

Interfata medii stocare

put(address, data)
get(address)



Sisteme de Fisiere

SO Curs 10



Trei abstractii fundamentale

Memorie virtuală

Iluzia unei memorii continue

Procese

Unitate de executie

Sistem de Fisier
Organizația și ordonarea informației
Ofere o interfață facilă și intuitivă
• trimitere
• achiziție (cumpărare, închiriere, înzinsă)
Accesare în linie cu orice terminal disponibil
de către noi

Cine este un fisier în UNIX
Fisier este un fișier de date și informații
• include informații cu privire la legăturile externe ale acestui fișier
• poate fi folosit pentru a crea, actualiza sau să se șteargă
• poate fi citit
• poate fi editat
• poate fi săzgătuit

Sisteme de Fisiere

SO Curs 10



Concluzii
Fisierele reprezintă unitățile de stocare a informației.
Sistemele de fisier oferă o interfață comună
utilizatorilor pentru a lucra cu fisier și directăru.
Operari principale:
• gestionarea datelor ce blocaj sunt într-un fisier?
• organizarea terenură directăruile sunt fisier
• Generarea spațiului liber

fisiere

Orice este un fisier in UNIX

Fisierul este unitatea de stocare a informatiei:

- date
- metadate (infonu mai exista fragmentare externa este necesar doar primul bloc pentru a localiza fisierul matii despre date)

Ofera:

- persistenta datelor
- partajarea datelor
- protectia datelor

Sistem de fisiere

Organizeaza datele pe un suport persistent
Ofera o interfata simpla utilizatorului:

- fisiere, directoare, ierarhie
- atribut (drepturi, nume, timpi acces)

Ascunde utilizatorilor detaliile dispozitivelor de stocare

Sisteme de Fisiere

SO Curs 10

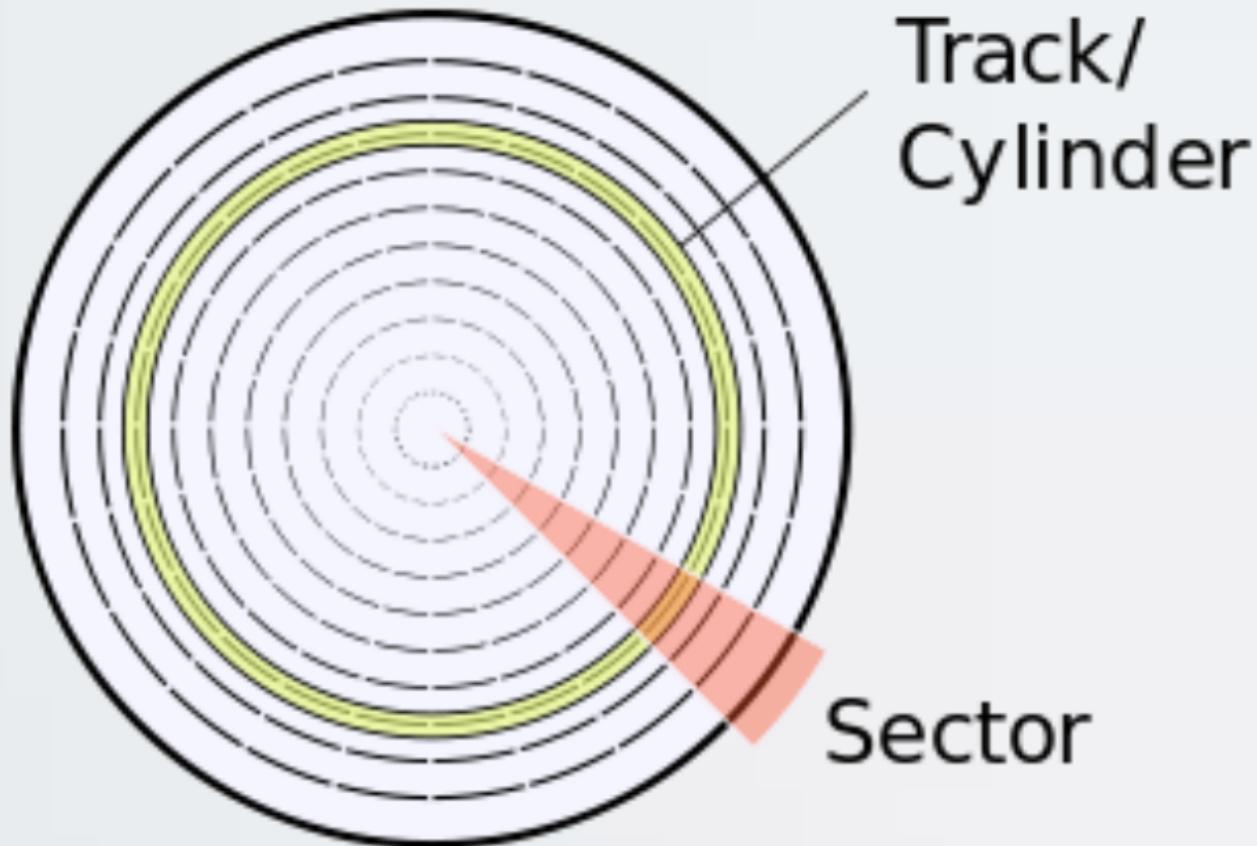


Interfata medii stocare

put(address, data)
get(address)

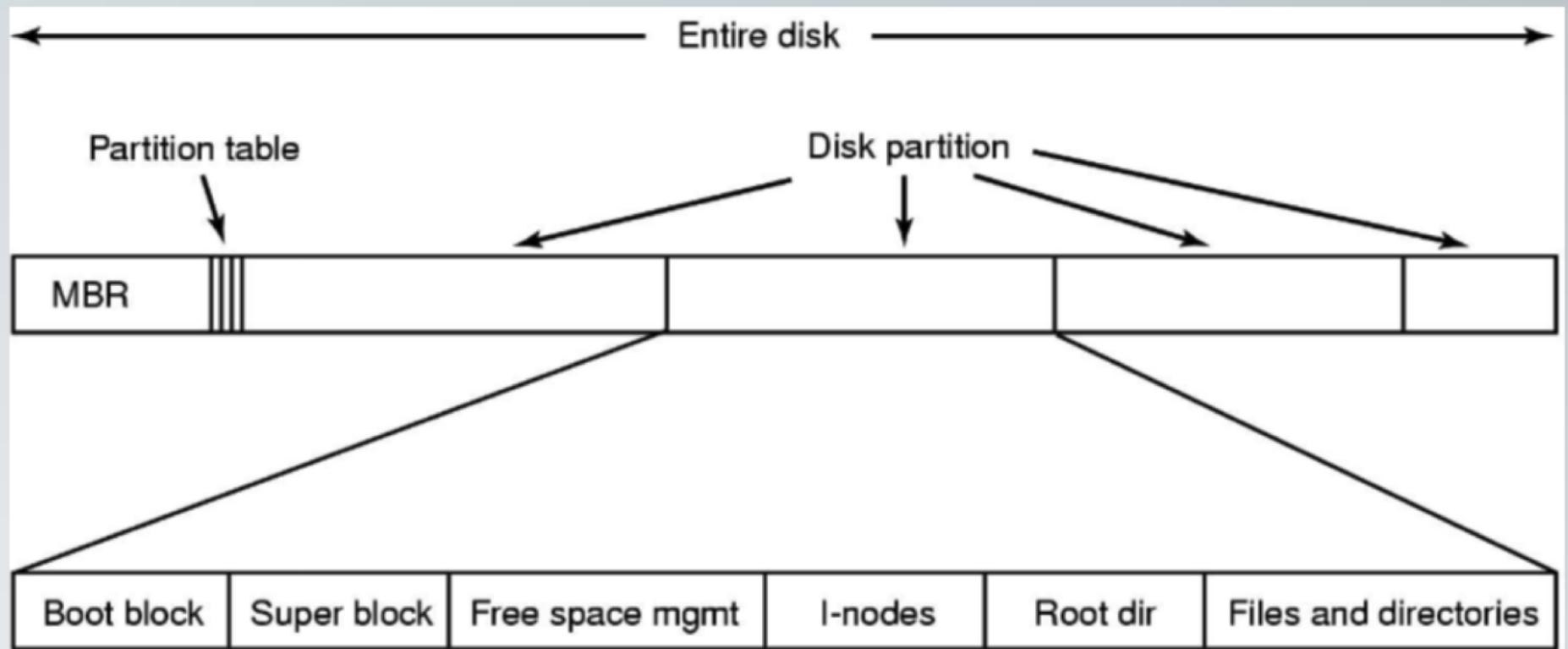






Heads
8 Heads,
4 Platters

Organizarea unui Hard Disc Drive



Alocarea spatiului pe disc

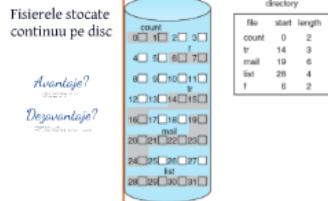
Cum tinem minte unde sunt datele fiecarui fisier?

