

mystart를 이용하는 문서 작성

Nova De Hi*

2010년 1월 1일

이 글에 대하여

ko \TeX Live가 성공적으로 제작되었다. 그리고 KC2008Plus도 함께 제공되고 있다. 이 글은 KC2008Plus의 작은 유틸리티인 mystart의 사용법을 소개하는 것이다.

1 ko \TeX Live로 책을 만들다

2009년 12월 31일, 나는 어떤 책의 마지막 교정과 최종 출력물 제작을 해야 했다. 출판사로 직접 가서 마지막 작업을 함께 하기로 하였는데 문제는 내가 가지고 다닐 만한 노트북이 없었다는 것. 그래서 나는 작업 파일을 USB 메모리에 넣어가지고 컴퓨터는 출판사의 것을 빌어쓰기로 작정하였다. 이 책은 ko \TeX Live로는, 그리고 X \TeX 으로는 처음 만드는 본격적인 단행본이 될 것이다.

내가 준비한 파일은 다음과 같다. 최종작업이었기 때문에 스타일이나 그런 것은 사전에 미리 다 준비되어 있었다. 하루만에 디자인까지 처음부터 책을 만들지는 못한다.

1. 매크로와 원고 .tex, .sty 파일
2. texmf-local 형식으로 이 책에서 사용하는 그러나 \TeX Live 기본 배포판에는 없는 폰트 파일들. 주로 상업용 폰트들로 이루어져 있다.

용량은 200메가가 조금 넘었다. 책에 들어갈 삽화 등도 같이 들어 있기 때문에 용량은 적은 편은 아니다. 아무튼 이 USB만을 달랑 들고 작업하러 갔다.

컴퓨터를 하나 빌려달라 하니 Windows Vista Home Edition이 운영체제인 것으로 쓰라 한다. 비스타..., 괜찮을까? Aero effect도 활성화되어 있지 않은 비스타를 쓰려니 마치 Virtual Box에서 쓰는 것 같은 기분도 들지만 반응속도는 괜찮았다.

일단 ko \TeX Live basic 버전을 설치했다. 설치 자체에는 약 3분 정도가 걸렸나? 다운로드까지 해서 좌우간 5분 이내에 끝난다. 그리고 KC2008Plus도 설치. 이걸 1분 이

* 최근 KTUG 사이트에 Progress 님이 '노바디'가 쓴 글에 대해 언급하신 것이 있다. 이 페이지를 구글 번역기에게 영문 번역을 하라고 시켰더니 노바디를 'Nova De Hi'라고 번역해주더군. 이거 마음에 들어서 필명으로 쓰기로 한다. 구글 카미사마께서 해주신 작명이다.

내에 설치가 종료된다. 그런데 비스타는 (분명히 환경변수와 레지스트리 등록이 되어 있음에도 불구하고) 설치된 환경이 활성화되지 않는다. 그래서 재부팅. 재부팅 후에 KC2008Plus의 "Setting up Inverse Search" 를 클릭하니 sumatrapdf 다운에서부터 시작해서 인버스 서치 설정까지 단숨에 끝내주었다.

그 다음 단계는 필요한 패키지를 설치하는 것인데, 일단 기본적으로 생각나는 대로 몇 개를 설치했다. 명령행 열기는 바탕화면의 낫흐파드뿔뿔을 더블클릭한 후에 Alt A 를 눌렀다.

```
C:> tlmgr install memoir lm xkeyval xcolor mathdesign listings
C:> tlmgr install caption ms eso-pic wallpaper pdfpages pgf metalogo
```

이제 준비해 온 texmf-local로 교체했다. 그런 후에 fontcache. 이걸로 작업 준비 끝. 여기까지 이것저것(비스타 때문에 발생한 시행착오까지 포함하여) 20분 정도 걸린 듯. 굴과 커피를 주면서 드시면서 하시란다. 설치 과정이 너무 빠르게 진행되는 바람에 굴 까먹을 시간도 없었음을 고백해둔다. 이 굴들은 나중에 본격적인 작업에 들어가서야 비로소 옷을 벗었다. 이호재 님 늘 강조하시는 "담배 피울 시간"이 아쉬웠다. 그래서 잠깐 쉬러 밖으로.

