# Emacs 초급자 HOWTO

Jeremy D. Zawodny: jzawodn@wcnet.org October 14th, 1999 ($Revision: 1.7 $)

이태철: tclee@prosys.korea.ac.kr 2001. 1. 28

이 문서는 리눅스 사용자들에게 Emacs라는 에디터를 소개하기 위해 만들어졌다. 기본적으로 vi와 같이 에디터에 대해 조금은 이미 알고 있다는 가정 하에 작성되었으며, 이문서의 최신버전은 http://www.wcnet.org/jzawodn/emacs/ 에서 찾을 수 있다.

## 목 차

# 제 1 절 소개

## 1.1 저작권



## 1.2 독자와 의도

이 문서는 이멕스에 대해 관심이 있으며 사용하고자 하는 리눅서 유저를 위해 작성되었다. 저자가 Toledo 지역 리눅스 유저모임인 http://www.talug.org/에 제출했던 간단한 안내서를 바탕으로 만들었다. 그 후 모임으로부터 여러 유용한 제안에 의해 점차 개선되었다. 자세한 내용은 감사부분을 보기 바란다.

이 문서에는 리눅스에만 적용할 수 있는 것들은 없다. 여러 종류의 유닉스나 심지어는 MS사의 윈도우에서 작동되는 이멕스에서도 사용할 수 있다. 그렇지만 이 문서는 리눅스 문서 프로젝트의 일부분으로 만들어진 것이기 때문에 리눅스 사용자들의 발전을 위해 작성이 되어 것이라는 것을 얘기를 하고 싶다.

마지막으로 단순히 "Linux"라고 하는 것보다 GNU/Linux라는 하는 것을 더 좋아하는 사람들은 이 문서에서 나오는 모든 Linux를 마음속으로 GNU/Linux로 바꾸기 바란다.(왜 그런지 알고 싶으면 http://www.gnu.org/gnu/linux-and-gnu.html 부분을 읽어보기 바란다.) 나는 그러한 이면의 생각에 동의를 하지 않고 GNU/Linux라고 굳이 쓰고 싶지도 않다.

## 1.3 이멕스란 무엇인가 ?

이멕스는 사람들에 따라 다르게 보일 수 있다. 여러분이 요구하는 것에 따라 다음의 것들로 생각할 수 있다.

- 텍스트 편집기
- 메일 보는 프로그램
- 뉴스 보는 프로그램
- 문서 작성기
- 종교
- 통합개발 환경
- 당신이 원하는 모든 것

그렇지만 현재의 목적에 따라 지금은 환상적 기능을 가지고 있는 문서편집기로 보자. 추후에 그 문제에 대해서 좀더 생각해 보기로 하겠다. 이멕스는 Richard Stallman(자유재단의 창시자 : http://www.fsf.org/ and the GNU project http://www.gnu.org/)에 의해 만들어졌으며 아직도 그가 관리하고 있다.

이멕스는 리눅스나 유닉스에서 가장 많이 사용되는 강력한 문서 편집기 가운데 하나이며, **vi**에 이어 두번째로 많이 사용되는 편집기이다. 또한 여러 가지 다양한 기능을 가지고 있으며, 사용하기 편하게 수정할 수 있으며 프로그램의 문제도 거의 없다. 자세한 모든 것을 얘기하기 전에 이멕스가 단순한 텍스트 에디터가 아니라는 것을 얘기하고 싶다. 이멕스는 대부분 **Lisp**라는 프로그래밍 언어에 의해 작성된 에디터이다. 이멕스의 가장 중요한 요소는 이 C로 작성된 모든 기능을 가지고 있는 Lisp라는 해석기이다. 단지 가장 기본

적이고 이멕스의 가장 낮은 영역만이 C로 작성이 되어있고, 대부분의 이멕스의 구성요소들은 Lisp로 작성되어있다. 그러므로 이멕스는 전체적으로 사용자가 수정, 확장하고 프로그램의 방식을 바꾸는데 사용할 수 있는 내장 프로그래밍 언어이다.

이멕스는 가장 오래된 에디터 가운데 하나이며, 거의 20(?)년 가량 많은 프로그래머에 의해 사용되어왔다는 점은 많은 추가 패키지들이 가능하다는 얘기와 일맥상통하다. 이러한 추가 패키지들에 의해 Stallman이 처음에 상상조차하지 못했던 많은 기능들이 가능해졌다. 좀더 자세한 것들은 다음 섹션에서 다루겠다.

이멕스에 관련된 역사나 문서들은 많은 웹사이트에서 찾을 수 있다. 이 문서에서 그러한 것들을 다시 논의하는 것보다 이문서의 6 (다른 자료들)장을 찾아보도록 추천한다.

#### 1.3.1 Ports and Versions

실질적으로 이멕스는 GNU Emacs와 XEmacs의 두 가지 형태로 존재한다. 또한 이 두 가지는 서로 많은 기능을 공유하고 있다. 이문서는 GNU 이멕스(버전 20.3)에 대해 설명을 하고 있지만 많은 부분은 XEmacs와 공유할 수 있다. 또한 문서 전체에서 단순히 "이멕스"이라고 부르는 것을 명심하기 바란다.

#### 1.3.2 이멕스 구하기

이멕스를 구하는 것은 쉽다. 만약 다른 여러 리눅스의 배포판, 데비안이나 레드햇, 슬랙웨어 등을 가지고 있다면 이멕스는 아마도 옵션으로 설치할 수 있는 패키지 가운데 하나일거다. 그렇지 않다면 이멕스 소스코드를 받아와서 직접설치를 해야만 한다. 정확히 GNU 웹사이트 : `http://www.gnu.org/software/emacs/emacs.html`에 가봐라.

# 제 2 절 이멕스 실행

## 2.1 시작 & 끝내기

사용자들이 컴퓨터를 처음 사용하다가 보면 이멕스라는 프로그램도 한번씩 해보는 프로그램일 것이다. 일단 이멕스를 실행을 했다면 나가는 방법을 찾아볼 것이다. 그렇지만 아마도 어떻게 해야 나갈 수 있는지 알기 힘들 것이다. 이전에 한번도 이멕스를 사용해보지 않았다면 지금부터 그 과정에 대해 설명을 하겠다. 먼저 쉘 상에서 `emacs`라고 타이핑을 치고 엔터키를 눌러라. 그러면 이멕스를 실행할 수 있다. 만약 실행이 되지 않는다면 프로그램을 설치하지 않았던가 경로변수(PATH)의 정의 안에 이멕스가 들어있지 않을 것이다.

일단 이멕스를 보았다면 나가는 방법도 알아야 한다. 이멕스를 나가기 위해서는 `C-x C-c`키를 눌러야 한다. 이 `C-x`라고 쓰여진 방법은 `Ctrl`키를 누르고 있는 상태에서 `x`를 누른다는 것을 의미한다. 이 경우에는 그 다음 `Ctrl`를 누른 상태에서 `c`키를 눌러야만 한다.

이멕스에서 사용되는 키의 조합은 좀 이상해 보이고 처음에 사용하기에는 불편할 수도 있다. 특히 여러분이 `vi`를 사용해 왔다면 더욱 그럴 것이다. 그렇지만 이멕스는 `vi`와 달리 편집모드와 명령모드를 구분해서 사용하지 않는다.

