でブロック化
・1 ブロックのバイト数
= 1 レコードのバイト数×ブロック化因数 + IBG のバイト数
= 600×6 + 250 = 3,850(byte)

・1 トラックに保存できるブロック数
= 1 トラックのバイト数÷1 ブロックのバイト数
= 20,000÷3,850 = 5.194… = 5 (block)

・1 トラックに保存できるレコード数
= 1 トラックに保存できるブロック数×ブロック化因数
= 5×6 = 30 (レコード)

・1 シリンダに保存できるレコード数
= 1 シリンダあたりのトラック数×1 トラックに保存できるレコード数
= 19×30 = 570(レコード)

・必要なシリンダ数
= レコード件数÷1 シリンダに保存できるレコード数
= 15,000÷570 = 26.31… = 27(シリンダ)

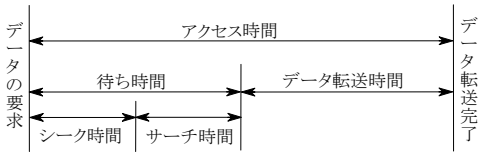
◎ブロック化なし
・1 ブロックのバイト数
= 600 + 250 = 850(byte)
・1 トラックに保存できるブロック数
= 20,000÷850 = 23.52… = 23(block)
・1 トラックに保存できるレコード数
= 23(レコード)
・1 シリンダに保存できるレコード数
= 19×23 = 437(レコード)
・必要なシリンダ数
= 15,000÷437 = 34.32… = 35(シリンダ)

№8 磁気ディスク装置のアクセス速度 ★★★★☆

①アクセス時間 (access time)

磁気ディスク装置にデータを記憶したり、記憶したデータを読み出したりする時間

アクセス時間 = シーク時間 + サーチ時間 + データ転送時間



1) **シーク時間 (seek time)**
磁気ヘッドを目的のトラック上まで移動させる時間。
位置決め時間とも言う。

2) **サーチ時間 (search time)**
・トラック上の特定のレコード位置が磁気ヘッドのところに回転してくるまでの時間
・磁気ヘッドとレコードの位置関係により、最小 0 回転，最大 1 回転
→ **平均回転待ち時間 = ディスク回転数(rpm)÷2**

3) **データ転送時間**
= 転送するバイト数÷データ転送速度

②アクセス時間の計算（例）

平均位置決め時間	20 ミリ秒
1 トラックあたりの記憶容量	20,000 バイト
データ容量	5,000 バイト
1 分間の回転数	3,000rpm

・1 回転時間
= 1 分÷1 分間の回転数 = 60,000 ミリ秒÷1 分間の回転数
= 60,000÷3,000 = 20(msec)

・平均回転待ち時間
= 1 回転時間÷2
= 20÷2 = 10 (msec)

・データ転送時間
= データ容量÷データ転送速度
= 5,000÷1,000 = 5(msec)

・アクセス時間
= 平均位置決め時間 + 平均回転待ち時間 + データ転送時間
= 20 + 10 + 5 = 35(msec)

③フラグメンテーション (fragmentation)

空き容量やファイルの連続性が断たれ、複数のセクタに散らばった状態。
ファイルの大きさは元のファイルと同じ。
↓
デフラグメンテーション (デフラグツール) で解消

№9 光ディスク装置と光磁気ディスク装置 ★★★★★

①CD-ROM 装置
1) CD-ROM の構造
・レーザ光線の反射でデータを読み取る
・反射層には平坦部分(**land**)とくぼみ(**pit**)があり、反射率の差でデータ表現
・セクタはらせん状で、中心部も外周部も同じ記憶密度

2) 読み出し方式

ⅰ. **CLV(Constant Linear Velocity)方式**

トラックに対するヘッドの相対速度(線速度)を一定にする方式

↓

ディスク回転数制御…中心部側(高速), 外周部側(低速)

ⅱ. **CAV(Constant Angular Velocity)方式**

ディスク回転数(角速度)を一定にする方式

↓

外周部ほどデータ転送速度は高速

3) 読み出し性能

ⅰ. シーク時間

ヘッド部分にレンズが必要で重くなり、らせん状トラック構造などから、磁気ディスク装置に比べて遅い

ⅱ. データ転送速度

標準で 150kbyte/sec (音楽 CD を標準として)

コンピュータ用では、標準の *x* 倍速という表現で高速化

4) **CD-R/-RW**

ⅰ. **CD-R**

・一度だけ書込むことができるが、消去することはできない

・記録層の有機色素をレーザ光線で分解して、レーザ光線の反射率を変化させることで、CD-ROM のピットに相当する状態をつくる

ⅱ. **CD-RW**

・書き換え可能な CD

・反射率が低いため、古い CD-ROM 装置では読み取れない

・記録層に相変化金属材料が使用され、レーザ光線の照射により結晶状態と非結晶状態に相変化させる

ⅲ. 書き込み方式

書込み方式		概　要
ディスクアットワンス (Disk At Once：DAO)		ディスク全体で一度だけ書き込み可能
マルチセッション (multi Session)	**セッションアットワンス (Session At Once：SAO)**	セッション単位で一度ずつ書き込み可能 空きがあればセッションの追記が可能
	トラックアットワンス (Track At Once：TAO)	セッションのデータ部が、トラック単位に分かれている
パケットライティング (Packet Writing)		リードイン，リードアウトを必要としないパケット単位で書込む

②DVD 装置

DVD … Digital Video Disk

→ **Digital Versatile Disk**

1) DVD の構造

・CD よりも波長の短いレーザ光線を使用

↓

・ピットを小型化し、高密度で記録

・両面を使用したり、記録層を 2 層にできる

2) DVD の種類

・DVD-ROM　…読み出し専用

・DVD-R　…書き込み可能

・DVD-RW　…書き換え可能

・DVD-RAM　…　〃

③光磁気ディスク

…**MO：Magneto Optical disk**

1) 特徴 → 書き込みと読み出しの方式が異なる

書み込み … レーザ光で加熱し、磁気で書き込み

読き込み … レーザ光の反射の偏光によって読み出し

2) 書き込み方式

・従来型

一度古いデータを消去してから新規にデータを書き込む

→ 読み出しの半分以下のデータ転送速度になる

・オーバライト型

現在の主流で、新規データを上書きして書き込む

№10　入力装置の種類と特徴　　★★☆☆☆

①キーボード (keyboard)

キーを押して文字コードや制御情報を入力する装置

②ポインティングデバイス (pointing device)

ディスプレイ画面上の位置情報を入力する装置

マウス (mouse)	底面のボールの回転で移動方向や移動量を指示し、ボタン操作で位置を入力する装置
トラックボール (track ball)	ボール型マウスをひっくり返した状態の装置
ジョイスティック (joy-stick)	操作棒を傾けて移動方向や移動量を指示する装置。ゲームに使われることが多い
タッチパネル (touch panel) **タッチスクリーン (touch screen)**	ディスプレイに直接触れて、位置を入力する装置 銀行の ATM などで広く普及
ディジタイザ(digitizer) **タブレット(tablet)**	平面パネル上でカーソルやペンを操作し、図面をなぞったり、画面のポインタを操作する装置

③ディジタイザ (digitizer)

平面パネル上図面をペンやカーソルでなぞることで座標を入力する装置

CAD(Computer Aided Design)システムなどに利用

※デジタイザ → 製図版のような大型専用機

タブレット → 卓上用の小型簡易版

④読取装置

1) **イメージスキャナ (image scanner)**

・紙面上の図形や写真などを小さな点の集合であるドットイメージに分解して、ドットごとの反射光の強さで読み取る

・読み取り精度 → **dpi (dots per inch)**

2) **OCR (Optical Character Reader)**

・紙などに書かれた文字を読み取り、文字コードとして入力する装置

・ソフトウェアによりパターンを解析して文字として認識

3) **OMR (Optical Mark Reader)**

・マークシートの黒く塗りつぶしたマークを読み取ることで、データを入力する装置

・読み取る情報はマークされた部分の位置情報

4) **バーコードリーダ (bar-code reader)**

・太さの異なる白線と黒線の集まりで情報をコード化したバーコードを読み取る装置

↓

POS システム※で利用され広く普及

※POS システム (Point Of Sales system)

…販売時点管理システム。商品を販売するごとに情報を記録し、在庫管理やマーケティングに活用できる

5) **磁気カード読取装置**

磁気カードに記録された情報を読み取る装置

磁気カード …紙やプラスチックのカードに磁気ストライプ層を施した媒体。ｷｬｯｼｭｶｰﾄﾞ，ﾃﾚﾎﾝｶｰﾄﾞなど

6) **IC カード**

・磁気カードのセキュリティの弱さを解決

・コストは高いが、記憶容量を増せ、情報処理機能を組み込むことができるので、高度なセキュリティ機能が可能

7) **磁気インク文字読取装置**

・MICR (Magnetic Ink Character Reader)

・磁気を帯びたインクで印刷された文字を磁気ヘッドで読み取る

⑤ディジタルカメラ (digital camera)

CCD という半導体素子で画像を読み取る装置

画像データの入力装置に位置づけ

※CCD (Charge Coupled Device：電化結合素子) … 光を電気信号に変換する素子

№11　出力装置の種類と特徴　　★★★★☆

①ディスプレイ装置 (display unit)

コンピュータ内のデータを画面に表示する装置

種　類	特　徴
CRT ディスプレイ (Cathode-Ray Tube display)	ブラウン管を使ったディスプレイ装置 大きく奥にせり出した筐体になる
フラットパネルディスプレイ (Flat Panel Display：FPD)	板状で省スペース筐体のディスプレイ装置
液晶ディスプレイ (Liquid Crystal Display：LCD)	液晶を利用したディスプレイ装置
プラズマディスプレイ (Plasma Display Panel：PDP)	放電による高圧ガスの発生を利用したディスプレイ装置
EL ディスプレイ (Electro Luminescence display)	電圧をかけると発光する物質を利用したディスプレイ装置

1) CRT ディスプレイ (Cathode-Ray Tube display)

・テレビと同じ構造のブラウン管(CRT：陰極線管)を使用

・電子銃から電子ビームを発射し、スクリーン内側の蛍光面に照射して発光させる。偏光ヨーク(電磁石の一種)でビームを曲げ、スクリーン全体を走査※する

※**走査 (scanning)** … 像を点の集合に分解し、電気信号に変え順番に送信する操作

走査方式	特　徴
インタレース (interlace)	1 回ずつ交互に奇数ラインと偶数ラインを走査する方法。動画向きでテレビに採用。
ノンインタレース (non-interlace)	毎回 1 本ずつ順番に走査する方法。 静止画や文字に向き、ｺﾝﾋﾟｭｰﾀﾃﾞｨｽﾌﾟﾚｲに採用。

2) 液晶ディスプレイ (Liquid Crystal Display：LCD)

・電圧をかけると分子が一方向に整列し、光の反射率や透過率が変化する液晶の性質を利用した表示装置

・画素制御方式

- **STN (Super Twisted Nematic)：単純マトリクス方式**
  縦横に配線した走査線と信号線の交点で画素を制御。
  視野角，表示速度，明瞭さに劣るが安価。
- **TFT (Thin Film Transistor)：アクティブマトリクス方式**
  画素ごとに薄膜トランジスタのスイッチング素子で制御
  STN に比べ、表示品質は優れるが高価

3) プラズマディスプレイ (Plasma Display Panel：PDP)

・2 枚のガラス基盤の間にヘリウムやネオンなどの高圧ガスを封入し、高電圧をかけて放電することで発光させる表示装置。

・コントラストが高く視野角も広くて、大画面化が容易だが、高電圧が必要。

4) EL ディスプレイ

・高電圧をかけると発光する物質を使った装置

・低電圧で高い輝度を得ることができる

無機 EL ディスプレイ
硫化亜鉛など無機物を使用
有機 EL ディスプレイ
ジアミン類など有機物を使用

5) VRAM の容量と表示能力

VRAM (Video RAM) …ディスプレイ装置に表示する内容を記憶する専用メモリ

VRAM の容量で解像度や色数が決まる

256 色	8 bit	1 byte
65,536 色	16 bit	2 byte
約 1,677 万色	24 bit	3 byte

例）解像度 800×600 ドットで 65,536 色の場合

$65{,}536 = 2^{16} = 2\text{byte}$

800×600×2 = 960,000 byte …必要な VRAM 容量

②プリンタ (printer)

1) プリンタの分類

印字方式	ｲﾝﾊﾟｸﾄ式	機械的に叩いて印字する衝撃式
	ﾉﾝｲﾝﾊﾟｸﾄ式	熱転写,ｲﾝｸｼﾞｪｯﾄ,レーザなどの非衝撃式
印字単位	シリアル	1 文字または 1 ドット単位で印字
	ライン	1 行単位で印字
	ページ	1 ページ単位で印字

印字方式	名称	印字単位	概　要
インパクト式	**ﾄﾞｯﾄｲﾝﾊﾟｸﾄﾌﾟﾘﾝﾀ**	リシアル	ワイヤー状のピンでインクリボンを叩いて転写
	ラインプリンタ	ライン	1行の桁数分の活字にインクリボンを叩いて転写
ノンインパクト式	**ｲﾝｸｼﾞｪｯﾄﾌﾟﾘﾝﾀ**	シリアル	ノズルからインクを吹き付けて印刷
	熱転写式ﾌﾟﾘﾝﾀ		熱でインクリボンのインクを溶かして転写
	感熱式プリンタ		熱で変色する専用の感熱紙を使用
	昇華型プリンタ	ページ	熱で気化させたインクを蒸着
	レーザプリンタ		電子写真式でトナーを転写して定着
プロッタ	**XY プロッタ**	シリアル	ペンまたは用紙を XY 方向に制御して描画

※感熱式や昇華型は熱転写式の一種である。 **サーマルプリンタ**とも言う
※レーザプリンタの印字原理　：　帯電→露光→現像→転写→定着→除電 の繰り返し

2) プリンタの性能
→ 解像度，色数，印字速度，用紙サイズ

ⅰ. **解像度 (dpi：dot per inch)**
1 インチ(2.54cm)あたりのドット数で表現

ⅱ. 印刷速度
・**cps (character per second)**　… 1 秒間に印字できる文字数
・**ppm (page per minute)**　… 1 分間に印刷できる枚数

№12　入出力インタフェース　★★★★★

①シリアルとパラレル

1) **シリアルインタフェース (serial interface)**
1 本の伝送路で 1 ビットずつ直列に転送する方式
ケーブルが細く、引き回しが容易

2) **パラレルインタフェース (parallel interface)**
8bit や 16bit といった処理単位の数だけの伝送路を持ち、並列に転送する方式
比較的短距離での接続に向く

3) 高速化
パラレル→伝送路が多いため高速化が容易に見えるが、同期を取る難しさや、隣合う線同士のノイズが影響しあう。
→ 高速化に限界 → シリアル化

②シリアルインタフェース（serial interface）

1) **RS-232C (Recommendation Standard-232C)**
・EIA(米国電子工業会)が規格化した、通信機器用インタフェース
・伝送速度は比較的低速

2) **USB (Universal Serial Bus)**
・パソコンと周辺機器を接続するインタフェース
・**ホットプラグ(hot plug)**対応 …電源を入れたまま接続可能
・USB ハブによりツリー状接続 …最大 6 階層 127 台

- USB1.1 ┬ 1.5Mbps (Low-speed mode)
  └ 12Mbps (Full-speed mode)
- USB2.0 ── 480Mbps (High-speed mode)

※ホットスワップ
…脱着可能な電気機器を、システムが通電された状態のまま取り外して、代替機器と交換･装着する操作

3) **IEEE1394**
・IEEE が制定した高速インタフェース
・Apple Computer 社の FireWire という規格を元に標準化
・ホストとなるパソコンを必要とせず、機器同士を直接接続可能
・デイジーチェーン接続やツリー状接続により最大 63 台接続可能
・ホットプラグ対応
・転送速度 … 100Mpbs, 200Mbps, 400Mbps

4) **IrDA (Infrared Data Association)**
・赤外線を使った無線通信の標準化を進める団体の名称でもあり、規格名でもある。

③パラレルインタフェース（parallel interface）

1) **セントロニクス**
プリンタ接続のデファクトスタンダード

2) **IEEE1284 (双方向パラレル)**
- IEEE が規格化した双方向インタフェース
- セントロニクス互換モード → プリンタ用として広く普及
- 伝送速度は最大 8MB/s （接続モードは 5 種類）
- モードにより 7 台のデイジーチェーン接続が可能

3) **SCSI (Small Computer System Interface)**
- ANSI がパソコン用に規格化し、
- USB の普及前に広く普及
- デイジーチェーン接続可能
  → 機器ごとに **SCSI ID** の設定
  チェーンの終端に**ターミネータ(terminator：終端抵抗)**
- 現在主流の SCSI-3 は最大 16 台まで
- 伝送速度は Ultra320 SCSI で 320MB/s
- シリアル SCSI も規格化

4) **GP-IB (General Purpose Interface Bus)**
- IEEE488 として規格化
- マイクロコンピュータと計測機器を接続
- 最大 32 台まで接続可能
- 伝送速度は数 MB/s

5) **IDE (Integrated Drive Electronics)**
- ハードディスク用インタフェース
- ANSI が **ATA**(AT Attachment)として規格化
- **ATAPI** (ATA Packet Interface) …ハードディスク以外の規格
- **シリアル ATA** が策定

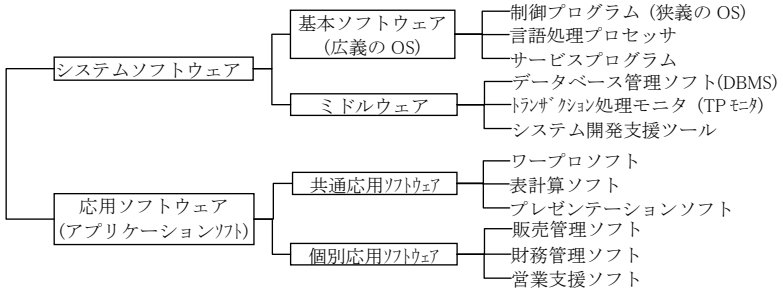
第 3 章　ソフトウェア

№1　ソフトウェアの分類　　★★☆☆☆

①ソフトウェアの分類

システムソフトウェア
　ハードウェアを稼動させるためのソフトウェア

応用ソフトウェア
　ユーザが利用するソフトウェア



②システムソフトウェア (system software)
　ハードウェアを制御し、効率的にコンピュータを活用するためのソフトウェア

1) **基本ソフトウェア (Operating System)**
- ハードウェアの基本的な制御を行い、ハードウェアの資源を有効活用するとともに、ユーザの使い勝手をよくするためのソフトウェア
- 広義：オペレーティングシステム（OS)
  狭義：制御プログラム

2) **ミドルウェア (middleware)**
- 基本ソフトウェアと応用ソフトウェアの中間に位置
- 多くのユーザが共通して使用するソフトウェア
- OS 上で動作し、アプリケーションに対して OS より高度で具体的な機能を提供

③応用ソフトウェア (application software)
　アプリケーションソフトとも呼ばれる。
　ユーザの利用目的に応じて作られたソフトウェア

1) **共通応用ソフトウェア**
- 多くの分野で共通して利用できる汎用的なソフトウェア
- 一般にソフトウェアパッケージにされ、市販されている

2) **個別応用ソフトウェア**
　ソフトウェアパッケージされた共通応用ソフトウェアでは十分な作業が行えない場合に、個別の業務ニーズに応じて開発したソフトウェア

№2　オペレーティングシステム　　★★★★☆

①　OS (Operating System)
　コンピュータシステムの中枢をなすもの

②OS の目的
　各機能を効率的に稼動させること。具体的には、

ⅰ．生産性を向上させること
ⅱ．ユーザインタフェース環境を向上させること

1) **生産性の向上**

ⅰ．**スループット(throughput)**の向上
　単位時間当たりの処理能力

ⅱ．**ターンアラウンドタイム(turn around time)**の短縮
　ジョブを出してから結果が出るまでの時間

ⅲ．**レスポンスタイム(response time)**の短縮
　データ入力終了から出力開始までの時間

オーバヘッドタイム
　OS の制御プログラムの実行など、仕事に直接関係ないが、その仕事を行うための間接的な時間

ⅳ. **RASIS** の向上
R：信頼性(reliability)　A：可用性(availability)
S：保守性(serviceability)　I：保全性(integrity)
S：安全性(security)
ⅴ. 応用ソフトウェアの負荷の軽減

2) **ユーザインタフェース環境の向上**
ⅰ. **GUI(Graphical User Interface)**による操作環境の提供
ⅱ. メモリや周辺機器の増設を容易に行える拡張性の提供
ⅲ. どのコンピュータでもデータやソフトが使える汎用性
ⅳ. **API** などによるソフトウェアの共通操作の提供

3) OS の**オーバヘッド**
OS などのシステムソフトウェアが走行することによってシステムにかかる時間的/空間的コスト。タスクスケジューラなど

③制御プログラム
応用ソフトウェアやミドルウェアが効率よく実行できる環境を作り出すためのソフトウェアの集まり
ジョブ管理 … ジョブの実行を監視・制御
タスク管理 … 処理装置の有効活用
データ管理 … ファイルの制御，データのアクセス制御
記憶管理 … 主記憶装置の有効活用
運用管理 … 運用管理の支援
障害管理 … 障害の制御
通信管理 … メッセージの送受信制御
入出力管理 … CPU と入出力装置とのデータの制御

1) **OS の三大管理**
ⅰ. **ジョブ管理**
通常、1 つのプログラムが 1 つの**ジョブ(job)**に対応
コンピュータ資源を割り付けながらジョブの実行を監視･制御

ⅱ. **タスク管理**
CPU の割り当て，割込み処理，マルチプログラミング処理など複数存在する**タスク(task)**※に対して、適切に CPU タイムを与える
※タスク … プロセッサ(CPU)の中での仕事の単位

ⅲ. **データ管理**
ファイルの制御，データのアクセス制御など記憶装置に格納されているデータを管理

④言語処理プログラム
プログラム言語で記述されたソースプログラムをコンパイル，インタプリタ，ジェネレータなどによって機械語に翻訳するソフトウェア

⑤サービスプログラム(ユーティリティソフト)
コンピュータの機能を補い、性能や操作性を向上させるソフトウェア

⑥代表的な OS
1) **Windows**
1986 年に米国 Microsoft 社が開発した MS-DOS の流れを汲むパソコン用の OS。MS-DOS のようなシングルタスクによるコマンドで命令を与える CUI 環境から、マルチタスクによるマウスやアイコン操作の GUI 環境になったことで操作性は向上。マルチメディア機能、ネットワーク機能を標準装備。

2) **Mac OS**
1984 年に Apple Computer 社のパソコン Macintosh 機に搭載された OS。GUI の環境をいち早く取り入れる。アーティストやデザイナーなどにユーザが多い。

3) **UNIX**
1970 年代に米国 AT&T ベル研究所が開発したマルチユーザ/マルチタスクの OS。UNIX を記述するためのシステム記述言語として C 言語が開発された。UNIX はソースコードが公開されているため、他のコンピュータへの移植や、改良が容易に行える特長を持つ。TCP/IP が標準装備されているので、インターネットのサーバとしても利用されている。ワークステーション標準 OS。

4) **Linux**
1991 年にフィンランドの学生 Linus Torvalds が中心となって開発したパソコン用 UNIX 互換 OS(PC-UNIX)。ソースコードが公開されている上、無償で配布されている。今後、Windows と対抗する OS として注目されている。

⑦ユーザインタフェース (User Interface)
ユーザとコンピュータが直接、対話のやり取りを行う接触領域
別名：**ヒューマンインタフェース(human interface)**

1) **CUI (Character User Interface)**
キーボードから文字を入力してコンピュータに指示を与える

2) **GUI (Graphical User Interface)**
視覚的に分かりやすい絵ボタン(アイコン)をクリックするなどしてコンピュータに指示を与える。

⑧**API (Application Program Interface)**
OS がアプリケーション開発に利用できるように、公開されているソフトウェア資源
(アプリケーションから、OS が用意する各種機能を利用するための仕組み)
↓
プログラム開発の負荷の軽減
インタフェースや操作方法の統一化

№3　ジョブ管理とタスク管理　　★★★★☆

①ジョブとタスクの関係

1) **ジョブ (job)**
ユーザがコンピュータに与える仕事の単位
JCL(Job Control Language：ジョブ制御言語)で指示を与える

2) **タスク (task)**
ハードウェア資源を使う実行の最小単位
(OS の管理下で実行される仕事の単位)

②**ジョブスケジューラ (job scheduler)**
・ジョブ管理の中枢を担う
・コンピュータに投入された複数のジョブは JCL で指定されたジョブの優先順位に従い、スケジュール管理し、ジョブを効率的に実行させる。

- **ジョブ管理プログラム**
  - **リーダ** …ジョブの読み込み
  - **ジョブスケジューラ** …ジョブの優先順位に従いｽｹｼﾞｭｰﾘﾝｸﾞ
    - **ジョブイニシエータ** …ジョブの開始
    - **ジョブターミネータ** …ジョブの終了
  - **マスタスケジューラ** …ユーザとｼﾞｮﾌﾞｽｹｼﾞｭｰﾗのやりとり
  - **ライタ** …ジョブの出力

