ESTRUTURA DE ARMAZENAMENTO EM MASSA

Grande parte dos dispositivos de armazenamento secundário é composta por:

- Discos magnéticos A velocidade do disco tem duas partes:
    i. A taxa de transferência: taxa de fluxo de dados entre o drive e o computador
    ii. O tempo de posicionamento(random-access time): consiste emduas partes: (1) O tempo necessário para acessar o cilindro do setor desejado(seek time); (2) O tempo necessário para o setor desejado rotacionar até a cabeça do disco(rotational latency).
- Discos de estado sólido Podem ser mais confiáveis e rápidos que HDs, pois não tempo partes móveis nem seek time e rotational latency. Além de que consomem menos energia. No entanto, possuem um preço mais alto por MB.
- Fitas magnéticas São relativamente permanentes e podem armazenar grandes quantidades de dados, no entanto o tempo de acesso é menor que o do HD(em cerca de 1000x). Portanto, não são muito úteis como armazenamento secundário.

Os drives de disco modernos são estruturados como grandes arrays unidimensionais de blocos de disco lógicos, que são mapeados pelo SO começando no setor 0 da primeira trilha no cilindro mais externo, depois o procedimento se repete através das trilhas do cilindro, depois pelo resto dos cilindros, do mais externo para o mais interno. Os discos podem ser anexados a um sistema de computação de duas maneiras:

1. através das portas de I/O locais no computador hospedeiro ou
2. por meio de uma conexão de rede.

As solicitações de I/O de disco são geradas pelo SO, processos de sistema ou processos de usuário. Cada solicitação especifica o endereço a ser referênciado no disco, número de setores a senrem transfiridos, endereço de memória. Os algoritmos de scheduling de disco podem melhorar a largura de banda efetiva, o tempo de resposta médio e a variância no tempo de resposta(minimizar o seek time). Algoritmos:

- SSTF -> É comum e tem uma abordagem natural. Escolha boa para algoritmo padrão
- SCAN -> Produz menos starvation
- C-SCAN
- LOOK -> Escolha boa para algoritmo padrão
- C-LOOK

Esses algoritmos foram projetados para implementar essas melhorias por meio de estratégias de ordenação das filas de disco. O desempenho dos algoritmos de scheduling de disco pode variar muito em discos magnéticos. Por outro lado, já que os discos de estado sólido não têm partes móveis, o desempenho varia pouco entre os algoritmos, e com muita frequência uma simples estratégia FCFS é utilizada.

O desempenho pode ser prejudicado pela fragmentação externa. Alguns sistemas têm utilitários que varrem o sistema de arquivos para identificar arquivos fragmentados; em seguida, eles mudam blocos de lugar para diminuir a fragmentação. A desfragmentação de um sistema de arquivos muito fragmentado pode melhorar significativamente o desempenho, mas o sistema pode ter o desempenho reduzido enquanto a desfragmentação está ocorrendo. Sistemas de arquivos sofisticados incorporam muitas estratégias para o controle da fragmentação durante a alocação de espaço de modo que a reorganização do disco não seja necessária.

O sistema operacional gerencia os blocos do disco.

1. O disco deve ser formatado em baixo nível para a criação dos setores no hardware bruto
2. Em seguida, o disco é particionado, sistemas de arquivos são criados (agrupa os blocos em clusters - I/O do disco é feito em blocos e I/O de arquivos é feito em clusters) e blocos de inicialização são alocados para armazenar o programa bootstrap do sistema
3. Finalmente, quando um bloco é corrompido, o sistema deve ter uma forma de submetê-lo a um lock ou de substituí-lo logicamente por um reserva. Já que um espaço de permuta eficiente é essencial a um bom desempenho, os sistemas usualmente ignoram o sistema de arquivos e usam o acesso ao disco bruto na paginação de I/O. Alguns sistemas dedicam uma partição de disco bruta para o espaço de permuta, e outros usam um arquivo dentro do sistema de arquivos. Outros sistemas permitem que o usuário ou o administrador do sistema tomem a decisão fornecendo as duas opções.

Algoritmos RAID(Importante) permitem que mais de um disco seja usado em determinada operação e permitem a operação continuada e até mesmo a recuperação automática em caso de uma falha no disco. Isso aumenta a confiabilidade e taxa de transferência de dados. Os algoritmos RAID são organizados em diferentes níveis; cada nível fornece alguma combinação de confiabilidade e altas taxas de transferência.

