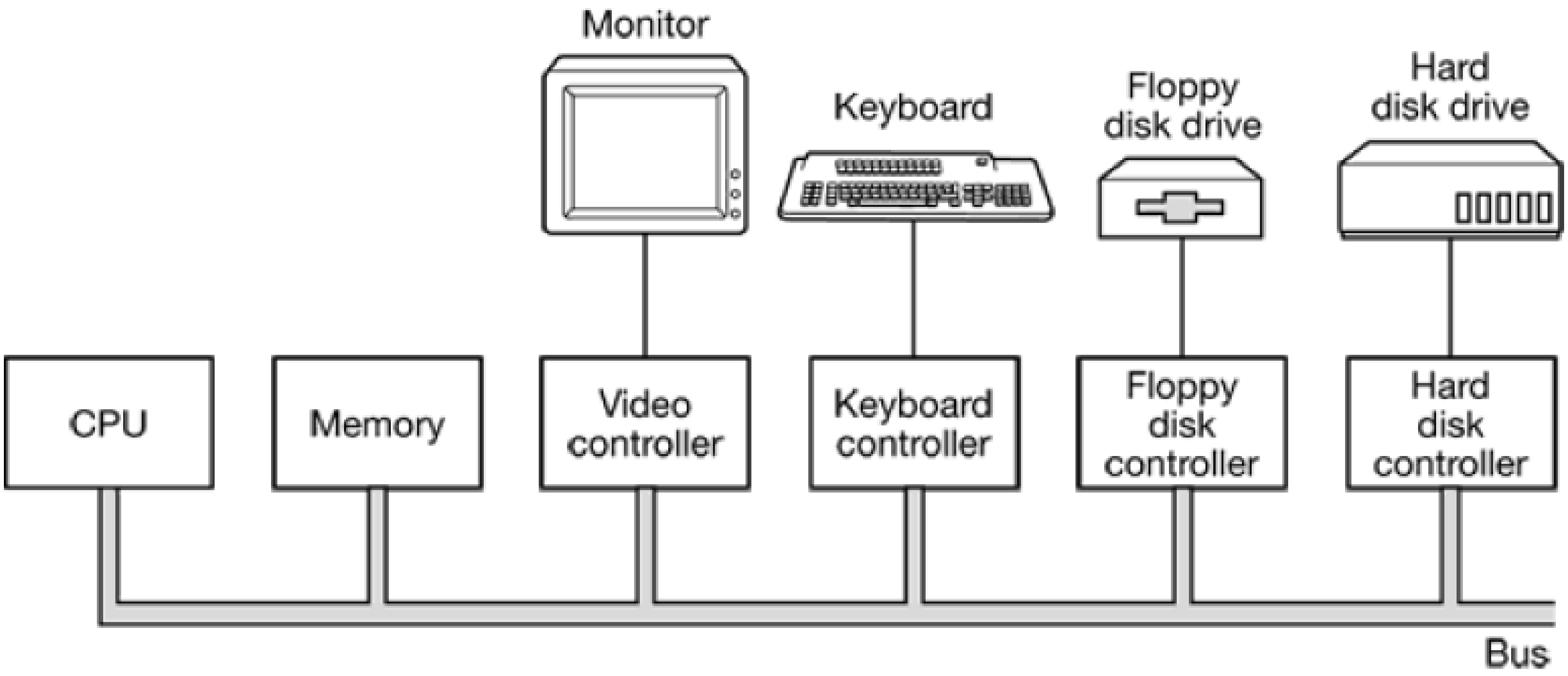
Architectures des Systèmes de Bases de Données

File system : Hard Disk Organisation

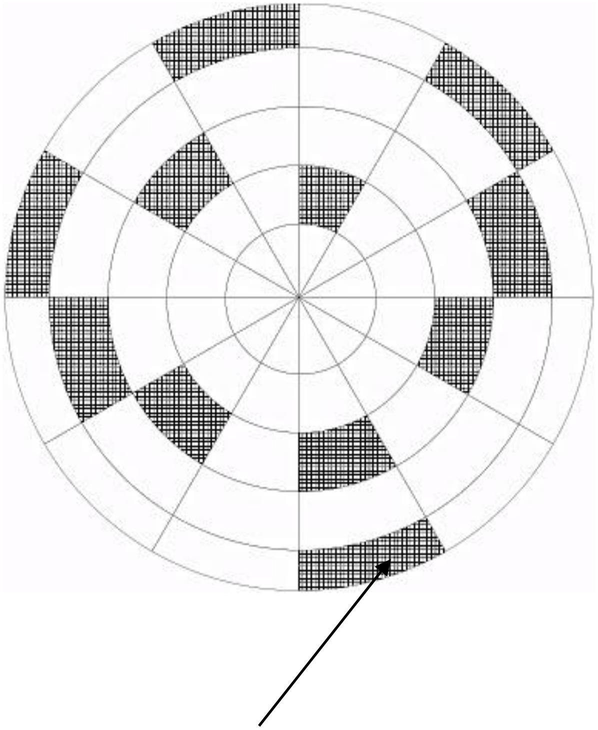
Traduction en cours

Computer Physical Architecture



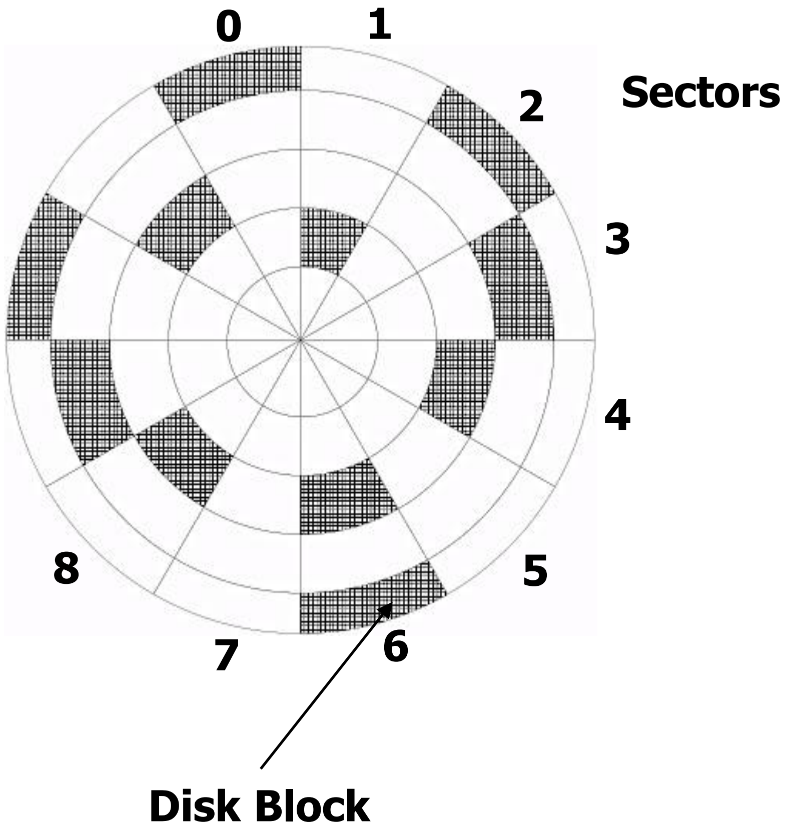
Source MOS : MODERN OPERATING SYSTEMS ANDREW S. TANENBAUM (A.S.T)

