

Sistemas Operacionais

Prof.: Boanerges Teixeira



1. Conceitos Básicos de SO

Tipos de sistemas operacionais

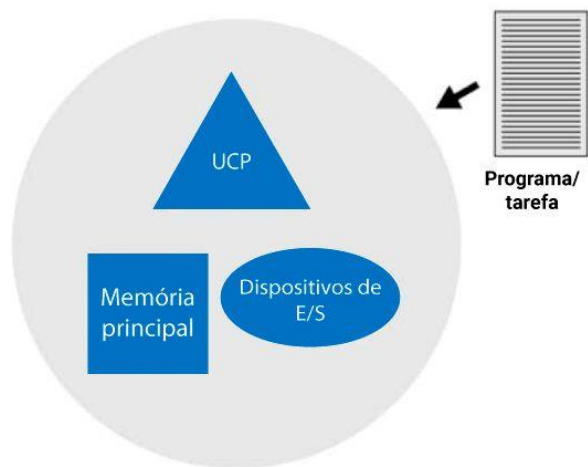
Cada sistema operacional possui características únicas que influenciam diretamente o seu desempenho e eficiência. A classificação em sistemas monoprogramáveis, multiprogramáveis e com múltiplos processadores ajuda a entender como os recursos de hardware são gerenciados e otimizados.



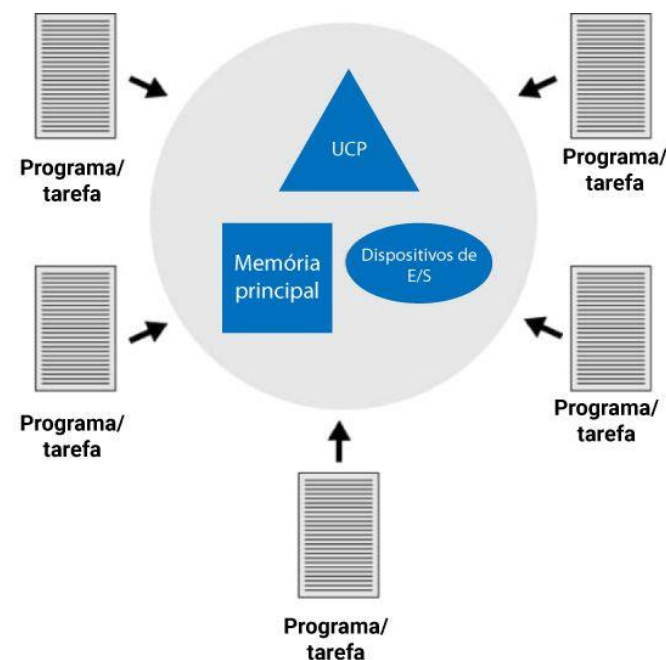
1. Conceitos Básicos de SO

Tipos de sistemas operacionais

Os **sistemas monoprogramáveis** também são chamados de **sistemas monotarefa**. Nesses sistemas, os principais módulos computacionais (processador, memória e periféricos) ficam alocados exclusivamente para a execução de um único programa, sendo que qualquer outra aplicação deve esperar para poder ser processada.



Nos **sistemas multiprogramáveis** ou **multitarefa**, os recursos computacionais passam a ser compartilhados entre usuários e aplicações, permitindo que muitas aplicações compartilhem os mesmos recursos.



