

SISTEMAS OPERACIONAIS

1 - CONCEITO

É o software responsável pelo gerenciamento do computador. É também o software que efetua a conexão entre o usuário e o computador, além de efetuar o controle dos periféricos de E/S.

2 - FASES DA COMPUTAÇÃO

2.1) 1ª. Fase (1945-1955)

- Válvulas
- Linguagem de Máquina
- Fabricação do Eniac (O 1º. computador digital. Foi criado para a realização de cálculos balísticos).
- Não existia o conceito de Sistema Operacional

2.2) 2ª. Fase (1956-1965)

- Transistores e memórias magnéticas
- Aumento da Velocidade e da Confiabilidade do Processamento
- Surgimento da 1ª Linguagem de Programação (Fortran e Assembly)

2.3) 3ª. Fase (1966-1980)

- Circuitos Integrados
- Diminuição do Tamanho dos Equipamentos
- Técnica de Multiprogramação
- Técnica de Spooling
- Processamento Batch
- Surgimento do Sistema Operacional Unix (1969)

2.4) 4ª. Fase (1981-1990)

- Integração dos computadores em Larga Escala (LSI) e a integração dos computadores em Muito Larga Escala (VLSI).
- Miniaturização e barateamento do hardware
- Surgimento dos Pc's e do Sistema Operacional MS-DOS
- Multiprocessamento e Multiusuário nos Sistemas Operacionais
- Surgimento das Redes

2.5) 5ª. Fase (1991-)

- Grandes Avanços de Hardware, Software e Telecomunicações
- Surgimento dos Sistemas Especialistas
- Surgimento dos Sistemas Multimídia
- Banco de Dados
- Inteligência Artificial
- Surgimento do WINDOWS 95 (SO GUI's)

3 - TIPOS DE SISTEMAS OPERACIONAIS

3.1) MONOPROGRAMAÇÃO OU MONOTAREFA

Os Sistemas monoprogramáveis ou monotarefas se caracterizam por permitir que o processador, a memória e os periféricos fiquem dedicados a um único usuário.

3.2) MULTIPROGRAMAÇÃO OU MULTITAREFA

Os Sistemas multiprogramáveis ou multitarefas se caracterizam por permitir que vários programas sejam processados ao mesmo tempo.

Este Sistema, caracteriza-se por ter compartilhamento de memória e de processador.

Os Sistemas de Multiprogramação podem ser classificados de acordo com a integração entre os usuários. São eles:

3.2.1) SISTEMAS BATCH

Caracterizam-se por terem os programas enfileirados numa espera para serem executados sequencialmente.

3.2.2) SISTEMAS DE TEMPO COMPARTILHADO (TIME-SHARING)

Permitem a interação dos usuários com o sistema, basicamente através de terminais de vídeo e teclado (Interação On-line).

Esse Sistema, caracteriza-se por compartilhamento do processador, memória e os periféricos. O sistema cria para o usuário um ambiente de trabalho próprio, dando a impressão de que todo o sistema está dedicado exclusivamente a ele.

3.2.3)SISTEMAS DE TEMPO REAL

Os sistemas de Tempo Real (Real Time) são semelhantes ao Sistema de Tempo Compartilhado. A grande diferença é o tempo de resposta exigido na execução das tarefas.

Nos sistemas de Tempo Real não existe a idéia de fatia de tempo, utilizada pelos sistemas de tempo compartilhado. Um programa executa o tempo que for necessário, ou até que apareça outro programa prioritário em função de sua importância no sistema.

Exemplo: Controle de Tráfego Aéreo

3.3)MULTIPROCESSAMENTO

Os sistemas multiprocessáveis caracterizam-se por possuir mais de um processador sendo que os processadores podem compartilhar o mesmo Sistema Operacional ou cada um possuir o seu próprio sistema independente.

3.3.1)CARACTERÍSTICAS

O multiprocessamento permite a RECONFIGURAÇÃO e o BALANCEAMENTO do sistema, tornando-o ainda mais poderoso e eficiente.

- **RECONFIGURAÇÃO**

A Reconfiguração é a capacidade de um sistema poder continuar o processamento, mesmo se um dos processadores falhar ou parar de funcionar, embora com menos capacidade de computação.