HDD Block Layout equivalent



Disk Block

HDD Block Layout equivalent



0

1

2

Sectors

3

4

5

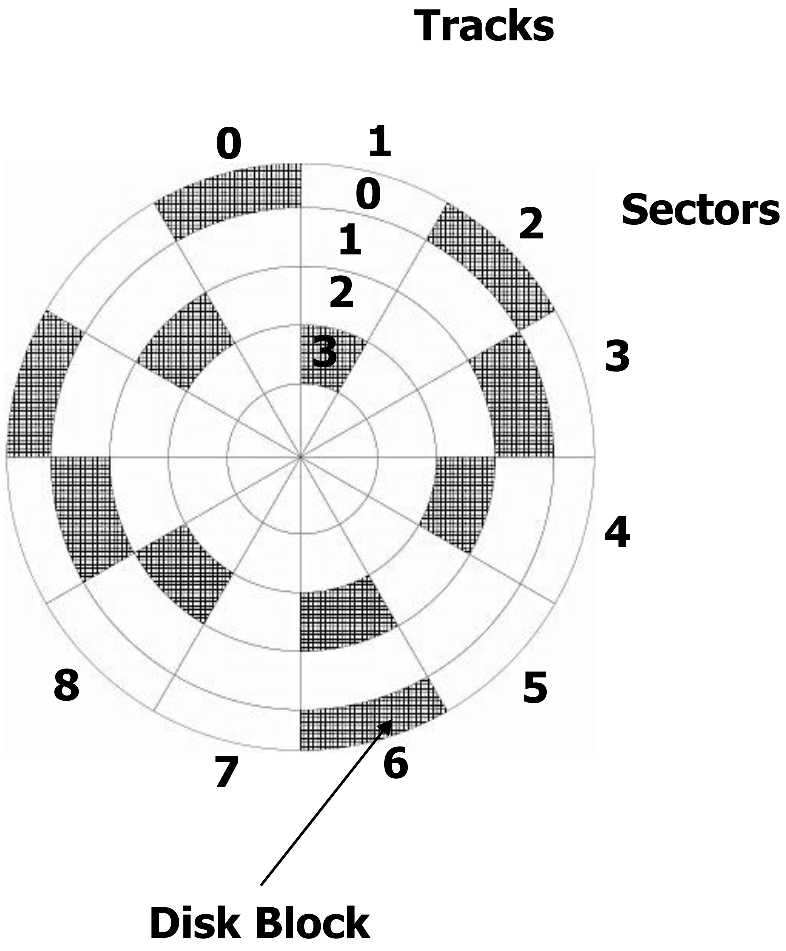
6

7

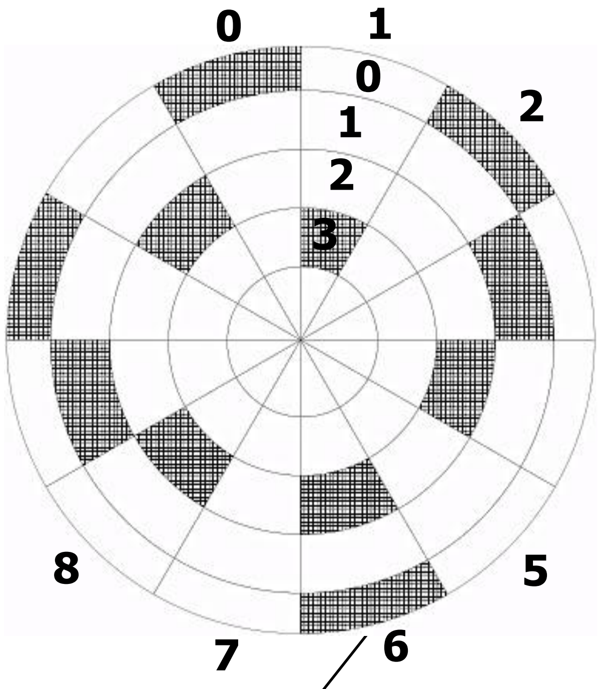
8

Disk Block

HDD Block Layout equivalent



HDD Block Layout equivalent



=

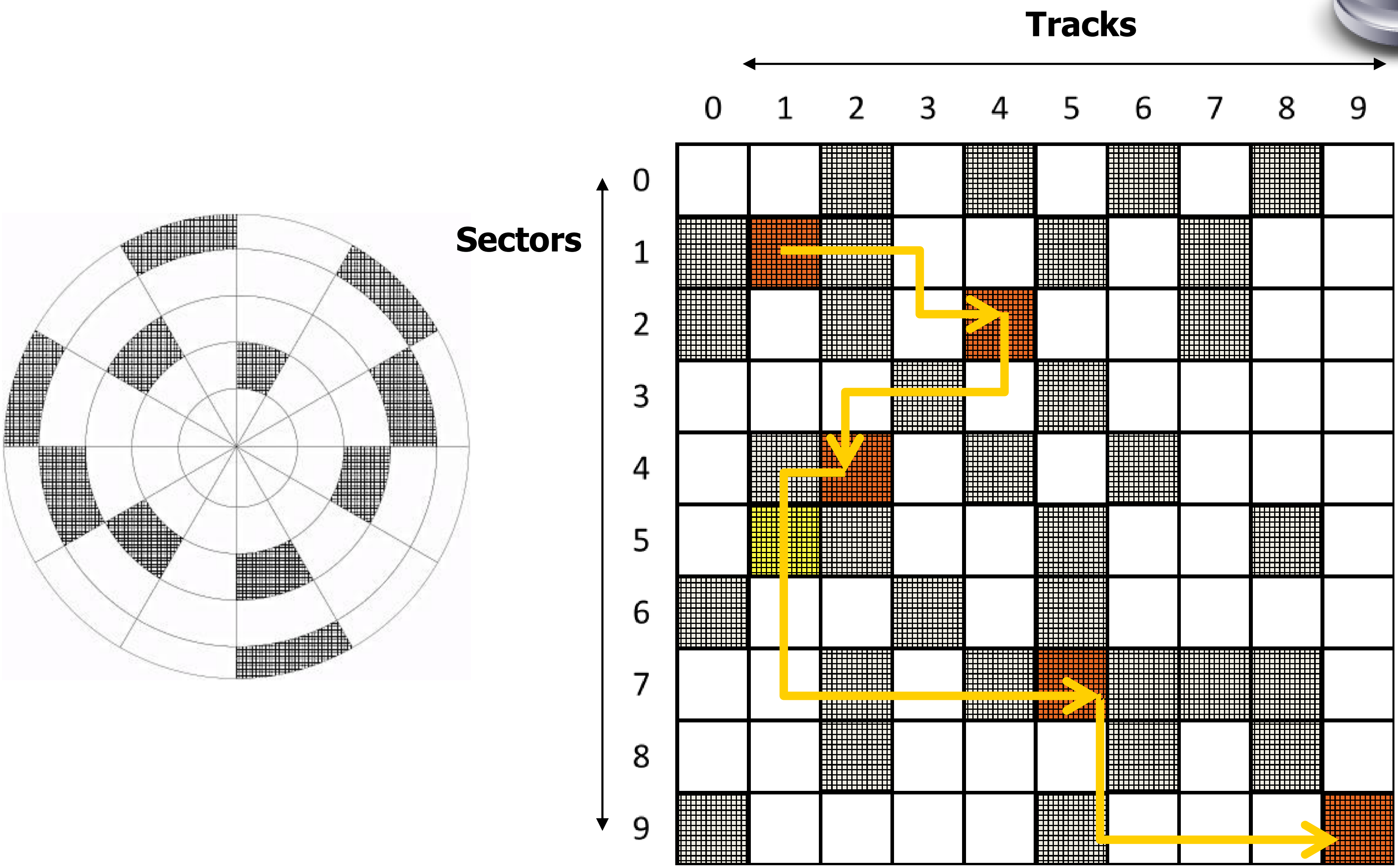
Tracks

Sectors

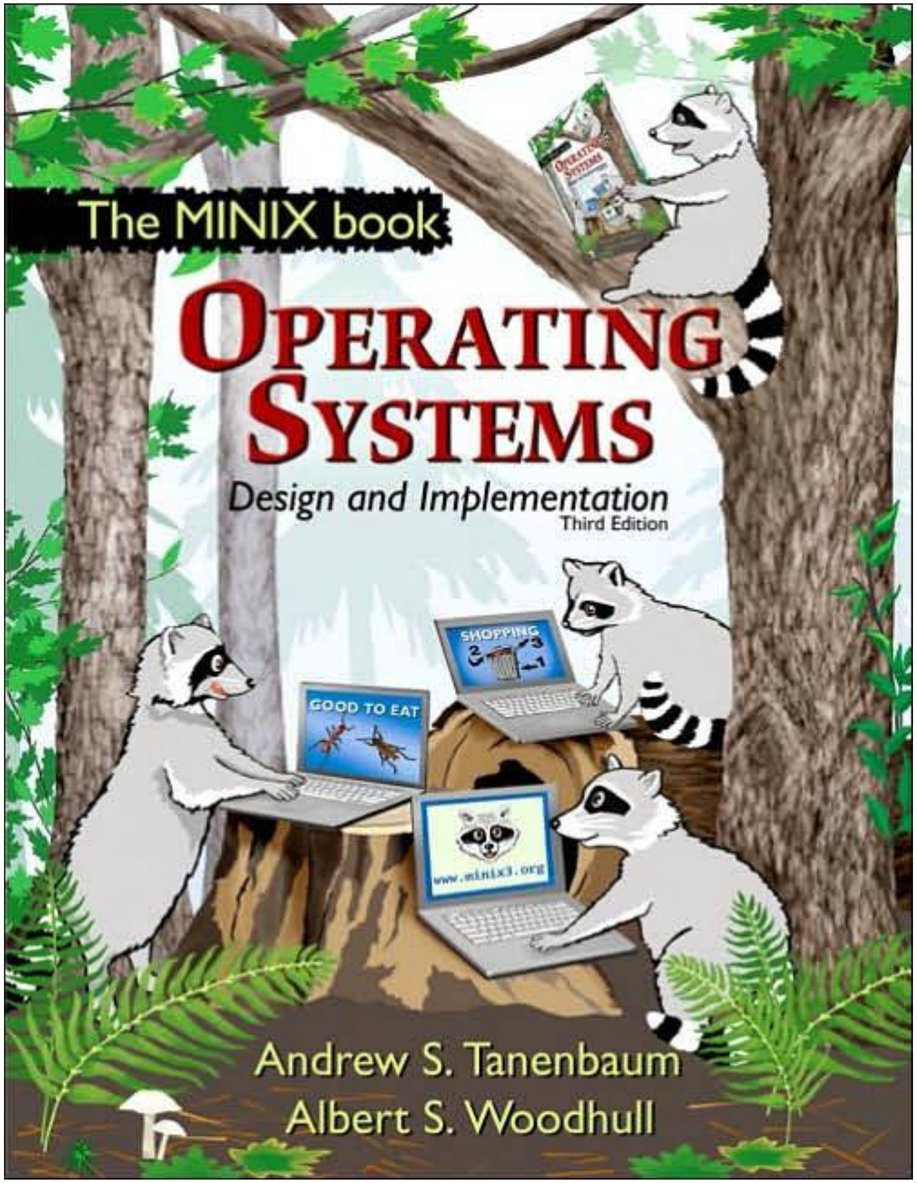
Sectors \ Tracks	0	1	2	3	4	5	6	7	8	9
0			▨		▨		▨		▨	
1	▨		▨			▨		▨		
2	▨		▨					▨		
3				▨		▨				
4		▨			▨		▨			
5			▨			▨			▨	
6	▨			▨		▨				
7			▨		▨	▨	▨	▨	▨	
8			▨				▨		▨	
9	▨					▨			▨	

Disk Block

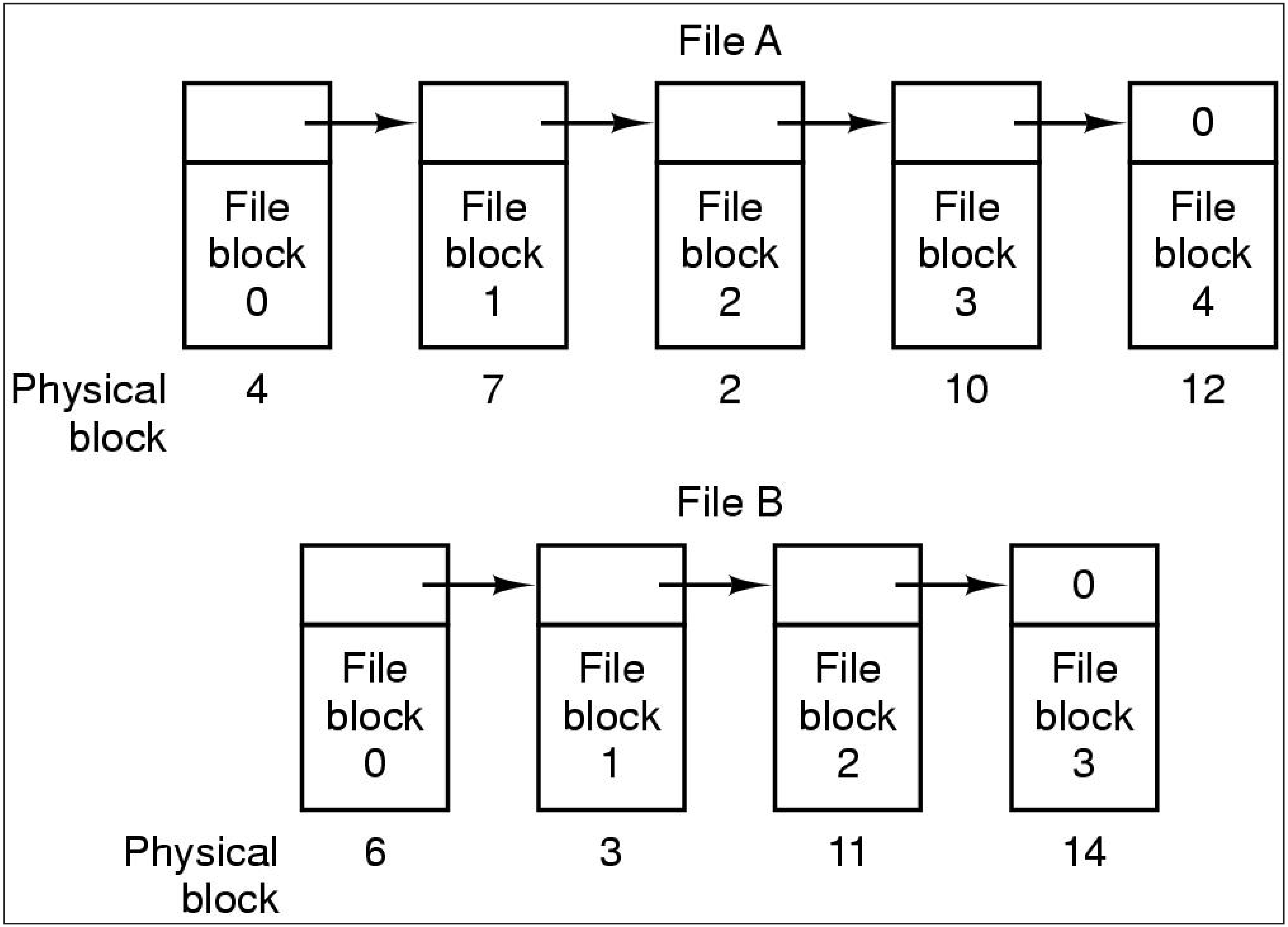
File Block list Layout equivalent



MOS MINIX


The MINIX book
OPERATING SYSTEMS
Design and Implementation
Third Edition
GOOD TO EAT
SHOPPING
www.minix3.org
Andrew S. Tanenbaum
Albert S. Woodhull

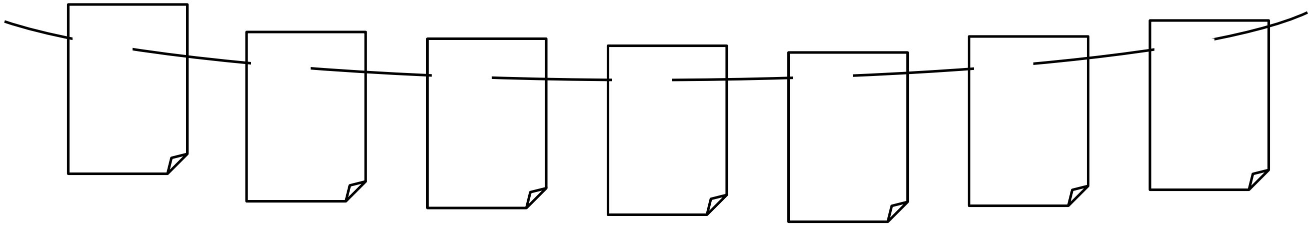
File Block Chain



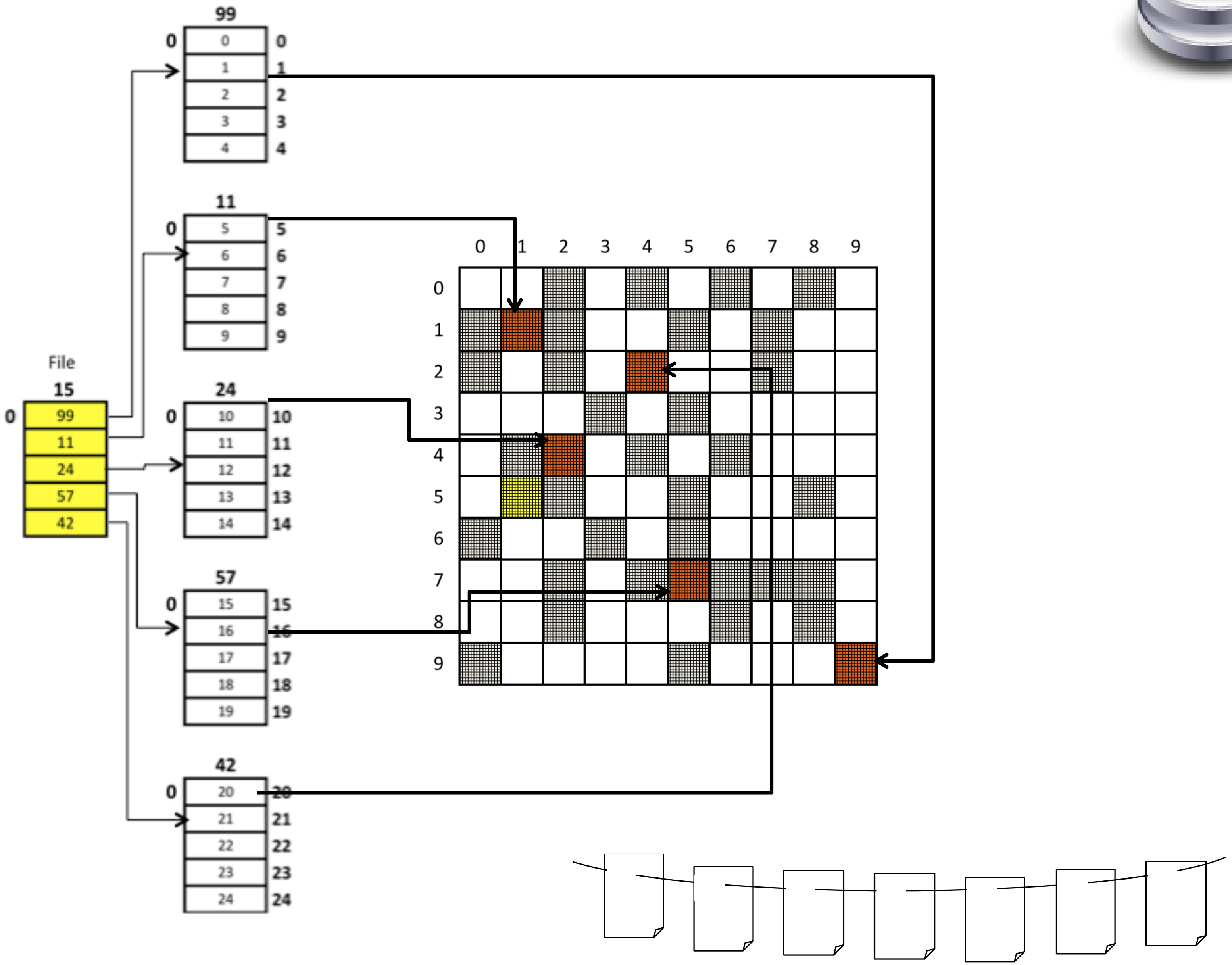
Source MOS : MODERN OPERATING SYSTEMS ANDREW S. TANENBAUM (A.S.T)

