APPENDIX-1

COMPUTER SYSTEM CONFIGURATION

APPENDIX-1
COMPUTER SYSTEM CONFIGURATION

1.1 Hardware and Software

(1) Computer system appearance

Computers installed at computer centers of business firms, universities and governmental institutions are used for a variety of purposes. These range form clerical work calculations to scientific and technological calculations. Normally, a computer is installed in an air-conditioned special room. Under-floor cables connect these devices. The devices are run by operators and other specially-assigned persons.

(2) Outline of devices of a computer system

The computer system consists of the devices shown in Fig.

These devices roughly consist of the central processing unit (CPU), storage unit and input/output (I/O) device. The storage unit has two parts. One is the main memory unit that can read and write data at a high speed. The other part is an auxiliary memory unit in which memory capacity is considered more important, rather than read/write speed.

An I/O device is a collective name of input device and output device.

The central processing unit and main memory unit are collectively called main frame. The auxiliary memory unit and I/O device are collectively called peripheral devices.

(3) Hardware and software

In a computer system, hardware and software function as a unified system. Hardware means the equipment. Software is a collective name for the functions, procedure and technique for effective operation of the hardware.

In software, the procedure that designates the processing and sequence of the hardware operation is called a programme. A person who creates a programme is called a programmer.

1.2 Concept of Programme (Fig. A.1.2)

(1) What is a programme?

To process data with a computer, the computer must be given instructions directing the job processing procedure. A set or a series of these instructions is called a programme.

A programme may be interpreted as a series of instructions necessary to execute a series of jobs in the sequence of processing.

---

[1] This section is based on the JICA training material "Computer Introduction; Training for Information Processing" OKINAWA International Center, Japan International Cooperation Agency.

