

주체가 최소한 마이크로소프트나 오라클이 아니길 바라고 있었을 것이며, 그 역할을 맡은 조직이 모질라재단이었기 때문에 더 흥미를 가지게 되는 것도 사실이다. 그렇다면 갑자기 왜 모질라재단에서는 파이어폭스 OS라는 모바일 플랫폼을 출시하게 된 것일까?

누구나 알다시피 파이어폭스는 인터넷 브라우저 제품이다. 사실 모질라재단은 제품을 만들기 위한 조직이 아니라 자유 소프트웨어 기반의 모질라 프로젝트를 지원하고 이끌기 위해 설립된 비영리 재단이다. 그렇다면 오픈소스 기반 비영리 단체에서 왜 브라우저를 만들게 되었는지부터 이해할 필요가 있다. 모질라재단 설립의 주축 인력들은 90년대말 인터넷 브라우저 제품으로 독점적인 위치를 가졌던 넷스케이프를 인수한 아메리카 온라인(AOL)에서 정리 해고된 사람들이다. 당시 넷스케이프와 마이크로소프트의 인터넷 익스플로러(IE)는 지금의 구글과 애플 같은 관계였다. 모질라 오픈소스에 기반을 둔 넷스케이프와 초기 상용제품이었던 '모자이크' 제품 기반으로 만들어진 IE 브라우저는 표준과 호환성과는 무관하게 독자적인 길을 걷게 되었다.

이렇게 상용화 과정에서 오픈소스의 정신이 왜곡되는 것을 목격한 모질라재단은 2003년부터 직접 파이어폭스라는 브라우저를 개발하게 되었다. 현재도 브라우저는 모질라 오픈소스인 게코(Gecko)와 사파리, 크롬에서 사용하고 있는 웹킷(Webkit) 오픈소스 그리고 IE에서 사용하고 있는 트라이던트(Trident)로 나뉘어져 있다. 브라우저들간의 호환성 문제와 표준화 문제는 십 수년이 지났지만 아직도 여전히 존재하고 있다. 최근에는 웹킷이 대세로 불리고 있지만, 웹킷 오픈소스도 예전과 달리 파편화가 많이 진행된 상태다.

이렇게 과거 브라우저 역사를 언급한 이유가 파이어폭스 OS가 나오게 된 배경과 일치하기 때문이다. 리눅스에 기반을 둔 안드로이드가 높은 시장 점유율을 가지고 있지만 모바일 파편화 문제를 일으키게 된 주범이기도 하다. 오픈소스에 기반을 둔 안드로이드 OS지만, 결국 개발자-소비자-생태계란 측면에서 구글의 권력하에 통제되고 있는 것이 현실이다. 이러한 비표준 상용화의 문제와 특정 기업에 의해 통제되는 기술에 대항하기 위해서 모질라재단에서는 파이어폭스를 상용화했던 이유와 마찬가지로 파이어폭스 OS를 상용화하기로 결정한 것이다.

그렇다면 파이어폭스 OS가 앞으로 얼마나 시장경쟁력을 가지게 될지 궁금하지 않을 수 없다. 분명히 브라우저와 모바일 플랫폼은 다른 영역이고, 특히 제조사와 마켓플레이스가 긴밀하게 연계된 독특한 생태계임을 감안하면 쉽게 성공적인 진입을 장담할 수 없다. 파이어폭스 OS 마케팅 전략을 보면 1차적으로 남미와 동구권을 타겟으로 하고 있다. 이 시장의 특징은 파이어폭스 브라우저의 시장점유율이 매우 높은 국가들이란 사실이다. 실제로 남미에서 파이어폭스 브라우저 점유율은 60%에 달하고 특정국가는 90%에 근접한다고 한다. 마치 전세계에서 유일하게 IE가 점유율이 90%가 넘었던 우리나라같이 매우 특수한 상황으로 보인다. 중요한 점은 이런 국가들에서 파이어폭스라는 브랜드는 마이크로소프트 만큼이나 신뢰할 수 있는 브랜드라는 사실이다. 이러한 브랜드 가치를 통해서 쉽게 모바일 플랫폼 시장에 진입이 가능할 것이라고 기대하고 있다.

