



Sistemas Operacionais

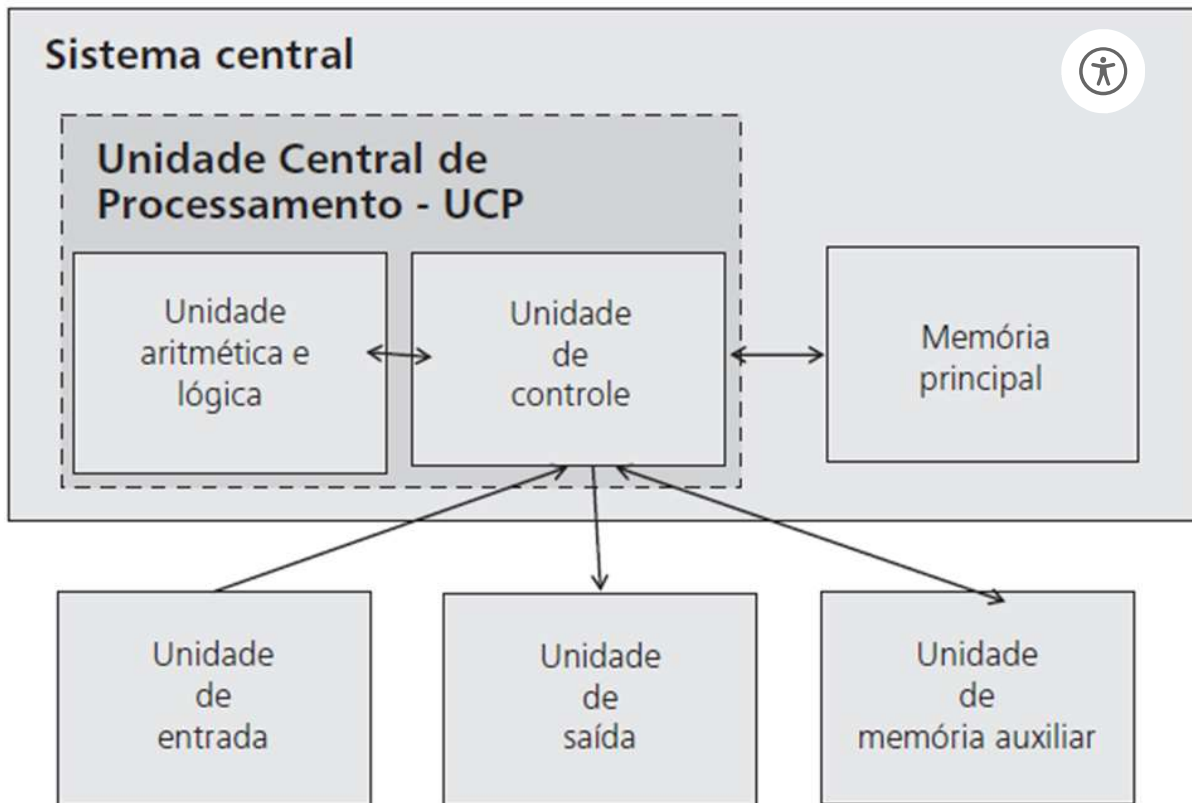


Modelo de von Neumann

Os componentes do modelo de von Neumann são:

- **Unidade de controle** (que busca cada instrução dos programas na memória e executa sobre os dados de entrada),
- **Unidade lógica e aritmética ULA** (parte da unidade central de processamento) e
- **Memória** (para armazenar dados e programas).

Os **dispositivos de entrada e saída** interagem com o modelo, mas não fazem parte dos componentes principais.



CPU – Unidade Central de Processamento

“A CPU é um componente vital para o computador. Muitos confundem a CPU com o gabinete, mas vale lembrar que o gabinete é a “carcaça” do computador, ou seja, uma estrutura para suporte e proteção. Já a CPU — ou o processador, como um termo mais conhecido — é considerada o cérebro do computador.”

Fonte: CÓRDOVA JUNIOR, R. S. et al. **Fundamentos computacionais**. Porto Alegre: SAGAH, 2018, p. 56

Unidade Lógica e Aritmética (ULA)

A unidade lógica e aritmética realiza ações indicadas nas instruções, executando operações numéricas (aritméticas) e não numéricas (lógicas),

além da preparação de informações para desvios do programa.



A Unidade de Controle (UC) e a unidade lógica e aritmética (ULA) formam a unidade central de processamento (UCP), ou simplesmente processador.

A ULA realiza operações aritméticas e operações lógicas sobre um ou mais operandos. Exemplos de operações realizadas pela ULA: soma de dois operandos; negação de um operando;

inversão de um operando; AND (“E” lógico) de dois operandos; OR (“OU” lógico) de dois operandos; deslocamento de um operando para a esquerda ou para a direita; rotação de um operando para a esquerda ou para a direita.

As operações da ULA são, geralmente, muito simples. Funções mais complexas, exigidas pelas instruções da máquina, são realizadas pela ativação sequencial das várias operações básicas disponíveis.

Memória

A memória de um sistema de computador tem a função de armazenar dados e instruções; é organizada em posições, que podem ser visualizadas como elementos em uma matriz.

Cada elemento da memória tem um endereço. Assim, pode-se falar de uma memória que tenha x posições: cada posição pode ser referenciada diretamente de acordo com a sua colocação na sequência.

Por exemplo, se uma memória tem 4096 posições, existem posições de memória 0, 1, 2, 3,..., 4094 e 4095.

Um registrador é como uma pequena unidade de memória.

Quando um endereço de memória aparece nos circuitos de controle conectados à memória, o conteúdo (o valor que está na posição) será lido da memória para os circuitos da unidade de processamento ou a informação na unidade de processamento será armazenada na memória, dependendo do trabalho associado com aquele endereço.

Instruções

Instruções em um computador são executadas em uma sequência determinada por suas posições de memória.

Na maioria dos computadores, instruções e dados são distribuídos em posições de memória.

