

The CWEB System of Structured Documentation

(Version 3.64 — February 2002)

Donald E. Knuth and Silvio Levy

T_EX is a trademark of the American Mathematical Society.
Acrobat Reader is a trademark of Adobe Systems Incorporated.

The printed form of this manual is copyright © 1994 by Addison-Wesley Publishing Company, Inc. All rights reserved.

The electronic form is copyright © 1987, 1990, 1993, 2000 by Silvio Levy and Donald E. Knuth.

Permission is granted to make and distribute verbatim copies of the electronic form of this document provided that the electronic copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of the electronic form of this document under the conditions for verbatim copying, provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Individuals may make copies of the documentation from the electronic files for their own personal use.

Internet page <http://www-cs-faculty.stanford.edu/~knuth/cweb.html> contains current info about CWEB and related topics.

구조적 문서화를 위한 CWEB 시스템

Donald E. Knuth, Silvio Levy

이 문서는 Don Knuth의 WEB 시스템에, Silvio Levy가 C 언어를 적용한 CWEB 시스템에 대한 글입니다. CWEB이 처음 만들어진 1987년 이후로, 이 시스템은 Knuth와 Levy에 의해서 다양한 방법으로 수정되고 보완되어서 현재는 거의 완벽한 시스템이 되었다고 할 수 있습니다. 하지만, 그래도 여전히 버그리포트, 개선 사항, 혹은 이 시스템에 도움 될 어떠한 사항들이 있다면, Levy(levy@math.berkeley.edu)에게 보내시거나 제안하면 됩니다.

CWEB과 WEB 시스템은 같은 철학을 공유하고 (근본적으로는) 동일한 문법을 사용하고 있습니다. 따라서, 이미 Knuth의 “The WEB System of Structured Documentation”이란 글에 익숙한 독자라면, 이 글의 상당 부분을 빠르게 읽으며 넘어 갈 수 있을 것입니다. 어떤 면에서는 CWEB 시스템은 WEB 시스템을 단순화한 버전이기도 합니다. 예를들면, Pascal 언어를 기반으로 했던 WEB은 Pascal 언어에는 없는 전처리기능을 필요했던 반면, C 언어를 기반으로 한 CWEB 시스템은 C 언어 자체가 전처리기능을 가지고 있으므로, CWEB 시스템에는 그러한 기능을 가질 필요가 없습니다. 비슷한 맥락으로, WEB 시스템은 8진수와 16진수를 표시하기 위해서 @'77 와 @'3f와 같은 별도의 표기 방식이 필요했지만, CWEB에서는 그러한 수를 나타내기 위해서 별도의 표기 방식의 필요 없이 C 언어에서 사용하는 그대로 077나 0x3f 처럼 그냥 사용하면 됩니다. 이처럼 CWEB은 WEB의 다른 모든 기능들을 가지고 있으면서, 새로운 기능이 더해진 시스템이라고 할 수 있습니다.

CWEB 시스템의 개발에 있어서 많은 제안과 비평을 해준 모든 분들께 감사의 말을 전합니다. 특히, 코드 작성에 기여한 Steve Avery, Nelson Beebe, Hans-Hermann Bode, Klaus Guntermann, Norman Ramsey, Joachim Schmitter, Saroj Mahapatra, 그리고 이 사용설명서 작성에 많은 도움을 준 Cameron Smith에 특별히 감사드립니다. Ramsey는 그의 SPIDER 시스템[*Communications of the ACM* 32 (1989), 1051–1055]을 통하여, Pascal이나 C 언어 이외의 다른 프로그래밍 언어를 이용하는 문학적 프로그램을 가능하게 해 주었습니다. Knuth의 책인 *Literate Programming*(1992)은 문학적 프로그래밍과 관련된 많은 참고 문헌들과 초기 작업에 대한 풍부하고 흥미로운 내용을 담고 있습니다. Bode, Schmitter, Mahapatra 등은 CWEB에서 C 언어 뿐만아니라, C++ 언어도 사용할 수 있도록 하여서, 여러분은 필요하다면, 이 글을 C에 대한 글이 아닌 C++에 대한 글이라고 간주해도 무방합니다.

