

# **TI DSP, MCU 및 Xilinx Zynq FPGA**

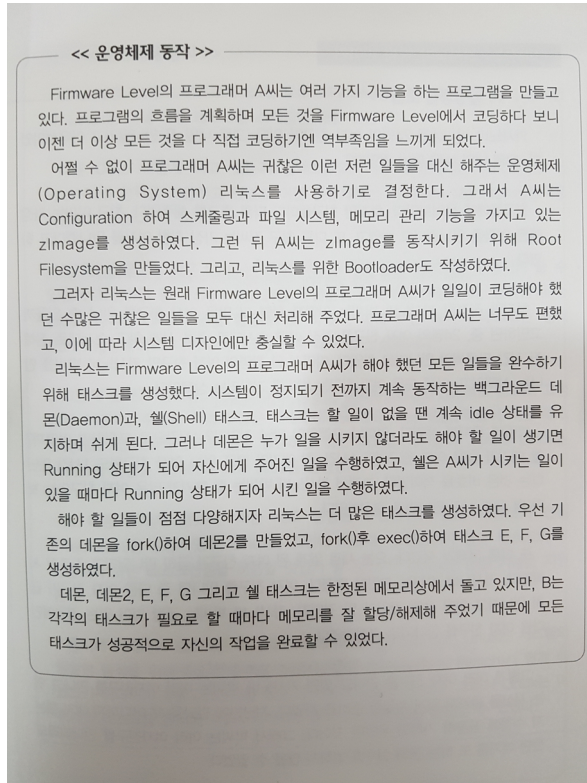
## **프로그래밍 전문가 과정**

**강사 - Innova Lee(이상훈)**  
**[gcccompil3r@gmail.com](mailto:gcccompil3r@gmail.com)**  
**학생 - 하성용**  
**[accept0108@naver.com](mailto:accept0108@naver.com)**

## 32 일차 리눅스 커널 내부 구조

### p10 운영체제는 자원관리자

### p17



Firmware 라는 기능을 만들고있었다는데  
Firmware 란?

운영체제가 없이 돌아가는 소프트웨어라는  
걸 의미

ex)avr 를 OS 없이 사용하면?

OS 없이 테스크 여러개(GPIO 등등) 사용하  
면 꼬임

이유는? 멀티태스킹이 안되서

한 저명한대학에서 발표한 논문이있는데  
컨텍스 스위칭과 멀티태스킹이 결국 무엇을  
기반으로 동작하는지? 스케줄링

스케줄링은 무엇을 의미?

이 프로세스에는 얼마나시간을 줄건지 저  
프로세스에는 얼마의 시간을 줄건지  
스케줄링은 타이밍값 계산을 다해줌  
결론은 OS 가 있어야한다

### 17

- 캐시가 적중한다 안한다뜻

「 CPU 」  
IC : 명령어캐시  
DC :Data

branch predictor : 예측기

아이캐시와 디캐시를 보고 사이즈를 보고 예측  
이 분계선은 절로 뭘거같은데 안뭘거같은데  
적중을 할것인지 미스가 날것인지  
적중을 하면 기록. 안하면 버림

arr[10]

arr[0]번에 먼저 적중하면 0 번에 기록  
캐시에서 바로 꺼내다쓰면 1clock 이 걸림

캐시라는게 없으면 컴퓨터성능이 17 배 떨어짐

적중 (hit)  
미스 (miss)

예측한 값이 캐시에 들어있다 = 적중  
우리가 예측한값이 안들어있다 = 미스

====

모든것을 ~역부족임을 느끼게 되었다 :  
운영체제없이 여러 작업(쓰레드등등) 을 하려면  
값이 다꼬인다

어쩔수없이~ 리눅스를 사용하기로 결정  
그래서 결론 : zimage(제트이미지)를 사용

즉 이사람이 어떤것에서 가장 힘든걸 느낀건 바로 스케줄링의 유무

-

콘솔에 OKOK 나오면서 나오는거는 초기화를 의미  
값을 초기화안하면 동작이 안됨

-

부트로더가 하는일은 초기화해주는게 그일

부트로더는 가상메모리 레이아웃도 잡아줘야함

mmu 없으면 가상메모리 안쓰면? 물리메모리 직접매핑하게됨

루트파일시스템은 ?????만 사용할수있음  
ex)pwd /→ /→

운영체제가 관리하는 5 대요소  
파일시스템  
메모리  
네트워크  
디바이스  
프로세스(task)  
이것까지 관리하는 이미지가 제트이미지에 들어가있다는것