요약 : `emacs`은 이멕스 시작. `C-x C-c`은 이멕스 종료.

#### 2.1.1 여러분이 볼 것

이멕스가 실행이 되면 일단 하나의 X 윈도우 전체에 걸쳐서 실행이 된다.(만일 콘솔에서 작업을 하고 있다면 콘솔전체) 위쪽에는 메뉴화면이 있고 화면의 중앙에서 약간의 텍스트들이 있으면 밑에 부분에는 몇 개의 줄이 있다.

아마 다음과 같은 모습의 화면일 것이다.

```
+----------------------------------------------------------------------+
|Buffers Files Tools Edit Search Mule Help                             |
|                                                                      |
|Welcome to GNU Emacs, one component of a Linux-based GNU system.      |
|                                                                      |
|                                                                      |
|                                                                      |
| ...                                                                  |
|                                                                      |
|---1:---F1  *scratch*         (Lisp Interaction)--L1--All-------------|
|For information about the GNU Project and its goals, type C-h C-p.    |
+----------------------------------------------------------------------+
```

**NOTE:** 이멕스는 일반적으로 전체화면이나 윈도우를 차지한다. 공간을 줄이기 위해서 위의 예제에서는 좀 줄였다. 처음 이멕스를 실행하면 환영한다는 메시지를 볼 수 있다. 그리고 나머지들을 "..."으로 나타냈다. 환영 메시지에는 당신이 사용하는 이멕스의 버전과 온라인 도움말과 그와 관련한 사항들을 있다.

**메뉴바** 이멕스의 가장 위에는 메뉴가 있다. X 에서 실행하였다면 일반적으로 마우스로 움직일 수 있는 풀다운 메뉴로 되어있다. 그렇지 않다면 단축키(여기서는 얘기하지 않지만)를 이용해서 메뉴에 접근할 수 있다.

**상태바와 미니버퍼** 이멕스의 밑에 있는 두 줄 가운데 바로 위에 있는 것은 상태바이다. 여기에는 현재 사용자가 무엇을 하고 있는지를 알려준다. 일단 지금은 그것이 있다는 것만 알고 있자.

가장 밑에 있는 줄은 **미니버퍼**라고 한다. 이것은 메인버퍼와 상태바로 구분이 되며, 이것을 이멕스의 "명령줄"이라고 생각하면 된다. 이곳에는 여러분이 이멕스에서 실행을 한 명령어와 그에 대한 이멕스의 응답이 있다.

여기서 내가 얘기한 상태바는 다른 이멕스와 관련된 문서에서는 모드라인이라고 한다. 여기에는 현재시간이나 날짜, 줄수, 파일의 크기를 포함해서 사용자가 사용하고 있는 현재모드에 대한 것들이 있다.

## 2.2 몇 가지 용어

이 섹션에서는 이멕스를 이용하면서 사용하게 되는 용어에 대해서 알아보자.

### 2.2.1 버퍼 & 파일

다른 에디터들과는 달리 이멕스에서 파일을 열면 여러분이 사용하는 동안 계속 파일을 "열어"두지 않는다. 대신 이멕스는 파일을 **버퍼**라는 메모리에 넣어둔다. 그러므로 버퍼를 편집하는 것이지 파일을 편집하는 것이 아니다. 실제 여러분이 저장을 했을 때만 디스크에 저장이 된다. 이렇게 하는 것은 장단점이 있지만 여러분이 이렇게 작업한다는 점을 이해하는 것이 중요하다.

결과적으로 여러분들은 이멕스와 관련된 문서, 예를 들어 모드나 패키지 등에서 "버퍼"라는 용어를 볼 수 있을 것이다. 그럴 때 버퍼라고 하는 것은 "메모리에 넣어둔 현재 파일의 사본"이라고 알면 된다. 종종 이멕스는 사용자의 명령의 결과로서 버퍼를 만든다. 그러한 버퍼는 명령의 결과나 선택할 수 있는 리스트 등을 포함하고 있다.

### 2.2.2 포인트 & 영역

이멕스 전문용어로 종종 **포인트**라는 용어를 사용하는 경우가 있다. 일반적인 용어로서 포인트는 커서를 의미한다. 처음 이멕스를 사용할 때에는 포인트나 커서의 차이점은 그리 중요하지 않다. 만일 관심이 있다면

이렇게 한번 생각해 봐라. 커서라고 하는 것은 포인트를 시각적으로 형상화 한 것이다. 커서는 항상 현재 버퍼의 어느 특정 글자위치 “위”에 있다. 그러므로 이렇게 얘기할 수 있다. “the”라는 단어의 ‘h’ 글자 위에 커서가 있다는 것은 ‘t’와 ‘h’ 사이에 포인트가 위치하고 있는 것이다.

다른 에디터들처럼 이멕스도 현재 버퍼의 부분에 여러 가지 기능(들여쓰기, 맞춤법검사, 재포멧, 잘라내기, 복사, 붙이기 등)을 할 수 있다. 사용자들은 키보드를 사용하던지 아니면 마우스를 사용해서 텍스트의 일부분을 돋보이게(혹은 “마크”) 설정하고 이 블록에 대해 위에 있는 기능들을 사용할 수 있다. 이멕스에서는 텍스트의 블록을 **영역**이라고 한다.

### 2.2.3 윈도우

이전에 GUI 인터페이스를 사용하던 사람들은 다소 이상할 것이다. 그러나 기억해야할 사항은 이멕스는 GUI 인터페이스나 윈도우 메니저가 일상화 되기 전에 이미 만들어졌다.

이멕스에서 **윈도우**는 버퍼가 표시되는 부분이다. 이멕스가 시작되면 사용자의 스크린위에 하나의 윈도우를 만든다. 몇몇 이멕스의 다른 기능(도움말이나 문서 등)들은 이멕스의 윈도우에 [임시로] 추가로 만든다.

이멕스는 GUI관점에서 보면 X 윈도우와 전혀 관계가 없다. 사용자는 새로운 이멕스 버퍼를 보기 위해 X 윈도우를 열어서 두 파일을 비교할 수도 있다. 이때 새로 열린 X 윈도우를 이멕스용어로는 **프레임**이라고 한다. 계속 읽어보자.

### 2.2.4 프레임

이멕스에서 **프레임**이라고 하는 것은 이멕스의 버퍼가 표시되는 분리된 X 윈도우이다. 그러나 두개는 동일한 이멕스 세션이다. 이러한 형태는 넷스케이프에서 Alt+N을 해서 열어보는 것과 같은 것이다.

## 2.3 Keyboard 기본

이 섹션에서는 이멕스에서 사용하는 기본적인 키에 대해서 알아본다. 모든 강력한 에디터와 동일하게 몇번의 타이핑으로 이멕스의 모든 것을 사용할 수 있다.

기존의 `vi` 사용자라면, `k, j, l, h` 키가 상하좌우로 움직인다는 것을 알 수 있을 것이다. 사실 `vi`에서 이러한 키를 익혀 파일을 이리 저리로 옮겨다니려면 여러 시간 아니 여러 주가 필요할지 모른다.

