

## RECURSOS DE GO EM THREADS

Dentro dos threads de um programa escrito em GO, várias atividades podem ocorrer, permitindo a execução simultânea e paralela de tarefas. Aqui estão alguns dos principais processos ou recursos que são comumente usados em threads GO:

1. **GOROUTINES:** São as unidades básicas de concorrência em GO. Eles são leves e facilmente gerenciáveis, permitindo que várias goroutines sejam executadas simultaneamente dentro de um único thread de sistema operacional.
2. **CANAIS:** São uma maneira de permitir a comunicação entre goroutines de forma segura e sincronizada. Eles são usados para enviar e receber dados entre rotinas e são úteis para coordenar a execução de tarefas concorrentes
3. **MUTEXES E SEMÁFOROS:** GO fornece suporte embutido para mutexes (Mutex) semáforos (Semaphore), que são usados para sincronizar ou acessar simultaneamente os recursos compartilhados e prevenir condições de corrida.
4. **WAITGROUPS:** WaitGroups é uma ferramenta útil para esperar que um grupo de goroutines conclua sua execução. Eles permitem que o programa aguarde até que todas tenham terminado antes de continuar.
5. **SELECTSTATEMENTS:** O 'select' é uma construção em Go que permite esperar em várias operações de comunicação. É comumente usado em conjunto com canais para implementar lógica de seleção não bloqueante.
6. **CONTEXTOS:** Contextos são usados para gerenciar a execução de goroutines de forma segura, especialmente em cenários em que é necessário controlar o cancelamento ou prazo de execução de operações.
7. **POOL DE GOROUTINES:** Em certos casos, pode ser útil usar pools de goroutines para reutilizar goroutines em vez de criar e destruir novas goroutines repetidamente.

Esses são alguns dos processos e recursos-chave que podem ser encontrados dentro dos threads em programas escritos em GO.