Esercizio FAT 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: 2^{24} blocchi (3 byte=24 bit) ◊ 2^{34} byte
2. Lunghezza della FAT: 2^{24} * 3 byte ◊ 48 MByte
3. Elementi della FAT che indirizzano il file e loro contenuto:

ELEMENTO FAT	CONTENUTO
15	30
30	16
16	64
64	40
40	Ø

Esercizio FAT 2

Un sistema operativo gestisce la memoria con paginazione a domanda con pagine di 2 Kbyte e utilizza un filesystem di tipo FAT-32 (con indirizzi a 32 bit) con blocchi di 2 Kbyte, ospitato da un disco di 10 Gbyte. La copia permanente della FAT è allocata sul disco a partire dal blocco 3. Gli elementi della FAT sono in corrispondenza biunivoca con i blocchi del disco, compresi i blocchi 0, 1 e 2 (riservati al codice di boot e ad altri dati del filesystem) e quelli, a partire dal blocco 3 occupati dalla FAT. I blocchi successivi all'ultimo blocco occupato dalla FAT sono i blocchi dati. Pertanto i valori legittimi dei puntatori contenuti nella FAT sono quelli non minori di 3+ LunghezzaFAT, dove LunghezzaFAT è espressa in blocchi.

Nel filesystem 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 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 800 e 14.000 (estremi inclusi).

Si chiede:

1. La lunghezza della FAT, espressa in numero di elementi e in numero di blocchi.
2. L'intervallo dei valori legittimi per gli indici degli elementi della FAT (e per i puntatori)
3. Quali blocchi logici del file `foto` sono interessati alla lettura
4. Quali blocchi fisici del file `foto` devono essere letti
5. Quali blocchi fisici contenenti la FAT vengono letti dal sistema operativo
6. 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

Premessa: ogni blocco del disco contiene $2*2^{10}/4$ (2KB/4 byte)= 2^9 elementi della FAT.

1. Lunghezza della FAT: $10*2^{30}/2*2^{10}$ (10GB/2KB)= $5*2^{20}$ elementi ◊ $5*2^{20}/2^9$= $5*2^{11}$ blocchi
2. I valori legittimi per gli indici degli elementi della FAT sono quelli compresi nell'intervallo [3+ $5*2^{11}$, $5*2^{20}$)= [10.243, 5.242.880)
3. Blocco logico del file `foto` che contiene il byte 800: 800 **div** 2048 = 0;
   Blocco logico del file `foto` che contiene il byte 14.000: 14.000 **div** 2048 = 6;
   Blocchi logici del file `foto` interessati alla lettura: 0, 1, 2, 3, 4, 5, 6
4. Blocchi fisici del file `foto` che devono essere letti:
   398.203 (puntatore contenuto nella directory `personale`, che è caricata in memoria)
   1.716.882 (puntatore contenuto nell'elemento della FAT di indice 398.203)
   106.987 (puntatore contenuto nell'elemento della FAT di indice 1.716.882)
   2.448.123 (puntatore contenuto nell'elemento della FAT di indice 106.987)
   12.186 (puntatore contenuto nell'elemento della FAT di indice 2.448.123)
   65.637 (puntatore contenuto nell'elemento della FAT di indice 12.186)
   4.876.999 (puntatore contenuto nell'elemento della FAT di indice 65.637)
5. Blocchi della FAT che devono essere letti:
   - Il puntatore 1.716.882 è contenuto nel blocco 398.203 **div** 512= 777 della FAT;
   - Il puntatore 106.987 è contenuto nel blocco 1.716.882 **div** 512= 3.353 della FAT;
   - Il puntatore 2.448.123 è contenuto nel blocco 106.987 **div** 512= 208 della FAT;
   - Il puntatore 12.186 è contenuto nel blocco 2.448.123 **div** 512= 4.781 della FAT;
   - Il puntatore 65.637 è contenuto nel blocco 12.186 **div** 512= 23 della FAT;
   - Il puntatore 4.876.999 è contenuto nel blocco 65.637 **div** 512= 128 della FAT.

Quindi i blocchi fisici da leggere per le operazioni sulla FAT sono: 777, 3.353, 208, 4.781, 23 e 128

6. Numero di page fault causati dall’operazione di lettura: 6
   I page faults sono determinati dal trasferimento in memoria dei blocchi del disco che contengono gli elementi della FAT interessati dall’operazione.

