

# TECNOLOGIE INFORMATICHE

*(appunti delle lezioni svolte in classe)*

## Table of Contents

Che cos'è l'informatica?.....	1
Cosa sono le tecnologie informatiche?.....	2
Inquadramento storico.....	2
Hardware.....	3
La CPU ed il suo funzionamento.....	4
RAM e unità di misura delle memorie di un elaboratore.....	5
Codifica delle Informazioni.....	8
La codifica dei numeri .....	8
Codifica dei dati alfabetici (codice ASCII).....	8
Dall' analogico al digitale: suoni e immagini.....	9
Analogico e digitale.....	10
I suoni.....	11
Le immagini.....	12
Software .....	12
Introduzione.....	13
Evoluzione del business del software.....	14
Software liberi.....	15

## INTRODUZIONE

### Che cos'è l'informatica?

La parola "informatica" deriva dal francese e più precisamente dalla contrazione delle due parole "information" e "automatique": informazione automatica.

E' una scienza interdisciplinare che tratta tutti gli aspetti della elaborazione delle informazioni con procedure automatiche.

Con le due semplici frasi sopra riportate abbiamo già introdotto una serie di concetti che dobbiamo approfondire.

Innanzitutto cosa significa "interdisciplinare". Significa che riguarda più discipline. Ad esempio riguarda la disciplina che si occupa della progettazione degli elaboratori, intesi come le macchine che elaborano le informazioni, ovvero l' elettronica oppure la disciplina che si occupa di creare i codici per poterlo fare, di trovare le soluzioni algoritmiche per elaborare i dati e in questo caso parliamo di programmazione, di realizzazione dei software.

Secondo concetto: cosa intendiamo per informazione. Per informazione intendiamo un concetto molto generale:

- è informazione il dato matematico
- è informazione il suono
- è informazione l'immagine
- è informazioni il sapere, la notizia su qualcuno o qualche cosa

Tutti questi elementi sono informazioni e vengono elaborate in modo automatico da delle macchine chiamate elaboratori.

## **Cosa sono le tecnologie informatiche?**

Sono il prodotto dell'applicazione creativa e originale del nostro bagaglio di conoscenze scientifiche e non solo. una tecnologia informatica un' insieme di idee unite alla progettazione, alla realizzazione e all'uso di macchine per elaborare informazioni, con il fine di risolvere uno o più problemi reali che riguardano la gestione delle informazioni da parte dell'uomo. Ad esempio la tecnologia con cui si produce uno specifico lettore mp3 riguarda le idee, la progettazione, la realizzazione specifica per tale tipo di lettore.

## **Inquadramento storico**

Se generalizziamo la definizione che abbiamo appena dato di informatica potremmo far risalire questa scienza ai tempi antichi perché già l' abaco e così anche i successivi calcolatori elaboravano automaticamente dei dati, eseguivano operazioni aritmetiche. I risultati di queste operazioni però venivano scritti per ricordarseli, non venivano memorizzati dal calcolatore e si poteva fare solo un' operazione alla volta.

L' informatica come noi siamo abituati a concepirla, chiamiamola “informatica moderna”, ha bisogno di due elementi molto importanti: la memoria e la programmabilità. Quando si è cominciato a capire come poter memorizzare dati e programmare delle operazioni, allora è cominciata l'evoluzione dell'informatica col significato che gli attribuiamo oggi.

L' utilizzo della logica binaria è alla base della memoria e della programmabilità. Prima della formalizzazione di questa logica, non si poteva pensare di realizzare le macchine dotate di memoria e programmabilità come le intendiamo ora.

Ma cosa intendiamo per logica binaria? Siamo nel 1703 quando Leibniz sviluppa in modo formale e matematico il sistema numerico binario, dove esistono solo 1 e 0. Basandosi su questo formalismo verso la metà del XIX secolo Boole sviluppa l'algebra che prende il suo nome “algebra di Boole” o “algebra Booleana” basata sulla logica dove tutto è vero o falso, on o off, 1 o 0. Parallelamente Babbage sfruttando tale logica elabora il primo prototipo di macchina programmabile. Tutta l' informatica è costruita usando la logica binaria.

E' con Babbage o meglio con Ada Byron, protettrice del lavoro di Babbage, che si comincia a pensare ad utilizzare queste macchine programmabili non solo per fare i conti, ma anche in

altri settori come il lavoro. Nasce in questo periodo il concetto di automazione del lavoro. E' con Ada Byron che si parla per la prima volta di applicare l'informatica al lavoro.

E' però solo verso la metà del secolo successivo, il XX, che si sviluppano nuovi e più potenti computer e si crea il primo computer interamente digitale, basato sulla trasmissione di segnali discreti, dove le parti meccaniche sono ridottissime. In questo periodo comincia a diffondersi l'utilizzo di queste macchine non come semplici calcolatori. Ricordiamo che siamo nel periodo della seconda guerra mondiale e quindi parliamo anche di ricerca e sviluppo di queste macchine a scopo bellico ad esempio nel campo delle telecomunicazioni.

E' solo dopo tutti questi eventi, molto più recentemente, che è nata la disciplina che si occupa della comunicazione delle informazioni. Ad un certo punto ci si è chiesti come poter comunicare da una macchina ad un'altra macchina le informazioni. Nasce così INTERNET, la rete delle reti, che permette la comunicazione tra i vari elaboratori, la trasmissione delle informazioni e quindi la comunicazione fra utenti. Internet comincia a svilupparsi intorno agli anni '50(1950), ma comincia a diffondersi in modo preponderante solo negli anni '70 (1970).