- Nível 0: capacidade de armazenamento lógico equivalente a capacidade de armazenamento somada de todos os arrays de disco no RAID. Mas sem nenhuma redundância (como o espelhamento ou bits de paridade)
- Nível 1: refere-se ao espelhamento de discos (sem dar muita atenção a quantidade de armazenamento)
- Nível 0+1: refere-se a uma combinação dos níveis 0 e 1. O nível 0 fornece o desempenho, enquanto o nível 1 fornece a comfiabilidade(também é caro), nele um conjunto de discos é distribuido, e a distribuição é espelhada em outra distribuição equivalente
- Nível 1+0: os discos são espelhados em pares e, então, os pares espelhados resultanes são distribuídos.
- Nível 2,3,4,etc: aumentar a redundancia com menos recursos necessários.

Diferenças entre 0+1 e 1+0: Se um disco falha no RAID 0+1, uma distribuição inteira fica inacessível, deixando apenas a outra distribuição disponível Se um disco no RAID 1+0, apenas um disco fica indisponível, mas o disco que o espelha continua disponível, assim como os outros discos.

INTERFACE DO SISTEMA DE ARQUIVOS

Um arquivo é uma sequência de registros lógicos. Um registro lógico pode representar programas e dados.

- Arquivos de dados: numéricos, alfabéticos, alfanuméricos ou binários

De modo geral, é uma sequência de bits, bytes, linhas ou registros, cujo significado é definido pelo usuário do arquivo. Arquivos podem ter estrutuas diferentes e os tipos de arquivos podem ser usados para indicar essa estrutura. Com isso o SO consegue ler o arquivo da forma desejada, no entanto isso pode tornar o SO muito pesado, pois precisa conter código para dar suporte ao diferentes tipo de estrutura de arquivo. Um arquivo pode ser considerado uma sequencia de blocos. Todas as funções básicas de I/O operam em termos de bloco, pois é dificil para o SO localizar um deslocamento dentro de um arquivo.

Arquivos podem ter atributos(mantidos na estrutura de diretório) como:

- Nome
- Identificador
- Tipo
- Localização
- Tamanho
- Proteção
- Hora, data, etc

Para abrir um aquivo(open file) é necessário:

- Tabela de open-file: mapeia arquivos abertos
- Apontador de arquivos
- Um contador de arquivos abertos
- Localização no disco
- Direitos de acesso

Um arquivo pode ter locks similiares a locks de escrita-leitura.

- Trancamento compartilhado(shared lock): diversos processos podem adquirir concorrentemente
- Trancamento exclusivo(exclusive lock): apenas um processo por vez pode adquirir esse lock
- Trancamento obrigatório: o sistema operacional assegura a integridade do trancamento
- Trancamento aconselhavel:é responsabilidade dos desenvolvedores do software assegurar que os locks sejam apropriadamente adquiridos e liberados. Caso contrário, eles impedirão que outros processos também o acessem

A principal tarefa do SO é mapear o conceito de arquivo lógico para dispositivos de armazenamento físicos como as fitas ou discos magnéticos. Já que o tamanho do registro físico do dispositivo pode não ser igual ao tamanho do registro lógico, é necessário ordenar os registros lógicos nos registros físicos. Novamente, esse tarefa pode ser suportada pelo SO, ou pode ser deixada para o programa de aplicação.

Métodos de acesso às informações de arquivos:

- Acesso sequencial: informações são processadas em ordem, um registro após o outro.
- Acesso direto: os arquivos são compostos por registros lógicos de tamanho fixo que permitem que os programas leiam e gravem registros rapidamente sem uma ordem específica.(função hash)

Cada dispositivo em um sistema de arquivos mantém um índice de volumes ou um diretório de dispositivos que lista a locação dos arquivos no dispositivo. Para permitir a organização dos arquivos é útil criar diretórios. Níveis de diretório:

- Nível 1: causa problemas de nomeação, já que cada arquivo deve ter um nome exclusivo
- Nível 2: soluciona o problema de nomeação criando um diretório separado para os arquivos de cada usuário. O diretório lista os arquivos por nome e inclui a localização do arquivo no disco, seu tamanho, tipo, proprietário, a hora de criação, a hora em que foi usado pela última vez, e assim por diante.
- Estruturado em árvore: permite que um usuário crie subdiretórios para organizar arquivos. As estruturas de diretório em grafo acíclico permitem que os usuários compartilhem subdiretórios e arquivos, mas complicam a busca e a exclusão.
- Estrutura de grafo geral: fornece flexibilidade ilimitada ao compartilhamento de arquivos e diretórios, mas às vezes requer que a coleta de lixo recupere espaço não utilizado em disco.

