

# Sistemas OPERATIVOS

Data -> 29 MAIO 2023

I

1-  
a)

O mecanismo de comutação de processos permite que o sistema operativo alterne entre a execução de diferentes processos de forma rápida e aparentemente simultânea. Isto é feito dividindo o tempo de CPU disponível em fatias pequenas e atribuindo uma fatia para cada processo. O Sistema operativo interrompe um processo em execução, salva o seu estado e contexto e restaura o estado e contexto de outro processo para continuar a execução. Esse processo de comutação ocorre num intervalo de tempo tão curto que os utilizadores ficam com a percepção que vários processos estão a ser executados simultaneamente.

b)

A escolha adequada do intervalo de tempo para a comutação de processos é importante para o desempenho das aplicações. Se o intervalo de tempo for muito curto, ocorrerão comutações frequentes, o que pode resultar em sobrecarga do sistema devido ao contexto de comutação e reduzir o tempo de CPU efetivamente utilizado pelas aplicações. Por outro lado, se o intervalo for muito longo, um processo pode monopolizar a CPU por um tempo excessivo, causando atrasos perceptíveis nas outras aplicações. É necessário encontrar um equilíbrio entre a frequência de comutação e a carga do sistema, garantindo assim, um desempenho eficiente do sistema operativo.

2→

- a) Para melhorar o desempenho da aplicação de armazenamento e leitura de arquivos, é mais apropriado usar blocos maiores de gravação, como 4096 bytes. Quando ocorre uma chamada ao sistema write, o sistema operativo geralmente agrupa os dados em blocos antes de os escrever no dispositivo de armazenamento. Escrever um tamanho maior de uma só vez reduz a quantidade de chamadas ao sistema e, conseqüentemente, a sobrecarga de operações de gravação e sincronização do sistema de arquivos. Portanto, escrever tamanhos maiores melhora o desempenho da aplicação.
- b) Para uma leitura mais eficiente de ficheiros, o acesso sequencial é preferível, pois permite a leitura contígua e utiliza a localidade espacial para otimizar a busca de dados adjacentes. Isso é mais eficiente do que a leitura aleatória, que pode levar a tempos de busca e acesso mais longos e, conseqüentemente, a um desempenho inferior.