

## Esercizio FileSystem 1

In un disco con blocchi di 1 Kbyte ( $= 2^{10}$  byte), è definito un file system FAT.

Gli elementi della FAT sono in corrispondenza biunivoca con i blocchi fisici del disco.

Ogni elemento ha lunghezza di 3 byte e indirizza un blocco del disco.

Ogni file è descritto da una lista concatenata di indirizzi di blocchi, realizzata sulla FAT.

Il primo blocco di ogni file è identificato dalla coppia (*nomefile*, *indiceblocco*) contenuto nella rispettiva directory.

1. Qual è la massima capacità del disco, espressa in blocchi e in byte?
2. Quanti byte occupa la FAT?
3. Supponendo che il file *pippo* occupi (ordinatamente) i blocchi fisici 15, 30, 16, 64 e 40, quali sono gli elementi della FAT che descrivono il file e quale è il loro contenuto?

### Soluzione:

1. Capacità del disco: ..... blocchi ..... byte
2. Lunghezza della FAT: ..... byte
3. Elementi della FAT che indirizzano il file e loro contenuto:

ELEMENTO FAT	CONTENUTO

## Esercizio FileSystem 2

Un sistema operativo gestisce la memoria con paginazione dinamica con pagine di 2 Kbyte e utilizza un file system di tipo FAT-32 (con indirizzi a 32 bit) con blocchi di 2 Kbyte per gestire un disco di 10 Gbyte. La FAT è allocata sul disco a partire dal blocco 3.

Nel file system il file `foto` contenuto nella directory `personale` occupa 9 blocchi logici allocati come segue:

Blocco logico:	0	1	2	3	4	5	6	7	8
Blocco fisico:	398.203	1.716.882	106.987	2.448.123	12.186	65.637	4.876.999	3.161.002	908.543

Ad un dato istante di tempo, quando nessun blocco della FAT è caricato in memoria principale e la directory `personale` è caricata in memoria, il sistema operativo riceve una richiesta di lettura dei caratteri del file `foto` compresi tra 8.000 e 14.000 (estremi inclusi).

Si chiede:

1. Quali blocchi logici del file `foto` vengono letti dal sistema operativo
2. Quali blocchi fisici del file `foto` vengono letti dal sistema operativo
3. Quali blocchi fisici contenenti la FAT vengono letti dal sistema operativo
4. Quanti page fault causa l'operazione di lettura nell'ipotesi che i buffer destinati ad accogliere i dati letti dal file siano già allocati e presenti in memoria.

## SOLUZIONE

1. Blocco logico del file `foto` che contiene il byte 8.000: .....;  
Blocco logico del file `foto` che contiene il byte 14.000: .....;  
Blocchi logici del file `foto` letti dal sistema operativo: .....;
2. Blocchi fisici del file `foto` letti dal sistema operativo: .....;
3. Blocchi della FAT letti:  
Ogni blocco del disco contiene ..... puntatori della FAT, di conseguenza:  
- Il puntatore ..... è contenuto nel blocco ..... della FAT, che corrisponde al blocco ..... del disco  
- Il puntatore ..... è contenuto nel blocco ..... della FAT, che corrisponde al blocco ..... del disco  
- Il puntatore ..... è contenuto nel blocco ..... della FAT, che corrisponde al blocco ..... del disco  
Il puntatore ..... è contenuto nel blocco ..... della FAT, che corrisponde al blocco ..... del disco  
Quindi i blocchi fisici da leggere per le operazioni sulla FAT sono: .....
4. Page fault causati dall'operazione di lettura: .....

### Esercizio FileSystem 3

In un file system UNIX i blocchi del disco hanno ampiezza di 1Kbyte e gli i-node contengono 10 indirizzi diretti e 3 indirizzi indiretti. Tutti gli indirizzi hanno una lunghezza di 4 byte.

Si chiede:

1. la massima capacità del disco che ospita il file system, in blocchi e in byte
2. la massima dimensione dei file indirizzabili dall'i-node, in blocchi e in byte.