Os discos são segmentados em um ou mais volumes, cada um contendo um sistema de arquivos ou deixado “bruto”. Os sistemas de arquivos podem ser montados nas estruturas de nomeação do sistema para torná-los disponíveis. O esquema de nomeação varia por sistema operacional. Uma vez montados, os arquivos do volume ficam disponíveis para uso. Os sistemas de arquivos podem ser desmontados para desabilitar o acesso ou para manutenção.

Os sistemas de arquivos distribuídos permitem que hospedeiros clientes montem volumes ou diretórios a partir de servidores, contanto que possam acessar um ao outro por uma rede. Os sistemas de arquivos remotos apresentam desafios quanto à confiabilidade, ao desempenho e à segurança. Os sistemas de informação distribuídos mantêm informações de usuário, hospedeiro e acesso, de modo que clientes e servidores possam compartilhar informações de estado para gerenciar o uso e o acesso.

Como os arquivos são o principal mecanismo de armazenamento de informações na maioria dos sistemas de computação, faz-se necessária a proteção de arquivos. O acesso a arquivos pode ser controlado separadamente para cada tipo de acesso — leitura, gravação, execução, acréscimo, exclusão, listagem de diretório, e assim por diante. A proteção de arquivos pode ser fornecida por listas de controle de acesso, especificando nomes de usuários e os tipos de acesso permitidos a cada usuário. Para condensar a lista de acesso, o SO reconhece três classificações:

- Modos de acesso: leitura, gravação, execução.
- Classes de acesso: Proprietário(usuário que criou o arquivo), Grupo(conjunto de usuários), Universo(todos os usuários no sistema).

SISTEMAS DE I/O

Os elementos básicos de hardware envolvidos no I/O são:

- Portas: pontos de conexação como o dispositivo
- Barramentos
- Controladores de dispositivos: parte eletronica que opera portas, barramentos e dispositivos
- Próprios dispositivos.

Os dipositivos possuem, também, seus próprios registradores onde armazenam:

- comandos
- endereços
- dados para escrita e leitura

Além disso cada dispositivo possui endereço que é usado para instruções de acesso direto e de memória mapeada

Tipos de entradas e saídas:

Sondagem (Polling) (caiu na prova)

É a interação entre controlador e hospedeiro que é feita atraveś do bit *busy*, quando o bit é 1 o controlador está ocupado, quando é 0 o controlador está pronto para aceitar o próximo comando.

1. Hospedeiro lê repetidamente o bit *busy* até que ele seja desligado
2. Hospedeiro liga o bit *write* no registrador *command* e grava um byte no registrador *data-out*
3. Hospedeiro liga o bit *command-ready*
4. Controlador nota o bit *command-ready* ligado e liga o bit *busy*
5. Controlador lê o registrador *command* e vê o comando *write*. Ele lê o registrador *data-out* para obter o byte e executa o I/O para o dispositivo
6. Controlador disliga o bit *command-ready*, desliga o bit *error* no registrador *status* para indicar que o I/O do dispositivo foi bem-sucedido, e desliga o bit *busy* para indicar que terminou

Interrupção (caiu na prova)

É o mecanismo de hardware que habilita um dispositivo para notificar a CPU. A CPU tem uma linha de solicitação de interrupção, quando a CPU detecta que um controlador confimou um sinal na linha de solicitação de interrupção, ela executa um salvamento de estado e salta para a rotina de manipulação de interrupções em um endereço fixo na memória. Depois realiza uma restauração de estado e executa uma instrução para retornar a CPU ao estado de execução anterior à interrupção.

- Interrupção não mascarável: reservada para eventos tais como erros de memória irrecuperáveis
- Interrupção mascarável: pode ser desativada pela CPU antes da execução de sequencias de intruções criticas que não devem ser interrompidas.

O mecanismo de interrupção também é usado para tratar *exeception* -> termina processos, "chasha" sistemas por conta de erros de hardware. Em sistemas multi-CPU as interrupções podem ser tratadas concorrentemente.

