

# [컬럼] 초보자를 위한 임베디드 리눅스 학습 가이드

## 목차

1. [개요](#)
2. [저작권](#)
3. [문서](#)
4. [임베디드 리눅스를 공부하기 위하여 자신의 상태에 따라서 무엇을 해야 하는가?](#)
  - 4.1 [임베디드 리눅스를 공부하고자 하는분들이 가장 먼저 해야 할일](#)
  - 4.2 [리눅스를 설치 해 본적도 없는 초 울트라 초차](#)
  - 4.3 [리눅스 명령에 익숙하지 않다면](#)
  - 4.4 [리눅스 디렉토리 구조를 모른다면](#)
  - 4.5 [셸 스크립트를 모른다면](#)
  - 4.6 [유닉스 운영 체제를 모른다면](#)
  - 4.7 [C 언어를 모른다면](#)
  - 4.8 [리눅스에서 도는 어플리케이션을 작성한 적이 없다.](#)
  - 4.9 [make 사용법을 모른다면](#)
  - 4.10 [리눅스가 동작하기 위한 최소한의 기본 구성에 대해 아는 것이 없다.](#)
  - 4.11 [커널 컴파일을 해 본적이 없었다면.](#)
  - 4.12 [커널의 구조를 모른다면](#)
  - 4.13 [어셈블러를 모른다면](#)
5. [윗글에서 제안한 리눅스의 개발과 관련한 기본을 다진후 진행하는 방법](#)
  - 5.1 [디바이스 드라이버를 정복하라!!!](#)
    - 5.1.1 [어떤 책이 좋을까?](#)
    - 5.1.2 [디바이스 드라이버 프로그램을 직접 해본다.](#)
    - 5.1.3 [프린터 포트를 이용한 캐릭터 디바이스를 만든다.](#)
    - 5.1.4 [자신이 사용하는 커널 버전이외의 내용은 과감하게 삭제한다.](#)
    - 5.1.4 [네트워크용 디바이스 드라이버는 공부해라.](#)
    - 5.1.5 [기존에 디바이스 드라이버를 한번쯤 스스로 재 작성해 보라.](#)
  - 5.2 [커널을 이해하라....](#)
  - 5.3 [커널을 이해하기 위한 방법론 및 어디까지?](#)
6. [정말 임베디드 리눅스를 하려면](#)
  - 6.1 [평가 보드를 구매하라....](#)
  - 6.2 [평가 보드를 구매 했다면....](#)
  - 6.3 [MCU를 공부하라..](#)
  - 6.4 [평가 보드의 회로를 공부해라....](#)
  - 6.5 [평가 보드에 사용되는 디바이스를 공부해라....](#)
  - 6.6 [크로스 컴파일 환경을 구축하는 것을 공부하라....](#)
  - 6.7 [부트 로더를 공부하라.](#)
  - 6.8 [NFS 시스템에 대해서 공부하라....](#)
  - 6.9 [그외 나머지는 어떻게?](#)
    - 6.9.1 [MMU가 없다..](#)
    - 6.9.2 [루트 디스크 이미지와 램 디스크](#)
    - 6.9.3 [플래쉬 메모리에 대해서 공부 할것](#)
    - 6.9.4 [계속 늘어 놓으면 한도 끝도 없을 것 같으므로 ...](#)
7. [빨리 개발하는 방법....](#)
  - 7.1 [이미 개발된 평가 보드를 이용할 것....](#)
- 8.0 [내가 하고 싶은 말은 ...](#)

## 1. 개요

이 문서의 내용은 정말 초보자가 임베디드 리눅스를 어떻게 접근하면 좋은가에 대한 생각을 적어 놓은 것이다.

## 2. 저작권

이 문서는 GNU 라이선스를 따른다.  
단 최초 작성인은 유명창임을 표시하여야 한다.