#### SOLUZIONE

1. Massima capacità del disco che ospita il file system: ..... blocchi; → ..... byte
2. considerato che:
  - lo i-node indirizza direttamente ..... blocchi
  - il blocco indiretto di primo livello puntato dall'indirizzo indiretto semplice indirizza ..... blocchi dati
  - il blocco indiretto di secondo livello puntato dall'indirizzo indiretto doppio indirizza ..... blocchi indiretti di primo livello, ciascuno dei quali indirizza ..... blocchi dati,
  - il blocco indiretto di terzo livello puntato dall'indirizzo indiretto triplo indirizza ..... blocchi indiretti di secondo livello, ciascuno dei quali indirizza ..... blocchi indiretti di primo livello, ciascuno dei quali indirizza ..... blocchi dati,

la massima dimensione dei file indirizzabili dall'i-node è pari a:

..... blocchi  
ovvero ..... byte.

## Esercizio FileSystem 4

In un file system UNIX i blocchi del disco hanno ampiezza di 1Kbyte e i puntatori ai blocchi sono a 32 bit. Gli i-node contengono, oltre agli altri attributi, 10 puntatori diretti e 3 puntatori indiretti.

Tenendo presente che il primo blocco del disco ha indice logico 0, si chiede:

1. il numero di puntatori che possono essere contenuti in un blocco indiretto;
2. l'indice logico del primo blocco e dell'ultimo blocco indirizzabili con puntatori diretti;
3. l'indice logico del primo blocco e dell'ultimo blocco indirizzabili con indirizzamento indiretto semplice;
4. l'indice logico del primo blocco e dell'ultimo blocco indirizzabili con indirizzamento indiretto doppio;
5. l'indice logico del primo blocco e dell'ultimo blocco indirizzabili con indirizzamento indiretto triplo;

Considerato il file (aperto) individuato dal file descriptor *fd*, la cui lunghezza corrente (in byte) è 278.538 e il cui *i-node* contiene i seguenti puntatori a blocchi:

Puntatore	0	1	2	3	4	5	6	7	8	9	10	11	12
Valore del puntatore	100	101	102	120	121	122	300	301	302	303	500	700	--

dove i blocchi indiretti 500, 700, e 800 hanno i seguenti contenuti parziali:

Blocco 500:

Indice di elemento nel blocco	0	1	2	3	4	5	.....
Valore del puntatore	304	305	306	307	308	309	.....

Blocco 700:

Indice di elemento nel blocco	0	1	2	3	4	5	.....
Valore del puntatore	800	801	802	850	851	852	.....

Blocco 800:

Indice di elemento nel blocco	0	1	2	3	4	5	.....
Valore del puntatore	1200	1201	1202	1203	1204	1205	.....

si chiede inoltre:

6. il numero di blocchi che compongono il file;
7. quali sono i blocchi indiretti che vengono letti per eseguire l'operazione *read(fd,&buf,1)* quando lo I/O pointer ha valore 12.298
8. quali sono i blocchi indiretti che vengono letti per eseguire l'operazione *read(fd,&buf,1)* quando lo I/O pointer ha valore 273.428.

## SOLUZIONE

1. il numero di puntatori che possono essere contenuti in un blocco indiretto è .....
2. il primo e l'ultimo blocco indirizzabili con puntatori diretti hanno rispettivamente indici logici ..... e .....;
3. il primo e l'ultimo blocco indirizzabili con indirizzamento indiretto semplice hanno rispettivamente indici logici .....;
4. il primo e l'ultimo blocco indirizzabili con indirizzamento indiretto doppio hanno rispettivamente indici logici ..... e .....;
5. il primo e l'ultimo blocco indirizzabili con indirizzamento indiretto triplo hanno rispettivamente indici logici ..... e .....;
6. l'ultimo carattere del file è contenuto nel blocco .....; quindi il file è composto da ..... blocchi
7. il carattere 12.298 sul quale è posizionato il puntatore di lettura, è contenuto nel blocco di indice logico ..... indirizzato dal .....  
Quindi per eseguire l'operazione *read(fd,&buf,1)* deve/devono essere letto il blocco/ letti i blocchi .....
8. il carattere 273.428 sul quale è posizionato il puntatore di lettura, è contenuto nel blocco ..... individuato dal puntatore di indice logico ..... indirizzato dal .....  
Quindi per eseguire l'operazione *read(fd,&buf,1)* deve/devono essere letto il blocco/ letti i blocchi .....