소개

CWEB의 배경이 되는 철학은, 자신이 작성하고 있는 프로그램에 대해서 가능한 최상의 문서를 제공하고자 하는 프로그래머는 문서 조판을 위해서 TeX과 같은 언어와 프로그래밍을 위해서 C 언어와 같은 두 가지 도구가 동시에 필요하다는 것입니다. 이 두 가지 언어 중 어느 한 가지만 가지고는 최상의 문서를 제공할 수 없습니다. 하지만 이 두 가지 언어를 적절히 조합하면, 각각의 언어가 제공할 수 있는 기능보다 더욱 유용한 기능을 갖춘 시스템을 구성할 수 있습니다.

소프트웨어 프로그램의 구조는 내부적으로 많은 작은 부분들이 서로 긴밀히 연결되어 있는 “웹(web)”이라고 생각할 수 있습니다. 그러한 프로그램을 문서화하기 하기 위해서 그 웹을 구성하는 각각의 개별적인 부분들을 설명하고, 그 부분들이 그의 이웃들과 어떠한 연관이 있는지를 설명하는 것이 좋은 방법일 것입니다. TeX으로 제공되는 조판 도구는 각 부분을 읽기 좋게 해 줌으로써 그의 구조를 쉽게 파악 할 수 있게 해주고, C 언어로 제공되는 프로그래밍 도구는 알고리즘들을 보다 수학적으로 명확하게 기술할 수 있게 해줍니다. 이러한 두 가지 도구를 조합함으로써, 소프트웨어의 복잡한 구조를 쉽게 파악 할 수 있도록하는 프로그래밍이 가능합니다. 동시에 그 문서화된 프로그램은 기계적으로 소프트웨어 시스템으로 변환되어 실행 시스템을 만들 수 있습니다.

CWEB 시스템은 CWEAVE 와 CTANGLE 이라고 불리는 두 개의 프로그램으로 구성되어 있습니다. CWEB 프로그래밍을 할 때는, CWEB 파일 이라고 불리는 하나의 파일에 C 코드와 문서화를 위한 코드를 동시에 작성하고, 일반적으로 something.w 과 같이 이름 짓습니다. 다음과 같은 명령어 ‘cweave something’는 something.tex 파일을 생성하고, 이 파일은 TeX 을 통하여 something.w을 페이지 레이아웃과 프로그램의 들여쓰기, 그리고 예약어나 변수명 혹은 수학 기호들을 각각 다른 폰트로 표현하는등 우리들이 쉽게 읽을 수 있도록 멋진 출력물로 만들어 냅니다. 이 출력물에는 cweave 명령어가 자동적으로 만들어 낸 색인도 포함됩니다. 비슷하게, ‘ctangle something’ 이란 명령을 실행하면 something.c란 이름의 C 파일을 얻을 수 있고, 이 파일은 컴파일을 통하여 실행 파일을 얻을 수 있습니다.

CWEB은 C 프로그래밍에 단순히 문서화 도구를 제공한다는 것 외에, C 언어를 보다 강화시키는 측면이 있습니다. 즉 보통의 C 프로그래밍에서는 불가능한, 전체 프로그램을 여러 작은 조각들로 나누고 그 조각들을 여기저기 새롭게 배치할 수 있습니다. 이렇게 함으로 프로그램을 순차적으로 설명하는 것이아니라, 알고리즘 혹은 자료 구조적으로 의미적으로 관련있는 부분들을 section 이라는 이름으로 한 곳에 배치해서 자세히 설명함으로써 각 section을 자세히 설명하고, 그러한 section들의 연관 관계를 설명함으로써 전체 프로그램을 보다 이해하기 쉽게 설명할 수 있습니다. CTANGLE 프로그램은 이처럼 C 문법에 어긋난 방식으로 작성된 CWEB 프로그램을 C 언어 문법에 맞게 다시 재배치 하기 때문에 ctangle 이라는 이름을 얻게 되었습니다. CWEB 으로 프로그래밍하면 얻을 수 있는 잇점은 각 section을 개별적으로 설명할 수 있다는 것을 이용해서 프로그램에 사용된 알고리즘들을 이러한 section들을 이용하여 묶어서 한 곳에서 “untangle”의 형태로 표현할 수 있다는 점입니다. CWEAVE 프로그램은 CWEB 파일을 입력 받아서 각 section을 구성하는 TeX 부분과 C 부분을 서로 엮은 후에 (intertwine) 그러한 section들로 구성된 직물(fabric)로부터 하나의 구조화된 문서를 짜내기(weave) 때문에 cweave 라는

이름을 얻었습니다.(이해가 되십니까?) 영어 단어 “weave”에 해당하는 독일 단어가 “webe” 이고, 그에 해당하는 라틴어가 “tewe” 라는 것에는 아마도 일종의 깊은 연관을 것입니다.

