# CWEB으로 즐기는 문학적 프로그래밍

## TEX과 C 프로그래밍 언어와의 만남

남수진

한글 TEX 사용자 그룹




- 이 저작물은 상업적 목적으로 사용하실 수 없습니다.
- 이 저작물은 개인적 목적으로 수정, 배포하시는 것은 괜찮으나, 그럴 경우 반드시 출처를 밝혀야 합니다.




Part I

# 컴퓨터 프로그래밍






## 컴퓨터 프로그램은 읽기 쉬워야 한다

- 소프트웨어 개발하는데 드는 노력 중 프로그램 작성에 들어가는 노력은 고작 10%, 나머지 90%는 이미 작성된 코드의 유지보수, 디버깅, 문서화 작업에 들어간다.
- 타인이 작성한 컴퓨터 프로그램은 읽기가 힘들다. 심지어 자신이 작성한 프로그램도 시간이 지나면 읽기 어려워진다.
- 우리는 다른 사람이 작성한 소스 코드를 보고 프로그래밍을 배운다.
- 프로그래밍 언어도 사람들 사이에 사용되는 언어이기도 하다
- “모든 프로그램은 먼저 사람이 읽을 수 있어야 하고, 두번째로 기계가 읽을 수 있어야 한다.”

# Computer Programming as an Art

- 1974년 Knuth의 ACM Turing Award 수상기념 강연 제목
- 좋은 프로그램은 쉽게 읽고 이해할 수 있는 것어야 한다.
- 프로그램을 효율성만 강조하여서 코드가 쓸데없이 복잡해져서 읽기 어렵게 되면, 디버그와 유지보수가 힘들어져서 결국 그 소프트웨어의 효율은 전체적으로 낮아진다.
- "Premature optimization is the root of all evil (or at least most of it) in programming."
- 소설가나 시인들이 독자들을 위해 문학 작품을 창작해 내듯이, 프로그램도 그것을 읽는 사람들을 위한 문학 작품(works of literature)이라 할 수 있다.






컴퓨터 프로그램은 실행하기 위해서 작성되는 것이 아니라, 읽기 위해서 작성되는 것이다.

Part II

# 문학적 프로그래밍






## 문학적 프로그래밍(Literate Programming)

- 프로그램을 작성할 때, 기본적인 마음가짐을 바꾸자.
  컴파일러가 쉽게 알아들을 수 있도록 프로그래밍 하기 보다는 사람이 쉽게 이해할 수 있도록 프로그래밍 하자. 결국은 이것이 프로그램의 효율을 높이는 것이다!
- 컴퓨터 프로그래밍의 정의 프로그래밍이란 컴퓨터가 해야 할 일들을 논리적이고 순차적으로 나열하는 것이 아니라, 컴퓨터가 해야할 일을 사람들에게 논리적이고 재미있게 설명하는 작업이다.
- 프로그래밍 언어의 문법과 규칙은 컴파일러에 적합한 것이어서 우리 사람들의 정서에는 맞지 않는다.
  따라서 프로그램은 사람이 작성하고 읽기 편하도록 사람에게 적합한 구조로 재배열되어야 한다.

# 문학적 프로그래밍(Literate Programming)

문학적 프로그래밍이란 사람들이 읽기 편하도록 프로그램을 재구성하여 자연스런 설명을 곁들인 프로그램 코드를 작성하는 것이다.

> I believe that the time is ripe for significantly better documentation of programs, and that we can best achieve this by considering programs to be *works of literature*. Hence, my title: "Literate Programming."
>
> Donald Knuth, Literate Programming






# 도대체 어떻게?

어떻게 하면 프로그램을 사람이 읽기 편하게 재배열 할 수 있을까?

프로그램의 구조적 특성을 이용하자.

# 프로그램의 구조는 "WEB" 이다

- 프로그램에는 수 많은 프로시져, 함수, 루틴들 있고 그 모든 것들은 프로그램의 논리적인 흐름에 따라 서로 서로를 호출하거나 이용한다.
- 소프트웨어 프로그램의 구조는 많은 작은 부분들이 내부적으로 서로 긴밀히 연결되어 있는 "웹(web)" 이라고 할 수 있다.
- 아무리 크고 복잡한 프로그램이라 할지라도, 그러한 프로그램을 구성하는 각각의 작은 부분들을 설명하고, 그 부분들이 서로 어떻게 연결되고, 어떠한 관련이 있는지를 설명하는 것이, 전체 프로그램을 보다 쉽게 이해할 수 있는 방법일 것이다.