Alocarea se face la nivel de bloc, care este:

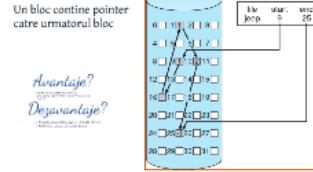
- multiplu de sector (512B)
- submultiplu de pagina (4096B)

Solutii posibile

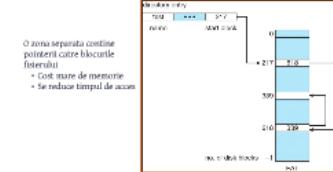
Alocare Contigua



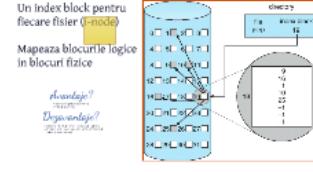
Alocare cu liste



File Allocation Table



Alocare indexata



Alocare Contigua

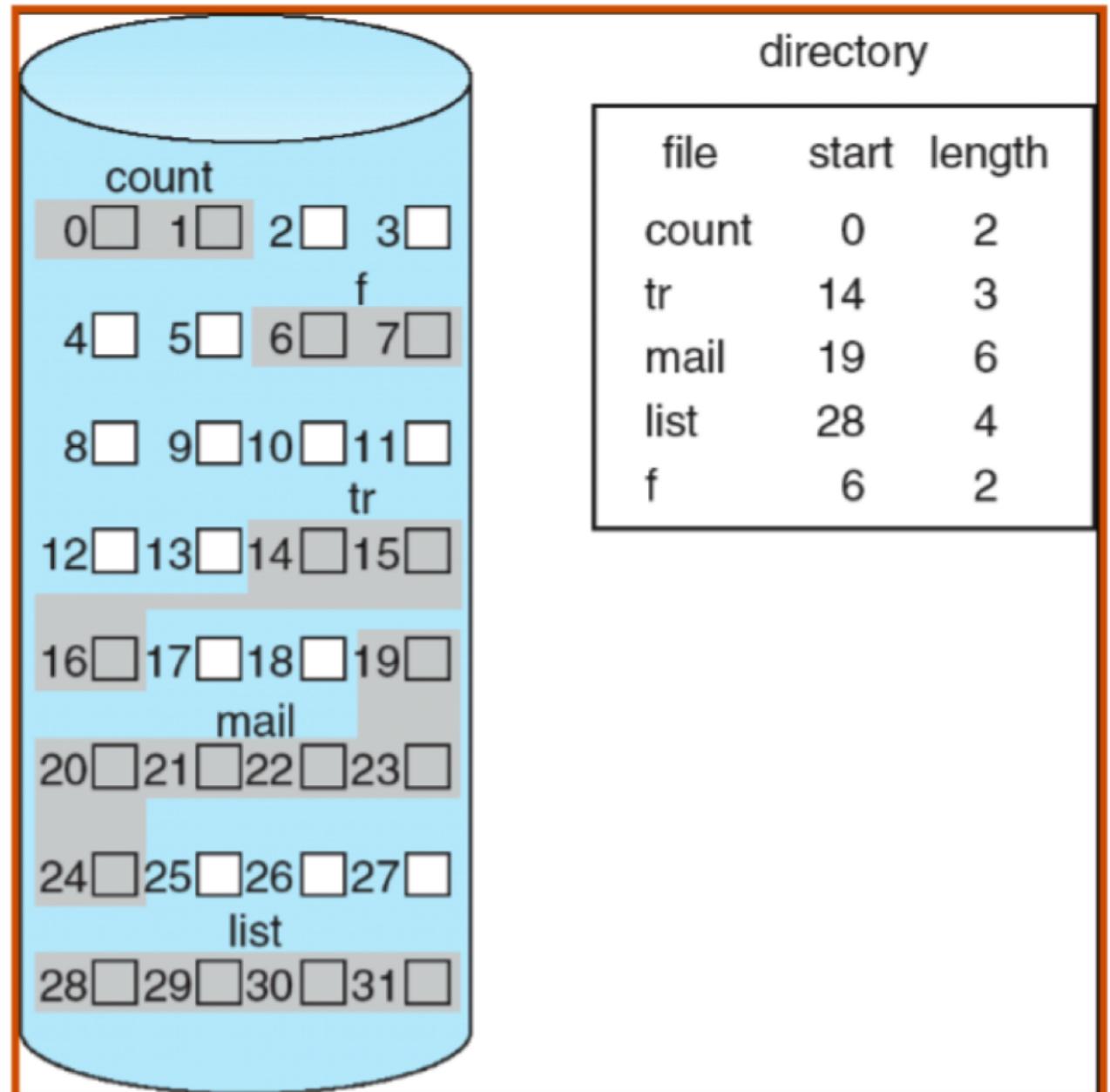
Fisierele stocate
continuu pe disc

Avantaje?

- Identificare simplă a blocurilor
- Viteză mare la citire

Dezavantaje?

- Fragmentare externă
- Trebuie specificată dimensiunea fisierului la creare



Avantaje?

- Identificare simplă a blocurilor
- Viteza mare la citire

- Identificare simplă a blocurilor
- Viteză mare la citire

Dezavantaje?

- Fragmentare externă
- Trebuie specificată dimensiunea fisierului la creare

Alocare cu liste

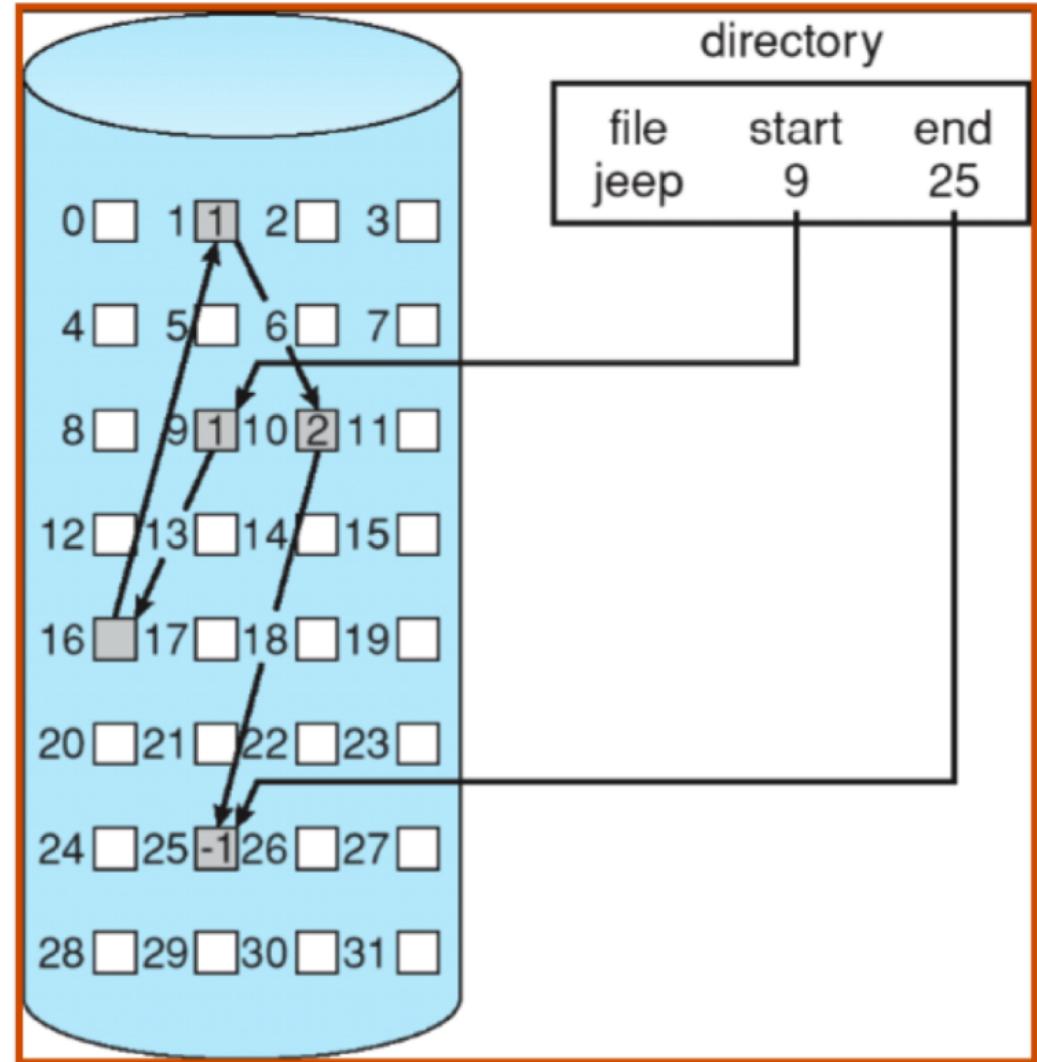
Un bloc contine pointer catre urmatorul bloc

Avantaje?

