Capítulo 9.

1 - As filas de mensagens no UNIX são geralmente classificadas como um mecanismo de comunicação de tipo 1 (comunicação de mensagem). Elas permitem a troca de mensagens entre processos por meio de uma fila de mensagens compartilhada.

2 - Os pipes UNIX são normalmente classificados como um mecanismo de comunicação de tipo 1 (comunicação de mensagem). Eles são usados para estabelecer um canal unidirecional para a comunicação entre processos.

3 - As áreas de memória compartilhada são geralmente classificadas como um mecanismo de comunicação de tipo 2 (compartilhamento de memória). Elas permitem que vários processos acessem uma região de memória compartilhada para compartilhar dados.

3 –

(a) Incorreta: As filas de mensagens POSIX têm uma capacidade máxima definida quando são criadas. Portanto, elas não têm capacidade nula.

(b) Correta

(c) Incorreta: A memória compartilhada é geralmente usada para comunicação entre processos em um sistema local, não necessariamente para comunicação em rede.

(d) Correta

(e) Incorreta: Os pipes Unix são um exemplo de comunicação 1:1, onde um processo escreve dados em um pipe e outro processo lê esses dados.

(f) Correta.

(g) Incorreta: As filas de mensagens POSIX são mais comumente usadas para a comunicação de mensagens, não para eventos.

(h) Incorreta: As filas de mensagens POSIX não utilizam arquivos em disco para a passagem de mensagens. Elas operam na memória do sistema.

(i) Correta

Capítulo 10.

1 - Condições de disputa ocorrem quando múltiplos processos tentam acessar recursos compartilhados sem coordenação, levando a resultados imprevisíveis.

2 –

1. Correta
2. Correta
3. Correta
4. Incorreta: Condições de disputa não ocorrem devido a diferenças de velocidade na execução dos processos.
5. Correta
6. Correta
7. Incorreta: O algoritmo de Peterson não garante justiça no acesso à região crítica.
8. Incorreta: Algoritmos com espera ocupada podem não otimizar o uso da CPU.
9. correta

3 - Espera ocupada é quando um processo ativamente verifica uma condição em vez de liberar a CPU; é ineficiente devido ao uso intensivo da CPU.

Capítulo 14

1 -

Os endereços lógicos são os endereços de memória usados pelos processos e pelo processador durante a execução, sendo definidos de acordo com o endereçamento do processado. Já os endereços físicos são aqueles que os endereços dos bytes de memória física do computador, sendo definidos pela quantidade de memória disponível na máquina.

2 -

A MMU é o dispositivo de hardware com capacidade de traduzir endereços virtuais em endereços físicos, normalmente é implementada como parte do CPU ou como um circuito integrado separado.

4 -

Alocação por partições:

* + Memória física dividida em partições de tamanhos fixos ou variáveis.
  + Uso de registradores de base e limite para tradução de endereços.
  + Simplicidade e velocidade, porém, sujeita à fragmentação externa.

Alocação por segmentos:

* + Espaço lógico dos processos dividido em segmentos de tamanhos variáveis.
  + Utilização de tabelas de descritores de segmentos para tradução de endereços.
  + Flexibilidade e proteção entre segmentos, mas também sujeita à fragmentação externa.

Alocação por páginas:

* + Espaço lógico dos processos dividido em páginas de tamanho fixo.
  + Mapeamento em quadros de memória de tamanho equivalente na memória física.
  + Tradução de endereços realizada com tabelas de páginas, que podem ter vários níveis.
  + Eliminação da fragmentação externa e suporte à memória virtual, porém com maior complexidade na gestão da memória.

Capítulo 15

**1 -**

O espaço de memória de um processo é organizado em seções ou segmentos, que são intervalos de endereços onde o processador tem permissão para acessar. Sendo essas as principais:

Text: onde fica contido o código binário a ser executado pelo processo, sendo gerado durante a compilação.

Data: contém as variáveis estáticas inicializadas, ou seja, variáveis que estão definidas do início ao fim da execução.

BSS: contém as variáveis estáticas não inicializadas.

HEAP: usada para armazenar variáveis alocadas dinamicamente, por meio de operadores como malloc(), new(), etc.

STACK: é utilizado para manter a pilha de execução do processo, sendo a estrutura responsável por gerenciar o fluxo de execução nas chamadas de funções, bem como armazenar os parâmetros, variáveis locais e o valor de retorno das funções.

Capítulo 20

1 -

Nível de Hardware:

Geração da Interrupção: As interrupções de hardware são geralmente geradas por dispositivos periféricos, como teclados, discos rígidos, ou controladores de dispositivos. Quando um desses dispositivos requer atenção, ele envia um sinal de interrupção ao processador.

Priorização de Interrupções: O sistema operacional precisa lidar com várias fontes de interrupção. Cada interrupção tem uma prioridade associada, e o hardware do sistema operacional deve ser capaz de identificar qual interrupção tem a prioridade mais alta no momento.

Salvamento do Estado Atual: Antes de iniciar o tratamento da interrupção, o estado atual da CPU, incluindo registros e instruções em execução, é geralmente salvo. Isso é crucial para que o sistema possa retornar ao ponto em que foi interrompido após lidar com a interrupção.

Chamada do Rotineiro de Tratamento: O sistema operacional possui rotineiros específicos (rotinas de tratamento de interrupção) para lidar com diferentes tipos de interrupções. A interrupção leva a execução para o endereço dessa rotina específica, onde a operação associada à interrupção é realizada.

Restauração do Estado: Após o tratamento da interrupção, o estado da CPU é restaurado para o ponto em que estava antes da interrupção, permitindo que a execução do programa seja retomada normalmente.

Nível de Software:

Identificação e Roteamento: No nível de software, o sistema operacional deve identificar a causa da interrupção e roteá-la para a rotina de tratamento apropriada. Isso envolve a consulta de tabelas de vetores de interrupção ou estruturas similares.

Tratamento Específico: Cada tipo de interrupção pode exigir um tratamento específico. Por exemplo, uma interrupção de disco pode envolver a transferência de dados entre a CPU e o disco, enquanto uma interrupção de temporizador pode estar relacionada ao agendamento de processos.

Coordenação com o Nível de Hardware: O nível de software interage com o nível de hardware para garantir uma resposta eficiente e adequada às interrupções. Isso pode incluir a configuração de controladores de dispositivos e a comunicação com o hardware para gerenciar interrupções.