

SISTEMAS OPERACIONAIS

Andréa Martins

E-book 2

fam
ONLINE

Neste E-Book:

INTRODUÇÃO	3
HARDWARE E SOFTWARE.....	4
PROCESSADORES.....	6
ARQUITETURAS RISC E CISC.....	8
CISC.....	8
RISC.....	9
MEMÓRIA PRINCIPAL.....	11
Memória Cache	14
Memória Secundária.....	15
Dispositivos de Entrada e Saída.....	17
SOFTWARE.....	19
Princípios básicos de software de entrada e saída	21
PROCESSOS	23
Estrutura de um processo	25
Estados de um processo	27
Bloco de controle de processos.....	28
CONSIDERAÇÕES FINAIS	30
SÍNTESE	31

INTRODUÇÃO

Começamos a nossa jornada falando sobre o mundo dos sistemas operacionais. Aqui, estudante, você poderá aprender ainda mais! Estudaremos tópicos bastante importantes, como conceitos de hardware e software, processadores, rotinas de acesso e muito mais!

Vamos começar?

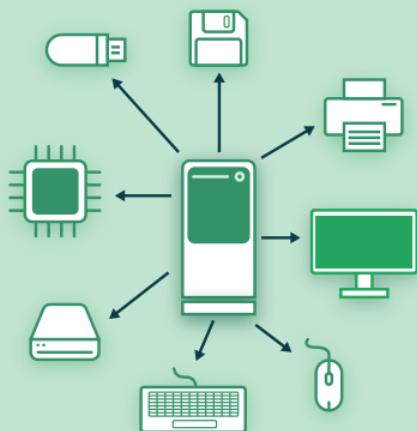
HARDWARE E SOFTWARE

Hardware e Software são considerados, respectivamente, o cérebro e o coração de um computador. São de fundamental importância para o funcionamento de um sistema computacional e são complementares, um necessita do outro, e os dois formam um elemento maior, único: os sistemas da informação que conhecemos. Todos os aparelhos que têm ligação com a tecnologia, como tablets, smartphones, computadores pessoais têm, em sua constituição básica, esses dois elementos.

Para que consigamos entender melhor, hardware é a parte física, o equipamento em si, e é composto, basicamente, por processadores, mouse, impressora, teclado, dispositivos de entrada e saída, entre outros componentes. São as ‘peças’ do computador.

E software é a parte lógica; são os programas, aplicativos, regras ou instruções que dão ‘vida’ e significado aos sistemas computacionais. É o software que faz nossos aparelhos funcionarem. Observe a representação abaixo:

Hardware



Software



Figura 1: Diferença entre hardware e software. **Fonte:** [GCFGlobal](#).

SAIBA MAIS

O vídeo a seguir nos dará detalhes importantes sobre hardware e software. Vamos assistir?

[Link do vídeo](#) .

[**Acesse o Podcast 1 em Módulos**](#)

PROCESSADORES

Você provavelmente já ouviu falar em CPU, certo? A Unidade Central de Processamento – ou *Central Processing Unit*. Pois é, a CPU é o processador da máquina. São os processadores (ou CPU) que interpretam as instruções de um programa. Lembra que falamos que o software e o hardware se complementam entre si? É exatamente isso! A CPU tem como tarefa fazer a correta execução da memória RAM, e essa execução é realizada por meio de operações matemáticas que conhecemos: como somar, subtrair, dividir e multiplicar (aritmética), e também utiliza operações lógicas, como movimentação de dados, comparações, entre outras tarefas.

Os processadores têm na sua composição elementos que compõe a UC, ou Unidade de Controle; a ULA, Unidade Lógica e Aritmética, e também por registradores.

A figura a seguir nos exemplifica como é um processador:

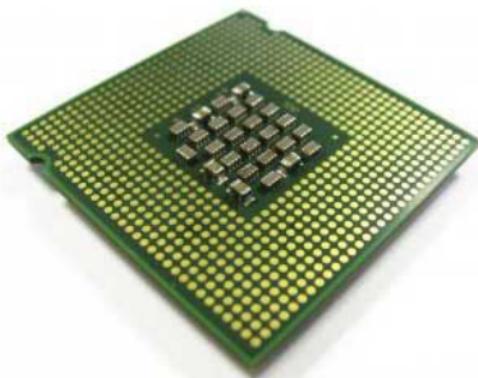


Figura 2: Processador. **Fonte:** [Zoom](#).