1. Conceitos Básicos de SO

Tipos de sistemas operacionais

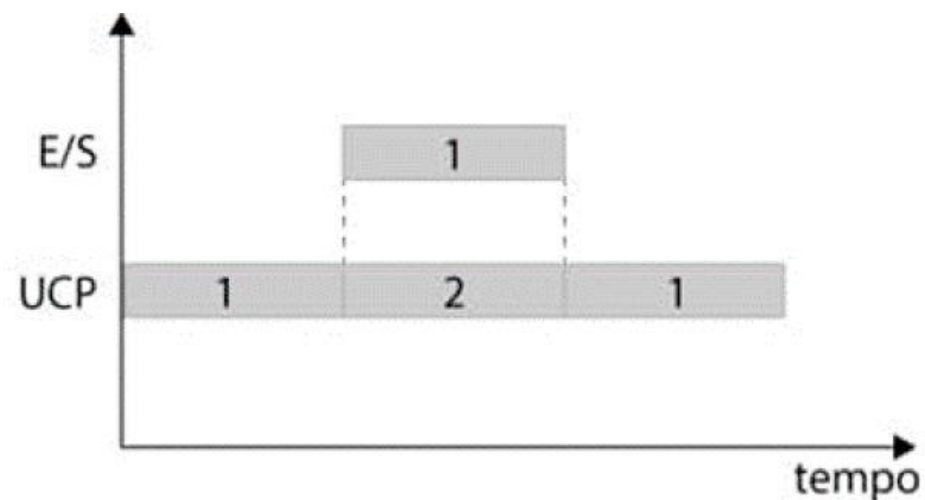
Monoprogramável

Neste sistema, enquanto uma leitura em disco era realizada, o processador permanecia ocioso, e o tempo de espera até que a leitura em disco fosse realizada era relativamente longo, pois as operações com dispositivos de E/S são muito lentas quando comparadas com a velocidade do processador para a execução de instruções.



Multiprogramável

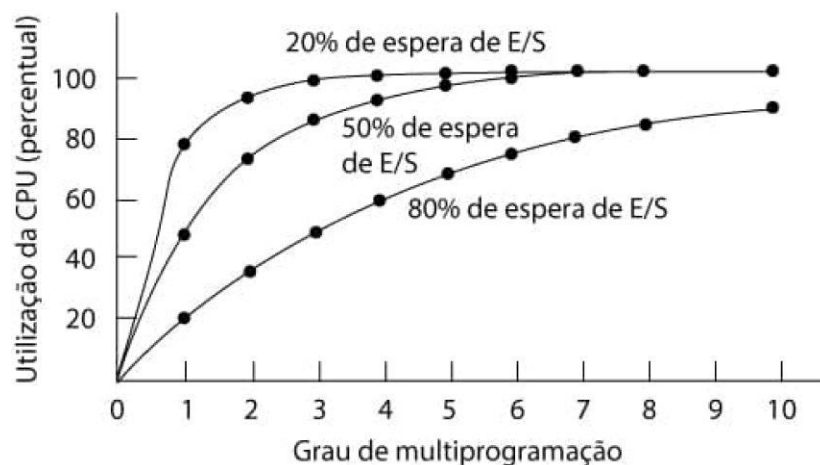
Já neste sistema, o processador é utilizado por diversos programas, ou processos, de forma concorrente, ou seja, quando um processo tem que fazer uma operação de E/S, outro processo pode utilizar o processador.



1. Conceitos Básicos de SO

Tipos de sistemas operacionais

A utilização da CPU como função da quantidade de processos carregados na memória simultaneamente é chamada grau de multiprogramação. A imagem a seguir apresenta um gráfico representando a utilização da CPU *versus* o grau de multiprogramação



A utilização da CPU é dada pela fórmula $U_{cpu} = 1 - p^n$, onde:

- p = Tempo de espera de (dispositivos de) E/S.

- n = Número de processos carregados na memória.

Por exemplo: Se $p = 65\%$ e $n = 3 \rightarrow U_{cpu} = 1 - 0,65^3 = 0,72$ ou 72%.

1. Conceitos Básicos de SO

Tipos de sistemas operacionais

Por fim, os **sistemas com múltiplos processadores** possuem dois ou mais processadores atuando juntos, oferecendo vantagens como:

Escalabilidade

Aumenta a capacidade de processamento.

Disponibilidade

Se um processador falhar, outro pode assumir a carga de trabalho.

Balanceamento de carga

Distribui a carga de trabalho entre os processadores.