- nu mai există fragmentare externă
- este necesar doar primul bloc pentru a localiza fisierul

Dezavantaje?

- Timp de acces ridicat pentru ultimele blocuri
- Performanța scăzută la acces aleator



Avantaje?

- nu mai există fragmentare externă
- este necesar doar primul bloc pentru a localiza fisierul

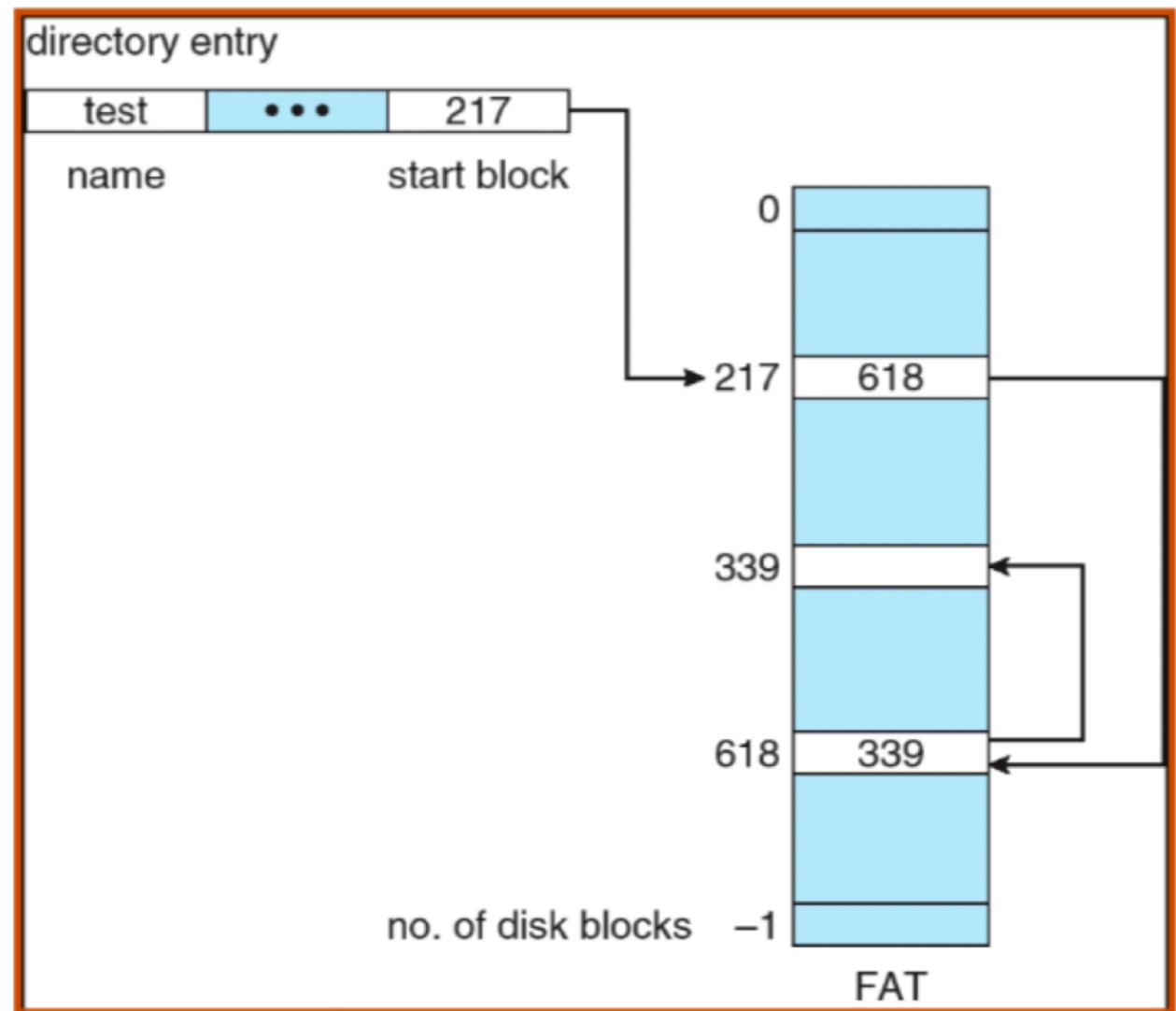
Dezavantaje?

- Timp de access ridicat pentru ultimele blocuri
- Performanta scăzuta la acces aleator

File Allocation Table

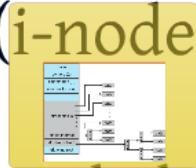
O zona separata contine pointerii catre blocurile fisierului

- Cost mare de memorie
- Se reduce timpul de acces



Alocare indexata

Un index block pentru fiecare fisier (i-node)



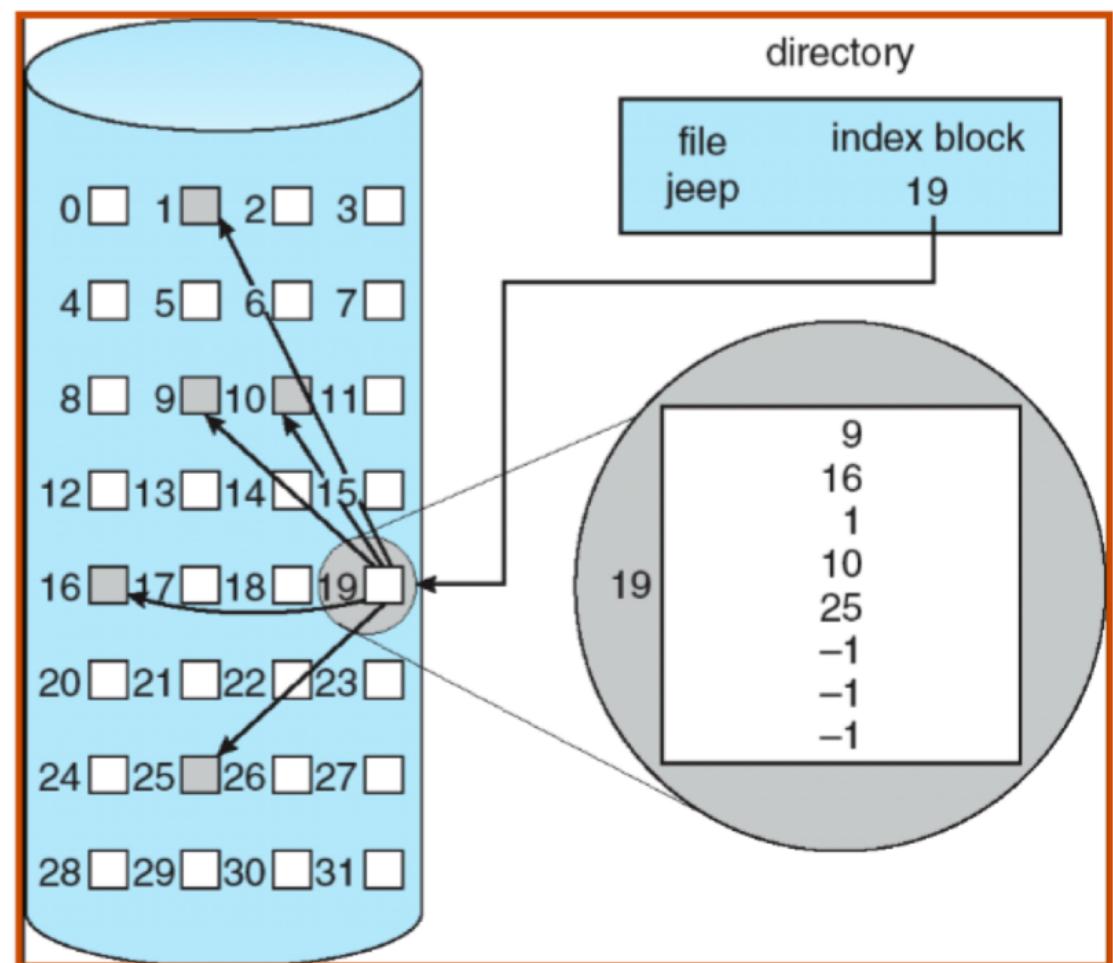
Mapeaza blocurile logice in blocuri fizice

Avantaje?

