

14주차 1차시 디바이스 드라이버 개념

【학습목표】

1. 디바이스 드라이버의 개념을 설명할 수 있다.
2. 디바이스 드라이버의 종류, 파일등을 설명할 수 있다.

학습내용1 : 디바이스 드라이버 종류, 파일

1. 개념

* 디바이스 : 컴퓨터시스템의 주변 하드웨어

- 하드디스크, 플로피디스크, 프린터, 단말기, 스캐너, 네트워크 어댑터, 터치 스크린, LCD디스플레이, 오디오 등등
- 디바이스 드라이버는 디바이스 구동 프로그램

* 디바이스 시스템 구성 : device + controller

- device : controller를 통해 입력된 명령을 수행하고 결과를 알려주는 역할
- controller : 디바이스와 디바이스 드라이버 사이의 인터페이스 담당

구성은 버퍼 및 제어/상태 레지스터 등

2. 디바이스의 종류

* 종류

- 문자 디바이스(character device)
- 블록 디바이스(block device)
- 네트워크 디바이스(network device)

* 디바이스 체계

- 주번호(major number)와 부번호(minor number)로 구분
- 동일한 디바이스는 동일한 주번호를 가지며 서로 구분하기 위해 부번호를 사용
- 구별하기 위해 시작문자를 통일
 - IDE 하드디스크 : hd
 - 터미널 : tty
 - 이더넷 : eth

* 문자 디바이스

- 파일시스템에서 하나의 노드 형태로 존재
- 자료의 순차성을 지닌 하드웨어
- 버퍼 캐시를 사용하지 않음.
- 데이터를 문자 단위 또는 연속적 바이트 흐름으로 전달하고 읽음.

* 터미널, 콘솔, 키보드, 사운드카드, 스캐너, 프린터, 직렬/병렬 포트, 마우스, 조이스틱 등

* 블록 디바이스

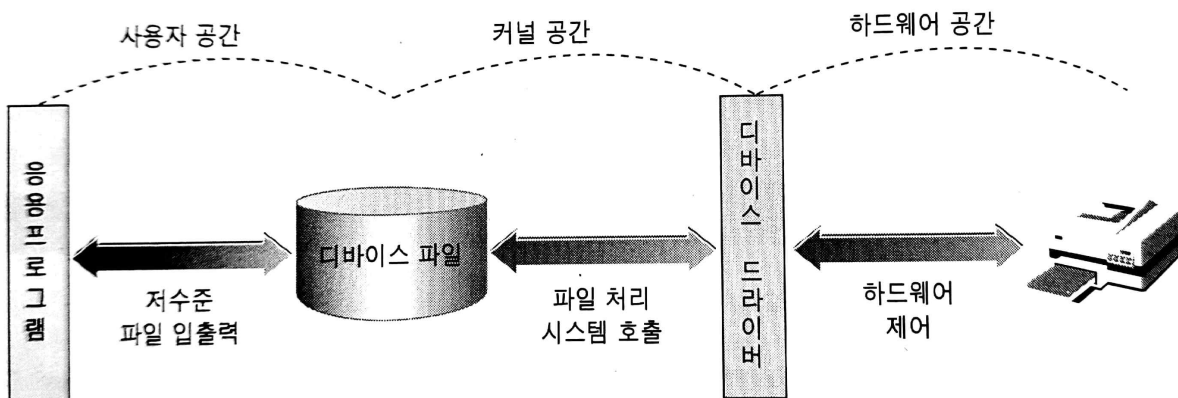
- 파일시스템에서 하나의 노드 형태로 존재
- 버퍼 캐시를 통해 임의 접근이 가능한
- 데이터를 블록 단위로 입출력
- 효율을 향상하기 위하여 버퍼를 이용
- 블록의 크기를 관리하고 블록 데이터의 전달을 담당하는 기능이 있음
- 하드디스크, 플로피 디스크, 램디스크, 테이프, CD-ROM, DVD 등

* 네트워크 디바이스

- 패킷 단위로 접근
- 네트워크 통신을 통해 네트워크 패킷을 송수신할 수 있는 디바이스
- 파일시스템의 노트 형태가 아닌 특별한 인터페이스를 사용
- 이더넷, PPP, SLIP, ATM, 네트워크 인터페이스 카드

* 디바이스 파일

- 리눅스 시스템은 디바이스를 /dev 디렉토리에 있는 일종의 파일로 취급. 즉, 디바이스를 파일로 추상화
- 디바이스 파일은 사용자에게 보이는 디바이스 드라이버의 인터페이스 부분
- 응용 프로그램은 디바이스에 접근하기 위하여 열기, 읽기, 쓰기 등과 같은 파일 연산을 이용
- 디바이스 파일은 파일시스템에서 고유한 번호와 이름을 할당받음
- 디바이스 파일은 파일시스템의 데이터 영역을 차지하지 않고 단지 디바이스 드라이버를 접근할 수 있는 관문 역할을 수행