File Word Origin

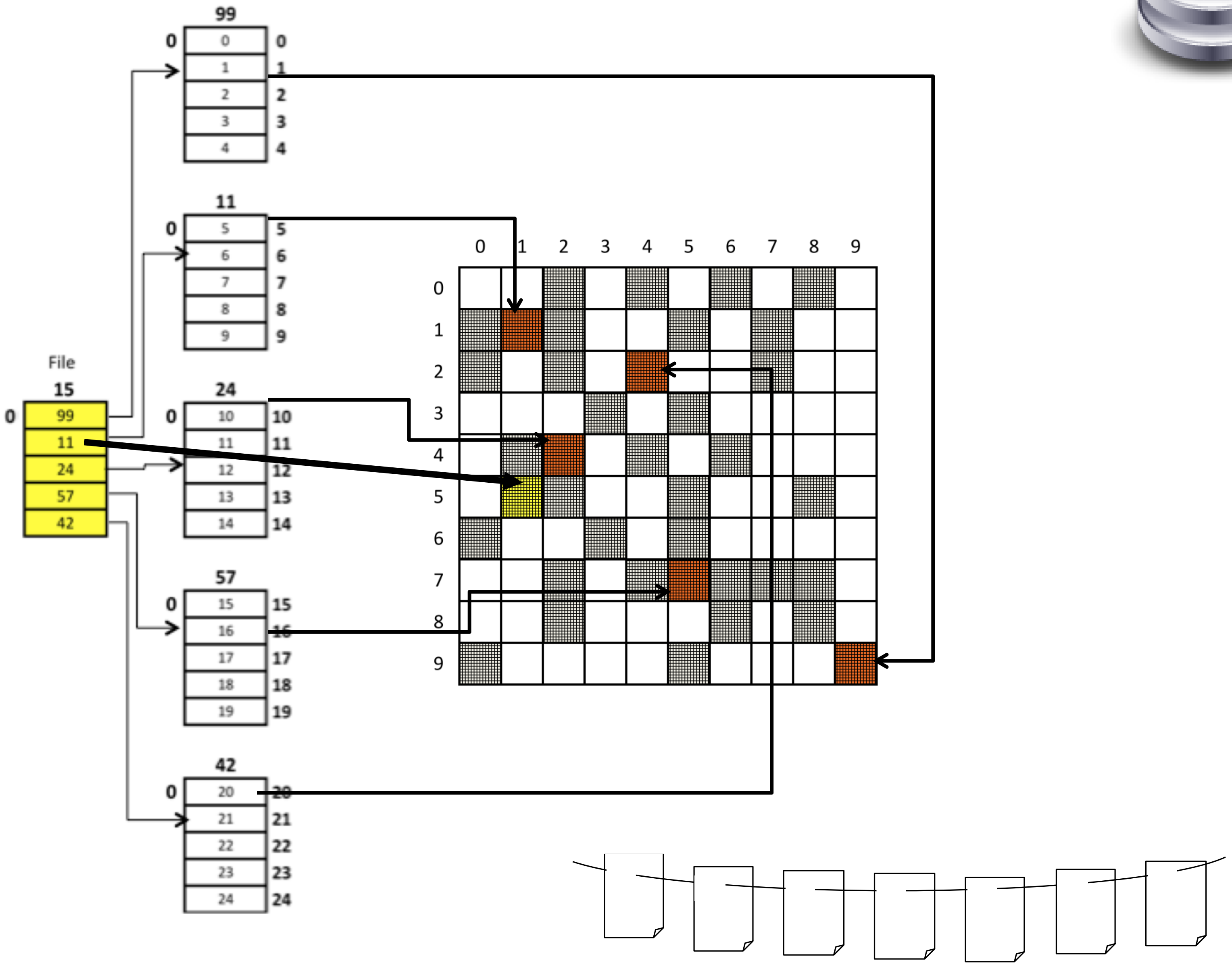
- From French : filer
  - to string documents on a thread or wire,
- The word "file" derives from the Latin *filum* ("a thread").



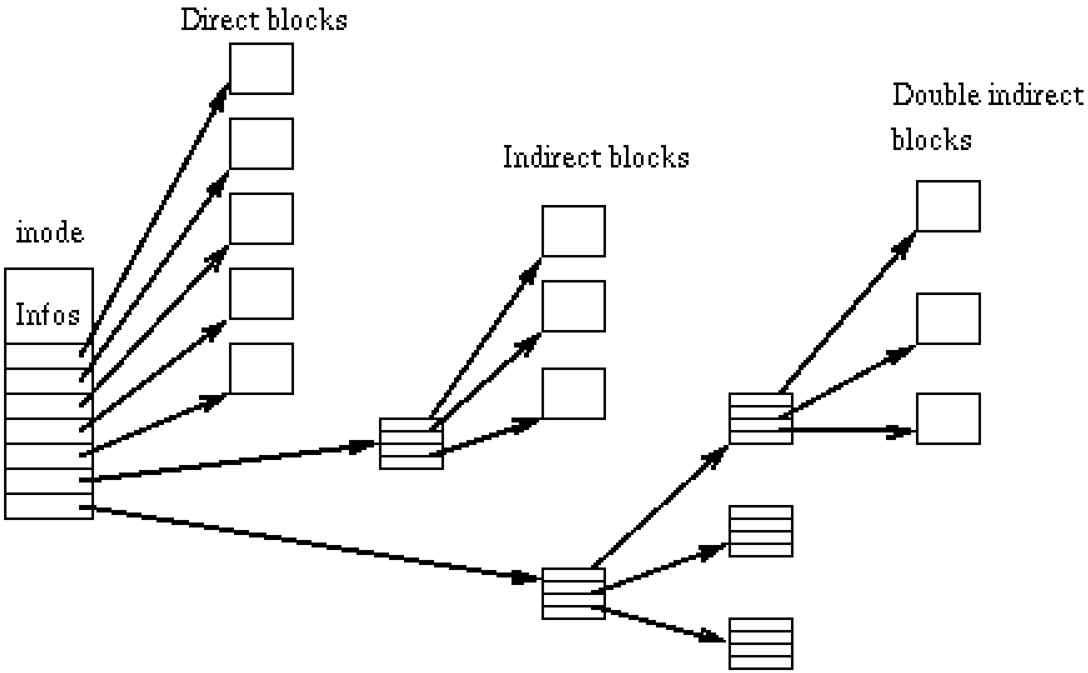
File descriptor



File descriptor



Unix File descriptor or inode

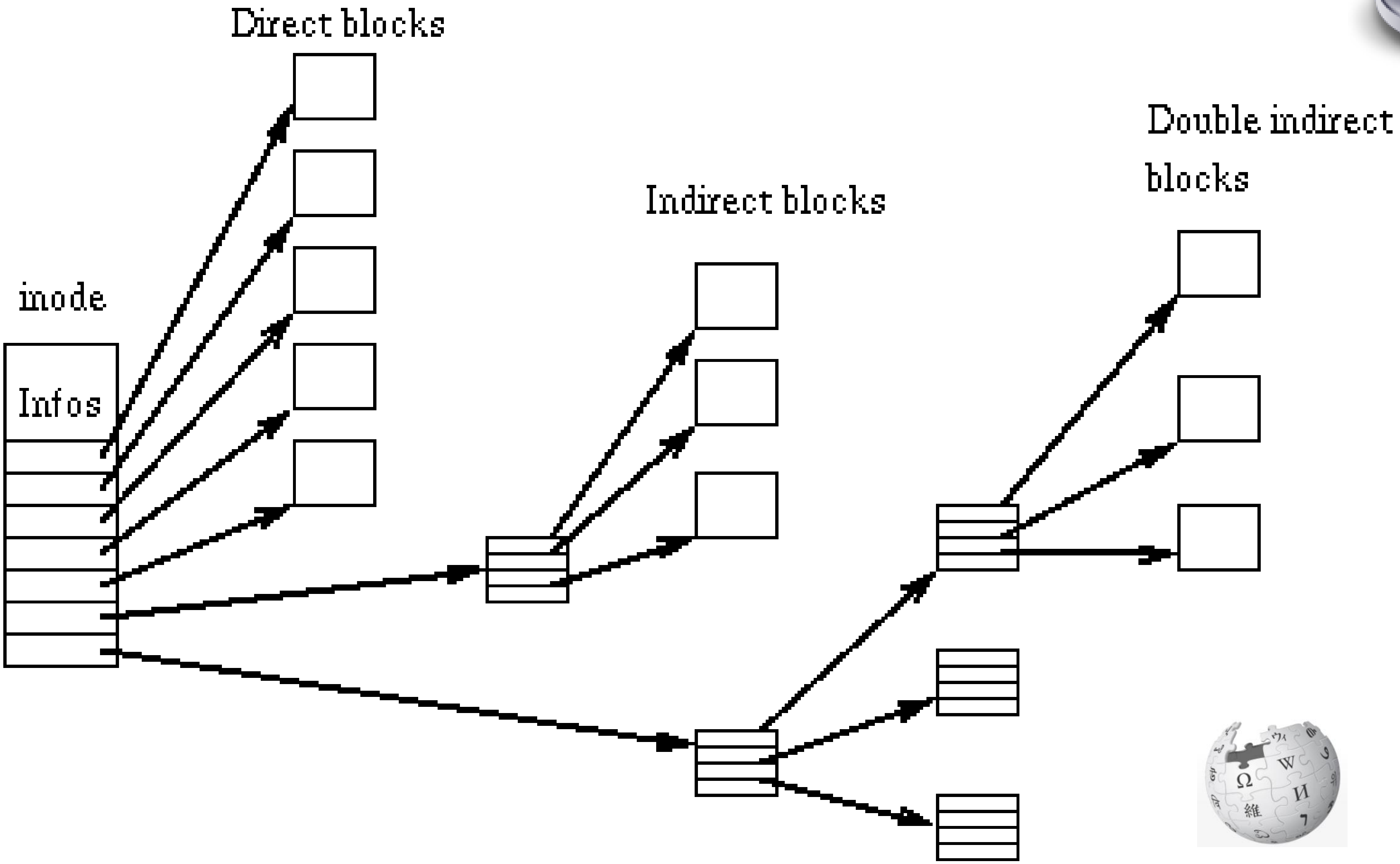


UNIX

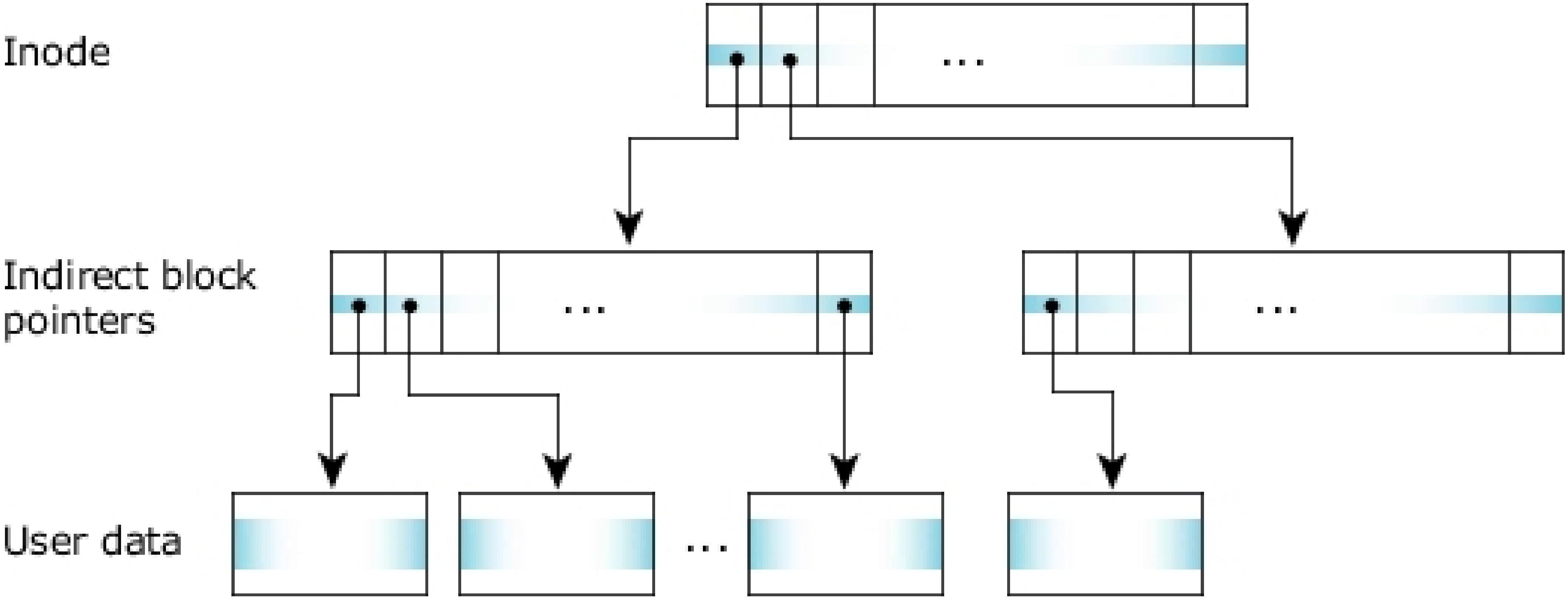
Unix is 32-bit multi-tasking, multi-user OS (Operating System) invented by Bell Labs that is used on many type of computer systems. UNIX was designed to be a small, flexible system used exclusively by programmers.