본격적인 작업이 시작되었는데, 지금까지 내가 겪은 여러 편집 상황 중에서 가장 편한 작업환경이었던 것이다. pdf에서 search 후 더블 클릭하면 즉시 소스의 제 위치로 찾아가는 것이, 수십 개의 서브파일들로 이루어진 이 방대한 프로젝트에서도 아주 쉽게 동기화가 되는 것이 일의 효율을 200% 정도 높여주었다. 화면이 와이드였던 것이 매우 크게 작용했다. pdf 프리뷰 창이 한 화면 안에 제자리를 차지하고 내가 하는 작업을 바로바로 반영해주고 있는 것은 참 감동적이었다. 집의 내 컴퓨터는 아직도 와이드 화면이 아니기 때문에 화면이 넓직한 것이 이렇게 좋은 건 줄 거의 처음 알았다고 해야 하나.

옆 자리에서 같이 작업한 편집 담당 직원이 묻는다. "pdf에서 여기 클릭하면 해당 위치로 가는 거는 저절로 되는 기능인가요?" "예, 저절로 되죠." "신기하네요." 이 대화는 pdf-편집기 동기화에 대한 대화가 아니고 pdf hyperlink에 대한 것이다.

koTeX Live 2009 —특히 basic— 은 걸작이다. KC2008을 무척 자랑스러워했던 나로서는 그보다 더 cute해진 koTeX Live 2009에 이르는 모든 과정이 이것을 만들기 위한 시행착오였다고까지 생각하게 되었다. 그 편리함과 강력함은 겪어보지 않은 사람은 절대로 모른다. 말로는 다 설명할 수 없네.

그러나 시스템이 아무리 똑똑해도 사용자가 멍청하니까 역시 어쩔 수 없더라. 나는 이 날, 집에 가는 차시간에 맞추기 위해서 부득이(라고 변명한다) \nofiles를 썼다는 것을 고백해둔다. 사람에게 자존심이 있다면 \TeX 편집자는 \nofiles만은 안 쓰는 것이 자존심일 수도 있는데.....

이 장황한 경험담은 \TeX 문서작업에서 작업 환경이 얼마나 중요한가를 새삼 깨닫는 계기가 되었다는 것이 결론이다. 그리고 koTeX Live와 KC2008Plus가 충분히 만족스런 작업 환경을 제공해주더라는 것도 포함하여.

2 작은 유틸리티 mystart

KC2008Plus에는 mystart라는 작은 배치파일이 하나 들어 있다. 이것은 나의 블로그에 그 만들어진 과정을 이미 올린 적이 있으나 내 블로그 독자가 무려 3-4명에 이르는 관계로 아는 분만 아시는 상황이 되어 있다. 그래도 이 유틸리티에 대해서만은 꼭 소개 하고 싶어서 이 글을 쓴다.

자신의 작업 전 과정을 생각해보자. 과연 어디에서 시간 손실이 가장 많을 것 같은가? 솔직하게 얘기하면 "내용을 만드는 것"이 정답이다. 그런데 그렇게 얘기가 진행된다면 곤란하므로 일단 내용은 만들어져 머릿속에 다 들어 있고 이제 그걸 글로 쓰기만 하면 되는 —있을 법하지 않은— 상황을 전제하자.

예전부터 \LaTeX 사용 안내 등에서는 "한 글자 타자하고 바로 컴파일해서 자신이 입력한 한 글자가 제대로 나오는지 확인하는" 작업 습관에 대해 상당히 비판적인 글이 많았다. 그러나 오늘날과 같은 비주얼시대에, 이런 습관을 계속 타박하기란 참 힘든 일이다. 어차피 궁금해지는 것은 해결하도록 해줘야 하지 않나? 말할 것도 없이 이런 습관이 그다지 장려할 만한 것은 아니다. 상상력이 그 간극을 메울 수 있도록 습관을 들이는 것은 당연히 좋다. 그러나 문제는 거기 있는 것이 아니라, 결과를 확인하기 위해서 상당한 시간을 소비해야 한다는 바로 그 점이다.