- **BALANCEAMENTO**

O Balanceamento possibilita balancear a carga de processamento e das operações de E/S entre os diversos processadores.

4 - SISTEMAS MULTIPROGRAMÁVEIS

4.1 - CONCORRÊNCIA

- O conceito de concorrência é fundamental para compreendermos o funcionamento de Sistema Operacional Multiprogramável.
- Surgiram devido a problemas existentes nos Sistemas Monoprogramáveis, que tem baixa utilização de recursos do sistema, como processador, memória e periféricos.
- Aumento em mais de 80% no tempo de processamento.
- A utilização concorrente da CPU deve ser implementada de maneira que, quando um programa perde o uso do processador e depois retorna para continuar o processamento, seu estado deve ser idêntico ao do momento em que foi interrompido. O programa deverá continuar sua execução exatamente na instrução seguinte àquela em que havia parado, aparentando que nada aconteceu ao usuário.

4.2 - INTERRUPÇÃO

- Na execução de um programa, alguns eventos (tarefas) podem ocorrer durante seu processamento, obrigando a intervenção do Sistema Operacional. Este tipo de intervenção é chamado INTERRUPÇÃO e obriga que o fluxo de execução do programa seja desviado para uma rotina de tratamento.
- Dependendo do tipo de evento, uma interrupção pode ser classificada como INTERNA ou EXTERNA.

INTERRUPÇÃO INTERNA

- É resultado direto da execução do próprio programa, ou seja, uma instrução é responsável pela ocorrência da interrupção.
- Este tipo também é chamado, em alguns Sistemas, TRAPS(ARMADILHAS) ou EXCEÇÕES, e deve ser tratado pelo sistema para que o programa possa continuar sua execução.
- Uma Interrupção Interna ocorre quando, por exemplo, um número é dividido por zero, ou, em uma operação aritmética, o resultado exceda o limite de representação dos números (OVERFLOW).

INTERRUPÇÃO EXTERNA

- Uma interrupção externa ou do sistema é gerada pelo sistema operacional ou por algum dispositivo e, neste caso, independe do programa que está sendo executado.
- Este tipo de interrupção ocorre, por exemplo, quando um periférico avisa à CPU que está pronto para transmitir algum dado. Neste caso, a CPU deve interromper o programa que está sendo executada para atender o dispositivo.

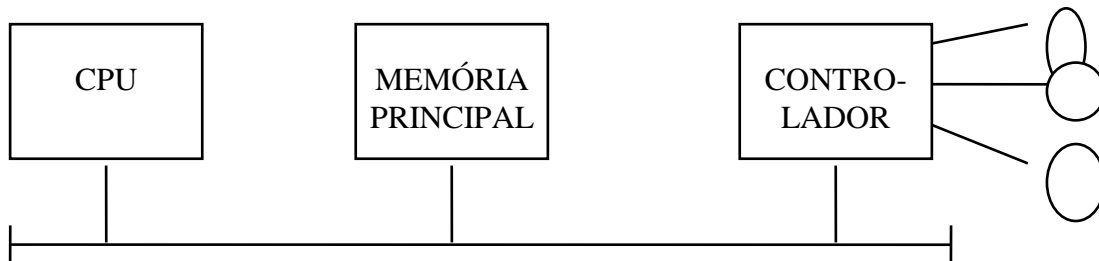
OBSERVAÇÃO 1: No momento em que a unidade de controle detecta a ocorrência de algum tipo de interrupção, o programa em execução é interrompido, e o controle é desviado para uma rotina responsável pelo tratamento da interrupção. Muitas vezes, após a execução dessa rotina, o controle deve voltar ao programa que, anteriormente, estava sendo processado.

OBSERVAÇÃO 2: A interrupção é o mecanismo que tornou possível a implementação da concorrência nos computadores, sendo o fundamento básico dos sistemas multiprogramáveis. É em função desse mecanismo que o sistema operacional sincroniza a execução de todas as suas rotinas e dos programas dos usuários, além de controlar os periféricos e dispositivos do sistema.

