

## ASSEMBLER

An assembler is a program that takes basic computer instructions and converts them into a pattern of bits that the computer's processor can use to perform its basic operations. Some people call these instructions assembler language and others use the term *assembly language*.

- Most computers come with a specified set of very basic instructions that correspond to the basic machine operations that the computer can perform. For example, a "Load" instruction causes the processor to move a string of bits from a location in the processor's memory to a special holding place called a register. Assuming the processor has at least eight registers, each numbered, the following instruction would move the value (string of bits of a certain length) at memory location 3000 into the holding place called register 8: `L 8, 3000`
- The programmer can write a program using a sequence of these assembler instructions.
- This sequence of assembler instructions, known as the source code or source program, is then specified to the assembler program when that program is started.
- The assembler program takes each program statement in the source program and generates a corresponding bit stream or pattern (a series of 0's and 1's of a given length).
- The output of the assembler program is called the object code or object program relative to the input source program.
- The sequence of 0's and 1's that constitute the object program is sometimes called machine code.
- The object program can then be run (or executed) whenever desired.

## ASSEMBLER

Assembler, temel bilgisayar komutlarını alır ve onları bilgisayar işlemcisinin temel işlemlerini yürütebileceği birer bitlik örneklerle dönüştürür. Bazı insanlar bu komutları assembler dili şeklinde adlandırır, bazıları ise assembly dili der.

- Çoğu bilgisayar, bir bilgisayarın yürütebileceği temel makine işlemlerine yanıt veren tanımlanmış ve oldukça temel bir komut kümesiyle birlikte gelir. Örneğin "Load" komutu işlemcinin, bir bit dizisini kendi iç hafızasından alıp register (saklayıcı) olarak adlandırılan özel bir hafıza alanına taşımaya sebep olur. İşlemcinin, her biri numaralanmış sekiz adet saklayıcısı olduğunu varsayarsak aşağıdaki komut, belirli uzunluktaki bir bit dizisini hafızadaki 3000 konumundan saklayıcı 8 olarak adlandırılan konuma taşıyacaktır. `L 8, 3000`
- Programcı, sıralanmış bu assembler komut dizileriyle bir program yazabilir.
- Kaynak kodu ya da kaynak program olarak bilinen bu assembler komut dizileri program başlatıldığı zaman assembler programına tanımlanır.
- Assembler programı kaynak programdaki her bir program ifadesini alır ve karşılık olarak bir bit akışı veya örnekleme üretir ( verilen uzunlukta bir, 0 ve 1'ler serisi).
- Assembler programının çıktısı nesne kodu olarak veya giriş kaynağı programına bağlı nesne programı olarak adlandırılır.
- Nesne programını oluşturan 0 ve 1'ler dizisi bazen makine kodu olarak da adlandırılmaktadır.
- Nesne programı her arzu edildiği zaman çalıştırılabilir veya yürütülebilir.

In the earliest computers, programmers actually wrote programs in machine code, but assembler languages or instruction sets were soon developed to speed up programming. Today, assembler programming is used only where very efficient control over processor operations is needed. It requires knowledge of a particular computer's instruction set, however. Historically, most programs have been written in "higher-level" languages such as COBOL, FORTRAN, PL/I, and C. These languages are easier to learn and faster to write programs with than assembler language. The program that processes the source code written in these languages is called a compiler. Like the assembler, a compiler takes higher-level language statements and reduces them to machine code.

A newer idea in program preparation and portability is the concept of a virtual machine. For example, using the Java programming language, language statements are compiled into a generic form of machine language known as bytecode that can be run by a virtual machine, a kind of theoretical machine that approximates most computer operations. The bytecode can then be sent to any computer platform that has previously downloaded or built in the Java virtual machine. The virtual machine is aware of the specific instruction lengths and other particularities of the platform and ensures that the Java bytecode can run.

İlk bilgisayarlarda programcılar gerçekten de programları makine koduyla yazdılar fakat yakın zaman sonra assembler dilleri veya komut kümeleri hızlanmış programcılığa terfi etti. Bugün assembler programlarla dili sadece işlemci işlemleri üzerinde tam ve etkili bir hâkimiyet gereken yerlerde kullanılmaktadır. Ancak bu belirli bir bilgisayar komut kümesinin bilinmesini gerektirir. Tarihsel olarak birçok program, COBOL, FORTRAN, PL/I, ve C. Gibi "yüksek seviyeli" diller ile yazıldı. Bu dilleri öğrenmek ve program yazmak assembler dilinden daha kolay ve hızlıdır. Bu dillerle yazılan kaynak kodunu işleyen program derleyici olarak adlandırılır. Assembler gibi bir derleyici, yüksek seviyeli dil ifadelerini alır ve onları makine kodu şeklinde özetler.

Program hazırlama ve taşımada daha yeni bir fikir ise sanal bir makine kavramıdır. Örneğin Java programla dili kullanılarak dil ifadeleri, bayt kodu olarak bilinen ve birçok bilgisayar işlemine yakın teorik bir makine olan sanal makine tarafından yürütülebilen genel bir makine dili biçiminde derlenir. Artık bayt kodu, önceden download (indirilmiş) edilmiş veya Java sanal makinesinde oluşturulmuş herhangi bir bilgisayar platformuna gönderilebilir. Sanal makine belirli komutların uzunluklarının ve platformun diğer özelliklerinin farkındadır ve Java bayt kodlarının çalışabilmesini garanti eder.