

FAT12 개요

FAT(파일 할당 테이블)은 하드 디스크나 플로피 디스크에 저장된 테이블로, 디스크에 있는 모든 데이터 클러스터의 상태와 위치를 나타냅니다. 파일 할당 테이블은 디스크의 "목차"로 간주될 수 있습니다. 파일 할당 테이블이 손상되거나 손실되면 디스크를 읽을 수 없습니다.

이 문서에서는 FAT12 파일 시스템에 대해 설명합니다. FAT12는 플로피 디스크의 파일 시스템입니다. 숫자 "12"는 FAT가 12비트 항목으로 구성된다는 사실에서 파생되었습니다.

플로피 디스크의 저장 공간은 섹터라는 단위로 구분됩니다. 더 큰 저장 장치에서는 여러 섹터가 클러스터를 형성합니다. 그러나 플로피 디스크의 경우 클러스터의 섹터 수는 1개입니다. 또한 플로피 디스크의 경우 섹터(및 클러스터)의 크기는 512바이트입니다.

1. 디스크 구성

플로피 디스크 레이아웃(FAT-12)은 부트 섹터, FAT 테이블, 루트 디렉터리 및 데이터 영역의 네 가지 주요 섹션으로 구성됩니다.

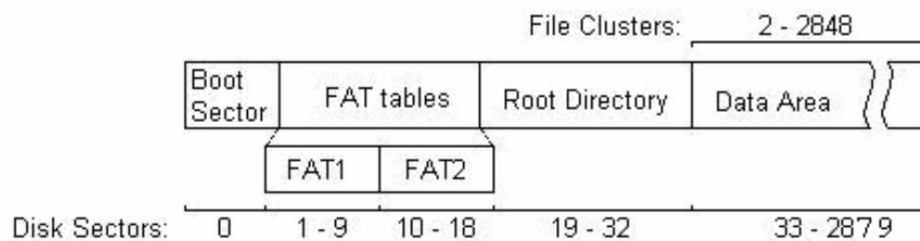


그림 1 FAT12 파일 시스템의 디스크 구성¹

- 부트 섹터는 볼륨 또는 디스크의 첫 번째 섹터(섹터 0)로 구성됩니다. 부트 섹터에는 존재하는 FAT 테이블의 복사본 수, 섹터 크기, 클러스터의 섹터 수 등을 포함하여 파일 시스템의 나머지 구성에 대한 특정 정보가 포함되어 있습니다.

등.

- FAT 테이블에는 디스크의 모든 클러스터에 대한 포인터가 포함되어 있으며 현재 클러스터 체인의 다음 클러스터 번호, 클러스터 체인의 끝, 클러스터가 비어 있는지 또는 오류가 있는지 여부를 나타냅니다. FAT 테이블은 디스크의 나머지 부분에서 파일과 디렉터리의 위치를 찾는 유일한 방법입니다. 일반적으로 데이터 보안 및 복구 목적으로 디스크에 FAT 테이블의 중복 복사본이 두 개 있습니다. 플로피에서는 클러스터가 단 하나의 섹터로 구성되므로 디스크의 모든 섹터에 대한 FAT 항목 포인터가 있습니다.

- 루트 디렉터리는 디스크의 기본 디렉터리입니다. 디스크의 데이터 영역에 위치한 다른 디렉터리와 달리 루트 디렉터리는 크기가 유한합니다(FAT12의 경우 14섹터 * 16디렉토리).

¹ 이 수치는 브리검 영 대학교의 CS324 과정 웹사이트에서 얻은 것입니다.

섹터당 항목 = 224개의 가능한 항목), 생성될 수 있는 파일 또는 디렉터리의 총량을 제한합니다.

- 데이터 영역. - 데이터 영역의 첫 번째 섹터 또는 클러스터는 파일의 클러스터 2에 해당합니다.

시스템(첫 번째 클러스터는 항상 클러스터 2임) 데이터 영역에는 파일 및 디렉터리 데이터가 포함되어 디스크의 나머지 섹터에 걸쳐 있습니다.

디스크 구성을 요약하면 다음과 같습니다.