## L'ELABORATORE o COMPUTER

Computer significa calcolatore per cui dovrebbe essere indicata con tale nome solo una macchina che elabora dati matematici, che esegue operazioni aritmetiche. In realtà viene chiamata computer la macchina in grado di elaborare automaticamente qualsiasi genere di informazioni. Noi chiameremo questa macchina indifferentemente computer o elaboratore. Tra le macchine che elaborano automaticamente informazioni possiamo considerare anche tablet, smartphone etc...

Il computer è costituito da due elementi fondamentali: l'hardware ed il software.

Per HARDWARE intendiamo l'insieme delle parti solide che formano la macchina: tutti i componenti meccanici, fisici, elettronici.

Per SOFTWARE intendiamo i dati, ma soprattutto i procedimenti di calcolo ed elaborazione che vengono applicati ai dati per poter consegnare all'utente dei risultati.

Il funzionamento di un computer viene ancora ben descritto dal diagramma di Von Neumann (figura 1) che risale addirittura al 1946:

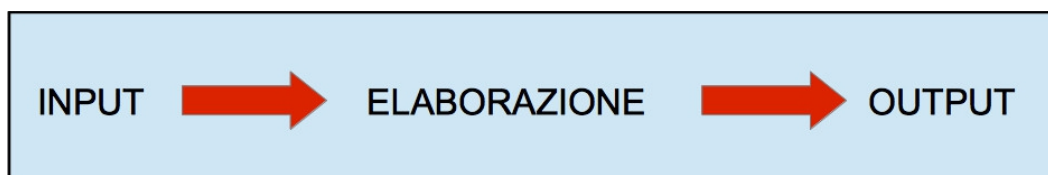
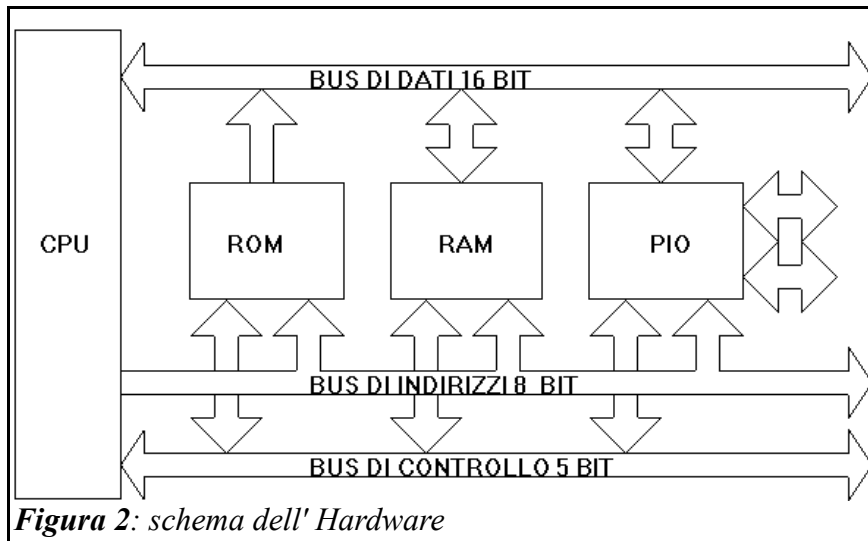


Figura 1: Diagramma di Von Neumann

### Hardware

Parliamo ora degli elementi base che caratterizzano la macchina e innanzitutto descriviamo l'hardware di un elaboratore con uno schema (figura 2):



In genere tutti questi elementi, eccetto gli I/O devices, sono contenuti in una scatola detta **case**. A questi elementi bisogna aggiungere anche l' **hard disk** (HD), quello principale che sta all' interno del case, ma si possono anche acquistare degli ulteriori hard disk che vengono esternamente collegati al computer. Il computer é quindi costituito da diversi componenti elettronici che vengono assemblati all' interno del case e possono essere anche sostituiti singolarmente.

### La CPU ed il suo funzionamento

#### CPU

CPU = Central Process Unit

E' l' unica parte attiva del computer. Essa comanda, elabora e fa le operazioni, prende e mette i dati dalla RAM, dalla ROM, comunica con gli INPUT ed OUTPUT devices.

La caratteristica principale da prendere in considerazione per quanto riguarda la CPU è la frequenza ovvero la velocità. E' dell' ordine del GHz, ad oggi i computer più veloci elaborano le informazioni con velocità superiori ai 4GHz.<sup>1</sup>

La CPU è formata da diverse parti:

- ALU: Unità Aritmetica Logica (Arithmetic Logic Unit)
- CU: Unità di Controllo (Control Unit)
- Registri: memorie della CPU dove vengono memorizzate dati durante lo svolgimento dell'istruzione

<sup>1</sup> Parlando di velocità di un computer si parla di frequenza non dell' unità di misura propria della velocità perchè abbiamo a che fare col il sistema binario, con apparecchi digitali e tutto si basa su impulsi. Quanto frequentemente riescono ad essere inviate le informazioni in byte? Questa è la domanda che ci si pone per valutare la velocità con cui lavora un computer

- Cache: memoria della CPU (in un personal computer è una delle memorie più piccole, ma al giorno d' oggi può raggiungere anche il MB (MegaByte) e soprattutto si tratta di una memoria piuttosto molto veloce)

La ALU esegue le operazioni in base alle istruzioni ricevute: fa i conti. E' la vera componente di calcolo dell' elaboratore. Queste istruzioni arrivano dalla CU che riceve i dati e le istruzioni su ciò che la macchina deve fare dalla memoria RAM, li decodifica e decide quali operazioni far svolgere alla ALU in base a ciò che è scritto nel programma, in base all'algoritmo, in base a ciò che dice il codice. Ad esempio se il codice dice: " se  $A < B$  fai questa operazione senno' no, allora è la CU che interpreta il codice, fa il confronto e decide se mandare o no l' operazione da fare alla ALU. In generale trasforma ciascuna istruzione ricevuta in flussi di dati fra i registri al fine di eseguire tutte le istruzioni.

