

Introdução

Conceitos de Arquitetura e Organização

Arquitetura são os atributos visíveis ao programador.

Conjunto de instruções, número de bits usados para representação de dados, mecanismos de E/S, técnicas de endereçamento.

Por exemplo, existe uma instrução de multiplicação?

Organização é como os recursos são implementados.

Sinais de controle, interfaces, tecnologia de memória.

Por exemplo, existe uma unidade de multiplicação no hardware ou ela é feita pela adição repetitiva?

Um computador quando equipado com o *software* adequado é capaz de armazenar, processar e recuperar informação, encontrar erros de sintaxe em textos e de engajar seu usuário em muitas outras atividades bastante produtivas.

O *software* de computador pode ser dividido em duas categorias:

- **Programas do Sistema** – gerenciam a operação do próprio computador;
- **Programas de Aplicação** – que resolvem problemas para o usuário.

O mais importante dos programas de sistema é o Sistema Operacional, que controla todos os recursos do computador, e fornece base sobre o qual os programas aplicativos são escritos. Há muitos anos, ficou claro que se deveria encontrar uma maneira de afastar o usuário da complexidade do *hardware*. A forma encontrada, foi a de colocar uma camada de *software* em cima do *hardware*, para gerenciar todos os componentes do sistema, apresentando-o ao usuário com uma interface muito simples de entender e de programar. Tal interface é conhecida como **Máquina Virtual**.

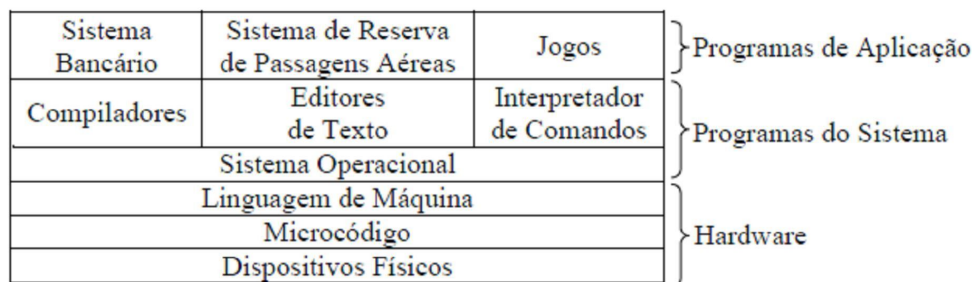


Figura 1: Modelo simplificado de Máquina Virtual

Na figura 1, observa-se que na parte de inferior está o *hardware*, o qual em muitos casos é composto de duas ou mais camadas. A mais baixa delas contém os dispositivos físicos, representado por chips de circuitos integrados, fios, fontes de alimentação, tubos de raios catódicos, etc.

Temos, a seguir, uma camada composta por um tipo de *software* bastante primitivo, que controla diretamente os dispositivos anteriormente citados. Tal *software* denominado microcódigo, composto por um conjunto de microprogramas, usualmente está gravado numa memória do tipo ROM, é, na verdade, um tipo de interpretador, que busca instruções de máquina na memória principal. O conjunto de instruções que o microcódigo interpreta é chamado **Linguagem de máquina**.

Sistema Operacional

Um sistema operacional é um programa que atua como intermediário entre o usuário e o *hardware* do computador. O propósito de um sistema operacional é fornecer um ambiente no qual o usuário possa executar

programas. O principal objetivo é, portanto, tornar o uso do sistema de computação conveniente de forma eficiente.

Parte dos problemas vem do fato de o sistema operacional realizar duas funções que não possuem nenhuma relação uma com a outra – O **sistema operacional visto como uma interface** e o **sistema operacional visto como um gerenciador de recursos**.

O Sistema Operacional como uma Interface

A arquitetura (conjunto de instruções, organização de memória, estrutura de E/S e estrutura de barramento) da maioria dos computadores é bastante primitiva e difícil de programar; em especial a parte de E/S.

Os comandos destinados a E/S são para a leitura/escrita de dados, movimento das cabeças de leitura e gravação, formatação de trilhas e inicialização, sensoramento, reinicialização e calibração do controlador e do driver de disquete (por exemplo).

Tomando como exemplo um controlador de driver de disquete, este executa dois comandos básicos – **READ** e o **WRITE**, com vários parâmetros. Tais parâmetros especificam o endereço do bloco a ser lido pelo disco, o número de setores por trilha, o modo de gravação no meio físico, entre outras coisas. Como se isto não bastasse, o programador da unidade de disco flexível tem que se preocupar com o fato do motor do disco estar ligado ou desligado. Se o motor estiver desligado, ele precisa ser ligado (retardando consideravelmente a operação), antes que o dado possa ser lido ou escrito. O motor não pode permanecer ligado por muito tempo, sob pena de danificar o disquete.

Sem precisar entrar em mais detalhes, fica claro que a maioria dos programadores não quer ou não pode envolver-se com detalhes da programação das unidades de disco flexível. No caso dos discos, uma abstração típica poderia fazer com que o disco fosse visto como uma coleção de arquivos identificados por nomes. Cada arquivo deve ser aberto para a leitura ou escrita, em seguida deve ser lido ou escrito, e finalmente deve ser fechado. Detalhes a respeito do tipo de modulação a ser usado no processo de gravação a respeito do estado corrente do motor não devem aparecer na abstração apresentada ao usuário.