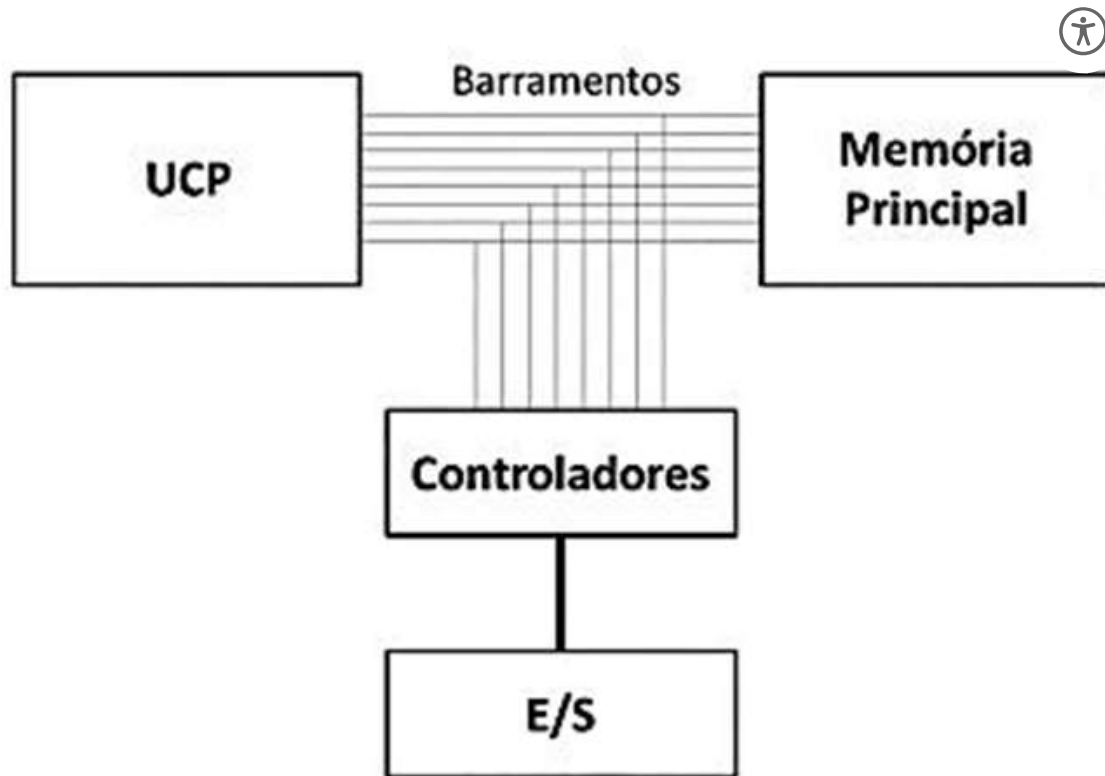
Drivers (Controladores)

São chamados de drivers os softwares que permitem a combinação do hardware, que é a parte física do computador, placa mãe e com outros dispositivos com o sistema operacional.

É o driver que converte as informações do hardware para o sistema operacional do computador.

É ele que cuida quando ao se abrir um requerimento, o processo seja executado, sendo permitida a interação do software com o dispositivo

Juntando as partes

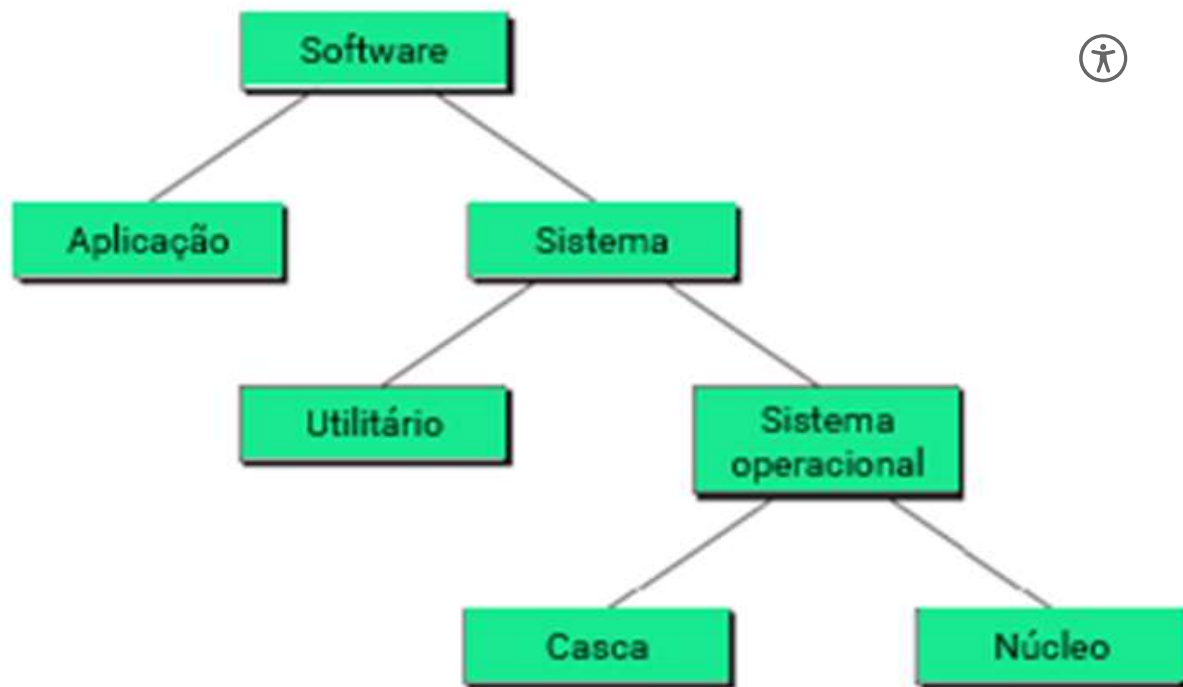


Programa

Um programa é constituído de uma sequência predeterminada de instruções, que deve ser seguida para que seja alcançado o objetivo computacional.

O programa e os dados correspondentes estão armazenados na memória da máquina; o conjunto de instruções (ou programa) deve ser interpretado para a realização do processamento, isto é, a informação codificada correspondente às ações e aos operandos precisa ser entendida e então processada.

