

1. O alocador Buddy é usado no núcleo Linux para a alocação de memória física (page frames) e também para a alocação de memória para o alocador de objetos do núcleo e para outros subsistemas.

Verdadeiro Falso

2. A Unidade de Gerenciamento de Memória (MMU) é um componente crucial do sistema operacional, responsável por traduzir endereços lógicos gerados por programas em endereços físicos na memória RAM. Sobre a MMU, qual alternativa está **CORRETA**?

A MMU é um software executado pelo processador, independente do hardware.

A MMU traduz endereços físicos em endereços lógicos, permitindo que programas acessem a memória virtual.

A MMU é responsável apenas pela paginação da memória, não realizando a tradução de endereços.

A MMU garante que cada processo tenha acesso exclusivo a toda a memória RAM do computador.

A MMU não gera interrupções de hardware em caso de tentativas de acesso indevido à memória.

3. O que é um TLB (Translation Lookaside Buffer) e qual é a sua função principal no contexto de um sistema computacional? Opções:

O TLB é um tipo de memória secundária que armazena temporariamente os dados usados pelos aplicativos para aumentar a velocidade do processador.

O TLB é um componente especializado de hardware dentro da MMU (Unidade de Gerência de Memória) que armazena resultados de traduções de endereços recentes para acelerar o processo de conversão de endereços virtuais em físicos.

O TLB é uma ferramenta de software que gerencia as conexões de rede para otimizar o desempenho do sistema operacional.

O TLB é um protocolo de segurança que verifica a integridade dos dados armazenados na memória antes de serem processados pelo CPU.

4. Sobre as afirmações a seguir, relativas ao uso da memória RAM pelos processos, indique quais são **incorretas**:

A área de memória TEXT contém o código-fonte a ser compilado e executado pelo processo. A área de memória TEXT contém o código binário do programa, gerado após a compilação e ligação do código-fonte. O código-fonte é a representação textual do programa, enquanto o código binário é a forma como o processador o entende.

A área de memória DATA é usada para armazenar todas as variáveis e constantes usadas pelo processo. A área de memória DATA armazena apenas as variáveis estáticas inicializadas, ou seja, aquelas que têm seus valores definidos no código-fonte. As constantes são geralmente armazenadas em uma seção separada, e as variáveis não-inicializadas ficam na seção BSS.

A área de memória HEAP é usada para as alocações dinâmicas de memória, sendo usada através de funções como malloc e free. A área de memória HEAP é a região de memória utilizada para alocar memória dinamicamente durante a execução do programa, utilizando funções como malloc() e free() em linguagens como C e C++.

A área de memória STACK contém as pilhas do programa principal e das demais threads do processo. A área de memória STACK contém a pilha do programa principal, enquanto cada

thread do processo possui sua própria pilha de execução, que geralmente é alocada dinamicamente no heap ou em uma área específica para pilhas de threads

5. Uma falta de página (page fault) ocorre quando um processo tenta acessar uma página que não está presente na memória RAM. Qual das alternativas abaixo NÃO descreve uma etapa do tratamento de uma falta de página pelo sistema operacional?

Verificar se a página é válida: Se a página é inválida, o processo tentou acessar um endereço inválido e deve ser abortado.

Carregar a página na memória: Se a página é válida, o sistema operacional carrega a página da área de swap para a memória RAM e atualiza a tabela de páginas.

Retomar a execução do processo: Após a página ser carregada, o processo é retomado e a instrução que causou a falta de página é reexecutada.

Escolher uma página "vítima" para ser removida da memória: Se a memória RAM estiver cheia, o sistema operacional precisa remover uma página existente para liberar espaço para a nova página.

Reiniciar o processo: Caso a página não seja encontrada na área de swap, o sistema operacional reinicia o processo para garantir a integridade dos dados.

6. O uso de endereços lógicos na memória virtual é justificado pela necessidade de simplificar o gerenciamento de memória e oferecer proteção aos processos, pois permite que o sistema operacional aloque e desaloque a memória física de forma mais eficiente, sem que os processos precisem se preocupar com os endereços físicos reais.

Verdadeiro Falso

7. Sobre a Unidade de Gerenciamento de Memória (MMU) , é CORRETO afirmar:

A MMU é um software executado pelo processador, independente do hardware.

A MMU traduz endereços físicos em endereços lógicos, permitindo que programas acessem a memória virtual.

A MMU é responsável apenas pela paginação da memória, não realizando a tradução de endereços.

A MMU garante que cada processo tenha acesso exclusivo a toda a memória RAM do computador.

A MMU não gera interrupções de hardware em caso de tentativas de acesso indevido à memória.

8. Considere os seguintes tipos de alocação de variáveis em programação. Assinale a alternativa que relaciona CORRETAMENTE cada tipo de alocação de variáveis com sua respectiva característica:

I. Alocação Estática: O espaço para a variável é definido em tempo de compilação e a memória é alocada no início da execução do processo.

II. Alocação Automática: O espaço para a variável é alocado na pilha de execução quando a função é chamada e liberado quando a função termina.

III. Alocação Dinâmica: O espaço para a variável é requisitado explicitamente em tempo de execução através de funções como ``malloc()`` ou ``new()``.

I - Variáveis globais e locais; II - Variáveis locais e parâmetros de função; III - Variáveis que podem ter seu tamanho alterado durante a execução.

I - Variáveis que podem ter seu tamanho alterado durante a execução; II - Variáveis globais e locais; III - Variáveis locais e parâmetros de função.