SAIBA MAIS

O assunto **processadores** é bastante interessante e extenso. Os processadores são considerados o “cérebro” da máquina. Para complementar seus estudos, recomendo a leitura do artigo abaixo:

[Link Unicamp](#) § .

ARQUITETURAS RISC E CISC

A arquitetura de um sistema computacional mostra qual o processador foi usado em nosso computador.

Para aumentar nosso conhecimento sobre processadores, falaremos um pouco sobre as arquiteturas Risc e Cisc. Essas arquiteturas são as instruções de processamento que a CPU é capaz de reconhecer.

CISC

A expressão CISC (*Complex Instruction Set Computing*) significa Computador com um Conjunto Complexo de Instruções.

Um processador que possui arquitetura do tipo CISC tem a capacidade de reconhecer centenas de instruções. Isso dá a essa arquitetura a vantagem de poder fazer a execução direta da grande maioria das operações programadas pelos softwares atuais.

Geralmente, a arquitetura CISC requer do processador instruções bastante complexas. Essas instruções têm sua interpretação executada por microprogramas. Outra característica é que possui um núcleo de registrador pequeno, e isso faz com que qualquer instrução referencie, diretamente, à memória principal. Esse processador é usado pela AMD e Intel. Como interpretam grande quantidade de informações, podem ter sua performance reduzida, o que os torna mais lentos.

RISC

A expressão RISC significa *Reduced Instruction Set Computing*, ou Computador com um Conjunto Reduzido de Instruções.

A principal característica desse tipo de processador é que tem a capacidade de interpretar instruções limitadas, se compararmos com os processadores CISC. Mas, a grande vantagem dele é que tem instruções otimizadas, o que significa ganho de performance e, consequentemente, mais rapidez na execução dos processos. Outra característica do Risc é que tem poucas instruções para fornecer ao sistema computacional, e essas instruções são executadas pelo próprio hardware. A maioria das informações dessa arquitetura é que as instruções não precisam acessar a memória principal, fazendo com que trabalhem como registradores – que, nesse tipo de processador, estão presentes em números elevados. É usada em processadores Power PC (Apple, Motorola, IBM, Sun).

CARACTERÍSTICA	CISC	RISC
Instruções por Ciclos	Complexas, executadas em muitos ciclos	Simples, executadas em um único ciclo
Acesso à Memória	Qualquer instrução tem a capacidade de referenciar à memória	Somente operações dos tipos LOAD/STORE referenciam à memória
Pipeline	Pouco ou mesmo nenhum pipeline	Usa a pipeline de forma intensa
Execução das Instruções	Executadas pelo próprio programa	Executadas pelo hardware

CARACTERÍSTICA	CISC	RISC
Formato das Instruções	Possui instruções de formato variável	As instruções possuem formato fixo
Quantidade de Instruções e Modos de Endereçamento	Múltiplos modos de endereçamento	Modos de endereçamento minimizados
Complexidade do sistema	A complexidade está contida no microprograma	A complexidade reside no compilador
Registradores	Apenas um conjunto de registradores	Múltiplos conjuntos de registradores

Tabela 1: A tabela acima nos mostra as principais diferenças entre CISC e RISC.

MEMÓRIA PRINCIPAL

Um computador tem muitos tipos de memórias diferentes em seu interior. Chamamos de memória RAM (*Random Access Memory*) a memória principal, ela é chamada também de memória de acesso aleatório. O nome da memória principal advém do fato de que é utilizada para manter programas em execução de ambos os lados: do usuário e do sistema operacional. Observe como é a memória RAM.



Figura 3: Memória RAM visualização. **Fonte:** [TechTudo](#).

FIQUE ATENTO

A memória RAM é o que nos faz ter acesso aos dados do computador. Ela não armazena nada de forma definitiva. Essa é a função de um HD. A RAM apenas lê os arquivos.

A base dessa memória são as células, que nada mais são do que as unidades de acesso. Essas células possuem um determinado tamanho, que é denominado **bit**. Mas, você deve estar se perguntando: como conseguimos acessar os conteúdos de uma célula específica? Muito bem, isso é feito por meio da designação de um número, que chamamos popularmente, de endereço. O endereço é, na verdade, um norte para a célula de memória, uma referência. Quando abrimos um programa através de um acesso e esse programa começa a executar operações em uma célula (seja uma operação de leitura ou de gravação), o primeiro passo é o acesso ao endereço de memória, ou seja, o computador localiza esse programa para depois realizar a operação. Por isso, o endereçamento é tão importante.