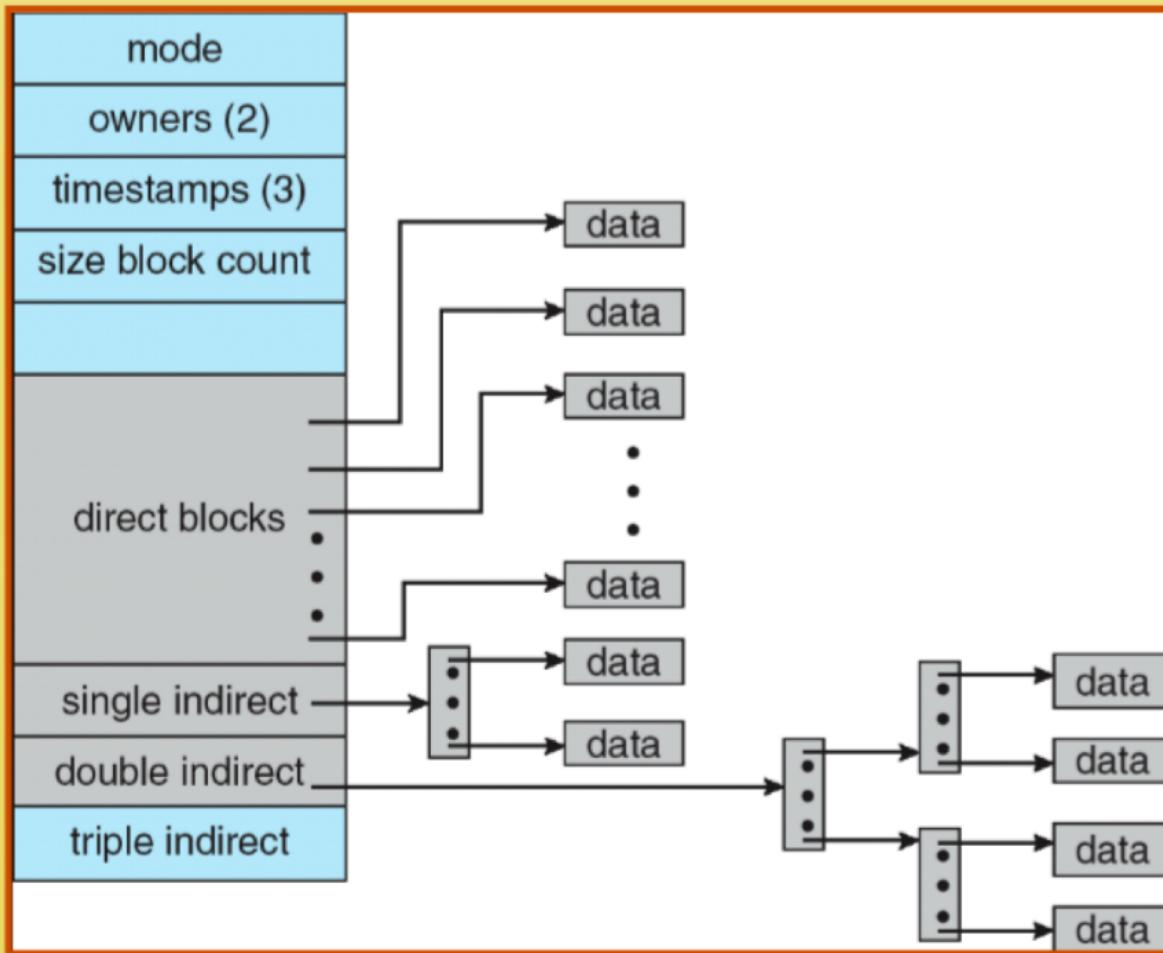
- nu există fragmentare externe
- timp de acces bun

Dezavantaje?

- numărul limitat de intrări în tabela de indexare
- ultimele intrări conțin pointeri către alte tabele de indecsi (crește latenta)

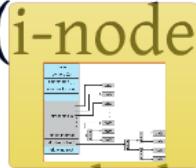


(i-node)



Alocare indexata

Un index block pentru fiecare fisier (i-node)



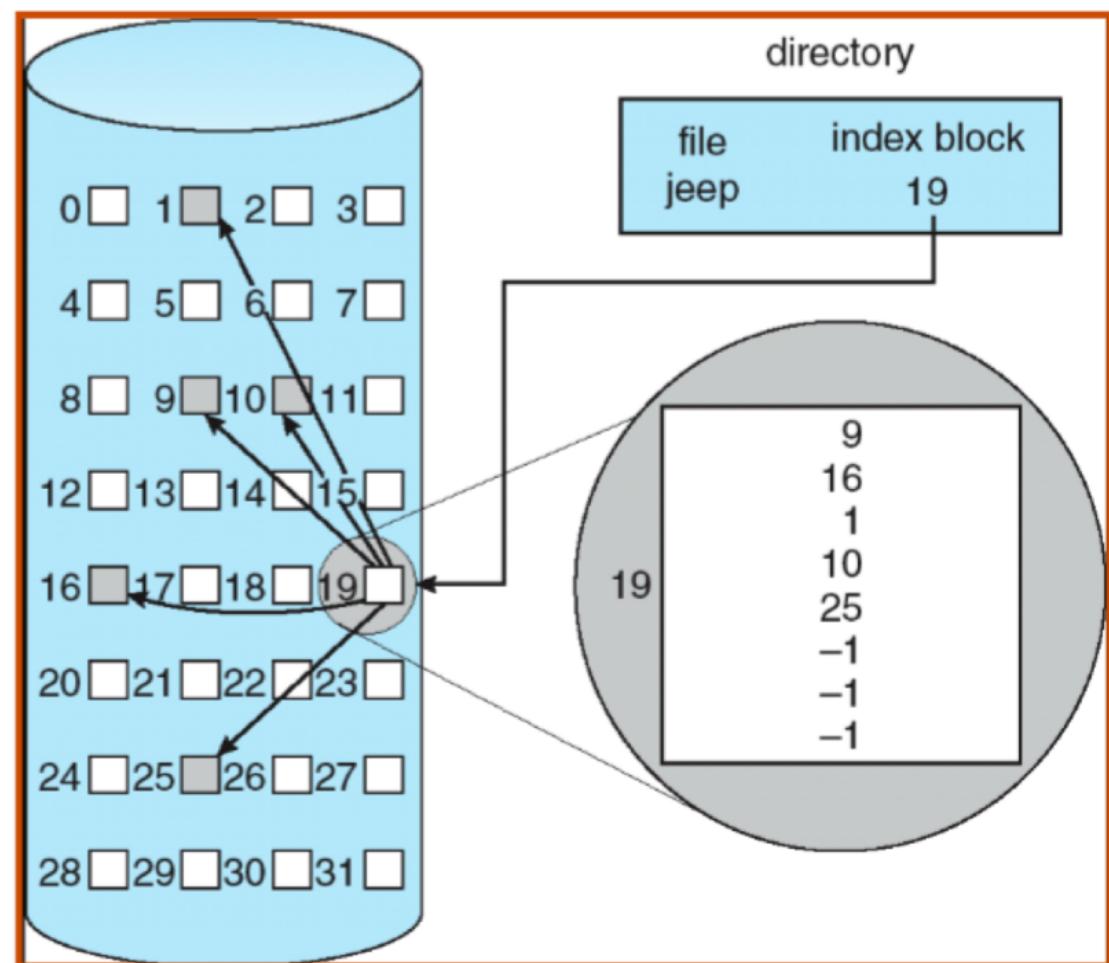
Mapeaza blocurile logice in blocuri fizice

Avantaje?

- nu există fragmentare externe
- timp de acces bun

Dezavantaje?

- numărul limitat de intrări în tabela de indexare
- ultimele intrări conțin pointeri către alte tabele de indecsi (crește latenta)



Avantaje?

- nu există fragmentare externă
- timp de acces bun

Dezavantaje?

- numarul limitat de intrari in tabela de indexare
- ultimele intrari contin pointeri catre alte tabele de indecsi (creste latenta)

Alocarea spatiului pe disc

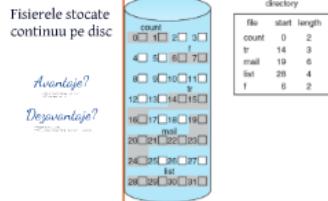
Cum tinem minte unde sunt datele fiecarui fisier?

