



# ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

## Sistemas operacionais modernos

*Andrew S. Tanenbaum*

Adriano da Silva Rosa - BP300838X

Matheus Felipe Zacarias - BP3008461

Pedro Guilherme Calasans de Souza - BP3008517

6 de março de 2020

## RESENHA

Uma das principais tarefas dos sistemas operacionais é esconder o hardware e em vez disso apresentar programas com abstrações de qualidade, limpas, elegantes e consistentes com as quais trabalhar. O Sistema Operacional é um software que opera em modo núcleo.

A arquitetura dos computadores em nível de linguagem de máquina é primitiva e complicada de programar. Por essa razão, todos os sistemas operacionais fornecem níveis de abstração. Boas abstrações transformam uma tarefa impossível em duas tarefas gerenciáveis. A primeira é definir e implementar as abstrações, e a segunda é utilizá-las para solucionar o problema à mão. A função dos S.O. é criar boas abstrações e então implementar e gerenciar os objetos abstratos criados desse modo. Além de fornecer uma alocação ordenada e controlada dos processadores, memórias e dispositivos de E/S entre os vários programas competindo por eles. A principal função é manter um controle sobre quais programas estão usando qual recurso, conceder recursos requisitados, contabilizar o seu uso, assim como mediar requisições conflitantes de diferentes programas e usuários. O gerenciamento de recursos inclui a multiplexação de recursos de duas maneiras diferentes. No tempo diferentes programas ou usuários se revezam usando-o; e no espaço em vez de os clientes se revezarem, cada um tem direito a uma parte do recurso.

No início, um único grupo de pessoas projetava, construía, programava, operava e mantinha cada máquina. Toda a programação era feita em código de máquina, ligando circuitos elétricos através de conexão de milhares de cabos a painéis de ligações para controlar as funções básicas da máquina. Com o tempo os computadores tornaram-se menores, confiáveis, manufaturados e houve a separação entre projetistas, construtores, operadores, programadores e pessoal da manutenção.

A multiprogramação divide a memória em várias partes, com uma tarefa diferente em cada partição. Enquanto uma tarefa ficava esperando pelo término da E/S, outra podia usar a CPU. Se um número suficiente de tarefas pudesse ser armazenado na memória principal ao mesmo tempo, a CPU podia se manter ocupada quase 100% do tempo. Spooling é sempre que uma tarefa sendo executada terminava, o sistema operacional podia carregar uma nova tarefa do disco para a partição agora vazia e executá-la. Timesharing é uma variante da multiprogramação na qual cada usuário tem um terminal on-line. O CTSS foi o primeiro sistema de compartilhamento de tempo para fins diversos, desenvolvido no M.I.T. em um 7094 modificado. O MULTICS é um computador utilitário que daria suporte a algumas centenas de usuários simultâneos com compartilhamento de tempo. O conceito de cloud computing é de que computadores relativamente pequenos estejam conectados a servidores em vastos e distantes centros de processamento de dados onde toda a computação é feita com o computador local apenas executando a interface com o usuário.

O conjunto de chamadas de sistema é o coração dos sistemas operacionais e o principal diferencial entre eles. Essas chamadas dizem o que o sistema operacional realmente faz. Sistemas operacionais são antigos, seu objetivo inicial era substituir operadores mas hoje já está além da multiprogramação moderna. Teve o nascimento do UNIX, que se tornou popular no mundo acadêmico, em agências do governo e em muitas empresas. Que deu origem a outros sistemas operacionais como o FreeBSD e Li-

nux. No UNIX as chamadas de sistema são divididos em quatro grupos: O primeiro grupo diz respeito à criação e ao término de processos. O segundo é para leitura e escrita de arquivos. O terceiro é para o gerenciamento de diretórios. E o quarto grupo contém chamadas diversas. Podendo ser estruturado de diversas maneiras os S.O. mais comuns são a hierarquia de camadas, o exonúcleo, o micronúcleo, o cliente-servidor, a máquina virtual e o sistema monolítico.

## ATIVIDADES

### 1 Quais as principais funções de um Sistema Operacional ?

Gerenciar recursos: O S.O. deve gerenciar a utilização dos recursos fornecidos pelo hardware, como processadores, memória, dispositivos de E/S, de modo que mantenha o controle sobre qual usuário/programa utiliza qual recurso, compartilhando os recursos entre os usuários/programas de modo seguro e sem conflitos.

Estender a máquina: O S.O. deve oferecer ao usuário uma maneira mais acessível de programar/utilizar o hardware do que as próprias instruções que este oferece. por exemplo: o usuário não precisa saber qual a trilha e o setor do disco se deseja gravar alguma informação, apenas faz a chamada ao sistema que estende as instruções de E/S., disponibilizando instruções mais amigáveis para estas e outras tarefas. Ou seja, o S.O. atua como uma interface entre o hardware e o ambiente de software.

### 2 Qual é a diferença entre sistemas de compartilhamento de tempo e de multiprogramação?

Compartilhamento de tempo: Possibilita que vários programas sejam executados através de uma divisão de tempo do processador em intervalos curtos.

Multiprogramação ou multitarefa: Possibilitam que enquanto um programa esteja em operação de leitura outros possam ser executados.

### 3 A idéia de família de computadores foi introduzida na década de 1960 com os computadores de grande porte System/360 da IBM. Essa idéia está ultrapassada ou ainda é válida?

Essa idéia continua válida, uma vez que fornece, por utilizar o mesmo conjunto de instruções para uma série de sistemas, algumas vantagens como: Portabilidade para diversos softwares, inclusive antigos; e uma maior variedade de opções para compra, que permite melhores escolhas e escalabilidade dentro de um sistema.

4 A chamada `count = write(fd,buffer,nbytes);` pode retornar qualquer valor `count` fora `nbyter`? Se a resposta for sim, por quê?

Sim, essa chamada pode retornar algum valor diferente de `nbyter`.

Principalmente por dois motivos, se um final de arquivo for encontrado antes, ou o arquivo não pôde ser lido. se a chamada falhar, por exemplo, porque `fd` está incorreto, ele pode retornar `-1`. Ele também pode falhar porque o disco está cheio e não é possível gravar o número de bytes solicitados. Em uma terminação correta, sempre retorna `nbytes`.