

Arthur Leite Padilha Filho

Uso de Java para Sistemas de Tempo real, vale a pena?

Não se for considerada a versão tradicional, já que possui vários fatores que trazem consigo um indeterminismo temporal inaceitável em sistemas de tempo real hard, como o sistema de coleta de lixo e a falta de um modelo de threads baseado em prioridades.

Para tentar resolver esse impedimento, foi introduzido um conjunto de especificações para a utilização de Java em sistemas de tempo real: a Real Time Specification for Java.

A RTSJ é composta por um conjunto de especificações de comportamento para a linguagem Java, permitindo que ela passe a apresentar um comportamento determinístico essencial ao desenvolvimento de sistemas de tempo real.

RTSJ aborda várias áreas problemáticas, incluindo a programação, gerenciamento de memória, threading, sincronização, tempo, relógios e manipulação de eventos assíncrona.

Passa a implementar a classe RawMemory para acesso direto a memória. Áreas da memória física podem ser definidas com determinadas características e usadas para alocação de objetos.

Já o sistema de threads de tempo real passa a sofrer uma subdivisão em Realtime Thread e NoHeapRealtimeThread. A primeira é mais aplicada em sistemas de tempo real soft e a segunda nos sistemas hard, uma vez que não usa a heap e nem o coletor de lixo, contudo esta segunda requer uma maior experiência do desenvolvedor de sistemas de tempo real.

O modelo de gerência de memória possui duas inovações principais: a primeira trata da inclusão de scoped memory que é representada por um objeto que é instanciado no código e então utilizado para criar outros objetos dentro. As threads que executam dentro desse espaço de scoped memory governam o tempo de vida desse objetos, quando não há threads no espaço de scoped memory, todos os objetos que estão dentro desse espaço são desalocados imediatamente; a segunda, denominada immortal memory, apresenta a característica de que, objetos criados nunca estão sujeitos à coleta de lixo e persistem até o fim da aplicação, portando, esse segundo tipo de memória deve ser utilizado com muito cuidado e moderação.

A RTSJ prevê áreas de memória não sujeitas à coleta de lixo. Essas áreas passam a utilizar o gerenciamento de memória baseado em região.

A conclusão que se obtém é que, a partir do lançamento da RTSJ, os desenvolvedores de sistemas de tempo real passaram a ter mais uma opção atraente, respaldada pela utilização desse sistema pela Marinha dos EUA.

Fontes:

**Real-time Java, Part 1: Using Java code to program real-time systems.** Disponível em: <http://www.ibm.com/developerworks/library/j-rtj1/index.html>. Acessado em: 30/09/2015.

**Real Time Java.** Disponível em: [https://en.wikipedia.org/wiki/Real\\_time\\_Java](https://en.wikipedia.org/wiki/Real_time_Java). Acessado em: 30/09/2015.

**Vale a pena usar Java em aplicações de tempo-real?.** Disponível em: <https://determinismotemporal.wordpress.com/2008/06/10/java-e-sistemas-de-tempo-real-vale-a-pena-utilizar/>. Acessado em: 27/09/2015.

**Java e Sistemas de Tempo Real: Vale a pena utilizar?.** Disponível em: <http://sisttemporeal.blogspot.com.br/2008/08/vale-pena-usar-java-em-aplicaes-de.html>. Acessado em: 27/09/2015.