**Trabalho para a V1**

Nome: Davi Milhome, Marcielen Maia

**Conceitos Básicos**

1. Explique o que é e pra que serve um sistema operacional (SO).

Um sistema operacional serve de interface, uma camada a mais entre usuário e hardware. Dessa forma, ele serve para facilitar a interação entre usuário e máquina e aplicações. Além disso, um S.O serve para abstrair o hardware, ou seja, o usuário não precisa ser um grande conhecedor de hardware e nem configurar suas próprias rotinas de entrada e saída para utilizar suas aplicações.

1. Cite e explique as funções básicas de um SO.

As principais funções de um sistema operacional são servir de interface entre o usuário e o hardware, ou seja, abstrair, de maneira geral, a necessidade do usuário interagir e configurar seu hardware e, além disso, ser o responsável pelo gerenciamento de recursos. Ou seja, suponha que você tenha uma impressora, é necessário que algum programa gerencie a fila de impressão para que a impressão de um determinado usuário não impacte a de outro, o S.O atua nesse gerenciamento não somente para usuários diferentes mas também para um mesmo usuário com diferentes aplicações.

1. Explique o que é um processamento batch.

O processamento batch é caracterizado por criar uma espécie de fila de processamento, basicamente, os programas ficam em fila de execução e passam a utilizar o recurso do processador assim que este é desocupado por outro programa.

1. Quais são os tipos de SO?

Existem os sistemas monoprogramáveis, nos quais somente é possível executar um programa por vez, ou seja, a execução de um determinado JOB toma para si todos os recursos da máquina.

Além disso, existem os sistemas multiprogramáveis que podem ser monousuários ou multiusuários. Esse tipo de sistema pode empregar uma ou mais estratégias de processamento, como o processamento BATCH, time-sharing e real-time.

Por fim, existem os sistemas multiprocessadores, que podem ser fortemente acoplados e fracamente acoplados.

1. Explique o que são os Sistemas Monoprogramáveis/Monotarefas.

Um sistema monoprogramável se caracteriza por utilizar todos os recursos da máquina para a execução de uma só aplicação, ou seja, os recursos da máquina ficam "travados" até o final da execução de um determinado programa.

1. Explique o que são os Sistemas Multiprogramáveis/Multitarefas.

Os sistemas multiprogramáveis têm a capacidade de executar mais de uma tarefa ao mesmo tempo, e de suportar mais de um usuário. Em geral, esse tipo de sistema apresenta vantagem, pois há barateamento de custos uma vez que os recursos podem ser compartilhados por diferentes programas e usuários, além disso, diminuição no tempo geral de execução de programas.

1. Explique o que são os Sistemas Batch.

O processamento batch é caracterizado por criar uma espécie de fila de processamento, basicamente, os programas ficam em fila de execução e passam a utilizar o recurso do processador assim que este é desocupado por outro programa

1. Explique os que são os Sistemas de Tempo Compartilhado.

No caso do time-sharing, cada programa recebe uma pequena fatia de tempo para a sua execução, sendo assim, mais de um programa pode ser executado "ao mesmo tempo". Nesse caso, mesmo que o recurso esteja sendo dividido, do ponto de vista da aplicação é como se todo o recurso estivesse disponível exclusivamente para ela.

1. Explique os que são os Sistemas de Tempo Real.

No caso do real-time, diferentemente do time-sharing, entende-se que o tempo de execução de um determinado programa não pode variar, pois isso causaria erros na sua execução. Sendo assim, os programas são executados por ordem de prioridade, só deixando de ser executados quando um programa de prioridade mais alta entra na pilha. Dessa forma os recursos do sistema ficam exclusivos para um determinado programa.

Aplicações recomendadas para real-time são as que utilizam grande processamento de dados e têm criticidade, como por exemplo, aplicações de usinas nucleares, tráfego aéreo, etc.

1. Explique o que são os Sistemas com Múltiplos Processadores.

São sistemas onde duas ou mais UCPs trabalham em conjunto. A vantagem desse tipo de sistema é permitir que vários programas sejam executados ao mesmo tempo ou que um mesmo programa seja subdividido para ser executado em diversos processadores. Geralmente esse tipo de sistema é utilizado para o processamento científico.

Outras características importantes que esses sistemas proporcionam são a escalabilidade, disponibilidade e balanceamento de carga.