Per svolgere tutte queste operazioni in velocità è necessario che ci siano delle piccole e veloci memorie dove mettere i dati su cui la ALU deve lavorare, dove immagazzinare i dati che serviranno per fare delle operazioni. Tra l'altro certi dati e certe istruzioni servono più e più volte, alcuni risultati di un' operazione servono quasi subito per fare altre operazioni e sarebbe una gran perdita di tempo passare sempre ogni cosa alla RAM e subito dopo riprenderle. Queste memorie della CPU sono i registri e la cache. Si tratta di memorie piccole dove i dati vengono sovrascritti quando sono piene. Per prima cosa entrano in gioco i registri che vengono riempiti dai dati che la CU ha decodificato e su cui la ALU deve fare i conti, poi interviene la cache quando oltre che a dati si devono memorizzare anche delle istruzioni del programma, ad esempio istruzioni che devono essere svolte più volte in modo che non debbano essere sempre richiamate dalla RAM e poi decodificate dalla CU. Una delle differenze fra registri e cache è proprio questa: nei registri non ci sono istruzioni, ma solo dati. Vedremo che i dati e le istruzioni di un programma devono essere sempre decodificate affinché le informazioni possano essere elaborate perché l'elaboratore è in grado di capire solo un linguaggio, il linguaggio macchina, che non è quello in cui normalmente il programmatore scrive.

La CPU è generalmente contenuto in un chip che è delle dimensioni di un qualche cm<sup>2</sup> installato nella scheda madre (motherboard) all'interno del case dove è alloggiata anche la RAM e le schede di espansione come ad esempio la scheda video. Questo è generalmente vero per i computer desktop e laptop.

### **RAM e unità di misura delle memorie di un elaboratore**

**RAM**

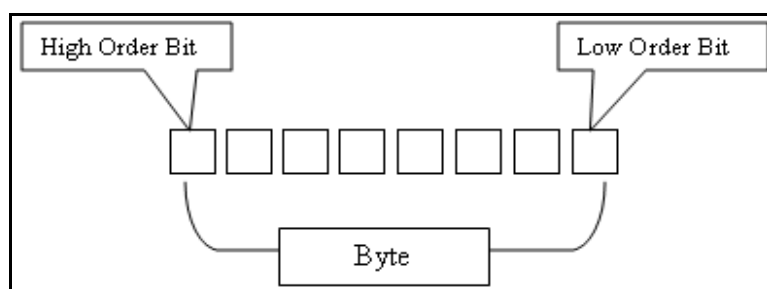
RAM = Random Access Memory

E' un'altra memoria presente nel computer. E' quella parte del computer dove vengono immagazzinati i dati durante l'elaborazione delle informazioni, ma non vengono salvati in modo

permanente. Si tratta di una memoria volatile, i dati in essa contenuti si perdono nel momento in cui si spegne il computer. Ad esempio quando aprite un programma, ad esempio word per scrivere un testo, esso viene caricato nella RAM insieme ai dati su cui lavorate (il documento) e vi rimane finché lo chiudete o finché viene spento il computer (buona prassi sarebbe chiudere tutti i programmi aperti e poi spegnere il computer). Per questo motivo quando avete troppi programmi aperti il computer alle volte rallenta o addirittura si blocca, perché la RAM non ce la fa a contenere tutto e non può arbitrariamente cancellare delle parti che non sono ancora state salvate. Alle volte, invece di bloccarsi, il computer ti dice out of memory: sono piena.

Ora che abbiamo capito a cosa serve la RAM vediamo come è fatta. Quello che leggete di seguito è vero per ogni tipo di memoria presente nella macchina ed introduce un argomento fondamentale dell'informatica, quello che ci dice come ragiona tale scienza, ci dice in che modo vengono immagazzinate tutte le informazioni anche durante l'elaborazione, il modo in cui vengono tradotte in linguaggio macchina tutte le informazioni e le istruzioni che vogliamo elaborare. Cominciamo quindi a conoscere la lingua che parla l'informatica, una lingua che dobbiamo imparare.

La RAM, come tutte le memorie, è formata da tante celle da 1 byte (B) che è l'unità di misura della memoria. Immaginatoci il byte come una sequenza di 8 cellette ognuna delle quali è il bit che sarà quindi un sottomultiplo del byte. L'informatica conosce solo lo 0 e l'1 vale a dire che ogni celletta può essere riempita solo da uno 0 o da un 1. Ne consegue che l'informatica conosce solo il sistema binario.



**Figura 3:** 1 byte (B) è formato da 8 bit rappresentati dalle cellette

Per questo l'elettronica digitale, che è la parte dell'informatica che si occupa della progettazione e dello sviluppo degli elaboratori, riguarda i circuiti che conoscono solo due stati: alto o basso, on o off.

In una celletta della RAM da 1B trovo un'informazione costituita da 8 bit cioè costituita da una riga di 8 spazi dove posso mettere solo 0 o 1 (Figura 3). Il numero massimo di combinazioni possibili di 0 e 1 negli 8 spazi è quindi  $2^8=256$ , quindi in una celletta da un byte si

può trovare una di queste 256 combinazioni. Ad ognuna di queste combinazioni viene associata un' informazione, sia essa un numero, una lettera etc. Vedremo poi con quali regole questo avviene.

