SGML-Tools 사용법

LDP 표준문서 포맷인 sgml에 대한 설명과 sgml-tools의 사용법

고려대학교 리눅스 동아리 KULS, 심상현artsilly@kuls.korea.ac.kr v0.6 1999년 6월 6일

sgml로 표현된 문서는 사용이 간단하며 다양한 다른 형식의 자료를 쉽게 만들 수 있도록 기본적인 마크업을 해 준다. 본 글에서는 *LDP(Linux Document Project)* <http://metalab.unc.edu/LDP>의 표준 문서양식인 linuxdoc DTD sgml의 마크업 언어에 대해 알아보고 이 sgml 문서를 다른 형식의 문서로 바꾸어 주는 유틸리티인 sgml-tools의 사용법에 대해 알아보도록 한다. 이 문서의 원문은 고려대학교 리눅스 동아리 KULS의 두번째 세미나 자료이다.

목 차

제 1 절 저작권

제 2 절 sgml이란 무엇인가?

단순한 텍스트의 나열이 아닌 보기에도 좋고 깨끗한 문서를 만들기 위해 흔히들 워드 프로세서라 불리우는 프로그램을 사용한다. 요즘의 워드프로세서는 대부분 위지위그(눈에 보이는대로 출력물을 보면서 작성하는 방식) 방식을 채용하여 출력하기 전에 문서의 형태를 직접 보며 조작할 수 있어 문서작성의 편안함을 가져다준다. 그러나 동일한 포맷의 장,절 구분으로 나뉘어진 문서를 작성하는데 있어서는 단순한 작업을 반복 해야만 하는 불편함이 있다. 또한 여러 사람으로 나누어 문서를 작성할 경우에는 동일한 포맷을 유지하기가 그리 쉬운 일이 아니다. 마지막으로 해당 문서를 보기 위해서는 반드시 그 워드프로세서 프로그램이 필요하다는 점이 쉽게 문서를 보기에 여간 답답한 점이 아니다.

tex으로 대표되는 문서 포매터(document formatter) 혹은 문서 컴파일러(document compiler)는 위의 워드 프로세서와는 반대의 개념을 가지고 있다고 볼 수 있다. 이런 문서 컴파일러는 위지위그 방식이 아니기 때문에 실제로 출력되는 모습을 볼 수 없다. 태그(tag)라 불리우는 표시자를 이용하여 문서를 꾸미고 다듬는다. 동일한 형태가 반복되어 나올 때 반복되는 작업을 줄여주고 여러 사람이 문서를 나누어 작성할 경우에도 동일한 문서 형태를 유지해 줄 수 있다. 또한 작성은 단순한 텍스트 에디터에서 하는 것이기 때문에 때와 장소를 가리지 않고 작성이 가능하다는 것을 장점으로 들 수 있다. 아래는 tex문서의 예제이다.

```
\section{리눅스의 장점}
리눅스는 일반적인 다른 OS와 비교하여 다음과 같은 것들을 장점으로
내세울 수 있습니다.

\begin{enumerate}
\item 리눅스는 무료입니다. 리눅스 시스템을 꾸미고 사용하는데는 아무런
비용이 들지 않습니다. FTP를 통해서 리눅스의 모든 프로그램을 구할 수
있습니다.
\item 리눅스의 거의 대부분의 소스가 공개되어 있습니다. 10여년의 세월을
걸치면서 버그를 잡고, 안정성을 확보한 리눅스 시스템의 소스는 누구나
쉽게 얻을 수 있습니다.
```

이러한 문서 컴파일러로는 tex이나 groff등이 대표적이며 리눅스의 매뉴얼인 man 형식이나 emacs 등에서 읽을 수 있는 info 형식 역시 문서 컴파일이나 문서 포매팅의 형태를 가지고 있다고 할 수 있다. 그러나 좀 더 유용하게 사용하기 위해서는 이런 다양한 형식이 서로 변환이 되어야 한다. 각각의 문서 포매팅 형태를 바꾸어주는 유틸리티들이 있지만 여전히 여러가지의 태그를 익혀야한다는 불편한 점이 있다. sgml(Standard Generalized Markup

