

Esercizi di Informatica Grafica

1. - Rappresentazione dei numeri naturali

Esercizio 1.1

Rappresentare in base 2 i seguenti numeri:

1) $(3241)_{b_5}$

2) $(3241)_{b_9}$

3) $(3241)_{b_8}$

Soluzione

Per i primi due casi, e' necessario passare dalla base 10. Tenendo presente il significato della rappresentazione posizionale dei numeri, otteniamo:

$$(3241)_{b_5} = 3 \cdot 5^3 + 2 \cdot 5^2 + 4 \cdot 5^1 + 1 \cdot 5^0 = 3 \cdot 125 + 2 \cdot 25 + 4 \cdot 5 + 1 = 446$$

$$(3241)_{b_9} = 3 \cdot 9^3 + 2 \cdot 9^2 + 4 \cdot 9^1 + 1 \cdot 9^0 = 3 \cdot 729 + 2 \cdot 81 + 4 \cdot 9 + 1 = 2386$$

Dai numeri espressi in base 10, possiamo ricavare la rappresentazione in base 2 applicando l'algoritmo mod&div, come descritto in Figura 1.

Div	Mod
446	0
223	1
111	1
55	1
27	1
13	1
6	0
3	1
1	1
0	

Div	Mod
2386	0
1193	1
596	0
298	0
149	1
74	0
37	1
18	0
9	1
4	0
2	0
1	1
0	

Figura 1 – algoritmo mod&div

Pertanto, abbiamo:

$$(3241)_{b_5} \equiv (110111110)_{b_2}$$

$$(3241)_{b_8} \equiv (100101010010)_{b_2}$$

3) Per quanto riguarda il terzo numero, si puo' applicare sempre lo stesso procedimento usato per i due precedenti. In questo caso, pero', la base di partenza e la base di arrivo sono correlate da una proprieta' particolare: la prima e' *potenza* della seconda, $8 = 2^3$. Ogni volta che si presenta questo caso, conviene fare la conversione *direttamente*, senza passare dalla base 10. La conversione puo' essere fatta osservando che *ogni cifra* in base 8 (cioe' $0..7 = 0..2^3 - 1$) puo' essere rappresentata con *tre cifre in base 2*. La conversione tra una cifra della base 8 ed un gruppo di tre cifre della base 2 puo' essere fatta semplicemente in maniera empirica, considerando che ciascuna cifra puo' essere scritta come somma delle prime tre potenze di due. Il risultato e' quello riportato in Figura 2.

Cifra	In base 2
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

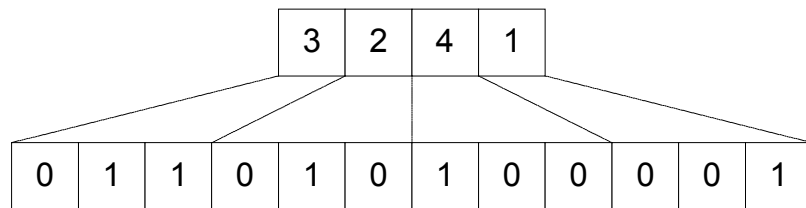


Figura 2 – conversione dalla base 8 alla base 2

Esercizio 1.2

Siano dati i due numeri naturali 13541 e 21256.

- 1) Calcolare il numero di bit necessari per rappresentare in base 2 i due numeri.
- 2) Calcolare la rappresentazione in base 2 dei due numeri.
- 3) Calcolare la rappresentazione in base 2 della loro somma.

Soluzione

- 1) Il numero di bit necessario alla rappresentazione di un numero naturale N e' il piu' piccolo $h: 2^h - 1 \geq N$. Pertanto, $2^h \geq N + 1$, cioe' $h \geq \log_2(N + 1)$. Visto che h e' il piu' piccolo intero che verifica questa proprieta', posso scrivere $h = \lceil \log_2(N + 1) \rceil$. Nei due casi si ottiene $h = 14$, $h = 15$.
- 2) Per rappresentare i numeri in base 2 utilizzo l'algoritmo mod&div. Il risultato e', nei due casi, 11010011100101 e 101001100001000.

3) Per rappresentare in base 2 la somma dei due numeri sopra scritti, posso procedere come segue: calcolo la loro somma in base 10, e converto il numero in base 2 applicando l'algoritmo mod&div. Alternativamente, posso osservare che l'algoritmo con il quale siamo abituati a calcolare le somme consiste nel sommare le cifre di uguale posto (cioè le potenze di uguale ordine), utilizzando le cifre di posto successivo qualora la somma non sia rappresentabile con una singola cifra (riporto). Quindi, tale algoritmo *e' una proprieta' della notazione posizionale*, indipendente dalla base che si usa per rappresentare i numeri. Posso pertanto eseguire le somme in base 2, applicando lo stesso algoritmo. La somma dei due numeri e' descritta in Figura 3. Si osservi che la rappresentazione della somma richiede un bit in piu' del piu' grande degli operandi (cioè 16 in totale), in quanto c'e' un *riporto*.



Figura 3 – Somma di numeri in base 2

2. - Architettura dei calcolatori

Esercizio 2.1

Si consideri una memoria con 1024 celle di 8 bit ciascuna.

- 1) Calcolare il numero di fili di indirizzo e di dati necessari sul bus.
- 2) Supponiamo che il processore voglia scrivere in memoria il numero 112, alla cella di indirizzo 516. Descrivere il contenuto dei fili di indirizzo e di dati.
- 3) Supponiamo che sia necessario portare nella cella di memoria di indirizzo 1010 il contenuto del registro di interfaccia (supposto anch'esso largo 8 bit) presente nello spazio di I/O all'indirizzo 37. Descrivere le operazioni che il processore deve fare per eseguire il trasferimento.

