

Revisão do conteúdo

Transcrição

No último capítulo já vimos como utilizar Threads. Foi necessário criar uma instância da classe `Thread` e passar uma tarefa. Essa tarefa, por sua vez, é um objeto do tipo `Runnable`. Quando criamos o objeto `Thread`, passamos a tarefa através do construtor da classe `Thread`. Por fim devemos chamar o método `start()` pra inicializar o Thread.

Mapeamento dos Threads para SO

Reparem que não foi necessário utilizar alguma biblioteca extra para trabalhar com threads. Quando o Java foi lançado, isso foi um grande diferencial pois a JVM já veio com suporte nativo para trabalhar com threads. Além disso, a máquina virtual mapeia os threads dela para threads ou processos do sistema operacional. Ou seja, se seu sistema operacional dá suporte à execução em paralelo, muito provável que podemos realmente executar código em paralelo. Claro que isso também depende da implementação da JVM em si, mas hoje em dia, pode ter certeza, se você usa a máquina virtual da Oracle ou da IBM, que essas máquinas virtuais mapeiam os threads para o sistema operacional. Ou seja, com Java em mãos, nada impede de realmente aproveitar a execução em paralelo.

