

# 17강. 운영체제의 큰 그림

지금까지는 컴퓨터 구조에 대해서 배웠음. 자세히는 컴퓨터를 이루고 있는 다양한 하드웨어에 대해 배웠음.

이번부터는 그런 하드웨어가 어떻게 소프트웨어적으로 관리될 수 있는지 배움.

Chapter 03-1 운영체제의 큰 그림(1)

- 커널(kernel) - 운영체제의 핵심 기능을 담당하는 부분
  - 자동차의 엔진이나 사람의 심장과도 같은 핵심부

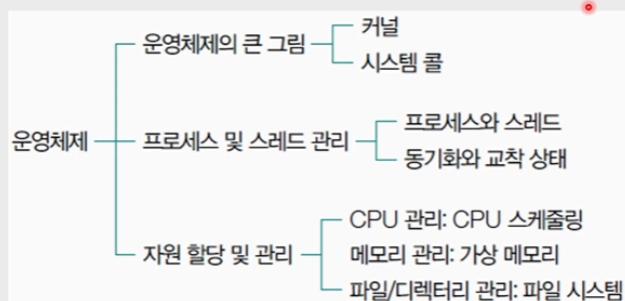
커널  
운영체제  
사용자 응용 프로그램  
사용자 응용 프로그램

◆ 이전이 최대한은 외화 컴퓨터 과학이다

3

## Chapter 03-1 운영체제의 큰 그림(2)

- 운영체제의 2가지 핵심 기능
  - 자원 할당 및 관리
  - 프로세스 및 스레드 관리



>> 이것이 취업을 위한 컴퓨터 과학이다

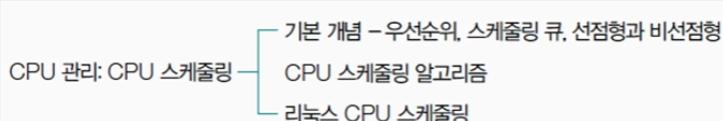
4

## Chapter 03-1 운영체제의 큰 그림(3)

### 운영체제의 역할

#### CPU 관리: CPU 스케줄링

- 운영체제는 실행 중인 모든 프로그램들이 공정하고 합리적으로 CPU를 할당받도록 CPU의 할당 순서와 사용 시간을 결정



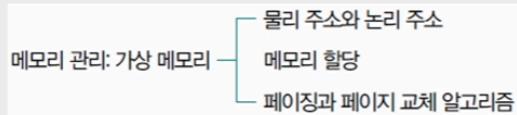
>> 이것이 취업을 위한 컴퓨터 과학이다

5

## Chapter 03-1 운영체제의 큰 그림(4)

### 메모리 관리: 가상 메모리

- 운영체제는 새롭게 실행하는 프로그램을 메모리에 적재하고, 종료된 프로그램을 메모리에서 삭제
- 동시에 낭비되는 메모리 용량이 없도록 효율적으로 관리



>> 이것이 취업을 위한 컴퓨터 과학이다

6

## Chapter 03-1 운영체제의 큰 그림(5)

### 파일/디렉터리 관리: 파일 시스템

- 운영체제는 보조기억장치를 효율적으로 관리하기 위해 파일 시스템을 활용
- 파일 시스템은 보조기억장치 내의 정보를 파일 및 폴더(디렉터리) 단위로 접근·관리할 수 있도록 만드는 운영체제 내부 프로그램



>> 이것이 취업을 위한 컴퓨터 과학이다

7

## Chapter 03-1 운영체제의 큰 그림(5)

### 운영체제의 입출력장치 및 캐시 메모리 관리

- 컴퓨터의 핵심 부품인 CPU, 메모리, 보조기억장치 외에 입출력장치와 캐시 메모리도 운영체제에 의해 관리되는 자원
- 운영체제는 일부 입출력장치의 장치 드라이버, 하드웨어 인터럽트 서비스 루틴을 제공하거나 캐시 메모리의 일관성을 유지하는 등의 기능을 제공

