

Disciplina: Sistemas Operacionais

Atividade: 3 questionario

Aluna: Vitor Hugo Klein

1.O que diferencia o núcleo do restante do sistema operacional?

O núcleo é o coração do sistema operacional, responsável pela gerência dos recursos do hardware, como CPU, memória e dispositivos de entrada/saída. Ele opera em um nível de privilégio mais alto, permitindo acesso direto ao hardware e controle sobre as operações críticas do sistema. Em contraste, o restante do sistema operacional (como bibliotecas e interfaces de usuário) opera em um nível de privilégio mais baixo e interage com o núcleo através de chamadas de sistema.

2. Seria possível construir um sistema operacional seguro usando um processador que não tenha níveis de privilégio? Por quê?

Não, pois um sistema operacional seguro depende de níveis de privilégio para proteger o núcleo e os recursos críticos do sistema de acessos não autorizados. Sem os níveis de privilégio, qualquer programa em execução teria acesso total ao hardware e à memória, tornando impossível implementar controles de segurança eficazes.

3. Os processadores da família x86 possuem dois bits para definir o nível de privilégio, resultando em 4 níveis distintos. A maioria dos sistemas operacionais para esses processadores usam somente os níveis extremos (0 e 3). Haveria alguma utilidade para os níveis intermediários?

Os níveis intermediários podem ser úteis para implementar características de segurança e modularidade. Por exemplo, um sistema pode utilizar um nível intermediário para fornecer serviços a um aplicativo que precisa de mais permissões do que um usuário normal, mas menos que o núcleo. Isso pode ajudar a limitar o acesso a recursos sensíveis enquanto permite uma maior funcionalidade para certos aplicativos.

4. Quais as diferenças entre interrupções, exceções e traps?

Interrupções: Sinais gerados por dispositivos de hardware que indicam que um evento externo ocorreu e que o processador deve interromper sua execução atual para lidar com o evento (por exemplo, I/O).

Exceções: Erros ou eventos que ocorrem durante a execução de um programa, como divisão por zero ou acesso a memória inválida, que resultam na interrupção do fluxo normal de execução.

Traps: Interrupções geradas por instruções específicas no programa, geralmente para solicitar serviços do núcleo (como chamadas de sistema).

5. O comando em linguagem C fopen é uma chamada de sistema ou uma função de biblioteca? Por quê?

Fopen é uma função de biblioteca que faz uma chamada de sistema para abrir um arquivo. Ele fornece uma interface mais conveniente e abstraída para o programador, mas internamente, ele invoca uma chamada de sistema para realizar a operação real de abertura do arquivo.

6. A operação em modo usuário permite ao processador executar somente parte das instruções disponíveis em seu conjunto de instruções. Quais das seguintes operações não deveriam ser permitidas em nível usuário? Por quê?

- (a) Ler uma porta de entrada/saída:** Essa operação deve ser restrita porque envolve acesso direto ao hardware, o que pode causar instabilidade no sistema.
- (b) Efetuar uma divisão inteira:** Essa operação pode ser permitida, pois não envolve acesso ao hardware ou a recursos críticos.
- (c) Escrever um valor em uma posição de memória:** Não deve ser permitido, pois pode causar corrupção de dados e violar a segurança da memória de outros processos.
- (d) Ajustar o valor do relógio do hardware:** Deve ser restrito, pois pode afetar todo o sistema e criar inconsistências.
- (e) Ler o valor dos registradores do processador:** Deve ser restrito, pois permite acesso a informações críticas que podem comprometer a segurança.
- (f) Mascaram uma ou mais interrupções:** Não deve ser permitido, pois pode desativar a resposta a eventos importantes e causar problemas de estabilidade.

7. Considerando um processo em um sistema operacional com proteção de memória entre o núcleo e as aplicações, indique quais das seguintes ações do processo teriam de ser realizadas através de chamadas de sistema, justificando suas respostas:

- (a) Ler o relógio de tempo real do hardware:** Chamada de sistema necessária para acessar o hardware diretamente.
- (b) Enviar um pacote através da rede:** Chamada de sistema necessária para interagir com a pilha de rede.
- (c) Calcular uma exponenciação:** Não é necessária uma chamada de sistema, pois é uma operação computacional que pode ser realizada em modo usuário.
- (d) Preencher uma área de memória do processo com zeros:** Não precisa de chamada de sistema, pois é uma operação válida em modo usuário.
- (e) Remover um arquivo do disco:** Chamada de sistema necessária para interagir com o sistema de arquivos.

