COLÉGIO ESTADUAL PROTÁSIO ALVES

COMPONENTE CURRICULAR: SISTEMAS OPERACIONAIS

Professora: Maria Helena - maira-hsilva368@educar.rs.gov.br

EMENTA

Conceito e implementação de Sistemas Operacionais
Conceito de Processos
Gerência de Processos/Processador
Comunicação, concorrência e sincronização dos processo
Gerenciamento de memória: memória virtual, paginação, segmentação e swap
Gerenciamento de arquivos
Gerenciamento de dispositivos de entrada e saída
Alocação de recursos
Sistemas Operacionais Modernos.

OBJETIVO DA DISCIPLINA: Fornecer ao discente habilidades que o permita identificar e gerenciar elementos fundamentais visando uma visão holística do Sistema Operacional. Assim, ampliar sua rede de conhecimento buscando gerenciamento dos recursos e implementação das funcionalidades do mesmo.

HABILIDADES E COMPETÊNCIAS

O aluno deverá entender a importância dos Sistemas Operacionais, além de compreender como se dá o gerenciamento dos recursos de hardware e software durante o funcionamento de um computador.

CONTEÚDO PROGRAMÁTICO

1. **Introdução aos Sistemas Operacionais**
   - 1.1. Conceitos básicos
   - 1.2. Diferenciar hardware de Software
   - 1.3. Funções do Sistemas Operacionais
   - 1.4. Tipos de Sistemas Operacionais
   - 1.5. Componentes dos Sistemas Operacionais
2. **Processos**
   - **2.1.** Definição e estrutura de processos
   - 2.2. Estados em um Processo
   - 2.3. Tipos de Processos
   - 2.4. Comunicação entre processos
   - 2.5. Escalonamento
3. **Gerência de Memória**
   - 3.1. Endereços lógicos e físicos
   - 3.2. Alocação de Memória
   - 3.3. Compartilhamento de Memória
   - 3.4. Paginação

3.5. Memória Virtual

4. Sistemas de Arquivos

4.1. Arquivos e diretórios
4.2. Alocação de arquivos
4.3. segurança e mecanismos de proteção da informação

5. Gerência de Dispositivos

5.1. Dispositivos de entrada e saída
5.2. Devices drivers
5.3. Controladores

6. **Sistemas Operacionais Distribuídos**

6.1. Comunicação síncrona e assíncrona
6.2. Modelos cliente-servidor e Peer-to-peer
6.3. Sockets
6.4. Chamada remota de procedimento

METODOLOGIA DE TRABALHO

- Aula Expositiva: Meet - explicação do conteúdo proposto e orientação na resolução da atividade
- Atividades de fixação dos conceitos trabalhados.
- Discussão dos tópicos apresentados na correção das atividades

Professora: Maria Helena - maria-hsilva368@educr;rs.gov;br

Aula 1 - 1ª semana - Sistemas Operacionais

Objetivo

- Compreender o papel do SO na visão do usuário e na visão do Hardware
- Diferenciar o Hardware do Software
- Conceituar e identificar os principais tipos de Software.

1. INTRODUÇÃO À SISTEMAS OPERACIONAL

Sistemas Operacionais

- Sistemas Operacionais é uma camada de software colocada entre o Hardware e os aplicativos, aqueles programas que fazem as tarefas dos usuários do computador. (Oliveira , 2021).
- É um Software básico de qualquer computador, pois fornece ao usuário uma interface conveniente e ao mesmo tempo gerencia o hardware controlado d e forma ordenada e eficiente o acesso ao processador, memória e dispositivos de entrada e saída dos aplicativos que os disputam. Pinto Neto (2014. p15)

RESUMO

Ao abrir um arquivo , ao usar um editor ou qualquer outro aplicativo. internet, você usou um Sistema Operacional Ele é responsável por facilitar o uso do computador, mostrando uma interface muito mais amigável para o usuário, assim, poder operar a máquina, mesmo sem saber o funcionamento do mesmo. Portanto, o SO é um mecanismo que faz o computador funcionar, ele trabalha nos bastidores como um intermediário do usuário e do programa, gerencia e orienta o hardware do computador. Ele lê e grava dados para o disco.

