

문제 1

(1) exam.txt 파일의 첫 번째 블록이 53번 디스크 블록에 할당되어 있다면 exam.txt 파일이 저장된 디스크 블록들을 순서대로 나열하라.

시작 블록: **53**

FAT에서 53번 항목의 값: **2**

FAT에서 2번 항목의 값: **124**

FAT에서 124번 항목의 값: **54**

FAT에서 54번 항목의 값: **320**

FAT에서 320번 항목의 값: **-1** (파일의 끝)

따라서 exam.txt 파일이 저장된 디스크 블록들은 순서대로 **53, 2, 124, 54, 320** 입니다.

1. (2) 블록 크기가 1KB라면 exam.txt 파일의 크기

로 가장 적절한 것은?

2. exam.txt 파일은 총 5개의 디스크 블록 (53, 2, 124, 54, 320)을 사용합니다.
3. 블록 크기가 1KB이므로, 파일의 총 크기는 5 블록×1KB/블록=5KB 입니다.
4. 선택지: ① 1KB ② 4KB ③ 4.5KB ④ 5KB
5. 가장 적절한 파일 크기는 ④ **5KB** 입니다.

주어진 FAT (File Allocation Table)을 참고하여 다음 질문에 답하십시오.

(1) score.txt 파일의 첫 번째 블록이 55번 디스크 블록에 할당되어 있다면 score.txt 파일이 저장된 디스크 블록들을 순서대로 나열하라.

score.txt 파일의 디스크 블록 할당 순서는 다음과 같습니다:

시작 블록: **55**

FAT에서 55번 항목의 값: **321**

1. FAT에서 321번 항목의 값: **52**
- 2.

3. FAT에서 52번 항목의 값: **123**
- 4.
5. FAT에서 123번 항목의 값: **-1** (파일의 끝)

따라서 score.txt 파일이 저장된 디스크 블록들은
순서대로 **55, 321, 52, 123** 입니다.

**(2) 블록 크기가 2KB일 때 score.txt 파일의 크기로
가장 적절한 것은?**

1. score.txt 파일은 총 4개의 디스크 블록 (55, 321, 52, 123)을 사용합니다.
2. 블록 크기가 2KB이므로, 파일의 총 크기는 4 블록×2KB/블록=8KB 입니다.
3. 선택지: ① 1KB ② 7KB ③ 8KB ④ 9KB
4. 가장 적절한 파일 크기는 **③ 8KB** 입니다.

문제 3유닉스 파일 시스템에서 /usr/dev/source/
app.c 파일을 읽으려고 한다. 다음 질문에 답하라.

1. **(1) 이 파일을 읽기 위해 운영체제는 몇 개의 i-node를 읽어야 하는가? 읽어야 하는 i-node를
순서대로 나열하라.**
2. 파일 경로의 각 구성요소(디렉터리 및 파일)마다

i-node가 존재합니다. 따라서 /usr/dev/source/app.c 파일을 읽기 위해 운영체제는 다음 순서대로 i-node를 읽어야 합니다:

3.

4. / (루트 디렉터리)의 i-node

5.

usr (디렉터리)의 i-node

- **dev** (디렉터리)의 i-node

-

source (디렉터리)의 i-node

app.c (파일)의 i-node

총 5개의 i-node를 읽어야 합니다.

(2) i-node와 디스크 블록들이 읽혀지는 순서를 기술하라.

파일 /usr/dev/source/app.c를 읽는 과정에서 i-node와 디스크 블록이 읽히는 순서는 다음과 같습니다:

루트 디렉터리 (/)의 **i-node**를 읽습니다.

루트 디렉터리의 ****데이터 블록(들)****을 읽어 usr 디렉터리의 i-node 번호를 찾습니다.

usr 디렉터리의 **i-node**를 읽습니다.

usr 디렉터리의 ****데이터 블록(들)****을 읽어 dev 디렉터리의 i-node 번호를 찾습니다.

dev 디렉터리의 **i-node**를 읽습니다.

dev 디렉터리의 ****데이터 블록(들)****을 읽어 source 디렉터리의 i-node 번호를 찾습니다.

source 디렉터리의 **i-node**를 읽습니다.

source 디렉터리의 ****데이터 블록(들)****을 읽어 app.c 파일의 i-node 번호를 찾습니다.

app.c 파일의 **i-node**를 읽습니다. 이 i-node에는 app.c 파일의 실제 데이터 블록들에 대한 포인터 정보가 들어있습니다.

app.c 파일의 ****데이터 블록(들)****을 읽습니다. 이것이 파일의 실제 내용입니다.

(3) 이 파일이 읽혀지는 과정을 생각했을 때, 운영체제가 i-node 테이블을 메모리에 적재해 두고 사용하는 이유는 무엇인지 설명하라.

운영체제가 i-node 테이블(또는 자주 사용되는 i-node들)을 메모리에 적재해 두고 사용하는 주된 이유는 **성능 향상** 때문입니다.

접근 속도 증가: 디스크는 메모리에 비해 접근 속도가 매우 느립니다. 파일 경로를 탐색하거나 파일 정보를 얻기 위해서는 여러 i-node를 순차적으로 읽어야 합니다 (문제 3-(2) 참고). 만약 매번 디스크에서 i-node를 읽어온다면 파일 접근 시간이 매우 길어질 것입니다. i-node를 메모리에 캐싱해두면 디스크 접근 횟수를 줄여 파일 시스템 연산 속도를 크게 향상시킬 수 있습니다.

디스크 I/O 감소: i-node 정보를 메모리에서 빠르게 가져올 수 있으므로, 불필요한 디스크 읽기/쓰기 작

업을 줄일 수 있습니다. 이는 전체 시스템의 응답성을 높이고 디스크의 부하를 줄이는 효과를 가져옵니다.