SAIBA MAIS

Vamos aprofundar nossos conhecimentos acerca de endereçamento e células da memória? Acesse os vídeos abaixo!

[Link da TechTudo](#) . [Link do Youtube](#) .

Quanto maior for a memória RAM de uma máquina, melhor será seu desempenho. Se a memória RAM da máquina não for grande o suficiente, o sistema terá de usar a memória virtual, que é bastante lenta, se comparada à RAM. Na verdade, podemos dizer que a memória virtual é uma extensão da memória RAM. As figuras abaixo demonstram isso de forma bastante simples; conseguimos observar os processos utilizando a memória virtual e o fluxo gerado por esse fluxo.

A primeira figura exemplifica o caminho que a informação percorre até chegar à memória virtual, e a segunda figura mostra como os processos se agrupam quando a memória física está comprometida:

Velocidade de acesso a memória virtual

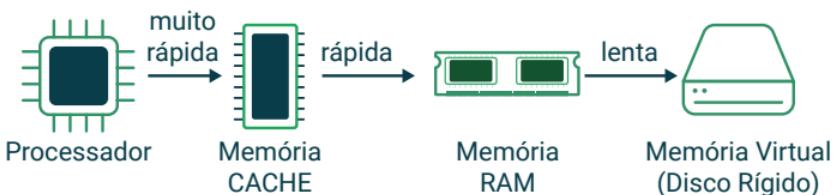


Figura 4: Como o computador acessa a memória virtual.
Fonte: gustavoschroeder.wordpress.com.

Memória Virtual

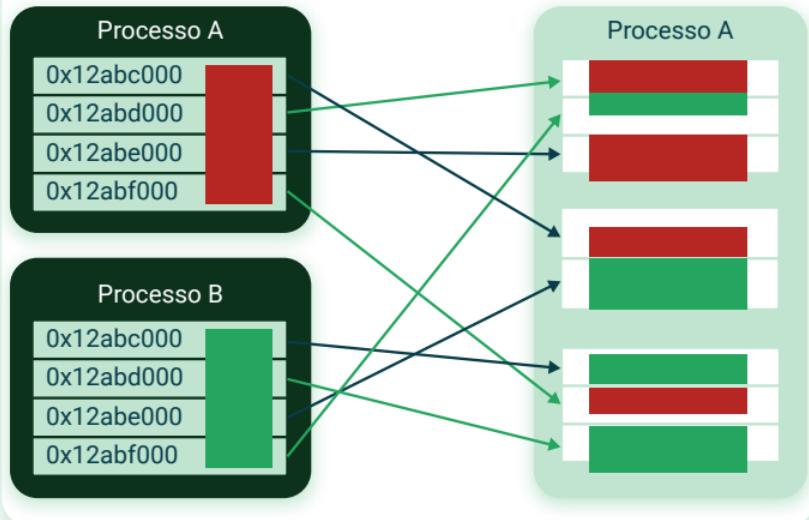


Figura 5: Acesso de memória virtual e física, aspectos da memória RAM.

Acesse o Podcast 2 em Módulos

Memória Cache

Podemos definir esta memória como uma memória de altíssima velocidade, bastante volátil e, via de regra, com baixo poder de armazenamento. Ela tem como meta mitigar a diferença que existe entre a velocidade com que são executadas instruções dentro de uma máquina e a velocidade que os dados conseguem ser armazenados ou gravados.

Essa memória possui a capacidade de armazenar os dados que são frequentemente acessados. Essa característica facilita, e muito, a vida de um processador: quando precisa acessar alguma informação, o processador rapidamente verifica se esse dado ou informação está presente na memória cache. É uma forma de otimizar seu desempenho. Caso essa informação não esteja armazenada na cache, o processador passa a fazer a busca da memória principal, o que, consequentemente, aumenta o tempo de resposta e diminui a performance.

Nas máquinas mais modernas, a memória cache possui em sua composição uma forma de vários níveis de memória, e é bastante simples entender o motivo desse particionamento: quanto menor for a capacidade que essa memória tiver de guardar uma informação, mais rápido será o acesso!