Em sistemas com múltiplos processadores, os processadores podem se comunicar de duas maneiras:

- *nos sistemas fortemente acoplados*
- *nos sistemas fracamente acoplados*

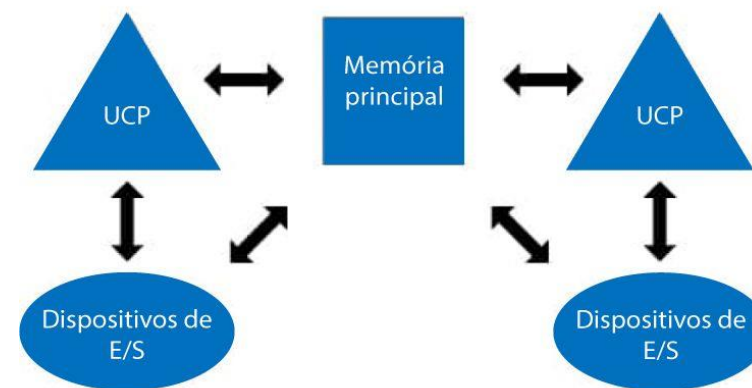
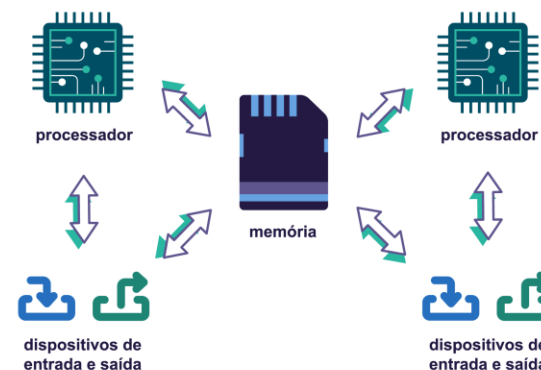
1. Conceitos Básicos de SO

Tipos de sistemas operacionais

Sistemas fortemente acoplados são sistemas de computação com vários processadores que compartilham uma memória física e dispositivos de entrada e saída. São também conhecidos como multiprocessadores.

Características:

- São máquinas de alto custo e de grande velocidade de processamento
- Possuem uma única memória física compartilhada por todos os processadores
- São gerenciados por um único sistema operacional
- Possuem uma fila de execução e um único sistema de arquivos mantidos em um espaço de memória compartilhada



1. Conceitos Básicos de SO

Tipos de sistemas operacionais

Os sistemas fortemente acoplados ainda podem ser divididos em:

SMP - Symmetric Multiprocessors



Os processadores acessam a memória uniformemente ao longo do tempo.

NUMA - Non-Uniform Memory Access



Como existem conjuntos de processadores e memória principal, e um conjunto se conecta aos outros por meio de uma rede de interconexão, o tempo de acesso dos processadores à memória varia conforme a localização física.

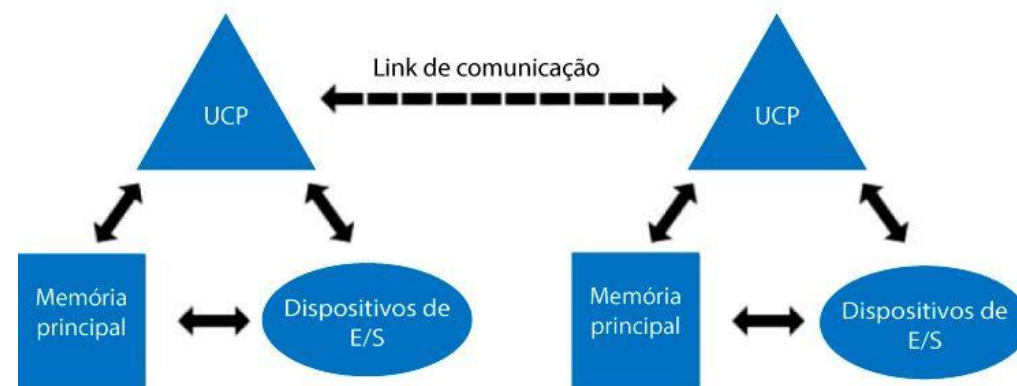
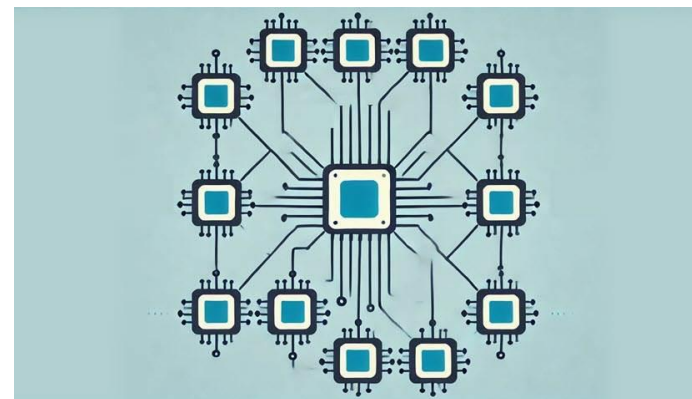
1. Conceitos Básicos de SO