논리 섹터	콘텐츠
0	부트 섹터
1	(첫 번째) FAT의 첫 번째 섹터
10	두 번째 FAT의 첫 번째 섹터
19	플로피 디스크 루트 디렉터리의 첫 번째 섹터
더블 엑스	루트 디렉터리의 마지막 섹터(부트 섹터의 바이트 17 및 18 참조)
XX + 1	플로피 디스크의 데이터 영역의 시작

FAT12의 경우 14개 섹터가 루트 디렉터리용으로 예약되어 있으므로 XX = 32입니다.

2. 부트 부문

부트 섹터는 디스크의 섹터 0에 존재하며 운영 체제가 디스크를 올바르게 사용하는 데 필요한 정보 세트인 기본 디스크 구조를 포함합니다. 디스크를 사용할 때마다 부트 섹터의 정보를 읽고 필요한 정보를 추출합니다. DOS 포맷 플로피의 부트 섹터는 다음과 같은 일련의 바이트입니다.

시작 바이트 길이(바이트)	저장된 데이터	
0	11	무시하다
11	2	섹터당 바이트
13	1	클러스터당 섹터
14	2	예약된 섹터 수
16	1	FAT의 수
17	2	최대 루트 디렉터리 항목 수
19	2	총 섹터 수
21	1	무시하다

22	2	FAT당 섹터
24	2	트랙당 섹터
26	2	헤드 수
28	4	무시하다
32	4	FAT32의 총 섹터 수(FAT12 및 FAT16의 경우 0)
36	2	무시하다
38	1	부팅 서명 ^b
39	4	볼륨 ID ^씨
43	11	볼륨 라벨
54	8	파일 시스템 유형(예: FAT12, FAT16) ^e
62	-	나머지 부트 섹터(무시)

†. 총 섹터 수 - 이 필드는 볼륨의 총 16비트 섹터 수입니다. 이 수에는 볼륨의 4개 지역 모두에 있는 모든 섹터 수가 포함됩니다.
다. FAT12 및 FAT16 볼륨의 경우 이 필드에는 섹터 수가 포함됩니다. FAT32의 경우 바이트 32-35를 참조하십시오.

비. 부팅 서명 - 확장된 부팅 서명입니다. 이는 다음을 나타내는 서명 바이트입니다.
부트 섹터에는 다음 세 개의 필드가 있습니다. 이를 나타내려면 값이 0x29여야 합니다.

씨. 볼륨 ID - 볼륨 일련 번호이기도 합니다. 이 필드는 볼륨 레이블과 함께 이동식 미디어에 대한 볼륨 추적을 지원합니다. 이러한 값을 사용하면 FAT 파일 시스템 드라이버가 이동식 드라이브에 잘못된 디스크가 삽입되었음을 감지할 수 있습니다. 이 ID는 일반적으로 현재 날짜와 시간을 32비트 값으로 결합하여 생성됩니다.

디. 볼륨 레이블 - 이 필드는 루트 디렉터리에 기록된 11바이트 볼륨 레이블과 일치합니다.
참고: FAT 파일 시스템 드라이버는 루트 디렉터리의 볼륨 레이블 파일 이름이 변경되거나 생성될 때 이 필드를 업데이트해야 합니다. 볼륨 레이블이 없을 때 이 필드의 설정은 "NO NAME" 문자열입니다.

이자형. 파일 시스템 유형 - "FAT12", "FAT16" 또는 "FAT" 문자열 중 하나입니다. 참고: 많은 사람들은 이 필드의 문자열이 해당 볼륨에 있는 FAT 유형(FAT12, FAT16 또는 FAT32)을 결정하는 것과 관련이 있다고 생각합니다. 이것은 사실이 아닙니다. 이 문자열은 정보 제공용일 뿐이며 올바르게 설정되지 않거나 존재하지 않는 경우가 많기 때문에 Microsoft 파일 시스템 드라이버에서 FAT 유형을 확인하는 데 사용되지 않습니다. 하지만 이 문자열은 Microsoft가 아닌 일부 FAT 파일 시스템 드라이버에서 확인하므로 FAT 유형을 기반으로 설정해야 합니다.