### 3. 문서

버전 1.0 - 2001. 06. 11

버전 1.1 - 2001. 06. (html version)

이 문서에 불만이 있거나 다른 생각이 있다면 여기로 메일을 보내 줄것

[frog6502@hanmail.net](mailto:frog6502@hanmail.net)

## 4. 임베디드 리눅스를 공부하기 위하여 자신의 상태에 따라서 무엇을 해야 하는가?

### 4.1 임베디드리눅스를 공부하고자 하는분들이 가장 먼저 해야 할일

임베디드 리눅스를 해야겠다는 생각만 가지고 있다가 이제 시작해야지 라는 생각을 하셨다면 우선 숨을 크게 들이키시고 내뿜는 행동을 반복하신후 임베디드 리눅스를 단번에 점령하겠다는 생각을 버린다. ^^; 정말 임베디드 리눅스는 한번 들어가 보면 끝이 없어서 한두달 만에 끝장을 보겠다는 생각을 가지셨다면 당장 그만 돌것! 당장 그만 두지 않더라도 조만간 그만 두게 될것이다. 왜? 한두달안에는 끝나지 않기 때문이다. 혹시 시작하는 시점의 여러분의 실력이 리눅스의 커널을 이미 분석이 끝난 실력이라면... 더구나 여러 아키텍처에 포팅까지 한 실력이라면 정말 한두달 사이에 끝날수 있을 것이다. 굳이 임베디드 리눅스를 따로 할 필요가 없는 실력자이므로...

### 4.2 리눅스를 설치 해 본적도 없는 초 울트라 초차

당장 리눅스 배포판을 구한다. PC에 리눅스를 설치한다.

### 4.3 리눅스 명령에 익숙하지 않다면

우선 리눅스에 관련된 책을 한권 산다. 그리고 한번 쭉욱 읽어 본다. [kldp.org](http://kldp.org) 사이트에 들락 날락 거리면서 [man.kldp.org](http://man.kldp.org) 도 참고 하면서.... 단 주의 할것은 암기력에 자신 있는 분을 제외하고는 명령을 굳이 외우려고 하지 말것 자주 사용하는 명령은 손가락이 외운다. 굳이 사용도 하지 않는 명령을 외우려고 스트레스 쌓이지 말고... 더구나 리눅스 명령어는 정말 정말 너무 많다.

### 4.4 리눅스 디렉토리 구조를 모른다면

최상위 디렉토리를 외운다. 몇 개 안되므로... 그리고 서점에 간다... 리눅스 관련 책들을 보면서 디렉토리에 관련된 내용을 중점적으로 읽는다. 읽어야 하는 내용은 디렉토리 구현에 대한 것이 아니라.. 각 디렉토리를 어떻게 구분해 놓았는가를 설명한 내용들을 읽는다. 대부분의 책들이 조금씩 만 소개하므로 굳이 책들을 사려고 하지 말고 서점에서 여러권을 섭렵하는 것이 낫다. 그리고 이것저것 패키지들을 손수 설치해 본다. 그러면 저절로 깨닫게 될것이다.

### 4.5 셸 스크립트를 모른다면

리눅스를 하면서 셸 스크립트를 모르면 고생문이 흰하다 사실 내가 그렇다. 그러나 셸 스크립트를 아주 잘 할 필요는 없다. 셸 스크립트를 읽을 정도면 된다. 셸 스크립트를 작성할 필요가 있다면 스크립트에 대한 강좌를 인터넷에서 보고 하거나 다른 스크립트를 보면서 하면 되니까....

### 4.6 유닉스 운영 체제를 모른다면

유닉스 운영체제에 대한 최대한 쉽게 설명한 책을 골라서 그냥 소설 읽는 셈 치고 읽어 본다. (참고) 쉽게 설명한 책들의 특징은

1. 그림이 많다.
2. 글씨가 크다.
3. 가격이 싸다.
4. 많은 사람들이 본다.
5. 두께가 얇다.

