

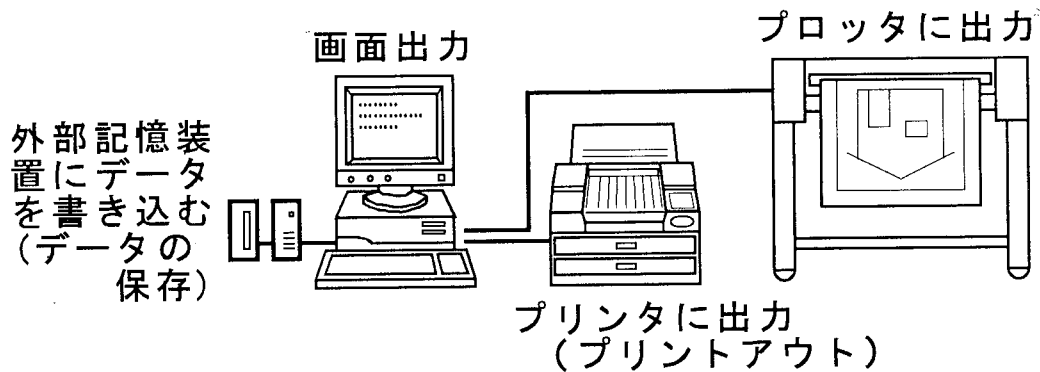
### Ⅲ 日本語（片仮名）

#### 1. ア行

[ア]

**アウトプット output** : [情報処理概論] [情報技術概論] , レベル1

外部記憶装置にファイルとしてデータを書き込む、又は、ディスプレイ、プリンタ、プロッタなどの周辺機器にデータを送ること（画面表示、印刷など）を示す（図Ⅲア-1）。



図Ⅲア-1 様々な“アウトプット”

<ファイル名 : ア 0001.HTM >

**アンインストール uninstall** : [情報処理概論] [情報技術概論] , レベル1

インストールしたシステムやアプリケーションを削除して、インストールする前の状態に戻すこと。主として、ハードディスクにインストールしたアプリケーション（プログラム及びそのプログラムが呼び出す関連のファイル）を削除して、インストール以前の状態に戻すことを示す。Windows95用のアプリケーションでは、このアンインストール機能の装備が推奨されている。

なお、アンインストール機能のないアプリケーションを削除する場合は、プログラムの削除だけでなく、すべての関連ファイルの削除が必要となり、場合によっては、システム側の設定ファイルの書き換えも必要となる。

<ファイル名 : ア 0002.HTM >

**アンデリート undelete** : [情報処理概論] [情報技術概論] , レベル1

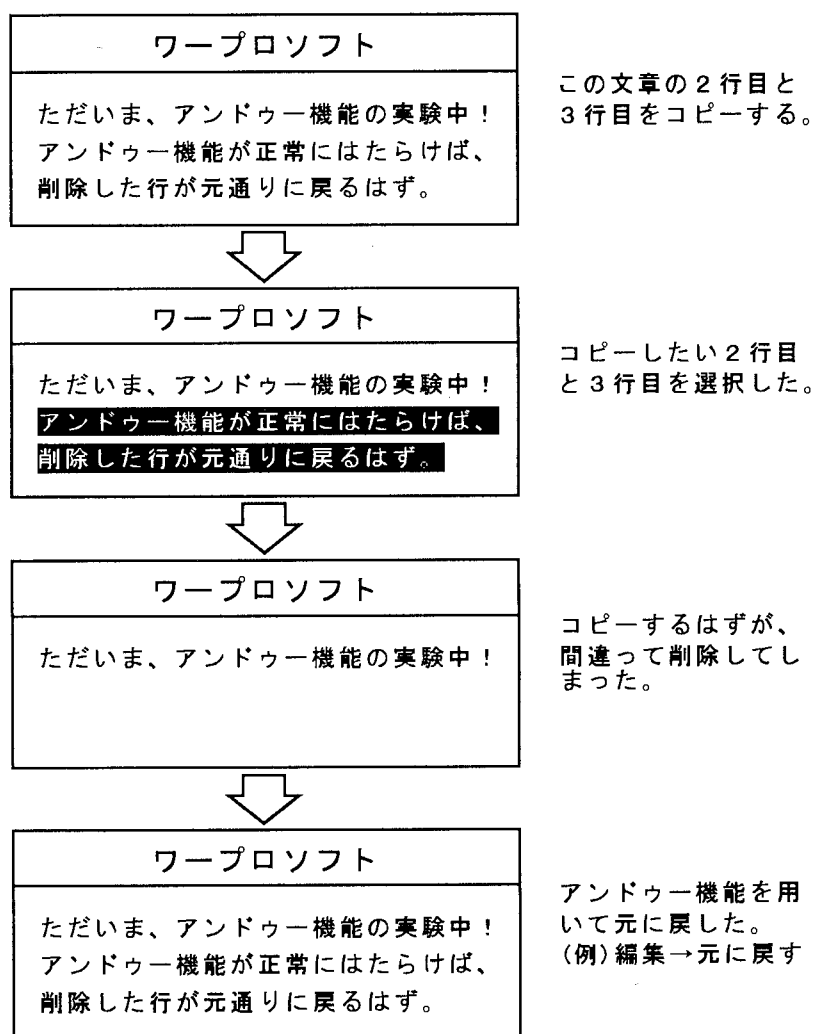
デリート（削除）したファイルを、デリートする以前の状態に戻すこと。すなわちファイルを元に戻すこと。間違ってファイルをデリートした場合、直後にアンデリートを行うと、ファイルを元に戻すことができる。MS-DOSのUNDELETEコマンドに対応する。

<ファイル名 : ア 0003.HTM >

## アンドゥー undo : [情報処理概論] [情報技術概論] , レベル1

ワープロソフトや表計算ソフトなどのアプリケーションソフトにおいて、直前に実行した処理を取り消して、その処理以前の状態に戻すこと。最近のアプリケーションソフトでは、必ずといってよいほどアンドゥー機能が装備されている。

例えば、ワープロソフトにおいて間違っって行やページの削除を行った場合、また、表計算ソフトにおいて間違っってセルの削除を行った場合などは、このアンドゥー機能で元に戻すことができる(図Ⅲア-4)。



図Ⅲア-4 アンドゥー機能の一例

ただし、個々のソフトウェア若しくはそこで行う処理によってアンドゥー機能の適用に大差があり、アンドゥーが行えない処理やアンドゥーが1回のみしか行えない場合もある。アンドゥーが1回のみしか行えない場合は、間違っった処理をした後に何らかの処理をすると、間違っう以前の状態に戻すことは不可能となる。

<ファイル名 : ア 0004.HTM >

アンフォーマット **unformat** : [情報処理概論] [情報技術概論] , レベル1

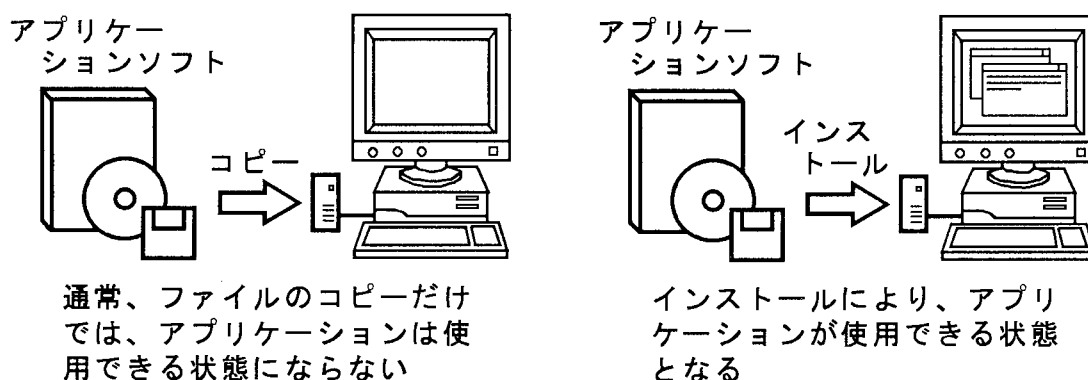
フォーマット (初期化) したディスクを、フォーマットする前の状態に戻すこと。間違っ  
てフォーマットした場合、直後にアンフォーマットを行うと、フォーマットする以前の状態  
に戻すことができる。MS-DOS の UNFORMAT コマンドに対応する。

<ファイル名 : ア 0005.HTM >

[イ]

インストール **install** : [情報処理概論] [情報技術概論] , レベル1

ソフトウェアパッケージは、通常、そのままの状態では使用できない。インストールとは、  
ソフトウェアをハードディスクやフロッピーディスクなどの記憶装置に入れ、ユーザがいつ  
でも使える状態にすることであり (図Ⅲイ-1)、単にファイルをコピーすることは異なる。  
最近では、処理速度の問題から、ハードディスクにインストールする場合はほとんどで  
ある。



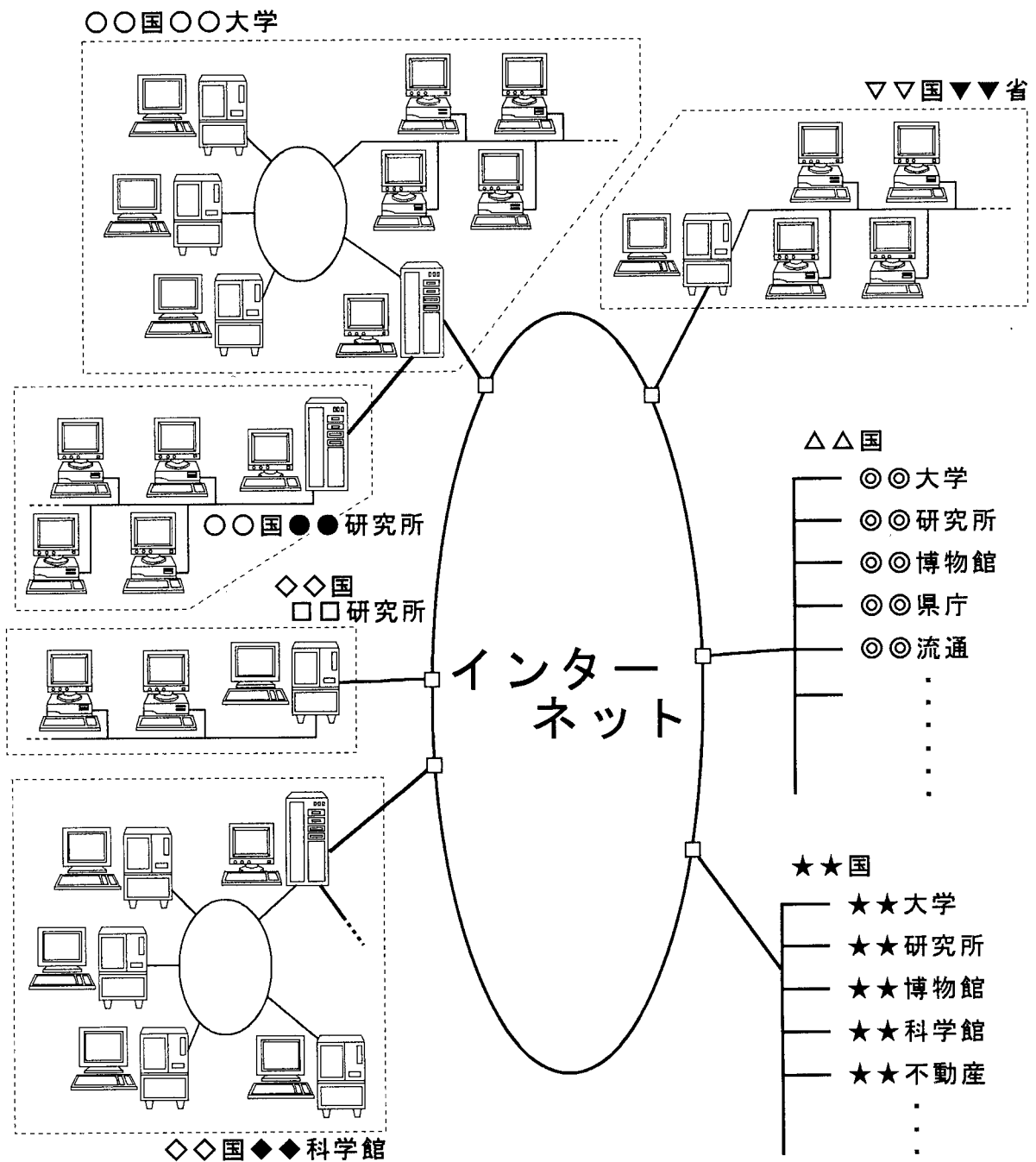
図Ⅲイ-1 コピーとインストールの違い

<ファイル名 : イ 0001.HTM >

インターネット **Internet** : [情報通信工学] [コンピュータネットワーク] , レベル1

世界中のあらゆる機関のあらゆるコンピュータが接続された最大のコンピュータネットワーク (図Ⅲイ-2)。インターネットの起源は 1969 年に開始された ARPANET (アーパネット) であり、当初の軍事目的から、学術目的、商用目的、趣味・娯楽目的へと発展し、今日のような複合的な目的をもつ大規模なネットワークが構築されることとなった。現在では、各国の政府機関や大学、研究機関、企業はもとより、各種の公共機関、更には個人など、利用者は年々急増している。

インターネットの最大の利点は、国境が存在しないということであり、例えば、日本の任意のネットワークからアメリカの任意のネットワークへ、瞬時にして接続できるということである。電話サービスや FAX サービスにおいても同様のことが行えるが、インターネッ



図Ⅲイ-2 インターネット概念図

トでは WWW ブラウザを用いることにより、より視覚的に、より速く、より大量のデータを扱える利点がある。更に最近では、ネットワーク上で動画を表示（再生）することも可能となり、高速回線の普及によってデータ転送速度が向上すれば、更に利用価値は高まるものと考えられる。

なお、個人でインターネットを使用するときは、電話の接続を電話会社に申し込む場合と

同様、インターネットへの接続を商用プロバイダに申し込むだけである。自宅からインターネットで、アメリカの任意のネットワークに接続したいからといって、アメリカのプロバイダやネットワーク機関に申し込む必要など一切ない。