## Esercizio FileSystem 5

Si consideri la chiamata *read(4, &buf, 2000)* di un sistema simile a UNIX, dove il file descriptor 4 corrisponde all'i-node 15.

Lo i-node contiene 5 indirizzi di blocchi diretti, che hanno rispettivamente valore 512, 567, 45, 34, 28, oltre agli indirizzi di 2 blocchi indiretti. La lunghezza degli indirizzi è di 2 byte.

Gli indirizzi 0, 1, 2, 3, 4 del blocco indiretto di primo livello raggiungibile con indirizzamento indiretto semplice hanno ordinatamente i valori 56, 47, 67, 89, 23.

I blocchi del disco hanno ampiezza di 1024 byte e la lunghezza corrente del file è di 10.000 byte.

Il puntatore alla posizione corrente di lettura ha il valore 8500.

Domande:

1. Quali blocchi fisici vengono letti per eseguire l'operazione ?
2. Quanti caratteri vengono copiati in *buf* da ogni blocco interessato alla lettura?
3. Qual è il valore intero restituito dalla chiamata?

## Soluzione:

Considerato che:

Il puntatore di lettura è posizionato sul carattere ..... del blocco logico .....

Il blocco fisico corrispondente a questo blocco logico è individuato dall'indirizzo .....

Il carattere EOF occupa la posizione ..... del blocco logico .....

Il blocco fisico corrispondente a questo blocco logico è individuato dall'indirizzo .....

Ogni blocco indice contiene ..... indirizzi,

Il file è necessariamente aperto al momento della chiamata di sistema *read*;

1. Per eseguire l'operazione si leggono i seguenti blocchi:

.....  
.....  
.....

2. Numero di caratteri copiati da ciascun blocco interessato alla lettura:

..... caratteri dal blocco .....

..... caratteri dal blocco .....

.....

3. Valore restituito dalla chiamata: .....

## Esercizio FileSystem 6

In un file system UNIX dove gli i-node occupano 1 blocco, si consideri il file individuato dal pathname *compiti/terzo\_appello*, relativo alla directory corrente. La directory corrente è caricata in memoria, mentre la directory *compiti* (che occupa 1 blocco) e il rispettivo i-node risiede su disco. Tutti gli i-node occupano 1 blocco. Gli i-node della directory *compiti* e del file *terzo\_appello* sono caricati, rispettivamente, nei blocchi fisici 35 e 71. Gli indirizzi diretti dello i-node della directory *compiti* hanno i valori:

Indirizzo diretto	0	1	2	3	4	5	6	7	8	9
Valore	25									

Gli indirizzi diretti dello i-node del file *terzo\_appello* hanno i valori:

Indirizzo diretto	0	1	2	3	4	5	6	7	8	9
Valore	29	30	42	64	65	66	78	90	91	93

Quali sono i blocchi fisici da leggere per portare in memoria i primi 2 blocchi del file *terzo\_appello*, supponendo che questo file sia già stato aperto?

## Soluzione

Vengono letti i blocchi:

1. ....
2. ....
3. ....
4. ....
- .....
- .....
- .....

## Esercizio FileSystem 7

In un file system UNIX si consideri il file */usr/tizio/appunti/esercitazione*, creato dall'utente *tizio*. I diritti associati alle directory *usr*, *tizio*, *appunti* e al file *esercitazione* sono i seguenti

	<i>owner (r w x)</i>	<i>group (r w x)</i>	<i>others (r w x)</i>
<i>usr</i>	<i>1 0 1</i>	<i>1 0 1</i>	<i>1 0 1</i>
<i>tizio</i>	<i>1 1 1</i>	<i>0 0 1</i>	<i>0 0 1</i>
<i>appunti</i>	<i>1 1 1</i>	<i>1 0 1</i>	<i>1 0 0</i>
<i>esercitazione</i>	<i>1 1 0</i>	<i>1 1 0</i>	<i>1 0 0</i>

Quali tra le operazioni di lettura, scrittura e cancellazione possono essere eseguite sul file *esercitazione* dall'utente *caio* se:

1. *caio* e *tizio* appartengono allo stesso gruppo;
2. *caio* e *tizio* appartengono a gruppi diversi.

