

LCMAT – Laboratório de Ciências Matemáticas

Curso de Bacharelado em Ciência da Computação

Data: 27/ 09/2023

Nota: ____/1,5

Professor: Eduardo Carvalho

Grupo: 1

Integrantes: Gabriel Almeida, Enzo Marques, Gabriel Costa, Enzo Alberoni, Mariana Cossetti, Daniel Gomes.

1) Vale 0,1. Quais são as principais funções de um sistema operacional?

Os sistemas operacionais realizam duas funções essencialmente não relacionadas: fornecer a programadores de aplicativos um conjunto de recursos abstratos limpo em vez de recursos confusos de hardware, e gerenciar esses recursos de hardware. A principal função do sistema operacional é manter um controle sobre quais programas estão usando qual recurso, conceder recursos requisitados, contabilizar o seu uso, assim como mediar requisições conflitantes de diferentes programas e usuários.

2) Vale 0,1. Quais são os modos de operação de um sistema operacional. Qual é a diferença entre eles?

Existem dois possíveis: Modo usuário e modo kernel: No Modo Usuário, aplicativos de usuário são executados com restrições de acesso aos recursos do sistema, enquanto no Modo Kernel, o sistema operacional detém altos privilégios, permitindo ter controle absoluto sobre o hardware e os recursos críticos.

3) Vale 0,1. O que é uma instrução trap? Quando a execução de um processo é retomada, ela deve continuar de qual ponto?

É um mecanismo fundamental em sistemas operacionais e arquiteturas de computadores que permite que o sistema operacional interrompa a execução normal de um programa ou processo em resposta a eventos excepcionais ou situações de erro. Após a retomada de execução de um processo ela retorna ao ponto em que se encontrava antes.

4) Vale 0,1. Qual é a estrutura de dados utilizada pelo conjunto dos processos num sistema operacional?

Uma das principais estruturas de dados utilizada pelo conjunto dos processos num sistema operacional é a Tabela de Processos.

5) Vale 0,1. O quê ocorre quando o espaço de endereçamento dos processos excede a capacidade de processamento da memória principal?

Quando o espaço de endereçamento dos processos excede a capacidade da memória principal, podem ocorrer várias situações. Em sistemas modernos, a técnica de memória virtual é comumente usada para lidar com essa limitação. A memória virtual envolve a alocação de parte do espaço de endereçamento de um processo na memória principal e parte no disco mantendo os processos em execução, embora com possíveis penalidades de desempenho devido à necessidade de acesso à memória no disco.

6) Vale 0,1. Explique o conceito de multiplexação, seus diferentes tipos e exemplo de utilização?

A multiplexação é o conceito de compartilhar recursos entre vários processos ou usuários para melhorar a eficiência e o uso dos recursos de um sistema. Existem dois tipos principais de multiplexação: multiplexação no tempo e multiplexação no espaço. Na multiplexação no tempo, os recursos são compartilhados em intervalos de tempo. Isso significa que cada processo ou usuário recebe acesso aos recursos em turnos, durante um período específico. Um exemplo clássico é a alocação de CPU em sistemas de multiprogramação. Vários processos estão prontos para serem executados, mas a CPU é compartilhada entre eles em pequenos intervalos de tempo. Na multiplexação no espaço, os recursos são divididos fisicamente e cada processo ou usuário recebe uma parte separada dos recursos. Um exemplo comum é a alocação de memória em sistemas de multiprogramação. A memória física é dividida entre os processos em execução, e cada processo possui seu próprio espaço de endereçamento na memória, evitando que eles interfiram uns nos outros.

7) Vale 0,1. O quê ocorre no sistema de arquivos ao inserir um pendrive na entrada USB?

Identificação do Hardware: O sistema operacional detecta que um novo dispositivo USB foi conectado e verifica suas características.

Associação de Drivers: O sistema operacional pode automaticamente carregar os drivers necessários para o dispositivo, a fim de garantir que ele seja reconhecido e funcione corretamente. Esses drivers são pequenos programas que permitem que o sistema operacional se comunique com o hardware específico do dispositivo.

Montagem do Dispositivo: O sistema operacional monta o dispositivo de armazenamento, tornando-o acessível ao sistema de arquivos do computador. Isso permite que você veja o conteúdo do pendrive e interaja com seus arquivos.

Atribuição de uma Letra ou Caminho: Em sistemas operacionais Windows, é comum que o pendrive seja atribuído a uma letra de unidade (por exemplo, "D:", "E:") para que você possa acessá-lo através do Explorador de Arquivos. Em sistemas operacionais baseados em Unix/Linux, o dispositivo pode ser montado em um ponto de montagem específico no sistema de arquivos.

Acesso aos Arquivos: Uma vez montado, você pode acessar, copiar, mover e gerenciar os arquivos no pendrive como faria com qualquer pasta ou unidade no seu computador.

Remoção Segura: É importante usar a opção de "Remoção Segura" (ou equivalente) no sistema operacional antes de desconectar o pendrive. Isso garante que todos os dados sejam gravados e que não ocorram danos ao sistema de arquivos durante a remoção.