O Sistema Operacional como uma Máquina Virtual

A arquitetura (conjunto de instruções, organização de memória, E/S e estrutura de barramento) da maioria dos computadores a nível de linguagem de máquina é primitiva e difícil de programar, especificamente para operações de entrada e saída. É preferível para um programador trabalhar com abstrações de mais alto nível onde detalhes de implementação das abstrações não são visíveis. No caso de discos, por exemplo, uma abstração típica é que estes armazenam uma coleção de arquivos identificados por nomes simbólicos. O programa que esconde os detalhes de implementação das abstrações é o sistema operacional. A abstração apresentada ao usuário pelo sistema operacional é simples e mais fácil de usar que o hardware original. Nesta visão, a função do sistema operacional é apresentada ao usuário como uma máquina estendida ou máquina virtual que é mais fácil de programar que o hardware que a suporta.

O Sistema Operacional como um Gerenciador de Recursos

Um computador moderno é composto de vários subsistemas tais como processadores, memórias, discos, terminais, fitas magnéticas, interfaces de rede, impressoras, e outros dispositivos de E/S. Neste ponto de vista, o sistema operacional tem a função de gerenciar de forma adequada estes recursos de sorte que as tarefas impostas pelos usuários sejam atendidas da forma mais rápida e confiável possível. Um exemplo típico é o compartilhamento da unidade central de processamento (CPU) entre as várias tarefas (programas) em sistemas multiprogramados. O sistema operacional é o responsável pela distribuição de forma otimizada da CPU entre as tarefas em execução.

Função do Sistema Operacional

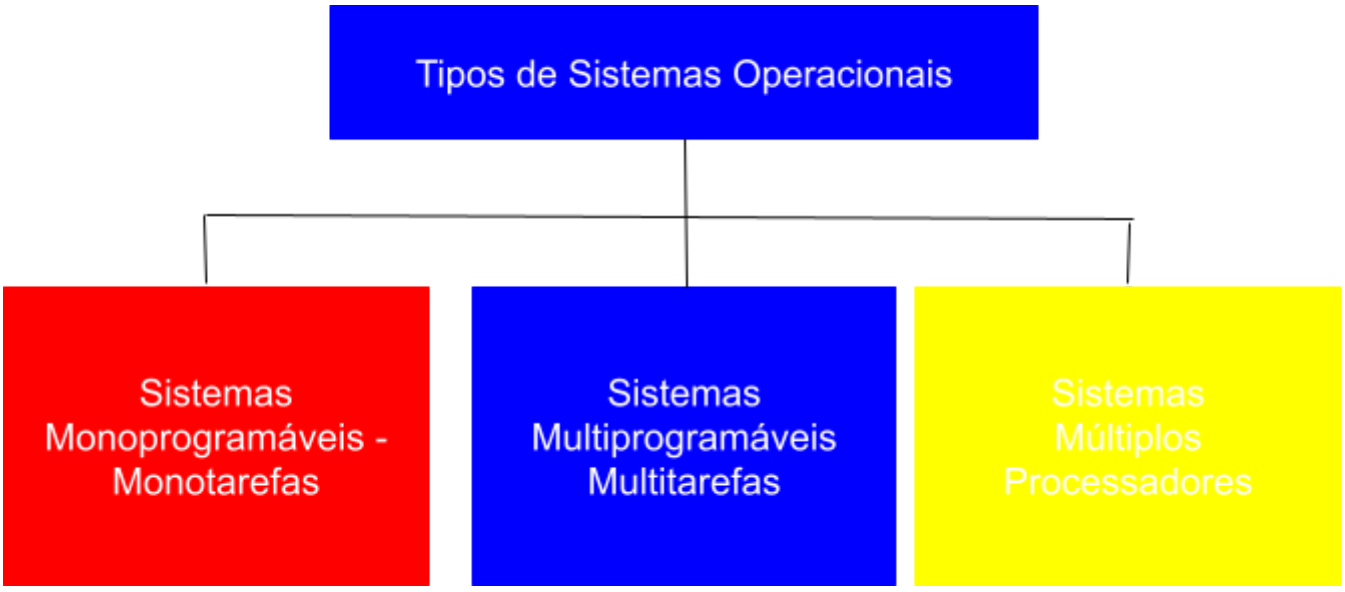
- O Sistema Operacional na verdade cria um ambiente virtual mais seguro e mais fácil de entender, operar e programar. (tanenbaum, 2008).
- Outra função básica do SO está relacionada ao controle de acesso ao recursos do sistema
- Controla quem está usando qual recurso, garantir os pedidos de recursos, medir a utilização e resolver conflitos de diferentes programas é uma tarefa básica do Sistema Operacional.
- Propiciar uma interface (de comando ou gráfica) para ao usuário se comunicar com o computador
- Gerenciar o sistema de arquivo em disco
- Dar suporte à outros programas
- Gerenciar os dispositivos de hardware

