Tarefa - 3

Nome: André Luiz Bernardes de Oliveira RA: 1460282313052

1. **Defina o que é um processo em SO e os estados e trocas possíveis de um processo.**

R: Um processo é um programa em execução no sistema operacional que pode estar em diferentes estados. Inicialmente, o processo é criado e fica esperando para ser executado. Quando recebe a CPU, ele começa a rodar, mas pode ser interrompido se precisar esperar por algo, como uma operação de entrada ou saída (I/O). Se isso acontecer, ele entra no estado de espera. Depois que o evento de espera termina, o processo volta para a fila de prontos e, ao final, completa sua execução.

1. **Descreva o conceito de thread e quais as vantagens em sua utilização:**

R: Uma thread é uma subdivisão de um processo, que compartilha seus recursos (como memória), mas tem seu próprio fluxo de execução. As threads permitem que o programa rode partes de seu código ao mesmo tempo, o que melhora o desempenho, especialmente em sistemas com múltiplos núcleos, onde elas podem ser executadas em paralelo. Além disso, como compartilham o mesmo espaço de memória, a comunicação entre as threads é mais rápida e eficiente, o que também torna o programa mais responsivo.

1. **Fundamente com suas palavras região crítica e exclusão mútua, e explique porque ela é importante**

R: A região crítica é uma área do código onde recursos compartilhados (como variáveis ou arquivos) são acessados, e a exclusão mútua é a regra que impede que mais de uma thread ou processo entre nessa região ao mesmo tempo. Isso é essencial para evitar conflitos e garantir que os dados permaneçam consistentes, pois, sem a exclusão mútua, múltiplas threads poderiam modificar os mesmos dados simultaneamente, resultando em erros.

1. **Diferencie as soluções espera ocupada, sincronização condicional, Semáforos, Monitores e Passagem de Mensagem**

R:

* Na espera ocupada, uma thread permanece em um loop verificando continuamente se pode acessar o recurso, o que consome a CPU desnecessariamente.
* Na sincronização condicional, a thread "dorme" até que uma condição seja atendida, evitando desperdício de CPU.
* Semáforos são variáveis que controlam o acesso de threads aos recursos, permitindo que elas se organizem ao acessar a região crítica.
* Monitores são uma estrutura que permite a entrada de uma thread por vez na região crítica, usando variáveis de condição para comunicação entre threads.
* Na passagem de mensagem, os processos se comunicam enviando mensagens, sem compartilhar diretamente a memória, o que facilita a troca de informações e evita o uso de recursos compartilhados.

1. **Detalhe o que é deadlock, quais as condições para obtê-lo com certeza e quais as soluções possíveis**

R: Deadlock é uma situação em que processos ou threads ficam presos, cada um esperando que outro libere um recurso de que precisam. Para que o deadlock aconteça, algumas condições devem existir, como a exclusão mútua e uma cadeia circular de espera. Para evitar deadlock, é possível prevenir essas condições, como não permitir cadeias circulares, ou detectá-lo em execução e resolver o problema, liberando alguns dos processos.