Tipos de sistemas operacionais

Sistemas fracamente acoplados são sistemas em que os componentes operam de forma independente, com pouca interdependência entre si.

Características

- Cada componente tem pouca ou nenhuma informação sobre os outros componentes
- Cada sistema tem a sua própria memória individual
- Os processadores podem estar em arquiteturas diferentes
- Os sistemas podem ser interligados por uma rede de computadores



1. Conceitos Básicos de SO

Tipos de sistemas operacionais

Sistemas em lote

O sistema em lote (batch) processa tarefas de rotina sem a presença interativa do usuário. Por exemplo: processamento de apólices de companhia de seguro; relatório de vendas de uma cadeia de lojas.



Sistemas de processamento de transações

Os sistemas de processamento de transações administram grandes quantidades de pequenas requisições. Cada unidade de trabalho é pequena, mas o sistema precisa tratar centenas ou milhares delas por segundo.



Por exemplo: processamento de verificações em um banco ou em reservas de passagens aéreas.

Sistemas de tempo compartilhado

Os sistemas de tempo compartilhado permitem que múltiplos usuários remotos executem suas tarefas simultaneamente no computador. Por exemplo: realização de consultas a um banco de dados.



1. Conceitos Básicos de SO

Tipos de sistemas operacionais

Os serviços de **sistemas de tempo real** são aqueles que possuem o **tempo** como parâmetro fundamental. Podemos citar, como exemplo, a linha de montagem de um carro. Esses sistemas podem ser de dois tipos:

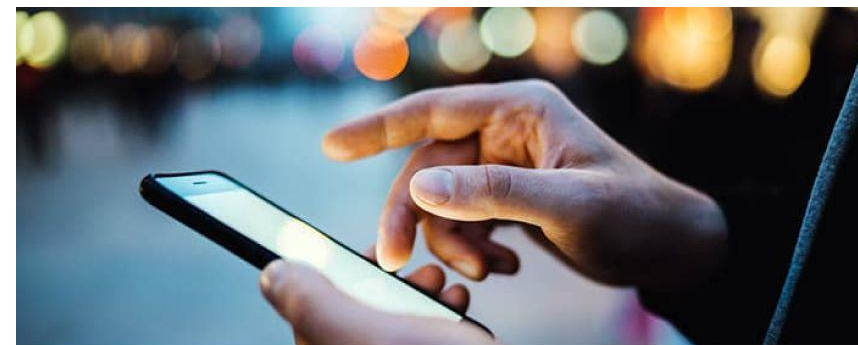
Sistema de tempo real crítico

Neste sistema, as ações precisam ocorrer em determinados instantes. Como exemplos, podemos citar processos industriais, de aviação e militares.



Sistema de tempo real não crítico

Neste tipo de sistema, o descumprimento de um prazo não causa dano permanente. Como exemplos, podemos citar sistema de áudio digital, multimídia, telefones digitais.



1. Conceitos Básicos de SO

Tipos de sistemas operacionais

Os sistemas operacionais variam quanto aos tipos de licenças, ou seja, o conjunto de ações que o usuário do SO pode ou não fazer.

Software proprietário



- Licenciado sob direitos legais exclusivos – copyright.
- Usualmente o código-fonte total ou parcial não é disponibilizado para modificação por qualquer pessoa, apenas a fabricante possui o acesso.

1. Conceitos Básicos de SO

Tipos de sistemas operacionais

Software livre (Free software)

Software livre se refere à **liberdade** dos usuários executarem, copiarem, distribuírem, estudarem, modificarem e aperfeiçoarem o software.