여러분이 CWEB을 사용하고자 한다면, 반드시 C 프로그래밍 언어에 익숙해야 할 것입니다. 최소한의 T_EX에 대한 지식도 있으면 좋겠지만, CWEB 프로그래밍을 하면서 T_EX에 대한 지식은 자동적으로 얻게 될 것입니다. 이는 T_EX에서 사용되는 단순한 텍스트는 T_EX에 대한 지식이 거의 없어도 가능하기 때문입니다. 물론 C 언어와 T_EX에 모두 익숙한 독자는 CWEB을 매우 쉽게 배울 수 있습니다.

개요

CWEB 파일에는 두가지 재료가 필요합니다: T_EX 텍스트와 C 텍스트. CWEB 프로그래밍을 하는 프로그래머는 만들고자 하는 문서와 C 프로그램을 동시에 생각해야 합니다. 즉, 프로그래머는 본능적으로 CWEB 파일 상에 취해질 CWEAVE와 CTANGLE 프로그램의 작용을 알고 있어야만 합니다. CTANGLE은 T_EX 파일을 만들어 낼때, CWEB 파일 내의 T_EX 텍스트를 그대로 T_EX 파일에 복사하지만, CTANGLE은 C 파일을 만들때, T_EX 텍스트는 모두 제거되어 전혀 포함시키지 않습니다. 이처럼 T_EX 텍스트는 순전히 문서화를 위한 텍스트로써 CTNAGLE에는 완전히 무시되는 반면에, C 텍스트는 CWEAVE에 의해서 보기 좋게 문서화 되고, CTNAGLE에 의해서는 본래 C 문법에 맞게 재배치 됩니다. 재배치 되는 규칙은 나중에 자세히 설명하게 될 것입니다. 지금 이순간에 여러분이 기억해야 할 점은 두 종류의 텍스트가 있다는 점입니다. CWEB 프로그램을 만든다는 것은 보통의 T_EX 문서를 만든다는 것과 동일하고, 단지 T_EX의 기본적인 모드인 수평모드, 수직모드, 수학모드에 “C모드”가 더해진 것이라고 생각하면 됩니다.

하나의 CWEB 파일은 섹션(sections)이라고 불리는 다소의 스스로 완전하다고 할 수 있는 작은 단위들로 구성됩니다. 각 섹션은 세 요소를 가지고 있습니다.:

- T_EX 요소, 해당 섹션이 어떤 일을 하는지에 대한 설명을 담고 있습니다.
- 중간 요소, C 언어에서 제공하는 매크로 정의 부분을 담고 있습니다. 이 요소는 CTNAGLE에 의해서 전처리기 매크로 정의로 변환됩니다.
- C 요소, CTANGLE이 생성할 C 프로그램 코드 부분. 이 C 코드는 이상적으로는 하나의 단위로 쉽게 이해될 수 있을 정도의 몇줄의 길이를 가지고 있는 프로그램 텍스트입니다. 이렇게 함으로 해서 이 구조를 쉽게 파악 할 수 있습니다.

이 세 요소는 반드시 위의 순서대로 나와야 합니다; 즉 T_EX 설명 요소가 가장 먼저 나와야 하고, 그다음에 중간 요소, 마지막으로 C 코드가 나와야 합니다. 이 요소 중 어느 부분도 생략이 가능합니다.

섹션은 ‘@_’ 혹은 ‘@*’와 같은 기호로 시작됩니다. 여기서 ‘_’는 공백 문자를 나타냅니다. 섹션은 다음 섹션이 시작하는 곳에서 끝나거나 (즉, 다음 ‘@_’ 혹은 ‘@*’) 파일의 끝에서 끝납니다. 또한 CWEB 파일에 어떠한 섹션에도 포함되지 않는 텍스트가 있습니다. 이른바 맨 첫번째 섹션 앞에 나오는 텍스트가 그것인데, 그러한 텍스트를 “in limbo”라고 부릅니다. CTNAGLE은 이 텍스트를 완전히 무시하고, CWEAVE는 그대로 T_EX 파일에 복사하여 T_EX 파일에서 흔히 사용되는 파일 윗부분의 매크로 정의나 문서를 만드는데 필요한 명령어들을 한곳에 모아두는 역할을 합니다. 앞 절에서의 설명에서 처럼 CWEB 작성하는 것 역시 T_EX 파일 작성하는 것과 크게 다르지 않기 때문에 T_EX 파일 작성시 그렇듯이 CWEB 파일 작성할 때, 역시 이 in limbo 부분에 특별한 폰트를 로드하거나 특별 매크로를 정의 하거나 페이지의 크기를 정한다거나 혹은 타이틀 페이지를 만드는 등을 담아서 파일의 맨 앞에 두는 것이 관례입니다.

