

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PB
CAMPUS DE ESPERANÇA

CURSO: Análise e Desenvolvimento de Sistemas DATA: 17/10/2024

DISCIPLINA: Sistemas Operacionais (SO)

PROFESSOR (A): Arthur

ALUNO: Maria Eduarda Coelho da Silva

1. Quais são as duas maneiras de análise dos SO's? Explique

Máquina Estendida:

Nesta perspectiva, o sistema operacional (SO) é examinado como uma camada que facilita a interação entre o usuário e o hardware. A ênfase está em como o SO simplifica as complexidades subjacentes e oferece uma interface amigável, permitindo que o usuário execute tarefas de baixo nível com facilidade.

Gerenciador de Recursos:

Neste contexto, a atenção se concentra na função do SO em gerenciar e otimizar o uso dos recursos de hardware. A análise investiga como o SO controla o acesso e a alocação de recursos como CPU, memória e dispositivos, assegurando tanto eficiência quanto segurança.

2. Conceitue:

a) Multi-Programação

Consiste em segmentar a memória do sistema em várias partes, atribuindo a cada uma delas um job específico. O objetivo é manter uma quantidade adequada de jobs na memória para ocupar 100% do tempo do processador, reduzindo a ociosidade e melhorando a eficiência. É fundamental ressaltar que, nesse modelo, é o hardware que protege cada job contra acessos indevidos por parte de outros jobs, assegurando a segurança e a integridade dos dados.

b) Spooling

Spooling (Operação Simultânea de Periféricos em Linha):

É uma técnica que possibilita a leitura de cartões de jobs diretamente do disco, eliminando a necessidade de entrada manual. Assim que um job é concluído, o sistema operacional imediatamente aloca um novo job em uma partição livre na memória, proveniente do disco. Isso otimiza o uso dos recursos e assegura uma transição ágil entre as tarefas.

c) TimeSharing

Em um sistema de time-sharing, cada usuário tem acesso a um terminal online, permitindo o uso simultâneo do computador. O primeiro sistema de time-sharing foi o CTSS (Compatible Time Sharing System), que

funcionava em um computador 7094 modificado. Essa abordagem dá a cada usuário a impressão de que possui o computador exclusivamente para si. Por exemplo, se 20 usuários estão ativos, mas 17 estão ausentes, o processador é distribuído entre os 3 jobs em execução, maximizando a utilização do sistema e proporcionando uma experiência interativa e responsiva.

3. Quais os tipos de Sistemas Operacionais? Caracterize cada um deles.

Permitem que múltiplos usuários acessem o computador simultaneamente por meio de terminais online, criando a impressão de que cada um possui o computador para si. Um exemplo é o CTSS.

Sistemas Operacionais de Lote (Batch OS):

Processam jobs em lotes de forma sequencial, sem interação em tempo real, utilizando a técnica de spooling para otimizar a entrada e saída de dados (I/O).

Sistemas Operacionais em Tempo Real (Real-Time OS):

Asseguram respostas dentro de prazos rigorosos, sendo essenciais em aplicações críticas. Podem ser classificados como duros (sem tolerância a atrasos) ou suaves (com alguma tolerância).

Sistemas Operacionais de Rede (Network OS):

Facilitam a comunicação e o compartilhamento de recursos entre vários computadores em uma rede, garantindo a segurança das informações.

Sistemas Operacionais Móveis (Mobile OS):

Desenvolvidos para dispositivos móveis, focando na eficiência energética e na usabilidade. Exemplos incluem Android e iOS.

Sistemas Operacionais de Usuário Único (Single-User OS):

Destinados a um único usuário, são comumente encontrados em desktops e laptops, priorizando a facilidade de uso.

4. Qual a importância das System calls?

As system calls (chamadas de sistema) são fundamentais para a operação dos sistemas operacionais, pois fornecem uma interface que permite que aplicativos de usuário acessem recursos de hardware de forma controlada e segura. Elas abstraem a complexidade do hardware, facilitando o gerenciamento eficiente de recursos como memória e processos, além de implementar mecanismos de segurança que restringem o acesso não autorizado. As system calls também permitem a comunicação e sincronização entre processos, contribuindo para o desenvolvimento de aplicativos robustos e portáteis. Em essência, elas funcionam como a ponte que conecta a lógica de alto nível dos aplicativos ao funcionamento de baixo nível do sistema, garantindo eficiência e segurança.

