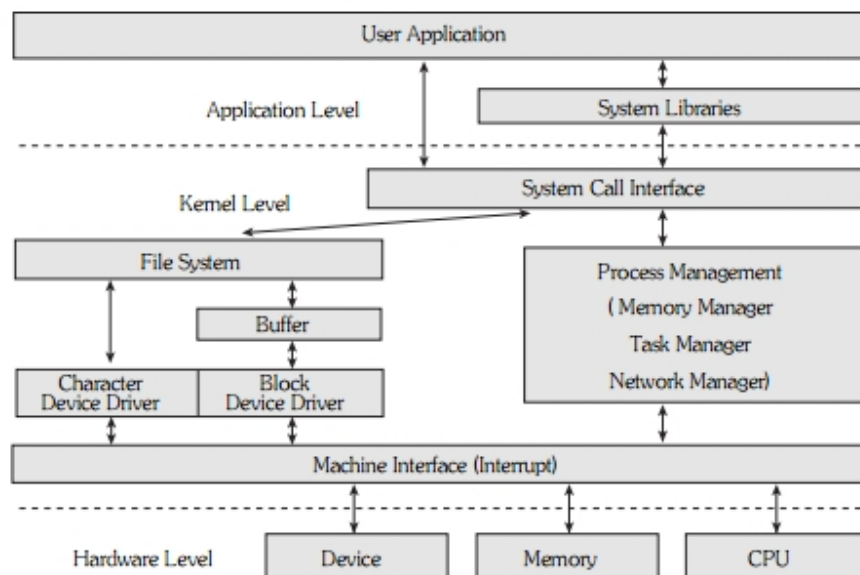


Chapter 04. Note

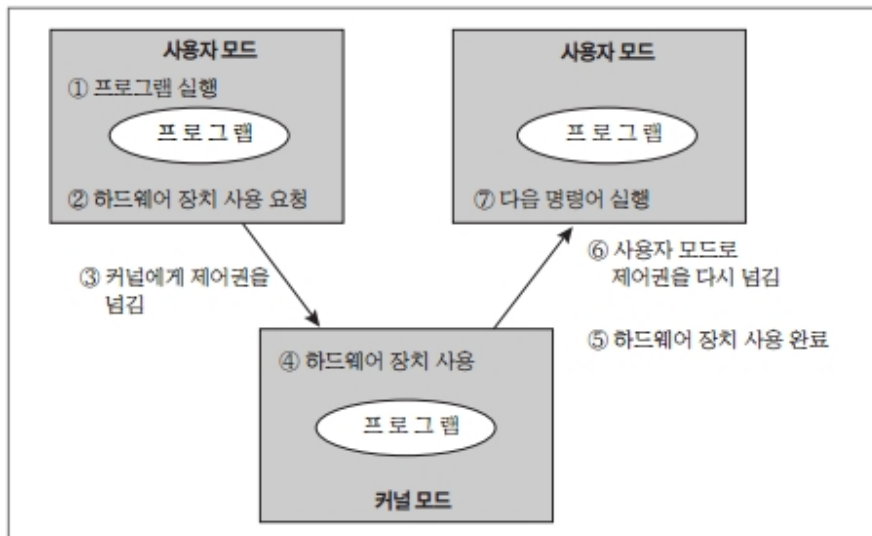
4.1 리눅스의 기본 구조와 파일 시스템

4.1.1 리눅스 시스템의 구조

리눅스의 주요 기능은 프로세스, 메모리, 파일, 입출력 등의 자원들을 관리하는 커널을 통해 제공된다. 커널은 시스템과 사용자 영역의 프로세스가 상호작용할 수 있는 인터페이스를 제공하여, 애플리케이션들이 컴퓨터 하드웨어나 시스템을 다룰 수 있는 보다 효율적인 애플리케이션 프로그래밍 환경을 지원한다.



리눅스는 커널 모드와 유저 모드로 동작하는데, 일반적으로 유저 모드에서는 시스템의 메모리에 접근하거나 하드웨어 같은 중요 장치를 사용할 수 없다. 이러한 디바이스의 사용이 필요한 경우에만 커널 모드로 동작하도록 해서 시스템을 보호하고 있다.



Reference

<https://kjvvv.kr/1766>

4.1.2 리눅스 파일 시스템

운영체제에서는 커널 프로그램과 유틸리티 등 다양한 시스템의 실행과 관련된 시스템 파일들을 제공한다. 그리고 사용자 데이터의 저장을 위해서도 파일을 사용하는데, 이러한 파일들은 파일 시스템으로 관리된다.