섹션은 1부터 시작하여 차례로 번호가 붙여집니다. 이 번호들은 CWEAVE가 만들어 내는 T_EX 문서의 각 섹션 시작부에 나타냅니다. 그리고 이 번호들은 CTANGLE이 만들어내는 C 프로그램의 섹션에 의해서 생성된 코드의 시작부와 끝부분에 C 주석문으로 나타나기도 합니다.

섹션 이름

다행인 점은, 여러분은 CWEB 파일을 작성할때, 섹션의 번호를 매기느라 수고할 필요가 없다는 것입니다. 여러분은 단순히 각 섹션의 시작부에 ‘@_’ 또는 ‘@*’ 기호를 붙여주면, 번호는 CWEAVE와 CTANGLE이 자동적으로 매겨집니다. 여러분에게 있어서 섹션은 번호로 인식되는 것이 아니라 이름(name)으로 인식됩니다. 섹션 이름은 처음에 ‘@<’가 나오고, 그 다음에 T_EX 텍스트, 마지막으로 ‘@>’가 나오는 모양을 갖고 있습니다. CWEAVE가 섹션 이름을 출력할 때는 ‘@<’와 ‘@>’를 꺾쇠 괄호로 바꾸고 그 안에 작은 폰트의 섹션 번호를 넣습니다. 따라서 여러분이 CWEAVE의 출력물을 읽을 때 하나의 섹션에서 언급된 다른 섹션을 찾기가 쉬워집니다.

섹션에 대한 상세한 설명을 목적으로, 섹션 이름은 그 섹션의 내용을 잘 설명할 수 있어야 합니다. 즉, 섹션 이름은 그 이름만 보고도 이 섹션이 무엇을 하는 섹션인지 알 수 있을 정도의 간략화된 표현이어야 합니다. 그렇게 되면 그 섹션은 하나 혹은 다른 많은 섹션에 포함되어 그 섹션을 포함하는 섹션의 일부가 이 섹션으로 대치되어서 설명될 수 있습니다. 따라서 섹션 이름은 그 섹션이 전달하고자 하는 의미를 분명히 반영할 수 있을 정도의 충분한 길이어야 합니다.

해석하게도, 이 처럼 섹션이름을 길게 하면 매번 그 섹션이 필요할 때마다 그 섹션을 언급하기 위해서 그 긴 섹션 이름을 매번 타이핑해야하는 것이 참으로 수고로운 작업이 되고, CWEAVE와 CTANGLE가 그 섹션 이름과 섹션을 매치할 수 있도록, 그 긴 이름을 정확하게 타이핑 하는 것도 또한 힘든 작업이 됩니다. 이러한 곤란한 상황을 덜기 위해서, CWEAVE와 CTANGLE은 섹션 이름을 간략화하는 기능을 제공합니다. CWEB 파일내의 어디에선가 정확한 섹션이름이 한번이라도 언급이 되면, 여러분은 단순히 ‘@< α ...@>’ 처럼 타이핑 할 수 있습니다. 여기서 α 는 하나의 섹션을 정확하게 구분해

낼 수 있는 prefix입니다. 예를들면, ‘@<Clear the arrays@>’ 라는 섹션 이름은 ‘Clear’라는 다섯 글자로 시작되는 다른 섹션 이름이 없다면 ‘@<Clear...@>’으로 간략화 될 수 있습니다. 만약에 ‘Clear’로 시작하는 다른 섹션이 있다면, ‘@<Clear t...@>’ 등과 같은 간략화도 가능합니다.

그렇지않다면 섹션 이름들은 연속적인 공백 문자(스페이스들, 탭문자, 개행문자 또는 form feed)들은 하나의 공백 문자와 동일하고 섹션 이름의 앞과 뒤에 나오는 그런 공백 문자들은 제거된다는 사실을 제외하고는 반드시 문자 대 문자로 매치되어야 합니다. 따라서 섹션 이름 ‘@< Clear the arrays @>’ 또한 앞의 간략화 예와 정확히 일치합니다. 간략화 이름에서 생략기호(...) 다음에 나오는 공백도 마찬가지로 무시되지만, 생략기호 앞에 나오는 공백문자는 공백으로 인식됩니다. 따라서 ‘@<Clear t ...@>’ 는 ‘@<Clear the arrays@>’과 매치되지 않습니다.