Soluzione

- 1) Il numero di fili di dati e' ovviamente 8, tanti quanti sono i bit da trasportare. Il numero di fili di indirizzo e' $h : 2^h = 1024$. Quindi, $h = \log_2 1024 = 10$.
- 2) Per rispondere alla domanda e' necessario convertire i numeri 112 e 516 in base 2. Si applica quindi l'algoritmo mod&div, come riportato in Figura 4. Il risultato e' $112 \equiv (1110000)_{b_2}$, $516 \equiv (1000000100)_{b_2}$. Quindi, la configurazione dei fili del bus dati ed indirizzi sara' quella riportata in Figura 5. Si osservi che, visto che il numero 112 e' rappresentabile su 7 bit, mentre il bus dati e' costituito da 8 fili, il bit piu' significativo deve essere impostato a zero.

Div	Mod
112	0
56	0
28	0
14	0
7	1
3	1
1	1
0	

Div	Mod
516	0
258	0
129	1
64	0
32	0
16	0
8	0
4	0
2	0
1	1
0	

Figura 4 – algoritmo mod&div

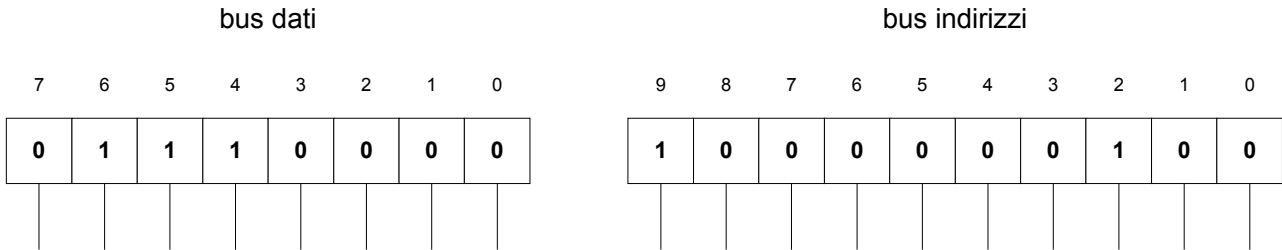


Figura 5 – bus dati ed indirizzi

3) Ogni trasferimento di informazione deve necessariamente passare attraverso il processore, e vedere coinvolto uno dei suoi registri generali. Pertanto, e' necessario *prima* portare il contenuto del registro di interfaccia in un registro generale del processore, e *poi* scrivere in memoria il contenuto di quel registro generale. In termini di istruzioni (scritte, per brevità', in linguaggio mnemonico), si ha:

```
IN R1 37
STORE R1 1010
```

Esercizio 2.2

Si consideri il frammento di programma riportato in Figura 6, e lo si immagini contenuto in una memoria RAM a partire nelle celle di indirizzo 100-106. Si consideri inoltre che le celle di indirizzo 50 e 51 contengono i dati riportati nei due casi a e b.

100	LOAD R1 50
101	LOAD R2 51
102	SUB R2 R1
103	JZ 106
104	STORE R1 52
105	JMP 107
106	STORE R2 52
107

50	120
51	120
52

50	8
51	150
52

Figura 6 – Contenuto della memoria

1) Stabilire cosa viene inserito nella cella di indirizzo 52 al termine dell'esecuzione del frammento di programma sopra scritto nei due casi a) e b).

- 2) Supponendo che in tempo di accesso alla memoria sia di $t = 50ns$, stabilire il tempo di esecuzione del suddetto programma nei due casi a) e b). A tal scopo, si supponga trascurabile il tempo di esecuzione delle istruzioni di elaborazione e controllo.

Soluzione

- 1) Per risolvere l'esercizio conviene tener traccia di quello che viene scritto nei registri del processore.

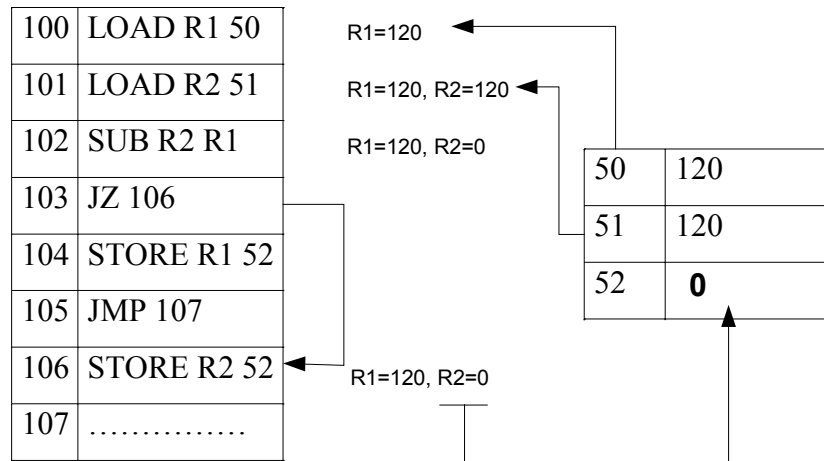


Figura 7 – esecuzione nel caso a)

Nel caso a), le due celle di memoria di indirizzo 50 e 51 contengono lo stesso numero. Tale numero viene caricato nei registri generali del processore R1 ed R2. La successiva sottrazione, pertanto, darà risultato nullo. Ciò rende vera la condizione testata dal salto condizionato JZ, e quindi l'esecuzione del programma passa alla cella di indirizzo 106. A tale cella, l'istruzione richiede di portare nella cella di indirizzo 52 il contenuto del registro R2, che è zero.