Le informazioni con cui abbiamo a che fare però sono tantissime quindi le RAM devono essere molto grandi, devono essere dotate di tantissime celle. Poichè parliamo di milioni e miliardi di informazioni, è necessario definire i multipli del Byte:

$1024 \text{ byte} = 2^{10} \text{ byte} = 1 \text{ kilobyte (kB)}$
$(1024)^2 \text{ byte} = 2^{20} \text{ byte} = 1048576 \text{ byte} (\sim 10^6 \text{ byte}) = 1 \text{ megabyte (MB)}$
$(1024)^3 \text{ byte} = 2^{30} \text{ byte} = 1073741824 \text{ byte} (\sim 10^9 \text{ byte}) = 1 \text{ Gigabyte (GB)}$
$(1024)^4 \text{ byte} = 2^{40} \text{ byte} = 1099511627776 \text{ byte} (\sim 10^{12} \text{ byte}) = 1 \text{ Terabyte (TB)}$

Le definizioni della tabella qui sopra sono non standard, nel senso che sono quelle usate dagli informatici, ma non sono quelle accettate dal Sistema Internazionale (SI) che definisce il kilobyte ed i suoi multipli come per tutte le grandezze che conosciamo:

$1000 \text{ byte} = 10^3 \text{ byte} = 1 \text{ kilobyte (kB)}$
$1000000 \text{ byte} = 10^6 \text{ byte} = 1 \text{ megabyte (MB)}$
$1000000000 \text{ byte} = 10^9 \text{ byte} = 1 \text{ Gigabyte (GB)}$
$1000000000000 \text{ byte} = 10^{12} \text{ byte} = 1 \text{ Terabyte (TB)}$

In pratica l' uso delle une o delle altre unità dipende dal contesto: se ci stiamo riferendo a potenzi di 2 si usano in genere le definizioni non standard, mentre in altri contesti si usano quelle standard. La cosa da ricordare é di porsi sempre questa domanda, di fare attenzione a questo quando parlate di memorie. In genere quando sono i programmatori a parlare essi si riferiscono alle definizioni non standard, ma in ambito commerciale, quando sono le aziende a parlare di memorie in commercio la definizione usata é quella prevista dal SI.

Quali sono a grandi linee le dimensioni delle RAM nei personal computer oggi? Ormai abbiamo a che fare con memorie di svariati GB. In linea di massima in un personal computer possiamo avere RAM da 2 a 16/32 GB. Ad ogni modo 2 GB di RAM al giorno d'oggi sono un

po' pochetti per un computer, ma ovviamente dipende da quello che l' utente deve fare con lo strumento. Negli smartphone più moderni oggi la RAM va dagli 1 ai 3 GB

## Codifica delle Informazioni

Cominciamo a vedere ora come si trasformano i dati e le informazioni in modo che la macchina sia in grado di capire ed elaborare. Poichè la macchina capisce solo 0 e 1 è necessario scrivere tutte le informazioni in tale formato. Partiamo dai numeri e cominciamo quindi a parlare di sistema binario e di conversione dei numeri da sistema decimale a binario e viceversa, poi vedremo anche come viene digitalizzato un testo, un immagine, un suono.

### La codifica dei numeri

Innanzitutto è necessario saper trasformare un numero espresso in base decimale nel suo corrispondente nel sistema binario e viceversa.

Il metodo per farlo è ben spiegato nel libro a pag 16-17.

### Codifica dei dati alfabetici (codice ASCII)

Dopo aver visto l' hardware, abbiamo parlato di memorie e abbiamo detto che ci sono delle cellette in cui ci stanno solo 0 e 1, perché ciò è l' unica cosa che la macchina è in grado di capire. Quindi ogni dato che voglio elaborare dovrà essere trasformato in una serie di 0 e 1. Per questo abbiamo fatto i cambi di base decimale-binario e viceversa.

Ora vediamo come sono trattati i dati alfabetici. Innanzitutto cosa intendiamo per dato alfabetico. Si tratta delle 26 lettere dell'alfabeto che possono essere sia minuscole che maiuscole, le dieci cifre numeriche (0,1,2,3,4,5,6,7,8,9), le parentesi tonde, gli operatori (+,-,...) e alcuni altri caratteri particolari. In pratica stiamo parlando dei tasti della tastiera. Cosa succede quando pigiate un tasto della tastiera? Per poterlo elaborare la prima cosa che deve fare la CPU è convertirlo in qualche modo in una sequenza di 0 e 1. Significa quindi che ad ognuno di quei caratteri viene associata una combinazione di zeri e uno. Quanto è lunga la combinazione di zero e uno che rappresenta un carattere?

Questi caratteri non sono tantissimi e potrebbero bastare solo 7 bit per rappresentarli tutti poiché con tale numero di bit posso fare  $2^7=128$  combinazioni diverse mettendo nei sette bit o 0 o 1 e questo basterebbe perché i caratteri da rappresentare sono in numero inferiore. Esiste però una codice molto diffuso, che viene usato da quasi tutti i produttori di hardware e software che si chiama codice **ASCII (American Standard Code for Information Interchange)** (vedi Figura 4) che utilizza 7 bit per ogni carattere e quindi può rappresentare fino ad un massimo di  $2^7=128$  caratteri. Questo è il codice che si usa per associare ad ogni carattere un'informazione fatta di 0 e 1. In realtà questo codice è stato esteso e per rappresentare un carattere si usa 1 byte e quindi le 128 combinazioni in più sono state usate per rappresentare altri simboli .