3. FAT(파일 할당 테이블)

FAT는 앞서 설명한 대로 저장 장치의 데이터 섹터를 매핑하는 데이터 구조입니다. 이는 배열과 유사하며 FAT의 각 항목은 디스크의 데이터 클러스터에 해당합니다. 관심 있는 FAT의 각 항목 값은 다음과 같습니다.

- 이 데이터 클러스터가 파일의 마지막 클러스터를 나타내는 값
- 이 데이터 클러스터가 현재 사용되지 않음을 나타내는 값
- 현재 파일의 NEXT 데이터 클러스터가 위치하는 위치를 나타내는 값.

특히 FAT 항목 값은 다음을 의미합니다.

값	의미
0x00	미사용
0xFF0-0xFF6 예약된 클러스터	
0xFF7	불량 클러스터
0xFF8-0xFFFF 파일의 마지막 클러스터	
(그 외 모든 것) 파일의 다음 클러스터 번호	

물리적 데이터 섹터 번호를 논리적 데이터 섹터 번호로 변환:

FAT는 논리적 데이터 섹터 값에서 작동합니다. FAT12 시스템의 경우 물리 섹터 번호에서 논리 섹터 번호를 결정하는 동안 다음 두 가지 요소를 고려해야 합니다.
계정.

- 디스크 구성을 보면 처음 33개 섹터가 미리 정의되어 있는 것을 알 수 있습니다. 사용자 데이터를 보유하는 실제 데이터 섹터는 처음 33개 섹터에 존재하지 않으며 섹터 번호 33에서 시작합니다(0부터 시작한다는 점을 기억하세요).
- FAT의 위치 0과 1에 있는 항목은 예약되어 있습니다. 따라서 실제로 물리적 섹터 번호 33에 대한 설명을 포함하는 것은 FAT의 항목 2입니다.

따라서 물리 섹터 번호 = 33 + FAT 항목 번호 - 2

예를 들어 FAT의 항목 5는 실제로 물리적 데이터 섹터 번호 36을 나타냅니다.

4. 디렉토리

디렉토리(예: 루트 디렉토리)는 하나 이상의 섹터를 차지한다는 점에서 디스크의 파일처럼 존재합니다. 디렉토리의 각 섹터(512바이트)에는 16개의 디렉토리 항목(각각 32바이트 길이)이 포함되어 있습니다. 각 디렉토리 항목은 디스크의 일부 파일이나 하위 디렉토리를 설명하고 가리킵니다. 따라서 디렉토리에 대한 디렉토리 항목 모음은 해당 디렉토리의 파일과 하위 디렉토리를 지정합니다.

각 디렉토리 항목에는 해당 항목이 가리키는 파일이나 하위 디렉토리에 대한 다음 정보가 포함되어 있습니다.

오프셋(바이트) 길이(바이트) 설명		
0	8	파일 이름(단, 이 필드의 첫 번째 바이트에 대한 아래 참고 사항 참조)
8	삼	확대
11	1	속성(아래 세부정보 참조)
12	2	예약된
14	2	생성 시간
16	2	생산 일
18	2	마지막 액세스 날짜
20	2	FAT12에서는 무시
22	2	마지막 쓰기 시간
24	2	마지막 쓰기 날짜
26	2	첫 번째 논리적 클러스터
28	4	파일 크기(바이트)

참고: FAT12 시스템에서는 클러스터가 단 하나의 섹터만 보유한다는 것을 이미 확인했습니다.
따라서 이 문서의 나머지 부분에서는 두 단어가 같은 의미로 사용됩니다.

디렉토리 항목에 대한 참고사항:

1. 첫 번째 논리적 클러스터 필드는 파일 또는 하위 디렉터리가 시작되는 위치를 지정합니다. 그래서 디렉터리 항목은 파일이나 하위 디렉터리를 가리킵니다. FAT 인덱스의 값을 제공합니다.
예를 들어 첫 번째 논리적 클러스터 값이 "2"인 경우 FAT 배열에 대한 인덱스는 FAT12 시스템의 물리적 클러스터 "33"인 "2"여야 함을 의미합니다. 첫 번째 논리적 클러스터의 값이 "0"인 경우 루트 디렉터리의 첫 번째 클러스터를 참조하므로 해당 디렉터리 항목은 루트 디렉터리를 설명합니다. (루트 디렉토리는 "." 항목으로 나열됩니다. 즉, 모든 하위 디렉터리의 상위 디렉토리입니다.)
2. 파일 이름 필드의 첫 번째 바이트가 0xE5이면 디렉토리 항목은 사용 가능 (즉, 현재 사용되지 않음)이므로 디렉토리 항목과 연관된 파일이나 하위 디렉터리가 없습니다.
3. 파일 이름 필드의 첫 번째 바이트가 0x00이면 이 디렉토리 항목은 비어 있고 이 디렉터리의 나머지 모든 디렉토리 항목도 비어 있습니다.
4. 디렉토리 항목의 속성 필드는 각 비트가 항목을 참조하는 8비트 수량입니다.
이 디렉토리 항목이 가리키는 파일 또는 하위 디렉터리의 속성(속성)은 다음과 같습니다.

비트 마스크 속성	
0 0x01 읽기 전용	
1 0x02 숨김	
2 0x04 시스템	
3 0x08 볼륨 레이블	
4 0x10 하위 디렉터리	
5 0x20 아카이브	
6 0x40 미사용	
7 0x80 미사용	

ㅏ. 속성 필드의 비트가 설정되면(예: 1) 이는 이 디렉토리 항목이 가리키는 파일이나 하위 디렉토리에 해당 비트와 관련된 속성이 있음을 의미합니다. 예를 들어 속성 필드가 0001 0010이면 이 디렉토리 항목이 가리키는 파일/하위 디렉토리는 숨겨진 하위 디렉토리입니다. (비트 1이 켜져 있으면 숨겨져 있음을 나타냅니다. 비트 4도 켜져 있으며 파일이 아니라 하위 디렉터리임을 나타냅니다. 비트 번호는 오른쪽에서 왼쪽으로 지정됩니다.)

비. 속성 바이트가 0x0F인 경우 이 디렉토리 항목은 긴 파일 이름의 일부이며 이 할당 목적을 위해 무시될 수 있습니다. (FAT 시스템에 대한 Microsoft 백서의 업데이트된 버전에는 FAT12의 긴 파일 이름을 처리하려는 경우에 대한 세부 정보가 포함되어 있습니다.)

5. 시간 및 데이터 필드의 형식은 FAT 시스템에 대한 Microsoft 백서에 지정되어 있습니다. (그러나 이 과제에서는 이러한 형식을 알 필요가 없습니다.)

6. 디렉토리 항목은 파일 또는 하위 디렉토리가 시작되는 위치(첫 번째 논리적 클러스터 필드)와 파일 또는 하위 디렉토리의 길이(파일 크기 필드)를 지정합니다. 그러나 일반적으로 파일이나 하위 디렉터리는 연속적으로 저장되지 않습니다. 클러스터 길이가 1개를 초과하는 파일의 경우 이 문서의 다음 섹션에 따라 FAT를 사용하여 나머지 클러스터를 찾아야 합니다.

FAT-12 파일 이름 및 확장자 표시

DOS의 파일 이름은 전통적으로 이름은 8자, 확장자는 3자로 제한됩니다. 알아야 할 몇 가지 사항이 있습니다.

- 파일/디렉터리 이름 및 확장자는 디렉터리 항목 내에서 null로 끝나지 않습니다. • 파일/디렉터리 이름은 항상 8바이트를 차지합니다. 파일/디렉터리 이름이 8바이트(문자)보다 짧은 경우 나머지 바이트를 공백으로 채웁니다(ASCII 32). 또는 16진수 0x20). 이는 3자 확장에도 적용됩니다.

- 파일/디렉터리 이름과 확장자는 항상 대문자입니다. 주어진 파일/디렉터리 이름을 항상 대문자로 변환하십시오. • 디렉터리 이름에도 확장자가 있을 수 있습니다. • "FILE1" 및 "FILE1.TXT"는 고유합니다(확장자는 중요 함). • 파일과 디렉터리는 속성이 달라도 같은 이름을 가질 수 없습니다.