# 프로그램의 구조는 "WEB" 이다

- 프로그램의 WEB의 특성을 이용해서 작성한 프로그램을 "WEB 프로그램" 이라고 부른다.
- 웹프로그램을 구성하는 작은 부분들을 "섹션(section)" 이라고 부르고, 그 섹션은 §1, §2 처럼 순차적으로 번호가 붙어 있다.
- 각각의 섹션은 그 자체로 독립적인 의미를 가질 수 있을 정도의 십여줄 내외의 소스 코드 크기가 적당하다.
- 커다란 소프트웨어 프로그램은 태생적으로 복잡할 수 밖에 없는 것이어서 그 프로그램의 미묘한 부분깐지 단번에 이해할 수 있는 방법은 존재하지 않는다.
- 하지만, 그 프로그램이 웹프로그램이라면, §1 부터 시작하여, 한 번에 섹션 하나씩 읽고 이해해 나감으로해서 그 프로그램의 전체적인 구조를 보다 쉽게 이해할 수 있다.

# Part III

# CWEB 프로그래밍






## CWEB

- 가장 많이 사용되는 대표적인 문학적 프로그래밍 시스템.
- C 언어로 작성하는 문학적 프로그래밍 언어, 도구, 시스템.
- C 프로그램이란 사실은 섹션들이 유기적으로 결합 되어 있는 CWEB 프로그램에서 C 코드 만을 뽑아내어서 컴파일러가 이해하기 쉬운 구조로 재배열한 프로그램이다.
- CWEB 프로그램은 C 프로그램을 섹션으로 나눈 다음, 사람들이 이해하기 쉬운 구조로 설명을 곁들여 재배열한 프로그램이다.
- CWEB 프로그램과 C 프로그램은 근본적으로 동일한 것이고, 단지 그 배열 순서만이 틀릴 뿐이다.

# CWEB의 섹션

- 웹프로그램의 섹션은 그 섹션의 목적과 하는 일을 설명하는 간단한 설명으로 시작하는데, 그 설명은 주어진 형식이 없이 자연스러운 인간의 언어로된 설명문이다.
- 그 자연스러운 설명 다음에는 그 설명에 해당하는 논리적이고 형식적인 프로그램 코드가 나온다.
- 그리고, 설명과 프로그램 코드 사이에는 짧고 간단한 하나 이상의 매크로 정의가 올 수 있다.
- 섹션의 구조
  - 형식이 없는 한글로 된 자유로운 설명: TEX 코드
  - 매크로 정의들: `#define`문
  - 논리적이고 정형화된 프로그램 코드: C 코드
- TEX으로 만들어지는 자유로운 설명과 그에 C 프로그램 코드를 통한 두 번의 설명으로 섹션을 보다 잘 이해할 수 있다.






# CWEB의 문서화: TEX

- CWEB 프로그램에서 문서화는 TEX이 담당한다.
- TEX은 수식이 많이 사용되는 아름다운 책을 만들기 위한 시스템이다.

$$\prod_{j\geq 0}\left(\sum_{k\geq 0} a_{jk}z^k\right) = \sum_{n\geq 0} z^n \left(\sum_{\substack{k_0,k_1,\ldots\geq 0\\ k_0+k_1+\cdots=n}} a_{0k_0}a_{1k_1}\ldots\right).$$

- TEX은 워드프로세서가 아니라 조판 시스템으로 컴퓨터 프로그래밍 언어이기도 하다.
- TEX으로 문서를 작성한다고 하면, 십중팔구 LATEX을 뜻한다.

# CWEB의 문서화

CWEB과 같은 문학적 프로그램에 워드 프로세서가 아닌 TEX과 같은 조판 언어(문서화 언어)가 필요한 이유는?

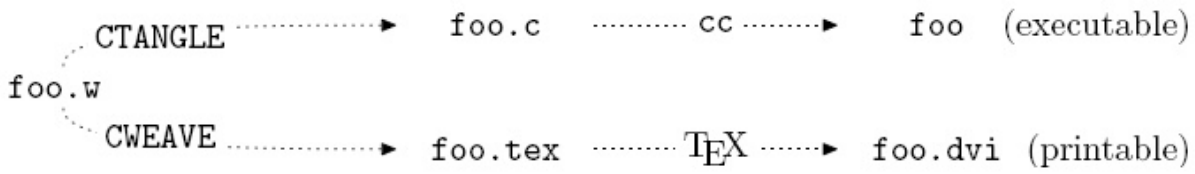
문학적 프로그램에서 문서화를 맡는 부분도 프로그래밍과 동일하게 소스코드 작성, 컴파일의 과정을 거쳐야 하기 때문이고, 이렇게 함으로써 프로그램과 문서를 하나의 파일에서 작성 할 수 있다.