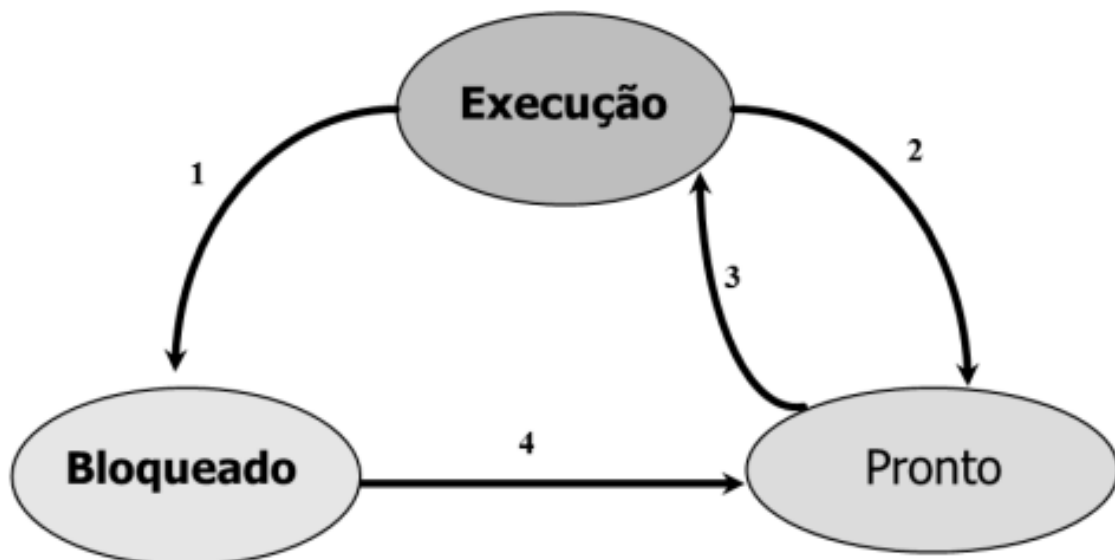
5. Qual a importância da instrução trap?

As chamadas de sistema (system calls) são essenciais para o funcionamento dos sistemas operacionais, pois oferecem uma interface que possibilita que aplicativos de usuário acessem recursos de hardware de maneira controlada e segura. Elas simplificam a complexidade do hardware, facilitando o gerenciamento eficiente de recursos como memória e processos, além de implementar mecanismos de segurança que previnem o acesso não autorizado. As chamadas de sistema também permitem a comunicação e sincronização entre processos, contribuindo para o desenvolvimento de aplicativos robustos e portáteis. Em suma, elas atuam como uma ponte entre a lógica de alto nível dos aplicativos e o funcionamento de baixo nível do sistema, assegurando eficiência e segurança.

6. Defina PCB. Quais os principais dados da PCB?

O PCB (Process Control Block) é uma estrutura de dados crucial utilizada pelo sistema operacional para armazenar informações sobre um processo em execução, permitindo uma gestão eficiente dos processos. Os principais dados contidos em um PCB incluem o PID (Process ID), que identifica o processo, o estado do processo (por exemplo, pronto, em execução ou bloqueado), informações de controle como prioridade e tempo de CPU utilizado, o contexto do processo com contadores de programa e registradores, além de informações de memória como endereços alocados e tabelas de páginas. Também são incluídos dados de entrada/saída relacionados e ponteiros para outros PCBs, facilitando a gestão das listas de processos. Essa estrutura é essencial para a troca de contexto e o controle eficaz da execução dos processos.

7. Explique os estados dos processos do SO. Comente cada mudança de estado.



O diagrama representa os estados dos processos em um sistema operacional: Execução, Bloqueado e Pronto. No estado de Execução, o processo está ativo e sendo executado pela CPU. A transição de Bloqueado para Execução ocorre quando um processo que estava aguardando um recurso (como I/O) tem esse

recurso liberado. No estado Bloqueado, o processo permanece em espera até que um evento específico aconteça. A mudança de Pronto para Execução ocorre quando a CPU fica disponível para o processo que está preparado para ser executado. Um processo em Execução pode mudar para o estado Bloqueado ao solicitar um recurso que não está disponível ou retornar ao estado Pronto se for interrompido para permitir que outro processo seja executado. Esses estados e suas transições são fundamentais para o gerenciamento eficaz dos processos.