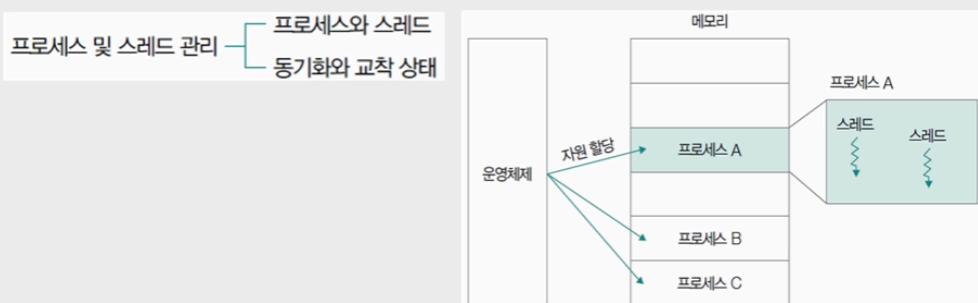
>> 이것이 취업을 위한 컴퓨터 과학이다

8

## Chapter 03-1 운영체제의 큰 그림(6)

### 프로세스 및 스레드 관리

- 프로세스(process) - '실행 중인 프로그램'
- 스레드(thread) - 프로세스를 이루는 실행의 단위
- 운영체제는 동시다발적으로 실행되는 프로세스와 스레드가 올바르게 처리되도록 실행의 순서를 제어하고, 프로세스와 스레드가 요구하는 자원을 적절하게 배분



>> 이것이 취업을 위한 컴퓨터 과학이다

9

한 프로세스가 여러개 스레드가 있다면 동시다발적으로 실행 가능. 여러 실행 갈래로.

## Chapter 03-1 운영체제의 큰 그림(8)

### 시스템 콜과 이중 모드

- 커널 영역(kernel space)
  - 운영체제는 메모리 내의 커널 영역에 따로 적재되어 실행
- 사용자 영역(user space)
  - 운영체제가 적재되는 커널 영역 외에 사용자 응용 프로그램이 적재되는 공간
  - 운영체제의 기능을 제공받기 위해서는 커널 영역에 적재된 운영체제 코드를 실행해야 함

>> 이것이 취업을 위한 컴퓨터 과학이다

11

커널 영역에 적재되어있는 운영체제에 부탁해야 됨.

## Chapter 03-1 운영체제의 큰 그림(9)

- 시스템 콜(system call)을 호출하여 운영체제 코드를 실행
  - 시스템 콜은 운영체제의 서비스를 제공받기 위한 수단(인터페이스)으로, 호출 가능한 함수의 형태를 가짐

~~~ 여기서 최선의 의향과 표준화된 표기법입니다

12

그 부탁이 바로 시스템 콜이라 하는 것.

## Chapter 03-1 운영체제의 큰 그림(10)

- 운영체제에 따라 제공하는 시스템 콜의 종류와 개수는 다양함
- 예시) 유닉스 계열의 운영체제에서 사용하는 대표적인 시스템 콜의 종류

| 구분        | 시스템 콜     | 설명                                   |
|-----------|-----------|--------------------------------------|
| 프로세스 관리   | fork()    | 새 자식 프로세스 생성                         |
|           | execve()  | 프로세스 실행(메모리 공간을 새로운 프로그램의 내용으로 덮어씌움) |
|           | exit()    | 프로세스 종료                              |
|           | waitpid() | 자식 프로세스가 종료할 때까지 대기                  |
| 파일 관리     | open()    | 파일 열기                                |
|           | close()   | 파일 닫기                                |
|           | read()    | 파일 읽기                                |
|           | write()   | 파일 쓰기                                |
|           | stat()    | 파일 정보 획득                             |
| 디렉터리 관리   | chdir()   | 작업 디렉터리 변경                           |
|           | mkdir()   | 디렉터리 생성                              |
|           | rmdir()   | 비어 있는 디렉터리 삭제                        |
| 파일 시스템 관리 | mount()   | 파일 시스템 마운트                           |
|           | umount()  | 파일 시스템 마운트 해제                        |

13

외울 필요 없음.

## Chapter 03-1 운영체제의 큰 그림(11)

여기서 잠깐

### 프로세스의 계층 구조

- fork() 시스템 콜을 통해 알 수 있듯 프로세스는 시스템 콜을 통해 또 다른 프로세스를 생성하고, 그렇게 생성된 프로세스는 또 다른 프로세스를 생성할 수 있음
- 많은 운영체제에서 프로세스들은 이렇게 계층적으로 관리
- 부모 프로세스(parent process) - 새 프로세스를 생성한 프로세스
- 자식 프로세스(child process) - 부모 프로세스에 의해 생성된 프로세스



>> 이것이 취업을 위한 컴퓨터 과학이다

14

## Chapter 03-1 운영체제의 큰 그림(12)

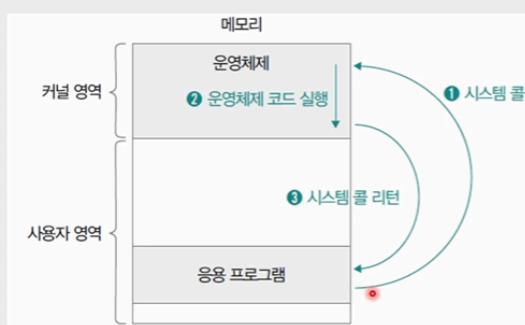
- 컴퓨터 내부에서 시스템 콜이 호출되면 다음과 같은 작업이 수행
  - 소프트웨어 인터럽트, 커널 모드, 사용자 모드



>> 이것이 취업을 위한 컴퓨터 과학이다

## Chapter 03-1 운영체제의 큰 그림(13)

- ① 사용자 영역을 실행하는 과정에서 시스템 콜 호출  
여느 인터럽트와 마찬가지로 CPU는 현재 수행 중인 작업을 백업
- ② 커널 영역 내의 인터럽트를 처리하기 위한 코드(시스템 콜을 구성하는 코드) 실행
- ③ 다시 사용자 영역의 코드 실행 재개



>> 이것이 취업을 위한 컴퓨터 과학이다

16

## Chapter 03-1 운영체제의 큰 그림(14)

- 이중 모드(dual mode)
  - CPU는 명령어를 실행하는 과정에서 사용자 영역을 실행할 때의 모드와 커널 영역을 실행할 때의 모드를 구분하여 실행
  - 사용자 모드(user mode) - 사용자 영역에 적재된 코드를 실행할 때의 실행 모드
    - 운영체제 서비스를 제공받을 수 없는 실행 모드
    - 사용자 모드로 실행되는 명령어는 실수로라도 자원에 접근할 수 없음
  - 커널 모드(kernel mode) - 커널 영역에 적재된 코드를 실행할 때의 실행 모드
    - 운영체제 서비스를 제공받을 수 있는 실행 모드
    - CPU가 커널 모드로 명령어를 실행하면 자원에 접근하는 명령어를 비롯한 모든 명령어를 실행할 수 있음
    - 운영체제는 이 커널 모드로 실행되기 때문에 자원에 접근 가능

## Chapter 03-1 운영체제의 큰 그림(15)

- 응용 프로그램은 실행 과정에서 시스템 콜을 매우 빈번하게 호출
  - 예) 화면에 “hello world”라는 문자열을 출력하는 단순한 프로그램조차 실행 과정에서 무려 600회가 넘는 시스템 콜을 호출