8. Coloque na ordem correta as ações abaixo, que ocorrem durante a execução da função printf("Hello world") por um processo (observe que nem todas as Disciplina: Sistemas Operacionais Prof. Agnaldo, da Costa Ementa: Classificação dos Sistemas Operacionais, Estruturas de Sistemas Operacionais, Elementos de Hardware, Chamadas de Sistemas. ações indicadas fazem parte da sequência).

- (a) A rotina de tratamento da interrupção de software é ativada dentro do núcleo.**
- (b) A função printf finaliza sua execução e devolve o controle ao código do processo.**

- (c) A função de biblioteca printf recebe e processa os parâmetros de entrada (a string "Hello world").
- (d) A função de biblioteca printf prepara os registradores para solicitar a chamada de sistema write().
- (e) O disco rígido gera uma interrupção indicando a conclusão da operação.
- (f) O escalonador escolhe o processo mais prioritário para execução.
- (g) Uma interrupção de software é acionada.
- (h) O processo chama a função printf da biblioteca C.
- (i) A operação de escrita no terminal é efetuada ou agendada pela rotina de tratamento da interrupção.
- (j) O controle volta para a função printf em modo usuário.

Ordenação correta: **(h)** O processo chama a função printf da biblioteca C.

(c) A função de biblioteca printf recebe e processa os parâmetros de entrada (a string "Hello world").

(d) A função de biblioteca printf prepara os registradores para solicitar a chamada de sistema write().

(g) Uma interrupção de software é acionada.

(a) A rotina de tratamento da interrupção de software é ativada dentro do núcleo.

(i) A operação de escrita no terminal é efetuada ou agendada pela rotina de tratamento da interrupção.

(b) A função printf finaliza sua execução e devolve o controle ao código do processo.

9. Monte uma tabela com os benefícios e deficiências mais relevantes das principais arquiteturas de sistemas operacionais.

Arquitetura	Benefícios	Deficiências
Monolítica	Simplicidade de implementação e eficiência.	Difícil de manter e modificar; falhas podem travar o sistema todo
Microkernel	Maior modularidade e segurança.	Possível perda de desempenho devido a chamadas de sistema mais frequentes
Híbrida	Combina benefícios de microkernel e monolítico.	Complexidade na implementação e no gerenciamento de módulos

Em Camadas	Clareza na separação de funções.	Sobrecarga de comunicação entre camadas
------------	----------------------------------	---

10. Qual é a diferença entre sistemas de compartilhamento de tempo e de multiprogramação?

Compartilhamento de tempo: Permite que múltiplos usuários interajam com o sistema ao mesmo tempo, garantindo que cada um tenha uma resposta rápida e eficiente. Os processos recebem pequenas fatias de tempo do processador.

Multiprogramação: Foca na utilização eficiente do processador ao ter vários processos carregados na memória, mas não necessariamente permite que múltiplos usuários interajam simultaneamente. A troca de processos é feita com o objetivo de maximizar a utilização do CPU.

11. Explique como a separação da política e mecanismo ajuda na construção de sistemas operacionais baseados em micronúcleos.

A separação da política e mecanismo permite que a lógica de decisão (política) seja distinta das operações de implementação (mecanismo). Em micronúcleos, isso facilita a adaptação de diferentes políticas de gerenciamento de recursos sem a necessidade de reimplementar os mecanismos subjacentes, promovendo flexibilidade e extensibilidade no design do sistema operacional.

12. Para um programador, uma chamada de sistema parece com qualquer outra chamada para uma rotina de biblioteca. É importante que um programador saiba quais rotinas de biblioteca resultam em chamadas de sistema? Em quais circunstâncias e por quê?

Sim, é importante porque chamadas de sistema geralmente têm custos associados, como mudanças de contexto e maior latência. Um programador deve estar ciente de quais operações podem ser mais eficientes quando realizadas diretamente no código, evitando chamadas desnecessárias que possam impactar o desempenho do sistema.

13. Quais das instruções a seguir devem ser deixadas somente em modo núcleo?

(a) Desabilitar todas as interrupções: Deve ser restrito para evitar que processos em nível usuário desativem a resposta a eventos críticos.

(b) Ler o relógio da hora do dia: Não necessariamente deve ser restrito, pois é uma operação que pode ser permitida em modo usuário.

(c) Configurar o relógio da hora do dia: Deve ser restrito, pois altera o comportamento do sistema.

(d) Mudar o mapa de memória: Deve ser restrito, pois envolve a proteção da memória e pode comprometer a segurança e estabilidade do sistema.