I - Variáveis globais e locais; II - Variáveis que podem ter seu tamanho alterado durante a execução; III - Variáveis locais e parâmetros de função.

I - Variáveis que podem ter seu tamanho alterado durante a execução; II - Variáveis locais e parâmetros de função; III - Variáveis globais e locais

I - Variáveis locais e parâmetros de função; II - Variáveis que podem ter seu tamanho alterado durante a execução; III - Variáveis globais e locais.

10. O texto a seguir descreve duas técnicas de gerenciamento de memória, A e B. Identifique qual técnica cada descrição representa, assinale a alternativa que identifica corretamente qual técnica é descrita por cada letra:

A: Move um processo inteiro para o disco, liberando a memória RAM. Essa técnica é simples e eficiente para sistemas com pouca memória RAM, mas pode ser lenta para processos grandes.

B: Move páginas individuais para o disco, permitindo que vários processos compartilhem a memória RAM. Essa técnica é mais flexível e eficiente, pois não exige mover o processo inteiro. É ideal para sistemas com pouca memória, pois permite compartilhar a memória disponível entre vários processos.

A descreve Swapping; B descreve Paging

A descreve Paging; B descreve Overlays

A descreve Overlays; B descreve Swapping

A descreve Paging; B descreve Swapping

Nenhuma alternativa está correta

11. Com base nas afirmações a seguir sobre Endereços Físicos e Lógicos, assinale as alternativas corretas:

A) Endereços físicos são sempre menores ou iguais aos endereços lógicos.

B) O espaço de endereçamento lógico de um processo é independente da quantidade de memória física disponível.

C) A MMU (Unidade de Gerência de Memória) é responsável por traduzir endereços lógicos em endereços físicos.

D) O sistema operacional não tem acesso aos endereços físicos da memória.

E) O tamanho do espaço de endereçamento lógico é determinado pelo tamanho da memória física.

12. Considere as afirmações sobre fragmentação de memória em sistemas operacionais, assinale a alternativa que contém as afirmações CORRETAS:

A: A fragmentação externa ocorre quando a memória livre está fragmentada em pequenos blocos não contíguos, dificultando a alocação de blocos maiores.

B: A fragmentação interna ocorre quando um bloco de memória alocado é maior que o tamanho realmente necessário, resultando em espaço desperdiçado.

C: A fragmentação interna é um problema exclusivo da alocação de memória paginada, pois a alocação é feita em múltiplos inteiros de páginas.

D: A fragmentação externa é um problema que pode ser completamente eliminado através de algoritmos de alocação eficientes.

13. O ato de suspender uma tarefa e reativar outra é denominado uma troca de contexto. Nela o sistema operacional pode suspender e retomar a execução de tarefas de

forma transparente (sem que as tarefas o percebam), é necessário definir operações para salvar o contexto atual de uma tarefa em seu TCB e restaurá-lo mais tarde no processador.

Verdadeiro Falso

14. Endereços físicos (ou reais) são os endereços dos bytes de memória física do computador. Estes endereços são definidos pela quantidade de memória disponível na máquina

Verdadeiro Falso

15. A divisão de espaço de kernel vs espaço de usuário é:

A - Linux de 32 bits: Espaço de usuário 3 GB, espaço de kernel 1 GB.

B - Windows de 32 bits: Espaço de usuário 2 GB, espaço de kernel 2 GB.

C - Linux de 64 bits: Espaço de usuário 128 TB, espaço de kernel 128 TB.

D - Windows de 64 bits: Espaço de usuário 8TB, espaço de kernel 8TB.

16. Endereços lógicos (ou virtuais) são os endereços de memória usados pelos processos e pelo sistema operacional e, portanto, usados pelo processador durante a execução. Estes endereços são definidos de acordo com o espaço de endereçamento do processador.

Verdadeiro Falso

17. Por que é necessário limpar o cache TLB (Translation Lookaside Buffer) após cada troca de contexto entre processos, mas não é necessário nas trocas de contexto entre threads?

Cada processo possui seu próprio espaço de endereçamento virtual, e o TLB armazena informações específicas para o processo ativo, portanto, precisa ser limpo para evitar conflitos de tradução de endereços.

Threads compartilham o mesmo espaço de endereçamento virtual e, portanto, o TLB pode ser reutilizado entre elas sem necessidade de limpeza.

O TLB é um recurso compartilhado entre todos os processos do sistema, e a limpeza é necessária para evitar que um processo acesse informações de outro.

A limpeza do TLB é um procedimento padrão em qualquer troca de contexto, independentemente de ser entre processos ou threads.

O TLB é um recurso do hardware e não é afetado pelas trocas de contexto entre processos ou threads.

A necessidade de limpar o TLB em trocas de contexto entre processos se deve à diferença no espaço de endereçamento virtual entre os processos. Já em trocas de contexto entre threads, o espaço de endereçamento é compartilhado, portanto, a limpeza do TLB não é necessária.

18. Qual das seguintes razões NÃO justifica a inviabilidade de implementar as conversões de endereços da MMU em software, ao invés de hardware dedicado?

O desempenho do sistema seria significativamente prejudicado, pois a conversão de endereços em software seria um processo lento.

O desenvolvimento e a manutenção do software para a conversão de endereços seriam tarefas complexas e custosas.

A implementação em software tornaria o sistema mais suscetível a falhas de hardware, como erros na memória RAM.

A implementação em software demandaria um processador mais potente para lidar com a complexidade e o tempo de execução do software.