CTANGLE이 하는 일

앞에서 섹션 이름이 ‘@_’ 또는 ‘@*’로 시작된다고 하였지만, 섹션이 어떻게 TeX 요소, 중간 요소, C 요소의 세 부분으로 나뉘는지에 대해서는 설명하지 않았습니다. 섹션 내에서 중간 요소는 ‘@d’ 또는 ‘@f’가 처음 나타나는 곳부터 시작되고, C 요소는 ‘@c’ 또는 ‘@<section name@>=’가 처음 나타나는 곳 부터 시작됩니다. 후자의 경우 처럼 C 요소를 시작한다는 것은 효과적으로, 섹션 이름이 곧 나타날 C 텍스트를 대표한다는 것을 의미합니다. 다른 경우인, C 요소가 섹션 이름 대신에 ‘@c’로 시작한다는 것은 현재의 섹션은 이름이 없다는 것을 뜻합니다.(unnamed)

섹션 이름을 구성하는 ‘@<section name@>’이와 같은 코드는 임의의 여러 섹션의 각 C 요소에 나타날 수 있습니다: 이렇게 섹션 이름이 여러 번 나타나는 것은 그 섹션이 “정의”된다는 의미가 아니라 “사용”된다는 의미입니다. 다른 말로 하면, 이미 다른 곳에서 정의된 섹션의 C 코드는 C 프로그램의 바로 이 지점에 삽입되어야 한다는 것입니다. 실로, CTANGLE의 주요 개념은 CTANGLE은 각각의 이름이 있는 섹션과 없는 섹션들로 부터 C 프로그램을 만들어 낸다는 것입니다. 그 만들어내는 방법은 다음과 같습니다: 먼저 모든 ‘@d’로 표시되는 모든 매크로 정의들이 C 언어의 전처리기 매크로 정의들로 변환되고, 파일의 맨 처음으로 복사됩니다. 그리고 나서 이름 없는 섹션의 C 요소들이 그 다음에 차례로 복사됩니다; CTANGLE은 여러 번의 패스를 통하여 CWEB 파일로 부터 C 프로그램을 만들어 내는데, 지금까지의 절차로 인하여 1차 패스의 결과물을 얻게됩니다. (CWEB 파일에는 적어도 하나의 이름 없는 섹션이 있어야 합니다. 그렇지 않으면 C 프로그램은 생성되지 않습니다.) 그리고 나서, 1차 패스 결과물내의 모든 섹션 이름들은 해당하는 섹션의 C 요소로 치환되고, 이 치환 작업은 이름 있는 섹션이 하나도 남지 않을 때까지 계속됩니다. 또한 C 요소내의 모든 주석문들은 제거됩니다. CTANGLE이 만들어 내는 C 프로그램은 오로지 C 컴파일러 용으로 출력물을 만들기 때문입니다.

만일 동일한 섹션 이름이 여러 섹션의 C 요소로 사용된다면, 그 사용되는 이름의 섹션이 가지고 있는 C 요소는 이 섹션을 사용하는 섹션의 C 요소들의 텍스트의 합으로 이루어집니다. 이는 매우 유용한 특징인데, 왜냐하면, 예를들어 ‘Global variables’ 이란 이름을 가진 섹션이 있고, 이 섹션이 여러 섹션의 C 요소로 사용된다면, 전역 변수들은 여러 섹션에 걸쳐서 정의 되지만, 결국은 이 모든 전역 변수들이 ‘Global variables’ 이란 이름을 가진 섹션에 모이게 된다는 의미입니다. 여러 개의 섹션이 동일한 이름을 가질 때마다, CWEAVE는 그 이름의 섹션이 처음 정의되는 곳에 그 섹션의 번호를 할당하고, 그 섹션 밑에 각주써 이 섹션이 다른 섹션에서 사용된다는 것을 알려주기 위해서 ‘See also sections so-and-so’ 라고 써넣습니다; 이 각주에는 이 섹션을 C 요소로 사용하는 모든 섹션의 번호들이 나타나게 됩니다. 섹션의 C 텍스트는 CWEB 파일 내에서 등호(=)기호로 나타나는데, 이는 CWEAVE에 의해서 ≡ 기호로 바뀝니다. 즉, ‘<섹션 이름> ≡ C 텍스트’ 로 바뀌게 됩니다. 그러나 동일한 이름의 섹션이 여러 번 나타날 때, 두번째 부터는 현재 나타나는 섹션은 다른 섹션의 C 텍스트에 첨가된다는 의미로 ‘≡’ 기호가 ‘+≡’로 대체됩니다.

