

직접 수정할 수도 있습니다. 액트온을 위키로 만들어졌으니까요! 그럼, 익명의
네트를 맘껏 항해하시길 바래요!

@ ActOn 1001

레니의 테크토크

- 300 Right or Not, 포털사이트 광고 차단
- 304 나는 네가 어디에서 왔는지 알고 있다 - 리퍼러
- 307 사악한 KT는 어떻게 공유기를 검출하는가
- 322 여론 조작에 대한 사소한 오해 - 멀티서버 카운터
- 327 공유의 미덕 - 공유기

Right or Not, 포털사이트 광고 차단

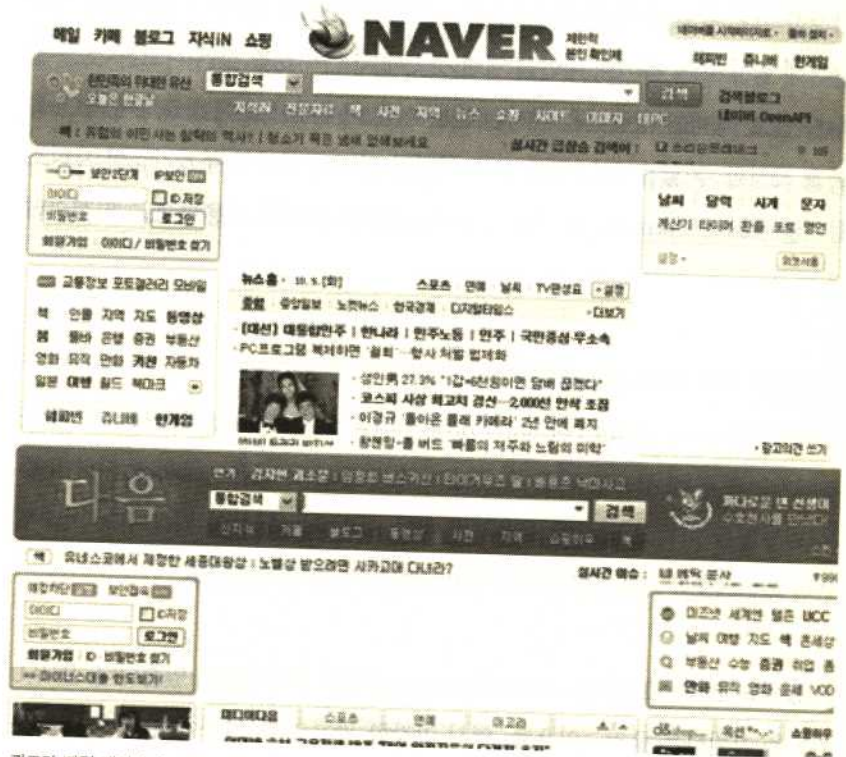
레니 | 진보네트워크센터 자원활동가 <http://blog.jinbo.net/renegeade>

요즘 “네이버 플래쉬/동영상 광고 차단”이라는 재미있는 글이 돌아다니고 있습니다. 이 글의 원본이 어디 있는지는 모르겠지만, 게시판, 댓글, 블로그 등을 통해 빠른 속도로 복사되어 확산되고 있죠.

포털 사이트의 광고를 차단하는 방법은 예전부터 많이 있어 왔기 때문에, 이 글에서 제시하는 방법 자체가 신선한 내용은 아닙니다. 광고-특히 배너 광고에 대한 사용자들의 반감은 상당한 수준에 도달해 있기 때문에, 광고를 보지 않고도 문제없이 포털 사이트를 사용할 수 있는 방법이 많이 개발되었습니다.

일단 위의 글에서 소개한 방법은 IE의 보안 설정을 이용한 방법입니다. IE 6.0 이상 버전에서는 “제한된 사이트”라는 목록에 특정 사이트를 입력하면 해당 사이트에 요청을 아예 보내지 않는 기능이 있습니다. 대부분의 광고 시스템들은 광고를 노출시키는 서버를 따로 운영하기 때문에 광고 서버는 별도의 도메인을 갖고 있습니다. “제한된 사이트” 목록에 포털 사이트의 광고 서버 도메인을 추가 시켜 놓으면, 광고 서버에 요청을 보내지 않기 때문에 광고 영역은 빠진

채로 사이트의 내용이 나오는 것이죠.



광고가 빠진 네이버와 다음의 메인 화면

그럼 이렇게만 하면 모든 광고를 안 볼 수 있느냐 하면, 반드시 그런 것도 아닙니다. 이 “제한된 사이트” 기능은 좀 단순한 측면이 있어서 iframe 등을 통해 요청하는 경우에 있어선 확실히 동작하긴 하지만, 자바스크립트 등의 동작은 막고 있지 않습니다. 따라서 스크립트와 DHTML로 조금 꼬아서 만든 광고 태그에 대해선 정상적으로 광고가 노출되게 되어 있죠.

불여우는 IE처럼 기본 보안 설정에서 특정 사이트를 배제하는 기능은 없습니다. 다만, 불여우의 부가기능(Extension)인 Adblock(<http://adblockplus.org/en/>)을 통해 보다 강력한 광고 필터 기능을 사용할 수 있습니다. Adblock은 자바스

크립트나 DHTML 이런 거 봐주지 않고 모조리 차단해 버리는 강력한 기능을 자랑합니다. 광고 서버 도메인이나 광고 이미지 서버 도메인을 Adblock의 차단 목록에 넣고 포털 사이트 메인을 열면 곳곳에 구멍이 뿜뿜 뚫려 나오는 것을 볼 수가 있죠. 게다가 정규식을 아시는 분은 이를 이용해서 보다 정교하게 차단 목록을 작성할 수 있습니다. 물론 조금 어렵긴 하지만 막강한 부가기능인 그리스몽키(greasemonkey)(<http://www.greasespot.net/>)를 사용해서도 되겠습니다.

사파리 역시 PithHelmet

(<http://www.culater.net/software/PithHelmet/PithHelmet.php>)이 라는 부가기능을 통해 광고를 막을 수 있습니다. PithHelmet은 Adblock 이상의 많은 기능들을 가지고 있지만 사용하기 약간 어렵다는 단점이 있습니다. 어떻게 보면 Adblock과 그리스몽키와의 중간 난이도라 할 수 있겠군요.

IE에도 IE7Pro(<http://www.ie7pro.com/>)라는 부가기능이 있습니다. IE 7.0 이상을 지원하는 플러그인으로, 역시 막강한 광고 필터 기능이 있으며, 사용하기 그다지 어렵지 않죠.

이렇게 관련 자료를 찾아 보면 광고를 싫어하는 유저들이 무척 많다는 것이 새삼 느껴집니다. 앞에서 소개한 글 역시 이러한 맥락에 이어져 있긴 하지만, 약간 다른 점이 있다면 개인적인 귀차니즘/광고혐오증을 넘어선 정치적 동기가 배경이 되었다는 것입니다. 사건의 발단은 생뚱맞게도 2007년 대선이었습니다. 선거법을 명분으로 하여 인터넷 세상에 제한적 실명제 등의 여러 제한들이 가해지는 상황에서, 네이버는 최근 정치 기사 댓글을 정치토론장으로 일원화 하고, 뉴스 댓글을 선택적으로 노출시키는 등의 개편을 단행했습니다. 이에 네티즌들의 반발이 이어졌는데, 이것이 네이버 광고 차단 운동으로까지 발전하게 된

것이죠.

광고 차단과 관련된 논란이 어제 오늘 일이 아닌 것만큼, 광고를 막는 것이 정당인가 부당한가의 논쟁을 단순히 소비자 주권 문제-광고가 싫으면 포털 서비스 쓰지 마라-로 협소화시킬 순 없겠습니다. 아무래도 이 문제는 조금 더 본질적인 논쟁으로 발전할 수 있을 것 같은 생각입니다.

나는 내가 어디에서 왔는지 알고 있다 - 리퍼러

레니 | 진보네트워크센터 자원활동가 <http://blog.jinbo.net/renegade>

블로그같은 1인 미디어를 갖고 계신 분들은 자신의 블로그에 들어갔을 때 어디를 제일 먼저 볼까요? 물론 사람마다 다르겠지만, 저 같은 경우 새로 쓴 포스트에 달린 댓글을 먼저 살펴봅니다. 그 다음에는 방명록에 새 글이 올라왔나 살펴보고요. 그런 다음 확인하는 것이 “접속 통계”입니다.

대부분의 1인 미디어 서비스들은 사용자의 페이지에 접속 통계를 제공합니다. 얼마나 많은 사람들이 자신의 페이지를 찾았는지, 그 사람들은 어떤 경로를 통해 내 페이지에 들어오게 되었는지 등의 정보를 요약해서 보여주는 것이 접속 통계 서비스입니다. 제 블로그는 하루에 100명도 채 안 되는 방문자가 오고 있으며, 대부분 구글, 네이버, 다음 등의 검색 엔진을 통해 찾아오고 있다는 사실을 알 수 있는 것도 접속 통계 서비스 덕분이죠.

그럼 접속 통계는 어떤 데이터를 근거로 산출되는 것일까요? 이 중 일일 접속자 수 같은 통계는 구하기 쉽습니다. 가장 단순하게 생각하면 사용자가 페이지에 접속하는 순간 로그를 남기고 나중에 그 로그의 개수를 합치면 접속자 수가 나오게 되지요. 물론 블로그 주인이 접속했을 때에는 통계를 잡지 않는다던지, UV(Unique Visitor, 순방문자)를 측정한다던지 하는 것은 좀 더 고려해 봐야

할 문제이긴 하지만, 기본적으로 접속자 수 자체를 구하는 것은 특별한 정보가 없어도 접속하는 순간의 기록만 있으면 계산 가능합니다.

문제는 방문 경로를 구하는 것입니다. 블로그 사이트에서는 어떻게 방문자가 바로 이전에 어떤 사이트에 있었는지를 알 수 있을까요? 바로 리퍼러(referer)라는 정보가 있기 때문입니다. 방문자가 A 사이트에서 링크를 클릭하여 B 사이트로 이동하는 경우, 웹브라우저는 B 사이트에 페이지에 대한 요청을 보내면서 HTTP 헤더에 A 사이트의 URL을 같이 실어 보냅니다. 이 URL을 리퍼러라고 부르며, 특정 사이트에서 사용자의 접속 경로를 추적하기 위해 심는 악성 코드 등과는 다른, HTTP 규약에 따른 정보입니다. B 사이트는 페이지 요청을 받음과 동시에 방문자가 있었던 이전 사이트를 좋건 싫건 자연스럽게 알 수밖에 없는데, 이 리퍼러 정보를 버리지 않고 적극적으로 활용하는 예가 바로 접속 통계 서비스라고 할 수 있겠습니다.

그럼 방문자는 방문하는 사이트마다 이전의 방문지를 남기게 되는 것일까요? 정확히 말하면 방문한 모든 사이트에 리퍼러가 남지는 않습니다. 리퍼러가 전송되기 위해서는 몇 가지 조건이 충족되어야 하는데, 이 중 가장 기본적인 조건은 링크를 클릭해서 이동한 경우에만 리퍼러가 남는다는 것입니다. 즉, 브라우저의 주소창에 직접 URL을 치고 들어간 사이트에는 이전에 특정 사이트에서 넘어갔다고 하더라도 리퍼러가 남지 않습니다. 같은 이유로 북마크(책갈피)를 클릭하고 들어간 사이트나 브라우저의 뒤로 가기 버튼이나 앞으로 가기 버튼을 누르고 이동한 경우에도 리퍼러가 제대로 남지를 않죠. 접속 통계에 리퍼러를 알 수 없다는 통계가 잡히는 경우가 많은데, 바로 리퍼러가 남지 않는 경우가 많기 때문입니다.

리퍼러가 남는 조건은 브라우저마다 차이를 보이기도 합니다. 예를 들면, 특

정 버튼이나 링크를 클릭해서 팝업 창을 띄우는 경우, 인터넷 익스플로러 6의 경우 리퍼러가 남지 않는 반면 파이어폭스는 리퍼러를 남겨줍니다. 이는 리퍼러 정보를 서버에 보낼지 말지는 요청을 보내는 브라우저의 몫이기 때문에, 브라우저마다 경우에 따라 리퍼러를 남기는 방식의 차이를 보이는 것이죠.

리퍼러를 사용하는 또 다른 예는 어뷰징abusing 체크 로직입니다. 어떤 게시판에 글을 쓰는 경우를 생각해 봅시다. 게시판에 글을 남기기 위해서 일단 편집기에 글을 쓴 다음, 이 글을 요청 파라미터에 넣어 서버에 저장하는 URL로 보내게 됩니다. 그런데 만약 글을 최종적으로 저장하는 URL을 어떤 방법으로 알게 되면, 편집기에 글을 쓰지 않고도 요청 파라미터에만 글 내용을 넣은 다음 그 URL로 요청을 보낼 수도 있습니다. 이 방법을 사용하면 편집기에서 글을 작성하는 과정 없이 단시간 내에 많은 양의 글을 쓸 수가 있습니다. 게시판이나 댓글 도배가 이런 방식으로 많이 이뤄지는데, 이는 시스템을 악용하는 행위라 하여 어뷰징이라 합니다. 이런 경우 이전 페이지가 편집기의 URL인 경우에만 글을 저장하도록 체크하는 로직을 넣는다면, 글을 저장하는 URL로 바로 요청을 보내는 경우를 차단할 수 있습니다. 이를 리퍼러 체크라고 하는데, 가장 기초적인 어뷰징 방지법으로 많이 사용되고 있죠.