Alocarea se face la nivel de bloc, care este:

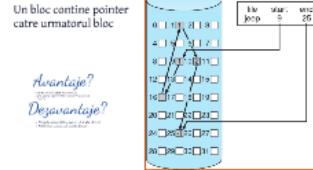
- multiplu de sector (512B)
- submultiplu de pagina (4096B)

Solutii posibile

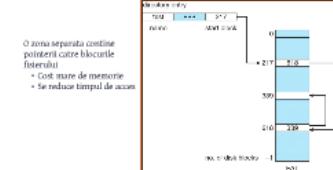
Alocare Contigua



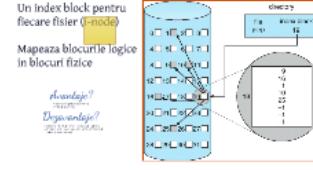
Alocare cu liste



File Allocation Table



Alocare indexata



Implementarea directoarelor

O structura dentry descrie un fisier:

- nume fisier
- un identificator al file-control block (i-node)

Directorul este un fisier special care contine informatii despre alte fisiere

- Directorul are deasemenea un FCB (i-node, intrare in FAT)
- Intrarile . si .. sunt speciale
- Dimensiunea este data de structurile continute

Exemplu: ext2

	inode	rec_len	name_len	file_type	name
0	21	12	1	2	.
12	22	12	2	2	..
24	53	16	5	2	h o m e
40	67	28	3	2	u s r
52	0	16	7	1	o l d f i l e
68	34	12	4	2	s b i n

Link-uri

Permit utilizatorilor sa refere acelasi fisier din mai multe directoare/cu mai multe nume

Hard-links

Doua structuri dentry care refera acelasi i-node
Pot fi folosite intre sisteme de fisiere diferite?

Soft-links

i-node specializat, contine numele fisierului referit

Exemplu: ext2

	inode	rec_len	file_type	name_len	name						
0	21	12	1	2	.	\0	\0	\0			
12	22	12	2	2	.	.	\0	\0			
24	53	16	5	2	h	o	m	e	1	\0	\0
40	67	28	3	2	u	s	r	\0			
52	0	16	7	1	o	1	d	f	i	1	e
68	34	12	4	2	s	b	i	n			

Link-uri

Permit utilizatorilor sa refere acelasi fisier din mai multe directoare/cu mai multe nume

Hard-links

Doua structuri dentry care refera acelasi i-node
Pot fi folosite intre sisteme de fisiere diferite?

Soft-links

i-node specializat, contine numele fisierului referit

Gestiunea spatiului liber

Care blocuri sunt libere?

Vector de biti

1 bit pentru fiecare bloc
overhead constant

Liste inlantuite

Primul bloc liber are pointer catre al doilea

Capatul listei tinut in memorie
Optimizari:

- tabele de blocuri libere
- zone contigue

Considerente de Performanta

Discul este mult mai lent decat memoria:

- 4Gb/s interfata SATA (2Gb/s in practica)
- 100Gb/s memoria RAM

Timp de acces:

- 10ms (disc)
- 1ns (memorie)

Aplicatiile pot citi:

- sequential sau aleator
- mai multe sau mai putine date

Cum imbunatatim performantele sistemelor de fisiere?



Tehnici folosite	
cache	Evita accesul la disc
pre-fetch	Optimizare acces secvential
grupeaza cereri	Reducere timp de seek

Ordonarea cererilor

- pentru a creste throughput-ul
- a reduce timpul de seek
- a minimiza inechitatea



Sisteme de Fisiere Log-Structured

- Scrie totul secvential
- Tine minte unde este cea mai recentă copie pentru fiecare bloc
- În paralel, sterge informațiile inutile

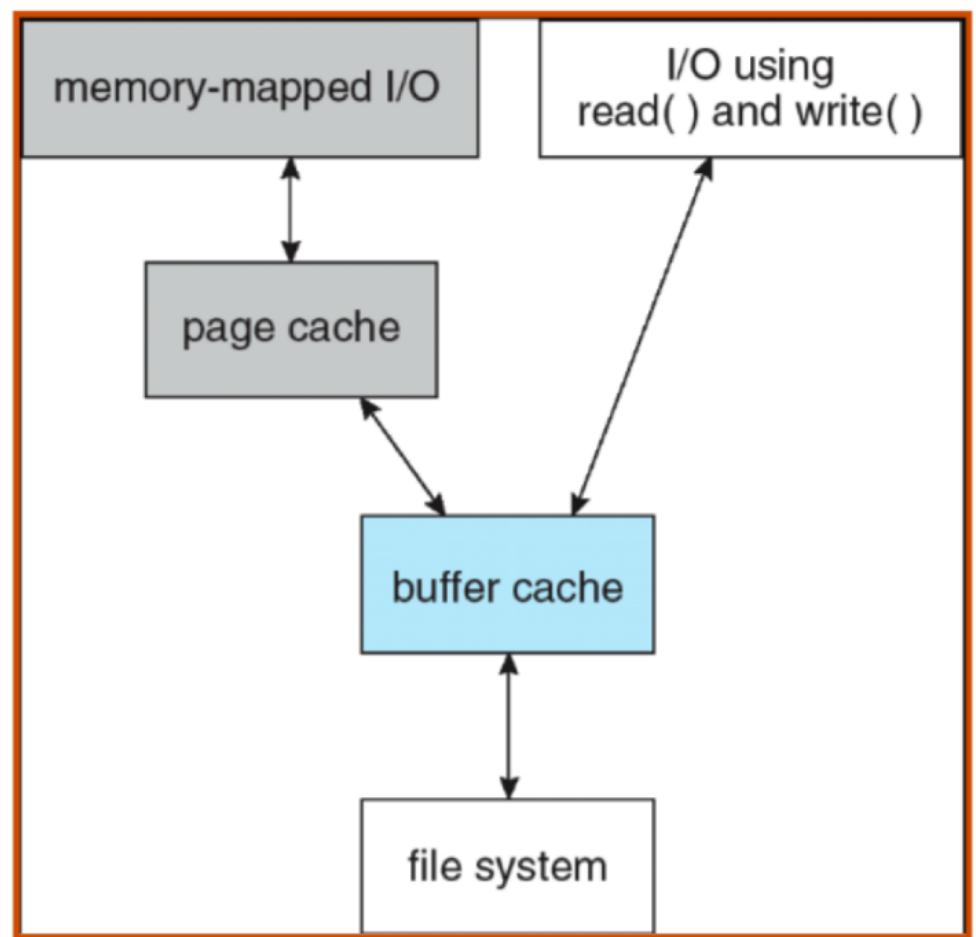
Cache

Buffer cache

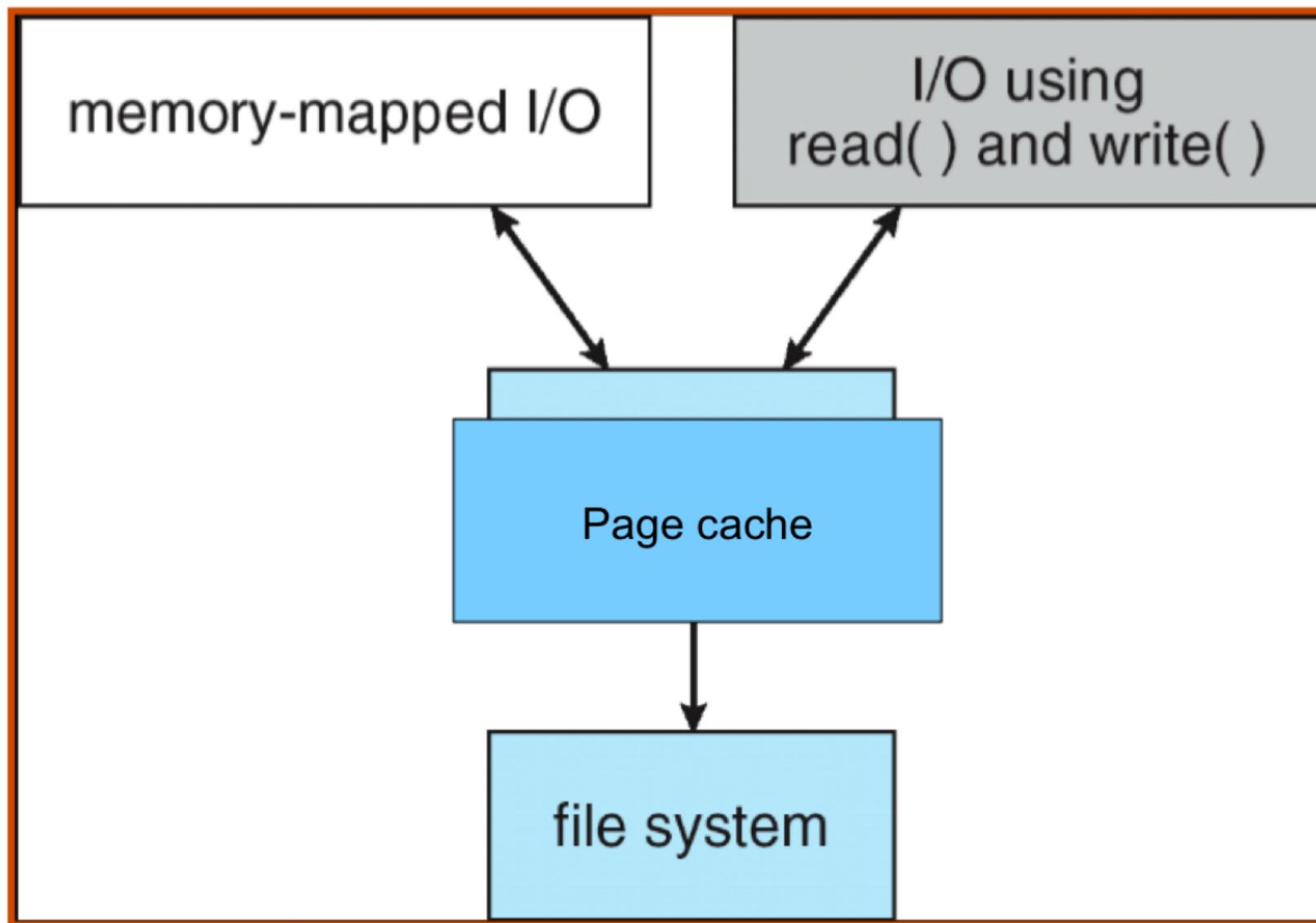
- Mentine in memorie blocurile recent citite de pe disc
- foloseste tabele Hash
- algoritmi de inlocuire: LRU, FIFO