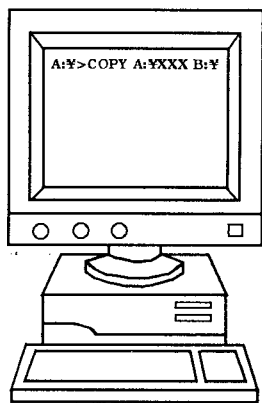
<ファイル名：イ 0002.HTM >

### インタフェース interface：[計算機工学] [マイコンシステム設計]，レベル1

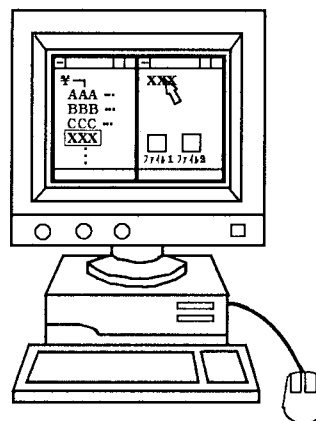
コンピュータ本体とそれに接続される周辺装置、OS（オペレーティングシステム）とアプリケーション、OS及びアプリケーションとユーザなど、異なる2つ以上の要素の接続部分（ハードウェア及びソフトウェア）若しくはそこで行われる情報伝達の方法（ソフトウェア）。

通常、ハードウェアを意味することが多く、有名なインタフェースとして、ハードディスクやCD-ROMなどを接続するときに用いるSCSI、プリンタを接続するときに用いるセントロニクスなどがある。

また、コンピュータとユーザとのインタフェースは、マンマシンインタフェース若しくは単にユーザインタフェースとよばれ、WindowsやMac OS、現在のUNIXなど、ユーザがキーボードからコマンドを入力するのではなく、マウスを用いてアイコンをクリックすることで様々な操作を可能にした環境をGUI（Graphical User Interface）と呼ぶ（図Ⅲイ-3）。



**キャラクタベースの環境**  
キーボードからコマンドを入力しなければならぬので、コマンドを覚えていなければ何もできない。



**GUI環境**  
マウスを使って、アイコンをクリック、若しくはメニューをクリックすることで操作可能。難しいコマンドを覚える必要はない。

図Ⅲイ-3 キャラクタベースの環境と GUI 環境の違い

今日のパソコンの一般家庭への普及は、パソコンのハードウェア的な高速化や高機能化よりむしろ GUI による操作性の向上によるところが大きい。

<ファイル名：イ 0003.HTM >

#### インタプリタ **interpreter**：[ソフトウェア工学]，レベル1

BASIC などのインタプリタ言語を 1 行（1 ステップ）ずつ翻訳しながら実行するプログラム。翻訳と実行を交互に行うため、コンパイラと比較して処理時間が多くかかるという欠点がある。反面、プログラムにエラーがあった場合、その箇所ですぐプログラムがとまるので、デバッグしやすいという利点がある。ただし、最近では、各種コンパイラのプログラム開発環境が非常に高度化されてきていることから、インタプリタのデバッグに関する優位性も薄れつつある。なお、コンパイラ言語として有名な C 言語、FORTRAN、COBOL などのインタプリタ版が、教育用として各ソフトウェアメーカーから市販されている。

なお、ここ数年、インタプリタ言語の代表格である BASIC について、「BASIC を用いてプログラミングするような機会はずいぶん減った」、「BASIC では古すぎる」などという話をよく聞くが、すべての場合においてこれが正しい訳ではない。周辺機器制御用のプログラミング言語としては現在でも BASIC が主流であり、また、プログラミングの容易さや過去の膨大な資産から、今後もこの分野では BASIC が大きな地位を占めるであろう。

<ファイル名：イ 0004.HTM >

#### インテル互換プロセッサ：[計算機工学]，レベル2

Intel（インテル）社製の CPU と互換性をもつ他社製の CPU。現在、有名な供給元としては、AMD（エイエムディー）社や Cyrix（サイリクス）社が挙げられ、前者は 5x86 及び 5k86 シリーズ、後者は 5x86 及び 6x86 シリーズを生産販売している。両社の 5x86 シリーズは i486 互換の CPU で、AMD 社の 5k86 シリーズ及び Cyrix 社の 6x86 シリーズは Pentium 互換の CPU である。いずれの CPU も、インテル社の CPU に対して、そのコストパフォーマンスの高さを売りにしている。

<ファイル名：イ 0005.HTM >

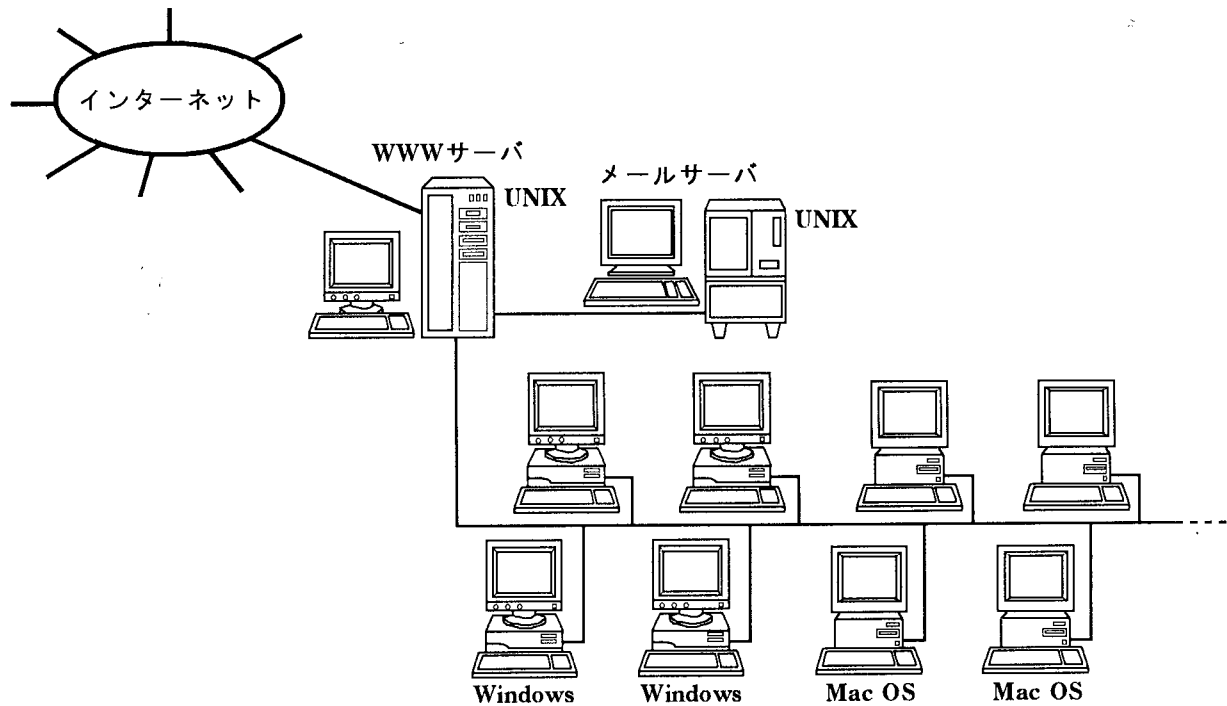
#### イントラネット **intranet**：[情報通信工学] [コンピュータネットワーク]，レベル2

インターネットで用いられている WWW（World Wide Web）や電子メールなどの様々な技術を、企業内の情報システム、すなわち社内 LAN に適用したもの。WWW ブラウザを用いることで、従来のテキストのみの情報に画像や音声を加えたデータを発信・受信することができ、また、ワークステーションやパソコンなど、コンピュータの機種を問わずに利用できる利点がある（図 III イ-6）。

同様のシステムとして、最近グループウェアが注目されているが、サーバ及びユーザの人数分若しくはコンピュータの台数分のクライアントを購入するために多額の費用がかかるこ

とから、WWWブラウザのみ基本的に揃えておけばよいイントラネットの方が、現在のところ利用価値は高い。また、インターネットと同様の技術を用いていることから、ごく簡単にインターネットへ接続できるという利点もある。

なお、同様の手法を用いて、社内だけのネットワークだけでなく、グループ傘下の関連企業や取引先企業も含めた形態のネットワークを構築することができる。社内LANと表現するには規模が大きすぎるが、これも一種のイントラネットと言える。



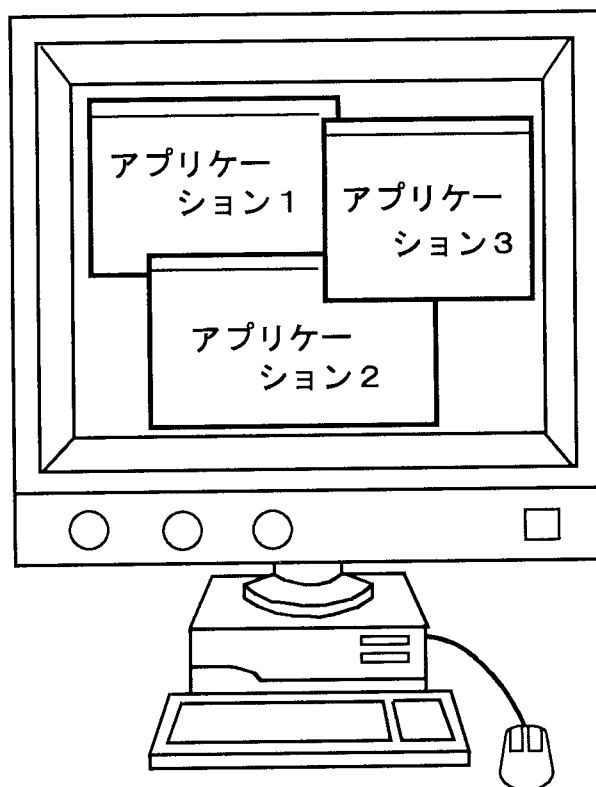
図Ⅲイ-6 イン트라ネットの一例

<ファイル名：イ 0006.HTM >

[ウ]

ウィンドウ window：[ソフトウェア工学]，レベル1

Windows や Mac OS、それらのアプリケーションなどで、画面上に表示される複数の独立した窓（これをウィンドウと呼ぶ。）のこと。また、MS-DOSアプリケーションにおける同様の機能もウィンドウと呼ぶ。ウィンドウを用いることにより、1台のディスプレイ上に複数の情報を表示することができ、これより、複数のアプリケーションを起動し切り換えながら作業することが可能となる（図Ⅲウ-1）。



図Ⅲウ-1 ウィンドウシステム

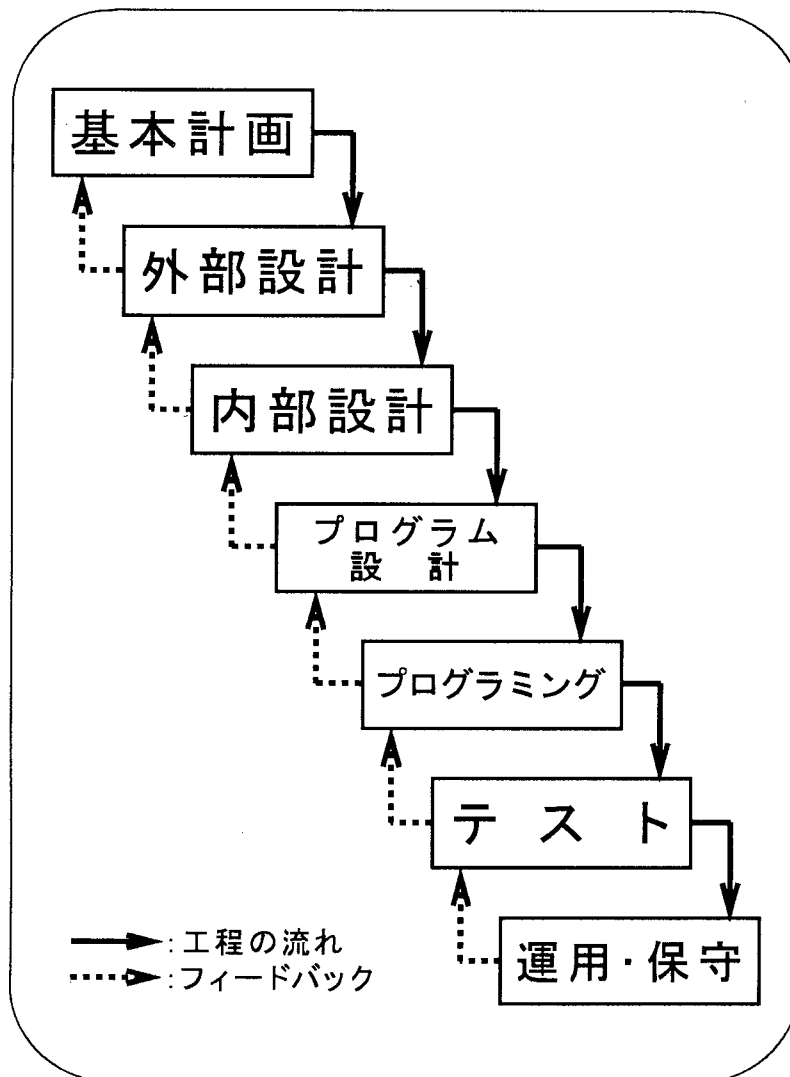
<ファイル名：ウ 000X.HTM >

ウォーターフォールモデル **waterfall model**：[システム設計] [ソフトウェア開発技法]，  
レベル1

現在のシステム開発手法の基礎となっているのがウォーターフォールモデルである（図Ⅲウ-2-1）。ウォーターフォールモデルによるシステムのライフサイクルは、基本計画、外部設計（概要設計）、内部設計（詳細設計）、プログラム設計、プログラミング、テスト、運用・保守という順に進められており、この様子が、水が滝を流れ落ちるようであることからこの名が付けられた。もちろん、水が滝を流れ落ちる過程と同様に、開発工程が逆戻りすることや所定の工程を飛び越すことはあり得ない。

従来から多くのシステムがこのサイクルに従って開発されており、現在でも多くのプロジェクトで採用されている。その大きな理由としては、各工程が明確に分かれており、それぞれの工程で作業内容のチェック・評価が可能で、しかも、ある工程での気付かなかった問題や小さなミスなどが次の工程で発見された場合は、フィードバックが許されるという点である。





図Ⅲウ-2-1 ウォータフォールモデルによる開発工程

ウォータフォールモデルでは、基本計画からプログラミングまでの工程では段階的詳細化技法（トップダウン開発）が用いられ、テスト工程では段階的統合化技法（ボトムアップ開発）が用いられる（図Ⅲウ-2-2）。なお、このときのテスト工程での結果は、