Language) 은 이런 여러가지 태그의 표준화 방안이다. 엄격히 말해 sgml은 문서 컴파일러라기 보다는 tex이나 groff등의 태그를 생성해주는 문서 전처리기라 할 수 있다. sgml은 이런 이유로 조판 언어(markup language)라고 불린다. 최근에는 sgml을 확장하여 더욱 유연한 문서를 작성할 수 있도록 해 주는 XML(Extensible Markup Language)이 나왔다. sgml 예제는 차츰 설명하도록 한다.

2.1 왜 sgml을 쓰는가?

위에 언급한 것처럼 sgml은 여러 문서 포매터의 태그를 생성해 주는 역할을 한다. 우리는 sgml 태그를 이용한 하나의 문서로부터 tex, html, gnu info, man, ascii등 다양한 포맷의 문서를 추출할 수 있다. 물론 여기에서 또 가지를 뻗어 나가면 ps나 pdf, dvi등 출력을 위한 여러 다양한 포맷을 얻을 수 있다. 단지 sgml의 태그를 익히는 것 만으로 충분하다.

2.2 그렇다면 sgml 태그는 쉬운가?

절대 그렇지 않다. 여러 태그의 표준이라면 얼마나 많고 다양한 태그를 지원하겠는가? 일반적인 태그는 문서를 장, 절로 구분하는 것에서 시작하여 문서를 꾸미거나 그림을 삽입하거나 인터넷 프로토콜을 연결해 주는 링크등 다양하다. sgml을 쓰기 위한 전용 프로그램도 어렵지 않게 만날 수 있을 정도이다.

2.3 그래도 sgml을 쓰라고 하는 이유는 무엇인가?

우리가 쓰려는 sgml은 쉽기 때문이다. 이 다양한 sgml 태그 중 특정 양식의 문서를 만들 때 쓰는 태그는 한정될 것이다. 이런 한정된 태그를 쓸 수 있도록 해 놓은 양식을 DTD라 하며 이것은 전체 sgml의 부분집합이 된다. 우리가 쓰는 DTD는 linuxdoc DTD이다. 여기에 있는 태그는 그 수가 적으며 사용하기에도 쉽다. linuxdoc DTD는 sgml-tools를 이용해 구현되며 현재 `<ftp://ftp.kr.FreeBsd.ORG/pub/users/cjh/sgml-tools-ko/>`에서 구할 수 있다. 특별히 우리가 쓰려는 프로그램은 한글이 가능해야 하므로 위의 링크에서 구하기 바란다. 최준호님께서 수고하여 한글 패치가 되어있다.

linuxdoc DTD는 리눅스 HOWTO 문서와 그밖의 문서의 표준 포맷이며 이 형식을 이용하여 여러 HOWTO 문서, mini HOWTO 문서등이 번역되었다. 우리나라에서도 번역본 HOWTO 문서의 표준포맷으로 sgml을 권장하고 있으며 본인 역시 *리눅스 한글문서 프로젝트* `<http://kldp.org>`에 갔다가 거기 관리자인 권순선님의 권유로 sgml에 빠져들게 되었다.

제 3 절 sgml의 기초

이제부터 sgml이라 하면 linuxdoc DTD에 해당하는 sgml만 말한다.

3.1 태그란?

태그란 html 문서에서 보는 것과 같은 < >로 묶인 사이에 문서의 형식을 지정하는 예약어이다. 사실 html은 html DTD이며 sgml의 부분집합이다. 다음과 같은 태그가 일반적으로 쓰인다.

```
<sect>
<sect1>
<author>
<p>
<item>
<itemize>
<enum>
```

```
<verb>
<code>
<quote>
<tt>
```

사실 문서를 좀 더 체계적으로 만들기 위해서는 좀 더 많은 태그를 알아야 하지만 기본적인 문서를 위해서는 위의 형식 정도면 충분하다.