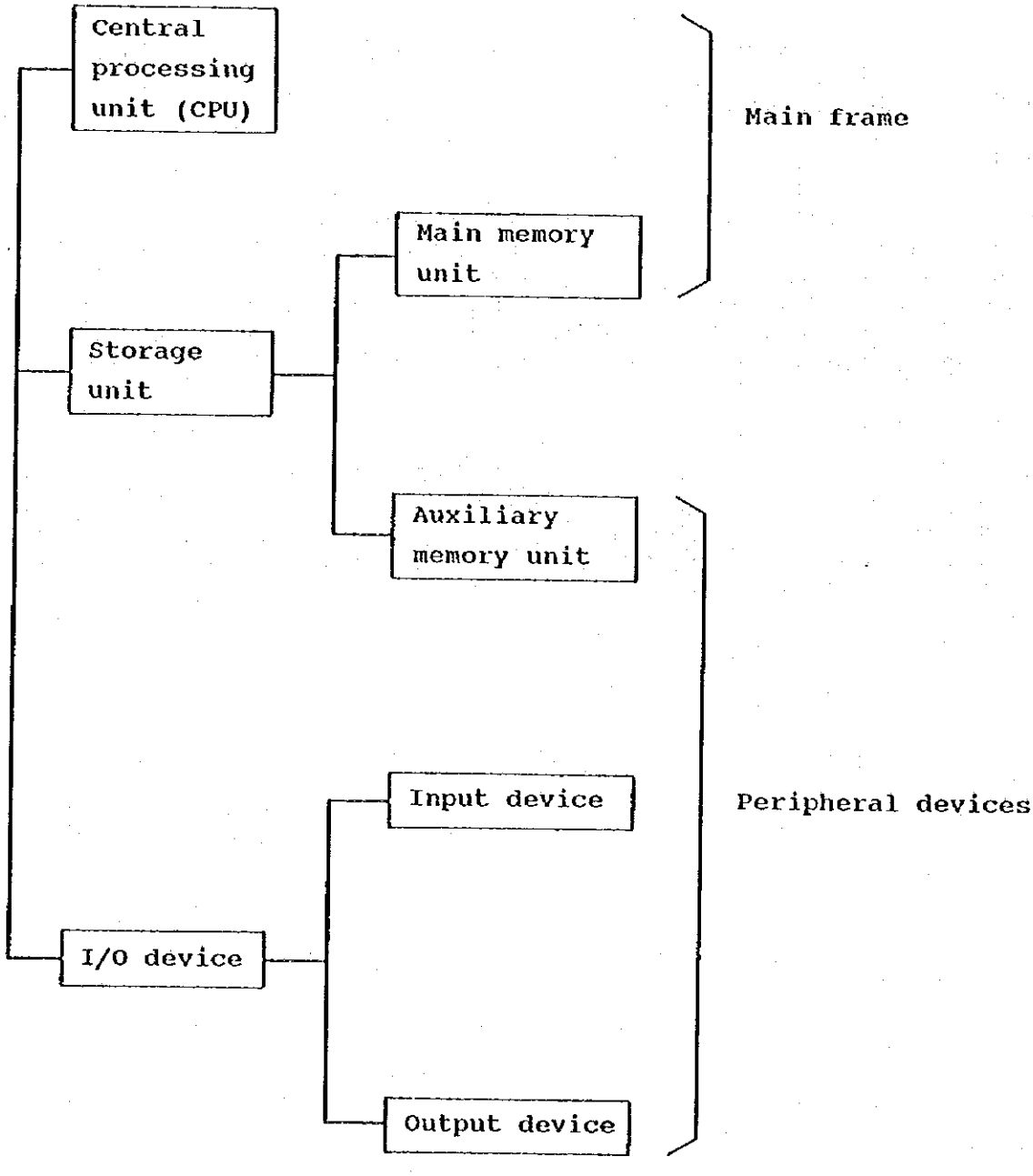

Fig. A.1.1 COMPUTER SYSTEM DEVICES

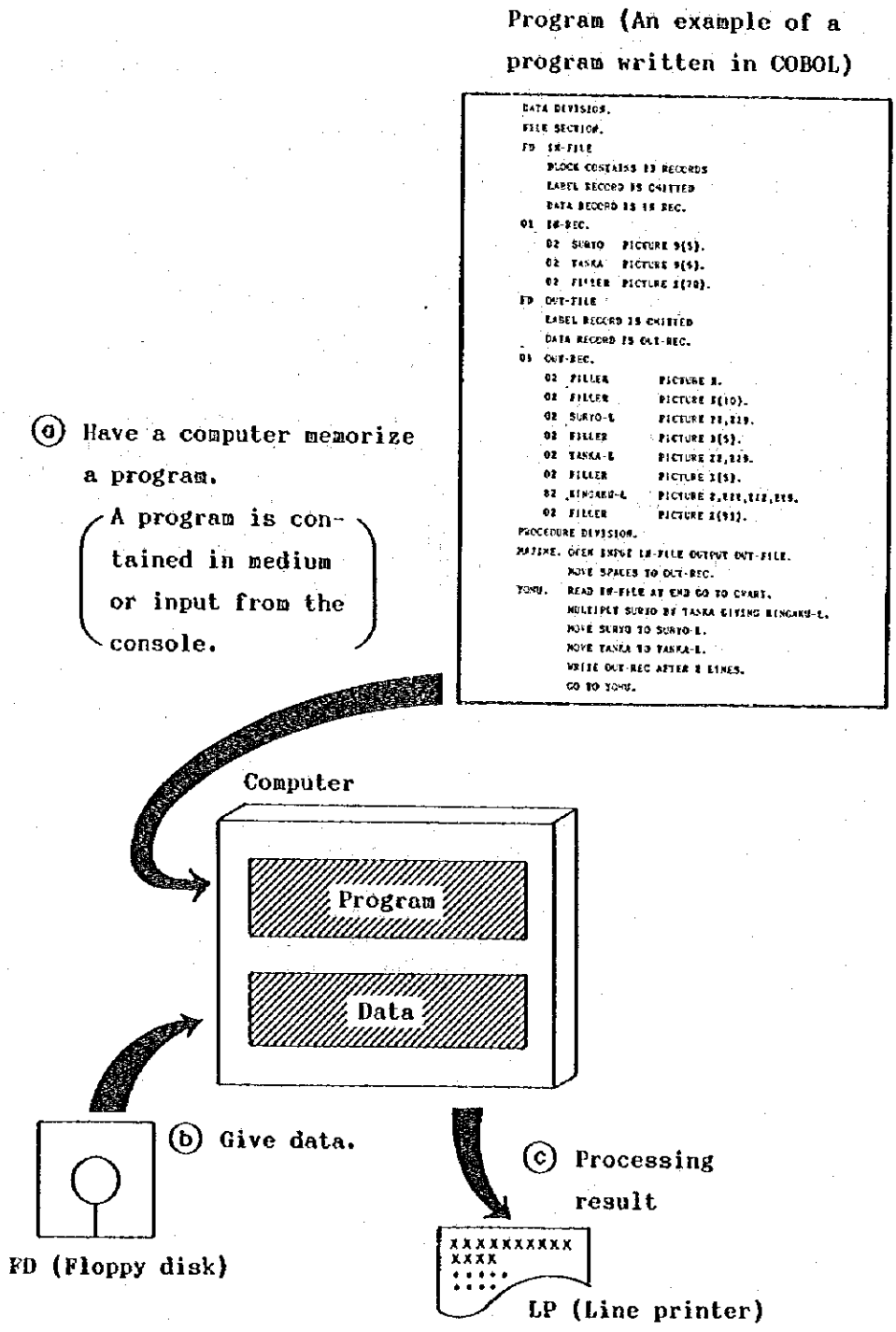

Fig. A.1.2 DATA PROCESSING

(2) Programme storing method

A computer is similar to a puppet. A computer must be manipulated by a person, but a programme can take the place of a man. In other words, a programme resembles the scenario of a puppet-show. The programme is a contrivance to manipulate the puppet's arms and legs. It makes the show progress as written in the scenario.

Therefore, once a computer memorizes a programme and the data to process is given to it, the computer automatically executes the instructions, step by step, in the sequence described in the programme.

To have a computer memorize a programme and execute jobs is called Stored Programme System.

1.3 Configuration and Functions of Hardware

(1) Hardware configuration

Refer to Fig. A.1.3. It shows the devices that configure a computer system, and their functions.

a. Central processing unit (CPU)
The CPU consists of a control part and an operation part. The CPU controls the entire computer system and its operation functions.

b. Storage unit
The storage unit consists of a main memory and an auxiliary memory. It stores programmes and data.

c. Channel
The channel controls data transfer between the main memory and I/O device, auxiliary memory, and communication control processor.

d. Communication control processor
This controls data transfer between remote terminals and the computer.

e. Input device
This inputs data to the computer.

f. Output device
This preserves a large quantity of results processed by the computer. It then outputs these result to visible media such as paper and display.

1.4 Off-line and On-line

(1) Off-line processing

In off-line processing, telecommunication lines are not used to transmit processing data to a computer. Instead, the processing data is contained in slips, data cards or magnetic tape. They are manually transferred to a computer location.

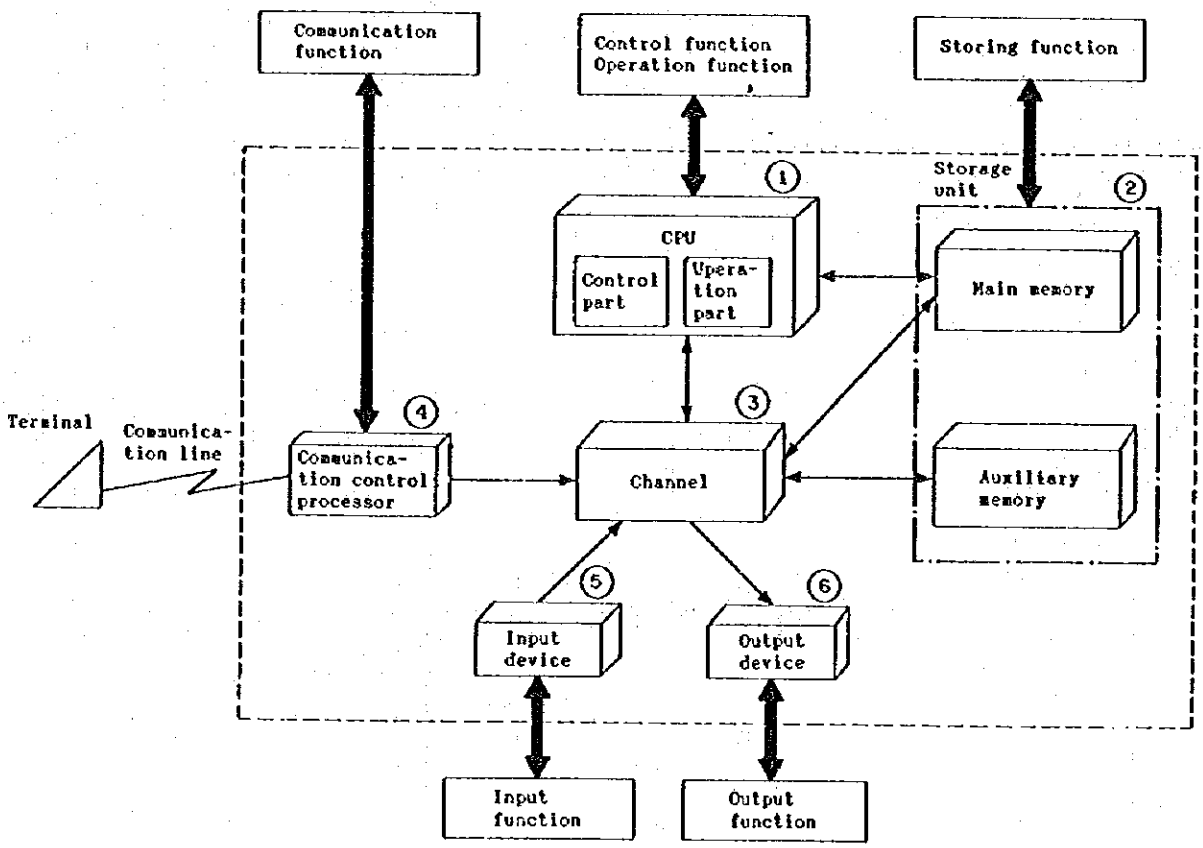

Fig. A.1.3 HARDWARE BASIC CONFIGURATION AND FUNCTIONS

(2) On-line processing

In on-line processing, a computer and its terminals are connected with telecommunications lines. Processing information is then directly transmitted to the computer through the telecommunications lines. As shown in Fig. A.1.4, the information is output to the computer in the form of electrical signals through the telecommunications lines.