4.3 - OPERAÇÕES DE E/S

ANTES: Em sistemas mais primitivos, a comunicação entre a CPU e os periféricos era controlada por um conjunto de instruções especiais, chamadas de INSTRUÇÕES DE E/S, executadas pela própria CPU. Essas instruções continha detalhes específicos de cada periférico. Esse tipo de instrução limitava a comunicação do processador a um grupo particular de dispositivos.

ATUAL: A implementação de um dispositivo eletrônico chamado CONTROLADOR ou INTERFACE permitiu a CPU agir de maneira independente dos dispositivos de E/S. Com esse novo elemento, a CPU não se comunicava mais diretamente com os periféricos, mas, sim, através do CONTROLADOR. Isso implicou as instruções de E/S, por não ser mais preciso especificar detalhes de operação dos periféricos, tarefa essa realizada pelo controlador.



- Com a nova implementação, existiam duas maneiras básicas pela qual o processador controlava as operações de E/S. Na primeira, a CPU sincronizava-se com o periférico para o início da transferência de dados e, após iniciada a transferência, o sistema ficava

permanentemente testando o estado do periférico para saber quando a operação chegaria ao seu final. Este controle, chamado de E/S controlada por programa, mantinha a CPU ocupada até o término da operação de E/S. Como a CPU, geralmente, executa uma instrução muito mais rapidamente que a realização de uma operação de E/S pelo controlador, existia um desperdício de tempo da CPU.

- Outra forma derivada da anterior, porém mais inteligente, era permitir que após o início da transferência dos dados, a CPU ficasse livre para se ocupar de outras tarefas. Assim, em determinados intervalos de tempo, o sistema operacional deveria realizar um teste para saber o término ou não da operação de E/S em cada dispositivo (POLLING). Esse tipo de operação introduziu um certo grau de paralelismo de operações, já que um programa deveria ser processado, enquanto outro esperava pelo término de uma operação de E/S. ISSO PERMITIU O SURGIMENTO DOS PRIMEIROS SISTEMAS MULTIPROGRAMÁVEIS, ONDE VÁRIOS programas poderiam executar concorrentemente, já que o tempo para execução de uma operação de E/S é relativamente grande.
- A operação de E/S controlada por interrupção é muito mais eficiente que a operação de E/S controlada por programa, já que elimina a necessidade de a CPU esperar pelo término da operação, além de permitir que várias operações de E/S sejam executadas simultaneamente. Apesar disso, essa implementação ainda sobrecarregava a CPU, já que toda transferência de dados entre memória e periféricos exigia a intervenção da CPU. A solução para esse problema foi a implementação, por parte do controlador, de uma técnica de transferência de dados denominada DMA (Direct Memory Access).

TÉCNICA DE DMA

A técnica de DMA permite que um bloco de dados seja transferido entre memória e periféricos, sem a intervenção da CPU, exceto no início e no final da transferência. Quando o sistema deseja ler ou gravar um bloco de dados, são passadas da CPU para o controlador informações como: onde o dado está localizado, qual o dispositivo de E/S envolvido na operação, posição inicial da memória de onde os dados serão lidos ou gravados e o tamanho do bloco de dados. Com estas informações, o controlador realiza a transferência entre o periférico e a memória principal, e CPU é somente interrompida no final da operação. A área de memória utilizada pelo controlador na técnica de DMA é chamada de BUFFER.

4.4 - BUFFERING

A técnica de Buffering consiste na utilização de uma área de memória (buffer) para a transferência de dados entre os periféricos e a memória principal. O buffering veio permitir que, quando um dado fosse transferido para o buffer após uma operação de leitura, o dispositivo de entrada pudesse iniciar uma nova leitura. Neste caso, enquanto a CPU manipula o dado localizado no buffer, o dispositivo realiza outra operação de leitura no mesmo instante. O mesmo raciocínio pode ser aplicado para operações de gravação, onde a CPU coloca o dado no buffer para um dispositivo de saída manipular.