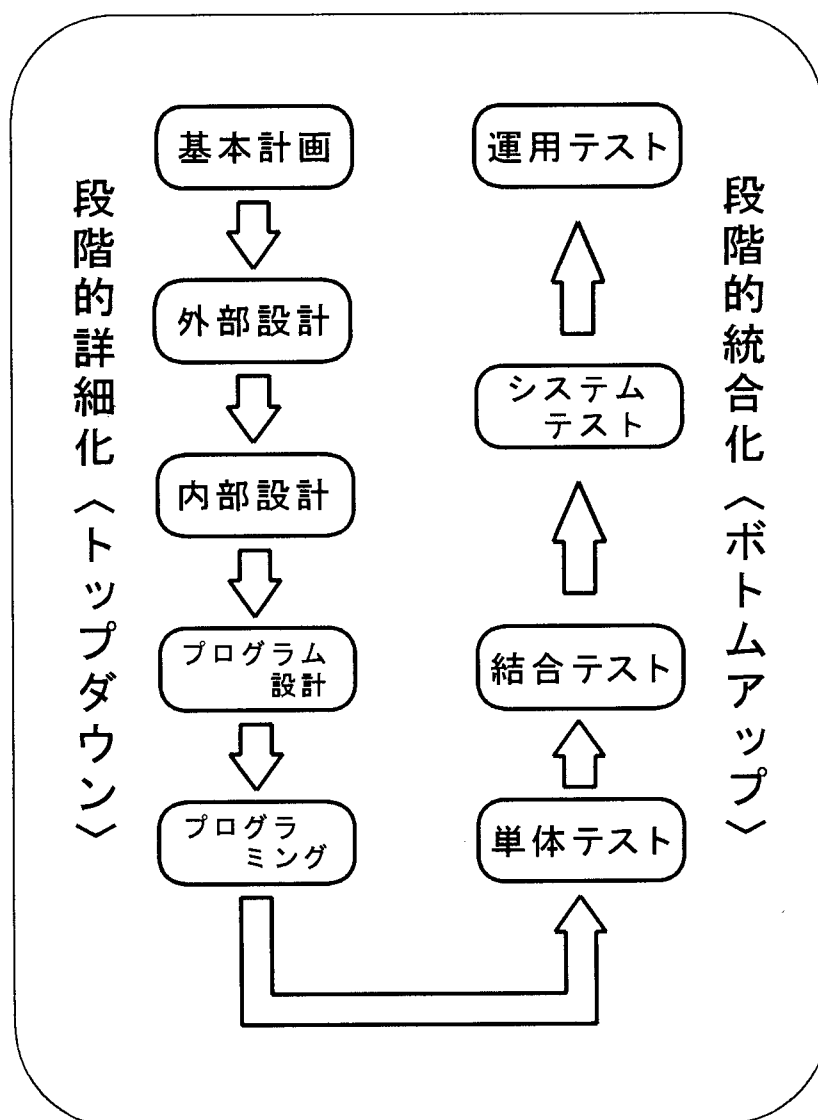
単体テスト → プログラミング

結合テスト → プログラム設計

システムテスト → 内部設計（詳細設計），外部設計（概要設計）

運用テスト → 基本計画

へと反映される。



図Ⅲウ-2-2 トップダウン開発とボトムアップ開発

<ファイル名：ウ 0002.HTM >

[エ]

エディタ editor：[ソフトウェア工学]，レベル1

テキストを編集するためのソフトウェア。ワープロソフトにあるような高度な文字装飾やページレイアウト、印刷機能をもたないため高速編集が可能である。プログラミングや環境設定ファイルの編集によく使用され、特に、プログラミング言語では、統合環境として専用のエディタが付属されている場合が多い。

<ファイル名：エ 0001.HTM >

**エンハンスドモード enhanced mode** : [ソフトウェア工学] , レベル1

Windowsの動作モードのひとつ。一般に広く普及したとされるWindows3.0は、リアルモード、スタンダードモード、エンハンスドモードの3つの動作モードを備えていた。このうち、エンハンスドモードは、アプリケーションの複数同時実行(マルチタスク)機能及び仮想メモリ(ハードディスクを仮想的なメモリとして利用する)機能などの特徴をもつ。現在主流となっているWindows95/NTでは、リアルモード及びスタンダードモードは除外され、エンハンスドモードのみとなっている。

なお、エンハンスドモードは、CPUとしてi386(互換CPUを含む)以降を対象としており、このため386エンハンスドモードともよばれる。

<ファイル名 : エ 0002.HTM >

[オ]

**オーバードライブプロセッサ ODP (Over Drive Processor)** : [計算機工学] ,

レベル2

Intel(インテル)社製のアップグレード用CPU。主として、クロックアップによる処理速度の高速化を目的としている。最近では、i486搭載機をPentium相当にアップグレードするPentium ODP、Pentium搭載機のクロックアップを目的としたPentium ODP等が市販されている。以前は、マザーボード上のODPソケットにCPUを差し込むタイプのもが多かったが、最近ではCPUを差し替えるタイプのもがほとんどとなっている。いずれの場合も、CPUだけの交換で高速化が望めることから、コストパフォーマンスは非常に高いと言える。

なお、Intel社製ODPと同様の高速化手法として、各コンピュータメーカーからCPUアクセラレータ(ボード)という製品が市販されている。Intel社製ODPがワンチップであるのに対し、CPUアクセラレータは、小さな基板の上にCPUを装荷した形状となっている。CPUアクセラレータの場合も、ODPの場合と同様、以前のCPUと差し替えて処理を高速化するものである。

<ファイル名 : オ 0001.HTM >

**オーバフロー overflow** : [情報処理概論] [情報技術概論] [計算機工学] ,

レベル2

あらゆるコンピュータには、それらが扱う値の範囲に上限(プラス側及びマイナス側)がある。オーバフローとは、その値を超えてしまうことをいう(ソフトウェア的に無限大の計算を可能としたものもあるが、ここではそのような場合を除外する)。

プログラミング言語においては、変数の型の桁数を上げることでオーバフローを回避できるが、複雑な行列演算などにおいては、計算のアルゴリズムの問題から、絶対値の極端に大きな数を絶対値の極端に小さな数で割ることもあり、このような場合、オーバフローは避け

られない。理論的にオーバーフローが起こりうる場合は論外であるが、プログラムのオーバーフローが発生するような場合は、そのアルゴリズムを変更する必要がある。

また、三角関数であるタンジェント  $\tan$  を使用する場合は、 $(2n+1)\pi/2$  近傍（ただし、 $n$  は整数）で極端に大きな値をとるので注意が必要である。

<ファイル名：オ 0002.HTM >

### オンラインサインアップ **online sign up** : [情報通信工学] [コンピュータネットワーク] , レベル2

パソコン通信サービスにおいて、書類を送付せずに、ネットワーク上で入会手続きを行うこと。通常、接続先にゲストとしてアクセスし、画面の表示にしたがって、氏名、住所、クレジットカードの番号などを入力する。

オンラインサインアップの場合、郵送による申し込みと違い、その日から通信サービスが利用できるという利点がある。ただし、不正アクセスを防止するために、仮のパスワードが発行され、正式なパスワードは後日郵送されてくる場合が多い。当然のことであるが、この間、パスワードを自由に変更することはできない。

<ファイル名：オ 0003.HTM >

### オンラインシステム **online system** : [情報通信工学] [コンピュータネットワーク] [オンラインシステム] , レベル2

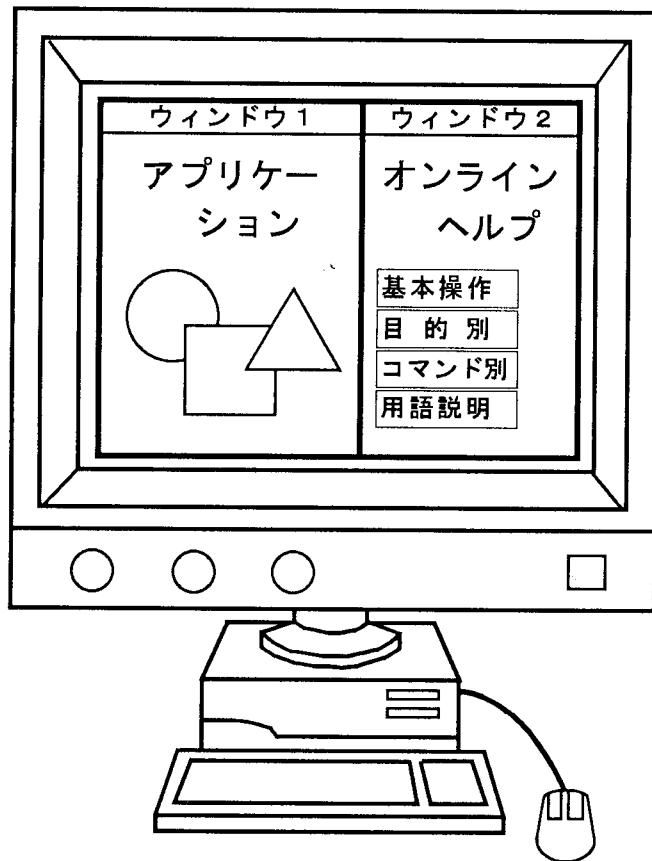
ホストコンピュータとその端末が何らかの通信回線で接続され、端末から様々なデータ処理が行えるシステム。JR や飛行機の座席予約、ホテルの宿泊予約、郵便局や銀行などの金融機関における CD サービスなどがあり、最近では、インターネットによる新聞・雑誌掲載、通信販売（オンラインショッピング）なども普及してきている。

<ファイル名：オ 0004.HTM >

### オンラインヘルプ **online help** : [ソフトウェア工学] , レベル2

ソフトウェア化された操作マニュアル。Windows や Mac OS、そしてその上で動作するアプリケーションで採用されており、実行中に操作が分からなくなった場合、この機能を用いることで、画面上に操作説明が表示される（図Ⅲオ-5）。

オンラインヘルプの形態には、目的別の操作法を階層的に検索していくものや、キーワードで検索できるものなど、アプリケーションによって様々な種類がある。なお、Windows の場合は、その上で動作するアプリケーション間で、オンラインヘルプの操作が統一化されている。また、ユーザ自らが RTF で記述したソースをヘルプコンパイラでコンパイルすることにより、簡単にヘルプファイルを作成することができ、オンラインヘルプとして活用することもできる。



図Ⅲオ-5 オンラインヘルプ

<ファイル名：オ 0005.HTM >

## 2. カ行

[カ]

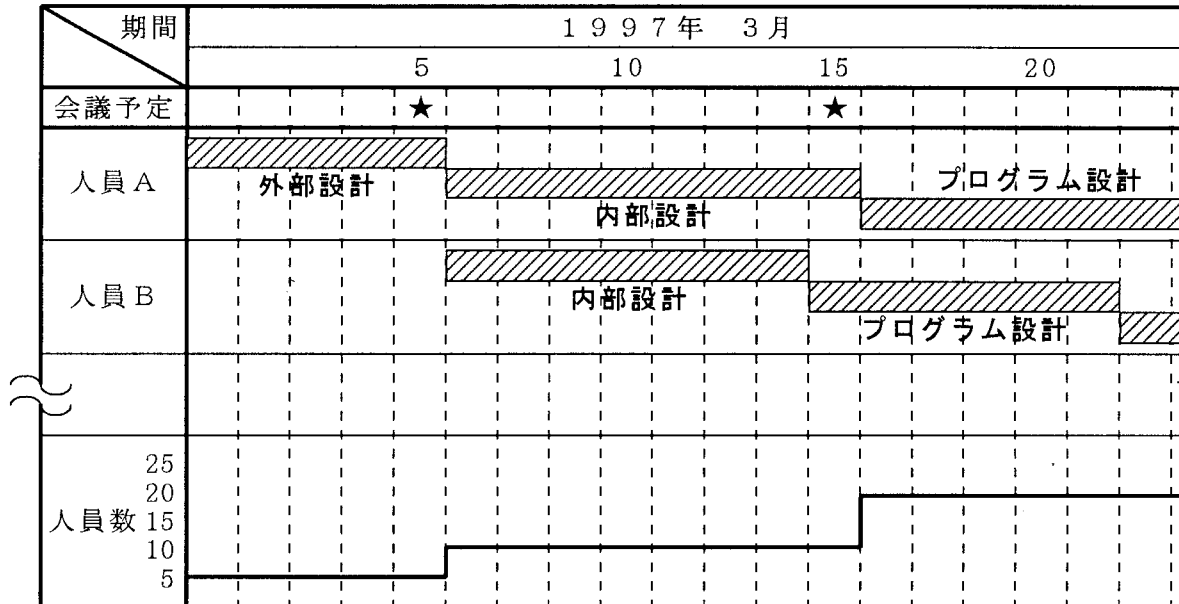
ガントチャート **Gantt chart**：[システム設計] [ソフトウェア開発技法]，レベル2  
システム開発で用いられる日程計画表。

従来のシステム開発で用いられてきた中日程計画、小日程計画のそれぞれの長所を統合化したチャートであり、横軸を時間、縦軸を担当者とし、チャートに作業内容と作業時間を棒グラフで示したものである（図Ⅲカ-1）。

ガントチャートの特徴は、システム開発全行程における日程とその進捗状況を一つのグラフで表せることである。また、納期や会議、レビューなど、日程上移動することが困難な項目に対しては、マイルストーンとよばれる目印を記入する（図カⅢ-Xでは★印を用いているが、特に決まった目印はない。）。

ガントチャートでは、システム開発の各工程において必要となる人員やその作業時間を明確に表すことができるので、他の日程計画に対して、全体的な作業量の把握がし易いという

利点をもつ。しかしながら、各工程間の関連を表すことができないという欠点を持ち、そのため大規模なシステム開発を行う場合は、PERT (Program Evaluation and Review Technique) とよばれる計画手法と併用することが望まれる。



図Ⅲカ-1 ガントチャート

<ファイル名 : カ 0001.HTM >

[キ]

[ク]

[ケ]

[コ]

