

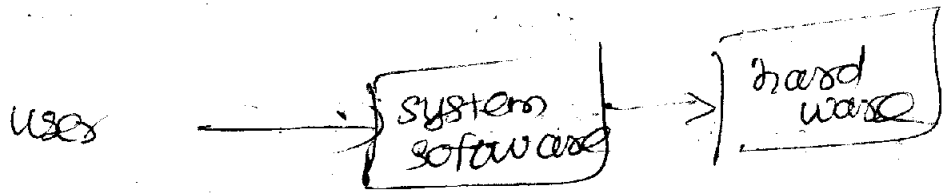
System software

software - set of instructions or programs to carry out certain tasks

System software Application software

System software

- * system software acts as an intermediate between users and hardware
- * A set of programs that creates the environment to work an application software.
- * system software is a computer software designed to operate the computer hardware and provides & maintains a platform for running application software.



(ex) OS, Translators, drivers, loaders, assembler, compiler, interpreter

Application software

A set of programs written for specific purpose. Different organizations need different application programs specific to their need.

(e.g.)

word processing software, database software, Accounting software, spread sheet software and Graphics software.

These application software

includes some of the programs that do real work for the users.

(e.g.) payroll systems, Inventory systems and student maintenance systems, library management systems.

Difference between system software and application software

System software	Application software
1) It is used in the operation of computers	It is used to perform some user task.
2) It is machine dependent software	It is machine independent software.

3. The programmer should know the designing of the computer

Not necessary to know the architecture of the system

4. System software is not meant to be run by the end user

Application softwares can be run by the end user

components of system software

1) Assemblers

3) macros

5) Operating system.

2. loaders

4. compilers or interpreters

1. Assembler

It is a program that translates assembly language program into machine language program.

Assembly language program - source program
" " " " - object program
machine " " " " -

2. Loader

loader is a system software that loads the programs into main memory and prepares them for execution.

macro

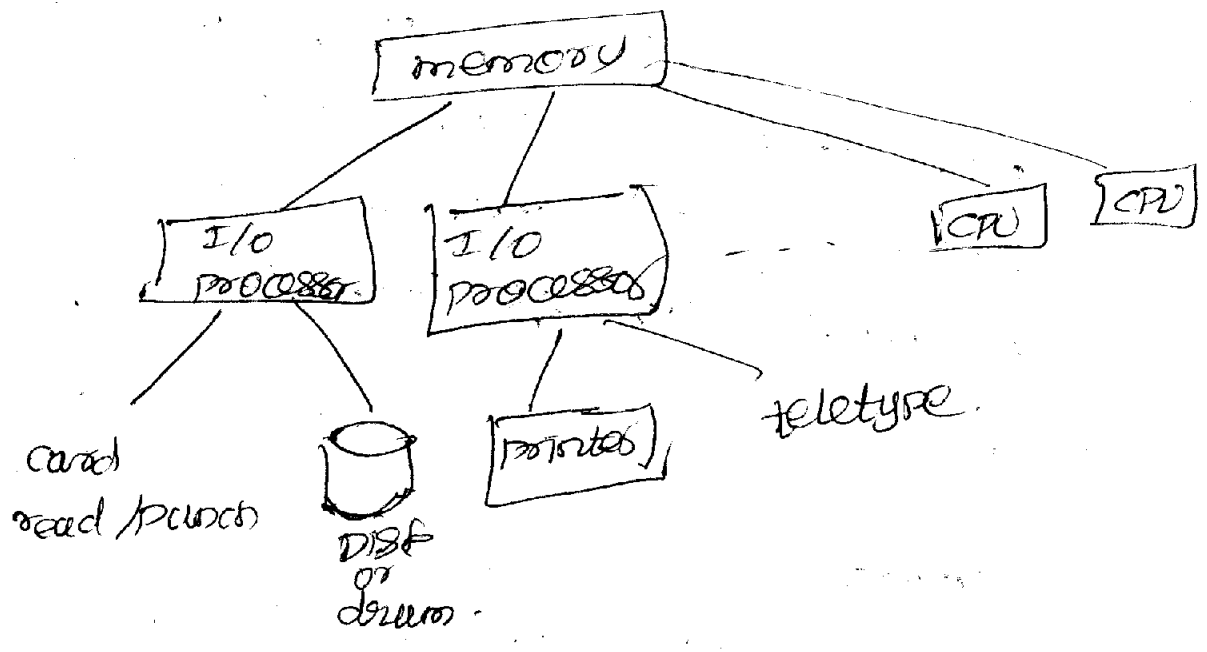
macro processor is a program that substitutes the macro definitions for macro calls.

compiler

compiler is a program that accepts a source program in high level language and produces its corresponding object program.

operating system

machine structure



memory

memory is the device where the information is stored and retrieved. Information is stored in the form of 0's and 1's. Each 1/0 is a separate binary digit called a bit. Bits are grouped into words, characters or bytes.