수많은 귀찮은 일들을 모두대신~처리해주었다

-

시스템 디자인에만 충실  
: 하드웨어 디자인에만 집중한다는거

커널개발하는사람들이 만들어준다는얘기

시스템디자인이라는게 어떤 a 라는신호가 들어왔을때 이걸 어떤 신호로 구동하게 할것인가  
이런걸 시스템디자인이라함  
input 이들어오면 output 을 어떻게 할것인가

모두 대신처리 해주었다 → 디바이스 드라이버가 일다해줌  
디바이스 드라이버는 뭘해줌? 어떤장치가 켜지면 어떤장치가 무엇인지 알고  
그것에 맞는 드라이버를 구동

일들을 완수하기 위해 테스트를 생성  
데몬 쉘~~

ps -ef |grep 하면 세션부분이 물음표로 나옴 → 데몬  
컴퓨터전원을 누르기전까지 애는? 안죽음

데몬은 누가 일을 시키지않아도~  
: 시그널이 들어오면 애를 깨울수있다  
리얼타임시그널은 배우지않았지만 특정상황에만 동작하게 할수도있음

해야할일이 더 다양해지자 리눅스는 더 많은 테스트를 생성하였다  
~테스크 E,F,G 를 생성하였다

: 그냥 fork 했을때 어떤일이벌어짐?  
exec 이후에 썼던내용이 증발함  
대처법 : exec 의 자식이 하게함

p19

여기서 중요한건 24 개의 태스크를 어떻게 잘관리할것이지? 스케줄링

FIFO 를 쓰는건 절대 RTOS 는 아님  
상대편한테 미사일이 나오고있음  
더 중요한건 통신? 레이다? → 레이다지만  
통신이 먼저라면 통신하고 레이다차레왔을때 죽음

즉 실시간기능에서 L 쓰면안됨

레이트모노틱정책을 사용 : 비율  
우선순위높은녀석에게 할당량을 더줌  
실시간계열에선 RM 정책을 주는게 좋음

효율성이 높은~~ 태스크를 수행 ~~ time 슬라이스만큼만 자원할당 :  
리눅스에서 레이다시스템에돌 들어가야하니까 리얼타임 리눅스도있고  
일반적으로 사용하는 범용적인 리눅스가 있음  
여기서 얘기하는 타임슬라이스만큼의 자원할당은 범용적인 상황을 말함  
범용에서 리눅스를 하이브리드 상황을 채택함  
하이브리드란? 리눅스에도 우선순위가 정해진게 있고 변하는애가있는데  
우선순위가 변한다는건 유저에서 만든 태스크가 있다고했을대  
유저가 막그냥 때려넣고있음. 작업량이 겁나 많은애가있음 (ex. 스타,.룰,프로게이머)  
이걸 다처리하려면?  
게임=사용자태스크  
다른태스크들한테 액가 밀리지않으려면  
입력이 엄청많네? 애우선순위를 높여줘야겠네

즉 리눅스는 상황에따라 동적으로 변할수도있고 고정으로될수도있다

단하나의 태스크만 러닝상태이고 나머지 태스크는:  
싱글코어 cpu o에서만 통용되는 사례

타임슬라이스 어떻게 결정 : 우선순위를 보겠다는거

-

또 다른 문제가 생겼다 :  
프렘션(선점)  
우선순위가 높은 태스크의 우선순위가 낮은 태스크선점을 허용할것인가 말것인가?

누구를 만나기위해 갔다면 선점을했다는소리

중요한 작업이 있다면 타임슬라이스가 완료가지않았음에도? 바꾼다는소리

..