コンパイラ **compiler** : [ソフトウェア工学] , レベル1

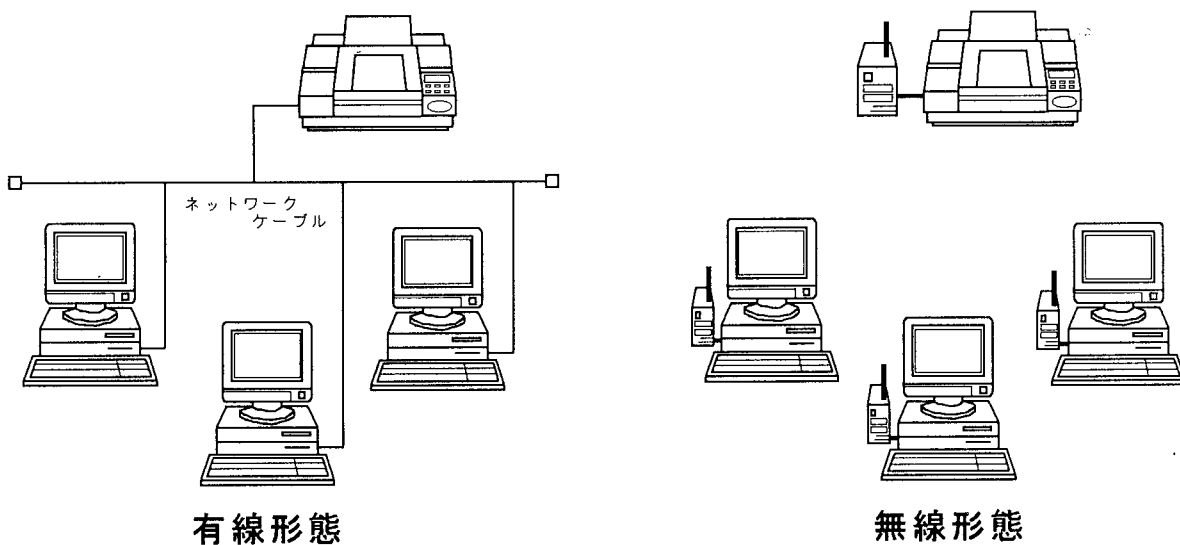
C言語、FORTRAN、COBOLなどに代表されるコンパイラ言語で記述されたソースプログラム(原始プログラム)を翻訳(コンパイル)して、コンピュータが実行できる機械語に変換するためのプログラム。通常、インタプリタ言語を用いて開発したプログラムは、その開発環境で用いた機種でしか動作しない場合が多いが、コンパイラ言語を用いて開発したプログラムは、他の機種においても、その機種で用いられるコンパイラでソースプログラムをコンパイルし直すことで実行できるという利点がある。ただし、これは、現在市販されているコンパイラ及びインタプリタに限ったことであり、あくまでコンパイラのシェアが広いことから、その汎用性が重視されているためである。

<ファイル名 : コ 0001.HTM >

コンピュータネットワーク **computer network** : [情報通信工学]

[コンピュータネットワーク] , レベル1

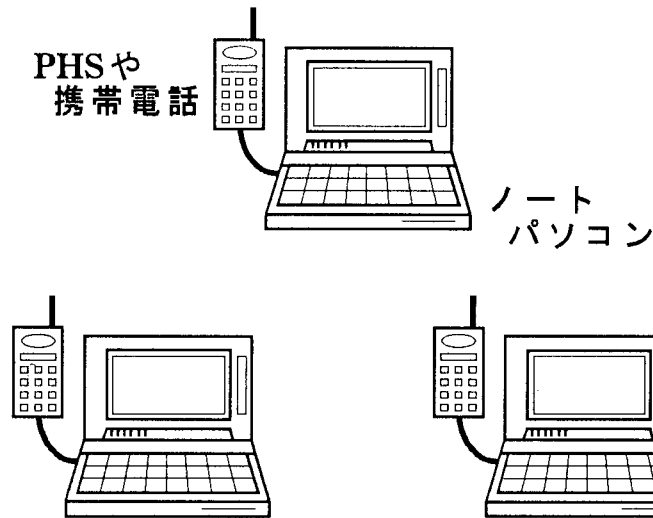
2台以上のコンピュータがネットワークケーブルを介して接続（有線形態）、又は空間に電磁波（電波）を放射することで接続（無線形態）されたシステム（図Ⅲコ-2-1）。単体のコンピュータでは不可能なデータの転送、メールの送受信、ファイルの転送及び共有化などに用いられる。



図Ⅲコ-2-1 有線形態及び無線形態のコンピュータネットワーク

現在、世界最大規模のコンピュータネットワークとなっているインターネットでは、大学、研究機関、企業、個人を問わず、世界中のあらゆるコンピュータが接続されている。

なお、最近では、PHSや携帯電話などの携帯端末をノートパソコンと接続し、場所の束縛を受けないコンピュータネットワークを構築することも可能となっている（図Ⅲコ-2-2）。



図Ⅲコ-2-2 携帯端末を用いたコンピュータネットワーク

<ファイル名：コ 0002.HTM >

### 3. サ行

[サ]

[シ]

システム開発 system development：[システム設計] [ソフトウェア開発技法]，

レベル1

システム開発とは、企業内や学内における大規模な情報システムを開発することを意味し、単なるプログラム設計・開発とは、その要素及び適用範囲が全く異なる。

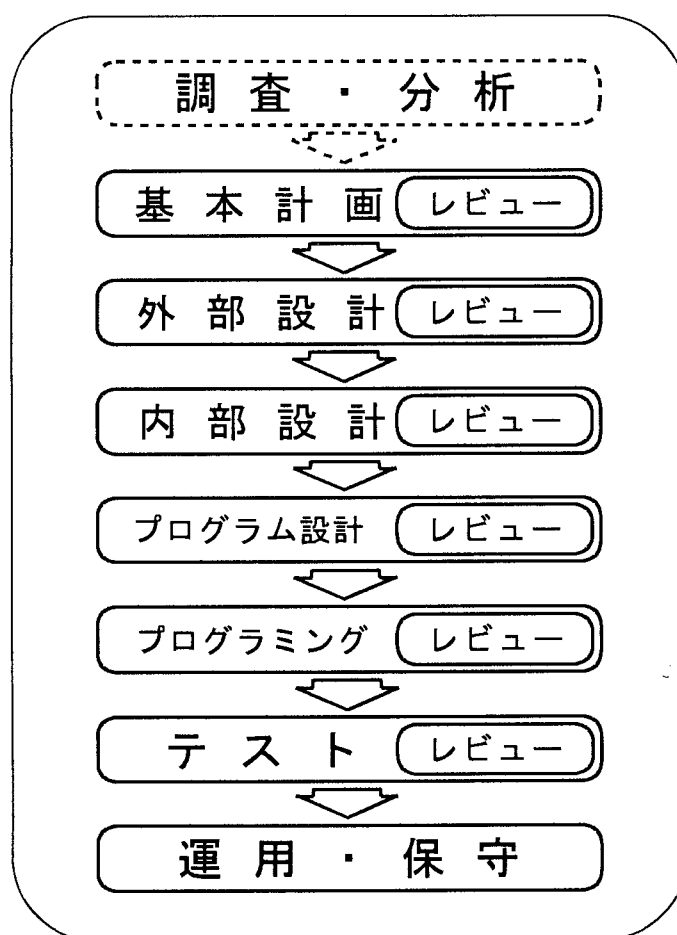
システム開発における基本的な作業手順を示したシステム開発工程のフローを図Ⅲシ-1に示す。システム自体が本来備えていなければならないと考えられる機能やユーザからの要求をもとに、開発対象となっているシステムの目的・目標を明確にするための調査・分析の工程が、システム開発のスタートとなる（図中において、この工程を破線で囲んでいる理由は、情報処理技術者試験において、この工程が次の工程である基本計画に含まれているからである。ここでは、調査・分析の重要性を考慮し、別の工程として独立させた。）。その後、基本計画、外部設計（概要設計）、内部設計（詳細設計）、プログラム設計、プログラミング、テストという工程を経て、最終的に運用・保守という工程でシステム開発は終了する。一般に、運用や保守といった表現を用いると、開発とは程遠い印象を受けるが、ここでいう運用・保守は、次のシステムへの要求を抽出するための最も優れた工程であり、システム開発において必要不可欠な工程となっている。



また、各開発工程では、その工程における作業内容を集約したドキュメント（文書）が作成されるが、最後に、このドキュメントのチェックのためにレビューと呼ばれる検証作業が行われる。

システム開発の工程は、調査・分析からプログラミングまでは、トップダウン的に順次行われていく。トップダウンとは、概要から徐々に詳細へと作業を進めていく方法であり、設計が段階的に次第に詳細となっていくことから、段階的詳細化技法とよばれている。テスト工程においては、この逆で、ボトムアップ的に細部から徐々に全体へと作業を進めていく。この方法は、テストの内容やそれぞれのプログラムが、段階的に次第に全体へと統合していくことから、段階的統合化技法とよばれている。

なお、これはあくまで一例であり、全ての企業においてこの開発工程が採用されているという訳ではない。各企業やその企業内の各プロジェクトチームでは、その企業及びプロジェクトチーム独自のシステム開発法を用いている場合が多くあり、その基本若しくは基準となっているのが、ここで示した開発工程の手順である。



図Ⅲシ-1 システム開発工程

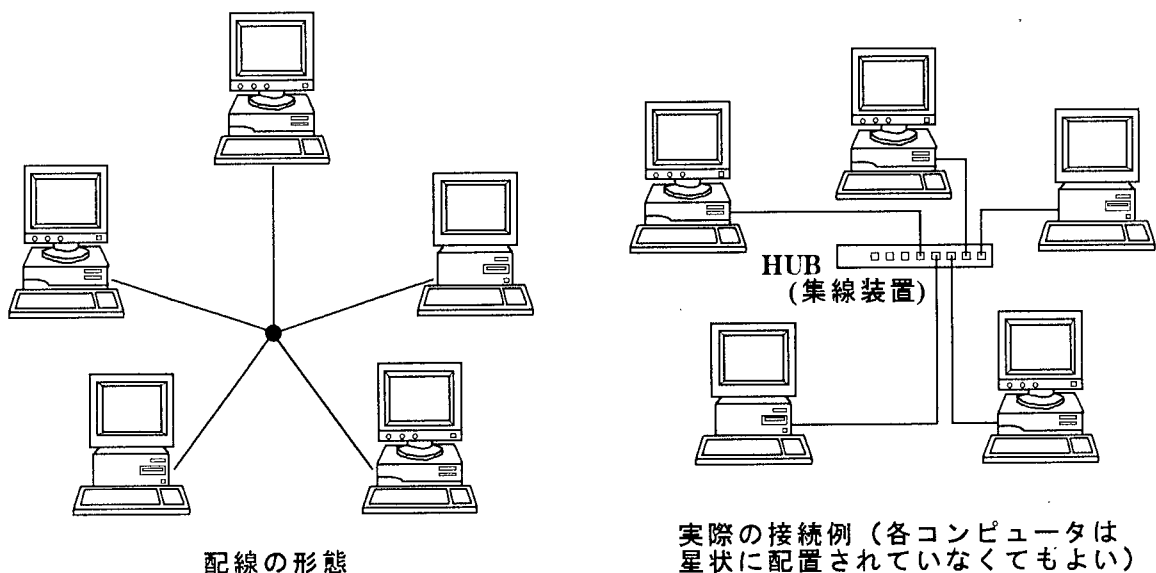
<ファイル名：シ 0001.HTM >

[ス]

スター型 LAN (スター方式 LAN) : [情報通信工学] [コンピュータネットワーク],  
レベル2

集線装置を中心に、コンピュータを星状に接続した形態の LAN。ただし、スター型 (星状) といっても、実際の物理的な配置が星状になっているわけではなく、どのような配置でもよいから、集線装置を中心として、そこから放射状に各コンピュータが接続されていればよい。通常、集線装置としては、HUB (ハブ: 10Base-T 用) が用いられる (図Ⅲス-1)。

なお、スター型では、HUB をカスケード接続することにより、1 台の HUB のポート数を上回るコンピュータを接続することができる。また、HUB には、10Base-T 用のポートの他に、10Base-2 用及び 10Base-5 用のポートが装備されていることが多く、このような場合は、接続形態において複合型のネットワークを構築することができる。



図Ⅲス-1 スター型 LAN

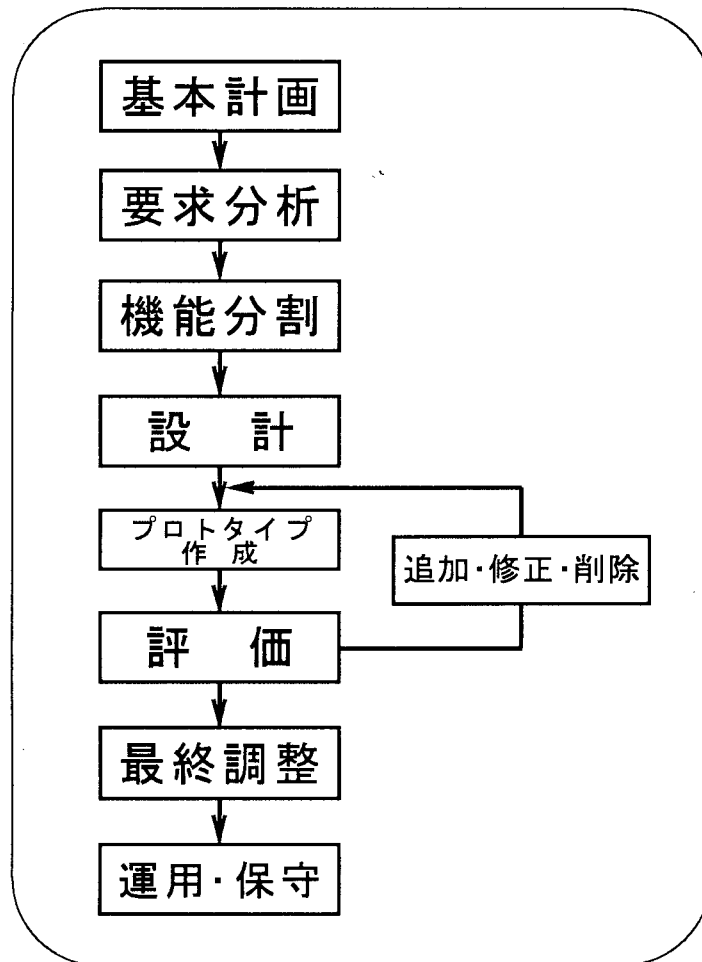
<ファイル名: ス 0001.HTM >

スパイラルモデル spiral model : [システム設計] [ソフトウェア開発技法], レベル2

スパイラルモデルは、ウォーターフォールモデルとプロトタイプモデルの利点をうまく融合したモデルである (図Ⅲス-2)。開発するシステムが独立性の高い部分・部品に分割可能などとき、すなわち機能分割が比較的容易に行える場合に、このモデルが使用できる。このモデルでは、機能分割された各ユニットごとに、設計、プログラミング、テストというウォーターフォールモデルで採用した工程を順次進めていく。また、画面設計や帳票設計など、ユーザ側に大きく関わる部分は、プロトタイプモデルで採用した試作品を作成することとなる。