특히 파이어폭스 OS 전략중에 하나는 100달러 수준의 저가형 단말기의 출시와 파이어폭스 스토어를 통해서 다양한 제품들을 선보일 예정이라고 한다. 안드로이드 종속화에 두려움을 느낀 많은 통신사들이 파이어폭스 OS 연합에 참여하고 있다. 현재 우리나라 KT를 포함한 18개 통신사가 참여하고 있으며, 제조사는 LG전자를 비롯하여 알카텔, 화웨이, ZTE 등이 참여하고 있다. 그리고 칩셋은 모두 퀄컴의 스냅드래곤을 사용한다.

분명 파이어폭스 OS의 등장은 신선한 바람이 될 수 있을 것이다. 어느때보다도 '안티 구글' 분위기가 높아진 상황에서 통신사, 제조사들이 파이어폭스 OS를 지원하기 위해서 연합군을 형성하는 과정은 지극히 시장 경쟁적인 흐름이라고 본다. 하지만 모바일 플랫폼은 단순히 하드웨어와 운영체제만으로 끝나지 않는다. 기존의 모질라재단이 이루어낸 업적들보다 더 힘든 것이 생태계를 플랫폼 안으로 끌어들이는 일이다. 물론 모질라재단도 이러한 문제를 모르지 않기 때문에 다양한 대안들을 만들기 위한 노력을 하고 있겠지만, 안드로이드만큼 단기간에 시장 점유율에서 두각을 나타내기에는 쉽지 않을 것으로 보인다.

우분투 터치 - "어떤 스마트폰에서도 동작한다"

정확한 표현은 우분투 포 폰(Ubuntu for phones)이라고 불러야 한다. 하지만 우분투 온 태블릿(Ubuntu on tablets)와 합쳐서 우분투 터치라는 말을 많이 사용하고 있기 때문에 본고에서는 우분투 터치라는 용어로 사용하고자 한다. 그런데 왜 갑자기 우분투가 모바일 플랫폼을 만들게 되었을까?

현존하는 모바일 운영체제 분포를 보면 애플의 iOS를 제외하면 안드로이드를 포함하여 대부분이 리눅스 커널을 기반으로 만들어져 있다. 따라서 리눅스 기반 모바일 운영체제가 사실상 모바일에 가장 최적화 되어 있고, 이미 안드로이드를 통해서 충분한 시행착오와 경험을 보유하고 있기 때문에 상당한 수준의 안정성을 확보하고 있다고 볼 수 있다. 앞서 설명한 파이어폭스 OS도 하부에 리눅스 커널을 사용하고 있음은 말할 것도 없다. 우분투라는 용어는 남아프리카 공화국의 건국 이념인 "다른 사람을 위한 인간애" 라는 뜻을 가진 단어에서 가져온 이름이다. 영국에 기반을 둔 캐노니컬(Canonical)이란 회사의 지원을 받고 있는 우분투재단에서 운영하고 있다. 알다시피 우분투는 리눅스 배포판을 상용화하여 사업을 영위하는 조직이며, 현재 가장 사용하기 쉬운 리눅스 버전을 배포하고 있다.

우분투 터치가 나오기 이전에 2008년에 인텔 아톰 프로세스에 기반한 "우분투 모바일"이 배포된 적이 있었지만, 현재는 개발이 중단된 상태다. 그리고 올해 다시 "우분투 포 폰"이라는 이름으로 MWC에 소개가 되면서 주목을 받기 시작하였다. 우분투 터치는 파이어폭스 OS와 제품 특성은 비슷하지만 사업의 방향은 좀 다르다. 우분투가 리눅스 배포판을 상용화하는 조직의 특성에서 파악이 되듯이 우분투 터치는 다양한 스마트폰에 모바일 플랫폼을 배포하는 것을 목적으로 하고 있다. 물론 통신사나 제조사가 우분투를 이용하여 자사 고유의 제품을 출시하는 것도 큰 문제는 없다. 이러한 특성들 때문에 우분투 터치는 다양한 스마트폰에 포팅되어 제품을 선보이고 있으며, 현재 가장 기본이 되고 있는 단말기는 구글 넥서스 시리즈들이고 그 외에 삼성 갤럭시 시리즈, LG 옵티머스, 모토롤라 등 이미 대중화된 스마트폰을 대상으로 포팅이 예정되어 있다. 그렇다면 이런 모바일 운영체제의 배포 사업이 모바일 플랫폼 시장상황에 어떤 영향을 미칠지 생각해 볼 필요가 있다.

