Lista

1 - Como funciona a interface homem máquina em um sistema operacional?
O Hardware fornece os recursos computacionais para o sistema operacional que controla as aplicações sobre o manuseio do homem.

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.
São sistemas onde um programa principal invoca a rotina do serviço, um conjunto de rotinas de serviço executam as chamadas de sistema e um conjunto de rotinas utilitárias que auxiliam as rotinas de serviço.
O sistema operacional é executado como um único programa, escrito como uma coleção de procedimentos com parâmetros e resultados que podem chamar umas as outras.
Tem como principal vantagem o tempo de resposta das tarefas.

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.
São sistemas com hierarquia de serviço entre as camadas.
Camadas: Operador;
Programa usuário;
Gerenciamento de entrada e saída;
Comunicação operador-processo;
Gerenciamento de memória;
Alocação do processador e multiprogramação.
As camadas superiores prestam serviço aos inferiores e cada camada é dependente.

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.
O sistema operacional é dividido em partes com cada uma responsável por um tipo de serviço, é dividido em: comunicação entre processos, gerenciamento de memória e escalonamento de processos. A comunicação acontece por troca de mensagens entre os programas e os serviços.

5 - Para cada um desses tipos de SO, apresente um caso de uso onde a arquitetura do kernel favorece a sua utilização.
No Sistema Monolítico a arquitetura de kernel oferece a vantagem dos procedimentos com parâmetros e resultados que podem chamar umas as outras e diminuir o tempo de resposta das tarefas.
No Microkernel deu a opção de separar o sistema operacional em partes sendo cada uma responsável por um tipo de serviço e também permite a troca de mensagens entre ambos.
No sistema em camadas surgiu uma maneira de fazer uma hierarquia em camadas onde as superiores prestam serviços as inferiores.

6 - Explique a diferença entre os seguintes conceitos:
(a) Tarefas: conjunto de ações a serem realizadas para o cumprimento de um objetivo em um determinado tempo

(b) Processos: são programas que estão sendo executados em um espaço virtual de endereçamento exclusivo.
(c) Threads: é a linha de execução dentro de um processo

7 - Apresente um exemplo prático para ilustrar as diferenças que você definiu.
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 - O que é a abordagem de Von Neumann? Como ela afeta a distribuição de tarefas em um Sistema Operacional?
Consiste em uma memória , uma unidade aritimetica lógica (ULA) e uma unidade de controle(CU).Ela possibilitou as maquinas armazenarem seus programas no mesmo espaço que os dados assim podendo manipular os programas

9 - Quais são os possíveis estados dos processos em um sistema operacional?
Run que está sendo executado no processador, ready ou executável dispõe de todos os recursos que precisa e está pronto para ser executado,sleep ou dormente bloqueado à espera de algum recurso, e só pode ser desbloqueado se receber um sinal de outro processo, zumbi onde um processo é criado por um programa, que por sua vez é finalizado antes de receber o resultado do processo,parado Recebeu ordem do administrador para interromper a execução. Será reiniciado se receber um sinal de continuação

10 - Qual é a diferença entre programação sequencial e multiprogramação? Apresente um exemplo para ilustrar suas justificativas.
Programação sequencial é uma programação que realiza um conjunto predeterminado de comandos de forma sequencial na ordem em que foram feitos.
L1 – ação 1
L2 – ação 2
L3 – ação 3
Assim por diante.
Multiprogramaçao: uma técnica para maximizar o uso da CPU
Um exemplo dela é a utilização mais inteligente nos recursos de hardware

11- Descreva como é a utilização do processador em cada um dos exemplos, descrevendo os possíveis estados dos processos em cada etapa.
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- Descreva, com exemplos, como ocorre o problema da exclusão mútua e apresente duas possíveis soluções.
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" pararealizar o "tie-breaak", e a solução de Peterson.

13- O que são chamadas de sistema (SYSCALL)?
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- Qual a diferença entre uma chamada trap e uma interrupção?
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- Para um programados, uma chamada de sistema se parece com qualquer outra rotina de 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`.

16 - 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.
Há vários exemplos, como *execve, link, mount, unmount, chmod, kill.*

17. 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.
- Armazena os bytes;
- Carrega no buffer;
- Gera o descritor de arquivo (fd);
- Chama a rotina da biblioteca (call);
- Executa a instrução TRAP. Nesse momento a chamada é promovida ao modo kernel;
- Passa a instrução para um endereço específico do kernel;
- Ativa o endereço específico para a chamada (registradores);
- Rotina de tratamento das chamadas de sistema;
- Retorna para a instrução TRAP. Podem também bloquear o

programa que a chamou;
- Retorna ao programa do usuário;
- Limpa a pilha.