

Thì m c con n m trong vùng data

Cách xác nh v trí, làm sao nh v trí c th c a th m c con trong file

Câu h i: t i sao FAT 12 và 16 ko th truy c p c vùng nh l n h n 32MB, là do 12 và 16 thôi

12 tr c n t i a 2 m 12

16 t ng t

Subdirectory n m âu, sub folder

Tìm hi u v c u trúc c a boot sector

FAT ch a hi u v cluster là nh th nào

M i cluster có ph i 12 bit i v i FAT 12

12 t c là 1 byte + 4 bit c a byte li n sau nó úng ko

Trl: nó dùng 12 bit th hi n d i giá tr c a cluster

T ng ng v i m i giá tr c a 12 bit ó thì chúng ta có th xem c là, bi t c là vùng data hay vùng

Trl: t t c u tr n vùng data

V c ph n boot sector thì nó có các byte trong ph n ó thì nó quy nh v nh th i gian ch nh s a t o ra file (? Liên quan gì n boot sector) => b nh m

ó là ph n root directory entry ko ph i boot sector

trong 1 file doc có bao g m nhi u sector, 1 sector là 512 bytes

h i: trong file floppy ó g m nhi u sector và 1 sector thì có 512 bytes úng ko

ch a xác nh c file cha và file th m c con c a nó làm sao

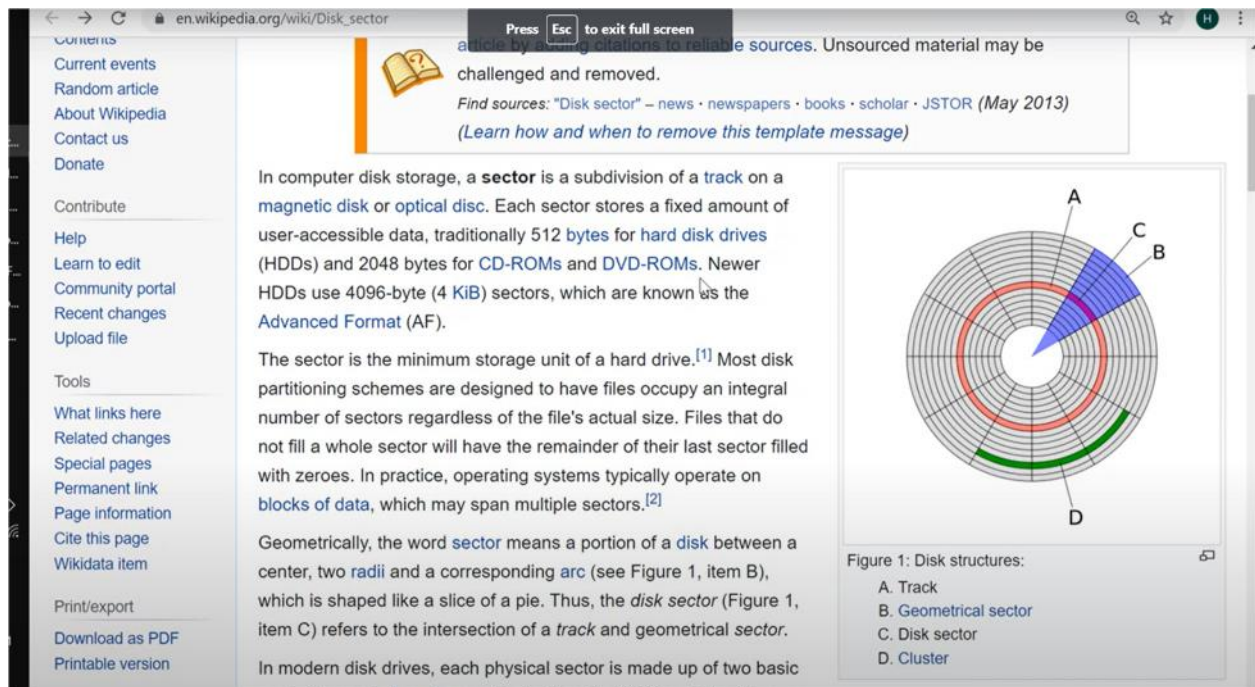
trl: m h

h i: ko h i n a.

Làm rõ v ph n bài t p

C u trúc c a FAT và làm rõ h n v c u trúc c a vi c c n làm gì

u tiên c n hi u v nh ng khái ni m v nh ng n v c a c ng



Số đơn vị trong đĩa, tức là tổng số phân vùng đĩa là phân vùng HDD và tổng số tác vụ đĩa thì nó phụ thuộc vào quy định của các nhà sản xuất vì các đơn vị đo lường quy định khác nhau là cm, hay mm

Và khi làm việc với đĩa thì nó phụ thuộc vào quy định của các nhà sản xuất, đơn vị thường là sector, sector là 1 phân vùng đĩa

Nhìn vào đĩa cứng hay nhìn vào đĩa CD nhìn vào nó có rất nhiều các cái rãnh tròn thì khi có 1 vòng rãnh thì cái đó gọi là track. Hoặc sau này thì 1 track tương đương với 1 đơn vị (1 đĩa cứng thì 1 track là 1 phân vùng đĩa của nó trong đĩa)

Ví dụ 1 đĩa 10 phân vùng đĩa thì hình dung là 10 track

Về mặt physical thì 1 track tương đương 1 vòng quay của 1 cái đĩa như vậy

Nhìn 1 track là 1 cluster, 1 đĩa chia thành nhiều phân vùng thì 1 cái cluster bao gồm nhiều phân vùng

Phân vùng nhỏ nhất gọi là sector

Sector là đơn vị nhỏ nhất truy xuất vào đĩa

Trên sector là cluster

Cluster có thể là 1, hoặc cluster nhỏ nhất có thể bằng 1 sector hoặc là nó bằng multiple sector, 1 track = 1 multiple cluster. Đó là về mặt physical