AT&T

Unix Linux Inode

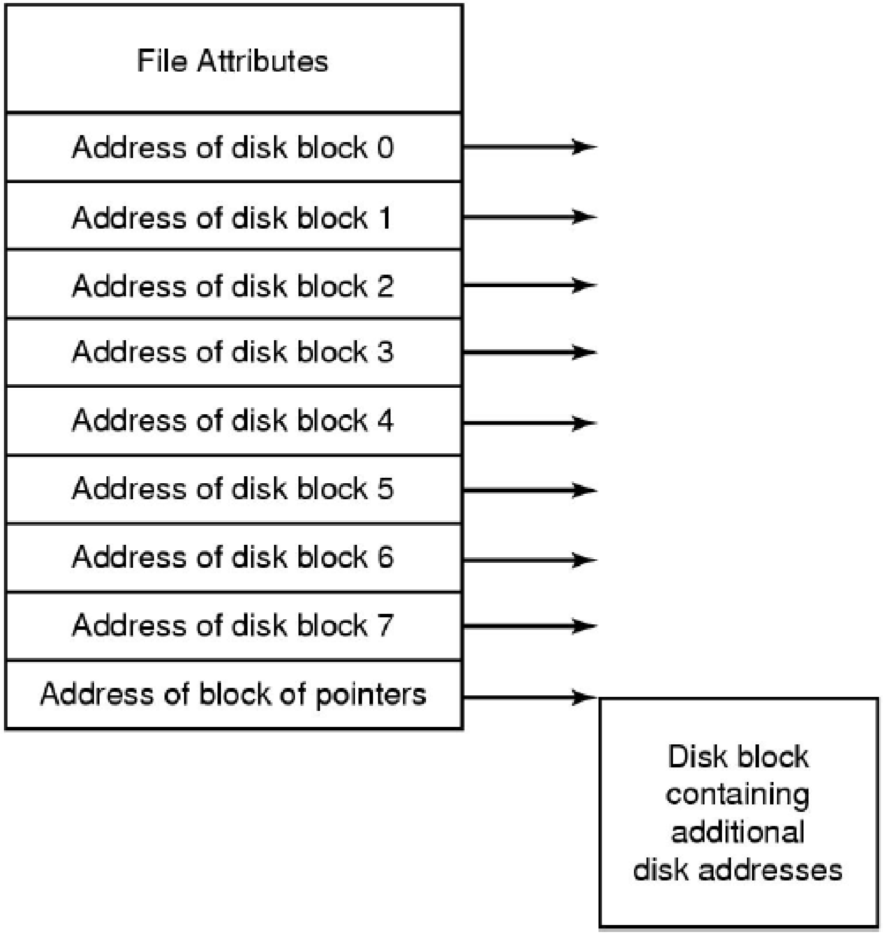


Unix Linux Inode : indirect block pointers



Unix Linux Inode

I-nodes



File Attributes

Address of disk block 0

Address of disk block 1

Address of disk block 2

Address of disk block 3

Address of disk block 4

Address of disk block 5

Address of disk block 6

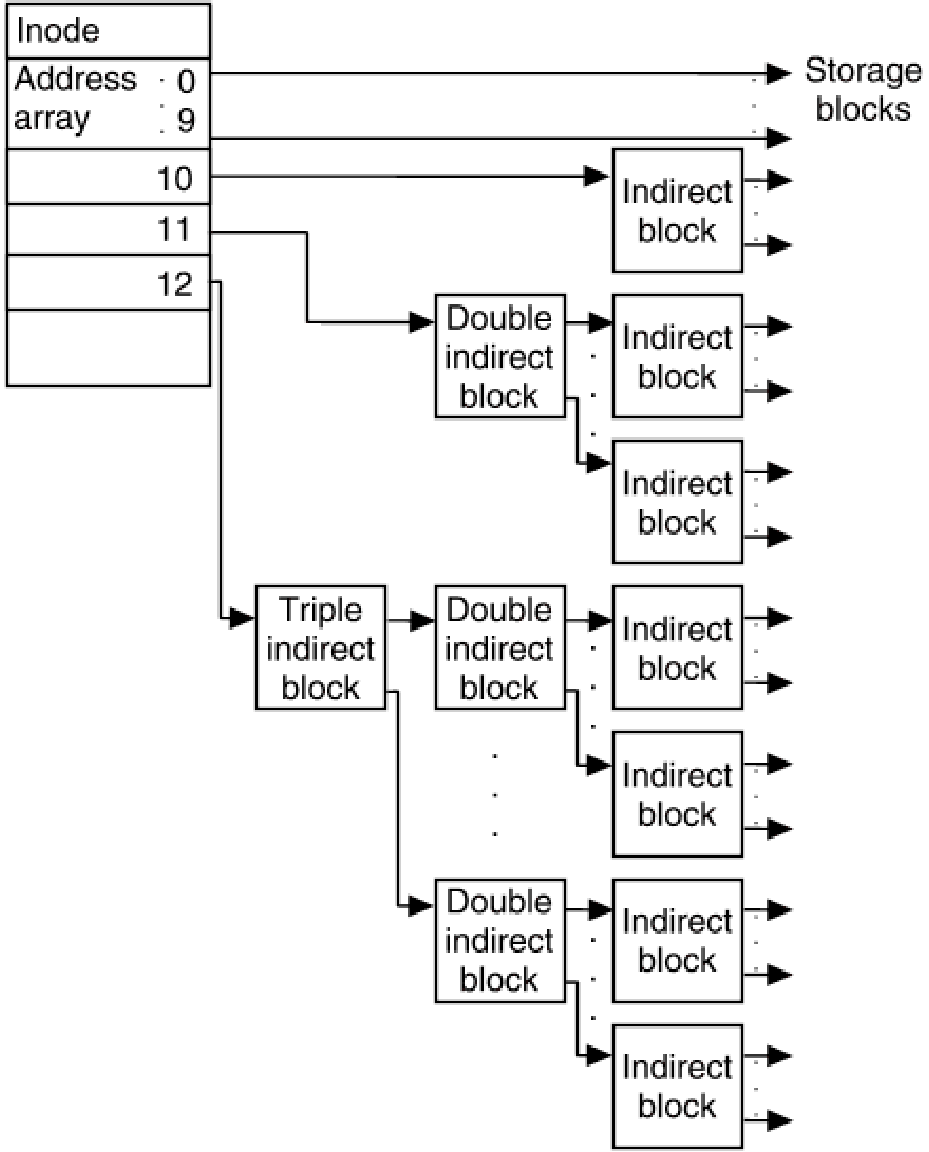
Address of disk block 7

Address of block of pointers

Disk block containing additional disk addresses

Figure 4-13. An example i-node.

UnixWare



c.xinuos.com/en/Navpages/index.html

Inode

The UNIX V7 File System (2)

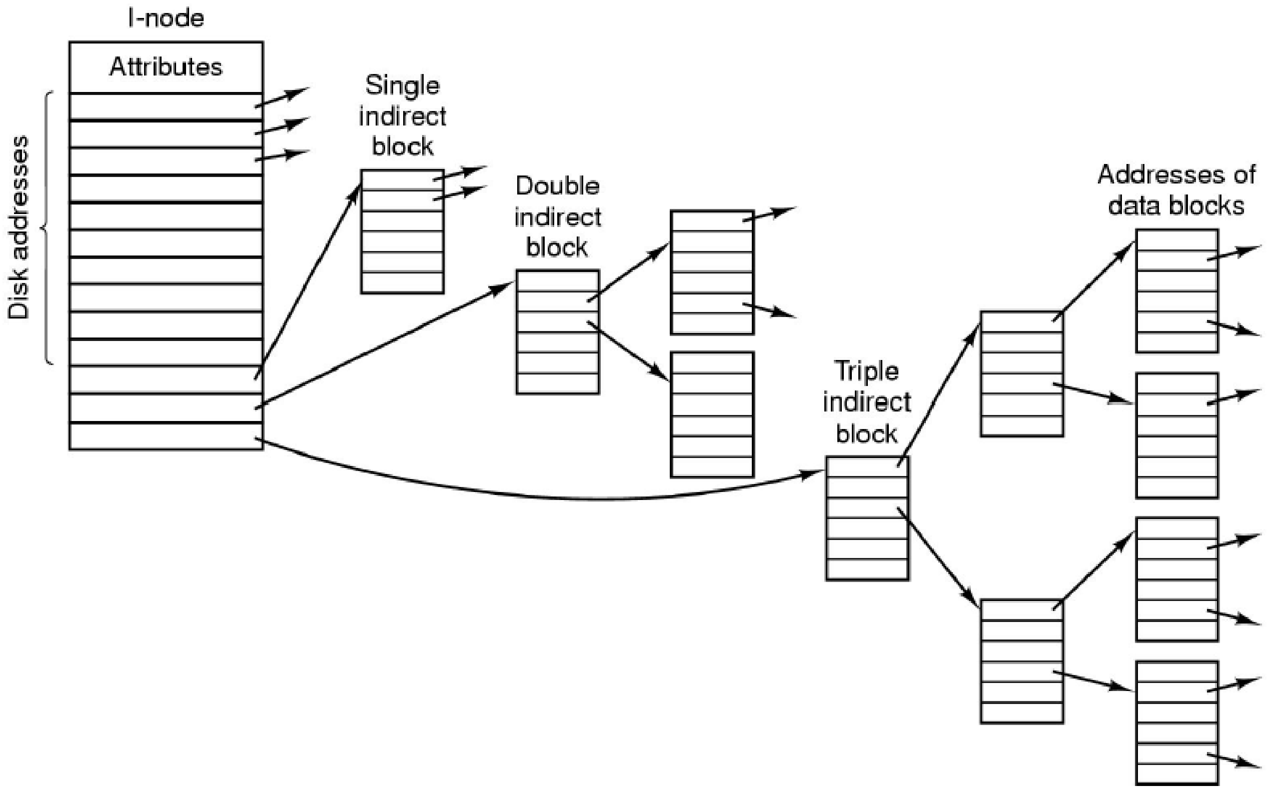


Figure 4-34. A UNIX i-node.

Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-6006639

Unix Linux Inode

Hierarchical Directory Systems (2)

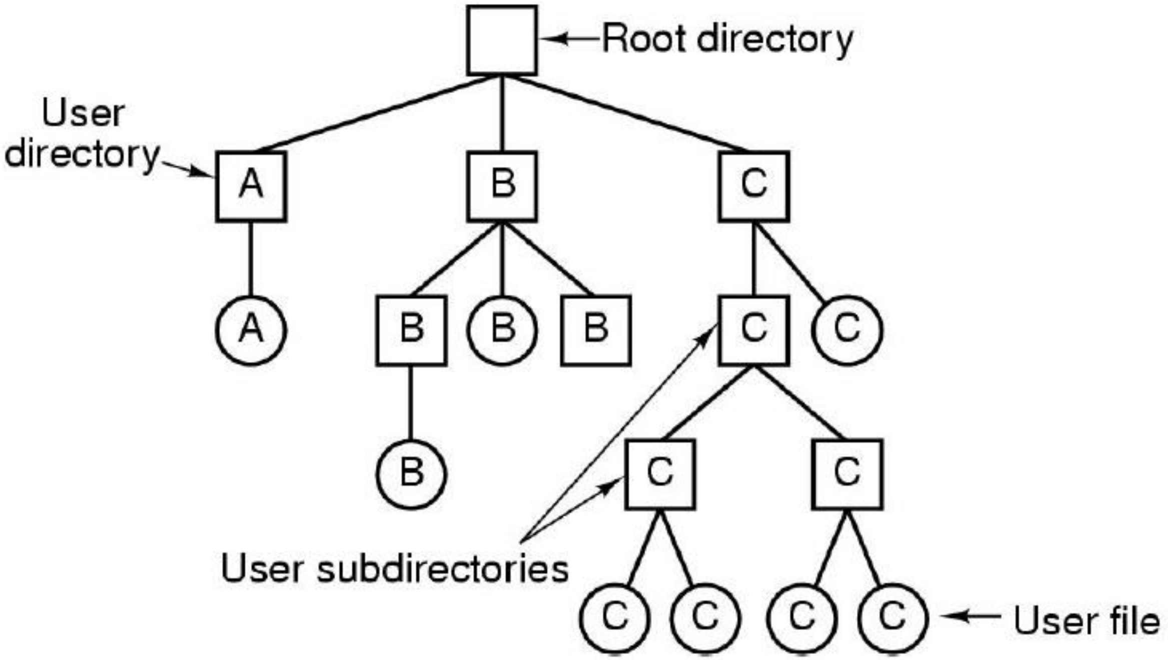
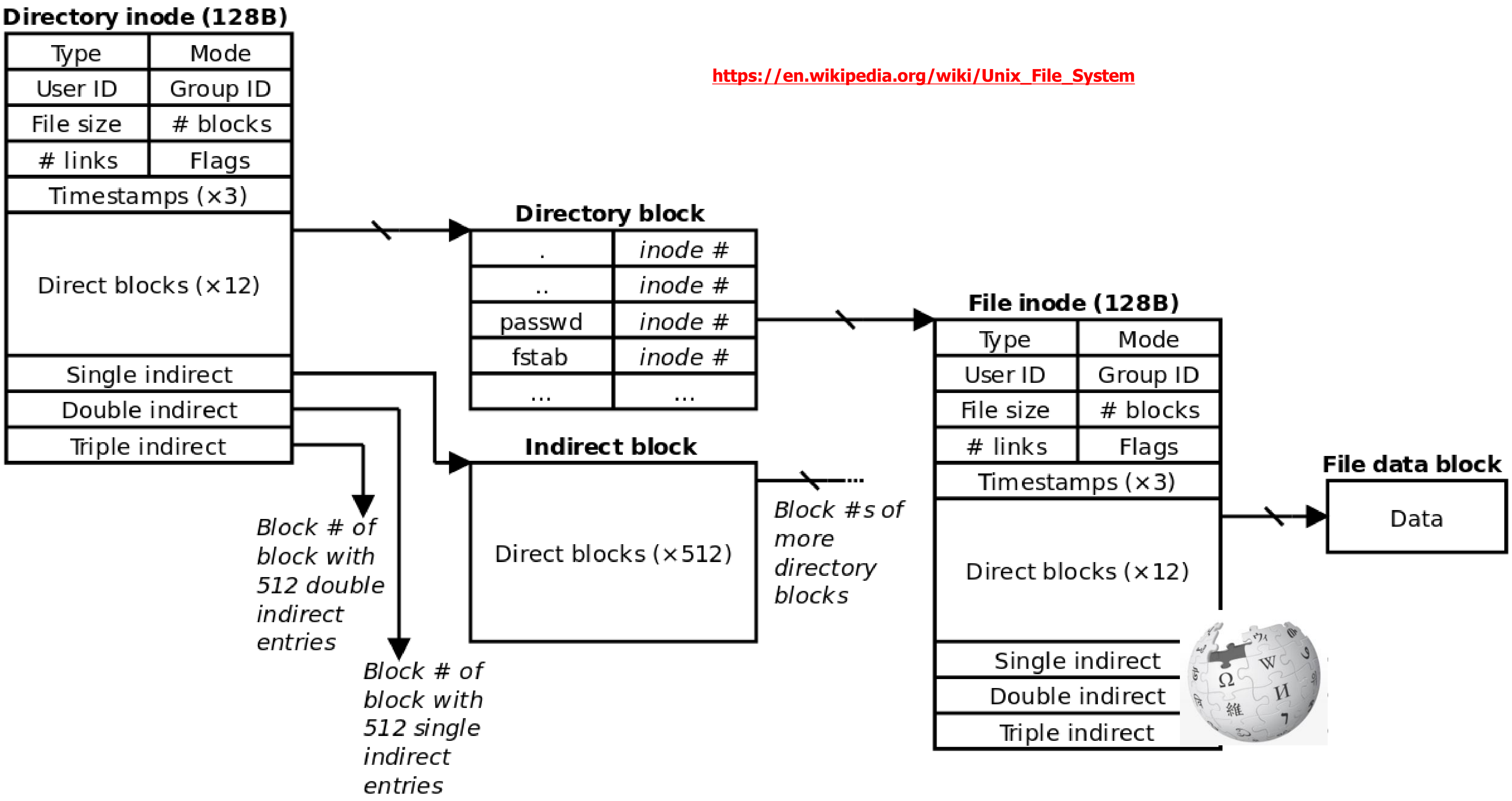