리퍼러는 사용자에게 도움이 되는 정보로 집계될 수도 있고, 어뷰징을 방지함으로써 시스템의 안정성을 향상시키는 데 사용되기도 합니다. 하지만 이를 악용하는 사례도 많이 볼 수 있는데요, 대표적인 것이 리퍼러 스팸입니다. 블로거들이 자신의 블로그를 찾아온 경로에 관심이 많단 사실을 알게된 스팸머들은 블로그를 방문하면서 홍보할 사이트를 리퍼러로 남겨 놓습니다. 블로거는 호기심에 그 사이트에 들어가 보면 스팸머들은 소기의 목적을 달성하게 되는 셈이죠. 나 날이 진보하는 스팸 방식을 보면, 참 광고쟁이들의 상상력은 정말 끝이 없는 것 같단 생각이 드네요~

레니의 테크토크

사악한 KT는 어떻게 공유기를 검출하는가

레니 | 진보네트워크센터 자원활동가 <http://blog.jinbo.net/renegade>

서론

2005년 KT는 자사의 인터넷 라인에 공유기를 꼽아 여러 대의 IP를 따서 사용하는 유저를 식별할 수 있는 시스템, 즉 공유기 검출 시스템 개발에 성공했다고 발표했습니다.

하지만 이것이 기술적으로 완벽하지 않다는 반론과 함께, 기존 사용자들의 거센 반발로 인해 사용 가능 PC를 2대로 제한한다는 정책으로 선회한 뒤 이 문제는 그대로 묻히나 했는데... 작년부턴 KT와 하나로 사용자들을 중심으로 인터넷 연결 중 이상한 현상에 대한 보고가 들어오기 시작합니다.

참고1:

http://www.phpschool.com/gnuboard4/bbs/board.php?bo_table=talkbox&wr_id=943726

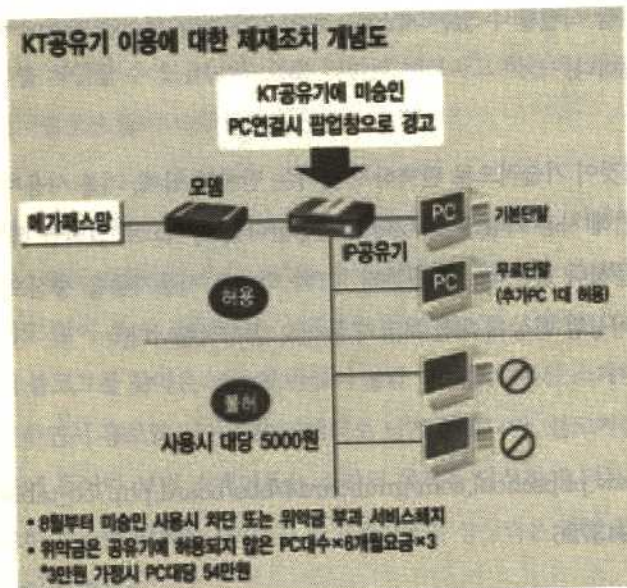
참고2 :

http://www.phpschool.com/gnuboard4/bbs/board.php?bo_table=talkbox&wr_id=884134

이를 정리하면, KT등의 ISP(인터넷 서비스 공급자, Internet Service Provider) 라인을 사용하는 유저들이 웹브라우저로 특정 사이트에 접속할 때, KT에서는 임의로 공유기 검출을 위해 특정 서버로 먼저 요청을 보낸 다음, 사용자가 요청한 페이지로 다시 이동시킨다는 것입니다.

참고3 : <http://wariua.egloos.com/2255292>

명목상 이런 과정은 “접속 환경 개선을 위한 속도 테스트”나 “불량 사이트 차단”을 위한 것이라는 설명을 많이 하는데, 실제로 앞에서 얘기한 KT 서버에서 하는 일은 사용자가 공유기를 사용하는지를 체크하는 것입니다. 그럼 이런 과정이 어떻게 일어나는지 한 번 추적해 보겠습니다.



KT 공유기 이용제한 조치 개념도

로직 추적

* 브라우저에 <http://www.jinbo.net> 으로 요청을 보냅니다.

보통의 경우라면 바로 www.jinbo.net으로 요청이 전달되어 진보넷 페이지가 브라우저에 뜨게 됩니다. 하지만 공유기 검출 로직이 실행되는 중이라면 요청한 주소는 다음과 같은 HTML로 치환되어 보내집니다.

```

<HTML>
<HEAD></HEAD>
<FRAMESET border=0 frameSpacing=0 rows=0,*
frameBorder=0>
<FRAME name=dolla src
="http://221.147.67.202/headseed.asp">
<FRAME src="http://221.147.67.202/xs_btn/uxcsdo.asp?
n=2&u=8825&i=0&y=0&m=0&k=0&t=N&b=78&x=www.jinbo.net&s=600&d=2008-01-01">
</FRAMESET>
</HTML>

```

화면이 두 개의 프레임으로 나뉘고, 둘 다 특정한 KT 서버에 요청이 갑니다. 이 중 첫 번째 요청(<http://221.147.67.202/headseed.asp>)은 서버에서 정확히 어떤 일을 하는지 알 수 없고 단지 빈 HTML이 돌아올 뿐입니다. 여기서 주목해야 할 것은 두 번째 요청입니다.

* http://221.147.67.202/xs_btn/uxcsdo.asp

http://221.147.67.202/xs_btn/uxcsdo.asp?n=2&u=8825&i=0&y=0&m=0&k=0&t=N&b=78&x=www.jinbo.net&s=600&d=2008-01-01

```

<html>
<frameset rows='0,*' border='0'>
<frame src='/xs_btn/uxcn.asp?
n=2&u=8825&i=0&y=0&m=0&k=0&t=N&b=78&s=600&x=
www.jinbo.net&d=2008-01-01'>
<frame src='http://www.jinbo.net/?' name='rosea'>
</frameset>
<body oncontextmenu="return false" onselectstart="return
false" ondragstart="return false">
</body>
</html>

```

응답의 결과로 역시 두 개의 프레임을 나눠서 줍니다. 첫 번째 프레임은 hidden 처리되어 특정 코드를 수행하고, 두 번째 프레임은 요청한 주소의 페이지를 보여줍니다. 결과적으로 화면에는 진보넷 페이지가 보이게 되니까 사용자는 크게 문제점을 인식하지 못할 가능성이 높습니다. 하지만 실제로는 사용자가 눈치채지 못하게 첫 번째 프레임의 동작이 실행되는 셈입니다. 그럼 첫 번째 프레임의 요청을 따라가 봅시다.

* http://221.147.67.202/xs_btn/uxcn.asp?

```

n=2&u=8825&i=0&y=0&m=0&k=0&t=N&b=78&s=600&
x=www.jinbo.net&d=2008-01-01

```

```

<input type='hidden' id='PTe' value='http://www.jinbo.net'
target='_top'>

```

```

<script language='javascript'>
var tcheck = document.cookie;
window.onload = gonogo;

function gonogo() {
if (tcheck == ) {
if (navigator.appName == 'Microsoft Internet Explorer') {
document.all.PTAe.click();
}
if (navigator.appName == 'Netscape') {

self.location.href=document.getElementById('PTe').value;
}
} else {
today = new Date();
var YY, MM, DD;
YY = today.getYear();
MM = today.getMonth()+1;
DD = today.getDate();

self.location.href='uxcn01.asp?mode=SSL3H&n=2&ha='+YY
+'&ba='+MM+'&ya='+DD+'&x=www.jinbo.net&i=0&y=0&m=
0&k=0&t=N&b=78&u=8825&d=2008-01-01';
}
}
</script>

```

```

<html>
<head>
  <title></title>
  <meta http-equiv="Content-Type" content="text/html;
charset=euc-kr">
</head>
<body oncontextmenu="return false" onselectstart="return
false" ondragstart="return false">
</body>
</html>

```

특정한 스크립트가 응답으로 날아옵니다. 페이지가 로드되면서 스크립트를 실행시켜 쿠키가 있는 경우 날짜와 기타 정보들을 파라미터로 싣고 특정 사이트로 다시 보내버립니다. 다시 페이지로 따라가 보겠습니다.

```

* http://221.147.67.202/xs_btn/uxcn01.asp?
mode=SSL3H&n=2&ha=2008&ba=4&ya=22&x=www.jinbo.net&i=0&
y=0&m=0&k=0&t=N&b=78&u=8825&d=2008-01-01

```

```

<SCRIPT LANGUAGE="JavaScript">
</SCRIPT>
<input type='hidden' id='PT' value='http://www.jinbo.net'
target='_top'>

<script language='javascript'>
  if (navigator.appName == 'Microsoft Internet Explorer') {

```

```

window.onload = function() {
  document.all.PTA.click();
}
}
if (navigator.appName == 'Netscape') {
  window.onload = function() {
    self.location.href=document.getElementById('PT').value;
  }
}
</script>

```

역시 다른 페이지로 보내버립니다. 또 따라가 봅시다.

```

* http://221.147.67.202/xs_btn/april.asp?n=2&u=8825&d=2008-01-
01&i=0&y=0&m=0&k=0&b=78&x=www.jinbo.net

```

```

<HTML><HEAD><TITLE>Untitled Document</TITLE>
<META http-equiv=Content-Type content="text/html;
charset=euc-kr">
<SCRIPT language=javascript>
<!--
//오늘은 더이상 팝업 윈도우를 띄우지 않습니다
function GetCookie( name )
{
  var nameOfCookie = name + "=";
  var x = 0;

```



```

while ( x <= document.cookie.length )
{
    var y = (x+nameOfCookie.length);
    if ( document.cookie.substring( x, y ) == nameOfCookie ) {
        if ( (endOfCookie=document.cookie.indexOf( ";", y ) == -
1)
            endOfCookie = document.cookie.length;
            return unescape( document.cookie.substring( y,
endOfCookie ) );
        }
        x = document.cookie.indexOf( " ", x ) + 1;
        if ( x == 0 )
            break;
    }
    return "";
}

```

```

//*****
*****

```

```
popup();
```

```

function popup()
{
    var g1 = "78";
    var n_type = "2";

```

```

var u_idx = "8825";
var cnt_g = "0";
var cnt_i = "0";
var msgUrl = "";
var width = "";
var height = "";
var msgUrl2 = "";
var width2 = "";
var height2 = "";
var total = "0";
var xt = "www.google.com";
var dsa = "2008-01-01";
var mainsvIP = ""; //임시 (공유기 서비스가입)

var tmpwin = "";
var tmpwin2 = "";
var param = "";
param = "g1="+g1+"&u_idx="+u_idx+"&cnt_g="+cnt_g+"&
cnt_i="+cnt_i+"&total="+total+"&xt="+xt+"&mainsvIP="+mai
nsvIP+"&dsa="+dsa;

if (n_type != 2 && msgUrl != "") // 유도
{
    url = "evsol.asp?n_type="+n_type+"&"+param+"&
msgUrl="+msgUrl;
    tmpwin=window.open

```

```

(url, "pop1", "width="+width+", height="+height+", top=0, left=0"
);
}

if (msgUrl2 != "") // 일반
{
//오늘은 더이상 팝업 윈도우를 띄우지 않습니다 if 문
var erer = GetCookie("Notice");
if ( erer != "done" )
{
url2 = "evsol.asp?n_type=1&"+param+"&
msgUrl="+msgUrl2;
tmpwin2=window.open
(url2, "pop2", "width="+width2+", height="+height2+", top=0, lef
t=370);
//tmpwin2.focus();
}
}

if (n_type == 2)
{
var strHTML = "<APPLET CODE='fnqlxmfhvl.class'
WIDTH='1' HEIGHT='1' name='Internet_Loading'>";
strHTML += "<PARAM NAME='URL'
VALUE='dotori_app.asp?b=78&n=2&u=8825&i=0&y=0&t=0
&x=www.daum.net&re_ip='>";

```

```

strHTML += "<PARAM NAME='TARGET' VALUE='_self'>";
strHTML += "</APPLET>";
document.write (strHTML);
}
} // popup() end

//-->
</SCRIPT>
</HEAD>
<BODY oncontextmenu="return false" onselectstart="return
false" ondragstart="return false"></BODY></HTML>

```

드디어 정말 하고 싶었던 일을 하는 페이지가 나왔습니다. 이 복잡한 과정의 목적은 결국 공유기 사용자에게 경고 팝업을 띄우고, fnqlxmfhvl.class란 java 애플릿을 실행시키는 것입니다. 일단 fnqlxmfhvl.class가 무엇을 하는지 다운 받아 리버스 컴파일해 보았습니다.

* fnqlxmfhvl.java

```

import java.applet.Applet;
import java.applet.AppletContext;
import java.net.*;

public class fnqlxmfhvl extends Applet
{
public fnqlxmfhvl()

```

```

{
    flema = "";
    mkbro = "unknown";
    fx0073 = "";
}

public void init()
{
    mkbro = rusianxx();
    if(getParameter("URL") != null)
    {
        fx0073 = getParameter("URL");
        fx0073 = fx0073 + mkbro;
        if(getParameter("TARGET") != null)
            flema = getParameter("TARGET");
    }
}

public void start()
{
    try
    {
        URL url = new URL(getDocumentBase(), fx0073);
        getAppletContext().showDocument(url, flema);
    }
    catch(Exception exception) {}
}

```

```

}

private String rusianxx()
{
    String s = "";
    String s1 = getDocumentBase().getHost();
    int i = 80;
    if(getDocumentBase().getPort() != -1)
        i = getDocumentBase().getPort();
    try
    {
        s = (new Socket(s1,
            i).getLocalAddress().getHostAddress());
    }
    catch(Exception exception) {}
    return s;
}

String flema;
String mkbro;
String fx0073;
}

```

이 프로그램이 하는 일은 매우 간단합니다. 사용자의 내부 IP를 알아낸 다음, 이 IP를 dotori_app.asp로 전송해 줍니다. dotori_app.asp로 요청을 보냈을 때 특별한 응답은 날아오지 않았지만, 짐작으로는 아마 dotori_app.asp에서 사용

자의 내부 IP를 받아 공유기를 사용한 IP인지, ISP에서 정상적으로 발급한 IP 인지를 판단할 것입니다.

로직 설명

일반적으로 KT나 하나로텔레콤 등의 ISP는 사용자에게 유동 IP를 발급합니다. 유동 IP는 모뎀이 ISP에 접속할 때마다 일정 범위 내에 있는 IP를 발급해 주는 IP입니다. 사용자가 ISP에서 제공하는 라인을 컴퓨터에 바로 꽂아 쓰는 경우에는 사용자 PC의 IP는 외부에서나 내부에서나 ISP가 발급한 유동 IP로 동일하게 됩니다.

