

Silberschatz:

Capítulo 1

- 1) - Proporcionar a interface entre o usuário e o hardware;
 - Controlar todos os dispositivos de hardware;
- 2) - O processo bloqueia para entrada;
 - O agendador seleciona outro processo;
 - O agendador seleciona esse processo;
 - Entrada torna-se disponível;
- 3) Vários jobs são mantidos na memória ao mesmo tempo, aumentando a utilização da CPU e diminuindo o tempo total para executar os trabalhos.
- 4) A interface do sistema operacional para computadores pessoais é mais amigável do que a dos mainframes, computadores pessoais são menores e mais baratos.
- 6)
 - a) Batch: sistema operacional carregado por cartões perfurados e ficava armazenado na memória e sua principal função era transferir o controle automaticamente de uma tarefa para a próxima, sem limite de tempo para resposta;
 - b) Interativo: sistema que permite a comunicação entre o usuário e sistema;
 - c) Compartilhado: permite o acesso de vários usuários simultaneamente;
 - d) Tempo Real: sistema onde o tempo é importante, usado geralmente como dispositivo de controle em aplicação dedicada e o processamento tem de ser feito dentro dos limites definidos ou o sistema falhará;
 - e) Rede: sistema que fornece recursos como compartilhamento de arquivos pela rede, um computador com sistema de rede atua independente dos outros computadores na rede, podendo se comunicar com os computadores na rede;

f) Distribuído: esse sistema não compartilha memória e nem clock, o processador tem sua própria memória local, os processadores se comunicam através de barramentos ou linhas telefônicas;

8) A principal vantagem desse esquema é que os programas podem ser maiores que a memória física, libera os programadores da preocupação relativa aos limites da memória.

9) A diferença é que no assimétrico existe um processador mestre que controla o sistema, os outros processadores recebem instruções do processador mestre, no simétrico isso não ocorre, não existe relação mestre-escravo, todos tem uma cópia do S.O.

Vantagens: maior produção, economia de dinheiro (compartilhar periférico) e aumento de confiabilidade.

Desvantagens: a taxa de aumento na velocidade com N processadores, entretanto não é N, é menor que N, Esforço dos processadores para resolver determinadas tarefas mais disputada por recursos compartilhados, diminui o ganho esperado dos processadores adicionais.

Capítulo 2

2) Funciona com o bit de modalidade, monitor (0) e usuário (1), com esse bit dá para distinguir entre tarefas que serão executadas em nome do S.O. ou em nome de usuário.

3) Interrupção: intervenção do SO na execução de um programa devido a ocorrência de um evento pode ser tanto de hardware ou de software geradas por eventos assíncronos.

Exceção: interrupção gerada por software, causada tanto por um erro quanto por uma solicitação específica geradas por eventos síncronos.

4) Para dispositivos de I/O de alta velocidade, um dispositivo de alta velocidade é capaz de transmitir informações próximas à da memória, dependendo do tempo que levará para responder as interrupções não sobrar muito tempo para execução de processos.

5) a - timer.

6) Sim, mais irá conter falhas.

7) Criação de interrupções.

8) b,c,d,e.

9) - Armazenamento mais rápido

- Auxilia a memória principal

problemas: - gerenciamento do cache (devido tamanho limitado)

Por causa do custo.

12) LANs: permite a comunicação entre processadores distribuídos por uma área geográfica pequena, conectadas por uma rede de computadores;

WANs: possibilitam a comunicação entre processadores distribuídos por uma grande área geográfica;

Normalmente as LANs são mais rápidas.

13) a) LAN, b) LAN, c) WAN, d) WAN

Capítulo 3

1) - criar e apagar processos tanto de sistema como de usuário;

- suspender e recomendar processos;

- fornecer mecanismos para sincronização de processos;

- fornecer mecanismos para comunicação entre processos;

- fornecer mecanismos para manipulação de deadlock;

2) - monitorar as porções de memória que estão sendo corretamente usadas e por quem estão sendo usadas;

- decidir que processos estão prontos para ser carregados na memória quando o espaço de memória tornar-se disponível;

- alocar e desalocar espaço de memória conforme necessário;

3) - gerenciamento do espaço livre;

- alocação do espaço de armazenamento;
- programação de alocação de disco.

4) - criar e apagar arquivos;

- criar e apagar diretórios;
- suportar primitivos para manipulação de arquivos e diretórios;
- mapear arquivos em memória secundária;
- criar cópias de arquivos em mídias de armazenamento estáveis (não-voláteis).

5) Proporcionar a interface entre usuário e sistema operacional.

6) - execução do programa, controla a execução do programa para que o usuário utilize os programas;

- operações de I/O, caso algum programa utilize operações de entrada e saída, ex: audio e video;
- manipulação do sistema de arquivos, acesso a determinados arquivos, isso varia conforme o

SO.

7) Constituem a interface entre um processo e o sistema operacional. Ex: quando se está programando com arquivo, a abertura, criação, remoção de arquivos, tudo isso faz chamada ao sistema.

10) - proporcionar um ambiente eficiente para o desenvolvimento e execução de programas;
- resolvem problemas comuns e operações comuns;

São divididos em categorias:

- gerenciamento de arquivos;
- informação de estado;
- modificação de arquivos;
- suporte a linguagens de programação;
- carga e execução de programas;
- comunicação.

11) Modularidade.

12) "Deixa o kernel limpo". Estrutura o sistema operacional removendo todos os componentes não essenciais do kernel e implementando-os como programas de nível de sistema e de usuário.

13) Para usuário: fazer parecer que cada usuário tem seu próprio processador.

14) Aumenta o desempenho. Porque converte os bytecodes independentes de arquitetura para a linguagem e máquina do compilador hospedeiro.

Capítulo 5

1) - navegador web;

- editor de texto.

3) Threads de usuário:

- são suportados acima do kernel;

- criação e gerência de threads é mais rápido;

Em caso do kernel ser organizado com único thread. Pior threads de kernel.

- são suportados diretamente pelo SO;

- criação e regência de threads é mais lenta;

Em caso do kernel ser organizado com vários threads. Pior.

4) O kernel realiza a criação, o scheduling e o gerenciamento de threads, no espaço do kernel. Se algum thread executar uma chamada de sistema de bloqueio, o kernel poderá agendar outro thread na aplicação para entrar em execução.

5) A biblioteca fornece suporte a criação, o scheduling e o gerenciamento de threads, sem suporte proveniente do kernel, toda a criação e o scheduling dos threads é realizado no espaço do usuário sem precisar da intervenção do kernel.