

# Como ler programas CWEB

Tradução livre  
de parte do capítulo 4 (“How to read CWEB programs”)  
do *Stanford GraphBase* de D. E. Knuth

Fevereiro de 2000

Um programa CWEB é um programa C que foi cortado em pedaços e rearranjado de modo que um ser humano possa entendê-lo mais facilmente. Pode-se dizer, reciprocamente, que um programa C nada mais é que um programa CWEB rearranjado de modo que possa ser entendido por um computador.

Os próximos parágrafos explicam como um programa CWEB pode ser transformado mecanicamente em um programa C por um processo chamado “tangling” [“to tangle” = entrelaçar, embaraçar].

**Seções.** Um programa CWEB é uma seqüência de seções numeradas. [As seções são chamadas *blocos* (“chunks”) na versão L<sup>A</sup>T<sub>E</sub>X do CWEB.] Um programa completo pode ser encarado como uma teia (“web”) ou rede, que consiste em pequenos pedaços de código e interconexões entre os pedaços. Para entender o complexo todo, basta entender cada pedaço e as relações entre pedaços vizinhos.

Cada seção de um programa CWEB começa com um comentário sobre os objetivos ou sobre algum aspecto importante da seção. Tais comentários são escritos em português, de maneira informal. A seção termina com uma parte formal, expressa em linguagem C; esta é a parte que realmente conta, e supõe-se que o código formal faz exatamente o que os comentários informais sugerem.

Entre os comentários introdutórios e o código C pode haver também uma ou mais definições de macros, que serão explicadas logo abaixo. Assim, cada seção tem três partes:

- comentário informal,
- definições (`#define`) de macros,
- código C,

e as partes sempre estão nesta ordem. Qualquer uma das partes pode estar vazia: algumas seções não têm comentário introdutório, outras não têm definições de macros, e outras não têm código C.

**Texto de substituição.** Algumas seções têm *nome* (“named section”); outras são *anônimas* (“unnamed section”). [Não estamos falando aqui do *título* que aparece em negrito no início da parte de comentários de algumas seções; esse título é independente do nome da seção.] O código C de uma seção com nome começa com

⟨Nome da seção⟩ ≡

e é seguido por um *texto de substituição* (“replacement text”). Por exemplo, o código C da seção 23 poderia ser

```

⟨Limpe os vetores 23⟩ ≡
    for (k = 1; k ≤ n; k++) {
        a[k] = 0 ;
        str[k] = "" ;
    }

```

e isso significa que o nome da seção é ‘Limpe os vetores’ enquanto o texto de substituição é ‘for (k = 1; k <= n; k++) { a[k] = 0; str[k] = ""; }’.

**Tipografia.** A descrição CWEB dos textos de substituição é formatada de uma maneira especial. Isso torna o código C mais fácil de ler. Por exemplo, palavras reservadas como ‘for’ são escritas em negrito; identificadores como ‘k’ e ‘str’ são escritas em itálico; o conteúdo de cadeias de caracteres (“strings”) é escrito em fonte máquina-de-escrever. Os seguintes símbolos especiais são usados para melhorar a legibilidade:

≤	significa	<= ,
≥	significa	>= ,
≠	significa	!= ,
≡	significa	== ,
∧	significa	&& (E lógico),
∨	significa	(OU lógico),
¬	significa	! (NÃO lógico),
→	significa	-> ,
&	significa	& (E bit-a-bit),
	significa	(OU INCLUSIVO bit-a-bit),
⊕	significa	^ (OU EXCLUSIVO bit-a-bit),
~	significa	~ (NEGAÇÃO bit-a-bit),
Λ	significa	NULL .

[Eu também gosto de usar ▷ para representar ->.] Constantes numéricas têm suas próprias convenções tipográficas. Uma constante **long**, escrita como 3L em um programa C, aparece como 3<sub>L</sub> na tipografia CWEB. Uma constante octal, escrita como 077, aparece como °77 na documentação. Uma constante hexadecimal, escrita como 0x7ff, aparece como #7ff. Uma constante ponto flutuante, escrita como 1.4E6, aparece como 1.4 · 10<sup>6</sup>.

**Seções citadas por outras seções.** Quando o nome de uma seção aparece em outra seção, ele deve ser substituído pelo correspondente texto de substituição. Por exemplo, suponha que o código C da seção 20 seja

```
⟨Inicialize as estruturas de dados 20⟩ ≡  
    soma = 0;  
    ⟨Limpe os vetores 23⟩;
```

Isso significa que o texto de substituição para ‘Limpe os vetores’ deve ser inserido, no lugar indicado, no texto de substituição de ‘Inicialize as estruturas de dados’. O nome de uma seção é geralmente um pequeno resumo d’o que a seção faz. Assim, os detalhes do código não vão distrair sua atenção, a menos que você precise conhecê-los. Se você precisa de todos os detalhes, o ‘23’ em ‘⟨Limpe os vetores 23⟩’ permite localizar o texto de substituição subsidiário. A propósito, a seção 23 de nosso exemplo terá uma nota de rodapé que dirá ‘Este código é usado na seção 20’; assim, você pode navegar facilmente do texto de substituição para o seu contexto e vice-versa.

**Juntando os pedaços.** O processo de conversão de um programa CWEB em um programa C é bastante simples, pelo menos conceitualmente. Primeiro, junte todo o código C que aparece em seções anônimas, preservando a ordem relativa dos vários pedaços. Depois, substitua cada seção com nome pelo correspondente texto de substituição. Continue fazendo isso até obter código C puro.

Pode-se dar o mesmo nome a várias seções diferentes. Continuando nosso exemplo, suponha que o código C da seção 30 seja

```
⟨Inicialize as estruturas de dados 20⟩ +≡  
    arq_fonte = fopen (nome_do_arq_fonte, "r") ;
```

Aqui, ‘+≡’ indica que o nome ‘Inicialize as estruturas de dados’ já apareceu antes; o código C que segue ‘+≡’ será *acrescentado* ao texto de substituição anterior daquele nome. Assim, se ‘Inicialize as estruturas de dados’ é definido apenas pelas seções 20 e 30, o texto de substituição completo para ⟨Inicialize as estruturas de dados 20⟩ será

```
    soma = 0 ;  
    ⟨Limpe os vetores 23⟩ ;  
    arq_fonte = fopen (nome_do_arq_fonte, "r") ;
```

A seção 20 também conterà a nota de rodapé ‘Veja também a seção 30’. Desta forma, diferentes partes da inicialização do programa podem ser tratadas quando o leitor estiver preparado para entendê-las, e não todas de uma só vez.

**Macros.** As definições de macros, que podem aparecer imediatamente antes do código C de qualquer seção, são reunidas num ponto chamado ⟨Definições para o pré-processador⟩, que normalmente vem logo depois das declarações `#include` no início do programa. (Se o programa não indicar especificamente a posição das ⟨Definições para o pré-processador⟩,

as macros reunidas serão simplesmente colocadas no início da saída emaranhada (“tangled”), antes de qualquer outro material C.)

A definição de uma macro pode ocupar várias linhas consecutivas; quando o código C é extraído do programa *CWEB*, essas linhas são devidamente emendadas.

**Geração de arquivos auxiliares.** Às vezes o nome de uma seção é o nome de um arquivo, escrito em fonte máquina-de-escrever, como por exemplo `<gb.save.h 1>`. Isso significa que o texto de substituição daquela seção será gravado no arquivo especificado, em vez de ser integrado ao programa C propriamente dito. Portanto, um só programa *CWEB* pode descrever vários arquivos correlatos.