Para melhorar ainda mais o desempenho da memória cache, criou-se a hierarquização em vários níveis, assim, o nível mais alto dessa memória é denominado L1 e se caracteriza por ter baixa capacidade de

armazenamento e alta velocidade de acesso. A seguir temos o L2, com mais capacidade de armazenar, porém, com velocidade menor que a L1.

Memória Secundária

Agora que já sabemos o que é memória RAM, memória Virtual, e memória Cache, vamos conhecer mais um componente importante dentro dos sistemas operacionais: a memória secundária.

A memória secundária é usada para guardar, permanentemente, os dados no computador. Sim, a memória secundária guarda as informações de forma definitiva. Ela possui um armazenamento que não precisa de alimentação, totalmente diferente da memória principal, que precisa ter energia para permanecer com as informações.

Em contrapartida, quando o computador realiza um acesso a essa memória secundária para executar a leitura ou gravação de algum dado, o processo é bem mais lento, se compararmos o desempenho com a memória principal; mas ela possui a vantagem de ter um custo bem mais baixo e ter capacidade de armazenamento bastante superior. Outra informação que devemos saber é que a memória secundária não consegue ser processada diretamente pela CPU: num primeiro momento, obrigatoriamente, a CPU acessará a memória RAM.

Para ter uma ideia da diferença de tempos entre as duas memórias, atente-se aos dados a seguir:

- Memória secundária: funciona em milissegundos
- Memória principal: funciona em nanossegundos

São exemplos de memória secundária:

- Discos magnéticos (discos rígidos – HD)
- Discos ópticos (CD, DVD, Blu-Ray)
- Memória Flash (Pendrive)

Observe a figura abaixo, que compara a velocidade e o desempenho das memórias que estudamos até agora:

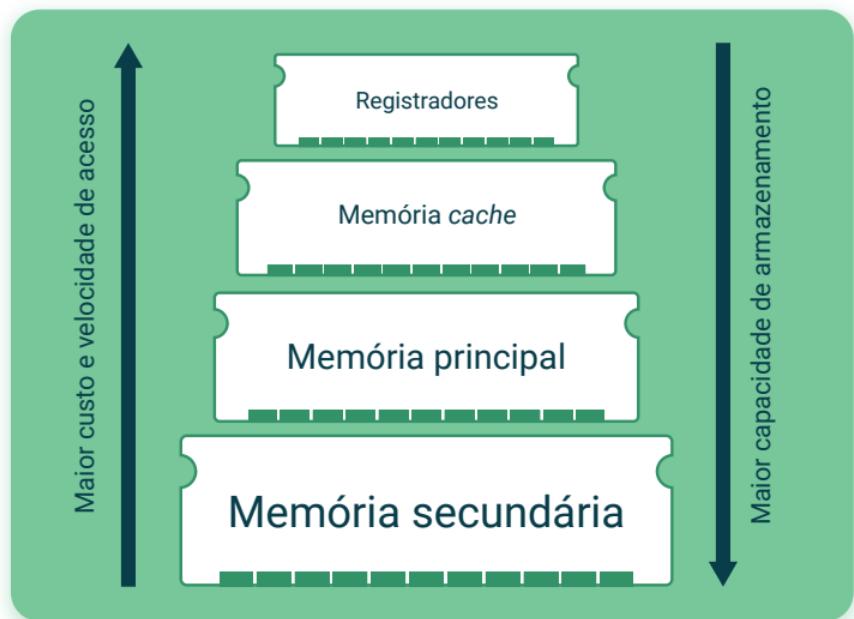


Figura 6: Relação entre dispositivos de armazenamento.
Fonte: Adaptado de Maia, 2007

Dispositivos de Entrada e Saída

Dispositivos de entrada e saída também são conhecidos como periféricos. São partes do computador que fazem com que ele tenha a capacidade de expandir as suas funcionalidades. Os periféricos são divididos em duas grandes categorias: os periféricos que fazem parte das memórias secundárias e os que se enquadram na categoria de interface usuário-máquina. Vamos aprofundar nossos conhecimentos nesse assunto?