O programa que esconde o verdadeiro *hardware* do usuário e apresenta-lhe um esquema simples de arquivos identificados que podem ser lidos ou escritos, é naturalmente o sistema operacional. Da mesma forma que o sistema operacional isola o usuário dos detalhes de da operação do disco, ele também trata de uma série de outras questões um tanto ou quanto incômodas, tais como as interrupções, os temporizadores, a gerência da memória, além de outras questões.

Neste aspecto, a função do sistema operacional é a de apresentar ao usuário uma **máquina estendida** (interface) ou uma **máquina virtual** equivalente ao *hardware*, porém muito mais simples de programar.

O Sistema Operacional como um Gerente de Recursos

O conceito de sistema operacional como fornecedor de uma interface a seus usuários é uma visão *topdown*. Os computadores modernos são compostos de processadores, memórias, temporizadores, discos, terminais, dispositivos de fita magnética, interfaces de rede, impressoras laser e outros dispositivos.

No caso de o computador possuir múltiplos usuários, a necessidade de gerência e proteção dos dispositivos citados anteriormente fica ainda mais aparente. Tal necessidade vem do fato de ser frequentemente necessário aos usuários fazer o compartilhamento de recursos relativamente caros como é o caso dos dispositivos de fitas e impressoras laser. Além das questões econômicas, é comum entre os usuários que trabalham juntos no mesmo sistema a necessidade de compartilhar informações.

Enfim, pelo exposto acima, mostra que a tarefa principal do sistema operacional é a de gerenciar os usuários de cada um dos recursos de máquina, contabilizando o tempo de uso de cada um e garantindo o acesso ordenado de usuários a recursos através da mediação dos conflitos entre as requisições dos diversos processos usuários do sistema.

Aspectos Históricos dos sistemas computacionais

Os sistemas operacionais e a arquitetura de computadores tiveram grande influência mútua. À medida que os sistemas operacionais foram criados e utilizados, ficou claro que as mudanças no projeto de *hardware* poderiam simplificá-los. Vamos observar como a identificação dos problemas de sistemas operacionais levou a introdução de novos recursos de *hardware*.

Sistemas em lote (Batch)

Os primeiros computadores eram máquinas exageradamente grandes, operadas a partir de um console, aos quais os dispositivos de entrada mais comuns eram leitoras de cartão e unidades de fita e os dispositivos de saída mais comuns eram impressoras de linha, unidades de fita e perfuradoras de cartão.

Nestes casos os usuários não interagiam diretamente com os sistemas de computação. Em vez disso, eles preparavam um *job* (tarefa), que consistia no programa, dados e algumas informações de controle sobre a natureza da tarefa (cartões de controle), e o submetia ao operador de computador.

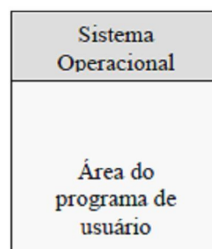


Figura 2: Layout da memória para um sistema em batch simples.

Para acelerar o processamento, os operadores reuniam os *jobs* em lotes com necessidades semelhantes e os executavam no computador como um grupo. Assim, os programadores deixavam seus programas com o operador. O operador classificava os programas em lotes com requisitos semelhantes e, à medida que o computador fica disponível, executava cada lote.

Neste ambiente de execução, a CPU muitas vezes fica ociosa, porque as velocidades dos dispositivos mecânicos de E/S são mais lentas do que as dos dispositivos eletrônicos.

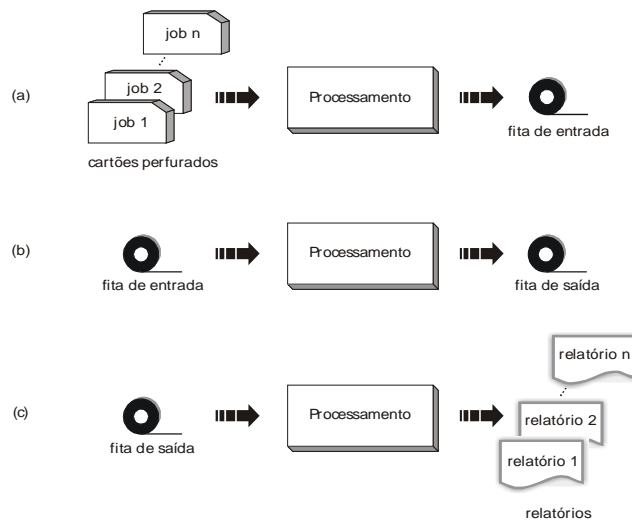
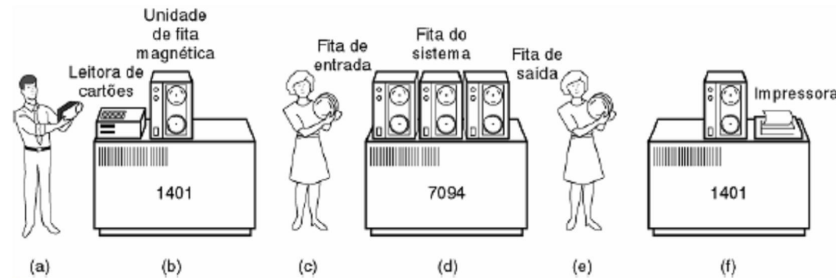


Figura 3: Esquema simplificado dos sistemas em lote.