### 4.7 C 언어를 모른다면

이건 사람 얼굴에 입이 없는 셈이다. 밥 어떻게 먹으려나... 곧 굶어 죽게 될것이다. 바로 C의 기본적인 설명에 대한 책을 독파한다. 이걸 정성 들여서 독파해야 한다. 임베디드 리눅스는 거의 C로 시작해서 C로 끝난다. 가끔 C++도 있

지만....

**4.8** 리눅스에서 도는 어플리케이션을 작성한 적이 없다.

<http://users.unitel.co.kr/~sangeun5/linux/lpg.html> 에 가본다. 정말 도움이 될것이다

**4.9** make 사용법을 모른다면

make 사용법을 아주 잘 알 필요는 없다 그러나 기본적인 사용법은 알고 있어야 한다. 아주 복잡한 사용법은 이미 기술된것을 이용하는 것으로 충분하다.

**4.10** 리눅스가 동작하기 위한 최소한의 기본 구성에 대해 아는 것이 없다.

부팅 디스크를 만들어 본다. 여기서 주의 할것! 대부분의 분들이 쉘프에 있는 부팅 디스크 만들기를 따라만 하시는데 이걸 별로 효과가 없다. 부팅 디스크를 만드는 목적은 리눅스가 동작하기 위한 최소한의 기본 구성을 알고자 하는데 목적이 있다. 또한 로더의 필요성 커널과 루트 이미지의 분리와 램디스크에서 동작하는 것에 대해서 또 커널이 동작한 후에 실행되는 최소한의 진행과정을 이해하기 위한 것이다. 이를 알고자 하는데 목적을 두어야지 만드는데에만 열중하면 안된다. 또한 다른 부팅 디스크 이미지를 이용하여 그 내용을 살펴보고 공부하여야 한다.

**4.11** 커널 컴파일을 해 본적이 없다면.

임베디드 리눅스는 커널을 가지고 놀아야 한다. 물론 안하고 하는 방법은 있다. 그러나 대부분의 임베디드 리눅스에서 커널 컴파일과 환경 설정은 거의 필수이다. 임베디드상에서 리눅스의 커널 환경을 설정하고 컴파일하는 방식과 우리가 일반적으로 다루는 PC의 커널을 다루는 방법은 거의 대부분이 같다. PC상에서 커널을 자유자재로 가지고 논다는 것은 "임베디드상에서도 자유롭다"는 것을 의미한다.

**4.12** 커널의 구조를 모른다면

Unix kernel 완전분석으로 가는길이라는 책을 한번 볼것

약간은 예전의 커널에 대한 이야기가 나오지만 커널이 어떻게 동작하는 지를 알수 있다.

**4.13** 어셈블러를 모른다면

임베디드 리눅스를 함에 있어서 어셈블러를 몰라도 가능한 영역은 있다. 그러나 많은 부분에서 부딪히게 된다. 고로 어셈블러는 필요 충분 조건이다. 어셈블러라는 것이 너무도 많은 종류가 존재하므로 모두 다 익히기는 어렵지만 자신이 사용하려는 MCU의 어셈블러는 익혀야 한다. 일단 PC를 이용한 임베디드에 대해서 익히려면 PC의 어셈블러는 알아야 한다. 자유 자재로 구사할 정도로 익힐 필요는 없다. 다른 쏘스의 명령을 이해하고 자신이 필요로 하는 기능을 추가하거나 수정 할수 있을 정도면 된다.