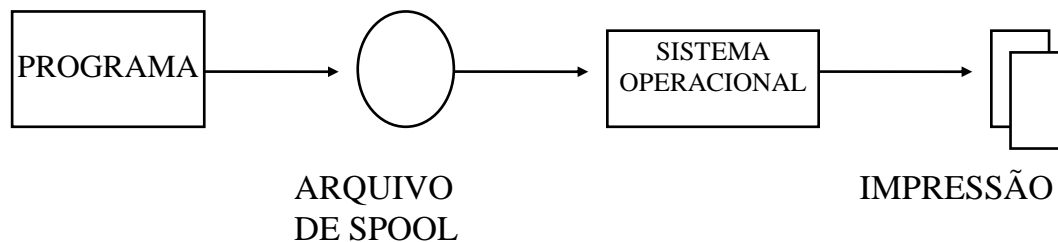
O buffering é outra implementação para minimizar o problema de disparidade da velocidade de processamento existente entre a CPU e os dispositivos de E/S. O objetivo do buffering é manter, na maior parte do tempo, CPU e dispositivos de E/S ocupados.

4.5 - SPOOLING

A técnica de Spooling (Simultaneous Peripheral Operation On-Line) foi introduzido no final dos anos 50 para aumentar a produtividade e a eficiência dos sistemas operacionais.

Naquela época, os programas dos usuários eram submetidos um a um para processamento pelo operador. Como a velocidade de operação dos dispositivos de E/S é muito lenta se comparado às operações do processador, era comum que a CPU ficasse ociosa à espera de programas e dados de entrada ou pelo término de uma impressão.

A solução foi armazenar os vários programas e seus dados, também chamados de JOBS, em um dispositivo de gravação (FITA MAGNÉTICA) e, em seguida, submetê-los a processamento. Desta forma a CPU poderia processar sequencialmente cada job, diminuindo o tempo de execução dos jobs e o tempo de transição entre eles. Esta base de processamento é chamada de SPOOLING e foi a base dos SISTEMAS BATCH.



OBSERVAÇÃO: Atualmente, a técnica de Spooling é implementada na maioria dos sistemas operacionais, fazendo com que tanto a CPU quanto os dispositivos de E/S sejam aproveitados de forma mais eficiente.

4.6 - PROTEÇÃO DO SISTEMA

Nos sistemas multiprogramáveis, onde diversos usuários compartilham os mesmos recursos, deve existir uma preocupação por parte do sistema operacional a fim de garantir integridade dos dados pertencentes a cada usuário. Problemas como um programa acessar (acidentalmente ou não) a área de memória pertencente a outro programa ou do próprio S.O., tornaria o sistema pouco confiável. Para isso, todo sistema implementa algum tipo de proteção aos diversos recursos que são compartilhados no sistema, como memória, dispositivos de E/S e CPU.

VIOLAÇÃO DE ACESSO

- Como vários programas ocupam a memória simultaneamente e cada usuário possui uma área onde dados e código são armazenados, o S.O. deve possuir mecanismos de proteção à memória fora de sua área, um erro do tipo de violação de acesso ocorre e o programa é encerrado.

CONTROLE DE COMPARTILHAMENTO DE DISPOSITIVOS DE E/S

- Há outro problema quando um programa reserva um periférico para realizar alguma operação. Nesta situação, como por exemplo, na utilização de uma impressora, nenhum outro programa deve interferir até que o primeiro programa a libere. O compartilhamento de dispositivos de E/S deve ser controlado de forma centralizada pelo S.O.

CONTROLE DE ACESSO SIMULTÂNEO A ARQUIVOS

- O compartilhamento de arquivos em disco permite que dois ou mais usuários acessem um mesmo arquivo simultaneamente. Caso o acesso concorrente não seja controlado pelo S.O., problemas de inconsistência nas informações obtidas dos arquivos podem ocorrer. Suponhamos que um usuário consulte, de forma on-line, o preço de um determinado produto em um arquivo de preços. Se, no mesmo instante um outro usuário estiver atualizando todos os preços do arquivo, o primeiro usuário poderá ler o preço desatualizado.

INTERRUPÇÃO DE PROCESSAMENTO EM LOOP

- * Um programa mal escrito, ao ganhar a CPU, pode possuir em sua estrutura um loop infinito, de forma a alocar o processador por tempo indeterminado. Para evitar este tipo de problema, a CPU possui um mecanismo denominado TIMER, que interromperá o processamento em determinados intervalos de tempo. Desta forma, o S.O. controla a utilização da CPU entre os diversos programas e, no caso de detecção do loop infinito, o programa deverá ser interrompido.