

# Embedded System Software [CSE4116]

## 실습 3 주차 : System Call

Department of Computer Science and Engineering, Sogang University, Seoul, South Korea

Data-Intensive and Computing and System Laboratory

### 1. Adding System Call

보드 커널 소스 폴더 : /work/achroimx\_kernel

#### 1.1. Allocate a system call number

/work/achroimx\_kernel/arch/arm/include/asm/unistd.h 파일 수정

```
403 #define __NR_sendmsg      (__NR_SYSCALL_BASE+374)
404 #define __NR_setns        (__NR_SYSCALL_BASE+375)
405 //new syscall
406 #define __NR_newcall      (__NR_SYSCALL_BASE+376)
407 #define __NR_newcall2     (__NR_SYSCALL_BASE+377)
```

#### 1.2. Register the system call table

##### A. Add a new service routine

/work/achroimx\_kernel/arch/arm/kernel/calls.S 파일 수정

```
387 /* 375 */ CALL(sys_setns)
388 //new syscall
389 CALL(sys_newcall)
390 CALL(sys_newcall2)
```

##### B. Expose the prototype of new system call

/work/achroimx\_kernel/include/linux/syscalls.h 파일 수정

```
65 struct file_handle;
66 //new syscall
67 struct mystruct;
68
```

```
851 asmlinkage long sys_setns(int fd, int nstype);
852 //new syscall
853 asmlinkage int sys_newcall(int a);
854 asmlinkage int sys_newcall2(struct mystruct *dd);
855
856 #endif
```

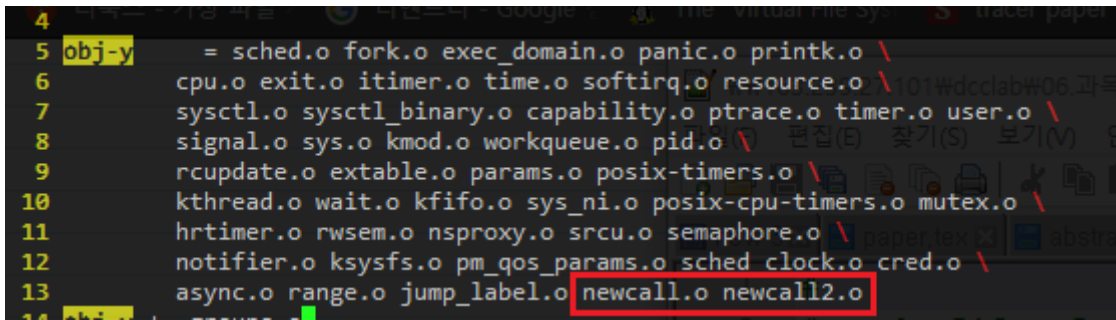
#### 1.3. Make out new system call

/work/achroimx\_kernel/kernel에 제공한 system call 파일 추가

- 함수는 매우 간단한 코드이니 읽고 내용 숙지, print 값과 return 값 확인
- newcall.c / newcall2.c: system call 함수
- test.c: 새로 생성한 두 system call 함수를 테스트하는 user 프로그램

## 1.4.Recompile the kernel

- Make 시 추가한 함수들을 포함할 수 있도록 Makefile (/work/achroimx\_kernel/kernel/Makefile) 수정



```

4
5 obj-y = sched.o fork.o exec_domain.o panic.o printk.o \
6       cpu.o exit.o itimer.o time.o softirq.o resource.o \
7       sysctl.o sysctl_binary.o capability.o ptrace.o timer.o user.o \
8       signal.o sys.o kmod.o workqueue.o pid.o \
9       rcupdate.o extable.o params.o posix-timers.o \
10      kthread.o wait.o kfifo.o sys_ni.o posix-cpu-timers.o mutex.o \
11      hrtimer.o rwsem.o nsproxy.o srcu.o semaphore.o \
12      notifier.o ksysfs.o pm_qos_params.o sched_clock.o cred.o \
13      async.o range.o jump_label.o newcall.o newcall2.o

```

- Make (make clean 하지 말것)

```

$ cd /work/achroimx_kernel
$ make achroimx_defconfig
$ make -j [number of cores]

```

- 커널 이미지 생성 및 전송

### 1. Host-side

```

$ cd /work/android
$ ./make_bootimg.sh

```

### 2. Device-side

u-boot 진입 후

```
$ fastboot
```

### 3. Host-side

```

$ fastboot erase boot
$ fastboot flash boot boot.img
$ fastboot reboot

```

## 1.5.Make an application which uses the new system call

- 다운로드받은 예제 test.c 파일을 크로스컴파일 하여 보드로 전송 및 실행
- 로그 레벨을 변경하여 printk 출력을 확인할 것

```
$ echo "7 6 1 7" > /proc/sys/kernel/printk
```

- test 프로그램은 linux/unistd.h 와 sys/syscall.h 헤더파일을 반드시 포함해야함

## 2. 실습 과제 : Generate new system call

- 새로운 system call (new call3) 작성
  - 함수는 int 형 변수를 두 개 포함하는 struct 를 parameter 로 사용한다.
  - Struct 안의 두 변수 값을 출력, 또 둘을 더한 값을 출력
- 새로운 system call (new call3)를 호출하는 user program 작성
  - Struct 변수들에 본인의 학번 앞 4 자리와 나머지 수를 각각 넣는다.
  - 새로 생성한 system call 을 위 struct 변수를 parameter 로 사용하여 호출한다.
- 제출물
  - 보드에서 새로운 system call (new call3)이 실행되는 프로그램 작동을 수행한 화면(minicom)을 캡처
  - System call 함수가 작성된 c 파일
  - 새로운 system call 함수를 test 하는 user program 이 작성된 c 파일
  - 위 파일들을 tar 로 압축하여 학번\_이름.tar 로 제출 (ex: 120221234\_홍길동.tar)