## 5. 윗글에서 제안한 리눅스의 개발과 관련한 기본을 다진후 진행하는 방법

임베디드 리눅스를 할때 많은 분들이 평가 보드를 이용한 학습을 준비한다. 그러나 이런 분들은 대부분 좌절한다. 왜? 기본기가 없는 분들이 평가 보드에 포팅하겠다고 덤비기 때문에 좌절하는 것이다. 포팅이 그리 쉬운 것은 아니다. 일단 PC에서 내공을 쌓은 후 접근하는 것이 수월하다. 임베디드 리눅스라고 PC의 리눅스와 다른 점은 거의 없다. 단지 아키텍처만이 다를 뿐이며 이 부분은 전체에 3%도 되지 않는다. PC에서 리눅스의 내부를 자유 자재로 드나드는 분이면 임베디드 리눅스에 접근은 너무도 쉽다. 사실 몇가지 사항을 제외하고는 임베디드 리눅스보다 PC 시스템이 더 복잡하다. 가격도 비교해보면 임베디드 장비가 더 싸다.. ^^; 자 이젠 PC에서 임베디드 리눅스를 하기 위한 접근 방법을 알아 보자. 그리고 공부하기 위하여 평가 보드를 구입하기에는 개인이 구입하는 가격이 너무 고가이다. 컴퓨터를 또하나 사는 셈인데...이게 말처럼 쉽겠는가? 고액 연봉자라면 모를까....

### 5.1 디바이스 드라이버를 정복하라!!!

임베디드 시스템을 개발하는 과정에서 하드웨어쪽을 제외한 나머지 소프트웨어 측면에서 개발 비중은 로더 및 하드웨어 테스트 프로그램 10% 커널 포팅 10% 디바이스 드라이버 40% 임베디드 어플리케이션 40% 이렇다. 물론 내 개인적인 생각이므로 상황에 따라 다르겠지만 .... 그만큼 디바이스 드라이버는 중요하다. 디바이스 드라이버를 개발한다고 평가보드를 구매 해야 하는가? 이에 대답은 "아니오" 이다. PC는 매우 복잡한 시스템이다. 임베디드에 사용되는 대부분의 기능이 PC에는 있다. 그러므로 PC에 도는 하드웨어에 관련된 디바이스 드라이버를 스스로 만들 능력이 된다면 임베디드용 디바이스 드라이버를 만들 능력은 충분해 진다. 일단 돈 안드는 PC부터 정복해라. PC가 없다면 모를까 ^^; 자 그럼 디바이스 드라이버는 어떻게 정복하는 것이 좋을까? 한번 알아보자..

#### 5.1.1 어떤 책이 좋을까?

디바이스 드라이버를 만드는 방법에 대한 책으로 대표적인것에는 오렐리의 책이 있다. 그러나 이책을 초보자가 보기에는 너무 어렵다. 그렇다고 마땅한 다른 책이 있는 것도 아니다. 고로 선택권이 없으므로 이 책을 여러번 읽어 본다. 완벽하게 이해하려고는 하지 말기를 부탁한다. 여러번 읽어 보면 어렵듯이 무슨 내용인지 이해하는 시점이 있을 것이다.

### 5.1.2 디바이스 드라이버 프로그램을 직접 해본다.

우선 오렐리의 책에 소개된 싸이트에 가서 소스를 가져 온다. 그리고 소스를 이해하려고 노력한다. 또한 KERP의 내가 쓰고 있는 디바이스 드라이버의 강좌를 계속 지켜 보거나 현재 진행되고 있는 박철님이 진행하고 있는 목요 세미나에 참가한다. 아니면 웹을 뒤지면 의외로 디바이스 드라이버 강좌를 하는 싸이트가 꽤 된다. 싸이트 이름을 내가 일일이 외우기는 힘들지만 ... ^^; ( 천재는 건망증이 심하다고들 한다. )

### 5.1.3 프린터 포트를 이용한 캐릭터 디바이스를 만든다.

대부분의 임베디드 장비에서 사용되는 디바이스는 캐릭터 디바이스로 충분하다. 고로 캐릭터 디바이스를 안다는 것은 임베디드용 디바이스 드라이버의 절반에 해당하는 내용을 안다는 의미도 된다.

