

## **Atividade 09 – Pesquisa**

### **1. Defina e Cite Exemplos de Sistemas monolíticos**

A arquitetura monolítica é a arquitetura de sistema operacional mais antiga e mais comum conhecida. Ela se caracteriza pelo fato de cada componente do sistema operacional ser contido no núcleo (kernel) e pode comunicar-se diretamente com qualquer outro componente (utilizando chamadas à função), justamente por isso o núcleo normalmente tem acesso irrestrito ao sistema de computador.

No núcleo/kernel monolítico, o sistema operacional é escrito como se fosse um conjunto de rotinas, de forma que cada rotina pode chamar, ou ainda, se comunicar, com outra rotina, sempre que for necessário. Esse sistema possui um núcleo grande e complexo que engloba todos os serviços. Todos os componentes funcionais do kernel têm acesso a todas as suas estruturas de dados e rotinas internas, com isso, um erro em uma rotina pode comprometer todo o núcleo.

O sistema monolítico é estruturado em um único arquivo binário, em apenas um processo que executa inteiramente em modo protegido.

Atualmente, apenas sistemas operacionais embutidos usam essa arquitetura, devido às limitações do hardware sobre o qual executam.

Exemplos: Versões Mac OS abaixo do 8.6, Windows 9x.

### **2. Defina e Cite Exemplos de Sistemas micronúcleo;**

Micronúcleo, ou microkernel, é uma arquitetura de núcleo (kernel) de um sistema operativo cujas funcionalidades são quase todas executadas fora do núcleo, em oposição a um núcleo monolítico. Os processos se comunicam com um núcleo mínimo, usando o mínimo possível o "espaço do sistema" (kernel space). Neste local os aplicativos tem acesso a todas as instruções e a todo o hardware e deixando o máximo de recursos rodando no "espaço do usuário" (user-space) em que o software tem algumas restrições, não podendo acessar algumas hardwares, nem tem acesso a todas as instruções).

Basicamente o Microkernel executa a maioria dos processos fora do kernel, ele carrega o mínimo de processos possíveis no kernel space. O restante dos módulos são executados como processos de usuário comuns para o sistema.

Um dos principais benefícios de utilizar a arquitetura dessa maneira, com divisão de tarefas entre módulos, é uma menor preocupação com os erros, já que sistemas operacionais são muito sujeitos a eles. Isso não quer dizer que erros não vão acontecer, porém, erros que provavelmente causariam uma falha e derrubariam todo o sistema, agora irão apenas causar uma falha naquele módulo em específico, trazendo alta confiabilidade.

Exemplos: Hurd, MINIX, QNX e L4

### **3. Defina e Cite Exemplos de Sistemas em camadas;**

O sistema operacional é organizado em camadas construídas uma sobre a outra. O primeiro sistema construído dessa maneira foi o sistema criado no Technische Hogeschool Eindhoven, na Holanda, por E. W. Dijkstra (1968) e seus alunos. O sistema THE era um sistema de lote simples para um computador holandês, o Electrologica X8, que tinha 32K de palavras de 27 bits (bits eram caros naquela época).

O sistema tinha seis camadas:

- A camada 0 lidava com a alocação do processador, alternando entre processos quando ocorriam interrupções ou quando os temporizadores expiravam, essa camada proporcionava a multiprogramação básica da CPU.
- A camada 1 fazia o gerenciamento da memória. Ela alocava espaço para os processos da memória principal e em um tambor (Antigo meio magnético de armazenamento de dados) com 512K de palavras utilizado para armazenar partes do processo (páginas) para os quais não havia lugar na memória principal. Acima da camada 1, os processos não tinham que se preocupar com o fato de eles estarem em memória ou no tambor, o software da camada 1 cuidava de assegurar que as páginas fossem levadas para a memória sempre que fossem necessárias.
- Já a camada 2 fazia a comunicação entre o console do operador e cada processo.
- A camada 3 gerenciava dispositivos de entrada e saída.
- A camada 4 localizavam-se os programas de usuários.
- A camada 5 era o usuário.

No entanto, a maioria dos sistemas de uso geral usa apenas duas camadas, mesmo que o hardware em que são executados forneça mais modos de CPU do que isso.

Exemplos: Windows 7 e o Windows Server 2008.

### **4. Defina e Cite Exemplos de Máquinas Virtuais;**

Uma máquina virtual (VM) ou virtual machine é um ambiente virtualizado que funciona como um sistema de computação com sua própria CPU, memória, interface de rede e armazenamento. Esse sistema virtual é criado a partir de um sistema de hardware físico localizado on-premise ou não. Um software chamado hipervisor separa do hardware os recursos utilizados pela máquina virtual e os provisiona adequadamente.

A máquina física, onde o hipervisor, como a máquina virtual baseada em Kernel (KVM), está instalada é chamada de host. As VMs que usam os recursos da máquina host são chamadas de guest. O hipervisor trata os recursos de computação (por exemplo, CPU, memória e armazenamento) como um pool que pode ser realocado com facilidade entre os guests existentes ou para novas máquinas virtuais.

Exemplos: virtualização de servidores, o qual particiona um servidor físico em vários servidores virtuais, VDI (Virtualização de Desktop), onde apenas um desktop gerencia vários outros desktops simultaneamente, virtualização de SOs, o qual foca apenas em abstrair o SO.

## **5. Defina e Cite Exemplos de Sistemas de Contêineres;**

Container nada mais é do que um ambiente isolado contido em um servidor que, diferentemente das máquinas virtuais, divide um único host de controle.

Esse isolamento de container, possibilita uma utilização limitada do HD, memória RAM e processador. Ao utilizar um tipo de compartilhamento de kernel, os containers apresentam uma capacidade de economia de recursos maior do que as máquinas virtuais.

Exemplos: os mais conhecidos são o OpenShift e o Kubernetes. O Docker também tem um gerenciador próprio, chamado de Docker Swarm.