

Disciplina: Sistema Operacional
Profa: Dra. Glenda Botelho
Aluno(s): Mateus Alves Araújo.

Tema: Introdução ao SO - Processos e Threads

1) Defina Sistema Operacional.

Um sistema operacional é um software essencial que atua como intermediário entre os componentes de hardware de um computador e os programas de software que são executados nesse hardware. Ele desempenha um papel crítico na gestão e coordenação de todas as atividades do sistema de computador. Os principais pontos sobre sistemas operacionais, conforme discutidos no livro "Sistemas Operacionais Modernos" de Andrew S. Tanenbaum, incluem:

Gerenciamento de Recursos

- Alocação e gerenciamento de recursos de hardware.
- Garantia de acesso adequado a recursos sem conflitos

Abstração de Hardware

- Fornecimento de uma camada de abstração sobre o hardware.
- Facilitação do desenvolvimento de software e portabilidade.

Gerenciamento de Processos

- Controle da execução de processos.
- Escalonamento e coordenação justa dos processos.
- Criação, término, suspensão e retomada de processos.

Gerenciamento de Memória

- Controle do uso da memória física.
- Alocação e desalocação de espaço de memória.
- Proteção de processos uns contra os outros.

Sistema de Arquivos

- Organização e armazenamento hierárquico de dados.
- Leitura e gravação de dados em dispositivos de armazenamento.

Disciplina: Sistema Operacional

Profa: Dra. Glenda Botelho

Aluno(s):Mateus Alves Araújo.

- Gerenciamento de Dispositivos

- Facilitação da comunicação entre software e hardware.
- Controle de entrada e saída de dados.
- Gerenciamento de drivers de dispositivo.

Segurança e Controle de Acesso

- Implementação de mecanismos de segurança.
- Proteção contra acessos não autorizados.
- Autenticação de usuários e controle de acesso a recursos.

-Interface com o Usuário

- Fornecimento de uma interface para interação.
- Pode ser uma linha de comando ou uma interface gráfica.

No geral, um sistema operacional é a base do funcionamento de um computador, supervisionando e coordenando todas as atividades do sistema, garantindo que os recursos sejam utilizados eficientemente, os processos sejam gerenciados e os aplicativos possam interagir de forma produtiva com o hardware e os usuários.

2) Descreva quais são as principais tarefas de um Sistema Operacional.

As principais tarefas de um sistema operacional são:

- Gerenciamento de hardware: O sistema operacional é responsável por gerenciar o hardware do computador, incluindo a CPU, a memória, os dispositivos de entrada e saída, e o armazenamento de dados. Ele deve garantir que esses recursos sejam usados de forma eficiente e segura.
- Gerenciamento de processos: O sistema operacional deve gerenciar os processos que estão em execução no computador. Ele deve garantir que cada processo receba uma quantidade justa de recursos e que não haja conflitos entre processos.
- Gerenciamento de memória: O sistema operacional deve gerenciar a memória disponível no computador. Ele deve garantir que a memória seja usada de forma

Disciplina: Sistema Operacional

Profa: Dra. Glenda Botelho

Aluno(s):Mateus Alves Araújo.

eficiente e que não haja vazamentos de memória.

- Gerenciamento de arquivos: O sistema operacional deve gerenciar os arquivos que estão armazenados no computador. Ele deve garantir que os arquivos sejam acessados de forma eficiente e segura.
- Gerenciamento de usuários: O sistema operacional deve gerenciar os usuários que estão acessando o computador. Ele deve garantir que cada usuário tenha acesso aos recursos apropriados.
- Gerenciamento de segurança: O sistema operacional deve proteger o computador de ataques externos. Ele deve garantir que os dados do usuário sejam mantidos seguros.

Além dessas tarefas básicas, os sistemas operacionais modernos também podem oferecer uma variedade de outros recursos, como:

- Interface gráfica do usuário: A interface gráfica do usuário (GUI) permite que os usuários interajam com o computador por meio de menus, ícones e outros elementos visuais.
- Suporte a redes: Os sistemas operacionais modernos podem suportar redes locais e remotas, permitindo que os usuários compartilhem arquivos, impressoras e outros recursos.
- Suporte a multitarefa: Os sistemas operacionais modernos podem suportar a execução de vários programas ao mesmo tempo.
- Suporte a multiusuário: Os sistemas operacionais modernos podem permitir que vários usuários acessem o computador ao mesmo tempo.