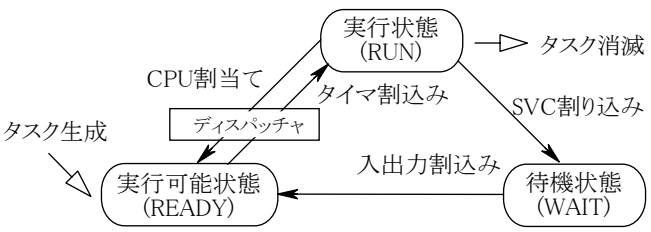
③**スプーリング (スプール処理：spooling)**
低速なプリンタの入出力処理から CPU を解放するための処理形式
コンピュータ本体と周辺装置のデータ転送速度の違いから起こる処理効率の低下を、補助記憶装置に保存することで、周辺装置の出力待ち時間を短縮すること
↓
スループットの向上
入出力装置は磁気ディスクなどの補助記憶装置にいったんデータを書き込んでおき、CPU を解放した後、CPU 処理と平行して補助記憶装置から入出力処理を適宜行う

④タスクの状態遷移
タスクの生成から消滅までの 3 つの状態遷移

1) 実行状態（**RUN**）
CPU の使用権が与えられ実行できる状態のタスク

2) 実行可能状態（**READY**）
いつでも実行できる状態にあり、CPU の空きを待っている状態のタスク

3) 待機状態（**WAIT**）
現在、入出力処理を行っており、その終了を待っている状態のタスク



※ディスパッチャ (dhispatcher)
…OS のディスパッチ機能を持つプログラム

ディスパッチ…実行準備が出来ているタスクやジョブに CPU の時間を割当てること

⑤プログラム構造

1) プログラム構造の分類

- **再使用不可能プログラム** …使用するたびに再ロードする
- **再使用可能プログラム** …再ロードしなくても使用できる
  - **逐次再使用可能プログラム** …複数のタスクを逐次使用
  - **再入可能プログラム** …同時に複数のタスクが使用可能
- **再帰的プログラム** …自分自身を呼び出す

2) **再使用不可能プログラム**

プログラム実行中に、プログラムの値(初期値など)が変更され、再度同じプログラムをそのまま実行することが出来ず、再度実行しなおす必要のあるプログラム

3) **再使用可能プログラム (reusable program：リユーザブル)**

実行中に変更した値などが初期状態に戻り、再ロードすることなく使用可能なプログラム

ⅰ. **逐次再使用可能プログラム (serial reusable program)**

1つのタスク終了後、同じプログラムを再ロードせずに次のタスクで繰り返し使用できるもの

ⅱ. **再入可能プログラム (reentrant program：リエントラント)**

タスクごとにデータ領域を持つことによって、実行中に自らの内容を変えず、複数のタスクを同時に使用できるプログラム。
(複数のタスクから平行して実行され、かつどのタスクにも正しい結果を返すために必要な性質)

4) **再帰的プログラム (recursive program：リカーシブ)**

プログラムの中から自分自身のルーチンを呼び出しても正しく実行できるプログラム

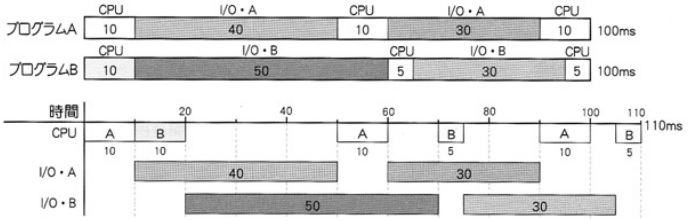
5) **再配置可能プログラム (リロケータブル)**

記憶装置上の配置位置に依存せずに実行が可能なプログラムの性質

№4　マルチプログラミングと割込み　　★★★★★

①マルチプログラミング (multiprogramming)

・複数のプログラムが見かけ上、同時に実行されているように見せる方式
・タスク管理プログラムのひとつ



②マルチプログラミングの実行

例) プログラム A,B それぞれの所要時間は 100ms
順次実行では、200ms かかる
マルチプログラミングでは、110ms となる

③セマフォ

複数のタスクやプログラムを並列処理する際に、
それらの実行をコントロールして混乱を防ぐ仕組み。
プロセスの排他制御に用いられる

④割込み (interrupt)

実行中のプログラムに何か別の要求が生じた場合、実行中のプログラムを一時中断し、要求に応じた別のプログラムを実行する。

1) **外部割込み**

プログラム以外で生じる割込み

ⅰ. **入出力割込み**
入出力処理の終了や、誤動作が生じたときの割込み

ⅱ. **マシンチェック割込み**
ハードウェアの異常、誤動作が生じたときの割込み

ⅲ. **コンソール割込み**
オペレータが介入するときの割込み

ⅳ. **タイマ割込み**
監視タイマなどで一定時間経過したときに生じる割込み

2) **内部割込み**
プログラム上から生じる割込み

ⅰ. **プログラム割込み**
プログラム実行中に異常が発生したときの割込み

ⅱ. **SVC(Super Visor Call)割込み**
プログラム上から入出力処理が必要になったときに生じる割込み
※Super Visor …メモリに常駐し、アプリケーションプログラム管理などを行う OS の中核プログラム。割込みの監視も行う。

№5　実記憶管理　　★★★☆☆

①主記憶装置の区画管理
主記憶装置をいくつかの区画(**partition：パーティション**)に分割し、その区画ごとに異なるプログラムをロードする

1) **単一区画方式**
・主記憶装置のユーザプログラムの区画は 1 つだけで、1 つのプログラムのみ実行する方法
・主記憶装置の利用効率はあまりよくない

2) **多重区画方式**
・主記憶装置を一定の大きさに分割し、それぞれのプログラムを割り当てる方式
・区画より大きいプログラムは実行できない
・区画より小さいプログラムは、空き容量が大きくなる
・主記憶装置の利用効率はあまりよくない

3) **可変区画方式**
主記憶装置をそれぞれのプログラムの大きさに応じた区画に割り当てる方式
↓
固定区画方式に比べ、はるかに利用効率は高い
↓
ただし、可変区画方式でも空いた区画に、より小さいプログラムのロードを繰り返していると、フラグメンテーション(fragmentation)が多数発生する。

②再配置
1) **コンパクション (compaction)**
断片化した主記憶装置上の空き容量(フラグメンテーション)をまとめて 1 つの連続した領域にすること。
ガーベージコレクションともいう。
プログラムの終了時に実行される。
↓
プログラムを主記憶装置内に移動
↓

2) **再配置 (relocation)**
プログラムで使用している各命令アドレス(相対アドレスやベースアドレス)を再配置する

ⅰ. **静的再配置 (static relocation)**
プログラムの実行前に再配置し、実行中は変更しないこと

ⅱ. **動的再配置 (dynamic relocation)**
プログラムの実行中に再配置を行うこと

③**スワッピング方式 (swapping)**
・主に優先順位が高いジョブが割込む場合に用いられる方式
・ジョブを割込ませるときに、主記憶装置上のプログラムやデータ領域を補助記憶装置に退避(スワップアウト，ロールアウト)し、ジョブの実行が終わると退避したプログラムを回復(スワップイン，ロールイン)する。

④**オーバレイ方式**
一つのプログラムを小さなセグメント(segment)単位に分割して、主記憶装置の容量以上の大きなプログラムを実行する方式

№6　仮想記憶システム　　★★★★☆

①**仮想記憶システム (virtual storage system)**
実際の主記憶装置の容量よりもはるかに大きな記憶空間を提供することを可能にしたシステム

実記憶 (real storage：実空間)
…実際の主記憶領域
実アドレス (real address：物理アドレス)
仮想記憶 (virtual storage：仮想空間)
…補助記憶装置を利用して作られた仮想記憶領域
仮想アドレス (virtual address)

仮想記憶に分割しておかれたプログラムは、そのプログラムが必要になった時点で実記憶に割り当てられる（動的再配置）

↓

動的アドレス変換 (dynamic address translation：DAT)
仮想記憶 OS で、仮想記憶装置上の論理アドレスを、実記憶装置上の物理アドレスに変換する、ハードウェア機構

仮想記憶システム ┬ ページング方式
└ セグメント方式

②ページング(paging)方式

・ページ (page)
プログラムを単純に 2～4KB の一定の大きさに分割したもの
ワーキングセット…実記憶上に設定されたページ数

・ページング (paging)
仮想記憶の構成単位であるページを主記憶から補助記憶に書き出したり、補助記憶から主記憶に読み込んだりする

プログラムは、ページ単位で分割され、実記憶にロードするため、まとまった空き領域がなくても、ページ単位の記憶領域があればロード可能
※**スロット** …仮想記憶方式において、ページアウトするための外部記憶装置のページ単位の領域
※**フレーム** …仮想記憶システムにおいて対応する実ページ（メモリ）のこと
↓
実記憶領域の利用率は高く、領域簡易が行い易い
アドレス変換機構によって管理

1)《ページング方式での処理》
1. プログラムは、ページの大きさに分割され、仮想記憶上に配置(静的再配置)される
2. 動的アドレス変換機構により、仮想アドレスを動的アドレスに変換(動的再配置)する。
3. 実記憶上に必要なページが存在しないというページフォルトが生じた場合、その必要なページを主記憶に転送(**ページイン**)する割込み処理を行う
4. 実記憶上で最も使用頻度の低かったページから順に仮想記憶に戻し、ページの入れ換え(**ページアウト**)をする

※**ページフォルト**の流れ
ページフォルト → 置き換え対象ページの決定 → ページアウト → ページイン

2) **ページリプレースメント (page replacement)**
不要なページと必要なページの出し入れ操作

ⅰ. **FIFO 方式 (First In First Out)**
時間的に最も古くページインしたページからページアウトし、一番新しくページインされたページが最後に取り出される方式

ⅱ. **LRU 方式 (Least Recently Used)**
最も長い時間参照されなかったﾍﾟｰｼﾞからページアウトする方式
その時点以降に参照される頻度が最も低いページがどれか推測する、ページ置き換えアルゴリズム

③セグメント(segment)方式

・セグメント (segment)
… プログラムを論理的なまとまりで分割したもの
処理形態はページング方式と同じ

ページング方式		セグメント方式
ページイン	→	ロールイン
ページアウト	→	ロールアウト

各セグメントの大きさが一定でないため、主記憶装置でフラグメンテーションが発生する

④仮想記憶システム利用における注意

・**スラッシング (thrashing)**

仮想記憶システムのうち、特に多重プログラム方式で頻繁にページの置き換えが起こること

↓

スラッシングを回避するには、

・実記憶を大きくする

・1 ページの大きさを小さくし、ワーキングセットを増やす

№7　プログラム言語　　★★★☆☆

①プログラム言語の変遷

機械語 (machine language)

コンピュータが理解できる唯一のハードウェア固有言語

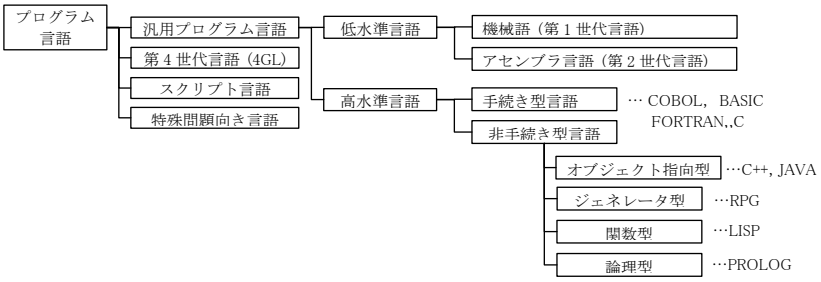
↓

アセンブラ言語 (assembler language)

機械語を人にわかりやすいように表意記号(**ニモックコード**)で表した言語

②プログラム言語の分類

プログラム言語



1) **汎用プログラム言語 (general purpose programming language)**

広範囲にわたる業務処理や問題解決するのに適した汎用性の高いプログラム

低水準言語	コンピュータが理解しやすい機械語またはより機械語に近い形式で記述
高水準言語	より人間の言語表現に近い形式で記述
手続き型言語	コンピュータで行う処理手順をプログラムの命令文で順次記述していく言語
非手続き型言語	パラメータ形式で一定の条件処理を入力する言語

2) **スクリプト言語 (script language)**

機械語への変換処理を省略し、テキスト形式で保存することによって簡易的に実行できるようにした比較的小規模なエンドユーザ向けプログラム言語

HTML，Perl，JavaScript，PHP など

3) **第 4 世代言語 (4GL：fourth generation language)**

処理内容をパラメータ入力による対話形式で開発する言語

プログラム開発が容易にできるエンドユーザ向けプログラム言語

4) **特殊問題向き言語**

ある特定の業務処理や問題解決をすることを目的に作られたプログラム言語

③代表的なプログラム言語

1) 高水準言語

ｉ．手続き型言語

C	UNIX のミニコン用の OS を記述する言語。現在では大型コンピュータからパソコンまで、さまざまな規模のコンピュータで使用されている。
BASIC	初心者向きの会話型言語として開発されたインタプリタ形式の言語。技術計算や事務計算さらにはゲームなどパソコンで広範囲に使用
COBOL	事務計算用言語。英語に近い文章表現なのでプログラマにとってなじみやすく、80 年代まで広く使用されていた
FORTRAN	科学技術計算用言語。複雑な数式をほぼそのままの形式でプログラミングできる。80 年代まで広く使用されていた。

ⅱ. 非手続き型言語

オブジェクト指向型	**C++**	C 言語をオブジェクト指向型に改良した言語
	Java	オブジェクト指向でインタプリタ形式の言語。インターネットや分散システム環境で利用される。OS やコンピュータの種類に依存することなくプログラムを実行できる。
	Ruby	オブジェクト指向のスクリプト言語。インタプリタ形式で強力なテキスト処理能力を持っている。UNIX,Windows,MacOS などの各プラットフォームも移植されている。
	Smalltalk	オブジェクト指向型のプログラム。マウスでコマンドを選択する方法やマルチウィンドウ，アイコンといった GUI が提供されている
ｼﾞｪﾈﾚｰﾀ型	**RPG**	事務用簡易言語。ファイル仕様，入出力仕様，計算仕様などを記入することによりプログラムを作成し、容易に報告書の作成ができる
関数型	**LISP**	リスト処理用言語。最近では、人工知能の研究に多用されている。
論理型	**PROLOG**	LISP と同様、人工知能の研究として使われている。

2) スクリプト言語

Perl	インタプリタ形式の言語。さまざまな OS で、Web ページのアクセスカウンタや CGI プログラムの作成に利用されている
PHP	HTML ファイルの中に組み込んで、動的な Web ページを作成する。XML のサポートや各種データベースとの連携が優れている

№8　言語処理プログラム　　★★★★☆

①言語処理プログラム

人間の言葉に近いプログラム言語によって作られた**ソース(原始)プログラム**を機械語に翻訳するプログラム
↓
翻訳されたプログラム … **オブジェクト(目的)プログラム**

②アセンブラ (assembler)

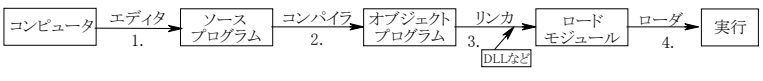
・アセンブラ言語で記述されたプログラムを翻訳する言語処理プログラム
・アセンブラ言語の命令語は、機械語と 1 対 1 に対応していて無駄のない効率の高いプログラムが作れる
・プログラム中でマクロ命令が使えるアセンブラ言語が多い
※マクロ命令 …定期的に使われる命令の 1 つの命令に定義したもの

③コンパイラ (compiler)

高水準言語で記述されたソースプログラムを一括してオブジェクトプログラムに翻訳する言語処理プログラム
C, FORTRAN, COBOL など
プログラム中で使用している外部関数を見つけ、未解決アドレスとして次のステップに渡す機能を持つ

《実行過程》

1. ソースプログラムの作成
   エディタを使ってソースプログラムを作成する
2. **コンパイル**する
   ソースプログラムを翻訳し、オブジェクトプログラムを作成
   翻訳作業は、**文字解析→構文解析→意味解析→最適化→コード作成**
3. **ロードモジュール(実行可能プログラム)**の作成
   関数やサブルーチンなど別々にコンパイルされたオブジェクトプログラムをリンカ(リンケージエディタまたは連係編集プログラム)によって連結し、ロードモジュールを作成する
   このとき、プログラムの中で共通に使われるデータやルーチンは、1 つの**モジュール**にまとめておき、プログラムの実行時に参照できるように **DLL(Dynamic Link Library)**を作っておく
4. 実行
   ローダを使ってロードモジュールを主記憶装置にロードし、プログラムを実行する



④インタプリタ (interpreter)

・ソースプログラムの命令を 1 つ 1 つ必要なところだけ翻訳しては実行する言語処理プログラム
・オブジェクトプログラムは作成されない
・実行速度は遅い
・BASIC, Ruby, Perl などの会話型言語

⑤ジェネレータ (generator：生成プログラム)

・プログラムの雛形が用意されてあり、利用目的に応じたデータやパラメータをキーボードなどから入力することで自動的にオブジェクトプログラムを作成してくれる言語処理プログラム
・PRG など

⑥プリコンパイラ (pre compiler)

・ソースプログラムをコンパイルする前に、マクロ機能やデータの変換,整列,条件合わせなどの事前処理を行わせる言語処理プログラム
・高水準言語で用いられる特定の文法や単語を、別の文法や単語または命令に置き換える動作を行うもの

⑦クロスコンパイラ (cross compiler)

・実際にプログラムを実行するコンピュータではなく、別のコンピュータでコンパイルし、オブジェクトプログラムを生成する言語処理プログラム

⑧プリティプリンタ

自動的に字下げなどを行い、ソースコードを見やすいように整形するツール

第 4 章　データ構造とアルゴリズム

№1　基本データ構造①ー配列構造　　★★★☆☆

①データ構造 (data structure)

プログラムがデータを記憶して操作するための表現方法

②配列 (array)

・同じ属性を持ったデータを集め、そのデータを添え字(**index**)によって識別する構造を持ったもの
・添え字の数によって、一次元配列，二次元配列，…
・添字の開始が 0 であるか 1 であるかに注意
・スタック構造，キュー構造，リスト構造がある

※リスト(list)

データ部とポインタ部からなるノードの連鎖によって一覧性のあるデータを実現する構造
リスト構造の最大の利点は、データの挿入/削除がポインタの付け替えのみができ、容易である

③スタック構造 (stuck structure：積み上げ方式)

・1 次元配列で格納されたデータにおいて、後から入れたものを先に出す構造
　＝ **LIFO (Last-In First-Out：後入れ先出し)方式**
・**push** …スタックにデータを格納すること
・**pop** …格納したデータを取り出すこと
・プログラムの実行中にサブルーチンを呼び出すときなどに使用

④キュー構造 (queue structure：待ち行列)

・1 次元配列で格納されたデータにおいて、一方の端からデータを入れ、他方の端からデータを取り出す構造
　＝ **FIFO (First-In First-Out：先入れ先出し)方式**
・並行して生起する複数の事象を、直列化することができる
・キューは資源の排他制御等にも使われる
・**enqueue** …キューにデータを格納すること
・**dequeue** …格納したデータを取り出すこと

⑤リスト構造 (list structure)

・順序付けされたデータの集合体
・各データは、データ部とポインタ部を対として持っている**セル**によって構成される
・リスト構造には**ポインタ**※の管理のしかたによって、単方向リスト，双方向リスト，循環リストがある
　※ポインタ … 他のデータへの参照を示す値で、そのデータのアドレスが入っている

1) **単方向リスト (one-way list)**

・ポインタには次のセルの場所が位置づけられていて、一方向だけにたどっていくリスト構造
・データの挿入，追加，削除が、ポインタを変更することで容易に行える
・最後のセルにはポインタはなく、データのみ存在

2) **双方向リスト (doubly-linked list)**

・前のセルへのポインタと次のセルへのポインタといった 2 つのポインタを持たせ、前後に自由にリストをたどることが出来るようにしたリスト構造
・最後のセルには前ポインタのみ存在

3) **循環リスト (circular list)**

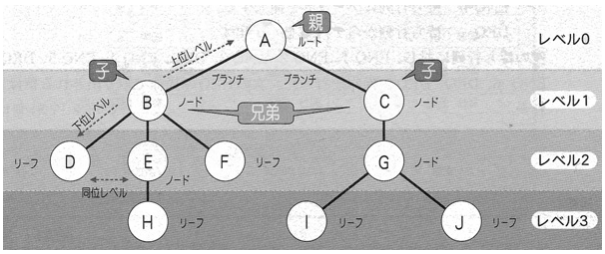
・双方向リストと同様に、前のセルへのポインタと次のセルへのポインタの 2 つのポインタを持つリスト構造
・最後のセルの次ポインタに先頭データがあるセルのアドレスが示されており、各データが環状に連結する構造

№2　基本データ構造②ーツリー構造　　★★★☆☆

①ツリー構造 (tree structure)

・データ間の関係を階層的に表現したデータ構造
・OS における階層ディレクトリで使用

・ノード (node：節)
　個々のデータを示す○の部分
・ルート (root：根)
　最も上位のレベルにある唯一のノード



↓
ノードとそのすぐ下のノードは親子関係で表現
↓

・リーフ (leaf：葉)
子のないノード

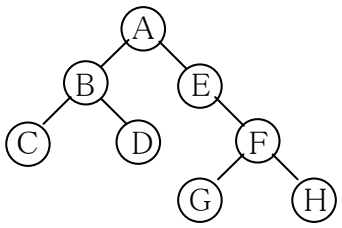
・ブランチ (branch：枝)
データ間を結ぶ

②二分木 (binary tree)
ノードからブランチの数が 2 本以下、つまり最大でも 2 つの子しか持たないツリー構造

○完全二分木
ルートから全てのリーフまでブランチの数が全て 2 の二分木

○二分木のデータ構造
左のポインタ部，データ部，右のデータ部　で構成
左のポインタ部 → 左の子のデータ位置
右のポインタ部 → 右の子のデータ位置

◎木の走査



ⅰ．**前順・先行順**
走査順は①親節点、②左節点、③右節点
A→B→C→D→E→F→G→H

ⅱ．**間順・中間順**
走査順①左節点、②親節点、③右節点
C→B→D→A→E→G→F→H

ⅲ．**後順・後行順**
走査順①左節点、②右節点、③親節点
C→D→B→G→H→F→E→A

ⅳ．**レベル順・幅優先順**
根から走査を開始し、走査順は①現レベル内全節点を左から走査、②次レベルの走査
A→B→E→C→D→F→G→H

③二分探索木 (binary search tree)
二分木の各ノードに要素(データ)を持たせたもの
・親のデータより小さいデータは左の子　　→ ページ検索を容易に行える
・親のデータより大きいデータは右の子

④AVL 木 (AVL tree)
二分探索木に全てのノードにおいて左右の部分木の高さが 1 以内である制約を加えたもの
↓
探索効率は悪くはないが、B 木の方が実用的

⑤B 木 (B tree)
m 分木をベースにしたもの
→ノードが最大 *m* 個(*m*≧2)の子を持つことができるツリー構造 （**多分木(multi-way tree)**）

◎多分探索木 (multi-way search tree)
多分木の探索木
ⅰ．ルートはリーフであるか、2～*m* 個の子を持つ
ⅱ．ルート,リーフ以外のノードは、$\frac{m}{2}$ ～*m* 個(小数点切上げ)を持つ
ⅲ．ルートから全てのリーフまで深さが全て等しい
↓
上の 3 つの条件を満たす *m* 分探索木を ***m* 階の B 木**という

二分探索木 → 全てのノードがデータを持つ
B木 → データを持つのはリーフのみ。
リーフ以外のノードはキーだけを持つ

№3　アルゴリズムの基礎　　★★★☆☆

①アルゴリズム (算法：algorithm)
問題を解くための手順のこと
問題を解くためのものであって、明確に定義され、順序付けられた有限個の規則からなる集合

②プログラム流れ図

流れ図 (flow chart) …プログラムのアルゴリズムを設計するための図式

1) **端子記号** ⬭

流れ図の開始を終了を示す
流れ図の最初と最後に必ずつける

2) **処理記号** ▭

あらゆる種類の処理を示す
記号中に処理内容を書く

3) **判断記号** ◇

条件を判断し、その結果で異なった処理を実行

ⅰ．条件による双岐分岐
ⅱ．比較による双岐分岐
ⅲ．多岐分岐

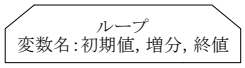
4) **データ記号** ▱

データの入力と出力を示す

5) **ループ記号** ⬠ ⬡

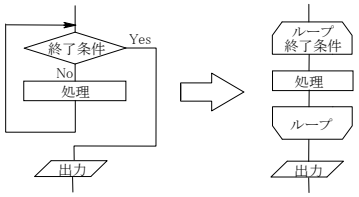
ループ(繰り返し)の始端と終端を示す

ループの繰り返し指定法



ⅰ．**前判定型**

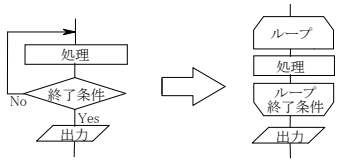
ループ始端に終了条件を書く



前判定型では、処理前に終了条件を判定するので本体を一度も処理しないことがある

ⅱ．**後判定型**

ループ終端に終了条件を書く



後判定型では、処理後に終了条件を判定するので、少なくとも一度は本体を処理する

③**トレース (trace)**

流れ図に沿って変数の値を追跡すること
(プログラムの実行順序，実行結果などの履歴情報を出力するデバッグツール)
プログラムを**デバッグ**※する上で重要

※デバッグ（debug）…誤りを探し出して修正すること

№4　探索　　★★★☆☆

①**探索 (search)**

複数のデータの中から目的のデータを探し出すこと

②**線形探索法 (linear search)**

・別名：**順次探索(sequential search)**
・配列の最初のデータから一つずつ順番に比較して探索する方法
・データが整列されている必要はないが、時間がかかる
・最後のデータの次に探索データと同じデータを番兵(sentinel)として置く
　↓
データ N 個の中に探索データがなくても、N+1 個目で必ず見つかるため、データの終わりに達したか毎回チェックする必要がない