3.2 시작하기/한번만 알아두기

여기에는 문서 머리에 넣어야 하는 몇몇 태그가 더 나온다. 그러나 이 태그는 특별히 외워둘 필요는 없다. 한번 작성해 두고 매번 복사해 쓰면 된다. 아래 예는 본 문서의 0.4버전 문서의 sgml 소스이다.

```
<!-- Korea Universiy Linux Study Team Seminar 2 -->
<!-- 쿨스 세미나 자료 #2 -->

<!doctype linuxdoc system>
<article>
<title>SGML-Tools 사용하기, 세미나 방법
<subtitle>쿨스 세미나 자료 #2
<author>심상현 <htmlurl url="mailto:artsilly@kuls.korea.ac.kr"
name="artsilly@kuls.korea.ac.kr">
<date>v0.4 1999년 4월 3일

<abstract>
이 문서는 고려대학교 리눅스 동아리 쿨스의 세미나자료이다. 누구나 쉽게
사용할 수 있으나 이 문서의 저작권은 쿨스에서 갖는다. 이 문서는 두번째
세미나자료로 sgml-tools을 이용하여 세미나 자료를 작성하는 방법과
앞으로의 세미나 방향에 대해 담고 있다.
</abstract>

<toc>
```

<!-- --> 로 나오는 줄은 주석이며 <!doctype linuxdoc system>로 DTD 타입을 지정해 주고 <article>로 문서의 시작을 표시한다. 이 태그는 문서의 마지막에 </article> 를 사용함으로 문서 전체를 끝마치게 된다. 이정도 보면 html을 작성해 본 사람이라면 대략 sgml에 대한 감이 올 것이다. <title>이나 <subtitle>은 제목과 부제목 정도가 된다. 이정도는 쉽게 알 수 있을 것이다.

<toc>는 Table of Contents 즉 차례를 자동으로 만들어 나타내라는 말이다. sgml에서는 자동으로 차례까지 만들어 준다.(얼마나 편한가!!) <author>, <date>등으로 문서의 작성자나 날짜를 표시하고 마지막으로 <abstract> 열고 </abstract> 닫음으로 문서의 간략한 개요를 적는다. 이런 헤더 소스는 한번만 작성해 놓으면 꾸준히 쓸 수 있는 기본적인 내용이다. 다들 적당히 카피해서 쓰기 바란다. 문서 작성자 <author>등은 바꾸도록 한다 –;

최준호님께서 패치하신 한글 sgml-tools에는 <trans>, <tdate> 라는 새로운 태그가 있다. 이것은 번역본 소스에 쓰이는 태그로 각각 번역자와 번역일을 나타낸다. <author>, <date> 처럼 써주면 된다. 아래는 번역문에 쓰이는 헤더 소스이다. 달라진 것은 저자 외에도 역자가 들어간다는 것이 다르다.

```
<!-- BogoMips mini-HOWTO의 sgml 소스
 번역/변환 심상현 artsilly@kuls.korea.ac.kr
-->
```

```
<!doctype linuxdoc system>
<article>
<title>BogoMips mini-HOWTO
<subtitle>보고밉스란 무엇인가?
<author>빔 완 도르스트(Wim van Dorst), <htmlurl url="mailto:baron@clifton.hobby.nl" name="baron@clifton.hobby.nl">
<date>v, 13 December 1997
<trans>심상현, <htmlurl url="mailto:artsilly@kuls.korea.ac.kr" name="artsilly@kuls.korea.ac.kr">
<tdate>v1.0s 1998년 12월 27일

<abstract>
이 문서는 유즈넷이나 e-mail을 통해 얻어진 보고밉스에 대한 여러 가지
정보를 담고 있다.
                                  (중략)
</abstract>

<!-- 차례 -->
<toc>

<!-- 본문의 시작 -->
```

앞으로 나올 태그는 대부분 위의 예에 나온 <author>등의 것과 달리 <abstract>, <abstract> 형식과 같이 열고 닫는 식으로 되어 있다. 몇가지만 구분하면 되기 때문에 그리 신경쓰지 않기 바란다. 하여간 여기까지 했다면 우리가 익혀야 할 태그의 반정도를 익혔다고 보면 될 것이다.

