Embedded System Software

SystemCall

Dept. of Computer Science and Engineering
Sogang University, Seoul, KOREA

실습 시간 확인받을 것

- **Generate new system call**
  - 사칙연산(+, -, *, /)을 실행하는 새로운 **system call (newcall3)** 제작

Adding System Call

- Allocate a system call number
- Register the system call table
- Make out new system call
- Recompile the kernel
- Make an application which uses the new system call

Adding System Call

- **커널소스폴더 : /work/achroimx_kernel**
- **Allocate a new system call number**
  - **add new system call number (번호는 연속적으로 할당할 것)**
    - 커널소스폴더/arch/arm/include/asm/unistd.h

```
403 #define __NR_sendmmsg           (__NR_SYSCALL_BASE+374)
404 #define __NR_setns          (__NR_SYSCALL_BASE+375)
405 ///////new systemcall//////////////////////////////
406 #define __NR_newcall            (__NR_SYSCALL_BASE+376)
407 #define __NR_newcall2           (__NR_SYSCALL_BASE+377)
408
```

- **Register into system call table**
  - **Add a new service routine**
    - 커널소스폴더/arch/arm/kernel/calls.S

```
387 /* 375 */   CALL(sys_setns)
388 ////////////////new systemcall////////////
389             CALL(sys_newcall)
390             CALL(sys_newcall2)
391
```

  - **Expose the prototype of new system call**
    - 커널소스폴더/include/linux/syscalls.h

```
65 struct file_handle;
66 //////new_systemcall
67 struct mystruct;
68
```

Adding System Call

- **Register into system call table**
  - **Expose the prototype of new system call**
    - 커널소스폴더/include/linux/syscalls.h
    - 만약 새로운 struct를 parameter로 쓴다면 다음과 같이 추가

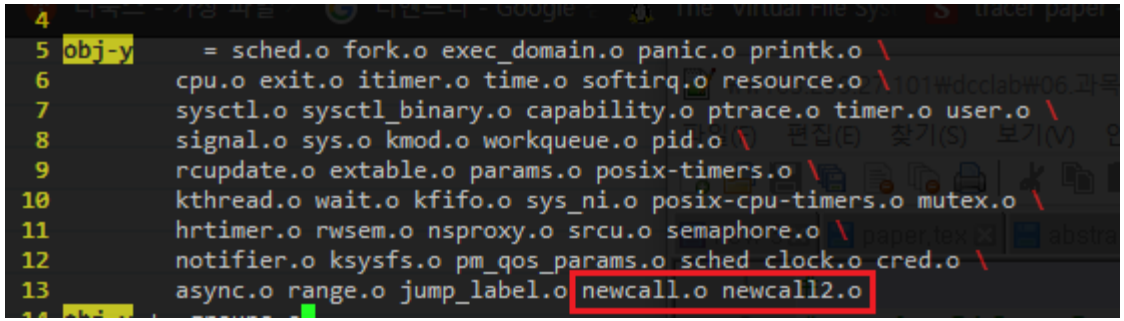
```
65 struct file_handle;
66 //////new_systemcall
67 struct mystruct;
68
```

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

Adding System Call

- **Make out new system call**
  - 커널소스폴더/kernel/
  - #include <linux/kernel.h>

- **Recompile the kernel**
  - 커널소스폴더/kernel/Makefile



Kernel Compile

- **make clean 하지 마세요**
- **source /root/.bashrc**
- **make achroimx_defconfig**
- **make -j[# of core] - ex) make -j4**

- **cd /work/android**
- **./make_bootimage.sh**
- **u-boot모드 진입 (device)**
- **fastboot (device)**
- **fastboot erase boot**
- **fastboot flash boot boot.img**
- **fastboot reboot**

Using the New System Call

- #include <linux/unistd.h>
- #include <sys/syscall.h>

- syscall(systemcall#, argument1, argument2, ...) 을 사용하는 Application 제작

- Tip : printk 출력 확인
  - echo "7 6 1 7" > /proc/sys/kernel/printk

copy_from_user(), copy_to_user()

- **copy_from_user()**

기능	사용자 메모리 블록 데이터를 커널 메모리 블록 데이터에 써넣는다.
형태	#include <asm/uaccess.h> int copy_from_user(void* to, const void __user* from, unsigned long n)
설명	from이 가리키는 주소의 사용자 메모리 블록 데이터를 to가 가리키는 커널 메모리 블록 데이터에 바이트 크기 단위인 n 만큼 써넣는다. 이 함수는 읽어올 공간의 유효성 검사를 수행한다.
매개변수	• from: 사용자 메모리 블록 선두 주소 • to: 커널 메모리 블록 선두 주소 • n: 써넣을 바이트 단위의 크기
반환값	정상적으로 수행이 되었다면 0 이상의 값, 그렇지 않다면 0보다 작은 값을 반환한다.

- **copy_to_user()**

기능	커널 메모리 블록 데이터를 사용자 메모리 블록 데이터에 써넣는다.
형태	#include <asm/uaccess.h> int copy_to_user(void __user* to, const void* from, unsigned long n)
설명	from이 가리키는 주소의 커널 메모리 블록 데이터를 to가 가리키는 사용자 메모리 블록 데이터에 바이트 크기 단위인 n 만큼 써넣는다. 이 함수는 써넣은 공간의 유효성 검사를 수행한다.
매개변수	• from: 커널 메모리 블록 선두 주소 • to: 사용자 메모리 블록 선두 주소 • n: 써넣을 바이트 단위의 크기
반환값	정상적으로 수행이 되었다면 0 이상의 값, 그렇지 않다면 0보다 작은 값을 반환한다.