하지만 공유기를 사용하게 되면 약간 다른 상황이 발생합니다. 공유기는 하나의 외부 IP를 여러 개의 내부 IP로 분할해 사용 가능하게 해 줍니다. 이 경우 유동 IP는 공유기가 갖게 되고 사용자의 PC는 공유기가 발급한 내부 IP를 사용하게 됩니다. 따라서 사용자 PC의 IP는 10.0.0.2나 192.168.0.2 같은 형태의 내부 IP가 됩니다. 하지만 외부에서 볼 때 사용자 PC의 IP는 공유기가 가지고 있는 IP, 즉 ISP가 발급한 외부 IP로 보이게 되죠. 따라서 이 경우는 외부 IP와 내부 IP가 달라지는 현상이 발생합니다.

어쨌든 ISP는 사용자들이 공유기를 사용하여 여러 대의 PC가 한 개의 라인에 몰려있는지 판단할 수 있는 작업을 사용자 몰래 하고 있는 셈입니다. 현실적으로 집에서 2~3개의 PC를 공유기로 사용하는 경우는 크게 문제를 삼고 있지 않지만, 사무실 등에서 10개 이상의 PC가 한 개의 공유기를 통해 사용하는 경우는 적발하여 추가적으로 라인을 구매하도록 유도하고 있다고 합니다. 이런 경우를 알 수 있는 것도 이러한 공유기 검출 시스템이 있기 때문이죠. 아마 사용자의

약관 어딘가에 사용자 동의 없이 하는 이런 작업을 정당화 하는 문구도 넣어놨으리라 생각합니다.

결론

현실적으로 ISP에서 제공하는 망을 사용하는 이상, 개인 PC에 침입하여 벌어지는 이런 일들을 막기가 매우 어렵습니다. 어떻게 보면 인터넷 세상의 진정한 지배자는 ISP가 아닐까요...

* 글이 너무 길어져서 공유기가 어떻게 IP를 분배하는지는 다른 글에서 다루도록 하겠습니다.

여론 조작에 대한 사소한 오해 - 멀티서버 카운터

레니 | 진보네트워크센터 자원활동가 <http://blog.jinbo.net/renegeade>

이명박 정부가 들어서면서 엉뚱하게도 다음 아고라가 유명세를 타게 되었습니다. 물론 그 전에도 사회적 이슈에 대한 토론을 주목적으로 하는 게시판 서비스로서 한겨레 토론방과 함께 나름 알려지긴 했지만, 기존 언론이 차지하고 있던 사회적 논의의 장에 있어 인터넷이 그렇게 큰 역할을 차지하진 못했기 때문에 아고라는 아는 사람만 아는 곳일 뿐이었습니다. 하지만 이번 쇠고기 파동을 통해 아고라는 거의 실시간으로 이슈를 생산하고 재생산하는 인터넷의 매체적인 특성을 잘 보여주며 단순한 토론장 이상의 기능을 하고 있습니다.

아고라는 이명박 탄핵 서명이 폭발적으로 일어나며 사회적인 관심을 받기 시작했습니다. 순식간에 서명 회수가 100만을 돌파하면서 반이명박 정서를 보여주는 대표적인 사건이 되었는데, 이 서명수는 현재 130만을 넘어서 있는 상태이고 지금도 여전히 서명이 올라오고 있습니다. 물론 다음의 회원 정책상 한 사람이 복수의 아이디를 만들 수 있기 때문에 130만명 이상이라고 단언할 수는 없

겠지만, 대선-총선에 걸친 승리로 의기양양하던 정부와 여당의 독선적인 정책에 대해 이루어진 강력한 조직적 반대라는 점에 있어 무시할 수 없는 숫자가 아닐까 합니다.

그런데 이 서명이 한창 불꽃 튀며 올라가고 있을 당시, 서명수 조작에 대한 불신이 널리 퍼졌습니다. 서명수가 너무 가파르게 올라가다 보니까 정부에서 다음에 압력을 넣어, 또는 다음이 알아서 서명수를 줄이고 있다는 것입니다. 서명을 했을 때 서명수보다 서명을 하고 나서의 서명수가 더 적더라 라는 이야기가 돌기 시작했고, 어떤 사람은 아예 화면을 동영상으로 캡처하여 실제로 F5를 눌러 화면을 갱신했을 때 실제로 서명수가 줄어드는 장면을 올리기도 했습니다.

이는 모든 정보를 독점하고 있는 포털의 구조상 언제라도 제기될 수 있는 문제입니다. 의견을 낼 수 있는 통로를 포털이 독점하고 있고, 의견을 수렴한 결과를 투명하게 공개하지 않으며, 이러한 포털은 기업 논리상 정부 기관에 협조적일 수밖에 없다는 인식이 이러한 문제제기를 낳게 되는 거죠. 실제로 포털은 미디어의 역할을 일부 수행하고 있으며, 그것도 사회에 상당한 영향력을 미치고 있습니다. 포털은 뉴스를 생산하진 않지만 이를 유통시키고 있으며, 제휴사들로부터 들어온 다양한 뉴스와 사용자가 작성한 UCC를 어떻게 편집하느냐에 따라 논조를 만들 수 있기 때문이죠.

그런데 아고라의 서명수 조작 문제는 약간의 오해가 들어간 부분이 있습니다. 일단 아고라 같이 대량의 트래픽을 다루는 게시판은 여러 대의 웹서버가 트래픽을 분산해 처리하는 멀티서버 구조를 가지고 있습니다. 만약 아고라 게시판 웹서버가 10대 있고 모든 웹서버는 동일한 역할을 담당하도록 만들면, 서버가 한 대 있을 때보다 산술적으로 10배 많은 트래픽을 처리할 수 있게 되는 것입니다. 보통 이렇게 구성한 서버의 제일 앞에는 요청을 분배하는 서버가 따로 있어서,

요청이 들어오면 가장 한가한 서버에게 요청을 돌려주는 역할을 하게 되는데 이를 로드밸런싱(Load Balancing)이라고 합니다.

이렇게 멀티서버 구조를 지닌 시스템에선 다음과 같은 현상이 일어납니다. 1번 글을 읽을 땐 1번 서버에서 요청을 처리하고, 2번 글을 읽을 땐 5번 서버에서 요청을 처리해서, 결과적으로 매번 다른 서버에서 글을 읽어오게 되는 것이죠. 글을 쓸 때도 마찬가지로 처음 쓴 글은 2번 서버에서 저장시키고 그 다음 쓴 댓글은 9번 서버에서 저장시키는 식으로 됩니다. 물론 최종적으로 글이 저장되는 공간인 데이터베이스는 하나만 존재하여 2번 서버에서 쓴 글이라도 5번 서버에서 읽을 수 있도록 되어 있습니다.

이런 경우 가장 문제가 되는 부분이 어디일까요? 일단 웹서버를 여러대 확보했기 때문에 사용자가 웹서버에 보내는 트래픽은 웬만큼 처리가 됩니다. 하지만 웹서버는 여러 대인데 데이터베이스 서버는 한 대뿐이기 때문에, 웹서버에서 데이터베이스 서버로 요청이 가는 부분에서 병목현상(Bottleneck)이 발생하게 되죠. 그래서 멀티서버 구조에선 데이터베이스로의 요청을 최대한 줄이기 위해 캐싱(Caching)을 사용합니다. 데이터베이스의 데이터를 미리 가져와서 보관하고 있다가 자신이 가지고 있던 글에 대해 읽기 요청이 발생하면 굳이 데이터베이스를 갔다 오지 않고 조회수를 하나 올린 다음 자신의 보관하고 있던 데이터를 돌려주는 것이죠.

이렇게 캐싱을 활용하면 웹서버와 데이터베이스와의 병목을 크게 줄일 수 있습니다. 하지만 모든 요청을 이렇게만 처리한다고 가정하면 큰 문제가 발생합니다. 여러 대의 서버가 서로 다른 요청을 처리하기 때문에 서버들이 갖고 있는 데이터에 점점 차이가 생기게 되는 것이죠.

예를 들어 조회수가 10인 1번 글을 1번 서버에서 읽었다고 가정하면, 1번 서버는 1번 글의 조회수를 11로 갖고 있지만 나머지 서버들은 여전히 10으로 갖고 있게 되는 셈입니다. 따라서 이렇게 서버마다 발생한 차이들을 한꺼번에 맞춰주는 일이 필요한데, 이를 동기화(Synchronization)라고 부릅니다. 동기화는 사용자가 이런 차이를 눈치 채지 못하도록 충분히 짧은 시간 동안 이루어져야 하며, 동기화가 오히려 서버에 부담을 주지 않을 만큼의 충분한 시간 간격을 뒤야 합니다. 따라서 멀티서버 시스템에선 동기화를 어떤 간격으로 잘 하느냐가 매우 중요한 일이 되는 거죠.

그럼 아고라 조회수 조작 문제로 돌아가 봅시다. 아고라의 게시판 시스템이 어떻게 구성되어 있는지는 잘 모르지만, 아마 멀티서버 구조로 되어 있음은 분명합니다. 아고라의 추천수는 각각의 서버들이 처리한 결과를 동기화를 통해 맞춰진 값으로 나올텐데요, 이명박 탄핵 서명 같이 순식간에 엄청난 양의 트래픽이 몰리는 경우에는 동기화가 추천수의 변화량을 따라잡지 못하는 현상이 발생합니다. 이런 경우 F5를 눌러 화면이 갱신되었을 때 미처 추천수를 동기화하지 못한 서버에서 추천수 값을 가져와 추천수가 더 줄어든 것처럼 보이는 현상이 발생할 수도 있습니다. 이런 경우 추천수의 감소량은 매우 적은 수로 나타날 것이며 시간이 지나 동기화가 이루어지면 제대로 된 추천수 값으로 보일 것입니다.

포털에서 여론을 조작하는지 안하는지는 확실하지 않습니다. 하지만 여론을 조작하고 싶다 하더라도 데이터를 임의로 지우거나 조작하는 방법은 너무 거칠고 위험한 방법이라 생각됩니다. 그런 경우 아마 블라인드 처리를 하거나 서버락을 발생시켜서 접근을 어렵게 하는 방법을 사용하겠죠.

어쨌거나 아고라의 급부상과 인터넷 여론의 표출은 미디어 지형의 큰 변화를 가져올 것임이 분명합니다. 온라인에서 발생하는 다수 여론의 힘을 알게 되었고 이 힘이 오프라인까지 진출하게 된 사건을 경험하였기 때문이죠. 이 힘을 민주적인 방법으로 발생시키고 민주적으로 사용할 수만 있다면, 타락한 대의민주주의를 탈피한 새로운 형태의 민주주의를 실현할 수도 있지 않을까요.

레니의 테크토크

공유의 미덕 - 공유기

레니 | 진보네트워크센터 자원활동가 <http://blog.jinbo.net/reneegade>

한때 공유기는 컴퓨터 좀 쓴다고 하는 집에서나 사용하던 물건이었습니다만, 이젠 웬만한 집에선 무선 공유기 하나쯤 있을 정도로 상당히 대중화된 기기가 되었습니다. 언젠가 글에서 KT의 공유기 추적에 대해 다룬 바가 있습니다만, 사용자에게서 공유기가 일반화 되다 보니 공유기 사용 사실을 알아도 현재로서는 제재할 방법이 없다고 할 정도니깐요. 이번에는 멀뚱하게 생긴 공유기 하나가 어떻게 선 하나에서 들어오는 인터넷 라인을 여러 개로 나눠 사용 가능하게 해 주는지 알아보도록 하겠습니다.

공유기의 가장 기본적인 기능은 하나의 선을 여러 개의 선으로 나눠 주는 겁니다. 좀 더 정확하게 얘기하면, 하나의 IP 주소를 여러 개의 IP 주소로 나눠주는 것이라 할 수 있겠군요. 보다 정확하게 얘기하면, 여러 개의 IP 주소를 사용 가능하게 하되, 하나의 IP 주소로 외부와 연결되도록 도와주는 것이라고 하겠습니다.

IP 주소란-간단하게 말하면-네트워크와 연결된 컴퓨터의 고유 주소 같은 겁니다.

니다. 실세계에서도 집주소가 있듯이, 네트워크로 연결된 컴퓨터들의 세계에서도 IP 주소라는 주소가 존재합니다. 다른 컴퓨터를 네트워크를 통해 접근하려면 반드시 이 IP 주소를 알아야만 하는 것이죠. 관련해서 예전에 네트워크에 연재한 글을 참조하시면 조금 쉬우려나...모르겠습니다;;

* IP 주소 :

http://networker.jinbo.net/zine/view.php?board=networker_4&id=1756&page=1&category2=66

* 도메인 :

http://networker.jinbo.net/zine/view.php?board=networker_4&id=1799&page=1&category2=66

실생활에서도 같은 주소가 두 개 존재하면 안되듯 네트워크에서도 같은 주소가 중복해서 존재할 수 없습니다. 일반적으로 사용하는 IP 주소, 즉 IPv4는 xxx.xxx.xxx.xxx의 형식으로 0~255사이의 숫자 4개를 조합해서 만들어집니다. 이 방식으로는 이론적으로 256의 네제곱, 즉, 약 43억 개의 주소를 나타낼 수 있게 됩니다. 여기서 특수한 용도로 예약된 주소가 존재하기 때문에 실제로 사용할 수 있는 주소는 이보다 좀 더 적겠지만 말이지요. 하지만 네트워크에 연결된 컴퓨터가 폭발적으로 증가하여 더 많은 주소를 표시할 수 있는 방법이 요구되었죠. IPv6는 이런 상황에 따라 대안으로 제시된 IP 주소 표시 방법인데, 앞으로는 점차 IPv6를 지원하는 곳이 늘어나게 될 것 같습니다.

IP 주소를 더 많이 사용하기 위해서는 이렇게 주소 체계를 바꿔버리는 방법이 가장 근본적인 대안이 되겠지만, IPv6가 제안되기 이전부터 다른 방법들을 통해 IP 주소를 확장하는 방법이 사용되고 있었습니다. 바로 유동 IP나 사설 IP의

사용이 바로 그것입니다.