Uma hierarquia do software

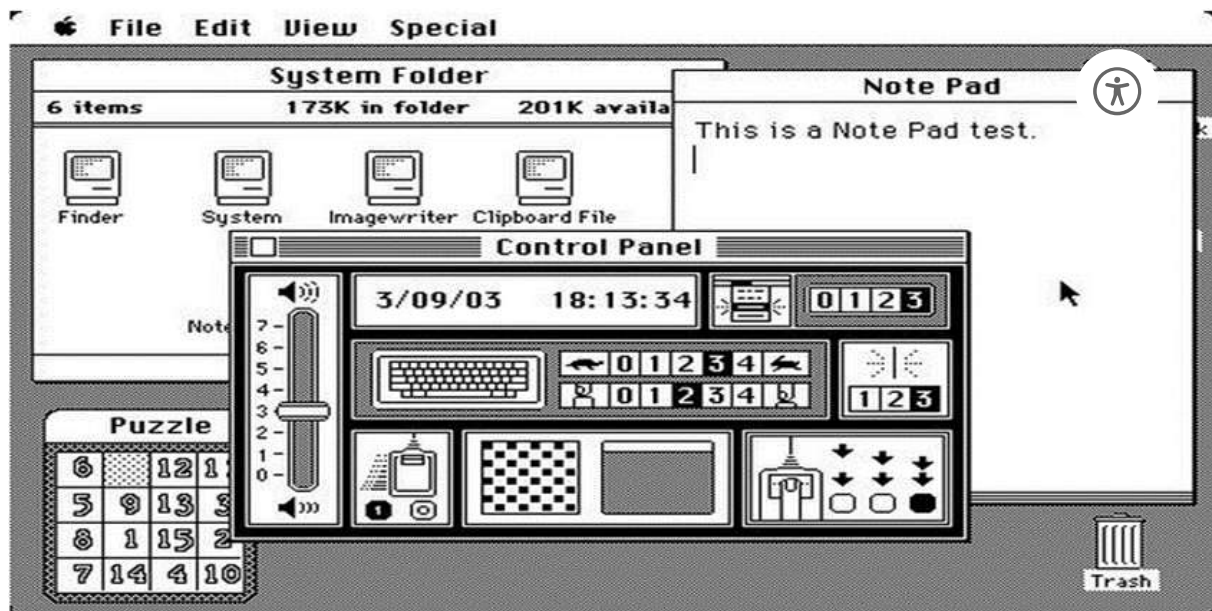


Sistema operacionais

Um sistema operacional visa abstrair o acesso e gerenciar os recursos de hardware, provendo aos aplicativos um ambiente de execução no qual o acesso aos recursos se faz através de interfaces simples, independentes das características e detalhes de baixo nível, e no qual os conflitos no uso do hardware são minimizados.

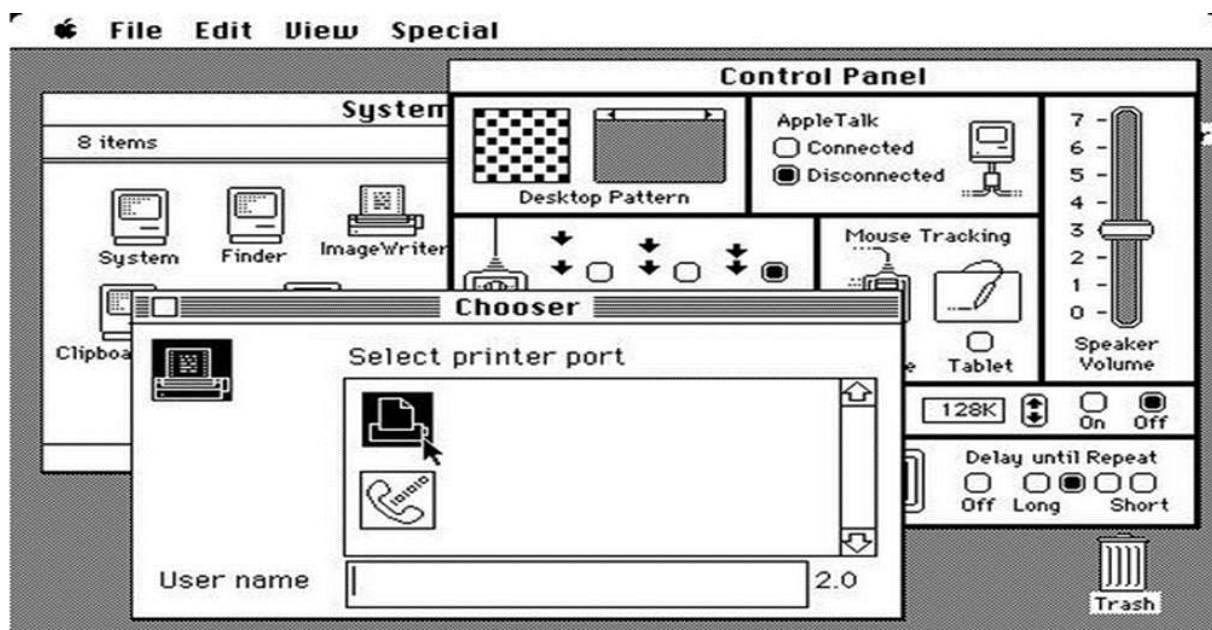
Os sistemas operacionais podem ser classificados segundo diversos parâmetros e perspectivas, como tamanho, velocidade, suporte a recursos específicos, acesso à rede, etc.

1984: Apple introduz o GUI no Mac (*Graphical User Interface*)

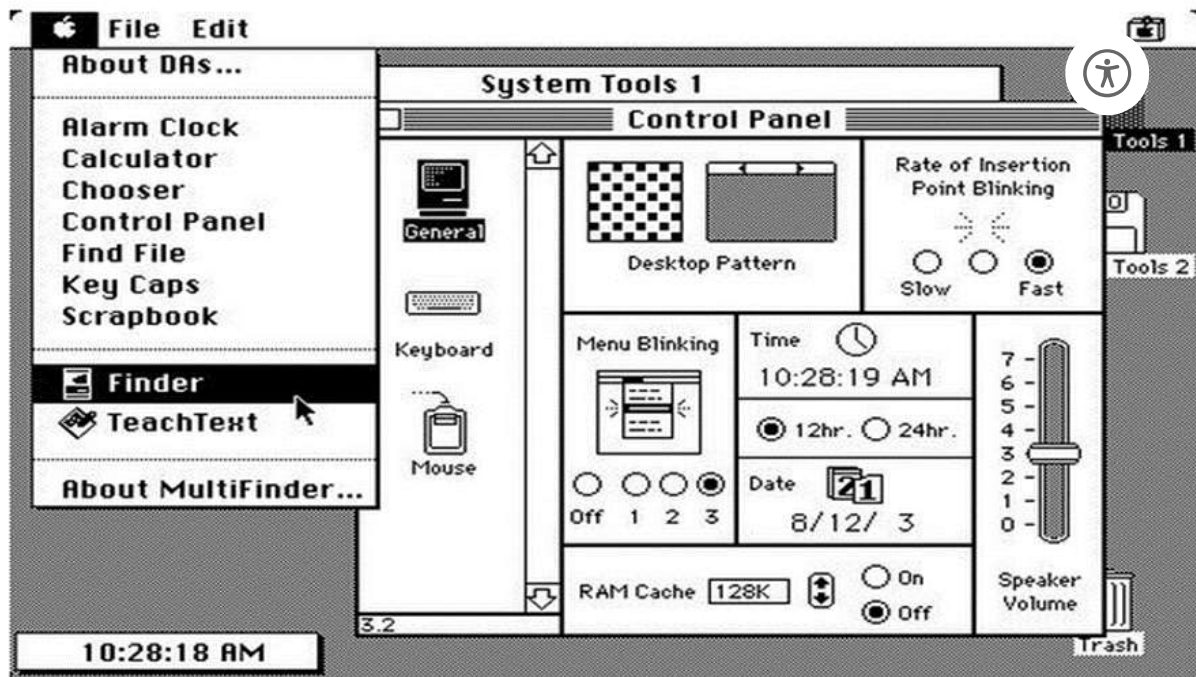


Abordagem de utilizar metáforas e desenhos para tornar mais fácil e intuitivo usar o computador em comparação a ter que lembrar comandos e digitar linhas de código: isso influenciou tudo que veio depois.