### 5.1.4 자신이 사용하는 커널 버전이외의 내용은 과감하게 삭제한다.

현재 임베디드용 커널은 2.2대를 근거로 사용한다. 2.4대는 최근에 나왔으므로 아직은 적용중이다. 임베디드용 커널은 버전이 낮다. 또 2.0 대 이전 버전은 약간 적용 방식이 다른 것으로 기억한다. 이를 위해서 오렐리 책에서는 이런 저런 내용을 기술하고 있는데 실전에 사용되는 임베디드 리눅스는 사실 커널을 여러가지로 사용하겠끔 굳이 설계하지 않아도 된다. 즉 커널 버전을 덜 타는 것이다. 범용적인 PC용 디바이스 드라이버를 작성할때나 고려할 내용이다. 그러므로 여러분은 괜히 소스를 어지럽히는 커널의 버전 관리 에 대한 내용은 과감하게 제거하고 진행하는 것이 수명 연장에 지름길이 된다.

### 5.1.4 네트워크용 디바이스 드라이버는 공부해라.

임베디드 장비에 리눅스를 사용하는 가장 큰 목적 중에 하나가 네트워크 스택을 이용하기 위한 것이다. 리눅스 자체가 덩치가 크므로 조그마한 임베디드 장비에는 아예 탑재하지 않는다. 채산성이 전혀 맞지 않기 때문이다. 더구나 리눅스의 대부분의 어플리케이션이 네트워크에 연관되어 있다. 고로 네트워크 디바이스 드라이버는 공부해야 한다.

### 5.1.5 기존에 디바이스 드라이버를 한번쯤 스스로 재 작성해 보라.

계속 말하는 것이지만 PC는 매우 복잡한 시스템이다. 고로 PC의 디바이스 드라이버를 구현 할수 있다는 것은 임베디드용 디바이스 드라이버를 구현 할 수 있는 능력이 된다는 것과 동일한 내용이다. 여기서 주의 할것은 PC에서 구현된 디바이스 드라이버들은 매우 안정된 상태이다. 여러분이 이렇게 완벽하게 안정된 디바이스 드라이버를 만들 수는 없다. 왜? 초짜이므로.... 똑같이 안정된 디바이스 드라이버를 작성할수 있다면 바로 리눅스의 개발에 앞장 서 주시길 부탁한다. 보통 많은 분들이 PC용 디바이스 드라이버를 재 작성할때 시리얼 포트용 디바이스 드라이버 부터 하는데 제발 시리얼 포트 부터 하지 말것... PC의 리눅스 상에서 시리얼은 정말 복잡하다. 보통 임베디드의 시리얼과는 상대가 되지 않는다. 터미날을 지원하는 기능과 가상 터미날과 연계 되어 있기 때문이다. 고로 차라리 프린터 포트나 마우스 같은 디바이스 드라이버 부터 만들어 가라... 어쩌면 랜카드용 디바이스 드라이버가 더 쉬울지도 모른다. 다시 당부 드리지만 시리얼 포트 부터 접근하지 말것....

### 5.2 커널을 이해하라....

디바이스 드라이버를 공부하다보면 많은 부분들이 커널과 연계되어 있음을 알게 된다. 그러므로 디바이스 드라이버에 대해서 공부가 끝났다면 커널의 40%쯤은 이해 했다고 볼 수도 있겠다. 커널을 공부함에 있어서 구분해야 할것이 있다. 하나는 아키텍처 의존적인 부분과 아키텍처와 독립적인 부분이 있다. 리눅스는 가급적 아키텍처 독립적으로 구성 되려고 노력한 흔적이 여기저기 보인다. 그래서 리눅스가 다른 아키텍처에 포팅 될수 있었던 것이다. 만약 PC의 i386 계열에 의존적인 부분이 매우 많았다면 지금처럼 임베디드용으로 사용되지도 못했을 것이다. 자 커널을 이해하는 방법은 어떻게 좋을까? 어디까지 이해해야 할까? 이에 대한 나의 생각을 풀어 보겠다.