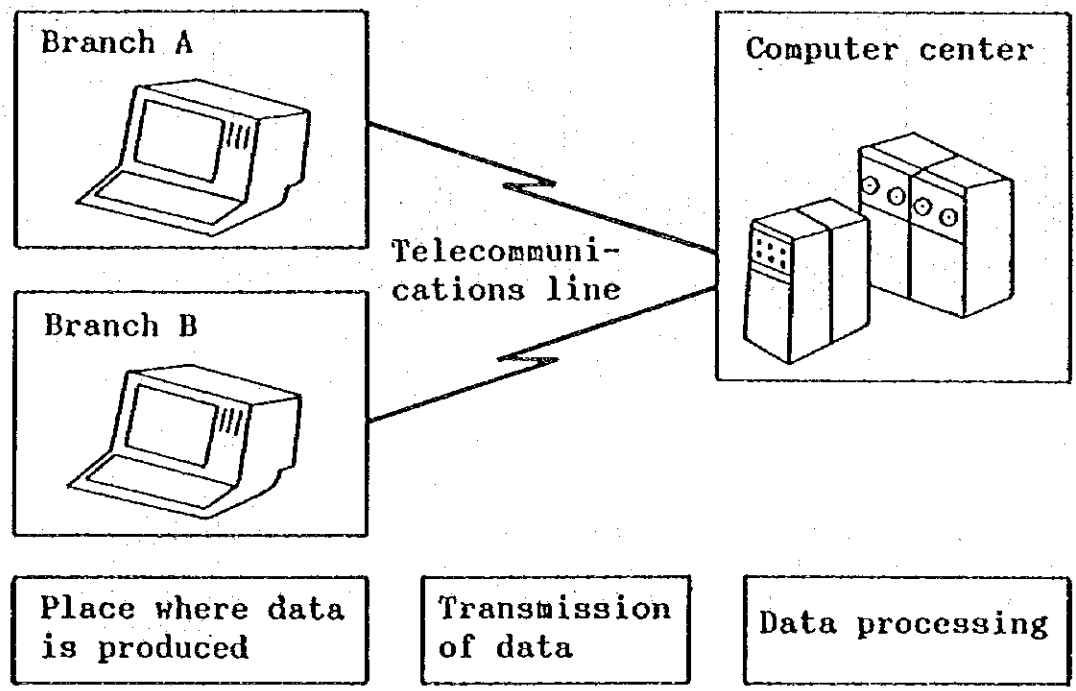

Fig. A.1.4 On-Line Real Time Processing

1.5 Batch Processing

In batch processing, information is collated to a certain unit. In this type of processing, many users request a variety of jobs for data processing. Priority is therefore assigned to increasing computer use efficiency and work volume. Responsiveness to users takes a lower priority. This means that notification of processed results to users would take longer than on-line processing. On a large job, the user may have to wait for a day or longer to receive the processed result.

There is also a type of processing in which data for batch processing is input/output with a remote terminal through a telecommunications line (that is, on-line processing). This is called remote batch processing. In this system, since data is input-output through a telecommunications line, manpower is not required for the transmission.

1.6 Real-Time Processing

In real-time processing, data sent from each terminal is immediately processed and the processed result is sent back to the pertinent terminal, as shown in Fig. A.1.5.

There are two types of real-time processing. One is on-line real time processing on which predetermined standard processing is conducted within a certain time (normally within several seconds). The other is time sharing processing in which non-predetermined and non-standard processing are conducted in conversation form with the users' terminals.

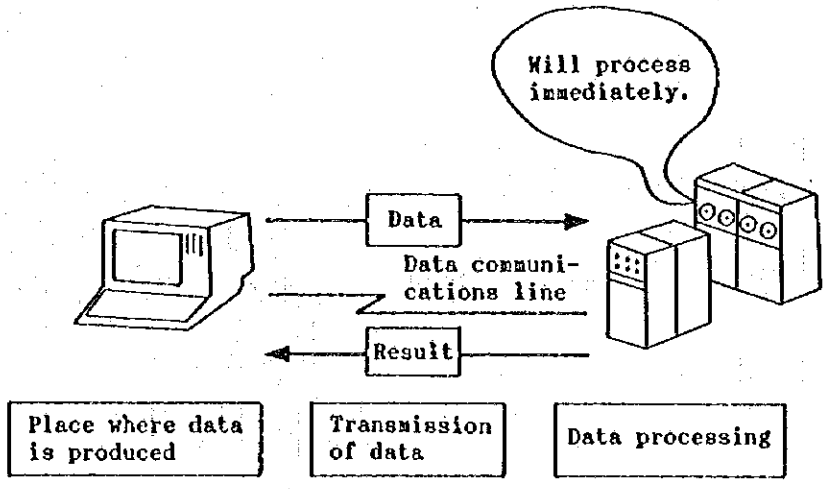

Fig. A.1.5 REAL-TIME PROCESSING

(1) On-line real time processing

This type of processing is now most popular. It is applied to various data communications systems, such as cash dispensers (Fig. A.1.6), airline reservations, security transactions, etc.

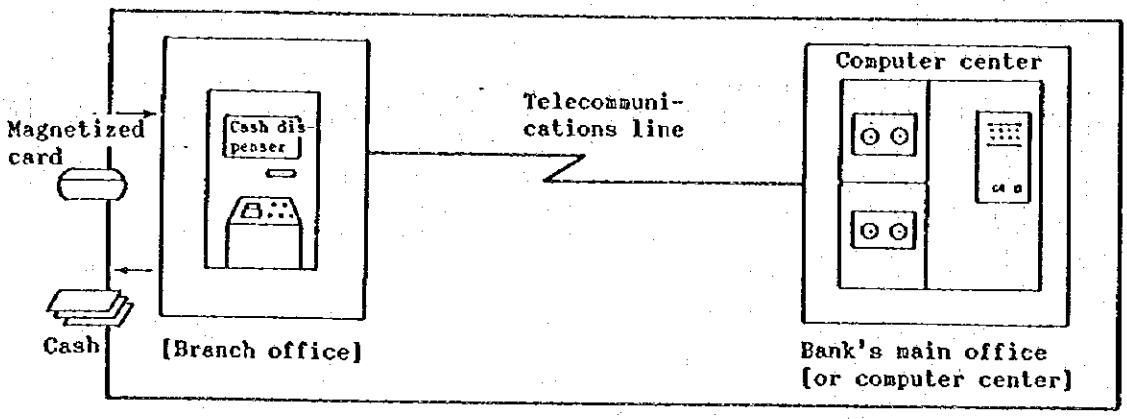

Fig. A.1.6 On-Line Real Time Processing

(2) Time sharing processing

Time sharing processing is not standardized. Therefore, such activities as data input from a terminal, data processing by a computer, output of the result to the terminal, and cogitation by the terminal operator, are conducted intermittently between the user and the computer. Since a large number of users share the CPU's operation in this type of processing, the operation is controlled in such a manner that each user can proceed as if he had a monopoly on the use of the computer. (Fig. A.1.7)