제 4 절 sgml을 이용하여 본격적으로 문서 작성하기

이 장은 본론이며 여기에 있는 내용들은 충분히 숙지하기 바란다.

4.1 장,절의 구분

역시 예를 보며 설명하도록 한다.

```
<sect>sgml의 기초
<p>
이제부터 sgml이라 하면 linuxdoc DTD에 해당하는 sgml만 말한다.

<sect1>태그란?
<p>
태그란 html 문서에서 보는 것과 같은 < >로 묶인 사이에 문서의 형식을
```

장 절의 구분은 닫기 태그가 필요없는 <sect> 태그로 이루어 진다. 쉽게 이해하겠지만 <sect>는 큰 제목, <sect1> 작은 제목, <sect2>는 더 작은 제목, <sect4>는 더더 작은 제목이다. 문서를 자세한 장, 절, 소제목으로 나눈는 것은 나쁘지 않은 일이나 일반적으로 <sect2> 정도면 충분하며 더 이상 문서가 깊이 있어지면 읽는데에 불편하며 브라우징 하기에도 불편하다.

<sect> 태그는 뒤에 제목을 쓰는 것으로 끝난다. 장,절에 맞추어 각각 다른 형식으로 변환 될 때 적당한 폰트와 들여쓰기 장,절 구분 숫자등을 자동으로 써 주게 된다. 이는 새로운 절이 추가 될 경우 숫자나 순서에 신경쓸 필요가 없다는 것이다. 단순히 추가만 하면 나머지는 자동으로 해 준다. 단 <sect> 밑에는 <p>를 써 줌으

로써 새로운 문단이 시작된다는 것을 알려줘야 한다. 초보시절에 가장 틀리기 쉬운 버그이다. 그밖의 문서 곳곳에서도 <p>나 한줄 띄워줌으로서 새로운 문단이 시작되는 것을 알릴 수 있다. 이것에 대해서는 추후 이야기하도록 한다. 하여간 이렇게 새로운 문단을 알리거나 아래 나올 항목 나열등의 새로운 태그가 나오지 않는 이상 모든 글은 위에 가서 붙게 된다. 중간에 한줄을 띄어놓거나 <p>를 사용하지 않는 이상 모두 한줄로 붙어 한 문단임을 나타내 준다는 말이다. 이 역시 sgml로 처음 만들때 쉽게 틀리기 쉬운 부분이다. 아래에 설명할 인용 태그를 쓰지 않고 인용한다고 그냥 무덤덤하게 copy-paste를 하면 아래처럼 끔찍한 결과를 나타낸다.

[linux]$ cat /proc/cpuinfo processor : 0 vendor_id : GenuineIntel cpu family : 5 model : 2 model name : Pentium 75 - 200

위의 예에는 어떤 인용 태그도 사용하지 않았다. 이것에 대한 올바른 인용 결과는 인용 태그를 보며 추후 예를 들도록 하겠다.

4.2 문자의 꾸밈

사실 우리가 쓰는 linuxdoc DTD의 경우 문자 꾸미는 것에 인색하다 할 수 있다. 다음과 같은 세가지의 태그를 이용한다.

<tt> </tt> 문자를 타이프라이터체로 나타낸다.

영문일 경우 대부분 courier체를 사용하며 한글은 설정에 따라 다르다.

`이 문자는 타이프라이터체 입니다. Hello, World.`

** 문자를 강조한다. 대부분 이탤릭체로 나타낸다.**

이 문자는 강조된 문자입니다. Hello, World.