Vì một file system thì nó cần quy định các thứ dựa trên các physical ntn

Nó sẽ quy định như thế nào thì là sector và lớn hơn là cluster

Chúng ta quan tâm 2 thứ này

Còn như track hay geometrical sector này thì không quan tâm

Hiện rõ hơn cấu trúc của FAT

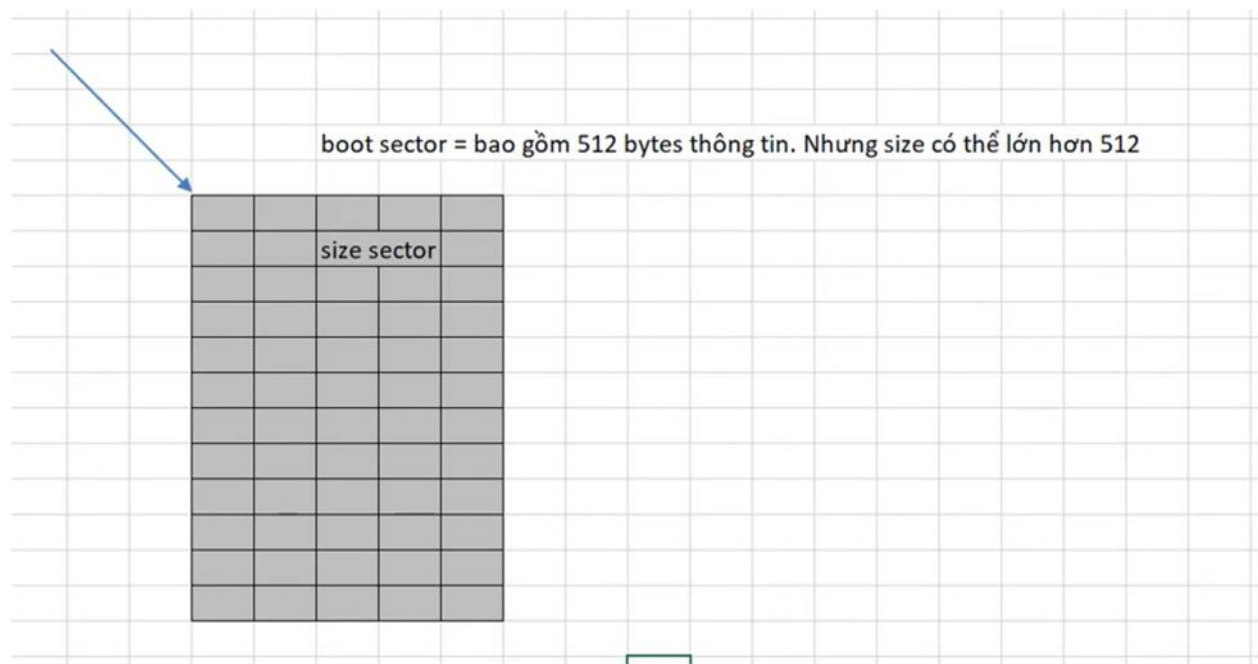
Khi mở file ra thì ý đầu tiên là boot sector, luôn luôn chứa, bao gồm 512 byte thông tin (không có nghĩa là riêng của boot sector là 512) nhưng riêng có thể lớn hơn 512 byte.

Trong boot sector, khi mở file ra thì còn thấy file track đầu tiên của file là track đầu tiên vùng boot sector và khi 512 byte ra thì chứa 512 byte thông tin của bootsector

Khi chứa 1 vị trí nào đó thì nó sẽ quy định như thế nào vị trí nào đó trong 512 byte

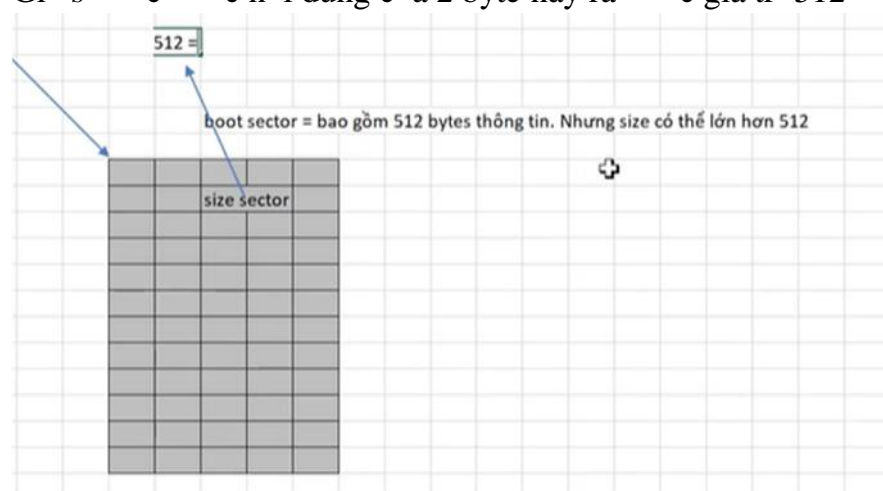
Nó sẽ quy định là có 2 byte quy định size của 1 sector là bao nhiêu

Ví dụ 2 byte này quy định sector size là bao nhiêu đó.



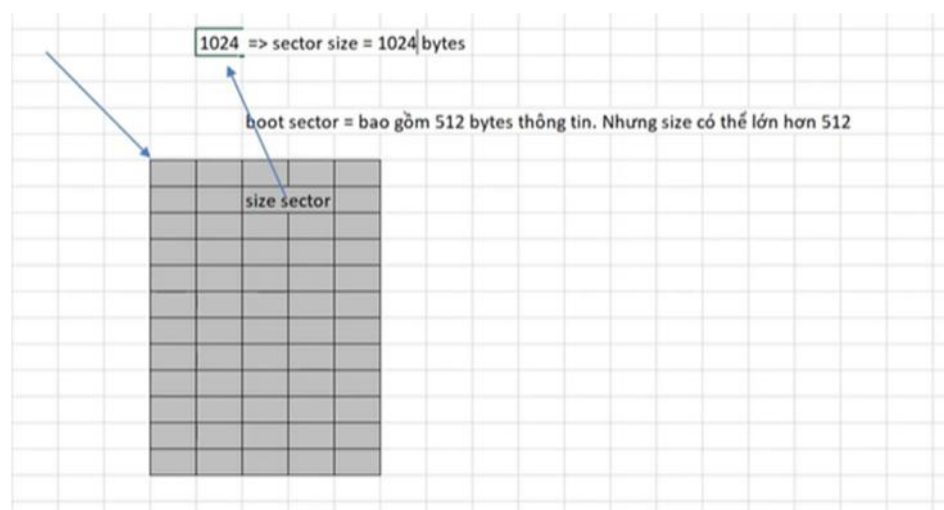
Ví dụ đây có thể chứa gì sẽ chứa nội dung của 2 byte đó và ra giá trị bằng bao nhiêu

Giả sử các nội dung của 2 byte này ra các giá trị 512



Chúng ta thì khi có thông tin về kích thước sector size bằng 512 bytes, thông tin về kích thước 1 sector size bằng 512 bytes

Nếu như có ra sector size nó không bằng 512 mà là 1024 thì có nghĩa là sector size nó là 1024 bytes