이멕스도 차이는 없다. 단지 다른 키와 명령어로 구성되어 있을 뿐이다. `vi`와 같이 기본적인 것을 익히려면 많은 시간이 필요하다. 그리고 시간이 지남에 따라 천천히 키구성에 대해 좀더 많은 것을 배우고 좀더 빠른 방법을 배울 수 있다.

### 2.3.1 명령키 (Meta, Esc, Control, Alt)

조금 후에 배우겠지만 이멕스는 여러 가지 키를 같이 사용하는 경우가 많다. 이멕스는 `vi`처럼 모드에 따라 결정되는 에디터가 아니기 때문에 커서를 움직이거나 명령을 실행하기 위해 “명령모드”나 “편집모드”와 같은 것을 알 필요는 없다. 대신 일반적으로 키를 올바르게 조합해서 입력해야만 이멕스가 정확한 명령을 수행을 할 수 있다.

이멕스에서 사용하는 키는 거의 Control이나 Ctrl에 상응하는 `C`나 Meta키에 대한 `M`과 같이 약자를 사용한다. 일반적인 현대의 키보드는 한두 개의 `Ctrl`키와 거의 없지만 `Meta`라는 키가 있다. 그러나 대부분의 사용자들은 이 Meta키에 대해서 `Esc`나 `Alt`키를 사용한다. 일반적인 구성에서 Esc나 Alt는 기능적으로 거의 같다.

그래서 이멕스에 관한 문서에서 참조로 `C-x f`라는 키의 조합이 나온다면 이것은 “Ctrl키를 누르면서 x키를 누르고 그 다음 f”를 누르라는 것이다. 또 `M-x shell` 이러한 조합으로 되어 있는 것은 “Alt키를 누르면서 x를 누르고 그 다음 shell을 입력하라”라는 의미이다.

초보자들에게 아주 유용한 명령어는 M-x apropos나 C-h a이다. apropos는 이멕스의 모든 함수에 대한 온라인 메뉴얼에서 사용자가 입력한 정규표현에 대해 검색을 수행한다. 프레임에 관련된 모든 명령어를 찾고 싶은 경우 간단히 C-h a를 입력한 후 frame을 입력하면 된다.

### 2.3.2 버퍼에서 돌아다니기

이제 여러분은 약어에 대한 의미를 어느 정도 알았을 것이고 이제 버퍼에서 움직여 다닐 수 있는 모든 키에 대해 알아보자.

```
입력키        행동
-----------------------------------
C-p           한줄 위로
C-n           한줄 아래로
C-f           한글자 앞으로
C-b           한글자 뒤로
C-a           줄의 시작으로
C-e           줄의 끝으로
C-v           한페이지 밑으로
M-v           한페이지 위로
M-f           한단어 앞으로
M-b           한단어 뒤로
M-<           버퍼의 처음으로
M->           버퍼의 끝으로
C-g           현재 명령종료
-----------------------------------
```

그리고 커서키(방향키)는 일반적으로 예상했던 것과 동일한 방향으로 움직인다. 그렇지만 백스페이스는 그렇지 않다. :-(

### 2.3.3 필수 명령어

자 이제 여러분들은 버퍼에서 커서를 움직이는 것이나 파일을 열고 닫는 방법, 찾는 방법에 대해 알았다. 이제부터는 기본적인 명령어에 대해서 알아보자.

이러한 명령어로 넘어가기 전에 어떻게 이러한 명령어들이 실행이 되는지에 대해 간단히 설명하고자 한다.

이멕스에서 사용하는 모든 "명령키"는 실제 이멕스의 한 부분인 함수로의 단축키이다.(예를 들어 M-x 아무것이나 C-아무것) 사용자는 이러한 함수를 M-x 함수이름과 엔터키를 누름으로 실행시킬 수 있다. 또한 사용자들은 만일 함수에 대한 단축키가 있다면, 함수에 대한 단축키를 이용해서 사용할 수도 있다.

예를 들어 버퍼를 디스크에 저장하는 이멕스 함수는 save-buffer이다. 디폴트 값으로는 C-x C-s로 되어있다. 그러므로 이 키를 이용해서 현재의 버퍼를 저장할 수도 있고, 또한 M-x save-buffer를 이용해서도 똑같은 일을 할 수 있다.

대부분의 기본적인 명령어는 단축키가 존재하며 그 가운데 몇 가지는 다음 표에 나타내었다.

```
단축키        함수                설명
------------------------------------------------------------------
C-x C-s       save-buffer         현재 버퍼를 디스크에 저장한다.
C-x u         undo                마지막 명령을 취소한다.
C-c C-f       find-file           디스크에서 파일을 연다.
C-s           isearch-forward     단어를 순차적으로 찾는다.
C-r           isearch-backward    단어를 역순으로 찾는다.
```

```
            replace-string      단어를 찾아 고체한다.
            replace-regexp      정규표현식을 이용한 찾기와 고체
C-h t       help-with-tutorial  안내서 사용
C-h f       describe-function   함수에 대한 안내문보기
C-h v       describe-variable   변수에 대한 안내문보기
C-h x       describe-key        키 순서에 대해 보기
C-h a       apropos             단어나 정규표현에 대한 안내문 찾기
C-h F       view-emacs-FAQ      이멕스 FAQ 표시
C-h i       info                info 문서 읽기
C-x r m     bookmark-set        북마크표시
C-x r b     bookmark-jump       북마크로 가기
---------------------------------------------------------------
```

이러한 함수를 여러 번 사용하다가 보면 대부분 경우 이멕스가 사용자의 입력을 기다리고 있다는 것을 알게될 것이다. 이러한 것은 미니 버퍼에서 생기며 vi에서 :를 이용할 때나 유닉스 쉘 상에서 사용하는 것과 비슷하다.

이멕스는 실질적으로 수백 개의 내장함수를 가지고 있다. 위에 있는 리스트는 내가 사용하고 있는 아주 적은 수의 샘플을 나타낸 것이다. 온라인 안내문에 있는 사용 가능한 함수에 대한 것과 좀 전에 얘기한 것에 대한 자세한 안내문을 찾아보기 바란다.

### 2.3.4 탭완성

bash나 csh, tcsh과 같은 많은 일반적인 유닉스 쉘과 같이 이멕스는 Tab키를 이용한 명령어 완성 기능을 제공한다. 사실 bash에서 명령어 완성기능은 이멕스의 기능으로부터 나왔으므로 bash에서 여러분이 사용하는 기능은 이멕스에서 다 될 것이다.

예를 들어 M-x search라고 치고 Tab을 쳐봐라. 이멕스는 여러 가지 다른 가능성을 가지고 있는 단어를 나타내기 위해서 하이픈을 붙여서 표시할 것이다. 그렇지만 그것들은 모두 하이픈을 포함한 단어들이다. Tab을 한번 더 치면 사용자가 선택할 수 있는 단어들의 리스트를 표시할 것이다. 이것은 *새 윈도우*에서 만들어진다. 새 윈도우는 임시로 여러분의 화면을 두개의 화면으로 나눈다. 하나는 현재 여러분의 버퍼가 있으며, 또 다른 하나는 "search-"로 시작하는 가능한 단어들이 있을 것이다. 선택을 마치고 난 후 C-g를 입력하면 새 윈도우는 닫힌다.