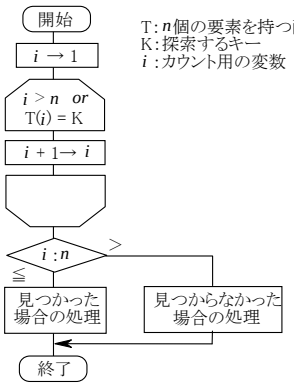
データ件数が N 件のとき
　平均探索回数 ＝ (N+1)÷2 回
　最大探索回数 ＝ N 回

③二分探索法 (binary search)

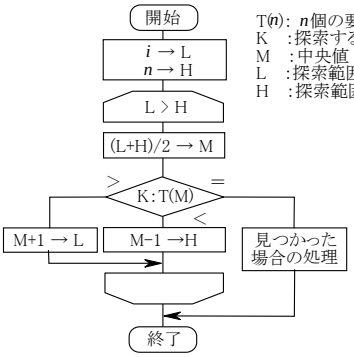
・配列のデータを次々に、二等分して探索範囲を狭めていく方法
・配列データは降順もしくは昇順で整列されていなければならない

> データ件数が N 件のとき
> 平均探索回数 X ＝ $\log_2 N$ 回
> 最大探索回数 Y ＝ $\log_2 N+1$ 回
> $2^X \leqq N < 2^Y$

▼線形探索の流れ図



▼2分探索の流れ図



④ハッシュ法 (hash method)

・高速探索法の一つで、はじめから探索しやすいように、データ格納時に格納位置を**ハッシュ関数**という計算式を使って決める
・文字列を 1 字ずつ比較して探索すると処理に時間がかかるため、文字列をハッシュに変換し、その値を検索する
・除算法，乗算法，基数変換法など

例) 7361 という 4 桁の数値を配列に格納したい場合、
各桁の値を加えて 7 で割った余りをハッシュ値とし、格納する位置にすると、
(7+3+6+1) mod 7 = 格納位置 3
2483 の場合
(2+4+8+1) mod 7 = 格納位置 3
↓
衝突(コリジョン：collision) …異なるデータが同じ格納位置になる

コリジョンが頻繁に発生する場合、探索効率は低下する

№5 基本整列法 ★★★★☆

①整列 (sort)

…同一属性を持った複数のデータをある特定の順番に並び替えること
- **昇順整列** … 小さい順に並べる
- **降順整列** … 大きい順に並べる

②オーダ記法

アルゴリズムの評価手法の一つで、処理時間のおおよその計算量を表す
注目する変量(データ数等)の変化に対する計算量の変化傾向を示す指標

データ件数が 2 倍，3 倍となると、計算量も比例して、2 倍，3 倍になるとき、O(n)と表す
計算量が 4 倍，9 倍になるとき、O(n^2)と表す

	整列法	オーダ
基本整列法	バブルソート	O(n^2)
	選択ソート	O(n^2)
	挿入ソート	O(n^2)
高速整列法	クイックソート	O($n\log_2 n$) 最大 O(n^2)
	ヒープソート	O($n\log_2 n$)
	シェルソート	O($n^{1.5}$)

例) データ件数が 1000 の場合、バブルソートとクイックソートの比較回数の差は、
$O(n^2) \div O(n\log_2 n) = 1000^2 \div 1000\ \log_2 1000$
$= 10^6 \div 10^3 \log_2 2^{10}$ $(1000 \fallingdotseq 2^{10})$
$= 10^6 \div 10^4 = 100$(倍)

③**バブルソート**
(bubble sort：基本交換法)

隣り合うデータ同士を比較し、大小関係が逆の場合にそれらのデータを並べ替えるという作業を繰り返し行う方法

整列時の比較回数 $=(n-1)+(n-2)+\cdots+2+1=n(n-1)\div 2$

④**選択ソート (selection sort：基本選択法)**

データの中で最小値(または最大値)を選び出し、最後のデータと入れ換える作業を繰り返し行う方法

整列時の比較回数 $=n(n-1)\div 2$

⑤**挿入ソート (insert sort：基本挿入法)**

・配列データにおいて、挿入するデータより手前までのデータが整列済みであるという前提で行う方法
・挿入データの手前のデータから順次データを比較し、適当な場所にそのデータを挿入する作業を繰り返し行う。

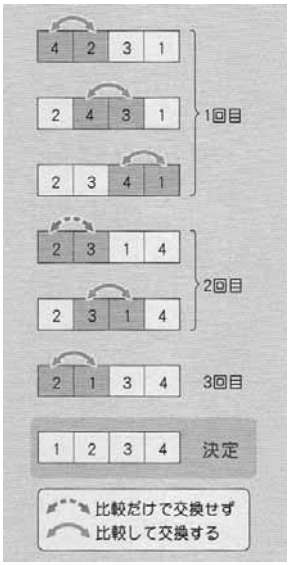
整列時の比較回数 $=n(n-1)\div 2$　　最小 n 回

⑥**マージソート**

複数の整列済みデータ列を１つのデータ列にすることを併合(マージ(marge))という。
併合を繰り返して整列すること

⑦**基数ソート**

データをバケットに振り分けて取り出す。これを末尾桁の数から順に先頭桁の数まで行う整列法



バブルソート

基本選択法

基本挿入法

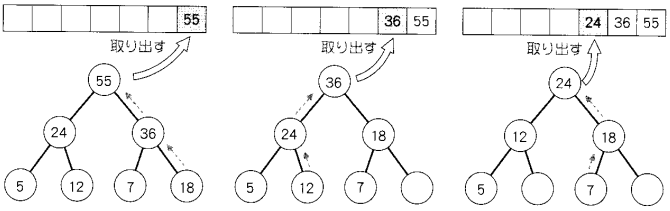
№6　高速整列法　　★★★☆☆

①**ヒープソート (heap sort)**

・基本選択法を応用した方法
・配列をヒープ※に変換してルートの値と各ノードの値を比較して整列を　行う。

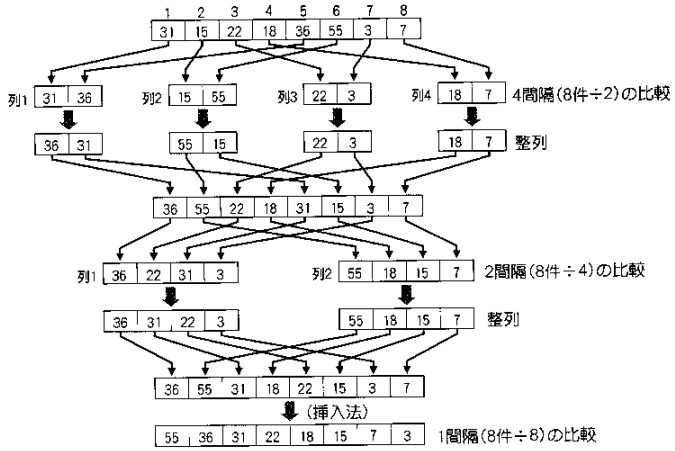
※**ヒープ(heap)**…完全二分木を配列で実現するデータ構造。ルートには最大値が入る、親ノード≧子ノード。またはルートには最小値で入る、親ノード≦子ノード。

左右の子に大小関係はない
ルートから最大値または最小値を取り出して配列の末尾に配置し、新たなヒープを作る作業を繰り返す。



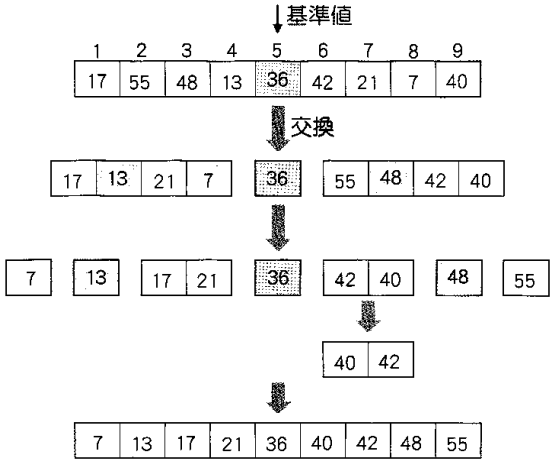
②**シェルソート (shell sort)**

・基本挿入法を応用した方法
・ある一定の間隔ごとのデータ列に分け、挿入法で整列させ、その間隔を順次狭くしていく方法
・間隔は通常、データ件数の 1/2, 1/4, 1/8,…とする　(間隔が１になった場合、基本挿入法)



③クイックソート (quick sort)

・全データから基準値を選び出し、その値より大きいか小さいかで振り分ける操作を繰り返す方法
・基準値は通常、中央のデータを使うことが多い



第 5 章　ファイルとデータベース

№1　ファイル構成　　★★☆☆☆

①ファイルの構成要素

1) **フィールド (field：データ項目)**
- ファイルを構成する要素の最小単位
- プログラムはそれぞれのフィールドに対して属性を定義する

2) **レコード (record)**
- 関連するフィールドを 1 つにまとめたもの

ⅰ. **論理レコード (logical record)**
プログラムの内部で処理する論理的な処理の単位となるレコード

ⅱ. **物理レコード (physical record)(ブロック:block)**
ディスク装置への格納効率，物理的なデータの入出力の効率性を図るために通常は複数件のレコードを 1 つのブロックとしてまとめたレコード
ブロックに含まれるレコード数 → **ブロック化因数(blocking factor)**

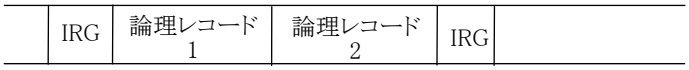
3) **ファイル (file)**
- 物理レコードを 1 つにまとめたもの
- OS の**データセット(data set)**によって管理される

②**レコード形式**

1) **固定長レコード (fixed length record)**
ファイルを構成する論理レコードの長さが全て同じ形式
非ブロック化レコードとブロック化レコードがある

ⅰ. **非ブロック化レコード**

ⅱ. **ブロック化レコード**

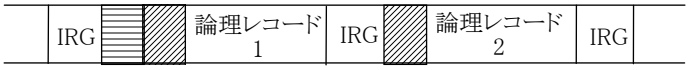


- IBG (Inter Block Gap)
  ブロック間の無効領域
- IRB (Inter Record Gap)
  ブロック化しない場合のレコード間の無効領域

2) **可変長レコード (variable length record)**
論理レコードの長さがレコードによって異なる形式
非ブロック化レコードとブロック化レコードがある
制御フィールド (control field) …各レコードやブロックの長さを記録している

ⅰ. **非ブロック化レコード**

ⅱ. **ブロック化レコード**



…ブロック制御フィールド
…レコード制御フィールド

3) **不定長レコード (undefined length record)**
- 論理レコード長が不定で、制御フィールドを持たず、各レコードの長さを記録しない形式
- 非ブロック化レコードしか扱えない
- 一般にロードモジュールなどを格納する場合に利用

※スパンレコード
- 複数の物理レコードを 1 つの論理レコードの中に格納した形式のレコード形式
- 論理レコードが非常に大きい場合などに論理レコードを複数の物理レコードに分割して記録する

③ファイルの分類

1) 用途による分類

マスタファイル (master file)	商品ファイルや人事ファイルなどデータ処理の基本となるファイル
トランザクションファイル (transaction file)	伝票ファイルなど逐次発生する変動データファイル
ワークファイル (work file)	整列処理やデータ処理などの作業領域として使用するファイル
バックアップファイル (backup file)	障害に備え、マスタファイルの内容をコピーしたファイル
ログファイル (log file)	システムの稼動状況や更新や通信などの履歴を記録したファイル
チェックポイントファイル (check point file)	障害回復に備え、ある一定のタイミングで記憶装置やレジスタの内容を記録したファイル

2) 利用者による分類

システムファイル (system file)	OS を動作させるのに必要なファイル
ユーザファイル (user file)	ユーザがﾌﾟﾛｸﾞﾗﾑやｱﾌﾟﾘｹｰｼｮﾝｿﾌﾄなどを実行するときに使ったﾃﾞｰﾀﾌｧｲﾙ

3) 使用時間による分類

パーマネントファイル (permanent file)	マスタファイルのように長時間保存されるファイル
テンポラリファイル (temporary file)	トランザクションファイルやワークファイルのように一時的に保存しておくファイル

④ファイルのアクセス方法

1) **順次アクセス (sequential access：シーケンシャルアクセス)**
- ファイル中のレコードに物理的に並んでいる順にアクセスしていく方法
- 磁気テープは順次アクセス

2) **直接アクセス (random access：ランダムアクセス)**
- ファイル中の任意のレコードに直接アクセスする方法
- 磁気ディスクや光ディスクなどが直接アクセス方式
- 直接アクセスではアクセスするレコードのアドレスかそのアドレスを算出できる情報をファイル中に持っている

3) **動的アクセス (dynamic access：ダイナミックアクセス)**
- 順次アクセスと直接アクセスの両機能を備えた方法
- 特定のレコードが直接アクセス可能で、それ以外は順次アクセスする

⑤その他

OLTP (オンライントランザクションプロセッシング)
… データベースアクセスの性能

№2　ファイル編成　　★★★★☆

①ファイル編成

磁気ディスクファイルなどの記憶媒体上でレコードの記憶手順や構成を示したもの

順編成法	順番に頭から記憶する
直接編成法	レコードのキー項目の値から記録する位置を算出
索引編成法	順次アクセスと直接アクセスの両方の機能を備える
区分編成法	順、索引編成法の特徴を併せ持つ
仮想記憶編成法	順、直接、索引、区分編成法の考えを 1 つにまとめる

②**順編成法 (SAM：Sequential Access Method)**
- 記録媒体に順番に頭から記録する最も単純な編成方法
- 論理的にはファイルの大きさに制約はない
- 記憶領域に無駄がなく(格納効率が高く)、大量データを効率よく処理できる
- 先頭から順次アクセスするため、目的のレコードに到達するまでレコードを順次読まなければならず、レコードのランダム処理には不適
- バッチ処理のマスタファイルやトランザクションファイルに利用
- レコードを変更する場合、一度新しいファイルに書きながら、関連するレコードを更新,挿入,削除しなければならない

③**索引編成法 (ISAM：Index Sequential Access Method)**
- **索引域**（索引を格納する領域）
- **基本域**（レコードを記録する領域）
- **あふれ域**（追加により基本域からあふれたデータを記録する領域）

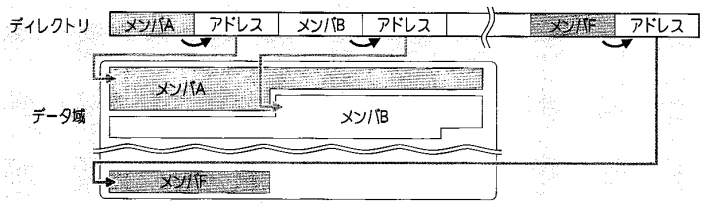
・順次アクセスと索引キーによる直接アクセスの両方の機能を備える
・記憶媒体は直接アクセス記憶装置(DASD)に限定
・最も基本となるのは、レコードのキー項目順に格納されている基本域
・3 階層の索引を順次使用してアクセスするため、直接編成(DAM)ファイルより検索に時間を要する
・レコードのあふれが頻発する場合、基本データ領域とあふれ領域の双方にレコードが混在することとなり、検索などが非効率になる
→ **再編成処理**(レコードの並び順の整理)が必要

④直接編成法 (DAM：Direct Access Method)
・各レコードにキー項目を持ち、この項目の値からレコードアドレスを算出し、その位置にレコードを記録する編成法
・オンライン処理に適し、順次アクセスも可能
・記憶媒体は磁気ディスクなどの直接アクセス記憶装置(DASD)

1) **直接アドレス方式**
・レコードのキー項目の値をそのままアドレスにする方式
・レコードの更新，挿入，削除は、他のファイルと比べて最も簡単で、高速アクセスが可能
・コード番号がとびとびの場合、使用されない記憶領域がかなりあり、無駄な領域となる

2) **間接アドレス方式**
・キー項目の値に対して一定の計算を行い、その値から相対レコードアドレスに変換する(**ハッシング(hashing)**)
→ 除算法，基数変換法，重合せ法など
・**シノニム (synonym)**
…アドレス変換において異なるキーが同じアドレスに変換されること

⑤区分編成法 (PAM：Partitioned Access Method)
・順編成法と索引編成法の両方の特徴を持つ
・順編成法でできている**メンバ**とメンバがどこに位置するかを示す**ディレクトリ**から構成される
・通常、プログラムライブラリとして使用される
・記憶媒体は直接アクセス記憶装置(DASD)に限定
・メンバの削除や追加で無効領域が発生するため、定期的に**再編成処理**が必要
・ディレクトリは各メンバの名称や先頭アドレスを持っていて、各メンバの取り出しはランダムに行える
・メンバの内容を扱うときは順次アクセス
・メンバ全体の追加/挿入/削除は比較的簡単にできるが、メンバごとのレコード単位では順編成法と同様時間がかかる



⑥仮想記憶編成法 (VSAM：Virtual Storage Access Method)
・仮想記憶システムで利用される順，直接，索引，区分編成法の考え方を 1 つにまとめた編成法
・論理レコードだけ意識すれば、後はシステムに任せ、効率のよい一元的なアクセスが可能
【VSAM ファイルの利点】
・処理に応じたファイルの編成方法の選択が可能
・データセット管理機能が提供される
・アクセス効率が高い
・相対バイトアドレス(RBA)による記憶装置からの独立

○データスペース（d**ata space)**
磁気ディスク装置などの補助記憶装置上に仮想記憶編成ファイルを作成する領域
クラスタ (cluster)
データスペース中に確保された領域
コンポーネント (component)
クラスタ中にあるレコードを記憶する領域
制御域 (CA：Control Area)
コンポーネント中にあり、ファイルを構成する
制御インタバル (CI：Control interval)
CA は複数の制御インタバルで構成される

○**データセット (data set)**
仮想記憶編成法によるファイル

ⅰ. **入力順データセット (ESDS：Entry Sequenced Data Set)**
・ファイル内のレコードが入力順に並んでいる
・順編成ファイルに相当
・レコードの最後に追加・更新することはできるが、挿入・削除には適さない

ⅱ. **キー順データセット (KSDS：Key Sequenced Data Set)**
・ファイル内のレコードがキー順に並んでいる
・索引編成ファイルに相当
・レコードの挿入・追加・削除が自由にできる

ⅲ. **相対レコードデータセット (RRDS：Relative Record Data Set)**
・相対レコード番号が自動的に付けられ、その番号をもとにデータに直接アクセスする
・直接編成ファイルに相当
・レコードの挿入・削除や追加・更新も可能

№3　ディレクトリ構成　　★★★☆☆

①**ディレクトリ (directory)**
・パソコンやワークステーションで使われている MS-DOS や UNIX といった OS におけるファイル管理の単位
・階層構造

1) **ルートディレクトリ (root directory)**
・最上位のディレクトリ
・ルートディレクトリは常に１つ

2) **サブディレクトリ (sub directory)**
・ルートディレクトリの下層にある複数のディレクトリ
・ユーザが自由に作ることができ、サブディレクトリの中にはファイルやディレクトリをさらに作ることができる

3) **カレントディレクトリ (current directory)**
・ユーザが現在の作業を行っているディレクトリ
・ユーザの作業状況によって位置は変化する

②**パス (path)**
ファイルやディレクトリまでの経路

- **絶対パス** … ルートディレクトリから指定
- **相対パス** … カレントディレクトリから指定

③**ワイルドカード (wild card)**
MS-DOS や UNIX などで、ファイルを指定するときに用いられる機能
“？” 任意の１文字
“＊” 任意の文字列

№4　データベース　　★★★★☆

①データベースの基本概念
個別に作成していたファイルを整理し、一元化することでデータの共有化とプログラムからの独立性を図り、データの質と安全性を向上させる

②**データモデル (data model)**
データ構造やコンピュータ操作をどのようにするかを決めたもの

1) **階層モデル (hierarchical data model)**
・別名：**ツリー構造**
・１つ上位レベルのレコードに対して、下位レベルのレコードは複数対応
・下位レベルに対しては、上位レコードは１つしか対応しない （＝親は常に一つ）

2) **ネットワークデータモデル (network data model)**
・ネットワーク構造によってデータ構造を表現したもの
・階層型モデルを改造した方法　→ １つのレコードに対して上位レコードも下位レコードも複数存在する

3) **関係モデル (relational data model)**
・2 次元の表構造によってデータ構造を表現
・別名：リレーショナルデータモデル

・**関係(relation)** … 属性と組で構成された表
　属性(attribute) … 表の列
　組 (tuple) … 表の行

・**主キー (primary key)**
　…1 つの表の中で各組全て、その値が異なるもの
　→値は重複せず、空ではないという実体の完全性を持つ

・**外部キー (foreign key)**
　…主キーが他の表の属性として使われるときの属性のこと

ⅰ. **参照の完全性**
「主キーと外部キーは表 A に主キー、表 B に外部キーがある場合、B の属性欄のデータは A の属性欄の値として必ず存在しなければならない」

ⅱ. **リレーショナルデータベース（RDB：Relational DataBase）**

ⅲ. 関係演算

・**射影 (projection)**	：	表の中から特定の属性(列)だけを抽出する
・**選択 (selection)**	：	特定の組(行)だけを抽出する
・**結合 (join)**	：	共通する属性をもとに複数の表を 1 つにまとめる

ⅳ. **ビュー表**
関係データベースの表に関係演算などを施してできた仮想表

№5　データの正規化　　★★★★☆

①**正規化 (normalization)**
関係データベースにおいて、表から集合や配列などを取り除き、データの整合性を保てるような表にすること

1) **第一正規化**
非正規形のデータ項目群に繰り返し部分がある場合、別のレコードとして分離させる
導出項目(演算などによる算出部など)がある場合は項目を削除する

2) **第二正規化**
第一正規化した表から、各組すべてで異なっている項目をキー項目として設定する
部分関数従属する項目群ごとに表を分離する
※部分関数従属 …キー項目に付随(関数従属)している項目群のこと

3) **第三正規化**
キー項目以外に関数従属している項目群があれば分離する

№6　DBMS　　★★★★★

①**データベース管理システム (DBMS：DataBase Management System)**
さまざまな情報をデータベースに蓄積し、ユーザの要求に応じて、情報を検索し応答する機能を持つソフトウェアシステム
【主な機能】
・定義機能　　・操作機能　　・制御機能

データベース管理者 (DBA：DataBase Administrator)
外部スキーマ以外のデータベースの設計，編成，保守，運用監視，障害回復などのデータベースに関する全体的な管理を行う人

②データベースの保護機能
1) **機密保護 (security)**
・**ユーザ ID(user IDentifier)**や**パスワード(password)**を許可されたユーザに与えることによってユーザの識別を行う
・**ログ(log)**を採取し、利用状況を確認したりデータを暗号化する

2) **排他制御 (exclusive access control)**
・複数のユーザが同時に同じデータにアクセスすることを制御すること

・参照するだけは問題ないが、ほぼ同時に更新した場合、二重更新され、片方の更新が無視される場合がある
↓
あるユーザが**トランザクション処理**※をしている間､他のトランザクションで同一のデータベースに更新をかけられないようにロックする
※トランザクション(transaction) …データベースの更新やロールバック処理の単位

3) 共有ロックと占有ロック

ⅰ. **共有ロック**
データベースを参照するときにかけるロック
他から参照できるが更新はできない

ⅱ. **占有ロック**
データベースを更新するときにかけるロック
他から参照も更新もできない
占有ロックをかけると他からロックは一切かけられない

4) **デッドロック (dead rock)**
・2 つ以上のトランザクション処理が、それぞれロックのかかったデータにアクセスしようとしたとき、アンロックができない待ち状態が無限に続くこと
・デッドロックが生じた場合、一方のトランザクション処理を強制キャンセルする必要がある

5) **2 相コミット**
・分散データベースシステムでデータの整合性を確保するために、一連の更新処理を行う際、複数のデータベース間で同期を取り、更新する制御方式

6) **ストアドプロシージャ (stored procedure)**
・データベースサーバを利用したクライアントサーバで、定期的に使う SQL 文などデータベース処理を行うプログラムを一つにまとめ、データベースサーバに保存しておくこと
・処理時間の軽減とネットワークのトラフィック削除が可能になる

③障害復旧 (recovery)
・データベースの障害対策
- **バックアップファイル**
  … 定期的にコピーをとる
- **ログファイル (ジャーナルファイル)**
  … バックアップの更新情報の履歴を記録する

1) **ロールバック (roll back)**
論理的障害が生じたとき、ジャーナルファイルを参照し、障害を起こす 1 つ前の更新状態にデータベースを戻す復旧方法

2) **ロールフォワード (roll forward)**
物理的障害が生じたとき、データ自体には問題がないので、バックアップファイルをコピーしてデータベースをある時点まで戻し、次にジャーナルファイルを参照し、障害発生前に状態の戻す復旧方法

3) **チェックポイント (check point)**
データベースの障害復旧に備え、システムを実行しているとき、ある一定のタイミングごとに記憶装置やレジスタの内容を記録した時点のこと
→ **チェックポイントファイル** …チェックポイントを記録したファイル
ロールバックやロールフォワードを行う祭、チェックポイントを参照

4) データベースの再編成
データの追加・削除・更新を繰り返すとフラグメンテーションが発生
↓
データベースの格納効率やアクセス効率の低下
↓
再編成 … データ域へ連続的にデータを格納する

④スキーマモデル (schema model)
スキーマ(schema) … データベースの論理的なデータ構造の定義を記述

1) **外部スキーマ (external schema)**
・各個人の利用目的に応じて定義することが可能
・プログラムから見たデータベースの定義の一部で、概念モデルで定義されたデータ構造の部分的な集まり

2) **概念スキーマ (conceptual schema)**
・データ処理上、実際のデータ全体の論理構造を定義したもの
・各データベースに 1 つ存在
・階層モデル，ネットワークモデル，関係データモデルが相当