Os componentes do sistema operacional

O sistema operacional é composto por um conjunto de softwares que permitem administrar as interações com o hardware. Neste conjunto de softwares distinguem-se os seguintes elementos: o núcleo (kernel), que representa as funções fundamentais do sistema operacional tais como gestão da memória, processos, arquivos, entradas/saídas e das funcionalidades de comunicação; o Intérprete de comandos (shell), ou seja, a camada externa, por oposição ao núcleo, que permite a comunicação com o sistema operacional por meio de uma linguagem de comandos para o usuário pilotar os periféricos ignorando muitas das características do hardware como, por exemplo, a gestão dos endereços físicos; e o sistema de arquivos (file system), que permite registrar os arquivos em arborescência.

Tipos de Sistemas Operacionais

- A maneira como os sistemas operacionais realizam as tarefas pode ser um elemento importante para determinar a sua classificação.
- Um sistema Operacional pode ser classificado de acordo com o número de programas executados simultaneamente e pelo número de processadores por ele controlado.



Sistemas Monoprogramáveis - Monotarefa

- MS-DOS foi um dos primeiros Sistemas Operacionais comerciais monoprogramáveis

- Nos sistemas monoprogramáveis todo o recurso de hardware está à serviço de um único processo
- À restrição de executar um único programa causa causa ineficiência dos recursos de máquina
- Uma tarefa simples pode significar uma perda de tempo considerável para o usuário.

Sistemas Multiprogramáveis - Multitarefas

- A evolução do Sistema Operacional de monotarefas, introduziu o conceito de multiprogramação, um ambiente onde uma única UCP é compartilhada juntamente com uma área de memória limitada.
- Nesse ambiente, o sistema Operacional altera a execução dos programas, dando à falsa impressão ao usuário de que os programas estão sendo executados simultaneamente.

Sistemas com Múltiplos Processadores

- Os sistemas com múltiplos processadores são semelhantes aos sistemas multitarefas. O que os diferenciam por apresentarem mais de dois processadores .
- Essa característica permite a execução de tantos programas quanto forem o número de processadores do sistema computacional.
- Sistemas com múltiplos processadores foram desenvolvidos para atender as aplicações que exijam um grande poder de processamento.

Software

- É um conjunto de instruções lógicas que executam determinadas tarefas, um computador sem programas, não possui nenhuma utilidade. O software vai torar o computador produtivo, permitindo ao usuário armazenar dados, receber e enviar mensagens, ouvir música, assistir e produzir vídeos, entre tantas outras aplicações.
- O termo Software é usado para definir o programa.

