1. FFS је један од првих система датотека који је диск третирао као диск, а не као RAM меморију, и тако побољшао перформансе преко просторне локалности.

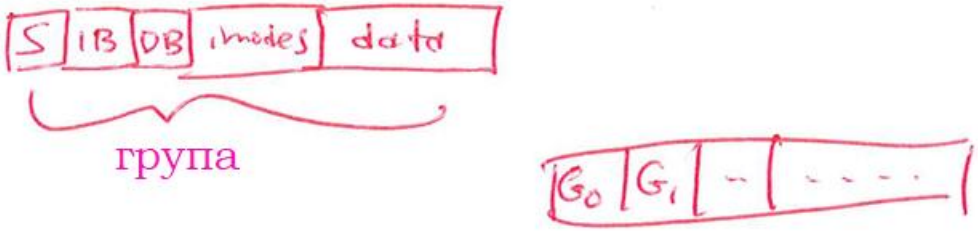
а) Коју структуру на диску користи FFS како би пратио **алокацију** индексних чворова и блокова података?

Мапе битова

б) Зашто је ово било унапређење у односу на класичне UNIX системе?

Стари системи су користили уланчане листе, а то је водило до фрагментације.

в) Нацртајте слику изгледа FFS система датотека. Како се ово разликује од других једноставних система, као нпр. VSFS (Very Simple File System).



г) Узмимо да је максимална брзина диска 100MB/s, а да су времена претраге и ротације, просечно, 20ms. На колике комаде би требало поделити велике датотеке, тако да се остварује 75% максималне пропусности (када се великим датотекама приступа секвенцијално)?

време претраге = 20 милисекунди → ово је 25%

време трансфера = 60 милисекунди → ово је 75%

60ms * 1s/1000ms * 100MB/s = **6MB**

2. Велике датотеке.
Већина система датотека подржава веома велике датотеке. Овде треба да израчунате колико велике датотеке подржавају различити системи датотека. Претпоставите да је, код свих ових питања доле, величина блока 4KB.

а) Рецимо да имамо веома једноставан систем датотека где сваки индексни чвор подржава само десет директних показивача, а сваки од њих може да укаже на један блок. Директни показивачи заузимају 32 бита (тј. 4 бајта). Колика је највећа величина датотеке која може да се смести?

10 * 4KB = **40KB**

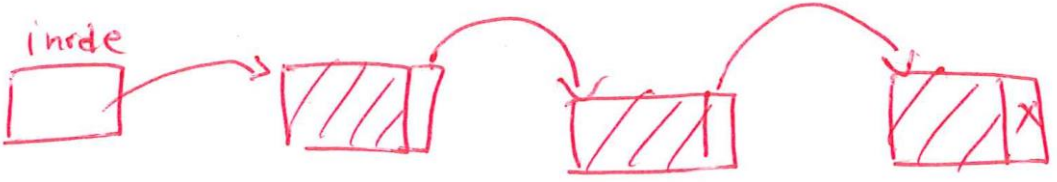
б) Рецимо да имамо систем датотека који има структуру која се назива *опсег*. Опсези имају показиваче (ово је базна адреса) и дужину (изражену у броју блокова). Нека је величина опсега 8 битова (1 бајт). И нека сваки индексни чвор има само један опсег. Колика је највећа величина датотеке која може да се смести?

2^8 * 4KB → 256 * 4KB = **1MB**

в) Нови систем датотека користи директне показиваче, али и индиректне показиваче, као и дупле индиректне показиваче. Показивачи су 32-битни. Колика је највећа величина датотеке која може да се смести? (Не морате потпуно да рачунате, довољан је само поступак.)

[(1024 * 1024) + (1024) + 1] * 4KB ~ **4GB**

г) Рецимо да имамо неки компактан систем датотека код којег се покушава уштедети што више простора у индексном чвору. Чува се само један показивач на датотеке и он показује на први блок датотеке. Али блокови код овог система чувају 4KB корисничких података, као и *next* показивач (како код уланчане листе), тако да показују на следећи блок података (или на NULL, уколико је ту крај). Нацртајте слику индексног чвора и датотеке од 12KB.



д) Колика је највећа величина датотеке која може да се смести у систему датом под г?

2^32 * 4KB → јер је величина блока 2^32

17) Код система датотека налик FFS, више индексних чворова може да указује на исти дескриптор датотеке.

Нетачно, и.ч. не показују на дескрипторе датотека.

18) Код система датотека налик FFS, више индексних чворова може да указује на исту путању датотеке.

Нетачно, путање показују на и.ч, а не обрнуто.

19) Предност меких веза у односу на тврде везе је у томе што меке везе могу да се користе да покажу на директоријуме.

Тачно, тврде везе не могу, а меке могу да се користе за директоријуме.

20) **File-Allocation Table** имаману што допушта велику екстерну фрагментацију.

Нетачно, FAT табела доводи до коришћења страница фиксне величине, а оне пате од интерне, а не екстерне фрагментације.

21) Индексни чворови садрже поље које указује на име дате датотеке.

Нетачно, и.ч. не садрже име датотеке, а разлог је то што више имена датотека може да показује на исти индексни чвор.

22) Структура и.ч. садржи поље које указује на време када је датотеци последњи пут приступано.

Тачно.

23) Структура и.ч. садржи показиваче на податке директоријума.

Тачно, и.ч. садрже показиваче према датотекама или директоријумима; директоријуми су обично смештени баш као и датотеке.

24) FFS покушава да смести индексне чворове и блокове података исте датотеке у оквиру истог цилиндра.

Тачно, на овај начин се постиже добра локалност.

25) FFS покушава да смести индексне чворове датотека из истог директоријума у оквиру истог цилиндра.

Тачно, на овај начин се постиже добра локалност.

26) Код смештања делова великих датотека у оквиру „нове“ групе, FFS тражи групе са натпросечним бројем слободних индексних чворова.

Нетачно, велике датотеке се смештају у новој групи, али се тражи група са мање од просечног броја слободних блокова података јер ће они и бити потребни за смештање велике датотеке (дакле, нису потребни индексни чворови).