スパイラルモデルでは、システムが機能分割されることにより、各ユニットごとの同時開発が可能となり、その結果、そこで必要となる人員をより明確にすることができる。

システムの機能分割が比較的容易に行える場合、このモデルの信頼性は非常に高いといえ、逆に、機能分割が困難な場合は、他のモデルによるシステム開発が妥当となる。



図Ⅲス-2 スパイラルモデルによる開発工程

<ファイル名：ス 0002.HTM >

[セ]

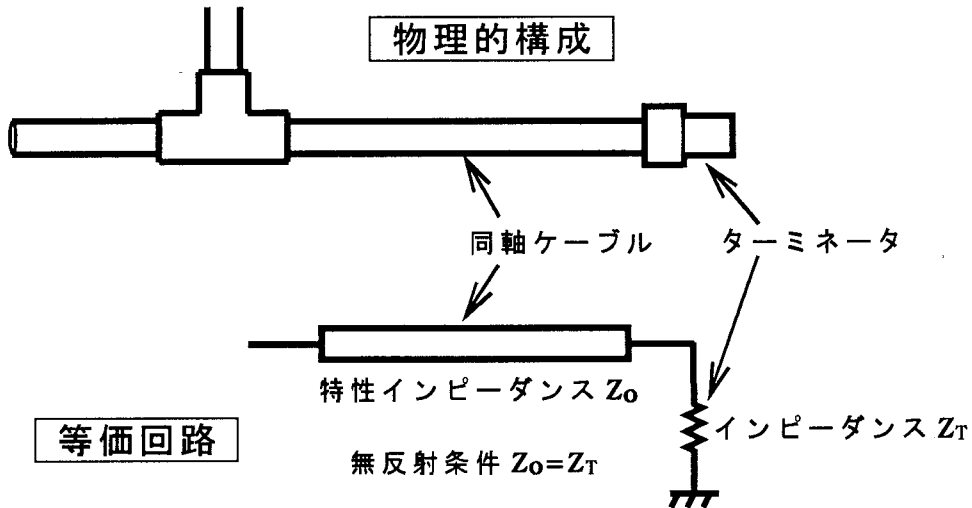
[ソ]

#### 4. タ行

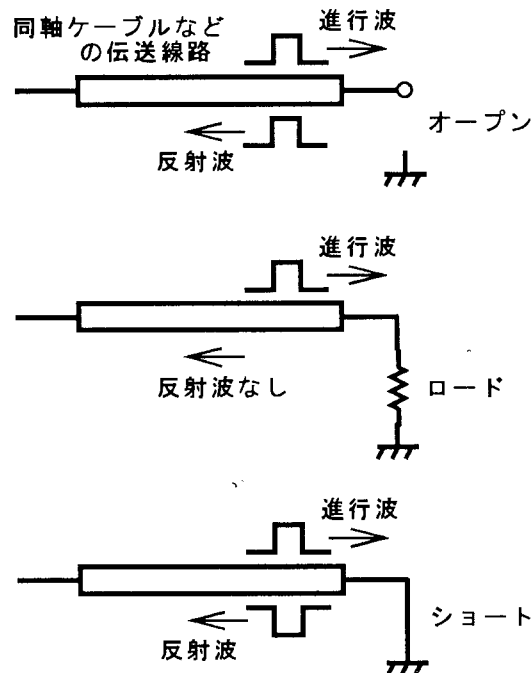
[タ]

ターミネータ terminator：[情報通信工学] [コンピュータネットワーク]，レベル2 SCSI 機器やネットワークケーブルの終端に取り付けるコネクタで、終端抵抗や、単に終端器ともよばれる。ターミネータは、ケーブル内を伝わる信号の反射を防ぐためのもので、

ネットワーク構築時に使用する場合は、ケーブルの特性インピーダンスと同値のインピーダンス（抵抗成分のみ）をもつものを取り付けなければならない（図Ⅲタ-1-1）。なお、ターミネータを取り付けなかった場合や間違って短絡用のキャップを取り付けた場合は、図Ⅲタ-1-2に示すように信号の反射が起こり、ケーブル内に不要な信号が伝搬することとなる。また、ケーブルの特性インピーダンスと異なったインピーダンスのターミネータを取り付けた場合も、同種のことが起こるので注意しなければならない。



図Ⅲタ-1-1 ケーブルとターミネータの関係



図Ⅲタ-1-2 インピーダンス不整合による信号の反射

<ファイル名：タ 0002.HTM >

[チ]

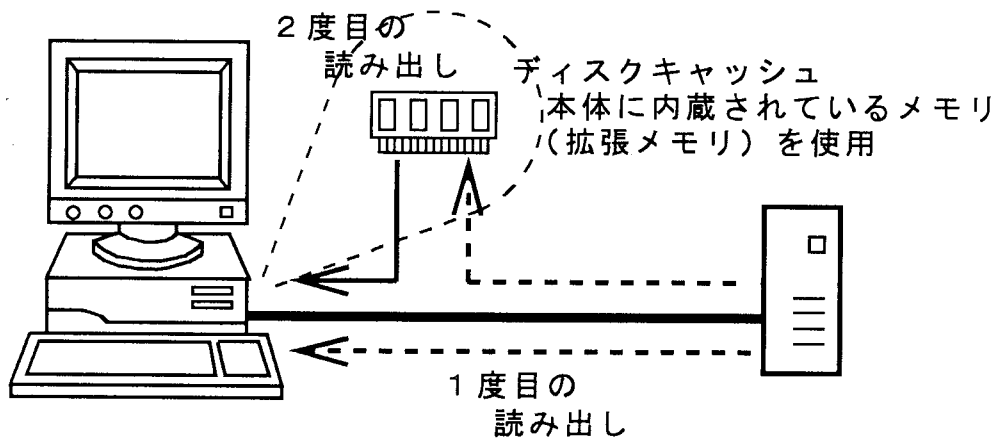
[ツ]

[テ]

ディスクキャッシュ **disk cache** : [情報処理概論] [情報技術概論] [計算機工学] ,

レベル2

主として、ハードディスクなどの機械的な動作を必要とする補助記憶装置の処理速度をカバーするために、メモリの一部をバッファとして確保したもの。同一ファイルの2度目の読み出しの際に動作する (図Ⅲテ-1)。



図Ⅲテ-1 ディスクキャッシュ

<ファイル名 : テ 0001.HTM >

データコンテキストダイアグラム **DCD (Data Context Diagram)** :

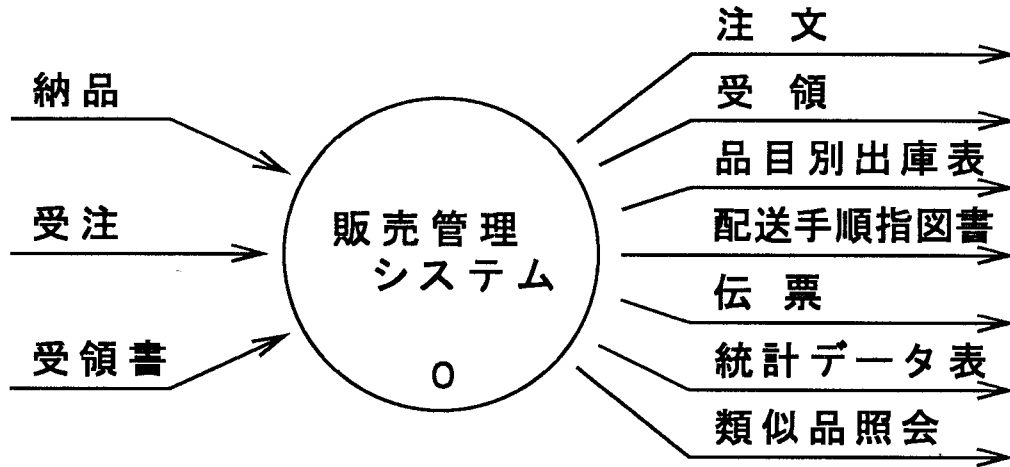
[システム設計] [ソフトウェア開発技法] , レベル1

システム開発で用いられるシステム設計手法 (システム設計図) であり、主として外部設計 (概要設計) の工程で利用される。

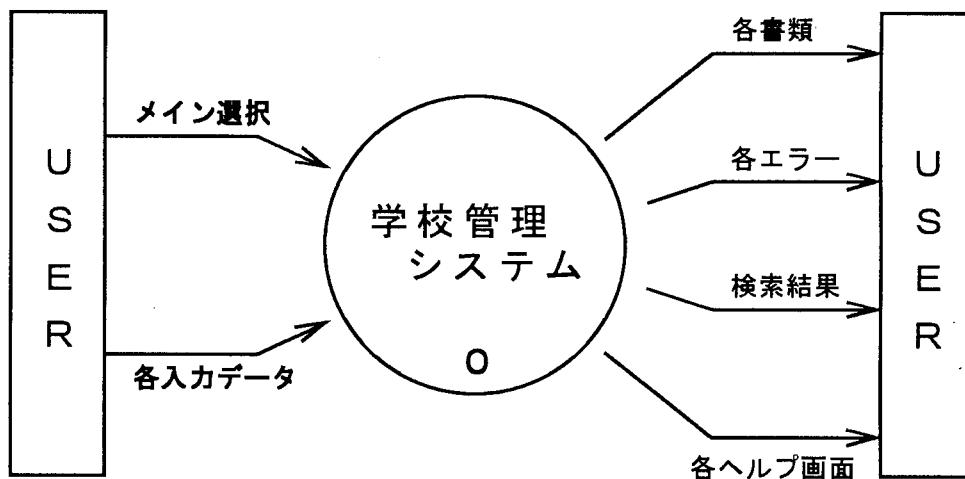
DCDは、従来の処理中心の設計手法ではなく、データ中心の考え、すなわちシステムにどのようなデータが必要であるのかという考えを重視した設計手法である。

DCDを作成する際は、まず、システムで必要となる入出力のデータを定義し、次にその変換工程を決定する。ここでは、入力源から発生した入力データを、それぞれに対応する出力データに変換するための処理を一つのバブル (円) で表現する。すなわち開発するシステムにどのようなデータが必要であり、どのようなデータを出力すればよいのかということを明確にする。このとき、入力データと出力データの整合性を考えることとなる。すなわち、これだけの入力データがあれば、この出力データが得られるという発想である。システム内

部での処理はあまり考えずに、必要となるデータに重点をおく。このようにして作成された図を DCD と呼ぶ。図Ⅲテ-2-1, 2 に DCD の一例を示す。



図Ⅲテ-2-1 DCD の例1



図Ⅲテ-2-2 DCD の例2

<ファイル名：テ 0002.HTM >

データフローダイアグラム DFD (Data Flow Diagram) :



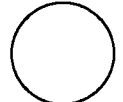

[システム設計] [ソフトウェア開発技法], レベル1

システム開発で用いられるシステム設計手法 (システム設計図) であり、主として外部設計 (概要設計) の工程で利用される。DFD は DCD を詳細化したものにあたる。

外部設計では、開発対象となっているシステムをサブシステムに分割しなければならない。ここで、従来は業務 (処理) を中心に考えてシステムを分割していたが、最近では、情報 (データ) を中心に考えて分割することが多くなってきている。このとき有用となるのが

DFDである。DFDは、表Ⅲテ-3に示すように、たった4つの図式記号（ソース/シンク、データフロー、プロセス、データストア）で表すことができ、そのため、システム開発について詳しい知識を有していないユーザにも簡単に理解できるという利点がある。したがって、分析結果及び新システムの構成をユーザ側や開発スタッフに提示するとき、非常に説明しやすいという利点をもっている。

表Ⅲテ-3 DFDで用いる図記号

記号	名称	意味
	ソース シンク	ソース：データの発生源を示す。 シンク：データの行先（吸収先）を示す。
	データフロー	各処理間のデータの流れを示す。
	プロセス	処理（プロセス）を示す。
	データストア	データが保存（蓄積）される場所を示す。

#### ① ソース/シンク

ソースとはデータの入力源のことであり、システムの処理対象となっている大元のデータの発生源を示す。シンクとはデータの出力先、すなわち最終的なデータの行き先・吸収先のことであり、システムが処理したデータの受け渡し先を示す。

ソース及びシンクは、いずれも長方形又は正方形で示し、その中に対象となる組織名や人名を記入する。

なお、入力源及び出力先が一つずつとは限らないので、ソース/シンク共に複数個存在してもよい。

#### ② データフロー

データフローとはデータの流れのことであり、プロセス間のデータの流れを矢印を用いて示す。それぞれのデータフローには、矢印の近傍に個別のデータ名を記入する。

#### ③ プロセス

プロセスとはデータの処理のことであり、入力データを出力データに変換するための処理を示す。それぞれのプロセスは、円（バブル）を用いて示し、その中に処理概要を記入する。

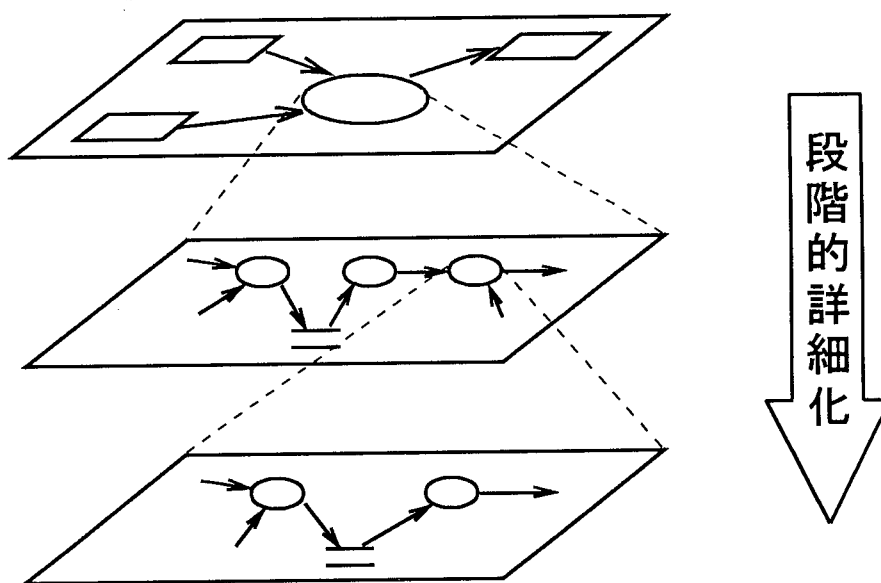
#### ④ データストア

データストアとはデータの保存（蓄積）場所のことであり、具体的にはデータを記憶するファイルを示す。データには、入力された後、処理（変換）されて出力されるといった形態

