

# **Exercício**

## **1) Defina o conceito de processo.**

Um processo pode ser definido como o ambiente onde um programa é executado. Este ambiente, além das informações sobre a execução, possui também o quanto de recursos do sistema cada programa pode utilizar, como o espaço de endereçamento, tempo de processador e área em disco.

## **2) Por que o conceito de processo é tão importante no projeto de sistemas multiprogramáveis?**

Através de processos, um programa pode alocar recursos, compartilhar dados, trocar informações e sincronizar sua execução. Nos sistemas multiprogramáveis os processos são executados concorrentemente, compartilhando o uso do processador, memória principal, dispositivos de E/S dentre outros recursos.

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

Sim, pois a execução de um programa pode necessitar de recursos do sistema que um processo pode possuir enquanto outro não.

## **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 juntos 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 da 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. Quando um processo está em execução, o seu contexto de hardware está armazenado nos registradores do processador. No momento em que o processo perde a utilização da UCP, o sistema salva as informações no contexto de hardware do processo.

## **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. O contexto de software é composto por três grupos de informações sobre o processo: identificação, quotas e privilégios.

## **7) O que é o espaço de endereçamento de um processo?**

O espaço de endereçamento é a área de memória pertencente ao processo onde as instruções e dados do programa são armazenados para execução. Cada processo possui seu próprio espaço de endereçamento, que deve ser devidamente protegido do acesso dos demais processos.

## **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 do processo (Process Control Block — PCB). A partir do PCB, o sistema operacional mantém todas as informações sobre o contexto de hardware, contexto de software e espaço de endereçamento de cada processo.

## **9) Defina os cinco estados possíveis de um processo.**

Estado de Execução: processo que está sendo processado pela UCP no momento.

Estado de Pronto: processo que aguarda para ser executado.

Estado de Espera: processo que aguarda por algum evento ou recurso para prosseguir processamento.

Estado de Criação: processo cujo PCB já foi criado porém ainda não teve seu processamento iniciado.

Estado de Terminado: processo que não pode ter mais nenhum programa executado no seu contexto, porém o sistema operacional mantém suas informações de controle presentes na memória.

## **10) Diferencie processos multithreads, subprocessos e processos independentes.**

Processos independentes não têm vínculo com os processos criadores. A criação de um processo independente exige a alocação de um PCB, possuindo contextos de hardware, contexto de software e espaço de endereçamento próprios. Subprocessos são processos criados dentro de uma estrutura hierárquica. Caso um processo pai deixe de existir, os subprocessos subordinados são automaticamente eliminados.

Semelhante aos processos independentes, subprocessos possuem seu próprio PCB.

Além da dependência hierárquica entre processos e subprocessos, uma outra característica neste tipo de implementação é que subprocessos podem compartilhar quotas com o processo pai. Neste caso, quando um subprocesso é criado, o processo pai cede parte de suas quotas ao processo filho.

Processos multithreads suportam múltiplos threads, cada qual associado a uma parte do código da aplicação. Neste caso não é necessário haver diversos processos para a

implementação da concorrência. Threads compartilham o processador da mesma maneira que um processo, ou seja, enquanto um thread espera por uma operação de E/S, outro thread pode ser executado.

### **11) Explique a diferença entre processos foreground e background.**

Um processo foreground é aquele que permite a comunicação direta do usuário com o processo durante o seu processamento. Neste caso, tanto o canal de entrada quanto o de saída estão associados a um terminal com teclado, mouse e monitor, permitindo, assim, a interação com o usuário. Um processo background é aquele onde não existe a comunicação com o usuário durante o seu processamento. Neste caso, os canais de E/S não estão associados a nenhum dispositivo de E/S interativo, mas em geral a arquivos de E/S.

### **12) Qual a relação entre processo e arquitetura microkernel?**

A arquitetura microkernel baseia-se na utilização de processos em modo usuário para executar diversas funções relativas ao sistema operacional, como gerência de memória e escalonamento.

### **13) Dê exemplos de aplicações CPU-bound e I/O-bound.**

**CPU\_Bound**, exemplo temos os processos utilizados em aplicações científicas que efetuam muitos cálculos.

**IO/Bound, exemplo** aplicações comerciais, que se baseiam em leitura, processamento e gravação