다음은 일부 파일 이름이 파일/디렉터리 이름 및 디렉터리 항목의 확장명에 할당된 11바이트로 변환되는 방법에 대한 예입니다(따옴표 사이의 공백은 공백으로 간주되어야 함).

- 파일 이름 제공 [01234567012] • "foo.bar"
-> "푸 바"
- "FOO.BAR" -> "FOO 술집"
- "푸.바" -> "FOO 술집"
- "foo" • "foo." -> "FOO "
- > "FOO "
- "피클.A" -> "피클 A"
- "prettybg.big" -> "PRETTYBGBIG" -> 불법입니다! 파일/디렉터리 이름은 시작할 수 없습니다. • ".big"은 "."로 시작됩니다.

5. FAT가 왜 필요한가요?

디렉터리 항목에는 파일이나 하위 디렉터리가 시작되는 위치를 지정하는 첫 번째 논리적 클러스터 필드라는 필드가 있습니다. 파일과 디렉터리는 섹터보다 클 수 있으므로 디렉터리나 파일을 둘 이상의 섹터에 걸쳐 저장해야 할 수도 있습니다. 파일이나 디렉터리에 속하는 데이터 섹터는 항상 메모리의 연속적인 위치에 저장되지는 않습니다. 따라서 FAT는 어떤 섹터가 어떤 파일에 할당되었는지 추적하는 데 사용됩니다.

예를 들어 파일의 전체 내용을 검색하려면 첫 번째 논리적 클러스터 필드가 처음 512바이트의 데이터를 보유하는 섹터 번호를 가리킵니다. 이 섹터의 데이터를 읽어야 합니다. 더 많은 데이터가 있는지 확인하려면 첫 번째 논리 클러스터에 해당하는 FAT 항목을 검사해야 합니다. FAT 항목 값을 검사하면 이 파일에 할당된 다른 섹터가 있는지 확인할 수 있습니다. 있는 경우 논리 섹터 값이 물리 섹터 값으로 변환되고 해당 섹터의 데이터를 읽습니다. 다음으로 두 번째 데이터 섹터에 대한 FAT 항목을 검사하여 파일의 끝인지 확인합니다. 그렇지 않은 경우 프로세스가 계속됩니다.

따라서 FAT를 사용하면 저장 장치의 연속되지 않은 섹터에 저장된 데이터에 액세스할 수 있습니다.

그림 2에서 File1.txt는 논리 섹터 2, 4, 6 및 7에 저장됩니다. 디렉터리 항목 필드 "Start Cluster", 즉 첫 번째 논리 클러스터 필드는 첫 번째 데이터 섹터인 섹터 번호 2를 가리킵니다. FAT에서 FAT 항목 2의 값은 4입니다. 이는 File1.txt의 다음 데이터 섹터가 논리 섹터 4에 저장되어 있음을 나타냅니다. 마지막 섹터는 섹터 7이며 이는 FAT 항목 7이 EOC 값을 보유하고 있다는 점에서 분명합니다.