- Dispositivos de Entrada: têm como função fazer a decodificação dos dados que entram no computador. Logo após a conexão desse dispositivo, os dados são decodificados e processados pela máquina. Podemos citar como exemplos desse tipo de periférico:
 - Câmeras digitais
 - Canetas óticas
 - Blu-ray
 - CD ou DVD
 - Joystick
 - Leitora de código de barras
 - Microfone
 - Mouse
 - Pen drive
 - Scanner
 - Teclado
 - Tela touch screen

- Dispositivos de Saída: esses dispositivos fazem o contrário dos periféricos de entrada: eles fazem a decodificação dos dados que são emitidos pelos dispositivos de entrada para que possam ser compreendidos pelo usuário. Em outras palavras, os periféricos de saída permitem a extração e visualização dos dados que estão na máquina. Alguns exemplos desse tipo de dispositivo são:
 - Caixas de som
 - Gravador de DVD
 - Gravador de CD
 - Impressora
 - Monitor de vídeo
 - Pendrive
 - Plotter
 - Projetores digitais

SOFTWARE

O software é o coração do Sistema Operacional, enquanto o hardware é o cérebro. Quando ligamos um computador, temos contato com a interface, que é a ‘cara’ do software, ele atende as necessidades dos usuários e também do hardware. Vamos conhecer alguns utilitários que o sistema operacional pode possuir?

- Tradutor: Essa funcionalidade exigia extrema perícia por parte dos programadores, quando teve sua primeira versão lançada. Após o surgimento da linguagem de programação chamada **assembly**, essa situação melhorou um pouco; mas, apesar do avanço proporcionado por essa linguagem, o código fonte, que é gerado pelos usuários, ainda não consegue ser executado diretamente pelo processador. Para essa conversão é necessária uma criação simbólica para a linguagem da máquina, e a essa conversão damos o nome de **tradutor**. O resultado desse processo é um módulo que chamamos de “módulo-objeto”. Dependendo do código fonte que usarmos para criar os módulos tradutores, podemos gerar dois tipos de módulos-objeto: o montador e o compilador.
- Interpretador: O interpretador de códigos é uma espécie de tradutor, só que sem a geração de módulo-objeto. Quando executamos um programa classificado como de alto nível (PHP, Basic, Perl, etc.), o interpretador faz a tradução de todas as instruções e, logo após, as executa.

- Linker: O linker nada mais é do que um editor de ligação, que tem como meta realizar a geração de um ou mais módulos-objeto, transformando-os num programa que seja executável.
- Loader: O loader atua junto ao sistema operacional e tem como principal função carregar na memória principal um programa para que esse programa consiga ser executado; podem ser absolutos ou realocáveis.
- Depurador: Sabemos que, para desenvolver um programa, o programador precisa de técnicas e bastante atenção, mas mesmo o melhor programador pode cometer erros, e é aí que entra o depurador. A função do depurador é ajudar o usuário na correta identificação e correção de erros (sintaxe, lógica e escrita do código), corrigindo esses erros. Com o depurador, o programador consegue verificar e acompanhar a execução do programa do início até o final. Alguns dos recursos que o depurador oferece são:
 - Acompanhamento da execução do programa, por todo o código
 - Visualização dos conteúdos variáveis
 - Marcação de pontos de parada dentro do programa (break points)

Princípios básicos de software de entrada e saída

Quando falamos de softwares de sistema operacional, precisamos falar dos princípios básicos de software de entrada e saída. Trata-se de um software de entrada e saída com complexidade bastante grande. Esses softwares podem pertencerem a classes diferentes, como:

- Placas de rede com protocolos
- Discos magnéticos do tipo SCSI
- IDE
- EIDE
- ZIP

Esse subsistema de software tem como objetivo principal conseguir o máximo de padronização possível na forma de acesso dos dispositivos de entrada e saída. Para conseguir essa organização, esse software é composto, normalmente, por quatro camadas e cada uma dessas camadas possui comandos que fornecem tarefas à camada superior. A figura a seguir mostra como funciona essa função:

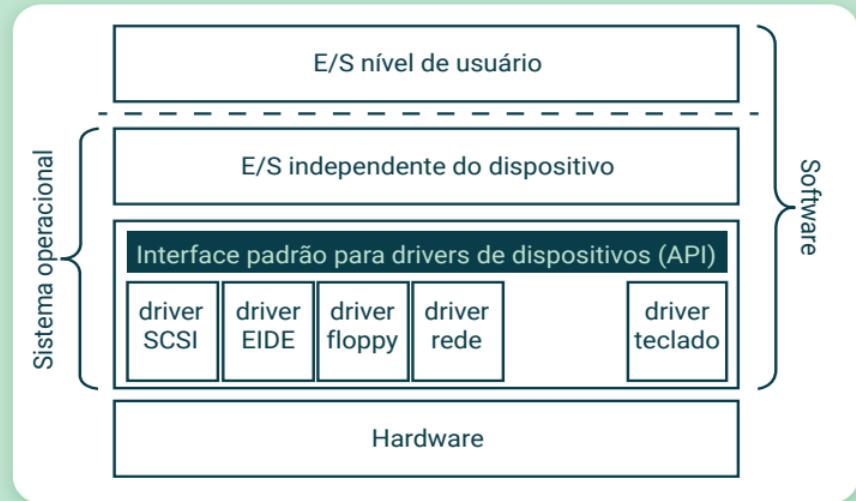


Figura 7: Estrutura de camadas em um subsistema de entrada e saída de software.

PROCESSOS

Quando falamos de processos, dentro do sistema operacional, entendemos que um programa de computador está sendo executado. Trata-se de um módulo que é executável e único, que funciona com outros módulos executáveis concorrentemente. Os sistemas operacionais não executam os programas que nós vemos quando acessamos algum aplicativo, por exemplo. Os programas e os aplicativos que executamos possuem em seu interior um emaranhado de instruções e comandos, mas não nos damos conta desses detalhes porque são os processos que realmente executam essas tarefas. Com isso, podemos concluir que um único programa pode ter com ele vários processos que têm relação direta com ele.

O Google Chrome, por exemplo, um browser de internet que usamos frequentemente, toda vez que é acionado, executa uma série de tarefas, e isso acontece novamente cada vez que abrimos uma aba nova. Isso otimiza o gerenciamento das abas, pois faz com que cada uma das abas seja gerenciada de forma independente e, portanto, se uma das abas travar, as outras podem continuar funcionando perfeitamente.

Em outras palavras, os processos dentro dos sistemas operacionais nada mais são do que tarefas que estão sendo executadas; mas nem todos os processos (ou tarefas) estão diretamente ligadas a algum aplicativo, pois vários processos representam as tarefas que mantêm o sistema funcionando e trabalhando corretamente. Alguns desses processos são:

- Gerenciamento de redes
- Memória
- Disco
- Execução do antivírus

Dessa forma, podemos concluir que os processos nada mais são que softwares em execução; e que o usuário consegue ter controle desses programas que estão sendo executados ou, pelo, menos, de grande parte desses programas. A figura abaixo exemplifica o fluxo de processos dentro de um sistema operacional; conseguimos observar a ordem de cada tarefa. Todas as tarefas demonstradas na figura abaixo podem ser consideradas processos:

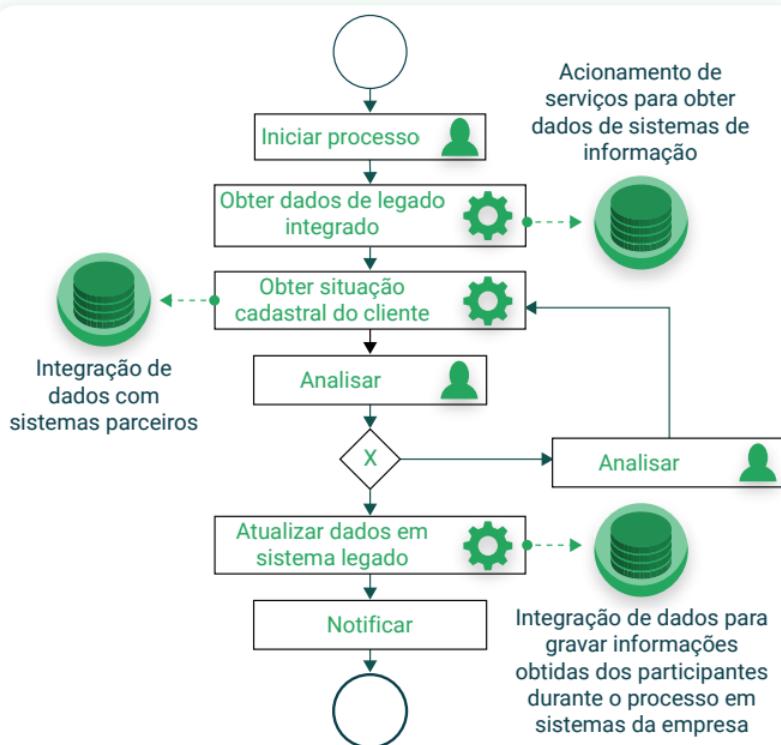


Figura 8: Fluxo de processos dentro de um sistema operacional. **Fonte:** Adaptado de [Blog Iprocess](#).