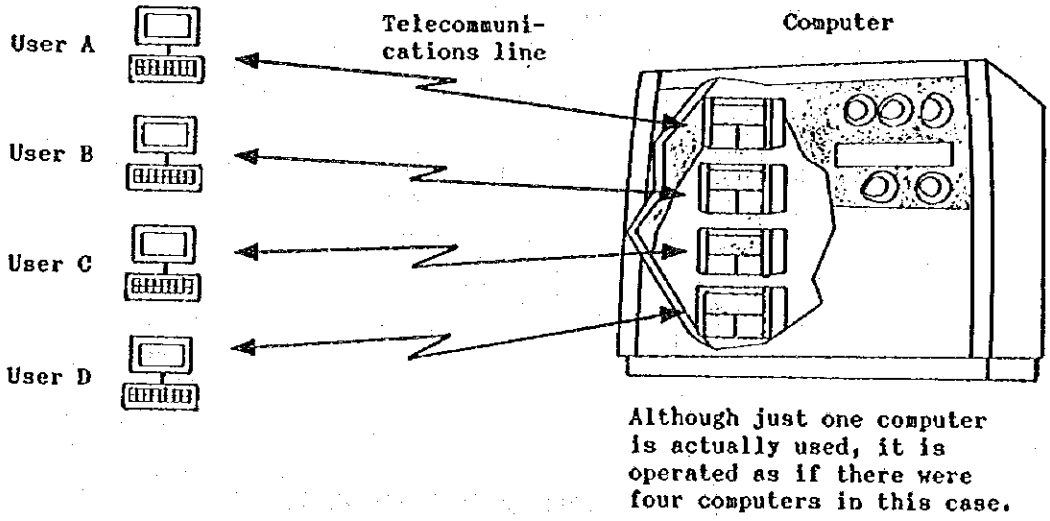

Fig. A.1.7 Time Sharing Processing

APPENDIX-2
SOFTWARE SYSTEM CONFIGURATION [1]

The hardware mentioned in Section 1 requires specific instructions to know what operations to perform. These specific instructions take the form of electronic impulses in the form of bits, and are represented in the diagram by broken arrows. These impulses are coded in preset order, because they may be stored by the computer, and they can be executed repeatedly.

Sets of instructions which control a computer system or perform various tasks are called programmes or software. These terms are often used interchangeably.

2.1 Operating System (OS)

(1) Outlines of an OS

A television, radio, and lamp can all be used immediately after we turn the power on. A computer, however, is not always immediately useful after the power is switched on. For the PC hardware to be functional, it needs some special software to supervise and control all of list functions. This basic software is called the operating system (OS). The PC can perform useful work only after the OS is loaded.

For many computers, the operating system is loaded as soon as the power is switched on, for others it must be loaded by the operator.

(2) Roles of an OS

The OS is the basic software procedure which is loaded into the computer before it can perform useful work. An OS has four main roles:

- Command processing
  The OS accepts all commands and executes them.

- Input/output (I/O) control
  The OS controls and monitors all input/output devices.

- Disk control
  The OS manages the processing, reading, and writing of all data.

- Task control
  The OS supervises and controls the execution of all programmes (tasks).

---

[1] This section is based on the JICA training material "Computer Introduction: Training for Information Processing" OKINAWA International Center, Japan International Cooperation Agency.

Each of these roles will now be discussed in more detail.

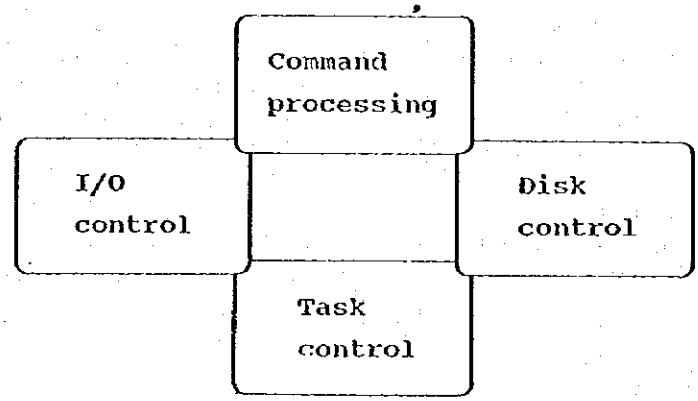

Fig. A.2.1 Roles of OS

Figure A.2.2 shows the four main components of the OS working together in executing a programme.

When the command PROGA is entered, the following processing occurs:

a. The command processor accepts the command (programme name) and searches for the programme.

b. The disk controller retrieves and manages the file information relating to the programme from the directory or disk.

c. The I/O controller provides the input/output methods for loading the programme, inputting the data from the keyboard, and outputting the data to the display unit or the printer.

d. The task controller starts and terminates execution of the programme.

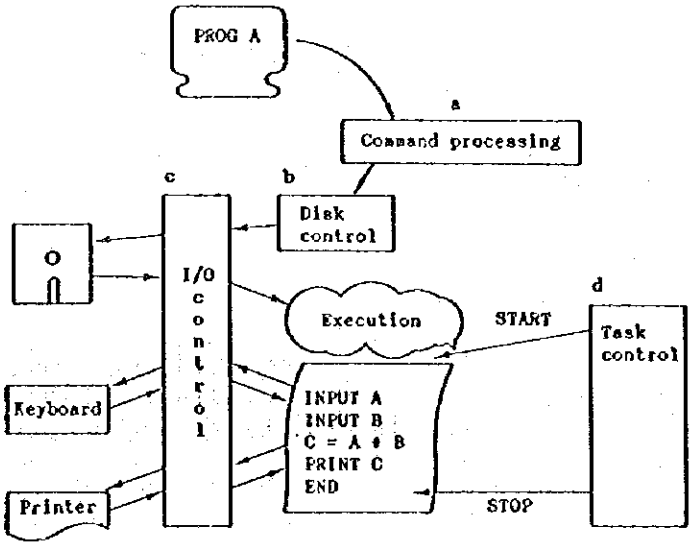

Fig. A.2.2 Structure of OS

(3) Command processing

Every command that we enter causes the system to process a number of instructions. Each command can be thought of as an executable programme. These commands are processed by the command processor.

As shown in the diagram, the command processor of the operating system performs the following roles:

Role 1: accepts a command
Role 2: searches for the specified command (programme), and loads it onto memory
Role 3: executes the programme

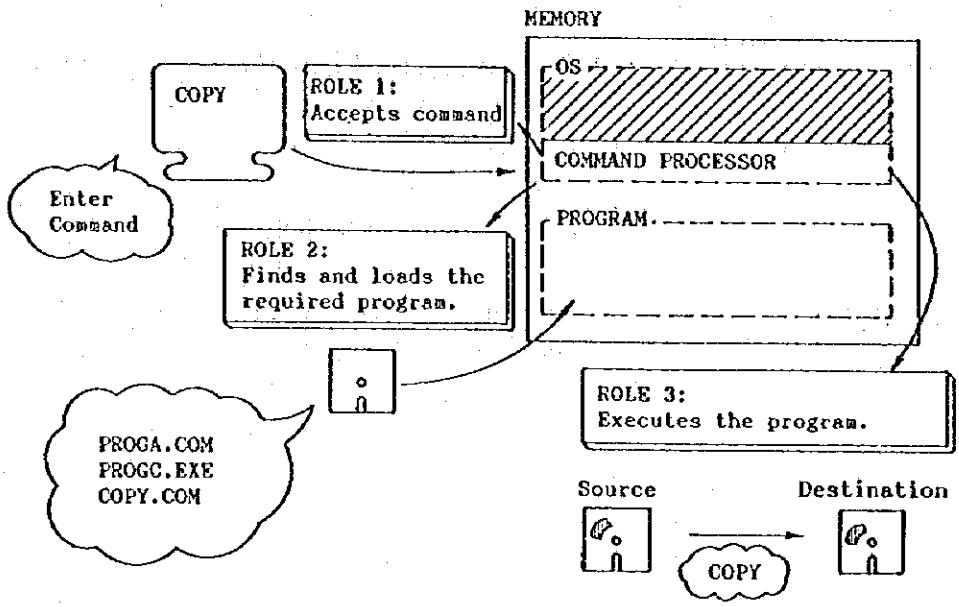

Fig. A.2.3 Command Processing

(4) Introduction of Standard OS

Fig. A.2.4 shows some various of the 16-bit OS.