のものだけではなく、各プロセスが共通で使用するために保存しておくデータや、処理時間にずれが生じた場合に一時的に記憶しておくデータ、すなわち一時記憶データなどがある。これらのデータをデータストアとして表す。データストアは、2本の直線を用いて示し、その間にファイル名を記入する。

#### ⑤ 作成手順

DFDを作成する際、いきなり処理の明確化やそこでのデータ受け渡しを考えてはならない。簡単な部分から複雑な部分へと詳細化していく構造化の概念に従うことが基本となる(図Ⅲテ-3-1)。



図Ⅲテ-3-1 DFDの段階的詳細化

DFDの作成手順を以下に示す。

#### ① ソース/シンクの設定

開発対象となっているシステムにおいて、そのデータの提供元となっている入力源、すなわちソースと、データの最終的な受け手となっている出力先、すなわちシンクを決定する。

#### ② DCDの作成

#### ③ バブルの階層化(詳細化)

DCDのバブルを、幾つかのより小さい機能に分割する。このとき作成されたDFDをレベル0ダイアグラムといい、その中の各バブルはサブシステムに対応する。更に、レベル0ダイアグラムの各バブルを、より小さい機能に分割したものをレベル1ダイアグラムと呼ぶ。このように、段階的詳細化技法を用いて次のレベルのDFDを作成していく。

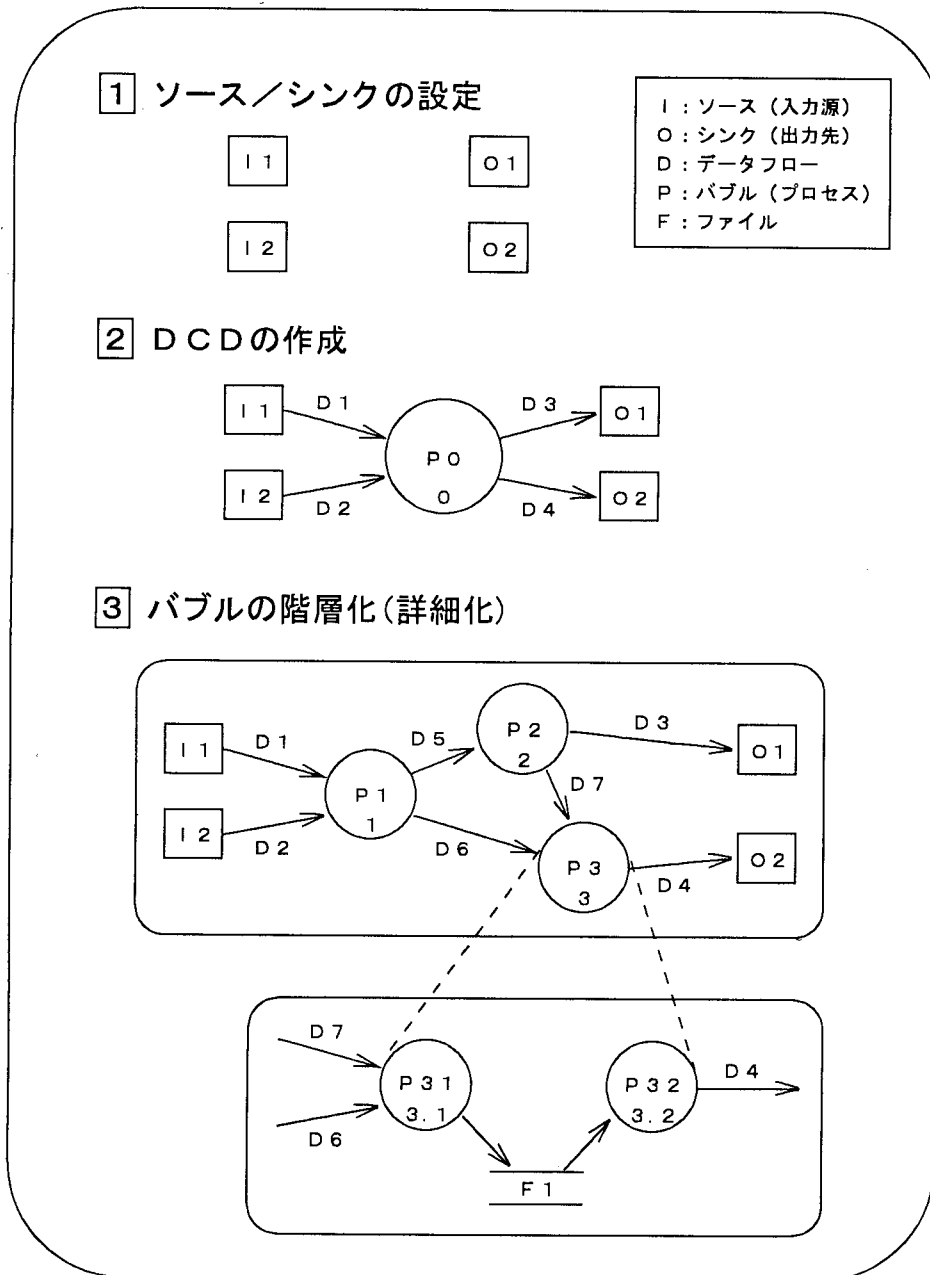
このとき、レベル1のDFDは、レベル0のDFDの主要プロセスを、そのプロセス



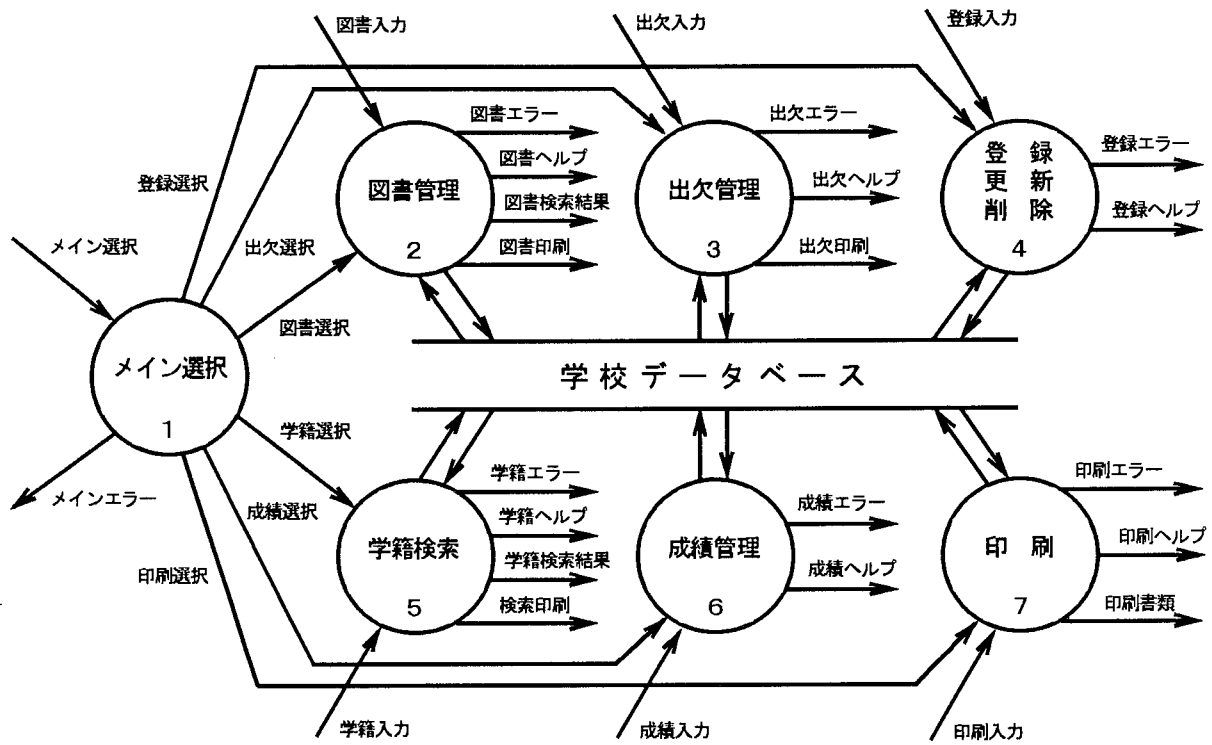
を構成する詳細ないくつかの子プロセスに分割したのになっている。また、データフローについても、レベル0のDFDから基本的に引き継ぐこととなる。

同様に、順次下位レベルのDFDを記述していき、プロセスへの入出力データフローが項目単位で表され、プロセスが単一機能となるまでレベルを下げていく。すなわち、最下位レベルのDFDのバブルは一つのプログラムを示すこととなる。なお、外部設計（概要設計）の工程においては、サブシステムまでの分割でとめておき、内部設計（詳細設計）の工程でより詳細なプログラム分割を行う。

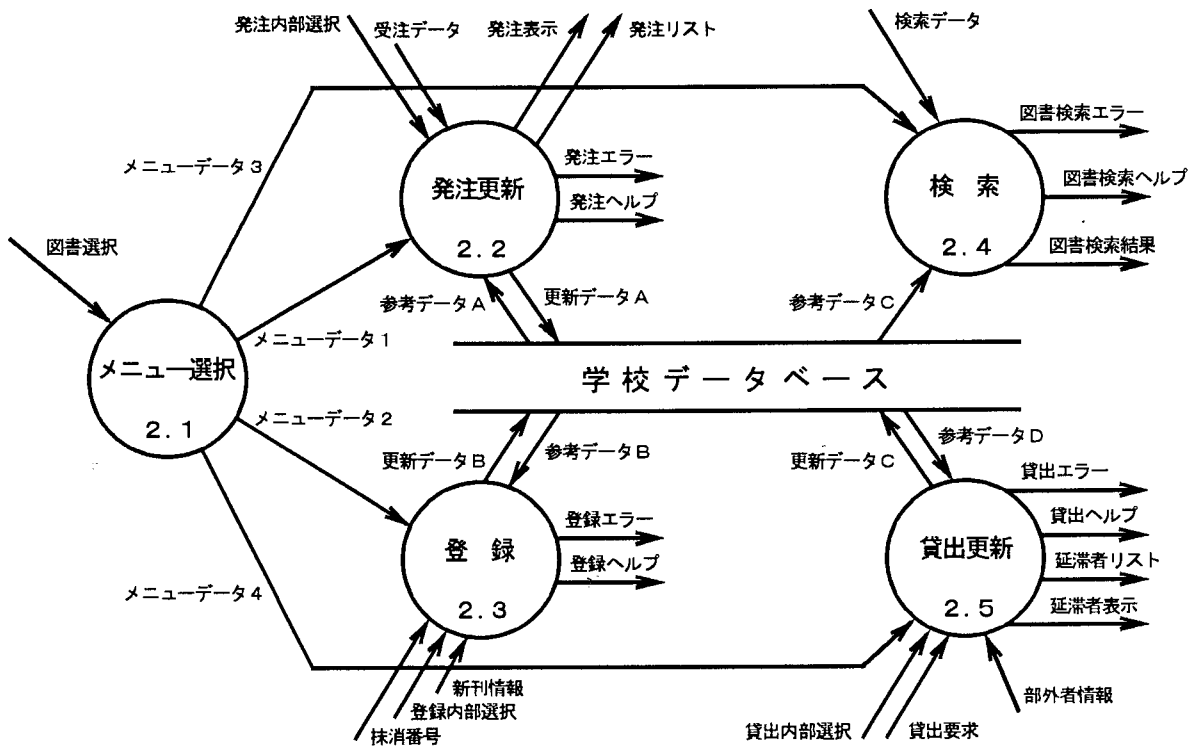
以上、①～③によりDFDが詳細化され、開発すべきシステムが明確化されていく（図Ⅲテ-3-2）。なお、DFDの一例を図Ⅲテ-3-3、4に示す。



図Ⅲテ-3-2 DFDの作成手順



図Ⅲテ-3-3 DFDの例1 (レベル0ダイアグラム)



図Ⅲテ-3-4 DFDの例2 (レベル1ダイアグラム)

<ファイル名:テ 0003.HTM >

**テンポラリファイル temporary file** : [ソフトウェア工学] , レベル1

システムやアプリケーションソフトが動作するときに、ハードディスクや RAM ディスク上に一時的に作成されるファイル。処理速度を向上させるためには、機械的な動作を必要としない RAM ディスクを用いた方が効果的であるが、作業している途中でハングアップ若しくは停電などにみまわれると、元の状態に戻すことが困難であることから、通常はハードディスク上にテンポラリファイルを作成することが多い。また、最近のハードディスクは、以前と比べると非常に高速化されており、また、ディスクキャッシュを併用することで実際の処理速度を更に向上させることもできる。このことから、ハードディスク上にテンポラリファイルを作成することが多くなってきている。

システムとして MS-DOS を使用している場合、テンポラリファイルの作成に当たっては、特別な処理が必要な訳ではなく、環境設定ファイル CONFIG.SYS に、

**SET TEMP=[ドライブ名:][ディレクトリ名]**

又は、

**SET TMP=[ドライブ名:][ディレクトリ名]**

と記述するだけで、汎用的なテンポラリファイルとして用いることができる。なお、グラフィックソフトや最近のワープロソフトなど、比較的大きなデータを扱うアプリケーションソフトでは、そのソフトの中でテンポラリファイル（作業領域や編集領域と表現されている場合が多い）の設定を行う必要がある。

テンポラリファイルは、いずれの場合にせよ、プログラム終了時に自動的に削除されることが多い。

<ファイル名 : テ 0004.HTM >

[ト]

## 5. ナ行

[ナ]

[ニ]

[ヌ]

[ネ]