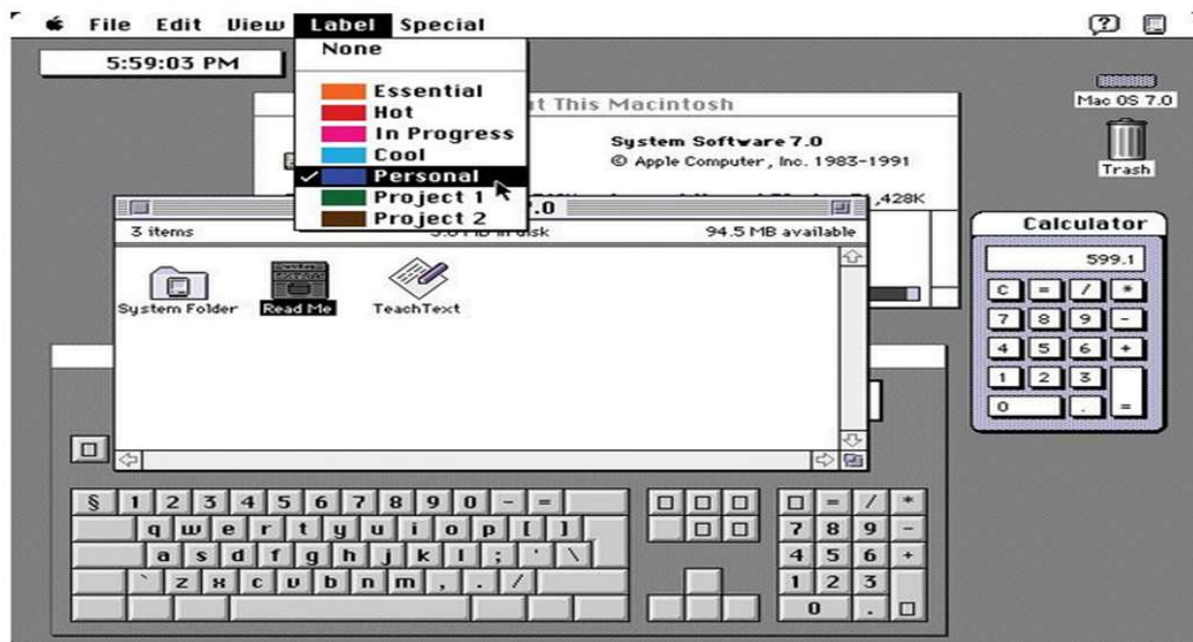
1986: Sistema operacional do Mac



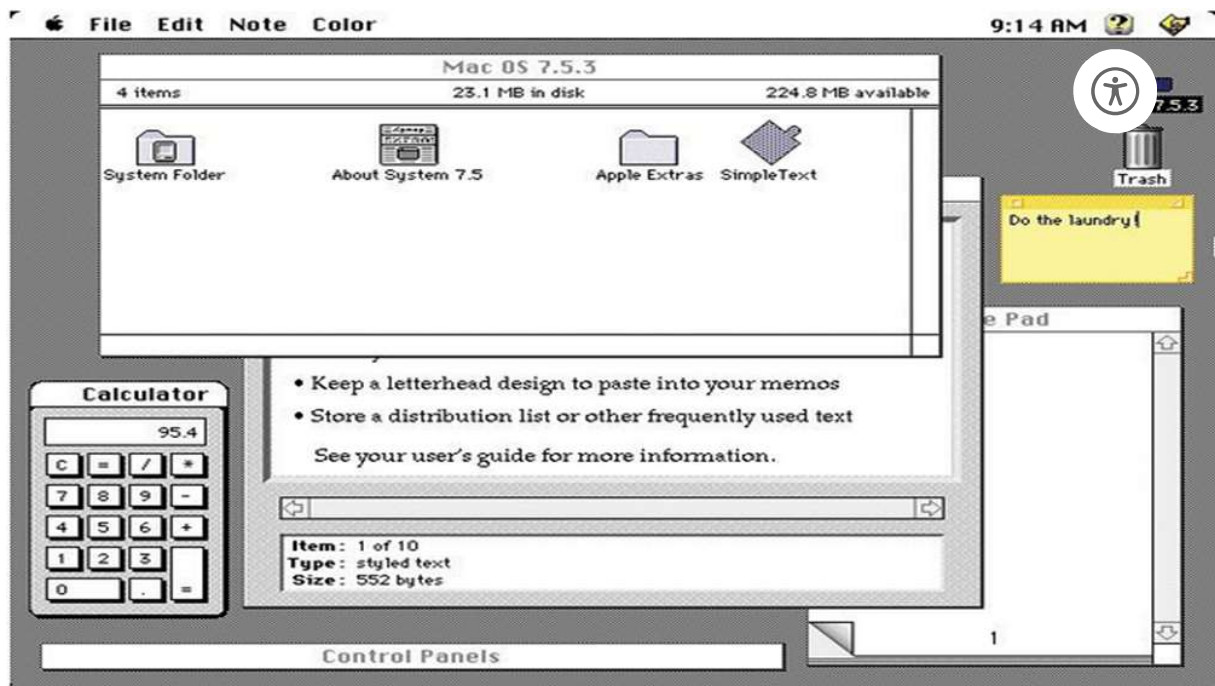
1987: Sistema Operacional do Mac



1991: Sistema Operacional do Mac

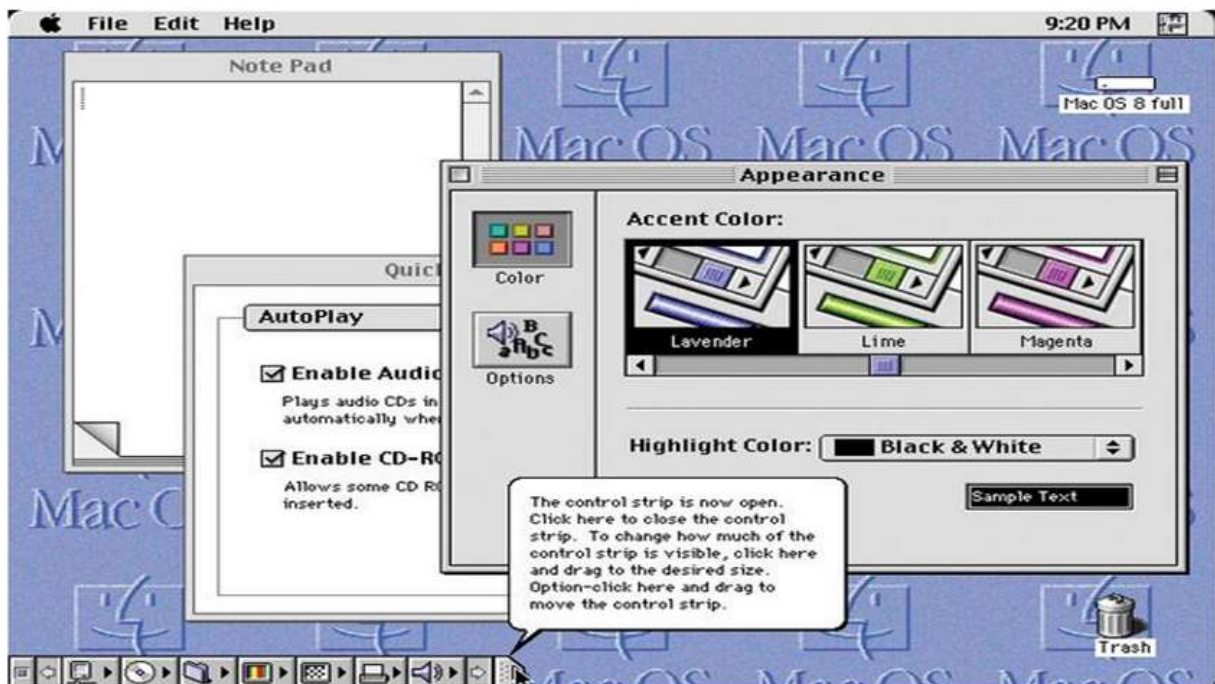


