

**Introdução:** A computação moderna está intrinsecamente ligada à alocação de recursos, um dos pilares fundamentais que impulsionam a eficiência e a confiabilidade dos sistemas. À medida que organizações e indivíduos buscam otimizar suas operações, é essencial compreender o que está por trás da escolha entre recursos preemptíveis e não preemptíveis.

Vivemos em uma era em que a demanda por recursos computacionais é ampla e variada. Desde a execução de cargas de trabalho de alto desempenho até a hospedagem de aplicativos críticos para os negócios, as organizações enfrentam a tarefa de encontrar o equilíbrio entre eficiência financeira e confiabilidade operacional. A escolha entre recursos preemptíveis, que podem ser interrompidos a qualquer momento, e recursos não preemptíveis, que garantem estabilidade contínua, é um dilema que permeia a tomada de decisões em muitos ambientes de computação. Este trabalho tem como objetivo explorar a complexidade desse tópico crucial na computação.

Recursos preemptíveis referem-se a recursos que podem ser interrompidos e realocados pelo sistema operacional, geralmente sem causar problemas significativos ou perda de dados

**Recursos Preemptíveis:** Um recurso preemptível é um recurso que pode ser interrompido ou suspenso temporariamente sem causar danos ao sistema ou ao processo proprietário. Em outras palavras, é um recurso que pode ser interrompido a qualquer momento, sem afetar o resultado final do processo. Esse tipo de recurso é comum em sistemas operacionais, onde a memória é um exemplo de recurso preemptível. Quando um processo precisa de mais memória, o sistema operacional pode suspender temporariamente outro processo para liberar a memória necessária. Isso é feito sem afetar o resultado final do processo suspenso. Em resumo, um recurso preemptível é um recurso que pode ser interrompido temporariamente sem causar danos ao sistema ou ao processo proprietário. A memória é um exemplo de um recurso preemptível. Considere, por exemplo, um sistema com uma memória de usuário de 1 GB, uma impressora e dois processos

de 1 GB cada que querem imprimir algo. O processo A solicita e ganha a impressora, então começa a computar os valores para imprimir. Antes que ele termine a computação, ele excede a sua parcela de tempo e é mandado para o disco. O processo B executa agora e tenta, sem sucesso no fim das contas, ficar com a impressora. Potencialmente, temos agora uma situação de impasse, pois A tem a impressora e B tem a memória, e nenhum dos dois pode proceder sem o recurso contido pelo outro. Felizmente, é possível obter por preempção (tomar a memória) de B enviando-o para o disco e trazendo A de volta. Agora A pode executar, fazer sua impressão e então liberar a impressora. Nenhum impasse ocorre. O tempo de CPU é outro recurso preemptível, pois o sistema operacional pode interromper a execução de um processo para permitir que outro processo use a CPU. Alguns dispositivos de entrada/saída, como teclados e mouses, são preemptíveis, pois o sistema operacional pode interromper a entrada de um processo para permitir que outro processo use o dispositivo.

#### Vantagens:

**Priorização de tarefas:** Os recursos preemptíveis permitem que o sistema operacional aloque recursos com base nas prioridades das tarefas em execução. Isso pode garantir que tarefas críticas obtenham acesso aos recursos necessários quando necessário.

**Melhor desempenho do sistema:** O uso eficaz de recursos preemptíveis pode ajudar a evitar bloqueios e atrasos, garantindo que tarefas críticas sejam tratadas de forma rápida e eficiente.

**Flexibilidade:** Os recursos preemptíveis podem ser alocados dinamicamente com base nas necessidades em tempo real do sistema. Isso permite a adaptação a situações em que as prioridades mudam rapidamente.

#### Desvantagens:

**Complexidade:** A implementação de recursos preemptíveis pode adicionar complexidade ao sistema operacional. Isso pode tornar o desenvolvimento e a manutenção mais desafiadores.

Potencial para interrupções indesejadas: Se não implementado corretamente, o uso indevido de recursos preemptíveis pode levar a interrupções indesejadas em tarefas críticas, causando problemas de estabilidade no sistema.

Overhead: A gestão de recursos preemptíveis requer algum overhead, o que pode consumir recursos do sistema, especialmente em sistemas com recursos limitados.

OBS: "Overhead" é um termo amplamente utilizado em várias áreas, incluindo tecnologia da informação, negócios, engenharia e muitos outros campos. Ele se refere a custos, recursos ou tempo adicionais necessários para realizar uma determinada tarefa ou processo que não está diretamente relacionado ao objetivo principal desse processo. Em resumo, overhead é um fator que não contribui diretamente para a realização de uma tarefa ou objetivo, mas ainda precisa ser considerado.