왜 dvi가 그토록 오래도록 살아 남았는가? 왜 pdf로의 스위칭이 바로 이루어지지 못했을까? 답은, Adobe Reader의 독특한 동작에 있었다. Adobe Reader를 뷰어로 쓰는 한, 자신이 입력한 한 글자가 식자되는 결과를 보려면 다음 과정을 거쳐야 했다.

1. Adobe Reader를 닫는다. Adobe Reader를 닫지 않으면 열린 pdf 파일을 쓰기(write)할 수 없기 때문에 꼭 이걸 닫아주어야 한다.
2. 컴파일 버튼을 누르거나 명령을 실행한다.
3. pdf가 만들어지면 이것을 Adobe Reader로 다시 연다.
4. 아까 작업 중이던 페이지로 **PgDn** 키를 눌러서 찾아간다. 작업 중이던 페이지가 253쪽이었다면 **PgDn**을 200번 정도 누르든가 아니면 페이지 가기로 아까 기억해둔 페이지로 이동하든가.

이거야말로 인생의 낭비였다. WinEdt이 각광받은 이유 중의 하나가 Adobe Reader 닫기와 열기를 에디터 매크로로 해주었기 때문이라는 점을 생각하면, 그리고 KC2006 시절에 pdfopen/pdfclose 유틸리티가 중요하게 취급되었던 점을 생각하면 이 상황이야말로 \TeX 작업이 피곤하게 느껴지는 최대의 난관이었던 것이다.

KC2008과 그 이후의 texworks 에디터는 이 문제를 근본적으로 해결했다. 즉 열려 있는 pdf를 닫지 않아도 컴파일이 가능하고 결과의 자동갱신이 이루어지며 소스와 pdf 사이의 바로찾아가기(동기화)가 구현되어버렸던 것이다. 그것도 Windows에서!

이거 사실 엄청난 거다. 이걸 \TeX 작업을 WYSIWYG로 만들었다. WYSIWYG라는 게, 자신이 입력한 결과를 바로 볼 수 있는 거라고 이해하기로 한다면 입력한 결과를 즉시즉시 볼 수 있다면 그게 바로 WYSIWYG 아니겠는가. 앞서 말한 '인생의 낭비'를

돌이켜보라. 이 새로운 상황에 열광하지 않으면 안 된다. 너무나 당연한 게 구현된 거라고 다들 생각하고 계신 듯도 하지만.

mystart의 발상은 매우 단순하다. 이것은 원래 \TeX daemon에서 아이디어를 가져온 것이다. 작업 중인 문서에 대하여 백그라운드로 컴파일이 계속해서 이루어진다면 pdf 뷰어 화면에서는 자신이 입력하는 글이 즉시 갱신되지 않겠는가? 예전 TeXtures[®] 처럼.

그런데 가만히 생각해보면, 이게 만만한 일이 아니다. 입력이 계속되고 있는 동안 백그라운드로 컴파일하여 결과를 지속적으로 갱신해가기 위해서는, 현재 입력된 위치의 전후 환경을 검사하여, 만약 열린 환경이 닫히지 않았다면 임시로 닫아주어야 하고 열린 `\bgroup`을 임시로 `\egroup` 처리해주어야 하며 에러가 발생한 행은 무시하게 해야 하는 등 복잡한 문제가 한두 가지가 아닌 것이다. 그러자면 매번 메모리 상에서 현재 입력 중인 문서에 대응하는 임시 문서를 생성하고 이것을 처리해서 현재 입력 중인 문서의 `synctex` 위치 정보를 재할당하여 pdf를 생성하여야 하는 난관이 가로놓여 있는 것이다. 그런 면에서 (pdf 기술을 이용한 것은 아니었지만) TeXtures[®]는 놀라운 프로그램이었던 셈인데, 그건 그렇다치고 지금 우리가 이 모든 문제를 일거에 해결하는 것은 사실상 불가능에 가깝다. \TeX daemon이라는 프로그램은 이 문제를 다음과 같이 우회하였다.