1994: Sistema Operacional do Mac



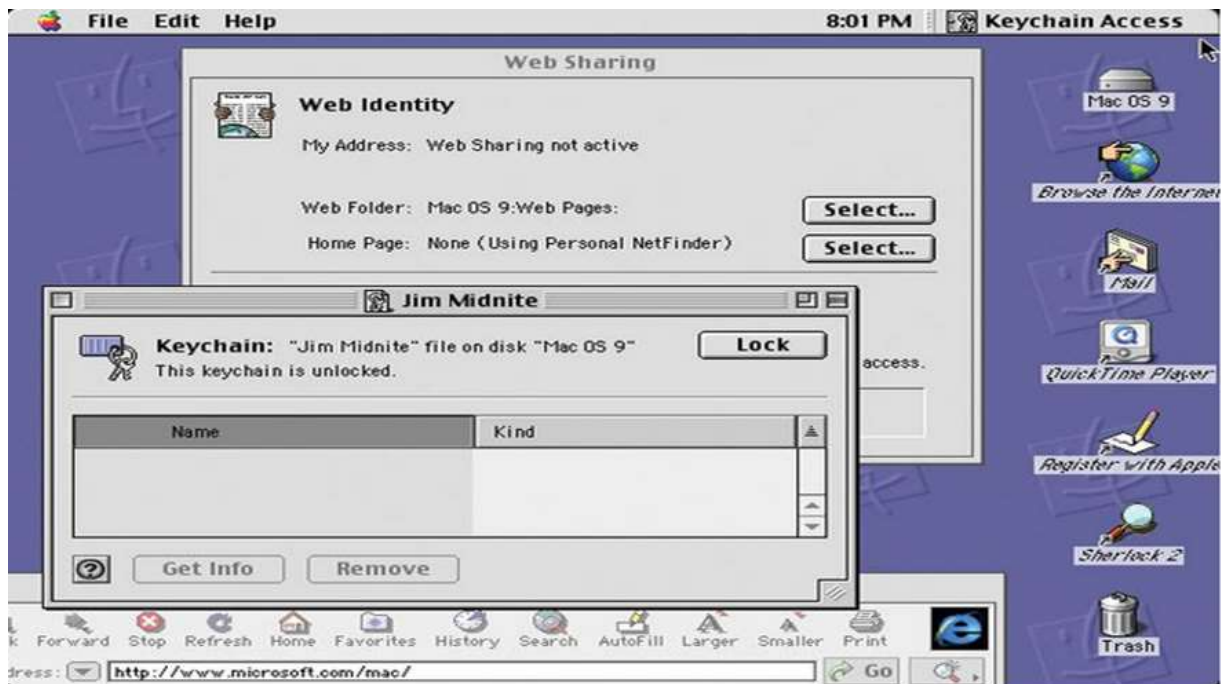
1997: Sistema Operacional do Mac

Nesta versão há muitas novidades: relógio na tela, minimizar janelas, menus hierárquicos, o computador poderia se conectar à Internet.



