

# Lista 02

## 1. Defina conceito de processo

Um processo pode ser considerado como um programa em execução, também um conjunto de informações necessárias para a concorrência de programas no sistema operacional. Além disso, também pode ser entendido como o ambiente onde um programa é executado

## 2. Porque o conceito de processos é tão importante no projeto de sistemas multiprogramáveis?

Através de processos, um programa pode alocar recursos, compartilhar dados, trocar informações e sincronizar sua execução

## 3. É possível que um programa execute no contexto de um processo e não execute no contexto de um outro? Por que?

Não. Cada processo é direcionado para um programa em execução, logo um programa pode não ir executar no contexto de outro processo, por este não possuir os requisitos necessários para que ele execute.

## 4. Quais partes compõem um processo?

Um processo é formado por três partes, conhecidas como contexto de hardware, contexto de software e espaço de endereçamento, que juntas mantêm todas as informações necessárias à execução de um programa

## 5. O que é contexto de hardware de um processo e como é a implementação de troca de contexto?

O contexto de hardware armazena o conteúdo dos registradores gerais da UCP, além dos registradores de uso específico como program counter (PC), stack pointer (SP) e registrador de status.

## 6. Qual a função do contexto de software? Exemplifique cada grupo de informação.

No contexto de software são especificadas características e limites dos recursos que podem ser alocados pelo processo, como o número máximo de arquivos abertos simultaneamente, prioridade de execução e tamanho do buffer para operações de E/S

## 7. O que é espaço de endereçamento de um processo?

Espaço de endereçamento é o intervalo de endereços de memória que um processo pode acessar.

8. Como o sistema operacional implementa o conceito de processo? Qual a estrutura de dados indicada para organizar os diversos processos na memória principal?

O processo é implementado pelo sistema operacional através de uma estrutura de dados

chamada bloco de controle de processo (PCB)

9. Defina os cinco estados possíveis de um processo.
- **Ativo** (ou executando), quando estiver fisicamente ocupando a cpu;
  - **Pronto**, quando estiver apenas esperando por uma chance de ocupar a cpu;
  - **Em espera** (ou bloqueado), quando estiver esperando pela ocorrência de algum evento externo que o permita ficar esperando apenas pela cpu;
  - **Inativo** (ou morto), quando não estiver executando.
10. Dê exemplo que apresenta todas as mudanças do estado de um processo, juntamente com o evento associado a cada mudança.
- **Pronto para Ativo**: quando o *dispatcher* escalonar o processo;
  - **Ativo para Pronto**: quando o *dispatcher* decidir que outro processo deve ocupar a cpu (por estouro do tempo de cpu, por exemplo);
  - **Ativo para Em espera**: quando o processo solicitar uma operação de entrada/saída ou for bloqueado por algum semáforo (ou outro mecanismo de bloqueio);
  - **Ativo para Inativo**: quando o processo for retirado da memória ou terminar sua execução;
  - **Inativo para Pronto**: quando o *scheduler* trazer o processo do disco para a memória;
  - **Em espera para Pronto**: quando a operação de E/S for completada ou o processo for desbloqueado.
11. Diferencie processos multithreads, subprocessos e processos independentes.

Processos multithreads é um conceito onde vários processos são executados simultaneamente sem que um interfira no outro.

Subprocessos são processo que derivão de outros processo, como em uma estrutura hierarquica

processo independentes, são processo que não possui vinculo a outro processo

**12. Explique a diferença entre processos foreground e background**

Os processos executado em foreground são aqueles que necessitam de interação direta com o usuário, incluindo troca de informações. Os processo em background não necessitam desta interação com o usuário.

**13. Qual a relação entre processo e arquitetura microkernel?**

A arquitetura microkernel implementa o uso intensivo de processos que disponibilizam serviços para processos das aplicações e do próprio sistema operacional.

Dê exemplos de aplicações CPU-bound e I/O-bound?

**14. Justifique com um exemplo a frase “o sinal esta para o processo assim como as interrupções e exceções estão para os sistemas operacionais”**

Quando ocorre uma divisão por zero, por exemplo, o sistema operacional é notificado do problema através de uma exceção. Por sua vez, o sistema deve nocar ao processo que gerou o problema através de um sinal.

**15. Explique como a eliminação de um processo utiliza o mecanismo de sinais.**

Quando um processo é eliminado, o sistema ativa o sinal associado a este evento. O processo somente será excluído do sistema quando for selecionado para execução. Neste caso, é possível que o processo demore algum período de tempo até ser eliminado de fato.