Acesso direto à Memória (caiu na prova)

É usado para evitar I/O programado (um byte por vez) para grandes quantidades de dados. Para que isso seja possível é necessário um controlador de DMA que ignora a CPU para transferir dado diretametente entre dispositivo I/O e memória.

O hospedeiro grava um bloco de comando DMA na memória. Esse bloco contém um ponteiro para a origem da transferência, um ponteiro para o destino da transferência, e uma contagem de número de bytes a serem transferidos. A CPU grava o endereço desse bloco de comando no controlador de DMA e, então, continua com outra tarefa. O controlador DMA passa a operar o bus da memória diretamente, inserindo endereços no bus para executar transferências sem a ajuda da CPU principal.

A interface de chamadas de sistema fornecida para aplicações é projetada para manipular várias categorias básicas de hardware, como:

- Dispositivos de blocos
- Dispositivos de caracteres
- Arquivos mapeados para a memória
- Sockets de rede e timers de intervalos programados

As chamadas de sistema encapsulam comportamentos de dispositivos em classes gerais. As camadas de driver de dispositivo escondem diferenças entre controladores de I/O e kernel. As chamadas usualmente bloqueiam os processos que as emitem (drives de disco e dispositivos de caracteres), ou seja, suspendem o processo até completar o I/O, chamadas sem bloqueio retornam no momento que estão disponíveis e as chamadas assíncronas processam enquanto o I/O executa. As chamadas de bloqueio e assíncronas são usadas pelo próprio kernel e por aplicações que não devem ser suspensas enquanto esperam que uma operação de I/O seja concluída.

Alguns SO também oferecem I/O vetorizado que permite que uma única chamada de sistema execute múltiplas operações de I/O envolvendo múltiplas locações.

O subsistema de I/O do kernel fornece numerosos serviços. Entre eles estão o scheduling de I/O, o armazenamento em buffer, o armazenamento em cache, o spooling, a reserva de dispositivos e a manipulação de erros. Outro serviço, a tradução de nomes, faz a conexão entre dispositivos de hardware e nomes de arquivo simbólicos usados pelas aplicações. Ele envolve vários níveis de mapeamento que traduzem nomes formados por cadeias de caracteres para drivers de dispositivos e endereços de dispositivos específicos e, então, para endereços físicos de portas de I/O ou controladores de bus. Esse mapeamento pode ocorrer dentro do espaço de nomes do sistema de arquivos ou em um espaço de nomes de dispositivos separado.

Transformando Solicitações de I/O em Operações de Hardware(importante)

1. Processo emite um read() com bloqueio para o descritor de um arquivo que foi aberto previamente
2. O código da chamada de sistema no kernel verifica a precisão dos parametros. Se os dados estão disponível na cache do buffer, eles são retornados ao processo, e a solicitação de I/O concluída.
3. Caso contrário, um I/O físico deve ser executado. O processo é removido da fila de execução e inserido na fila de espera. Eventualmente o subsistema de I/O envia a solicitação ao driver do dispositivo
4. O driver aloca espaço no buffer do kernel para receber os dados e inclui o I/O no schedule. Eventualmente, o driver envia comandos ao controlador do dispositivo gravando nos registradores de controle do dispositivo
5. O controlador do dispositivo opera o hardware do dispositivo para executar a transferência de dados
6. O driver pode sondar o status e os dados, ou pode ter estabelecido uma transferência DMA para a memória do kernel. A transferência é gerenciada por um controlador DMA que gera uma interrupão quando a transferência é concluída
7. O manipulador de interrupções recebe por meio da tabela de vetores de interrupções e armazena quaiquer dados necessários.
8. O driver do dispositivo recebe o sinal e determina que a solicitação foi concluida
9. O kernel transfere os dados ou retorna códigos para o espaço de endereçamento do processo solicitante e transfere o processo da fila de espera de volta para a fila de prontos
10. A transferência do processo para a fila de prontos o desbloqueia. Quando o scheduler atribui o processo à CPU, ele rotoma a sua execução quando se completa a chamada de sistema

O STREAMS é uma implementação e uma metodologia que fornece uma base estrutural para a programação de drivers de dispositivos e protocolos de rede usando abordagem modular e incremental. Por meio de fluxos (streams), drivers podem ser empilhados, com dados passando por eles para processamento, de maneira sequencial e bidirecional.

