

Computação Embarcada-Fluxo de projeto embarcados

Felipe Frid Buniac

February 13, 2017

1 Cross-compiler

1.1 Item I- O que é cross compilação (cross-compiler)

Um cross compiler é um compilador capaz de criar código executável para uma plataforma diferente daquela em que o compilador está sendo executado [1, 2, 3].

2 Embarcados

2.1 Item I- O que é um RTOS, descreva uma utilizações.

A maioria dos sistemas operacionais parece permitir que vários programas sejam executados ao mesmo tempo. Isso é chamado de multitarefa. Na realidade, cada núcleo de processador só pode estar executando um único segmento de execução em qualquer ponto do tempo. Uma parte do sistema operacional chamada de planejador é responsável por decidir qual programa executar e fornece a ilusão de execução simultânea ao alternar rapidamente entre cada programa. RTOS (real-time operating system) é um sistema operacional que foi desenvolvido para aplicações em tempo real e normalmente usado para aplicações incorporadas em um computador de propósito especial com um conjunto limitado de funções determinadas pelo design do hardware.[4, 5, 6, 7]

RTOS ajuda a resolver os problemas comuns relacionados à programação de microcontroladores, como a necessidade de executar múltiplas tarefas simultaneamente, a resposta rápida a eventos de alta prioridade, o gerenciamento de recursos de hardware da MCU etc. Em MCUs de 16 bits ou 32 bits, RTOS são usados com frequência. Em sistemas menores de 8 bits não é tão comum pois esses sistemas têm memória e poder de CPU limitados , portanto, é mais eficiente escrever o programa necessário sem o uso RTOS nesse caso.[8]

2.2 Item II- O que é desenvolvimento de projetos em V (Modelo V).

O modelo V é uma variante simples do modelo de cascata¹ tradicional de desenvolvimento de sistema ou software, este, baseia-se no modelo de cachoeira, enfatizando a verificação e validação. O modelo V toma a metade inferior do modelo de cachoeira e o reverte para cima na forma de um V, de modo que as atividades à direita verificam ou validam os produtos de trabalho da atividade à esquerda. Mais especificamente, o lado esquerdo do V representa as atividades de análise que decompõem as necessidades dos usuários em pequenas peças gerenciáveis, enquanto o lado direito do V mostra as atividades de síntese correspondentes que agregam (e testam) essas peças em um sistema que atende as necessidades dos usuários[9].

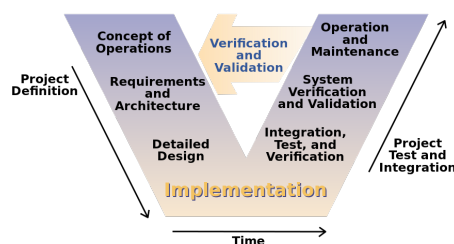


Figure 1: Esquemático em inglês do modelo em V [20]

¹ - O modelo de cachoeira é um processo de projeto sequencial (não-iterativo), usado em processos de desenvolvimento de software, em que o progresso é visto como fluindo de forma constante para baixo (como uma cachoeira) através das fases de concepção, iniciação, análise, Produção / implementação e manutenção[10].

3 C

3.1 Item I- Descreva a funcionalidade do:Compilador C; Assembler; Linker

Compiler C

Um equívoco comumente dito é de que os computadores entendem código. Isso não é verdade. Os computadores só compreendem o código de máquina, isto é, o código que é montado para a arquitetura do processador do computador. Alguns programas vêm em versões de 32 bits e 64 bits, e aqueles que usam um sistema operacional de 32 bits não podem executar programas de 64 bits. Os compiladores transformam o código que o programador escreve em código de máquina. No entanto, nem todas as linguagens precisam ser compiladas. Algumas línguas são conhecidas como linguagens interpretadas, estas são linguagens que são compiladas em tempo de execução. Linguagens interpretadas ainda precisam ser compiladas em código de máquina, mas isso ocorre quando o programa está sendo executado. Uma vez que o código de uma linguagem compilada é transformado em um programa compilado, não é necessário que o usuário final desse programa tenha o compilador em sua máquina. No entanto, o usuário de um programa escrito em uma linguagem interpretada é obrigado a ter o intérprete em sua máquina[11].

O compilador C traduz o código fonte para código assembly. O código-fonte é recebido do pré-processador.[12]

Assembler

Um assembler é um programa que transforma instruções básicas do computador em um padrão de bits que o processador do computador pode usar para executar suas operações básicas. Algumas pessoas chamam essas instruções de linguagem assembler e outros usam o termo linguagem assembly [13].

Linker

Na computação, um linker ou editor de links é um programa de computador que leva um ou mais arquivos de objetos gerados por um compilador e os combina em um único arquivo executável, arquivo de biblioteca ou outro arquivo de objeto [14].

3.2 Item II- Qual a diferença entre C e C++

A principal diferença entre C e C++ é que C é uma linguagem de programação procedural e não suporta classes e objetos, enquanto que o C++ é uma combinação de linguagem de programação orientada a procedimentos e objetos; Portanto, C++ pode ser chamada de uma linguagem híbrida.

Mais informações sobre as diferenças entre estas linguagens podem ser encontradas em[15, 16, 17, 18].