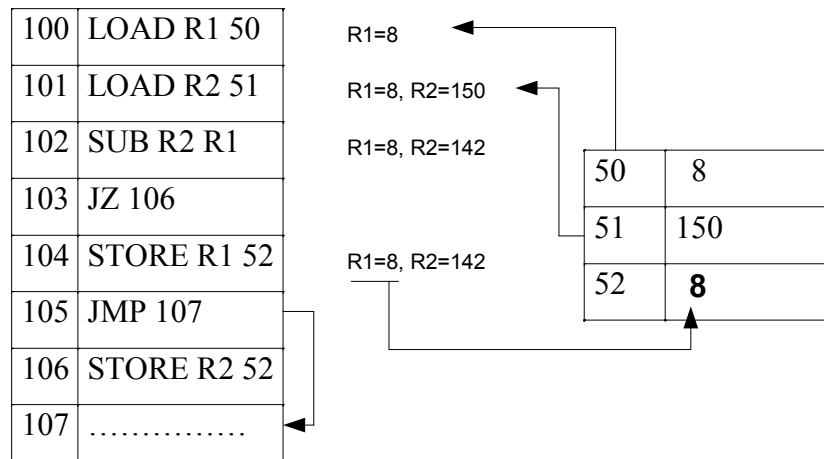


Figura 8 – esecuzione nel caso b)

Nel caso b), le due celle di memoria di indirizzo 50 e 51 contengono numeri diversi.. La sottrazione, pertanto, non darà risultato nullo. Ciò rende falsa la condizione testata dal salto condizionato JZ, e quindi l'esecuzione del programma continua dall'istruzione ad essa successiva. Tale istruzione richiede di portare nella cella di indirizzo 52 il contenuto del registro R1, che contiene il valore letto nella memoria all'indirizzo 50, cioè 8.

- 2) L'esecuzione del programma nei due casi a) e b) differisce semplicemente per la veridicità o meno della condizione testata all'istruzione 103. Nel caso in a), la condizione testata è vera, e quindi eseguo le istruzioni 100-103, 106 (5 istruzioni in totale). Ciascuna istruzione deve essere prelevata con una lettura in memoria. Inoltre, le istruzioni 100, 101, 106, richiedono la lettura/scrittura di dati in memoria nella loro fase di esecuzione. Pertanto, dovrò eseguire in tutto 8 accessi in memoria, per un tempo totale di $T = 8t = 400ns$. Nel caso b), invece, eseguo le istruzioni 100-105 (6 in totale), delle quali la 100, 101, 105 richiedono lettura o scrittura di dati in memoria. In questo caso, quindi, avremo $T = 9t = 450ns$.

Esercizio 2.3

Si supponga di disporre di un floppy disk caratterizzato dai seguenti dati tecnici: $T = 80$ tracce, $S = 18$ settori per traccia, $B = 512$ byte per settore, velocità di rotazione $V = 360$ rpm, tempo di spostamento della testina da una traccia alla successiva $t_s = 1ms/tr$. Sia F un file memorizzato sul settore j -simo di 18 tracce *consecutive*, come indicato nella Figura 9.

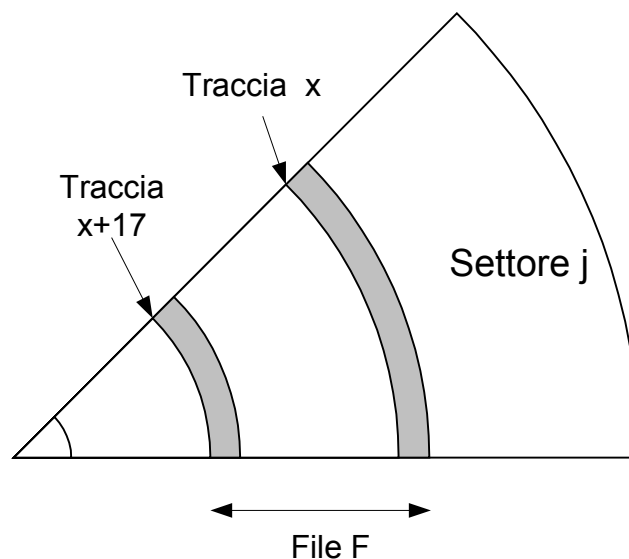


Figura 9 – File memorizzato su floppy disk

- 1) Si calcoli il tempo *medio* di lettura del file F.

- 2) Si supponga adesso che il file venga *deframmentato*, in modo tale da occupare i 18 settori di un'intera traccia. Si calcoli nuovamente il tempo *medio* di lettura del file F.

Soluzione

Il disco fa 360 giri al minuto, cioè un giro ogni 60 di secondo. Quindi, un giro occupa un tempo $t_r \cong 167ms$.

1) Per leggere l'intero file sarà necessario:

- a. Spostare la testina sulla traccia x (tempo di ricerca).
- b. Attendere il passaggio del settore j sotto la testina (latenza).
- c. Leggere un settore (tempo di trasferimento).
- d. Spostare la testina sulla traccia successiva ed attendere nuovamente il passaggio del settore j . Si osservi che il tempo che il lettore di floppy disk ci mette a spostare la testina su una traccia adiacente, $t_s = 1ms/tr$, è tale per cui lo spostamento può avvenire *senza perdere un giro*. Infatti, per riportare il settore j sotto la testina, il disco impiegherà $t = 17/18t_r \gg t_s$. Ciò garantisce che al successivo giro la testina sarà effettivamente in grado di leggere il settore j dalla traccia adiacente.
- e. Ripetere l'operazione d. fino ad aver letto tutte le 18 tracce.

