Noman Shoaib

Bscs-f19-m-86

Harris bin Abid

Bscs-f19-m-84

1.Differentiate between computer organization and computer architecture by giving relevant examples.

Ans] The basic distinction between the computer Organization and computer architecture is the explanation of the operating of the computer. Computer architecture explains what computers ought to do and the computer organization explains how computers work.

Difference between CO and CA

The Computer organization is the study of the behavior and structure of different building blocks of a computer system while Computer design/architecture refers to the transformation of the behavior to the corresponding structure. Moreover, the detailed study of the science of dividing the system functions in software and hardware modules and establishing the interface among them is understood as computer architecture.

The merging of computers and therefore communication technologies have already ushered in a significant technological revolution. This successively will cause the merging of electronics with computers. New and innovative designs to suit the precise requirements of the changing scenario are expected to further evolve within the near future. Hence, computer organization study is a must.

Computer Architecture

Computer architecture is science for designing a computing system. Even as a building architect prepares a design for a building, similarly a computer architect designs a high performance-system at an inexpensive cost, fulfilling all other requirements. Computer architecture provides various attributes to the pc system, which are needed by a machine language programmer or a system software designer to develop a program. It's a conceptual model.

- Computer architecture is an interface between hardware and software
- It is an abstract model and is a programmer's view in terms of instruction, addressing modes and registers
- It describes what the PC does.
- While designing computing system architecture is taken into account first
- it deals with high-level design issues

Computer Organization

Computer Organization gives an in-depth analysis of its functional structure and logical interconnections between different functional blocks. It includes hardware. Any of two computers with an equivalent architecture can have different organizations. And two computers having an equivalent organization can differ in their architecture. Also, it's worth noting that for designing a computer, its architecture is fixed first then its organization is set.

- Computer Organization deals with the components of connection during a system.
- It expresses the belief of architecture.
- It describes how a computer does a task.
- Computer Organization is completed on the idea of architecture deals with low-level design issues.

Difference between Computer Organization and Computer Architecture in tabular form:

Computer architecture explains what a computer ought to do.	Computer organization clarifies how a computer works.
It deals with the interface between software and hardware	It deals with the lower-level details.
It deals with Hardware System Architecture (HSA).	It deals with Instruction Set Architecture (ISA).
The task is to seek out and investigate the organizational structure for its proper operation.	It investigates instruction sets, instruction formats, and addressing technology. It also includes the specifications of varied functional modules like CPU and memories.
While designing, it is not fixed first.	The computer's organization is set after the architecture is fixed.
It shows computer's performance	It shows the computer's hardware.

4. What is the importance of memory in computers? Briefly explain about ROM, RAM, Cache, Flash memory.

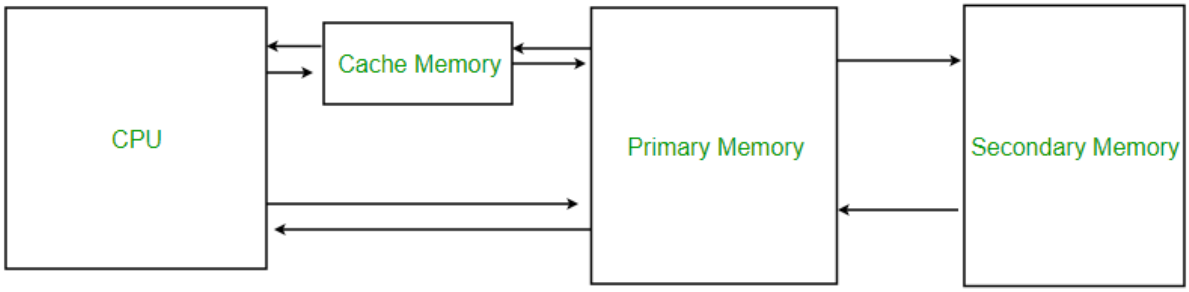
Ans] **Computer memories** are liable for storing data into memory. Entered data and directions must be stored in a storage unit before the particular processing starts. Similarly, the result produced by the pc after processing must even be kept somewhere inside the pc system before being passed on to the output units. Moreover, the results produced by the system must be preserved for an ongoing processing. Functions that are performed by the memory unit are as follows -

(a) It holds input files and directions for processing.

(b) Holds intermediate results of processing.

(c) Holds results of processed data, before these results are released to an output device.