<bf> </bf> Bold Face, 볼드체로 나타낸다.

이 문자는 볼드체입니다. Hello, World.

위와 같은 태그를 쓸 때 길지 않은 내용이라면 열어주고 닫아주는 것이 불편할 수도 있다. 이럴 때는 간이 태그를 이용할 수 있다. 즉

```
<tt/ Hello, World/
```

와 같이 / 를 사용하여 열고 닫을 수 있다. 이런 방식의 간이 태그는 긴 문장에라도 가능하지만 이럴경우 문서의 내용에 / 가 있다면 혼동을 일으킬 염려가 있으며 중복하여 쓰면 간이 태그 안에는 / 가 겹치기 때문에 중복하여 간이태그를 쓸 수 없다.

4.3 항목의 나열

항목을 나열하는 것은 세가지 종류가 있다.

<itemize> </itemize> 간단한 블랫(동그라미나 점등)을 이용하여 나열한다.

- 애프터 스텝
- FVWM
- 윈도 메이커

<enum> </enum> 순서대로 표시한다..

1. 애프터 스텝
2. FVWM

3. 윈도 메이커

이 두가지 <itemize>와 <enum>는 각 항목을 나타낼 때 닫는 태그 없는 <item> 이라는 태그를 사용한다. 즉 위의 예의 소스를 본다면

```
<enum>
<item>애프터 스텝
<item>FVWM
<item>윈도 메이커
</enum>
```

처럼 쓴다.

마지막 항목 나열 태그는 각 항목을 굵은 글자체로 나타내고 그 아래에 추가글을 들여 쓰는 형식으로 바로 이 절이 그 형식으로 되어있다.

<descrip> </descrip>굵은 글자체로 항목을 표시한다.

애프터 스텝
: 좋다. 아주 좋다.

FVWM
: PAKKUM이는 이걸 쓴다.

윈도 메이커
: 선근이는 이걸쓴다.

이 방법은 <item> 대신에 굵은 글자로 표현해야할 부분을 <tag>라는 태그로 대체한다. 위의 소스를 보면

```
<descrip>
<tag>애프터 스텝</tag> 좋다. 아주 좋다.<p>
<tag>FVWM</tag> PAKKUM이는 이걸 쓴다.<p>
<tag>윈도 메이커</tag> 선근이는 이걸쓴다.<p>
</descrip>
```

처럼 되어 있다.

4.4 인용하기

위에 말한 것처럼 적절한 인용 태그를 사용하지 않으면 sgml-tools에서는 한 문단으로 보고 모든 문장을 붙여 버린다. 인용 태그를 쓸 경우 인용문이라는 표시를 하기 위해 적절한 폰트 변환을 해 주며 여러 방법으로 인용문이라는 느낌이 들게 해 준다. 인용하는 방법에는 세가지 정도가 있다. <verb> 태그는 자동으로 문자를 타이프라이터체로 나타낸다. 즉 인용문의 태그는 자동으로 <tt> 가 붙는 것으로 이해해도 좋다.

<verb> </verb> 단순한 인용

```
이렇게
```

<code> </code> 위에 두줄이 나타난다.

```
요렇게
```

<quote> </quote> 이것은 <verb>와 같으나 <tt>형식이 아니며 들여쓰기가 된다.

이렇게

<code>를 너무 많이 쓰면 위 아래 두줄로 인해 정신이 사나울 정도니까 정말 필요하다고 생각하는 것에만 쓰길 바란다. 바라는 결과대로 <verb> 태그와 <code> 태그는 쓰여진 모습 그대로 대로 인용을 한다. 그러나 <quote> 태그는 일반 문단과 같이 또 앞줄에 주루륵 붙여 버리니까 주의 하기 바란다. 즉 <verb>를 쓸 경우

```
[linux ~]$ cat /proc/cpuinfo
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 5
model           : 2
model name      : Pentium 75 - 200
stepping        : 12
```

이렇게 나오는 것이(사실 이 예는 위에 나왔던 그! 예제이다.) <quote> 태그를 쓰면

> [linux]$ cat /proc/cpuinfo processor : 0 vendor_id : GenuineIntel cpu family : 5 model : 2 model name : Pentium 75 - 200 stepping : 12

이렇게 되어 버린다. 이 태그는 들여쓰기 정도로 이해하는 것이 좋을 것이며 사람 말의 인용등에 사용가능하다.