## BOLSONARO

**Recursos não-preemptíveis:** São aqueles que não podem ser interrompidos sem causar problemas significativos. Dispositivos de entrada e saída (I/O), como discos rígidos e impressoras, são exemplos clássicos de recursos não preemptíveis. A interrupção da operação de um dispositivo I/O em andamento pode resultar em perda de dados ou corrupção, tornando esses recursos mais desafiadores de gerenciar. it does not interrupt a process running CPU in the middle of the execution. In a non-preemptive environment, once a process enters the execution state, it cannot be brought forward until it has completed its allocation time. With non-preemptive OS, the process itself decides whether it wants to leave the processor. CPU can only be taken away from a process when it is terminated or blocked. A process can occupy the CPU as long as it wants. Non-preemptive scheduling has both advantages and disadvantages.

## VANTAGENS:

- Minimal scheduling overhead: Non-preemptive scheduling is simple and doesn't require a lot of computational resources.

- High throughput: It can execute a high volume of processes in a certain length of time.
- Simplicity: It is easy to understand and implement.

#### **DESVANTAGENS:**

- Poor response time: The response time to the process can be quite high<sup>1</sup>.
- System freeze: If there are bugs in the process, it can cause the computer to freeze up<sup>1</sup>.
- Inflexibility: Once a process has been allocated the CPU, it cannot be interrupted until it completes its execution or reaches a waiting state<sup>1</sup>.
- Starvation: In certain scenarios, lower priority tasks may have to wait for an indefinite time if higher priority tasks keep coming.

#### **EXAMPLES:**

##### **CD:**

When you refer to a CD as being non-preemptive, it's likely in the context of how data is read from the CD. Once the reading process starts, it continues until it's complete without being interrupted by other processes. This is similar to non-preemptive scheduling in operating systems, where once a process starts, it runs to completion without being preempted by other processes<sup>1</sup>.

In the context of a CD player, once the player starts reading data (e.g., playing a song), it doesn't stop or get interrupted to do something else until it's done (e.g., the song ends). This ensures that the data is read smoothly and continuously, which is crucial for applications like audio and video playback where a smooth, uninterrupted flow of data is needed. This is because the drive's read/write head needs to physically move across the disc surface, and interrupting this process could result in data corruption

**Printers:** A printer is another example of a non-preemptive device. Once a print job starts, it must run to completion. Interrupting a print job partway through could result in wasted paper and ink, and could potentially damage the printer.

**Pen drive:** (also known as a USB flash drive) can be considered non-preemptive in the sense that once a read or write operation starts, it generally needs to complete before another operation can start<sup>1</sup>. This is because interrupting a read or write operation partway through could result in data corruption<sup>1</sup>.

### **Considerações de Custos**

Uma das principais considerações de custo entre recursos preemptíveis e não preemptíveis é o preço. Em geral, recursos preemptíveis são mais baratos que recursos não preemptíveis, pois eles não garantem a disponibilidade e a continuidade do uso. Por outro lado, recursos não preemptíveis são mais caros, pois eles garantem a exclusividade e a continuidade do uso. No entanto, o preço também depende do tipo, da demanda e do provedor do recurso, bem como do seu impacto na execução e no resultado do processo. Cada provedor de nuvem tem seus próprios nomes, preços e regras para as instâncias para recursos preemptíveis e não preemptíveis. Por exemplo, a Alibaba Cloud chama de ECS Preemptible Instances, a Amazon chama de EC2 Spot Instances, a Azure chama de Spot Virtual Machines e a Google Cloud chama de Preemptible Virtual Machines.

### **Comparação de Custos:**

Uma das principais comparações de custo entre recursos preemptíveis e não preemptíveis é o custo-benefício. Em geral, recursos preemptíveis oferecem um maior custo-benefício que recursos não preemptíveis, pois eles permitem economizar até 90% em relação aos recursos regulares, dependendo da oferta e da demanda. Por outro lado, recursos não preemptíveis oferecem um menor custo-benefício que recursos preemptíveis, pois eles cobram um preço fixo, independente da oferta e da demanda. No entanto, o custo-benefício também depende do valor, do risco e da viabilidade de cada opção, considerando os objetivos e as necessidades do negócio. Por exemplo, se o negócio precisa de um recurso que garanta a disponibilidade e a continuidade do serviço, talvez seja mais vantajoso optar por um recurso não preemptível, mesmo que seja mais caro. Por outro lado, se o negócio pode tolerar alguma interrupção ou instabilidade no serviço, talvez seja mais econômico optar por um recurso preemptível, que custa menos.

### **Estratégias para otimizar o custo:**

**Diversificação de Fornecedores:** Usar recursos preemptíveis de diferentes provedores de nuvem para reduzir o risco de interrupções.

**Monitoramento e Reinicialização:** Monitorar recursos preemptíveis e reiniciá-los quando interrompidos para minimizar interrupções em tarefas de longa duração.

**Balanceamento de Cargas:** Distribuir tarefas críticas em recursos não preemptíveis e tarefas não críticas em recursos preemptíveis, como análise de dados em lote e servidores web, respectivamente.

**Análise de Custos e Otimização Contínua:** Realizar análises regulares dos custos e ajustar a combinação de recursos com base em padrões de uso e economia potencial.