8) Vale 0,1. O quê são os drivers e com o quê eles se comunicam?

Os drivers são programas de software que funcionam como intermediários entre o sistema operacional de um computador e os dispositivos de hardware conectados a ele. Eles são essenciais para permitir que o sistema operacional e os aplicativos interajam com e controlem o hardware de forma adequada. Os drivers atuam como tradutores, convertendo comandos e solicitações do software em instruções compreensíveis pelo hardware específico. Isso permite que os dispositivos funcionem corretamente e sejam reconhecidos pelo sistema operacional. Cada dispositivo de hardware, como impressoras, placas de vídeo, discos rígidos e periféricos, geralmente requer um driver específico para funcionar corretamente em um sistema.

Os drivers se comunicam diretamente com o hardware dos dispositivos. Eles traduzem as instruções fornecidas pelo sistema operacional e pelos aplicativos em comandos compreensíveis pelo hardware específico. Essa comunicação permite que o software controle e interaja com o hardware de maneira eficaz, garantindo que os dispositivos funcionem corretamente e que as informações sejam trocadas de maneira adequada entre o software e o hardware.

9) Vale 0,1. Descreva um conteúdo na hierarquia de sistemas de arquivos UNIX-like protegido com -rwxrw----.

Um conteúdo em um sistema de arquivos UNIX-like protegido com permissões "rwxrw----" significa que esse conteúdo possui permissões específicas que determinam quem pode acessá-lo e de que maneira.

Exemplo: Suponha que você tenha um arquivo chamado "documento-secreto.txt" com permissões "rwxrw----".

O usuário "Alberoni" é o proprietário do arquivo "documento-secreto.txt". Portanto, Alice pode ler(r), escrever(w) e executar(x) o arquivo.

O arquivo pertence a um grupo chamado "Equipe". Os membros deste grupo, como "Mariana" e "Costa", podem ler(r) e escrever(w) no arquivo, mas não podem executá-lo.

Todos os outros usuários, como "Almeida" e "Enzo", não têm nenhuma permissão para acessar o arquivo. Eles não podem lê-lo, escrever nele ou executá-lo.

10) Vale 0,1. O que é POSIX?

Portable Operating System Interface — interface portátil para sistemas operacionais. O POSIX define uma interface minimalista de chamadas de sistema à qual os sistemas UNIX em conformidade devem dar suporte. Na realidade, alguns outros sistemas operacionais também dão suporte hoje em dia à interface POSIX.

11) Vale 0,15. Como ocorre a criação de um processo, num sistema operacional UNIX-like? Quais são os segmentos de um processo?

A chamada fork é a única maneira para se criar um processo novo em POSIX. Ela cria uma cópia exata do processo original, incluindo todos os descritores de arquivos, registradores - tudo. Processos em UNIX têm sua memória dividida em três segmentos: o segmento de texto (isto é, código de programa), o segmento de dados (isto é, as variáveis) e o segmento de pilha.

12) Vale 0,15. O processo filho é idêntico ao pai? Explique.

Após a criação do processo pelo fork, o processo original e a cópia (o processo pai e o processo filho) seguem seus próprios caminhos separados. Todas as variáveis têm valores idênticos no momento da fork, mas como os dados do

processo pai são copiados para criar o processo filho, mudanças subsequentes em um deles não afetam o outro. Por exemplo, o processo pai pode continuar executando seu código original, enquanto o processo filho pode executar um código diferente usando a função "exec". Portanto, embora o processo filho seja inicialmente idêntico ao processo pai, eles podem divergir em suas execuções posteriores. Cada processo tem sua própria cópia dos recursos do sistema e é executado de forma independente.

13) Vale 0,2. Dê um exemplo e explique a estrutura de sistema operacional que apresenta uma alta coesão e um baixo acoplamento. Compare-o com uma estrutura monolítica.

Microkernel é uma arquitetura de sistema operacional que exemplifica esses conceitos.

Alta Coesão: O microkernel é projetado para ser pequeno e conter apenas as funcionalidades essenciais, como gerenciamento de processos, comunicação entre processos e gerenciamento de memória. Cada um desses componentes é altamente coeso, ou seja, eles se concentram em uma única tarefa e são independentes um do outro.

Baixo Acoplamento: Devido à abordagem modular do microkernel, os componentes podem ser desenvolvidos, modificados e substituídos independentemente uns dos outros. Isso significa que, por exemplo, você pode atualizar o gerenciamento de processos sem afetar o gerenciamento de memória e vice-versa. Isso reduz o acoplamento entre os componentes e torna o sistema mais flexível.

Em contraste, uma estrutura monolítica, como o **kernel monolítico**, possui todas as funcionalidades do sistema operacional integradas em um único componente. Isso resulta em uma coesão menor, porque várias responsabilidades estão agrupadas em um único lugar, tornando o componente menos focado. Além disso, o acoplamento é geralmente maior em sistemas monolíticos, pois as partes do kernel dependem umas das outras de maneira mais estreita. Isso torna os sistemas monolíticos mais difíceis de manter e modificar, especialmente quando se deseja adicionar ou remover funcionalidades.