Thus, the CPU makes use of the subsequent memory subsystems for completing its processing operations.



RAM

The main memory is referred to as the internal memory, primary memory, or temporary memory of the pc. It's also referred to as Random Access Memory (RAM). It's a short-lived data-storage medium that holds the info just for a short period hence the name "temporary". The data stored within the RAM gets erased when the pc is turned off. The memory space of RAM is restricted and thus not all the files and instructions can be stored in it. These files and instructions are usually stored in a different location referred to as secondary storage and are copied from secondary storage to main memory before execution. This method is stated as swapping.

The memory space available in RAM also affects the speed of a computing system. In straightforward term,

We have a C drive in the computer system wherever all window programs are installed. A computer would like an area where it can store the OS like Windows 10, 7, 8, etc. When an operating system runs on a system, It desires a RAM where the computer saves files for some time for temporary use. If the space of RAM is additional, the pc can run quicker. If the RAM has low space like one GB then the PC will work slowly.

For example: If there's less traffic on the road you'll drive your car quickly however in case of high traffic you'll drive slow. A similar issue with computers. Once you have an additional GB of ram, your laptop can run quicker. however, once you use Less GB RAM your computer will run slow.

In most straightforward terms, RAM is memory within which windows systems run and execute programs. The additional RAM makes it extremely efficient for running any application.

The importance of memory is that once a computer has massive space for temporary storage, it'll make your computer run quicker.

The main memory is additionally liable for holding intermediate data transferred between the Central Processing Unit and Input/output devices.

ROM

It is the memory that stores the data permanently. This suggests that it doesn't lose the data when the power is turned off. The data are often easily read from this sort of memory but can not be changed. ROM is usually utilized in devices like calculators, laser printers, etc.

ROM doesn't allow the random access of data rather it allows sequential access of data. It's less costly as compared to RAM and other storage devices like magnetic discs, etc.

ROM is split into the following **types** -

Programmable Read-Only Memory (PROM) is often programmed by the user for converting critical and lengthy operations into microprograms that are fused into a chip. they will be executed at a high speed. Once operations are written into a PROM, they can't be altered.

Erasable Programmable Read-Only Memory (EPROM) is often erased and reprogrammed. Before to simply accept any new content, it's far away from the processor and exposed to ultraviolet a few times.

Electrically Erasable Programmable Read-Only Memory (EEPROM) is often erased and reprogrammed with special electric pulses.

Flash ROM (FROM) may be a sort of EEPROM that stores the knowledge using floating-gate transistors, which may store charge for an extended period as compared to the traditional transistors.

This memory is especially utilized in the memory cards of mobile phones, digital cameras, and iPods for storing data. The info stored in flash ROM memory is often easily transferred using transmission mediums like data cable, Bluetooth, and infrared technology.

Flash ROM has a faster speed of reading data as compared to other sorts of ROM. It uses continuous memory cells for storing data.

It is a little, fast, and expensive memory that stores the copies of data that are needed to be accessed frequently from the main memory. The processor, before reading data from or writing data to the main memory, checks for equivalent data within the cache memory. If it finds the data within the cache memory, the processor reads the data from or writes the data to the cache itself because its time interval is far faster than the main memory. The cache memory is placed between the CPU and the main memory of the pc system, the transfer of knowledge between the processor and therefore the cache memory is bidirectional. The availability of data within the cache is known as a cache hit. The potential of cache memory is measured based on cache hit. There are two sorts of cache memory in a computing system –

- **Primary Cache** – it's also referred to as level 1(L1) cache or internal cache. The primary cache is found inside the CPU. It's a smaller but fastest kind of cache that gives fast access to the frequently accessed data by the microprocessor.

- **Secondary Cache** – it's also referred to as level 2 (L2) cache or external cache. The secondary cache is found outside the CPU. It's normally positioned on the motherboard of a computer. The secondary cache is larger than the first cache but slower.

What is a Cache?

The cache is a very high speed, expensive piece of memory, which is used to speed up the memory retrieval process. Due to it's higher cost, the CPU comes with a relatively small amount of cache compared with the main memory. Without cache memory, every time the CPU requests for data, it would send the request to the main memory which would then be sent back across the system bus to the CPU. This is a slow process. The idea of introducing cache is that this extremely fast memory would store data that is frequently accessed and if possible, the data that is around it. This is to achieve the quickest possible response time to the CPU.