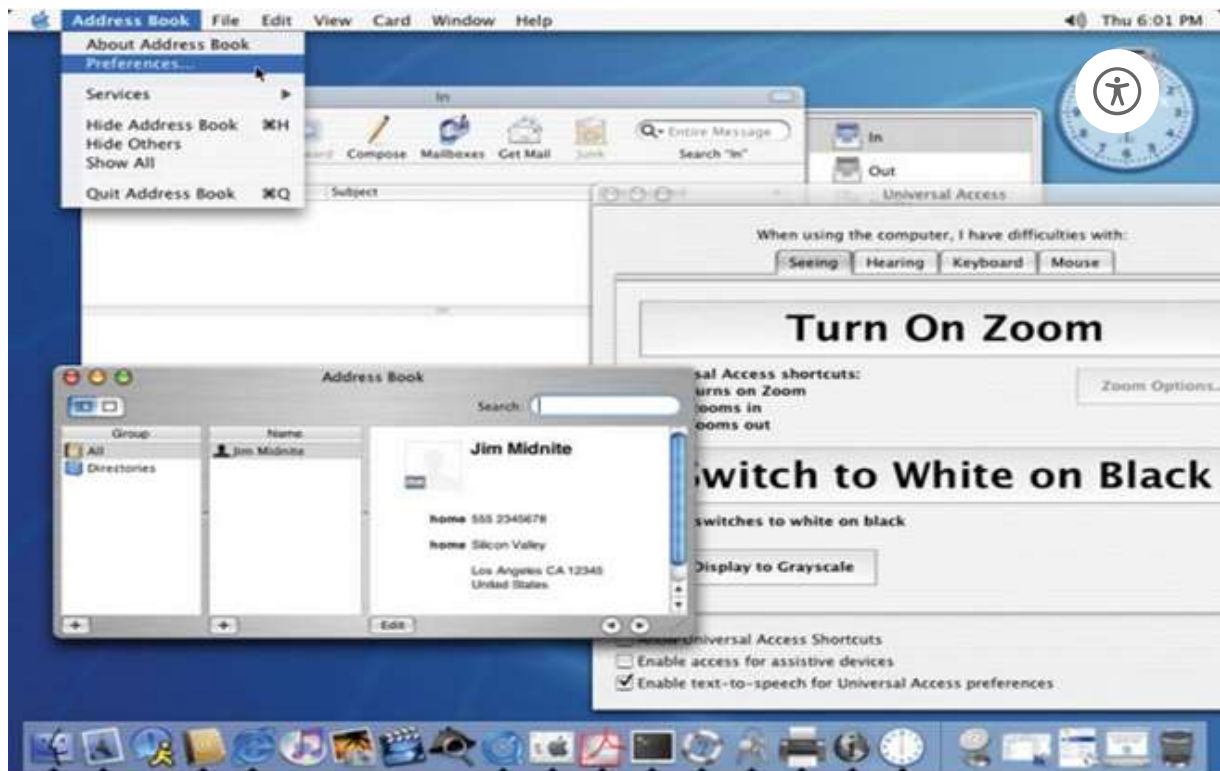
1999: Sistema Operacional do Mac

Atualização pela Internet, sistema de senhas e um sistema de busca no HD e na Internet.



Versões de 2001 e 2002

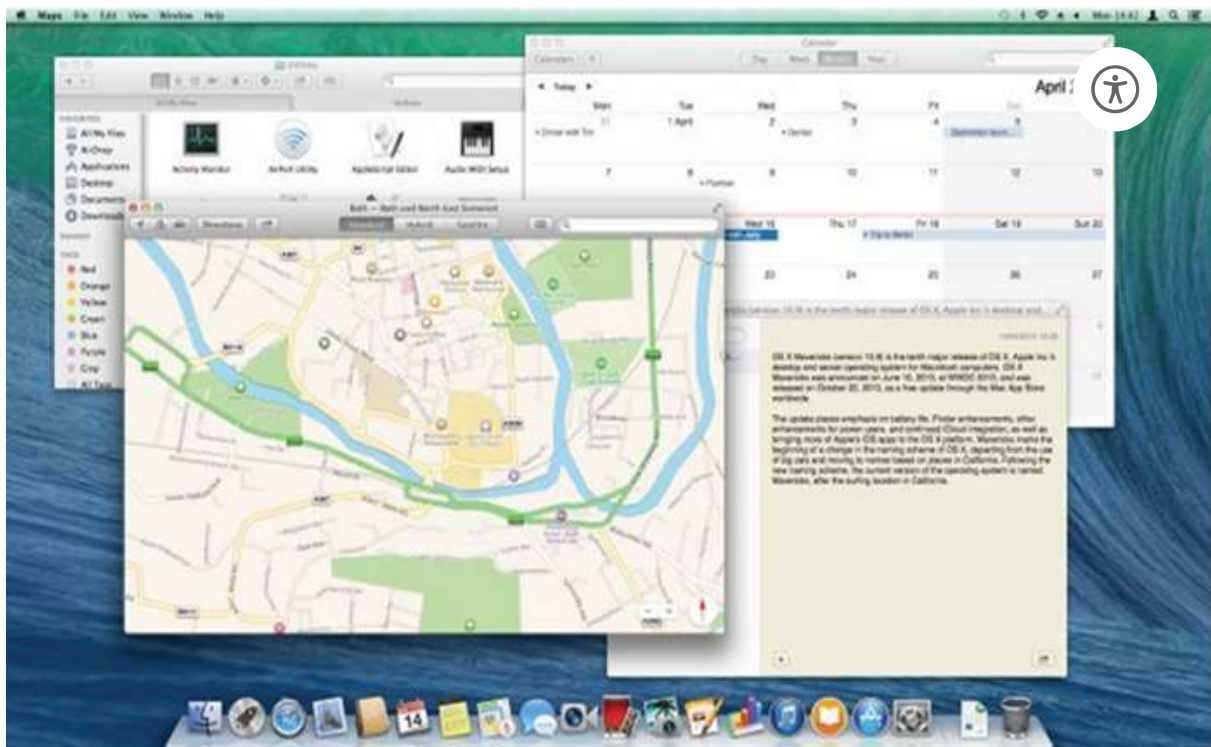




Versões de 2007



Versões de 2013



Para que serve um sistema operacional?

Cada computador possui normalmente um só processador.


O uso do processador deve ser distribuído entre os programas presentes no sistema, de forma que cada um deles possa executar na velocidade adequada para cumprir suas funções sem prejudicar os outros.

Também a memória RAM deve ser gerenciada e distribuída de forma justa entre as aplicações.

Gerenciar as impressões em computadores em rede, por exemplo, é uma tarefa necessária que o sistema operacional faz

Tipos de sistemas operacionais

Um sistema operacional visa abstrair o acesso e gerenciar os recursos de hardware, provendo aos aplicativos um ambiente de execução abstrato, no

qual o acesso aos recursos se faz através de interfaces simples, independentes das características e detalhes de baixo nível, e não  os conflitos no uso do hardware são minimizados.

Os sistemas operacionais podem ser classificados segundo diversos parâmetros e perspectivas, como tamanho, velocidade, suporte a recursos específicos, acesso à rede, etc.

Desktop: um sistema operacional “de mesa” é voltado ao atendimento do usuário doméstico e corporativo para a realização de atividades corriqueiras, como edição de textos e gráficos, navegação na Internet e reprodução de mídias simples. Suas principais características são a interface gráfica, o suporte à interatividade e a operação em rede. Exemplos de sistemas desktop são o Windows 7, MacOS X e Linux.

Servidor: um sistema operacional servidor deve permitir a gestão eficiente de grandes quantidades de recursos (disco, memória, processadores), impondo prioridades e limites sobre o uso dos recursos pelos usuários e seus aplicativos. Normalmente um sistema operacional servidor também tem suporte a rede e multiusuários.

Distribuído: em um sistema operacional distribuído, os recursos de cada máquina estão disponíveis globalmente, de forma transparente aos usuários.

Ao lançar uma aplicação, o usuário interage com sua janela, mas não sabe onde ela está executando ou armazenando seus arquivos: o sistema é quem decide, de forma transparente.

Multiusuário: um sistema operacional multiusuário deve suportar a identificação do proprietário de cada recurso dentro do sistema (arquivos, processos, áreas de memória, conexões de rede) e impor regras de controle de acesso para impedir o uso desses recursos por usuários não autorizados. Essa funcionalidade é fundamental para a segurança dos sistemas

operacionais de rede e distribuídos. Grande parte dos sistemas atuais são multiusuários.



De rede: um sistema operacional de rede deve possuir suporte à operação em rede, ou seja, a capacidade de oferecer às aplicações locais recursos que estejam localizados em outros computadores da rede, como arquivos e impressoras. Ele também deve disponibilizar seus recursos locais aos demais computadores, de forma controlada. A maioria dos sistemas operacionais atuais oferecem esse tipo de funcionalidade.


