Nome: Matheus de Amorim Favero – Ciência da Computação

Sistemas Operacionais – Quiz 05/09/2023

Questões Teóricas

1. Definições básicas (o que é um processo, thread, etc.).

* **Processo**

Um processo é uma instância de um programa em execução em um sistema operacional. Ele inclui o código executável, dados, variáveis, registradores e um contexto de execução. Um processo é uma entidade independente que pode ser gerenciada pelo sistema operacional, e cada processo possui seu próprio espaço de endereçamento e recursos, o que o torna isolado de outros processos em execução.

* **Thread**

Uma thread, também conhecida como "thread de execução" ou "thread de processo", é a menor unidade de execução dentro de um processo. Diferentes threads dentro do mesmo processo compartilham o mesmo espaço de endereçamento e recursos, mas cada uma executa seu próprio fluxo de controle independente. As threads são usadas para tornar a execução de programas mais eficiente e permitir a execução concorrente de tarefas em sistemas multitarefa.

* **Kernel**

O kernel é a parte central do sistema operacional que fornece serviços essenciais de gerenciamento de hardware e recursos. Ele age como uma camada de abstração entre o hardware e os programas de usuário, controlando o acesso aos recursos do sistema e garantindo a execução segura e eficiente dos processos e threads.

* **Escalonamento**

O kernel é a parte central do sistema operacional que fornece serviços essenciais de gerenciamento de hardware e recursos. Ele age como uma camada de abstração entre o hardware e os programas de usuário, controlando o acesso aos recursos do sistema e garantindo a execução segura e eficiente dos processos e threads.

1. Diferenças entre tipos de processos (por exemplo, processos de usuário x processos de sistema).

Os sistemas operacionais frequentemente classificam os processos em diferentes categorias com base em seu contexto e privilégios. Duas categorias comuns são os "processos de usuário" e os "processos de sistema".

* **Processos de Usuário:** Origem: Os processos de usuário são iniciados e controlados pelos usuários do sistema. Eles são geralmente programas ou aplicativos que um usuário executa, como navegadores da web, processadores de texto, jogos, etc.
* **Privacidade:** Os processos de usuário têm privilégios limitados e não têm acesso direto ao hardware ou ao núcleo do sistema operacional. Eles operam em um ambiente virtual que é gerenciado e protegido pelo sistema operacional.
* **Tempo de CPU:** Geralmente, os processos de usuário têm uma fatia limitada do tempo de CPU e são escalonados pelo sistema operacional para evitar que monopolizem os recursos do sistema.
* **Interatividade:** Muitos processos de usuário são interativos e interagem diretamente com o usuário por meio de interfaces gráficas ou de linha de comando.
* **Processos de Sistema:** Origem: Os processos de sistema são iniciados e gerenciados pelo próprio sistema operacional para realizar funções internas e tarefas de gerenciamento do sistema. Eles não são iniciados pelos usuários.
* **Privilégios:** Os processos de sistema geralmente têm privilégios mais elevados e acesso direto a recursos críticos do sistema, como o kernel do sistema operacional e o hardware.
* **Funções:** Os processos de sistema executam funções essenciais para o funcionamento do sistema, como gerenciamento de memória, gerenciamento de dispositivos, escalonamento de tarefas, entre outros.
* **Estabilidade**: A estabilidade dos processos de sistema é crucial para o funcionamento do sistema operacional. Falhas nesses processos podem levar a problemas graves de sistema, como travamentos ou panes.
* **Não interativos:** Os processos de sistema normalmente não têm interação direta com os usuários. Eles operam em segundo plano e não fornecem uma interface de usuário visível.

1. Conceitos de escalonamento de processos

O escalonamento de processos é um componente crucial dos sistemas operacionais multitarefa, e seu principal objetivo é alocar a CPU de forma eficiente e justa entre os diferentes processos em execução.

* Escalonador: Componente do sistema que escolhe qual processo será executado na CPU.
* Política de Escalonamento: Regras para selecionar processos na fila de prontidão, como FIFO, Round Robin e Prioridade.
* Fila de Prontidão: Mantém processos prontos para execução.
* Tempo de Turnaround: Tempo entre a submissão de um processo e sua conclusão.
* Tempo de Espera: Tempo que um processo passa aguardando na fila de prontidão.
* Tempo de Execução: Tempo que um processo passa executando na CPU
* Inversão de Prioridade: Processo de alta prioridade aguarda recurso de baixa prioridade.
* Afiliação de Processo: Associação de processo a um núcleo de CPU.
* Preempção: Capacidade de interromper a execução de um processo em andamento.