As tarefas específicas de um sistema operacional podem variar dependendo de sua arquitetura e do tipo de computador em que ele está sendo executado. No entanto, as tarefas listadas acima são comuns a todos os sistemas operacionais.

Disciplina: Sistema Operacional

Profa: Dra. Glenda Botelho

Aluno(s):Mateus Alves Araújo.

3) Faça um breve resumo das principais características de cada geração dos SOs.

A evolução dos sistemas operacionais pode ser dividida em quatro gerações principais:

1ª Geração (1945-1959)

Os sistemas operacionais da primeira geração eram simples e primitivos, projetados para máquinas de grande porte. Eles não tinham interface gráfica e eram totalmente controlados por comandos digitados pelo usuário.

2ª Geração (1959-1964)

Os sistemas operacionais da segunda geração eram mais complexos e poderosos, graças ao desenvolvimento de novos componentes de hardware, como os transistores. Eles introduziram alguns recursos novos, como o multiprogramação, que permite que o computador execute vários programas ao mesmo tempo.

3ª Geração (1964-1971)

Os sistemas operacionais da terceira geração eram mais modulares e flexíveis, permitindo que fossem adaptados a diferentes tipos de hardware. Eles introduziram novos recursos, como o compartilhamento de arquivos, que permite que vários usuários acessem os mesmos arquivos.

4ª Geração (1971-presente)

Os sistemas operacionais da quarta geração são os mais modernos e avançados. Eles são baseados em microprocessadores e oferecem uma ampla gama de recursos, incluindo interface gráfica, multitarefa e multiusuário.

Disciplina: Sistema Operacional

Profa: Dra. Glenda Botelho

Aluno(s):Mateus Alves Araújo.

Aqui estão alguns exemplos de sistemas operacionais de cada geração:

- 1ª geração: IBM OS/360, Univac 1100/2200
- 2ª geração: IBM CP/CMS, DEC VAX/VMS
- 3ª geração: IBM MVS, UNIX
- 4ª geração: Microsoft Windows, macOS, Linux

É importante notar que essas gerações não são mutuamente exclusivas. Muitos sistemas operacionais modernos incluem recursos de várias gerações.

- 4) Cite pelo menos duas vantagens do uso de sistemas operacionais baseados na estrutura em Camadas.

Duas vantagens do uso de sistemas operacionais baseados na estrutura em camadas são:

- Facilidade de manutenção e depuração: A estrutura em camadas permite que cada camada seja isolada das outras. Isso torna mais fácil para os desenvolvedores trabalharem em uma camada específica sem afetar as outras camadas. Além disso, a estrutura em camadas facilita a depuração de problemas, pois os desenvolvedores podem identificar a camada responsável pelo problema.
- Flexibilidade: A estrutura em camadas permite que o sistema operacional seja modificado ou expandido sem afetar as outras partes do sistema. Isso torna o sistema mais flexível e adaptável às mudanças nas necessidades dos usuários.

- 5) No que diz respeito às vantagens da arquitetura de micro-núcleo (micro-kernel) para sistemas operacionais em relação à arquitetura de núcleo monolítico, quais das seguintes afirmações são verdadeiras?

- I. A arquitetura de micro-núcleo facilita a depuração do S.O.
- II. A arquitetura de micro-núcleo permite um número menor de mudanças de contexto.
- III. A arquitetura de micro-núcleo facilita a reconfiguração de serviços do SO, pois a maioria deles reside em espaço de usuário.

- (a) apenas I;
- (b) II e III;
- (c) I e III; CERTA
- (d) I e II;
- (e) todas são verdadeiras.

Disciplina: Sistema Operacional
Profa: Dra. Glenda Botelho
Aluno(s):Mateus Alves Araújo.

