

A Linguagem Assembly

Assembly é uma linguagem de programação de baixo nível criada em 1970 por Ken Thompson para facilitar o entendimento e a leitura da “linguagem de máquina”.

Cada arquitetura de processador, por exemplo x86 ou ARM, tem um código de máquina diferente com instruções diferentes. Após o trabalho de projetar uma nova arquitetura é terminado, não existem ferramentas como compiladores para essa arquitetura, então é necessário fazer a documentação dessa arquitetura para que os desenvolvedores sejam capazes de estudá-la e projetar ferramentas para as mesmas, ou então, portar ferramentas já existentes para essa arquitetura.

Dentre as diversas informações que podemos encontrar sobre uma arquitetura, existe o ISA que é o conjunto de instruções do código de máquina que aquela arquitetura contém. Porém, o código de máquina é um tanto quanto difícil de ler. Para facilitar o entendimento e memorização pelos desenvolvedores, as instruções recebem nomes e um formato em texto “mnemônico” que facilita a leitura e o entendimento.

Assembly nada mais é que uma forma mais fácil de escrever o código de máquina usando-se de uma notação em formato de texto. Todas as instruções da linguagem Assembly são, na verdade, também instruções do código de máquina. É exatamente pelo fato de Assembly ser uma notação em texto do código de máquina que cada arquitetura diferente tem uma linguagem Assembly completamente diferente, já que, se cada arquitetura tem um código de máquina diferente, consequentemente a sua linguagem Assembly também será diferente.

Então é por isso que Assembly é uma linguagem de baixo nível, já que suas instruções são as mesmas instruções do código de máquina. Há uma correspondência direta com quase nenhuma abstração do código de máquina.

O assembler é o software que converte o código em Assembly para o código de máquina. Existem também adições que o assembler pode fazer para que facilite a programação em Assembly, como por exemplo o pré-processador do NASM que oferece bastante recursos. Isso não quer dizer que seja uma “versão diferente” da linguagem Assembly e nem torna a linguagem “mais alto nível”, isto é apenas um conforto dado pelo assembler.

Programar em Assembly pode trazer desvantagens, como grande desproporção entre o conjunto de instruções e as tarefas que o microprocessador normalmente é chamado a executar. Esta desproporção obriga a decompor manualmente cada tarefa em operações elementares que, além de ser um processo demorado e sujeito a erros, não ajuda a manter o código estruturado. Também pode trazer grande dependência relativamente aos detalhes internos do hardware que tem como consequência direta reduzir a portabilidade dos programas.

A principal vantagem de programar em assembly é poder aproveitar ao máximo as características particulares do hardware onde vai ser executado, conseguindo assim resultados otimizados em tempo de execução e tamanho de código. Outra vantagem é a existência de assembladores gratuitos ou de baixo custo para a totalidade dos microprocessadores, sendo sempre possível programar em assembly, qualquer que seja o microprocessador escolhido. O mesmo já não acontece com linguagens de alto nível, onde nem sempre é possível encontrar um compilador para um dado microprocessador.