Che cos' è un codice? Si dice spesso dire “parole in codice”, “codice segreto”, ma cosa significa. Sono dei simboli, anche parole, che vengono usati per rappresentare qualche cosa. Se il codice è segreto solo alcune persone sapranno cosa corrisponde ai vari simboli, ma non sempre il codice è segreto, ad esempio nella lingua italiana se diciamo gallina sappiamo tutti cosa intendiamo e lo sappiamo perché tale codice lo sanno tutti coloro che conoscono l'italiano, si tratta comunque di un codice. Abbiamo una codifica cioè abbiamo associato all'animale che fa le uova che ci mangiamo la parola, il simbolo gallina.

La stessa cosa vale per il codice ASCII dove ad ognuno dei caratteri che vi ho detto sopra viene associata una sequenza specifica di 0 e 1 lunga 8bit. Questa è una codifica e tutti la accettano così, come tutti quando diciamo gallina associamo un determinato animale. Nella Figura 4 è rappresentato tutto il codice ASCII

Ora è nato un nuovo standard che si chiama **UNICODE** che tenta di associare una combinazioni di 0 e 1 ad ogni carattere di tutti gli alfabeti. Ovviamente non bastano più sequenze di 8bit, perché i caratteri da rappresentare sono molti di più di 256. Questo nuovo standard prevede stringhe di 16 bit. Con stringhe di 16bit, il numero massimo di combinazioni che posso fare sono  $2^{16}=65536$  che sono sufficienti. La necessità di creare un nuovo standard è dovuta alla nascita di internet che mette in contatto gente di tutto il mondo. La sequenza dei caratteri usata dallo standard ASCII non è cambiata per i caratteri che traduceva tale standard per cui unicode capisce ASCII perché lo contiene, ma non viceversa.

Byte	Cod	Char	Byte	Cod	Char	Byte	Cod	Char	Byte	Cod	Char
00000000	0	Null	00100000	32	Spc	01000000	64	@	01100000	96	`
00000001	1	Start of heading	00100001	33	!	01000001	65	A	01100001	97	a
00000010	2	Start of text	00100010	34	"	01000010	66	B	01100010	98	b
00000011	3	End of text	00100011	35	#	01000011	67	C	01100011	99	c
00000100	4	End of transmit	00100100	36	\$	01000100	68	D	01100100	100	d
00000101	5	Enquiry	00100101	37	%	01000101	69	E	01100101	101	e
00000110	6	Acknowledge	00100110	38	&	01000110	70	F	01100110	102	f
00000111	7	Audible bell	00100111	39	'	01000111	71	G	01100111	103	g
00001000	8	Backspace	00101000	40	(	01001000	72	H	01101000	104	h
00001001	9	Horizontal tab	00101001	41	)	01001001	73	I	01101001	105	i
00001010	10	Line feed	00101010	42	*	01001010	74	J	01101010	106	j
00001011	11	Vertical tab	00101011	43	+	01001011	75	K	01101011	107	k
00001100	12	Form Feed	00101100	44	,	01001100	76	L	01101100	108	l
00001101	13	Carriage return	00101101	45	-	01001101	77	M	01101101	109	m
00001110	14	Shift out	00101110	46	.	01001110	78	N	01101110	110	n
00001111	15	Shift in	00101111	47	/	01001111	79	O	01101111	111	o
00010000	16	Data link escape	00110000	48	0	01010000	80	P	01110000	112	p
00010001	17	Device control 1	00110001	49	1	01010001	81	Q	01110001	113	q
00010010	18	Device control 2	00110010	50	2	01010010	82	R	01110010	114	r
00010011	19	Device control 3	00110011	51	3	01010011	83	S	01110011	115	s
00010100	20	Device control 4	00110100	52	4	01010100	84	T	01110100	116	t
00010101	21	Neg. acknowledge	00110101	53	5	01010101	85	U	01110101	117	u
00010110	22	Synchronous idle	00110110	54	6	01010110	86	V	01110110	118	v
00010111	23	End trans. block	00110111	55	7	01010111	87	W	01110111	119	w
00011000	24	Cancel	00111000	56	8	01011000	88	X	01111000	120	x
00011001	25	End of medium	00111001	57	9	01011001	89	Y	01111001	121	y
00011010	26	Substitution	00111010	58	:	01011010	90	Z	01111010	122	z
00011011	27	Escape	00111011	59	;	01011011	91	[	01111011	123	{
00011100	28	File separator	00111100	60	<	01011100	92	\	01111100	124	
00011101	29	Group separator	00111101	61	=	01011101	93	]	01111101	125	}
00011110	30	Record Separator	00111110	62	>	01011110	94	^	01111110	126	~
00011111	31	Unit separator	00111111	63	?	01011111	95	_	01111111	127	Del

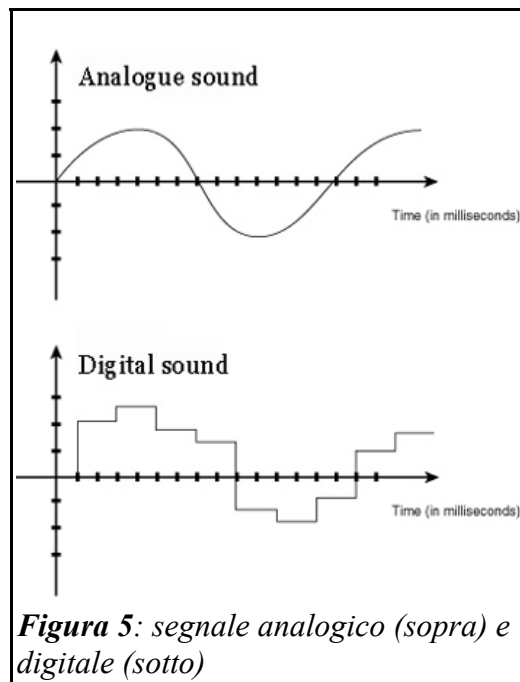
Figura 4: codice ASCII

## Dall' analogico al digitale: suoni e immagini

### Analogico e digitale

Abbiamo già detto che gli elaboratori o computer elaborano informazioni in modo automatico e abbiamo detto che il concetto di informazione é da intendersi come concetto esteso non limitato alla produzione di testi o all'elaborazione di calcoli matematici, ma anche all'elaborazione di suoni e immagini.