1. Explique o que são os Sistemas Fortemente Acoplados.

Os sistemas fortemente acoplados são uma divisão dos sistemas com múltiplos processadores, eles se caracterizam por dividir uma única memória principal e um único S.O entre as unidades de processamento.

1. Explique o que são os Sistemas Fracamente Acoplados.

Nos sistemas fracamente acoplados, cada conjunto de processador tem sua memória e S.O individuais. Por conta disso, os sistemas fracamente acoplados também são conhecidos como multicomputadores.

**Componentes do SO**

1. Explique a arquitetura de von Neumann.

A arquitetura de Von Neumann é baseada no princípio do armazenamento de dados e programas na memória, esse fator implica que as instruções dos programas e os dados sejam manipulados da mesma forma, ou seja, pelos mesmos componentes de hardware. Dessa forma, essa arquitetura é composta de cinco elementos principais: CPU, memória, unidades de entrada e saída, barramento e sistema de interrupções.

1. Quais são e explique os registradores mais importantes da Unidade Central de Processamento (UCP).

Program Counter (PC)

Contém o endereço da próxima instrução que o processador deve buscar e executar. Toda vez que o processador buscar uma nova instrução, esse registrador é atualizado com o endereço da memória da instrução seguinte a ser executada.

Stack Pointer (SP)

Contém o endereço de memória do topo da pilha, que é a estrutura de dados onde o sistema que mantém informações sobre programas que estão sendo executados e tiveram que ser interrompidos.

Program Status Word

Responsável por armazenar informações sobre a execução de instruções, como a ocorrência de overflow. A maioria das instruções, quando executadas, alteram o registrador de status conforme o resultado.

1. Quais são os tipos de memórias que um computador tem? Diferencie-as.

Um computador é composto por memória principal, memória secundária e memória cache, além disso, os registradores também podem ser considerados um tipo de memória.

A memória principal é o local onde são armazenadas instruções e dados. A memória principal pode ser classificada em função da sua volatilidade, que é a capacidade de a memória preservar seu conteúdo mesmo sem uma fonte de alimentação ativa, dessa forma, as memórias do tipo RAM são voláteis, quantos as ROM (ready only memory) e EPROM (erasable programmable ROM) são do tipo não voláteis.

A memória cache é uma memória volátil de alta velocidade, porém, com uma pequena capacidade de armazenamento. O tempo de acesso a ela é muito menor que o da memória principal. Seu tamanho é limitado em função de seu alto custo. Então, a memória cache armazena uma pequena parte do conteúdo da memória principal. Toda vez que o processador faz referência a um dado armazenado na memória, verifica-se se este dado não está na cache. Caso o dado esteja na cache, a situação é chamada de cache hit, caso não, é chamada de cache miss. Em algumas construções, há a hierarquização da construção de cache em múltiplos níveis. O nível mais alto é chamado de L1 (Level 1), com baixa capacidade de armazenamento e altíssima velocidade, e assim por diante.

Já a memória secundária é uma memória não volátil para armazenamento de dados e programas. Seu acesso é lento, porém, o custo é baixo.

Por fim os registradores são um tipo de memória de alta velocidade, que armazenam dados temporariamente. Os registradores são alocados na CPU.

1. Explique o Princípio da Localidade.

Os dados são alocados na cache segundo o princípio da localidade, que é a tendência do processador, ao longo da execução de um programa, referenciar instruções da memória principal localizados em endereços próximos. Essa tendência é justificada pela alta incidência de sub-rotinas e acesso a estruturas de dados como vetores e tabelas. O princípio da localidade garante então que após a transferência de um novo bloco para a cache, haverá uma alta probabilidade de cache hits em futuras referências.

1. Quais são e explique os tipos de barramentos.

Os barramentos ainda podem ser de três tipos: barramentos de processador/memória, barramentos de E/S e barramentos de blackpane. Os barramentos de processador/memória são de alta velocidade e curta extensão, diferentemente do barramento de E/S que possui uma maior extensão e são mais lentos. Algumas arquiteturas de alto desempenho utilizam o terceiro barramento, blackplane. Nessa organização, o barramento de E/S não se conecta diretamente ao barramento do processador/memória. A principal vantagem nesse caso é reduzir o número de adaptadores existentes no barramento processador/memória e, dessa forma, otimizar seu desempenho.

1. Explique a Arquitetura Pipeline e explique a sua vantagem em relação a uma arquitetura sequencial comum.

Da mesma forma em que uma linha de montagem, a execução de uma instrução pode ser dividida em subtarefas, como as fases de busca da instrução e dos operandos, execução e armazenamento dos resultados. O processador, através de suas várias unidades funcionais pipeline, funciona de forma a permitir que, enquanto uma instrução esteja em fase de execução, uma outra instrução possa estar na fase de busca simultaneamente.