- nibble - 4 bits
- byte - 8 bits
- half word - 16 bits
- word - 32 bits
- double word - 64 bits

Basic unit of memory is byte.

Data values to be operated.

Instructions - operations to be performed.

A code is a set of rules for interpreting group of bits.

BCD - decimal digits

EBCDIC or ASCII - characters

processor opcode instructions

Processor

(6)

processor is a device that performs a sequence of operations specified by instructions. The sequence of instructions is called a program or procedure.

There are two types of processors.

1. CPU (Central Processing Unit)
2. Input/Output processors.

Central Processing Unit (CPU)

It is the brain of the computer. It consists of ALU, memory unit and control unit. The CPU is responsible for all events inside the computer. ALU performs arithmetic and logical operations in a computer. All the instructions and programs are stored in memory unit (The control unit fetches the instruction and decodes and executes them). The CPU carries the instructions of program in sequence.

Input / output processors

Input / output processors transfers data between memory and peripheral devices such as disks, drums and printers.

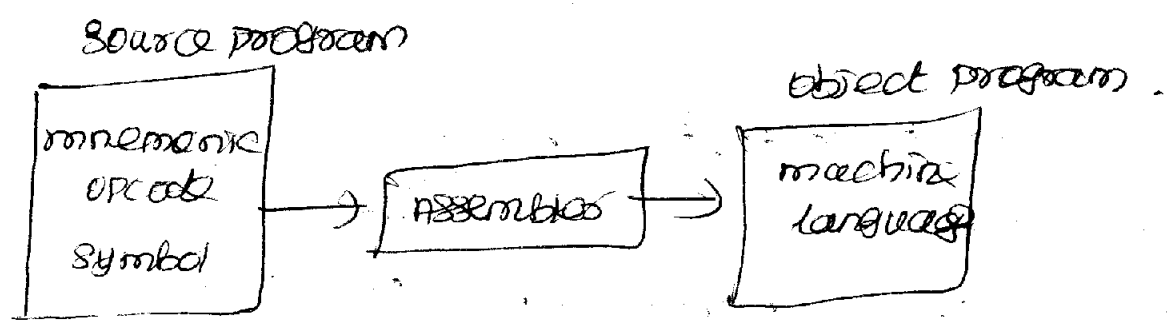
I/O instructions are stored in memory. I/O processors executes these instructions which are activated by a command from the CPU.

Evolution of the components of a programming system.

Assembler

In earlier stage, the computer programmer used to write programs using 0's and 1's (machine language). Programmers found difficulty in writing or reading programs using machine language. Then they began to use assembly level language, where they use mnemonic (symbol) for each machine instruction. It's very easy to remember the mnemonics (symbols and letters).

Assembler is a system software which is used to convert an assembly language program to its equivalent object code.