3) **内部スキーマ (internal schema)**
・概念モデルで定義されたデータベースをいかにコンピュータシステム上に実現するか内部構造を定義したもの
・番地の割付やアドレッシングなどを記述

⑤データベース言語

1) **データ定義言語 (DDL：Data Description Language)**
概念スキーマや外部スキーマの定義をするデータベースの枠組みを行う

2) **データ操作言語 (DML：Data Manipulation Language)**
概念スキーマで、データの検索，修正，更新，削除などを行う

⑦その他

・コントロールブレーク
… 整列の対象となっているキー項目をコントロール項目といい、グループの変わり目のことをコントロールブレークという

・グループ制御
… 整列済みの入力ファイルにおいて、キー項目を比較することで、レコードのグループを見分けて集計等の処理を行うこと

⑥ACID 特性

トランザクション処理における特性

- **・A：Atomicity (原子性)**
  データベースに対する一連の操作をあらかじめそれ以上に細かく分けることができない 1 つの操作単位で扱うこと
- **・C：Consistency (一貫性)**
  状態変化が正しく反映されるという性質、あるいは矛盾しないという性質
- **・I：Isolationn (分離性)**
  並行して処理される複数のトランザクションが互いに干渉しないという性質
- **・D：Durability (持続性)**
  トランザクション処理がコミットすると、その後に障害が発生してもトランザクションによる効果は持続する性質

№7 SQL ★★★★☆

①SQL (Structured Query Language)

関係データベースを操作するための言語

- SQL-DDL … データの定義言語
- SQL-DML … データの操作言語

独立言語方式※や**親言語方式**※で利用可能

データベース操作言語
・親言語方式
COBOL などの親言語からデータベースを利用する方式
・独立言語方式 (ユーザ言語方式)
COBOL 等の言語とは別に独立してデータベース操作専用の言語を用意する方式
・コマンド方式
操作コマンドにより対話的にデータベースを利用する方式

②SQL の基本構成

データ定義言語 (SQL-DDL)	**CREATE SCHEMA**	スキーマの定義
	CREATE TABLE	表の定義
	CREATE VIEW	ビュー表の定義
	GRANT	処理権限の定義
	REVOKE	処理権限の解除
データ操作言語 (SQL-DML)	**SELECT**	データの抽出
	INSERT	データの挿入
	UPDATE	データの更新
	DELETE	データの削除

③SELECT 文による関係演算

SELECT 文の書式

SELECT　列名　FROM　表名　WHERE　条件

列名：抽出する列名を指定する
表名：抽出する表名を指定する（複数可）
条件：抽出する条件を指定する

例）表 A　社員№，氏名，年齢，課名
　　表 B　課名，内線番号

1) **射影**

```
SELECT　氏名，課名　FROM　表 A
```

表 A から氏名と課名の列を抽出する

2) **選択**

```
SELECT　*　FROM　表 A　WHERE　年齢>35
```

表 A から年齢が 35 より大きい行を選び、列はすべて抽出
列名に「**＊**」を指定するとすべての列を指定する

3) **結合**

```
SELECT　氏名,内線番号　FROM　表 A,表 B
　WHERE　表 A.課名＝表 B.課名
```

・表 A と表 B を結合し、氏名,内線番号を抽出
・「**＝**」で異なる表の共通項目(列)を指定し、表を１つに統合
・異なる表の共通する項目の列名は減っていてもかまわない

4) **条件設定**

```
SELECT　社員№,氏名　FROM　表 A
　WHERE　年齢>=40　OR　年齢<=30
```

・表 A から年齢が 40 以上または 30 以下である社員№と氏名の列を抽出する

5) **LIKE 設定**

```
SELECT　社員№,氏名　FROM　表 A
　WHERE　年齢　LIKE　'3%'
```

・表 A から年齢が 30 代である社員№と氏名を抽出
・**LIKE** は文字列を指定する演算子
・「**%**」は任意の文字列をあらわす

6) 文字定数の設定

```
SELECT　社員№,氏名　FROM　表 A
　WHERE　課名=N'企画'
```

・表 A から課名が企画である社員№と氏名を抽出
・文字定数を指定する時は、指定する文字を「**‘ ’**」で囲む
・日本語の文字列を指定する場合は「’’」の前に N をつける

7) 集計設定

```
SELECT　課名, AVG(年齢)　FROM　表 A
　GROUP BY　課名
```

・表 A から課名と同じ課に所属する社員の平均年齢を求める
・**GROUP BY** 句はデータをグループ化する

8) 整列設定

```
SELECT　社員№, 氏名, 年齢　FROM　表 A
　ORDER BY　年齢
```

・表 A から社員№,氏名,年齢を抽出し、年齢の若い順に並び替え
・**ORDER BY** 句はデータを並べ替える（**ASC(昇順)，DESC(降順)**）

④CREATE VIEW 文による関係演算

CREATE VIEW 文により、実表から必要な行や列だけを抽出した仮表を作ることができる

CREATE VIEW 文の書式

CREATE VIEW　ビュー表名　AS　SELECT　ｽﾃｰﾄﾒﾝﾄ

ビュー表名：新たに作られるビュー表の名前を定義する
AS：SELECT ステートメントによってビュー表を構成する列を実表から抽出する

例）表 C　社員№，氏名，営業所，勤続年数

```
CREATE VIEW　リスト表
　AS　SELECT　＊　FROM　表 C
　WHERE　営業所=N'東京'　AND　勤続年数>=10
```

・表 C から営業所が東京かつ勤続年数が 10 年以上であるデータを持つすべての列を抽出してリスト表という仮表をつくる

第 6 章　通信の仕組み

№1　通信機器と通信回路　　★★★☆☆

①構成機器

1) **DTE (Data Terminal Equipment)**
- 通信機能を備えたコンピュータや端末装置の総称

2) **DCE (Data Circuit-terminating Equipment：データ回線終端装置)**
信号変換および符号化を行う装置。ディジタル回線では DSU が、アナログ回線ではモデムが DCE になる

3) **モデム (MODEM：Modulator-DEModulator)**
- 別名：**変復調装置**
- 電話回線などのアナログ回線でデータ伝送を行うとき、DTE からのディジタル信号をアナログ信号に変換(**変調**)して回線に送信したり、その送信されたアナログ信号をディジタル信号に戻す(**復調**)装置
- データ転送速度は、一般に **56Kbps**
- モデムの規格は、ITU-T(国際電気通信連合 電気通信標準化部門)で勧告されている

4) **DSU (Digital Service Unit：宅内回線終端装置)**
- ISDN や高速ディジタル回線を用いるとき設置される DTE と通信回線とのディジタル信号や速度の変換を行う装置

5) **NCU (Network Control Unit：網制御装置)**
- アナログ回線の回線変換を使う場合に使用する装置
- モデムのほとんどが内臓

6) **CCU (Communication Control Unit：通信制御装置)**
- 通信回線とコンピュータシステムの中間に介在し、直並列変換や通信回線の監視，バッファリング，文字の組み立て，誤り制御などを行う
- CPU の負荷軽減をねらう

7) **PBX (Private Branch eXchange)**
- 企業などで内線電話同士の変換や外線電話との接続を行う交換機
- PBX を互いに専用線で接続することによって、広域の内線電話網構築が可能

②通信回線

1) **公衆通信回線 (public network)**
- 電気通信事業者が不特定多数のユーザに対して通信サービスを提供する通信回線
- 公衆電話交換網や公衆データ交換網など

2) **専用線 (leased line)**
- 利用者が電気通信事業者から借り受け、専有して使用できる回線
- 料金は定額制で、伝送速度と通信距離による課金
- 通信品質がよい

№2　通信サービスと技術　　★★★☆☆

①電気通信事業法 (telecommunications business law)

1985 年 4 月　施行

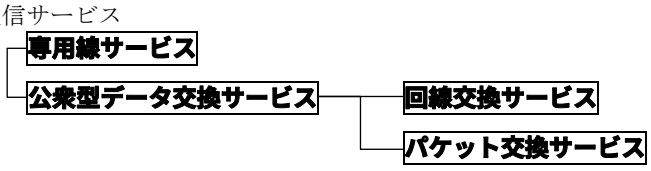
・第一種電気通信事業者 (許可制)
…自ら電気通信ネットワークを保有し、第三者に電気通信サービスを提供する

・第二種電気通信事業者 (届出制)
…第一種電気通信事業者から回線を借りて、さまざまなサービスを付加して提供する

2003 年 7 月　規制緩和

2004 年 7 月　第一種と第二種の区別がなくなり、すべて届出制

②通信サービス



1) **専用線サービス (private line service)**
- NTT などの電気通信事業者から指定先までの区間の回線を借り受け、占有利用するサービス
- 大量のデータ交換に適している

2) **回線交換サービス (circuit switching service)**
- 通信が行われるたびに、交換機を使って相手先と接続し、物理的な伝送路を確保する通信サービス
- 接続中は回線を占有
- 通信料金は接続時間により課金
- 接続相手先を選択でき、端末間の伝送遅延がないことからテレビ会議などに利用
- パケット交換サービスに比べ、大容量データ転送に向くが、長時間の使用に適さない

3) **パケット交換サービス (packet switching service)**
- パケット交換網を使い、ディジタルデータを一定長(128 または 256byte)のパケット単位に分割して伝送するデータ通信サービス
- パケットに宛先が付加されているため、個々のパケットごとに目的地に送れる
- 固定的な伝送路を確保せず、空いている回線を使って伝送
- 複数のユーザが同一の回線を共有でき、回線の利用効率は高い
- 一定時間またはある程度データが貯まらないと伝送しないため、遅延時間が発生する
- 通信料金は、伝送データ量に対して課金

4) **ISDN (Integrated Service Digital Network)**
- 従来の電話網、データ通信網、FAX 通信網などの異なるメディアの一元化を図り、統合するディジタルサービス網
- **DSU(Digital Service Unit)**と **TA(Terminal Adapter)**が必要

ⅰ.基本インタフェース
回線：電話回線
日本での名称：INS ネット 64
代表的な構成：2B+D
最大速度：144kbps

ⅱ. 1 次群インタフェース
回線：光ファイバ網
日本での名称：INS ネット 1500
代表的な構成：23B+D
最大速度：1,536kbps

※B チャンネル
…情報チャンネルとして
データ伝送に使用
D チャンネル
…制御チャンネルとして
制御情報伝送に使用

③通信技術

1) **フレームリレー (frame relay)**
- 従来のパケット交換で使われていた**プロトコル(protocol)**※の *x*.25 を簡素化し、データ転送の高速化を図った通信方法

※プロトコル … データ通信規約

- **ITU-T**(国際電気通信連合 電気通信標準化部門)で標準化

2) **ATM (Asynchronous Transfer Mode：非同期転送モード)**
- 高速･広帯域ネットワークの **B-ISDN**(Broad band-ISDN)を支える伝送交換方式
- 156Mbps の高速通信が可能
- 回線交換方式とパケット交換方式の両方の長所を持つ
- データは分割された**セル単位**(cell)※で伝送交換

※セル…すべてのデータを 48byte に分割し、それぞれの宛先の 5byte のヘッダーをつけた 53byte の固定長の単位

3) ***x*DSL (*x* Digital Subscriber Line)**
- 既存の電話回線を利用して行う高速ディジタル通信技術の名称

名称	通信	伝送速度
HDSL (High-speed DSL)	同じ (対称)	128k～1.5Mbps
SDSL (Symmetric DSL)		128k～2.3Mbps
ADSL (Asymmetric DSL)	異なる (非対称)	128k～1Mbps (上り) 1.5M～40Mbps (下り)
VDSL (Very high-speed DSL)		1.5M～26Mbps (上り) 13M～52Mbps (下り)

4) **ADSL (Asymmetric DSL)**
- 既存の電話回線(ツイストペア線)を利用
- 下り最大 40Mbps、上り最大 1Mbps
- **スプリッタ**…音声通話と通信データを分離する
- 通信データは電話交換機を通さない為、電話料金がかからずに常時接続が可能

5) **FTTH (Fiber To The Home)**
- NTT が推進する電話局から全家庭までのアクセス網を光ファイバにし、電話，テレビ，インターネットなどの統合サービスを提供する
- 最大 100Mbps の常時接続が可能

6) **VPN (Virtual Private Network：仮想私設網)**
- 公衆回線を専用回線のように利用するサービス
- 専用線よりコストが安価

№3　データ伝送の仕組み　　★★★☆☆

①データ伝送の速度単位

1) **変調速度 (baud rate)**
- アナログ回線において 1 秒間に行われる変調回数を単位とする
- 単位：**baud (ボー)**

2) **データ転送速度 (data transmission rate)**
- 単位時間に伝送されるデータ量を表す
- 単位：bps , byte/sec

②変調方式

アナログ伝送では通信回線上で送信する搬送波(carrier wave)の振幅，周波数，位相などを変調することでデータ伝送を行う

1) **AM 方式 (Amplitude Modulation：振幅変調)**
- 搬送波の振幅を変化させることでデータ伝送する方式
- 送信データの 0, 1 の値を振幅の大小に対応させて伝送
- 低速のデータ伝送に利用

2) **FM 方式 (Frequency Modulation：周波数変調)**
- 搬送波の周波数を変化させることでデータ伝送する方式
- AM 方式に比べてノイズが少なく雑音に強い
- 低速のデータ伝送に多く利用

3) **PM 方式 (Phase Modulation：位相変調)**
- 搬送波の位相を変化させることでデータ伝送する方式
  - 2 相位相変調方式 … 180 度位相をずらしたもの　1 回の変調で 1 ビットを伝送
  - 4 相位相変調方式 … 90 度位相をずらしたもの　1 回の変調で 2 ビットを伝送

4) **PCM 方式 (Pulse Code Modulation：パルス符号変調)**

アナログ信号の波形を一定間隔ごと時間で細かく区切り、計測することを**サンプリング(sampling：標本化)**という

↓

- サンプリングした波形の高さを**量子化**(値を整数化)し、2 進数に変換(**符号化**)して伝送する方式
- 電話音声 →サンプリング 1/8,000 秒　波形の高さ 8bit (256 段階)
- **CD-DA (Compact Disk Digital Audio)** →サンプリング 1/40,000 秒　波形の高さ 16bit (65,536 段階)
- ノイズに強く、情報の加工や制御が容易

③多重化方式 (multiplexing system)

1 本の高速回線に複数の端末を接続することによって経済的に利用する方式

◎多重化装置 (MUX：multiplexer)

多重化方式では、端末と通信回線の接点に MUX を置くことで、1 つの通信回線で多数の端末に同時にデータを送ることができる

1) **FDM 方式 (Frequency Division Multiplexing：周波数分割多重化)**
- 周波数帯域の広いアナログ回線で使用
- 周波数を一定の狭い周波数帯に区切り、各区間ごとに 1 チャネルずつ割り当てる
- テレビやラジオ放送と同じ原理

2) **TDM 方式 (Time Division Multiplexing：時分割多重化)**
- 高速なデジタル回線で使用
- 高速回線を細かく一定時間間隔で区切り(タイムスロット)、低速回線に各チャネルを割り当てる方式

3) **WDM 方式 (Wavelength Division Multiplexing：波長分割多重化)**
- 広帯域の光ファイバで使用
- 光は波長が異なると干渉しないという性質から、異なる複数の波長を 1 本の光ファイバケーブル上に流すことで多重化を図る方式
- 通常、2.5Gbps の伝送速度をもつ光の波長を 1 チャネルとし、4 チャネルの多重化を図った 10Gbps の超高速伝送を行う

↓

DWDM (Dense WDM)

WDM をさらに高密度化した多重化方式。1 本の光ファイバで数 Gbps～数 Tbps の超高速伝送を可能にする

№4　回線接続方式と通信方式　　★★☆☆☆

①回線接続方式

1) **ポイントツーポイント方式 (Point-to-Point system)**
- 2 台の端末を交換機や中継器を介さずに専用線だけで接続し、データの送受信を行う方式
- 1 回線を占有できるため、伝送効率は高い
- 各端末を 1 対 1 で接続し、大量のデータ通信に適しているが、端末が増えた分だけ回線が必要なためコストがかかる
- 端末間が近く、**トラフィック(traffic)**の多い場合に適する　　※トラフィック …通信ネットワークにおける通信量
- 回線制御：**コンテンシャン方式**

2) **マルチポイント方式 (multi-point system)**
- 1 回線に複数の端末を接続する方式
- 別名：**分岐接続**
- 回線距離が長く、各端末へのトラフィック量が比較的少ない場合に適しているが、端末が増えると通信時間が増える
- 回線制御：**ポーリング/セレクティング方式**

3) **集線方式**
- 情報処理装置と集線装置を 1 対 1 で接続し、集線装置と端末は別の回線で接続する方式
- ポイントツーポイント方式より経済的であるが、情報処理装置と集線装置を高速化する方式がある

4) **交換方式**
- それぞれの端末を交換網に接続する方式

②通信方式

1) **単方向通信 (simplex)**
- 情報があらかじめ決められた一方向にのみ伝送され、その逆からの伝送はできない
- 伝送路は 2 線式回線

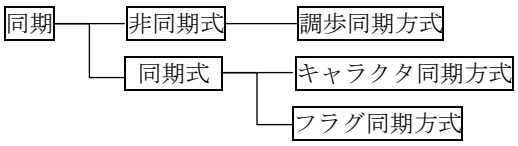
2) **半二重通信 (half duplex)**
- 送信と受信を切り替えることにより、両方向から情報を伝送することが可能
- 伝送路は 2 線式回線

3) **全二重通信 (full duplex)**
- 両方向から同時に送受信ができる方式
- 伝送路は一般に 4 線式回線（2 線式回線も可）

№5　同期制御と誤り制御　　★★★★☆

①**同期方式 (synchronization system)**

同期 … データを正しく伝送するために、送信側と受信側の間でタイミングを取ること



1) **調歩同期方式 (start-stop synchronization system)**
- 送信する文字ごとに先頭にスタートビット"0"，末尾にストップビット"1"を付加して伝送する方式
- 別名：**スタートストップ同期方式**
- 1 文字 8bit の場合、1+8+1= 10bit
- 低速通信で利用

2) **キャラクタ同期方式 (character synchronization system)**
- 送信する連続データの前に、同期を示す**「SYN」制御キャラクタ**を付加して伝送する方式
- 別名：**SYN 同期方式**
- 通常、同期の認識を確認するために、2 個以上の SYN 符号を付けて送信する
- ベーシック手順の伝送制御手順に利用

3) **フラグ同期方式 (flag synchronization system)**
・伝送するデータブロックごとの最初と最後に、フラグパターンの「01111110」を付加して、データの区切りを示す方式
→ データブロック … **フレーム**
・別名：**フレーム同期方式 (frame synchronization system)**
・文字だけでなく画像データなど、他の異なったビット長の符号も送ることが可能
・HDLC 手順の伝送制御手順に利用

②誤り制御
データ伝送を行うとその間に様々なひずみや雑音が生じ、その影響でデータの値が変化してしまう（**ビット誤り**）

1) **再送訂正方式**
i . **パリティチェック(奇偶検査) (parity check：誤り検査)**
・伝送されるデータ(ビット列)に**パリティビット**を 1bit 加え、各データ列の”1”のビットの数が偶数(または奇数)になるようにパリティビットを調整する方式

ii . **CRC (Cyclic Redundancy Check：巡回冗長検査)**
・情報ビット列から特定の式(**生成多項式**)を用いて、検査ビット値を生成して、情報ビット列に付加する方式
高速かつ検査精度が良いため、外部記憶装置とのデータ転送での誤り検出等に広く利用
・伝送されるビット列にある桁数のビット列(**冗長ビット**)を加える
・1bit 誤りだけでなく、連続誤り(バースト誤り)を検出可能

2) **自己訂正方式**
ビット誤りを検出したら誤り訂正をする方式

i . **水平垂直パリティチェック**
・垂直パリティ … 各データ列に 1 つのパリティビットを付加
・水平パリティ … 伝送するブロック単位ごとに各データの同じ位置(水平)にパリティビットを付加
↓
1 つのブロックに 1 文字分の**パリティデータ**が作成
(**BCC：Block Character Check**)
・1bit の誤り → 誤りを検出し訂正
・2bit の誤り → 誤りの検出のみ

ii . **ハミングコード(ハミング符号方式) (humming code)**
・情報ビットのほかに数ビットのチェック用ビットを付加
→ 所定の計算を行い、ビット誤りを起こした位置を検出し自動訂正する
・検査点(*m* ビット)、情報点(*n* ビット)とすれば、検査行列[*m*, *n*]と情報点から検査点のビット値を定める方式
1 ビットの誤りが訂正可能
・2bit の誤りは検出できるが訂正できない

№6　伝送制御手順　　★★★☆☆

①伝送制御
1) **同期制御**
データが正しく伝送されるように、送信者側と受信者側で同期を取る
2) **誤り制御**
伝送されたデータに誤りがないか検出したり、検出された誤りを訂正する
3) **回線制御**
モデムや DSU といった DCE の監視制御，変換回線における回線の接続と切断を行う
4) **データリンク制御**
通信相手を確認し送受信者との間に物理的な伝送路(**データリンク**)を確保して，データ伝送ができるようにする(**データリンクの確立**)

②伝送制御手順 (transmission control procedure)
データ転送の手続き
1. 回線接続（回線制御）
2. データリンクの確立（データリンク制御）
3. データ転送（同期制御，誤り制御）
4. データリンクの解放（データリンク制御）
5. 回線切断（回線制御）

伝送制御手順	同期制御	誤り制御	伝送速度
無手順	ビット同期方式	なし	低速
ベーシック手順	キャラクタ同期方式	ﾊﾟﾘﾃｨﾁｪｯｸ	中速
HDLC 手順	フラグ同期方式	CRC	高速

③無手順 (non-procedure：たれ流し)

・非同期の調歩同期方式
・打ち込んだデータがそのまま通信回線に流れる
・会話型のサービスに適している

④ベーシック手順 (basic mode data transmission control procedure)

・10 種類の伝送制御キャラクタを使用したキャラクタ同期方式
・受信者側は正常であれば「**ACK**」、異常であれば「**NAK**」の応答を行う
・データリンクの確立にはポーリング/セレクティング方式とコンテンション方式が選択可能

◎コンテンション方式 (contention sysytem)

・ポイントツーポイント方式で接続されている場合
・送信要求を早く出したほうが主局となりデータ伝送の制御を行う
・別名：**早い者勝ち方式**
・送信側が受信側に「受信可能か」を問い合わせ、受信側から「受信可能」を受信したときデータを送る

◎ポーリング/セレクティング方式 (polling/selecting system)

・マルチポイント方式のように 1 本の回線から多くの端末が接続されている場合に用いられる方式

ⅰ.**ポーリング (polling)**
ホスト側(制御局)が接続されている各端末(従局)に送るデータがあるかを聞いて回る方式

ⅱ.**セレクティング (selecting)**
ホスト側からある端末に送りたいデータがあるとき、その端末に「受信準備 OK」を確認する方式

⑤HDLC 手順 (High-level Data Link Control procedure)

・フラグ同期方式で同期を取り、高速なデータ転送を行う
・CRC 誤り制御によりベーシック手順より高い信頼性
・複数相手と同時通信が可能
・OSI 基本参照モデルのデータリンク層のプロトコル

◎1 次局と 2 次局
… HDLC 手順では制御局を**1 次局**、従属局を**2 次局**と呼ぶ

◎HDLC 手順のデータリンクの確立方式

1) **不平衡型データリンク**
1 次局と 2 次局の間でデータの授受を行う構成

a. **正規応答モード**（**NRM**：Normal Response Mode）
ポーリング/セレクティング方式にあたる方式
1 次局の送信許可が出たときのみ、2 次局からのレスポンスを送ることができる

b. **非同期応答モード**（**ARM**：Asynchronous Response Mode）
コンテンション方式にあたる方式
いつでも 2 次局からレスポンスを送ることができる

2) **平衡型データリンク**（**ABM**：Asynchronous Balanced Mode）
1 次局と 2 次局の機能を持つ複合局のみで構成される
複合局はいつでもコマンドやレスポンスを送信することができる

◎フレームの構成

フラグシーケンス(F)	フレームの開始と終了を示す。 [01111110] の 8bit からなる
アドレスフィールド (A)	コマンドでは宛先となる 2 次局のアドレス レスポンスでは発信元となる 2 次局のアドレス
制御フィールド (C)	フレームの種類，送信順序番号， 受信順序番号など各種制御情報
情報フィールド (I)	送信データ
フレームチェックシーケンス (FCS)	CRC 方式によるチェックビット

◎コマンドとレスポンス

・**コマンド** … 1 次局から 2 次局に送られるデータ
・**レスポンス** … 2 次局から 1 次局に送られるデータ

第 7 章　ネットワークシステム

№1　ネットワークアーキテクチャ　　★★★★★

①プロトコル (protocol)

データ通信を行うために必要な通信規約

②ネットワークアーキテクチャ (network architecture)

データ通信に必要となる全ての機能を階層的に体系化し、まとめたもの

ネットワークの階層化

→機能別にまとめられているというだけでなく、プロトコルの変更が生じた際に全体を修正をしないでも階層単位で効率的に変更できる利点がある

③OSI 基本参照モデル (OSI basic reference model)

ISO がオープンシステムの相互接続に関するとりまとめをした **OSI(Open System Interconnection：開放型システム間相互接続)**の 7 階層モデル

- 下位層 … 第 1 層～第 4 層
  役割：いかに高品質なデータを送るか
- 上位層 … 第 5 層～第 7 層
  役割：下位層の機能を利用した通信サービスの提供

	層	層の名称
上位層	第 7 層	アプリケーション層
	第 6 層	プレゼンテーション層
	第 5 層	セッション層
下位層	第 4 層	トランスポート層
	第 3 層	ネットワーク層
	第 2 層	データリンク層
	第 1 層	物理層