Nel caso medio, il tempo di cui al punto a. sarà $T_r = t_s \cdot T/2 = 40ms$. Il tempo di cui al punto b. sarà $T_l = t_r/2 \cong 83ms$, il tempo di cui al punto c. sarà $T_t = t_r/18 \cong 9.3ms$.

A partire dall'istante in cui la testina ha letto il settore j sulla traccia x , sarà quindi necessario attendere altre 17 rotazioni complete del disco per leggere tutto il file F . Il tempo totale di lettura sarà quindi: $T_{tot} = T_r + T_l + T_t + 17t_r \cong 3s$

- 2) In questo caso, il file si trova tutto sulla stessa traccia. Il tempo medio di lettura del file è quindi calcolato come somma del tempo di ricerca, della latenza, e del tempo di trasferimento di tutta una traccia, quest'ultimo pari al tempo di rotazione del disco. Quindi: $T_{tot} = T_r + T_l + t_r \cong 0.29s$.
Si osservi che la deframmentazione ha ridotto di un fattore 10 il tempo di lettura del file.

Esercizio 2.4

Si consideri un hard disk caratterizzato dai seguenti parametri: tempo medio di ricerca $T_s=5ms$; velocità di rotazione $\omega=10800$ rpm, velocità di trasferimento dati $B = 400$ Mbyte/s.

- 1) Determinare il tempo *medio* di lettura T_L di un file che occupa un quarto di traccia
- 2) Determinare il numero M_t di Mbyte memorizzati su una traccia

- 3) Si consideri un file disposto su due tracce consecutive t e $t+1$, come mostrato in Figura 10. Determinare la distanza d che minimizza il tempo di lettura sapendo che il tempo necessario alla testina per muoversi da una traccia all'altra è circa $T_t = 0.5$ ms. Si esprima la distanza d come frazione di traccia.

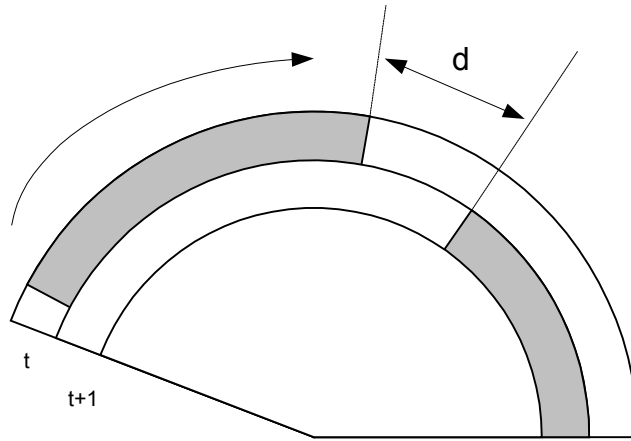


Figura 10 – File memorizzato sull'hard disk

Soluzione

- 1) Il tempo necessario a compiere una rotazione del disco è $T_r = \frac{60}{\omega} = 5.56ms$. Pertanto, il tempo medio di lettura di un file che occupi un quarto di traccia è dato da (si veda anche l'Esercizio 2.3): $T_L = T_s + T_r/2 + T_r/4 = 9.17ms$.
- 2) Durante una rotazione del disco vengono trasferiti $T_r \cdot B$ Mbyte di informazione. Pertanto, su una singola traccia sono memorizzati $T_r \cdot B = 5.56 \cdot 10^{-3} \cdot 400 = 2.2$ Mbyte,
- 3) Nel tempo $T_t = 0.5$ ms durante il quale la testina si sposta dalla traccia t alla traccia $t+1$, il disco compie una frazione di giro pari a $T_t/T_r \cong 0.09$. Pertanto, se la distanza d è maggiore di 0.09 parti di giro, si dovrà attendere del tempo per iniziare a leggere dalla traccia $t+1$ dopo aver spostato la testina. Per contro, se $d < 0.09$, si dovrà attendere il giro successivo per cominciare a leggere dalla traccia $t+1$. Pertanto, la distanza che minimizza il tempo di lettura è $d = 0.09$.

Esercizio 2.5

Si consideri un hard disk caratterizzato dai seguenti parametri: $T = 1024$ tracce, $S = 512$ settori per traccia, $B = 512$ byte per settore, velocità di rotazione $V = 5400$ rpm, tempo di spostamento della testina da una traccia alla successiva $t_s = 0.01ms/tr$.

- 1) Si calcoli il tempo *massimo* di accesso al disco t_{\max}
- 2) Di quanto si dovrebbe aumentare la velocità di rotazione del disco affinché il tempo massimo di risposta diventi $t_{\max}/2$? Di quanto affinché diventi $t_{\max}/4$?
- 3) Si calcoli il massimo tasso al quale è possibile leggere dei dati sul disco.

Soluzione

1) Il tempo massimo di accesso al disco è dato da:

a. Tempo di ricerca massimo: $(T-1)t_s = 10.23ms$

b. Latenza massima pari ad un giro: $t_l = 60/5400 = 11.1ms$

c. Tempo di trasferimento, pari a t_l/S , trascurabile rispetto ai precedenti.

Il tempo massimo di trasferimento è quindi pari a 21.33 ms.

2) Supponendo di aumentare di k volte la velocità di rotazione, la latenza diminuisce di k volte.