Por exemplo, a licença GNU da Free Software Foundation (FSF) possui as seguintes liberdades:

- Liberdade nº 0: Executar o programa, para qualquer propósito.
- Liberdade nº 1: Liberdade de estudar como o programa funciona, e adaptá-lo para as suas necessidades. Acesso ao código-fonte é um pré-requisito para esta liberdade.
- Liberdade nº 2: Liberdade de redistribuir cópias de modo que você possa ajudar ao seu próximo.
- Liberdade nº 3: Liberdade de aperfeiçoar o programa e liberar os seus aperfeiçoamentos, de modo que toda a comunidade se beneficie. Acesso ao código-fonte é um pré-requisito para esta liberdade.

1. Conceitos Básicos de SO

Tipos de sistemas operacionais

Software de código aberto (Open source)

O código-fonte é disponibilizado por meio de uma licença de código aberto para modificação ou melhoria por qualquer pessoa. O termo "código aberto" foi criado pela OSI (Open Source Initiative).

O software de código aberto difere-se de um software livre por não respeitar as 4 liberdades definidas pela Free Software Foundation (FSF), compartilhadas também pelo projeto Debian em "Debian Free Software Guidelines (DFSG)". Vale ressaltar que qualquer licença de software livre é também uma licença de código aberto (Open Source), mas o contrário nem sempre é verdade.

São exemplos de licenças de código aberto:

- Apache License.
 - MIT License.
 - Mozilla Public License.
 - Common Development and Distribution License.
 - Eclipse Public License.
-

1. Conceitos Básicos de SO

A estrutura do SO

Estrutura do SO: kernel, system calls, modos de acesso

O sistema operacional é formado por um conjunto de rotinas que oferecem serviços aos usuários, às suas aplicações e ao próprio sistema. Esse conjunto de rotinas é denominado **núcleo** do sistema ou **kernel**, cuja localização na máquina de níveis está ilustrada na imagem a seguir:



1. Conceitos Básicos de SO

A estrutura do SO

O sistema operacional funciona com algumas particularidades quanto à execução das rotinas:

- Tarefas não sequenciais, ou seja, as rotinas são executadas concorrentemente.
- Sem ordem predefinida.
- Eventos assíncronos.
- Eventos relacionados ao hardware e eventos relacionados às tarefas internas do próprio SO.

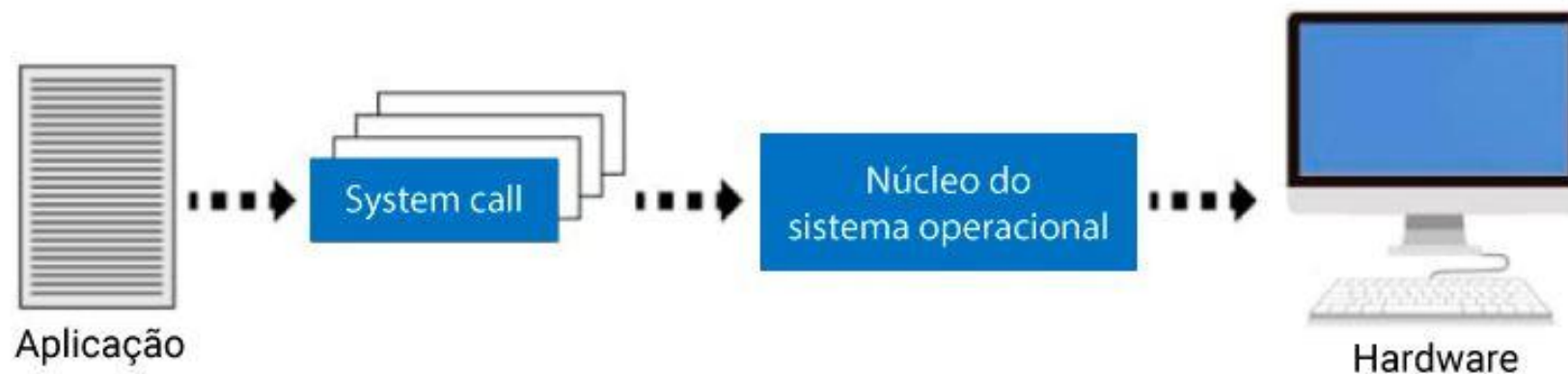
Já o núcleo do SO deve atuar considerando essas particularidades, e para isso implementa funções como:

- Tratamento de interrupções e exceções.
- Criação e eliminação de processos e threads.
- Sincronização e comunicação entre processos e threads.
- Escalonamento e controle de processos e threads.
- Gerência de memória.
- Gerência do sistema de arquivos.
- Gerência dos dispositivos de E/S.
- Suporte a redes locais e distribuídas.
- Contabilização do uso do sistema.
- Auditoria e segurança do sistema.