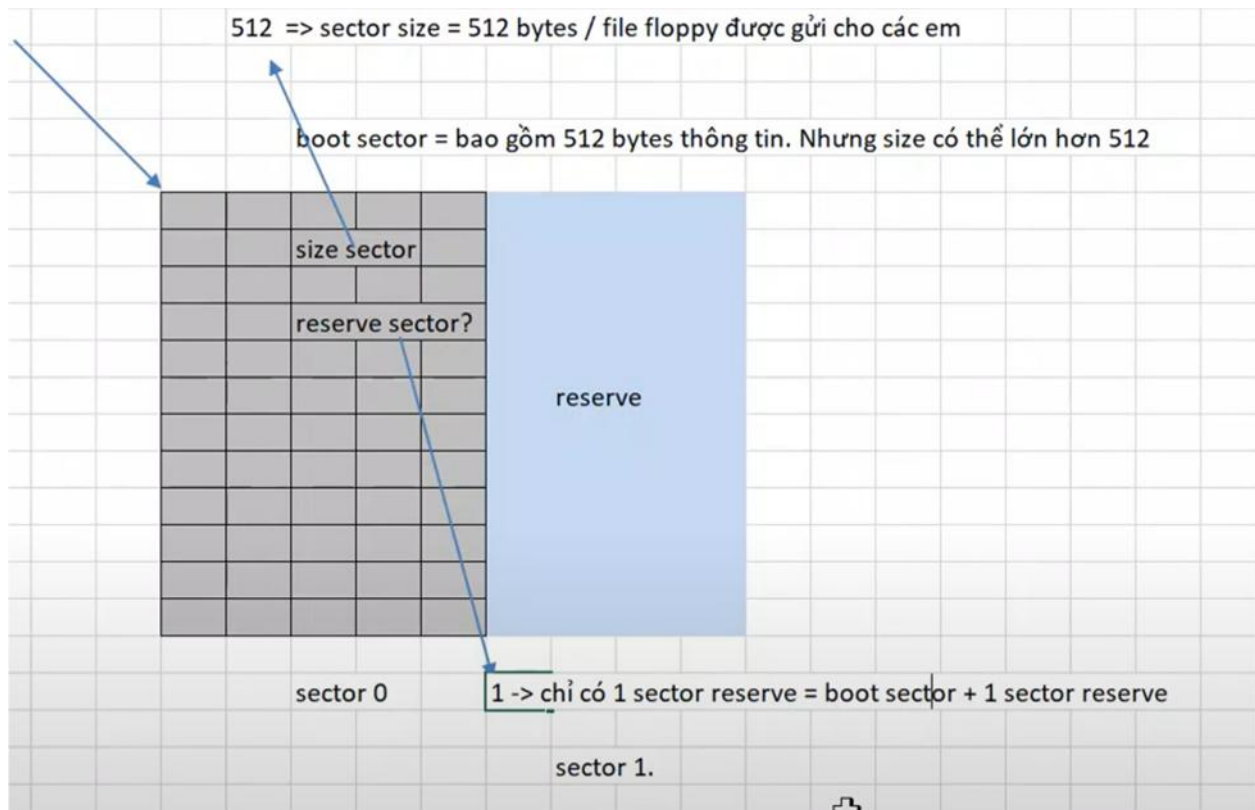


Và file floppy các giá trị thì nó cũng có giá trị là 512. Phần mềm mình viết phải hỗ trợ cả hai file floppy này mà nó phải hỗ trợ cả hai vì thế chúng ta phải viết phần mềm file floppy có thể hỗ trợ cả hai. Tức là viết chương trình có thể hỗ trợ cả hai file floppy có thể quy định sector size là 512 hay 1024

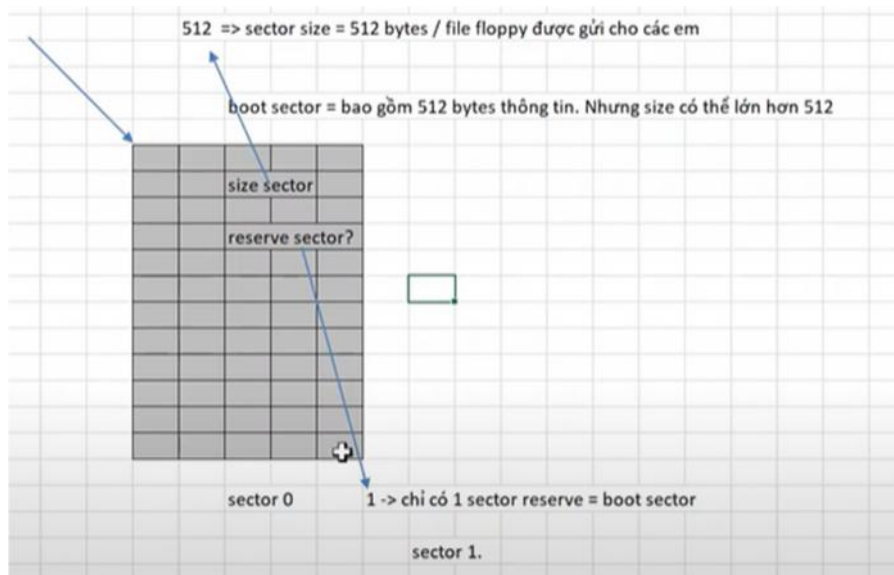
Sector size luôn luôn là 1 bội số của 512 và chúng ta cần ghi nhớ