Estrutura de um processo

Se tomarmos como base um sistema multiusuário que explicamos anteriormente, cada um dos usuários que fazem parte do sistema terá um processo que estará diretamente associado a um programa de computador. Funciona da seguinte forma: quando o usuário executa algum programa, ele tem a impressão de que todos os recursos do computador estão disponíveis somente para a execução do programa (que ele está executando), e isso não corresponde à realidade, pois em um sistema multiusuário, todos os recursos disponíveis são compartilhados entre todos os usuários, inclusive o processador.

Para conseguir fazer o gerenciamento de todas as tarefas que são executadas por todos os usuários de uma corporação, o processador da máquina faz uso do programa do usuário durante um tempo determinado, e muda de processo para atender aos demais. Ou seja, esse tipo de sistema consegue gerenciar todos os processos de maneira ordenada.

O processo de um sistema operacional é composto basicamente por três partes:

- Contexto de hardware: faz o armazenamento de informações sobre registradores gerais do processo, registradores de uso específico e registrador de status.
- Contexto de software: tem em sua base de formação três grupos de informação: identificação (número da identificação do processo ou PID: **Process**

Identifier), cotas (trata-se do limite dos recursos disponíveis que o processo tem a capacidade de fazer a alocação), e privilégios (define o que um processo faz dentro do sistema operacional, diante dos demais processos ou mesmo diante dele mesmo).

- Espaço de endereçamento: são as posições de memória a que um processo pode fazer referência.

São essas três partes integradas que mantêm todas as informações importantes para a correta execução do processo. A figura abaixo demonstra isso:



Figura 9: Estrutura de um processo. **Fonte:** Adaptado de Maia, 2007.

Estados de um processo

Os processos dentro no sistema operacional não são fixos, eles mudam de status durante o período de tempo em que estão sendo executados. Dessa forma, uma das maneiras que temos para definir seu estado no momento é verificando a atividade em curso. Esses são os estados para um processo:

- Novo: o processo acaba de ser criado
- Execução: o processo está em andamento
- Espera: o processo aguarda a ocorrência de um determinado evento
- Pronto: o processo está apto para ser acionado por um processador
- Terminado: a execução do processo foi finalizada

Observe essa sequência na figura abaixo:

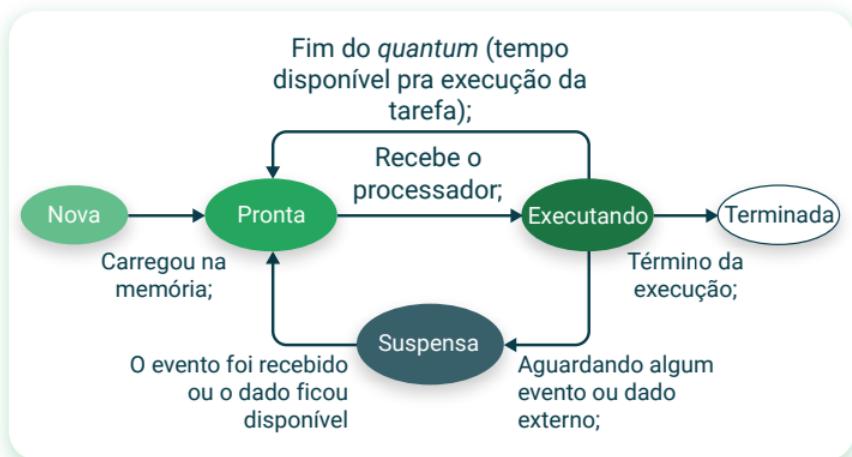


Figura 10: Fluxo de processos. **Fonte:** Adaptado de [TreinaWeb](#).

Bloco de controle de processos

Os processos que estão presentes dentro do sistema operacional são representados por blocos. Nesse bloco encontramos informações bastante importantes acerca de um determinado processo que pode estar em execução. Algumas dessas informações são sobre:

- Em qual status está o processo;
- Qual é o nome do processo;
- Se o processo tem ou não prioridade;
- O contador do programa
- Os registradores que a CPU contém;
- Como está sendo executado o gerenciamento da memória;
- Informações sobre contabilização.