Flash Memory is also called Non-Volatile memory, where the term "Non-Volatile" means that the memory of a device can retain data irrelevant to whether or not the device is electrically powered or not. It gained its name (flash) as such, because of the very fact that the memory cells section is erased in the blink of an eye.

An example of non-volatile memory is a hard disc, which retains all the data, even when the system is turned off. However, flash memory has significant differences from that of hard disks. As both can retain data, even if the power is turned off. However, flash memory has a different size, weight, and working function than that of a hard disk.
If you happen to use any kind of digital components that are compatible with the system, then you probably own flash memory. Digital cameras, camcorders, video games, and even GPS units use flash memory, to capture the related important information, which can be later downloaded onto your PC.

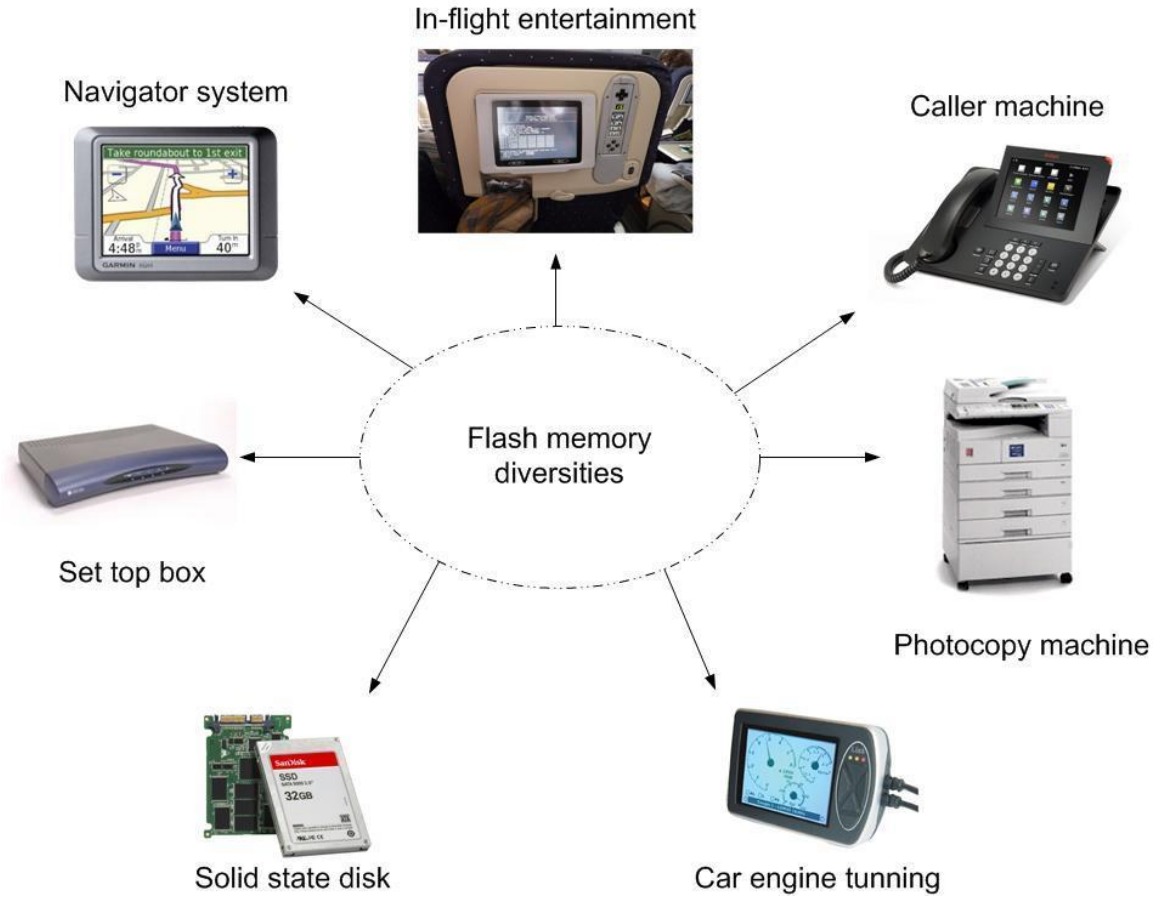
Flash memory is available on different devices such as memory cards, memory sticks, USB drives, and so on. However, the basic operating principle is the same. The flash memory is simply versatile as it cost-efficient and is offered with the operating flexibility of a simple plug-and-play option. The memory stored on this storage media can be erased and rewritten again according to its use.