### 5.3 커널을 이해하기 위한 방법론 및 어디까지?

커널을 이해하기 위해서 가장 좋은 방법은 소스에서 흐름을 쫓아 보아 보는 것이다. 이를 위한 방법은 쉘프에서 제안한 숙제중 부팅 메시지를 시리얼로 보내는 방법을 제안한다. 물론 이때 "Uinx kernel 완전분석으로 가는길이라는 책"을 옆에 끼고 있는 것이 좋을 것이다. 물론 쉘프에 조형기님이 진행하고 있는 강좌란도 무척 도움이 될것이다. 너무 완벽하게 쫓아 가려고 하지 않는 것이 좋을 것이다. 이 놈의 리눅스 커널을 쫓아 가는 것에 맛이 들리면 도통 헤어나질 못하게 된다. 필요한 만큼만 쫓아 다니면 된다. 임베디드 리눅스를 구현함에 있어서 대부분은 기존에 이미 포팅되어 있는 아키텍처를 이용하게 되는데 이 방법이 의미하는 것은 커널을 완벽하게 이해하기 보다는 커널 소스의 어떤 부분에 어떤 기능이 있는가 정도를 파악하는 선이면 된다는 것이다. 필요할때 좀더 자세하게 소스를 쫓아 다니면 된다. 또 내가 권유하는 방식은 당신이 필요로 하는 기능은 대부분 다른 사람이 이미 해 놓았을 가능성이 높다는 것이다. 커널에 집중하기 보다는 웹싸이트를 뒤지는 시간이 더 짧을 수 있다. 요즘은 Know Where!! 아닌가....

## 6. 정말 임베디드 리눅스를 하려면

5항까지 진행된 분이라면 6항 이후에 설명하는 과정을 수행해도 문제가 없을 것이다. 여기서 부터는 보통 일반인들이 말하는 임베디드 시스템용 리눅스를 어떻게 하는가를 설명하는 것이다.

### 6.1 평가 보드를 구매하라...

만약 당신이 가난한 개발자라면 이 이후에 진행은 포기해 달라.. 왜? 개인이 하기에는 너무도 벅찬 돈이 든다. 그냥 평가 보드만 사서 될일이 아니다. (이것도 정말 비싸다) 물론 평가 보드만 사도 될수 도 있지만 새로운 하드웨어적인 기능을 추가하고 시험하려면 개발 장비가 보통 고가가 아니다. 그냥 개인이 취미로 하기에는 이미 금전적인 부담이 넘어선다. 단 평가보드상에서 만족한다면 평가보드 살 능력이 되어 한다. 평가보드만으로도 꽤 많은 것을 공부 할수 있다. 단 평가보드를 구매할때 JTAG가 지원되는 평가보드를 구매하라 그나마 JTAG를 사용하는 공부는 돈이 적게 든다.

### 6.2 평가 보드를 구매 했다면....

평가 보드를 보통 구매하면 대부분의 문서가 따라 오게 된다. 물론 사용되는 디바이스의 설명서들이 포함되 있거나 최소한 어디서 구하는지에 대한 설명은 있을 것이다. 이런 자료를 모두 일단 확보할 것

### 6.3 MCU를 공부하라..

평가 보드에 사용되고 있는 MCU에 대해서 공부해야 한다. MCU를 모르고 임베디드를 도전하는 것은 시간낭비의 지름길이다. MCU의 구조 MCU의 하드웨어 사양 MCU의 프로그램 방법 ( 어셈블러 ) MCU의 동작 방식 뭐 이런 것이 있겠다.

### 6.4 평가 보드의 회로를 공부해라...

평가 보드의 회로를 읽는 법을 알아야 한다. 회로를 읽지 못하고 임베디드를 공부하려 한다면 진행하는데 한계를 느낄수 있다. 회로를 읽을수 있어야 외부 연결 커넥터에 어떻게 연결하는지를 이해 할수 있다.