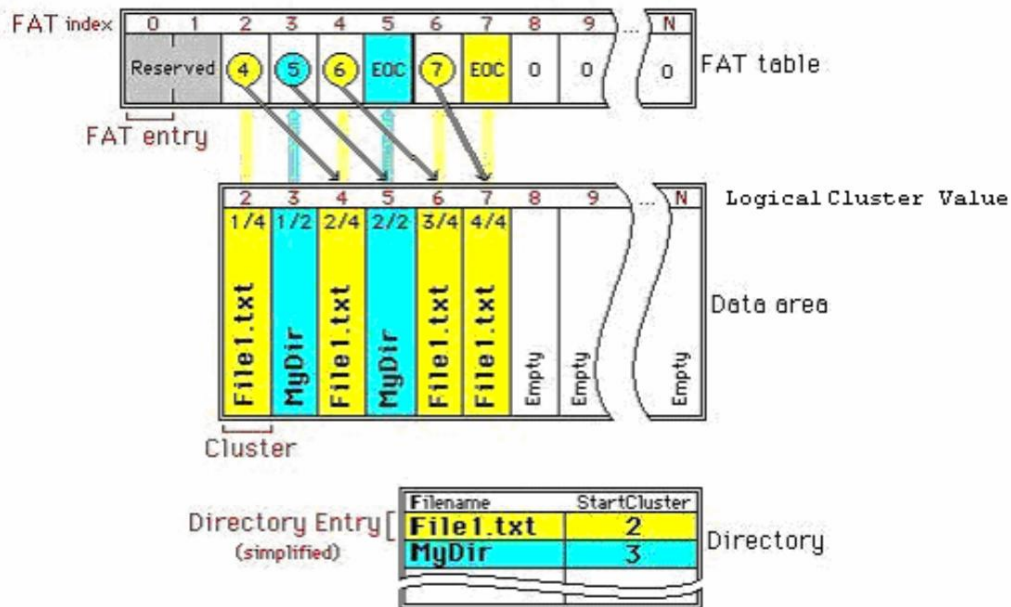


그림 2 FAT1의 사용을 보여주는 예

6. 지방 포장

이 섹션에서는 "12" 값을 선택하는 방법과 12비트 값이 FAT에 저장되는 방법에 대해 설명합니다.

플로피 디스크 공간 = 1.44MB.

섹터의 바이트 수 = 512 1.44MB의 섹터 수

= $x \div 2812$ 따라서 "x" 섹터를 주소 지정하는 데 필요한 최

소 비트 수 = 12비트 ($2^{11} < 2812 < 2^{12}$)

위의 계산에서 볼 수 있듯이 12비트는 플로피 디스크의 전체 1.44M 공간에 액세스하는 데 필요한 최소 비트 수입니다.

12비트의 문제점은 컴퓨터가 모든 것을 8비트(1바이트)의 배수로 저장한다는 것입니다. 따라서 12비트 양을 저장할 때 16비트를 사용하여 12비트를 저장하는 옵션은 모든 FAT 항목에 대해 4비트가 사용되지 않게 되기 때문에 만족스럽지 못했습니다. 플로피에서는 디스크 공간이 이미 부족하기 때문에 다른 솔루션이 설계되었습니다. 이 솔루션에는 2개의 FAT 항목(총 24비트)을 3개의 8비트 위치로 압축하는 작업이 포함됩니다. 이는 효율성 측면에서는 훌륭하지만 단일 항목을 추출하려면 약간의 작업을 수행해야 함을 의미합니다. 더 명확히 하기 위해 FAT의 스냅샷을 살펴보세요. 8비트 항목이 검사됩니다.

¹ 이 수치는 브리검 영 대학교의 CS324 과정 웹사이트에서 얻은 것입니다.

CSSE 332 – 운영 체제
2004-2005년 봄

Rose-Hulman Institute of Technology 컴
퓨터 과학 및 소프트웨어 공학 교수 Archana
Chidanandan

위치 바이트 0

76543210

1 54321098

2 32109876

이 공간에는 2개의 FAT 항목이 있습니다. 첫 번째 항목은 109876543210이며 처음 4비트는 위치 1에서 나옵니다. 두 번째 항목은 321098765432이며 마지막 4비트는 위치 1에서 나옵니다. FAT는 IBM PC 시스템용으로 개발되었으므로 데이터 저장소는 리틀 엔디안 형식입니다. 즉, 최하위 바이트가 가장 낮은 주소에 배치됩니다.

그렇다면 FAT를 어떻게 사용합니까? 첫째, FAT를 C에서 표현할 수 있는 유일한 방법이기 때문에 바이트 배열(8비트 수량)로 생각합니다. 이제 n 번째 FAT 항목에 액세스하려면 다음과 같이 변환해야 합니다. 12비트 및 8비트 값.

- n 이 짝수이면 항목의 물리적 위치는 위치 $1+(3*n)/2$ 의 하위 4비트입니다.
위치 $(3*n)/2$ 의 8비트
- n 이 홀수인 경우 항목의 물리적 위치는 위치 $(3*n)/2$ 의 상위 4비트이고
위치 $1+(3*n)/2$ 의 8비트

FAT에 값을 읽고 쓰는 기능이 제공됩니다.

참고자료

- http://students.cs.byu.edu/~cs345ta/labs/fall03_specs/lab_fat_help.htm#Directory%20structures%20and%20their%20fields •

FAT 파일 시스템에 대한 백서.