Com o tempo, melhorias na tecnologia e a introdução de discos resultaram em dispositivos de E/S mais rápidos. No entanto, as velocidades de CPU aumentaram ainda mais, por isso o problema não só não foi resolvido, mas também exacerbado.

Com a introdução da tecnologia de discos, ao invés da tecnologia de leitora de cartões seriais, inseriu-se o **escalonamento de jobs**, com este, o acesso direto a vários *jobs* poderia ser executado para usar recursos e realizar tarefas de forma eficiente. O aspecto mais importante é a capacidade de **multiprogramação**.



Resumo:

- Traz os cartões para o 1401;
- Lê os cartões para a fita;
- Coloca a fita no 7094 que executa o processamento;
- Coloca a fita no 1401 que imprime a saída.

Figura 4: Modelo simplificado do processamento em sistemas em batch com multiprogramação.

Sistemas Multiprogramados

A **multiprogramação** aumenta a utilização de CPU organizando *jobs* de forma que a CPU sempre tenha um *job* a executar.

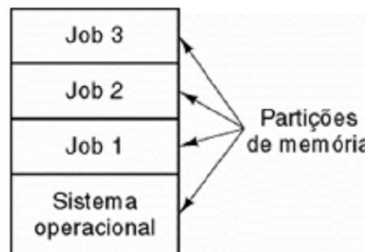


Figura 5: Layout da memória para um sistema de multiprogramação.

O sistema operacional mantém vários *jobs* na memória ao mesmo tempo. Esse conjunto de *jobs* é um subconjunto de *jobs* mantidos no pool de *jobs*. O sistema operacional escolhe e começa a executar um dos *jobs* na memória. Em um sistema não multiprogramado a CPU ficaria ociosa. Em um sistema de multiprogramação, o sistema operacional passa para outro *job* e o executa. Quando esse *job* precisa esperar, a CPU passa para outro *job* e assim por diante. Por fim, o primeiro *job* termina a espera e tem a CPU de volta. Desde que haja pelo menos um *job* para executar, a CPU nunca fica ociosa.

Sistemas de Tempo Compartilhado

Este sistema é uma extensão lógica da multiprogramação. A CPU executa vários *jobs* alternando entre eles, mas as trocas ocorrem com tanta frequência que os usuários podem interagir com cada programa durante a sua execução.

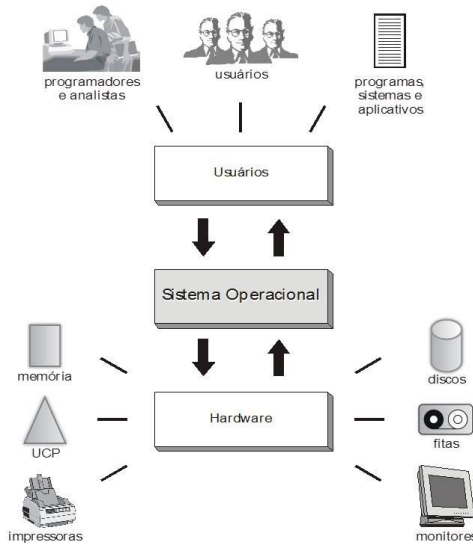


Figura 6: Modelo simplificado do processamento em sistemas de tempo compartilhado.

Um **sistema operacional de tempo compartilhado** permite aos muitos usuários compartilharem o computador ao mesmo tempo. Como cada ação ou comando neste tipo de sistema tende a ser curto, apenas um pequeno tempo de CPU é necessário para cada usuário. Como o sistema alterna rapidamente de um usuário para o outro, cada usuário tem a impressão de que todo o sistema de computação está dedicado ao seu uso, enquanto, na verdade, um computador está sendo compartilhado por muitos usuários.

Este tipo de sistema utiliza o escalonamento de CPU e a multiprogramação para fornecer a cada usuário uma pequena parte de um computador de tempo compartilhado. Um programa carregado na memória e em execução é normalmente chamado de processo. Um processo é normalmente executado durante um curto espaço de tempo antes de terminar ou de precisar realizar uma operação de E/S.

Os sistemas operacionais de tempo compartilhado são ainda mais complexos dos que os sistemas multiprogramados. Para alcançar um tempo de resposta razoável, os *jobs* talvez precisem ser passados rapidamente da memória principal para o disco que agora serve como extensão da memória principal. Um método comum de alcançar essa meta é a **memória virtual**, uma técnica que permite a execução de um *job* que talvez não esteja completamente na memória.

Devido a maioria dos sistemas hoje utilizarem tempo compartilhado e multiprogramação, estes são os temas centrais dos sistemas operacionais modernos.