# CWEB 시스템 = CWEAVE + CTANGLE

문학적 프로그래밍이란 사람들이 읽기 편하도록 자연스런 설명과 프로그램 코드를 하나의 소스 파일에서 동시에 작성하는 것이다.




- CWEAVE: 웹파일로 부터 TEX 파일을 만들어 낸다. 소스 코드와 문서를 보기 좋게 잘 조합하여 직물을 짜내듯이(weave) 보기 좋은 문서를 만든다.
- CTANGLE: 웹파일의 각 섹션들에서 C 코드만 뽑아내어 C 파일을 만들어 낸다. 사람들의 기준에서 잘 정돈된(untangle) 코드를 뒤섞어서(tangle) 컴파일러가 이해 할 수 있는 코드를 생성한다.

# CWEB의 장단점

장점

- 프로그래밍 작성하기와 읽기가 쉬워진다.
- 프로그램 소스 코드를 신경쓰지 않아도 된다.
- Emacs 내에서 gdb를 이용한 디버깅으로 쉽고 빠른 디버깅이 가능하다.

단점

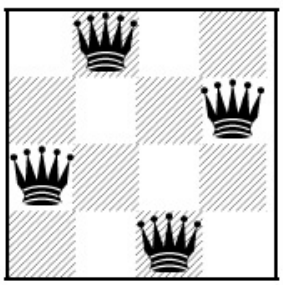
- 프로그래밍 언어는 C, C++, Java만 사용할 수 있다.
- 고품질의 문서 생성해내기 때문에, 관련 명령어(control code)가 많다.
- plain TeX을 이용한다.

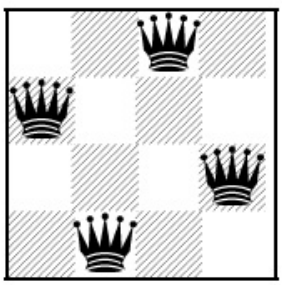





# CWEB 프로그래밍 시연

N queens Problem





# CWEB의 한글화

- TEX의 한글화로 가능해졌다.
- CWEB시스템의 소스 컴파일
- 한글 사용(UTF-8)을 위해서 `common.w` 소스를 수정해야 한다.
- 아래와 같은 내용을 포함하는 `comm-utf8.ch` 작성

```
@x
char *buffer_end=buffer+buf_size-2; /* end of |buffer| */
@y
char *buffer_end=buffer+long_buf_size-2; /* end of |buffer| */
@z
```

- CWEB의 문서화는 TEX이 담당한다.
  - `ucsplain.tex`: TEX에서 한글 사용
  - `cwbucsol.tex`: PDF 파일에서 한글 책갈피
  - `hangulcweb.tex`: CWEB의 매크로 파일인 `cwebmac.tex`에서 한글화가 필요한 매크로를 재정의






# 그밖의 문학적 프로그래밍 도구

- WEB: Pascal 언어를 이용. 최초의 웹프로그래밍 시스템
- noweb: LATEX + 모든 프로그래밍 언어. 매우 간단.
- Rambutan: TEX + Java 언어, CWEB와 동일.
- FWEB: Fortran 언어
- xmlLP
- ...

# Part IV

# 결론






## 컴퓨터 프로그래밍은 예술이다

- 세상에서 가장 즐거운 일 중 하나는 우리들이 작성한 컴퓨터 프로그램을 다른 사람들 혹은 여러분 자신이 읽고 기쁨을 얻는 것이다.
- 컴퓨터 프로그래밍은 음악, 미술, 문학과 같은 예술이다.
  - 축적된 지식을 세상에 적용해서 그렇고,
  - 기술과 독창력을 요구해서 그렇고,
  - 무엇보다도 아름다움을 만들어내기 때문에 그렇다.
- 어렴풋이나마 자신을 예술가로 인식하는 프로그래머는 자신이 하는 일을 즐길 것이며, 더욱 잘할 것이다.

# Knuth's Interview: TUGboat(2005년)

Question: Please, tell our readers briefly what made you decide to start the project, which tools you used, and how many people you had at the core of the TEX team.

Answer: "The tools I used were home grown and became known as Literate Programming. I am enormously biased about Literate Programming, which is surely the greatest thing since sliced bread. I continue to use it to write programs almost every day, and it helps me get efficient, robust, maintainable code much more successfully than any other way I know. Of course, I realize that other people might find other approaches more to their liking; but wow, I love the tools I've got now. I couldn't have written difficult programs like the MMIX meta-simulator at all if I hadn't had Literate Programming; the task would have been too difficult."