1. 입력에 대하여 항상 문서를 재컴파일하는 것이 아니라 문서 파일이 갱신되는 시점, 즉 “저장되는” 순간 백그라운드 재컴파일한다.
2. 따라서 사용자는 문서를 저장하는 순간에 열어둔 환경을 닫고 에러 발생이 나타나지 않도록 하여 저장하기만 하면 저장 키 한 번으로 화면을 매번 갱신해준다.

이것은 매우 적당한 선택인 것 같다. 저장하는 순간 갱신해준다는 아이디어에 대해서 말했더니, “texworks의 `Ctrl-T`를 누르는 것과 뭐가 다르냐”고 질문하신 분이 계셨는데, 많이 다르다. 결정적인 차이는, 백그라운드 컴파일이 진행되고 있는 그 순간에도 문서는 계속 편집 가능하다는 것이다. 이거 의외로 무시못하는 점이다.

mystart는 이러한 저장-시점-백그라운드-컴파일-뷰어갱신이라는 전략을 채택하고 그 환경을 배치파일로 즉시 설정할 수 있도록 한 것이다. 단 여기에는 \TeX daemon이 아니라 `kcltxmk`를 사용했을 뿐이다. `kcltxmk`와 그 원본인 `latexmk`는 현재 작업 중인 문서의 관련 파일의 변경 여부를 검사하는 기능을 가지고 있는데, 이를 이용한 것이다.

이것을 제대로 쓰려면 코딩 습관을 일부 바꿔야 한다. 자신이 저장 키를 누르는 순간에 그 문서에 에러가 발생할 것 같은 상황을 만들지 않는 쪽으로 코딩하는 습관을 들여야 하는 것이다. 예를 들면

```
\begin{enumerate}
\item
```

까지 쓴 상태에서 저장하면 에러가 나올 것이다. 열린 `enumerate`이 닫히지 않았기 때문이다. 그러므로

```
\begin{enumerate}

\end{enumerate}
```

과 같이 먼저 코딩한 다음, `\item` 항목은 나중에 그 사이에 채워 넣는 방식이 필요하다. 이것은 아주 약간 습관을 바꿀 것을 요구한다. 물론 모든 에러를 완전히 피하는 방법은 없다. 에러가 발생한 경우 이에 대처하는 방법은 뒤에 따로 서술한다. 여기서 강조하고자 하는 것은 어쩔 수 없는 에러 말고 불필요한 에러를 일부러 만들지 않도록 하자는 것이다.¹

3 mystart 사용 방법

이 배치파일은 명령행에서 실행한다.

```
C:> mystart [engine] [projectname]
```

실행이 성공적으로 이루어지면, 네 개의 창을 화면에 띄운다. 각각 편집기(Notepad++), pdf 뷰어(sumatrapdf), 백그라운드 컴파일 창(cmd), 마지막 하나는 Adobe Reader이다 (그림 1은 Adobe Reader를 닫은 상태). Adobe Reader는 최종 출력물의 상태를 확인하기 위하여 뜨는 것인데 작업을 위해서는 닫아두는 것이 좋다. Adobe Reader가 pdf 파일의 쓰기(write)를 잠그기 때문이다. 나중에 편집기로부터 Adobe Reader 최종 화면을 언제든지 열어볼 수 있다.

이제 편집기에서 `Ctrl-S`를 누르거나 저장 메뉴를 선택하는 순간, 백그라운드 컴파일 창이 시작된다. 배경에서 컴파일이 이루어지고 있는 순간에도 편집기에서는 수정과 추가가 가능하다. 컴파일이 종료되면 뷰어가 갱신된다. 거의 실시간 작업이라는 느낌이 들 것이다. 컴퓨터가 충분히 빠르고 문서양이 얼마 되지 않는다면 저장에서 갱신까지 시간차를 거의 느낄 수 없을 정도라고 한다. 이 문서는 상당히 느린 기계에서 작성하고 있는데도 참을 수 있을 정도의 갱신 속도를 보여주었다.