Classificação do Software

- Software de Sistemas:São os programas responsáveis por controlar o hardware: Exemplo. Sistemas Operacionais. Linux, Windows e Android. O teclado por exemplo, precisa de controle para que, necessita que o sistema operacional possa receber os dados que o usuário digita.
- Software de Programação: São programas para criar outros programas, utilizados por desenvolvedores e software ou programadores, como linguagem de programação, compiladores e outra ferramenta para criar software. Ex: Java, C + +, C# e outras.
- Sistema de Aplicação: Esses programas também chamados de aplicativos, podem ser de negócios específicos como programa de contabilidade, edição de texto, multimídia e telefonia, desenho e imagem entre outros. ex: Skype, Strike, counter, Ms Office, etc,,,

Hardware

- Conjunto de componentes físicos de suma importância nessa engrenagem chamada computador, ele é quem fornece subsídios que visa ampliar o recursos. Potencializar as ações que usuário necessita da ferramenta. A partir do processador, memórias, placa mãe, entre outros componentes que influenciam diretamente no desempenho e eficiência da máquina. Quanto maior for à tecnologia aplicada em determinado componente, maior será o desempenho e eficácia no desempenhar das tarefas.

Aula2 -

Objetivo da Aula

- Diferenciar Programa X Processo
- Identificar o contexto de um processo
- Conceituar e descrever estado de um processo
- Conceituar Multiprogramação
- Distribuir execução de modo de usuário do modo de núcleo
- conceituar escalonamento

.Processos

Programa é o código fonte escrito em alguma linguagem de programação (por exemplo, C, Pascal, Fortran), já executável é esse código traduzido para a linguagem específica (binário) da máquina real onde ele será executado.

Um processo pode ser definido como um fluxo de controle sequencial e seu espaço de endereçamento, ou seja, é a execução de um programa junto com os dados usados por ele. Processo, de forma resumida, é então um programa em execução. Como exemplo, o mesmo programa pode ser executado ao mesmo tempo por dois usuários, gerando dois processos distintos. Além disso, na sua execução um programa pode originar vários processos.

O contexto de um processo é o conjunto de dados necessários a sua execução no tempo como, por exemplo, a identificação do processo (PID – Process Identification), contador de programa (PC Program Counter), ponteiros (SP Stack Pointer), as variáveis e dados armazenados em memória, o conteúdo dos registradores da CPU, a lista de arquivos que estão sendo utilizados, tempo de CPU disponível e prioridade de execução, entre outros. Essas informações são fundamentais para que o processo interrompido pelo escalonador possa voltar a executar exatamente a partir do ponto de parada sem perda de dados ou inconsistências. Tais informações são armazenadas em estruturas de dados conhecidas como Tabela de Processos.

O processo, quando em execução na CPU, está no estado executando (run). Ao terminar sua quota de tempo para execução, o escalonador o interrompe e é colocado no estado pronto. Ao chegar novamente sua vez de executar, o escalonador invoca (acorda) o processo permitindo sua continuidade. Um processo pode ser bloqueado se os dados de entrada ainda não estiverem disponíveis e ele não poder continuar sua execução, e isto pode ocorrer pelo sinal enviado por outro processo ou pela própria decisão do escalonador.

O processo é desbloqueado por um evento externo (chegada dos dados, sinalização de outro processo), e então passa ao estado pronto. Para efetuar o compartilhamento da CPU entre processos, o sistema operacional possui duas filas de controle: a de processos prontos (ReadyList) e a de processos bloqueados (BlockedList). A manipulação dessas filas depende da política de escalonamento adotada pelo sistema, que é um critério para determinar, entre os diversos processos no estado pronto, qual o próximo processo a executar. Concorrência Num sistema multitarefa, normalmente vários programas são executados simultaneamente e o número de processos é maior que o de CPUs. Um dos problemas mais difíceis na

administração de recursos está relacionado ao fato de muitos processos existirem simultaneamente, ocorrendo em alguns casos uma disputa entre processos pelo uso do mesmo recurso. No caso de haver disputa, dizemos então que esses processos são concorrentes. Por exemplo, processos concorrentes frequentemente acessam o mesmo arquivo, e o sistema operacional deve garantir que um processo não possa alterar os dados que outro processo esteja usando.