Flash Memory Types

There are two types of Flash Memory called NAND Memory & NOR Memory, where NAND is programmed and is read in blocks and NOR allows it to be written or read as a byte.

- **NOR Flash**: In a NOR gate, each cell of the gate contains a direct connection of its one end to the ground and the other end to the bit line. Since, this arrangement acts as a NOR gate (a combination of OR gate followed by an inverter, which provides an output as TRUE if both inputs are false, alternatively the output is False), it's called as NOR flash, which can be explained as, When one-word lines, which is connected to the control gate(CG) is carried high, the output bit is put low in corresponding to storage transistor. This is why this low-read latencies feature of NOR makes the flash-driven devices a single memory product to make possible for both direct code execution and data storage. As a result, the NOR flash is currently being used as the technology of selection for embedded applications entailing a discrete non-volatile storage device.

- **NAND Flash**: NAND flash uses Floating Gate Transistors, but they're connected in a way that replicates NAND circuits where several transistors are connected in series and only just in case all word lines are pulled high, the bit line is pulled low. The NAND flash transistors group is connected to a NOR style bit line array within the same way that single transistors are linked in NOR flash. an additional level of addressing is seen in NAND, as replacing single transistors with serially linked groups is completed. Bit-level addressing will enable one bit at time access. But every bit within the word must be simultaneously addressed to execute in-place applications.



5: What do you know about machine language? Also, differentiate between mnemonics and opcode/ operation code.

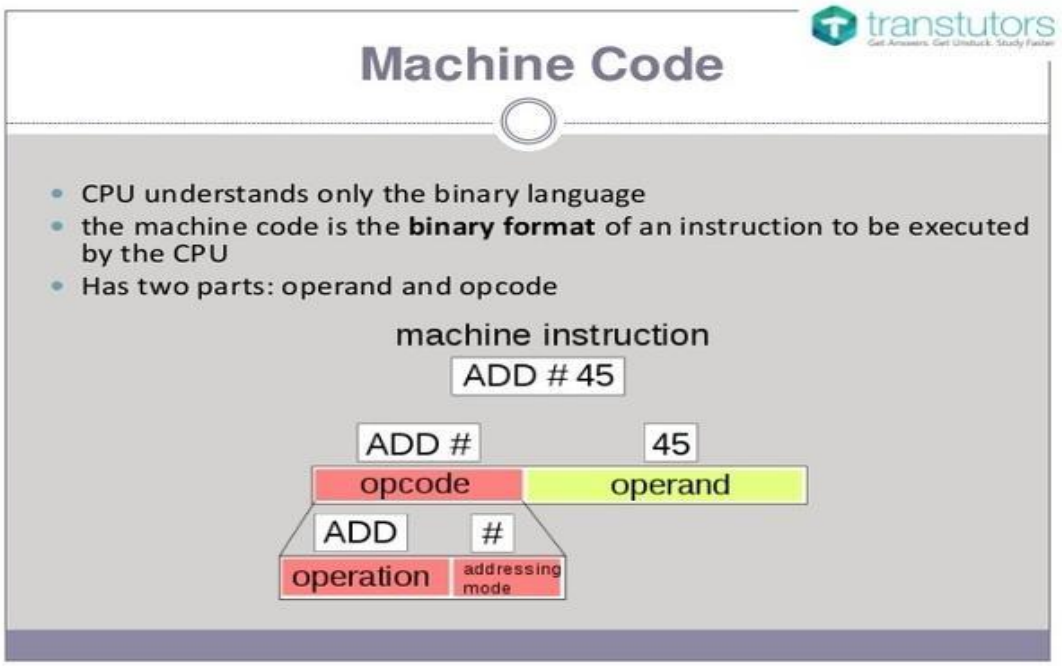
Ans] **Machine code**, which is also referred to as machine language, is that the elemental language of computers. The computer's processing unit (CPU) reads it, which consists of digital binary code and appears to be a sequence of zeros and ones. So, the source code of every human-readable programing language must be translated to machine language by a compiler or an interpreter, because binary code is the only language that hardware can understand.