As chamadas de sistema de I/O são caras em termos de consumo da CPU por causa das muitas camadas de software existentes entre um dispositivo físico e uma aplicação. Essas camadas geram overhead proveniente de várias fontes: mudanças de contexto para atravessar o limite de proteção do kernel, manipulação de sinais e de interrupções para servir os dispositivos de I/O, e a carga sobre a CPU e o sistema de memória para copiar dados entre os buffers do kernel e o espaço da aplicação.

PROTEÇÃO

Os sistemas de computação contêm objetos que precisam ser protegidos contra a má utilização. Os objetos podem ser de:

- Hardware: memória, tempo de CPU e dispositivo de I/O
- Software: arquivos, programas e semáforos.

Direito de acesso é a permissão para executar uma operação sobre um objeto. Domínio é um conjunto de direitos de acesso.

Os processos são executados em domínios e podem usar qualquer um dos direitos de acesso do domínio para acessar e manipular objetos. Um processo pode ficar limitado a um domínio de proteção ou ter permissões para permutar de um domínio para outro.

A matriz de acesso é um modelo geral de proteção que fornece um mecanismo de proteção sem impor uma política de proteção específica ao sistema ou aos seus usuários.

A matriz de acesso é, normalmente, implementada como listas de acesso associadas a cada objeto, ou como listas de competências associadas a cada domínio. Podem incluir a proteção dinâmica no modelo de matriz de acesso considerando os domínios e a própria matriz de acesso como objetos. A revogação de direitos de acesso em um modelo de proteção dinâmica costuma ser mais fácil de implementar com um esquema de lista de acesso do que com uma lista de competências.

Os sistemas reais tendem a fornecer proteção apenas para arquivo. O UNIX é representativo, fornecendo proteção de leitura, gravação e execução separadamente para o proprietário, o grupo e o público geral de cada arquivo.

A proteção baseada em linguagens fornece uma arbitragem refinada de solicitações e privilégios do que o SO é capaz de fornecer. Por exemplo, uma única JVM Java pode executar vários threads, cada um em uma classa de proteção diferente. Ela impõe as solicitações de recursos por intermédio de uma sofisticada inspeção de pilha e de segurança de tipos de linguagem.

SEGURANÇA

Proteção é um problema interno, já segurança deve considerar tanto o sistema de computação quanto o ambiente - pessoas, prédios, empresas, objetos de valor, ameaças - dentro do qual o sistema é usado. (caiu na prova - ler livro para mais detalhes)

Os dados armazenados no sistema de computação devem ser protegidos contra:

- Acesso não autorizado
- Destruição
- Alteração maliciosa
- Intrudução acidental de inconsistências

É mais fácil se proteger contra a perda acidental da consistência dos dados do que se proteger contra o acesso malicioso aos dados.

A proteção absoluta das informações armazenadas em um sistema de computação contra abuso malicioso não é possível; mas o custo para infrator pode ser suficiente alto para deter quase todas (quando não todas) as tentativas de acesso a essas informações sem autorização apropriada.

Vários tipos de ataques podem ser lançados contra programas e contra computadores individuais ou coletivos:

- Próprios dispositivos.
- Estouro de pilha ou de buffer: permitem que invasores bem sucedidos alterem seu nível de acesso ao sistema
- Vírus e vermes: são autoperpetuáveis e às vezes infectam milhares de computadores
- Ataques de recusa de serviço: impedem o uso legítimo de sistemas-alvo

Criptografia (caiu na prova - ler livro para mais detalhes) : limita o domínio de receptores de dados enquanto

A criptografia é usada para fornecer sigilo aos dados que estão sendo armazenados ou transferidos. Existem dois tipos:

- Simétrica: requer uma chave compartilhada
- Assimétrica: fornece uma chave pública e uma chave privada

Autenticação: limita o domínio de emissores.

A autenticação, quando combinada com o hashing, pode comprovar que os dados não foram alterados.

Além da proteção-padrão com nome de usuário e senha outros métodos de autentificação são usados para identificar os usuários legítimos de um sistema.