동시에 여러개의 태스크를 하나의 CPU  
컨텍스트스위칭~~

: 즉 멀티태스킹을 구현했단소리

p20

a 태스크는 B 태스크에게 전달해줄 사건이 발생 :  
시그널을 얘기하는중

시그널을 받은 태스크는? 기본적으로 종료  
시그널 핸들러를 등록 : 시그인트 시그핸들러 동작하게

왜 보냈는지 궁금할수있다  
~번호를 저장한뒤 그 번호를 보내주기로 하였다  
어보트 : 6 번  
시그인트 : 컨트롤 c  
정말 죽어야한다 : 9 번  
시그널번호 총 64 개였음

같은 시그널이 또 배달되지않도록 :  
시그애틀등등 → 이곳에 시그널이 들어오는게 맞는지아닌지 확인하는것  
시그널을 처리하는도중에 시그널이 또들어오면

ex)키보드 → io  
io 는 느림 컨텍스트스위칭이 더빠르다보니까 ctrl+c 누르는정도논 처리되지만  
시그널을 막주고받는상태가되면 시그널핸들러를 처리하는도중에 시그널이 들어오면 시그널이 처리가 안됨

시그널블록 : 사용자가 직접 코드짜서(해당번호 막겠다)

마지막부분 - 시그널이 빠르지만 대량의 정보를 주고받을수없다 :  
시그널은? 번호만 보낼수있다  
구현하기위해서 집어넣은게 쉘어드메모리,등등,  
소켓은 무엇? 유일하게 원격으로 ipc 와 소통할수있는거

클라이언트에 소켓이 생성되었다는거자체가 나랑얘가 동기화됐다 는것

p21

배우지않았기때문에 적을수없었을것  
tcp/ip 에서 서버에서 클라이언트와 동기화를 하게되는데  
동기화를 어떤식으로 하게되는거?  
3-way 핸드셰이킹  
3 번왔다갔다하면서 핸드셰이킹을 한다는것

3 번째줄에 3 번째문단 → syn(야 있냐)을 보냈고  
받고 ack(ㅇ 있어) + syn(너도있냐) 을 보내고  
받고 ack(ㅇ 있어) 보냄

통신활성화됨

FIN 은 종료할때 나오는 통신

원격은 무조건 이렇게 통신하는것

p22

커널 ~~파일 읽어오라고 명령 :  
디스크찾는거~~?

파일시스템 a.txt 찾기 시작 :  
먼저 a.txt 를 찾기위해선 최상위 디렉토리에서 검색하는게 빠른데  
최상위 디렉토리에서 검색하기위해 최상위디렉토리를 찾음

---/--/super-block 이 있음

파일시스템을 지정하게되면 우리가 생각해야하는거  
파일언더바 오퍼레이션  
여기엔 무엇이들어있음? 리드 오픈 클로즈  
여기엔 무엇? 함수포인터가 들어있음  
함수포인터가 들어있는이유는? ??가 제각기 다르기때문에

## 챕터 1

p24

386 보호모드는 인텔계열만있음

중요한건 운영체제를 만드려면 페이징은 할수있어야함  
MMU 를 사용하지못하게되니까 하드웨어적인 리소스를 사용불가  
코어텍스 에이나 운영체제를 만드려면 페이징까지는 구현이되야 동작가능

오픈소스는 책임이있어야하고 프리소스는 다공개

gpl 라이선스

돈받고 팔아도되지만 코드는 공개해야함

라이선스 선불리 기입하면안됨

p29

리눅스장점

## 챕터 2

그림 2.1 중요한 구조

모든게 테스크\_struct 로 관리됨

유저가 커널에게 요청할수있는 방법은 시스템콜밖에없음

운영체제가 고나리하는 5 대요소

파일시스템 매니지먼트

메모리매니지먼트

테스크 매니지먼트

디바이스매니지먼트

네트워크매니지먼트

디렘 인식하려면 디바이스 드라이버가 필요함

프로그램이 올라가기전에 디스크인데

디스크가 어디있는지 관리하는게 파일시스템 매니지먼트

원격은 네트워크 매니지먼트

프로그램이 메모리에 올라가야 구동이되는데

네트워크에서 구동이되는건 어디서=메모리 매니지먼트

프로그램이 올라가려면 태스크 매니지먼트가 필요하고

네트워크통신하는게

이렇게 서로 물려있음

밑에보면 하드웨어 인터페이스가 보이는데 하.인이 어디에 묶여있냐면

네트워크랑 디바이스매니지먼트에 묶여져있음

이유는 네트워크라 네트워크는 하드웨어이기때문에 묶여있음

실제 물리장치 dev1..~~..