O pipelinening pode ser empregado em sistemas de um ou mais processadores, em diversos níveis, e tem sido a técnica de paralelismo mais utilizada para aumentar o desempenho dos sistemas computacionais.

1. Explique e diferencie as Arquiteturas RISC e CISC.

RISC (Reduced Instruction Set Computer): Essa arquitetura se caracteriza por possuir poucas instruções de máquina, em geral bastante simples, que são executadas diretamente pelo hardware. Na sua maioria, essas instruções não acessam a memória principal, trabalhando principalmente com registradores (que se apresentam em grande número). Essa arquitetura permite mais velocidade e facilita o pipelining.

CISC (Complex Instruction Set Computers): Essa arquitetura possui instruções complexas, que são interpretadas por microprogramas. O número de registradores é pequeno, e qualquer instrução pode referenciar a memória principal. Os microprogramas definem a linguagem de máquina de um computador CISC.

1. Explique e diferencie o Compilador e o Interpretador.

O compilador e o interpretador são tipos de tradutores que têm como objetivo final possibilitar a execução de um código de alto nível pelo CPU, porém, eles atuam de forma diferente.

O compilador funciona com a tradução do código de alto nível diretamente em linguagem de máquina, gerando um módulo-objeto (também chamado de executável). Além disso, durante a tradução, o compilador pode fazer análises mais profundas e implementar algumas otimizações de execução. Sua vantagem é possibilitar a portabilidade do código de maneira mais simples e, além disso, reduzir o tempo de execução uma vez que o código é compilado.

O interpretador, por outro lado, não gera um módulo objeto, ele traduz e executa as instruções imediatamente. A desvantagem de se utilizar um interpretador é o tempo gasto em cada execução, que pode se repetir, por outro lado, o interpretador tem a vantagem de ser mais flexível, possibilitando a implementação de tipos de dados dinâmicos (dados que podem mudar seu tipo durante a execução).

1. Explique o papel do Tradutor.

O tradutor, que pode ser de vários tipos, tem sua atuação de maneira geral na transformação de um programa fonte em um programa objeto. Resumindo, o tradutor converte as representações simbólicas de um determinado código fonte em linguagem de máquina, ou seja, sem o Tradutor, seria necessário escrever programas em linguagem de máquina.

1. Explique o papel do Linker.

O linker é responsável por buscar as referências externas/dependências de um determinado código e acoplá-las para que o código possa ser executado. Ou seja, o linker associa o programa fonte a suas sub-rotinas que não foram definidas dentro do próprio programa. A partir da linkagem é gerado o arquivo executável.

1. Explique o papel do Loader

O Loader é o responsável por carregar os programas na memória antes de eles serem executados. A atuação do loader irá depender do linker presente no sistema, dessa forma,existem dois tipos de loader

Loader absoluto: Nesse caso mais simples, o loader precisa apenas saber o endereço definido para a primeira instrução e o tamanho do programa. Então, o loader transfere o programa da memória secundária para a memória principal e inicia a execução.

Loader relocável: No caso do código relocável, o programa pode ser carregado em qualquer posição de memória, e o loader é responsável pela realocação no momento do carregamento.

1. Explique o papel do Depurador (Debugger).

O depurador é um utilitário auxiliar no desenvolvimento de aplicações. Ele serve para facilitar e acelerar a correção de erros em códigos. Basicamente o debugger fornece ferramentas ao usuário para identificar erros no programas executando uma instrução de cada vez. dentre as possibilidades que o debbuger pode apresentar, estão: Acompanhamento de instrução por instrução, possibilidade de alterar e visualizar o conteúdo de variáveis, implementar pontos de parada no programa para melhoria e análise, criar um watchpoint, ou seja, escolher uma variável e informar toda vez que essa variável mudar de valor.

CAP 4

1. Explique o procedimento de Ativação do Sistema Operacional.

Os componentes do S.O devem ser carregados para a memória principal toda vez que o computador é ligado, isso é feito por intermédio de um processo denominado boot.