## Soluzione

IPOTESI	LETTURA	SCRITTURA	CANCELLAZIONE
<i>caio</i> e <i>tizio</i> appartengono allo stesso gruppo			
<i>caio</i> e <i>tizio</i> appartengono a gruppi diversi.			

## Esercizio FileSystem 8

Ricordando che per la creazione di un *pipe* il sistema UNIX utilizza la chiamata di sistema:

```
int pipe (int fd[2])
```

si chiede quali chiamate di sistema sono utilizzate per:

1. scrivere k byte nel pipe
2. leggere k byte dal pipe
3. chiudere il pipe per la scrittura.

## Soluzione

1. per scrivere k byte: .....
2. per leggere k byte : .....
3. per chiudere il pipe in scrittura: .....



## Esercizio FileSystem 9

Si consideri un File System simile a NTFS che alloca i file in sequenze contigue (*run*) individuate mediante coppie del tipo (*inizio, lunghezza*), dove *inizio* è l'indice del primo blocco fisico del *run* e *lunghezza* esprime il numero di blocchi che la compongono. Ogni file è descritto da un *Master Record* che contiene, oltre ad altri attributi, una o più coppie (*inizio, lunghezza*). Il File System è ospitato da un disco con  $NCilindri = 50$ ,  $NFacce = 4$  e  $NSettori = 20$ . Il tempo necessario per percorrere un settore è di  $0,1 \text{ msec}$  e il tempo medio di esecuzione di un'operazione di *seek* (comprensivo del ritardo rotazionale per raggiungere il primo settore indirizzato) è di  $5 \text{ msec}$ .

A un certo tempo viene eseguita l'operazione *read(filename, &buffer, Ncaratteri)*, per effetto della quale si leggono 10 blocchi a partire dal nono blocco del file, cioè da quello di indice logico 8. Nel *Master Record* del file *filename* sono definite, nell'ordine, i seguenti *run* contigui:

1. (1525, 15)
2. (3170, 12)

dove l'indice di blocco 1525 corrisponde alla terna (*cilindro*= 19, *faccia*= 0, *settore*= 5) e l'indice di blocco 3170 corrisponde alla terna (*cilindro*= 39, *faccia*= 2, *settore*= 10).

Si calcoli il tempo necessario per eseguire la lettura, supponendo che le teste di lettura scrittura siano inizialmente posizionate sul cilindro 12 e che tempo di esecuzione delle eventuali operazioni di *seek* sia sempre uguale a quello medio.

## SOLUZIONE

- Il primo blocco da leggere ha indice logico ..... e indice fisico .....; pertanto il primo blocco fisico estratto dal primo *run* 1 ha indice .....
- Numero di blocchi estratti dal primo *run*: ..... blocchi
- Numero di cilindri su cui è distribuito il primo *run*: .....
- Tempo necessario per estrarre i blocchi del primo *run*: ..... msec
- Indice del primo blocco fisico estratto dal secondo *run*: blocco .....
- Numero di blocchi estratti dal secondo *run*: ..... blocchi
- Numero di cilindri su cui è distribuito il secondo *run*: .....
- Tempo necessario per estrarre i blocchi del secondo *run*: ..... msec
- Numero di operazioni di *seek* eseguite: .....
- Tempo totale impiegato: ..... msec.

Esercizio FileSystem 10

Data la seguente matrice di protezione:

	File 1	File 2	File 3	File 4	Scanner	Stampante
Utente Mario	R,W	X	R,X		R	
Utente Franco		R,X	W			
Utente Rosa	R	W				W
Utente Nina	R		R,X	R,W	R	W

convertirla in:

- 1. liste di capability
- 2. liste di controllo degli accessi

SOLUZIONE

1. liste di capability:

.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....

3. liste di controllo degli accessi:

.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....