4 Paralelismo vs Concorrência

O texto parte da afirmação da importância da Concorrência (Concurrency) para sistemas embutidos/embarcados (Embedded Systems), e a partir disto compara o paralelismo com a concorrência em programas de computação.

Um programa concorrente (concurrent) é um programa que conceitualmente executa diferentes partes do programa de forma simultânea. Um programa é definido como paralelo (parallel) se partes diferentes do programa fisicamente executam de forma simultânea em hardware distinto. Programas não concorrentes, especificam uma sequência de instruções a serem executadas, uma linguagem de computação que se comporta de tal forma é chamada de imperativa. A linguagem

C é definida como imperativa, pois ao utilizar C para escrever programas concorrentes utilizamos uma biblioteca de threads- biblioteca que utiliza instalações que não são fornecidas por C, mas sim fornecidas pelo sistema operacional e / ou pelo hardware. Toda execução de um programa em uma linguagem imperativa deve se comportar como se as instruções fossem executadas exatamente na sequência especificada para definir a execução como correta, no entanto, muitas vezes é possível executar instruções em paralelo ou em uma ordem diferente da especificada pelo programa e ainda obter um comportamento que corresponda ao que teria acontecido se tivessem sido executados em sequência.

Muitos microprocessadores hoje suportam a execução paralela, usando fluxos de instrução multi-issue ou arquiteturas VLIW (very large instruction word) podendo executar instruções independentes simultaneamente. O hardware analisa instruções buscando dependências, quando não há nenhuma dependência, executa mais de uma instrução de cada vez. Um compilador pode analisar as dependências entre operações em um programa e produzir código paralelo, se a máquina de destino o suportar. Esta análise é chamada análise de fluxo de dados. Este código paralelo é utilizado em máquinas avançadas de VLIW, que tem instruções em nível de assembly que especificam várias operações a serem executadas em conjunto. A análise neste caso é feita no nível de linguagem assembly, não no nível das linhas de C. O objetivo é aumentar performance e aumentar a velocidade dos programas.

Em sistemas embutidos a concorrência tem um papel mais importante do que aumento de performance pois a simultaneidade é fundamental em processos físicos. No mundo físico muitas atividades progridem ao mesmo tempo e um programa embutido deve monitorar e reagir de forma simultânea à diferentes estímulos, estas ações realizadas no mundo físico precisam ser feitas no momento certo para obter sucesso.

Assim como os programas imperativos, os programas concorrentes podem ser executados sequencialmente ou em paralelo. A execução sequencial de programas concorrentes são feitas em sistemas operacionais de multitarefas, responsável por intercalar a execução de múltiplas tarefas em um único fluxo sequencial de instruções.

A partir do texto tiramos a conclusão de que aos conceitos de paralelismo e concorrência estão relacionados a simultaneidade na execução de programas. Do ponto de vista do programador, a concorrência surge como uma consequência do hardware projetado para melhorar o desempenho, não como uma consequência do problema da aplicação sendo resolvido, ou seja, não necessariamente atividades devem acontecer simultaneamente e sim devem acontecer rapidamente. Muitas vezes existe a combinação destas duas para diversas aplicações. Portanto nem sempre a mesma técnica é utilizada para diferentes soluções como em programas simultâneos ou em paralelo.

References

- [1] <http://programming4.us/desktop/440.aspx>
- [2] https://en.wikipedia.org/wiki/Cross_compiler
- [3] <http://www.pcmag.com/encyclopedia/term/40491/cross-compiler>
- [4] https://en.wikipedia.org/wiki/Real-time_operating_system
- [5] http://rtos.com/PDFs/What_Is_An_RTOS_and_Why_Use_One_Embedded.com_.pdf
- [6] <http://www.webopedia.com/TERM/R/RTOS.html>
- [7] <http://www.freertos.org/about-RTOS.html>
- [8] <http://www.naun.org/main/NAUN/ijmmas/20-566.pdf>
- [9] https://insights.sei.cmu.edu/sei_blog/2013/11/using-v-models-for-testing.html
- [10] https://en.wikipedia.org/wiki/Waterfall_model
- [11] <https://www.reference.com/technology/functions-compiler-395080178618f9d4>
- [12] <https://users.cs.cf.ac.uk/Dave.Marshall/C/node3.html#SECTION00322000000000000000>
- [13] <http://searchdatacenter.techtarget.com/definition/assembler>

- [14] [https://en.wikipedia.org/wiki/Linker_\(computing\)](https://en.wikipedia.org/wiki/Linker_(computing))
- [15] <http://cs-fundamentals.com/tech-interview/c/difference-between-c-and-cpp.php>
- [16] https://pt.wikibooks.org/wiki/Programar_em_C%2B%2B/Diferen%C3%A7as_entre_C_e_C%2B%2B
- [17] <http://softwareengineering.stackexchange.com/questions/16390/what-are-the-fundamental-differences-between-c-and-c>
- [18] <http://techwelkin.com/difference-between-c-and-c-plus-plus>
- [19] <http://stackoverflow.com/questions/3996651/what-is-compiler-linker-loader>
- [20] [https://en.wikipedia.org/wiki/V-Model_\(software_development\)](https://en.wikipedia.org/wiki/V-Model_(software_development))

5 GITHUB

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