어디서 접속하던지 관계없이 접근할 수 있는 고정된 주소를 고정(static) IP라고 하는 반면, 접속할 때마다 변경된 IP를 받는 것을 유동(dynamic) IP라고 합니다. 유동 IP를 사용하는 대표적인 사례가 KT, 하나로 등이 제공하는 ISP 망을 사용할 때입니다. 이런 ISP 업체들의 사용자 PC가 항상 켜져 있지 않기 때문에, DHCP를 통해 이미 확보해 놓은 일정 범위 내의 IP를 ISP 망에 접속할 때 임대해 줍니다. 그리고 PC가 망에서 접속을 끊을 때 다시 이 IP를 회수해서 다른 사용자의 PC가 접속해 올 때 다시 내 주는 식의 재활용을 하게 되죠. 따라서 사용자의 PC는 망에 접속할 때마다 IP 주소가 달라지기도 합니다. 대부분의 사용자는 자신의 IP가 몇 번이 되던 크게 관계가 없는 경우가 많지만, PC를 서버로 사용하기 위해 고정 IP를 부여받을 필요가 있을 때는 ISP 업체에 추가 요금을 지불해야 하는 것이 일반적입니다.

이에 비해 사설(private) IP를 사용하면 외부에서 접근 가능한 IP는 하나로 고정하되 내부에서 사용하는 IP를 임의로 부여하여, 하나의 IP 주소를 여러 대의 PC가 나눠 쓰게 됩니다. 이때 외부에서 접근할 수 있는 IP, 즉 공인(public) IP를 여러 개의 사설 IP로 분배하는 기능을 하는 것이 공유기입니다. 공유기는 외부에서 받아온 IP 주소를 내부 컴퓨터에 연결해 줌과 동시에, 내부에 연결된 컴퓨터들에게 사설 IP를 부여하여 공유기에 연결된 컴퓨터끼리는 서로의 IP 주소를 통해 식별 가능하게 해 줍니다.

예를 들어, 공유기를 사용하지 않는 컴퓨터 A가 네트워크에 연결되었을 때 100.101.102.103이라는 IP를 부여받는다고 생각해 봅시다. 이 경우, 외부에 있는 컴퓨터 B에서는 100.101.102.103이라는 IP를 통해 컴퓨터 A로 접근할 수 있습니다. 그럼 공유기를 통해 컴퓨터 A가 연결되어 있는 경우를 생각해 봅시다.

이때는 공유기가 네트워크에 접속하면서 100.101.102.103이라는 IP 주소를 부여받습니다. 동시에 공유기는 컴퓨터 A에게 10.0.1.2라는 사설 IP를 부여해 줌과 동시에 자신은 10.0.1.1이라는 사설 IP 주소를 갖게 되죠. 이 경우 외부의 컴퓨터 B는 100.101.102.103이라는 공인 IP로는 컴퓨터 A로 접근할 수 없습니다. 컴퓨터 A는 공인 IP가 없고 사설 IP만 있기 때문에 외부에서 이 컴퓨터로 접근할 수 있는 주소 자체가 없는 셈이죠. 하지만 컴퓨터 B를 공유기에 연결하면 B는 10.0.1.3이라는 사설 IP를 부여받게 되고, 이제 10.0.1.2라는 IP 주소를 통해 컴퓨터 A에 접근할 수 있게 됩니다. 이렇게 사설 IP는 IP 영역을 확장시키는 역할을 하면서, 동시에 폐쇄된 네트워크 내의 컴퓨터들을 외부로부터 접근할 수 없도록 보호하는 역할도 하게 됩니다.

그리하여 현재의 상황은 유동 IP와 공유기의 활발한 사용을 통해 더 많은 컴퓨터가 네트워크에 연결되어 있다고 할 수 있습니다. 이는 어떻게 보면 공유기 사용자와 유선 사업자 사이의 암묵적이고 한시적인 평화 협정이 맺어진 상황이라고도 할 수 있겠습니다. 하지만 과점된 시장인 유/무선 인터넷 사업의 본질적인 한계와 정부의 인터넷 규제 움직임 등, 이러한 평화가 언제까지 보장될지... 모르겠군요. 공유기에 대한 강력한 제재가 가해진다면, 앞으로는 선택된 컴퓨터만이 네트워크에 접속할 수 있는 영광을 누리게 될 수도 있지 않을까요.

블로거 To 블로거

332 그 사람이 바라보는 곳을 상상하게 되는 블로거 / 녀름

334 그러니까 염동이님 블로그는 재밌다 / 썩은돼지

그 사람이 바라보는 곳을 상상하게 되는 블로거

- 세상도 썩고 돼지도 썩었다, <http://blog.jinbo.net/batblue>

너름 | 블로거 <http://blog.jinbo.net/mbc112>

0.

블로그에서조차 사회화된 인간의 모습으로 반드시 살아야 한다면 아마도 미쳐버릴 지도 모른다. 그래서 나와 가능한 한 멀리 있어서 그 사람의 사람 됨됨이 따위 내가 알(경험할) 필요도 없는 사람의 블로그를 소개하고 싶었다. 블로그에서 어쩌면 이 사람이 보여주고 싶어 하는 모습 그대로 나도 보면 그만이지 않나.

1. 기억

라면 진짜 맛있게 먹는 방법(<http://blog.jinbo.net/batblue/?pid=350>)을 읽고 사실은 “썰린다”라고 생각했었다. 특히 마요네즈, 개인적으로 정말 마요네즈를 안 좋아해서 처음에 “혀” 했다. 하지만 남은 라면을 버리지 않고 또 먹는 센스, 언제나 빈곤한 자취생인 내게 조금은 요긴하고 기발하게 보였다. 그리고



강한 이펙트를 주는 이런 포스팅은 블로거를 기억하게 한다. “썩은 돼지” 웬지 이름과 포스팅이 잘 어울린다고 생각했었다.

2. 상상

썩은 돼지는 많은 사진을 올리지만 내가 제일 꽂혔던 것은 “백수들 (<http://blog.jinbo.net/batblue/?pid=499>)” 이다. 제목을 보고 바로 클릭을 했다. 그리고 나도 진짜 크게 웃었다. 아하하하. 백수 너도 이렇게 사진으로 보니 너도 별거 아니네 했다. 사진을 보면서 줄에 매달려 널부러진 장갑을 발견하고 자신의 처지를 생각하고 그리고 찍었을 사람을 생각하게 됐다. 그리고 그 찍은 사진을 보면서 내 처지까지 떠올리는 나도 생각했다.

썩은 돼지의 풍경 사진들을 나는 좋아한다. 왜냐면 왠지 이 사람이 이 풍경 뒤에서 우두커니 보고 있었을 거 같은 느낌이 들기 때문이다. 그래서 나도 썩은 돼지의 사진을 천천히 본다. 사진에 찍힌 사건을 보기도 하지만 사진 찍고 있을 이 사람을 상상하는 시간이기도 하다.

3. 프락티카

썩은 돼지 사진 이야기의 첫번째 포스팅으로 갔었다. 첫번째 사용했던 카메라가 프락티카였다는 구절에서 너무 반가웠다. 내가 처음으로 사용해본 카메라가 프락티카였거든. 그 첼퍼덕 소리를 내는 수동카메라와 내가 놀았던 기억이 나잖아.

0.

사실 나는 다른 이들의 블로그의 글을 아무리 많이 읽어도 결국은 나로 돌아온다. 이 얼마나 자기적인 블로그 생활인가. 하지만 다른 이들의 경험과 감상을 느끼면서 그 사람들을 상상하는 시간은 결국 내가 나로 오는 과정이어서 너무 중요하다. 그래서 자신의 이야기를 블로그로 나누어 주는 당신들에게 오늘도 매우 고마워 하고 있어요. 헤헤헤

블로거 To 블로거

그러니까 염동이님 블로그는 재밌다

- 염동이 <http://blog.jinbo.net/ys1917/>

썩은돼지 | 블로거 <http://blog.jinbo.net/batblue>

솔직히 진보 블로그는 그닥 재미가 없다. 자주 포스팅하는 나름 유명한 진보 블러거들의 글은 진지한 주제가 많다. 나조차도 사진을 올리다 보면 스스로 진지해 질려고 하니 말이다. 그런 진보 블로그 내에 그나마 몇몇 있는 유쾌한 블로거가 있는데 염동님 블로그는 진보블로거 계에서 그 유쾌함으로 치자면 단연 북쪽하늘 전체를 밝게 비추는 별이라 할 수 있다. 화를 내도 화나 보이지 않으시고, 웃기는 얘기도 그 절제가 있으시는, 당신은 우후훗 야구쟁이.



나는 야구를 좋아하지 않는다. 사실 싫어하기 까지 한다. 그러나 가끔 염동님 블로그를 보고나면 야구에 관심을 갖게 된다. “야구선수 님은 꿀(수정버전) *이벤트있음(<http://blog.jinbo.net/ys1917/?pid=594>)”을 보고 나면 그 범상치 않은 포스팅 철학에 동참하지 않을 수가 없다.

“이 포스팅의 목적은 야구를 널리 알려 세상을 이롭게 하려 함이
라.”

아아 포스팅 하나조차도 이리도 명쾌하게 목적을 밝히면서 던져 주시니 반하
지 아니할 수가 없다.

염동님은 분석이 매우 정확해서 재밌다. 염동님의 정치적 감각은 리더의 유형
(<http://blog.jinbo.net/ys1917/?pid=598>)이라는 포스트에서 그 탁월함이 드러
난다.

머리 나쁘고 부지런한 유형

- 일을 정말 대책없이 많이 벌린다.
 - 고집이 유독 강하고 남탓을 정말 많이 한다.
 - 부지런하면 안될 일이 없다는 미신을 믿는다.
 - '잠재적 능력에 반비례하는 욕심'이라는 제6의 장기를 늘 달고
다닌다.
 - 수습할 게 너무 많아 후임자가 개고생한다.
 - 주변 사람을 가장 고생시키는 스타일이다.
- 대개 이런 사람이 리더가 되면 그 조직은 '고난의 행군'을 각오해
야 한다.
- > 이명박
----> 트윈스 팬으로선 도저히 입에 담을 수 없는 금지어. 순
Fe.

이 야구와 정치를 접목한 이 정치 감각이야말로 읽는 이를 털려하게 만든다.

김호형

<http://blog.jinbo.net/ys1917>



김호형

내가 본 것이 무엇이나

최근 포스트

- PS 일정
- 알다가 잠든 아가...
- 은적

최근 댓글

- 나도...내가 영랑으로 놀며
논 각도까지...
- 나랑 성격이 비슷하군. 나
두 누가 내 ...
- 정말 많이 많이 불평하셨것
여요! 정말 ...

전체보기

- 시달
- 미술미랑
- 피끄러기

PS 일정 | 피끄러기

2008년 09월 26일 09:54

추천

포스트시즌 일정이 나왔다.
좋은 문헌에만 올리고 잠실엔 가지 않았다.
가지 않아서 결과가 그만큼 것인지
결과가 그래서 가지 않게 된 것인지
지금으로선 그조차 생각할 겨를이 없고
포스트시즌 일정한 일단 올려놓는다

일자	요일	경기 일정(구장)
10월 8일	수	준PO 1 (승률 3위팀 홈구장)
9일	목	준PO 2 (승률 3위팀 홈구장)
10일	금	미동일
11일	토	준PO 3 (승률 4위팀 홈구장)
12일	일	준PO 4 (승률 4위팀 홈구장)
13일	월	미동일
14일	화	준PO 5 (승률 3위팀 홈구장)
15일	수	미동일
16일	목	PO 1 (승률 2위팀 홈구장)
17일	금	PO 2 (승률 2위팀 홈구장)
18일	토	미동일

그런 분석력은 일상의 삶에서도 마구 드러난다.

“나는 가끔 아버지가 가만히 앉아서 어머니 우리들이 갓다 주는 밥
을 먹을 때마다

남자들과 애완건의 공통점 세 가지를 떠올리곤 한다.”

- 1) 털이 많다.
- 2) 어려운 얘기는 못 알아 듣는다.
- 3) 삼시 세 끼 밥을 꼭 줘야 먹는다.

여기서 아버지는 털은 별로 없으니 2번과 3번만 올인이다.

오오 염동님의 통찰력이 주는 이 유쾌함에 배를 잡고 웃곤한다.

그러나 염동님의 재치가 빛났던 곳은 앞발로 쓴 이슬이 일기 (<http://blog.jinbo.net/ys1917/?cid=3>)와 이슬이의 눈 (<http://blog.jinbo.net/ys1917/?cid=2>)이었다. 이젠 YS와 영영 헤어져서 더 이상 블로그에서는 볼 수 없지만 이슬이의 눈과 앞발을 통해 본 세상사는 염동님의 독백인 것 같기도 하고 아닌 것 같기도 하면서 YS의 일상으로 재미있게 그려 지곤 했다. 나도 이슬이 같은 앞발과 세상을 보는 눈을 기르고 싶은 생각이 들곤 했지만 이슬이의 일기는 2006년 06월 19일 자로 끝났다. 그러나 염동님의 블로그는 계속 되고 있다.

@ ActOn 1011

기획 : 열린문서를 만들자!

- 340 누구나 읽을 수 있는 열린 문서를 만들자! / 정보공유연대 IPLeif
- 346 '열린 문서'로 문서의 가치는 쑥 올리고, 다른 이들을 배려하는 센스까지 / 김지성
- 353 리눅스를 쓰고 싶어요... / 지각생)
- 358 PDF 문서 만들기, 어렵지 않아! / 오병일

누구나 읽을 수 있는 열린 문서를 만들자!