Esercizio FFS 1

In un filesystem 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 filesystem, 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: 2^{32} blocchi (4 byte=32 bit) ◊ $2^{32}*2^{10}=2^{42}$ byte (4 Tbyte)
2. considerato che:
   a. - lo i-node indirizza direttamente 10 blocchi
   b. - ogni blocco indiretto contiene 2^{10} **div** 4= 2^8 indirizzi
   c. - il blocco indiretto di primo livello puntato dall'indirizzo indiretto semplice indirizza 2^8 blocchi dati
   d. - il blocco indiretto di secondo livello puntato dall'indirizzo indiretto doppio indirizza 2^8 blocchi indiretti di primo livello, ciascuno dei quali indirizza 2^8 blocchi dati,
   e. - il blocco indiretto di terzo livello puntato dall'indirizzo indiretto triplo indirizza 2^8 blocchi indiretti di secondo livello, ciascuno dei quali indirizza 2^8 blocchi indiretti di primo livello, ciascuno dei quali indirizza 2^8 blocchi dati,

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

   f. $10 + 2^8 + 2^{16} + 2^{24} = 10+ 256+ 65536+ 16777216 =16.843.018$ blocchi ◊ 16.843.018 Kbyte ◊
      16.843.018 / 2^{20} Gbyte = 16,06 Gbyte

Esercizio FFS 2

In un filesystem 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 2^10/4 (espresso in byte) = 2^8= 256;
2. il primo e l'ultimo blocco indirizzabili con puntatori diretti hanno rispettivamente indici logici 0 e 9;
3. il primo e l'ultimo blocco indirizzabili con indirizzamento indiretto semplice hanno rispettivamente indici logici 10 e (10+ 2^8)- 1= 265;
4. il primo e l'ultimo blocco indirizzabili con indirizzamento indiretto doppio hanno rispettivamente indici logici 10+ 2^8= 266 e (10+ 2^8+ 2^16)- 1= 65.801;
5. il primo e l'ultimo blocco indirizzabili con indirizzamento indiretto triplo hanno rispettivamente indici logici 10+ 2^8+ 2^16= 65.802 e (10+ 2^8+ 2^16+ 2^24)- 1= 16.843.017;
6. l'ultimo carattere del file è contenuto nel blocco (278.538- 1) ***div*** 2^10= 272; quindi il file è composto da 273 blocchi
7. il carattere *12.298*, sul quale è posizionato il puntatore di lettura, è contenuto nel blocco di indice logico *BloccoLogico=12.298* ***div*** 2^10= 12.

Poiché BloccoLogico≥10 si utilizza l'indirizzamento indiretto mediante un blocco indice.

Il *BloccoIndicePrimoLiv* utilizzato ha indice *ind1= (12- 10) div 256= 0;* il puntatore occupa in questo blocco la posizione *PuntatoreNelBloccoPrimoLiv= 12-10) mod 256= 2*

Poiché *ind1=0,* il blocco indice *BloccoIndicePrimoLiv* è raggiunto tramite il puntatore *puntatore[10]* dello i-node e occupa nel disco il blocco di indice *500.* Il blocco dati da leggere è raggiunto tramite il puntatore *BloccoIndicePrimoLiv[2] e* occupa nel disco il blocco di indice *306.*

Quindi per eseguire l'operazione *read(fd,&buf,1*) deve essere letto il blocco indiretto di indice 500.

8. Il carattere *273.428,* sul quale è posizionato il puntatore di lettura, è contenuto nel blocco di indice logico *BloccoLogico=273.428* ***div*** 2^10= 267.

   Poiché BloccoLogico≥10 si utilizza l'indirizzamento indiretto mediante un blocco indice.

   Il *BloccoIndicePrimoLiv* utilizzato ha indice *ind1= (267- 10) div 256= 1;* il puntatore occupa in questo blocco la posizione *PuntatoreNelBloccoPrimoLiv= ((257-10) mod 256= 1. (Più semplicemente, questo risultato si poteva dedurre dalla risposta 4).*

   Poiché *ind1> 0,* il blocco *BloccoIndicePrimoLiv* è raggiunto tramite un puntatore *PuntatoreNelBloccoSecondoLiv* contenuto in un blocco indice di secondo livello. L'indice di questo blocco è i*nd2= (1- 1) div 256= 0* e il puntatore utilizzato occupa in questo blocco la posizione *PuntatoreNelBloccoSecondoLiv = (1- 1) mod 256= 0. (Più semplicemente, questo risultato si poteva dedurre dalla risposta 4).*

   Poiché *ind2=0,* il blocco indice *BloccoIndiceSecondoLiv* è raggiunto tramite il puntatore *puntatore[11]* dello i-node e occupa nel disco il blocco di indice *700.* Il blocco *BloccoIndicePrimoLiv* è raggiunto tramite il puntatore *BloccoIndiceSecondoLiv[0]* e occupa nel disco il blocco di indice *800.* Il blocco dati da leggere è raggiunto tramite il puntatore*BloccoIndicePrimoLiv[10].*

   Quindi per eseguire l'operazione *read(fd,&buf,1*) devono essere letti il blocco indiretto di secondo livello 700 e il blocco indiretto di primo livello 800.