O processo se inicia com a execução de um programa chamado bootloader, esse programa chama a execução de um outro programa, o POST (Power-On Self Test), que indica possíveis problemas de hardware. Após isso, o procedimento de ativação verifica se há sistema operacional. Se um dispositivo com S.O for encontrado, um conjunto de instruções é carregado para a memória principal. Além da carga, a ativação do sistema também consiste na execução de arquivos de inicialização onde são especificados procedimentos de customização e configuração de hardware..

1. Apenas cite as funções do núcleo do SO.

De maneira geral, podemos atribuir como principais funções da kernel:

\* Tratamento de interrupções e exceções

\* criação e eliminação de processos e threads

\* sincronização e comunicação entre processos e threads

\* escalonamento e controle dos processos e threads

\* gerência de memória

\* gerência de arquivos

\* gerências de I/O

\* suporte a redes locais e distribuídas

\* contabilização do uso do sistema

\* auditoria e segurança do sistema

Além disso, o S.O deve ser responsável pelo controle da utilização da CPU, de forma a impedir que algum programa monopolize o seu uso inadequadamente.

1. Explique o que é uma System Call.

O núcleo do sistema é composto por rotinas que oferecem serviços aos usuários e aplicações, todas as funções do núcleo são implementadas em rotinas que necessariamente possuem instruções privilegiadas. Ou seja, isso implica que o processador necessariamente deve estar em modo kernel, então, foi necessária a implementação de um mecanismo de proteção.

Dessa forma, todo o controle de execução de rotinas do sistema operacional é realizado pelo mecanismo do system call, também chamado de syscall. Resumindo, quando uma aplicação deseja utilizar uma rotina do S.O, o mecanismo de syscall é ativado. Inicialmente é verificado os privilégios da aplicação, em caso de privilégios suficientes, o sistema salva o conteúdo corrente nos registradores, troca o modo de acesso para Kernel e realiza o desvio para a rotina alterando o registrador PC com o endereço da rotina chamada. Ao término da execução da rotina, o modo de acesso volta para o usuário e o contexto dos registradores é restaurado para que a aplicação continue a execução a partir da instrução que a syscall foi evocada.

1. Explique a Arquitetura Monolítica de um SO.

A arquitetura monolítica pode ser comparada com uma aplicação formada por vários módulos que são compilados separadamente e depois linkados, formando um único grande executável. Os primeiros S.Os foram desenvolvidos nesse modelo, porém, isso torna o código desnecessariamente complexo, o que faz com que a manutenção e o desenvolvimento sejam bastante difíceis.

1. Explique a Arquitetura em Camadas de um SO.

Com o aumento da complexidade e tamanho dos S.Os técnicas de programação estruturada e modular foram incorporadas ao seu projeto. Na arquitetura de camadas o sistema é dividido em níveis sobrepostos. Cada camada oferece um conjunto de funções que podem ser utilizadas exclusivamente pelas camadas superiores.

A vantagem da arquitetura de camadas é isolar as funções do S.O, facilitando sua manutenção e depuração, além disso, cria uma hierarquia de níveis de modo de acesso, protegendo mais as camadas mais internas. Cada nova camada implica uma mudança no modo de acesso.

1. Explique a Arquitetura de Máquina Virtual de um SO.

Nesse caso,um sistema computacional é formado por níveis, onde a camada de mais baixo nível é o hardware. O modelo de máquina virtual (ou VM) cria um nível intermediário entre o hardware e o S.O, denominado gerência de máquinas virtuais. Esse nível cria diversas máquinas virtuais independentes, onde cada uma oferece uma cópia virtual do hardware.

Como cada máquina virtual é independente das demais, é possível que cada VM tenha seu próprio sistema e que seus usuários executem suas aplicações como se todo o computador estivesse dedicado a eles.

Além de permitir a convivência de sistemas operacionais diferentes no mesmo computador, esse modelo cria um isolamento total entre cada VM, oferecendo grande segurança para cada máquina individualmente. A desvantagem dessa arquitetura é sua grande complexidade devido a necessidade de compartilhar e gerenciar recursos de hardware entre as VMs.

1. Explique a Arquitetura Microkernel de um SO.

Uma tendência nos sistemas operacionais modernos é tornar o núcleo do sistema operacional menor e o mais simples possível. Para implementar essa ideia, os serviços do sistema são disponibilizados através de processos onde cada um é responsável por oferecer um conjunto específico de funções, como gerência de arquivos, gerência de memória e escalonamento.

Sempre que uma aplicação deseja algum serviço, é realizada uma solicitação ao processo responsável. Neste caso, a aplicação que solicita o serviço é chamada de cliente, enquanto o processo que responde a solicitação é chamado de servidor. A principal função do núcleo nessa arquitetura é realizar a comunicação,ou seja, a troca de mensagens entre cliente e servidor.