### **Tolerância a falhas:**

#### **Como são usados?**

Em ambientes de tolerância a falhas, os recursos preemptivos são usados para aumentar a confiabilidade e a disponibilidade do sistema. A ideia por trás do uso de recursos preemptivos é permitir que o sistema operacional ou pela infraestrutura de nuvem interrompa ou suspenda tarefas em andamento para lidar com situações de falha ou para priorizar tarefas mais críticas.

Ao usar recursos preemptivos, é importante implementar estratégias adequadas para lidar com a interrupção desses recursos. Uma abordagem comum é o uso de redundância, onde múltiplas instâncias dos recursos são executadas em paralelo. Se uma instância preemptiva for interrompida, outra instância pode assumir o trabalho, garantindo a continuidade do serviço.

Além disso, é possível utilizar técnicas como o balanceamento de carga e a distribuição de tarefas entre os recursos preemptivos disponíveis. Isso ajuda a garantir que a carga de trabalho seja distribuída de forma equilibrada e que nenhum recurso fique sobrecarregado.

No entanto, é importante ressaltar que o uso de recursos preemptivos também apresenta alguns desafios. Por exemplo, é necessário considerar o tempo de resposta para substituir uma instância preemptiva interrompida e garantir que o sistema seja capaz de lidar com a interrupção de recursos de forma eficiente.

Recursos preemptivos podem ser utilizados em ambientes de tolerância a falhas para aumentar a disponibilidade do sistema, mas é necessário implementar estratégias adequadas para lidar com a interrupção desses recursos e garantir a continuidade do serviço.

<https://www.inf.ufrgs.br/~taisy/disciplinas/textos/ConceitosDependabilidade.PDF>

<https://www.controle.net/faq/tolerancia-a-falhas-o-que-sao-sistemas-tolerantes-a-falha>

<https://www.cin.ufpe.br/~jvob/introducao.html>

### **Importância da escalabilidade e redundância:**

A escalabilidade e a redundância são conceitos fundamentais ao utilizar recursos preemptivos, especialmente em ambientes de computação em nuvem e em situações em que a disponibilidade e a confiabilidade são críticas. Recursos preemptivos são aqueles que podem ser interrompidos a qualquer momento pela plataforma de computação, como instâncias preemptivas em serviços de nuvem. A importância da escalabilidade e redundância nesse contexto pode ser entendida da seguinte maneira:

**Minimizar o impacto da interrupção:** Recursos preemptivos são mais econômicos, mas também menos confiáveis do que recursos dedicados. Escalabilidade permite que você distribua sua carga de trabalho em várias instâncias preemptivas, de modo que a interrupção de uma instância não prejudique significativamente o funcionamento geral do sistema.

**Garantir disponibilidade:** A redundância envolve a duplicação de recursos ou instâncias em diferentes zonas de disponibilidade ou regiões da nuvem. Isso garante que, se uma instância preemptiva for interrompida, outras estejam disponíveis para assumir a carga de trabalho, mantendo a disponibilidade do serviço.

**Flexibilidade e elasticidade:** Escalabilidade permite aumentar ou diminuir a capacidade de recursos preemptivos conforme necessário. Isso é crucial para

atender às demandas variáveis de carga de trabalho, proporcionando flexibilidade operacional.

Mitigação de riscos: Redundância e escalabilidade ajudam a mitigar o risco de interrupções não planejadas. A combinação de várias instâncias preemptivas e distribuição em diferentes zonas de disponibilidade reduz a probabilidade de todas as instâncias serem interrompidas simultaneamente.

Economia de custos: A utilização de recursos preemptivos é uma maneira eficaz de economizar em custos de computação, mas a falta de escalabilidade e redundância pode levar a interrupções e custos indesejados. Ao implementar esses conceitos, você pode maximizar a economia de custos enquanto mantém a confiabilidade.

Alta disponibilidade geográfica: A redundância geográfica, ou seja, distribuir recursos em diferentes regiões geográficas, ajuda a proteger contra interrupções regionais, desastres naturais ou outros eventos imprevistos.

Para implementar a escalabilidade e a redundância com recursos preemptivos, é importante usar práticas de arquitetura de alta disponibilidade e fazer uso de serviços de gerenciamento de recursos da nuvem, como balanceadores de carga, grupos de instâncias, armazenamento replicado e monitoramento constante. Essas estratégias garantem que sua aplicação continue funcionando de forma confiável, mesmo em ambientes com recursos preemptivos sujeitos a interrupções.

<https://wiki.inf.ufpr.br/maziero/lib/exe/fetch.php?media=socm:socm-livro.pdf>



OBS: <https://www.guru99.com/preemptive-vs-non-preemptive-scheduling.html>

exemplo de recursos preemptíveis e não preemptíveis para explicar na hora. O preemptível ele começa com um processo, para, vai pro próximo, para e volta tudo. O burst time é o tempo que precisa ser executado. Por isso que a memória é preemptível, pq tem mt requisição, daí não fica uma fila infinita.

OBS: Apenas um processo pode ser utilizado por vez. Irá se criar uma fila de processos e quando um terminar, vai verificar qual tem menor burst time e vai coloca-lo para executar.