명령행을 열고 실행시키는 것이 불편하다고 생각할 수 있는데, 여기에 대해서는 먼저 Notepad++를 실행한 다음, 여기에서 `Alt-A`로 커맨드 창을 열어서 `mystart`를 써넣는 쪽이 편리하다. 이미 문서가 편집기에 열려 있더라도 새로운 Notepad++가 또 뜬다거나 하는 일은 없다.

이제 `mystart`의 옵션과 실행 방법을 각 경우에 따라 살펴보자.

¹cron 같은 것을 걸거나 에디터의 자동 저장 기능을 이용하면 어떠한 제의가 있었다. 그러나 에러 발생이라는 부분 때문에 예를 들어 자동 저장 기능이 시간 단위로 동작할 경우, 저장되는 시점에 모든 환경이 다 닫혀 있다고 보증할 수 없기 때문에 이 방법은 좋은 점도 있겠지만 문제를 귀찮게 만들 수도 있다.

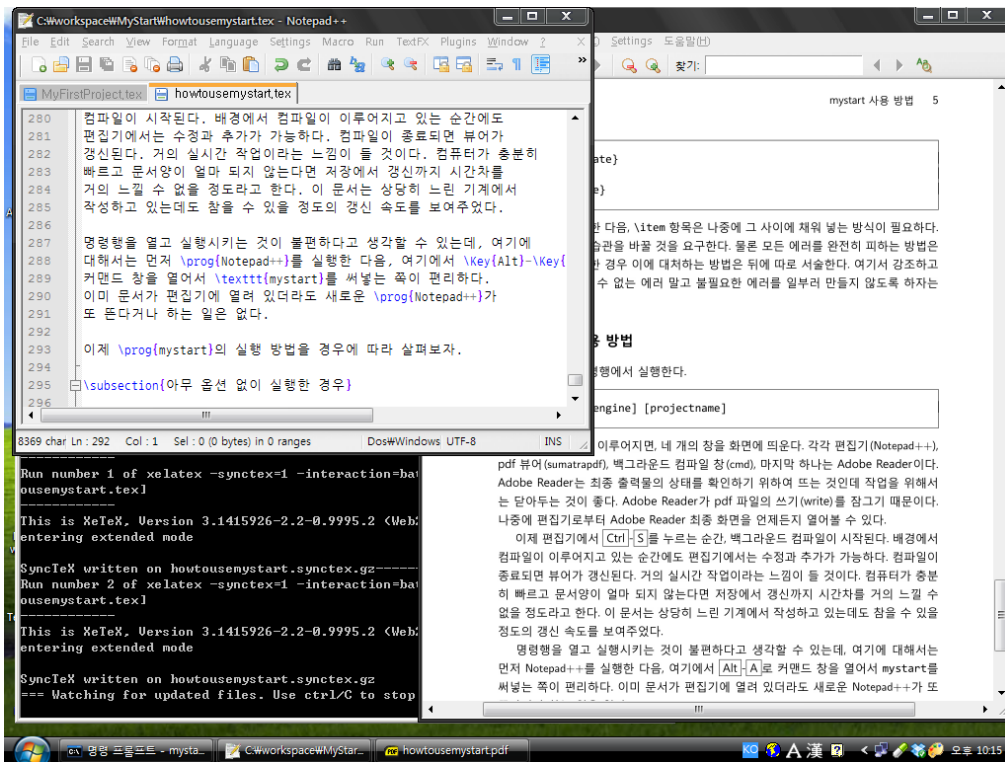


그림 1: mystart로 구성한 작업 환경

3.1 아무 인자 없이 실행한 경우

아무 옵션도 주지 않고 실행하면 간단한 사용법을 다음과 같이 보여준다.

```
Usage : mystart <engine> projectname
engine is one of
    xe = xelatex
    pdf = pdflatex (default)
    dpx = latex and dvipdfmx
    ps = latex, dvips and ps2pdf.
When the engine isn't given, the first argument will be
regarded as projectname.
```