Figure 4-7. A hierarchical directory system.

Tanenbaum, Modern Operating Systems 3 e, (c) 2008 Prentice-Hall, Inc. All rights reserved. 0-13-6006639

UFS (Unix File System) structure : Directory

https://en.wikipedia.org/wiki/Unix_File_System



Directory inode (128B)

Type	Mode
User ID	Group ID
File size	# blocks
# links	Flags
Timestamps (×3)	
Direct blocks (×12)	
Single indirect	
Double indirect	
Triple indirect	

Directory block

.	*inode #*
..	*inode #*
passwd	*inode #*
fstab	*inode #*
…	…

Indirect block

Direct blocks (×512)

Block #s of more directory blocks

Block # of block with 512 double indirect entries

Block # of block with 512 single indirect entries

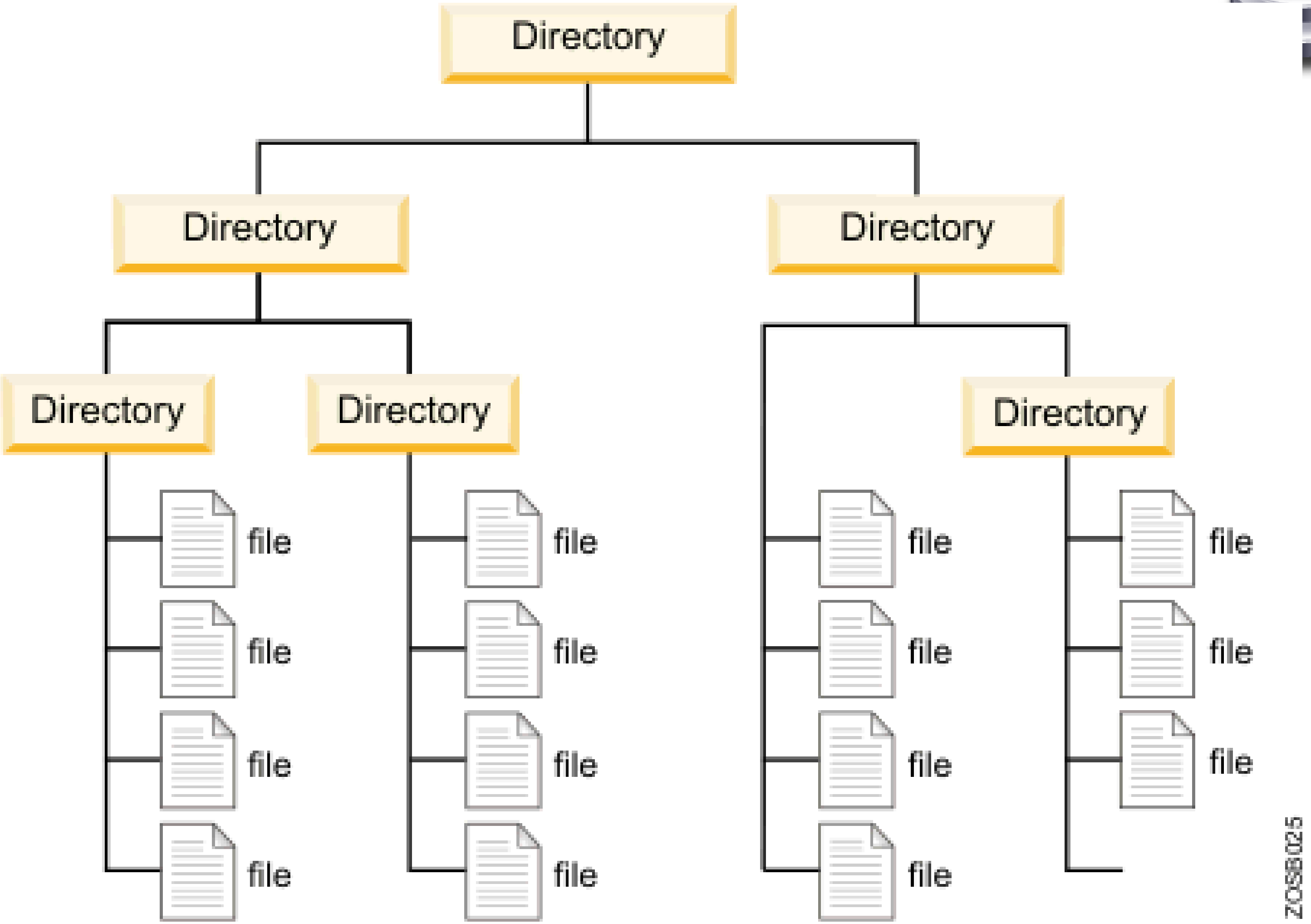
File inode (128B)

Type	Mode
User ID	Group ID
File size	# blocks
# links	Flags
Timestamps (×3)	
Direct blocks (×12)	
Single indirect	
Double indirect	
Triple indirect	

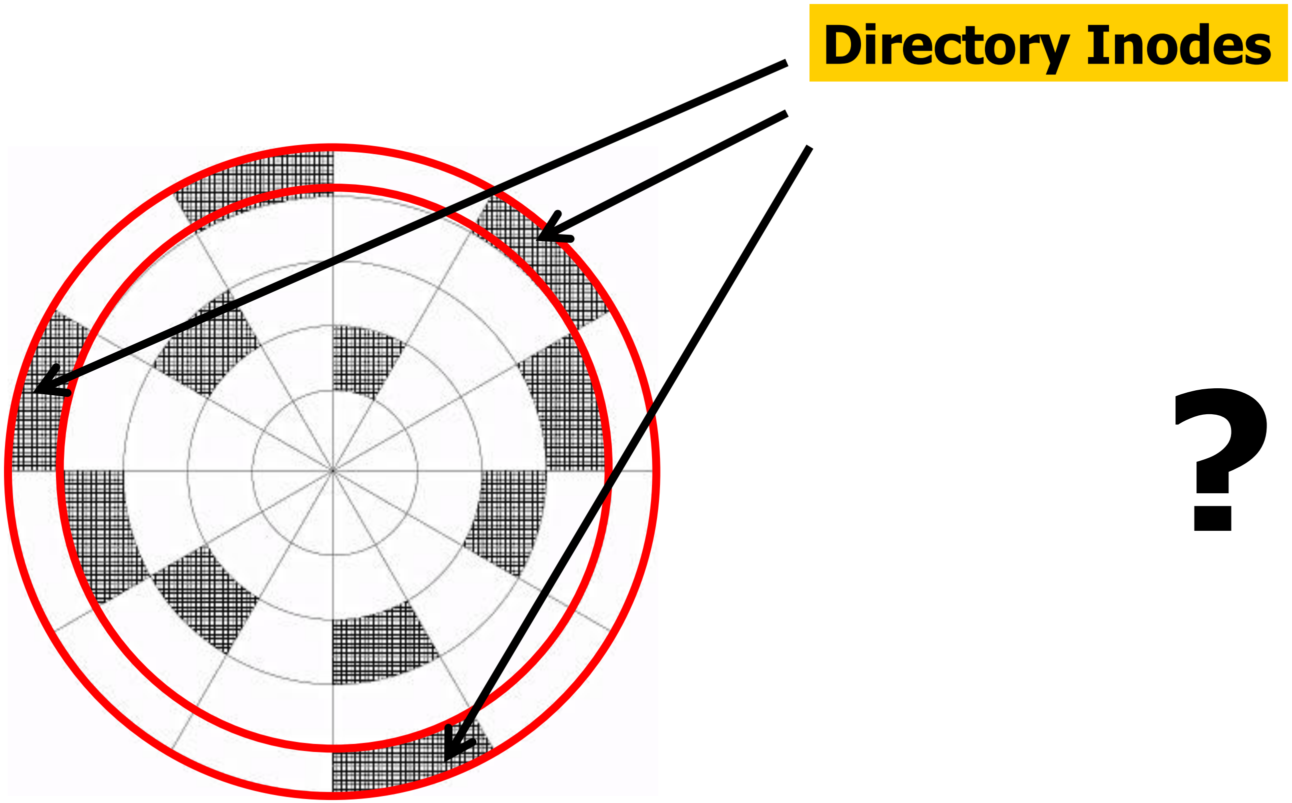
File data block

Data

Directory


Directory
Directory
Directory
Directory
Directory
Directory
file
file
file
file
file
file
file
file
file
file
file
file
file
file
file
ZOS8025

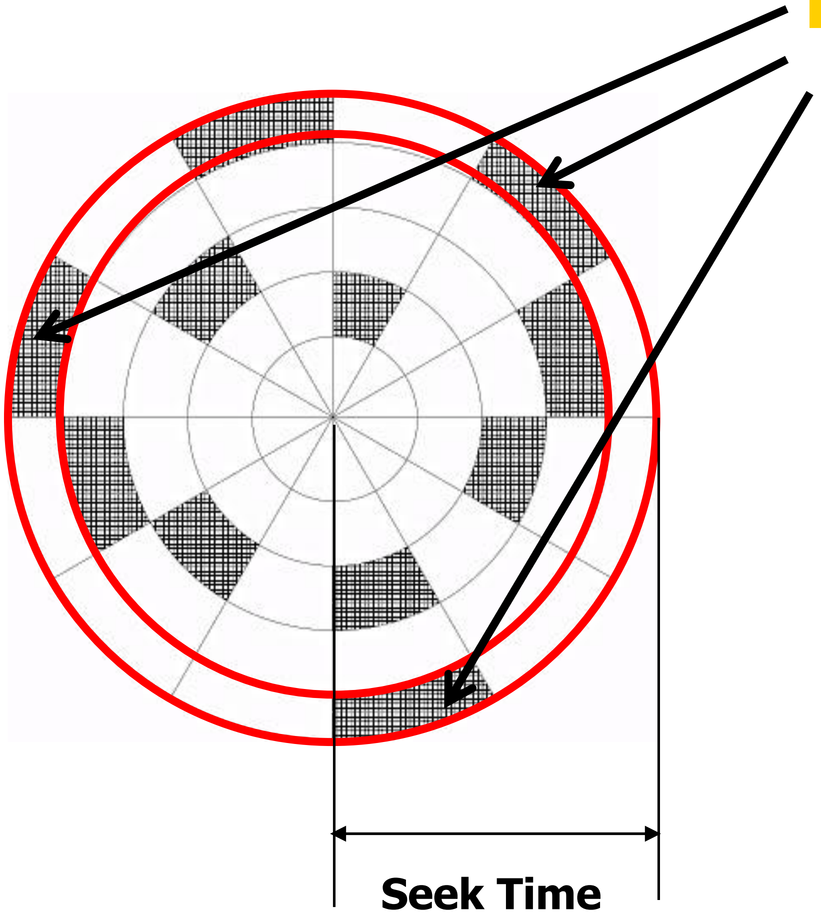
Directories Inodes location



Directory Inodes

?