우선 리눅스와 같이 우분투 터치가 쉽게 스마트폰에 배포될 수 있을지에 대한 의문이다. 현재 상당수의 우분투 사용자들이 1차 고객이 되겠지만, 현재 대상 단말의 대부분이 안드로이드 폰임을 감안하면, 상당한 얼리어답터 수준이 아니면 기존에 최적화된 안드로이드 폰을 우분투 터치로 바꿔서 사용할 일은 많아 보이지 않는다. 하지만 우분투에서 이렇게 다양한 스마트폰에 포팅을 하는 이유는 향후 해당 스마트폰 제조사들이 안심하고 기존 하드웨어 스펙에다가 우분투를 최적화하여 출시할 수 있도록 레퍼런스가 되고자하는 목적이 크다고 생각한다.

우분투 터치의 잘못 알려진 오해중에 하나는 안드로이드와 동일한 커널과 드라이버를 사용하고 있기 때문에 안드로이드 앱이 우분투 터치에서 호환이 될 수 있을 것이라는 기대다. 하지만 우분투에서는 안드로이드 앱을 구동시키기 위한 가상머신인 달빅(Dalvik)을 탑재하고 있지 않기 때문에 안드로이드 앱을 구동시킬 수는 없다. 하지만 다른 장점으로서는 가상머신이 없어서 보다 더 네이티브하게 앱들이 구동되기 때문에 성능이란 측면에서는 더 효과적이라고 볼 수 있다. 그래서 우분투를 위한 앱 개발은 아이폰, 안드로이드와 별개로 또 새로운 개발 환경을 익혀야 된다는 문제를 야기시킬 수도 있겠지만, 우분투 터치도 파이어폭스 OS와 마찬가지로 기본적으로 HTML5 기반의 웹앱에 주력하고 있기 때문에 네이티브 앱 개발에 대한 부담은 상대적으로 줄어든다고 볼 수 있다.

우분투의 모바일 카테고리에서 관심을 끌만한 제품은 "우분투 포 안드로이드" 기능이다. 여기에는 재밌는 기능이 포함되어 있는데, 평소에는 안드로이드 폰으로 사용하다가 폰을 모니터에 연결하면 우분투 운영체제로 전환되어 PC처럼 사용할 수 있게 해준다. 이러한 기능은 기존 우분투 사용자들에게 우분투 모바일 제품군으로 관심을 유도하고 있다.

하지만 분명한 사실은 아직 우분투 터치는 파이어폭스 OS에 비해서 공식 버전의 출시시점이 늦고 연합전선의 파워도 파이어폭스 OS보다는 약해보인다. 그럼에도 불구하고 우분투 터치에는 분명한 틈새시장이 보인다. 특히 우분투 온 태블릿의 경우 셋탑박스나 스마트TV, 카 네비게이션 같은 저사양 디바이스에서 효과적인 성능을

보여줄 수 있다. 즉 “Connected Things”형 디바이스에서는 안드로이드나 파이오폭스 OS보다 더 좋은 선택이 될 수 있다는 것이다. 예전에 마이크로소프트가 모바일 운영체제 전략을 스마트폰용 윈도우모바일과 산업용 단말을 지원하기 위한 윈도우CE로 구분하였듯이 우분투 터치는 비 스마트폰 계열 디바이스에서 더 관심을 가질만 하다.

타이젠폰 - “새로운 하이엔드 제품을 꿈꾸다”

삼성전자는 그동안 웨이브폰(바다OS)을 진행하면서 상당한 경험과 노하우를 가지게 되었다. 안드로이드 폰의 세계 최고 제조사임은 말할 것도 없다. 기존 바다OS가 파이어폭스 OS나 우분투 터치와 같이 저사양, 저가형 단말기를 목표로 하였다만, 타이젠폰은 하이엔드폰을 목표로 하고 있다. 그동안 경험에서 터득한 자신감의 표현으로 보인다.