그런 다음 새로운 프로젝트를 시작한다. 먼저 엔진을 선택하게 한다.

```
+++++++ select engine ++++++++
Select Processing Engine
[Enter]
```

디폴트가 pdflatex이기 때문에 그냥 `Enter`를 치면 pdflatex이 선택된다. 다른 엔진을 원하면 `Enter`를 치고 `Enter`하면 된다. 우리는 xe라고 써넣어보기로 하자.

```
You selected pdfxe.
+++++++ project file ++++++++
default project name is newprj.tex
Enter your project file name without extension.
```

그 다음은 파일 이름(또는 프로젝트 이름)을 선택하는 것이다. 그냥 `Enter`만을 치면 파일 이름이 newprj.tex으로 설정된다. 그리고 만약 이 파일이 없으면 새로 만들면서 환경 설정을 위한 초기화를 행한다. 초기화라는 것은 일단 해당 파일을 컴파일하여 부수파일 등을 만든 후에 각각의 파일에 대하여 변경 여부를 체크하기 위한 설정을 행하는 것을 말한다. 즉, 두어 번 컴파일이 진행된다.

파일 이름을 입력하면 해당 파일이 있는지를 체크한다. 만약 tex 파일도 있고 pdf 파일도 있으면 이 두 파일을 이용해서 작업 환경을 설정한다. pdf가 없으면 컴파일을 진행하여 먼저 이를 만드는 과정이 진행된다.

파일 이름을 입력하지 않은 경우에도 해당 폴더에 newprj.tex이 있으면 그 이름을 입력한 것과 같은 방식으로 동작한다.

마지막 단계에 가서 "Press anykey..." 메시지가 나오면 아무 키나 누른다. 그러면 네 개의 프로그램이 차례로 실행된다. 앞서 말한 대로 Adobe Reader를 죽이면 작업 준비가 끝나는 것이다. 이제부터 tex 파일을 저장하기만 하면 백그라운드 컴파일과 pdf 갱신이 계속된다.

여러 파일로 이루어진 프로젝트의 경우에는 어떠한가? 부수 파일 중의 하나라도 변경되면 main file이 컴파일되는데, 이것은 매우 편리하다. 심지어 .aux 파일만 변경되어도 컴파일이 새로 진행된다. 다만 부수 파일은 에디터에서 직접 열어 편집해야 한다. 그러므로 명령행에서는 항상 main file을 인자 이름으로 주어야 한다. 부수 파일을 mystart의 인자로 지정하면 부수 파일 단독으로 컴파일되지 않아서 장애가 발생한다.

3.2 명령행에서 인자 지정하기

다음과 같은 몇 가지 형식으로 실행할 수 있다.

mystart file filename만을 인자로 지정하고 실행하면 엔진은 pdflatex으로 설정되고 filename을 projectname으로 간주하여 환경설정을 행한다.

mystart xe file 첫째 인자로 엔진을, 둘째 인자로 projectname을 지정하는 방법이다. 앞서 설명한 대로 projectname.tex이라는 파일이 없으면 만들고 pdf를 먼저 생성하는 과정을 거친다. 엔진 이름으로는 xe, pdf, dpx, ps 넷 중의 하나를 지정해야 한다. 만약 이들 가운데 하나가 아니면 첫째 인자가 projectname으로 간주된다.

mystart . file 엔진 이름에 그냥 점만 찍는 방식인데, 이것은 엔진을 pdflatex으로 설정하기 때문에 점을 안 찍은 경우와 같다.

없는 프로젝트를 새로 만들려 할 때 반드시 주의해야 할 점이 있다. 그것은 새로 만들어지는 파일 이름이면 newprj.tex의 내용이 한글이 한 글자도 없는 영문만으로 된 ANSI 인코딩 파일이라는 점이다. 여기에 한글을 추가하였을 때 반드시 Notepad++의 Format(형식) 메뉴에서 "Convert to UTF-8(UTF-8으로 변환)"을 한 번 지시해주어야 한다는 것이다. 즉, 파일을 Unicode UTF-8으로 저장하도록 하라.