Directories Inodes location

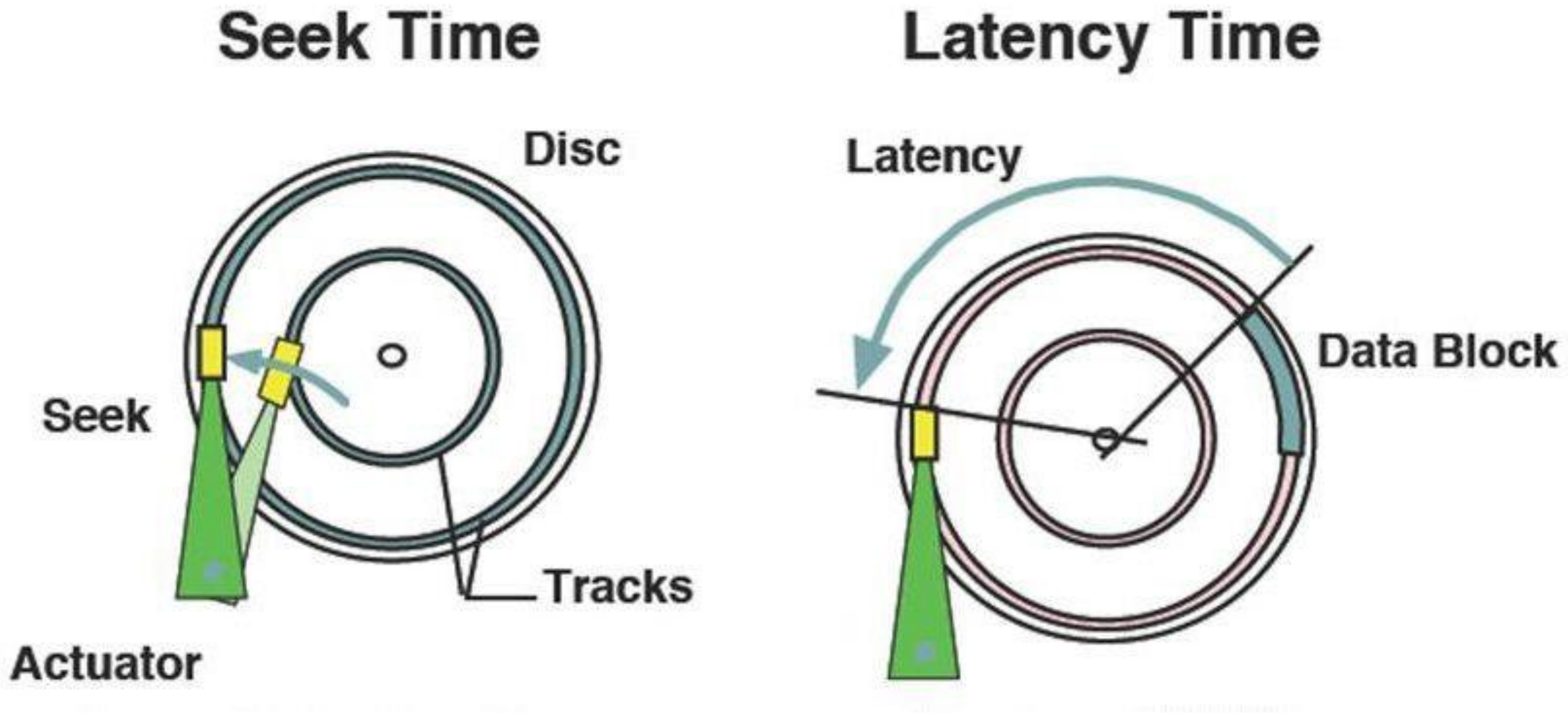


Directory Inodes

Seek Time

?

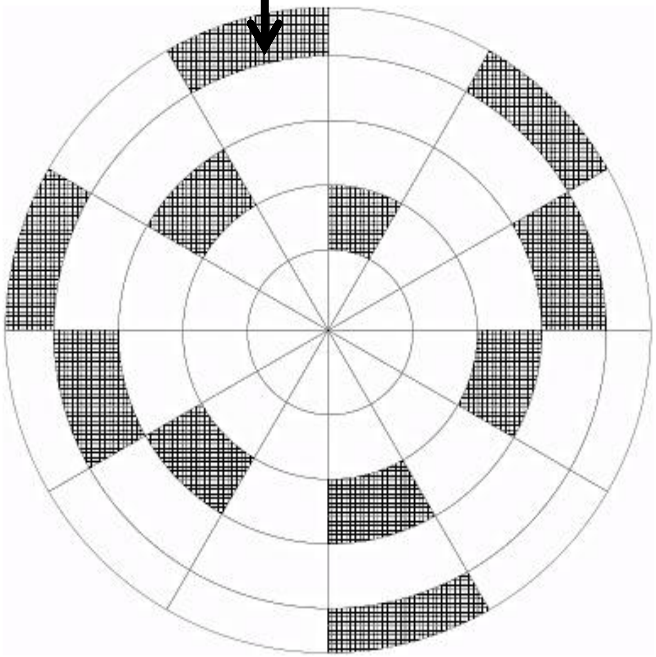
Seek Time, Latency Time



https://dbadiaryy.wordpress.com/2018/01/31/ioping/

Master Block Record

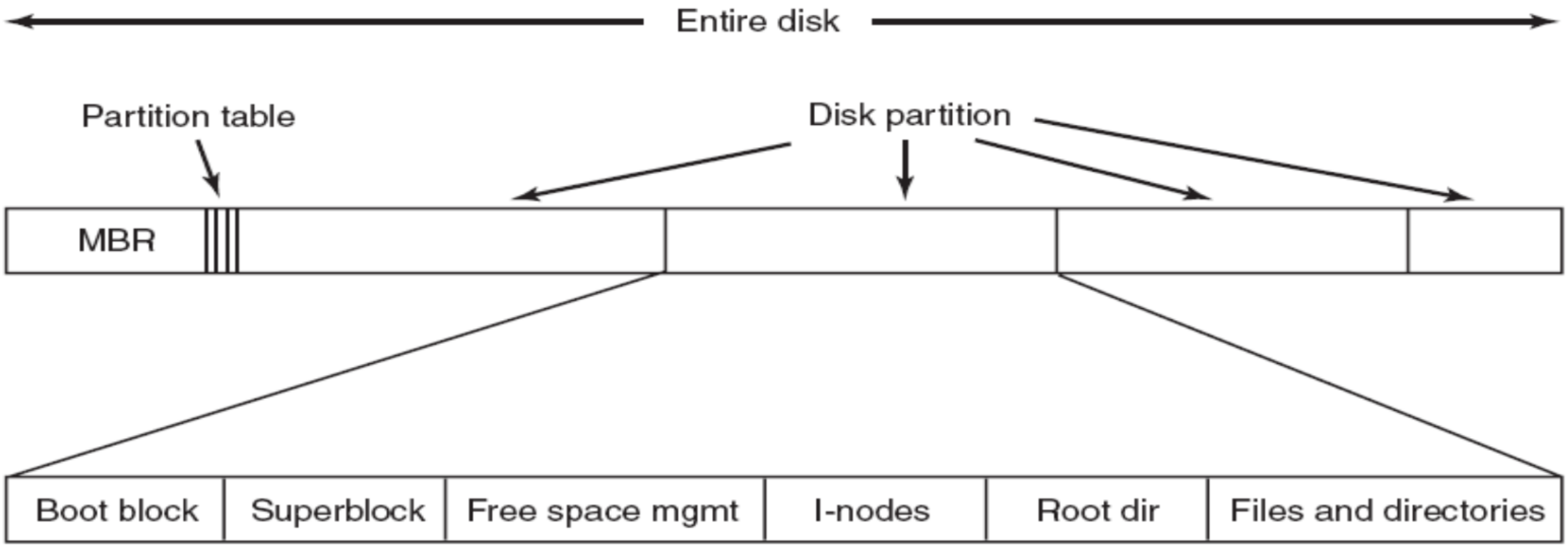
Master Block Record



	0	1	2	3	4	5	6	7	8	9
0	🟧		▨		▨		▨		▨	
1	▨		▨			▨		▨		
2	▨		▨					▨		
3				▨		▨				
4		▨			▨		▨			
5			▨			▨			▨	
6	▨			▨		▨				
7			▨		▨	▨	▨	▨	▨	
8			▨				▨		▨	
9	▨					▨			▨	

AST : Disk Format, disk partitions

File System Layout



Free blocks

- Part of the hard drive stores a map of blocks that have already been used up and others that are still free.
- When the computer wants to store new information, it takes a look at the map to find some free blocks.

Block Free List

Free Block Number

	0	1	2	3	4	5	6	7	8	9
0	MBR	1	2	3	4	5	6	7	8	9
1	10	11	12	13	14	15	16	17	18	19
2	20	21	22	23	24	25	26	27	28	29
3	30	31	32	33	34	35	36	37	38	39
4	40	41	42	43	44	45	46	47	48	49
5	50	51	52	53	54	55	56	57	58	59
6	60	61	62	63	64	65	66	67	68	69
7	70	71	72	73	74	75	76	77	78	79
8	80	81	82	83	84	85	86	87	88	89
9	90	91	92	93	94	95	96	97	98	99

Free Blocks

Free List

Free List

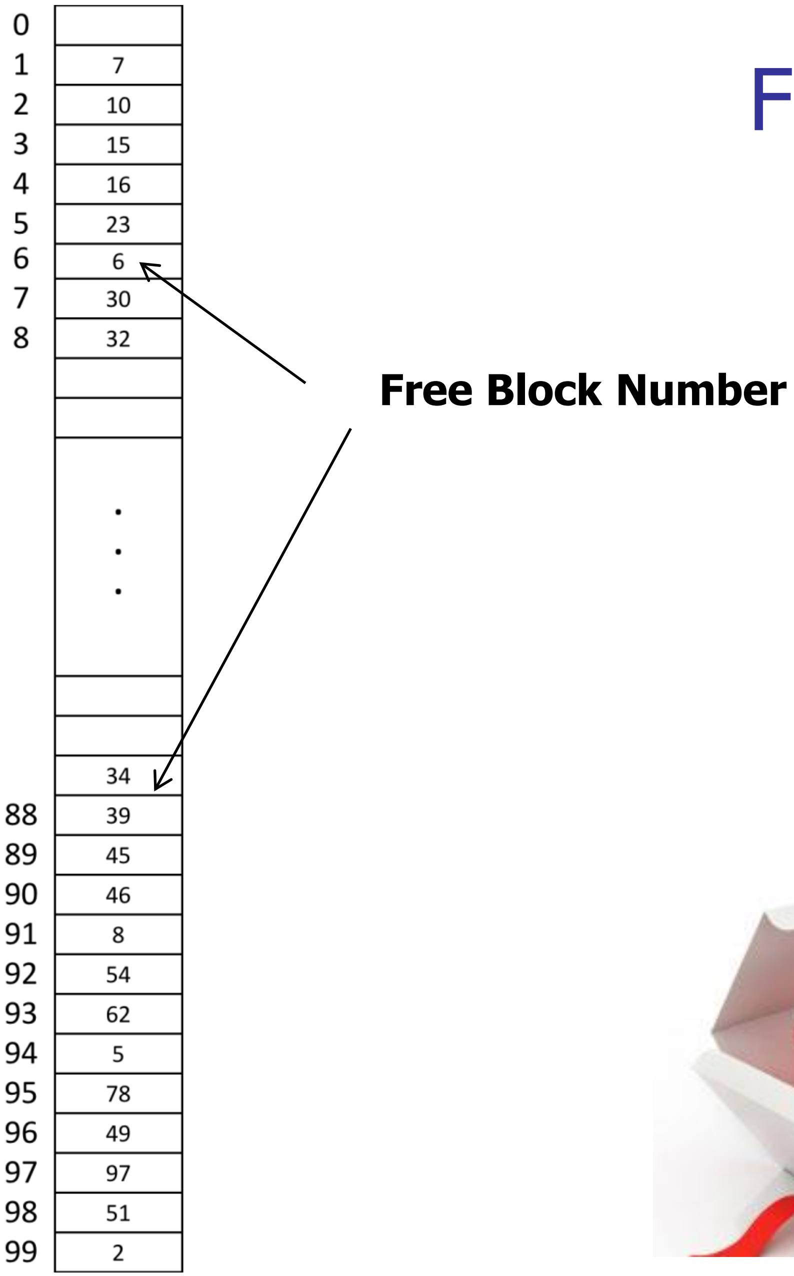
0	
1	7
2	10
3	15
4	16
5	23
6	6
7	30
8	32
	⋮
	34
88	39
89	45
90	46
91	8
92	54
93	62
94	5
95	78
96	49
97	97
98	51
99	2

Free Block Number

	0	1	2	3	4	5	6	7	8	9
0	MBR	1	2	3	4	5	6	7	8	9
1	10	11	12	13	14	15	16	17	18	19
2	20	21	22	23	24	25	26	27	28	29
3	30	31	32	33	34	35	36	37	38	39
4	40	41	42	43	44	45	46	47	48	49
5	50	51	52	53	54	55	56	57	58	59
6	60	61	62	63	64	65	66	67	68	69
7	70	71	72	73	74	75	76	77	78	79
8	80	81	82	83	84	85	86	87	88	89
9	90	91	92	93	94	95	96	97	98	99

Free List

Free List



0	
1	7
2	10
3	15
4	16
5	23
6	6
7	30
8	32
	. . .
	34
88	39
89	45
90	46
91	8
92	54
93	62
94	5
95	78
96	49
97	97
98	51
99	2