CTANGLE이 섹션을 처리할 때마다, CTANGLE의 출력물인 C 파일에 전처리기 명령어인 #line문을 써넣습니다. 이는 C 파일을 컴파일 할때, 컴파일러가 에러 메시지를 내거나, 혹은 여러분이 디버깅 할때, 여러분은 C 파일이 아니라 CWEB 파일의 라인 넘버가 나오므로 원래 여러분이 작성한 파일을 가지고 작업을 할 수 있도록 하기 위함입니다. 따라서 대부분의 경우 여러분은 C 파일을 직접 다루는 일은 없을 것입니다.

CWEAVE가 하는 일

CWEAVE의 일반적인 개념은 다음과 같은 방법으로 CWEB 파일로 부터 .tex 파일을 만들어내는 것입니다: 우선 .tex의 맨 첫번째 줄은 다른 .tex 파일들이 그렇듯이, CWEB의 문서화를 위한 규칙들을 매크로로 만들어 놓은 파일을 포함하는 것입니다. 그리고 다음 줄부터 CWEB 파일의 “in limbo”에 정의된 TeX 텍스트들이 그대로 복사됩니다. 그 다음 부터 각 섹션에 대한 출력 결과가 차례로 나오게 됩니다. 이 과정에서 각 페이지에 섹션들이 적절히 배치됩니다. 마지막으로, CWEAVE는 색인 페이지를 만듭니다. 이 색인 페이지에는 각각의 C 프로그램에 사용된 변수명 이라든가 여러 식별자 들이 나오고, 이 파일을 구성하는 모든 섹션들이 섹션 이름의 알파벳 순으로 나옵니다. 또한 색인 페이지 뿐만 아니라 별표 섹션과 그의 섹션 번호를 나타내는 목차도 만들어 냅니다.

“별표” 섹션이 뭐냐구요? ‘@_’가 아닌 ‘@*’로 시작되는 섹션으로써, 다른 보통의 섹션과 크게 다른 점은 없고, 단순히 여러 섹션들로 구성된 그룹을 대표하는 섹션이 됩니다. ‘@*’ 다음에는 반드시 이 그룹의 제목이 나오고 그 뒤에 마침표가 나와야 합니다. 이러한 섹션들은 TeX 출력물에서 보면 반드시 새로운 페이지로 시작되고, 제목은 다음 별표 섹션이 나오기 전까지 그 페이지와 뒤따라 나오는 페이지의 헤드라인이 됩니다. 그리고, 이 제목은 목차에도 나오고, 섹션의 시작에 굵은 글씨체로 출력됩니다. 주의: 이러한 제목에 TeX의 제어문자는 사용하지 마시기 바랍니다. 사용하지 않으면, cwebmac

매크로가 엉뚱한 일을 할 경우는 없습니다. 이 제목들이 각 페이지의 헤드에 사용될 때는 각 글자들은 대문자로 바뀌고, 섹션의 처음에 나올 때는 굵은 글씨체로 바뀌고, 또한 목차에도 사용되기 때문입니다. 이 세가지 경우에 모두 말뚝을 안 일으킬 제어문자라면 사용해도 됩니다.

각 섹션에 대해서 CWEAVE가 만들어 내는 T_EX 출력물은 다음과 같은 것들로 구성됩니다: 먼저 섹션 번호가 옵니다. (예를 들어 123번 섹션의 시작에 ‘\M123.’가 옵니다. 만약 해당 섹션이 별표 섹션이라면, ‘\M’ 대신에 ‘\N’가 쓰입니다.) 그리고 나서 그 섹션의 T_EX 요소가 CWEB 파일에 여러분이 타이핑해 넣은 그대로 복사되어 옵니다. 그 뒤로 그 뒤로 중간 요소, C 요소가 그들 사이에 약간의 공간을 사이에 두고 옵니다. 중간 요소와 C 요소는 T_EX 요소처럼 그냥 그대로 복사되어 오는 것은 아니고, 멋진 문서를 만들어 내기 위해서 `cwebmac.tex`에 정의된 매크로들이 여기 저기 사용되어 옵니다.