p.35

page fault

3-5

파일시스템이 파일에 생성과 접근제어

슈퍼블록관리 (루트파일 위치를 알아오는)

네트워크 사용하는이유는 TCP/IP

네트워크를 아무리 작게만들어도 Ip와 맥주소, 여차하면 포트번호도필요함

디바이스드라이버는 실제장치를 구동시키는거

펌웨어냐 디바이스드라이버냐 차이는 OS 없이 구동시키냐 OS있이 구동시키냐 차이

실제로 하는일은 펌웨어와 같음

4-1

운영체제는 무엇을 위하여 어떻게 자원을 관리

파일을 열거나 호출

지금 니가열려는 파일의 정보를 알려주는게 커널

우리가 하는일은 요청만하는것

이 복잡한걸 해주느게 커널

system call

p36

커널의 소스코드

~/kernel/linux-4.4\$ ls arch

```
yong@yong-Z20NH-AS51B5U:~/kernel/linux-4.4$ ls arch
alpha  avr32    frv      Kconfig  microblaze  openrisc  score  um
arc     blackfin h8300    m32r     mips        parisc    sh      unicore32
arm     c6x      hexagon  m68k     mn10300     powerpc   sparc   x86
arm64   cris     ia64     metag    nios2       s390      tile    xtensa
```

blackfin : 오디오 dsp

openrisc:리스크프로세스를 한번 오픈플랫폼으로 만들어보자해서 fpga로 만든 오픈소스의 리스크프로세스

x86 : 인텔 cpu

c6x : ti사의 dsp

ia64 : 멸망당한 vi의 어찌구.. 이유 : 8년전에 병렬처리쓰기시작할때 병렬이뜨다보니까 모든명령어를 심드로 만들어보자 했다가 성능안나와서 망한 아키텍처  
그런데 망한덕에 그래픽카드가 엄청 발전하게됨

microblaze : fpga에서 구동되는 가상의 cpu

spark :슈퍼컴퓨터용 cpu

arm : 임베디드 디바이스용(ex.폰)

powerpc : 주로 자동차에 들어감

arm64 : 암도 64비트가 나왔다는소리

avr32 : 우리도 먹고살아보자 해서 커널을 건드려봄

nios2 : 알테라 fpga = 알테라는 인텔에 인수당함, 즉 인텔에서 밀고있는 아키텍처

c6x  
arm  
arm64  
를 주로 다루게될것

~/kernel/linux-4.4\$ ls arch/arm

```
yong@yong-Z20NH-AS51BSU:~/kernel/linux-4.4$ ls arch/arm
boot      mach-davinci  mach-lpc32xx  mach-rpc      mach-zynq
common    mach-digicolor  mach-mediatek  mach-s3c24xx  Makefile
configs    mach-dove      mach-meson     mach-s3c64xx  mm
crypto     mach-ebsa110   mach-mmp       mach-s5pv210  net
firmware   mach-efm32     mach-moxart    mach-sa1100   nwfpe
include    mach-ep93xx    mach-mv78xx0   mach-shmobile  oprofile
Kconfig    mach-exynos    mach-mvebu     mach-socfpga   plat-iop
Kconfig.debug  mach-footbridge  mach-mxs      mach-spear     plat-omap
Kconfig-nommu  mach-gemini     mach-netx      mach-sti        plat-orion
kernel      mach-highbank  mach-nomadik   mach-stm32     plat-pxa
kvm         mach-hisi      mach-nspire    mach-sunxi     plat-samsung
lib         mach-imx       mach-omap1     mach-tegra     plat-versatile
mach-alpine  mach-integrator  mach-omap2     mach-u300      probes
mach-asm9260  mach-iop13xx    mach-orion5x   mach-uniphier  tools
mach-at91     mach-iop32x     mach-picoxcell  mach-ux500     vdso
mach-axxia    mach-iop33x     mach-prima2    mach-versatile  vfp
mach-bcm      mach-ixp4xx     mach-pxa       mach-vexpress  xen
mach-berlin   mach-keystone   mach-qcom      mach-vt8500
mach-clps711x  mach-ks8695     mach-realview  mach-w90x900
mach-cns3xxx  mach-lpc18xx    mach-rockchip  mach-zx
```