Their recent versions are:

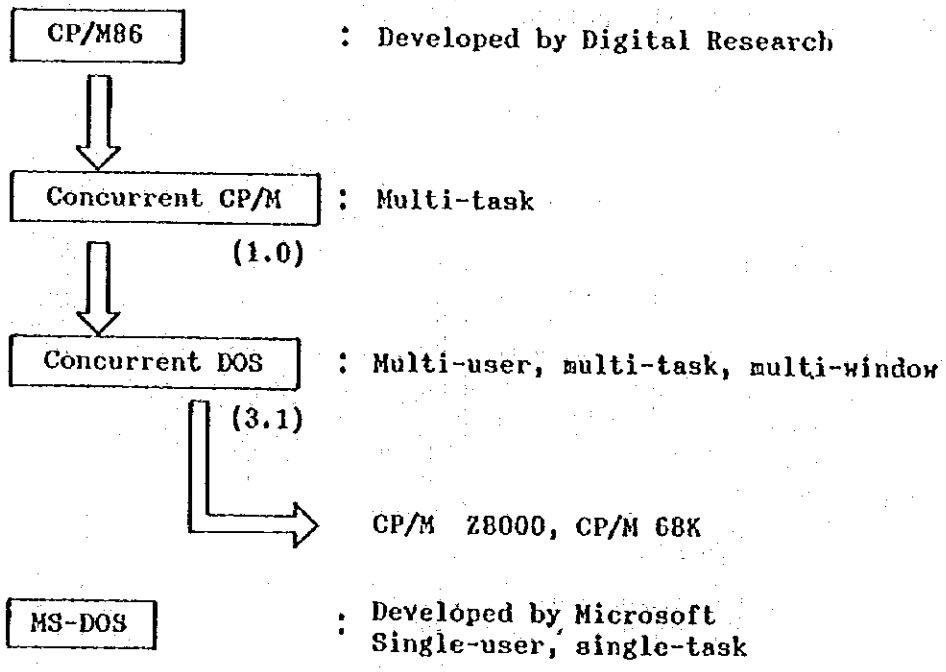

Fig. A.2.4 16-Bits OS

The roles of both CP/M86 and MS-DOS files are the same. Figure A.2.5 shows how each file is used when we input a COPY command from the keyboard:

(1) The I/O controller accepts the command and passes it to the command processor.

(2) The command processor analyzes the command. It uses two kinds of command files. One is stored in RAM and the other is stored on an auxiliary device. The former is an internal command and the latter is an external command.

(3) If the COPY command is stored din RAM, this command would be executed immediately.

(4) If the COPY command is not stored in RAM, the command processor asks the disk controller to load the command.

(5) The command is then loaded by the I/O controller and executed.

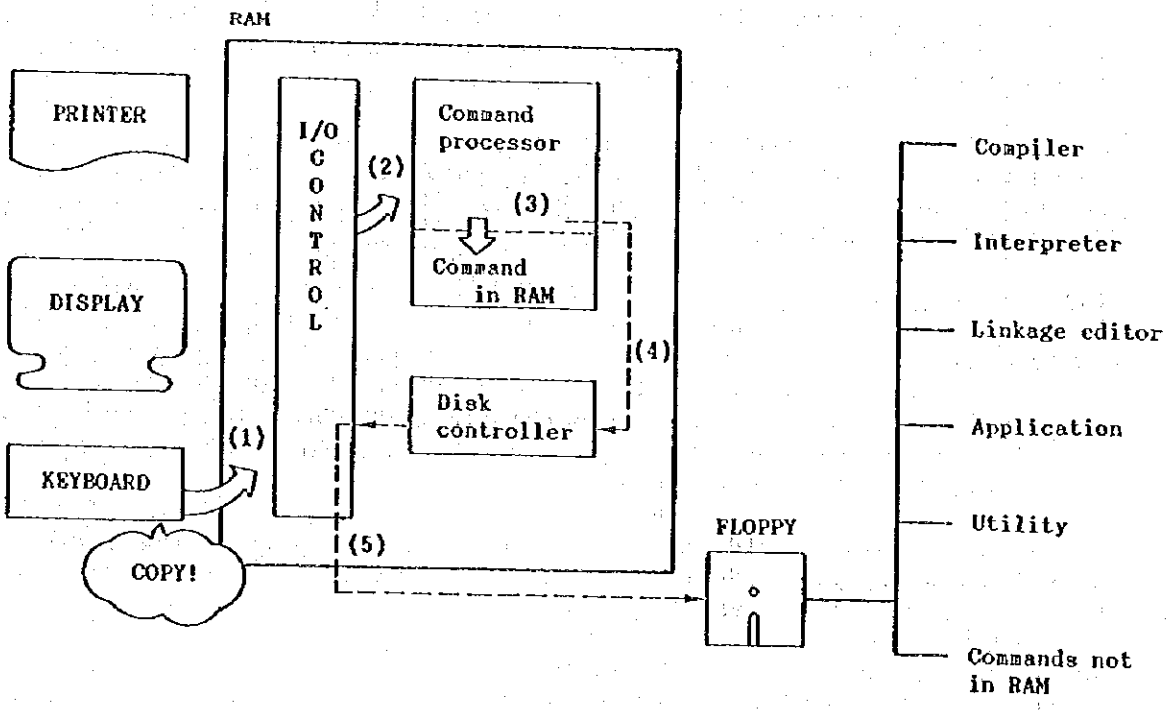

Fig. A.2.5 The Roles of MS-DOS

2.2 Programmes

(1) Outline of Programming Languages

There are a number of different programming languages as follows:

a. Machine language
b. Programming languages
c. Programme translation
d. Software packages
   (Type of software packages)

(2) Machine language

As already discussed, the PC can understand only two conditions, 0 and 1. Every item of data and every instruction must be converted to binary form before it can be processed.

A special language called machine language is composed of binary codes which the PC can understand. Each type of PC has its own unique machine language and, quite often, the machine language of one PC is not understood by another PC.

(3) Programming Languages

a. Assembly language

Machine language is not easy for people to read and write. If we wrote a programme in machine language, we would have to refer to a special code table for every instruction.

To overcome this, a wide variety of easy-to use programming languages have been developed. Most of these are similar to written languages such as English and Japanese.

One such language is assembly language. Each element of this language corresponds to machine language but it can be understood more easily than machine language.

The main features of the assembly language and a number of other languages will now be discussed.

Each assembly statement corresponds to one machine language instruction.

The programme example below is coded in assembly language, and must be translated into machine language before it can be processed.