- PrSenhas descartáveis: mudam de uma sessão para outra para evitar ataques de reexecução
- A autenticação com dois fatores: requer dois tipos de autenticação, tal como uma calculadora em hardware junto com um PIN de ativação
- A autenticação com múltiplos fatores: usa três tipos ou mais de autenticação

Os métodos de prevenção ou detecção de incidentes de segurança incluem:

- Sistemas de detecção de invasões
- Softwares antivírus
- Auditoria e registro em log de eventos dos sitema
- Monitoramento de alterações em softwares do sistema
- Mnitoramento de chamadas de sistema e firewalls

MAQUINAS VIRTUAIS (caiu na prova)

A virtualização fornece a um convidado uma duplicata do hardware subjacente de um sistema.

Múltiplos convidados podem ser executados em um sistema, cada um acreditando que é o SO nativo com controle total do sistema.

A virtualização tipo 0: é implementada no hardware e requer modificações no SO para garantir operação apropriada. Essas modificações oferecem um exemplo de paravirtualização, em que o SO não desconhece a virtualização, mas, em vez disso, tem recursos adicionados e algoritmos alterados para melhorar as funções e o desempenho da virtualização.

A virtualização tipo 1: um monitor de máquina virtual (VMM) do hospedeiro fornece o ambiente e os recursos necessários à criação, execução e destruição de máquinas virtuais convidadas. Cada convidado inclui todos os softwares tipicamente associados a um sistema nativo completo, inclusive o SO, drivers de dispositivo, aplicações, contas de usuários e assim por diante.

Os hipervisores tipo 2: são simplesmente aplicações executadas em outros SO, que não sabem que a virtualização está ocorrendo. Esses hipervisores não se beneficiam de suporte de hardware ou do hospedeiro e, assim, devem executar todas as atividades de virtualização no contexto de um processo.

A virtualização de ambientes de programação: a linguagem especifíca uma aplicação container em que programas são executados, e essa aplicação fornece serviços aos programas. A emulação é usada quando um sistem hospedeiro tem uma arquitetura e o convidado foi compilado para uma arquitetura diferente. Todas as instruções que o convidado quiser executar devem ser traduzidas de seu conjunto de instruções para o do hardware nativo. Embora esse método envolva alguma queda de desempenho, ele é compensado pela utilidade de poder executar programas antigos em hardware incompátivel mais recente ou executar jogos projetados para consoles antigos em hardware moderno.

Os VMMs tiram partido de qualquer suporte de hardware que esteja disponível quando da otimização do scheduling da CPU, do gerenciamento de memória e dos módulos de I/O para fornecer aos convidados um uso de recursos ótimo enquanto se protegem dos convidados e protegem os convidados uns dos outros.

Em qual situação é melhor a utilização do Polling, e do Interrupt?

A estratégia de polling é útil quando há uma necessidade de verificar periodicamente o estado de um dispositivo de entrada ou saída para determinar se houve alguma mudança. Por exemplo, ao ler um teclado, o sistema operacional pode usar a estratégia de polling para verificar se alguma tecla foi pressionada e, se sim, qual tecla foi pressionada.

A interrupção, por outro lado, é mais adequada para situações em que é necessário que o sistema operacional reaja imediatamente a mudanças no estado de um dispositivo. Por exemplo, ao receber uma mensagem de rede, o sistema operacional pode usar uma interrupção para interromper sua execução atual e tratar a mensagem de rede de forma apropriada.

Em geral, a interrupção é preferível quando é necessário tratar eventos de forma rápida e precisa, enquanto a estratégia de polling é mais adequada para situações em que a resposta precisa ser mais lenta ou o custo de interrupção é alto.

Questão 2 - Em que consiste a estratégia de DMA?

A sigla DMA significa Direct Memory Access, ou Acesso Direto à Memória. É uma estratégia utilizada pelo sistema operacional para transferência de dados entre a memória principal e dispositivos de entrada e saída sem a intervenção do processador.

Na estratégia de DMA, o processador configura um controlador DMA para realizar a transferência de dados. O controlador DMA é responsável por gerenciar a transferência de dados diretamente entre a memória e o dispositivo sem a necessidade de envolver o processador em cada etapa da transferência. Isso permite que o processador continue sua execução normal enquanto a transferência de dados é realizada pelo controlador DMA.

A estratégia de DMA é útil para acelerar a transferência de grandes quantidades de dados, pois evita que o processador seja sobrecarregado com a tarefa de realizar a transferência de dados. Além disso, a estratégia de DMA permite que o processador continue sua execução normal sem interrupções, o que melhora o desempenho geral do sistema.

