시스템 콜 추가

========================

1. 커널에 시스템 콜 추가하기

2. C 라이브러리 헤더에 시스템 콜 번호 추가하기

3. 시스템콜 사용 app 작성

4. 커널 재빌드 & 재시작

1. 커널에 시스템 콜 추가하기

1) 사용자 헤더에 추가할 시스템콜 번호 추가

arch/arm/include/uapi/asm/unistd.h

#define \_\_NR\_hello (\_\_NR\_SYSCALL\_BASE+391)

2) 시스템 콜 개수 추가

/usr/src/linux/arch/arm/include/asm/unistd.h

#define \_\_NR\_syscalls (392) //1) 추가에 따른 조정 -> 4의 배수로 추가하기

3) 시스템 콜 SWI 처리 코드에 추가

/usr/src/linux/arch/arm/kernel/calls.S

CALL(sys\_hello)

4) 커널 top include syscalls.h 파일에 시스템 콜 프로토타입 추가

/usr/src/linux/include/linux/syscalls.h

asmlinkage long sys\_hello(int a, int b, int \*res);

5) 커널 소스에 시스템 콜 구현코드 추가

/usr/src/linux/drivers/char/hello.c

#include <linux/kernel.h>

#include <asm/uaccess.h>

asmlinkage long sys\_hello(int a, int b, int \*res)

{

int err=0, ret=0;

printk(KERN\_INFO "Hello SysCall in Kernel!!!\n");

err = access\_ok(VERIFY\_WRITE, res, sizeof(int));

if(err == -EFAULT) return err;

ret = a + b;

copy\_to\_user(res, &ret, sizeof(int));

//put\_user(ret, res);

printk(KERN\_INFO "%d + %d = %d from Kernel!!\n", a, b, ret);

return 0;

}

6) hello.c 컴파일을 위한 Makefile 수정

/usr/src/linux/drivers/char/Makefile

obj-y += hello.o

2 C 라이브러리 헤더에 시스템 콜 번호 추가하기

/usr/include/arm-linux-gnueabihf/asm/unistd.h

#define \_\_NR\_hello (\_\_NR\_SYSCALL\_BASE+391)

3 시스템콜 사용 app 작성

hello\_syscall\_test.c

#include <stdio.h>

#include <stdlib.h>

#include <linux/unistd.h>

int hello\_test(int a, int b, int \*res)

{

return syscall(\_\_NR\_hello, a, b, res);

}

int main(int argc, char\*\*argv)

{

int ret = 0, err=0;

err = hello\_test(3, 5, &ret);

if(!err) printf("hello\_result : %d\n", ret);

else printf("System Call Error!!\n");

return ret;

}

4. 커널 재빌드 & 재시작

# make -j4 zImage (약 30분 소요됨)

# mv /boot/kernel7.img /boot/kernel7.img.pre

# scripts/mkknlimg arch/arm/boot/zImage /boot/kernel7.img

# reboot