## 2.4 Tutorial, Help, & Info

이멕스는 기본적인 편집기능이나 모든 사람들이 알아야할 함수에 대한 온라인 안내서를 가지고 있다. 또한 이 안내서는 이멕스에서 사용할 수 있는 다른 도움말에 대한 기능도 설명하고 있다.

만일 여러분들이 이멕스를 심도 있게 배울 생각이라면 이 안내서를 꼼꼼히 읽어보길 바란다. 위의 표에 나와있듯이 C-h t를 입력하면 안내문을 볼 수 있다. 안내문은 이멕스를 시작하는 사람들을 위해 만들어졌으며 혼자 해볼 수 있는 것으로 구성이 되어있다.

만일 X에서 이멕스를 사용하고 있다면 메뉴의 맨 오른쪽 help라고 되어있는 것을 찾을 수 있다. 여기에서 명령어를 살펴보면 어떠한 것들은 단축키가 지정이 된 것들이 있으며 이러한 것들은 메뉴의 오른쪽에 표시되어 있다.

마지막으로 이멕스에 포함되어 있는 문서들을 볼 수 있는데 이러한 것들을 보는 명령어들은 M-x info나 C-h i로 실행시킬 수 있으며 이멕스 문서 탐색기에 Info파일을 보여준다.

# 제 3 절 이멕스 모드

이멕스의 모드는 다른 기능과 형태로 되어있는데 필요에 따라 그것들을 실행하거나 끌 수 있다. 이러한 모드는 문서를 작성하거나 여러 가지 언어로 프로그래밍을 한다든지 홈페이지를 만들거나 메일을 보내고 뉴스를 보고 약속을 기록해 두거나 심지어 게임을 할 수 있도록 기능을 제공한다.

이멕스의 모드는 이멕스의 기능을 확장하고 변형 발전해 나갈 수 있는 Lisp 코드로 만들어진 라이브러리이다.

## 3.1 주 모드와 부 모드

두 가지 기본적인 모드가 가능한데 그것은 주 모드와 부 모드이다. 그 기능상의 차이점은 그 기능을 중단하거나 실행하기 전에는 잘 이해할 수 없으나 일단 시작을 해보자.

한번에 한가지의 주모드를 실행할 수 있으며, 여러 가지 부모드는 동시에 실행할 수 있다. 주모드는 언어나 주어진 문제와 직접적으로 관련되어 있는 반면 부모드는 좀더 많은 작업을 해 나가는데 특정적으로 사용되어지는 것이다.

간단하게 예를 들어보자. 내가 주로 단순한 텍스트 문서만을 작업하는 경우에 많이 사용하는 주모드는 `text-mode`이다. 이 모드는 README 파일과 같은 것을 편집할 때 사용하기 위해 만들어졌다. 단어와 문단을 구분하고 일반적 방향키가 내가 원하는 대로 움직이도록 되어있다.

만일 내가 인간소모에 관한 글을 쓸 때 그것이 보기 좋게 되기를 원한다. 예를 들어 알맞은 값에서 줄을 바꿔주는 기능이나 등등 여러 가지가 필요하다. `auto-fill` 부모드를 실행하여 줄바꿈을 실행할 수 있다. 이 모드는 줄의 마지막을 입력했을 때 알아서 실행이 되며, 이 부모드는 다른 여러 가지 주모드와 같이 실행이 될 수 있다. 여기서 "알아서 실행" 이라는 것은 `text-mode`에서 실행하는 경우와 예를 들어 `java-mode`에서 실행하는 경우와는 다르다. 난 java 코드가 영어의 일반 문장처럼 줄바꿈이 생기는 것은 원하지 않는다. 그러나 java코드의 블록코멘트는 줄바꿈이 일어나길 원한다. 이 `auto-fill` 모드는 그러한 것들을 충분히 잘 감지하여 작동한다.

여러 가지 이멕스 모드를 만드는 사람들은 주모드와 부모드에서 벌어질 것들을 충분히 잘 이해해야한다.

이멕스 프로그램의 아스키 그림으로 돌아가서 보면 모드라인에 현재 이멕스가 사용하고 있는 모드가 나타나 있다. 위의 경우는 기본모드인 "Lisp Interaction" 모드이다. 이 모드는 여러분이 Lisp 코드를 만들때만 유용하다.(그러나 대부분의 이멕스는 Lisp로 만들어져 있다.)

## 3.2 프로그래밍 모드

처음부터 이멕스는 프로그래머를 위해 프로그래머에 의해 만들어졌다. 여러분이 생각할 수 있는 대부분의 여러 프로그램에 대해 높은 수준의 모드가 있다. 이러한 것들에 대해 간단히 설명하고자 한다.

대부분의 프로그램모드는 여러 가지 비슷한 특징이 있다. 일반적으로 다음에 있는 기능을 거의 다 가지고 있을 것이다.

- 언어에 대해 칼라로 된 문장강조기능을 제공한다.
- 언어에 대해 들여쓰기와 코드 포멧기능을 제공한다.
- 문맥에 관련된 도움말 기능을 제공한다.
- 여러분의 디버거와 연결된 인터페이스를 제공한다.
- 메뉴바에 언어와 관련된 메뉴를 더한다.

게다가 특정언어와 관련되어 있지는 않지만 프로그래밍과 관련되어 있는 여러 가지 부가기능이 있다. 버젼 콘트롤기능, 코드에 자동적으로 설명추가, Makefile만들기, Change Logs 업데이트기능 들이다.

이러한 여러 가지 기능을 함께 잘 사용한다면 일반적인 상용 통합개발환경(IDE)와도 비교해서 뒤질 것이 없다. 게다가 물론 무료다.

### 3.2.1 C/C++/Java

C나 C++, Java는 거의 비슷한 문장을 가지기 때문에 하나의 이멕스 모드가 위의 세 가지 언어를 한꺼번에 다룰 수 있다. 이러한 모드는 이미 잘 만들어져 있기 때문에 이멕스와 기본적으로 배포가 된다. 이 모드는 `cc-mode`나 `CC Mode`로 알려져 있다.

좀더 자세한 내용이나 새로운 버전을 받기 위해서는 `http://www.python.org/emacs/`로 가보기 바란다.

### 3.2.2 Perl

이멕스에서 Perl 코드를 편집하기 위해서는 실제 두 가지 모드가 제공이 된다. 첫 번째 것은 `perl-mode`이고 두 번째 것은 `cperl-mode`이다. 왜 이렇게 두가지 모드가 있는지는 정확히 알 수 없지만 이멕스에서 Perl 코드를 편집하기 위해서는 `perl-mode`가 기본 모드니다. 그러나 이 모드는 `cperl-mode`보다 약간 기능이 적으며 Perl의 몇 가지 뛰어난 언어의 구조를 인식하는데 약간의 문제가 있다.

개인적으로는 `cperl-mode`를 사용하고 있으며 추천한다. 왜냐면 이 모드는 현재 꽤 활발하게 유지되고 있으며 지금까지 내가 원하던 모든 기능이 들어 있다. 마지막 판은 `ftp://ftp.math.ohio-state.edu/pub/users/ilya/emacs`에 있다.

그렇지만 내가 한 말에 치중하지 말고 두 가지를 다 사용해 보고 여러분에게 가장 마음에 드는 것을 고르기 바란다.

