

리눅스 시스템 프로그래밍

day4 참고 자료



고강태

010-8269-3535

james@thinkbee.kr



<https://www.linkedin.com/in/thinkbeekr/>

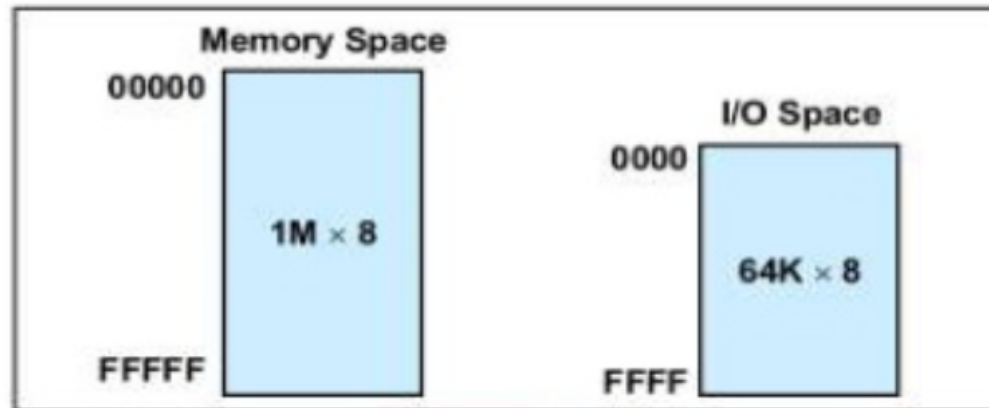


<http://www.facebook.com/gangtai.goh>

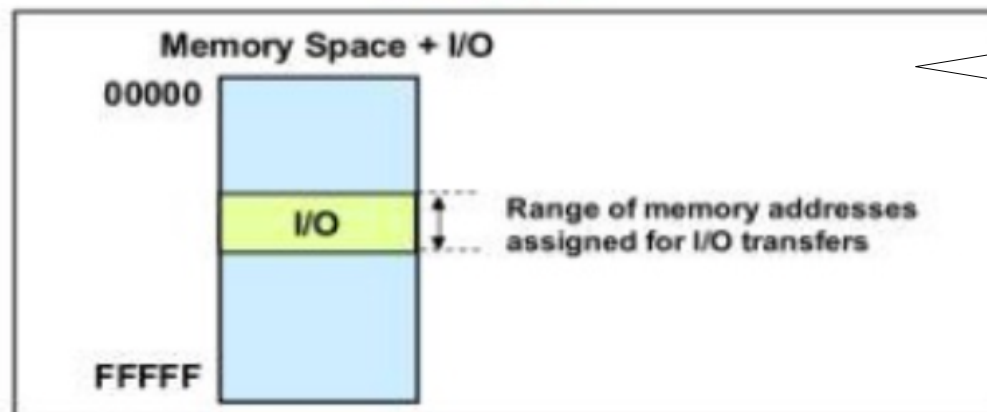
Memory Mapped

Memory Mapped I/O

Isolated vs. Memory Mapped I/O



Isolated I/O



Memory-Mapped I/O

MMR
은 volatile 사용

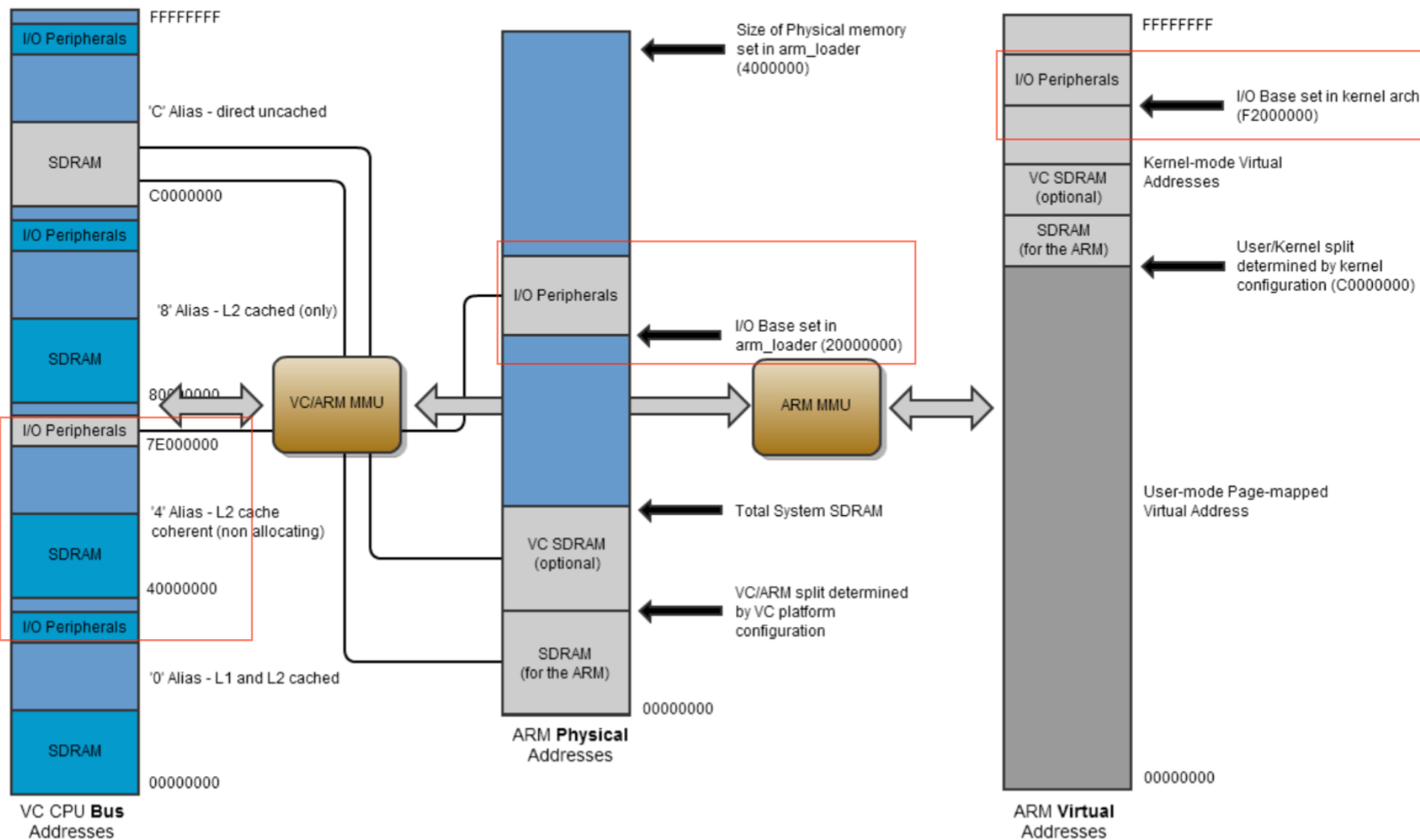
Memory Mapped I/O Programming

```
volatile unsigned long *LED = (volatile unsigned*)0xF0001000;

void Led_On(void)
{
    unsigned long led_status;
    *LED = 0x100;           // 0xF0001000이 저장된다
    led_status = *LED;      // 0xF0001000의 값을 led_status로 읽는다.
}
```



BCM2835 ARM Peripherals



mmap

물리 메모리의 주소를 입력하면 가상화 메모리 주소로 변환시켜주는 함수 'mmap'을 사용해야 한다.

```
$man mmap
```

NAME

```
mmap, munmap - map or unmap files or devices into memory
```

SYNOPSIS

```
#include <sys/mman.h>
```

```
void *mmap(void *addr, size_t length, int prot, int flags,  
           int fd, off_t offset);  
int munmap(void *addr, size_t length);
```

- addr: 가상주소의 시작 번지, 보통 0으로 시작.
- length: 매핑할 크기
- prot: 매핑 영역의 퍼미션, PROT_READ, PROT_WRITE...
- flags: 매핑 영역의 접근 허용 여부, 허용시 MAP_SHARED 반대는 MAP_PRIVATE
- fd: 메모리 매핑 장치의 파일 디스크립터, '/dev/mem'
- offset: 매핑 물리주소