Esse tipo de arquitetura tem sua vantagem na facilidade de adicionar novos servidores à medida que o número de clientes aumenta, conferindo uma grande escalabilidade ao S.O. Porém, apesar de suas vantagens, a implementação deste tipo de modelo na prática é muito difícil. Primeiramente existe o problema de desempenho devido a necessidade de mudança de modo de acesso a cada comunicação entre cliente e servidor. Outro problema é que certas funções do S.O exigem acesso direto ao hardware, como operações de E/S.

**Gerência de Processos – Estados**

1. Explique o que é o Multiprocessamento

O multiprocessamento é quando o sistema operacional consegue realizar mais de um processo ao mesmo tempo ou de um só por mais de um processador. Para isso, precisa conter dois ou mais processadores físicos ou lógicos, não tem uma central de controle, mas sim cada um com sua unidade de controle; os processadores dividem o mesmo espaço de memória e o hardware é controlado por um sistema operacional.

1. Quais os três contextos que compõem a Estrutura de Processo. Cite as informações que são armazenadas em cada contexto.

Contexto de Software; Contexto de Hardware e Espaço de Endereçamento.

No contexto de Software, vai dizer quais as limitações e o que vai ter dentro do processo, de limite pode ser o máximo de arquivos que vão abrir no mesmo momento, qual vai ser a prioridade pra executar os processos e o tamanho do buffer. Esse contexto é constituído por grupos que informam sobre o processo: Identificação, em que cada processo recebe um PID, ou seja, um número para identificação própria, o nome, o usuário e nisso existe uma certa segurança em que só pode acessar os processos, arquivos etc, se tiver a mesma identificação de usuário; Quotas, que coloca limite nos recursos do processo, como tamanho máximo de memória, número de arquivos abertos ao mesmo tempo etc; Privilégios, em que decide o que o processo pode fazer em relação a ele mesmo, aos outros processos e ao sistema operacional.

No contexto de Hardware, armazena registradores de status, registrador de PC (program counter), registrador SP (stack pointer), registradores gerais.

Espaço de endereçamento, o qual pertence na parte em que as rotinas e os dados vão ser armazenados para que o processo seja executado e cada processo tem esse espaço e outros processos não podem interferir.

1. Contexto de Hardware

O contexto de Hardware, armazena registradores de status, registrador de PC (program counter), registrador SP (stack pointer), registradores gerais. Quando algo está sendo executado, a sua parte de hardware, que é o seu contexto, vai tá armazenado nos registradores da CPU.

1. Explique o fluxo de mudança de contexto entre dois programas.

Visando que a concorrência entre os programas ocorra sem problemas, é necessário que todas as informações do programa interrompido sejam guardadas para que, quando este volte a ser executado, não lhe falte nenhuma informação necessária ao processamento. Estas informações são fundamentais para que o sistema operacional possa gerenciar a execução concorrente de programas, e é a base de qualquer ambiente multiprogramável. Dessa forma, o conceito de processo pode ser definido como sendo o conjunto necessário de informações para que o sistema operacional implemente a concorrência de programas.

Então, a troca de um processo para outro no processador é comandada pelo sistema operacional e é denominada mudança de contexto. No caso, cada usuário tem a impressão de possuir o processador e todos os demais recursos do hardware, porém, o que de fato acontece é que o processador executa a instrução de um processo em uma determinada quantidade de tempo, guarda seu contexto em registradores e alterna para outro processo.

1. Explique o que é um Bloco de Controle de Processos (PCB).

É uma estrutura de dados em que o processo é elaborado pelo Sistema Operacional e a partir dele, o sistema vai ter todas as informações que ficam dentro dos contextos do processo, como o de software, hardware e também o espaço de endereçamento. Esses blocos ficam dentro da memória principal em um espaço só deles no sistema operacional e esse espaço é limitado por um critério do próprio sistema.

1. Explique os estados em que um processo pode estar. Explique também as mudanças de estados, mencionando as listas em que os PCBs podem ficar.

