
1: Cross-compiler

(1) O que é cross compilação (cross-compiler) ?

Como é explicado em [2], cross compilação é o ato de desenvolver um binário de um programa em uma plataforma que irá rodar em uma outra, a primeira plataforma sendo a de desenvolvimento e a segunda, a plataforma destinatária. Para que esse programa seja feito, é necessário um cross compiler, compilador capaz de compilar um código para diversas plataformas, por exemplo: um compilador que roda em Windows e gerá um código para Android.

2: Embarcados

(1) O que é um RTOS, descreva uma utilizações.

Segundo [6], RTOS é uma sigla que significa Real Time Operating System, ou seja, um sistema operacional onde o scheduler, programa responsável por cuidar dos intervalos entre multi-threads, tenha um tempo de resposta pré-definido. Esse tipo de sistema operacional é muito comum em sistemas embarcados, pois muitas vezes os sistemas embarcados acabam tendo necessidades em tempo real, sendo assim necessário um sistema operacional que consiga otimizar as multi-tarefas para essas necessidades.

(3) O que é um DSP ? O que difere de um microcontrolador ?

Como é dito em [5], DSP é uma sigla que significa Digital Signal Processors. DSPs são microprocessadores especializados em processamento de sinais digitais. Os DSPs são comparados a Microcontroladores em [4]. Por serem especializados nessa área, acabam tendo uma velocidade de processamento muito maior que microcontroladores para essas tarefas. Microcontroladores são circuitos integrados menos especializados, com processadores, memória, periféricos. Já os DSPs são simplesmente processadores que apenas alteram sinais digitais.

3: C

(1) Qual a diferença entre C e C++ ?

Segundo as comparações em [3] e [1], C e C++ são linguagens muito parecidas em certos aspectos, sendo o C++ uma versão posterior ao C. A diferença mais notável entre as duas linguagens é o fato do C ser uma linguagem estrutural, já C++, uma linguagem orientada a objetos, ou seja, o C++ é considerado uma versão atualizada e orientada a objetos do C. Por ser uma versão do C com mais funcionalidades, a maioria dos compiladores de C++ também compilam C. Mesmo sendo uma versão de C atualizada, o C++ é considerado uma linguagem de mais alto nível que o C, tendo assim diferentes aplicações.

4: Paralelismo vs Concorrência

(1) Analise o texto a seguir extraído do livro : “Introduction to Embedded Systems - A CyberPhysical Systems Approach (7.2.1)” e faça uma resenha sobre paralelismo e concorrência

O texto começa explicando brevemente o conceito de paralelismo e concorrência, dizendo que um programa concorrente seria aquele que diferentes partes do programa são executadas ao mesmo tempo conceitualmente, já um programa paralelo seria aquele em que suas partes são executadas fisicamente ao mesmo tempo, ou seja em hardwares distintos.

Após explicar brevemente esses dois conceitos, o autor analisa suas aplicações em duas diferentes linguagens, o C e o Java. É apresentado um novo conceito chamado de linguagem imperativa, uma linguagem que é executada como uma sequência de operações. o C é uma dessas linguagens, e não suporta nativamente programas concorrentes. Já o Java tem suporte nativo para esse tipo de execução.

Em seguida é mostrado um exemplo de código C para explicar um programa dependente e independente, acrescentando que microprocessadores atuais suportam execução em paralelo, analisando o código para definir se suas instruções são independentes, se sim, as executam de forma paralela. O objetivo disso é melhorar a performance, com a ideia de que terminar uma tarefa antes é melhor do que terminá-la depois.

O autor fala então na aplicação dessa ideia em sistemas embarcados, dizendo que concorrência tem um papel muito mais importante do que melhorar a performance, controlando o tempo de saída de cada tarefa do sistema, pois a ordem de tarefas no mundo físico é essencial.

Ele termina o capítulo retomando a ideia que, em aplicações como sistemas embarcados, não há sempre a necessidade que programas sejam executados ao mesmo tempo, somente a necessidade que eles sejam executados rapidamente, sendo assim muito interessante combinar concorrência para ordenar e organizar as tarefas, e paralelismo no hardware para otimização do programa.

Referências

- [1] *10 Major Differences Between C And C++ 344*. Disponível em <http://durofy.com/10-major-differences-between-c-and-c/>, Acessado em 12-02-2017.
- [2] *Cross-Compilation*. Disponível em https://www.gnu.org/software/automake/manual/html_node/Cross_002dCompilation.html, Acessado em 12-02-2017.
- [3] *Difference Between C and C++ Programming Languages*. Disponível em <http://techwelkin.com/difference-between-c-and-c-plus-plus>, Acessado em 12-02-2017.
- [4] *Difference between DSP and Microcontroller*. Disponível em <http://www.differencebetween.info/difference-between-dsp-and-microcontroller>, Acessado em 12-02-2017.
- [5] *PROCESSADOR DIGITAL DE SINAIS - DSP*. Disponível em <http://www.cbpf.br/~rastuto/>, Acessado em 12-02-2017.
- [6] *What is An RTOS?* Disponível em <https://www.arduino.cc/en/Products/Compare>, Acessado em 08-02-2017.