

O que é Kernel? componente principal de um sistema operacional , serve como ponte entre o hardware e o software

O que é Processos cpu bound? é quando o tempo de processamento depende mais do processador do que das entradas e saídas, fazendo assim com que atrapalhe o tempo total de processamento.

Processos e conceitos Estado do Processo

Execução (running) – O processo está sendo executado pela CPU. Pronto (ready) – O processo está pronto e esperando para ser executado pela CPU. Espera (wait) – O processo está esperando algum evento externo ou por algum recurso para poder prosseguir seu processamento.

O que é paralelismo? Paralelismo é sobre a execução paralela de tarefas, ou seja, mais de uma por vez (de forma simultânea), a depender da quantidade de núcleos (cores) do processador.

O que é política de escalonamento de um sistema operacional? Trata-se da abordagem utilizada pelo sistema operacional para gerenciar o processador e tornar possível a multiprogramação do sistema operacional.

tipos de servidores de rede? Servidor de aplicação ou application server. ... Servidor de Arquivos. ... Servidor de banco de dados. ... Servidor de mídia. ... Servidor de email. ... Servidor de backup. ... Servidor FTP. ... Servidor Proxy.

Qual a diferença entre sistema operacional e kernel? O sistema operacional fornece interface entre o usuário e o hardware. Kernel fornece interface entre aplicativos e hardware. Ele também oferece proteção e segurança. Seu objetivo principal é o gerenciamento de memória, gerenciamento de disco, gerenciamento de processos e gerenciamento de tarefas.

Tipos de kernel? **Kernel monolítico:** os serviços do usuário e os serviços do kernel são implementados no mesmo espaço da memória. ... **Microkernel:** separa os serviços do usuário e os serviços do kernel em espaços diferentes. ... Kernel híbrido: usa a velocidade do monolítico e a modularidade do micro.

Modos de operações?

Existem dois modos de operação: Thread e Handler. O modo Thread é definido quando a operação do processador é iniciada (modo de Reset). O processador entra em modo Handler quando ocorre uma exceção (interrupção). Quando o contexto é restabelecido, o processador retorna para o estado Thread.

O que é um sistema de entrada e saída? Resultado de imagem para o que é entrada e saída Os dispositivos de entrada são os que inserem informações em um computador a partir de uma fonte externa. Exemplos incluem: teclado, mouse, microfone, scanner e telas sensíveis ao toque. Já os dispositivos de saída são os aparelhos usados por um computador para comunicar informações.

O que é threads de usuário? Resultado de imagem para threads de usuários e de núcleo As threads podem ser divididas em threads de usuário, as quais são executadas dentro de um processo e correspondem às tarefas executadas, e threads de núcleo, que correspondem à fluxos de execução reconhecidos e gerenciados pelo núcleo do sistema, ou também chamadas de kernel threads.

Threads e núcleos Quando se fala que um determinado processador é dual-core, quad-core, hexa-core, e assim por diante, significa que a CPU tem diversos núcleos independentes que funcionam como processadores individuais. Cada um desses corresponde a uma thread, ou linha de execução.

