

Exercícios Teóricos – Processos

1 - Qual a diferença entre programa e processo?

Programa é o código parado. Processo é o programa em execução, usando memória, CPU e outros recursos.

2 - Quais são os estados de um processo e quando ocorrem as transições?

- Novo: o processo está sendo criado.
 - Pronto: esperando a vez de usar a CPU.
 - Executando: está usando a CPU.
 - Esperando/Bloqueado: esperando algum recurso (ex: disco).
 - Finalizado: terminou sua execução.
- As transições ocorrem quando o processo inicia, é interrompido, volta a esperar ou termina.

3 - O que contém um Process Control Block (PCB)?

Informações como ID do processo, estado, registradores, área de memória, arquivos abertos e prioridades.

4 - O que acontece com os recursos de um processo quando ele termina?

São liberados pelo sistema (memória, arquivos, CPU), e o processo é removido da tabela de processos.

5 - Qual a diferença entre fork() e exec() no UNIX?

fork() cria um novo processo igual ao pai. exec() troca o código do processo atual por outro programa.

6 - Como funciona a hierarquia de processos em UNIX?

Um processo pode criar outros (filhos). Todos os processos vêm de um processo inicial (geralmente o init/systemd).

7 - Compare memória compartilhada e troca de mensagens (IPC).

Memória compartilhada é mais rápida, mas exige controle. Troca de mensagens é mais segura e organizada, porém mais lenta.

8 - Cite exemplos de chamadas de sistema usadas em IPC.

- Memória compartilhada: shmget, shmat, shmdt
- Mensagens: msgget, msgsnd, msgrcv
- Sockets: socket, send, recv

9 - Por que é importante que o sistema operacional faça gerenciamento de processos?

Para organizar a execução, evitar conflitos e garantir que os recursos sejam usados de forma eficiente.

10 - Explique a diferença entre processos independentes e processos cooperativos.

Independentes não se comunicam com outros. Cooperativos trocam dados e dependem uns dos outros.

11 - O que é um processo zumbi em UNIX/Linux?

É um processo que já terminou, mas ainda está na tabela até o processo pai ler seu status com wait().

12 - Explique a diferença entre chamadas bloqueantes e não bloqueantes em IPC.

Bloqueantes fazem o processo esperar. Não bloqueantes permitem que o processo continue executando.

13 - Qual a diferença entre processo pesado (process) e thread (processo leve)?

Processo tem recursos próprios. Thread é mais leve e compartilha recursos com outras threads do mesmo processo.

14 - Por que sistemas operacionais multiprogramados precisam de troca de contexto (context switch)?

Para alternar entre processos e aproveitar melhor a CPU enquanto um processo espera por algo.

15 - Cite vantagens e desvantagens da comunicação via memória compartilhada.

Vantagens: rápida e eficiente.

Desvantagens: precisa de controle para evitar erros e conflitos entre processos.