Quando o sistema é multiprogramável, como o processador é compartilhado entre os processos, enquanto vão acontecendo as execuções, os processos ficam mudando de estado para dar espaço para os outros. Esses estados são: Execução, que é quando está sendo executado pelo processador e quando só tem um processador, tem um tempo que vai de acordo com cada sistema, para o processador ficar nesse estado e quando tem mais de um processador, pode acontecer de mais de um processo estar sendo executado ou um só sendo executado em duas UCP; Pronto, quando o processo está aguardando para ser executado e quem decide qual e quando vai ser executado, é o Sistema Operacional de acordo com o critério de cada um e esses processos em estado de pronto ficam em ordem de importância, assim os com maior prioridade vão sendo executados primeiro; Espera, em que os processos ficam em listas e cada lista é de um evento, quando o evento acontece, todas as listas que a ele estão associada, vão para a lista de pronto.   
  
 As mudanças de estado do processo acontecem por eles mesmo, de forma voluntário ou pelo sistema operacional, de forma involuntária.   
Mudanças: Pronto para Execução, quando o processo já foi criado, fica na lista de pronto esperando para ser executado; Execução para Espera, quando acontece do sistema operacional suspender a execução do processo ou quando o próprio gera isso, com uma operação de entrada e saída; Espera para Pronto, quando o que foi requisitado é respondido ou o recurso esperado é permitido e sempre o processo vai passar por isso; Execução para pronto, quando por exemplo, acaba o tempo limite de execução e acaba tendo que voltar para a lista de pronto e esperar para ser executado novamente.

1. Explique os conceitos de Processos Independentes, Subprocessos e Multithreads.

Os Processos Independentes não vão ser associados ao criador, só vai precisar do PCB, pra ter o contexto de software, contexto de hardware e o próprio espaço de endereçamento.

Os Subprocessos são criados com uma hierarquia, o primeiro é o processo-pai e os que vão sendo criados, são os subprocessos e esses podem criar seus próprios subprocessos, e cada subprocesso tem seu PCB, o que tem de desvantagem é que se o processo principal for deletado, todos os outros também vão ser, além de demandar recursos, já que pra cada processo vai ter um PCB, o que também pode complicar a comunicação entre eles.

Quando é Multithread, tem a capacidade de sustentar várias threads e cada uma vai ser relacionada a uma parte do código de aplicação, compartilhando o processador e tendo seu próprio contexto de hardware, porém compartilha o contexto de software e o espaço de endereçamento, o que ajuda na comunicação entre elas.

1. Explique os conceitos de Processos Foreground e Background.

Os processos foreground permitem que o usuário tenha uma comunicação direta com os processos durante a execução, por exemplo, um teclado, em que o ao digitar está tendo contato com a entrada de dados.

Os processos Background funcionam por trás das telas, sem que o usuário tenha permissão e interação, apenas vão ter arquivos de entrada/saída, como o processamento do tipo batch.

1. Explique os conceitos de Processos CPU-bound e I/O-bound.

Os processos de CPU-bound são os que estão ligados a CPU, na maior parte do tempo sendo executados ou em estado de pronto, sem muitas mudanças de estado, tendo poucas execuções de leitura e de gravação, operando mais cálculos.  
 I/O-bound passa mais tempo em estado de espera, já que precisa realizar muitas operações de entrada/saída, tendo como base as leituras, os processamentos e as gravações. Os processos foreground são um bom exemplo de I/O-bound, pelo fato de ter entrada/saída e os usuários conseguem interagir, porém é mais lento.

**Gerência de Processos – Escalonamento**

1. Explique o que é um algoritmo de Escalonamento e quais são as preocupações principais de um algoritmo de Escalonamento.

Escalonamento determina qual processo vai ser executado, em qual momento e também por quanto tempo. Tem como função fazer o equilíbrio da UCP entre a execução dos processos, deixar o processador ocupado a maior parte do tempo possível, entre outras coisas que funcionam para diminuir tempo de espera e otimizar tempo de processamento. Existem os critérios de Escalonamento, tem a rotina do sistema que aplica esses critérios, que pode ser chamada de escalonador, tem o Scheduler e Dispatcher. Os critérios de escalonamento são: Utilização do Processador, Throughput, Tempo de UCP, Tempo de espera, Tempo de Turnaround e Tempo de resposta.

2. Qual o papel de um Scheduler e de um Dispatcher? Explique o que é latência do dispatcher.

Scheduler é uma rotina do sistema operacional, responsável por implementar os critérios do escalonamento e é fundamental porque todo o compartilhamento do processador depende dessa rotina. O dispatcher também é uma rotina, mas responsável pela troca de contexto, após o scheduler decidir quem vai usar o processador, e na troca desse contexto, o tempo que for utilizado para trocar, é chamado de latência do dispacther.