제 5 절 상호 참조 넣기

상호 참조는 하이퍼텍스트 기능을 갖춘 문서(대표적으로 html)로 변환 될 때 사용가능하다. 지금 문서를 html로 보고있다면 3(이 버튼)을 누르면 'sgml의 기초'로 다시 되돌아 갈 것이다.

이는 다음과 같은 태그를 이용하여 사용 가능하다.

대상 쪽

```
<sect>sgml의 기초<label id="test">
```

누르는 쪽

```
보고있다면 <ref id="test" name="이 버튼">을
```

그러나 최근 경향에 비추어 보면 문서 내의 상호참조는 별로 사용되지 않는 것 같다. 특별히 생각하지 않아도 된다.

제 6 절 인터넷 주소 넣기

6.1 기본형 ¡url¿

인터넷 주소는 html로 변환 할 때뿐만 아니라 여러 다른 문서로 변환이 될 때도 유용하다. <url>태그를 사용한다.

```
<url url="http://kulsnet.korea.ac.kr/~artsilly/"
name="아트실리의 홈페이지">
```

형식은 위와 같이 된다. URL을 따라갈 수 있는 문서형식(html)일때는 name= 태그 뒤에 있는 글자가 화면에 나오고 그 글자를 누르면 그 URL을 따라서 점프하게 된다. 그러나 URL을 따라갈 수 없는 일반 문서형식(tex, ascii등)은 뒤에 < >를 이용해 주소가 나오게 된다.

즉 위 예제는

아트실리의 홈페이지 <http://kulsnet.korea.ac.kr/~artsilly/>

와 같이 보일 것이며 만일 지금 보고있는 문서가 html이라면 '아트실리의 홈페이지'를 누르면 아트실리의 홈페이지로 갈 것이다. 그러나 일반 문서라면 그냥 '아트실리의 홈페이지 <http://kulsnet.korea.ac.kr/ artsilly/>'이라고 나와서 일반 문서 사용자를 위한 배려를 하게 된다. 지금 보고 있는 문서 형식에 따라 조금씩 다를 것이다. 만일 name= 부분을 생략한다면 자동으로 주소가 name으로 들어가게 된다. 아래 예을 참고 하기 바란다.

<http://kulsnet.korea.ac.kr/~artsilly/>

6.2 변형형 ¡htmlurl¿

이 태그의 변형형으로는 <htmlurl> 태그가 있다. 이것은 일반 문서형식이나 점프가능한(브라우저) 문서형식(html)에서나 모두 name= 에 넣은 내용만을 보여준다. 이 태그는 잘 사용해야 한다. 일반적인 본문에 이 태그를 쓴다면 일반 문서형식으로 변형시 URL이 표시되지 않아 낭패를 겪게 된다. 즉 아래의 예

```
<htmlurl url="http://kulsnet.korea.ac.kr/~artsilly/"
name="아트실리의 홈페이지">
```

는 모든 형식에서 아래와 같이 name= 뒤에 있는 내용만 나오게 될 것이며 특정 형식(tex)에서는 아예 나오지 않을 수도 있다.