많은데 arm 은 하위호환을 하지않음  
arm 이라는 아키텍처안에 엄청 많음  
mach-exynos : 삼성  
omap1 : ti 꺼  
omap2 : ti 꺼  
zynq : 자이닉스  
s3c24xx : 삼성  
s3c64xx : 삼성  
keystone: dsp  
plat-samsung : 삼성 플랫폼  
plat-omap : ti  
이렇게 센곳은 커널에 자기회사명을 올려놓음  
lpc18xx : nxp 즉 차량용  
lpc32xx : 이하동문  
bcm : 라즈베리파이  
davinci : ti(블랙박스용,스마트 tv,cctv 즉 비디오시장은 다빈치가 다잡음)

stm32 :  
tegra : nvidia 에서 사용하는 리눅스 플랫폼

-

cpu 모델따라 바뀌어야하는 코드가 kernel/linux-4.4/arch 에 들어가있음

커널은 순수한 소프트웨어만 관리  
아키는 하드웨어적인 부분을 관리  
컨텍스트위칭코드  
부트로더  
페이징을 하는 코드가 아키에 mm  
페이징에 소프트웨어적인 부분은 mm 에 구현되었음

아키는 cpu 에 종료불로 cpu 에 분할된다

lib 는 ==  
tools 에는 커널에서아용할수있는 간단간단한 코드들



block 디렉토리에는  
파일에종류 머머잇었음 .블록파일 캐릭터파일 파이프 일반파일디렉토리 소켓  
블록디바이스드라이버와 관련된 내용이 여기있음  
왜 디바이스드라이버만? 블록이 크기가 크기때문

ssd 유명한게 무엇? 삼성  
ssd 가 블록을 통해 지원함  
블록장치들을 관리하기위해 알고리즘이 필요한데  
이 알고리즘을 관리하는게 블록디렉토리

init 에는 머가 있을까  
init 뜻은 초기화  
커널의 시작코드는 init 에 있음

vi -t start\_kernel 하면

9 번에 있음

스타트 언더바 커널  
스타트 언더바 커널에 들어가면 하드웨어 초기화  
rest\_init 까지 하게되면 로그인창이 뜨는거

부트로더의 분석은  
멀티프로세서를 지원하는 커널이 어떻게 동작하는지 알게됨

sample 에는 말그대로 sample 로 사용할수있는 커널이있음

certs 에는 암호화 알고리즘

documentation : 커널개발에 관련한 많은문서가있음

ipc : ipc 에 관련된게 저기다있음

scripts : 커널을 구동시키는데 필요한 스크립트들이 여깁쓰음

virt : 운영체제에 의해 운영체제를 올르수있게해줌  
운영체제를 프로세스처럼 구동시키게 해줄수있다는뜻  
네스케이팅기법? 이라는것도있음  
젠이나 클라우드서비스  
즉 클라우드서비스를 하기위한 서비스가 들어잇음

drivers : 디바이스 드라이버에관한게 엄청나게많음

ls drivers/

pci :

amba

macintosh

fmc : 초고속 무선통신

gpio

gpu:그래픽카드

power:전원장비

block

tty

bus

cdrom

char

usb

i2c

mtd

regulator:정전압

video

iio : 이거가지고 오실로스코프를 만들수있음

오실로스코프에 올라가는 운영체제는 리눅스

dma : 메모리에 직접 액세스하는것

xen :

wl : 무선과관련된것

phy(피직스) : 무선?

Net:

ls drivers/net

```
yong@yong-Z20NH-AS51B5U:~/kernel/linux-4.4$ ls drivers/net
appletalk  ethernet  ipvlan    mii.c      slip        vrf.c
arcnet     fddi      irda      netconsole.c  Space.c    vxlan.c
bonding    fjes      Kconfig   nlmon.c     sungem_phy.c  wan
caif       geneve.c  LICENSE.SRC  ntb_netdev.c  team        wimax
can        hamradio  loopback.c  phy          tun.c        wireless
cris       hippi     macvlan.c   plip         usb          xen-netback
dsa        hyperv    macvtap.c   ppp          veth.c       xen-netfront.c
dummy.c    ieee802154  Makefile    rionet.c     virtio_net.c
eql.c      ifb.c     mdio.c      sb1000.c     vmxnet3
```