L'equazione da impostare nel primo caso è la seguente:

$$(T-1)t_s + t_l/k = \frac{1}{2}[(T-1)t_s + t_l]$$

che risolta da:

$$k = \frac{2t_l}{t_l - (T-1)t_s} \cong 25.5$$

Per dimezzare il tempo massimo di trasferimento è necessario aumentare di oltre 25 volte la velocità di rotazione del disco.

Si osservi che non è possibile diminuire oltre un certo limite il tempo massimo di accesso agendo semplicemente sulla velocità di rotazione del disco, in quanto quest'ultima interviene soltanto sulla latenza (e sul tempo di trasferimento, che comunque in questo caso è trascurabile) e *non* sul tempo di ricerca. Come si evince dal risultato del punto 1), il tempo di ricerca costituisce circa metà del tempo massimo di accesso. Pertanto, sarà impossibile ridurre quest'ultimo ad un quarto del suo valore semplicemente aumentando la velocità di rotazione del disco. Infatti, l'equazione da impostare nel secondo caso sarebbe:

$$(T-1)t_s + t_l/k = \frac{1}{4}[(T-1)t_s + t_l]$$

che porta alla seguente espressione:

$$k = \frac{4t_l}{t_l - 3(T-1)t_s} < 0$$

- 3) Il tasso *massimo* al quale e' possibile leggere informazioni dal disco si calcola imponendo che i tempi *variabili* di accesso al disco (tempo di ricerca e latenza) abbiano il valore minimo, che e' zero. In questo caso, si riesce a trasferire una quantita' di informazione pari a quella contenuta in 512 settori ($512 \cdot 512 = 2^{18}$ byte) nel tempo di rotazione del disco $t_r = 60/5400 = 11.1ms$. La velocita' e' quindi:

$$V = \frac{2^{18}}{11.1 \cdot 10^{-3}} = \frac{10^3}{11.1 \cdot 4} 2^{20} \cong 22.5 \text{ Mbyte/s}^1$$

Esercizio 2.6

Una scheda video supporta una modalita' grafica 1024x768 pixel, con C=65536 colori. Determinare:

- 1) il numero N di bit di indirizzo della memoria video
- 2) la dimensione W di una cella della memoria video
- 3) la dimensione Z (espressa in Mbyte) della memoria video.
- 4) Di quanto e' necessario aumentare dimensione della memoria video affinche' si possa raddoppiare il numero di colori visualizzabili. Si esprima tale numero come frazione della dimensione della memoria.

Soluzione

- 1) Il numero totale di celle di memoria video deve essere sufficiente a contenere tutti i pixel rappresentati nella modalita' grafica descritta. Pertanto:
 - N. di pixel totali: $R = 1024 \cdot 768 = 2^{10} \cdot 2^{10} \cdot 3/4 = 3 \cdot 2^{18}$
 - Il numero di celle video deve essere una potenza di 2 che da' un numero maggiore o uguale a R . Quindi, servono $V = 2^{20}$ celle video.
 - Per indirizzare $V = 2^{20}$ celle video servono $N = 20$ fili di indirizzo.
- 2) La dimensione in bit di una cella video deve essere tale per cui l'insieme delle possibili diverse combinazioni di bit sia maggiore o uguale del numero di colori richiesto. Quindi devo trovare un numero di bit $W : 2^W \geq 65536$. Passando ai logaritmi, $W \geq \log_2 65536$. Visto che W e' un intero, posso scrivere: $W = \lceil \log_2 65536 \rceil = 16$. Ogni cella deve essere di 2 byte.
- 3) La dimensione della memoria video e' data dal numero di celle per la dimensione di ciascuna cella. Quindi ho: $Z = 2 \text{ byte} \cdot 2^{20} \text{ celle} = 2 \text{ Mbyte}$
- 4) Riprendendo quanto scritto al punto 2, e' immediato osservare che la relazione da impostare e':

¹ Si osservi che un Mbyte sono 2^{20} byte.

$2^W \geq 2 \cdot 65536 \Rightarrow 2^{W-1} \geq 65536$. Quindi, per ogni cella di memoria video servono $W = 16 + 1 = 17$ bit. L'aumento della dimensione della memoria video richiesto, pertanto, è di $1/16$.

Esercizio 2.7

Sia dato uno schermo CRT a 19" capace di una modalità grafica XGA (1024x768), con aspect ratio 4:3.

- 1) Calcolare il dot-pitch e la risoluzione dello schermo.
- 2) Calcolare il numero di celle di memoria video sufficienti a memorizzare un'intera schermata
- 3) Sapendo che l'occhio umano distingue circa 10 milioni di colori, stabilire la grandezza di una cella di memoria video necessaria a rappresentare tutti i colori visibili.
- 4) Calcolare la dimensione della memoria video.
- 5) Supponendo che il *refresh rate* dello schermo sia di 100Hz, determinare il massimo tempo di accesso che la memoria video deve avere per supportare la modalità grafica sopra descritta.

Soluzione

Come mostrato in Figura 11, la dimensione *netta* dello schermo è inferiore di un pollice alla dimensione riportata nei dati di produzione. Pertanto, la diagonale dello schermo è lunga $D = 18"$.