Each CPU has its specific machine language. The processor reads and handles instructions, which tell the CPU to perform a straightforward task. Instructions are a particular number of bits. If instructions for a specific processor are 8 bits, for instance, the first 4 bits part (the opcode) tells the computer what to try and do then the second of the 4 bits (the operand) instructs the computer what data to use.

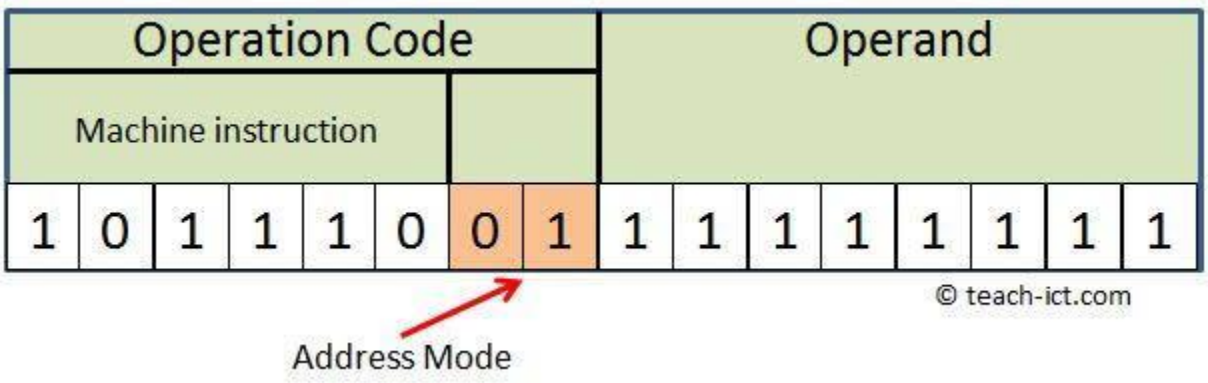
010111110 00100001

Depending upon the processor, a computer's instruction sets may all be an equivalent length, or they'll vary, depending upon the particular instruction. The architecture of the actual processor determines how instructions are patterned. The firmware or the CPU's internal wiring controls the execution of instructions.

Programmers/Developers don't deal directly with machine language anymore. If developers are debugging a program at a low level, they could use a printout that shows the program in its machine language 0/1 form. The printout, which is named a dump, is extremely difficult and to figure with a tool called a dump. Utility programs that are used to create dumps will often represent four bits by one hexadecimal to form the machine language easier to understand and contain other information about the computer's operation, such as the address of instructions that were executed when the run time of the dump was initiated.



OPCODE: it's a number interpreted by your machine(virtual or silicon) that represents the operation to perform.



MNEMONIC: English word mnemonic means that "A device sort of a pattern of letters, ideas, or associations that assists in remembering one thing.". So, it's typically utilized by assembly language programmers to remember the "OPERATIONS '' a machine will do, like "ADD" and "MUL '' and "MOV '' etc. which is often program-specific.

Address	Instruction	Data
0001	LD	00000
0002	OR	10015
0003	AND NOT	00001
0004	OUT	10015
0005	LD	10015
0006	AND NOT	TIM001
0007	AND	00002
0008	OUT	10000
0009	LD	10015
0010	AND NOT	TIM001
0011	AND	00003
0012	OUT	10001
0013	LD	10015
0014	AND NOT	TIM001
0015	AND	00004
0016	OUT	10002
0017	LD	10000
0018	OR	10001
0019	OR	10002
0020	AND	TIM000
0021	AND NOT	TIM001
0022	OUT	10004
0023	LD	10015
0024	AND	00005
0025	TIM	001 #0020
0026	LD	10000
0027	OR	10001
0028	OR	10002
0029	TIM	000 #0020

Difference

- In computer assembly program (or assembly) language, a mnemotechnic is an abbreviation for associate degree operation. It's entered within the code field of each assembly program instruction. as an example AND AC,37 that suggests and therefore the AC register with thirty-seven. so AND, SUB, and MUL are mnemonic. they will get translated by the assembly program.
- Instructions (statements) in a programming language are usually very straightforward, in contrast to those in high-level programming languages. Generally, a mnemonic could also be a symbolic name for one executable machine language instruction (an opcode), and there is a minimum of 1 opcode mnemonic defined for each machine language instruction. every instruction generally consists of an operation or opcode, plus zero or more operands.

6) Give your understanding of the following listed items.