wireless

```
yong@yong-Z20NH-AS51B5U:~/kernel/linux-4.4$ ls drivers/net/wireless
adm8211.c  atmel_cs.c  iwlegacy  mwifiex    rndis_wlan.c
adm8211.h  atmel.h     iwlwifi   mwl8k.c    rsi
airo.c     atmel_pci.c Kconfig   orinoco     rt2x00
airo_cs.c  b43         libertas  p54        ti
airo.h     b43legacy  libertas_tf  prism54    wl3501_cs.c
at76c50x-usb.c  brcm80211  mac80211_hwsim.c  ray_cs.c  wl3501.h
at76c50x-usb.h  cw1200     mac80211_hwsim.h  ray_cs.h  zd1201.c
ath         hostap     Makefile    rayctl.h  zd1201.h
atmel.c     ipw2x00    mediatek   realtek    zd1211rw
```

brcm

iwlwifi

ti

-

ls

firmware //

net //네트워크와 관련된 모든정보

sound //사운드 디바이스 드라이버, ALSA 드라이버만 잘해도 먹고사는게 지장없음

video 의 → V4L2 이것도잘해도 먹고사는데 지장없음 //영상스리밍 그것의 핵심

crypto : 암호학, 암호화시키는 알고리즘, 복호화 복호화시키는 알고리즘들이 있음

fs : 파일 시스템, 파일에 관련된 작업, open read close

kernal:커널에 관련된 기능들이 구현-- 스케줄링

~39 페이지까지의 설명

44

챕터 3

제어권을 가지게되고 구동을 하면서 작업을 하게됨

즉 태스크의 생명사이클

변화를 준다는건 이그젝을 통해(즉 사건을 통해 인간말종이 사람이되는) 변화를 보는걸 보실수있다

생명을 준다는건 곧 피스나 포크를 준다는거고 변화를 준다는건 이그젝을 한다는거

컨텍스트스위칭=문맥교환

주로 영어로 써져있으니 영어로 기억하게하는게 좋음

50

1. 프로세스와 쓰레드 그리고 태스크  
프로세스와 쓰레드 두개를 구분짓는 핵심적인 요소가있는데  
kernel  
vi -t task\_struct  
로들어가서 144  
1480

pid\_t pid 가 이재용;  
pid\_t tgid 가 삼성  
쓰레드의 리딩을 하는사람  
tgid 와 pid 가 같으면

완수하기위해서  
이런것들을 모아서 쓰레드삼성  
쓰레드삼성의 리더는 쓰레드이재용이되는거

tgid 는 임베디드과정  
pid 는 강사님, 학생들이됨  
tgid 의 대표격이되는게 강사

같다는게 리더프로세스  
쓰레드 리더가 즉 프로세스랑 같다는거  
쓰레드 리더가 아니면 쓰레드(그냥 멤버 구성멤버)

fork 해서 만든건 pid 값과 tgid 와 같은 즉 프로세스

50

태스크란 무엇일까?  
pid 와 tgid 가 같으면 프로세스  
pid 가 다르면 같은 그룹에 속해있는 쓰레드들중에 하나가됨  
fork 와 같이쓰면 프로세스  
그럼에도 불구하고 pid 값은 언제나 같을수없음  
피아이드와 티취아이드가 각각 100 200  
그다음 200 100 100 200  
프로세스도 리더쓰레드로 봐야함  
ex)1 인창업

file a.out  
윈도우 파일포맷 PE  
실행파일은 ELF  
안되면 디버그  
디버깅 파일포맷 DWARF 드워프?  
readlf -h a.out

이것을통해 어떤머신을통해 동작하는지알게됨

51

사용자입장에서 프로세스 구조  
가상메모리를 얘기하는데 가사

vi -t task\_struct

pgd\_t pgd //주석 달아놓을것

mm\_struct

start\_code, end\_code,

// 텍스트영역의 시작과끝  
~의 시작과끝  
스택의 시작과끝

가상메모리는 8 바이트

51~52 그림

사진을 옆에 추가하는걸 빠뜨렸습니다 죄송합니다