

1.1.1 Linguagens, níveis e máquinas virtuais

O texto discute duas abordagens para lidar com a criação de um conjunto de instruções mais conveniente para as pessoas usarem em computadores. A primeira abordagem é a tradução, onde um programa escrito na nova linguagem (chamada de L1) é convertido em uma sequência equivalente de instruções na linguagem de máquina existente (chamada de linguagem de máquina M1). O programa resultante é então executado pelo computador na linguagem M1. A segunda abordagem é a interpretação, onde um programa em M1 é escrito para interpretar e executar os programas escritos na linguagem L1. Essas abordagens podem ser combinadas.

O texto também sugere a possibilidade de inventar um terceiro conjunto de instruções, chamado de L2, que seja mais orientado para pessoas do que para máquinas. Os programas escritos em L2 podem ser traduzidos para a linguagem L1 ou executados por um interpretador escrito em L1. O objetivo é criar uma série de linguagens sucessivas, cada uma mais conveniente do que sua antecessora, até que uma linguagem adequada seja alcançada.

Em resumo, o texto discute diferentes abordagens, como tradução e interpretação, para lidar com a criação de um conjunto de instruções mais conveniente para as pessoas usarem em computadores. Também sugere a possibilidade de inventar linguagens sucessivas, cada uma mais adequada, em uma série de níveis ou camadas.

O texto afirma que há uma relação importante entre uma linguagem e uma máquina virtual. Cada máquina tem sua própria linguagem de máquina, que consiste em todas as instruções que ela pode executar. Da mesma forma, uma linguagem define uma máquina, ou seja, aquela que pode executar todos os programas escritos nessa linguagem.

Embora seja possível imaginar a construção de uma máquina que execute linguagens como C, C++ ou Java, isso não seria econômico em comparação com outras técnicas disponíveis atualmente. Um projeto prático precisa ser tanto factível quanto econômico.

Em um computador com vários níveis, cada nível pode ser visto como uma máquina virtual com uma linguagem de máquina diferente. Os termos "nível" e "máquina virtual" são usados de forma intercambiável. Apenas programas escritos na linguagem de máquina mais baixa (LO) podem ser executados diretamente pelos circuitos eletrônicos, sem a necessidade de tradução ou interpretação. Os programas escritos em linguagens de nível mais alto (L1, L2, etc.) devem ser interpretados por um interpretador em um nível inferior ou traduzidos para outra linguagem correspondente a um nível inferior.

Um programador que escreve programas para uma máquina virtual de determinado nível não precisa se preocupar com os interpretadores e tradutores subjacentes. A estrutura da máquina garante que esses programas devem executados de alguma forma, independentemente de serem interpretados passo a passo por interpretadores ou executados diretamente pelos circuitos eletrônicos. O resultado é o mesmo: os programas são executados.

A maioria dos programadores que usam uma máquina de determinado nível está interessada apenas no nível mais alto, que é o que difere mais da linguagem de máquina do nível mais baixo.

1.1.2 Evolução de máquinas multiníveis

O texto discute o desenvolvimento histórico das máquinas multiníveis, destacando a evolução do número e da natureza desses níveis ao longo dos anos. No nível 0, os programas escritos em linguagem de máquina podem ser executados diretamente pelos circuitos eletrônicos do computador, sem a necessidade de interpretadores ou tradutores. O hardware do computador é composto por tais circuitos eletrônicos, memória e dispositivos de entrada/saída.

Por outro lado, o software consiste em algoritmos e programas, que são representações das instruções detalhadas para realizar determinadas tarefas. O software pode ser armazenado em diferentes meios físicos, como disco rígido ou CD-ROM, mas sua essência está nas instruções dos programas, não no meio físico em que estão armazenadas.

No passado, a distinção entre hardware e software era clara, mas ao longo do tempo, essa fronteira se tornou indistinta devido à adição, remoção e fusão de níveis à medida que os computadores evoluíram. Atualmente, é difícil distinguir claramente essa fronteira. O livro enfatiza o tema de que hardware e software são logicamente equivalentes.

Todas as operações realizadas por software podem ser incorporadas diretamente no hardware, uma vez que sejam suficientemente compreendidas. Da mesma forma, qualquer instrução executada em hardware pode ser simulada em software. A decisão de implementar funções no hardware ou no software é baseada em fatores como custo, velocidade, confiabilidade e frequência de mudanças esperadas. Não há regras rígidas e imutáveis para determinar o que deve ser implementado no hardware ou no software, pois essas decisões variam de acordo com as tendências econômicas, demanda e uso dos computadores.

A invenção da microprogramação trouxe uma simplificação significativa aos primeiros computadores digitais da década de 1940. Inicialmente, esses computadores tinham apenas dois níveis: o nível ISA, onde a programação era realizada, e o nível lógico digital, que executava os programas. No entanto, os

circuitos do nível lógico digital eram complicados, difíceis de entender, montar e pouco confiáveis.

Em 1951, Maurice Wilkes, pesquisador da Universidade de Cambridge, propôs a ideia de um computador de três níveis para simplificar o hardware e reduzir o número de válvulas necessárias. Esse computador teria um interpretador embutido, chamado de microprograma, que executaria programas do nível ISA por meio de interpretação. Essa abordagem reduziria o número de circuitos eletrônicos necessários, uma vez que o hardware só teria que executar microprogramas com um conjunto limitado de instruções, em vez de programas do nível ISA com conjuntos de instruções muito maiores. Isso prometia aumentar a confiabilidade dos circuitos eletrônicos.

Durante a década de 1950, poucas máquinas de três níveis foram construídas, mas a ideia ganhou popularidade na década de 1960. Por volta de 1970, a interpretação do nível ISA por meio de microprograma se tornou dominante, sendo utilizada em todas as principais máquinas da época.

Outro avanço importante foi a invenção do sistema operacional. Nos primeiros anos dos computadores, o programador precisava operar a máquina pessoalmente, reservando um período de tempo e levando cartões perfurados com o programa. Era um processo demorado e propenso a erros. Para automatizar esse trabalho e reduzir o tempo de espera, foram desenvolvidos sistemas operacionais que permitiam que vários programadores se comunicassem diretamente com o computador. Esses sistemas compartilhados, conhecidos como sistemas de tempo compartilhado ou time sharing, tinham terminais remotos conectados ao computador central por linhas telefônicas.

Com o surgimento da microprogramação, os projetistas perceberam que poderiam adicionar novas instruções ao microprograma, ampliando assim as capacidades do hardware. Isso levou a um aumento significativo nos conjuntos de instruções de máquina, com projetistas competindo para produzir conjuntos maiores e melhores. Além disso, outras características e facilidades foram adicionadas aos microprogramas para acelerar cálculos, permitir a movimentação de programas na memória, lidar com interrupções, entre outras funcionalidades.

No entanto, com o tempo, os microprogramas ficaram mais volumosos e lentos. Alguns pesquisadores perceberam que eliminando o microprograma e executando as instruções diretamente no hardware, as máquinas poderiam ser mais rápidas. Isso marcou um retorno ao projeto de computadores anterior à invenção da microprogramação.

É importante destacar que, mesmo com a eliminação da microprogramação, os processadores modernos ainda usam microcódigo interno para traduzir instruções complexas e executá-las diretamente no hardware prepar