- interpreter
- compiler
- assembler
- linker

Interpreters

An interpreter is also a program that translates a high-level language into a low-level one, but it does it at the instant the program runs. You write the program employing a text editor or something similar, then instruct the interpreter to run the program. It takes the program, one line at a time, and translates each line before running it: It translates the primary line and runs it, then translates the second line and runs it, etc.

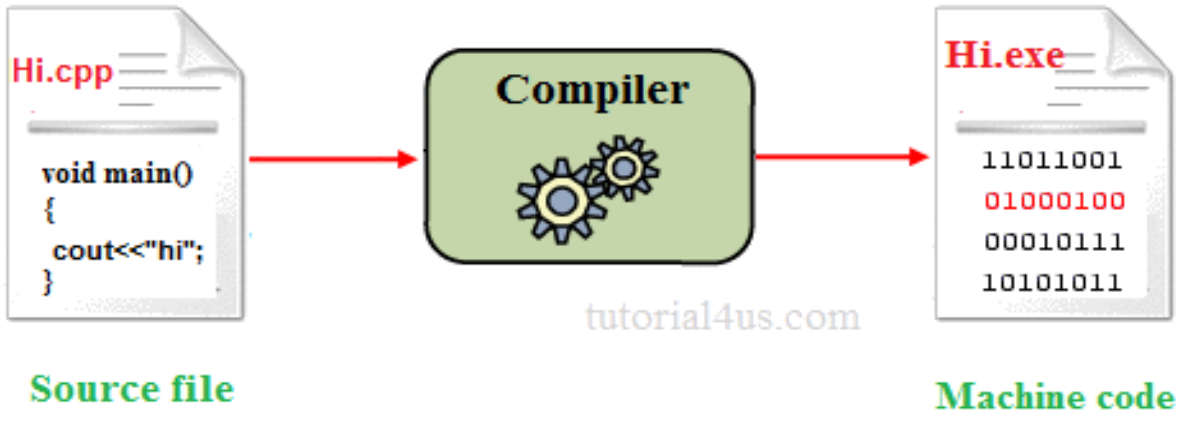
Interpreter characteristics:

- A relatively small amount of time is spent on analyzing and processing the program.
- The resulting code is a few forms of intermediate code.
- The resulting code is interpreted by another program.
- Program execution is comparatively slow.

Compiler

Compilers are utility programs that transform your code into executable machine language files. When you run a compiler on your code, first, the preprocessor reads the source code (the C++ file

you simply wrote) and searches for any preprocessor directory(lines of code beginning with #). Preprocessor directives cause the preprocessor to alter your code in some way (by typically adding some library or another C++ file). Next, the compiler works through the preprocessed code line by line translating every line into the acceptable machine language instruction. this can additionally uncover any syntax errors that are found in your ASCII text file and can throw an error to the instruction. Finally, if no errors are found, the compiler creates an object file with the machine language binary necessary to run on your machine. whereas the issue file that the compiler just created is probably going enough to try to do something on your system, it still isn't a working executable of your C++ program. there's a final vital step to succeed in an executable program. C++ contains an enormous library to help in performing tough tasks like I/O and hardware manipulation. you'll be able to include these libraries with preprocessor directives, however, the preprocessor doesn't automatically add them to your code. For you to possess a final executable program, another utility stated as the linker should combine your object files with the library functions necessary to run the code. Think of it as having all the mandatory blocks to create a house. The compiler created all the blocks however the linker is that the one that sticks all of them together to finally produce a house. Once typically|this can be} often done, you now have a functioning executable file!