### 3.2.3 Python

Python도 또한 이멕스 모드가 있다.(현재 가장 많이 사용하는 스크립트 언어) 지금까지는 이멕스와 같이 배포되지는 않지만 XEmacs와는 배포가 된다. 그렇지만 두 가지 에디터에서 잘 작동을 한다.

여러분은 공식 파이썬 홈페이지 `http://www.python.org/emacs/python-mode/`에서 `python-mode`를 얻을 수도 있다.

### 3.2.4 다른 것들

프로그래머에 도움이 되는 다른 편집모드들이 있는데 다음과 같은 것이다.

- Shell Scripts (Bash, sh, ksh, csh, ...)
- Awk, Sed, Tcl, ...
- Makefiles
- Change Logs
- Documentation
- Debugging

그리고 이 문서의 마지막 부분에 더 많은 모드와 추가설치 내용들이 있다.

## 3.3 Authoring

재미있는 이멕스 모드는 단순히 코드만을 쓰는 사람들을 위한 것은 아니다. 다른 종류의 문서를 작성하는 사람들도 이멕스모드를 이용할 수 있다.

### 3.3.1 철자검색 (ispell mode)

여러 종류의 문서를 작성하는 사람들도 한번씩은 철자검색이 필요하다. 여러분이 **GNU ispell**를 설치하였다면 `M-x ispell`를 이용해서 현재 버퍼의 철자를 검색할 수 있다. 만일 ispell이 알 수 없는 단어를 찾는 경우 교체를 할 수 있는 단어들의 리스트를 보여주면서 여러분이 그중 하나를 선택할 수 있게 해 준다. 이 프로그램은 여러분들이 사용하는 다른 상용 프로그램의 철자 검색프로그램과 동일하다.

### 3.3.2 HTML (html-helper mode)

여러분들이 가끔 HTML파일을 작성하는 경우 `html-helper-mode`를 사용해 보기 바란다. `http://www.santafe.edu/\~{}nelson/tools/`에서 관련된 문서와 다른 것들을 포함해서 구할 수 있다.

이름이 보여주듯이 `html-helper-mode`는 고전적 스타일로 손으로 HTML코드를 사용하는 사람들을 도와주도록 만들어졌다.

### 3.3.3 TeX (tex-mode)

TeX을 이용해서 문서를 작성하는 경우 이맥스는 역슬래쉬나 중괄호 기호 등과 다른 여러 문자에 색과 강조기능을 더해준다.

사실 더 이상 TeX을 직접 사용하지는 않지만 예전에 사용할 때 이 모드를 이용하면 TeX 소스 파일을 좀더 쉽게 읽을 수 있었다.

### 3.3.4 SGML (sgml-mode)

지금 여러분이 읽고 있는 이문서는 SGML로 작성되었다. `sgml-mode`는 SGML문서의 모든 기본적인 요소를 제공해 준다. 예를 들어 validation, highlighting, forward-tag, backward-tag 등이 있다. 이것은 이맥스의 기본인 한 부분이다.

## 3.4 다른 모드들

물론 이맥스를 사용하기 편하게 하는 다른 모드들이 있는데 여기서는 그 가운데 자주 사용되는 몇 가지를 소개한다.

### 3.4.1 Version Control (vc mode)

`vc` 모드는 자주 사용되는 버전 제어 백엔드인 RCS, SCCS, CVS 등과 함께 파일의 포함과 제외나 판수를 관리를 쉽게 해준다. 이맥스의 기본적인 것중에 하나며 이맥스 문서에 이것에 대한 것을 찾을 수 있다.

### 3.4.2 Shell Mode

왜 간단한 몇 가지 쉘명령어를 사용하기 위해서 X윈도우나 가상 콘솔을 사용하는가..? 이맥스에서 번거로운 것을 해서 그 문제로부터 여러분들을 구하라. :-)

`M-x shell`명령어는 이맥스 버퍼에 쉘을 작동시킨다. 여러분이 쉘 상에서 사용하는 명령어를 실행시킬 수 있지만 `vi`나 `pine`과 같이 풀 스크린에서 작동하는 프로그램은 실행시킬 수 없다. 왜냐면 이맥스가 화면 뒤에서 여러분의 실제 쉘과 연결을 해서 실행시키기 때문이다..

이것 또한 이맥스의 기본적인 부분이고 이맥스 문서에서 찾을 수 있다.

#### 3.4.3 Telnet and FTP

텔넷이나 FTP를 사용하기 위해 다른 X 윈도우나 가상콘솔을 사용해야만 하는가 ? 이멕스에서 이것들을 직접 실행하라.

텔넷이나 ftp를 사용하기 위해서는 `M-x telnet`나 `M-x ftp`를 입력하라. 자세한 내용은 이멕스 문서를 보기 바란다.

#### 3.4.4 Man

매뉴얼 페이지를 읽으려고 X 윈도우나 가상콘솔을 이용해야하는가 ? 이것 또한 이멕스에서 사용할 수 있다.

이멕스 내의 쉘을 실행시켜 메뉴얼 페이지를 읽을 수 있다. `M-x man`를 입력해서 사용할 수 있다. 자세한 것은 문서를 보기 바란다.

#### 3.4.5 Ange-FTP

`ange-ftp`문서에서 인용하였다.

> 이 팩키지는 이멕스 내에서 간단하고 명확하게 파일 및 디렉토리에 접근하는 것을 용이하게 해준다. 일반적인 파일을 다루는 루틴을 FTP와 함께 사용할 수 있도록 확장하였다.

위의 말은 다른 컴퓨터에 있는 파일을 마치 여러분의 컴퓨터에 있는 것처럼 사용할 수 있게 해준다는 의미다. 그러므로 여러분이 다른 컴퓨터에 있는 파일을 편집할 경우 간단히 이멕스가 그 파일을 열라고 명령을 내리면 된다. 물론 약간 다른 경로를 사용한다. 파일을 다루는 모든 명령어는 기록이 되며 파일을 복구 할 수도 있다. 그리고 `C-x C-s`를 입력해 저장을 하면, `ange-ftp`는 파일의 저장과 쓰기를 다른 컴퓨터에 해 준다.

약간 다른 디렉토리 문법은 다음과 같다. 파일이름이 "myfile"이고 디렉토리가 "user"이며 컴퓨터가 "my.host.org"라면 파일을 (`C-x f`)를 이용해서 다음과 같이 열수 있다.

```
/user@my.host.org:~user/myfile
```

이것 또한 이멕스의 기본 팩키지이고 이멕스 문서에서 내용을 찾을 수 있다.

Etienne Grossmann (`etienne@anonimo.isr.ist.utl.pt`)에게 위의 예제에 대한 감사의 말을 전한다.

# 제 4 절 커스터마이징 이멕스

사실 모든 이멕스 커스터마이징은 Lisp코드를 통해 이루어진다. 이멕스를 수행하는데 필요한 변수를 바꿀 수 있으며 새로운 함수를 더하거나 기존에 있는 함수를 대체할 수도 있다.

## 4.1 일시적 커스터마이즈

이멕스를 커스터마이징할 때 여러분들이 이러한 것들을 일단 임시로 실행해 보길 원할 것이다. 만약 여러분들이 엄청 이상한 것을 했다면 단순히 `C-x C-c`입력해 이멕스를 닫은 다음 다시 실행하면 된다. 여러분이 수정한 내용이 지속적으로 유지되기 위해서는 그러한 내용을 여러분의 `.emacs`에 넣어서 이멕스가 실행될 때마다 읽히도록 하면 될 것이다. 이것은 다음 섹션에서 다루겠다.

