

Оперативни системи 2 – вежбе

1. КАРАКТЕРИСТИКЕ ДИСКА. Рецимо да имате диск са следећим карактеристикама:

- Број површина: 16
- Број сектора по цилиндру: 4096
- Број стаза по површини: 512

а) Колико плоча има овај диск?

б) Колико има сектора по стази?

в) Колика је укупна величина (тј. капацитет) овог диска?

2. РАСПОРЕЂИВАЧИ ДИСКА.

а) Рангирајте распоређиваче диска FCFS, SSTF, SCAN и C-SCAN по томе колико су **фер** према захтевима код опслуживања неког насумичног низа захтева за диском; рангирајте их од најмање фер до највише фер распоређивача.

б) Напишите по ком распореду ће следећи низ захтева за неким цилиндром бити опслужен код горенаведених распоређивача. Низ захтева је: 11, 15, 18, 6, 13, 7, 5, 8. Иницијална позиција главе диска је 9 и креће се нагоре (помоћ: ови подаци су битни код неких, а не свих алгоритама). Такође претпоставите да алгоритми SCAN и C-SCAN не иду скроз до ивице диска уколико нема захтева у том делу.

3. **RAID**. Спојте сваки од RAID нива са његовим главним недостатком.

_____ RAID 0	а) Не искоришћава капацитет диска
_____ RAID 1	б) Компликовано рачунање позиције података и парности
_____ RAID 4	в) Отказ једног диска узрокује губитак података
_____ RAID 5	г) Диск парности је уско грло за перформансе

4. **ВЕЗЕ**. Рецимо да имамо неку постојећу датотеку (или директоријум) под именом А и да правимо везу (тврду или симболичку) према новој датотеци (или директоријуму) којој дајемо име В. У табели испод означите знаком + да ли су тврдње односи на тврду везу, симболичку везу, на оба типа веза, или ни на један од ова два типа.

Тврдња	Тврде везе	Симб. везе	Оба типа	Ниједан
Када се креира В, број референци у иноду А се инкрементира.				
Када се измени садржај А, садржај В се такође мења.				
Када се измени садржај В, садржај А се такође мења.				
Када се обрише А (рецимо, преко "rm А"), В и даље постоји.				
Када се обрише В (рецимо, преко "rm В"), А и даље постоји.				
А и В могу да буду директоријуми.				
Овај тип веза може да се креира за прављење циклуса у директоријумској хијерархији.				

5. **АЛОКАЦИЈА И МЕТА-ПОДАЦИ**. Спојте алгоритме стратегија за алокацију простора са метаподацима који су им потребни да би чували податке о томе где се налазе подаци датотеке.

_____	Уланчавање блокова (<i>linked</i>)
_____	Метода индексних блокова (<i>indexed</i>)
_____	Додела континуалног простора (<i>Contiguous</i>)
_____	Везивање региона (<i>Extent-based</i>)

- а) Један показивач на податке датотеке и њену величину
- б) Низ који садржи више показивача и њихове величине
- в) Показивач на први блок датотеке

г) Низ који садржи показиваче на сваки блок датотеке

6. **СИСТЕМ ДАТОТЕКА.** Рецимо да имате систем датотека на UNIX-у код ког се у оквиру инода чува десет директних показивача, један индиректни показивач, један дупло-индиректни показивач, као и један троструко-индиректни показивач. Рецимо да су блокови диска величине 4 KB, и нека показивачи буду величине четири бајта.

а) Која је највећа величина датотеке на овом систему датотека? (Напомена, не морате дати тачн број, само поступак.)

б) Рецимо да је оперативни систем већ прочитао инод ваше датотеке у главну меморију. Колико читања диска је потребно да блок података под бројем 800 учита у меморију? Заокружите тачан одговор.

- једно • два • три • 790 • 800

7. **FFS.**

а) Посматрамо морерне ситеме датотека. Заокружите речи које допуњују реченицу.

- Већина датотека које креирају корисници су: *мале / велике*
- Већина простора у систему датотека се потроши на датотеке које су: *мале / велике*
- У оквиру система датотека, важно је оптимизовати проток за приступања која су: *секвенцијална / насумична*

б) Када је највероватније да ће FFS прећи на нову групу цилиндара? (Заокружите све одговоре који су тачни.)

- Када се алоцира нова датотека.
- када се алоцира први блок података у оквиру датотеке.
- Након алоцирања првих 48 KB датотеке.
- Када се алоцира нови директоријум.

в) FFS решава проблем који је постојао на ранијим системима који су користили неорганизоване листе слободних блокова. Како? (Један тачн одговор.)

- Периодично реорганизује листе слободних тако да су оне сортиране.
- Убацује сваки од блокова података по тачно сортираном распоред у оквиру листе слободних.
- Организује листу слободних блокова као мапу битова слободних блокова.
- Користи бинарно стабло да укаже на узастобне регионе слободних блокова.