Embutido: um sistema operacional é dito embutido (*embedded*) quando é construído para operar sobre um hardware com poucos recursos de processamento, armazenamento e energia. Aplicações típicas desse tipo de sistema aparecem em máquinas industriais e automotivas, equipamentos eletrônicos de uso doméstico (leitores de DVD, eletrônicos, etc.).

Ciclo de execução

“Quando a CPU quer ler uma palavra, ou da memória ou de uma porta de E/S, ela coloca o endereço necessário nas linhas de endereço do barramento e, então, envia um sinal READ em uma linha de controle do barramento. Uma segunda linha de sinal é usada para dizer se é necessário espaço de E/S ou espaço de memória. Se for espaço de memória, a memória responderá a requisição. Se for espaço de E/S, é o dispositivo de E/S que responderá.”

Um **programa se torna um processo** quando um arquivo executável é carregado na memória.

Existem duas técnicas comuns para carregar arquivos executáveis.

Uma é clicar duas vezes em um ícone que representa o arquivo executável, a outra é digitar o nome do arquivo executável na linha de comando. 

Embora dois processos possam estar associados ao mesmo programa, são considerados duas sequências de execução separadas.

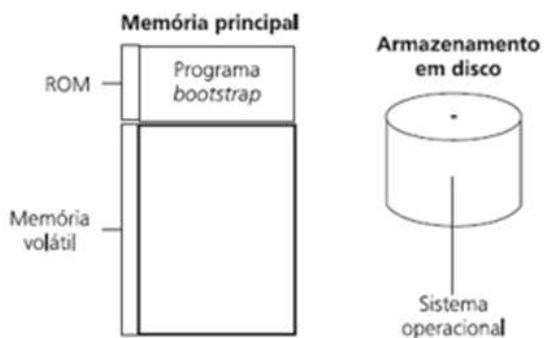
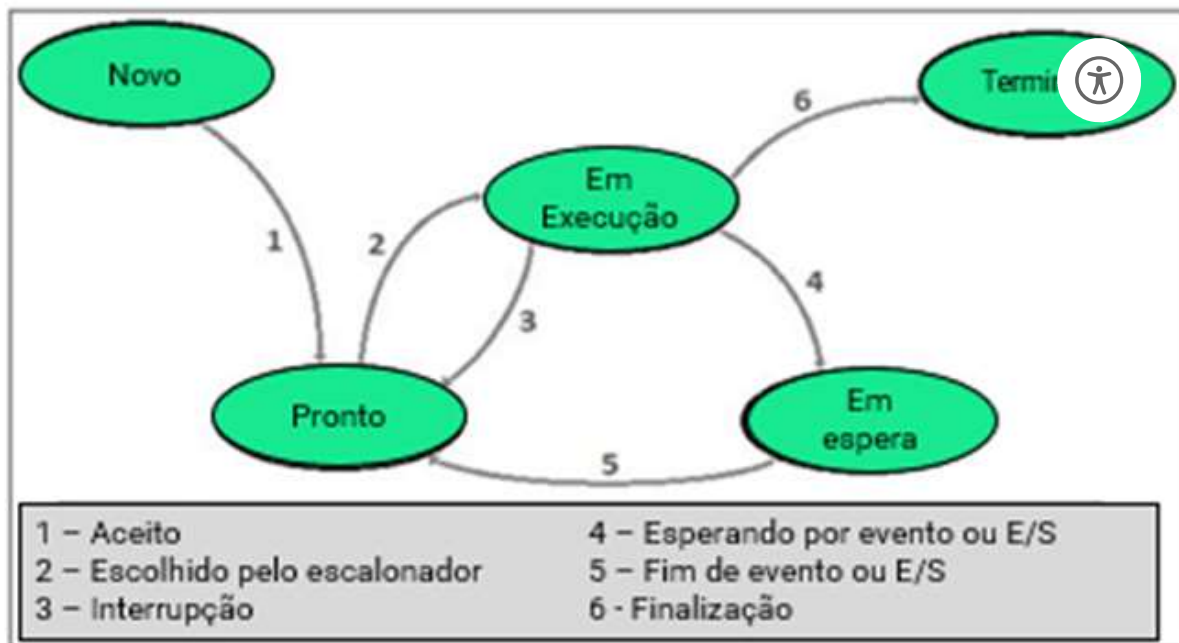
Por exemplo, vários usuários podem estar executando cópias diferentes do programa de e-mail ou o mesmo usuário pode invocar várias cópias do programa do navegador da web.

Cada um desses é um **processo separado**; e, embora as seções de texto sejam equivalentes, variam as seções de dados, heap e pilha (estrutura do processo).

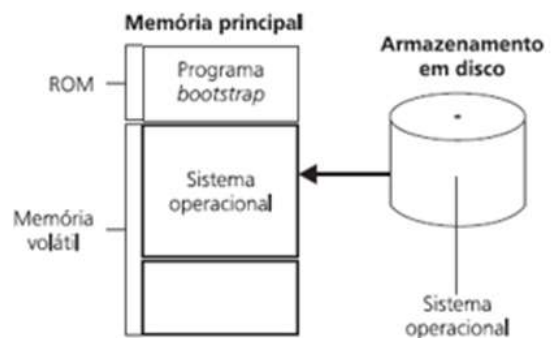
Quanto às fases do ciclo de vida de um processo, podemos dividi-las da seguinte forma:

- **Novo:** Um novo processo foi adicionado à fila de execuções, então a CPU deverá buscar os dados necessários para executar o processo;
- **Pronto:** O processo encontra-se pronto para ser executado, nesse ponto ele pode ser enviado para a CPU pois os dados básicos para sua execução já foram selecionados;
- **Em Execução:** O processo está na CPU sendo executado;
- **Em Espera:** O processo executa alguma operação de Entrada e Saída, ou depende de algum dado que deve ser fornecido por outro processo ou que não foi previamente carregado;
- **Terminado:** O processo encerrou seu ciclo de execução.

A figura representa de este ciclo de forma:



Passo 1: A máquina inicia executando a rotina *bootstrap*, residente na memória. O sistema operacional está armazenado em disco.



Passo 2: A rotina *bootstrap* controla a transferência do sistema operacional para a memória principal e, em seguida, lhe transfere o controle.

Exemplo de uma execução com registradores r1, r2, e r3

Qual o valor do registrador r3 depois que a seguinte sequência for executada?