6) Defina e diferencie: programa, processo e *thread*.

Programa é um conjunto de instruções que descrevem uma tarefa a ser executada pelo computador. Um programa pode ser escrito em uma variedade de linguagens de programação, como C, C++, Java, Python, etc.

Processo é um programa em execução. Um processo tem um conjunto de recursos alocados pelo sistema operacional, como memória, tempo de CPU, arquivos e dispositivos. Um processo também tem um conjunto de estados, que descrevem seu status de execução.

Thread é um fluxo de controle dentro de um processo. Um processo pode ter um ou mais threads. Cada thread tem seu próprio conjunto de registradores, pilha e contador de programa. Threads compartilham o mesmo espaço de memória do processo, mas têm seu próprio conjunto de recursos alocados, como tempo de CPU e arquivos.

Característica	Programa	Processo	Thread
Definição	Conjunto de instruções	Programa em execução	Fluxo de controle dentro de um processo
Recursos alocados	Nenhum	Memória, tempo de CPU, arquivos e dispositivos	Registradores, pilha e contador de programa
Estado	Não tem	Em execução, pronto, bloqueado, etc.	Em execução, pronto, bloqueado, etc.
Compartilha recursos com	Nenhum	Processo	Processo

tabela que resume diferenças entre programa, processo e thread:

Disciplina: Sistema Operacional
Profa: Dra. Glenda Botelho
Aluno(s): Mateus Alves Araújo.

- 7) O custo (e consequentemente o tempo) de criação de um thread (seja ela um *thread* de usuário ou um *thread* de *kernel*) é menor que o custo de criação de um processo. Por quê?

O custo de criação de um thread é menor que o custo de criação de um processo porque threads compartilham muitos recursos com o processo pai. Esses recursos incluem:

- Espaço de endereçamento virtual
- Tabelas de páginas
- Arquivos abertos
- Dispositivos de E/S

Ao compartilhar esses recursos, threads são mais eficientes em termos de memória e tempo de inicialização.

Além disso, threads são menos complexos de criar do que processos. Isso ocorre porque threads não precisam de um novo espaço de memória virtual ou tabelas de páginas. Eles simplesmente compartilham o espaço de memória virtual e as tabelas de páginas do processo pai.

Aqui está uma tabela que resume as diferenças entre os custos de criação de threads e processos:

Recurso	Thread	Processo
Espaço de memória virtual	Compartilhado	Não compartilhado
Tabelas de páginas	Compartilhadas	Não compartilhadas
Arquivos abertos	Compartilhados	Não compartilhados
Dispositivos de E/S	Compartilhados	Não compartilhados
Tempo de inicialização	Menor	Maior

Em geral, threads são uma maneira mais eficiente de paralelizar tarefas em um sistema operacional. Eles são mais rápidos de criar e inicializar, e eles compartilham recursos com o processo pai, o que torna mais eficiente o uso de memória e recursos de E/S.

Disciplina: Sistema Operacional

Profa: Dra. Glenda Botelho

Aluno(s):Mateus Alves Araújo.

- 8) Um processo é caracterizado por ser um “programa em execução”. Durante sua existência no sistema, um processo pode assumir basicamente três estados. Cite quais são esses estados e descreva os eventos que fazem com que um processo mude de um estado para outro.

Em um sistema operacional, um processo é como um aluno na escola. Ele nasce, cresce, estuda, faz provas, tira notas e, eventualmente, se forma.

Um processo nasce quando é criado por um usuário ou por outro processo. Ele começa sua vida no estado pronto, esperando para ser chamado para a sala de aula, ou seja, para ser executado pelo processador.

Quando um processo é escalonado para o estado executando, ele entra na sala de aula e começa a estudar, ou seja, a executar suas instruções.

Um processo pode ser bloqueado quando ele precisa esperar por algo, como a conclusão de uma operação de E/S. Nesse caso, ele vai para o estado bloqueado, esperando pela sua vez de voltar para a sala de aula.

Quando o processo que o processo bloqueado estava esperando terminar, o processo bloqueado é desbloqueado e volta para o estado pronto.