윈도우에서는 FAT32, NTFS 같은 파일 시스템을 사용하고, 리눅스에서는 EXT2, 3, 4 등의 파일 시스템을 사용한다.

마운트(mount) 뜻, 관련 명령어 & 문제 (fdisk, df, mkfs, 디스크마운트)

리눅스 완전 정복 : 리눅스 목차 아마 마운트라는게 윈도우 운영체제에서는 우리가 실감할 일이 없기 때문에 리눅스를 처음 접하시는 분들은 헛갈릴 수 있는 개념이에요. 오늘은 마운트에 대해서 끄적여보도록 하..

🔗 <https://jhnyang.tistory.com/12>

```
mount
type sysfs (rw,nosuid,nodev,noexec,relatime)
proc (rw,nosuid,nodev,noexec,relatime)
devtmpfs (rw,nosuid,nodev,noexec,relatime,size=978388k,nr_inodes=244747,mode=755)
type devpts (rw,nosuid,noexec,relatime,gid=5,mode=660,ptmxmode=0)
type tmpfs (rw,nosuid,noexec,relatime,size=201728k,mode=755)
type ext4 (rw,relatime,data=ordered)
fs/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
type tmpfs (rw,nosuid,nodev)
type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k)
cgroup type tmpfs (ro,nosuid,nodev,noexec,mode=755)
/cgroup/unified type cgroup2 (rw,nosuid,nodev,noexec,relatime)
/cgroup/systemd type cgroup (rw,nosuid,nodev,noexec,relatime,xattr)
cpustore type cpustore (rw,nosuid,nodev,noexec,relatime)
/cgroup/rdma type cgroup (rw,nosuid,nodev,noexec,relatime,rdma)
/cgroup/devices type cgroup (rw,nosuid,nodev,noexec,relatime,devices)
/cgroup/hugetlb type cgroup (rw,nosuid,nodev,noexec,relatime,hugepages)
/cgroup/net_cls,net_prio type cgroup (rw,nosuid,nodev,noexec,relatime,net_cls,net_prio)
/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,memor
```

4.2 파일 처리와 표준 입출력

4.2.1 저수준 파일 입출력

함수	내용	비고
open()	파일을 읽거나 쓰기 위해 열거나 생성한다	
creat()	비어 있는 파일을 생성한다.	open() 함수로 대체 가능
close()	열려 있는 파일을 닫는다.	
read()	열려 있는 파일로부터 데이터를 읽어온다.	
write()	열려 있는 파일에 데이터를 저장한다.	
lseek()	파일 포인터를 특정 위치로 이동한다.	
unlink()	파일을 삭제한다.	
remove()	파일이나 디렉터리를 삭제한다.	
fcntl()	파일과 관련되어 있는 속성을 설정하거나 조정한다.	ioctl()
dup()	파일 디스크립터를 복사한다.	dup2()

- open() 열기 뿐만 아니라 생성 가능
- creat() 는 open(path, O_WRONLY | O_CREAT | O_TRUNC, mode) 기능 수행
- close() - 프로세스가 종료되는 경우에는 운영체제에 의해 자동으로 닫힘
- read() 데이터를 읽은 만큼 파일 내의 오프셋이 증가
- write() 데이터를 쓴 만큼 파일 내의 오프셋이 증가
- lseek() 읽고 쓰기 위한 원하는 위치로 파일 내의 오프셋 이동

i2c-dev 디바이스 드라이버를 이용한 쓰기 읽기의 조합

KS24C080의 경우에는 이와 같은 방식으로 한번은 쓰기 명령 형식을 이용하여 write 함수로 읽을 주소를 지정하고 read 함수를 이용하여 읽기를 처리할 수 있다. 이 경우에는 START와 STOP이 반복적으로 사용된다. 하지만 경우에 따라서는 START ... START ... STOP 형식으로만 사용해야 하는 디바이스가 있다. 이런 경우에는 단순히 write read 함수의 조합만으로는 구현이 불가능하다. KS24C080의 경우에도 이와 같은 형식으로 데이터를 읽을 수 있는데 이런 경우의 데이터 형식을 보면 <그림 11>과 같다.

출처:

<https://sisyphus13.tistory.com/entry/linux-i2c-driver>

[시지프스...영원한 줄다리기...:티스토리]