I segnali che caratterizzano un suono o un immagine possono essere analogici o digitali. Al giorno d'oggi molto più spesso si ascolta musica in formato digitale e le macchine fotografiche non digitali sono una rarità. I segnali analogici sono continui nel tempo. Con il termine di segnale analogico si indica la rappresentazione di una grandezza fisica con un' altra grandezza analoga che ben la descrive. Volendo vedere il concetto di analogico graficamente guardiamo la figura 5: il grafico sopra di figura 5 rappresenta un segnale continuo. Il segnale (in questo caso un segnale sonoro) viene registrato ad ogni istante.



Il segnale della penna é continuo, disegno una linea continua.

Il computer però non elabora il segnale partendo dalla sua rappresentazione analogica, non é capace di fare ciò, lo deve trasformare in un segnale digitale, vale a dire discreto, cioè in un segnale che viene raccolto con una certa frequenza, non in modo continuo. Come agisce quindi un computer? Si dice che il segnale viene “campionato” con una certa frequenza: lo registro ad un certo istante e lo mantengo fisso finche non lo “campiono” un'altra volta dopo un certo tempo  $\Delta t$  fissato (vedi figura 5 in basso). Si parla di campionamento di un segnale. E' in questo modo che un segnale continuo viene digitalizzato. Più piccolo é  $\Delta t$  più il segnale digitale assomiglia ad un segnale continuo, analogico.

Se il segnale non é già discreto, prima di essere elaborato da un computer deve essere digitalizzato. Per questo quando il segnale analogico arriva al vostro dispositivo deve essere innanzitutto convertito in un segnale digitale per mezzo di oggetti che prendono il nome di A/D converter (convertitore analogico digitali), poi passano al microprocessore dove vengono elaborati ed infine per mezzo dei D/A converter (convertitori digitali analogici) vengono ritrasformati nuovamente in segnali analogici. Il vantaggio del segnale digitale é che non si altera mai, può essere riprodotto miliardi di volte e rimanere sempre lo stesso, la qualità non si degrada.

## I suoni

Vediamo ora il caso del suono, cerchiamo di capire come viene elaborata la musica che ascoltate. Come si fa a digitalizzare un suono?

Per prima cosa dobbiamo trasformare il suono in un segnale elettrico tramite un microfono

(trasduzione). Questa operazione comune agli apparecchi digitali e ai vecchi apparecchi analogici trasforma l'intensità della pressione dell'aria nel tempo (il suono) in un equivalente intensità del segnale elettrico.

I dispositivi elettronici sono in grado di digitalizzare un suono ossia trasformarlo da un segnale elettrico continuo a una sequenza di numeri.

Questa operazione viene fatta misurando il segnale ad intervalli regolari (campionamento) e trasformando questa misura in numero ed infine il numero in una sequenza di bit attraverso l'aritmetica binaria (discretizzazione).

Il dato digitale può rappresentare con maggiore o minore fedeltà l'originale analogico. Questo dipende principalmente da due fattori:

- la frequenza con cui le misure vengono effettuate (ossia il numero di misure in un intervallo di tempo)
- la precisione con cui si associa il numero alla misura. Questo dipende da quanti numeri ho a disposizione: potrebbe essere che a due segnali molto simili debba associare lo stesso numero perché ho finito i numeri che ho a disposizione.

L'orecchio umano può percepire frequenze da 20Hz a 20.000Hz (in realtà la maggior parte dei suoni importanti (parlato, musica, suoni comuni) è nel range fra 200 e i 4000 Hz)

Si può dimostrare matematicamente che se si vuole campionare un segnale bisogna utilizzare una frequenza almeno doppia della massima frequenza presente nel segnale altrimenti le frequenze più alte vengono eliminate. Inoltre si è visto che per migliorare la qualità è conveniente aumentare la precisione delle misure e non il numero dei campionamenti oltre a quello minimo richiesto. Per tali motivi lo standard dei CD-ROM ha scelto di campionare il suono a 44.100 Hz (maggiore del doppio di 20.000 Hz, la massima frequenza udibile)

Per quanto riguarda la precisione l'utilizzo di un byte (256 possibili livelli) per ciascuna misura sarebbe sufficiente per una qualità bassa (conversazione telefonica) ma non per l'ascolto di musica di qualità e per questo si è scelto nei CD di utilizzare 2 byte (216=65536 valori).

A questo punto si capisce perché nella custodia di un CD viene in genere riportato 80min di registrazione, 700Mbyte di memoria. Questi valori non sono casuali e indipendenti l'uno dall'altro:  $44100$  (campionamenti al secondo)  $\times$  2 (2byte occupati da ogni campione)  $\times$  2 (canali, dx e sx)  $\times$  60 (secondi)  $\times$  80 (minuti) =  $176000 \times 60 \times 80 = 768\ 000\ 000 = 846\text{Mbyte} \sim 700\text{Mbyte}$  perché parte della memoria viene persa per l'organizzazione del file system del CD.

## Le immagini

Anche il dato immagine deve essere trasformato in una sequenza di 0 e 1 come tutti gli altri dati. Ma cosa significa? Quando fate una foto col vostro telefono non ci sono pellicole che

vengono impressionate, ma una memoria su cui vengono memorizzate le immagini. La memoria é una di quelle di cui abbiamo parlato fino ad ora, fatta di cellette che posso riempire solo con 0 e 1, dove ogni celletta é fatta dal solito byte.