Free Block Number

Free List

Free List

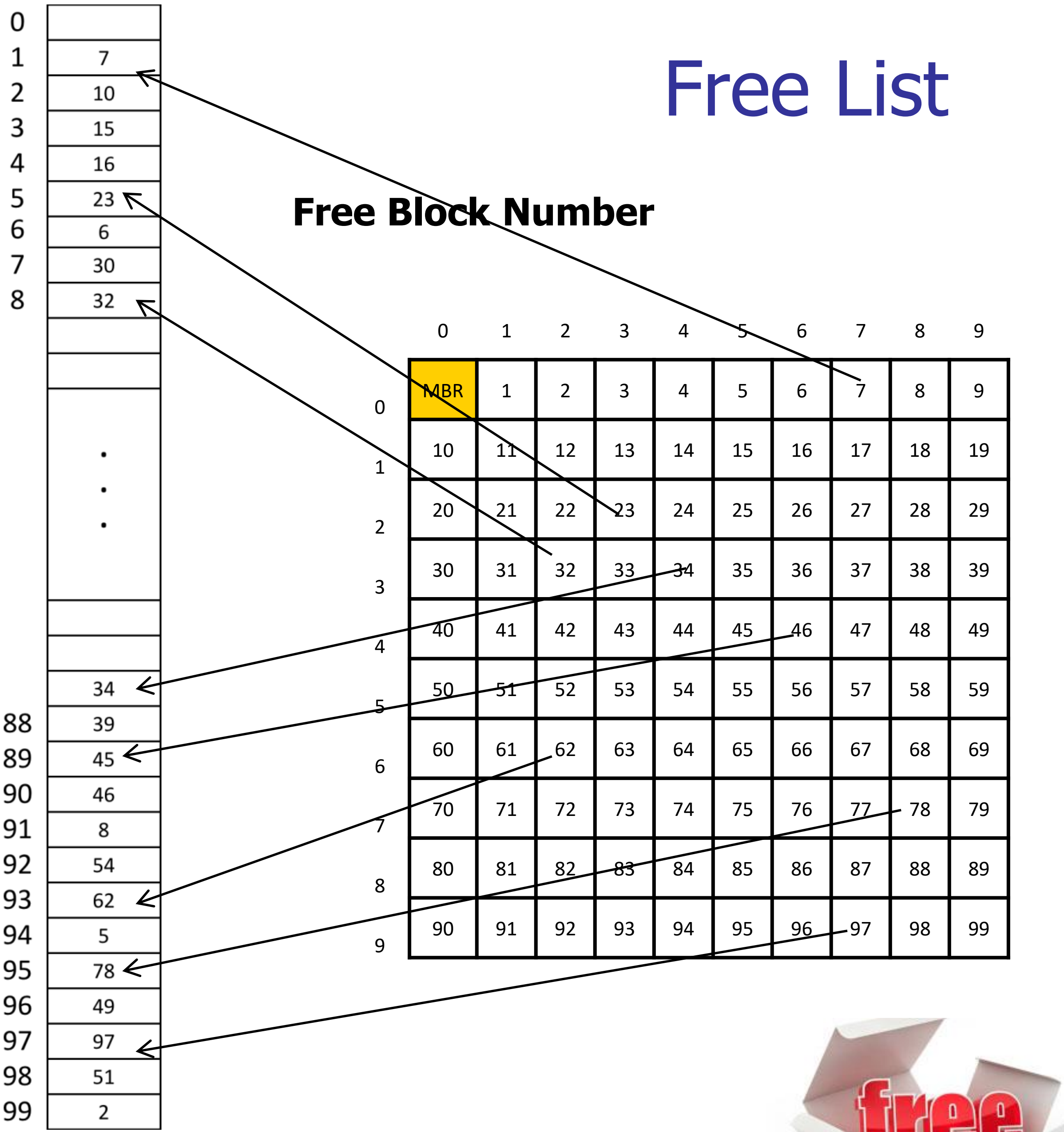
0	
1	7
2	10
3	15
4	16
5	23
6	6
7	30
8	32
	. . .
	34
88	39
89	45
90	46
91	8
92	54
93	62
94	5
95	78
96	49
97	97
98	51
99	2

Free Block Number

	0	1	2	3	4	5	6	7	8	9
0	MBR	1	2	3	4	5	6	7	8	9
1	10	11	12	13	14	15	16	17	18	19
2	20	21	22	23	24	25	26	27	28	29
3	30	31	32	33	34	35	36	37	38	39
4	40	41	42	43	44	45	46	47	48	49
5	50	51	52	53	54	55	56	57	58	59
6	60	61	62	63	64	65	66	67	68	69
7	70	71	72	73	74	75	76	77	78	79
8	80	81	82	83	84	85	86	87	88	89
9	90	91	92	93	94	95	96	97	98	99

Free List

Free List



Free Block Number

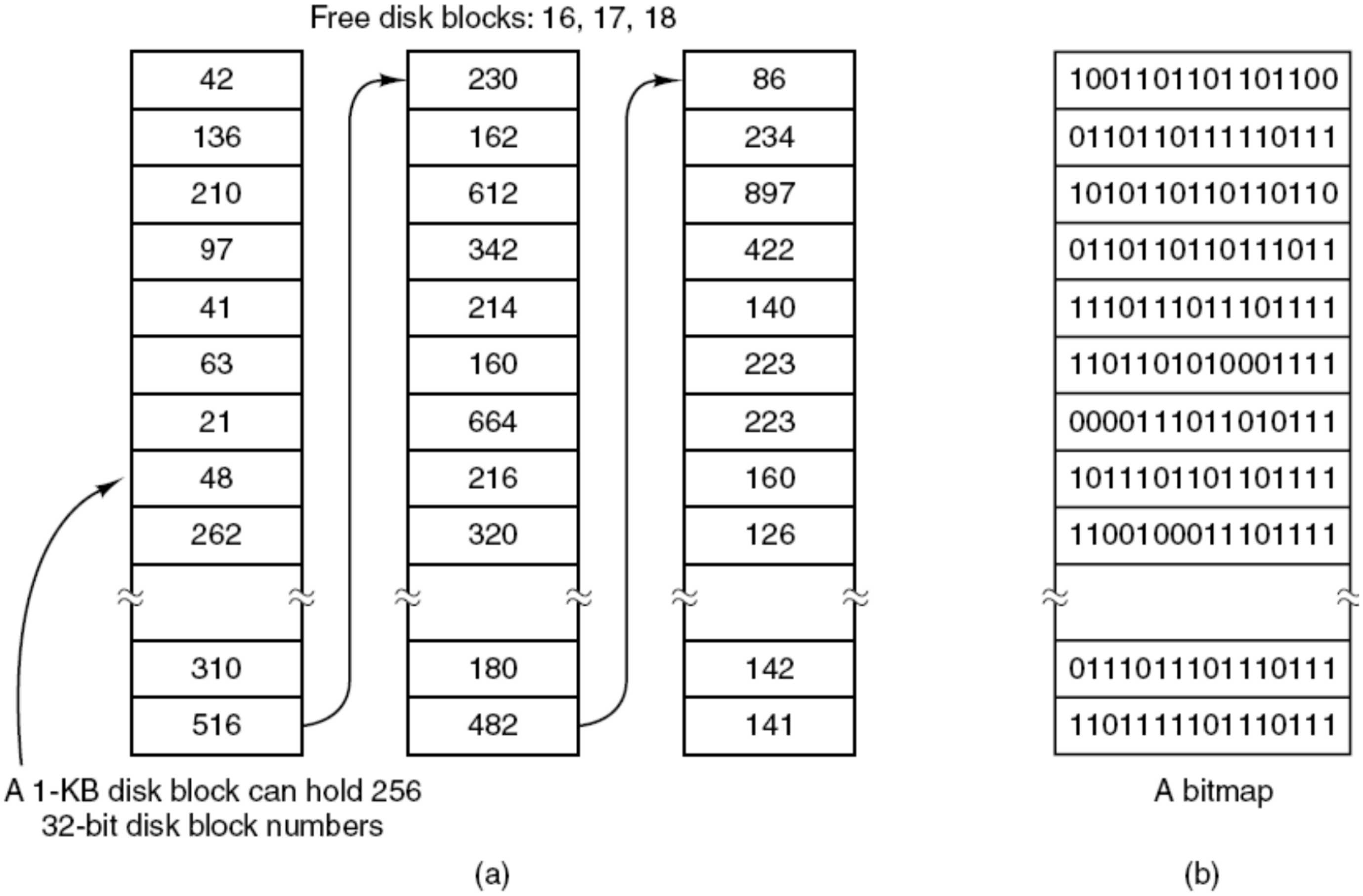
0	
1	7
2	10
3	15
4	16
5	23
6	6
7	30
8	32
	. . .
	34
88	39
89	45
90	46
91	8
92	54
93	62
94	5
95	78
96	49
97	97
98	51
99	2

	0	1	2	3	4	5	6	7	8	9
0	MBR	1	2	3	4	5	6	7	8	9
1	10	11	12	13	14	15	16	17	18	19
2	20	21	22	23	24	25	26	27	28	29
3	30	31	32	33	34	35	36	37	38	39
4	40	41	42	43	44	45	46	47	48	49
5	50	51	52	53	54	55	56	57	58	59
6	60	61	62	63	64	65	66	67	68	69
7	70	71	72	73	74	75	76	77	78	79
8	80	81	82	83	84	85	86	87	88	89
9	90	91	92	93	94	95	96	97	98	99

Free List

AST : Free list bitmap

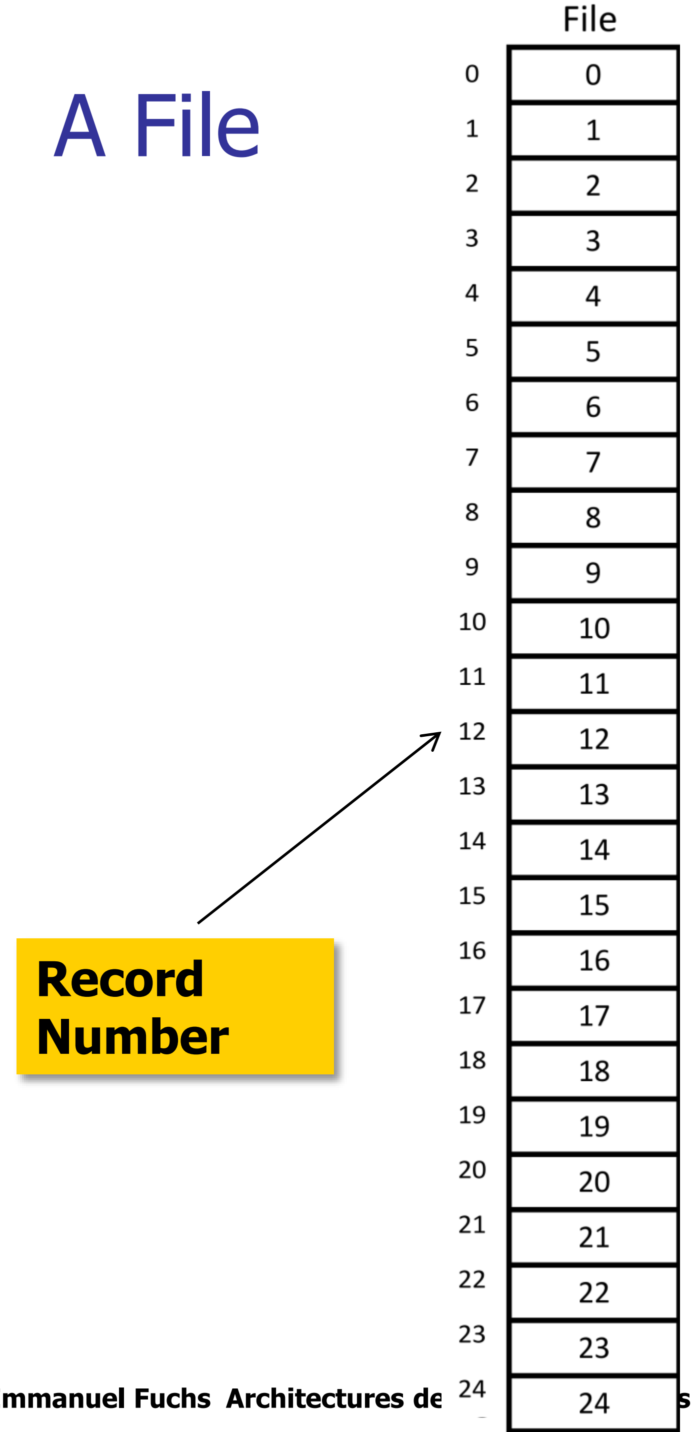
Keeping Track of Free Blocks (1)


Free disk blocks: 16, 17, 18
A 1-KB disk block can hold 256 32-bit disk block numbers
A bitmap
(a)
(b)

42	230	86	1001101101101100
136	162	234	0110110111110111
210	612	897	1010110110110110
97	342	422	0110110110111011
41	214	140	1110111011101111
63	160	223	1101101010001111
21	664	223	0000111011010111
48	216	160	1011101101101111
262	320	126	1100100011101111
≈	≈	≈	≈
310	180	142	0111011101110111
516	482	141	1101111101110111

Figure 4-22. (a) Storing the free list on a linked list. (b) A bit

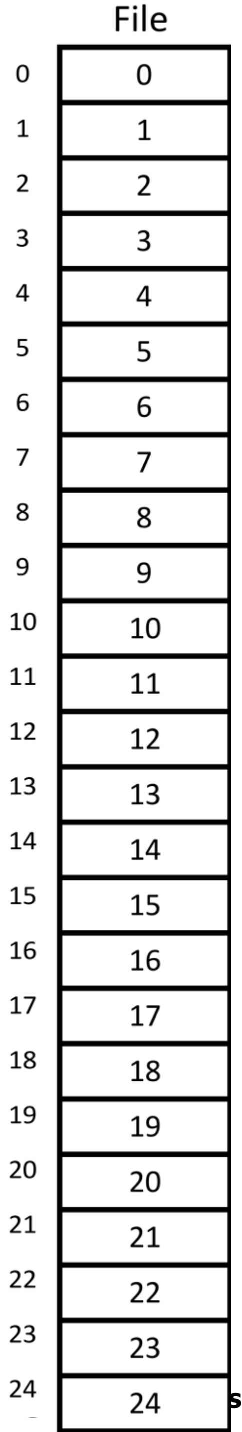
A File



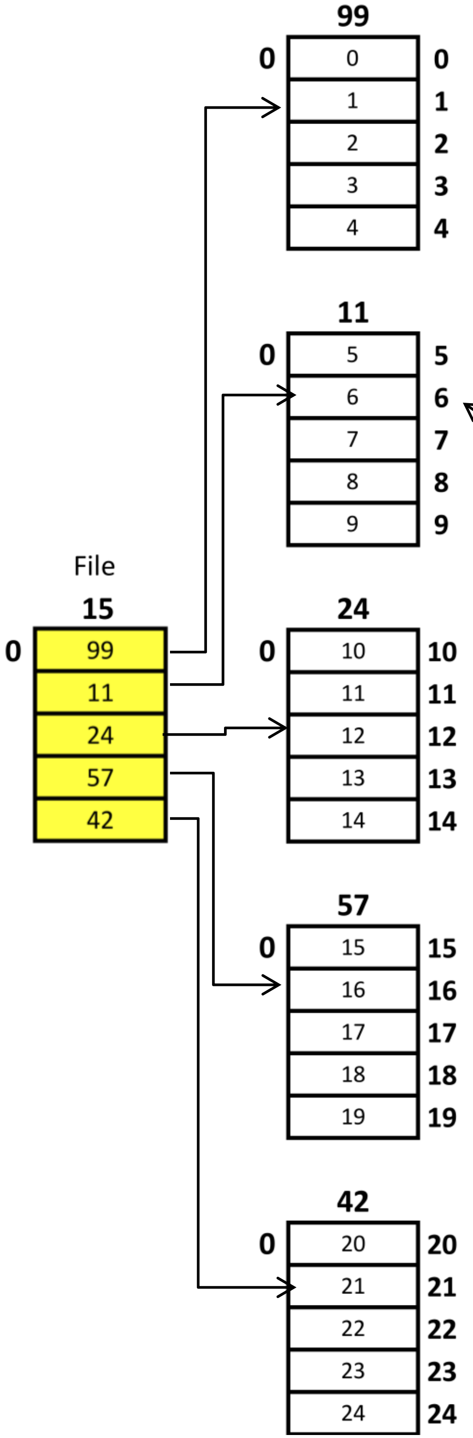
	File
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14
15	15
16	16
17	17
18	18
19	19
20	20
21	21
22	22
23	23
24	24