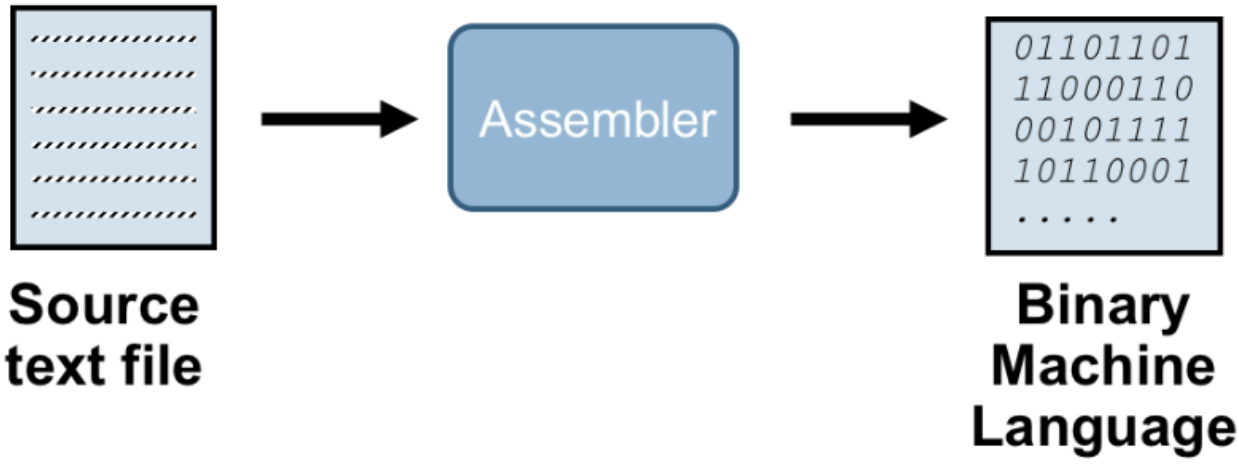


Assembler

Assembly Language is similar to compilers. The main difference is that most Compilers convert a High-Level Language like C, C++, Python to the Assembly language equivalent then use Assemblers to convert it into low-level binary Machine Language of the given Microprocessor.

All Microprocessors accompany a Language set of Instructions at a really low level (Binary Bit sequences) involving a Program Counter, assorted Registers, an Accumulator, and an Arithmetic Logic Unit (ALU). This makes it easier for a Programmer, Assembly Language was created to put source code Instructions in mostly 3 character Mnemonics easier for a human to read as compared to its Binary Equivalent. Examples are like MOV X, Y to move data from register X to Register Y; or, AND B A to perform an AND operation between A and B and place the result in the Accumulator A, or NOP for a no Operation Instruction which will fill an instruction slot in memory and to eat clock cycles for that instruction for timing.

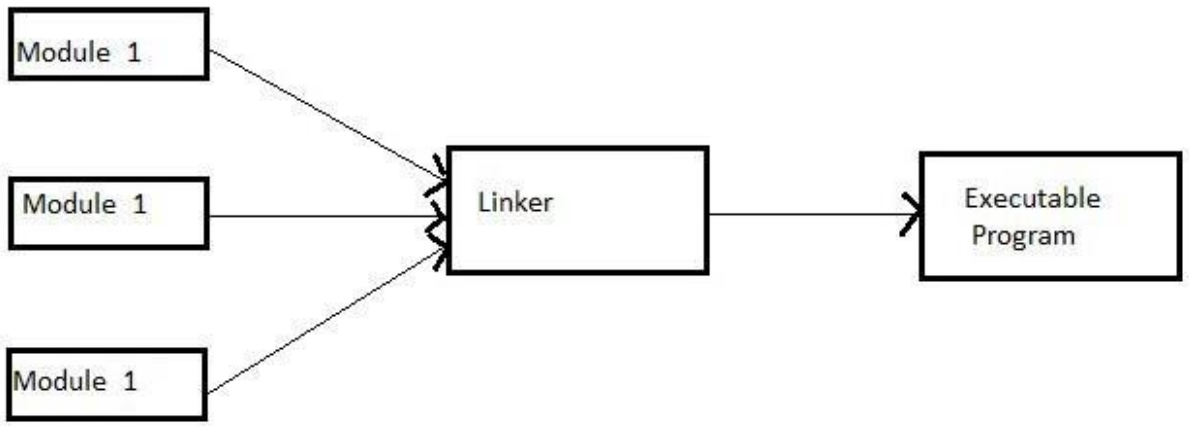
What is interesting is that there are a lot of Microprocessors from many different Vendors - Intel, Motorola, AMD, Spars, etc. - and their Instruction Sets will be radically different also as their history of migration. So, Compiler Writing will be a true challenge to form Compilers consistent and usable for these Microprocessors and their Computer Systems. Keep in Mind if an OS is involved, also as an assortment of proprietary Peripherals, that brings an entirely different level of complexity involved, as a programming language is only written for a selected Microprocessor and not all the opposite hardware features being integrated with it. This is why some Compilers handshake with the low-level OS access points, maybe with the System BIOS, and generic communication protocols with other integrated embedded hardware devices.