L' immagine viene suddivisa in tanti quadratini: i pixel. Ad ogni pixel viene associato un dato fatto da una sequenza di 0 e 1 che dice di che colore é associato quel quadratino.

Ci sono due considerazioni da fare. Associamo ad ogni pixel un colore, quindi se io divido la mia foto in pochi quadratini, non avrò molta risoluzione e vedrò l' immagine a quadratini, più quadratini metto e più avrò la possibilità di distinguere un punto dall'altro . Si parla in genere di Mega pixel per ogni foto, ad esempio 10 Mpixel. Cosa significa? Significa che ho diviso la foto in 10 000 000 quadratini. Con un tale numero di quadratini posso già dire che riesco a distinguere tanti punti molto vicini della foto, se poi ad ogni punto ci posso mettere un colore diverso se necessario, avrò una bella foto.

La seconda considerazione infatti riguarda quanti colori ho a disposizione. Se io ho la foto divisa in milioni di quadretti, ma ho solo 10 colori a disposizione, comunque non avrò una buona fotografia, visto che dovrò mettere lo stesso colore su svariati quadratini.

Teniamo inoltre sempre presente che ad ogni colore dovrà corrispondere una combinazione di 0 e 1. Se volessimo memorizzare un colore con una combinazione di 8bit, cioè se ad ogni colore venisse associato un byte avremmo a disposizione solo 256 colori. Sono un po' pochi per rappresentare tutte le sfumature di colore. Per i colori si usano 3byte. Il meccanismo funziona nel seguente modo: nella macchina fotografica ci sono tre sensori, uno per il rosso, uno per il verde e uno per il blu (scala RGB). Ad ogni pixel verrà quindi associato un colore che occupa una memoria di 3byte. In conclusione quindi per un immagine di 10Mpixel avrò occupato una memoria di  $3\text{byte} \times 10\text{Mpixel} = 30\text{Mbyte}$

## Software

### Introduzione

Un elaboratore non può elaborare alcun dato se non è stato precedentemente programmato. Questo significa che devono essere installati al suo interno, devono essere messi nella sua memoria e inizializzati, i programmi che gli permettono di elaborare i dati e produrre informazioni.

Che cosa intendiamo per programmi (software)?

Si tratta di istruzioni per elaborare i dati. Queste istruzioni implementano gli algoritmi che vengono usati per risolvere problemi.

In che “lingua” scrive il programmatore?

Ci sono diversi livelli di programmazione. Esiste il linguaggio macchina che è caratterizzato dall'alfabeto binario fatto di 0 e 1 che la CPU è in grado di comprendere. Il linguaggio macchina é l' unico che la CPU é in grado di capire. Il programmatore, colui che scrive i

programmi, può scrivere direttamente usando questo linguaggio, ma nella maggior parte dei casi non è così, perché sarebbe troppo complicato. Generalmente un programmatore scrive usando un linguaggio che non è subito comprensibile alla macchina, ma deve essere compilato con dei programmi detti compilatori oppure interpretato. I compilatori sono a loro volta dei programmi installati nelle macchine. Immaginiamoci il compilatore come un traduttore automatico.

I linguaggi generalmente usati dai programmatori sono detti di alto livello e sono costituiti da una serie di parole chiave e regole sintattiche che sono più comprensibili al programmatore, ma non lo sono per la CPU, per questo, una volta scritto, il programma deve essere compilato affinché la CPU lo capisca. La serie di parole chiave e le regole sintattiche danno origine ad una serie di istruzioni che vengono applicate dalla CPU per l'elaborazione dei dati.

Alcuni di questi programmi di alto livello sono Pascal, Fortran, C++, Java. Essi sono più intuitivi per il programmatore rispetto al linguaggio macchina e sono maggiormente indipendenti dalla macchina che si sta utilizzando.

Esiste una classificazione dei software. Si possono distinguere due sottoinsiemi:

- SOFTWARE di BASE
- SOFTWARE APPLICATIVO

Per **software di base** intendiamo quei programmi in grado di far funzionare la macchina semplificando il lavoro dell'uomo e ottimizzando l'uso delle risorse del sistema. Tra i software di base ricordiamo : il sistema operativo, il bios, il firmware.

Per **software applicativo** intendiamo tutti quei programmi che vengono installati dall'utente a seconda delle proprie necessità. Ad esempio quasi tutti installiamo un word processor (programma di video scrittura) per poter scrivere documenti, oppure i disegnatori installano programmi per fare disegni, o chi fa paghe installa programmi per fare le paghe o chi fa contabilità programmi di contabilità. Questi sono tutti programmi applicativi.

### **Evoluzione del business del software**

Come si è evoluto il business dei software?

Come tutte le cose anche questo settore si è evoluto nel tempo. Alla fine degli anni '70 (1970) nelle aziende si cominciano a richiedere **programmi ad hoc**, realizzati esclusivamente per un'azienda per soddisfare le proprie esigenze. Questi programmi venivano realizzati all'interno dell'azienda stessa o da società esterne (software house) che mandano i loro programmatori direttamente sul posto. Il ruolo di queste società è quello di fornire consulenti che aiutano a realizzare il codice dall'interno andando loro stessi a programmare presso l'

azienda e usando i terminali dell'azienda stessa.

I pacchetti realizzati per delle specifiche aziende, una volta elaborati dalle società, divennero pian piano dei codici più generali da adattare anche ad altre realtà. Fu così che nacquero i primi **programmi standard** che a volte vengono venduti alle aziende raggruppati in pacchetti. Da società di consulenza le software house divennero vere e proprie aziende produttrici di software.

