Pró-Reitoria Acadêmica

Diretoria Acadêmica

Assessoria Pedagógica da Diretoria Acadêmica

**FACULDADE:** CENTRO UNIVERSITÁRIO DE BRASÍLIA – UniCEUB

**CURSO:** CIÊNCIA DA COMPUTAÇÃO

**DISCIPLINA**: SISTEMAS OPERACIONAIS

**CARGA HORÁRIA**: 60 H. A. **ANO/SEMESTRE**: 2020/02

**PROFESSOR**: EDUARDO FERREIRA DOS SANTOS

**HORÁRIOS: TURMA B SEGUNDA, TURMA A QUARTA.**

**LISTA DE EXERCÍCIOS 01**

|  |  |
| --- | --- |
|  | **EXERCÍCIO 01** |
| 1. | Como funciona a interface homem máquina em um sistema operacional?  A interface é ,basicamente ,tudo que vemos no painel da máquina, ela serve para tornar mais fácil(simplificado) a interação do usuário, ou seja, ela funciona através de pré-progamações feitas para auxiliar o usuário a interagir com a maquina. |
| 2. | O que são sistemas operacionais de kernel monolítico? Descreva a interação entre os programas hospedados pelo sistema operacional e o hardware para esse tipo de SO.  Se caracteriza pelo fato de que cada componente do SO pode ser continho no núcleo e interagir com outros componentes. No kernel monolítico o SO é escrito em forma de rotinas e cada rotina deve ter uma interface bem organizada, com parâmetros e resultados bem defindos, para que haja a intertaçao entre rotinas e para que o hardware consiga interagir com os mesmos sem nem um problema. |
| 3. | O que são sistemas operacionais baseados em camadas? Descreva a interação entre os programas hospedados pelo sistema operacional e o hardware para esse tipo de SO.  É um SO baseado em hierarquia/preferencia de processos, ou seja, faz uma fila/camada para cada processo que será feito na ordem da hierarquia. Como dito antes, a interação entre os programs e o hardware serão feitos de forma hierárquica, ou seja , com ordem de prioridade para cada processo. |
| 4. | O que são sistemas operacionais baseados em microkernel? Descreva a interação entre os programas hospedados pelo sistema operacional e o hardware para esse tipo de SO.  São sistemas em que quase todas as suas funcionalidades são realizadas fora do núcleo.Os programas se comunicam com um núcleo mínimo que utiliza o menor espaço possível dentro do espaço do sistema. |
| 5. | Para cada um desses tipos de SO, apresente um caso de uso onde a arquitetura do kernel favorece a sua utilização.  o kernel monolítico é mais vantajoso quando há uma limitação no hardware utilizado.  os SO de camadas é mais vantajoso quando se faz processos que são muito “delicados” e que precisam de mais “atenção”  o microkernel é mais vantajoso por ser possível utilizalo em situações diferentes(quando o hardware possui apenas um processador, multiprocessadores ou um sistema distribuído) e tem uma fácil manutenção. |
| **EXERCÍCIO 02** | |
| 1. Explique a diferença entre os seguintes conceitos:    1. Tarefas;   Processos;(c) Threads.  Tarefas são o objetivo a se realizar, ou seja, o que o computador deve fazer.Processo é a execução dos comandos gerados pela tarefa/objetivo.Thread é um pequeno programa que serve para um processo se auto dividir em uma ou mais tarefas.   1. Apresente um exemplo prático para ilustrar as diferenças que você definiu.   O usuário, ao abrir o navegador, gera uma tarefa que ,ao ser interpretado pelo computador, é processada e antes de executar o navegador. Se ,por algum motivo, o usuário tenta abrir mais de um aplicativo, o processo pode gerar uma ou mais threads para realizar mais tarefas quase que simultaneamente, tornando a maquina mais eficiente.   1. O que é a abordagem de Von Neumann? Como ela afeta a distribuição de tarefas em um Sistema Operacional?   Se caracteriza pela possibilidade da maquina digital armazenar seus programas no mesmo espaço de memoria que os dados, para poder manipular esses programas. A arquitetura de von Neuman deixa a realização de tarefas mais eficaz por serem trabalhadas e distribuídas diretamente na memória.   1. Quais são os possíveis estados dos processos em um sistema operacional?   Estado de pronto, estado de execução e estado de espera.   1. Qual é a diferença entre programação sequencial e multiprogramação? Apresente um exemplo para ilustrar suas justificativas.   Na programação sequencial os processos são realizados na ordem que foram declarados, enquanto na multiprogramação podem ter um ou mais processos “ready” e o SO decidirá qual deve ser executado primeiro.um exemplo de programação sequencial seria a soma de dois números enquanto o exemplo para mutiprogramação seria uma programação de prioridades/hierarquia.   1. Descreva como é a utilização do processador em cada um dos exemplos, descrevendo os possíveis estados dos processos em cada etapa.   Sistemas multiprogramaveis tornam mais eficiente a utilização de recursos computacionais por permitirem a execução simultânea de vários programas concorrendo pela utilização do processador. Na programação sequencial o processador realiza uma tarefa por vez, o que torna o processamento menos eficiente mas pesa menos na maquina.   1. Descreva, com exemplos, como ocorre o problema da exclusão mútua e apresente duas possíveis soluções.   Considerando o caso de uma impressora partilhada por várias pessoas como exemplo se não se garantir que apenas um documento é impresso de cada vez podem saír páginas de vários documentos intercaladas.  A utilização de um semaforo binário é uma boa solução para esse problema,assim como Os protocolos de entrada e saída garantirem a exclusão mútua da região crítica de um programa. | |
| **EXERCÍCIO 03** | |
| 1. O que são chamadas de sistema (SYSCALL)?   É o mecanismo pelo qual um programa solicita um serviço do núcleo do SO.   1. Qual a diferença entre uma chamada *trap* e uma interrupção?   A principal diferença é Interrupções são interrupções de hardware, enquanto traps são interrupções invocadas por software.   1. Para um programador, uma chamada de sistema se parece com qualquer outra rotina de | |

Pró-Reitoria Acadêmica

Diretoria Acadêmica

Assessoria Pedagógica da Diretoria Acadêmica

|  |
| --- |
| **EXERCÍCIO 03** |
| biblioteca. É importante que um programador saiba quais rotinas de biblioteca resultam em chamadas de sistema? Sob quais circunstâncias e por quê?  É muito importante sobre a questão do desempenho.Chamadas ao sistema requerem desvio de fluxo e tratamento das chamadas que trazem consigo armazenamento e resgate de contexto, coisas que tomam tempo.   1. A tabela da Figura 1 apresenta um comparativo entre a API em C Win32 e o API Unix POSIX. Considerando o que você sabe sobre chamadas de sistema, apresente um exemplo prático onde a comunicação entre processos é dificultada pela incompatibilidade da API Win32.     *Figura 1: Comparação da API Win32 e Unix*   1. Considere a chamada *read* apresentada em sala de aula. Descreva o comportamento do SO em cada passo de sua execução, assim como o fluxo dos dados na memória principal. |

|  |
| --- |
| **BIBLIOGRAFIA** |
| SILBERSCHATZ, Abraham et al. **Operating system concepts**. Reading: Addison-Wesley, 1998.  TANENBAUM, Andrew S.; MACHADO FILHO, Nery. **Sistemas operacionais modernos** Prentice-Hall, 1995. |

.