### 6.5 평가 보드에 사용되는 디바이스를 공부해라...

평가 보드에 사용되는 디바이스를 모두 공부할 필요는 없다. 자신이 사용하고자 하는 디바이스에 대해서 하드웨어적으로는 어떻게 연결되고 디바이스를 제어하기 위해서는 소프트웨어 적으로 어떻게 해야 하는 가를 이해 해야 한다. 이런 이해를 위해서는 해당 디바이스에 대한 리퍼런스 매뉴얼의 확보는 필수적이다. 항상 나에 가슴을 아프게 하는 것은 기술된 언어가 영어라는 점이다. 빨리 한글이 전세계 공통어로 되어야 하는데.... 나를 세계 대통령으로 밀어 주면 가장 먼저 한글을 공용어로 하고 모든 표기를 한글로 하는 세계법을 제정하겠다.

### 6.6 크로스 컴파일 환경을 구축하는 것을 공부하라...

임베디드 리눅스를 공부하는 것에 가장 큰 관문중에 하나가 크로스 컴파일 환경을 구축하는 것이다. 그나마 다행인것은 요즘은 아예 rpm 패키지로도 나온다는 점이다. 그래도 몇몇 cpu에 필요한 환경은 자신이 스스로 구축해야 한다. 또 평가 보드에 따라 구축하는 방법이 조금씩 다를 수 있다. 물론 평가보드에 따라오는 CD에는 일반적으로 크로스 컴파일에 필요한 화일도 따라오므로 좀 관찮기는 하지만 아예 공부하지 않으면 조금만 문제가 생겨도 속수 무책일수 있다.

### 6.7 부트 로더를 공부하라.

임베디드 리눅스를 공부하기 위해서 가장 처음 부딪히는 것이 부트 로더다. PC에서는 그 개념이 별로 중요하지 않으므로 이 문제가 별로 대두되지 않지만 임베디드에서는 바로 대두 된다. 보통은 이미 개발된 오픈된 부트 로더를 자신의 시스템에 맞게 바꾸게 된다. 그러므로 부트로더의 소스는 어느정도 분석하고 있어야 한다.

### 6.8 NFS 시스템에 대해서 공부하라...

임베디드 시스템에는 커다란 보조 기억 장치가 없기 때문에 필히 네트워크를 이용한 다른 시스템의 보조 기억 장치를 마운트 사용해야 편리하다. 이를 위해 사용되는 것이 NFS이다. 고로 이에 대한 공부는 해두는 것이 좋다.

### 6.9 그외 나머지는 어떻게?

커널을 올리고 바꾸고 하는 것은 PC에서 공부했다면 별로 문제가 되지 않는다. 또 대부분 커널은 포팅했다. 단 일반적으로 부딪히는 문제를 늘어 놓는다면 다음과 같다.

#### 6.9.1 MMU가 없다..

보통 PC에서는 MMU가 있으므로 문제가 되지 않지만 임베디드 시스템에서는 MMU가 없는 것이 일반적이다. 왜? 가격이 싸기 때문이다. 그렇다면 어떤 문제가 있을까 바로 이 문제에 대해서 공부해야 한다는 것이다. 50100용을 사용하거나 드래곤볼을 사용한 임베디드 리눅스의 하우투 문서를 찾아 보고서 공부하는 것이 좋다.

### 6.9.2 루트 디스크 이미지와 램 디스크

임베디드 리눅스의 거의 대부분이 보조 기억 장치가 없거나 있어도 플래쉬 메모리이다. 그래서 루트와 마운트 되는 기억 장치로 램 디스크를 많이 사용하게 된다. 문제는 램 디스크용 루트 이미지가 ROM에 있거나 플래쉬 메모리에 저장되기 때문이다. 커널의 루트 이미지 복사는 부분에 대한 공부를 필요로 하는 것이 이 때문이다.