Outro ponto importante quando falamos de bloco de processos é o PCB – bloco de controle de processo –, que tem a função de armazenar todas as informações que mudem de um processo para o outro.

Todos os registros e atividades dos PCB que estão em um sistema operacional ficam dentro da memória principal do sistema, em uma área projetada para eles. Essa área tem tamanho limitado; esse limite é que será capaz de dizer qual será a quantidade máxima de processos executados de maneira simultânea, que podem ser suportados pela máquina.

O gerenciamento de todo esse processo é feito por meio das chamadas rotinas do sistema, que executam operações como: visualização de processos, eliminação, alteração, entre outras.

CONSIDERAÇÕES FINAIS

Neste capítulo, começamos a estudar as funcionalidades dos sistemas operacionais. Pudemos observar aspectos importantes sobre o funcionamento do hardware e software, dispositivos de entrada e saída, como é o funcionamento da memória principal, secundária, memória cache, e finalizamos falando sobre algo extremamente importante dentro dos sistemas operacionais: os processos.

Recomendo que você assista aos vídeos, especialmente os que foram indicados neste e-book, pois isso ajudará você no processo de compreensão dos sistemas operacionais.

Até a próxima!



SISTEMAS OPERACIONAIS

Nesse segundo e-book ampliamos um pouco mais nossos conhecimentos sobre sistemas operacionais. Entramos no universo dos processadores e suas múltiplas funções. Os temas abordados foram:

- A diferença entre hardware e software, entendendo as peculiaridades de cada uma das partes que compõe um computador
- Depois demos mais um passo entrando no assunto dos processadores, entendemos como eles funcionam, o que é uma CPU, o que são UCs
- Logo a seguir entendemos o que são e como funcionam as arquiteturas RISC e CISC
- Depois estudamos a memória principal, as funções de um HD, entendemos as diferenças entre memória física e memória virtual
- Depois abordamos o tema de memória secundária e o papel dela dentro de um sistema operacional
- Estudamos os dispositivos de entrada e saída, sua importância e suas funções
- Entendemos a função do software, os princípios básicos dos softwares de entrada e saída
- Por fim, entendemos como funcionam os processos dentro de um sistema operacional. Sabemos o que são estruturas de processos, os estados de um processo e os blocos de controle de processos.

Referências Bibliográficas & Consultadas

BITTENCOURT, P. H. M. **Ambientes operacionais.** São Paulo: Pearson Education do Brasil, 2014. [Minha Biblioteca]

CÓRDOVA JUNIOR, R. S.; LEDUR, C. L.; MORAIS, I. S.; **Sistemas operacionais.** Porto Alegre: SAGAH, 2018. [Minha Biblioteca]

COULOURIS, G.; DOLLIMORE, J.; KINDBERG, T.; BLAIR, G.; **Sistemas distribuídos conceitos e projetos.** 5. ed. Porto Alegre: Bookman, 2013. [Minha Biblioteca]

MACHADO, F. B.; MAIA, L. P. **Arquitetura de sistemas operacionais.** 5. ed. Rio de Janeiro: LTC, 2017. [Minha Biblioteca]

MACHADO, F. B.; MAIA, L. P. **Fundamentos de sistemas operacionais.** Rio de Janeiro: LTC, 2011. [Minha Biblioteca]

MACHADO, F. B.; MAIA, L. P. **Introdução à arquitetura de sistemas operacionais.** São Paulo: LTC, 1992.

MAIA, L. P. **Introdução à arquitetura de sistemas operacionais.** São Paulo: LTC, 2007.

OLIVEIRA, R. S.; CARISSIMI, A. da S.; TOSCANI, S. S. **Sistemas operacionais**. 4. ed. Porto Alegre: Bookman: Instituto de Informática da UFRGS, 2010. [Minha Biblioteca]

SILBERSCHATZ, A.; GALVIN, P. B.; GAGNE, G. **Fundamentos de sistemas operacionais**: princípios básicos. Rio de Janeiro: LTC, 2013. [Minha Biblioteca]

TANEMBAUM, A. S.; BOS, H. **Sistemas Operacionais Modernos**. 4. ed. São Paulo: Pearson Education do Brasil, 2016. [Biblioteca Virtual]

Fam
ONLINE