Um processo pode ser encerrado quando ele termina sua execução, quando ele é interrompido por um sinal ou quando ele é morto pelo sistema operacional.

Um exemplo lúdico

Imagine um processo que está executando um programa de edição de texto. Quando o usuário começa a digitar, o processo entra no estado executando. Quando o usuário pressiona a tecla "Enter", o processo é bloqueado até que a operação de E/S de escrita seja concluída. Quando a operação de E/S é concluída, o processo é desbloqueado e volta para o estado pronto.

Conclusão

Os estados de um processo são importantes para o gerenciamento de processos, que é a função do sistema operacional que controla a execução de processos. O gerenciamento de processos é responsável por garantir que os processos sejam executados de forma eficiente e justa, e que os recursos do sistema sejam alocados de forma eficaz.

- 9) Qual a diferença entre processos *CPU-bound* e *I/O-bound*? Quais são os problemas que podem ocorrer caso o sistema tenha muitos processos *CPU-bound* ou muitos processos *I/O-bound*?

A diferença entre processos *CPU-bound* e *I/O-bound* é que os processos *CPU-bound* são aqueles que passam a maior parte do tempo executando instruções no processador, enquanto os processos *I/O-bound* são aqueles que passam a maior parte do tempo esperando por operações de entrada e saída.

Processos *CPU-bound* são geralmente tarefas que envolvem cálculos complexos, como renderização de gráficos, processamento de dados ou aprendizado de máquina. Processos *I/O-bound* são geralmente tarefas que envolvem acesso a arquivos, como cópia de arquivos,

Disciplina: Sistema Operacional

Profa: Dra. Glenda Botelho

Aluno(s):Mateus Alves Araújo.

leitura de dados da web ou carregamento de vídeos.

Caso um sistema tenha muitos processos CPU-bound, isso pode levar a um aumento na carga do processador, o que pode afetar o desempenho de outros processos e do sistema como um todo.

Em casos extremos, isso pode até levar a um travamento do sistema.

Caso um sistema tenha muitos processos I/O-bound, isso pode levar a um aumento na fila de operações de E/S, o que pode afetar o desempenho de outros processos e do sistema como um todo. Em casos extremos, isso pode até levar a uma perda de dados.

Para evitar esses problemas, é importante que o sistema operacional seja projetado para lidar com diferentes tipos de processos de forma eficiente. Isso pode ser feito usando algoritmos de escalonamento que priorizam os processos I/O-bound ou usando técnicas de paralelização para executar processos CPU-bound em múltiplos processadores.

Aqui estão alguns exemplos específicos de problemas que podem ocorrer caso o sistema tenha muitos processos CPU-bound ou muitos processos I/O-bound:

- Processos CPU-bound
 - Redução do desempenho
 - Travamento do sistema
- Processos I/O-bound
 - Aumento da fila de operações de E/S
 - Perda de dados

É importante notar que esses problemas podem variar de acordo com o sistema operacional e com as configurações específicas do hardware.

fonte:

www.passeidireto.com/arquivo/80037654/lista-de-sistemas-operacionais-ssc-5723-2020/2

- 10)** A figura abaixo apresenta um diagrama de estado básico mostrando três estados de um processo em computador gerenciado por um sistema operacional. Considerando que o computador, onde o sistema operacional se encontra, tem um único processador de um núcleo, assinale a alternativa que melhor define um desses estados

Disciplina: Sistema Operacional

Profa: Dra. Glenda Botelho

Aluno(s):Mateus Alves Araújo.



- a)** Em execução: neste estado o processo é executado, podendo ter mais de um estado.
- b)** Bloqueado: o processo ao chegar neste estado, além de ser bloqueado ele é descartado, não podendo mais voltar ao estado em execução.
- c)** Pronto: neste estado, o processo foi executado e finalizado.
- d)** Em execução / pronto: ambos estados o processo é executado, entretanto no último o processo é executado e finalizado.
- e)** Pronto: o processo está temporariamente parado enquanto aguarda ser executado. :CERTO

Obs. Pode ser feita em Dupla ou Individual