### 4.1.1 변수에 대입

가장 쉽게 수정하는 방법은 이멕스 변수의 값을 바꾸는 방법이다. 이렇게 하는 코드는 다음과 같다.

```
(setq variable-name new-value)
```


```
(setq variable-name new-value)
```


`variable-name`는 변수의 이름이고 `new-value`는 여러분이 변수에 지정할 값이다. (Lisp에서 얘기를 할 때에는 변수에 값을 바인딩 했다고 한다.) lisp에서 `setq` 함수는 다른 프로그램에서 대입연산자(일반적으로 `=`)이다.

**주의:** 나는 여기서 간단히 얘기하기 위해서 자세한 내용은 대강 얼버무렸다. 아마도 여러분들은 나나 다른 사람들이 Lisp함수 `set`이나 심지어 `setq-default`를 사용하는 것을 보게 될 것이다. 여러분이 이 부분에 대해 많은 관심이 없다면 단지 그러한 것들은 이멕스의 Lisp과 관련된 것이라고 보면 된다.

자 이제 내 `.emacs` 파일의 한 줄을 보자.

```
(setq-default transient-mark-mode t)
```

`transient-mark-mode`은 문서의 일부분을 마크했을 때 그 부분이 강조될 것인가를 결정하는 변수이다. 많은 GUI 응용프로그램에서는 마우스를 이용해서 원하는 부분을 클릭 끌어 선택을 하면 화면과 반대색이나 다른 색으로 바꾸게 된다. 이멕스에서는 이러한 것을 `transient-mark-mode`를 설정하므로 할 수 있다.(nil이 아닌 값으로)

*어떠한 값?*

좋다. 간단히 얘기를 하자. 많은 프로그래밍 언어는 참값과 거짓값에 대한 기호를 가지고 있다. C/C++에서는 0이 아닌 값이 참값이다. Perl에서는 0이 아니거나 비어있는 값이 아니면 참이다. Lisp에서도 비슷한 것을 사용하는데 이름과 기호가 다르다.

참값은 `t`라고 하고 거짓은 `nil`이라고 한다. 다른 언어들과 동일하게 nil이 아닌 어떠한 값도 참이다.

`transient-mark-mode`에 대한 자세한 정보를 얻기 위해서는 온라인 도움말을 사용할 수 있다. `C-h v`나 `M-x describe-variable`를 입력하고 난 후 `transient-mark-mode`를 입력해라. 여러분이 나처럼 게으른 사람이라면 `Tab`을 사용해 완성기능을 이용할 수도 있다. 간단히 변수이름만 입력하고 `Tab`을 입력하면 된다. 만약 여러분이 이멕스가 유일하게 알고 있는 단어를 입력하는 경우 여러분을 위해서 단어가 완성되어 있는 것을 보게 될 것이다.

다른 사람들이 자주 셋팅을 해 두는 변수는 `fill-column`이다. 이 변수는 워드 랩핑을 위해 사용하는 화면을 크기를 결정해준다. (`auto-fill-mode`는 이 값을 사용한다.) 좀 이상하게 해 보기 위해서 다음과 같이 해봐라.

```
(setq fill-column 20)
```

위의 명령어는 어떠한 값을 지정하는 것은 아니다. 그러므로 여러분이 입력한 표현의 결과를 이멕스가 **계산**하라고 할 필요가 있다. 이렇게 하기 위해서는 표현의 마지막으로 가서 `C-x C-e`를 입력하면 된다. 이 명령은 `eval-last-sep`이라는 함수를 불러 값을 계산하도록 할 수 있다. 그렇게 하면 20이라는 숫자가 화면의 밑에 있는 미니 버퍼에 나오는 것을 볼 수 있다. 이것은 단순히 여러분이 계산을 원하는 표현의 값을 표시해준다.

위의 명령어가 작동하는 것을 확인하기 위해 한두 줄의 문장을 입력해 봐라. `auto-fill-mode`를 작동하게 했다면 문서가 20번째 컬럼에서 잘려지는 것을 보게 될 것이다. 그렇지 않으면 이미 여러분이 어떠한 문장이 타이핑된 후라면 `M-q`를 쳐서 `fill-paragraph`함수를 불러서 워드랩핑을 해 볼 수도 있다.

### 4.1.2 파일 연결

여러분들은 어떠한 특정한 파일을 열었을 때 이멕스가 자동으로 뭔가를 하도록 설정할 수 있다. (어떤 GUI 프로그램들은 여러분이 특정 파일에 대한 아이콘을 클릭 했을 때 자동으로 프로그램이 실행된다.) 예를 들어 `.txt` 확장자를 가지는 파일을 열 때마다 이멕스가 자동으로 `text-mode`로 설정되어 있도록 하는 것이다. 음.. 그런데 이건 이미 되어 있다. :-) 그래서 이번에는 여러분들이 "README" 파일을 열때 이멕스가 `text-mode`가 되도록 해보자.

```
(setq auto-mode-alist (cons '("README" . text-mode) auto-mode-alist))
```

헉?

사실 별로 알고 싶어하지 않는 Lisp 프로그래밍에 대해 배우는 것보다 일단 `auto-mode-alist`는 여러 가지의 조합을 가지고 있다라고 하자. 각각의 조합은 정규 표현과 이멕스모드를 가지고 있다. 여러분이 여는 파일이 정규표현과 일치한다면 (이 경우에는 `README`라는 단어) 이멕스는 여러분이 정의한 모드를 실행할 것이다.

그런데 재미있는 것은 그러한 모드에 다른 또 하나의 조합을 설정할 수 있다는 것이다. 여러분은 `auto-mode-alist`가 이미 가지고 있는 값에 대해서 정확히 알기 전에는 그것에 대응시키지 않을 것이다.

이멕스가 `.html`이나 `.htm`의 확장자를 가지는 파일을 열 때 자동으로 `html-helper-mode`로 바뀌도록 하려면 .emacs에 다음과 같은 것을 추가해두면 된다.

```
(setq auto-mode-alist (cons '("\\.html$" . html-helper-mode) auto-mode-alist))
(setq auto-mode-alist (cons '("\\.htm$" . html-helper-mode) auto-mode-alist))
```

이러한 가능성은 사실 끝도 없다.

## 4.2 .emacs파일 이용하기

여러분들이 이멕스를 사용해보고 여러 가지 기본적인 커스터마이징에 대해 알고 난 후에는 아마도 계속적으로 이러한 커스터마이징을 사용하고 싶을 것이다. (맘이 안 바뀌면 계속 사용하겠죠.) 여러분이 계속적으로 이멕스를 사용하다가 보면 여러분의 `.emacs` 파일이 날로 증가되는 것을 알게 될 것이다. 이러한 현상은 아주 좋다. 왜냐면 여러분은 이멕스가 어떻게 작동을 하는지 그리고 어떻게 하면 여러분이 원하는 것을 할 수 있는지 알고 있다는 얘기가 되기 때문이다. 여러분들이 이렇게 하지 못하는 소프트웨어는 좀 부끄럽다는 것을 알아야 된다.