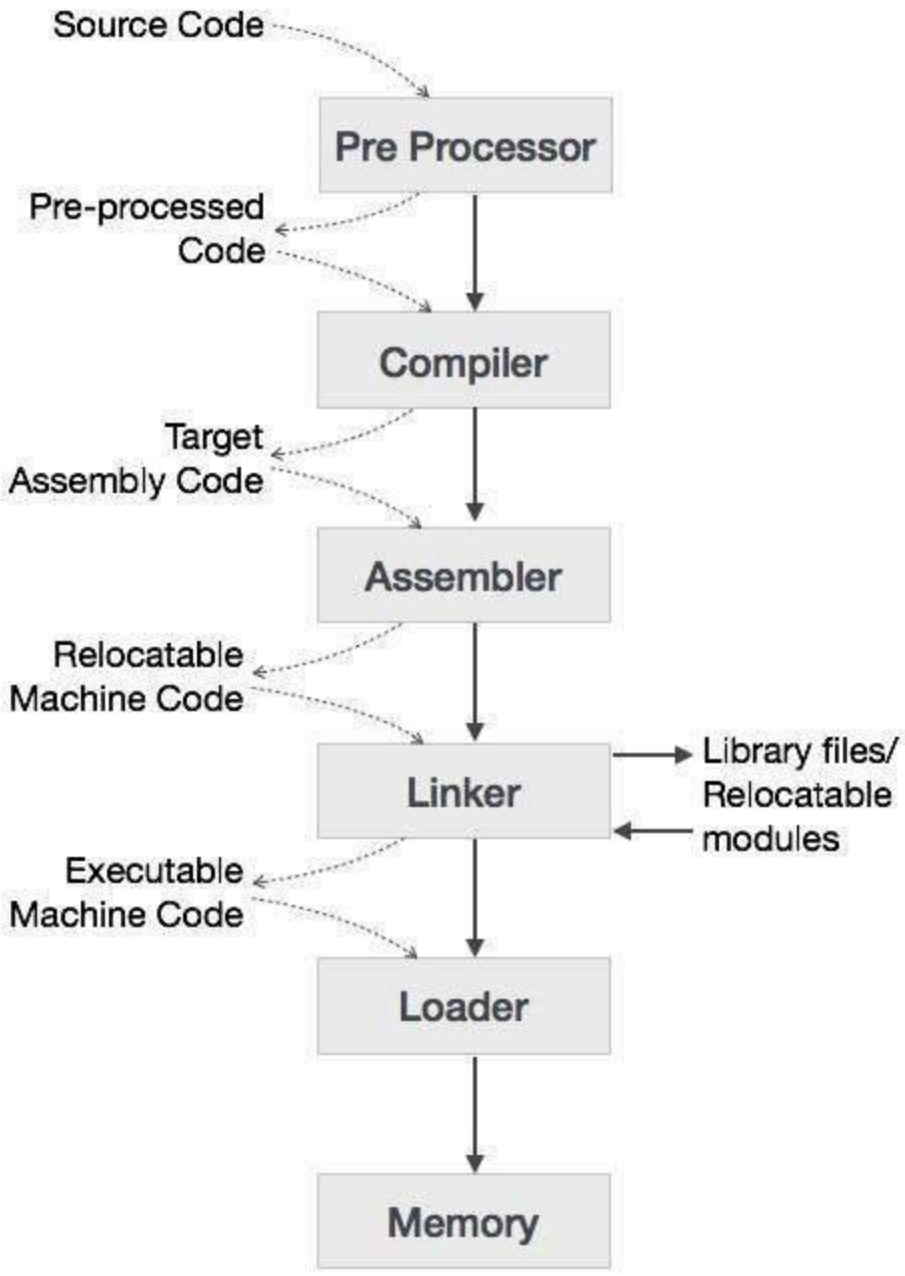


Linker

In computing, a linker is a computer program that takes one or more object files generated by a compiler and combines them into one, executable program.

Computer programs are usually made from multiple modules that span separate object files, each being a compiled computer program. The program as an entire refers to those separately compiled object files using symbols. The linker combines these separate files into one, unified program, resolving the symbolic references as it goes along.




Source Code
Pre Processor
Pre-processed
Code
Compiler
Target
Assembly Code
Assembler
Relocatable
Machine Code
Linker
Library files/
Relocatable
modules
Executable
Machine Code
Loader
Memory