1. Conceitos Básicos de SO

A estrutura do SO

A execução das rotinas no sistema operacional é controlada pelo mecanismo denominado system call ou chamada ao sistema, que é um mecanismo de proteção por software que garante a execução somente de rotinas previamente autorizadas.



1. Conceitos Básicos de SO

A estrutura do SO

System calls

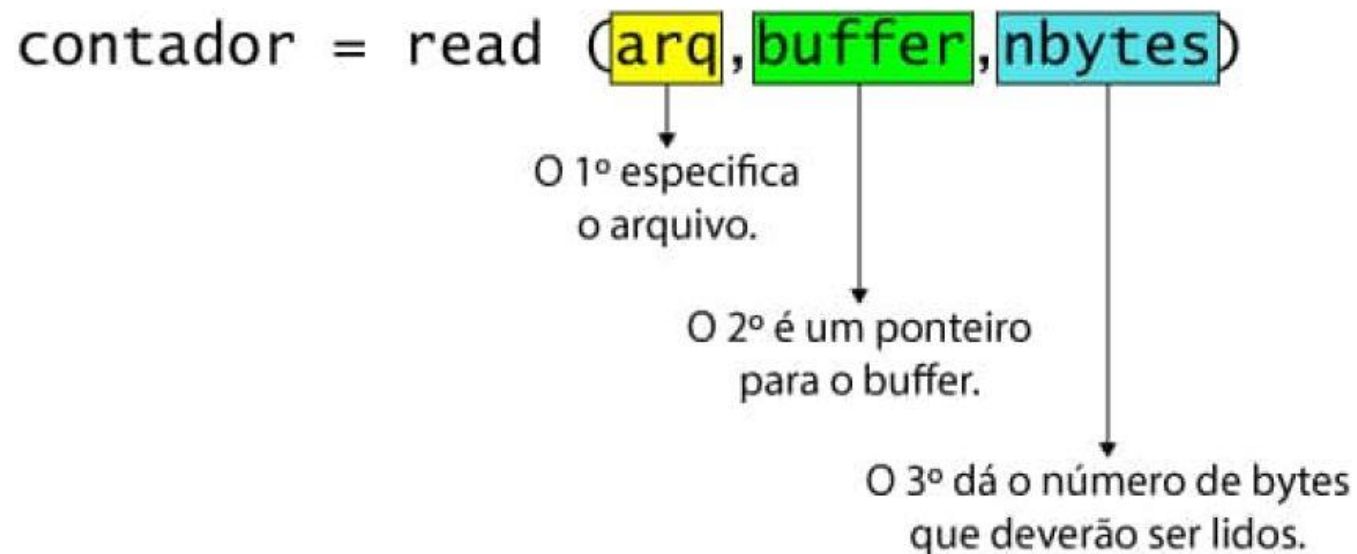
Quando qualquer aplicação deseja chamar uma rotina do sistema operacional, ela aciona a system call. O sistema operacional verifica se a aplicação possui os privilégios suficientes para a execução da rotina desejada. Se não tiver, o sistema operacional impede o desvio para a rotina do sistema e informa ao programa de origem que a operação não será executada.

A system call é uma “porta de entrada” para o acesso ao núcleo do SO e aos seus serviços. Cada serviço possui uma system call associada, e cada SO possui seu próprio conjunto de chamadas, com nomes, parâmetros e formas de ativação específicas.