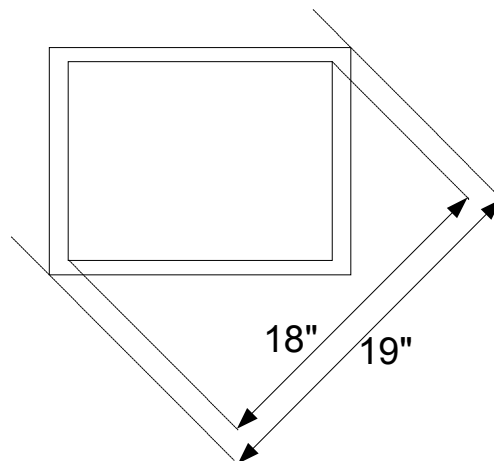


Figura 11 – dimensioni dello schermo

Dato il rapporto di forma 4:3, si calcolano le dimensioni dei due lati tenendo presente che la diagonale ed i due lati formano una terna pitagorica. Pertanto:

$$\begin{cases} x = 4/5 \cdot D = 14.4" \\ y = 3/5 \cdot D = 10.8" \end{cases}$$

- 1) Il dot-pitch dello schermo è dato dal rapporto tra la dimensione di uno qualunque dei due lati, quest'ultima espressa in millimetri, ed il numero di pixel sul lato. A tal proposito, giova ricordare che un pollice è pari a 25.4 mm.

$$\text{Pertanto, il dot-pitch è } d = \frac{14.4 \cdot 25.4}{1024} \cong \frac{14.4}{40} \cong 0.36 \text{ mm.}$$

La risoluzione si calcola come il rapporto tra il numero di pixel e la lunghezza del lato, convenendo di esprimere il lato in pollici (dpi, *dots per inch*). Pertanto:

$$r = \frac{1024}{14.4} \cong 71 \text{ dpi.}$$

- 2) Il numero totale di celle di memoria video deve essere sufficiente a contenere tutti i pixel rappresentati nella modalità grafica descritta. Pertanto:

- N. di pixel totali: $R = 1024 \cdot 768 = 2^{10} \cdot 2^{10} \cdot 3/4 = 3 \cdot 2^{18}$

- Il numero di celle video deve essere una potenza di 2 che dà un numero maggiore o uguale a R . Quindi, servono $V = 2^{20}$ celle di memoria video.

- 3) Seguendo lo stesso procedimento dell'Esercizio 2.6, punto 2, si ottiene:

$$W = \lceil \log_2 10000000 \rceil = 24$$

- 4) Seguendo lo stesso procedimento dell'Esercizio 2.6, punto 3, si ottiene:

$$Z = 3 \text{ byte} \cdot 2^{20} \text{ celle} = 3 \text{ Mbyte}$$

- 5) Per poter sostenere un *refresh rate* di 100 Hz, è necessario leggere tutte le celle di memoria video che contengono una schermata per 100 volte al secondo. Il numero di celle coinvolte è, come abbiamo visto, pari ad R . Quindi, una cella di memoria video deve poter essere acceduta in un tempo $t = 1/(100R) \cong 13.3 \text{ ns}$.

Esercizio 2.8

Dato uno schermo CRT 17" di tipo multisync, con aspect ratio 4:3, determinarne il dot pitch e la risoluzione quando la modalità grafica attivata è:

- 1) 640x480
- 2) 1024x768
- 3) Si supponga di impostare sul monitor una modalità di visualizzazione 1280x1024. Quale sarà la forma dei pixel?

Soluzione

La diagonale “netta” dello schermo (cfr. la soluzione dell’Esercizio 2.7) è di 16”. Pertanto le dimensioni lineari dello schermo sono:

$$\begin{cases} x = 4/5 \cdot 16" = 12.8" \\ y = 3/5 \cdot 16" = 9.6" \end{cases}$$

1) In questo caso, il dot-pitch è $d = \frac{12.8 \cdot 25.4}{640} = 2 \cdot \frac{25.4}{100} \cong 0.51$ mm. La risoluzione è

$$R = \frac{640}{12.8} = 50 \text{ dpi.}$$

2) In questo caso, il dot-pitch è $d = \frac{12.8 \cdot 25.4}{1024} \cong \frac{12.8}{40} \cong 0.32$ mm. La risoluzione è $R = \frac{1024}{12.8} = 80$ dpi.

3) Nei casi di cui ai punti 1) e 2) il rapporto tra il numero di pixel sui due assi è 4:3, pari a quello dello schermo, e quindi i pixel risulteranno quadrati. In questi casi, dot-pitch e risoluzione sono identici su entrambi gli assi, e quindi possono essere calcolati considerandone uno o l’altro indifferentemente. Se invece la modalità grafica utilizzata è 1280x1024, il rapporto tra il numero di pixel sui due assi è 5:4, come si può facilmente osservare riducendo ai minimi termini il rapporto dei due numeri sopra scritti. Tale rapporto è *differente* dal rapporto delle dimensioni lineari (*aspect ratio*) del monitor (4:3). Cio’ significa che i pixel non potranno che essere *rettangolari*, con il lato orizzontale più lungo. In questo caso il dot-pitch (distanza tra i centri di due pixel) o, analogamente, la risoluzione, sono *differenti* sui due assi; in particolare il dot-pitch e’ maggiore (la risoluzione e’ minore) sull’asse orizzontale che sull’asse verticale.

3. - Sistemi operativi

Esercizio 3.1

Si consideri un sistema operativo multitasking, nel quale il tempo di cambiamento di contesto è fisso e pari a $t_{cc} = 1$ ms. Si definisca “*time slice*” il tempo massimo per il quale un programma può eseguire senza bloccarsi prima che il sistema operativo passi il controllo ad un altro programma.