**ネットワークインタフェースカード NIC (Network Interface Card)** :

[情報通信工学] [コンピュータネットワーク] , レベル2

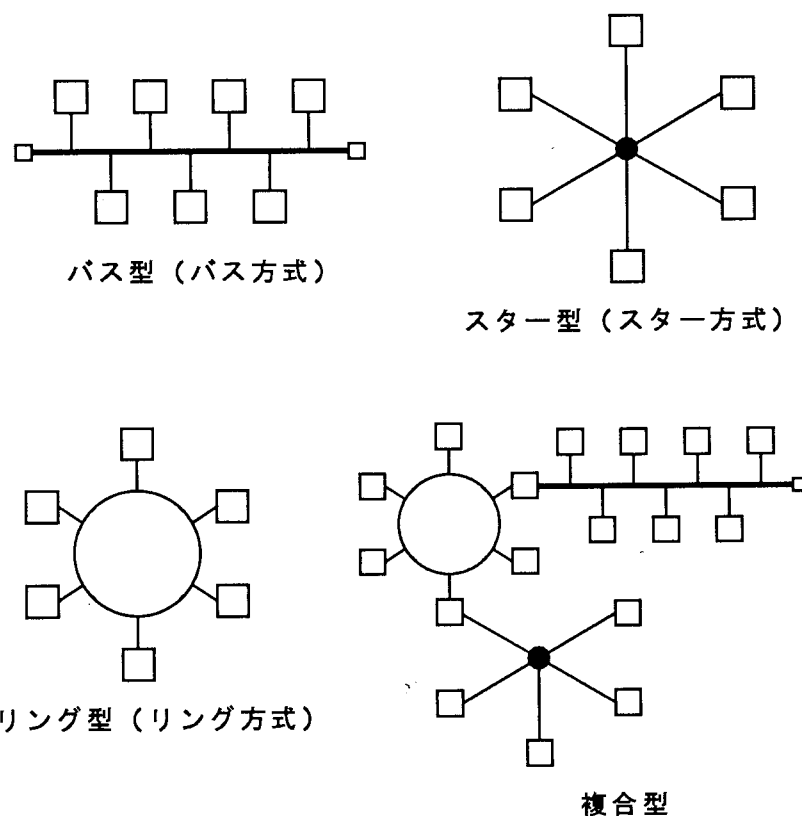
コンピュータ間を接続する際に用いるインタフェースカード、すなわちコンピュータネットワークで使用するインタフェースカード。LANアダプタやLANボード、また単に、ネットワークカードと呼ばれることもある。通常、1台のコンピュータにつき1枚あればよい。

なお、NICには、接続法の違いから10Base-2用、10Base-5用、10Base-T用などがあり、それぞれコネクタの形状が異なっている。また、汎用的な考えから、異なる2つ以上のコネクタを装備したものもある。

<ファイル名：ネ 0001.HTM >

ネットワーク形態 **network topology** : [情報通信工学] [コンピュータネットワーク] ,  
レベル2

コンピュータネットワークにおける物理的な接続形態。バス型 (バス方式)、スター型 (スター方式)、リング型 (リング方式) 及びその複合型があり (図Ⅲネ-2)、その接続法に関しても 10Base-2、10Base-5、10Base-T など数種のものがある。また、最近ではケーブルを用いない無線形態のネットワークも普及してきている。通常、これらはネットワークの規模や導入コストによって選択される。



図Ⅲネ-2 様々なネットワーク形態

<ファイル名：ネ 0002.HTM >

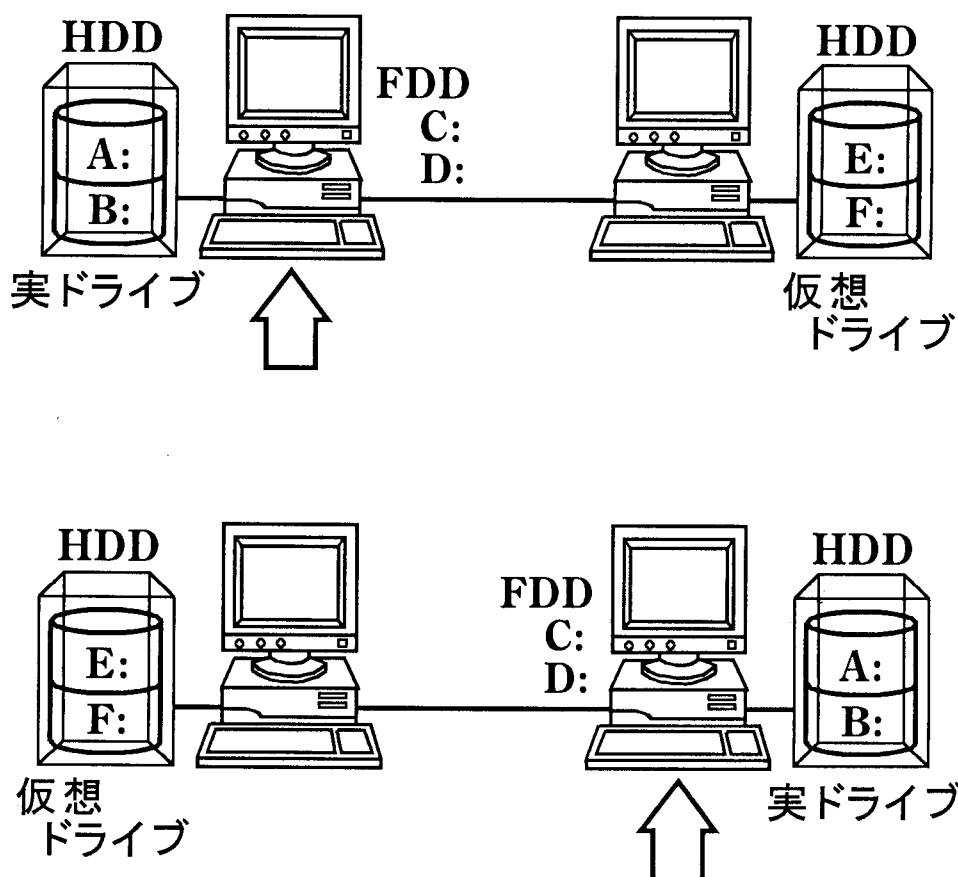
ネットワークケーブル **network cable** : [情報通信工学] [コンピュータネットワーク],  
レベル2

コンピュータ間を接続する際に用いるケーブル、すなわちコンピュータネットワークで使用するケーブル。コンピュータネットワークにおける接続法の違いから、シックイーサネット同軸ケーブル（イエローケーブル）やシンイーサネット同軸ケーブル、ツイストペアケーブルなど数種のもが用いられ、これらはそれぞれ物理的形狀や接続法が異なり、また、接続時に保証されている最大長も異なる。

<ファイル名：ネ 0003.HTM >

ネットワークドライブ **network drive** : [情報通信工学] [コンピュータネットワーク],  
レベル2

ネットワーク OS やネットワークユーティリティの機能によって、他のコンピュータ上につくられた仮想的なドライブ。ユーザは、ネットワーク上に接続された他のコンピュータのハードディスクなどを、実ドライブと同様に扱うことができる（図Ⅲネ-4）。



図Ⅲネ-4 実ドライブとネットワークドライブ

図に示したネットワークでは、実ドライブが A: と B: (ハードディスク) 及び C: と D: (フロッピーディスクドライブ) であり、仮想ドライブ (ネットワークドライブ) として他のコンピュータのハードディスクが E: と F: に割り当てられている。ユーザは、特に意識することなくネットワークドライブである E: と F: を使用することができる。すなわち、ファイルの読み込みや書き込みが自由にできるわけである。ただし、実ドライブと同様、対象となるネットワークドライブの容量による制限はあり、また、ネットワーク上のデータ転送速度による問題やトラフィックによる問題は避けられない。また、当然のことであるが、他のコンピュータの電源が入っていない場合、ネットワークドライブは使用できない。

<ファイル名 : ネットワークドライブ名>

[ノ]

## 6. ハ行

[ハ]

バス型 LAN (バス方式 LAN) : [情報通信工学] [コンピュータネットワーク],

レベル 2

1本のケーブル (実際は、複数本のケーブルを直列接続している場合が多い。) を中心に、コンピュータを直線的に接続した形態の LAN。中心となるケーブルは幹線と呼ばれ、そこから複数の支線を引き出すことで接続するコンピュータを増加させる (図 III ハ-1)。ただし、10Base-5 による接続の場合は、図に示した形態となるが、10Base-2 による接続の場合は、支線を用いず、T型コネクタを介して、直接、幹線に接続する場合が多い。

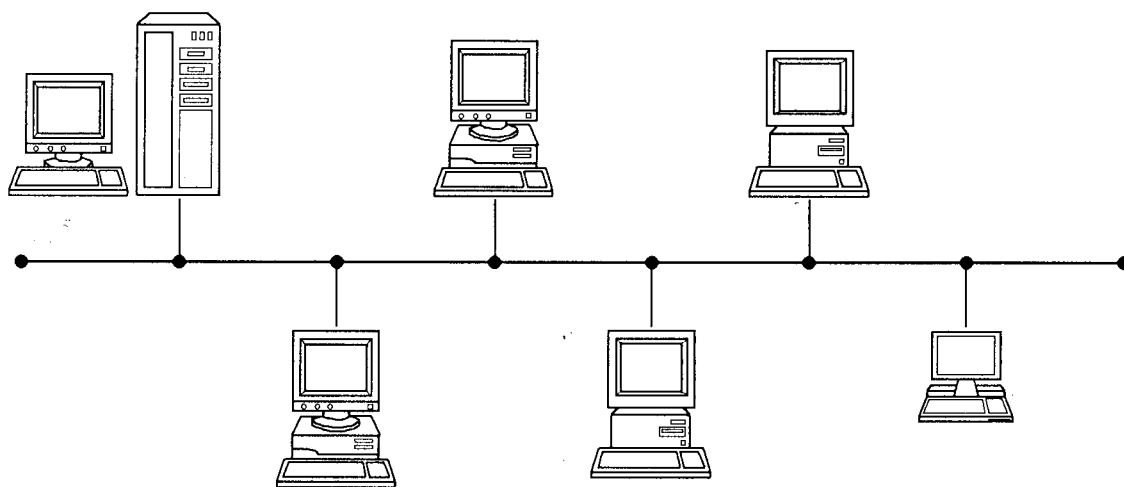


図 III ハ-1 バス型 LAN

イーサネットはバス型 LAN の代表格であり、パソコン LAN を含むありとあらゆるネットワークで採用されている。

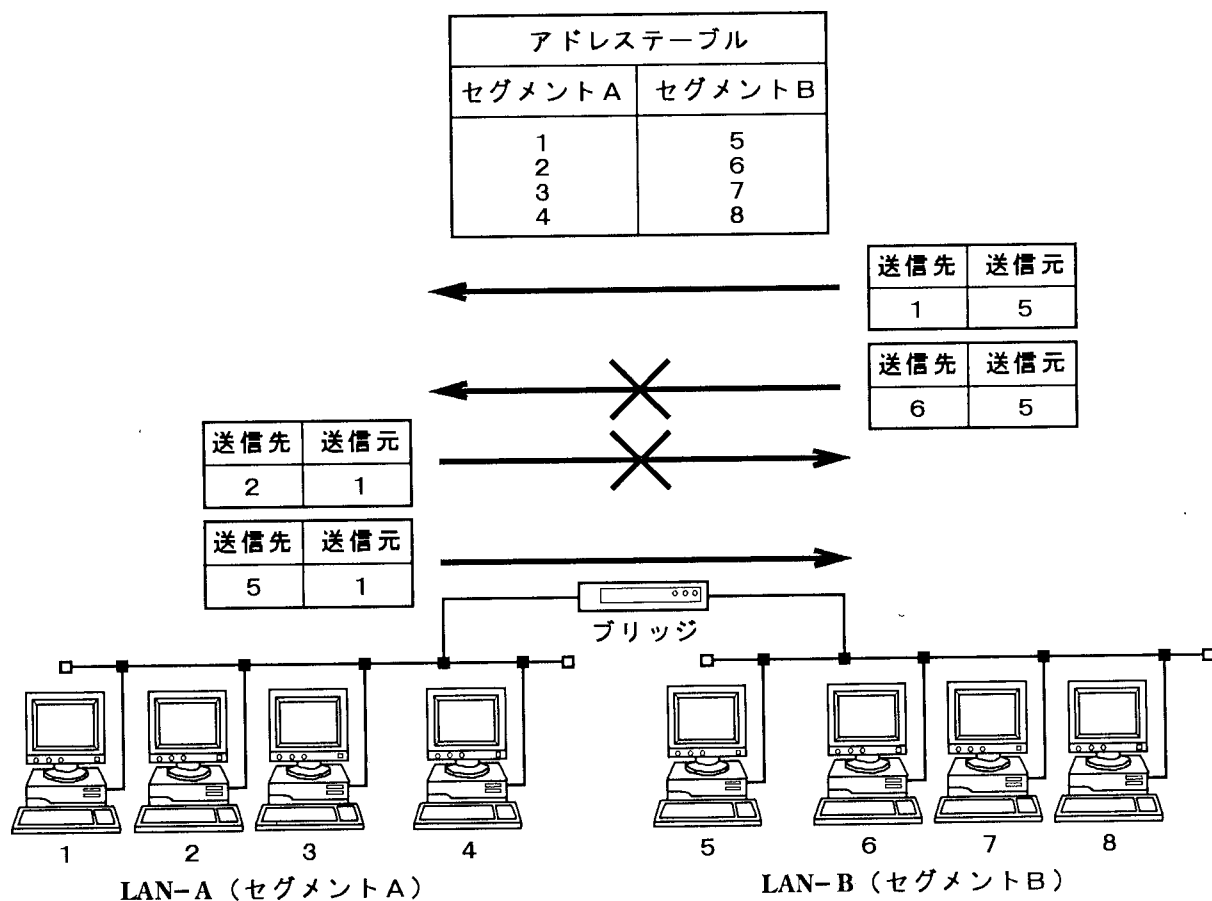
<ファイル名：ハ 0001.HTM >

[ヒ]

[フ]

ブリッジ bridge：[情報通信工学] [コンピュータネットワーク]，レベル3

複数の LAN を相互接続するための装置。パケットが送られてきたときにアドレス一覧（アドレステーブル）と照合し、外部の LAN に転送先があればそちらにパケットを送り（フォワーディング）、転送先がなければパケットを破棄する（フィルタリング）。すなわち、パケットの送信元と送信先がブリッジからみて同一セグメント内に位置するときは、他のセグメントにパケットを通過させずに破棄する機能をもつ。