그렇다면 이런 변형형은 어느 경우에 쓰는가? 그것은 누군가의 메일 주소를 나타낼 때 유용하다. 메일임을 나타내는 프로토콜 표시자(mailto:)는 꿀꿀하며 만일 이것을 url로 썼을 경우 하이퍼 텍스트 문서는 괜찮지만 일반 문서에서는 artsilly@kuls.korea.ac.kr <mailto:artsilly@kuls.korea.ac.kr> 과 같이 메일 주소가 두번 표시되거나(name= 을 썼을 경우) 아니면 아예 <mailto:artsilly@kuls.korea.ac.kr> 처럼 (name= 을 생략한 경우) 처럼 나타날 것이며 이것은 상당히 꿀꿀하다. 평상시에

> 난 메일주소가 mailto:artsilly@kuls.korea.ac.kr이다

라고 말하지는 않으며 항상 mailto는 빼고 이야기 한다. 이렇게 사용하기 위해 이때는 일반 문서든 html이든 공히 name= 만 나타나게 하기 위해 이 태그를 쓴다.

artsilly@kuls.korea.ac.kr

제 7 절 특수문자

특수문자나 별 특수문자 같지 않은 것들을 표시하는 예약어가 있다. 이것은 태그안에 태그를 인용하거나 하는 경우에 쓰인다.(이 세미나 자료 준비하면서 특수문자 코드를 너무 많이 사용했다. –; 미치겠다.) 가장 일반적이며 많이들 알고있는 에서 시작해서 여러가지가 있다. 이것에 대해서는 /usr/doc/sgml-tools-* 등에 있는 guide 문서를 참고하기 바란다. 평상시에는 특별히 사용할 만한 특수문자는 없을 것이다. 몇몇 사용 빈도가 높은 것은 다음과 같다.

- & - & - 앰퍼썬드 –; 라 불리우는 기호를 표시한다.
- < - < - 왼쪽 꺽쇠
- > - > - 오른쪽 꺽쇠
- " - " - 따옴표
-   -   - 아무 의미없는 빈칸

<p> 태그는 문단을 나누는 태그이다. <sect> 밑에는 꼭 써야 하며 그 밖의 위치에서는 한줄 비워두는 것으로 이 태그를 대신 할 수 있다. 즉 아래의 두 코드는 동일하게 나온다. 출력은 바로 이 문단과 바로 위의 문단이다.

```
있다. 이것에 대해서는 /usr/doc/sgml-tools-* 등에 있는 guide 문서를
참고하기 바란다.<p>
<p> 태그는 문단을 나누는 태그이다. <sect> 밑에는 꼭 써야
하며 그 밖의 위치에서는 한줄 비워두는 것으로 이 태그를 대신 할 수
```

```
있다. 이것에 대해서는 /usr/doc/sgml-tools-* 등에 있는 guide 문서를
참고하기 바란다.

<p> 태그는 문단을 나누는 태그이다. <sect> 밑에는 꼭 써야
하며 그 밖의 위치에서는 한줄 비워두는 것으로 이 태그를 대신 할 수
```

한줄 비워두는 것이나 <p>를 쓰지 않은 곳은 문단이 나뉘어 지지 않고 위로 붙게 된다. 자동으로 양 옆 폭을 맞추어 조절이 됨을 기억하기 바란다. 또 이야기하지만 이점에 주의하기 바란다.

제 8 절 문서 변환

우리는 2byte 문자를 사용하는 사람들이기 때문에 sgml-tools를 바로 가져다 쓸 수 없다. 한글이 사용가능하게 패치된 것은 출력물로 tex, ps, dvi, 일반텍스트, html등의 변환이 가능하다. info와 rtf 출력은 수요가 그리 없어 패치대상에서 제외한 것으로 보인다. 그럼 하나씩 살펴보도록 하자.

8.1 html로 변환

```
sgml2html test.sgml
```

와 같이 해 주는 것으로도 충분하다. 이렇게 할 경우 html은 각 장별로 다른 문서로 저장이 된다. 즉 7개의 장이 있을 경우 목차가 들어있는 test.html에서부터 test1.html test7.html이 생기게 된다. 이렇게 나뉘는 것이 싫다면 `-s 0` 옵션을 넣으면 된다. 즉 아래와 같이

```
sgml2html -s 0 test.sgml
```

라고 해 주면 test.html 문서 하나만 생기며 HTML 문서 역시 범용적으로 (때론 출력 목적에까지) 유용하다.