Questão 3 - Definição de criptografia simétrica e assimétrica com relação à quantidade de chaves utilizadas.

Na criptografia simétrica, uma única chave é usada tanto para cifrar quanto para decifrar os dados. Isso significa que a mesma chave é usada para proteger os dados durante a transmissão e para desprotegê-los na chegada. Este tipo de criptografia é rápido e eficiente em termos de recursos, mas requer que a chave seja compartilhada de forma segura entre as partes envolvidas na comunicação.

Na criptografia assimétrica, também conhecida como criptografia de chave pública, dois pares de chaves são usados: uma chave pública e uma chave privada. A chave pública é compartilhada com todos e é usada para cifrar os dados. A chave privada é mantida secreta e é usada para decifrar os dados. Isso permite que as informações sejam protegidas durante a transmissão, mesmo que a chave pública seja amplamente conhecida. Este tipo de criptografia é mais seguro, mas geralmente requer mais recursos e é mais lento do que a criptografia simétrica.

Em resumo, a criptografia simétrica é usada quando há uma necessidade de proteger a comunicação com uma chave compartilhada, enquanto a criptografia assimétrica é usada quando há uma necessidade de proteger a comunicação com chaves separadas para cifrar e decifrar os dados.

Questão 4 - Em que consiste o ataque de recusa de serviço

O ataque de negação de serviço (DoS, na sigla em inglês) é uma forma de interrupção maliciosa de um serviço, com o objetivo de torná-lo indisponível para os seus usuários legítimos. Isso é feito enviando uma grande quantidade de requisições ao serviço, com o objetivo de sobrecarregar seus recursos, tornando-o incapaz de atender aos requisitos dos usuários.

Os ataques de negação de serviço são particularmente perigosos porque podem interromper a disponibilidade de serviços críticos, como bancos de dados, sistemas financeiros e outros serviços sensíveis. Além disso, esses ataques são difíceis de serem detectados e de serem protegidos contra, uma vez que envolvem uma grande quantidade de tráfego na rede.

Para proteger contra ataques de negação de serviço, é importante implementar medidas de segurança, como a configuração de firewalls e a implementação de técnicas de balanceamento de carga, além de monitorar constantemente a rede para detectar e responder a ataques. Além disso, é importante manter os sistemas e aplicativos atualizados com as últimas correções de segurança para minimizar a exposição a ataques.

Questão 5 - Dê a definição dos termos host, guest e VMM. Além disso, diferencie emulação de virtualização.

Host: Refere-se ao sistema operacional e hardware subjacente que executa a máquina virtual. É o sistema operacional real que gerencia os recursos do computador, como CPU, memória, armazenamento, entre outros.

Guest: Refere-se ao sistema operacional e aplicativos que são executados dentro da máquina virtual. O sistema operacional convidado "acredita" que tem acesso direto aos recursos do hardware, mas na realidade está sendo virtualizado e compartilhado pelo host.

VMM (Virtual Machine Monitor): É o componente software responsável por gerenciar e monitorar a máquina virtual. O VMM é responsável por criar e executar máquinas virtuais, alocar recursos do host para as máquinas virtuais e garantir a separação e a segurança entre as máquinas virtuais.

Em resumo, o host é o sistema operacional e hardware subjacente, o guest é o sistema operacional e aplicativos executados na máquina virtual e o VMM é o componente de software responsável por gerenciar a máquina virtual.

Emulação: É o processo de imitar o comportamento de um sistema ou dispositivo, geralmente para ser executado em um ambiente diferente. Em uma máquina virtual, a emulação é usada para emular um sistema operacional ou hardware diferente, permitindo que o sistema operacional convidado execute aplicativos que normalmente não funcionariam no sistema operacional host.

Virtualização: É o processo de criar uma representação virtual de algum recurso, como CPU, memória, armazenamento, rede, entre outros. Em uma máquina virtual, a virtualização é usada para fornecer ao sistema operacional convidado acesso aos recursos do hardware do host, enquanto mantém a separação e a segurança entre as máquinas virtuais.

Em resumo, a emulação é usada para imitar o comportamento de um sistema ou dispositivo, enquanto a virtualização é usada para fornecer acesso virtual aos recursos do hardware. Ambas as técnicas são importantes na máquina virtual, mas servem propósitos diferentes.