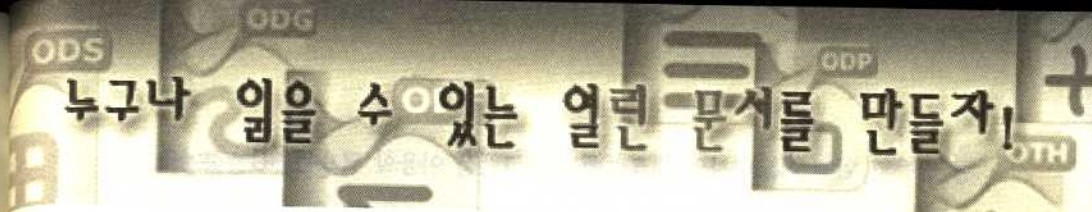
정보공유연대 IPLeft | <http://ipleft.or.kr>

취지

아직 한글97를 쓰고 있어서, 다른 사람이 보내 준 한글 2007 첨부파일을 읽지 못해 분통을 터뜨린 일은 없습니까? 혹은 MS 워드 프로그램이 없어서 첨부파일을 읽지 못한 경우는요? 단순히 텍스트로만 되어있는 문서인데도 첨부파일로 보낸 경우, 파일을 다운받고 프로그램을 실행시키기 귀찮아서 그냥 읽지 않고 넘어간 경우도 많지 않나요?

아직도 시민사회단체에서 주로 이용하고 있는 문서 편집기는 '아래아한글'입니다. 대부분의 문서들이 '아래아한글'로 작성되며, 메일링리스트나 홈페이지 게시판을 통해 유통되고 있습니다. 대부분이 '아래아한글' 문서를 기반으로 하고 있으므로, 상호간의 문서 유통에 별 불편함을 느끼지 못하고 있습니다. 그러

* 정보공유연대와 진보네트워크센터에서는 누구나 문서를 읽을 수 있도록 열린문서를 만들어서 사용자는 캠페인을 시작하였습니다. 웹진 액트온에서는 열린 문서를 사용하는 것이 왜 좋은지, 지금은 어떠한 문제들이 발생하고 있는 건지 몇 호에 걸쳐 기획연재로 소개하려고 합니다. 많은 관심과 참여를 부탁드립니다!



나 '아래아한글' 문서인 hwp나 MS워드 문서인 doc는 표준 문서포맷이 아니며, 그래서 다음과 같은 문제를 야기할 수 있습니다.

hwp나 doc와 같이 표준이 아닌 문서포맷을 이용하는 경우, '아래아한글'이나 'MS워드' 프로그램을 가지고 있지 않은 사람이나 쓰기를 원하지 않는 사람은 그 문서를 읽을 수 없습니다. 예를 들어, 비록 소수이지만 리눅스 운영체제를 이용하는 이용자의 경우에는 '아래아한글'에 기반한 문서 유통에 어려움을 겪고 있습니다. (물론 리눅스용 한글 프로그램이 있기는 합니다만, 유료로 구입을 해야 할 뿐만 아니라, 자유/오픈소스 소프트웨어 이용자로서 독점 소프트웨어를 이용해야 한다는 것은 곤혹스러운 일이기도 합니다.)

뿐만 아니라, 낮은 버전의 소프트웨어를 이용하고 있는 이용자는 높은 버전의 소프트웨어로 만들어진 문서를 읽을 수 없습니다. 예를 들어, 한글97 프로그램을 이용하고 있는 사용자는 한글2007로 만들어진 문서를 읽을 수 없습니다. 아무런 생각 없이 한글2007 문서로 배포하는 것은 한글 프로그램을 가지고 있지 않거나, 낮은 버전의 한글 프로그램을 가지고 있는 (대부분 재정적으로 취약한) 사람들에게 읽지 말라고 하는 것이나 다름없습니다. 보다 많은 사람이 접근하고 이용하도록 해야 할 시민사회단체에서 나온 문서들이 이와 같이 특정한 이용자를 배제하는 문서포맷을 이용하는 것이 합당한 일일까요?

hwp나 doc와 같이 표준이 아닌 문서포맷을 이용하는 것은 그러한 문서포맷을 제공하는 특정 업체의 프로그램에 종속되는 결과를 초래합니다. 표준 문서포맷인 txt, pdf, odf 문서를 만들기 위해 우리는 특정한 문서 프로그램만을 이용할 필요는 없습니다. txt 문서를 만들기 위해 '메모장'을 이용해야 한다면, pdf

를 만들기 위해 어도비사의 애크로벳 프로그램을 이용할 필요는 없는 것이죠. 그러나 hwp 문서를 만들기 위해서는 '아래아한글' 프로그램을 이용할 수밖에 없죠. 이는 문서의 작성자나 이용자 모두 특정한 프로그램에 '종속' 되는 결과를 초래합니다. 더불어 'MS워드'나 '아래아한글' 프로그램은 자유/오픈소스 소프트웨어가 아니라, 상용/독점 소프트웨어입니다. 이러한 상용/독점 프로그램을 이용하기 위해서는 그것을 구입, 업그레이드하는데 비용이 듭니다. 만일 불법 복제 프로그램을 이용한다면, 언제라도 고발 및 손해배상 요구를 당할 수 있는 위험에 처하게 됩니다.

만일 txt, pdf, odf와 같은 '표준 문서 포맷'을 이용한다면, 특정한 워드 프로그램에 종속될 필요도 없고, 특정한 프로그램을 갖고 있지 않은 사람들이 배제될 이유도 없을 것입니다. 이는 독점 소프트웨어에 대해 반대하며 자유/오픈소스 소프트웨어를 지지하는 운동과도 연결됩니다. 또한, 파이어폭스와 같은 인터넷 익스플로러 외의 웹 브라우저를 이용하는 사람이나 장애인의 원활한 인터넷 이용을 보장하기 위한 웹접근성 운동과도 연결됩니다.

현실적으로 지금까지 보편적으로 이용하던 프로그램을 바꾸는 것은 쉽지 않아 보입니다. 정치적인 활동보다 자신의 일상생활이나 습관을 바꾸는 것은 더욱 어려운 법입니다. 그러나 아래와 같은 실천을 쉬운 것보다 하나하나 해가다보면, 문서에 대한 접근성은 훨씬 향상될 것이고, 언젠가는 지금과 같이 특정 프로그램에 의존할 필요도 없어질 것입니다.

이렇게 합시다

편집이 필요 없는 텍스트 위주의 내용은 '텍스트' 형태로 유통합시다.

간혹 간단한 메시지나 주로 텍스트로 이루어진 문서도 한글 파일로 첨부하여 메일링리스트로 보내거나 게시판에 올리는 경우가 있습니다. hwp 파일로 첨부하면 리눅스 이용자는 읽기가 곤란할 뿐 아니라, 여타 이용자에게도 첨부된 파일을 다운로드한 후 열어보아야 하는 불편함이 있습니다. 불필요하게 전송 용량이 커지는 문제도 있겠죠. 편집이 필요 없는 텍스트 위주의 내용은 그냥 텍스트 형태로 유통하도록 합니다.

다소간 편집이 필요한 문서의 경우에도, 메일이나 게시판의 본문은 텍스트 형태로 채우고 해당 문서 파일을 첨부한다면, 이용자들이 문서 내용에 접근하기가 훨씬 용이해질 것입니다.

편집이 필요한 문서의 경우, 표준 문서포맷인 PDF 나 ODF(*.odt) 문서포맷을 이용합시다.

PDF (Portable Document Format)

PDF는 어도비사에서 개발한 전자문서 포맷입니다. 그러나 문서 포맷이나 제작 기술에 독점적인 기술이 사용되지 않았고, 문서 포맷이 공개되어, 어도비에서 개발한 소프트웨어 외에도 다양한 상용 혹은 자유 소프트웨어에서 PDF를 지원하고 있습니다. 또한 편집된 그대로 보여주기 때문에 전자 출판 용도로도 많이 이용되고 있습니다.

주고 받는 모든 문서를 PDF 문서 형식으로 하기는 힘들겠지만, 게시판이나 메일링리스트를 통해 외부로 공개되는 문서 형식으로는 PDF 문서를 이용하는 것이 좋겠지요.

* PDF 파일 읽기 : 어도비사에게 제공하는 애크로벳 리더를 이용하면 되는데, 무료로 다운로드 받아 설치할 수 있습니다. 이 외에 다양한 PDF 리더가 있으며, 리눅스 환경에서는 PDF 리더가 내장되어 있습니다.

* PDF 만들기 : 아래아한글, 오픈오피스 프로그램을 포함한 다양한 워드 프로그램에서 PDF 문서 생성기능을 제공합니다. 혹은 다양한 PDF 변환 프로그램이 있습니다. 윈도우에서 이용할 수 있는 PDF 변환 프로그램 중 무료로 다운로드 받을 수 있는 것은 PDF-Pro나 PDF-XP 등이 있습니다. (PDF-pro는 개인적 이용일 경우 무료 다운로드가 가능하며, PDF-xp는 프리웨어입니다. 프리웨어란 소스가 공개되는 자유/오픈소스 소프트웨어는 아니지만, 무료로 배포되는 소프트웨어를 의미합니다.) 리눅스 환경에서는 PDF 변환 기능이 내장되어 있습니다.

ODF (오픈다큐먼트포맷)

ODF는 편집가능 한 오피스 문서, 스프레드시트, 차트, 프레젠테이션을 저장하고 교환하기 위한 XML 기반의 공개된 문서 파일 포맷으로, 2006년 ISO와 IEC에 의해 국제표준으로 승인되었습니다. (한국 오픈다큐먼트 커뮤니티 : <http://opendocument.or.kr/blog/>)

ODF 는 국제표준 문서포맷이므로, 특정한 워드 프로그램에 종속될 필요가 없습니다. 오픈오피스, 스타오피스 등이 ODF를 지원하며, 2009년 상반기에 나

올 예정인 MS 오피스 2007 SP2에서 오픈도큐먼트 포맷을 지원할 계획이라고 합니다.

이중 오픈 오피스는 윈도 및 리눅스를 포함한 다양한 플랫폼을 지원하는 무료 오피스 프로그램입니다. 워드 외에도 스프레드시트, 프리젠테이션 등 다양한 파일을 읽고 쓸 수 있습니다. 아래아한글 3.0/97 및 MS 오피스 파일을 읽는 것도 가능합니다. PDF 형식으로서의 저장 기능 역시 제공합니다. 중요한 것은 개인적인 용도뿐만 아니라 업무상 이용할 경우에도 라이선스 걱정이 전혀 없는 자유 소프트웨어입니다.

‘열린 문서’로 문서의 가치는 쑥 올리고, 다른 이들을 배려하는 센스까지

김지성 | 진보네트워크센터/정보공유연대 활동가 to.jisung@gmail.com

“열린 문서 캠페인”을 가장 간단하게 이해하는 방식은 아마도 문서를 저장하는데 있어서 아래아한글의 hwp나 MS워드의 doc 파일 포맷과 같은 특정 회사의 문서편집 프로그램만의 독자적인 파일 포맷 대신에 txt나 HTML과 같이 공개된 표준 파일 포맷을 이용하자는 것일 것이다. 이렇게 이해한다고 해도 선뜻 “열린 문서” 또는 공개 문서 포맷을 사용해야 할 필요성을 느끼기는 그리 쉽지 않은 것 같다.

아마도 많은 분들이 이 캠페인이 특정한 문서편집 프로그램을 쓰지 말 것을 혹은 쓸 것을 주장하는 것이 아닌가 하고 생각하실 것 같다. 그러나 이 캠페인은 아래아한글이나 MS워드를 쓰지 말고 공개 문서편집 프로그램을 쓰라는 것이 아니다. 공개 소프트웨어를 쓰는 것의 장점도 있지만, 이 캠페인에 있어서 소프트웨어의 선택은 부차적인 문제다. 다시 한번 강조하자면 이 캠페인은 문서를 저장하는 포맷을 어떻게 선택할 것인가를 핵심적인 문제로 다루고 있다. 아래아한글이나 MS워드를 쓰더라도 저장하기 버튼을 눌렀을 때 고를 수 있는 파일 포맷은 최소한 대여섯 가지가 넘는다는 것을 생각해 보면, 여러분이 쓰는 소프트웨

어가 무엇이든지 여러분은 제공된 많은 선택지 중에서 어떤 파일 포맷으로 저장할 지를 결정해야 한다. “열린 문서 캠페인”은 여러분이 문서를 저장할 때 어떤 파일 포맷을 사용하면 좋을지 판단하는데 몇 가지 선택의 가능성과 판단 근거로 쓰일 수 있는 정보를 제공하려는 것이라고 이해해주었으면 한다.

문서의 저장 파일 포맷을 고민하고 선택한다는 것이 좀 어색하다고요?

머리 아픈 문제도 많고 선택해야 할 것이 너무나 많은 이 세상에서 문서를 저장하는 파일 포맷까지 고민을 한다니 너무 한 것 아니냐고 할 수도 있겠다. 그런데 자신도 모르게 이미 문서를 저장할 때마다, 그리고 문서를 다른 사람들에게 전달할 때마다 파일 포맷의 선택을 고민하고 있음을 생각한다면 문제는 달라질 것이다.

우리가 어떤 상황에서 어떤 선택을 하고 있는지 한번 살펴보자. 먼저 여러분이 속한 인터넷 동호회에 글을 올릴 때, 게시물에 간단한 제목 정도만 본문에 쓰고 hwp파일을 첨부하는지, 아니면 hwp 파일의 본문 내용을 복사해서 게시물 본문에 붙여넣기를 하는지 생각해 보라. 너무 긴 글이 아니라면 대부분의 경우는 아마도 복사해서 붙여넣기를 할 것이다. 그래야 동호회의 다른 회원들이 별도로 첨부파일을 다운로드 받고 아래아한글을 열어서 보는 복잡한 과정 없이도 바로 웹 브라우저에서 내용을 볼 수 있을테니 말이다. 이때, 복사해서 게시물 본문 작성란에 붙여넣기를 한 것은 기술적으로 보면 대부분의 경우 hwp 파일을 txt나 HTML 파일로 변환하여 저장한 것과 동일한 의미다.

다른 경우를 살펴보자. 같이 일을 하는 동료에게 새롭게 변경된 데이터를 바탕으로 작성된 표를 이메일로 보낸다고 생각해 보자. 표가 그리 복잡하지 않아서

표를 원래 작성했던 hwp 파일에서 해당하는 표를 복사해서 내가 쓰는 웹메일의 메시지 본문 작성 란에 붙여넣기를 하고 메일을 보낸다. 이 경우 심중팔구 이 표는 hwp가 아니라 HTML 포맷으로 바뀌어 동료에게 보내진다.