trong file floppy thì size của nó luôn là 512 cho đến khi tìm hiểu

Gi s n ó b ng 2



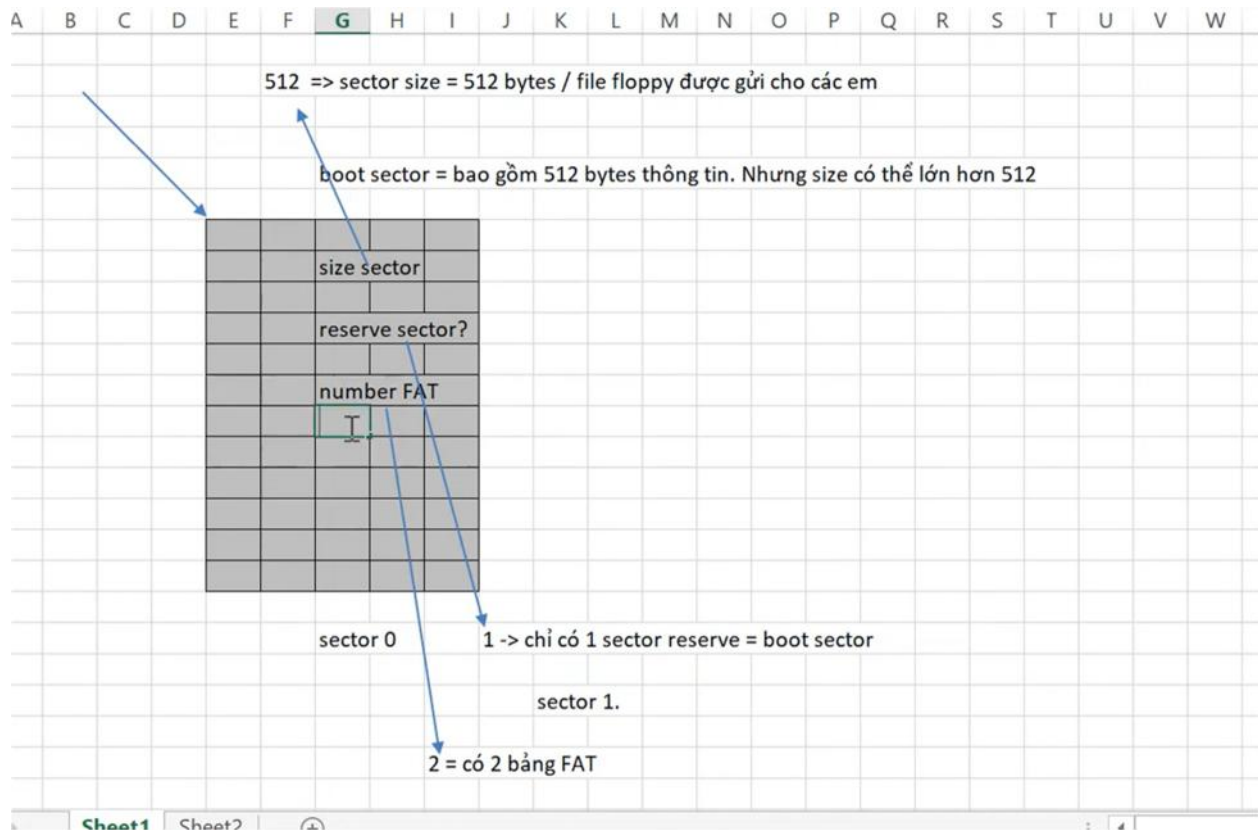
Vì cái file floppy thì c n h n thì nó s là 1 reserve sector nó s là 1 nh v y ch có 1 thôi và nó s không có ph n sector khác n a và ph n này s b i



C n c c n i dung c a boot sector n m c t t c nh ng thông tin này

ang a ra nh ng thông tin c n thì t n m, các thông tin các có th t c nh ng ko có tác d ng nhi u trong vì c phân tích ph n i theo sau cái ph n sector

ây là 1 t c là có 1 boot sector thôi, sau ó s quy nh là number FAT, quy nh là có bao nhiêu b ng FAT, nó có th là 1 b ng FAT hay 2, 3 b ng FAT thì s c cái này và trong file floppy c g i thì nó s quy nh là có 2



T c là 2 s có 2 b ng FAT

2 b ng FAT s i theo sau th ng boot sector, sau ó c nh ng thông tin l y ra c nh ng thông tin c a size of FAT. V i file floppy c g i s c c size of Fat s b ng 9 t ng ng v i vì c 1 b n FAT chỉ m 9 sector => 2 b n FAT chỉ m 18 sector



V y hình dung sau khi c c nh ng thông tin này thì s vi t c

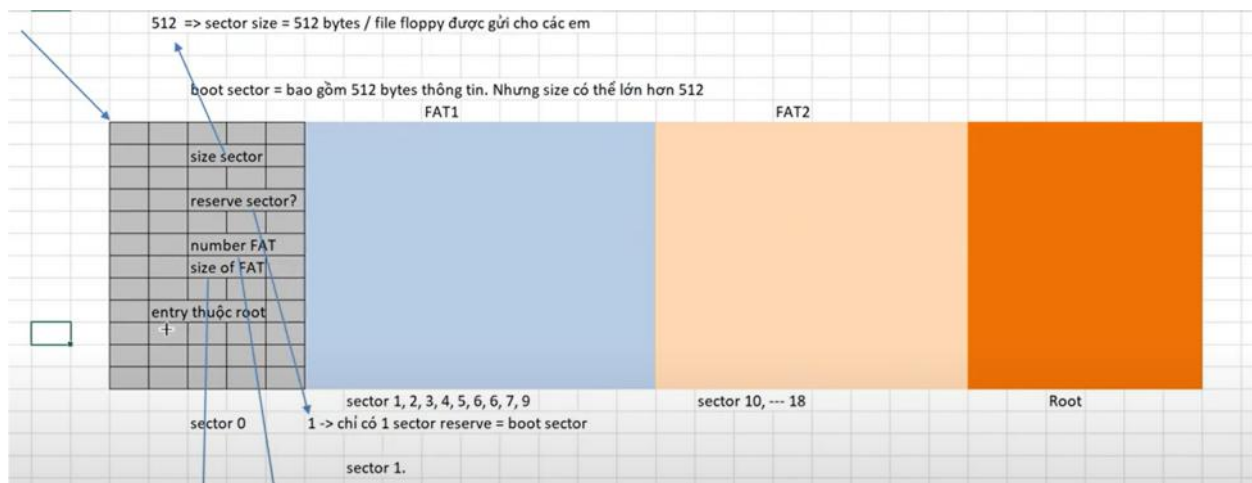
i theo cái này s là 1 b ng FAT

Fat1 v i size là 9 sector thì nó s n m t sector 1,2,3,4,...9 i t sector 1 n 9 cho b ng FAT

T ng t v i b ng FAT2 vì khi c thông tin trong boot sector, ta bi t c là trong c u trúc c a tôi s có 2 lo i FAT

FAT2 c ng chỉ m size là 9 sector, t sector 10 n 18

Sector 19 s là vùng root directory



c trong b ng FAT s bi t c là có bao nhiêu entry thu c

T entry này s tính ra c s l ng .. t cái s entry này

M i entry chỉ m

Khi c thông tin s bi t c m i entry chỉ m 32 bytes

Khi bi t c s entry thì nhân v i 32 và chia cho sector size thì ra c size c a root

L y c size c a vùng root thì file floppy âu ó c t sector 19 n sector 33.

c ph n boot sector phân tích và l y c các v trí offset c a nh ng vùng 32 ho c 33 g i ó.