1. Conceitos Básicos de SO

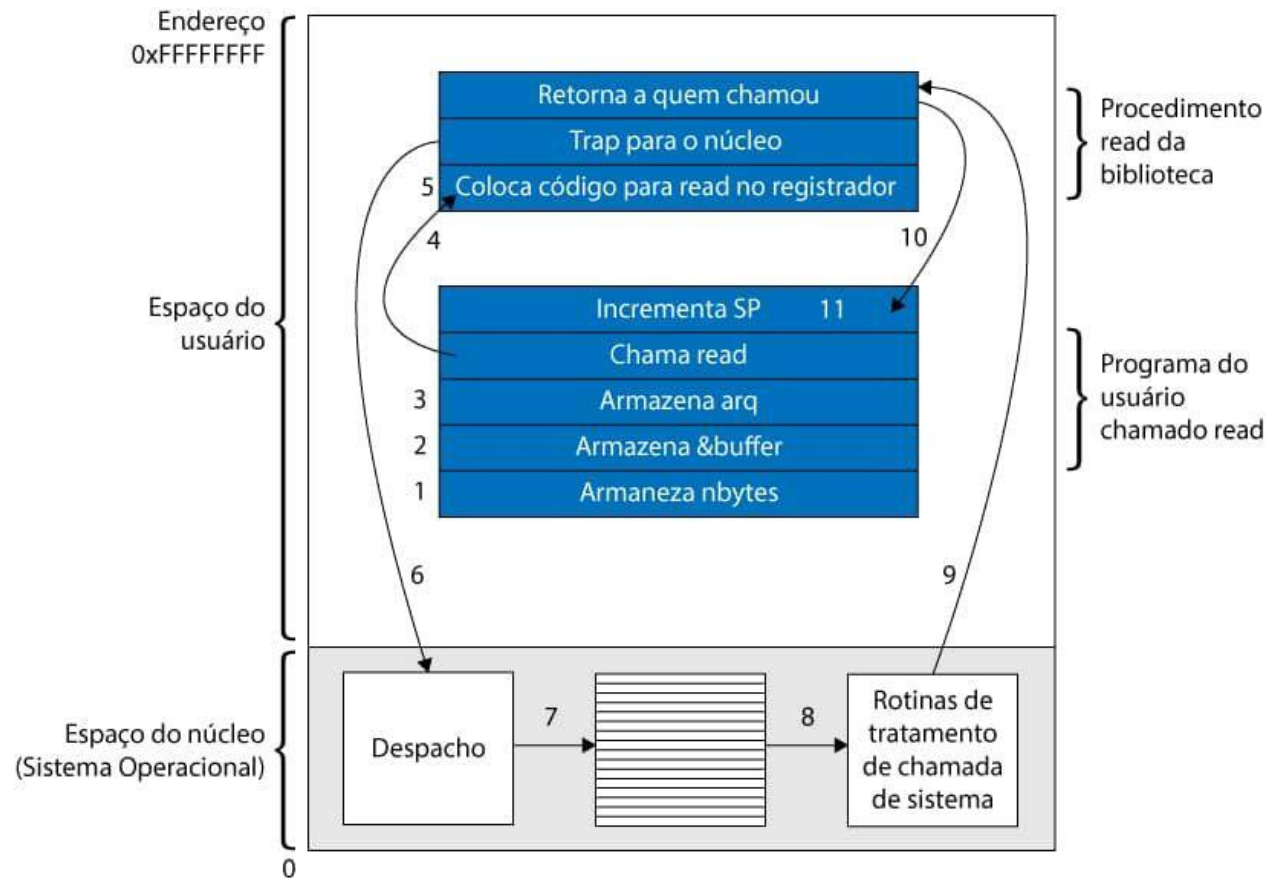
A estrutura do SO

A chamada ao sistema "read", apresentada na imagem a seguir, possui três parâmetros:



1. Conceitos Básicos de SO

A estrutura do SO



1. Conceitos Básicos de SO

A estrutura do SO

Passo 1 a 3

O programa que se chama read primeiramente armazena os parâmetros na pilha.

Passos 4 e 5

Segue-se para a chamada real à rotina de biblioteca (passo 4) que, em geral, coloca o número da chamada de sistema em um local esperado pelo SO, por exemplo, em um registrador (o passo 5).

Passo 6

É executada uma instrução TRAP para passar do modo usuário para o modo núcleo, iniciando a execução em um determinado endereço no núcleo.