- 1) Si stabilisca la durata della *time slice* in modo tale che l’efficienza del sistema, cioè la percentuale di tempo che quest’ultimo passa ad eseguire compiti significativi impartiti dall’utente, sia del 95%.
- 2) Si supponga che il processore sia in grado di eseguire 1000 istruzioni durante una *time slice*. Si calcoli il tempo al quale termina (in assenza di blocchi) un programma P che deve eseguire 10000 istruzioni, supponendo che all’istante $t = 0$ il programma P cominci la sua esecuzione, nell’ipotesi che altri 5 programmi siano “contemporaneamente” in esecuzione. Si consideri che il sistema operativo consente ai vari programmi di andare in esecuzione “a turno”, in modo equo, per una *time slice* ciascuno.
- 3) Si confronti il caso 2) con quello in cui sono contemporaneamente in esecuzione 2 altri programmi.

Soluzione

- 1) Se il sistema ha una *time slice* di durata t_s , la sua efficienza è $E = t_s / (t_s + t_{cc})$. Dovrà pertanto essere $t_s = \frac{E \cdot t_{cc}}{1 - E} = 19$ ms.
- 2) In questo caso, il programma P dovrà andare in esecuzione per 10 *time slice*. Per 9 volte, però, sarà interrotto da altri programmi, per una durata totale di 5 *time slice* più 5 tempi di cambio di contesto. Il programma P completerà quindi la sua esecuzione all’istante $T = 9 \cdot 6 \cdot (t_s + t_{cc}) + t_s = 1.1$ s.
- 3) In questo caso, avremo $T = 9 \cdot 3 \cdot (t_s + t_{cc}) + t_s = 559$ ms

Esercizio 3.2

Si consideri un interprete comandi a riga di comando. Sia “*dir*”, seguito dal percorso della directory il cui contenuto si vuole visualizzare, il comando necessario alla visualizzazione del contenuto di una directory (ad esempio: “*dir pippo*” mostra il contenuto della directory “pippo”). Supponiamo

che il disco identificato dalla lettera “c” abbia un file system descrivibile tramite il diagramma ad albero riportato in Figura 12, e supponiamo che l’interprete di comandi stia visualizzando il prompt riportato in Figura 13.

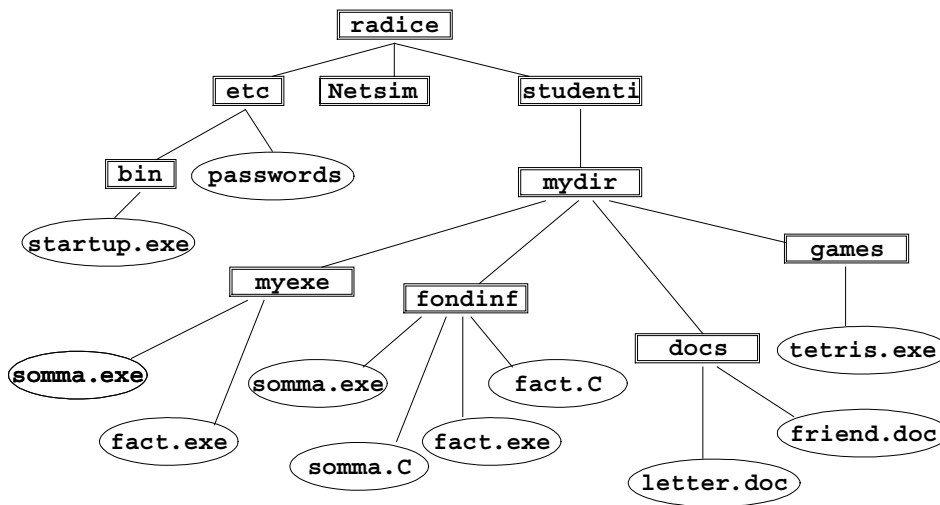


Figura 12 – Rappresentazione ad albero del file system

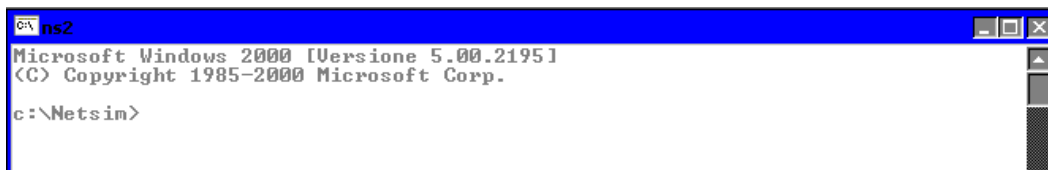


Figura 13 – prompt dell’interprete

Scrivere il comando che consente di visualizzare il contenuto della directory “docs”, riferendo quest’ultima

- a. In modo assoluto.
- b. In modo relativo, rispetto alla directory corrente.

Soluzione

Se voglio riferire una directory in modo assoluto, devo indicare il percorso a partire dal nodo radice dell’albero. Il percorso assoluto è quindi “\studenti\mydir\docs” (si noti che il percorso è assoluto, e quindi inizia con una barra invertita). Quindi, per il caso a), il comando è: “*dir \studenti\mydir\docs*”.

Nel caso b), è necessario calcolare il percorso relativamente alla directory corrente, che è quella specificata nel messaggio di prompt (“Netsim”). Per arrivare alla directory richiesta, è quindi necessario risalire di un livello l’albero del file system, per poi ridiscenderlo di nuovo. Il percorso relativo è quindi “..\studenti\mydir\docs” (si noti che il percorso è relativo, e quindi *non* inizia con una barra invertita). Quindi, per il caso b), il comando è: “*dir ..\studenti\mydir\docs*”.

4. - Rappresentazione delle immagini

Esercizio 4.1

Sia data una stampante laser B/N in grado di stampare 10 pagine A4 (21x29.7cm) al minuto.