# 참고문헌


- Knuth, *Literate Programming*, Center for the Study of Language and Information, 1992
- Knuth, *The CWEB System of Structured Documentation*, Reading, Massachusetts: Addison-Wesley, 1993
- Daniel Mall의 Literate Programming 웹사이트
  http://www.literateprogramming.com
- Chris Lee의 Literate Programming 간단한 소개 웹사이트
  http://vasc.ri.cmu.edu/old_help/Programming/Literate/literate.html
- 문학적 프로그래밍: http://faq.ktug.or.kr/faq/LiterateProgramming
- CWEB: http://faq.ktug.or.kr/faq/CWEB


# Part V

## KTUG: Korean TEX User Group






## KTUG: 한글 TEX 사용자 그룹

- LATEX의 한글화와 보급 http://www.ktug.or.kr
- 2001년 이전
  - ChoF's TeX Archive : MiKTeX 2.1, HLaTeX 0.98/0.99.
  - 도은이네 집 : fpTeX, HLaTeX 0.98/0.99
- 2001 ∼ 2002 년간
  - KTUG : MiKTeX-KTUG 2.2 (by ChoF)
  - KTUG : HPack (by 홍석호)
  - 2002년은 가장 생산적인 해 중 하나였음. 옛한글, dvipdfm-cjk(DVIPDFMx), pdftex subfont patch(by ChoF)[1], ConTeXt 한글화 등등...
- 2003년
  - MiKTeX-KTUG 2.3
  - cvs를 이용한 HLaTeX의 설치가 권장된 적이 있음.
  - 은글꼴과 DVIPDFMx를 이용한 검색/추출 가능한 pdf 제작 기술 발전. 은글꼴이라는 GPL 트루타입 글꼴은 이 분야의 발전에 획기적인 전기였음.
  - 옛한글 관련 Omega/Lambd를 이용한 다양한 실험.

# KTUG: 한글 TEX 사용자 그룹

- 2003 ∼ 2004년간
  - MiKTeX 2.4와 더불어 MiKTeX-KTUG 프로젝트 중단.
  - HPack for MiKTeX 2.4를 이용한 HLaTeX 설치 (홍석호)
  - DVIPDFMx의 MiKTeX 2.4를 위한 바이너리가 나오는 데 한참 걸려서 한동안 pdf 제작 기술이 실용화되지 못하였음. 이 때 CygWin teTeX에 대한 관심이 증가함.
  - pdf 제작 기술은 hangul-k(HANGULkStyle)로 발전함.
  - 한글 타이포그래피에 대한 관심이 증가하여 hlatex-interword 등이 제작됨.
  - 옛한글 및 고문헌 처리 시도들이 좋은 결과를 보이기 시작하고 DHHangul이 제작됨.
  - LaTeX2html에 대한 수요가 줄어들고 그 대신 TeX4ht의 한글화가 진행됨 (by synapse, HuidaeCho)
  - HanyangPuaTableProject가 진행됨.






# KTUG: 한글 TEX 사용자 그룹

- 2005년
  - Hangul-ucs 패키지의 발전. DHUcs라는 이름으로 시작되어 나중에 상당한 규모의 프로젝트로 발전하였음.
  - HLaTeX 설치 지원은 MiKTeX + KTUG patch(dvipdfmx, ttf2pk, etc.) + HPack의 방법이 표준적인 방식으로 자리잡음. (이 기간이 상당히 길었음)
  - KTUGCollection2005 제작. MiKTeX + HLaTeX/DHUcs 등, 그 시기까지 발전한 거의 모든 한글 환경을 하나의 통합 설치 패키지로 만든 것이었음.
  - 한글 pdf 제작에 관하여 요구되던 거의 모든 문제점을 Hangul-ucs가 해결함.
  - Hangul-ucs의 발전과 더불어 memoir의 한글화가 이루어짐.

# KTUG: 한글 TEX 사용자 그룹

- 2006년
  - ALee님에 의한 Hangul-ucs Debian 패키지가 나오고, WkPark 님에 의한 Fedora 설치 패키지를 시작으로 리눅스 쪽에서 Hangul-ucs가 광범위하게 받아들여짐.
  - Hangul-ucs까지 지원하는 HPack 1.2 (beta) 공개.
  - WinEdt의 정체 때문에 WinEdt은 HLaTeX 전용 에디터로 인식됨. Hangul-ucs 쪽에서는 Emacs나 EmEditor를 권장하게 됨.
  - KTUGCollection2006 제작. MiKTeX 대신 W32TeX/ko를 채택하고 최신 베타 버전의 pdftex을 탑재한 실험적인 TeX 환경이 선보임. 이 과정에서 ConTeXt, XeTeX의 한글 사용 가능성을 제시함.





# KTUG Collection2006 CD

# QnA

*Perhaps we will even one day find Pulitzer prizes awarded to computer programs.*

Donald Knuth, Literate Programming






감사합니다