

FACULDADE: CENTRO UNIVERSITÁRIO DE BRASÍLIA – UniCEUB

CURSO: CIÊNCIA DA COMPUTAÇÃO

DISCIPLINA: SISTEMAS OPERACIONAIS

NOME: PEDRO RIBEIRO F MELO

SEMESTRE/ANO: 2020/2

Lista 1

Exercício 1)

1) Primeiro, haverá a programação comum, como ocorre em um desktop de uso popular. Na máquina, a interface costuma ser simplificada e feita por meio de painéis ou os chamados computadores de uso industrial.

2) O núcleo monolítico é uma arquitetura de núcleo onde todo o conjunto de instruções de controle do hardware é executado no espaço de núcleo no modo de supervisão, ou seja, é um único executável que possui todos os códigos de suporte necessários agregados, a interação entre esse SO e os programas funcionam da seguinte forma: O GRUB o executa, que o faz iniciar o carregamento de drivers e inicializar o sistema. E uma vez executado, o kernel executa sob 1 slot da memória RAM chamado “espaço de kernel” o qual o usuário não tem acesso.

3) O SO é estruturado em uma hierarquia em camadas, onde cada camada implementa uma parte do sistema, na estruturação do sistema como hierarquia em camadas, o núcleo do sistema operacional é dividido em camadas, sendo que cada uma delas trata do gerenciamento de alguma parte do hardware, fornecendo para as camadas superiores uma visão abstrata da parte do hardware gerenciada por esta camada.

4) Micronúcleo, ou microkernel, é uma arquitetura de núcleo de um sistema operativo cujas funcionalidades são quase todas executadas fora do núcleo, em oposição a um núcleo monolítico. Os processos se comunicam com um núcleo mínimo, usando o mínimo possível o "espaço do sistema" Neste local os aplicativos tem acesso a todas as instruções e a todo o hardware e deixando o máximo de recursos rodando no "espaço do usuário" em que o software tem algumas restrições, não podendo acessar algumas hardwares, nem tem acesso

a todas as instruções).

5) O monolítico costuma ter melhor performance porque faz menos mudanças de contexto, micronúcleo (microkernel) tende a ser mais confiável e seguro e a em camadas tem como vantagens o isolamento das funções do SO facilitando sua manutenção e depuração.

Exercícios 2)

6) Um programa consiste num conjunto de instruções que você elabora para atingir determinado objectivo. Ou seja o programa consiste essencialmente no código fonte.

Uma thread é um componente responsável por executar as instruções que você elaborou no seu programa. O processador executa sempre instruções no contexto duma thread.

Um processo é uma entidade dinâmica, que altera seu estado à medida que avança sua execução. Assim, o processo pode ser encarado como uma abstração que representa um programa em execução.

7) No exemplo pratico podemos ver na memória a tarefa dela é armazenar o processo da memória tem obrigatoriamente de passar por três fases: a memorização, o armazenamento e a rememoração e a threads é a execução disso tudo.

8) A Arquitetura de von Neumann é uma arquitetura de computador que se caracteriza pela possibilidade de uma máquina digital armazenar seus programas no mesmo espaço de memória que os dados, podendo assim manipular tais programas. Esta arquitetura é um projeto modelo de um computador digital de programa armazenado que utiliza uma unidade de processamento (CPU) e uma de armazenamento ("memória") para comportar, respectivamente, instruções e dados.

9) Em execução, pronto e bloqueado.

10) Multiprogramação é uma execução, em paralelo, de múltiplos programas na mesma máquina e uma programação sequencial realiza um conjunto predeterminado de comandos de forma sequencial, de cima para baixo, na ordem em que foram declarados.

11) Processador é o conjunto da unidade lógica e aritmética, registradores e da unidade de controle. Sua função é executar os programas armazenados na

memória principal, buscando suas instruções, examinando-as, e então executando uma após a outra.

O processador é responsável pela realização de uma série de funções:

Busca de instruções e dados na memória;

Programa a transferência de dados entre a memória e os dispositivos de entrada/saída;

Decodifica as instruções;

Realiza as operações lógica e aritméticas;

Responde a sinais enviados por dispositivos de entrada/saída como RESET ou interrupções.

12) Exclusão mútua é um recurso ou técnica que evita que dois processos tenham acesso simultâneo a um recurso compartilhado no processador, isso ocorre, por exemplo, em um semáforo binário, ou seja, que só pode assumir dois valores, 0 e 1. Esse bloqueio acontece sempre antes do recurso ser utilizado, assim só uma thread usará o recurso, e após o recurso ser utilizado o semáforo entra em desuso. As soluções para a exclusão mútua são a solução de Dekker, em que é usada a variável "vez" para realizar o "tie-break", e a solução de Peterson.

Exercício 3)

13) Em computação, uma chamada de sistema é o mecanismo programático pelo qual um programa de computador solicita um serviço do núcleo do sistema operacional sobre o qual ele está sendo executado.

14) Uma *Trap* é uma exceção em um processo do usuário. É causado por divisão por zero ou acesso inválido à memória. É também a maneira usual de invocar uma rotina do kernel (uma chamada do sistema) porque é executada com uma prioridade mais alta que o código do usuário.

Uma interrupção é algo gerado pelo hardware (dispositivos como disco rígido, placa gráfica, portas de E / S, etc.). Elas são assíncronas (ou seja, não ocorrem em locais previsíveis no código do usuário) ou "passivas", pois o manipulador de interrupções precisa esperar que elas aconteçam eventualmente.

15) 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.

16) Gerência de processos e threads

Gerência de memória

Gerência do sistema de arquivos

Gerência de dispositivos

17) 1, 2 e 3 – Guarda parâmetros na pilha.

4 – chamada real ao procedimento na biblioteca.

5 – programa coloca o número da chamada ao sistema de um local esperado pelo SO.

6 – TRAP para passar do modo usuário para modo núcleo e iniciar a execução em um endereço no núcleo.

7 – despacha para procedimento correto de tratamento de chamada.

8 – executado o procedimento de tratamento da chamada ao sistema.

9 – retorna para o procedimento de biblioteca do espaço do usuário.

10 – retorno ao espaço do usuário.

11 – programa do usuário limpa pilha.