여러분이 이미 이것을 예상하지 못했는지 모르겠지만 매번 여러분이 이멕스를 실행할 때 여러분의 홈 디렉토리에서 `.emacs`라는 파일을 찾을 것이다. 여러분의 `.emacs`파일에는 자동으로 실행이 되길 원하는 Lisp 코드와 지금 여기서 다루고 있는 커스터마이징에 관련된 부분들이 들어 있다.

내 `.emacs` 파일에서 볼 수 있는 다른 예제:

```
(setq inhibit-startup-message t)
```

`inhibit-startup-message` 변수는 이멕스가 시작될 때 환영 메시지를 나타낼지를 결정한다. 한동안 난 이것을 찾느라 고생을 했으나 (왜냐면 난 어떻게 도움말을 찾는지는 알고 있었기 때문에) 곧 찾아서 이 기능을 꺼버렸다.

연습삼아 직접 여러분의 `.emacs`파일을 만들고 위의 줄을 추가해 봐라. 이멕스를 나가거나 들어갈 때 더 이상 환영메시지를 볼 수 없을 것이다.

이멕스 모드나 팩키지에 관한 글을 읽을 때 종종 그 문서에는 모드나 팩키지가 정확히 작동하기 위해 여러분의 `.emacs` 파일을 수정할 필요가 있다고 할 것이다.

GNU 이멕스 FAQ(`C-h F`)에는 `.emacs`파일과 관련된 여러 가지 유용한 아이템들이 있다.

### 4.3 커스터마이즈 팩키지

이멕스가 계속적으로 사용되어지고 발전해 나감에 따라 "초보자들이 좀더 쉽게 수정할 수 있는 방법이 있어야만 한다"고 많은 사람들이 얘기를 해왔다. 그리고 커스터마이즈가 탄생했다.

커스터마이즈는 이멕스를 좀더 쉽게 자기 입맛에 맞게 고칠 수 있는 방법을 제공한다. 직접 한번 해보기 위해 Help메뉴의 부메뉴인 Customize를 보거나 M-x customize를 입력하면 된다.

커스터마이즈는 "Editing", "Programming", "File"과 같은 세부 그룹으로 구분된다. 어떤 그룹은 세부그룹으로 구성되어 있다.

여러분이 커스터마이즈 환경을 이용해서 수정을 하는 경우 이멕스는 여러분의 .emacs 파일에 기록을 한다. 이 방법은 좀더 다루기 쉽다. 왜냐면 이렇게 하면 바꾼 것을 보던가 아님 다시 되돌리는 것을 좀더 쉽게 해볼 수 있기 때문이다.

*난 이런 커스터마이즈 환경으로 수정하지 않기 때문에 이것에 대해서는 더 이상 얘기를 하지 않겠다.*

### 4.4 X윈도우 디스플레이

다른 X윈도우 응용프로그램과 같이 이멕스는 여러분의 X 리소스에 따라 결정이 된다. 이 말은 사용자가 초기 색, 크기나 다른 여러 X에 관련된 것들을 여러분이 마치 xterm이나 nxterm을 수정하듯이 조절할 수 있다.

여기에 관련된 나의 ~/.Xdefault파일이 있다.

```
emacs*Background: DarkSlateGray
emacs*Foreground: Wheat
emacs*pointerColor: Orchid
emacs*cursorColor: Orchid
emacs*bitmapIcon: on
emacs*font: fixed
emacs.geometry: 80x25
```

X 리소스에 대한 것은 여러분의 X 메뉴얼 페이지를 참고하기 바란다.

Chris Gray (cgray4@po-box.mcgill.ca)는 다음과 같이 얘기한다.

> 데비안에서는 ~/.Xdefaults를 사용하지 않는다. 대신 데비안 사용자들은 /etc/X11/Xresources/emacs 파일에 관련된 사항을 넣어두면 RedHat를 사용하는 경우와 동일한 결과를 얻을 수 있다.

## 제 5 절 많이 사용하는 팩키지들

이멕스에는 여러 가지 다른 모드를 사용할 수 있기 때문에 부가적인 여러 개의 부속**팩키지**들이 있다. 이것을 팩키지라고 부르는 이유는 새로운 모드 이상의 것들이 때문이다. 추가적인 유틸리티들을 포함하거나 또는 너무 크기 때문에 이러한 것들을 단순히 모드라고 하기에는 맞지 않다. 다른 경우에는 이 팩키지는 다른 이멕스 모드나 팩키지와 결합되어 기능을 향상시킨다. 차이점을 쉽게 알기는 어렵지만 일단 받아들이자.

### 5.1 VM (Mail)

VM FAQ로부터 인용

> VM(View Mail)은 이멕스 안에서 메일을 보거나 정리할 수 있는 이멕스의 부속 시스템이다. 일반적인 메일클라이언트처럼 답장을 보낸다든지 폴더에 메일을 저장하거나 지우는 등 여러 가지를 할 수 있다. 또한 좀더 뛰어난 기능이 있는데 다이제스트를 만들거나 실행하고 메시지를 전달하고 여러가지 기준에 따라 메시지의 형태를 구성할 수 있다.

처음 내가 이멕스를 사용할 때에는 잠시 동안 VM를 사용할 수 있도록 시도만 해 봤었다. 그러다가 이 VM이 Pine이나 Elm과 같은 다른 메일 프로그램과 동일하다는 것을 발견했다. 그래서 더 이상은 메일을 보고 뉴스를 읽기 위해 다른 프로그램을 사용하지 않는다. VM은 계속적으로 활발히 개발되고 오늘날에도 잘 지원되고 있다.

`http://www.wonderworks.com/vm/`에서 얻을 수 있다.

## 5.2 Gnus (Mail and News)

GNUS 메뉴얼에서 인용:

> Gnus는 메시지를 읽기 위한 모든 일이 이루어지는 곳이다. 당신은 Gnus를 이용하여 모든 것들을 마치 newsgroup인 것처럼 바라볼 수가 있다. Gnus를 통하여 메일을 읽을 수도 있고, directory를 훑어볼 수도 있고, ftp를 할 수 있다. – 심지어는 Gnus를 이용하여 news를 읽을 수도 있다.
>
> Gnus는 이멕스가 문서를 편집하는 사람들에게 많은 도움이 되었듯이 뉴스를 읽는 사람들에게도 많은 도움이 된다. Gnus는 이러한 것을 하는데 어떠한 제한도 없으며 사용자들은 원하는대로 Gnus를 확장할 수 있다. 프로그램이 사람들을 제어하는 것이 아니라 사람들이 그들이 원하는대로 프로그램을 사용해서(혹은 오용해서) 프로그램을 수정할 수 있다.

Gnus는 현재 내가 메일과 뉴스를 위해 사용하는 프로그램이다. Gnus는 오늘날 활발히 개발되고 있으며 잘 지원되고 있다.

`http://www.gnus.org/`에서 구할 수 있다.

## 5.3 BBDB (A rollodex)

BBDB는 음흉한 Big Brother용 데이타베이스이다.(소설 1984에 나오는 Big brother를 연상하면 됩니다.) Emacs에서 대부분의 Emacs 메일 패키지(VM과 GNUS포함)와 동작하도록 만들어진 rollodex와 비슷한 프로그램입니다.