1）第 1 層：**物理層 (physical layer)**
- ビット単位(0, 1)の表現のしかた，伝送方式，コネクタのピンの数や大きさ，電気的仕様などネットワークで使用する機器の物理的状態を規定
- RS-232C，モデム，DSU など

2）第 2 層：**データリンク層 (data link layer)**
- 隣接した装置間のデータ管理やデータの誤り検出などを規定
- HDLC 手順や LAN における CDMA/CD 方式，トークンパッシング方式，FDDI など

3）第 3 層：**ネットワーク層 (network layer)**
- 経路設定を行うルーティング機能とその中継機能を規定
- インターネットの IP 層

4）第 4 層：**トランスポート層 (transport layer)**
- 透過的な伝送路を提供し、メッセージの紛失や順序の乱れの補正などを規定
- インターネットの TCP 層

5）第 5 層：**セッション層 (session layer)**
- 同期調整，半二重，全二重などの通信方式の選択を規定

6）第 6 層：**プレゼンテーション層 (presentation layer)**
- データの表現方式(データのタイプや符号)の変換やテキスト圧縮，暗号化などを規定

7）第 7 層：**アプリケーション層 (application layer)**
- 応用プログラムが存在し、そのサービスをどのように実行するかを規定
- 別名：**応用層**

④**TCP/IP (Transmission Control Protocol/Internet Protocol)**

インターネットの標準として、DARPA(米国防総省高等研究計画局)が開発したプロトコル

1) **TCP (Transmission Control Protocol)**

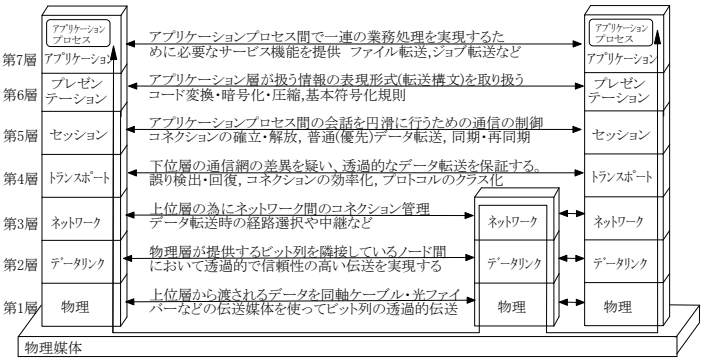
・OSI のトランスポート層に該当
・データの管理や誤り検出などを行って、パケットを転送
・データの送受信を行う端末間に論理的な伝送路を確立させる**コネクション型(connection oriented type)**
・高い信頼性

2) **IP (Internet Protocol)**

・OSI のネットワーク層に該当
・IP アドレスを元に経路を選択してパケットを転送
・データの送受信を行う端末間に論理的な伝送路を持たない**コネクションレス型(connectionless oriented type)**
・高速伝送が可能，TCP より信頼性は低い

3) TCP/IP と OSI 基本参照モデルの比較

TCP/IP		OSI 基本参照モデル
ｱﾌﾟﾘｹｰｼｮﾝ層		第 7 層
		第 6 層
		第 5 層
TCP		第 4 層
IP		第 3 層
ﾈｯﾄﾜｰｸｲﾝﾀﾌｪｰｽ層		第 2 層
		第 1 層



⑤**DHCP (Dynamic Host Configuration Protocol)**

・TCP/IP 環境でネットワークを接続するとき、接続する各コンピュータに IP アドレスなどの必要な情報を自動的に割り当てるプロトコル
・IP アドレスの管理を簡素化

№2　LAN の構成　　★★★☆☆

①**LAN (Local Area Network)**

・同一建物など限られた空間内で、コンピュータと関連機器を接続するためのデータ伝送システム
・ファイルやプリンタを共有することで資源の有効活用

②伝送媒体

1) **ツイストペアケーブル (twisted pair wire：より対線)**

・2 本の銅線をより合わせたケーブル
・安価で取り扱いが容易
・伝送速度は最大 1,000Mbps、伝送距離は数百 m

2) **同軸ケーブル (coaxial cable)**

・1 本の銅線を絶縁体で包み、それを円筒の中に入れ、さらに被覆したケーブル
・ツイストペアケーブルより雑音に強く、減衰も少ない
・伝送速度は 10～数百 Mbps、伝送距離は 185m～数十 km
・10BASE2，10BASE5 など

3) **光ファイバケーブル (optical fiber)**

・プラスチックまたはガラスで作られた髪の毛ほどの細い管を被覆したケーブル
・中継器なしで高速で長距離伝送が可能
・伝送速度は数百 Mpbs、伝送距離は最大 10km
・中規模，大規模の高速 LAN で利用

③LAN モデル

1) **クライアントサーバモデル (client server model)**

※**クライアント (client)** … サービスを受けるコンピュータ
サーバ (server) … クライアントから要求されたサービスを提供

・クライアントとサーバが分散して処理を行うため、それぞれのコンピュータの負荷を軽減できる
・サーバ →
LAN システム全体にまたがってデータ処理の基盤を支え、効率的に運用するための各種サービスを行う
例)ファイルサーバ，プリントサーバ，データベースサーバなど

ⅰ. **2 層クライアントサーバシステム**

GUI の関連処理やデータの加工処理をクライアント側で行う
↓
大量のデータがネットワークを通じて流れるためトラフィックが混雑
↓

ⅱ. **3 層クライアントサーバシステム**

クライアント側では GUI の関連処理のみを行い、サーバ側でデータの加工処理を行う

層	プレゼンテーション層	GUI の実現
	ファンクション層	データの加工処理
	データ層	データベースのアクセス

ⅲ. **CORBA (Common Object Request Broker Architecture)**

クライアントサーバモデルにおいて、分散環境化のもとでオブジェクト同士がメッセージ交換するための共通仕様で、プラットフォームやプログラム言語に依存しない

2) **ピアツーピア LAN (Peer-To-Peer LAN)**

・サーバとクライアントの区別がなく、両方の機能を兼用
・サーバとしての負荷の少ない小規模向けのパソコン LAN
・専用サーバ不要で、導入･管理が容易，低価格で構築可能
・クライアントサーバモデルに比べデータの保護や機密保護機能は弱い

3) **無線 LAN (wireless LAN)**

・無線 LAN カードを装着し、電波や赤外線を利用して通信する LAN

規格	周波数	伝送速度
IEEE802.11a	2.4GHz	11Mbps
IEEE802.11b	5.2GHz	54Mbps
IEEE802.11g	2.4GHz	54Mbps

電波の盗聴対策
ⅰ. **WEP (Wired Equivalent Privacy)**
ⅱ. **WPA (Wi-Fi Protect Access)**

・端末の場所が固定されず、無線の届く範囲なら比較的自由に移動可能

④その他

・**コンテンション**

…ネットワークシステムで、複数の端末が双方向の通信を介して送信要求を行う場合のアクセス方式

№3　LAN 接続方法　　★★★★★

①**トポロジ (topology)**

ネットワークの接続形態

ノード (node) … ネットワークを構成するサーバや端末、通信制御装置など

1) **スター型ネットワーク (star network)**

・中心となるサーバに全てのノードを直接つなぐ方式
・**ハブ(hub)**を用いたネットワークも該当
・システム構成は容易だが、混雑時にサーバに負荷がかかり能力が低下
・サーバの故障がシステム全体に影響を及ぼす

2) **バス型ネットワーク (bus network)**

・1 本の基幹となるケーブル(**バス**)に端末をランダムにつなげていく方式
・CSMA/CD LAN やトークンバス LAN がこの方式
・データはケーブル上を両端に向かって左右に流れ、途中に受信したい端末があればそのデータを受信し、末端まで行ったデータは消滅
・終端で電気信号が反射して雑音になるのを防ぐために**ターミネータ(終端抵抗器)**が必要
・伝送速度は中高速で、中小規模向け LAN
・リング型 LAN より低コストで、増設や拡張が容易
・ケーブルが 1 本しかないので、2 つ以上の端末からデータが送信されるとデータが**衝突(collision：コリジョン)**が発生する

3) **リング型ネットワーク (ring network)**

・リング状にケーブルをつなぎ、そこに全ての端末を接続する方式
・FDDI やトークンリング LAN がこの方式で、比較的大規模 LAN に用いられる
・データはリング上を一定方向に回り、バス型のようにデータ衝突は発生しない
・ケーブルが輪になっているため、1 箇所が切断されても支障はない
・通信効率はよいが、システムが複雑でコスト高

②LAN 間接続装置

1) **リピータ (repeater)**
- OSI 基本参照モデル第 1 層(物理層)での接続装置
- 電気信号の再生増幅を行う
- 通常、CSMA/CD のバス型 LAN に使用
- リピータ結合した LAN 同士は、１つの大きな LAN になるため、トラフィック量が増えると遅延時間が生じる

2) **ブリッジ (bridge)**
- OSI 基本参照モデル第 2 層(データリンク層)での接続装置
- アクセス制御の異なる CSMA/CD 方式とトークンリング方式の相互接続が可能
- 個々の端末の情報をデータベースとして内蔵し、管理する
- 不必要なデータの流れによるトラフィック量増大を防ぐ**フィルタリング機能**を備える

3) **ルータ (router)**
- OSI 基本参照モデル第 2 層(データリンク層)での接続装置
- 機能は基本的にブリッジと同じ
- 最も好ましい伝送ルートを選択する**ルーティングアルゴリズム機能**を備える

4) **ゲートウェイ (gateway)**
- OSI 基本参照モデル全階層を認識
- プロトコルが異なった異機種の LAN でも相互接続可能
- 機能は基本的にルータと同じ

5) **ハブ (hub)**
- LAN を構成する際に使用される集線装置
- ハブを直列接続する**カスケード接続(cascade joint)**が可能

層	名称	接続･中継する装置
全ての層	物理層～アプリケーション層	ゲートウェイ（異プロトコル間接続）
第 3 層	ネットワーク層	ルータ（機種 LAN 間接続）
第 2 層	データリンク層	ブリッジ（同一 LAN 間接続）
第 1 層	物理層	リピータ（配線延長）

№4　LAN のアクセス制御方式　　★★★★★

①イーサネット (Ethernet)
- 1980 年に米国の XEROX 社,DEC,Intel の 3 社が共同開発した IEEE802.3 に準拠したバス型の LAN 製品のこと
- アクセス制御方式は CSMA/CD 方式
- 伝送速度は 10Mbps、ファースト Ethernet (100Mbps)，ギガビット Ehternet (1,000Mbps)

②FDDI (Fiber Distributed Data Interface)
- ANSI (American National Standards Institute：米国規格協会)で標準化された二重トークンリング LAN
- 光ファイバによる 100Mbps，最大伝送距離は 100km

③LAN 規格

規格名	伝送速度	セグメントの最大長	ケーブルの種類	制御方式
10BASE2	10Mbps	200m	同軸	CSMA/CD
10BASE5	10Mbps	500m	同軸	CSMA/CD
10BASE-T	10Mbps	100m	ツイストペア	CSMA/CD
10BASE-F	10Mbps	500m	光ファイバ	CSMA/CD
100BASE-TX	100Mbps	100m	ツイストペア	CSMA/CD
100BASE-TX	100Mbps	10km	光ファイバ	CSMA/CD
1000BASE-T	1Gbps	100m	ツイストペア	CSMA/CD
1000BASE-SX	1Gbps	550m	光ファイバ	CSMA/CD
FDDI	100Mbps	200km	光ファイバ	ﾄｰｸﾝﾊﾟｯｼﾝｸﾞ

④CSMA/CD 方式 (Carrier Sense Multiple Access with Collision Detection system)
- バス型 LAN に用いられる
- データの転送はパケット単位で双方向に流し、流したパケットはネットワークの末端で消滅する
- CSMA/CD 方式は Ethernet の仕様をベースとしており、CSMA/CD 方式そのものを Ethernet と呼ぶこともある

・データ転送速度は 10Mbps～1Gbps
・パケットのデータ長は 50～1504 ビット
・ケーブルと端末の間には、ケーブル上の電流の流れをチェックし、使用中かあいているかを常時監視するボードがある
↓
このボードにより、流れてきた電流(データ)の宛先が自分宛のデータであれば取り込む
・データの衝突(collision)が発生した場合、データが消滅するため、各端末で感知し、しばらく時間を置いてから再度送信する

⑤トークンパッシング方式 (token passing system)

・主にリング型 LAN に用いられる
・データ伝送は、トークンという送信するデータを載せるものがケーブル上を巡回する仕組み
→CSMA/CD 方式のように、コリジョンは発生しない

フリートークン

…トークンにデータが載っていない状態

ビジートークン

…送信先の宛先とデータを載せている状態

↓
各端末は、自分宛のビジートークンが回ってきたとき、そのデータをコピーして取り込む
↓
送信先の端末に戻るとフリートークンとなる
・IEEE802 によって標準化

⑥TDMA 方式

各端末にタイムスロットという通信可能な時間を与え、データの衝突が発生しないように順番に送信する方式

№5　インターネットの機能　　★★★★☆

①IP 接続 (IP protocol connection)

1) インターネットサービスプロバイダ (ISP：Internet Service Provider)

・インターネットの接続サービスを行っている会社
・通常 TCP/IP を通信プロトコルとした接続を行う

2) 専用線 IP 接続

・専用線を利用し、インターネットと常時接続する場合の接続形態
・利用料金はダイヤルアップ IP 接続と比べて高いが高速通信,常時接続が可能

3) ダイヤルアップ IP 接続

・電話回線や ISDN などの公衆回線を利用し、モデムや TA を介してインターネットと接続する場合の接続形態
・最近は常時接続可能な電話回線を利用した ADSL が多く利用
・専用線 IP 接続に比べると通信速度は遅いが、利用料金が安い
・ダイヤルアップ接続では PPP(Point to Point Protocol)を使用

②IP アドレス (Internet Protocol address)

・TCP/IP でインターネット接続されているコンピュータを識別するためのアドレス
・32bit で構成され、同一アドレスがない一意性のもの
・**IP アドレス ＝ ネットワークアドレス部＋ホストアドレス部**

クラス	ﾈｯﾄﾜｰｸｱﾄﾞﾚｽ部	ﾎｽﾄｱﾄﾞﾚｽ部
A	8 bit	24 bit
B	16 bit	16 bit
C	24 bit	8 bit

1) サブネットマスク (subnet mask)

・IP アドレスと同じ 32 ビットのビット列
・IP アドレスをネットワークアドレスとホストアドレスに分けるためのビットパターン
・ネットワークアドレスは、LAN 内のデータのやり取りには無関係
↓
IP アドレスの上位を「マスク」で隠してしまう
↓
マスクで隠されなかった部分が同じサブネットのホスト数に相当
(具体的にはサブネットマスクと IP アドレスの理論積)
※サブネットに分けないときもサブネットマスクの指定は必要
サブネットに分けるためには、サブネットアドレス部分をホストアドレス部分から借りる必要がある
↓
クラス A～C という枠組みを越えて、ホスト部分の一部をネットワークアドレスとして扱う

▼サブネットマスクとネットワーク数，ホスト数

サブネットマスク	ネットワーク数	IPアドレス数	ホスト数
255.255.255.0	1	256	254
255.255.255.128	2	128	126
255.255.255.192	4	64	62
255.255.255.224	8	32	30
255.255.255.240	16	16	14
255.255.255.248	32	8	6
255.255.255.252	64	4	2

▼デフォルトサブネットマスク

クラス	ビットパターン	10進数形式
A	11111111 00000000 00000000 00000000	255.0.0.0
B	11111111 11111111 00000000 00000000	255.255.0.0
C	11111111 11111111 11111111 00000000	255.255.255.0

［関連］IPv6　：128ビットからなる次世代のIPアドレス

2) **プライベートIPアドレス**

LANのように外部とは接触を持たない閉鎖的な範囲で使用するIPアドレス

※**グローバルIPアドレス** …公式に管理されているアドレス

3) **NAT (Network Address Translation)**

ルータを通してプライベートIPアドレスとグローバルIPアドレスを「1対1」で相互変換する技術

4) **IPマスカレード (IP masquerade)**

ルータを通じて1つのグローバルIPアドレスを複数のポート番号に割り当て、複数のプライベートIPアドレスで同時接続することを可能にする技術

別名：**NAPT (Network Address Port Translation)**

③インターネットサービス

1) **WWW (World Wide Web)**

・世界中のWWWサーバに登録されているWebページがクモの巣のように互いに結びつきあい、誰でも閲覧できるように公開した大規模データベースシステム

・閲覧には**WWWブラウザ**が必要

2) **電子メール (electronic mail)**

・インターネットを通じてメッセージ交換するシステム

ⅰ. **SMTP (Simple Mail Transfer Protocol)** … 送信者側の使用するプロトコル

ⅱ. **POP3 (Post Office Protocol version 3)** … 受信者側の使用するプロトコル

3) **FTP (File Transfer Protocol)**

・TCP/IPにおけるファイル転送をするためのプロトコル

・インターネット上に多く存在するファイルをダウンロードする際などに利用する。最近はファイル転送サービスそのものを指す

4) **Net News**

・インターネット独自の電子掲示板システム

・1つのニュースサーバにニュースを登録すると世界中のニュースサーバに瞬時に登録される

・ニュースリーダというソフトウェアが必要

5) **Telnet**

・インターネットを使って遠隔地にあるコンピュータを操作する為のプロトコル

・**SSH (Secure SHell)**

…ネットワーク上を流れるデータを暗号化し、安全性を高める技術

④URL (Uniform Resource Locator)

インターネット上の情報資源の所在を特定するための固有のアドレス

・**HTTP (Hyper Text Transfer Protocol)**

Webブラウザと Webサーバ間において、HTMLファイルなどの文書を転送するために用いられるアプリケーションレベルのプロトコル。サーバ側のポート番号は80番が予約されている。

⑤**DNS (Dynamic Name System)**

最大 12 桁の IP アドレスを意味のある文字(ドメイン名)にインターネット上で変換するシステム

※DNS サーバ …変換を行っているサーバ

№6　Web ページの作成　　★★★★☆

①**マークアップ言語 (Markup Language)**

文章だけでなく、文字の装飾やレイアウト情報、ハイパーリンク情報などを言語の書式にのっとって文書中に記述していく、画面表示や印刷のための言語

②データの圧縮

1) **JPEG** (Joint Photographic Experts Group)
- 静止画データの圧縮･伸張に関する世界的な標準規格
- 圧縮率は 1/10～1/50 で、RGB256 階調(約 1677 万色)が扱える
- **非可逆符号化方式** (伸張したときに完全には元の画像に戻らない)

2) **GIF** (Graphics Interchange Format)
- 静止画の符号化方式
- 圧縮率は数分の 1 で、色数は 256 色まで減色 → イラスト調の画像に向く
- インターレース GIF …画像全体を徐々に細分表示していく
- 透過 GIF …背景を透かして見せる
- アニメーション GIF …複数の画像を合成して動画表現する

3) **PNG** (Portable Network Graphics)
- JPEG や GIF に代わり、普及を目指して開発されたインターネットの Web ページの画像形式
- GIF 形式より圧縮率は高く、インターレース，透過，アニメーションなども対応
- フルカラー(約 1677 万色)が扱える
- **可逆符号化方式**

4) **MPEG** (Motion Picture Expert Group)
- 動画の圧縮･伸張の世界標準符号化方式
- **MPEG1** …CD-ROM やビデオ CD など
- **MPEG2** …HDTV(高精度テレビ)や DVD など

③Web ページのイタラクティブ利用

1) **CGI (Common Gateway Interface)**
- クライアント側の WWW ブラウザからの要求により、WWW サーバ上の CGI プログラムを実行させ、その結果を HTML 形式でクライアントに送信するためのインタフェース
- Web アンケートや掲示板など
- 開発言語は、Perl や C など

2) **SSI (Server Side Include)**
- HTML 文書内に WWW サーバで処理する命令やデータを埋め込み、事前に送っておき、クライアントサーバが WWW サーバに接続したとき、WWW サーバ側でそれを実行する技術
- アクセスカウンタ，日付や時刻など

3) **サーブレット (servlet)**
- サーバ側で実行する Java プログラムのことで、Web サーバの機能を拡張することが可能になる
- CGI との違いは、一度呼び出されたものはメモリに常駐するため、高速処理が可能で、データを永続的に扱え、複数のユーザ間で情報を共有できる

※アプレット
- 本来は小さいアプリケーションプログラムの意味
- コンパイル済みのオブジェクトコードがサーバに格納されていて、クライアントからの要求によってクライアントへ転送されて実行される

サーブレット
- Web サーバ側で動作する Java プログラム(小さなアプリケーション部品)のこと
- 通常は動作の結果を HTML としてクライアント側に送る

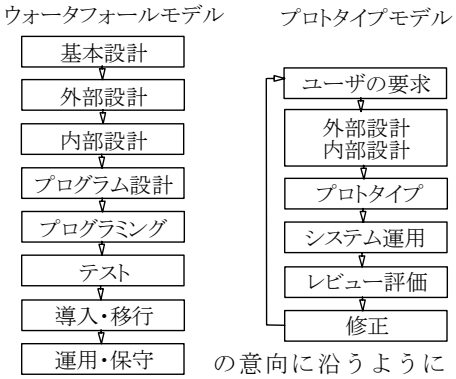
④その他
- **VRML** (Virtual Reality Modeling Language)
  …インターネット上で扱う 3 次元コンピュータグラフィックスを記述するための言語仕様

第 8 章　システム開発

№1　システム開発手順　　★★★★☆

①システム開発モデル（system development model）

1) **ウォータフォールモデル (water fall model)**
- 上流工程の基本計画から下流工程の運用･保守までの各工程を 1 工程ずつ順次開発していく手法
- 工程ごとに仕様書を作り、具体化していくため、管理しやすく大規模開発に向く
- 開発の初期段階からユーザの要求を的確に取り込まなければならない
- 前工程の不具合が次の工程に拡大して影響する
- 工程ごとに作業終了の確認作業を行わないと次工程に進めないことによって、工程間の並行作業ができない



2) **プロトタイプモデル (prototype model)**
- システム開発の初期段階で、**プロトタイプ(試作品)**を作成する手法
- ユーザは操作性などの問題点を**レビュー評価**し、開発側はそのユーザの意向に沿うようにプログラムの変更などを行う。
- 最終段階でのユーザと開発側との食い違いを極力減らすことができる
- 比較的小規模のシステムに適している

3) **スパイラルモデル (spiral model)**
- 開発するシステムを比較的独立性の高いサブシステムに分け、個々に設計，プログラミング，テストを行う手法
- ウォータフォールモデルとプロトタイプモデルの特長を活かした手法
- 同時開発する規模を抑制し、開発要員の一時的な増大を防ぐ利点がある

②**CASE ツール (Computer Aided Software Engineering Tool)**
- コンピュータ支援によるシステム開発をするソフトウェア
- システム開発に必要な要求仕様や設計情報などをデータベースで一元管理し、各工程を自動作成する

1) **上流 CASE ツール**
基本設計，外部設計，内部設計を支援する

2) **下流 CASE ツール**
プログラム設計，テストなどの下流工程を支援

3) **保守 CASE ツール**
保守を支援する（リエンジニアリング機能など）

4) **共通 CASE ツール**
システム開発全体で行われる作業を支援
（文書や図表などの文書化，プロジェクト管理など）

5) **統合 CASE ツール**
システム開発工程全体を支援

6) **開発プラットフォームサービス提供 CASE ツール**
各 CASE ツールの統合化を支援

※リポジトリ (repository) …CASE ツールの各種情報を格納するデータベース

③**リエンジニアリング (reengineering)**
- すでに稼動しているシステムを分解･解析し新しいシステムを作成する為の技術
- 保守 CASE ツールで利用されている

1) **リバースエンジニアリング (reverse engineering)**
すでに稼動しているプログラムやファイルなどを分解し、利用されている設計仕様を明らかにする技術

2) **フォワードエンジニアリング (forward engineering)**
リバースエンジニアリングから得た仕様から新たなシステムを開発する技術

№2　外部設計　　★★★★★

①基本計画

・システム開発する業務内容を調査し、問題点などの分析によって、システム化への必要条件を明確にする
・基本計画をもとに外部設計に入っていく

○システム化計画
現状調査と問題点を分析し、システム化を検討する

○プロジェクト実行計画
プロジェクトを組織し、開発の見積もりやスケジュールを決める

○要求定義
DFD や E-R 図を使って、システム要求を明確にし、要求仕様書をまとめる

②外部設計 (概要設計)

・ユーザ側から見たシステム設計
・コンピュータとユーザの接点となる**ユーザインタフェース**の部分が中心

1) 外部設計の手順

ⅰ. 要求仕様の確認
↓　基本計画で定義された内容を元にユーザの要求を確認
ⅱ. サブシステムの定義と展開
↓　システムをサブシステムに分割して定義する
ⅲ. 画面設計･帳票設計
↓　ﾕｰｻﾞｲﾝﾀﾌｪｰｽを考慮した画面レイアウトや帳票の入出力設計を行う
ⅳ. コード設計
↓　システム内で使用する情報をコード化する
ⅴ. 論理データ設計
↓　データ項目やデータ構造を決める
ⅵ. 外部設計書の作成
内部設計にスムーズに引き継げるよう外部設計書をまとめる

2) **要求仕様の確認**

・ユーザの立場に立って行う → ユーザ部門とスタッフの参加も必要
・ユーザ部門ごとのシステム化における要求事項を調査し、ユーザ要件をまとめる
・DFD などの業務フロー図を使ってユーザ要件を整理する

※ペトリネット
…ペトリによって提案された事象応答分析に基づいた要求モデル。並列的に動作する機能間どおしの同期を表現することができることから、制御中心のシステムの要求分析に適したモデル