Sau y s n vùng data là vùng còn l i



Vùng data s b t u t sector 34, do size = 512 byte và trong ph n boot sector s quy nh sector siz. Thông tin này s ch a ta s quy nh 1 cluster chỉ m bao nhiêu sector

V i file floppy c c cluster size = 1 d n n 1 cluster = 1 sector

D n n vùng data b t u t sector th 34

ây là ph n format chung c a file FAT 12 và FAT 16

FAT 32 s ko có vùng root directory mà s c tích h p vào trong vùng data. Khi ó s c ph n boot sector bi t c cái root directory n m âu trong vùng data.

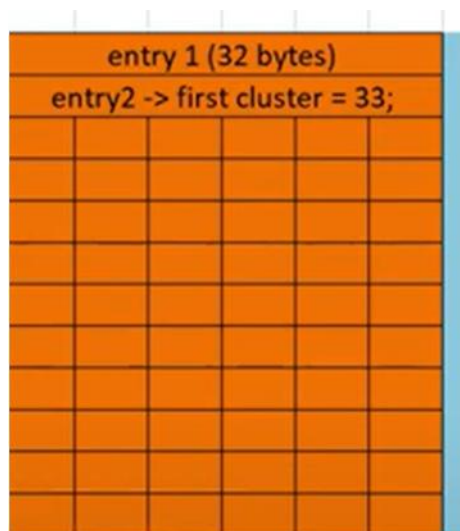
Ph n root directory s ch a nh ng, s chính là các th mà mình nhìn trong D

Ch a nh ng tt nh th này:

Name	Date modified	Type	Size
HaiND1	7/8/2021 4:18 PM	File folder	
Program Files	5/20/2019 6:14 PM	File folder	
PROJECT_2019	1/8/2020 2:50 PM	File folder	
PROJECT_2020	11/24/2020 10:40 PM	File folder	
PROJECT_2021	6/1/2021 8:52 AM	File folder	
Reskill	8/4/2020 8:46 AM	File folder	
Setup	7/15/2021 9:53 AM	File folder	
ToolPath	4/17/2021 6:50 AM	File folder	
haind1.txt	6/29/2021 4:15 PM	Text Document	1 KB

Mỗi cái folder hay 1 file cũng chỉ là 1 entry trong FAT

Mỗi 1 entry chỉ mất 32 bytes, ví dụ đây là entry1, entry 2 cũng 32 byte 1



Trong thông tin 32 byte sẽ có một quy định là first cluster, nó sẽ báo là nó nằm ở đâu trong vùng data, giả sử vùng data nó bắt đầu từ 33 thì nó sẽ

Giả sử entry này là 1 file text: nội dung chứa các ký tự "a" "b" "c"

Chỉ mất 1 cluster trong FAT

Thì các giá trị 33 này không chính xác cho 1 mà nó sẽ có phần offset sẽ bắt đầu từ 2

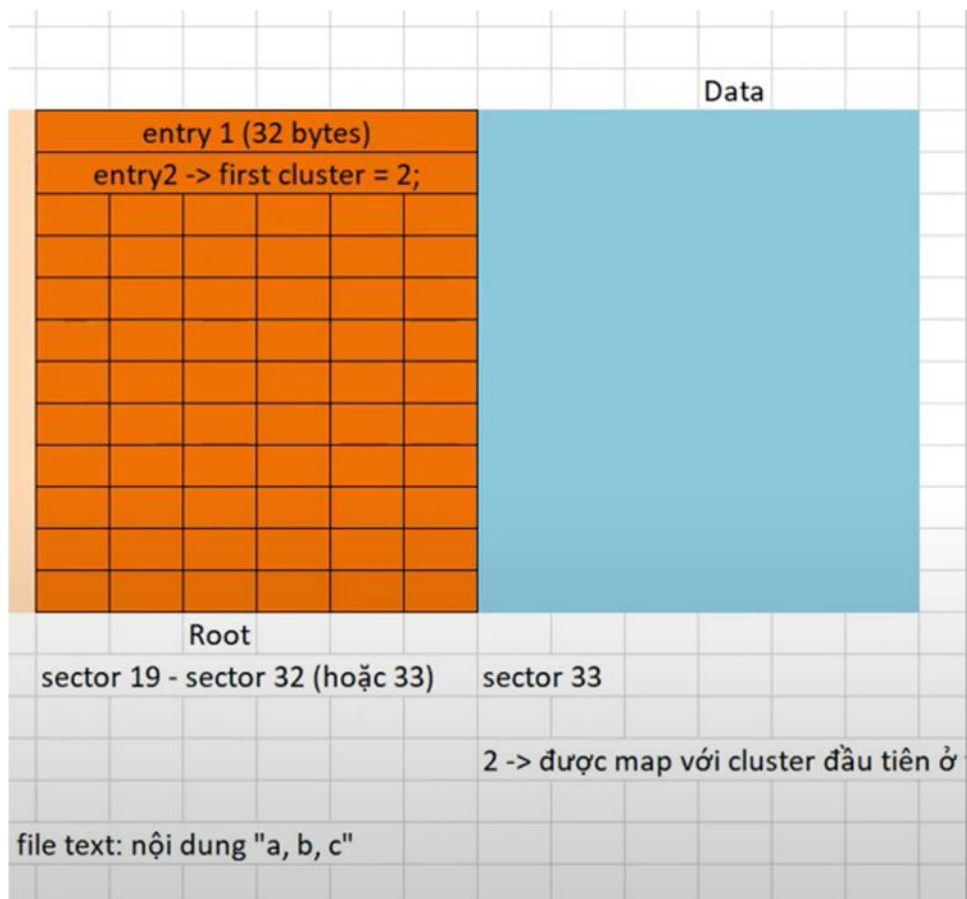
2 đây sẽ là các map về cluster đầu tiên vùng data