이 두 경우는 첨부할 것인지 본문에 바로 붙여넣기를 할 것인지와 관계된 것이지만, 우리가 문서 저장 포맷을 어떤 기준에 따라 선택하는지에 대해 많은 시사점을 보여준다. 조금은 비약해서 말하지만, 이 경우 파일 포맷의 선택의 기준이 된 것은 누가 이 문서를 어떤 환경에서 어떤 용도로 이용할 것인지에 대한 나의 판단이라고 할 수 있을 것이다.

자신만의 생각을 자신만이 볼 것을 전제로 문서를 작성한다면 파일 포맷의 선택은 상대적으로 간단해질 것이다. 내가 사용하는 컴퓨터에서 무리없이 작동하고 내가 익숙하고 구하기도 쉬운 문서편집 프로그램을 쓰면 될 것이다. 나는 어떤 소프트웨어가 내 컴퓨터에서 잘 작동하고 나에게 익숙한지 잘 알기 때문에 이러한 판단이 잘못될 확률은 아주 작다고 보아도 될 것이다.

그런데 우리가 작성하는 문서의 대부분은 누군가와 의사소통을 위한 경우가 많다. 우리는 내가 어떤 환경에 있으며 어떤 용도로 이용할지에 대해서는 상당히 분명하게 알고 있지만, 내가 작성한 문서를 받은 사람의 컴퓨터 환경과 이용할 용도에 대해서는 보다 적은 정보만을 가지고 있다. 내가 작성한 문서를 이용할 대상이 내가 특정하기 어렵거나 대단히 많은 사람들이 이용할 것이 예상되는 상황에서는 더욱 내가 작성한 문서를 보고 이용할 사람들의 컴퓨터 환경과 이용 방식에 대한 정보가 적어지고 불확실성은 높아진다. 이러한 불확실성이 높은 상황에서 어떻게 하면 여러분이 작성한 문서가 더 많은 사람들에게 유용하게 쓰일 수 있을까라는 고민이 바로 이 캠페인의 출발점이다. 그럼 이제 어떤 상황에서 '열린 문서'가 좋은 선택이 될 수 있는지 구체적인 상황을 살펴보자.

지금 저장하는 파일 포맷으로 문서를 보냈을 때 받은 사람이 볼 수 있다고 얼마나 확신하는가?

우선 내가 작성한 문서를 받아볼 사람의 컴퓨터(하드웨어와 소프트웨어)의 환경에 대해 가정하기 어려운 경우를 생각해보자. 예전에 내가 속했던 조직에서 전국에 고르게 분포한 수백 명의 활동가들을 대상으로 주간 소식지를 hwp파일로 정기적으로 이메일로 발송했다. 얼마나 많은 활동가들이 소식지를 제대로 보고 있었는지 모르지만, 평소에는 아래아한글 97 버전의 hwp파일로 보내던 것을 잘 못 해서 아래아한글 2004 버전의 hwp파일 포맷으로 보낸 적이 있다. 당장 전화로 항의를 해 온 활동가들이 있었다. 문제는 예상하다시피 아래아한글 2004 버전 소프트웨어가 없었던 것이다. 수십 페이지 정도의 분량에 제일 복잡한 서식이라고 해봐야 표 정도였던 소식지를 hwp로 보내지 않고 HTML 포맷으로 저장해서 보냈다면 어땠을까? 아래아한글 버전 문제로 이런 항의를 받지도 않았을 것이고, 혹시 리눅스를 쓰는 활동가가 있다고 해도 아래아한글을 구할 필요없이 웹 브라우저를 통해 볼 수가 있었을 것이다.

한글 뷰어 소프트웨어가 있으니 꼭 아래아한글 97 버전이 없어도 상관없는 것이 아닌가라고 생각해볼 수도 있다. 그래도 여전히 HTML로 보내는 것이 '안전한' 선택이 될 것이다. 지금도 그렇지만 앞으로 5년 후에 대부분의 컴퓨터에 아래아한글 97이나 한글 뷰어 소프트웨어가 있을 것이라는 가정보다는, 웹 브라우저(공개 소프트웨어 또는 윈도와 같은 운영체제를 만드는 회사가 추가 비용 없이 제공하는 웹 브라우저가 이미 윈도, 리눅스, 맥 OS 용으로 존재한다)는 최소한 있을 것이라는 가정이 더 현실적일 것이다. 나아가 웹 브라우저가 대부분의 컴퓨터에 깔려 있지 않더라도 HTML의 내용을 볼 수 있는 방법이 있다. 제목 글씨라든가 표의 모양은 제대로 보지 못하더라도 윈도의 메모장과 같은 텍스트 파일을 읽을 수 있는 간단한 소프트웨어에서 HTML 파일을 불러들이면 내

용은 다 볼 수 있다. hwp 파일이나 doc 파일은 이러한 텍스트 파일을 읽을 수 있는 소프트웨어에서 보면 많은 부분이 사람이 이해할 수 없는 기호 형태로 이미 변형되어 그 내용을 알 수 없는 경우가 많다.

문서를 받는 사람의 컴퓨터에 어떤 소프트웨어가 깔려 있을지 확실하지 않다면 가장 확실한 전달 방법은 줄 바꾸기 정도 외에는 아무 꾸미기 기능이 없는 txt 파일이나 웹 페이지 등을 만드는데 쓰이는 HTML 파일로 저장하는 것이 마음 편하지 않을까?

내가 지금 작성한 문서를 5년 후에 내가 다른 용도로 쓸 수도 있고, 지금 이 문서를 받은 사람이 내가 의도하지 않은 용도로 쓸 수도 있다고 생각해보았는가?

이제 문서를 작성한 내가, 그리고 문서를 받은 상대방이 문서를 이용하는 용도에 대해 가정하는 것이 어려운 경우를 생각해보자. 지금 촛불시국에서 인터넷에서 벌어지고 있는 보수 언론 불매운동에 대한 내 생각을 담은 글 한편을 내 친구 길동이에게 전하기 위해 작성했다고 가정해보자. 이 상황에서 파일 포맷은 내가 쓰기 편하고 친구 길동이 읽을 수 있는 포맷이면 될 것이고 별로 걱정할 필요가 없을 것이다. 그런데 5년이 지나고 10년이 지나서 돌이켜보니, 2008년 촛불시국이 대단히 중요한 일이었고 그때 길동이에게 보낸 글, 당시 다음 아고라에 올라온 글, 이곳 저곳에서 주최한 관련 토론회의 자료집 글들을 모아서 검색 가능한 온라인 데이터베이스로 만들어야겠다는 생각이 들었다. 수천 건을 모아서 개발자(또는 회사)를 찾아갔다. 그런데 개발자가 파일 포맷이 어떤 건 txt, 어떤 건 HTML, 어떤 건 hwp, 어떤 건 doc, 어떤 건 PDF라 데이터베이스에서 검색 가능하게 저장하는데 엄청난 비용이 든다거나 아니면 파일 포맷을 자동으로 변환할 기술이 없어서 못하겠다고 한다면 얼마나 가슴이 아프겠는가?

아마도 개발자도 아쉬워하면서 txt, HTML 또는 PDF 파일들 만이라면 포맷을 바꾸는데 드는 프로그램 개발 시간이나 수동으로 변화하는데 드는 인력을 거의 안들이고 금방 댈텐데요라고 말할 것이다.

앞에서 든 예가 좀 먼 이야기 같다면 좀 더 가까운 현실의 이야기를 해보자. 지금 여러분들이 일하는 조직에서 생산되는 각종 자료를 검색하는 시스템을 구축하려 한다고 가정하고 이는 개발자에게 이러한 검색 시스템을 만드는데 가장 많은 시간과 노력이 필요한 것이 무엇일까 물어보라. 아마도 백이면 백, 문서의 파일 포맷을 들 것이다. 공개 표준의 문서 포맷으로 문서들이 저장되어 있다면 대부분의 경우 이러한 포맷을 변환하거나 필요한 내용을 추출하는 것을 지원하는 소프트웨어가 공개 소프트웨어로 이미 존재하고 있을 확률이 아주 높기 때문이다. 있는 소프트웨어를 이용해서 약간의 자동화에 필요한 프로그래밍만 하면, 나머지는 일반적인 데이터베이스 프로그래밍을 할 줄 아는 사람이라면 상대적으로 쉽게 해줄 수가 있다.

이제까지 '열린 문서'로 저장하는 것이 문서를 받는 사람의 컴퓨터 환경과 해당 문서의 이용 용도에 대해 불확실한 가정을 할 수 밖에 없는 상황에서 어떻게 도움이 될 수 있는지, 가상의 또는 현실의 상황을 들어서 제시해왔다. 이런 불확실성이 여러분이 작성하는 문서에도 존재한다면, 인생은 불확실의 연속이니 지금 확실하다고 해도 시간이 지나면 불확실해지는 부분이 있다는 것까지 고려해본다면, 열린 문서로 저장하는 것이 손해 볼 일은 아니다.

이 글에서는 PDF는 잠시 언급하기는 했지만 PDF나 오픈다큐먼트 파일 포맷과 같은 '열린 문서' 표준 포맷은 거의 다루지 않았다. 이 두 포맷의 경우에는 txt나 HTML과는 좀 더 상황이 복잡한 측면이 있어서 이 글에서 다루기에는 무리일 것 같다. txt나 HTML 파일로는 내가 특정 문서에서 필요한 기능과 양식을

채워줄 수 없는 경우, 표준 포맷과 hwp나 doc와 비교하는 것은 좀 더 면밀히 다
를 필요가 있다. 다음 기회를 기약하며, 글을 읽어주신 여러분 감사합니다.

열린문서를 만들자

리눅스를 쓰고 싶어요...

지각생 | 정보공유연대/IT산업노조 활동가 fosswithyou@gmail.com

A : 저.. 부탁이 있어요..

B : 뭔데요.

A : 제가 리눅스를 쓸 수 있도록 도와주세요.

B : ... 전 리눅스 잘 몰라요. 다른데 가서 물어보세요.

A : 그런 게 아니랍니다.

B : 그럼 대체 뭐죠? 왜 이러세요!

A는 왜 그랬을까요? 그건 B가 하도 h*p 로 끝나는 문서를 많이 보내서입니
다. 무슨 소리냐구요?

리눅스용 그 프로그램도 있고, 여차하면 어느 사이트에서 이케저케하면 불러
올 수 있다는데? 리눅스 쓴다는 사람이 그것도 몰라? 혹은 그런걸 하기 싫어해?
이럴 지도 모르겠네요. 네, 압니다. 알아요. 다 방법이 있어요. 근데요, 그거 생
각보다 짜증나거든요. 리눅스용 아래아한글 요즘은 어떤지 모르지만 얼마 전만
해도 뭔가 잘 안돼서 삽질하다 겨우 조금 쓰고는 지워버렸구요. 그 인터넷의 서

비스도 일단 인터넷 연결하고, 그리 가서 로그인하고, 메뉴 클릭해서 불러오고.. 그래서 나온 결과는 그다 만족스럽지 않은데 그걸 다시 복사해서 원하는데 붙여야하고, 당신들은 그냥 더블클릭 한번이면 되는걸 난 얼마나 삽질을 해야하는지 알아?! 그리고 그게 한 두번이면 말도 안한다고, 그래서 하다 하다 그냥 나도 원도로 부팅하고, 한글 뷰어를 깔든 어둠의 경로를 통해 구하든 해서 그 파일들 몰아보고 치워 버린다고.

그래서 지금 A는 이메일로 날아오고, 게시판에 붙어 있는 hw* 파일들을 아예 무시해 버립니다. 먹고 사는 데와 직접적으로 연결됐을 때만 눈물을 머금고 어떻게든 열어볼 뿐입니다. A는 오직 리눅스가 좋아서, 흠 사실 독점 가식 구라 기업 MS를 싫어하긴 하지만 어쨌든 그냥 리눅스를 쓰고 싶어 하는 평범한 사람입니다. 귀찮은 거 싫어하고 끈기도 별로 없고, 철저하게 파고들고 잘 기억해두는 사람도 못 됩니다. A는 지금껏 윈도를 썼듯이 리눅스를 쓰고 싶고 이제 그런 것이 거의 가능하다는 것을 알게 됐습니다. 자, 이제 서로 서로 주고 받고 나누는 자유소프트웨어의 세계로 빠져드는 거야~

그런 A에겐 무슨 일이 일어났을까요? 네, 짐작할 수 있죠? 인터넷 बैं킹이 안 됩니다. ActiveX 꼭 필요한 서비스? 이용할 수 없습니다. 가끔 인터넷 익스플로러에만 먹는 짝퉁 코딩때문에 로그인도 안되는 사이트 많습니다. 그래도 곳곳이 참고 살아가립니다. 그까짓, 안가면 그만이야. 다 참을 수 있습니다. 늘 그 자리에 있는 것이 싫다면 안가면 되고, 돌아가면 되지요. 그러나.. 이게 끝이 아닙니다.

A는 인터넷 익스플로러 대신 모질라 불여우(Firefox)로 아주 만족스런 인터넷 서핑을 하고 있습니다. 어도비 포토샵 대신 '김프'로 웬만한 거 다 합니다. 아래아 한글이나 MS오피스 대신 오픈오피스로 필요한 문서들 다 만들어 내고 있

습니다. 새까만 창에서 명령 내리는 거 조금 배웠더니 이거 아주 컴퓨터로 할 수 있는 게 무지 많아지고 삶이 쾌적해집니다. 그 외에도 컴퓨터로 하는 것들이 거의 대부분 이제 리눅스에서도 별 차이 없이 할 수 있게 되었죠.