1. Prioridades e estados de um processo

Prioridades de um Processo:

* Prioridade do Processo: Valor atribuído a um processo para determinar sua ordem de execução. Processos de alta prioridade são executados antes dos de baixa prioridade.

Estados de um Processo:

* Pronto (Ready): O processo está pronto para ser executado, mas está esperando na fila de prontidão pelo tempo da CPU.
* Execução (Running): O processo está sendo executado atualmente na CPU.
* Bloqueado (Blocked): O processo está temporariamente impedido de executar, geralmente aguardando por algum evento, como entrada/saída (E/S).
* Terminado (Terminated): O processo concluiu sua execução e foi encerrado.

1. Mecanismos de comunicação entre processos

* Pipes: Canal unidirecional para transferência de dados entre processos.
* Fifos (Named Pipes): Pipes nomeados que permitem a comunicação entre processos não relacionados.
* Sockets: Conexões de rede para comunicação entre processos em diferentes máquinas.
* Memória Compartilhada: Área de memória compartilhada para troca de dados entre processos.
* Mensagens: Comunicação baseada em mensagens por meio de filas ou buffers.

1. Conceitos de segurança relacionados a processos

* Isolamento: Garantir que processos não interfiram uns com os outros e não acessem recursos não autorizados.
* Controle de Acesso: Gerenciar permissões para garantir que apenas processos autorizados acessem recursos protegidos.
* Privilégios Mínimos: Limitar os privilégios dos processos para reduzir riscos de segurança.
* Separação de Espaço de Endereçamento: Isolar o espaço de memória de processos para evitar vazamento de dados.
* Monitoramento e Auditoria: Acompanhar atividades de processos para detecção de comportamento malicioso.
* Autenticação e Autorização: Verificar identidade e conceder acesso com base em regras predefinidas.
* Criptografia: Proteger a comunicação e dados em processos por meio de criptografia.
* Gestão de Identidade: Gerenciar identidades de processos para garantir autenticidade e integridade.

1. O que são deadlocks e como podem ser evitados

Deadlock é uma situação em que dois ou mais processos (ou threads) ficam bloqueados permanentemente, incapazes de continuar a execução, porque cada um está aguardando um recurso que o outro possui. É um problema comum em sistemas multitarefa e requer estratégias de prevenção e recuperação para evitar impasses.

1. Conceitos de memória e como ela é gerenciada em relação aos processos

* Espaço de Endereçamento: Cada processo tem seu próprio espaço de endereçamento virtual, isolado de outros processos.
* Segmentação: Divisão do espaço de endereçamento em segmentos para dados, código e pilha.
* Paginação: Divisão do espaço de endereçamento em páginas de tamanho fixo para gerenciar a memória física.
* Memória Virtual: Uso de espaço de endereçamento virtual maior que a memória física disponível, com uso de paginação

1. O que são processos zumbis e órfãos

Processos Zumbis: Processos filhos que terminaram, mas ainda têm entrada na tabela de processos, esperando que o pai colete informações de término.

Processos Órfãos: Processos filhos que continuam a ser executados após o pai ter terminado. Eles são adotados pelo processo init (PID 1).

1. Conceitos de processos em sistemas distribuídos

* Processo Distribuído: Entidade executando programas em um sistema distribuído.
* Comunicação entre Processos: Mecanismos para compartilhar informações entre processos distribuídos.
* Coordenação: Gerenciamento da ordem e concorrência entre processos para manter a consistência dos dados.
* Tolerância a Falhas: Lidar com falhas de hardware ou software para manter a disponibilidade e confiabilidade.
* Transações Distribuídas: Garantir a atomicidade e consistência das operações em múltiplos processos distribuídos.
* Escalabilidade: Adicionar ou remover processos conforme necessário para lidar com aumento de carga.
* Sincronização: Coordenação de tempo e eventos entre processos para garantir a consistência e a ordem das operações.