Passo 7

O código do núcleo, iniciado após a instrução TRAP, verifica o número da chamada de sistema e despacha para a rotina correta de tratamento dessa chamada.

Passo 8

É a execução da rotina de tratamento da chamada de sistema.

Passos 9 e 10

Após o término dessa rotina, pode-se retornar o controle para a rotina de biblioteca no espaço do usuário, na instrução seguinte à instrução TRAP (passo 9), em geral do mesmo modo no qual são feitas as chamadas de rotina (passo 10).

Passos 11

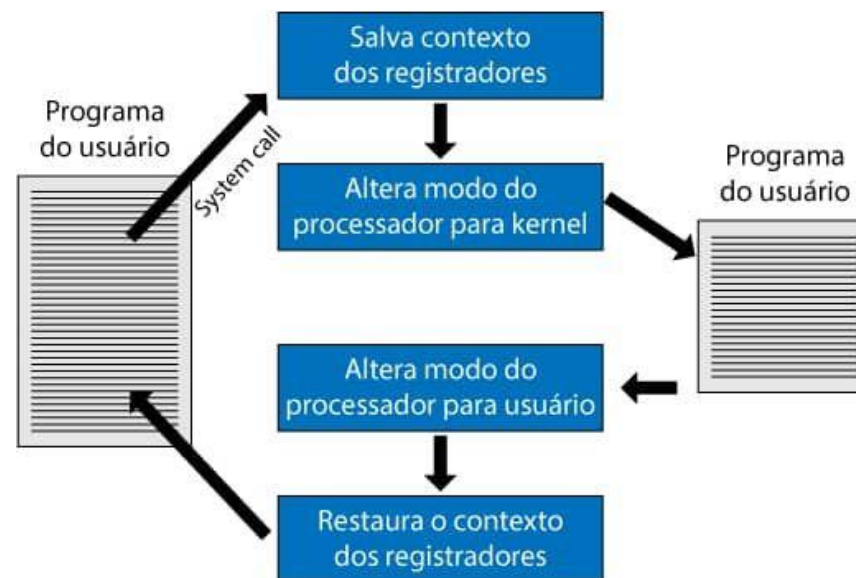
O programa do usuário deve limpar a pilha. O programa agora está liberado para fazer o que quiser.

1. Conceitos Básicos de SO

A estrutura do SO

Em geral, para qualquer situação, o funcionamento da system call consiste:

1. Na realização da chamada ao sistema pela aplicação que deseja executar rotina(s).
2. No processamento da solicitação pelo kernel.
3. No retorno, para a aplicação solicitante, de uma resposta, com um estado de conclusão, indicando se houve algum erro.



1. Conceitos Básicos de SO

A estrutura do SO

Um exemplo de conjunto de chamadas de sistema é o POSIX (Portable Operating System Interface for Unix), padrão internacional ISO/IEC/IEEE 9945 que foi uma tentativa de padronizar as system calls.

O objetivo inicial foi unificar as diversas versões existentes do sistema operacional UNIX, permitindo um aplicativo utilizando o padrão POSIX teoricamente poder ser executado em qualquer SO que possua suporte. É incorporado, atualmente, pela maioria dos sistemas operacionais modernos.

Já o conjunto de chamadas de sistemas Win32 é suportada desde o Windows 95. Cada nova versão do Windows traz novas chamadas que não existiam anteriormente, buscando não invalidar os programas existentes.

1. Conceitos Básicos de SO

A estrutura do SO

A API Win32 possui um grande número de chamadas para gerenciar aspectos da interface gráfica (como janelas, figuras geométricas, textos, fontes de caracteres, caixas de diálogos, menus etc.) e possui diferenças em relação ao POSIX. Veja algumas de suas características:

Desacoplamento de chamadas

Uma característica é o desacoplamento das chamadas de biblioteca e das chamadas reais ao sistema, o que pode dificultar a identificação do que é uma chamada ao sistema, e do que é uma chamada de biblioteca no espaço do usuário.

Acesso aos serviços do SO

A API Win32 também permite aos programadores acesso aos serviços do SO mesmo quando trabalhando em uma versão diferente com chamadas ao sistema modificadas. As chamadas Win32 chegam a milhares.