### 6.9.3 플래쉬 메모리에 대해서 공부 할것

플래쉬 메모리는 하드 디스크 처럼 쓰고 읽을 수 있는 저장 장치이다. 더구나 읽는 속도가 매우 빠르다. 그러나 플래쉬 메모리는 쓰는 횟수에 한계가 있다. 이에 대한 공부를 미리 하지 않으면 당신이 납품한 임베디드 장비에 대해서 고생 할 가능성이 농후하다. 이에 대하여 다른 사람은 어떤 방법론이 있는가에 대하여 미리 공부해 두는 것도 좋을 것이다. 가끔 배신감 느끼게 플래쉬에 대한 몇가지 소프트웨어가 특허에 걸려 있다.

### 6.9.4 계속 늘어 놓으면 한도 끝도 없을 것 같으므로 ...

정말이다.. 임베디드 리눅스라는 분야는 PC이외에 모두 다 라고 할 정도로 다양하다. 이 모든 것을 미리 공부할수는 없다. 역시 실전에 부딪히면서 배우는 수밖에 ....

## 7. 빨리 개발하는 방법....

아마도 이 글을 읽고서 벌써 임베디드 리눅스를 공부하고 싶은 마음이 없어지는 사람들의 목소리가 들린다. 음.... 아마도 이글을 읽고 있는 분들 중에는 당장 1달안에 개발을 완료 해야 하는 분들도 있을 것이다. 이에 대한 해답은 아니지만 속성 코스에 대해서 이야기 하려고 한다. 단 이 방법은 엔지니어로써 어느정도 구력이 있는 분에 해당하는 것이지만 완전 엔지니어 초짜에 해당되는 것은 아님을 미리 밝힌다.

### 7.1 이미 개발된 평가 보드를 이용할 것...

자 임베디드 리눅스를 이용한 시스템을 만든다는 것은 자신이 목적으로 하는 동작이 정확하게 수행되면 된다. 그러므로 구현하고자 하는 장비와 최대한 같은 기능을 담고 있는 평가보드(개발보드)를 구한다. 이 평가 보드에는 리눅스가 모두 설치 되어 있어야 한다. 그리고 캐릭터 디바이스의 가장 최소한에 대한 공부만 한다. 이것마저도 안한다면 어쩔 수 없다. 그리고 평가보드의 회로를 최대한 그대로 수용한 목적 시스템을 만든다. 이런 상태라면 다음과 같은 할일만 남는다.

1. 부트로더에서 초기화와 커널 부팅을 수행하는 부분만 남겨 놓고 제거한다.
2. 커널을 그대로 사용한다.
3. 대부분의 디바이스 드라이버도 그냥 사용한다.
4. 추가된 디바이스의 드라이버만 작성한다.
5. 어플리케이션을 작성한다.

이것이 끝이다.

## 8. 내가 하고 싶은 말은 ...

내가 제안한 초짜가 임베디드 리눅스로 가는 길이 맞다고는 할수 없다.

아직도 나는 초짜이므로

하지만 기존에 개발하는 과정을 지켜본 나로써 내린 것은 위와 같은 것이다.

누군가 말했다.

임베디드 리눅스의 세계에 발을 들여 놓는 순간은  
태산을 삼질해서 없애는 것과 같다는 것을

하지만 내가 해본 결과로는

임베디드 리눅스를 한다는 것은  
리눅스의 본질을 파악하는 것과 거의 일맥 상통한다.

엔지니어라면 한번쯤 도전해서 이루고 싶은 영역이다.

또 생각보다 재미 있다.

당신도 해보라... 겁만 먹지 말고

천천히 가다보면

언젠가 이미 득도한 해커와  
같은 반석위에 있는 자신을 발견 할 수 있을 지도 모른다.

어느날 저녁 새벽 3시쯤 근방에 이글을 마무리 하며.....

PS: 배고프다.... 그리고 졸립다....

-----