We use a special translator programme called assembler to do this. Assembler uses a correspondence table to convert each statement into the equivalent machine language instructions.

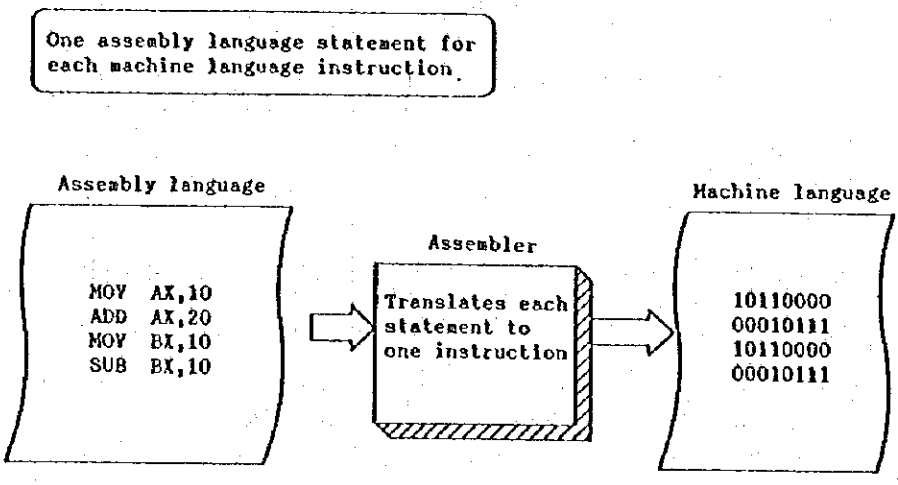

Fig. A.2.6 Assembly Language

b. High-level languages

The development of assembly language makes it easier to code programmes, but it is still very tedious and complex to use. A language which is very close to machine language, such as assembler, is called a low-level language. For more effective programme development, a number of high-level languages have been developed. Each high-level language has its purpose. The choice depends on what sort of processing is to be done.

A number of high-level languages are shown in the diagram opposite. Some of these are:

- COBAL : This is used for business processing.
- FORTRAN : This is used for scientific and technical calculation.
- PL/I : This is used for business, scientific, and technical processing

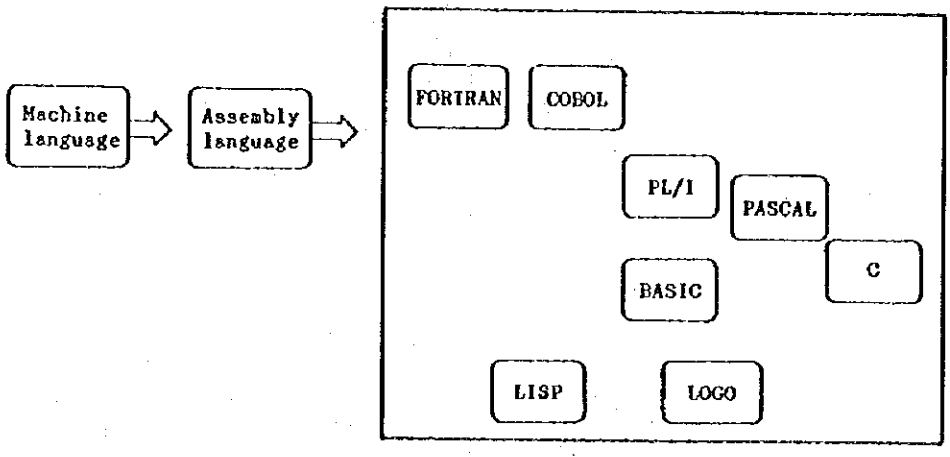

Fig. A.2.7 High-Level Languages

c. Programme translation

None of the example programmes can be processed by a PC in their present form. They must first be translated into machine language. There are two steps in translating a programme.

Compiler

The first step requires the use of a compiler. A compiler. A compiler is a programme which reads the high-level language and converts it into machine language. This process is called compiling.

The programme which is read by the compiler is called a source programme. The compiler produces a translated version of the programme. This translated version of the programme is called the object module.

Linkage editor

For most languages, the programme translated into an object module by the compiler still cannot be executed. To execute it, we need to combine it with some procedures (for example, specific function programmes, commonly used programmes) which are provided in object module form.

This process of combining modules is performed by a programme called the linkage editor. The link-edit process converts the programme to an executable form.

This executable version of the programme is called a load module.

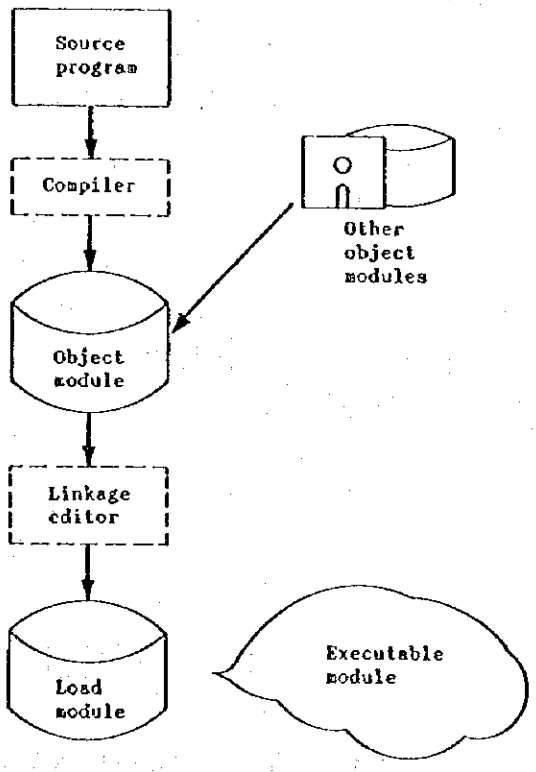

Fig. A.2.8 Programme Translation

d. Interpreter

Some programming languages can be translated and executed by an interpreter.

The interpreter reads a source programme one instruction at a time, translates that instruction into machine code, executes it, and then reads the next instruction.

This is different from compilation which translates all the instructions before execution begins. Interpreters are a useful way of testing programmes and running small programmes.

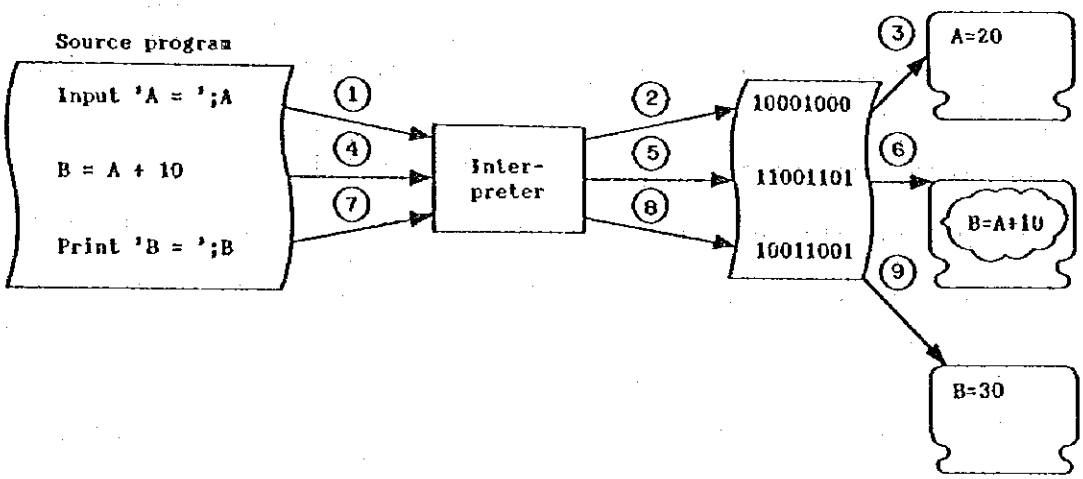

Fig. A.2.9 Interpreter

(4) Software packages

A software package is easy to use.