근데 A는 싸워야만 합니다. B와 싸웁니다. 그리고 C와도 싸웁니다. 왜 그럴까요? B와 C가 계속 자 이거 읽어 하며 무심코 던지는 "짹"들 때문입니다. 이메일에 첨부된 *wp 문서, 게시판에 올려져 있는 아*아한글 문서 파일. B와 C는 아마 스스로 얼마나 A에게 그것이 스트레스를 주는지 짐작도 못할 것입니다. 네, 그리고 실제로 B와 C는 죄가 없습니다. 죄가 있다면 호환성을 고려하지 않고(혹은 의도적으로) 소프트웨어를 개발한 사람들에게 있습니다. 그리고 언제나 효율성만 내세우며 사회적 소수에 대한 인식이 부족하고, 그래서 결국엔 더 큰 대가를 치르기를 반복하는 한국의 문화적 상황이 문제라면 문제죠. 어쨌든, B와 C는 오늘도 A가 자유소프트웨어를 쓰는 걸 힘들게 하고 있습니다. A가 독점 소프트웨어의 손에서 자유로워지는 것을 막는데 일조하고 있습니다. 알든 모르든 말이죠.

당신은 차별 중

의사소통을 위해선 어차피 말하는 사람과 듣는 사람과의 공통분모가 있어야 합니다. 같은 언어 등 서로 약속을 알고 있는 표현 수단을 써야하고, 비슷한 경험이 있거나 오래 같은 문화를 공유해야 그 사람의 "말"을 맥락 속에서 이해할 수 있지요. 그런 기반이 쉽고 익숙할수록 좀 더 "메시지"에 집중할 수 있을지 모르겠습니다. 컴퓨터로 뭔가 주고 받고 하는 것도 서로가 공통의 환경을 조성할 수 있는 가능성이 있어야 합니다. 즉 내가 아래아한글 문서를 만들어 보낸다는 것은, 받아볼 사람이 그 파일을 보고 바로 "아래아한글 문서라는 걸 알고", 아래

아한글을 “갖고 있거나 구할 수 있어야”하고, 필요한 만큼 사용할 수 있다는 전제를 두고 보내는 것입니다. 그러나 자꾸 하다보면 그 과정들에 대한 “가정”을 잊고 스스로 작성한 문서 내용이 마치 그대로 전달되는 것처럼 인식하게 되죠. 모든 문서를 출력해서 주고 받고 보관하고 열람하던 문화가 아직도 완전히 없어지지 않은 것도 관련이 있을 것 같습니다. 어쨌든 대체로 사람들은 특히 도구, 수단에 대해 쉬운 것을 선택해 익숙해진 다음, 잊어버리고 싶어 하는 면이 있다고 봐야 할 겁니다. “아 한국 사람들 아래아한글 많이 쓰잖아. 그리고 적어도 알긴 거의 알잖아. 괜히 다른 걸로 보내면 더 싫어할꺼야.”

네, 하지만 그렇게 이해하는 건 아무 일 없을 때나, 별로 첨부 파일의 내용이 궁금하지 않거나 할 때의 이야기고요. 이걸 꼭 읽어야 하는데 읽을 수 없는 상황이 되면 답답~합니다. 특히 보나마나 간단한 내용이고, 이쁘게 출력해서 볼 필요 없는, 정보만 잘 전달되면 되는 그런 종류의 문서일 때 단지 그것이 “내가 갖고 있지 않은 해석 수단”을 이용하게끔 강제해서 보냈기 때문에 못 본다는 것은 환장할 노릇입니다.

왜 그냥 텍스트 파일로 보내지 않습니까? 그렇게 보내는 것이 혹시 받는 이에 대한 무례라도 되는 것일까요? 간단하게 메모장 같은 거 열어 내용 넣고 저장해 보내는 것보다 워드프로세서로 작성해서 살짝 손봐 보내는 것이 한층 격이 있는 행동일까요? 내용이 아주 길거나 어려워서 가독성에 특히 유념해야 하는 경우가 아니라면 그냥 언제 어디나, 모든 컴퓨터로, 어떤 소프트웨어로도 읽을 수 있는 텍스트 파일로 그냥 보는 것이 오히려 받는 사람을 가장 잘 배려하는 행위입니다. 무심코 손에 익은 대로의 행동이 의도치 않게 누군가에 대한 차별이 될 수 있습니다. 존재를 망각하는 것, 그래서 소통과정에서 처음부터 배제시키는 행위다라고 말하면 너무 심한 표현인걸까요.

눈으로만 읽지 않는다

텍스트로 문서를 작성했을 때의 이점은 시간이 지날수록 커집니다. 컴퓨터 하드에 많은 파일이 쌓였을 때, 혹시 기억력이 좋은 분은 파일 이름이 어떻게 붙여져 있고, 어느 디렉토리에 있든 간에 필요할 때 바로 찾아볼 수 있는지 모르겠습니다. 정리를 잘 하시는 분은 디렉토리를 적절히 만들고 구조화해서 보관하실 거고, 파일 이름에 그 내용을 짐작할 수 있는 적절한 키워드나 날짜 등을 섞어두는 사람도 있을 겁니다. 하지만 이것도 파일이 몇백 몇천까지 늘어난다고 하면 나중에 필요한 내용을 찾는 것은 보통 힘든 일이 아닐 겁니다. 그런데 만일 파일이 텍스트로 저장되어 있다고 하면 파일이 아주 많아도 걱정할 필요가 없습니다. 컴퓨터한테 찾아달라고 하면 되니까요.

파일이 텍스트나 잘 알려진 표준 문서 포맷으로 되어 있을 경우 많은 파일 속에서 내용을 찾아내 보여 주는 것은 컴퓨터에겐 쉬운 일입니다. 윈도우보다 리눅스가 이런 일은 훨씬 잘합니다. grep 같은 명령어 등을 적절한 옵션과 함께 사용해 주면 아무리 파일이 많아도 잠시 후에는 원하는 파일이 제 앞에 모습을 드러낼 것입니다. 이런 좋은 환경을 두고 윈도우로 부팅해서 파일들을 뒤적이며 몇 번이고 반복해서 열고 닫아 찾으면 다행인 삶을 살라고 하는 것은 너무 잔인한 일이에요. 또 웹메일을 많이 쓰실텐데 역시 요즘 대부분의 서비스는 검색 기능을 제공합니다. 이때도 내용이 본문에 포함되어 있거나 텍스트 파일로 첨부되어 있다면 검색에 빠짐없이 포함될 수 있겠죠. 하지만 모든 내용이 특정한 워드프로세서만 이해할 수 있는 형식으로 저장되어 있을 경우라면 검색에는 엉뚱한 것만 쫓~금 나오고 말 것입니다.

당장 익숙한 환경을 버리기 힘들다면, 혹은 특정 워드프로세서의 기능을 꼭 이용해야 한다면 그냥 하던대로 하셔도 좋습니다. 하지만 맨 마지막에, “(반드시) 누군가는 내가 만든 문서를 못본다”는 걸 기억해주시고, 저장을 한번만 더 해주세요. 텍스트 파일이나, PDF로, PDF로 보내면 웹브라우저의 플러그인으로 바로 읽을 수 있다는 추가적인 이점들이 있습니다. 그리고 둘을 같이 보내는 겁니다. 아마 받는 사람은 처음에는 “왜 이래?” 그럴지 모르지만 나중에 이런 문제를 알게 되면, “오호.. 센스쟁이”하며 감탄할지도 모릅니다. 키보드 몇 번 더 두드리고 마우스 몇 번만 더 움직이면, 수많은 사람들이 이후로 계속 적은 수고로 그 문서를 읽을 수 있음을 잊지 말아주시기 바랍니다. 습관을 들이는 게 쉽지 않겠지만, 충분히 들일만한 수고입니다.

첨부파일 때문에 재부팅하는 경우가 있다면 믿겨지세요? 꼭 읽어야 할 독점 포맷 문서 몇 개가 있는데 인터넷이 안 될 때 읽어야 되서 재부팅하는 경우가 생기면 울고 싶어집니다. 인터넷에서 변환해 보는 서비스를 이용할 수 없으니까요. 윈도우에서야 업데이트를 하거나, 어떤 장치를 연결하거나, 프로그램을 설치하거나 그것도 아니면 무슨 무슨 에러 때문에 여러 번 재부팅을 하게 되지만 리눅스는 안 그렇거든요. 한번 전원을 넣으면 끝날 때까지 몇날 며칠이고 쓸 수 있는 게 리눅스입니다. 리눅스에서 재부팅한번 하는 건 윈도우에서 스무번 재부팅 하는 것보다 제게는 충격입니다. 부디 맘 편하게 리눅스 사용할 수 있도록 조금만 도와주세요. ㄱㄱ

PDF 문서 만들기, 어렵지 않아!

오병일 | 진보네트워크센터/정보공유연대 활동가 antiropy@jinbo.net

많은 분들이 열린문서 캠페인의 취지에는 공감(?)하면서도 참여하는데 현실적인 어려움을 겪고 있는 듯합니다. 간단한 문서는 텍스트 포맷으로 유통시키자는 것에는 큰 부담이 없을 것입니다. 캠페인 이전부터 이미 우리는 텍스트 포맷을 많이 사용해 왔구요. (물론, 열린문서 캠페인 이후에도, 편집이 필요 없는 문서임에도 불구하고, 한글 파일을 첨부해서 보내주시는 분이 여전히 계시더군요. 습관을 바꾸는 것은 역시 쉽지 않은 일이지요.) 그러나 PDF나 ODF는 아직 익숙하지 않아서 선뜻 사용하기 쉽지 않을 것입니다.

앞서 김지성씨 컬럼(‘열린 문서’로 문서의 가치는 쑥 올리고, 다른 이들을 배려하는 센스까지.)에서도 언급했습니다만, 열린문서 캠페인은 특정한 문서편집기 이용을 배제하자는 것은 아닙니다. 물론 아래아한글이나 MS 워드와 같은 독점 소프트웨어 대신, ‘오픈오피스’와 같은 자유 소프트웨어를 이용하는 것은 의미 있는 일입니다만, 그것이 열린문서 캠페인에서 주장하고자 하는 바는 아닙니다. 아래아한글이나 MS워드, 윈도우의 메모장, 구글의 문서도구 등 자신에게 익숙한 문서편집기를 이용해서 편집해도 상관없습니다. 다만, 다른 사람과 소통

하기 위한 공개된 문서의 경우에는 누구나 읽을 수 있는 표준적인 문서포맷을 이용하자는 것이지요.

열린문서 캠페인에서 PDF와 ODF를 제안했습니다만, 이 두개의 포맷만 의 미있다는 것은 아닙니다. 특정한 문서편집기에 종속되지 않은 표준 문서포맷이 라면 열린문서 캠페인의 취지에 모두 부합합니다. 예를 들어 HTML같은 경우 도 표준 문서 포맷입니다.

어떠한 문서편집기를 이용하여 작성한 문서이더라도, PDF나 ODF와 같은 표 준문서포맷으로 변환할 수 있습니다. 다만, 문서편집기에서 PDF 문서생성 기 능을 제공하지 않을 경우에는, 작성된 문서를 PDF 변환 프로그램을 이용하여 별도로 변환해주어야 하는 불편함이 따르겠지요.

현재 아래아한글은 ODF를 지원하지 않고 있습니다. (기사에 따르면, 2009년 하반기에 출시될 한글오피스에서는 지원할 계획이라고 하더군요.) 모든 버전을 확인해보지는 못했습니다만, 최근 버전에서는 PDF 변환을 지원하는 듯합니다. MS 워드는 PDF와 ODF 모두 지원하지 않고 있습니다. (MS 오피스도 내년 상 반기에 출시될 서비스팩2에 ODF와 PDF 지원을 포함한다고 하네요. 다만 PDF 변환 프로그램을 설치하면, 그 변환 프로그램을 이용하여 작성된 문서를 PDF 로 바로 변환할 수 있는 기능을 제공합니다.) 반면, 오픈오피스는 당연히 ODF 와 PDF 모두 지원합니다. 구글 문서편집 도구도 ODF와 PDF 모두 지원하네요.

이 글에서는 작성된 문서를 PDF 문서로 만드는 방법을 주로 소개하고자 합 니다. 별로 어렵지 않기 때문에, 방법만 알아두시면 문서 저장하듯이 쉽게 할 수 있습니다.

PDF 문서 읽기

다양한 PDF 리더(reader)가 있지만, 대표적인 것이 어도비사에서 제공하는 애크로벳 리더입니다. 홈페이지에서 무료로 다운로드 받아 설치할 수 있습니 다. 리눅스 환경에서는 PDF 리더가 내장되어 있습니다. 애크로벳 리더, Xpdf, Evince, KPDF 등이 리눅스용 PDF 리더입니다.

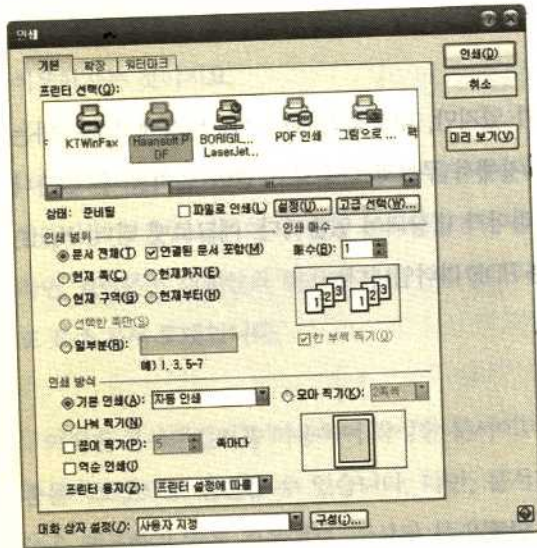
PDF 문서 만들기

1) 아래아한글에서 PDF 문서 만들기

한글 2005, 2007에서는 PDF 문서 생성 기능을 제공합니다. 다른 버전에서 는 확인이 필요합니다. 만일 자신이 갖고 있는 아래아한글에서 PDF 문서 생성 기능을 제공하지 않을 경우에는 '3-3) PDF 변환 프로그램 이용하기'를 참고하 세요.

PDF 문서를 만드는 방법은 간단합니다. 인쇄를 프린터가 아니라 PDF로 하 는 것입니다.

1. 아래아한글에서 문서를 작성한 후 [파일]-[인쇄] 메뉴를 선택합니다.(혹은 Alt-P를 누르면 됩니다.)
2. 프린터 선택에서 [Haansoft PDF] 혹은 [PDF 인쇄]를 선택합니다.
3. [인쇄]를 누르면 창이 나오는데, 생성되는 PDF 문서의 이름과 저장할 디렉 토리를 선택합니다.
4. [저장]을 누르면 PDF 문서로 변환합니다.