Principais diferenças entre semáforo e mutex Semaphore é um mecanismo de sinalização, pois a operação wait () e signal () executada na variável semáforo indica se um processo está adquirindo o recurso ou liberando o recurso. Por outro lado, o mutex é um mecanismo de bloqueio, para adquirir um recurso, um processo precisa bloquear o objeto mutex e, ao liberar um processo de recurso, deve desbloquear o objeto mutex. Semaphore é tipicamente uma variável inteira enquanto que mutex é um objeto . Semaphore permite que vários encadeamentos de programa acessem a instância finita de recursos . Por outro lado, o Mutex permite que vários encadeamentos de programa acessem um único recurso compartilhado, mas um de cada vez. O valor da variável de semáforo pode ser modificado por qualquer processo que adquira ou libere recursos executando operações wait () e signal (). Por outro lado, o bloqueio adquirido no objeto mutex pode ser liberado apenas pelo processo que adquiriu o bloqueio no objeto mutex. Semaphore são de dois tipos, semáforo binário e semáforo binário que é bastante semelhante ao mutex. O valor da variável do semáforo é modificado pela operação wait () e signal () além da inicialização. No entanto, o objeto mudo é bloqueado ou desbloqueado pelo processo de aquisição ou liberação do recurso. Se todos os recursos forem adquiridos pelo processo e nenhum recurso estiver livre, o processo que deseja adquirir o recurso executará a operação wait () na variável semáforo e bloqueará ele mesmo até que a contagem do semáforo se torne maior que 0. Mas se um objeto mutex já estiver bloqueado então o processo que deseja adquirir esperas de recursos e ser enfileirado pelo sistema até que o recurso seja liberado e o objeto mutex seja desbloqueado.

Deadlock, no contexto de sistemas operacionais, refere-se a uma situação em que ocorre um impasse, e dois ou mais processos ficam impedidos de continuar suas execuções - ou seja, ficam bloqueados, esperando uns pelos outros

Diferença de Threads de Processos? Resumindo: Um processo é um programa em execução, somado ao seu contexto. Threads: É um processo com múltiplos fluxos de controle.

Sistema? É um conjunto integrado de componentes regularmente inter-relacionados e interdependentes criados para realizar um objetivo definido, com relações definidas e mantidas entre seus componentes e cuja produção e operação como um todo é melhor que a simples soma de seus componentes.

Classificação dos Sistemas Operacionais

Monolítico - possui um conjunto de instruções de alto nível que possibilitam o gerenciamento de processos, são executados com privilégios especiais.

híbrido - São servidores externos, são executados em um modo chamado “protegido”, melhorando o desempenho geral, pode agregar ou desagregar funcionalidades sem perder a performance

Microkernel - Os processos se comunicam com um núcleo mínimo, usando o mínimo possível o "espaço do sistema" (*kernel space*). Neste local os aplicativos tem acesso a todas as instruções e a todo o hardware e deixando o máximo de recursos rodando no "espaço do usuário" (*user space*) em que o software tem algumas restrições, não podendo acessar algumas hardwares, nem tem acesso a todas as instruções).

kernel - é o componente central do sistema operativo da maioria dos computadores; ele serve de ponte entre aplicativos e o processamento real de dados feito a nível de hardware. As

responsabilidades do núcleo incluem gerenciar os recursos do sistema (a comunicação entre componentes de hardware e software).

monotarefa - executam uma tarefa por vez

Multitarefa - executam mais de uma tarefa por vez, ex: criar desenho, ouvir música, etc

Sistema de tempo compartilhados - Se houver alguma necessidade de executar mais de um programa, o S.O irá designar uma fatia de tempo para cada um, e se o tempo não for suficiente para a conclusão, ele irá ser interrompido.

Monousuario - O S.O foi criado para que um único usuário utilize os recursos do computador.

Multiusuário - O S.O permite que vários usuários utilize o computador.

Oque é um processo ? - Criação → execução → término

CPU Bound - São processos que ficam executando instruções na CPU

I/O Bound - Ficam o tempo todo fazendo solicitações de entrada e saída, na maior parte ele vai estar no estado de espera aguardando essas solicitações que ele faz.

Oque é Paralelismo ? - Ocorre quando as tarefas de um computador são executados em simultâneo em uma máquina com mais de um processador.

Buffering - Utilização da memória central para a transferencia de dados entre dispositivos de entrada e saída

Multiprogramação - É quando o S.O opera em vários programas em paralelo.

DeadLock : Ocorre quando dois ou mais processos estão esperando indefinidamente por um evento que só pode ocorrer por um dos processos de prioridade maior.

Thread - É uma forma como um processo / tarefa de um programa de computador é dividido em duas ou mais tarefas.