Page cache

- Foloseste pagini in loc de blocuri
- Memory-mapped I/O



Cache unification



Tehnici folosite	
cache	Evita accesul la disc
pre-fetch	Optimizare acces secvential
grupeaza	Reducere timp de seek
cereri	

Ordonarea cererilor

- pentru a creste throughput-ul
- a reduce timpul de seek
- a minimiza inechitatea

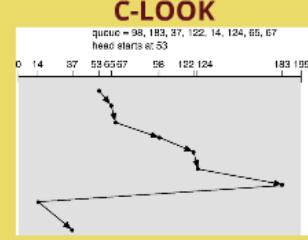
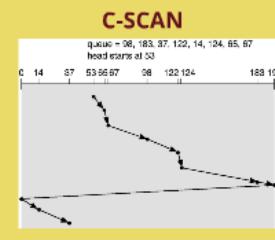
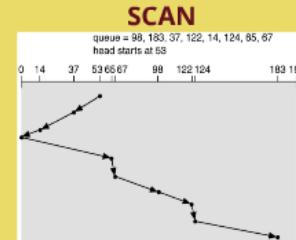
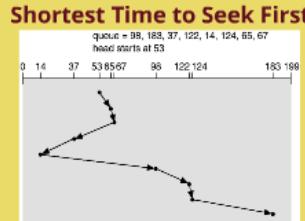
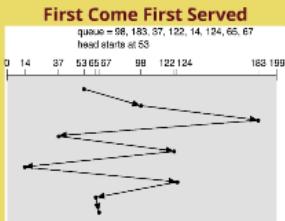


Sisteme de Fisiere Log-Structured

- Scrie totul secvential
- Tine minte unde este cea mai recentă copie pentru fiecare bloc
- În paralel, sterge informațiile inutile

Ordonarea cererilor

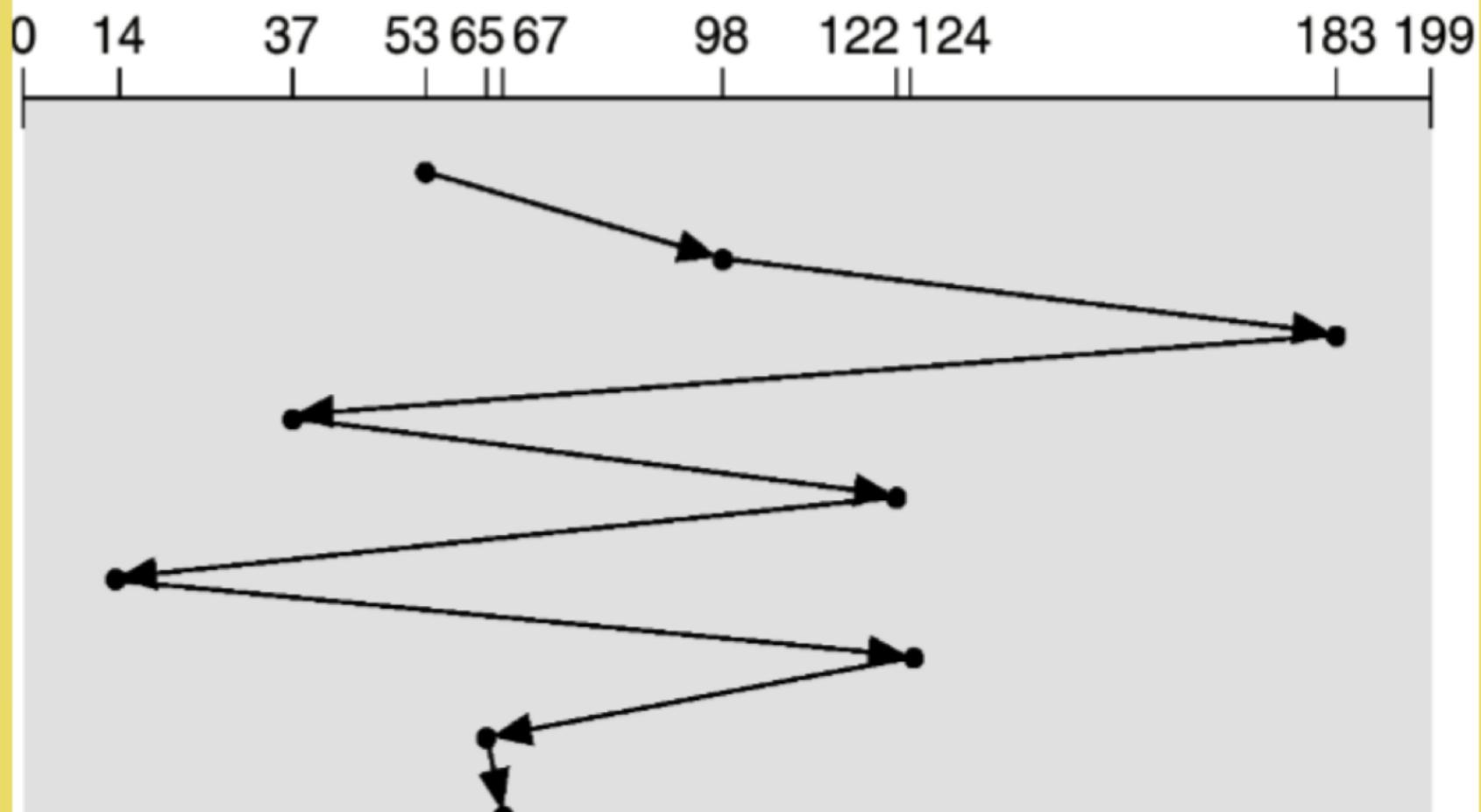
- pentru a creste throughput-ul
- a reduce timpul de seek
- a minimiza inechitatea



