***Continut fisier***:

I. Doua intrebari, fiecare avand raspunsuri imediat sub.

II. Modificari aduse proiectului meu fata de cel utilizat in exemplu.

I. Doua intrebari, fiecare avand raspunsuri imediat sub

1. Care sunt efectele/Ce se întâmplă dacă mărimea UA este 32, iar numărul de UA este 4096?

Pentru a definii problema schematic, avem urmatoarele:

- A, cu specificatii:

--UA = 16 bytes

-- 4096 clusters

-- ROOT = 64 entries

- B, cu specificatii:

--UA = 32 bytes

-- 4096 clusters

-- ROOT = 64 entries

Din punct de vedere al spatiului de stocare al mediului de stocare simulat:

Sistemul A are 4096 \* 32 = 65 536 bytes maxim stocati.

Sistemul B are 4096 \* 32 = 131 072 bytes maxim stocati.

Deci sistemul B va putea suporta o cantitate de fisiere mai mare.

Din punct de vedere al FAT:

Tabela FAT are aceeasi dimensiune pentru vectorul lanturilor de alocare de dimensiune 4096, ceea ce se traduce in 4096 de pozitii care ocupa 2 bytes = 8192 bytes fiind dimensiuea pe care fat ul o ocupa.

Sistemul A aloca 512 clustere (UA) pentru FAT : 8192 / 16 = 512

Sistemul B aloca 256 clustere (UA) pentru FAT : 8192 / 32 = 256

Din punct de vere al tabelei ROOT:

ROOT-ul nu are numarul de entry-uri modificat ( 64 entries), insa dimensiunea unui entry, chiar daca ramane la fel, 16 bytes, va ocupa minim un cluster.

Astfel ar fi normal sa marim dimensiunea unui entry la 32 pentru avea limite mai largi pentru nume, extensie si flags ( size si FAU nu o sa castige nimic, au acelaeasi dimensiuni, dimensiuni care se incadreaza in 2 bytes).

Scenariul acesta permite ROOT-ului sa ocupe un spatiu dublu: 64 \* 32 = 2048, in detrimetul scenariului initial care ocupa 64 \* 16 = 1024 bytes.

In alta ordine de idei, dimensiunea tabelei fat se reduce la jumatate, complementar, dimensiunea tabelei root se dubleaza.

Clar sistemul B este mai avantajos, spatiu de stocare mai mult pentru acelasi numar de UA.

Exemple:

Daca dorim stocarea a 2000 de fisiere de 1 - 16 bytes :

B – 2000 clustere cu 32 bytes/UA

A – 2000 clustere cu 16 bytes/UA

Daca dorim insa sa stocam 2000 de fisiere cu dimensiune 17 – 32 bytes :

B – 2000 clustere cu 32 bytes/UA

A – 4000 clustere cu 16 bytes/UA

2. Care sunt efectele/Ce se întâmplă dacă marimea UA este 16, iar numărul de UA este 8192?

Pentru a definii problema schematic, avem urmatoarele:

- A, cu specificatii:

--UA cu 16 bytes

-- 8192 UA

-- ROOT = 64 entries

- B, cu specificatii:

--UA = 16 bytes

-- 4096 clusters

-- ROOT = 64 entries

Din punct de vedere al spatiului de stocare al mediului de stocare simulat:

Sistem A : 8192 \* 16 = 2 \* 4096 \* 16

Sistem B : 4096 \* 16.

Spatiul de stocare este dublu pentru sistemul A

Din punct de vedere al FAT:

Pentru sistemul A FAT ocupa un spatiu dublu, 8192 de pozitii cu dimensiune 2 bytes.

Pentru sistemul B FAT 4096 de pozitii cu dimensiune 2 bytes.

Tabela fat ocupa un spatiu dublue pentru sistemul A fata de B.

Din punct de vere al tabelei ROOT:

Root ul ramane identic petnru ambele, 64 de intrari cu dimensiune 16 bytes - UA.

A se tine cont ca dimensiunea de stocare este mai mare, permitand acelasi numar de fisiere sa fie stocat, 64.

In alta ordine de idei, avem dublu unitati de alocare cu aceeasi dimensiune, deci un spatiu de stocare dublu, iar FAT ul ocupa un spatiu dublu.

Sistemul A este de preferat.

Exemple:

Daca dorim stocarea a 2000 de fisiere de 16 bytes:

B – 2000 clustere cu 16 bytes/UA

A – 2000 clustere cu 16 bytes/UA

Daca dorim insa sa stocam 2000 de fisiere cu dimensiune 32 bytes:

B – 4000 clustere cu 16 bytes/UA

A – 4000 clustere cu 16 bytes/UA

Daca dorim insa sa stocam 8192 de fisiere cu dimensiune 1 bytes:

B – 8000 din 4096 clustere cu 16 bytes/UA

A – 8000 din 8192 clustere cu 16 bytes/UA

II. Modificari aduse proiectului meu fata de cel utilizat in exemplu.

1. Am redus numarul de intrari posibile in ROOT de la 64 -> la 32 pentru a avea dimensiune dubla pentru fiecare intrare pentru ca un char in .net ocupa 2bytes, pentru a explica mai pe larg, a se vedea:

16 octet | 32 octeti

Nume 8 | 18 4 chars -> 9 chars

Extensie 3 | 8 1 char -> 4 chars

Dimensiune 2 | 2

FAU 2 | 2

Flags 1 | 2