때로 프린터 선택에서 [PDF 인쇄]를 선택하고 인쇄를 했는데, 아무런 작동이 일어나지 않는 경우가 있습니다. 이는 아래아한글 프로그램에 PDF 인쇄 기능을 하는 프로그램이 설치되지 않았기 때문입니다. [도움말]-[한글과 컴퓨터 자동업데이트]

를 선택하셔서 업데이트를 하시면 PDF 인쇄 기능 모듈을 설치할 수 있습니다.

2) MS 워드에서 PDF 문서 만들기

MS 워드는 PDF 변환 기능을 제공하지 않고 있습니다. 다만, 내년 상반기에 출시될 MS 오피스 서비스팩2에서는 PDF를 지원한다고 하네요. 그러나 PDF 변환 프로그램을 설치하면, 그 변환 프로그램을 이용하여 작성된 문서를 PDF로 바로 변환할 수 있는 기능을 제공합니다. 아래 '3-3) PDF 변환 프로그램 이용하기'를 참고하세요.

3) PDF 변환 프로그램 이용하기

아래아한글이나 MS 워드 문서 등을 PDF 변환 프로그램을 이용하여 PDF 문서를 생성할 수 있습니다. PDF-pro, Primo PDF 등 윈도우에서 이용할 수 있는 PDF 변환 프로그램 중 무료로 이용할 수 있는 것들이 많이 있습니다.

▶ Primo PDF(<http://www.primopdf.com/>)

Primo PDF는 프리웨어입니다. (프리웨어란 소스가 공개되는 자유/오픈소스 소프트웨어는 아니지만, 무료로 배포되는 소프트웨어를 의미합니다.) 메뉴 등이 영어로 되어있어 조금 불편할 수 있습니다.

1. 다음 사이트(<http://www.primopdf.com/>)에서 프로그램을 다운로드 받습니다. [Download - Free]를 클릭하시면 됩니다.

2. 다운로드된 파일을 더블클릭하여 설치(인스톨)합니다.

3. 설치가 완료되면 컴퓨터의 [프린터 및 팩스] 메뉴에 Primo PDF가 추가됩니다. 아래아한글이나 MS 워드 프로그램에서 문서를 작성하신 후, [인쇄] 메뉴의 프린터 선택에서 [Primo PDF]를 선택하시면 됩니다.

4. [Primo PDF]를 선택하고 [인쇄]를 누르면 창이 나옵니다. 이 화면에서 생성될 PDF 문서의 품질이나 속성 등을 세부적으로 지정할 수 있습니다. (일단은 기본으로 지정된 설정을 사용하도록 합니다.) [Save As:] 부분에서 PDF 문서를 저장할 디렉토리나 문서 이름을 지정합니다.

5. [Create PDF]를 누르면 PDF 문서를 생성합니다.



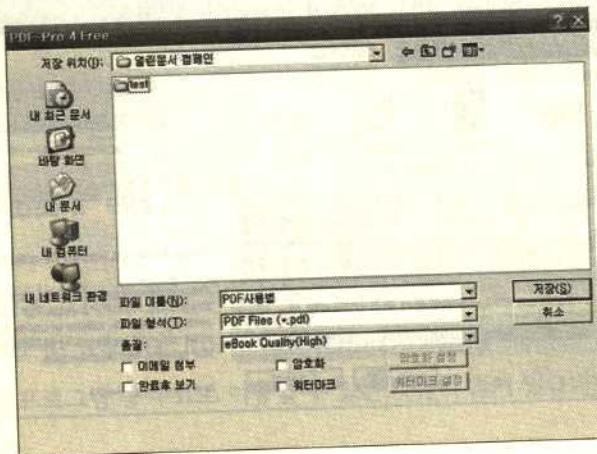
▶ PDF-pro (<http://www.pdfpro.co.kr/>)

PDF-pro는 개인적 이용일 경우 무료 다운로드가 가능하며, 한글 메뉴를 제공합니다. 그러나 기업이나 단체에서 이용할 때는 유료로 구입을 해야 합니다. PDF 문서 변환 뿐만 아니라, PDF 문서 읽기, 여러 개의 문서를 PDF로 일괄 변환하는 기능, PDF 문서의 편집 기능 등 다양한 기능을 제공합니다. 다만, PDF-pro로 생성된 PDF 문서는 하단에 “이 문서는 PDF-Pro 4 무료버전에서 생성된 문서입니다. www.pdfpro.co.kr”라는 문구가 자동으로 들어가게 되어 있어, 조금 거슬립니다.

1. 다음 사이트(http://www.pdfpro.co.kr/download/pdf-pro_free.php)에서 개인 사용자용 무료 프로그램을 다운로드 받으실 수 있습니다.

2. 다운로드된 파일을 더블클릭하여 설치(인스톨)합니다.

3. Primo PDF와 같이 [인쇄] 메뉴의 프린터 선택에 PDF-pro가 추가된 것을 확인할 수 있을 것입니다. MS 워드의 경우, [인쇄] 메뉴의 프린터 선택에 PDF-pro가 추가될 뿐만 아니라, [파일] 메뉴 아래에 [PDF-pro 4 free] 메뉴가 새로 만들어집니다. 이를 이용해서 MS 워드로 작성된 문서를 PDF 문서로 변환할 수도 있습니다.



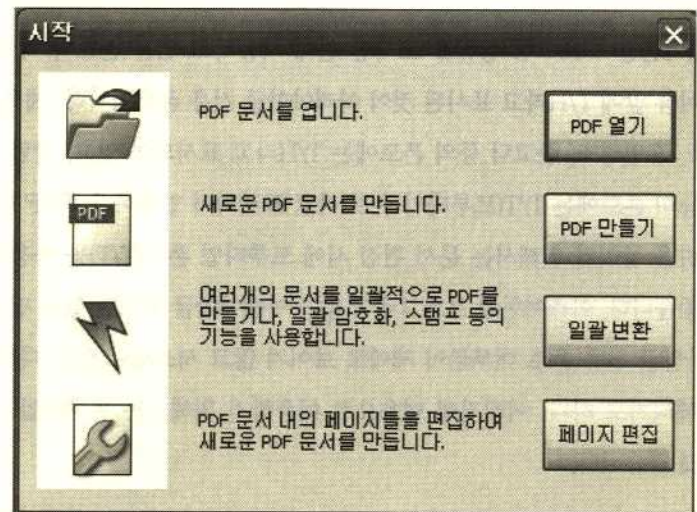
4. 프린터 선택에서 [PDF-pro]를 선택하고, [인쇄]를 누르면 창이 나옵니다. 여기서 PDF 문서 이름과 저장할 디렉토리, 그리고 PDF 문서 속성 등을 지정할 수 있습니다. [저장]을 누르면 PDF 문서가 생성됩니다.

아래아한글이나 MS 워드 등 문서편집기를 열지 않고, 문서 파일을 PDF-pro를 이용해서 PDF 문서로 변환할 수도 있습니다.

1. 윈도의 [시작]-[모든 프로그램] 메뉴를 보시면, [PDF-pro 4 Free] 메뉴가 추가된 것을 알 수 있습니다. [PDF-pro 4 Free] 메뉴를 누르면 프로그램이 실행됩니다.

2. 프로그램이 실행되면 [PDF 열기], [PDF 만들기], [일괄 변환], [페이지 편집] 등의 메뉴가 나타나는데, [PDF 만들기]를 선택합니다.

3. 먼저 변환할 문서를 선택합니다. [열기]를 누르고, 변환될 PDF 문서의 이름과 경로를 지정합니다. [저장]을 누르면 PDF 문서를 생성합니다. (이 때 PDF-pro를 기본 프린터로 지정할 것인지를 묻는 팝업창이 뜨기도 하는데, 기본 프린터로 지정할 필요는 없습니다.)



PDF 문서의 용량문제

아래아한글 문서나 워드 문서를 PDF 문서로 바꾸면, 보통 용량이 늘어납니다. 몇 페이지 안되는 작은 문서는 큰 상관이 없을 수도 있지만, 수백 페이지 분량의 큰 문서의 경우에는 PDF 문서로 바꾸는 것이 부담이 될 수도 있습니다. PDF 문서의 용량은 변환하는데 이용한 프로그램이나 PDF 문서의 해상도 등의 설정에 따라 달라집니다. 해상도가 높을수록 당연히 용량이 늘어나게 됩니다.

1) 문서 편집 시에 트루타입 폰트를 이용한다.

문서에 사용된 폰트에 따라서도 용량이 달라집니다. 특히 아래아한글에서 제공하는 전용 폰트를 사용하면 용량이 엄청나게(!) 늘어납니다. (해당 글자를 이미지로 처리하기 때문입니다.) 뿐만 아니라, 변환된 PDF 문서에서 글자가 역상으로 나타나 잘 보이지 않기도 하고, 검색이나 문서 내용의 복제 등도 되지 않습니다.

아래아한글 프로그램 상단을 보시면 현재 사용하고 있는 폰트가 나오는데, 폰트 이름 앞에 T/H라고 표시된 것이 아래아한글 전용 폰트입니다. 예를 들어, 신명조, 휴먼명조, 중고딕 등의 폰트에는 T/H라고 표시되어 있고, 굴림, 돋움, 바탕 등의 폰트에는 T/T(트루타입 폰트)라고 표시되어 있습니다. PDF 변환 시에 용량을 줄이기 위해서는 문서 편집 시에 트루타입 폰트(T/T)를 사용하시는 것이 좋습니다. 아래아한글 프로그램이 없이 아래아한글 뷰어만 있는 경우에도 아래아한글 전용 폰트 대부분이 제대로 보이지 않고 시스템에 있는 다른 폰트로 대체되기도 하니, 이래저래 범용으로 이용하기 위해서는 트루타입 폰트가 나온 것 같습니다.

2) 한글전용폰트를 트루타입 폰트로 바꾸기

그런데, 한글전용폰트가 사용된, 예전에 만들어둔 문서를 PDF로 바꾸려고 할 때 어떻게 하면 좋을까요? 그냥 PDF 문서로 전환하면 용량이 너무 커지게 됩니다. 그렇다고 예전 문서의 한글 전용 폰트를 트루타입 폰트로 바꾸자니, 매우 번거롭게 느껴질 수 있습니다.

그러나 아래아한글 문서에 사용된 한글전용폰트를 쉽게 트루타입 폰트로 바꾸는 방법이 있습니다. 만일 문서가 스타일을 써서 만들어져 있다면, 각각의 스타일(서식)에서 글꼴만 한글전용폰트에서 트루타입 폰트로 바꾸면 됩니다. 스타일을 쓰지 않은 경우에는 아래와 같이 하면 됩니다.

1. 아래아한글 메뉴 중에서 [파일]-[문서정보]를 선택하면 창이 열립니다.
2. 열린 창에서 [글꼴정보] 탭으로 이동합니다.
3. 현재 문서에서 쓰이고 있는 글꼴 중에서 한글전용글꼴(T/H로 표시되어 있는 글꼴)을 다른 트루타입글꼴(T/T)로 바꾸어주면 됩니다. 글꼴 바꾸기에 대한 자세한 설명은 다음 블로그의 글(<http://www.sonamu.biz/zbxe/87922>)을 참고하시기 바랍니다.



당신의 손끝에서 나오는 으랏차차 진보넷

19981114를 기억하시나요?

정보화라는 화려한 베일에 가려진 빅브라더와 독점자본의 질주에 브레이크를 걸기 위해, 자본과 권력으로부터 독립적인 네트워크를 꿈꾸면서 활동가들과 이를 지지해주는 수백명의 회원 여러분들이 진보네트워크센터를 만든 날입니다.

지문날인 거부운동, 주민등록제도에 대한 위헌소송 등을 기억하신다면, 그 안에 진보넷이 있었음을 기억해주세요. 우리 사회에서 최초의 의약품 강제 실시 운동에 진보넷이 역시 있었고, 정보공유라이선스를 만들어서 저작권의 다양한 활용을 고민했던 자리에도 진보넷은 꼭 있었어요.

진보넷의 '꼭 킴'은 여기에만 멈추지 않았습니다. 누구나 한 번쯤은 써보셨을 진보 메일링 리스트와 진보 공동체를 만들고, 사회운동 단체 활동가들이 프라이버시를 보호받으면서 표현의 자유를 마음껏 누릴 수 있는 호스팅 서비스를 구축하였습니다.

진보넷은 언제라도 어딘가에 누구에게 꼭 꼭 꼭 함께 하고 싶습니다. 항상 뛰어갈 수 있는 진보넷이 되기 위해서는 여러분의 도움이 필요합니다. 여러분의 손끝에서만이 으랏차차 기운 넘치는 진보넷이 나옵니다.

ella 4hacke bright igoomy ikd01 next421 **positro**

nenmai **wangkm** twofus shincl eeseob_n mari supul2

SHHA kambe anggoo10 **peacepia** Hserin tae21 may

jjinbo13 delphy action horong arcase meiste rainme

young82 redgan damon abird33 seapear underis **anbada01**

angle tsn eejoon **homin99** nojisimi

이 분들의 뒤를 이어 후원회원이 되고싶으시면...

쌀을 주셔도 되고,
반찬을 주셔도 됩니다.

쓰지 않는 이불을 주셔도 되고, 알뜰(?)을 주셔도 됩니다.
후원회원이 되셔서 지속적으로 으랏차차 기운 주는 방법도 있습니다.
후원회원이 되시면 빠르고 친절한 서비스를 자랑하는 **메일링리스트와 호스팅계정**, 즐겁고 풍성한 내용을 담은 **계간지 <액트온>**을 받아보실 수 있습니다. 또 하나! **독립 영화관 할인 및 무료 입장** 혜택을 받으실 수 있습니다.

후원회원이 되는 방법은 간단해요!
<http://center.jinbo.net/member/new.php> 에 방문해주시면 됩니다.



기업은행 057-036794-04-019
예금주 진보네트워크센터



To Be Continued