First Come First Served

queue = 98, 183, 37, 122, 14, 124, 65, 67

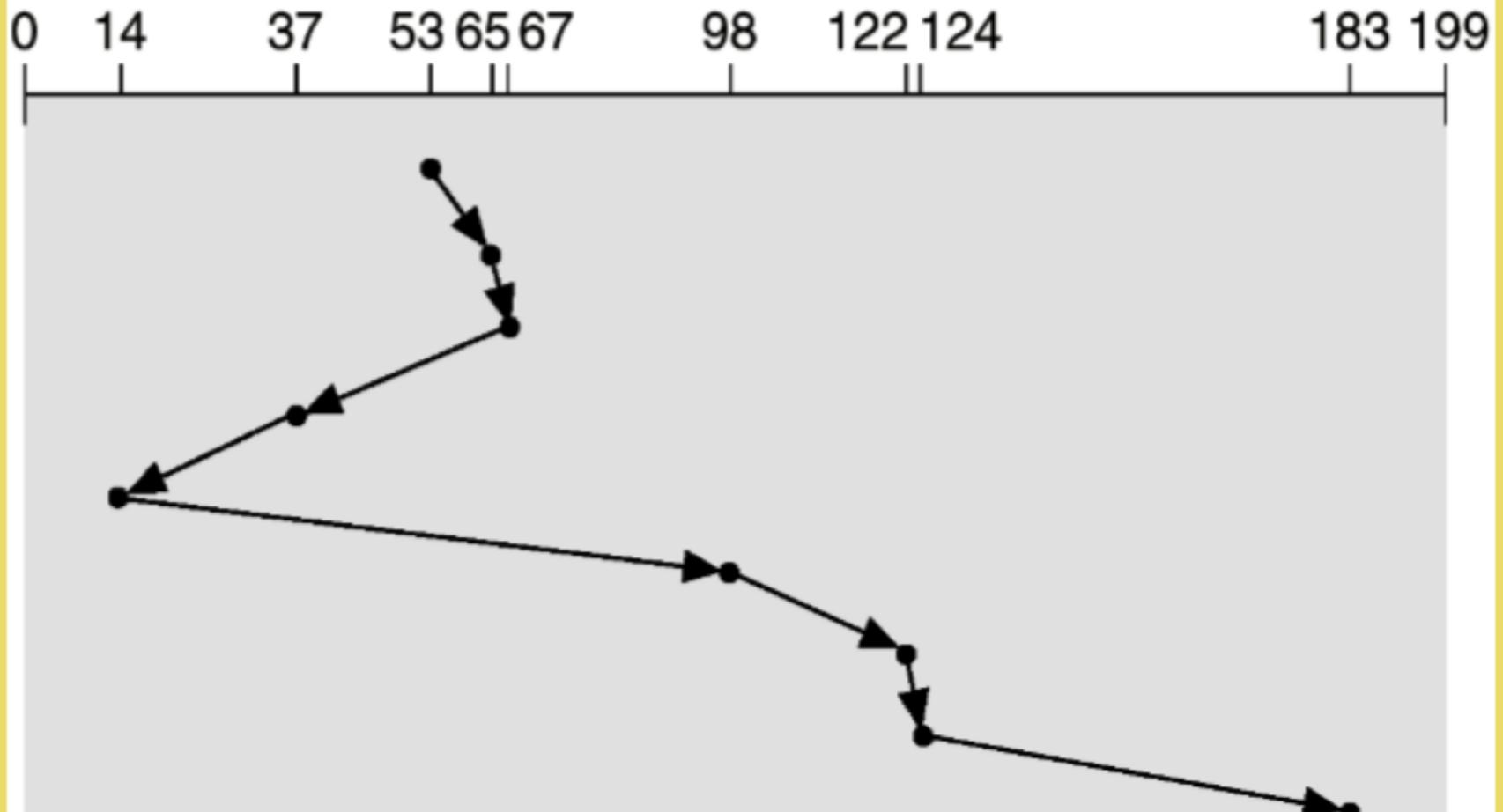
head starts at 53



Shortest Time to Seek First

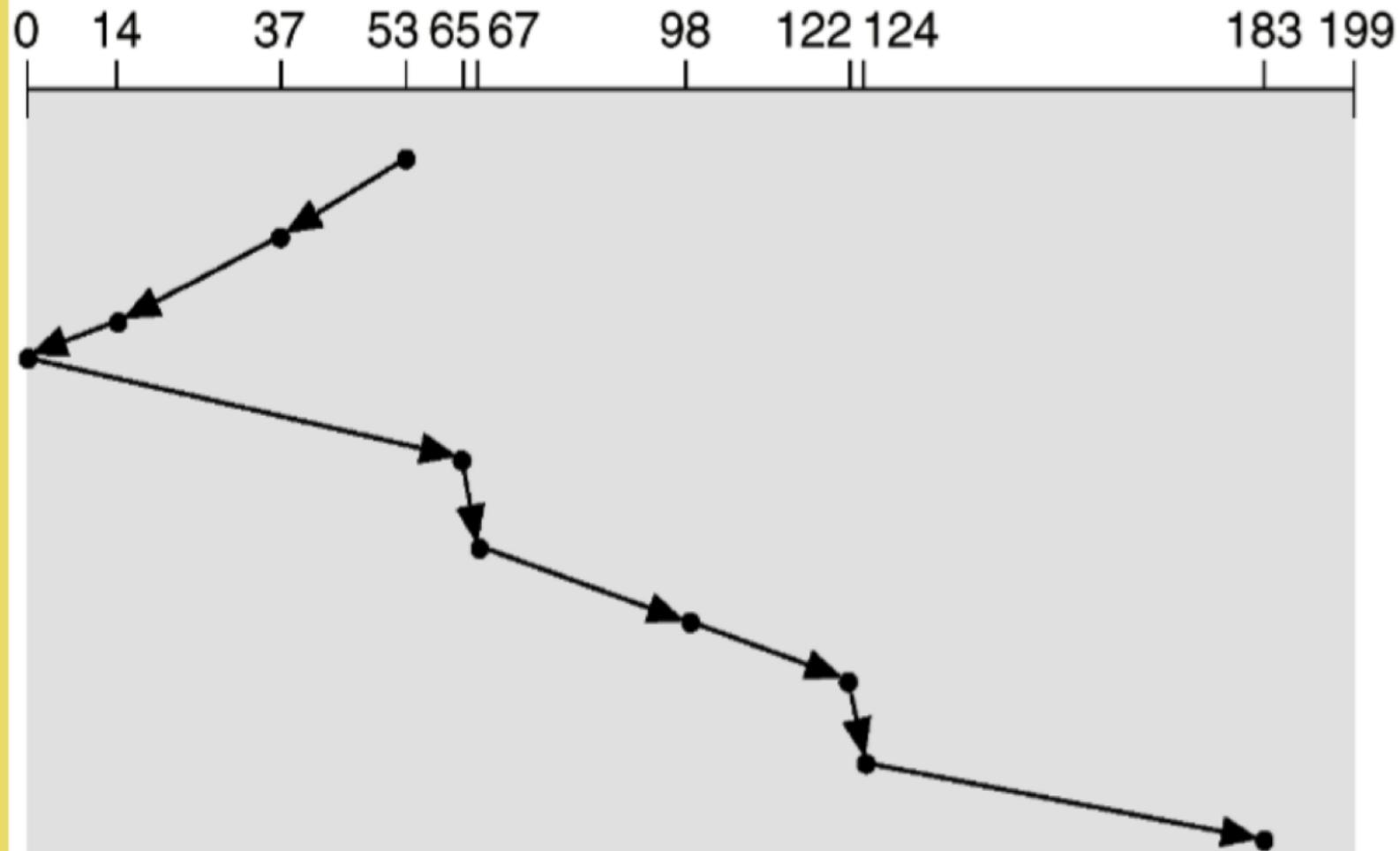
queue = 98, 183, 37, 122, 14, 124, 65, 67