Estados de um processo

Em um sistema multiprogramável, um processo não deve alocar a CPU com exclusividade, de forma que possa existir um compartilhamento no uso do processador. Os processos passam por diferentes estados ao longo do seu processamento, em função de eventos gerados pelo sistema operacional ou pelo próprio processo. Um processo pode
encontrar-se em três estados diferentes:

• **Execução** (running) Um processo é dito no estado de execução quando está sendo processado pela CPU. Os processos se alternam na utilização do processador seguindo uma política estabelecida pelo sistema operacional.

• **Pronto** (ready) Um processo está no estado de pronto quando ele tem condições lógicas de executar e apenas aguarda para ser executado. O sistema operacional é responsável por determinar a ordem e os critérios pelos quais os processos em estado de pronto devem fazer uso do processador. Esse mecanismo é conhecido como escalonamento. Em geral, existem vários processos no sistema no estado de pronto organizados em listas encadeadas.

• **Espera** (wait) Um processo no estado de espera aguarda por algum evento externo ou por algum recurso para prosseguir seu processamento, ou seja, ele não tem condições lógicas de executar. O sistema organiza os vários processos no estado de espera também em listas encadeadas, associadas a
cada tipo de evento.

Um processo muda de estado durante seu processamento em função de eventos originados por ele próprio (eventos voluntários) ou pelo sistema operacional (eventos involuntários). Basicamente, existem quatro mudanças de estado que podem ocorrer a um
processo:

Pronto⇒ Execução

Após a criação de um processo, o sistema o coloca em uma lista de processos no estado de pronto, onde aguarda uma oportunidade para ser executado. Cada sistema operacional tem seus próprios critérios e algoritmos para a escolha da ordem em que os processos serão executados.

Execução⇒ Espera

Um processo em execução passa para o estado de espera por eventos gerados pelo próprio processo, como uma operação de E/S, ou por eventos externos (sistema operacional suspende por um período de tempo a execução do processo).

Espera⇒ Pronto

Um processo no estado de espera passa para o estado pronto quando a operação solicitada é atendida ou o recurso esperado é concedido. Um processo em estado de espera sempre terá que passar pelo estado de pronto antes de poder ser novamente selecionado para execução.

Execução⇒ Pronto

Um processo em execução passa para o estado de pronto por eventos gerados pelo sistema, como o término da fatia de tempo que o processador possui para sua execução. Um processo em estado de pronto ou de espera pode não se encontrar na memória principal. Esta condição ocorre quando não existe espaço suficiente para todos os processos na memória principal e parte do contexto do processo é levada para a memória secundária. Uma técnica conhecida como swapping retira processos da memória principal e os traz de volta seguindo critérios de cada sistema operacional.

Tipos de processo

Além dos processos do usuário, a CPU também executa processos do sistema. Estes executam sempre, com certa prioridade, concorrendo com os processos do usuário. Os processos em execução, do usuário, podem assumir dois tipos diferentes, de acordo com suas características de uso de CPU e periféricos:

Processo CPU-bound

• É aquele processo que utiliza muito a CPU.
• Ele ganha uma fatia de tempo e a utiliza por inteiro, sem desperdiçar nenhum tempo.
• É o caso de programas científicos, de cálculo numérico, estatística, matemática, e também na área de simulação.
• Normalmente fazem pouca ou nenhuma entrada de dados, e muito processamento.

Processo I/O-bound

• é o tipo de processo que utiliza muito mais E/S do que CPU.
• Aplicações em Banco de Dados, onde se faz consultas e atualizações constantes em arquivos em disco são um bom exemplo deste tipo de processo.
• De acordo com essas características, podemos dizer que este tipo de processo permanece mais tempo em espera (tratando interrupções) do que propriamente em execução, ocupando a CPU por períodos mínimos de tempo.