Thì sao không có số 0 và 1: vì chúng ta thêm trong FAT system

First cluster sẽ có cách để là phần thứ 2

First cluster sẽ tương ứng với vị trí sector thứ 33

Cluster thứ 2 bắt đầu từ sector thứ 33

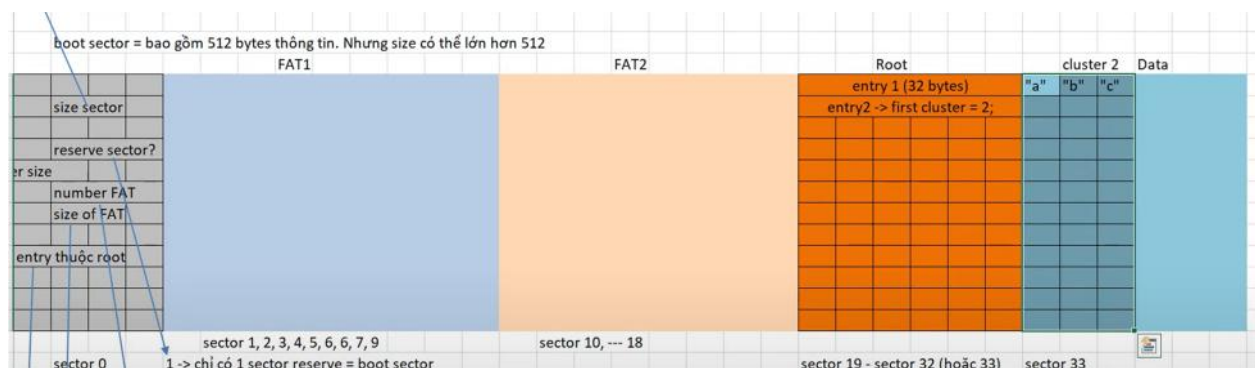


Và nếu cần thêm thông tin về boot sector, về thông tin về bảng FAT 2

Root directory này và vùng data này

Giả sử entry2 là 1 file text, nội dung chứa abc thì

Giả sử đây là cluster đầu tiên của vùng data, cluster số 2 thì mới ở trong vùng cluster đầu tiên nó sẽ chứa ký tự a, ô nhiễm thứ 2 chứa ký tự b và 3 chứa a c



Và toàn bộ phía sau không chứa gì cả tuy nhiên nó sẽ dẫn tới, nó sẽ vẽ nên chi m, cái file text vẽ nên chi m tới các vùng

1 cluster sẽ chỉ m 512 bytes vì 1 cluster = 1 sector, và sector = 512 vì file floppy cũg i

Khi làm việc với usb, khi copy 1 file, gì s usb không dùng bất kỳ 1 phần data nào, copy 1 phần file text có 3 byte abc, khi check dung lượng usb có thể chỉ m tới 1 KB tức là 1024 byte tức là nó đang chỉ m tới 1 cluster, 1 cluster này là 1024 bytes với Usb

Còn file floppy là 512B

Bình thường khi copy chỉ m 3 byte nhưng tại sao lại sẽ dẫn tới 1024 bytes của USB

Cái first cluster chính là cái bit của entry nằm đâu trong vùng data

Gì s có entry thứ 3

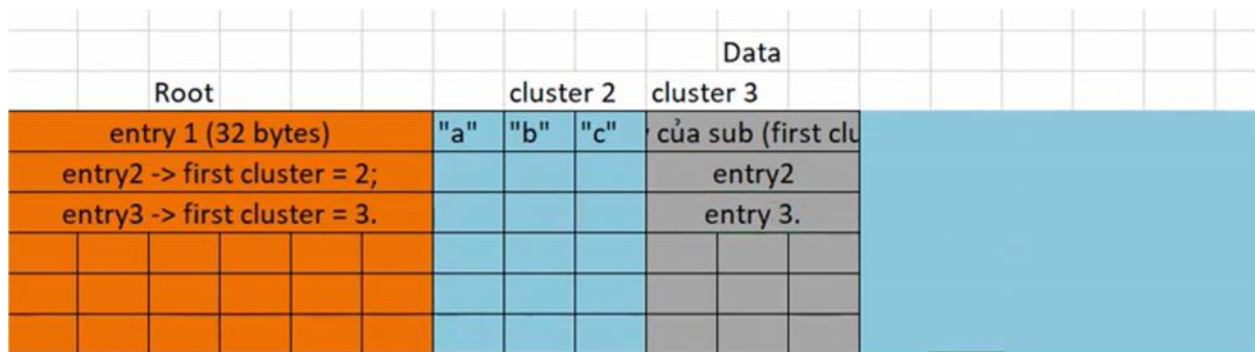
entry 1 (32 bytes)						"a"	"b"	"c"	
entry2 -> first cluster = 2;									
entry3 -> first cluster = 3.									

Vì entry 3 là 1 sub directory thì sao khi nó gì s c trong cluster sẽ 3 ch ng h n



Trong cluster s 3 n i dung g m 32 byte nó s chia thành t ng 32 byte , 32 byte là 1 cái entry c a sub. Gi ng nh vì c khi m 1 trong a thì bên trong 1 sub này thì project 2019 này là 1 cái sub directory c a a D c a root directory này.

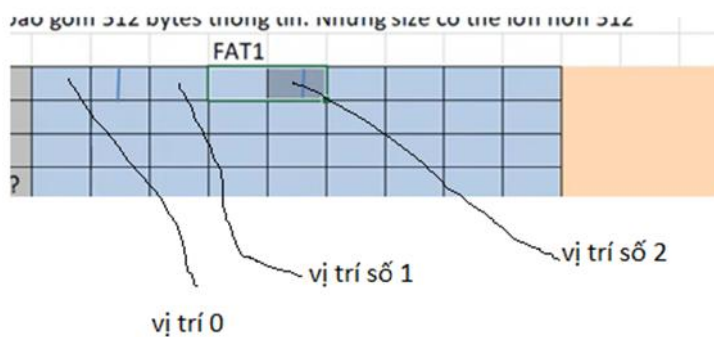
T₁ c là trong sub c₁ ng ch₁ a các cái entry t₁ ng t₁ nh₁ bên ngoài c₁ ng là t₁ ng 32 byte l₁ và nó c₁ ng có cái first cluster₁ quy₁ nh xem là entry₁ y n m₁ âu trong vùng data. T₁ ng t₁ 32 byte ti₁ p theo c₁ ng là entry ti₁ p theo



Ki u nh v y.