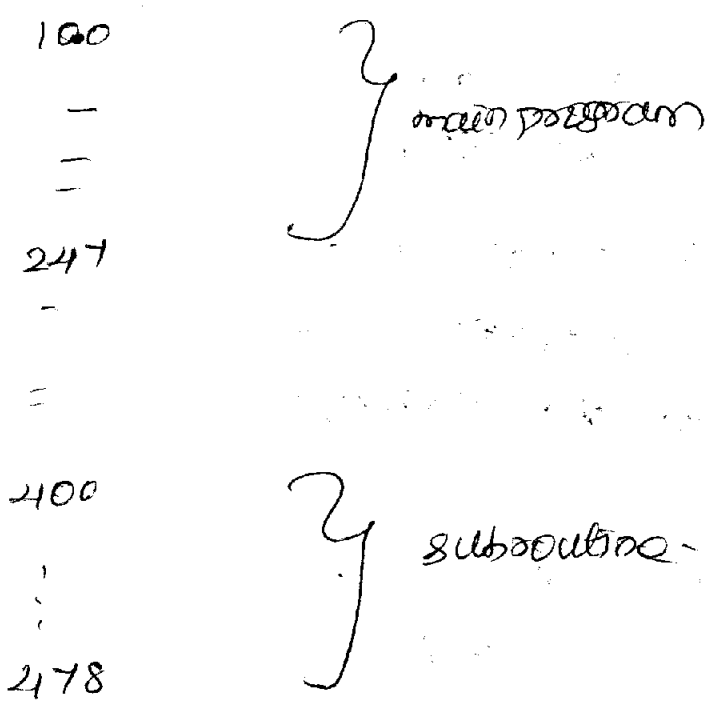


Loaders

A loader is a program that loads the object programs into memory in an executable form.

A loader is the part of a program that loads object programs into memory and prepares them for execution.

A subroutine is a body of computer instructions designed to be used by other routines to accomplish a task. For example, SORT is the subroutine. The main program is assigned to the locations 100-247 and SORT subroutines is assigned to the locations 400-477.



If changes were made to main that increased its length to more than 300 bytes end of main ($100 + 300 = 400$) would overlap the start to SRT (at 400). At that time the loader will relocate the SRT subroutine to new location.

Relocation

Relocation is the task of adjusting programs so that they may be placed in arbitrary memory locations.

Relocating loaders perform the following four functions

- 1) Allocation: Allocate space in the memory for programs
- 2) Linking: resolve the symbolic references between object codes.

3) Relocation : Adjust all address dependent location

4) Loading : Physically place the machine instructions & data into the memory.

Here depending upon the functions the loader is divided into different types. They are

1. compile and go loader
2. Absolute loader
3. re-locating loader
4. direct linking loader
5. Dynamic loading
6. Dynamic linking

Execution time :-

It is the period of execution of a user's program

compile time :-

It is the period of translating user's source program into object program.

Load time :- It is the period of loading and preparing an object program.

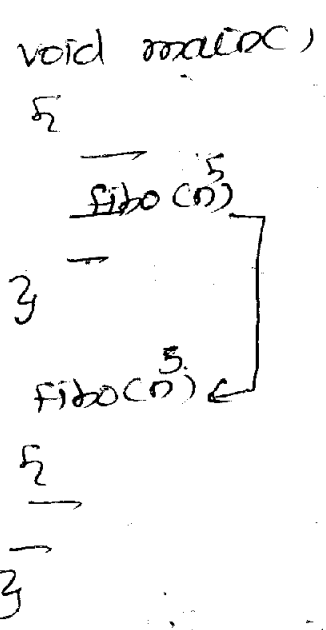
Macro

Subroutine is the part of the program. Two types of the subroutine

- (i) Open subroutine
- (ii) Close subroutine

closed subroutine

It is stored outside the main routine and during the subroutine call control is transferred to it. Associated with closed subroutine, the main program performs two tasks, it transfers the control and transfer of data.



Open Subroutine

Open subroutine or macro definition is one whose code is inserted

into the main program. It performs only one task. (2) insert a macro. If an open subroutine was called four different times, the macro definition would appear in four different places in calling program.

```
#define PI 3.14
```

```
void main
```

```
{
```

```
    (3.14)
    A = PI * r * r;
    (3.14)
    C = 2 * PI * r;
```

3 ← example of macro in 'c' programming
 To eliminate the need of repeating identical parts of the program, operating systems provide a macro processing facility.

macro permits to define an abbreviation for a part of the program. and to use the abbreviation anywhere in a program.

macro :- macro are single line abbreviation or name for some group of instructions

macro definition :- It is a sequence of code that has a name.

macro call :- It is the occurrence of the macro name in the source program.

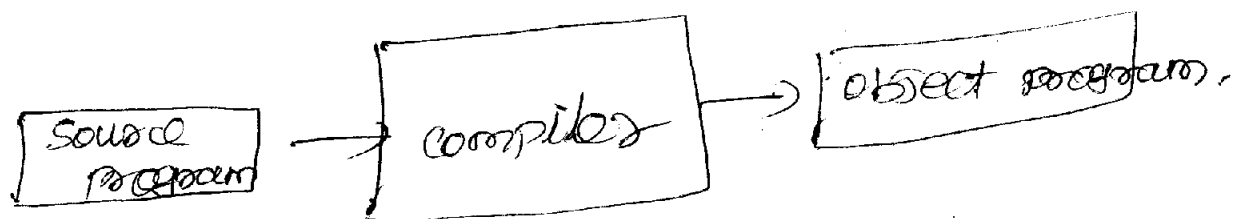
macro processor :- It is a program that substitutes the definition for all occurrences of the macro call.