Esercizio FFS 3

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 file contiene 10.000 byte, l'operazione restituirà 1.500 caratteri e terminerà al raggiungimento del carattere di EOF, successivo all'ultimo carattere di informazione.
Il puntatore di lettura è posizionato sul carattere 8500 **mod** 1024 = 308 del blocco logico 8500 **div** 1024 = 8.
Il carattere EOF occupa la posizione 10.000 **mod** 1024 = 784 del blocco logico 10.000 **div** 1024 = 9.
Ogni blocco indice contiene 1024/2= 512 indirizzi.
Il blocco fisico corrispondente al blocco logico di indice 8 è individuato dall'indirizzo (8-5) **mod** 512= 3 del blocco indiretto di primo livello (8-5) **div** 512= 0, a sua volta individuato con indirizzamento indiretto semplice.
Il blocco fisico corrispondente al blocco logico di indice 9 è individuato dall'indirizzo (9-5) **mod** 512= 4 del blocco indiretto di primo livello (8-5) **div** 512= 0, a sua volta individuato con indirizzamento indiretto semplice.
Quindi:
1. Considerato che il file è necessariamente già aperto e che pertanto lo i-node è caricato in memoria,
per eseguire l'operazione si leggono i seguenti blocchi:

- blocco indiretto di primo livello di indice 0, raggiungibile con indirizzamento indiretto semplice
- Blocco fisico 89, corrispondente al blocco logico 8 e individuato dall'indirizzo 3 di questo blocco indiretto. Dal blocco fisico 89 si leggono 1024-308= 716 caratteri e pertanto rimangono da leggere 1500-716= 784 caratteri
- Blocco fisico 23, corrispondente al blocco logico 9 e individuato dall'indirizzo 4 del medesimo blocco indiretto. Dal blocco fisico 23 si leggono caratteri fino a raggiungere il carattere EOF: quindi 784 caratteri

2. Numero di caratteri copiati da ciascun blocco interessato alla lettura:
   - 716 caratteri dal blocco 89
   - 784 caratteri dal blocco 23

3. Valore restituito dalla chiamata: 716+ 784= 1500

Esercizio FFS 4

In un filesystem UNIX i blocchi del disco hanno una lunghezza di 1024 caratteri e 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 directo*ry compiti* (che occupa 1 blocco) e il rispettivo i-node risiede su disco.
Gli i-node della directory *compiti* e del file *terzo_appello* risiedono, 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 trasferire in memoria per leggere i primi 2048 caratteri del file *terzo_appello,* supponendo che questo file sia già stato aperto?

Soluzione

Lo i-node del file *terzo_appello* risiede in memoria, dove è stato caricato al momento dell'apertura.

I blocchi, che devono essere letti sul disco sono i seguenti:

1. blocco 35 , che contiene lo i-node di *compiti*
2. blocco 25 , che contiene la directory *compiti*
3. blocco 29 , che contiene il primo blocco di *terzo_appello*
4. blocco 30 , che contiene il secondo blocco di *terzo_appello*

Esercizio NTFS 1

Si consideri un File System simile a NTFS che alloca i file in sequenze contigue *(extent)* individuate mediante coppie del tipo *(inizio, lunghezza),* dove *inizio* è l'indice del primo blocco fisico dell'*extent* 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 Filesystem è ospitato da un disco con *NCilindri= 50*, *NFacce= 4* e *NSettori= 20*. Il tempo necessario per percorrere un settore è di *0,1 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 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 *extent* 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

1. Il primo blocco da leggere ha indice logico 8 e indice fisico 1533; pertanto il primo blocco fisico estratto dal primo *extent* 1 ha indice 1533
2. Numero di blocchi estratti dal primo *extent*: 15- 8= 7 blocchi
3. Numero di cilindri su cui è distribuito il primo *extent*: 1
4. Tempo necessario per estrarre i blocchi del primo *extent*: 7 * 0,1= 0,7 msec
5. Indice del primo blocco fisico estratto dal secondo *extent*: blocco 3170
6. Numero di blocchi estratti dal secondo *extent*: 3 blocchi
7. Numero di cilindri su cui è distribuito il secondo *extent*: 1
8. Tempo necessario per estrarre i blocchi del secondo *extent*: 3* 0,1= 0,3 msec
9. Numero di operazioni di *seek* eseguite: 2
10. Tempo totale impiegato: 2*5 + 0,7+ 0,3= 11 msec.