Vậy bảng FAT đầu, bảng FAT làm gì, khi dùng lệnh copy file chỉ mìn hìn cluster thì nó sắp xếp như thế nào data tiếp theo nó nằm ở đâu trong vùng data, tức là dùng bảng FAT quy định, gì giống như 1 danh sách liên kết. cái cluster số 2, cái này vào tìm ra xem phần tiếp theo của nó nằm ở đâu. Giờ quay lại bài toán entry số 21 nh 512 byte

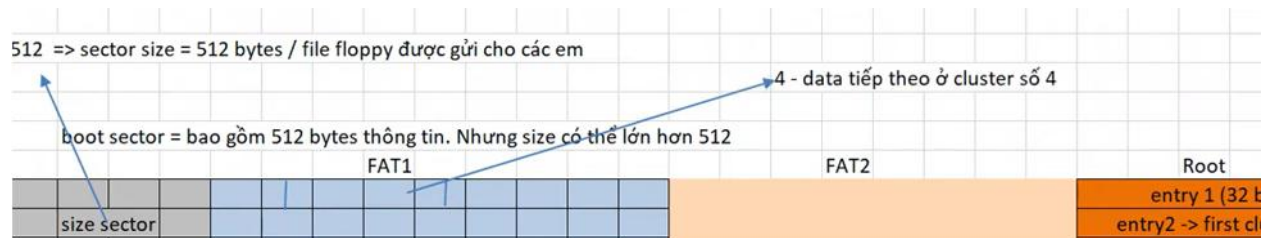
Gi s ây là ký t y g ì g ì ó



Ta cần dùng các vị trí số 2 này thì sẽ lấy giá trị, giá trị nó sẽ

Entry số 2 first cluster bằng 2 thì số 2 là index truy xuất trong bảng FAT cần dùng vị trí số 2 này là 1 số 12 bit tiếp theo

Giờ sẽ có bảng 4 chúng ta thì bảng vị trí là data tiếp theo cluster số 4



Sau khi có các số 4 thì tiếp cluster số 4

Root	Data					
	cluster 2			cluster 3	cluster số 4	
entry 1 (32 bytes)	"a"	"b"	"c"	của sub (first cluster)	y	y
entry2 -> first cluster = 2;	"a"	"b"	"c"	entry2		
entry3 -> first cluster = 3.	"a"	"b"	"c"	entry 3.		
	"a"	"b"	"c"			
	"a"	"b"	"c"			
	"a"	"b"	"c"			
	"a"	"b"	"c"			
	"a"	"b"	"c"			

Thì thấy 3 byte tiếp là entry này là 1 file text abc ... lúc này không phải abc nữa và cuối cùng là 3 ký tự n m cluster số 4. Tiếp vị trí cluster số 4 thì lại quay ra các bảng FAT thì vị trí số 0 1 2 3

Thì vị trí số 4 thôi, cần dùng vị trí số 4 thì nó sẽ ra giá trị 0xFFE -> end of file



Thì nó báo là file này đã kết thúc rồi nó sẽ không còn data nữa

Vậy khi nó kết thúc quá trình cần dùng các entry số 2

Tức là nó chứa nội dung của entry số 2. Bảng fat ghi nhận bằng next của entry số 5

Index của nó là giá trị first, còn giá trị cluster là 1 index rồi truy xuất ra bảng fat, rồi ra nội dung ở r i ... truy xuất phần tiếp theo.

Khi có toàn bộ các entry thì sẽ ra 1 cái list of entry -> menu option cho người dùng chọn

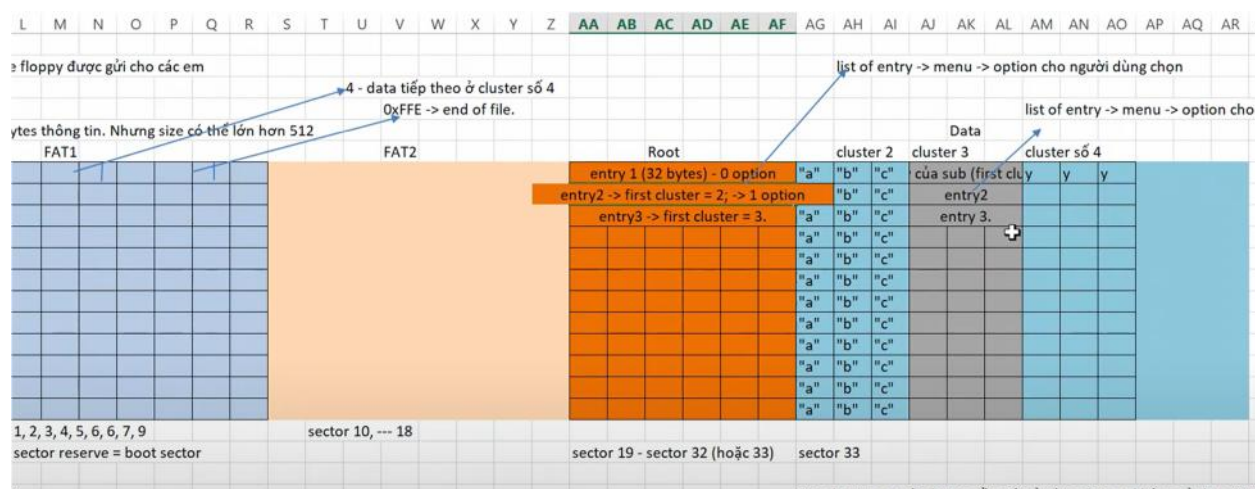
Tương tự như vậy khi có 1 sub directory thì cũng vậy

Đây sẽ tạo ra hình ảnh cái list of entry

thì cái đó tạo ra hình ảnh menu, option trên

ví dụ option này là option 0, sau khi tạo ra danh sách rồi thì tạo option 1: người dùng nhấn 1 thì in nội dung của

nội dung của cluster số 2 và ra cái next, next của số 2 truy xuất ra cluster số 4 và truy xuất ra nội dung của cluster số 4 và ta print toàn bộ thông tin của ô số 2 và 4 này ra sẽ có nội dung của file text tên là option



Tương tự như vậy, đó là cách mà mình sẽ làm bài tập này.

Hỏi: trong cluster này sẽ chia, làm sao phân biệt 1 cái cluster chứa sub directory