3) **サブシステムの定義と展開**

・サブシステムを定義し、システムにかかる負荷の分散を図る
・サブシステムの展開として、サブシステム間のインタフェースを明確にする

4) **画面設計･帳票設計**

・入力のしやすさや見易さを考慮し、ユーザインタフェース設計を行う

ⅰ. 画面設計
・画面設計の標準化を決める
・画面フロー設計を決め、画面展開を確認する
・画面レイアウト設計を行い、統一性，操作性を確認する

ⅱ. 帳票設計
・出力項目や必要な帳票を決める，出力方式と出力媒体を決める
・帳票のレイアウトを確認する

5) **コード設計**

・コード化する情報を抽出し、コード設計を行う
連番コード … データに一連の番号を振り分ける
区分コード … 分類項目を一連の番号であらわす
桁別コード … コードの各桁に意味をつける
表意コード … 文字や数字で認識番号をつける

6) **論理データ設計**

ⅰ. データ項目の洗い出し
↓ 業務で使われているデータ項目を洗い出し、必要な要素をチェックする

ⅱ. データの関連性の分析
↓ 画面や伝票などで使われるデータの関連性を分析し、同時使用の多いデータ要素ごとにグループ化する

ⅲ. ファイル候補の決定
↓ グループ化されたものがファイル候補となり、そのグループに属するデータ要素がデータ項目となる

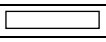
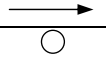
ⅳ. データ項目と処理の対応
↓ 処理を行う際にデータ項目に漏れがないかデータ項目と処理の対応

ⅴ. ファイル仕様の作成
ファイル名,キー項目,属性,桁数などファイルの仕様を決める

7) **外部設計書の作成**
外部設計の内容をまとめる

③DFD (Data Flow Diagram)

・システム化や業務効率の改善を図る際に、業務の流れを図式化し、**業務分析**を行う手法

	外部のデータの入力部・出力部
	データの流れ
	データの処理
	データを保存するファイル

④E-R 図 (Entity Relationship diagram)

実体(Entity：エンティティ)とその**関連(Relationship：リレーションシップ)**を図式化し、データの構造を分析するもの。実体とその関連は**属性(Attribute：アトリビュート)**を持つ

記号	名称	意味
	実体	人,物,場所,情報など
	関連	実体と実体の間にある相互関係
	属性	実体や関連に関わるデータ項目

№3 内部設計 ★★★★☆

①内部設計 (詳細設計)

・外部設計案に基づいて、システム開発の立場でコンピュータ側から見たシステム設計を行うこと
・分割したプログラム間の処理や流れを円滑にする為のプログラム間インタフェース部分が中心となる

外部設計書の確認
↓ 外部設計の仕様をもとに、どのようにシステム開発を行うか確認する
機能分割･構造化
↓ サブシステムの機能をプログラムに分割する
物理データ設計
↓ ファイル編成やレコードレイアウト，媒体などを設計する
入出力詳細設計
↓ 具体的な入出力項目の配置，GUI の設計，入出力媒体，入力データのチェック方式などを設計する
内部設計書の作成
次の工程にスムーズに引き継げるよう内部設計書をまとめる

②入出力詳細設計

1) **GUI の画面設計**

※GUI … Graphical User Interface

ユーザインタフェースの設計として GUI を使った画面設計を行う

ラジオボタン	複数項目から１つを選択する。○などにチェックを入れる
チェックボックス	複数の項目を選択する。□などにチェックマークを入れる
リストボックス	選択肢をリスト表示して、その中から選択させる
プルダウンメニュー	ある項目を選択すると、その項目に属したサブメニューが表示される
ポップアップメニュー	マウスの右クリックなどによって表示されるメニュー
スクロールバー	画面をスクロールするのに使われる
スライダー	メモリ表示されたボタンを操作してボリューム調整などをする

2) 入力データのチェック方式

ⅰ. **カウントチェック (件数検査)**
データ件数が妥当かチェックする

ⅱ. **ニューメリックチェック (数字検査)**
数値以外のデータが入っていないかチェックする

ⅲ. **リミットチェック (限度検査)**
上限値，下限値を超えていないかチェックする

ⅳ. **レンジチェック (範囲検査)**
データが定められた値の範囲内にあるかチェックする

ⅴ. **シーケンスチェック (順番検査)**
データがレコードなどのキー項目順に並んでいるかチェックする

ⅵ. **チェクディジットチェック**
検査数字を 1 桁付加し、一定の計算を行うことによって、データ入力に間違いないかチェックする

ⅶ. **バランスチェック (平衡検査)**
対となっている項目の値が一致しているかチェックする

ⅷ. **アンマッチチェック (照合検査)**
入力されたコードなどの検査がファイルやテーブルに存在するかチェックする

3) **チェックディジットチェック (check digit check)**
商品コードなどのデータ入力ミスがないかチェックする方法

【例】コード

3	5	7	2	4	5	9
×	×	×	×	×	×	×
8	7	6	5	4	3	2

1. 各桁に重みを設定する (8,7,6,5,4,3,2)
2. コードの桁と重みの積を合計する
24+35+42+10+16+15+18 = 160
3. 合計値をある数値で割った余りの数値をコードの末尾につける
160 mod 11 = 6
→ 35724596

④計算処理のチェック方式
・計算処理を行っているときに、入力データのチェック漏れや計算で新たに生じたエラーを発見する設計

1) **プラス･マイナスチェック**
計算結果がプラスまたはマイナスになるように設定した場合、そのようになっているかチェックする

2) **オーバーフロー・異常値チェック**
計算結果が桁あふれや規定の値を超えていないかチェックする

3) **不能･不定チェック**
除算の場合、除数，被除数がゼロでないかチェックする

№4　プログラム設計　　★★★★☆

①プログラム設計
内部設計書からどのようなプログラムを作成するか機能や構造を明確にするもの

②プログラムの分割
構造化設計 (structured design)
大規模，複雑化するプログラム開発を効率的に行うために、プログラムを小機能単位に分割し、それぞれの単位ごとにプログラミングを行う
↓
モジュール (module)
手続きやデータの宣言からなるプログラムのコンパイル単位

③構造化設計の良い設計
1. 複雑さを減少する
適切な大きさに分割し、独立性を高め、適切な階層構造にする
2. 分割する
開発言語によるが、高水準言語で 1 モジュールあたり 300～500 ステートメント程度
3. 独立性を高める

モジュールが機能的にまとまっていて、モジュール間の関連が単純になるように分割する

4. 階層構造化する

1 つのモジュールから呼び出す従属モジュールの数を制限し、階層もあまり深くならないようにする

④モジュール分割技法

- **プロセス中心設計** … データの流れに着目し分割する
- **データ中心設計** … データの構造に着目し分割する

1) **プロセス中心設計**

ⅰ. **STS 分割 (Source Transform Sink 分割)**

プログラム
- 入力モジュール（Source：源泉）
- 処理モジュール（Transform：変換）
- 出力モジュール（Sink：吸収）

・最大抽象入力点

この場所以降はデータ入力が発生しない境界点
Source と Transform の境目

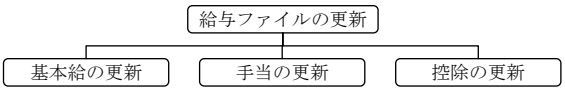
・最大抽象出力点

この場所以前は出力データが発生しない境界点
Transform と Sink の境目

ⅱ. **TR 分割 (TRansaction 分割)**

入力データの種類によってトランザクション処理が異なる場合に利用する分割法

例)



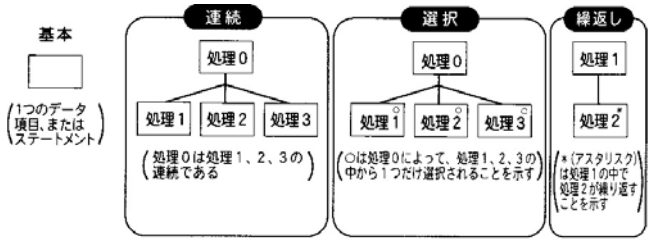
ⅲ. **共通機能分割**

モジュール全体で共通モジュールがあれば、それを独立したモジュールとして分割する技法

2) **データ中心設計**

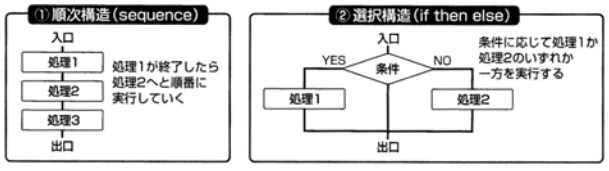
ⅰ. **ジャクソン法 (Jackson method)**

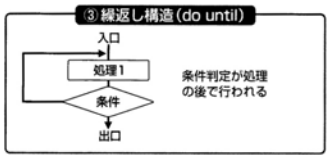
・入力データと出力データの構造を合わせて、プログラムを階層的に構造化していく手法
・階層化されたモジュールは、基本/連続/選択/繰返しの 4 つの基本図式を用いて表現する



ⅱ. **ワーニエ法 (Warnier-Orr method)**

・入力データを中心に構造化する手法
・データが「いつ・どこで・何回」使われるかを基本的に順次/選択/繰返しの 3 つのデータ構造で表現する





⑤モジュールの独立性

分割したモジュールは修正などを行った場合、他のモジュールへの影響を与えないように独立性を高めた設計にすることが重要

1) **モジュール強度**

モジュール同士の機能の関連性の強さを表す尺度

強度基準	説　明	関連性	独立性
機能的強度	1 つの機能を実現するために関連しあっている	強	高
情報的強度	同じデータ構造と関連しあっている	↑	↑
連絡的強度	共通のデータを参照し関連しあっている		
手順的強度	処理手順をモジュール化し関連しあっている		
時間的強度	処理する時期で関連しあっている		
論理的強度	関連ある複数の機能を扱うことで関連しあっている		
暗号的強度	関係のない複数の機能を 1 つにする	弱	低

2) **モジュール結合度**

・モジュール相互間でどの程度のデータのやり取りを行うかその関連性を表す尺度

・モジュール強度とは逆に、結合度が高いほど独立性は低い

結合度基準	説　明	関連性	独立性
データ結合	データのパラメータのみを受け渡す	弱	高
スタンプ結合	データ構造を決めるパラメータを渡す	↑	↑
制御結合	ほかのモジュールのパラメータを渡す		
外部結合	いくつかのモジュールが共通データを使用する		
共通結合	いくつかのモジュールが共通領域を使用する		
内容結合	ほかのモジュールの内容を直接参照する	強	低

№5　プログラミング　　★★★☆☆

①プログラミング手順

モジュール設計 … モジュールを構造化プログラミングする

↓

単体テスト計画 … テストデータ，テストの日程を作成する

↓

コーディング … プログラミング言語でプログラム記述する

↓

単体テスト … モジュールの単体テストを行う

②構造化プログラミング (structured programming)

プログラムの論理を順次/選択/繰返しの 3 つの基本構造に分けて記述

【プログラムの構成】

単純化され、見やすく分かりやすい。3 つの基本構造は、いずれも入口･出口の 1 つのブロック構造となる

【プログラムの流れ】

GOTO 文を用いないため、上から下へと一方向にアルゴリズムが流れるようになる

【プログラムの性質】

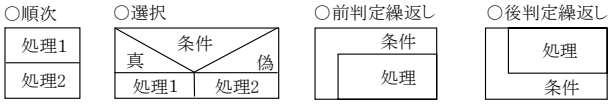
信頼性･保守性が高まる

③モジュール設計の図式表現法

1) **NS チャート (Nassi-Schneiderman chart)**

・構造化プログラミングの基本構造を図式表現したもので、階層構造を明確にする表現法
(構造化プログラミングで示された基本制御構造を表現するプログラム図式)

・特徴は矢線を使わない点と、1 つの入口と 1 つの出口を持つ点 ＝ GOTO 文表現を持たない



2) **HIPO (Hierarchy plus Input-Process-Output)**

・視覚的に理解しやすい図式表現と機能の階層的な記述ができる文書表現を用いた仕様書

・システム開発をトップダウンで行う場合によく利用される

階層構造 (Hierarchy)

図式目次 (VTOC：Virtual Table Process Output)

全体の機能構成を示し、プログラムやモジュール間の上下関係を階層的に表現

入力･処理･出力 (Input-Process-Output)

総括ダイアグラム (詳細ダイアグラム)

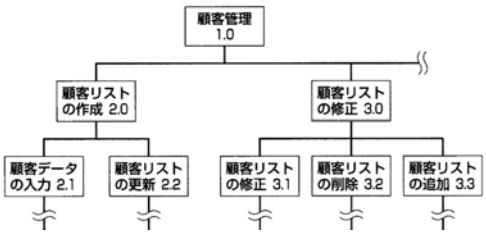
プログラムやモジュールへの入力･処理･出力を図で表現

○HIPO のメリット

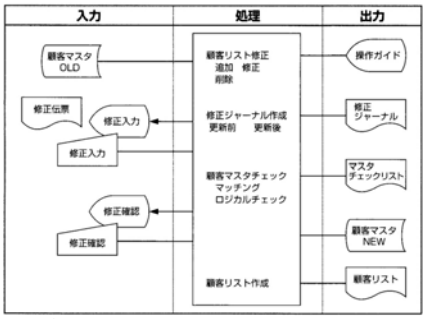
・体系化され、見やすく文章化されているので、変更や保守が容易

・機能と入出力データの流れが同時に表現されているため把握し易い

▼HIPO(図式目次)の例



▼HIPO(詳細ダイアグラム)の例



3) **決定表 (デシジョンテーブル：decision table)**

・条件とその条件に対する行動の関連性の真偽を「Y」,「N」で表して表にしたもの

・その条件の行動をとる場合は「X」を入力

条件を書く	データ閲覧申込	Y	Y	Y			
	データ印刷申込				Y	Y	Y
	データはあるか	Y	Y	N	Y	Y	N
	閲覧は許可できるか	Y	N		Y	N	
条件が満たされた時の行動を書く	ディスプレイへ表示	X			X		
	データ印刷				X		
	ｴﾗｰﾒｯｾｰｼﾞ表示		X	X		X	X

④デバッガ (debugger)

・**バグ**※を取り除く作業(**デバッグ**)を支援するソフトウェア

※バグ…プログラム上のエラー

・通常、ユーティリティプログラムとして提供

1) **ダンプ (dump)**

主記憶装置内のメモリやディスクなどの補助記憶装置内の状態を記録し、出力表示させる。プログラム開発時にプログラムの動作確認をするためのデータ追跡などで利用する。

2) **トレーサ (tracer)**

プログラム実行時の CPU や周辺機器への制御の流れの状態を記録し、出力表示させる。プログラムへの流れを命令単位で追跡する場合に利用される。

3) **スナップショット (snapshot)**

主記憶装置内の指定したアドレスやレジスタの内容を実行中に一定間隔で記録し、出力させる。プログラムのエラーの原因が分からないときに利用する。

4) **ブレークポイント (break point)**

プログラム開発などでプログラムの動作確認するために、ソースコードを適当な位置でプログラムを強制停止させ、停止直前のレジスタや変数の内容を確認する場合に利用される。

5) **インスペクタ**

デバック時にデータ構造の内容を確認するためのツール
変数やプロパティの内容を表示するツール

6) **クロスリファレンス**

オブジェクト間の関連情報をわかりやすく一覧表示するツール

⑤オープンソース

ソースコードを無償で公開し、誰でも自由にそのソフトを改良して再配布することを許可するもの。

№6　オブジェクト指向設計　　★★★★★

①オブジェクト指向 (object-oriented)

・システムが扱うオブジェクト(操作の対象物)を元にシステム構築するもの

・ユーザインタフェースの設計やシステム開発などに利用

・**カプセル化 (encapsulation)** …データの属性とそのデータに対する処理(メソッド：method)をひとまとめにしたもの

・C++, Java, small talk など

②クラスとインスタンス

1) **クラス (class)**

・共通する性質(機能や属性)を持つオブジェクトを定義したもの。

・同じ性質のオブジェクトをまとめて扱うことが出来る

2) **インスタンス (instance)**
・クラスの性質を持った 1 件のオブジェクトのこと
・「ひな形」(クラス)から作り出された実体のようなもの

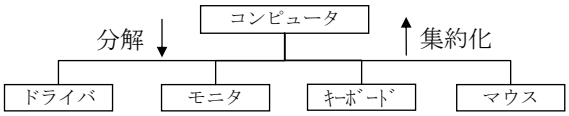
③継承 (インヘリタンス：inheritance)
クラスはデータの属性やメソッドの類似性によって階層化される。このときの上位クラスを**スーパークラス**、下位のクラスを**サブクラス**という
→ 階層化されたクラス間で、サブクラスがスーパークラスの性質を受け継ぐこと

④継承関係 (is-a 関係)
・特化 … スーパークラスをサブクラスに分解すること
・汎化 … サブクラスからスーパークラスを生成すること

例) コンピュータの性質を継承してより具体化すると、デスクトップパソコン，ノートパソコン，携帯端末などがある。

⑤集約関係 (has-a 関係)
複合オブジェクト
…複数のオブジェクトを組み合わせて 1 つのオブジェクトを作り上げたもの
・分解 … 上位クラスを下位クラスに分けること
・集約化 … 下位クラスから上位クラスを構成すること

例) パソコンは、ドライバ，モニタ，キーボード，マウスから構成されるので、「パソコン」は「ドライバ，モニタ，キーボード，マウス」の集約化となる



⑥多様性 (ポリモーフィズム：polymorphism)
同じメッセージを送ってもオブジェクトが異なると、オブジェクトはそれぞれ固有の異なった処理を行う

№7　ソフトウェアの品質管理　　★★★★☆

①ソフトウェアの品質特性
ISO ではソフトウェアの品質特性の尺度として次の 6 つを定める

1) **機能性 (functionality)**
・仕様書どおりに実際に稼動するかを示すもの
・機能性の品質は上流過程のシステム分析における要求仕様によって左右される

2) **信頼性 (reliability)**
・ソフトウェアが仕様書どおりに故障することなく正確な動作をするか、また故障時にはすぐに復旧できるバックアップ体制がとられているかを示すもの
・信頼性の品質評価は、テスト工程が大きな要因となる

3) **使用性 (usability)**
・ソフトウェアを操作するにあたり、ユーザにとっていかに使いやすいものとなっているかを示すもの
・ユーザインタフェースをいかに充実させるかが重要

4) **効率性 (efficiency)**
・プログラムを稼動したとき、コンピュータシステムの資源をいかに効率よく使用しているかを示したもの
時間経過性の評価
ターンアラウンドタイムやレスポンスタイムの短縮など
資源経済性の評価
主記憶装置や補助記憶装置の有効利用など

5) **保守性 (maintainability)**
・ソフトウェアに修正を施す場合、いかに修正しやすいかを示すもの
・保守しやすい構造化設計によるプログラム作成が必要

6) **移植性 (portability)**
・システムの入れ換えや動作環境の変化に対して、いかに容易にソフトウェアの移植ができるかを示したもの
・なるべく JIS 規格による標準仕様に沿ってプログラミングすることが望まれる

②**レビュー (review)**

人間がプログラミングリストや資料を元に討議し、あいまいな点や問題点を発見する方法

1) **ウォークスルー (walk-through)**

システム仕様書やプログラム設計書などにミスはないか、担当者や複数の人間を前に説明を行い、検討することによって、見落としを防ごうとする方法
ソフトウェア開発段階で、品質向上を図るために、複数のメンバーで開発工程の成果物を検証すること

2) **インスペクション (inspection)**

完成したプログラムを実行する前に、モデレータ(moderator：管理者)を中心とした管理統括の元で、大きな誤りがないか、プログラム仕様とコーディング内容をもう一度比較する方法

3) **成長曲線 (S 字カーブ)**

・ソフトウェアのエラー発生件数と期間は成長曲線(バグ曲線)に近似する
・最初は少ないが、ある時期から急激に増え、最終的に収束していく傾向がある

№8　プログラムテスト　　★★★★★

①プログラムテストの種類

・高品質なプログラムを開発するためには、十分なテストが必要であり、プログラムテストは、プログラム開発工程の半分以上を有するほど重要なウェイトである
・プログラムテストは、

単体テスト → 結合テスト → システムテスト → 運用テスト

の順に行われる

②**単体テスト (unit test)**

1 つのモジュールが仕様書通りに正しく動作するか事前に行うテスト

1) **ブラックボックステスト (black box test)**

・プログラムの外部仕様(ユーザインタフェース)に着目する。プログラムをブラックボックス化し、入力と出力との関係が仕様書どおりになっているかチェックするテスト
・モジュール間のインタフェースのテストをする上で、結合テストで使用することもある

テスト	特　徴
同値分割	テストデータをある 1 つの条件に対し、2 つ以上の同値クラスに分け、それぞれのクラスの中から代表するデータを 1 つ選択するテスト
限界値分析	それぞれのクラスの中から境界条件のデータを選択するテスト
因果グラフ	複合的な条件設定によって真偽判定が異なる場合、入力(原因)と出力(結果)の関係を示すテストデータを選択する方法
実験計画法	テストケースが非常に多く、すべてをテストできない場合、テストデータが最小限になるように選択するテスト

2) **ホワイトボックステスト (white box test)**

プログラムの内部仕様に基づいて、構造や制御，データの流れに着目して行うテスト

テスト	特　徴
命令網羅	プログラムの中の全ての命令を最低 1 回は行うようにするテスト
判定条件網羅	全ての判定条件における真偽を 1 回は行うようにするテスト
条件網羅	判定条件における真偽の全ての組み合わせを満たすように行うテスト(A と B の 2 条件ある場合、A と B それぞれの真偽を行う)
複数条件網羅	複数ある判定条件の真偽の全ての組み合わせを満たすように行うテスト（A と B の 2 条件ある場合、A と B における全ての組み合わせの真偽を行う)

③**結合テスト(joint test)**

・2 本以上のモジュールを連結して正しく動作するかをテストする
・主にモジュール間のインタフェースやプログラムの入出力機能のチェックを行う

1) **トップダウンテスト (top down testing)**

・上位モジュールから下位モジュールへと順次結合してテストする方法
・下位モジュールが存在しない状態でテストを行うとき、下位モジュールの代わりに**スタブ(stub)**※が必要

※スタブ (stub)

…トップダウンテストを行う際、完成した上位モジュールをテストするため、未完成の下位モジュールのインタフェース部分だけを用意したプログラム。テスト対象のモジュールからの呼出し命令の条件に合わせて、値を返す

2) **ボトムアップテスト (bottom up testing)**
・下位モジュールから上位モジュールへと順次結合してテストする方法
・上位モジュールが未完成の段階で下位モジュールのテストを行うときは、上位モジュールの代わりとなる**ドライバ(driver)**※が必要
※ドライバ (driver)
…ボトムアップテストを行う際、完成した下位モジュールをテストするため、未完成の上位プログラムの動作をシミュレートするプログラム

3) **サンドウィッチテスト (sandwich testing)**
・ボトムアップテストとトップダウンテストを同時に行うテスト方法
・両者が出会った時点でテストを終了する
・大規模プログラムのテストに適している

4) **ビッグバンテスト (big bung testing)**
プログラムの各モジュールを個々にテストし、すべてのテストが終わった段階で一度に全モジュールを結合し、全体としてテストする方法

④システムテスト (system test)
・全てのモジュールを連結して正しく動作するかテストする。システム全体として要求されている機能を十分に発揮できているかチェックする
・チェック項目…機能，性能，負荷，操作性，障害など

⑤運用テスト (acceptance test)
実際にシステムを運用させながら行う最終テスト。
実際のデータやシミュレーションデータを使用して、ユーザの要求どおりにシステムが稼動するか、使用しながら確認する。

※テストデータジェネレータ
…パラメータを与え、テストデータを自動的に生成するプログラム

⑥保守作業 (maintenance)
1) **修正作業**
バグを修正する作業

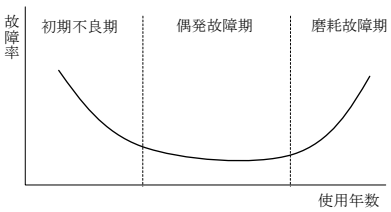
2) **変更作業**
年号や金利の変更，法律改正などの外部要因の変化に適応して行う作業

3) **改良作業**
ハードウェア，OS，周辺機器，データ仕様等、修正が必要なところを改良する作業。最も時間とコストを要する。

⑦保守作業の注意点
1) **リップル効果**
ソフトウェアの一部修正･改良は、時としてソフトウェア全体に悪影響を及ぼすことがある

2) **レグレッションテスト (regression test)**
・保守作業の結果を検証するテスト。新たなバグが生じていないか確認
・別名：**退行テスト**

3) **バスタブ曲線**
ハードウェアの故障頻度と経過時間の関係をグラフで表したもの



4) **信頼度成長曲線**
・テスト工程で、テストが順調に進んだ場合、発見されるバグ件数はテスト項目消化件数に比例して増加するが、ある時点から急激に減少する
・縦軸に累積バグ件数を、横軸にテスト項目消化件数を取ってグラフ化したもの

№9　システムの開発管理　　★★★★★

①日程管理
1) **PERT 図 (Program Evaluation Review Technique chart)**
・多くの作業からなるプロジェクトの日程管理を行う手法
・プロジェクトの作業手順を**アローダイアグラム(arrow diagram：矢線図)**で表し、重点管理を**クリティカルパス(critical-path)**で表す。

