

## AULA(10-12): MÁQUINAS VIRTUAIS E CONTAINERS

A aula foi dedicada ao estudo dos conceitos fundamentais de virtualização e suas aplicações na computação moderna. No início, discutiu-se como a virtualização atua como uma camada entre a aplicação e o sistema operacional, permitindo maior flexibilidade e abstração no uso dos recursos de hardware. A partir disso, a aula avançou para a diferenciação entre máquinas virtuais e containers, bem como o papel crucial dos hypervisors. Por fim, estudamos também os conceitos de emulação, suas aplicações e a utilização do Bochs para emular arquiteturas x86. Além disso, o professor transmitiu o vídeo [Software Emulators vs FPGAs](#) do canal “What’s Ken Making”

Abaixo, seguem algumas das definições e explicações apresentadas durante a aula:

**Virtualização:** a virtualização consiste em criar uma camada intermediária entre a aplicação e o sistema operacional em conjunto com o hardware. Essa camada cria uma abstração que permite ao software interagir com os recursos como se tivesse controle dedicado sobre eles. A principal vantagem é permitir isolamento, portabilidade e melhor aproveitamento dos recursos físicos, possibilitando que várias aplicações ou sistemas funcionem simultaneamente no mesmo hardware.

**Máquinas Virtuais:** uma máquina virtual leva a abordagem de camadas ao seu limite, tratando tanto o hardware quanto o kernel do sistema operacional como elementos virtualizados. A interface fornecida ao convidado é idêntica ao hardware real, permitindo que cada sistema operacional convidado tenha a ilusão de possuir seu próprio processador, memória e dispositivos. Cada máquina virtual recebe sua própria cópia virtual do computador, funcionando de forma isolada e independente. Sistemas como o Java também utilizam máquinas virtuais, como a JVM, para garantir a portabilidade do código.

**Hypervisor:** o hypervisor é a implementação mais próxima do hardware e responsável por gerenciar o ambiente virtualizado. Ele pode estar abaixo do sistema operacional (no caso dos hypervisores do tipo *bare-metal*), possibilitando que múltiplos sistemas operacionais diferentes rodem simultaneamente no mesmo hardware. Essa camada distribui os recursos físicos entre os sistemas convidados, garantindo isolamento e eficiência no uso dos recursos.

**Containers:** ao contrário das máquinas virtuais, containers não possuem um sistema operacional próprio. Todos os containers utilizam o mesmo kernel do sistema hospedeiro, o que os torna muito mais leves e rápidos. O motor de containers, como Docker ou Kubernetes, é responsável por gerenciar esses ambientes isolados. Containers são amplamente utilizados por fornecerem isolamento eficiente com baixo custo de recursos, oferecendo uma alternativa mais otimizada em muitos cenários que antes eram dominados por máquinas virtuais.

**Diferença entre VM e Container:** máquinas virtuais possuem um sistema operacional completo para cada convidado, consumindo mais recursos e oferecendo maior isolamento. Containers, por compartilharem o kernel do hospedeiro, são mais leves, rápidos e eficientes, mas oferecem um nível de isolamento mais superficial quando comparados às VMs.

**Emulação:** a emulação consiste em simular via software o comportamento de um hardware ou sistema operacional. Emuladores executam por cima do sistema operacional e muitas vezes são escritos em linguagens de alto nível. Eles precisam reproduzir componentes fundamentais do sistema, como entrada e saída, controladores e até instruções da CPU. A emulação pode ser também de hardware, reproduzindo arquiteturas distintas como x86 ou ARM. Na aula, exploramos especialmente o uso do Bochs para emular uma máquina x86, necessária para a execução do kernel que será desenvolvido na disciplina.

## **CONCLUSÃO**

A partir dos tópicos estudados, torna-se evidente que a virtualização é uma das tecnologias centrais da computação moderna, permitindo flexibilidade, segurança e melhor aproveitamento dos recursos físicos. A compreensão clara das diferenças entre máquinas virtuais, containers e emuladores é essencial para atuar em áreas como desenvolvimento de sistemas operacionais, administração de servidores e computação em nuvem. Além disso, recursos como hypervisors e motores de containers demonstram como a abstração de hardware molda a forma como sistemas são executados atualmente. Em síntese, o domínio desses conceitos amplia significativamente a capacidade do estudante de entender e criar soluções computacionais mais eficientes, modernas e portáteis.

**Aluno:** João Victor Oliveira

**Matrícula:** 20240008468