

# L02\_estrutura de um SO

Total de pontos 18/18

Endereço de e-mail \*

bruno.oliveira.duarte.bd@gmail.com

✓ O comando em linguagem C fopen é uma chamada de sistema ou uma função de biblioteca? Por quê? \* 1/1

- ☐ Chamada de biblioteca, não existe chamada de sistema para abrir arquivo
- ☐ NDA
- ☐ Chamada de sistema pois abre um arquivo, ou seja, interfere em recursos do SO
- ☒ Chamada de biblioteca, que verificará os parâmetros da chamada. Em estando tudo correto ela invocará a chamada de sistema open() ✓
- ☐ Chamada de sistema pois toda função da biblioteca C é uma chamada de sistema
- ☐ Nenhuma das duas, é apenas uma função

## Feedback

*Pelo próprio manual do C dá para verificar se é uma chamada de sistema ou não. Compare os manuais*

<https://man7.org/linux/man-pages/man3/fopen.3.html>

<https://man7.org/linux/man-pages/man2/open.2.html>

*O open diz claramente, "the open() system call opens" e a fopen "the fopen() function opens"*



✓ 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: \*

2/2

- ☒ Ler o relógio de tempo real do hardware ✓
- ☒ Enviar um pacote através da rede ✓
- ☐ Preencher uma área de memória do processo com zeros
- ☐ Calcular uma exponenciação
- ☒ Remover um arquivo do disco ✓

✓ Quais as diferenças entre interrupções, exceções e traps? Marque as alternativa correta. \*

- ☒ Interrupções e exceções são geradas por hardware, traps são geradas por software ✓
- ☒ Traps funcionam como interrupções, porém servem para invocar serviços do núcleo. Já interrupções e exceções servem para informar ao processador a ocorrência de um evento, interno (próprio processador) ou externo (dispositivos) ✓
- ☐ Interrupções são geradas por hardware (exceto processador), exceções são geradas pelo próprio processador e trap é uma exceção que irá travar o processo
- ☐ Não há diferenças, pois todas possuem um IRQ
- ☐ Interrupções e exceções são eventos gerados pelo próprio processador



✓ Seria possível construir um sistema operacional seguro usando um processador que não tenha níveis de privilégio? Por quê? \* 1/1

- ☐ Sim, o MSDOS é um exemplo de sistema seguro mesmo utilizando apenas um nível de privilégio
- ☒ Não, pois qualquer usuário mal intencionado poderia interferir nas configurações e gerências ✓
- ☐ NDA
- ☐ Não, pois os sistemas precisam usar as chamadas de sistema
- ☐ Sim, bastaria armazenar os dados do usuário exclusivamente no disco rígido

✓ O que diferencia o núcleo do restante do sistema operacional? Marque a 1/1 resposta correta \*

- ☐ Somente o kernel roda sobre modo privilegiado
- ☐ Kernel cuida do carregamento do SO e gerência de recursos, enquanto os drivers fazem a interface com os dispositivos e os programas utilitários fazem a interface com os usuários
- ☒ Ele é responsável único pela gerência dos recursos ✓
- ☐ NDA
- ☐ Nenhuma, pois todos os módulos do SO rodam sobre o modo privilegiado



✓ 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, ou 002 e 112). Haveria alguma utilidade para os níveis intermediários? \*

1/1

- ☒ Uma utilidade bastante comum nos dias de hoje é utilizar os níveis intermediários para hospedar os sistemas virtualizados (e.g. VMware, docker) ✓
- ☐ Não, foi uma implementação desnecessária e descartada com o passar dos anos
- ☒ Sim, outros sistemas operacionais podem usar os 4 níveis. Depende do design escolhido ✓
- ☐ Sim, mas apenas o Multics utiliza tal abordagem
- ☐ NDA

✓ A operação em modo usuário permite ao processador executar somente 2/2 parte das instruções disponíveis em seu conjunto de instruções. Quais das seguintes operações não deveriam ser permitidas em nível usuário? \*

- ☒ Ajustar o valor do relógio do hardware ✓
- ☒ Mascarar uma ou mais interrupções ✓
- ☐ Ler o valor dos registradores do processador
- ☐ Escrever um valor em uma posição de memória
- ☐ Efetuar uma divisão inteira
- ☒ Ler uma porta de entrada/saída ✓



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 ações indicadas fazem parte da sequência, use NA para marcar tais ações). \*

|                                                                                                                              | 1                                | 2                                | 3                                | 4                                | 5                                | 6                                | 7                     | 8                     | NA                    |
|------------------------------------------------------------------------------------------------------------------------------|----------------------------------|----------------------------------|----------------------------------|----------------------------------|----------------------------------|----------------------------------|-----------------------|-----------------------|-----------------------|
| A função de biblioteca <code>printf</code> prepara os registradores para solicitar a chamada de sistema <code>write()</code> | <input type="radio"/>            | <input type="radio"/>            | <input checked="" type="radio"/> | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| O processo chama a função <code>printf</code> da biblioteca C                                                                | <input checked="" type="radio"/> | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| A função de biblioteca <code>printf</code> recebe e processa os parâmetros de entrada (a string "Hello world")               | <input type="radio"/>            | <input checked="" type="radio"/> | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| A rotina de tratamento da interrupção de software é ativada dentro do núcleo                                                 | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/>            | <input checked="" type="radio"/> | <input type="radio"/>            | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Uma interrupção de software é acionada                                                                                       | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/>            | <input checked="" type="radio"/> | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| A operação de escrita no terminal é                                                                                          | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/>            | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |



efetuada ou  
agendada  
pela rotina  
de  
tratamento  
da  
interrupção

O  
escalonador  
escolhe o  
processo  
mais  
prioritário  
para  
execução

☐☐☐☐☐☐☐☐☒

A função  
printf finaliza  
sua  
execução e  
devolve o  
controle ao  
código do  
processo

☐☐☐☐☐☐☐☒☐

O controle  
volta para a  
função printf  
em modo  
usuário

☐☐☐☐☐☐☒☐☐

O disco  
rígido gera  
uma  
interrupção  
indicando a  
conclusão  
da operação

☐☐☐☐☐☐☐☐☒

Este conteúdo não foi criado nem aprovado pelo Google. - [Termos de Serviço](#) - [Política de Privacidade](#)

Google Formulários