4 에러 극복하기

배경의 컴파일 창에서는 에러 여부를 보여주지 않는다. 이것은 실행 속도를 높이기 위한 선택이었다.² 그러므로 저장 직전에 자신의 작업 위치에서 다음 사항을 염두에 두도록 하자.

1. 열린 중괄호는 반드시 닫아준다.
2. 행중 수식은 되도록 시작과 끝을 알아보기 쉽게 \ (와 \)로 작성한다. \$로 행중 수식을 작성하는 관행을 극복하는 것이 좋다.
3. \begin 행 다음에는 반드시 \end 행을 먼저 써놓고 그 다음에 가운데 내용을 채운다.
4. 중괄호로 묶는 경우 일단 { }와 같이 먼저 괄호를 닫아 놓고 그 후에 괄호 안의 내용을 써넣는다.
5. 습관적으로 \TeX 명령에 오타를 내는 사람이 있다. 적어도 \TeX 명령에서는 오타를 내지 않도록 주의한다.
6. 되도록이면 저장 간격을 너무 밟지 않게 한다. 적당한 간격으로 컴파일해가면 에러 발견이 훨씬 쉽다. 정말 한 글자 치고 저장하고 하는 것은 때로 곤란한 상황을 유발할 수도 있다.

그래도 에러가 발생한다. 에러가 발생한 경우 보여주는 동작은 대략 다음과 같다.

- 배경의 백그라운드 컴파일 창이 정지하지 않고 컴파일을 계속 반복한다. 지연된 컴파일을 하는 수도 있으므로 정말 에러로 인한 반복 컴파일인지 잘 확인하는 것이 중요하다.
- 변경 후 저장을 하고 있는데도 pdf 내용이 전혀 갱신되지 않는다.

²에러 발생시 정지하게 하는 방법도 있다. 이것은 mystart.bat의 kcltxmk 실행 옵션에서 -f를 빼주면 된다.

- sumatrapdf가 pdf를 로드하지 못하고 죽어버린다.

각각의 경우에 대한 처리 방법을 간략히 적어두겠는데, 이 외의 상황도 있을 수 있을 것이다. 최악의 경우는 모두 중단하고 문제를 수정 후 mystart를 재시작해야 하는 경우도 있을 수 있다.

먼저 백그라운드 컴파일 창이 무한반복되는 경우나 pdf가 갱신이 되지 않는 경우. 이 때는 배경의 컴파일 창은 그대로 잠시 내려두고 에디터에서 독립적인 컴파일을 시도해본다. 에러 발생 지점에서 정지할 것이다. 해당 부분을 수정하고 저장하면 문제가 해결된다.

sumatrapdf가 죽은 경우, 일단 문제점을 발견한 후 수정하라. 그런 다음 **F7**을 누르면 된다.

5 종료하기

모든 창을 다 닫으면 된다. 백그라운드 명령창은 **Ctrl-C**를 누르라는 메시지가 보일 텐데, 그렇게 해도 되고 그냥 창 자체를 닫아도 상관없다.

6 맺는 말

작업 환경을 자신에게 적합하게 만드는 것은 매우 중요한 일이다. 여기에는 약간의 학습과 경험이 필요하다. 일단 익숙해진 후에 더없이 편리해지는 것이 \LaTeX 의 특징이라면 특징이라 할 수 있는 것이다. 이 글에서는 mystart를 이용한 작업 환경 구축의 예를 들었지만 이보다 더 편리한 자신만의 환경을 얼마든지 구축하여 활용할 수 있으리라 생각한다.

최고의 \TeX 작업 환경인 \LaTeX Live와 KC2008Plus와 더불어 행복한 \TeX 작업을 하시는 새해가 되기를 바란다. 이 훌륭한 환경을 대가없이 손쉽게 구축하도록 해준 한국텍학회, 한글텍사용자그룹, 표준한글텍환경 프로젝트팀에게 가벼운 감사 인사를 남겨보자.