Comunicação entre processos

Frequentemente os processos do sistema precisam se comunicar com outros processos para a execução de determinadas tarefas. Neste contexto, dois paradigmas são mais utilizados: memória compartilhada (shared memory) e troca de mensagem (message passing). No primeiro, os processos têm acesso a uma área comum para leitura e escrita de informações. A segunda abordagem consiste no uso de primitivas para o envio/recebimento (send/receive) de mensagens entre processos. Quando há compartilhamento de recursos entre dois ou mais processos, diversas situações estranhas podem ocorrer durante o processamento, e isso devido principalmente à interrupção provocada pelo escalonador do sistema operacional (preempção). Por exemplo, a atualização "descontrolada" de uma variável global compartilhada pode gerar resultados inconsistentes. Este tipo de situação é conhecida como situações de corrida (race conditions), porque o resultado final em algumas situações especiais depende da "competição" entre os processos e não do algoritmo implementado. Uma tentativa de resolver este problema garante que os dados compartilhados só possam ser manipulados por um processo de cada vez, e esta estratégia é chamada de exclusão mútua (mutual exclusion) entre processos. A parte do programa que referencia dados compartilhados que possam sofrer condições de corrida com outros processos é denominada de região ou seção crítica (critical section) do processo. Portanto, uma maneira de evitar situações de corrida é garantir que dois ou mais processos nunca estejam executando simultaneamente na sua seção crítica. Entretanto, esta restrição não é suficiente e uma boa solução para o problema envolve quatro requisitos básicos:

● Dois processos não podem estar simultaneamente acessando suas regiões críticas;
● A velocidade e quan[dade de processadores não deve interferir no algoritmo;
● Um processo que esteja fora de sua seção crítica não deve impedir outro processo de usá-la;
● Um processo não deve esperar indefinidamente para acessar sua região crítica (evitar situação de starvation).

Escalonamento

A entidade responsável pelo escalonamento é o escalonador (scheduler), que é quem determina qual processo deve sair ou ir para a CPU em determinado momento. É ele o elemento responsável pela alocação de processos a processadores, definindo sua ordem de execução. A política de escalonamento (scheduling) é um problema complexo e depende do tipo de sistema suportado e da natureza das aplicações. Em sistemas do tipo lote (batch), o escalonamento era feito simplesmente selecionando o próximo processo na fila de espera, já em sistemas multiusuário de tempo repartido geralmente combinados a sistemas em lote, o algoritmo de escalonamento deve ser mais complexo em virtude da existência de diversos usuários interativos solicitando serviços, e da execução de tarefas em segundo plano (background). Um escalonador poderia funcionar da seguinte forma: quando a CPU tornasse disponível, o primeiro elemento da fila de processos prontos é retirado e inicia sua execução. Caso seja bloqueado, este irá para o final da fila de processos bloqueados. Se à sua quota de execução se esgotar, este será retirado da CPU e colocado no final da lista de processos prontos.
O escalonamento tem como principais objetivos:
● maximizar a u[lização do processador;
● maximizar o número de processos completados por unidade de tempo;
● garantir que todos os processos recebam o processador;
● minimizar o tempo de resposta para o usuário.

Os tipos de escalonamento

- Não-preemptivo: ocorre quando o processo que está executando não pode ser interrompido. Esse tipo de escalonamento estava presente nos primeiros sistemas multiprogramáveis, onde predominava o processamento em batch, e as políticas que implementam escalonamento não-preemptivo não são aplicáveis a sistemas de tempo compartilhado. Os primeiros sistemas de programas em lote (batch) inicialmente liam todas as instruções sequencialmente e depois as executavam uma após a outra.
- Preemptivo: o processo pode ser retirado do processador que está executando. Permite atenção imediata aos processos mais prioritários (tempo real), melhores tempos de resposta (tempo compartilhado) e compartilhamento uniforme do processador. Existe também a multitarefa cooperativa onde não existe a figura do escalonador, pois nesse caso são os aplicativos que cooperativamente se revezam no uso dos recursos de CPU e memória. Porém, nesse caso, se determinado aplicativo apresentar algum problema trava o sistema. Era a multitarefa cooperativa que dava aos aplicativos escritos para Windows 3.X a impressão de ser multitarefa, muito embora todos eles rodassem sobre o DOS, um sistema monotarefa. Era também esse o principal motivo das "travadas"