



03&04 – Pesquisa: Fluxo de Projeto

Rafael Corsi

rafael.corsi@insper.edu.br

8 de fevereiro de 2017

Matheus Marotzke

Engenharia da Computação – INSPER – 2017

INTRODUÇÃO

A atividade inicial da disciplina de Computação Embarcada consiste em uma pesquisa sobre diversos aspectos relacionados a Compilers, Embarcados, C e Paralelismo vs Concorrência.

CROSS-COMPILER

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

O Cross-compiler ou (compilador cruzado) nada mais é do que um compilador capaz de gerar um executável para uma plataforma diferente da qual a compilação foi executada. [1] Um exemplo seria um sistema que utiliza um OS Linux criando aplicações e executáveis Android, com Android Studio por exemplo, esse seria um cross-compiler.

EMBARCADOS

- **O que é um RTOS, descreva uma utilização.**

O RTOS (Real-Time Operating System) é um Sistema operacional desenhado para execução de tarefas em tempo real, ou seja, seu Scheduler é programado com uma capacidade de priorizar tarefas de maneira preditiva a fim de executar tarefas em tempo real. Diferentemente do Unix ou Windows em que o sistema garante uma divisão justa entre tarefas, gerando uma falsa impressão de paralelismo de processos. Muitas vezes são utilizados em dispositivos embarcados que possuem uma demanda por realtime o RTOS, como alternativa aos OS padrões. [2]

- **O que é desenvolvimento de projetos em V (Modelo V)?**

O Modelo de desenvolvimento de projetos V prevê que durante o processo de implementação e desenvolvimento de soluções, deve-se fazer testes e comparações com as especificações e padrões estabelecidos na definição do projeto, comparados ao trecho de integração, após sua implementação [3]

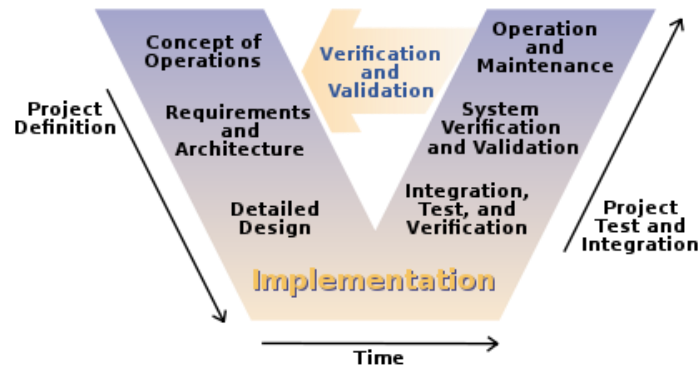


Imagem 1 - Modelo de Desenvolvimento V

- **O que é um DSP ? O que difere de um Microcontrolador?**

De maneira geral o DSP (Digital signal processor) é um microprocessador, comumente utilizado para medir, filtrar e comprimir dados analógicos. Assim como um microcontrolador contém seu próprio, o DSP também, contudo sua principal diferença é sua otimização para processamento de sinais digitais. [4]

C

- **Descreva a funcionalidade do:**

Antes mesmo do compilador ser acionado, o código C passa pelo processo de pré-processamento. Nesta etapa, o programa agrupa linhas continuadas por \; substitui temos usados com #define; retira comentários; entre outras atribuições antes da compilação. [5]

- **Compilador C**

A etapa de compilação tem como tarefa transformar o código C já pre-processado, em código assembly. [5]

- **Assembler**

Já neste estágio o código recém gerado em assembly será convertido para código de máquina.

- **Linker**

Na etapa final de linking, o programa arruma as bibliotecas e arquivos que ainda não são acessíveis do programa recém gerado. Em um código que utiliza *printf* a função de *puts* deve ser linkada ao programa antes de sua execução.

- **Qual a diferença entre C e C++?**

Para começar com as diferenças podemos apontar a anedota utilizada na nomeação do C++, o símbolo '++' na linguagem C, significa $C = C + 1$, ou seja, adicionar 1 à versão atual do C, criando uma nova versão do clássico C, melhorada.

A linguagem C, criada anteriormente ao C++ está contida no C++ e pode ser rodada por compiladores dessa linguagem.

O C++ é muitas vezes chamado de C com classes detém a possibilidade de orientação à objetos. [6]

PARALELISMO VS CONCORRÊNCIA

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

Ambos os conceitos de paralelismo e concorrência estão relacionados à necessidade de execução de programas simultaneamente.

Seja através do paralelismo em que o processamento simultâneo ocorre em dois *cores* distintos, ou através da concorrência ambos são executados alternadamente em um único *core* criando o efeito, com a alternância alta, de que estão executando simultaneamente.

Em linguagens de chamadas de **imperativas**, ou seja, não concorrente, o programa escrito tem uma ordem em que deve ser executado a fim de realizar uma tarefa. Quando buscamos a ferramenta da simultaneidade de execução saímos do padrão de várias linguagens e utilizamos **threads**.

Se considerarmos sistemas embarcados, por muitas vezes a simultaneidade dos acontecimentos de interação hardware e mundo físico ocorrem simultaneamente. Esse fenômeno exige que sistemas embarcados tenham seu software de leitura e atuação trabalhando de maneira otimizada simultaneamente.

Atualmente, linguagens, hardwares e SOs mais modernos estão cada vez mais preparados para o uso de paralelismo e concorrência. Com *multitasking operating system* e estruturas de análise de fluxo de dados, pode-se cada vez mais otimizar programas executando simultaneamente.

GITHUB

<https://github.com/MatheusDMD/EmbeddedComputing>

Bibliografia

- [1] Wikipédia, Enciclopédia Livre, "Cross compiler," Wikipédia, Enciclopédia Livre, [Online]. Available: https://en.wikipedia.org/wiki/Cross_compiler.
- [2] Wikipédia, Enciclopédia Livre, "Real-time operating system," Wikipédia, Enciclopédia Livre, [Online]. Available: https://en.wikipedia.org/wiki/Real-time_operating_system.
- [3] Wikipédia, "Modelo V," Wikipédia, [Online]. Available: https://pt.wikipedia.org/wiki/Modelo_V.
- [4] Wikipedia, Enciclopédia Livre, "Digital Signal Processor," Wikipedia, Enciclopédia Livre, [Online]. Available: https://en.wikipedia.org/wiki/Digital_signal_processor.
- [5] C. Landsson, "the-four-stages-of-compiling-a-c-program," [Online]. Available: <https://calleerlandsson.com/the-four-stages-of-compiling-a-c-program/>.
- [6] cs-Fundamentals.com, "What is the difference between C and C++?," cs-Fundamentals.com, [Online]. Available: <http://cs-fundamentals.com/tech-interview/c/difference-between-c-and-cpp.php>.