3. Explique o que são Não preempção e Preempção.

Escalonamentos não preemptivos são aqueles que não tem a possibilidade de trocar o processo de contexto e fica somente no estado de execução, o que pode mudar isso é se ele finalizar a execução dele ou no código dele mesmo tenha a instrução para ir para o estado de espera.

No escalonamento preemptivo, já é possível a mudança de estado, já que o sistema operacional pode colocar o processo no estado de pronto para que outro processo vá para a UCP, também é fundamental quando for preciso colocar certos processos em prioridade e distribuir esses processos de forma equilibrada para a UCP.

4. Explique o que são: Throughput, Tempo de CPU, Tempo de Espera, Tempo de Turnaround e Tempo de Resposta.

Os critérios a serem avaliados na política de escalonamento, são: Throughput, que é o número de processos dentro de um determinado período; Tempo de Processador, que é basicamente o tempo que um processo leva dentro do seu período de execução; Tempo de Espera, que é após o processo ficar pronto, quanto tempo ele leva dentro da fila enquanto espera para ser executado; Tempo de Turnaround, que é o tempo completo, desde a hora que foi criado até ser completamente executado, levando em conta o tempo de espera para ser alocado na memória, o tempo de espera após pronto, processamento, fila de espera; Tempo de Resposta, é o tempo em que solicita algo pro sistema e espera pela resposta dele.

5. Explique o algoritmo de escalonamento First-In-First-Out (FIFO).

Esse tipo de escalonamento é como uma fila, em que o primeiro a ficar pronto, já entra para ser executado no processador e assim vai sendo por ordem de chegada, então quando sai algum para o estado de espera, um pronto já é escalonado e todos que estiverem na espera, saem para ir para o final da fila dos processos que estão prontos. O problema que tem nesse escalonamento, é que não dá para saber quando um processo vai ser iniciado, já que depende do tempo dos outros processos que também estão na fila de pronto. Esse é do tipo não preemptivo e hoje em dia só é implementado de forma parcial.

6. Explique o algoritmo de escalonamento Shortest-Job-First (SJF).

Esse escalonamento escolhe o processo que estiver no estado de pronto com menor tempo para que seja executado primeiro e como não é possível ver o tempo preciso de cada processo, é feita uma análise para verificar a média daquele processo em outras execuções que já tinham sido realizadas. Esse é do tipo não preemptivo e a diferença que tem do FIFO, é a média do tempo de espera entre os processos que é reduzida e também pode acontecer de ter starvation para processos que são muito longos ou que são do tipo CPU-bound.

7. Explique o que é Starvation.

Basicamente, é quando um processo tem muita dificuldade para ser executado, pois sempre entra algum para que seja executado primeiro por ter maior prioridade, um exemplo é no escalonamento SJF, porque escolhe o processo com menor tempo sempre, então se tiver um processo muito longo e sempre entra um menor que ele para que seja executado, vai haver dificuldade para que ele consiga um tempo de processamento.

8. Explique o algoritmo de Escalonamento Shortest Remaining Time.

É uma implementação do escalonamento Shortest-Job-First, só que com preempção, ou seja, quando algum processo está em período de execução, o sistema faz uma preempção caso ocorra de aparecer um processo com o tempo que tiver menor execução do que aquele que está em execução.

9. Explique o algoritmo de Escalonamento Cooperativo.

O cooperativo é quando um processo que está em execução, de maneira voluntária, ele sai e deixa entrar um que esteja na fila de pronto e para isso, ele lê uma fila de mensagens para saber se tem processos na fila de pronto. Um problema que pode ocorrer é se o processo não ler essas mensagens e acabar deixando vários processos na fila de pronto para ser executado na cpu, já que não vai ser responsabilidade do sistema operacional verificar isso.

10. Explique o algoritmo de Escalonamento Circular (Round-Robin Scheduling).

É parecido com o FIFO, porém o sistema operacional define um tempo para cada processo ser executado. Quando sai da fila de pronto e vai para a execução, se o tempo que foi definido expirar, o sistema interrompe, direciona para a fila de pronto e substitui por outro processo, sendo chamado de preempção por tempo. O sistema vai escolher, assim como escolhe no FIFO, o próximo da fila de pronto. Se o valor desse tempo for muito alto, vai se comportar igual o FIFO e se for pequeno, vai haver várias preempções e acabar afetando o sistema operacional e o tempo completo que o processo utiliza para ser executado. Esse escalonamento é bom pelo fato de não deixar processo tomar todo o tempo da UCP, porém os que são CPU-bound tem mais vantagem dos que são I/O-bound, já que CPU-bound consegue utilizar totalmente esse tempo determinado pelo sistema, enquanto os I/O-bound já vão para o estado de espera antes mesmo de passar pela preempção por tempo.