타이젠 OS는 SKT, KT, NTT, 오렌지, 보다폰, 스프린트, 텔레포니카 등 전세계 다양한 통신사 및 화웨이, NEC, 파라소닉 등 중국-일본 아시아 중심 제조업체들과 연합하여 구글과 애플의 구속에서 벗어나기 위한 개방형 마켓을 지향하고 있다. 특히 다른 국가에 비해서 안드로이드 의존율이 높은 아시아권에서는 구글에 대한 견제로서 타이젠에 많은 관심을 가지고 있는 것이 사실이다.

분명 삼성-인텔이 주도하는 타이젠은 벤더 중심이라는 약점을 가지고 있지만, 안드로이드 사례에서 보았듯이 레퍼런스폰의 시장 반응이 무엇보다 중요한 시점에 삼성전자, 화웨이가 출시하게 될 하이엔드급 레퍼런스폰은 분명 의미있는 초기 시장 전략이 될 수 있을 것이다. 모바일 플랫폼 후발주자임에도 불구하고 하드웨어 최강자인 삼성전자가 쓸 수 있는 전략은 사실 뻔하지만 기대를 가지게 하는 부분이다. 운영체제와 디바이스 측면에서는 파이어폭스 OS나 우분투 터치보다 유리한 위치에 있는 것이 사실이지만, 플랫폼과 생태계란 측면에서 커다란 약점을 보이고있다. 비록 삼성앱스라는 마켓 플레이스가 존재하지만 그 수준은 아직 미약할 뿐이다. 하지만 반대로 삼성이 국내에서 소프트웨어 연합을 만든다면 상황은 또 달라질 수 있다. 카카오톡으로 시작된 모바일 게임 유통 채널은 삼성전자가 하고 싶었던 작은 생태계의 성공적인 사례라고 할 수 있다. 아마존 킨들 파이어가 초기 서비스 앱과 보유한 디지털 콘텐츠를 기반으로 자신들만의 생태계를 구축하듯이, 삼성전자가 카카오톡으로 시작된 게임 유통채널의

다변화를 주도한다면 초기 부족한 마켓 플레이스 기능을 활성화하는데 도움이 될 것으로 보인다.

신규 모바일 플랫폼 - "HTML5가 대안이다?"

지금까지 3종류의 모바일 플랫폼 전략에 대해서 분석해 보았다. 각기 다른 목적과 다른 전략으로 시장을 바라보고 있지만, 몇가지 중요한 공통점을 가지고 있다.

- 1) *HTML5 기반 웹앱 중심의 모바일 운영체제를 지향하고 있다.*
- 2) *리눅스 커널을 베이스로 하고 있다.*
- 3) *통신사 / 제조사 연합 세력을 확보하고 있다.*
- 4) *반 구글 세력의 움직임을 보인다.*
- 5) *생태계 조성을 목표로 하고 있다.*
- 6) *모바일 OS 파편화에 대한 개발 문제점을 해결하고자 한다.*

재미있는 사실은 이들이 의외로 HTML5를 전략의 중심으로 보고 있다는 점이다. 사실 이러한 시도는 생소한 것은 아니다. 2000년대 초반 웹기반 위젯이나 팜의 WebOS 그리고 최근에 WAC(Wholesale Application Community)에 이르기까지 웹기반으로 모바일 생태계를 구축하고자 하는 의지는 꾸준히 있어왔지만 대부분 실패하였다. 그럼에도 불구하고 이러한 노력이 지속되는 이유는 이 전략이 애플이나 구글에 비해 열약한 개발자 생태계를 보유한 조직에서는 초기 시장 진입에 유리한 유일한 전략이기 때문이다. 초기 자바라는 언어가 운영체제와 상관없이 동작하는 가상머신 기반의 시스템 비종속적인 특성으로 기존 C/C++을 시장에서 밀어낸 전략과도 크게 다르지 않다.