head starts at 53



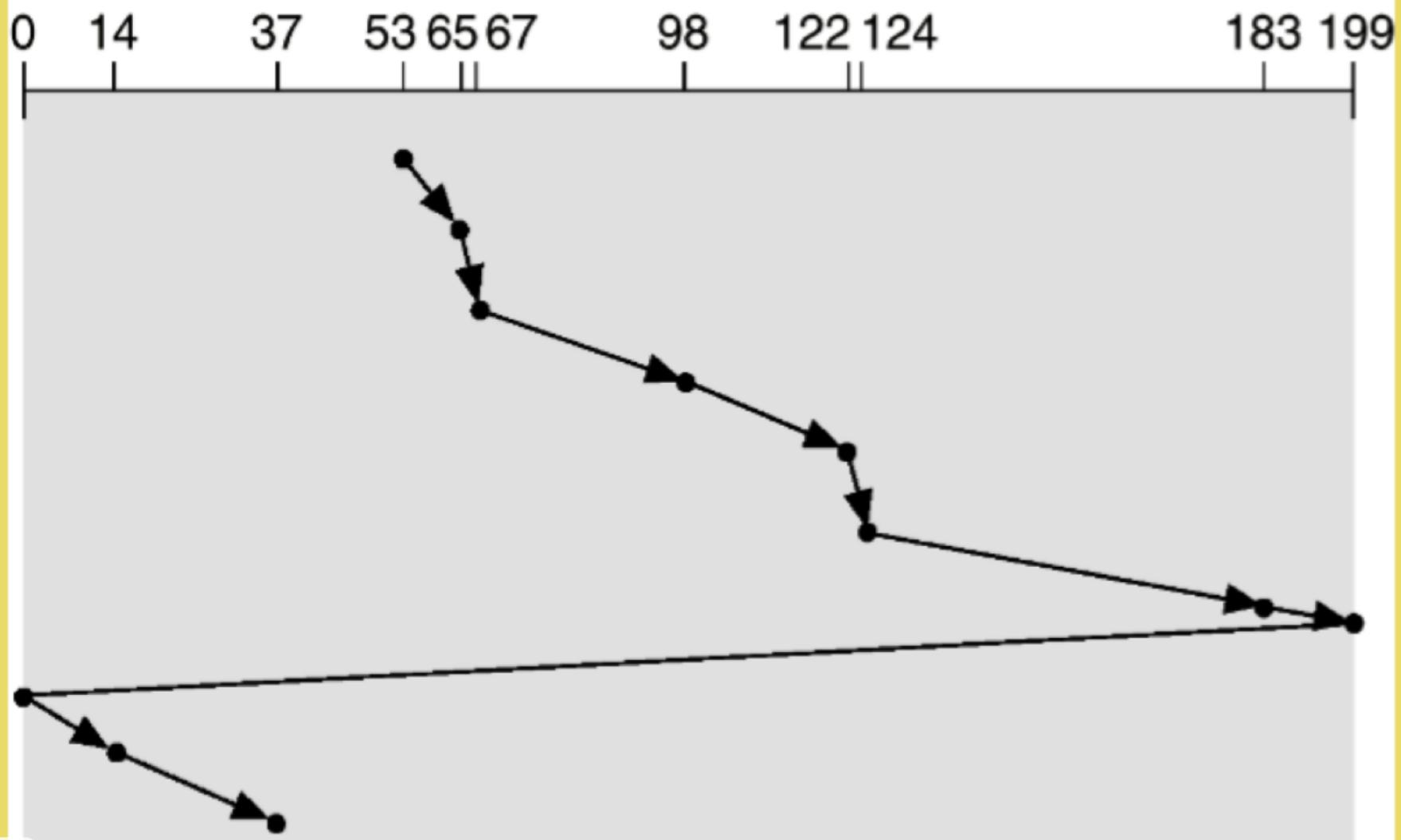
SCAN

queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53



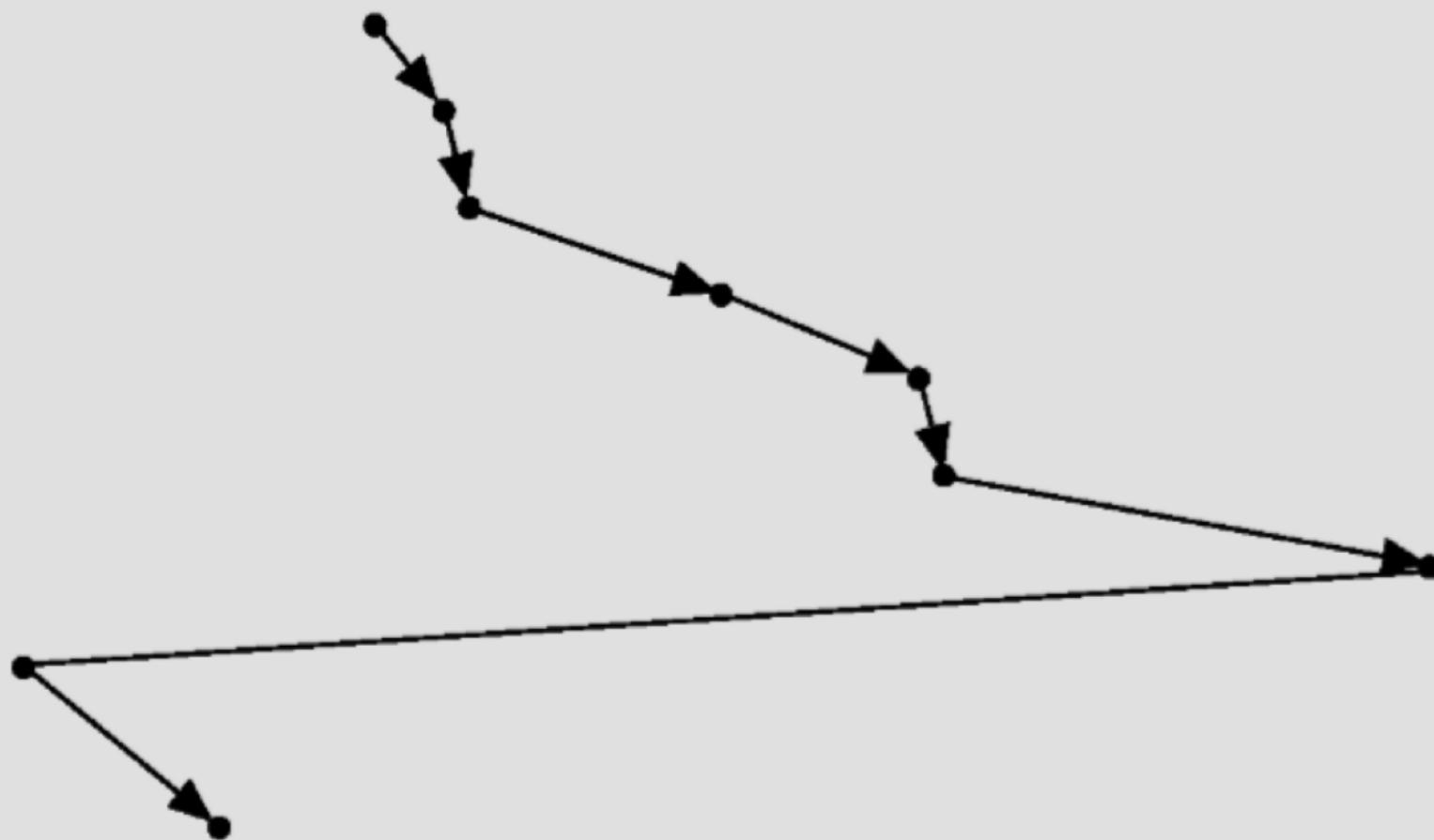
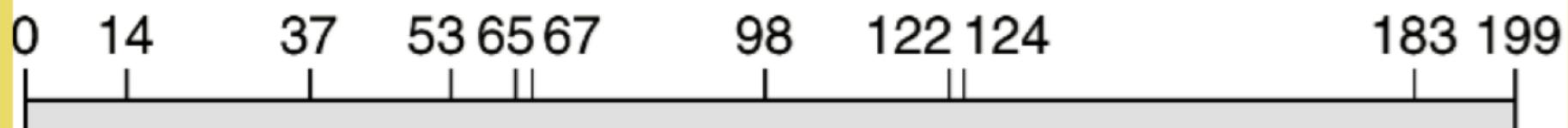
C-SCAN

queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53



C-LOOK

queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53



Sisteme de Fisiere Log-Structured

Scrie totul secvential

Tine minte unde este cea mai recenta copie pentru fiecare bloc

In paralel, sterge informatiile inutile

Consistentă unui sistem de fisiere

Pentru a scrie un singur fisier:

- scrie datele intr-un bloc sau mai multe
- scrie i-node-ul
- modifica lista de bloc-uri libere

Ce se întâmplă dacă există un defect?

Solutii posibile

FSCK: Exploată redundanța din FS

Teste de consistență pentru blocuri:

- se construiesc tabele de blocuri libere și ocupate
- un bloc trebuie să fie într-un singur tabel

Teste de consistență pentru fisiere:

- tabela cu numarul de referiri construită prin parcurgerea directoarelor
- se compară cu nr de referiri din i-node

Sisteme de fisiere jurnalizate

Idee:

- scrie modificările pe disc înainte de a le aplica
- Aplica modificările
- Sterge log-ul

La boot se verifică jurnalul și se efectuează operațiile din tranzacții (trebuie să fie idempotente)



FSCK: Exploateaza redundanta din FS

Teste de consistenta pentru blocuri:

- se construiesc table de blocuri libere si ocupate
- un bloc trebuie sa fie intr-un singur tabel

Teste de consistenta pentru fisiere:

- tabela cu numarul de referiri construita prin parcurgerea directoarelor
- se compara cu nr de referiri din i-node

Sisteme de fisiere jurnalizate

Idee:

- scrie modificarile pe disc inainte de a le aplica
- Aplica modificarile
- Sterge log-ul

La boot se verifica jurnalul si se efectueaza operatiile din tranzactii (trebuie sa fie idempotente)

Avantaje? Dezavantaje?

temp scurt de verificare a consistentei robustete

incetineste sistemul de fisiere

fica jurnalul si se efectueaza
(trebuie sa fie idempotent)

Avantaje? Dezavantaje?

timp scurt de verificare a consistentei
robustete

incetineste sistemul de fisiere

Concluzii

Fisierele reprezinta unitatea de stocare a informatiei

Sistemele de fisiere ofera o interfata comună utilizatorilor pentru a lucra cu fisiere si directoare

Operatii principale:

- gestiunea datelor ce blocuri sunt intr-un fisier?
- organizare ierarhica: directoarele sunt fisiere speciale
- Gestiunea spatiului liber