Esercizio NTFS 2 (run=extent)

Si consideri un File System simile ad 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 della sequenza e *lunghezza* esprime il numero di blocchi che la compongono. Ogni file è descritto da un *MFT Record* che contiene, oltre ad altri attributi, una o più coppie *(inizio, lunghezza)*. Il File System è ospitato da un disco con *NCilindri= 300*, *NFacce= 2* e *NSettori= 800*, in cui il tempo di seek è proporzionale al numero di cilindri attraversati, pari a 0,01 msec per ogni cilindro attraversato e il periodo di rotazione è di 20 msec, per cui il tempo impiegato per percorrere un settore è di 0,025 msec. Nel file system la dimensione del blocco è la stessa del settore, che è pari a 1Kbyte.

Nel file system è presente il file *luna*, di dimensione 32.000 Byte, allocati in due *run*: (95.000,19) e (60.001,13), dove l'indice di blocco 95.000 corrisponde alla terna (cilindro= 59, faccia= 0, settore= 600) e l'indice di blocco 60.001corrisponde alla terna (cilindro= 37, faccia= 1, settore= 1);

Ad un certo tempo, quando l'I/O pointer del file *luna* contiene 11.000, sul fine *luna* vengono eseguite le operazioni *read(luna, &buffer, 9000)* e *write(luna, &buffer, 9000)*.

Si calcoli il tempo necessario per eseguire le due operazioni, supponendo che siano eseguite sul disco in sequenza e che le testine di lettura/scrittura siano inizialmente posizionate sul cilindro 7.

Il tempo di esecuzione di ogni operazione è uguale alla somma dell'eventuale tempo di seek, del ritardo rotazionale (tempo necessario per raggiungere il settore indirizzato) e del tempo di percorrenza del settore indirizzato. Si suppone che il tempo necessario per raggiungere un qualsiasi settore dopo un'operazione di seek sia sempre uguale a un periodo di rotazione. Il controllore è dotato di sufficiente capacità di buffering ed è sempre in grado di accettare senza ritardo i dati letti dal disco o quelli da scrivere sul disco. Si assume inoltre che i comandi al disco vengano eseguiti in ordine FIFO.

Soluzione

read(luna, &buffer, 9000):

- Numero di byte letti dal 1' run: 19456-11.000= 8456
- Indice logico del primo blocco letto dal 1' run : 11000 div 1024= 10
- Blocchi estratti dal 1' run: 95.010 – 95.018
- Numero di cilindri su cui è distribuita la sequenza nel primo run: 1
- Tempo di seek per raggiungere il cilindro: 0,01* (59-7)=0,52 msec
- Tempo necessario per raggiungere il settore nel cilindro: 20 msec
- Tempo necessario per estrarre i blocchi della sequenza dal primo run: 9 * 0,025= 0,225 msec
- Tempo totale per estrarre i blocchi dalla prima sequenza: 20,745 msec

- Numero di byte letti dal 2' run: 9000 – 8456=544
- Indice logico del primo blocco letto dal 2' run : 19
- Blocchi estratti dal 2' run: 60.001
- Numero di cilindri su cui è distribuita la sequenza nel secondo run: 1
- Tempo di seek per raggiungere il cilindro: 0,01* (59-37)=0,22 msec
- Tempo necessario per raggiungere il settore nel cilindro: 20 msec
- Tempo necessario per estrarre i blocchi della sequenza dal primo run: 1 * 0,025= 0,025 msec
- Tempo totale per estrarre i blocchi dalla prima sequenza: 20,245 msec

Tempo totale impiegato: 40,99 msec.

write(luna, &buffer, 9000)

- Numero di byte scritti nel 1' run: 0

- Numero di byte scritti nel 2' run: 9000
- Indice logico del primo blocco scritto nel 2' run : 19
- Blocchi scritti nel 2' run: 60.001-60.010
- Numero di cilindri su cui è distribuita la sequenza nel secondo run: 1
- Tempo di seek per raggiungere il cilindro: 0 msec
- Tempo necessario per raggiungere il settore nel cilindro: 20 msec
- Tempo necessario per scrivere i blocchi della sequenza nel secondo run: 10 * 0,025= 0,25 msec
- Tempo totale per scrivere i blocchi nel 2' run: 20,25 msec

Tempo totale impiegato: 20,25 msec.