특히 개발자 분포도를 볼 때 웹개발자의 비중이 월등히 높은 것이 사실이다. 풍부한 웹개발자 자원을 활용할 수 있다는 측면에서 분명 웹기반 플랫폼이 갖는 잇점이 있을 것이다. 하지만 간과하는 부분 또한 적지 않다. 개발이란 측면에서 본다면 동일한 C언어를 사용하더라도 서버개발과 클라이언트 개발이 다르고, 각각이 전문적인 지식과

경험이 필요하다. 언어가 동일하다고 해서 갑자기 서버 개발자가 클라이언트 개발을 할 수는 없다. 예를 들어 JavaScript 개발에 경험이 많다고 누구나 Node.js로 서버 개발을 자유롭게 할 수는 없다는 것이다. 여기서 말하는 HTML5의 환상중에 하나가 실제로 HTML 경험자 누구나가 쉽게 모바일웹앱을 만들 수는 없다는 것이다. 또한 클라이언트의 특화된 기능을 제대로 사용하고자 한다면 JavaScript 지식이 필수적이며 Canvas에 대한 이해도 높아야 된다. 접근하기는 쉽지만 제대로 하기가 어려운 분야가 HTML5의 숨어있는 속성이다.

신생 모바일 플랫폼의 베이스는 리눅스 커널이어야 한다는 사실에 동의한다. 하지만 필자의 주관적인 생각으로는 아마존 킨들처럼 그동안 시장에서 검증된 안드로이드 오픈소스를 잘 활용하는 것도 방법이라 생각한다. 구글의 종속성을 벗어나면서도 운영체제의 안정성을 확보하고, 독자적인 생태계를 구축할 수 기반을 만들 수 있다고 생각한다.

생태계의 핵심이라고 할 수 있는 마켓 플레이스의 경우, 웹앱이 활성화된다면 WAC과 같은 통합 앱스토어로 연합하는 것이 필요하다. HTML5가 대안이라고 하지만 결국 크로스 플랫폼에 최적화된 언어는 현재로는 JavaScript 정도가 아닐까 생각한다. JavaScript에 최적화된 환경은 현재로서는 구글의 V8 엔진을 사용하는 것이 가장 효율적이지만, 구글의 종속성에서 벗어날 수 없다는 한계도 분명히 가지고 있다.

중요한 것은 어떤 디바이스와 어떤 운영체제가 아니라 누가 먼저 선순환적인 생태계를 조성할 수 있는가에 달려있다. 어떤 커널에 어떤 방식의 앱을 만들든 소비자와 생산자의 적극적인 참여를 만들어낼 수 있다면 성공의 가능성은 조금씩 높아질 것이다. 킨들 파이어의 사례처럼 페이스북, 카카오톡, NHN과 같이 서비스 종속적인 회원을 기반으로 하는 서비스 업체들은 한번쯤 고민해 볼만 하다. 하지만 이들이 아마존 만큼 가치있는 콘텐츠를 보유하고 있는지는 의문이다.

우분투 사례에서 보듯이 언젠가는 표준화된 더미 디바이스 시대가 올지도 모르겠다. 용산에서 PC를 조립해서 구매하고 원하는 운영체제를 설치해서 사용하듯이, 스마트폰도 운영체제가 없는 디바이스를 구매한 뒤에 사용자가 원하는 운영체제를 선택해서

사용하는 시대가 오지는 않을까?

[참고문헌]

- The Firefox OS platform by https://developer.mozilla.org/en-US/docs/Mozilla/Firefox_OS/Platform
- Ubuntu now fits your phone by <http://www.ubuntu.com/devices/phone>
- Smartphone OS World market shares Q2 2012 and 2011 : IDC
http://www.theregister.co.uk/2012/08/08/apple_google_dominant_q2_2012_smartphone_shipments/
- 게임 이론으로 살펴본 Web OS이 가능성 by 전종홍 <http://mobile2.tistory.com/1>
신생 모바일 플랫폼의 동향과 차세대 플랫폼의 비전 by 박민우
http://www.digieco.co.kr/KTFront/report/report_issue_trend_view.action?board_seq=6292&board_id=issue_trend