11. Explique o algoritmo de Escalonamento Circular Virtual.

Quando um processo sai da fila de espera, ele vai para a fila de pronto auxiliar, e essa fila tem prioridade em relação a fila de pronto, então os que estão na fila de pronto são começam a ser executados quando a fila de pronto auxiliar fica vazia. Quando o processo sai da fila de pronto auxiliar para ser executado, o tempo estimado para ele é igual ao tempo do sistema operacional menos o tempo que ele utilizou quando foi escalonado a partir da fila de pronto.

12. Explique o algoritmo de Escalonamento por Prioridade.

Nesse escalonamento, o processo vai ter um valor chamado de prioridade de execução, então sempre o que tiver maior prioridade vai ser o escolhido para ser executado e os outros que são do mesmo valor, vão ser como no FIFO. Os que tiverem prioridade, não podem sofrer preempção por tempo, ou seja, não tem tempo definido, ele só sai pra lista de espera se for de forma voluntária ou quando aparecer um com prioridade maior, e então o sistema vai realizar a preempção, para isso, a rotina do sistema revisa os que estão na fila de pronto e verifica se tem algum com maior prioridade. Também pode ter de forma não preemptiva, em que por mais que apareça um com prioridade maior, ele só vai passar pra fila de pronto para ser o próximo e não barrar aquele que está em execução.

13. Explique o algoritmo de Escalonamento Circular com Prioridade.

Esse escalonamento vai ter a junção de dois escalonamentos, com as definições de fatia de tempo e de prioridade de execução. O processo vai ficar sendo executado e só vai pra lista de espera se for voluntariamente ou se acontecer uma preempção, de tempo ou de prioridade. Esse tipo pode ter duas variações, circular com prioridade estática, em que a prioridade vai ser sempre a mesma pra cada processo e tem a circular com prioridade dinâmica, que a prioridade muda ou por quem administra o sistema ou pelo sistema operacional mesmo.

14. Explique o algoritmo de Escalonamento por Múltiplas Filas.

Esse escalonamento permite que tenha várias filas com processos no estado de pronto. Cada fila tem a possibilidade de ter seu tipo de escalonamento, como algo personalizado, mesmo sendo em um único sistema operacional, então pode ser que uma seja do tipo FIFO e outra do tipo circular. O processo que estiver em execução só sofre preempção, caso algum outro processo entre em uma fila de prioridade, já que a prioridade está associada às filas. O sistema operacional só pode escalonar quando todos os processos de filas com maior prioridade forem executados. Os processos nesse escalonamento são alocados em suas respectivas filas até o final da execução, não podendo mudar para uma fila mais adequada.

15. Explique o algoritmo de Escalonamento por Múltiplas Filas com Realimentação.

A diferença desse tipo para o Múltiplas filas, é que dá pra mudar os processos de fila durante seu processamento, fazendo com que o processo vá para filas com prioridade de execução e escalonamento mais apropriado durante sua execução. O FIFO é instalado com fatia de tempo em todas as filas e na de menor prioridade, é instalado o Circular. Nesse também só ocorre o processamento de uma fila, quando todas que têm maior prioridade forem executadas e sua fatia de tempo é de acordo com sua prioridade. Após o processo ser criado, ele vai pra fila de pronto com maior prioridade e quando estiver sendo executado, a cada preempção por tempo que sofrer, ele vai mudando de posição para a fila com menor prioridade.

16. Explique o algoritmo de Escalonamento em Sistemas de Tempo Compartilhado.

Esses sistemas usam o escalonamento circular com prioridade dinâmica, ou seja, ele compartilha os recursos de forma igual para que a UCP tenha um uso mais equilibrado entre a execução dos processos, assim o tempo que vai haver o processamento pode variar sem afetar algo que esteja sendo executado.

17. Explique o algoritmo de Escalonamento em Sistemas de Tempo Real.

Nesse sistema, os processos seguem um limite de tempo, para que não atrapalhe a execução. o escalonamento mais apropriado é o por prioridade, não tem fatia de tempo e cada prioridade deve ser do tipo estática.