ⅰ.**最早開始時刻**

先行作業がすべて完了し、最も早く後続作業を開始できる時刻のこと。複数の作業経路が結合するところでは最大値をとる。

※先行作業と後続作業

A→B→C と順に作業が行われた場合、B 点から見て A は先行作業、C は後続作業となる

ⅱ.**最遅開始時刻**

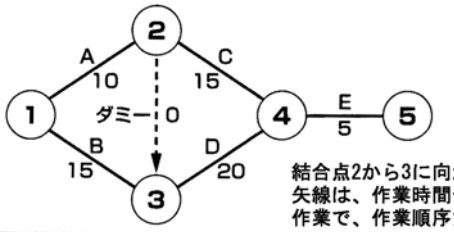
後続作業すべてに遅れが生じないように、先行作業を完了しておかなければならない最も遅い時刻のこと。複数の作業に分岐するところでは最小値をとる

ⅲ.**クリティカルパス (critical-path)**

・作業開始から終了までの間で、作業日程に余裕のない最短経路のこと

・求め方は、各作業の余裕時間を求め、余裕時間がゼロの作業を結ぶ

※**余裕時間**…最遅開始時刻と最早開始時刻の差



作業	最早開始時刻	最遅開始時刻	余裕時間
①	0	0 MIN (15−10, 15−15)	0
②	10	15 MIN (35−20−0, 35−15)	5
③	15 MAX (10+0, 15)	15 (35−20)	0
④	35 MAX (10+15, 15+20)	35 (40−5)	0
⑤	40	40	0

求める順番

②進捗管理

実作業が計画通りに進んでいるか **QCD(Quality:品質，Cost:コスト，Delivery:納期)**の観点から管理し把握するもの

1) **ガントチャート (Gantt chart)**

・各作業の日程をバー(横線)で表した作業の進捗管理図

・予定と実績を対比することで作業情況を把握しやすい

・別名：**バートチャート**

2) **WBS (Work Break down Structure)**

・プロジェクト開発に必要な作業項目をトップダウン的に洗い出し、それらを階層構造で表した作業分割図

③開発工数の見積もり

1) **ファンクションポイント法 (function point method)**

システム開発で生じる**外部入力**，**外部出力**，**内部論理ファイル**，**外部インタフェースファイル**，**外部照会**の 5 機能に分割し、それぞれの機能の難易度(単純，平均，複雑)別に応じた重みを掛け合わせてシステムの機能を定量化(ポイント)し、それに補正係数を掛け合わせて求める

【例】次の表のようなユーザファンクションタイプごとの測定個数および重み付け係数があり、この複雑さの補正係数が 0.7 である場合、

ﾕｰｻﾞﾌｧﾝｸｼｮﾝﾀｲﾌﾟ	測定個数	重み付け係数
外部入力	3	3
外部出力	5	4
内部論理ファイル	4	6
外部ｲﾝﾀﾌｪｰｽﾌｧｲﾙ	3	3
外部照会	2	4

ファンクションポイント = (3×3+5×4+4×6+3×3+2×4)×0.7 = 49

2) **COCOMO (COnstructive COst MOdel)**

開発するプログラムの規模(ソフトウェアの行数)を把握し、その行数を開発工数に換算する手法

3) 開発規模と工数の関係

開発規模がある規模を超えると、開発工数は指数関数的に急速に増える傾向がある

第 9 章　情報システムと社会

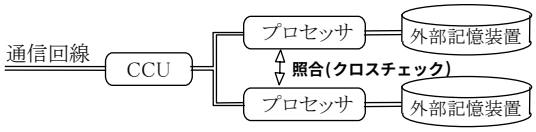
№1　コンピュータのシステム構成　　★★★★★

①シンプレックスシステム (simplex system)

・1 台のプロセッサと外部記憶装置で構成された単一システム
・経済的だが、1 つのシステムが故障するとシステム全体に影響する為、信頼性が低く、リアルタイム処理にはあまり向かない

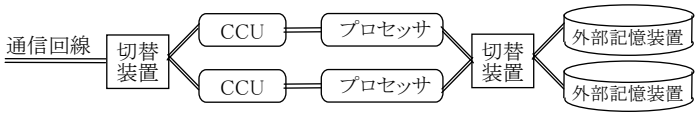
②デュアルシステム (dual system)

・構成機器を二重化して通常は 2 つのシステムで処理し、結果を**照合(クロスチェック)**しながら処理を進めていく
・障害が発生した場合は、故障したシステムを切り離し、残りのシステムのみで処理を続ける
・システムダウンが許されない信頼性の高いシステムに利用される



③デュプレックスシステム (duplex system)

・主系と従系の 2 系統を持つシステム構成。どちらか一方が主系として動作し、他方が従系として待機状態となる



1) **ホットスタンバイシステム**
　従系システムに主系システムと同じオンライン処理プログラムを起動した状態で待機させておき、主系に障害が生じた場合、速やかに従系に切り替えて処理を行う

2) **コールドスタンバイシステム**
　通常は従系システムは主系システムとは異なる処理を行っていて、主系に障害が生じた場合、速やかに従系に切り替えて主系処理を行う

④マルチプロセッサシステム (multiprocessor system)

・複数のプロセッサを並列に動作させることで処理能力，拡張性を向上させるシステム
・いずれのプロセッサが故障しても、そのプロセッサを切り離して処理を続行させるので、信頼性も高い

1) **密結合マルチプロセッサシステム**
　複数のプロセッサが 1 つの主記憶装置を共有し、1 つの OS によってシステム全体が管理されるシステム

2) **疎結合マルチプロセッサシステム**
　共有記憶がなく、プロセッサごとに主記憶装置と OS を持つシステム

⑤クラスタリング (clustering)

・複数の独立したコンピュータを相互接続し、全体として 1 台のコンピュータのように動作するシステム
・システムの一部に障害が生じてもシステム全体はダウンしない高い信頼性のシステムで、大規模なサーバシステムに利用される

⑥タンデムシステム (tandem system)

・1 台のプロセッサとそれより小型のプロセッサで構成されたもので、処理速度を高めるために直列にしたシステム
・それぞれのプロセッサが機能分担をして、処理効率の向上を図る

⑦タイムシェアリングシステム (時分割処理：TSS)

・CPU の処理時間を短い時間に分割し、複数の処理を単位時間ごとに切り替えて多重プログラミングを行う方式
・コンピュータの演算速度の高速性を利用して複数の端末の処理を可能にする
・対話型処理はユーザの判断の関与する部分が大きい
・対話型処理は適用業務に精通したエンドユーザが使用する
・コンピュータとユーザ間の意思疎通を円滑にするには、ユーザインタフェースの役割りが重要
・対話システムでは、命令を出してから結果が出るまでの時間である応答時間が短いことが重要

※ラウンドロビン方式
　…タイムシェアリングシステムで、各タスクに処理時間を均等に割り当て、順番に処理を進める管理方式

⑧**集中処理**と**分散処理**

	集中処理システム	分散処理システム
管理	集中して管理を行うことができるため容易である	分散しているため管理は難しい
セキュリティ	1ヶ所で管理するため、セキュリティの確保は容易	分散しているためセキュリティの確保は難しい
信頼性	ホストがダウンするとシステム全体が停止してしまうため、信頼性は低い	部分的に障害は発生しても、システム全体に与える影響は小さく信頼性は高い
柔軟性	処理量が増加すると、ホストを入れ替えなければならないため、柔軟性は低い	処理量が増加しても他のコンピュータで代替できるため、柔軟性は高い
システム開発	規模が大きくなると、少しの変更でも難しくなる	部分的なシステムの変更は容易
データ整合性	ホスト1ヶ所にデータを持つため、データの整合性をとるのは容易	データも分散しているため、整合性をとるのは難しい
EUC	対応するのは難しい	比較的自由に利用できる

情報処理システムの分類

- **バッチ処理システム**
  手順の定まった仕事を一括して処理するシステム
- **オンライントランザクション処理システム**
  端末からのデータをホスト側で処理し、すぐに結果を送り返す
- **リアルタイム制御処理**
  装置の状態を示すデータを収集し、状態に応じて即座に指示を出す
- **対話型処理システム**
  人間とコンピュータが対話しているかのように処理を進める
- **集中処理システム**
  ホストコンピュータで、集中的にデータ処理を行うシステム
- **分散処理システム**
  分散したコンピュータが、それぞれの処理を行うシステム

分散処理システムの形態
- **垂直分散**
  ホストコンピュータを頂点とした階層構造を持つ分散処理システム
- **水平分散**
  上下関係を持たない分散処理システム

№2　システムの信頼性　　★★★★★

①**RASIS**

コンピュータシステムの総合的な信頼性に対する概念

R:信頼性 (Reliability)	システムが与えられた条件の元でどれだけ故障せずに機能するかをいう。平均故障間隔(MTBF)で表す
A：可用性 (Availability)	システムが処理要求に対し機能を維持している確率、時間の割合をいう。稼働率で表す
S：保守性 (Serviceability)	システムが故障したとき最小限の時間で稼動できるように予防，修復することをいう。平均修理時間(MTTR)で表す
I：保全性 (Integrity)	システムが故障状態からどの程度修復できるかをいう
S：機密性 (Security)	不正利用などからどの程度システムを保護できるかをいう

②信頼性の尺度

1) **MTBF (平均故障間隔：Mean Time Between Failures)**
- RASIS における信頼性(Reliability)を表す尺度
- コンピュータがどれだけ長時間、安定しているかを評価する

$$\text{MTBF} = \frac{\text{稼働時間}}{\text{故障件数}}$$

2) **MTTR (平均修理時間：Mean Time To Repair)**
- RASIS における保守性(Serviceability)を表す尺度
- コンピュータシステムが故障したとき、どれだけ短時間に再生できるかを評価する

$$\text{MTTR} = \frac{\text{不稼動時間}}{\text{故障件数}}$$

3) **稼働率**

・RASIS における可用性(Availability)を表す尺度

・コンピュータシステムがどれだけ稼動したかを評価する

$$稼働率 = \frac{MTBF}{MTBF + MTTR}$$

ⅰ.直列システムの稼働率

各装置が直列に稼動しているシステムは、途中のどの装置が故障してもシステム全体の故障につながる

全体の稼働率 = $P_1 \times P_2 \times \cdots \times P_n$

ⅱ.並列システムの稼働率

各装置が並列に稼動しているシステムは、一般に 1 台の装置が故障してもシステム全体は稼動可能

全体の稼働率 = $1 - (1-P_1) \times (1-P_2) \times \cdots \times (1-P_n)$

③フォールトトレランス (fault tolerance)

コンピュータシステムの故障や誤動作などの人的な失敗が生じた場合でも、停止することなく常に稼動状態を保つ技術

1) システムの障害対策

ⅰ.**フェイルセーフ (fail safe)**

システムの障害が生じた時、安全な状態でシステムを停止させる技術

ⅱ.**フェイルソフト (fail soft)**

システムの障害が生じた場合、システム全体が停止しないよう影響を受けた部分を切り離すことで、処理を続行しようとする技術

縮退運転 …機能は低下するが被害は最小で稼動

ⅲ.**フールプルーフ (fool proof)**

ユーザが意図しない使い方をしたり、誤った操作をしても、システムが停止したり異常終了したりしないように設計段階で安全対策を施したもの

2) ハードディスクの障害対策

ⅰ.**ミラーリング (mirroring)**

故障に備えて 2 台以上のハードディスクを用意し、全てのディスク装置に同一データを書き込むことで信頼性を高める技術

ⅱ.**RAID (Redundant Array of Independent Disks)**

複数のハードディスクを並列に結合し 1 台のハードディスクとして使用することで、高速で大容量かつ信頼性を高める技術

レベル	内　容
RAID 0	データを複数のディスクに均等に振り分け、同時並列で記録することで高速化を図る方式（ストライピング）
RAID 1	2 台のディスクに同一データを同時に記録することで、信頼性の向上を図る方式（ミラーリング，デュプレックス）
RAID 2	ビット単位に分割されたデータとハミングコードという誤り訂正符号のパリティがディスクに分散させて記録する方式
RAID 3	RAID2 と同様だが、パリティは専用のディスクに記録する方式
RAID 4	RAID3 と同様だが、分割したデータをブロック単位で記録する方式
RAID 5	ブロック単位に分割されたデータとパリティが複数のディスクに分散して記録される方式

3) その他の障害対策

○**無停電電源装置 (UPS：Uninterruptible Power Supply)**

電池や発電機を内蔵し、停電してもしばらくの間システムを稼動できるようにする装置

○**CVCF** (Constant-Voltage Constant-Frequency)

電圧･周波数を安定化した電源

№3　セキュリティ　　★★★★★

①情報システムのセキュリティ

1) プライバシーの保護

情報通信システム上での個人情報の漏洩や不正使用を未然に防ぐこと

アクセス管理 (access control)	コンピュータシステムへの不正アクセスを防止するためにユーザに与えるアクセス権を明確にし、管理すること
認証 (authentication)	相手が本人であるかを確認すること。主にパスワードやコールバック，ディジタル署名などが利用される
暗号化 (encryption)	ネットワーク上で交わされる個人情報などのデータを第三者には理解できないように暗号に変換すること
復号化 (decode)	暗号化されたデータをもとのデータに戻すこと

2) **暗号化 (encryption)**

ⅰ.**秘密鍵暗号方式 (secret-key cryptosystem)**

・別名：**共通暗号化方式**
・暗号化と復号に同じ秘密鍵を使い、当事者(送信者と受信者)以外には秘密にしておく方式
・**DES**(Data Encryption Standard)など
・暗号化，復号の速度は速く、グループの利用に向くが、相手の数だけ鍵を作成する必要がある

ⅱ.**公開鍵暗号方式 (public-key cryptosystem)**

・暗号化鍵と復号化鍵を別にし、データを受信者の公開鍵で暗号化し、受信者は送られてきた暗号データを受信者の秘密鍵で複合化する
・**RSA**(Rivest-Shamir-Adleman scheme)など
・暗号化/復号化の速度がやや遅い。公開した鍵を用いるため、誰でも暗号化できるので**なりすまし**の危険性がある

3) **ディジタル署名 (digital signature)**

・送信者の秘密鍵で署名を暗号化して送り、送信者の公開鍵を使って復号化することで、本人から送られたものと確認する仕組み
・なりすましの防止として利用される

▼公開鍵暗号方式とディジタル署名の違い

	公開鍵暗号方式	ディジタル署名
暗号化	受信者の公開鍵	送信者の秘密鍵
復号化	受信者の秘密鍵	送信者の公開鍵

4) **SSL (Secure Sockets Layer)**

・インターネット上で使われている WWW や FTP などのデータを暗号化して送受信するプロトコル
・データの盗聴や改竄，なりすましを防ぐ

②**不正アクセス禁止法 (正式名：不正アクセス行為の禁止等に関する法律)**

・1999 年制定 → 2000 年施行
・ネットワークを介した不正アクセスを防止する法律
・不正アクセス行為の定義
　アクセス権限を持たないものが正規権限者のアクセス権限を不正に利用し、アクセス制限機能を持つコンピュータにネットワーク経由で不正アクセスすること
・**不正アクセス**
　├他人のユーザ ID やパスワードを無断で利用する行為
　└セキュリティホールの攻撃などによってアクセス制限機能を無効にする行為

③コンピュータウィルスとその駆除

1) **コンピュータウィルス (computer virus)**

ネットワークやディスクを介して、コンピュータシステム内に密かに不正侵入し、ディスク内のデータや SO をはじめとするプログラムの内容を改竄したり、破壊したりするプログラムのこと

▼コンピュータウィルスの機能

自己伝染機能	ほかのコンピュータに進入し、勝手にコピー繁殖すること
潜伏機能	一定の条件(時刻など)が整うまで症状を出さないこと
発病機能	ウィルスプログラムが活動すること

2) コンピュータウィルスの分類

ⅰ.**ウィルス**

・コンピュータ内のファイルに取り付いて悪さをするプログラム
・1 つのファイルに感染すると次々とパソコン内のファイルに取り付いて繁殖していく
・最近は特定のアプリケーションソフトのマクロ機能を利用して感染する**マクロウィルス**が増加

ⅱ.**ワーム**

・ネットワークを使って自分のコピーを他のコンピュータに送り込むプログラム
・ウィルスとは違い、ユーザの手を借りることなく自分で繁殖していくため瞬時に他のコンピュータに感染する

ⅲ.**トロイの木馬**

・ウィルスやワームのような繁殖能力はないが、あたかもユーザの興味を引く便利なソフトを装い、ダウンロードさせることでパソコンに取り付き感染していく

3) コンピュータウィルス対策

・**ワクチンソフト**

コンピュータウィルスを発見して駆除するソフトウェア

【ワクチンソフトの有効利用】

・パソコンに常駐させる
・定義ファイルは定期的に更新する
・使用期限が過ぎたら更新する

④**セキュリティポリシー (security policy)**

・企業全体のセキュリティに関する基本方針
・セキュリティポリシーを明示することで、責任の所在や判断基準や実施すべき対策が明確になる

⑤**情報セキュリティマネジメントシステム (ISMS)**

情報セキュリティを維持･向上させるための管理の仕組みについての規格

・**ISO/IEC17799 (情報セキュリティマネジメント実施基準)**

英国の規格の BS7799 Part1 を国際規格化
日本 → JIS X 5080：2002

【ISMS の目的】

JIS X5080：2002 と ISMS の仕様を定めた BS7799 Part2 によって、組織の ISMS が情報セキュリティを維持･向上させる上で十分な要件を備えて構築され適切に運用されているかを評価すること

⑥**アクセスコントロール**

不正アクセスを防止するためのセキュリティ対策

・アクセス権限は担当者の職務内容に合わせ、必要最小限の範囲で付与する
・ユーザ ID によるアクセス管理はアクセス権限と一致させ、きめ細かく設定する
・管理者は、アクセス権限とユーザ ID の設定状況をタイムリーに確認する
・担当者はパスワード管理を厳重に行う。定期的にパスワードを変更する
・アクセスログを記録し、管理者が定期的にアクセス状況を確認する

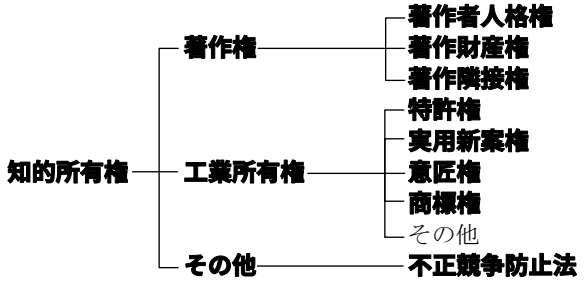
⑦**システム監査基準**

監査対象から独立かつ客観的立場のシステム監査人が情報システムを総合的に点検および評価し、組織体の長に助言および勧告するとともにフォローアップする一連の活動
システム監査は、システム化計画，開発，運用の全工程を対象とするもので、監査に耐えられるようになっていなければならない

№4　知的所有権　　★★★★☆

①**知的所有権**

・知的な創作活動の成果として生まれた発明や著作物などに対する権利



②**著作権**

執筆作品，エンターテイメント作品，コンピュータプログラム，データベースなどに対して、著作権の権利を認め保護するもの

▼著作権法の分類

著作者人格権	未発表の著作物を発表する公表権，著作者の氏名を表示する氏名表示権，著作物をむやみに改変させない同一性保持権がある
著作財産権	複製権，上演権，演奏権，公衆送信権，口述権，展示権，頒布権，二次的著作物の利用権などがある
著作隣接権	楽曲の演奏家など著作物を公衆に伝達する者に対する権利

③**工業所有権**

・工業技術，デザイン，識別標識などにかかわる知的財産を保護するための権利

・取得方法は先願主義

権利	保護対象	保護期間
特許権	発明に対する権利	出願日から 20 年
実用新案権	小さなアイディアや発明に対する権利	出願日から 6 年
意匠権	物品の色，形状，デザインに対する権利	登録日から 15 年
商標権	トレードマークやサービスマークに対する権利	登録日から 10 年

④ソフトウェアの著作権保護

○ソフトウェアの著作権対象

プログラム開発時の設計書，操作マニュアル，ソースプログラム，オブジェクトプログラム，データベース

↓

言語，規約，アルゴリズムは保護対象外

1) **開発したソフトウェアの著作権の帰属**

企業が社員に業務として開発させたソフトウェアの著作権は、特別な取り決めがない限り、開発者個人ではなく企業に帰属する

委託契約によって外部企業に開発させたソフトウェアは、契約上の取り決めが無ければ、開発を行った委託先企業に帰属する

2) **同一性保持権の制限**

著作権法では著作権者の了解なしに著作物に変更を加えてはならないという同一性保持権を認めているが、プログラムを自社のシステム環境で稼動させるため、処理性能を向上させるためなどの理由で変更することは、同一性保持権の制限として、著作権者の了解を得る必要はない

3) **複製権の制限**

バックアップのためにプログラムを複製し保管することは著作権の違反にはならない。

4) **プログラムの登録制度**

プログラムは公開されていなくても、作成日を持って著作物となる。

5) **違法複製プログラムの使用**

違法に複製されたプログラムの使用に関しては、複製プログラムの使用権限を取得した時点で違法複製プログラムであることを知っていた場合に限り、著作権違反となる。

⑤フリーウェアとシェアウェア

1) **フリーウェア (freeware)**

作者が著作権を留保したまま、広く自由に利用してもらうことを目的としたもので、無料で使用できる

2) **シェアウェア (shareware)**

作者が著作権を留保したまま、一定の試用期間を経て、気に入ればユーザ登録料を支払う方法のソフトウェア

3) **パブリックドメインソフト (PDS：Public Domain Software)**

著作権を放棄したソフトウェア。日本の法制度では著作権の放棄は認められていないため、厳密には PDS は日本に存在しない。

⑥著作権フリー

著作権保護の対象にならないもの。著作権が行使されないもの。

フリーソフト用のライセンスの代表的なものに、FSF(Free Software Foundation)が提唱する GNU GPL(General Public License)がある。GPL に準拠したフリーソフトは、誰がどのように改変してもフリーソフトであることが保証される。

⑦ビジネスモデル特許

IT の発展に伴い、IT を利用したビジネスの仕組みについても、特許要件を満たしていれば、特許が認められる

№5　企業の情報化と経営　　★★★★★

①CIO (Chief Information Officer)
企業の情報セキュリティや情報戦略にいたるまでの情報システムを統括する最高責任者

CEO (Chief Executive Officer)
経営最高責任者

COO (Chief Operating Officer)
最高執行責任者
CEO の決定した戦略に従って、実際の業務の執行及び業務の戦略的意思決定者

②ERP (Enterprise Resource Planning：企業資源計画)
経理，生産管理，販売管理，人事管理などの基幹業務の情報をコンピュータで一元管理し、企業活動の効率化を図るための手法

③アウトソーシング (outsourcing)
・経営資源を全て自前で持つのではなく、必要に応じて外部資源を活用すること
・アウトソーシングを行うことにより、企業は自社の競争力となる強みに資源を集中させ、コスト削減，生産性向上，体質強化などを図ることができる

④TCO (Total Cost of Ownership)
コンピュータシステムの導入から維持，管理にいたるまでに要する総費用

⑤SCM (Supply Chain Management)
・資源調達から生産，物流，販売，在庫管理，製品の発送までコンピュータを使って総合管理する企業活動の管理手法の１つ
・ジャスト･イン･タイムな管理を実現することで、余分な部品在庫，製品在庫を削減し、コストの引き下げを目指す

⑥OJT (On the Job Training)
管理者や上司が部下に対して実際の仕事を遂行しながら教育訓練を行うこと

⑦デルファイ法
・多数の専門家に同一内容のアンケート調査を繰り返し、回答者の意見をまとめる方法
・アンケート調査とは違い、前回の回答結果を回答者にフィードバックし、回答者が全体の意見を参考にして回答内容を修正することができる
・技術予測に用いられる手法であり、広範囲にわたる諸技術事象の変化を、相当期間に渡って推測し、将来起こりうる事象に関する予測をする

⑧ワークサンプリング法 (work sampling method)
・観測者がランダムに定めた時間に瞬時観測を行い、事務や生産における作業情況や作業量などを、サンプリングして作業時間を見積もる方法
・統計結果から問題点を解明し、効率よく作業が行えることを目的とする。

⑨企業との雇用契約

1) **派遣**
労働者は派遣元の会社ｔの雇用関係を継続して保つ。ただし、派遣先の会社の指揮命令を受けて就業する。雇用条件は派遣先と結ぶ。派遣元と派遣先は労働者派遣契約を結ぶ。

2) **請負**
労働者は請負業者と雇用関係を結び、請負業者の指揮命令を受けて就業する。雇用関係も作業上の指揮命令権も請負業者にあり、発注主には指揮命令権はない。注文主と請負業者は請負契約を結ぶ。

3) **出向**
雇用関係と作業場の指揮命令は出向元と出向先の両者に二重に発生する。出向元と出向先は出向契約を結ぶ。
・移籍出向
出向元の会社との雇用関係を終わらせ、出向先の会社と雇用関係を新たに結ぶ
・在籍出向
出向元の契約を保ちながら出向先の会社とも雇用関係を結ぶ

⑩品質マネージメント規格 (ISO9001)
・ISO9000 シリーズは、ISO(国際標準化機構)によって、国や企業による品質管理及び品質保証の違いによる商品の流通の支障をなくすために制定された国際標準規格
・ISO9000 シリーズの中の ISO9001 が品質マネージメントを規定している

【ISO9001：2000 の認証】

国内における ISO9001：2000 の認証審査は、日本の審査登録認定機関である(財)日本適合性認定協会(JAB)の認定を受けた認定登録機関によって行われる

⑪製造物責任法 (PL 法：Product Liability law)

消費者が日常使う製品に何らかの欠陥があって、怪我や事故などの損害をこうむった場合、その製造会社などに損害賠償を求められる法律制度

№6　インターネットの応用利用　　★★★☆☆

①インターネット (internet)