- 1) Supponendo che il tamburo della stampante abbia un diametro $D = 10$ cm, si calcoli la velocità di rotazione del medesimo (in radianti al secondo) necessaria a garantire tali prestazioni. Si supponga, per semplicità, che la stampante possa essere alimentata di carta con continuità.
- 2) Si supponga che la stampante debba stampare su un'area di 8"x11" un'immagine in bianco e nero con una risoluzione di 100 dpi. Si supponga che la trasmissione dei dati dal calcolatore alla stampante avvenga ad una velocità $C = 500$ Kbit/s. Si calcoli il tempo di invio dell'immagine alla stampante.
- 3) Supponendo che il processo di stampa della pagina j e di ricezione della pagina $j+1$ possano avvenire in parallelo, si calcoli quale deve essere la velocità minima di trasmissione dei dati dal calcolatore alla stampante perché quest'ultima possa stampare pagine del tipo descritto al punto 2 alla massima velocità di cui è capace.
- 4) Supponiamo adesso che la stampante sia a colori. Si calcolino:
 - a. La velocità di rotazione del tamburo necessaria a garantire la stampa 10 pagine al minuto.
 - b. La velocità di trasmissione dati necessaria a garantire la stampa di 10 pagine al minuto, sotto le ipotesi di cui al punto 2).

Soluzione

- 1) Il tamburo della stampante deve poter ruotare per una lunghezza di $L = 10 \cdot 29.7 = 297$ cm in un minuto. Deve poter quindi descrivere $\frac{L}{\pi D}$ circonferenze, cioè $\frac{L}{\pi D} \cdot 2\pi \cong 59.4$ radianti in un minuto. La velocità di rotazione sarà quindi pari a circa un radiante al secondo.
- 2) L'immagine è composta da un numero di punti pari a $P = 8 \cdot 100 \cdot 11 \cdot 100 = 880000$. Visto che l'immagine è in bianco e nero, ciascun pixel richiede soltanto un bit. In tal caso, il tempo di trasmissione sarà: $t_s = \frac{880000}{500000} = 1.76$ s.
- 3) Se la stampante è in grado di stampare 10 pagine al minuto, può stampare una pagina in 6 secondi. Quindi, per poter stampare a ciclo continuo deve poter ricevere 880000 bit in un tempo

pari a 6 secondi. La velocità minima del collegamento deve quindi essere:

$$V = \frac{880000}{6} \cong 147 \text{ Kbit/s.}$$

- 4) In questo caso, un foglio deve essere stampato 4 volte (una per ciascuno dei 4 colori fondamentali per la stampa in quadricromia: Cyan, Magenta, Yellow, black). Pertanto, la velocità di rotazione del tamburo deve essere 4 volte superiore a quella calcolata al punto 1, cioè circa 4 radianti al secondo. Per poter stampare un'immagine in quadricromia, è necessario mandare alla stampante 4 immagini successive, una per ciascuno dei 4 colori fondamentali. In ciascuna di queste immagini, un pixel è associato ad un singolo bit (ad esempio: 1=inchostro presente, 0=inchostro non presente). Pertanto, per quanto riguarda la trasmissione dati, la velocità deve essere 4 volte superiore, cioè $V = 4 \cdot \frac{880000}{6} \cong 587 \text{ Kbit/s}$

Esercizio 4.2

Sia data un'immagine raster 200x400 pixel, formata da bande verticali di colori diversi, ciascuna larga 5 pixel.

- 1) Si calcoli la dimensione del file che memorizza l'immagine sotto le seguenti ipotesi:
 - a. Il file è memorizzato in formato bitmap non compresso, con un numero di colori sufficiente alla sua corretta rappresentazione (si trascurino eventuali *header* del formato bitmap non considerati durante le lezioni).
 - b. Il file è memorizzato in un formato compresso che usa la tecnica *run length encoding*: dati j pixel consecutivi del medesimo colore, viene memorizzata la coppia $\langle j, c \rangle$, con c numero del colore. Il numero j è memorizzato su 2 byte, mentre c è memorizzato su 1 byte.
- 2) Si supponga adesso che l'immagine venga ruotata di 90° . Si risponda nuovamente alla domanda 1) nei 2 casi proposti.

Soluzione

- 1) Se ogni banda verticale è larga 5 pixel, nel file sono presenti contemporaneamente $C = 200/5 = 40$ colori. Pertanto, il file può essere salvato come bitmap a 256 colori senza che ci sia perdita di informazione. In tal caso, il numero di punti associato all'immagine sarà: $P = 200 \cdot 400 = 80000$. Va tenuta in conto anche la palette, che avrà dimensione pari a $3 \cdot 256 = 768$ byte. La dimensione totale del file associato all'immagine sarà quindi non meno di 80768 byte. Nel caso del formato compresso, invece, ogni gruppo di 5 pixel consecutivi può es-

sere rappresentato con 3 soli byte. Pertanto la dimensione del file sul disco sarà $D = 80000 \cdot 3/5 + 768 = 48768$ byte.

- 2) Nel caso in cui l'immagine venga ruotata, le bande da verticali diventano orizzontali. Il numero totale di pixel non cambia: pertanto, la dimensione del file nel caso di cui al punto a) rimane la stessa. Nel caso di cui al punto b), invece, il numero di pixel consecutivi dello stesso colore passa da 5 a $400 \times 5 = 2000$. Quindi, la dimensione del file diventa $D = 80000 \cdot 3/2000 + 768 = 888$ byte.