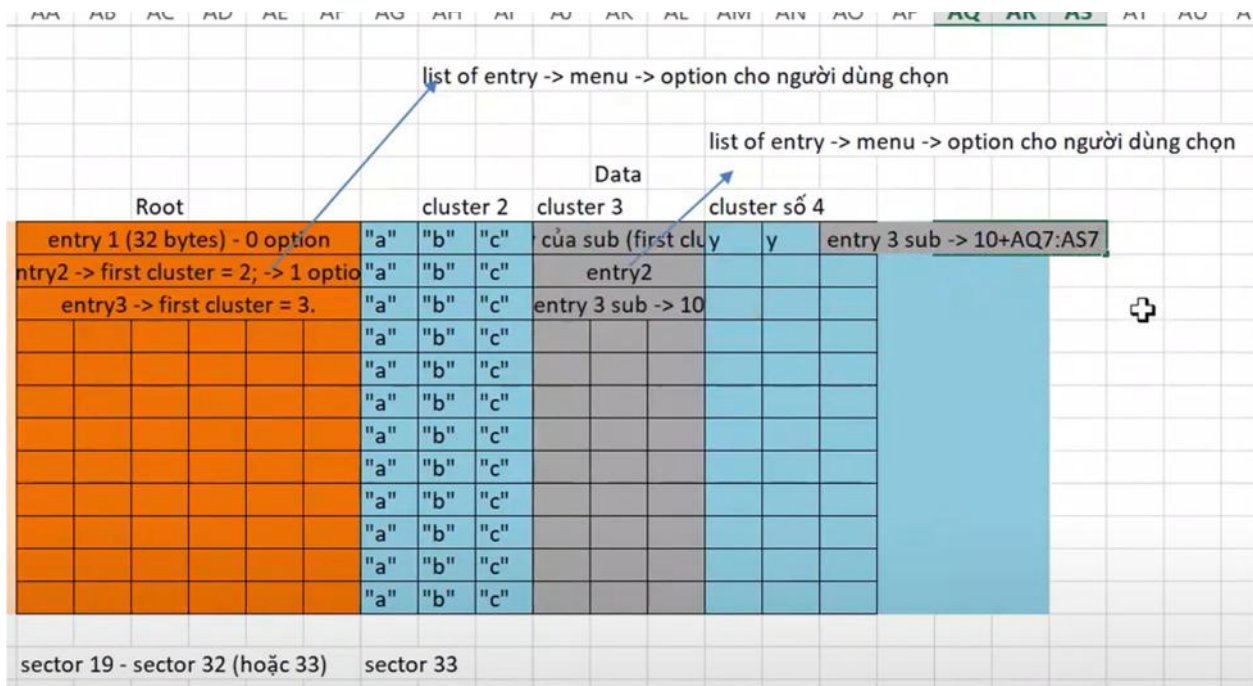
Khi có 32 byte quy định cái đó, 32 byte chỉ nói đến cái first xem chu trình nó có nhận ra nó còn các byte khác có 32 byte nữa

Nội dung của 32 byte sẽ là cái ngày modified, cái entry này ngày xóa ngày thêm, giá trị modified rồi cùng là abcxyz và file này là folder hay file, tức là có 32 byte đó sẽ có những thông tin như vậy.

Đây là phần kiến thức cần tìm hiểu là mất khoảng 1 tháng và làm thì toang

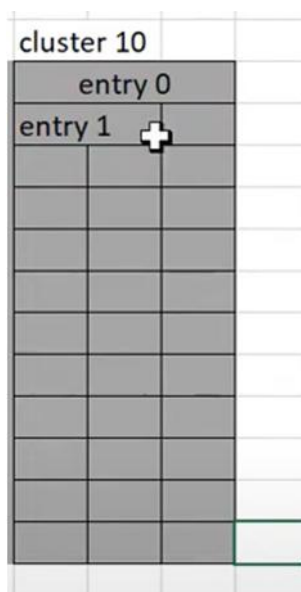
Phần data này là số 2 nhưng phần data tiếp theo có thể là số 4, số 10, nó có thể không liên tiếp. Và mỗi một khối có thể liên tiếp nhau. Còn lý thuyết thì nó truy xuất theo bảng FAT (bảng cấp phát file)

Nếu 1 file trong thư mục còn thì mình phải truy xuất đến những khối cluster 3 nhưng nó là thư mục trong. Nói chung phần format của nó sẽ là từng 32 byte mỗi, mỗi 32 byte là 1 entry tương ứng thì 1 entry là 1 sub khác và 1 nhóm sẽ là 10



Thì mình sẽ nhìn phần cluster số 10, giá trị entry 3 cluster số 10 và nó là 1 sub

Thì khi đó mình cần cluster số 10



Vùng FAT là nó quy nh xem là cái data ti p theo c a cluster y ầu còn root thì nó quy nh là

Làm tròn lên 2K, khi xóa i v n chi m dung l ãng là 1 K 2K nó s ko bao gi t ãng
th a thi u

Cần tìm hiểu

cách phân root dir, như vậy vị trí các thông tin con cháu trong file

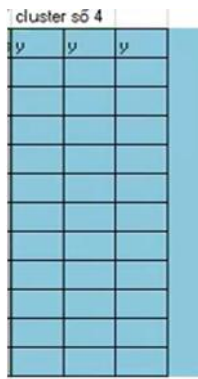
Tìm hiểu về FAT, root, cấu trúc boot sector, cluster

Có thể hiểu trong 32 byte đầu của root directory, byte nào là byte first cluster, byte nào quy định attribute của file

Lưu ý: 2 bảng FAT trên máy tính là backup cho nhau, tức là gì sẽ sinh ra entry số 2 này nó quy định size của file

Cluster số 4 quy định chỉ mất 3 byte của file thì nó tương ứng với cluster số 2 là 512 byte, cluster số 4 là 3 byte thì 515 của file thì trong 32 byte nó sẽ có 4 byte quy định size của file 512 hay 515

Nhưng mà khi kết thúc nội dung cluster số 4, sau này truy xuất vị trí số 4 nhưng nó không ra giá trị end of file (0xFFE) mà nó bằng 8 của file thì nó báo là còn cluster số 8 thì ra gì, ra nội dung phần data tiếp theo. Khi check thì entry chỉ có size 515 byte thì khi đó đây là cuối data rồi



Nhưng mà cái này không phải là end of file tức là đây hình dung ra các FAT bảng 1 -> các bảng FAT thì 2 là bản backup tức là nó hoàn toàn giống với FAT1

Mục đích là khi bảng 1 bị lỗi thì nó truy xuất đến bảng 2 để lấy

Nếu 2 bảng cùng lỗi thì báo FAT bị lỗi và khi này sẽ phải format lại

Vì các bảng 2 này là mình phải xử lý

Phần system bên dưới khi làm việc với bản FAT bảng 1 thì nó sẽ có bản FAT 2 và sau khi bản FAT 2 nó sẽ có việc copy bản FAT này backup về bản FAT 1, tức là bản FAT 2 này là backup cho các trường hợp lỗi thì ra

Nếu bạn 2 byte FAT byte 1 thì các byte treo nên khi truy xuất byte thì thì thông các byte có thể treo hoặc đến tình trạng không thể truy xuất vào các byte sau đây phải format do các byte FAT này bị sai

Khi viết chương trình lúc nào các byte có các trường hợp này, khi làm bài tập thì nên cao minh vẽ, hình dung phần các FAT lúc nào (làm bài) các byte không sai.

Nhưng còn size of sector hay là number FAT khi mà mình boot thêm sao cho linh hoạt

Phần thông tin trong boot sector, không fix các giá trị