Oggi esistono le aziende che producono programmi ad hoc, realizzati esclusivamente per specifiche aziende per fare specifiche cose, ma anche aziende che producono programmi standard, venduti così come sono, a pacchetti preconfezionati. Una famosa azienda di questo tipo è la Microsoft che produce software molto importanti come i sistemi operativi per i PC (windowsVista, windows8 etc) oppure il diffusissimo pacchetto office contenente i programmi Word, Power Point, Excel. I programmi standard sono generalmente pacchetti grossi perché devono cercare di coprire le esigenze di molti, generalmente ti ritrovi sempre con più cose di quelle di cui necessiti. Alle volte però questi pacchetti si possono acquistare a moduli. Gli errori o banchi di questi software vengono corretti all'interno dell'azienda stessa e gli aggiornamenti si devono acquistare.

Infine ci sono le aziende che producono i software **chiavi in mano**, vale a dire che producono dei software personalizzati per le varie aziende, ma partendo da dei programmi già esistenti che loro stessi hanno fatto oppure altri hanno fatto. Le aziende che producono questo tipo di software possono essere le stesse aziende che producono il software di base, ma anche le aziende distributrici del software di base oppure altre aziende che riescono ad integrare il pacchetto di base con degli applicativi ad hoc.

Se i programmatori o le aziende che hanno creato dei software richiedono i diritti d'autore allora si parla di software proprietario. Questo tipo di software può essere sia a pagamento che gratuito. Ad esempio proprietari sono tutti i software della Microsoft. Un esempio di formato proprietario, ma gratuito è il programma Adobe per leggere i file pdf. Attenzione abbiamo detto leggere non editare i file pdf. La stessa azienda infatti mentre lascia gratuito il programma per leggere i file pdf, fa pagare il programma per editare i file pdf (editare significa poter modificare i files).

### **Software liberi**

Innanzitutto dobbiamo ricordare che un software è caratterizzato dalla versione eseguibile che è il programma che l'utente usa quando deve fare qualcosa ed il file sorgente che è quello che il programmatore ha scritto. Il file sorgente è accessibile al programmatore che lo può modificare e cambiare per creare una nuova versione del file eseguibile, ma non è quasi mai accessibile all'utente, a voi che usate il programma.

Questa distinzione è importante quando parliamo di software gratuito. Si tratta di quei

programmi che non si pagano, che possono essere installati e usati da tutti nelle proprie macchine.

Ce ne sono di due tipi. Un tipo di programma gratuito è quello distribuito con la formula dello **shareware**. Si tratta di quei programmi eseguibili che sono liberi per un periodo limitato di tempo e poi li si devono acquistare. Dopo un certo numero di giorni può succedere che la versione installata non funzioni proprio più o che sia disponibile con funzioni limitate. Sentito mai parlare di DEMO? Le demo non sono altro che programmi liberi distribuiti con la formula dello shareware. Il vantaggio per le aziende di distribuire i programmi in questo modo è che : innanzitutto il programma viene testato dagli utenti e alla fine viene messa in commercio una versione più corretta, secondo viene data la possibilità ad un possibile compratore di provare il prodotto in modo che si convinca a comperarlo o meno. Il secondo tipo di programmi gratuiti é quelli distribuito con la formula **freeware**. Questi programmi sono e rimangono gratuiti o liberi. Il guadagno delle aziende che producono programmi di questo tipo può essere dato dalla pubblicità. Immaginatoci il software Meteo.it, programma che si scarica e utilizza liberamente. Si tratta di un programma libero, ma l' azienda che lo ha realizzato ci guadagna con la pubblicità inserita nel programma stesso.

Ma cos' è che viene distribuito a titolo gratuito? Fino ad ora abbiamo fatto riferimento ai file eseguibili dei programmi, non alla sorgente. Un particolare caso di programmi distribuiti con la formula freeware è quello in cui é libera non solo la versione eseguibile, ma anche la sorgente. Questi programmi prendono il nome di software **open source**. La sorgente può essere disponibile a vari livelli: la si può solo leggere o anche modificare, modificare e creare nuovi software anche a pagamento, modificare e creare nuovi software che però non si possono vendere, modificare e rendere pubblico il prodotto .

Facciamo qualche esempio:

LINUX e OPEN OFFICE: sono due programmi open source, le cui sorgenti si possono modificare e si può da essi creare nuovo software, che però non può essere rivenduto, deve essere messo a disposizione della comunità.

FREEBSD: è un software open source, può essere modificato, ma il nuovo software realizzato può anche essere venduto. Un esempio di software nato da questo programma e rivenduto è il sistema operativo dei computer Apple: OsX.

Come possono sopravvivere le aziende che producono software open source se non vendono i loro prodotti?

Alle volte il guadagno sta solo nel fatto che il proprio software messo in rete, utilizzato e modificato dalla comunità migliora ed anche il produttore iniziale ne ha un guadagno perché alla fine il suo software funziona meglio. Altri software open source nascono da progetti scientifici pubblici e come tali devono essere pubblicati e resi disponibili a tutti. Poi ci sono dei software finanziati dai produttori di hardware. Ad esempio la fondazione che produce linux vive



anche dei finanziamenti che riceve dai produttori di hardware dove gira tale sistema operativo. I finanziatori della fondazione Apache che produce il programma open office (programma di video scrittura) sarà finanziata dai concorrenti di Microsoft che vende il pacchetto office. Infine, come già accennato, alcuni programmi vivono delle pubblicità che riescono ad inserire nei loro prodotti, non vendono il software, ma gli spazi pubblicitari.