・インターネット技術を利用して企業などの内部ネットワーク(LAN)を構築し、情報の共有化や外部への情報提供などを行うシステム

・外部に公開できない重要情報はファイアウォールを使って情報を防御する必要がある

○**エクストラネット (extranet)** … イントラネット同士を接続した広範囲のネットワーク

②ファイアウォール (fire wall)

外部からの不正アクセスなどから内部ネットワークを守るセキュリティ対策機能のこと

○**プロキシサーバ (proxy server)**

内部ネットワークとインターネットの間に位置し、内部ネットワークへの不正アクセスなどのセキュリティ確保や高速アクセスを実現させるためのサーバ

○**フィルタリング (filtering)**

内部を管理するホストコンピュータ宛のデータはほかに転送せず、外部に宛てられたデータだけほかのホストコンピュータに転送する機能

③グループウェア (groupware)

インターネットや LAN などのネットワークを利用して、情報共有を効率的に行い、グループ作業を円滑に進めるためのソフトウェア

▼グループウェアの代表的な機能

電子メール機能	グループメンバや外部関連メンバとのコミュニケーションを円滑に図る
電子掲示板機能	グループメンバに広報を行う
ｽｹｼﾞｭｰﾙ管理機能	メンバでスケジュールを共有する
文書管理機能	アイディアやノウハウをデータベース化して共有する
ワークフロー機能	稟議書など企業内の届け処理や事務フローを電子化し、流通させる
電子会議機能	離れた場所にいるメンバとリアルタイムで打ち合わせをする

№7　ビジネスシステム　　★★★☆☆

①POS システム (Point Of Sales system)

・小売業において、どの商品がいつ何個売れたか把握し、商品全体の販売管理をするシステム

・別名：**販売時点情報管理システム**

②EOS (Electronic Ordering System)

・オンラインによる自動受発注システムのこと

・POS システムと連動させることにより在庫状況などを的確に管理できる

③EC (Electronic Commerce：電子商取引)

・ネットワーク環境を有効利用して、電子的に取引を行うシステムのこと

・通常、インターネットを利用し、オンラインショップなどを行っている

1) **B to C (Business to Consumer)**

・企業がインターネット上で開設したオンラインショップを通じて、商品やサービスを顧客に販売する消費者向け電子商取引のこと

・利点は、実店舗よりローコストで販路を拡大することができ、そのコスト削減による低価格化や迅速な納品が行える

2) **B to B (Business to Business)**

・インターネットを通じて企業同士が商談などを進める企業間電子商取引のこと

・利点は、営業や購買などに関わるコストを削減でき、顧客が注文した瞬間にその取引先だけでなく、仕入先への発注などをリアルタイムで行うことができる

④**EDI (Electronic Data Interchange：電子データ交換)**
・商品の受発注や見積もりなど企業間の商取引をデジタル化し、ネットワークを利用してやり取りすること
・情報の伝達スピードを向上させることにより販売促進を図り、事務処理における人件費削減効果を目指す

【EDI の 4 つの規約】
1. **情報伝達規約**
　コンピュータ間の通信手段に関する取り決め
2. **情報表現規約**
　交換するデータの記述方法を決める規約
3. **業務運用規約**
　EDI の運用方法に関する取り決め
4. **取引基本規約**
　EDI を用いた取引に関する基本的かつ総合的な契約

⑤**CALS (Commerce At Light Speed：高速商取引)**
・ネットワークとデータベールを活用し、調達，設計，生産，運用，保守にいたるまでの製品のライフサイクル全般に関する情報を一元管理して、複数の企業間で情報を共有するシステム
・データの共有を図る上で、SGML※が標準文書規格として使われる
　※SGML …文書の論理構造や意味構造をタグで記述するマークアップ言語

⑥**SFA (Sales Force Automation)**
・インターネットと携帯端末を使い、社内のイントラネットに接続する環境を整え、営業活動の効率化，迅速化を目指した営業支援システム
・SFA を有効活用するためには、**モバイルコンピューティング(mobile computing)**環境を整えるとともに、営業する者にある程度の権限を委譲しなければならない

⑦**CRM (Customer Relationship Management)**
・顧客情報や顧客対応履歴を蓄積して管理する、顧客管理システム
・多面的な情報を蓄積することで、次のニーズを予測することができる

⑧**ASP (Application Service Provider)**
・インターネットを通じてサーバやアプリケーションなどの機能を提供する企業のこと
・保守作業やバージョンアップ作業などをする必要がなくなり、人件費などのコストを削減できる

⑨**ファームバンキング (firm banking)**
・企業と金融機関をコンピュータでネットワーク接続し、金融機関の各種サービスをオフィスにいながら受けられるようにしたシステム
・振込みや送金が安い手数料で可能。最近は**ネットバンキング**で多く利用

⑩**データウェアハウス**
・データ分析による意思決定支援のための全社的なデータの集まり
・データを大量に蓄え整理し、ビジネス上の意思決定に利用する

№8　コンピュータの性能評価　　★★★★★

①**ベンチマークテスト (benchmark test)**
・実際に使用する環境の下で、コンピュータを実行させたときのハードウェアやソフトウェア性能を評価するテスト
・コンピュータの性能指標を示すことで計測したコンピュータ性能の高低を相対的に把握できる

※カーネルプログラムテスト
　基本的な計算問題について、対象とするコンピュータで CPU 使用頻度の高い命令を実行するプログラム(カーネル)をつくり、その実行時間を測定し、他のシステムでの結果と比較評価を行う

②**TPC (Transaction Processing Performance Council)**
・実環境の下でのコンピュータの実行速度の性能指標を明示する非営利団体
・別名：**トランザクション処理性能協議会**
・TPC に策定されたベンチマークは TPC-A，TPC-B，TPC-C，TPC-D の 4 タイプがある。TPC-A，TPC-B は現在はほとんど使われない。TPC-C が主流で、TPC-D は策定から日が浅い
・単位：tpm (transactions per minute)

1) **TPC-C**
・トランザクション処理システムの性能指標を示すもの

・主にデータベースシステムのトランザクション性能を測定する
・オンラインで端末からホストへ処理要求を出してから処理結果を端末に送り返すまでの処理 **OLTP(On-Line Transaction Processing：オンライントランザクション処理)**の性能評価で行われる
・単位：tpmC

2) TPC-D
・意思決定システムの性能指標を示すもの
・単位：tpmD

3) トランザクション処理能力の計算

1 秒間に処理できるトランザクション数
= (装置の MIPS 値×使用率)/1 トランザクションの命令数

【計算例】CPU の性能…20MIPS ， CPU 使用率…80%
1 トランザクションの命令数…100 万命令
→1 秒間に処理できるトランザクション数
$= (20\times10^6\times0.8)/(1\times10^6) = 16$ 件/秒

③**SPEC (Standard Performance Evaluation Corporation)**
・UNIX で動作するコンピュータの実行速度の性能指標を明示する非営利団体のこと

【SPEC の 3 部門】
1) **OSG (Open System Group)**
一般的なコンピュータの性能評価を行う部門。UNIX システムの CUP の性能評価をする SPECint (整数演算性能評価)や SPECfp (浮動小数点性能評価)、Java プログラムの実行速度を性能評価する SPECjvm や Web サーバを性能評価する SPECweb がある。

2) **HPG (High Performance Group)**
大規模コンピュータや高性能なコンピュータの性能評価をする部門。

3) **GPC (Graphics Performance Characterization group)**
グラフィックスの性能評価をする部門

№9　品質管理法　　★★★★★

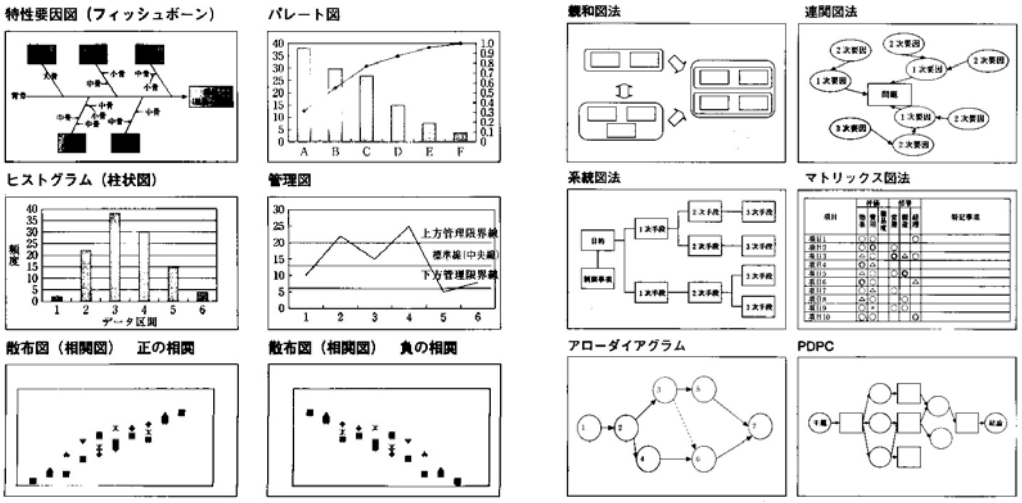
①**QC (品質管理：Quality Control)**
・消費者のニーズにあった品質の商品またはサービスを最も経済的な水準に基づいて正しく、より効率的に作り出すための手法
・品質管理は、
計画(Plan) － 実施(Do) － 調査(Check) － 処置(Action)
のサイクルで管理活動を進める

▼QC の 7 つ道具

名称	内　容
層別	データを項目分けして、項目間の違いから原因を探る
チェックシート	データにミスや漏れがないかチェックするためのシート
パレート図	影響の大きな項目を調べる。棒グラフ，折れ線グラフを使う。ABC 分析として使う
ヒストグラム	データ分布の形，中心，バラツキを把握する
特性要因図	結果に対して様々な原因を整理して体系的に表現する
散布図	測定データを 2 次元のグラフにプロットし、データの相関を把握する
管理図	工程が正常か異常か判断する

▼新 QC の 7 つ道具

名称	内　容
親和図	データ項目間の類似性や主従関係を調べ、問題点を明らかにする
連関図	複雑に絡み合う要因の因果関係を図示して明らかにする
系統図	問題解決する為の手段の関係をツリー状に表現する
マトリックス図	項目同士の関連や優劣を表にする
マトリックスデータ解析法	数値データの相関関係を分析する
アローダイアグラム	作業工程や日程を矢線で表し管理する
PDPC	目標達成までの道筋をフロー図で表す



1) **層別**
- データを何らかの基準(時間別，作業者別，機械別など)によってグループに分類し分析すること
- 小グループ(層)に分けることによって、データの整理や問題の原因を明らかにする

2) **チェックシート**
- あらかじめ行動内容や測定や観測内容を項目別にチェックし、数値を入力することで、行動やデータに漏れがないか分かるようにまとめた図

3) **パレート図**
- 改善すべき現象や問題点を原因別に分類し、それを大きい順に並べた棒グラフと、それらの累積和を折れ線グラフで表した図

◎**ABC 分析**

構成比率の高いものから順にデータを並べ、例えば構成比 80%までの項目を A、構成比 80～95%までの項目を B、その他を C というランク付けし、パレート図を用いて項目の重要度を分析する方法

【各クラス(グループ)の管理法】
- A グループ：**定期発注方式**
  少量ずつ頻繁に発注する
- B グループ：**定量発注方式(発注点方式)**
  ある程度まとめて発注し、発注の手間を削減する
- C グループ：**定量発注方式(発注点方式)**，**2 ビン法(ダブルビン法)**
  在庫管理に手間をかけないようにするため、簡便な管理を実施する

4) **ヒストグラム**
データの存在する範囲をいくつかの区分に分け、各区間に入る出現頻度を棒グラフにしたもの

5) **特性要因図 (フィッシュボーンダイアグラム)**
結果である特性に対して、影響を及ぼす様々な要因との因果関係を矢線でつないで体系的に整理した図

6) **散布図**
- 測定値を 2 次元グラフにプロットし、データの相関を調べる図

◎**相関係数**

相関関係の強さを測る指標のこと。相関係数は-1～+1 の値をとり、絶対値が 1 に近いほど相関が強い

◎**回帰分析**

相関のあるデータの分析をもとに傾向を掴んで予測する手法

7) **管理図**
- 工程における偶然原因によるバラツキと異常原因によるバラツキを判断して、工程を管理するもの
- 1 本の**中心線**(CL：Center Line)とその上下に合理的に決められた**管理限界線**(**上部管理限界線**(UCL：Upper Control Limit)，**下部管理限界線**(LCL：Lower Control Limit))で管理状態を判断

8) **連関図**
- 因果関係を整理し問題に深く関与している要因を見つけ出す図
- 解決すべき問題は分かっているが、その問題要因が複雑に絡み合っているため解決の糸口が見つけられない場合に有効

9) **アローダイアグラム**
- 作業の順序関係を矢線、作業と作業の結合点には○印を使用したネットワークチャート
- プロジェクトの作業管理などで利用

№10　企業の会計と経営分析　　★★★★☆

①**財務諸表 (financial statements)**

企業の財務状況を示す会計諸表

1) **貸借対照表 (B/S：Balance Sheet)**

資本に着目し、企業の資本がどれくらい増減したかを把握するための会計表

資産 ＝ 負債＋資本＋純利益

2) **損益計算書 (P/L：Profit and Loss statement)**

利益に着目し、どのような利益をどれくらい損失したかを把握するための計算書

費用＋純利益 ＝ 収益

3) **利益**

売上総利益 ＝ 売上高－売上原価
営業利益 ＝ 売上総利益－(販売費＋一般管理費)
経常利益 ＝ 営業利益＋営業外収益－営業外費用

4) **売上原価**

売上に対応する商品の仕入原価のこと

売上原価 ＝ 期首商品棚卸高＋当期商品仕入高－期首商品棚卸高

②経営分析

1) **損益分岐点 (breakeven point)**

利益が生まれるかどうかの分岐点

利益 ＝ 収益－費用
＝ 売上高－(変動費＋固定費) ＝ 0　…①

※**変動費**…売上高に比例して増減する費用
固定費…売上高に関係なく発生する費用

変動費は売上高に比例するので、

変動費 ＝ 売上高×変動費率　…②

よって、変動費率は、

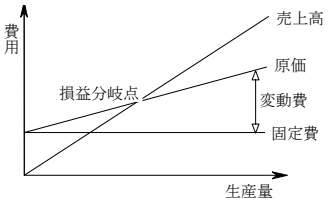
$$\text{変動費率} = \frac{\text{変動費}}{\text{売上高}} \quad \cdots③$$

①を②に導入すると、

売上高－(売上高×変動費率＋固定費) ＝ 0　…④

④より、

$$\text{損益分岐点売上} = \frac{\text{固定費}}{(1 - \text{変動費率})} \quad \cdots③$$



2) **線形計画法 (LP：Linear Programming)**

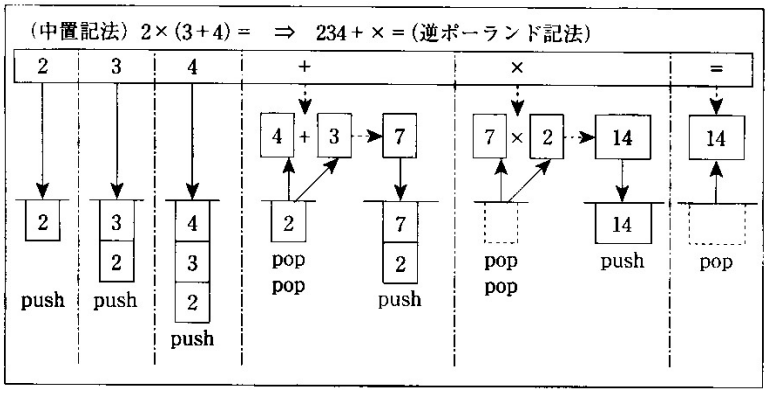
・複数の一次不等式等を制約条件として、最適解(最大，最小)を求める
・最適解を求める関数を目的関数という
　数式解法 … 数式で求める
　図式解法 … 図(グラフ)で求める

第 10 章　追加情報

№1　追加情報

○逆ポーランド記法

・計算順序を示す括弧が不要
・スタックデータ構造を用いると用意に演算機構が実現できる



○構文図(syntax chart)

・プログラム言語 Pascal の構文記述のために考案された表記法
・構文規則が視覚化されて分かりやすい

○排反事象

A∩B = φ(空事象)　⇒ A と B は排反事象である　⇒　P(A∩B) = P(φ) = 0

○エンドユーザコンピューティング (EUC：End User Computing)

エンドユーザが自分でコンピュータを操作して、必要な結果を入手すること

○エンドユーザデベロプメント (EUD：End User Development)

与えられた処理手順だけでなく、自発的に処理手順を開発しシステムの開発を行うこと

○システムインテグレーション (SI：System Integration)

ユーザの目的に合わせ、最適なハードウェア，ソフトウェア，ネットワークの組み合わせを選択し、異種システムの統合化，システムの企画，開発から運用･保守までを一貫して手がける仕事

○移行

1) **順次移行 (パイロット移行)**
受け入れてスト完了後、全てのユーザに対して機能を提供するのではなく、特定の部門などユーザを限定して運用を開始し、新システムの機能，レスポンス，運用などを評価する導入方法

2) **一斉移行**
・システム範囲の全てのユーザに対して一斉に移行する方法
・移行に伴う運用は保守の負担は順次移行に比べて少ない

○フォーマット処理

1) **物理フォーマット (ローレベルフォーマット)**
ディスク内をトラックやセクタに分割する

2) **論理フォーマット (イニシャライズ)**
物理フォーマット後のディスクに対して、OS が固有で使用する管理用データや実際に記録されるデータの論理的な位置を設定する

○リスク管理策

1) リスク低減　：対策を講じることでリスクを減少させる
2) リスク移転　：保険の利用などによってリスクを他者に移す
3) リスク回避　：脅威の発生要因を取り除くことでリスクの発生を防ぐ
4) リスク許容　：特別な措置をとらず、リスクの存在を認める

○**ソフトウェアライフサイクルプロセス (SLCP)**

ソフトウェアには企画から廃棄に至るライフサイクルがあり、ソフトウェアサイクルを構成する段階の組み合わせのこと

※**ソフトウェア開発取引の共通フレーム'98**

SLCP の国際標準化の流れに呼応し、国内で経済産業省の主導で作成･発表されたもの

○**ISO/IEC9126**

【ソフトウェアの品質特性】

1) 機能性　：必要な機能の実現度合い
2) 信頼性　：一定期間、一定の機能を維持する度合い
3) 使用性　：使用する上での労力の少なさ、操作のし易さの度合い
4) 効率性　：情報資源を無駄なく使用する度合い
5) 保守性　：変更し易さの度合い
6) 移植性　：他のシステム環境への移行し易さの度合い

○**不正競争防止法**

【営業秘密(トレードシークレット)が満たす要件】

秘密管理性	情報が秘密として管理されていること(営業秘密に該当することが客観的に認識できるレベルの管理が行われていること)
有用性	情報が営業上､技術上の有用な情報であること（客観的に見て､企業に収益をもたらしている､あるいはもたらす可能性が大きいと判断できること）
非公知性	情報が公然とは知られていないこと（業界内で周知の情報である､刊行物に掲載済みであるなど、公知されていることが無いこと）

○**経営情報システム**

1) **基幹情報処理システム (EDPS：Electronic Data Processing System)**

販売･生産･会計等の企業の基幹業務効率化による経営の合理化を目的としたシステム

2) **経営情報システム (MIS：Management Information System)**

経営管理者や専門スタッフに対して、全社レベルの情報を一元化･体系的に管理し、意思決定を支援するために提供することを目的としたシステム

3) **意思決定支援システム (DSS：Decision Support System)**

経営管理社会層の各管理者，経営者等の意思決定を支援し、有効性を高める目的のシステム
EDPS，MIS も意思決定を支援するが、DSS はユーザ自身が自分で自由にデータを検索し、データ分析に基づいて経営や業務で必要な戦略を立案し、アクションを起こすことを支援するシステム

※ **OLAP (OnLine Analytical Processing)**
…ユーザ･インタフェースとして、多次元分析ソフトなどのツール

4) **戦略情報システム (SIS：Strategic Information System)**

DSS の発展系として提唱された企業の競争戦略実現のための情報システム

○**在庫評価方法**

1) 先入先出法
在庫品を出庫する場合、入庫した古いものから順に取り出すことを前提として在庫評価額を算定する方法
2) 後入先出法
在庫品を出庫する場合、入庫した新しいものから順に取り出すことを前提として在庫評価額を算定する方法
3) 移動平均法
在庫が移動するごとに在庫評価額の平均単価を算出する方法
4) 総平均法
月末や決算日時点で締め、それまでの期間を通じて在庫評価額の平均単価を計算する方法
5) 最終仕入原価法
最終に仕入れ(入庫)した単価を常に在庫評価額として使用する方法

○意思決定理論

意思決定とは、将来起こりうる状態を予測し、それに基づいて行動を選択決定すること

例）利益表　　（単位：百万円）

	好転	現状維持	悪化
A1	900	300	200
A2	250	350	400

1）将来起こりうる状況の確率が予測できる場合

a）**期待値原理**

各方策の利益と経済状況の発生確率をかけて、足し合わせたものを計算し、これが最大のものを選択する

A1：900×0.3 + 300×0.6 + 200×0.1 = 470 百万円　⇒　○

A2：250×0.3 + 350×0.6 + 400×0.1 = 325 百万円

2）将来起こる状況の確率が予測できない場合

a）**ラプラスの原理**

全ての現状が同じ確率で起こると考えて方策を決定する

A1：900×1/3 + 300×1/3 + 200×1/3 = 467 百万円　⇒　○

A2：250×1/3 + 350×1/3 + 400×1/3 = 333 百万円

b）**マクシミン(Maxmin)原理**または**ミニマックス(Minimax)原理**

悲観的な考えに基づいた基準で、最悪の状態が起こった場合を想定し、各代替案の最小利益を最大にするものを選択する

A1 の最小利益 = 200 百万円、A2 の最小利益 = 250 百万円　⇒　A2 を選択

c）**マクシマックス(Maximax)原理**または**ミニミン(Minimin)原理**

楽観的な考えに基づいた基準で、最良の状態が起こった場合を想定し、各代替案の最大利益を最大にするものを選択する

A1 の最大利益 = 900 百万円、A2 の最大利益 = 400 百万円　⇒　A1 を選択

d）**ミンマックスリグレット(Minmax Regret)原理**

ある状況が起こった時、最良の方策を選択していなかったため後悔(リグレット)する度合いが最小となる方策を選択する

№2　SQL

①データ構造の定義

スキーマ定義　CREATE SCHEMA
表定義　CREATE TABLE
ビュー定義　CREATE VIEW
権限定義　GRANT

②データ操作

検索　SELECT
挿入　INSERT
削除　DELETE
更新　UPDATE

③データ定義

主キー　PRIMARY KEY
外部キー　FOREIGN KEY

④関係演算

・**射影**：表から必要なカラム(列)のデータを抽出
・**選択**：表から必要なタプル(行)のデータを抽出
・**結合**：複数表をある共通項目(結合キー)で連結し、1 つの表として扱う

⑤その他の SQL

(1) タプル(行)の排除：DISTINCT
データを取得する際、同じ内容を排除して表示したり、件数をカウントする時に使用する。

(2) 別名：**カラム名** AS **表示名**
AS で指定しなければ表のカラム名が使用されるが、AS を使用すれば別名で表示できる

(3) グループ処理：GROUP BY
指定したカラム(列)が、同じ値のタプル(行)をグループ化して集計などを行うことができる。この場合、合計や件数のカウントを行う関数を使用できる。

```
[公式]
SELECT ｷｰｶﾗﾑ, …, 関数 1(ｶﾗﾑ名), 関数 2(ｶﾗﾑ名), …
        FROM 表名
        GROUP BY ｷｰｶﾗﾑ 1, ｷｰｶﾗﾑ 2, …
```

なお、GROUP BY を使用する場合に、SELECT 句で指定できるカラムは、GROUP BY で指定しているカラム(集計キーカラム)か、関数を使って演算を行うカラムに限られている。しかし、GROUP BY で指定したカラムを必ず SELECT 句に記述する必要はない
SELECT 句で指定した集計関数を除く他の項目名を記述しなければならない

(4) グループ後データの検索条件：HAVING
グループ化した結果に対し、さらに条件指定したい場合は HAVING を使う。WHERE 句は集計前の選択に使用するが、HAVING は集計後の項目を使って条件判定を行う。

```
[公式]
SELECT ｶﾗﾑ名 1, 関数 1(ｶﾗﾑ名), 関数 2(ｶﾗﾑ名), …
        FROM 表名
        GROUP BY ｷｰｶﾗﾑ 1, ｷｰｶﾗﾑ 2, …
        HAVING (抽出条件)
```

(5) 並べ替え(ソート)：ORDER BY
指定したカラム(列)で取得したデータを並べ替えたい場合に使う。

```
[公式]
SELECT ｶﾗﾑ名 1, …　FROM 表名
        ORDER BY ｿｰﾄｶﾗﾑ名 1 ASC, ｿｰﾄｶﾗﾑ名 2 DESC, …
```

※　ASC … 昇順（省略可）
　　DESC … 降順

(6) その他

・BETWEEN

［公式］

WHERE ｶﾗﾑ名 BETWEEN “A” AND “B”

A から B の範囲のデータを抽出する
A, B の値も含まれる

・IN

［公式］

WHERE ｶﾗﾑ名 IN (A, B, C, …)

( )内に指定するいずれかの値が含まれているデータを抽出

・LIKE

［公式］

WHERE ｶﾗﾑ名 BETWEEN “A” AND “B”

・NULL

［公式］

WHERE ｶﾗﾑ名 IS NULL (1)
WHERE ｶﾗﾑ名 IS NOT NULL (2)

(1) データが空白(NULL)のものを抽出
(2) データが空白(NULL)でないものを抽出