학습내용2 : 디바이스 파일 생성

1. 특징

- 호스트 시스템의 디바이스 파일이 있는 /dev 디렉토리 내용관찰
- 디바이스 파일의 생성

```
mknod <디바이스 파일명> <디바이스 파일 형식> <주번호> <부번호>
```

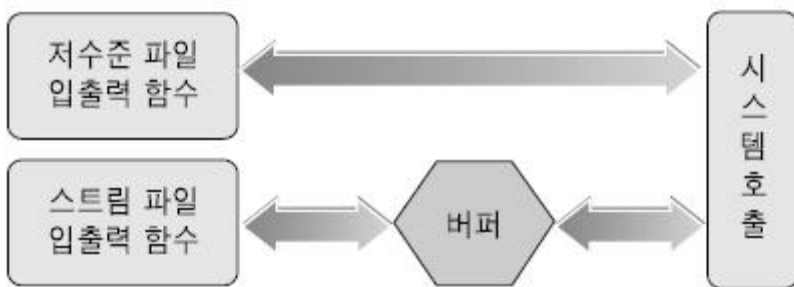
- * 디바이스 파일은 /dev 에서 관리
- 디바이스 파일 형식: 문자형 c , 블록형 b
- 리눅스에서 사용되는 디바이스 파일의 주번호

주번호	문자 디바이스	블록 디바이스
0	NFS, 네트워크 사용을 위해 이름이 없음	
1	메모리장치 (mem)	RAM 디스크
2	pseude TTY 마스터(pty*)	디스크(fd*)
3	pseudo TTY 슬레이브(ttyp*)	IDE 하드디스크(hd*)
4	터미널	
5	터미널과 Aux	
6	병렬 인터페이스	
7	가상 콘솔(vcs*)	루프백 디바이스
60-63		지역/실험적 사용 목적
120-127		지역/실험적 사용 목적
240-252		지역/실험적 사용 목적

- * mknod 명령으로 디바이스 파일 생성하기
 - ; 디바이스 이름이 fd0, 주번호가 250, 부번호가 0인 문자 디바이스 파일 생성
 - mknod fd0 c 250 0

2. 디바이스 드라이버

- 디바이스와 시스템 사이에 데이터를 주고받기 위한 인터페이스
- 표준적으로 동일한 서비스 제공을 목적
- 커널의 일부분으로 내장
 - 커널이 직접 하드웨어를 제어하지 않고 디바이스 드라이버의 인터페이스를 통해 하드웨어 제어
- 서브루틴과 데이터의 집합체
- 디바이스의 고유한 특성을 내포
- 디바이스 하드웨어에 데이터를 읽거나 쓰거나 제어하는 루틴의 집합
 - 루틴들은 운영체제에 잘 연동
 - 운영체제가 하드웨어를 제어할 수 있도록 인터페이스 제공



	저수준 파일 입출력 함수	고수준 파일 입출력함수
접근 형태	직접 접근	형식화된 형태로 접근, 사용하기 쉬움
버퍼 사용 유무	사용 않음	버퍼를 사용
디바이스 파일 다루기	가능	곤란
디바이스 드라이버 구현	사용	사용 불가
종류	open(), read(), write(), close(), fsync(), ioctl(), poll(), mmap() 등	fopen(), fread(), fwrite(), fclose() 등

【학습정리】

1. 디바이스 드라이버 종류, 파일

* 개념

- 디바이스: 컴퓨터시스템의 주변 하드웨어
- 하드디스크, 플로피디스크, 프린터, 단말기, 스캐너, 네트워크 댑터, 터치 스크린, LCD디스플레이, 오디오 등등
- 디바이스 드라이버는 디바이스 구동 프로그램

* 디바이스 시스템 구성 : device + controller

- device : controller를 통해 입력된 명령을 수행하고 결과를 알려주는 역할
- controller : 디바이스와 디바이스 드라이버 사이의 인터페이스 담당구성은 버퍼 및 제어/상태 레지스터 등

* 디바이스의 종류

- 종류

: 문자 디바이스(character device)

: 블록 디바이스(block device)

: 네트워크 디바이스(network device)

* 디바이스 체계

- 주번호(major number)와 부번호(minor number)로 구분
- 동일한 디바이스는 동일한 주번호를 가지며 서로 구분하기 위해 부번호를 사용
- 구별하기 위해 시작문자를 통일
(IDE 하드디스크 : hd, 터미널 : tty, 이더넷 : eth)

2. 디바이스 파일 생성

* 특징

- 호스트 시스템의 디바이스 파일이 있는 /dev 디렉토리 내용관찰
- 디바이스 파일의 생성
mknod <디바이스 파일명> <디바이스 파일 형식> <주번호> <부번호>
- 디바이스 파일은 /dev 에서 관리 : 디바이스 파일 형식: 문자형 c, 블록형 b