Compilers

The program that converts a high level language into machine language is called a translator. There are two types of translators. They are compilers and interpreters.

A compiler is program that accepts a program written in high level language and converts into an object program.

During the compilation process it checks each coded instruction to see that it follows the grammar for that language and uses the proper spelling and syntax. All the acceptable instructions are translated into the machine language and stored. In case of errors in the program, it communicates this information to the programmer in the form of messages. These messages appear on the output device along with the program listing.



Interpreter is another type of translator used for translating programs in high level language into machine code. Interpreter interprets the instruction line by line.

Formal systems

A formal system is an uninterpreted calculus. It consists of

1. An alphabet
2. A set of words.
3. A finite set of relations called rules.

Example

set theory, Boolean algebra

Uses of formal systems

1. used in design, implementation and study of programming languages.
2. used to specify the syntax and semantics of programming languages.

Operating system

An operating system (OS) is a program that controls the execution of an application program that acts as an interface between user and computer hardware.

Evolution of operating systems

OS has evolved over the last 60 years through several distinct phases or generations.

Serial processing

operating system performs its task serially.

Advantages

simple and ease of programming.

Disadvantages

Poor utilization of resources
time consuming.

Batch processing

In the operating system batch processing, the similar jobs are batched together and executes at a time.

Disadvantages

- (i) CPU is idle because the synchronization gap between the CPU speed and I/O devices.
- (ii) Increase in response time.
- (iii) no human intervention during execution.
- (iv) At a time all the system resources dedicated to a single program.

Multi programming

- It allows the OS to keep more than one job on the memory at a time.

- The control transfers or context switch occurs on I/O events.

Advantages

- Increase throughput
- Lower response time.

Disadvantages

- complex design
- protection issue;

Time sharing

- It is a special case of multi programming
- Each user process is allocated with a fixed time limit for execution called time slice / time quantum

Advantages

- more CPU utilization
- low response time

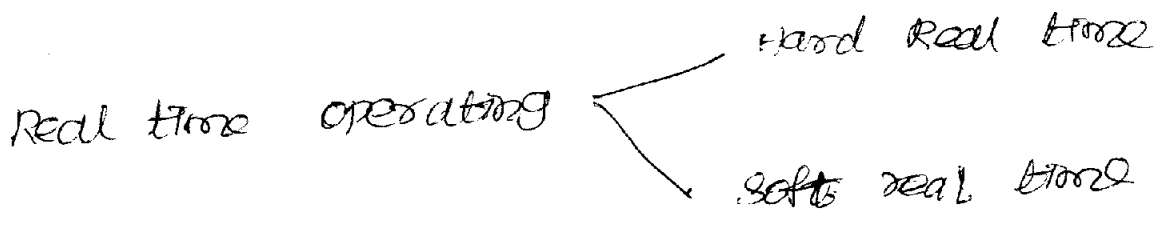
Disadvantages

- very complex
- more resource required

Real time

A real-time operating system is a multi tasking operating system for real-time applications. such applications include embedded systems, industrial robots.

The output of one process is input for another process -



Hard Real time

- It guarantees that critical tasks can be completed on time.

Soft Real time

It is less restricted. It indicates in soft real time system where a critical real time task gets priority over other tasks.

A real time operating system provides the following facilities for this purpose

1. multi tasking within an application
2. Ability to define the priorities of tasks.
3. priority driven scheduling.

Distributed systems

A recent trend in computer system is to distribute computation among several processors. The processors

communicate with one another using communication lines, such as an ethernet. The processors in a distributed systems may vary in size and function and is referred by a different names such as sites, nodes, computers and so on.

The major reasons for building distributed systems are

1. Resource sharing :- If a number of different sites are connected to one another, the user at one site may be able to use resources at the other.
2. Computation speed up :- If a particular computation can be partitioned into a number of sub partitions that can run concurrently, then a distributed system may allow user to distribute computation among various sites.
3. Reliability :- If one site fails in a distributed system, the remaining sites can potentially continues its operation.
4. Communication :- There are many instances in which program needs to exchange with one another.