`http://pweb.netcom.com/\~{}simmonmt/bbdb/index.html`에서 찾을 수 있다.

## 5.4 AucTeX (다른 TeX mode)

AucTeX는 TeX 파일을 편집할 수 있는 또 다른 모드이다.

AucTeX 웹사이트에서 인용:

> AUC TeX는 GNU Emacs의 여러 유사 프로그램들을 위해 만들어진 TeX 파일 쓰기, 포멧팅 팩키지이다. AMS TeX나 LaTeX, TeXinfo를 포함해서 많은 다른 메크로 팩키지들이 지원된다.

`http://sunsite.auc.dk/auctex/`에서 찾을 수 있다.

# 제 6 절 다른 자료들

이 장은 여러분들이 이멕스에 관련된 것들에 대한 정보를 찾을 수 있는 책이나 웹싸이트, 뉴스그룹, 메일링 리스트에 관한 것을 다룬다.

## 6.1 책

이멕스를 배우는데 좋은 책은 정말 드물다. 게다가 여러분은 많은 Linux나 Unix관련 책들이 한두 장에 걸쳐 이멕스(와 `vi`)를 다루고 있다는 것을 알게 될 것이다.

### 6.1.1 Learning GNU Emacs

저자: Debra Cameron, Bill Rosenblatt, Eric S. Raymond

출판사: O'Reilly & Associates - `http://www.ora.com/`

여러분은 Amozon.dom에서 관련 프로그램에 의해 저렴한 가격으로 구입할 수 있다. `http://www.amazon.com/exec/obidos/ASIN/1565921526/`

**서평:** 이 책은 아마 시작하기에 가장 좋은 책일 것이다. HOWTO를 읽고 난 후 FAQ를 찾아본 다음에 이 책이 자세하고 직접적으로 사용 가능한 지도서라는 것을 알게 될 것이다.

### 6.1.2 Writing GNU Emacs Extensions

저자: Bob Glickstein

출판사: O'Reilly & Associates - `http://www.ora.com/`

여러분은 Amozon.dom에서 관련 프로그램에 의해 저렴한 가격으로 구입할 수 있다. `http://www.amazon.com/exec/obidos/ASIN/1565922611/`

**서평:** 이멕스를 한동안 사용을 해보고 여러분이 여러분 자신만의 모드를 만들거나 좀더 커스터마이징을 하고 싶다면 바로 이 책을 읽기 바란다. Lisp에 대해서 설명을 하고 있지는 않지만 간단한 안내문이 있다.

### 6.1.3 Programming in Emacs Lisp: An Introduction

저자: Robert J. Chassell

README 파일에서 From the README file:

> 이 책은 프로그래머가 아니거나 실질적으로 프로그래밍에 관심이 없지만 자신의 컴퓨팅 환경을 확장하거나 커스터마이징을 하고 싶은 사람들이 이멕스의 Lisp 프로그래밍을 할 때 사용할 수 있는 기초 안내서이다.

여러분은 메뉴얼을 전부 GNU FTP 서버에 anonymous로 접속해서 얻을 수 있다.

또한 Amozon.dom에서 관련 프로그램에 의해 잘 프린트된 버젼을 구입할 수 있다

**서평:** 여러분이 프로그래머가 아닐지라도 이멕스 Lisp 프로그래밍을 할 수 있는 좋은 안내서이다.

### 6.1.4 The GNU Emacs Lisp Reference Manual

저자: Richard Stallman

출판사: The Free Software Foundation - `http://www.fsf.org/`

여러분이 이 메뉴얼을 전부 GNU FTP 서버에 anonymous로 접속해서 얻을 수 있다.

**서평:** 이멕스 Lisp 프로그래밍 언어를 위한 결정판 가이드이다.

## 6.2 웹싸이트 Web Sites

### 6.2.1 EMACSulation

EMACSulation은 온라인 잡지인 Linux Gazette `http://www.linuxgazette.com/`에 Eric Marsden이 쓴 컬럼이다. 가장 최근 컬럼은 `http://www.linuxgazette.com/issue39/marsden.html`에 있다. 기사의 맨 마지막 부분에 예전 기사에 관한 것이 있다.

## 6.3 Newsgroups

여러분의 뉴스서버에서 "emacs"에 관련된 뉴스그룹을 찾아봐라. 아마 여러 가지가 나올텐데 밑에 있는 것은 나의 뉴스서버에서 찾은 것들이다.

- comp.emacs
- comp.emacs.sources
- gnu.emacs
- gnu.emacs.bug
- gnu.emacs.help
- gnu.emacs.sources

## 6.4 Mailing Lists

이멕스에 관련된 메일링 리스트는 내가 지금 알기에 NT-Emacs 리스트이다. 이 리스트는 윈도그에서 이멕스를 사용하는 사람들을 위한 것이다. 좀더 자세한 정보는 NT-Emacs FAQ `http://www.cs.washington.edu/homes/voelker/ntemacs.html`에 있다.

## 6.5 The Emacs Lisp Archive

이멕스 Lisp Archive README에서

> ftp.cis.ohio-state.edu에 있는 이멕스 Lisp 아카이브에는 여러 가지 이멕스 Lisp 코드가 있다. 이멕스 Lisp는 자유재단에 의해 만들어진 이멕스의 기능을 확장하기 위해서 사용되는 언어이다. 많은 이멕스 Lisp 코드가 GNU 이멕스와 같이 배포되지만 많은 사람들이 다른 시스템과 연결을 하거나 또는 그들이 사용하는 프로그래밍 언어를 좀더 잘 편집하고, 새로운 기능을 더하고, 이멕스의 기본 동작을 바꾸기 위해 팩키지를 쓰고 있다. 이 아카이브의 많은 부분은 개인들에 의해 만들어졌으며 info-emacs나 info-gnu-emacs, comp.emacs, gnu.emacs, gnu.emacs.sources와 같은 메일링 리스트를 통해 인터넷에 배포된 것으로 구성이 되어있다.

아카이브는 `ftp://ftp.cis.ohio-state.edu/pub/emacs-lisp/`에서 anonymous FTP를 통해 얻을 수 있다.

**NOTE:** 내가 말할 수 있는 것 이멕스 Lisp 아카이브는 너무 느려서 거의 쓸모 없다. 내가 이미 존재하는 것을 안 팩키지도 그곳에서 거의 본 적이 없다. 일반적으로 `comp.emacs.sources` 뉴스그룹에 Lisp 팩키지들이 올라온다. (만약 내 생각이 잘못됐다면 고쳐달라.)

# 제 7 절 감사의 글

이 문서를 완성하는데 다음의 사람들이 도와주었다.

- Robert Vollmert rvollmer@gmx.net
- Larry Brasfield larrybr@seanet.com
- Etienne Grossmann etienne@anonimo.isr.ist.utl.pt
- Thomas Weinell kf6mli@amsat.org
- Adam C. Finnefrock adam@bigbro.biophys.cornell.edu
- Chris Gray cgray4@po-box.mcgill.ca
- Robert J. Chassell bob@rattlesnake.com
- Isaac To kkto@csis.hku.hk
- Matteo Valsasna valsasna@elet.polimi.it
- Tijs van Bakel smoke@casema.net