MOV r1, #31

MOV r3, #12

MOV r2, #14

SUB r3, r3, r2

MUL r2, r2, r2

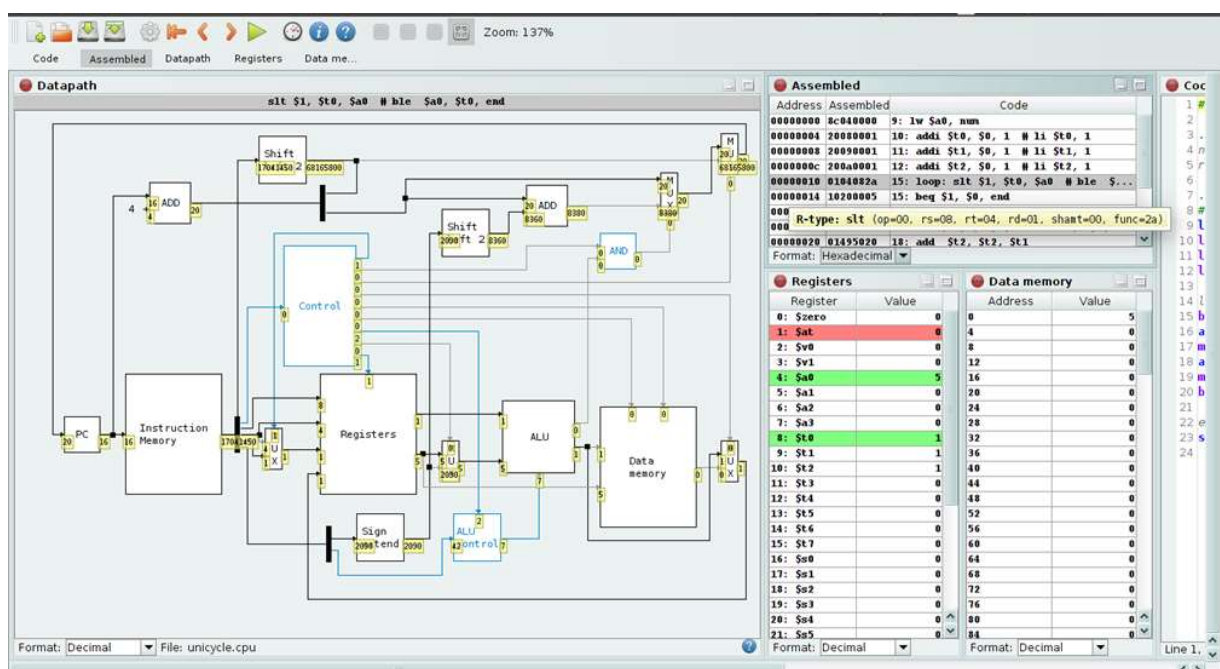
ADD r3, r1, r2



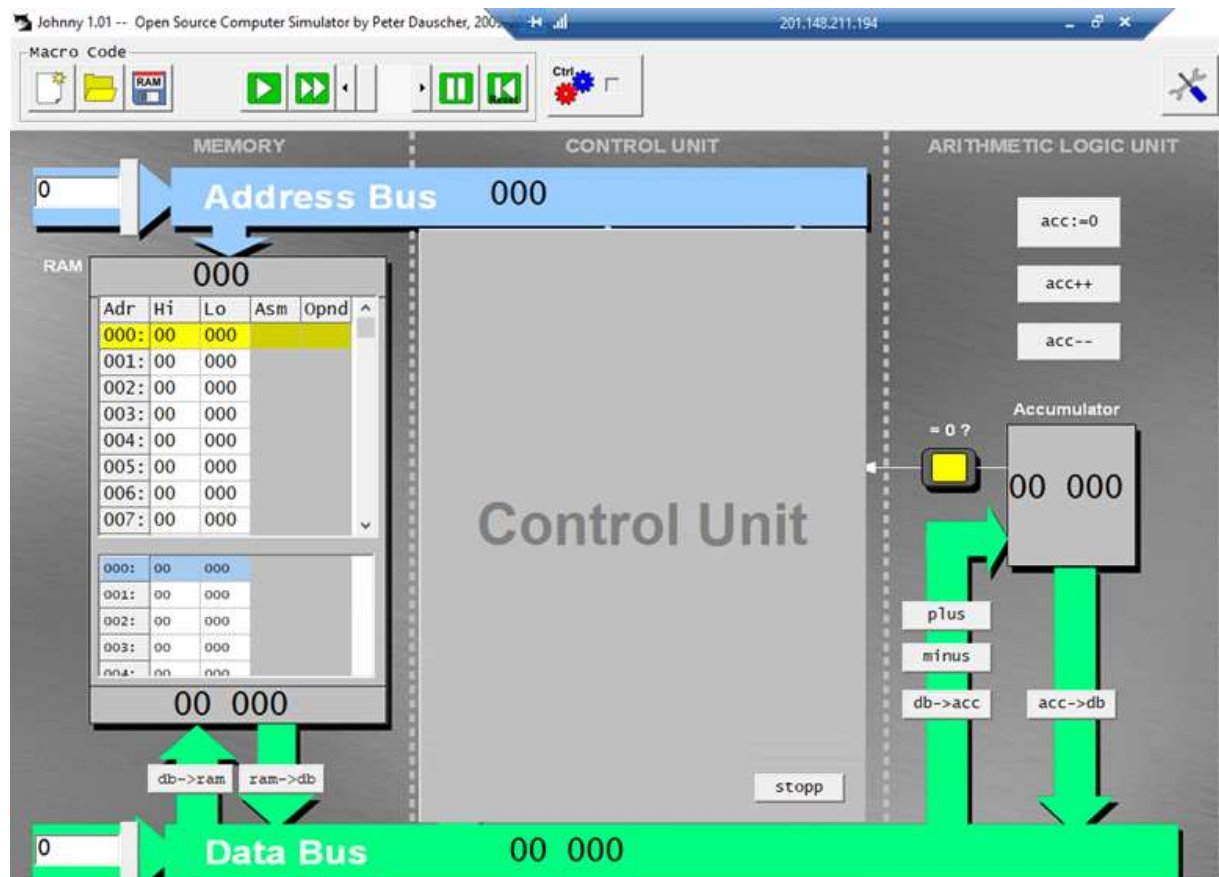
Dica:

Existem ferramentas interessantíssimas para se entender mais sobre Arquitetura de Computadores e Sistemas Operacionais. Aqui duas sugestões:

DrMips: simulador gráfico de funcionamento da CPU



Outra ferramenta muito interessante também é Johnny Simulator:



Referência Bibliográfica

BROOKSHEAR, J.G. **Ciência da Computação: uma visão abrangente**. Porto Alegre: Bookman, 2013.

TANENBAUM, A. S. **Sistemas Operacionais Modernos**. 3.ed. São Paulo: Pearson Prentice Hall, 2010.

SILBERSCHATZ, Abraham; GALVIN, Peter Baer; GAGNE, Greg. **Fundamentos de Sistemas Operacionais**. Grupo Gen-LTC, 2000.

Ir para exercício