참고로 다음/이전/목차의 링크를 이미지를 이용하는 옵션이 있으나 매번 이미지를 복사해 줘야 하는 불편함이 있기 때문에 사용하지 않는 것으로 한다. 예쁜 이미지를 만들어 넣고 싶은 사람은 도움말을 참고하기 바란다.

8.2 txt로 변환

```
sgml2txt -s -f test.sgml
```

반드시 위의 두 옵션을 넣도록 한다. -f 옵션을 넣지 않으면 한글이 깨지는 곳이 생기고 -s 옵션을 빼 놓으면 이상한 빈줄이 생긴다. 이것은 원 프로그램의 문제이며 추후 패치될 것이다.

8.3 dvi로 변환

```
sgml2latex test.sgml
```

과 같이하면 폰트를 포함하지 않는 ps인 dvi 포맷의 문서가 생긴다. 단 시스템에 tex과 Hlatex이 올바르게 깔려 있어야 한다. latex은 이 sgml을 위해서뿐 아니라 익혀두면 무척 유용하므로 깔아두는 것도 좋을 것 같다. 단 하드를 약 300M정도 잡아먹는다. –;

dvi로 변환된 문서는 xdvi로 볼 수 있으며 dvips를 통해 ps 문서로 변환가능하다. dvips의 사용법은 아래와 같다.

```
dvips -f test.dvi > test.ps
```

택 패키지의 기본 설정에 따라 다르겠지만 -f 옵션을 사용하지 않으면 필자의 경우 바로 프린터로 출력이 된다. -f 옵션은 화면으로 출력을 보내며 이것을 리다이렉션 시켜 저장하면 된다. 아예 처음부터 ps로 만들고 싶다면

```
sgml2latex -o ps test.sgml
```

와 같이 해도 된다. 참고로 택은 시스템 프린터 설정에 따라 각기 해상도가 다른 폰트를 사용하며 폰트 소스를 컴파일 하여 사용한다. 폰트 컴파일 하는데 몇분 정도 걸리니 꾸준히 기다리기 바란다.

제 9 절 그밖의 잡담들

9.1 문서 통일화를 위한 조언

9.1.1 인용에 있어서

앞으로의 문서에 있어서 특별히 강조하고 싶지 않은 코드라면 일반적으로 `<verb>`를 쓰는 것이 좋다. 위에 말한 것처럼 눈아프다.

9.1.2 인터넷 주소넣기

`<url>`과 `<hmlurl>`을 반드시 구분하여 사용하도록 한다. 잘못 사용한 sgml 소스는 하이퍼 텍스트 형식이 아닌 문서에서 아무런 정보를 제공해 주지 못한다. 예전에 번역했던 문서중에 아무런 URL이 표시가 안되어서 무척 궁금해 했던 글이 있었는데 그 글의 소스에서 원 저자가 `<hmlurl>`을 오용하여 일반 텍스트 형식에서 아무런 URL을 찾을 수 없던 경험이 있다.

9.1.3 문장 내의 강조

문서 내부에 "를 사용함으로 어떤 부분을 강조하기 보다는 문서 꾸밈 태그인 `<tt>`를 사용하는 것이 더욱 보기 좋은 문서를 보여준다.

9.2 sgml을 어디에 쓸 것인가?

특별히 잡다한 tex 소스에 익숙하지 않은 사용자라면 아주 범용적으로 sgml을 사용할 수 있다. 레포트나 일반 문서용으로도 좋다. 그러나 좀 시간이 있다면 바로 지금 *리눅스 한글문서 프로젝트* `<http://kldp.org>`로 달려가 리눅스 문서 번역에 열심인 다른 사람들을 보고 함께 문서 번역에 참여하는 것도 나쁘지 않은 생각이다.