

Síntese da Aula 09 e 10/11

A arquitetura de basicamente todos os computadores atuais baseia-se na arquitetura de Von Neumann, e isso é verdade desde os primeiros computadores eletrônicos, como o INIAC. A arquitetura de Von Neumann se baseia na ideia de separação da memória, processamento, controle e I/O.

Com o advento de novas tecnologias, especialmente de armazenamento, foi possível a criação de máquinas muito mais complexas, o que possibilitou a existência dos nossos atuais computadores, com interfaces de usuário gráficas e outras comodidades modernas. Nessas máquinas modernas existem diversas formas diferentes de memória, cada uma de capacidades e velocidades diferentes. Quanto mais rápida a memória, mais cara, portanto, menos quantidade temos dentro das máquinas. Hoje, em um computador, temos memórias persistentes, como HDs e SSDs e também memórias voláteis, como a memória RAM e as Cachês L3, L2 e L1, da memória mais lenta para a mais rápida, respectivamente. É necessário também notar a necessidade da sincronia de velocidades entre memórias e barramentos de mesmo padrão. O computador irá sempre trabalhar na velocidade do componente mais lento para evitar problemas de sincronia de barramento.

Por fim temos o avanço das linguagens de programação. Em um primeiro momento toda programação deveria ser feita em linguagem de máquina, código binário, o que tornava o ato de programar extremamente complicado, tedioso e desgastante. Com a criação de compiladores, programas que traduzem códigos mais próximos da língua humana para código de máquina, essa programação se tornou muito mais simples, e com isso surgiram as linguagens de alto-nível. Essas linguagens trabalham em algo próximo da língua humana (majoritariamente em inglês), e são mais muito fáceis de serem compreendidas e desenvolvidas. Elas variam em complexidade, mas todas são um passo gigantesco em relação à linguagem de máquina. Essas linguagens também são um dos principais motivos que permitiram a existência dos sistemas de informação modernos. Pode-se interpretar, em termos de complexidade, uma cadeia que inicia-se em linguagens de alto nível, como JAVA, C, Python, passa-se pela linguagem Assembly, uma espécie de simplificação dos comandos de máquina, e, através do Assembler, chega-se à linguagem de máquina, ou código binário.