There is no need to know how to programme a computer in order to use one effectively. There are a number of application programmes available called software packages. They can be easily used by those with little or no programming experience. Some of the features of these packages are:

a. Standard and flexible programmes are provided as a part of a package. They do not need to be written by the user.

b. A few simple commands are used for processing. After selecting or entering a command, instruction messages or prompts are displayed on the screen. This method of interacting with a computer is called interactive command processing.

c. Packages are available for many different applications, including table, graph and report creation.

(5) Types of software packages

There are four general types of software packages. Each has a specific purpose:

1) Spreadsheet software is used to create and manipulate data in tables.

2) Graph software is used to create graphs and charts.

3) Database software is used to create and manipulate a database.

4) Word-processing software is used to create documents and reports.

Each of these packages will now be discussed.

APPENDIX-3
EXPRESSION OF INFORMATION [1]

3.1 Bits and Bytes

(1) Two types of status

On data handled by a computer, codes like, +, -, x, /, =, (,) and symbols like *, $ and ¥are used in addition to numerals, and alphabetic characters, and Japanese characters. How is data expressed by a computer using these?

In a computer system, all data is expressed in a combination of two types of mechanical, electric or magnetic states.

A well known example of transmitting information using combinations of two types of states is the Morse code (dot and dash). Fig. 2.1 shows and example of the method adopted by a computer.

When numerals "0" and "1" are assigned to each of these two types of states, the data state (characters and others) can be expressed.

(2) Bits and bytes

A combination of "0" and "1", explained in (1) is equivalent to the binary notation. This unit of numerals, whether it is "0" or "1", is called a bit (bit stands for Binary digit).

One digit of the binary notation showing a numeral of "0" or "1" is called a bit, and "data handled by a computer is expressed by using several bits."

A numeral, alphabetic character, or symbol is expressed by eight bits. Thus, numerals, alphabetic characters and symbols are expressed and processed under a certain rule in a computer system.

Also, as Fig. A.3.1 shows, one data item consisting of eight bits is called a byte.

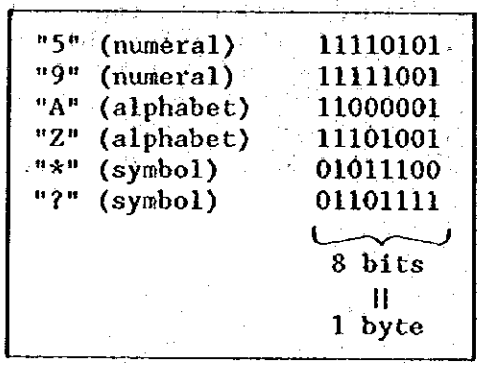

Fig. A.3.1 Bits and Bytes (EBCDIC Code)

[1] This section is based on the JICA training material "Computer Introduction: Training for Information Processing" OKINAWA International Center, Japan International Cooperation Agency.

3.2 Binary Numbers, Decimal Numbers

(1) Decimal numbers

The counting system we use daily is the decimal notation (counts of 1, 2, 3, ..., 9, 10, 11, ...). (It is said that the reason for using decimal notation is because a man has ten fingers and ten toes.) The numerals used in this are the types of "0" through "9". Normally, the count starts form "0" to "9" sequentially. The next number is 10, or moving up of the digit by one.

As shown in Fig. A.3.2, in decimal notation ten numerals from "0" to "9" are used (which is referred to as "10" being the base). Adding "1" to "9" makes "10", moving up the digit by one. The next number to "99" is "100". Thus the digit is sequentially added. Also, you all know that numeric values can be expressed by assigning weight to each digit from the end, like 10^0, 10^1, 10^2, 10^3 ...

Size of decimal number

Base	: 10 (numerals used:	ten pieces of "0" to "9")	
Weight of each digit:	first digit	10_1	= 1
	second digit	10^1	= 10
	third digit	10^2	= 100
	.		.
	.		.
	.		.
	.		.
	.		.

(Example) $(1988)_{10}$ = $(1 \times 10^3) + (9 \times 10^2) + (8 \times 10^1) + (8 \times 10)$
= 1000 + 900 + 80 + 8 = 1988

Fig. A.3.2 Decimal Notation

(2) Binary numbers

The basic number used by a computer system is binary. The concepts of base and weight of each digit used in decimal notation can be applied to binary notation. As shown in Fig. A.3.3, the base in binary notation is "2" and the numerals used are two types of "0" and "1". After "1", the digit is raised, creating "10". After "10" is "11" and then the digit is raised, creating "100", followed by "101", "110" .... The weight for each digit is expressed as 2^0 , 2^1 , 2^3 , ..., in succession.

Base : 2 (numerals used: two pieces of "0" and "1")

Weight of each digit:

first digit	2^0	= 1
second digit	2^1	= 2
third digit	2^2	= 4
forth digit	2^3	= 8
fifth digit	2^4	= 16
sixth digit	2^5	= 32
seventh digit	2^6	= 64

(Example) $(1010110)_2 = (1 \times 2^6) + (0 \times 2^5) + (1 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) + (1 \times 2^1) + (0 \times 2^0)$

$= 64 + 0 + 16 + 0 + 4 + 2 + 0 = (86)_{10}$

Fig. A.3.3 Binary Notation

APPENDIX-4
SYSTEM DEVELOPMENT

4.1 The Concept of A System

The entire system including Data Base Management System for making city plan, should have the main following characteristics:

(1) It should have the overall purpose of making a city plan.
(2) It should consist of many sectors.
(3) System sectors are related to each other in a particular way.
(4) Each system sector is related to the entire system in a particular way.

Data Base Management System plays an important role for the entire system as shown in Figure 2.

4.2 System Development

The purpose of system development is to organize a new information system which can resolve the problems in an existing system.

Fig. A.4.1 The Purpose of System Development

System development has different stages in developing a new system in DTCP.