```
$ strace -c python3 test.py
hello world!
% time    seconds   usecs/call     calls    errors syscall
----- -----
 20.19   0.000213      2      103      24 stat
 18.39   0.000194     13      14      72 getdents64
 10.14   0.000107      0      111      72 openat
  9.19   0.000097      1      54      3 read
  5.97   0.000063      1      63      1 fstat
  5.50   0.000058      1      52      3 lseek
  5.21   0.000055      0      68      1 rt_sigaction
  5.12   0.000054      1      42      1 close
  4.27   0.000045      0      48      1 mmap
  2.65   0.000028      1      28      22 ioctl
...
  0.00   0.000000      0      1      1 execve
  0.00   0.000000      0      2      1 arch_prctl
----- -----
100.00   0.001055
                                         645      125 total
```

개발 프로그램, 응용 프로그램들은 시스템 콜을 엄청 보냄. 간단한 저 프로그램조차 600번이 넘는 시스템 콜을 보냄.

## Chapter 03-1 운영체제의 큰 그림(16)

- 프로그램은 시스템 콜을 통해 사용자 모드와 커널 모드를 빈번히 오가며 운영체제의 소스 코드를 실행
  - 소스 코드로 작성된 프로그램뿐만 아니라 ls와 같은 명령어 또한 프로그램이기 때문에 마찬가지로 실행 과정에서 수많은 시스템 콜을 호출

```
$ strace -c ls
myfile1 myfile2
% time    seconds   usecs/call     calls    errors syscall
----- -----
100.00    0.000003      0       11        0  close
      0.00    0.000000      0        7        0  read
      0.00    0.000000      0        1        0  write
      0.00    0.000000      0        8        7  stat
...
      0.00    0.000000      0        1        0  set_tid_address
      0.00    0.000000      0       61       52  openat
      0.00    0.000000      0        1        0  set_robust_list
      0.00    0.000000      0        1        0  prlimit64
-----
100.00    0.000003      0      162        0  64 total
```

>> 이것이 취업을 위한 컴퓨터 과학이다

19