Record Number

A File

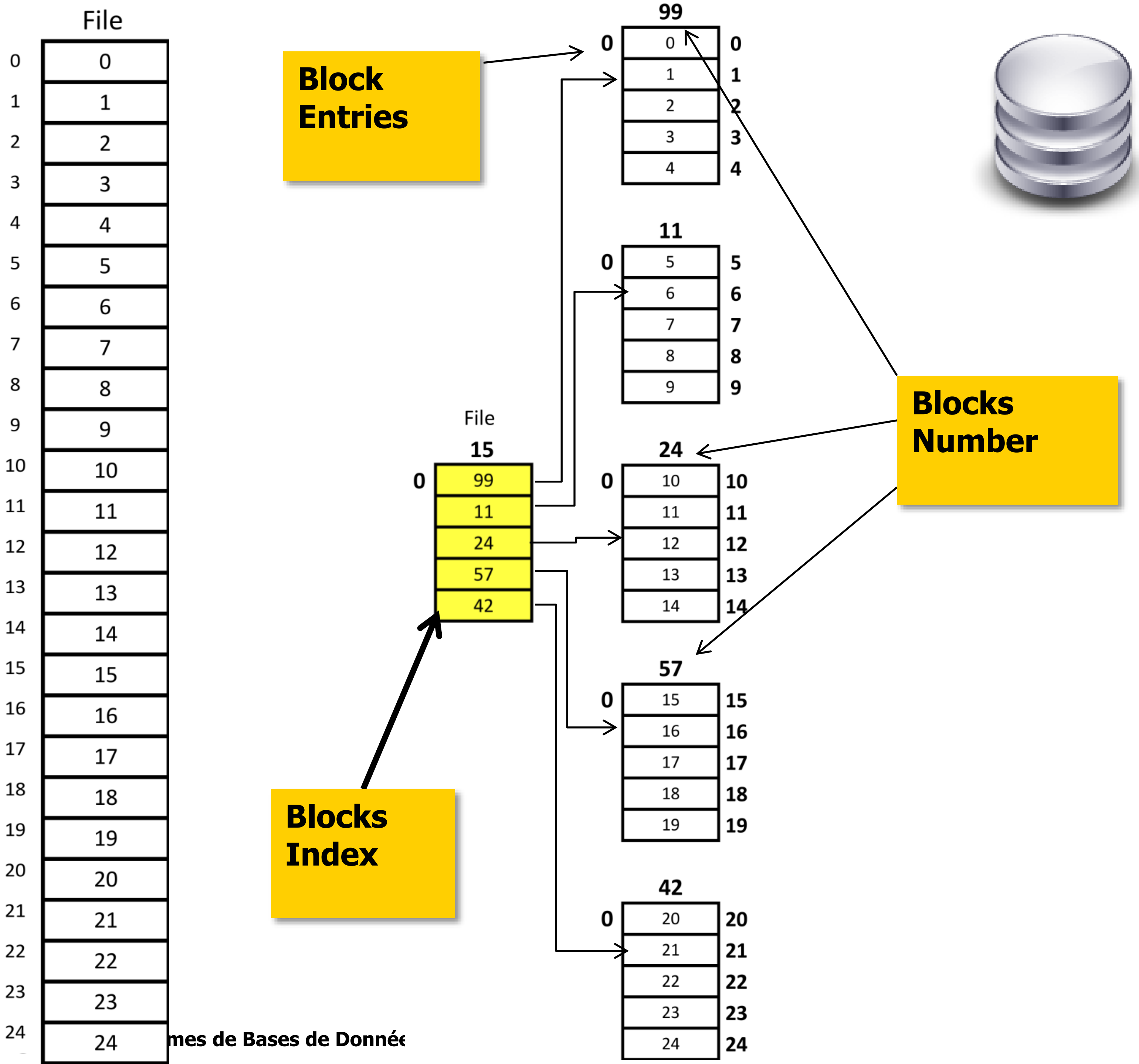


Record Number



Record Number

A File

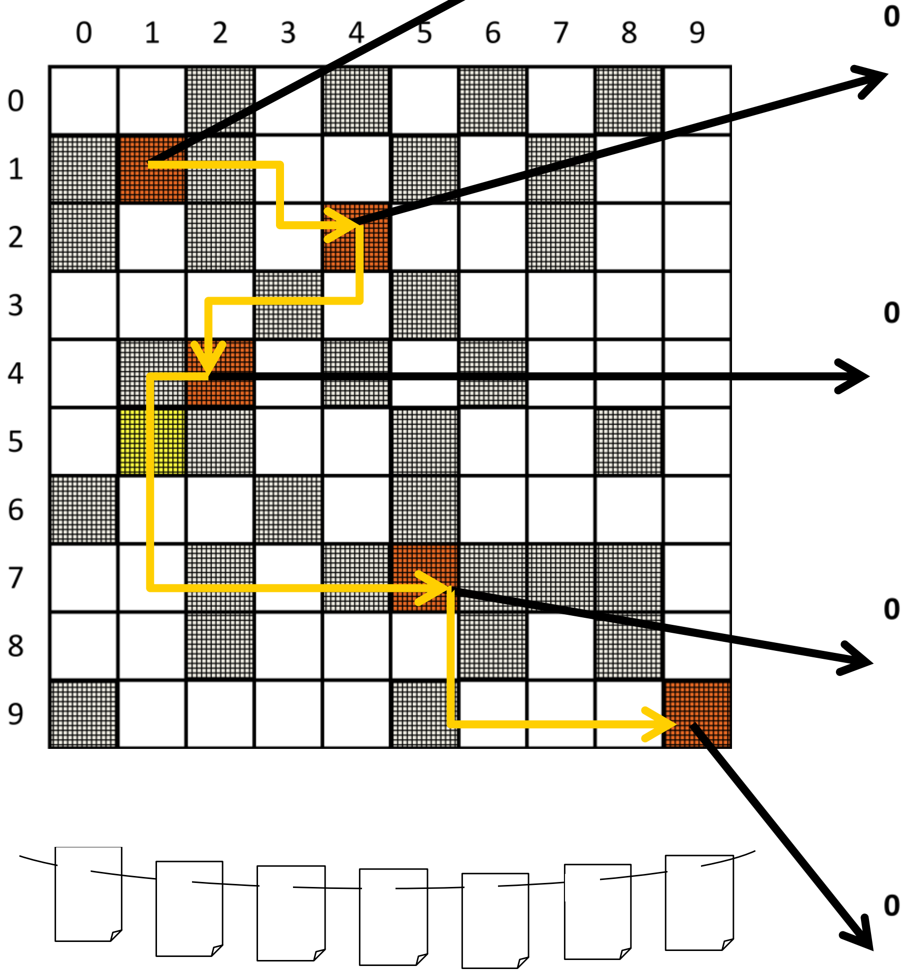


Block Entries

Blocks Number

Blocks Index

File



	11	
0	5	1
	6	2
	7	3
	8	4
	9	5

	24	
0	10	6
	11	7
	12	8
	13	9
	14	10

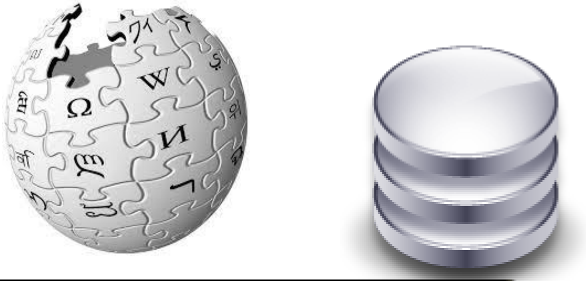
	42	
0	20	11
	21	12
	22	13
	23	14
	24	15

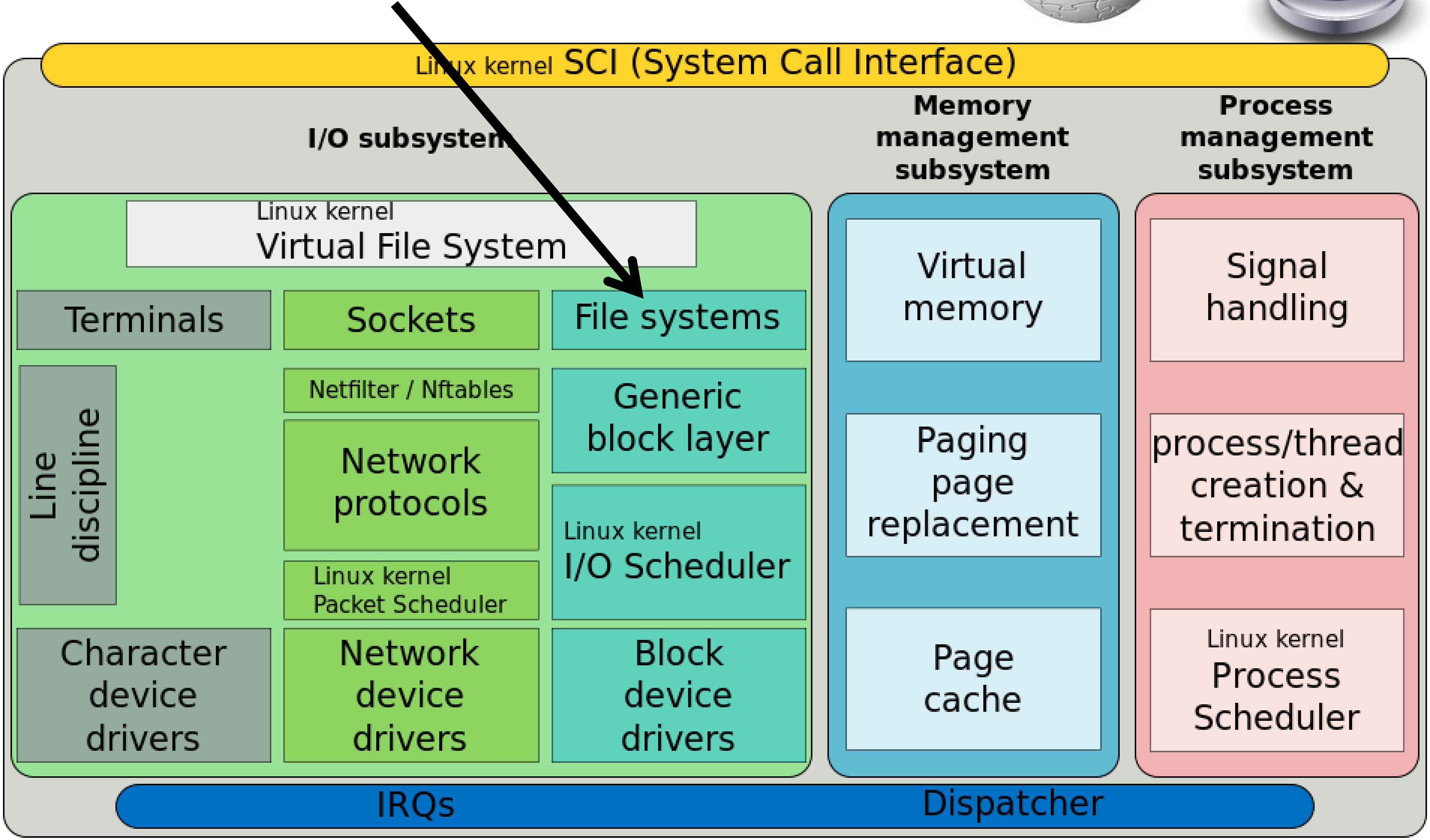
	57	
0	15	16
	16	17
	17	18
	18	19
	19	20

	99	
0	0	0
	1	1
	2	2
	3	3
	4	4

	File
0	5
1	6
2	7
3	8
4	9
5	10
6	11
7	12
8	13
9	14
10	20
11	21
12	22
13	23
14	24
15	15
16	16
17	17
18	18
19	19
20	0
21	1
22	2
23	3
24	4

Linux




Linux kernel SCI (System Call Interface)
I/O subsystem
Memory management subsystem
Process management subsystem
Linux kernel
Virtual File System
Terminals
Sockets
File systems
Line discipline
Netfilter / Nftables
Generic block layer
Network protocols
Linux kernel
I/O Scheduler
Linux kernel
Packet Scheduler
Character device drivers
Network device drivers
Block device drivers
Virtual memory
Paging page replacement
Page cache
Signal handling
process/thread creation & termination
Linux kernel
Process Scheduler
IRQs
Dispatcher