The major topics as:

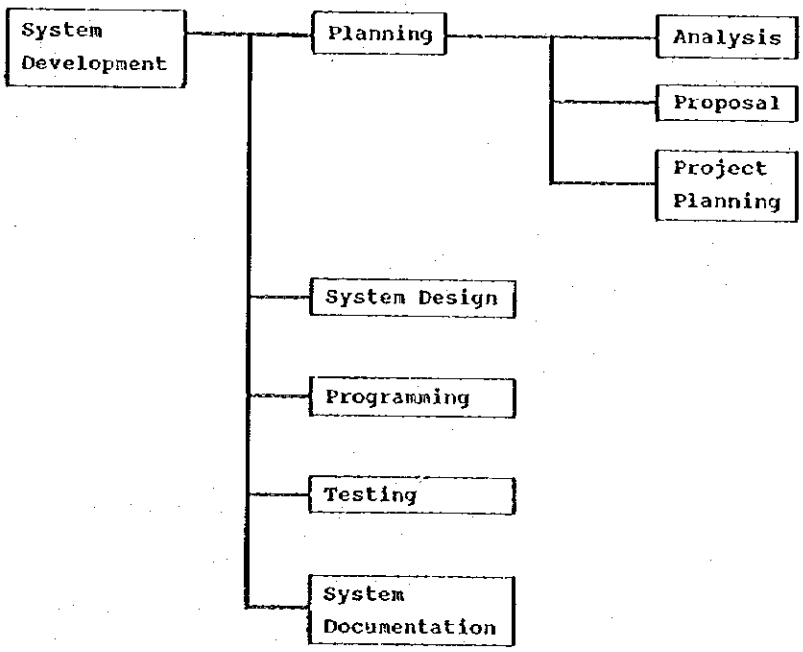

Fig. A.4.2 System Development

Table A.4.1 SUMMARY OF SYSTEM DEVELOPMENT

PLANNING	ANALYSIS	The old system is analyzed in order to identify and document the requirements of the new system.
	PROPOSAL	A proposal is prepared. It is discussed and reviewed by the users and management.
	PROJECT PLANNING	A detailed project plan is prepared.
SYSTEM DESIGN	PROGRAM DIVISION	The system is divided into separate functions of program modules.
	INPUT/OUTPUT ITEMS	The input/output items for each program are identified.
	CODE DESIGN	Codes are designed for some data items.
	FILE DESIGN	The record layouts of files are designed.
	INPUT/OUTPUT FORMAT	The input/output format is designed for printers or display devices.
	PROGRAM SPECIFICATION	The processing to be done by each program (module) is defined, document, and reviewed.
PROGRAMMING	FLOWCHART	A flowchart is produced for each program.
	CODING	The programs are coded.
	UNIT TEST	Each program is tested and debugged.
TESTING	TEST PLAN	A plan is developed for the integrated testing of all the programs (modules) of the systems.
	TESTING	The whole new system is tested according to the test plan.
SYSTEM DOCUMENTATION		The new system is documented for those who will use or maintain it.

4.3 Code Types

(1) Sequenced code

A sequenced code has a particular order of code values. In Fig. A.4.3, the last three digits of employee code represent the order in which the employee joined the company. The first employee to join has '001' as the last three digits of the employee code, the second has '002', and so on.

If a sequenced code is to be used, it is important to allow room for future growth. In our example, the employee sequence number provides for 999 new employees each year. If more than 999 people join the company, the code will have to be expanded to four or more digits.

In the example of a code for canned coffee, 150g and 250g cans are not being produced now. So the values are shown in parentheses.

A. CANNED COFFEE EXAMPLE

NET WEIGHT	50G	100G	150G	200G	250G	300G
C O D E	1	2	(3)	4	(5)	6

B. EMPLOYEE CODE EXAMPLE

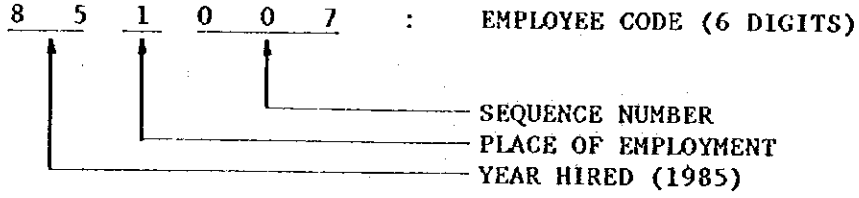

Fig. A.4.3 Sequential Code

[1] This section is based on the JICA training material "System Design Basid: Training for Information Processing" OKINAWA International Center, Japan International Cooperation Agency.

(2) Block code

In a block code, each valid value of the code represents an entity or a category. For example, on Table A.4.2, the code value of '03' means 'accounting department'. Block codes are often used for subfields.

Table A.4.2 Example - Department Codes

CODE	DEPARTMENT NAME
01	ADMINISTRATION
02	PERSONNEL
03	ACCOUNTING
04	FINANCE
05 TO 09	RESERVED
10	DATA PROCESSING
11	COMPUTER OPERATION

(3) Group classification code

Group classification codes are used to represent codes with many subfields by dividing the codes into several levels.

Group classification codes are more flexible and allow for more growth than block codes. For example, Table A.4.3 shows that there are currently only three sections within the administration department, but there can be as many as nine if a one-digit subfield is used for section.

Table A.4.3 Group Classification Code

LEVEL 1		LEVEL 2		LEVEL 3	
HEAD OFFICE	1	PERSONNEL DEPARTMENT	1	PERSONNEL SECTION	1
		ADMINISTRATION DEPARTMENT	2	SECRETARIAL SECTION	1
				GENERAL AFFAIRS SECTION	2
				FINANCIAL SECTION	3
		PLANNING DEPARTMENT	3		

(4) Mnemonic code

A mnemonic coding system consists of characters or digits that are directly related to the name being represented.

Table A.4.4 shows a mnemonic code for color televisions. Mnemonic codes remind us of the values that they represent.

14 INCH COLOR TV	T-14
18 INCH COLOR TV	T-18
20 INCH COLOR TV	T-20

Table A.4.4 MNEMONIC Code

(5) Decimal code

The decimal codes developed for use in libraries are divided into left and right sections. The left column is a group classification code. The right column represents a detailed classification according to the rules of that decimal system.

Table A.4.5 DECIMAL Code

CODE	CATEGORY
000	GENERAL
100	PHILOSOPHY
.	
.	
.	
500	NATURAL SCIENCE
510	MATHEMATICS
.	
.	
530	PHYSICS
531	MECHANICS
531-1	MECHANISM

THE LAST DIGIT FOLLOWS THE PRINCIPLES OF THE DECIMAL SYSTEM.

THE PRINCIPLE DOES NOT APPLY WHEN ADDITIONAL LEVELS ARE ADDED.

(6) Meaningful digit code

Meaningful digit codes consist of digits that represent characteristics of the data.

The first example in Table A.4.6 shows a meaningful digit code for iron plate. It is designed so that the code value represents all the dimensions of the plate.

The second example shows a meaningful digit code for railway stations. It can be designed so that each value represents the distance of the station from some starting point.

Table A.4.6 Example - Iron Plate Codes

WIDTH (M)	LENGTH (M)	THICKNESS (CM)	WEIGHT (KG)	CODE
1.0	4	0.3	20	10040620
1.5	10	1.2	75	15101275

Table A.4.7 Example - Station Codes

LINE #	STATION		CODE
01	TOKYO	000	01000
	YOKOHAMA	026	01026
	ATAMI	101	01101
	SHIMIZU	166	01166

STATION NAME CODES ARE MEANINGFUL DIGIT CODES. 026 (YOKOHAMA) REPRESENTS THE DISTANCE FROM TOKYO.

(7) Cryptic code

These codes are used to represent information that must be kept secret. Codes are created by replacing digits with other characters. This type of code can, for example, be used in payroll systems to keep salary information secret.

Table A.4.8 CRYPTIC Code

0	1	2	3	4	5	6	7	8	9
:	:	:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:	:	:
Z	O	T	R	F	B	S	E	G	N

Examples: 1234 = OTRF

7231 = ETRO

JICA