図Ⅲフ-1 ブリッジを用いた LAN の接続例

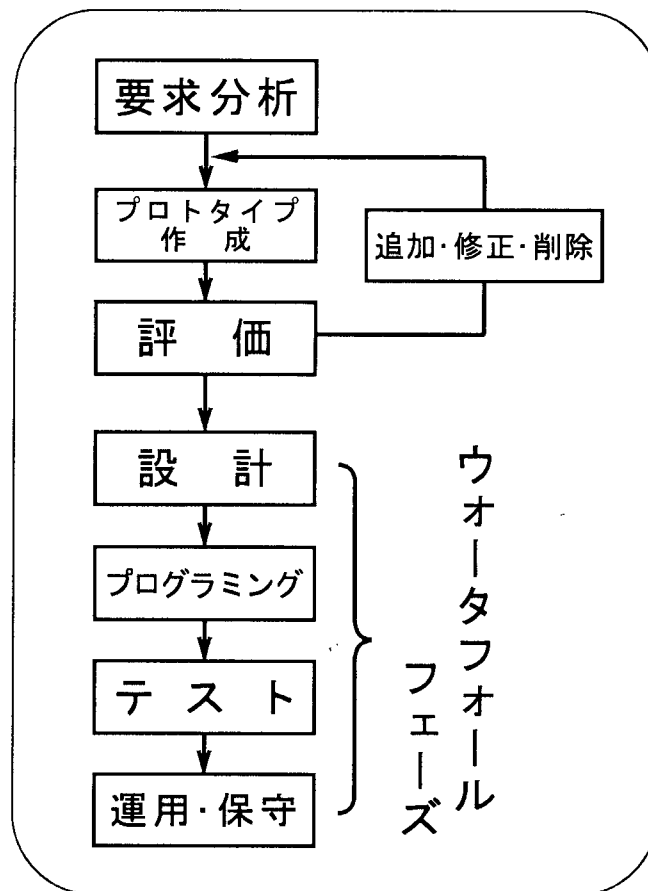
図Ⅲフ-1 にブリッジを用いた LAN の接続例とそのときのパケットの流れを示す。ブリッジは、リピータと同様、ケーブル内を伝送している信号を増幅・再生する機能を持ち、更にパケット識別機能も備えている。したがって、不要なパケットの流出を防ぐことができるので、ネットワーク全体のトラフィック低減やセキュリティ向上に貢献する。

<ファイル名：フ 0001.HTM >

プロトタイプモデル **prototype model** : [システム設計] [ソフトウェア開発技法] ,

レベル 1

開発の後半にならないとシステムの全容が分からないウォータフォールモデルの欠点を改善するために提案されたのが、プロトタイプモデルである（図Ⅲフ-2）。このモデルでは、システム開発の初期の段階から試作品を作成することにより、ユーザ側と開発者側との意見の食い違いを吸収し、設計工程以降に大幅な仕様変更が発生するのを未然に防ぐことができる。したがって、ウォータフォールモデルよりも、よりユーザの要求しているシステムを実現することができるという長所をもっている。また、試作品を作成しているという利点から、システム開発の初期段階でのミスの発見確率も向上し、結果として開発期間の短縮、コストの低減などが図れる。



図Ⅲフ-2 プロトタイプモデルによる開発工程



特に大きな問題が発生しない限り、プロトタイプモデルはウォーターフォールモデルよりも、システム開発全般にわたって優れた点を多く備えている。しかしながら、プロトタイプモデルでは、ユーザが納得する試作品が完成しない限り、次の工程に進めないという重大な欠点がある。すなわち、ユーザ側と開発者側の意見がくい違った場合は、初期段階から開発がストップしてしまうわけである。問題を回避しようと、試作品の作成に多大な労力を注ぎ込むことは、本来プロトタイプモデルがもっている長所が、逆に欠点となってしまう。全てのシステムがプロトタイプモデルで効率よく開発できるのではなく、対象となるシステムがプロトタイプモデルに適しているかどうかを見極めることが重要となる。

なお、プロトタイプモデルでは、試作品が作成された後はそれをもとに要求仕様書やインターフェース仕様書などが作成され、以降の工程はウォーターフォールモデルと同様に進められる。

<ファイル名 : フ 0002.HTM >

[へ]

[ホ]

## 7. マ行

[マ]

[ミ]

[ム]

[メ]

[モ]

## 8. ヤ行

[ヤ]

[ユ]

[ヨ]

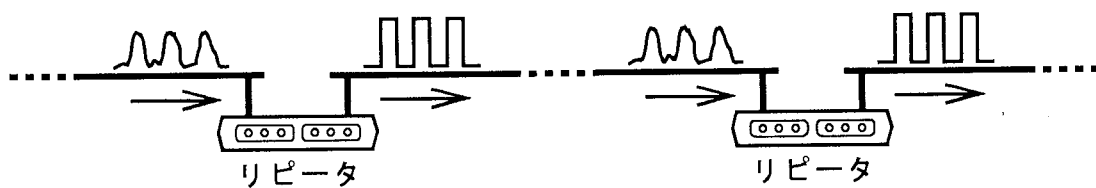
## 9. ラ行

[ラ]

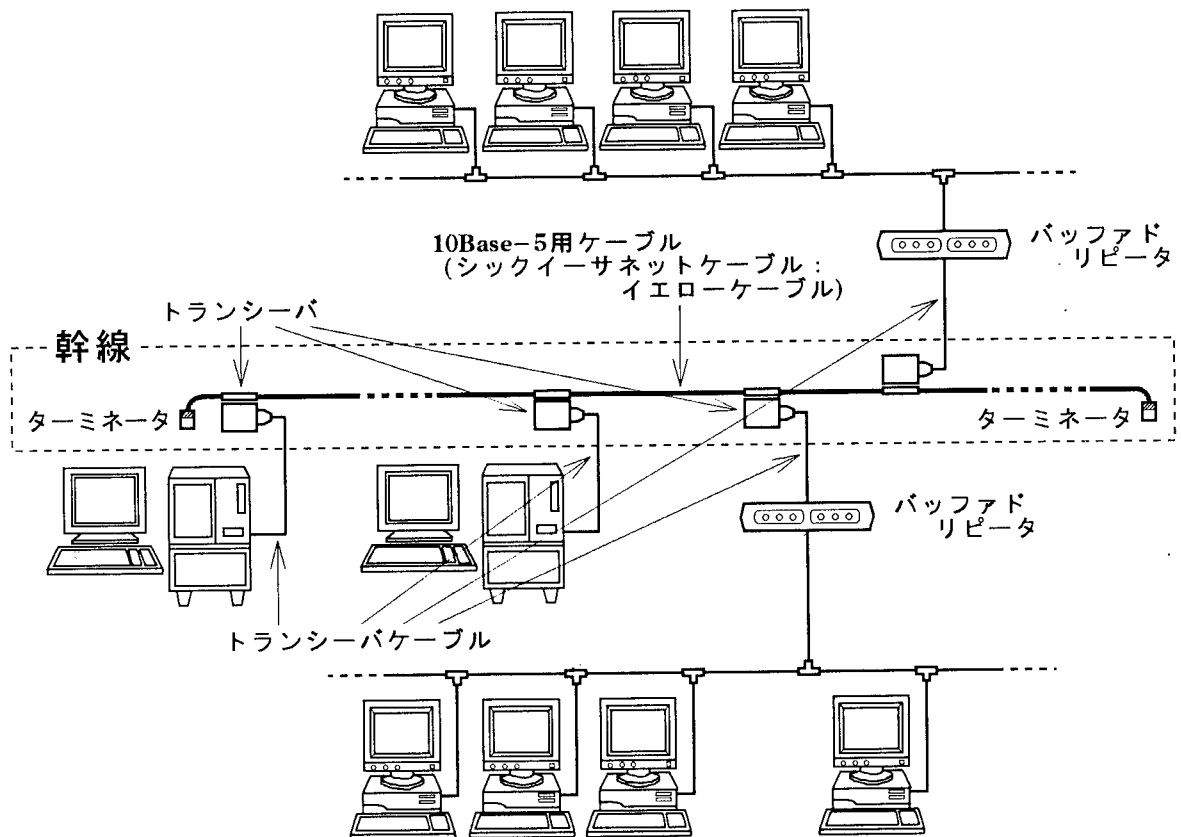
[リ]

リピータ repeater : [情報通信工学] [コンピュータネットワーク], レベル3

LANにおいて、ケーブル内を伝送している信号を増幅・再生して、伝送距離を延ばすための装置。例えば、10Base-5による接続では、リピータを用いることによりケーブルの最大長500mを越えた接続が可能となる。リピータは、単に信号を増幅・再生するための装置(図Ⅲリ-1-1)なので、ブリッジやルータのようなパケット識別機能は基本的にもたない。



図Ⅲリ-1-1 リピータを用いたネットワークの延長



図Ⅲリ-1-2 バッファドリピータを用いたネットワークの構成例

リピータには、大きく分けてビットリピータとバッファドリピータがあり、前者は単に信号の増幅・再生を行うためのもので、接続されるセグメントにおけるデータ転送速度は同一でなければならない。それに対して後者は、バッファ（データ蓄積）機能を有するので、セグメント間のデータ転送速度が異なってもよいという利点がある。なお、ビットリピータは、その名のとおりビットごとに増幅・再生を行っており、バッファドリピータはパケットごとに増幅・再生を行っている。また、ビットリピータは、異常なパケットが流れてきた場合、それを通過させてしまうが、バッファドリピータは、送られてきたパケットを内部のバッファに一時的に蓄え、そこでパケットのチェックを行った後に、正常なパケットのみを転送する機能をもつ。

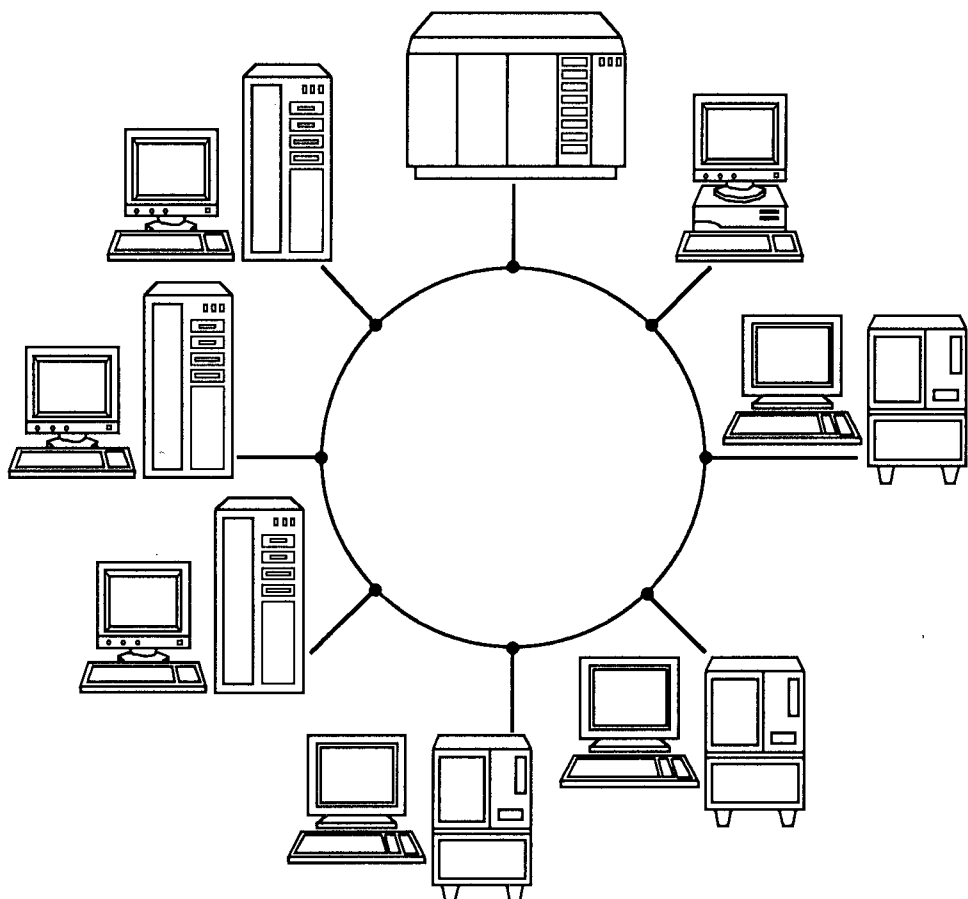
リピータを用いなければならないような LAN は、比較的大規模なもの（室内だけで LAN 接続しているのではなく、構内や学内で LAN を構築している）が多く、通常、幹線からトランシーバなどを介してバッファドリピータに接続される（図Ⅲリ-1-2）。

<ファイル名：リ 0001.HTM >

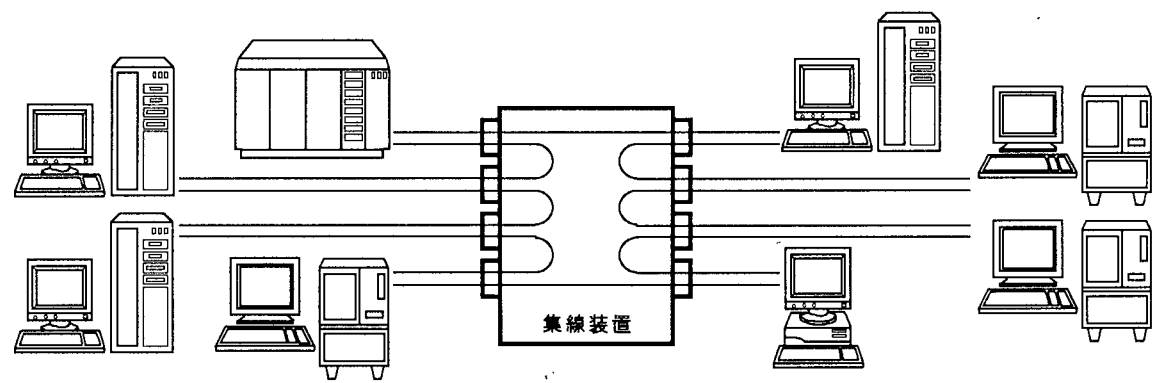
リング型 LAN（リング方式 LAN）：〔情報通信工学〕〔コンピュータネットワーク〕，  
レベル 2

リング状に構成したケーブルを中心として、その周りにコンピュータを接続した形態の LAN（図Ⅲリ-2-1）。ただし、実際は、1本のケーブルでリングを構成しているのではなく、複数本のケーブルとそのケーブルをリング状に接続するための集線装置で構成されている。バス型 LAN が終端をもつものに対して、図に示すようにリング型 LAN は終端をもたない。

トークンリングはリング型 LAN の代表格であり、大型汎用機やワークステーションを中心とした高速通信が必要となるネットワークで採用されている。なお、リング型 LAN では、その高速性を重視するために、伝送媒体として光ファイバケーブルを用いることが多い。なお、このときは、光集線装置を用いて図Ⅲリ-2-2に示すような接続形態をとる。



図Ⅲリ-2-1 リング型 LAN



図Ⅲリ-2-2 実際の接続例

<ファイル名：リ 0002.HTM >

[ル]

[レ]

[ロ]

10. ワ行

[ワ]