

Informatica di Base

Appunti introduttivi ai sistemi informativi

Author: Andrea Manni

Copyright: GFDL

Version: 1.5

Indice degli argomenti

Informatica

 Evoluzione nella societa'

Analogico e Digitale

Sistema Binario

Unita' di misura

 Multipli del Bit (TABELLE DI CONVERSIONE):

 Trasmissione dati su reti

 Velocita' di clock

Supporti di storriggio (dati)

 Hard Disks

 Memorie Flash

 Propieta' dei supporti di storriggio

 HARD DISK

Classi di elaboratori

 Calcolatori tascabili

 Palmari e smart phones

 Ultraportabili (netbook)

 Portatili

 Trasportabili

 Ultraportatili

 Personal Computer

 Work Station

 Server

 Mainframe

 Cluster

 Supercomputer

Linguaggi di programmazione

 Compilatori

 Interpreti

 Java e similari

Licenze software

Tipi di licenze

Publico dominio

Freeware

Shareware

Licenze per software liberi

Copyleft != Copyright

Licenza BSD

Licenza Apache

Eula

Generato con: <http://docutils.sourceforge.net/rst.html>

Informatica



Informatica:

L'informatica è una scienza interdisciplinare che riguarda tutti gli aspetti del trattamento dell'informazione mediante procedure automatizzate.

L'oggetto della materia sono quindi i **dati** e la **manipolazione** di questi, da un lato la capacità di *storare* i dati (cosa non così elementare quando il quantitativo diventa enorme) per poterli poi *manipolare*. Buona parte del lavoro del sistemista consisterà quindi nel garantire l'*integrità* e la *disponibilità* (intesa come *continuità* del servizio) dei dati. Starà poi all'*amministratore di rete* preoccuparsi di renderli disponibili (si pensi a un sito web che rende delle risorse disponibili) agli *utilizzatori*.

Link sull'argomento: <http://it.wikipedia.org/wiki/Informatica>

Evoluzione nella società

Per meglio percepire l'importanza dell'informatica si rifletta sull'influenza che le seguenti invenzioni nel campo del trattamento delle informazioni hanno avuto sulla società umana:

- **Diffusione della scrittura:**
 - *integrità* dell'informazione: dati come il costo di una merce cessano di essere arbitrari, con conseguenze negli scambi commerciali.
 - maggiore durata nel tempo delle informazioni (seppur limitata da supporti deperibili)
- **Invenzione della stampa: Gutenberg 1448**
 - Maggiore diffusione delle informazioni grazie all'abbattimento dei costi di stampa.
 - Progressivo ampliamento della base di utenza: sempre più persone possono almeno imparare a leggere e quindi emanciparsi da una *casta* ristretta che mantiene il controllo sulla fruizione delle informazioni. Seppure in misura minore aumenta anche la possibilità di produrre e diffondere contenuti.
- **Nascita dell'informatica/elettronica**
 - Superamento di parte dei problemi legati alla deperibilità dei supporti sui quali si conservano le informazioni.
 - Enorme aumento del quantitativo di dati gestibili e abbassamento del costo di gestione. Supporti magnetici e ottici hanno radicalmente cambiato il modo di archiviare le informazioni rispetto ai depositi cartacei.
 - Maggiore rapidità nella trattazione delle informazioni, per es. la possibilità di censire la popolazione rapidamente e gestire un apparato fiscale più efficiente.
 - Possibilità di incrociare i dati, ottenendo sostanzialmente nuovo valore da informazioni preesistenti

- **Diffusione della rete Internet**

- Disponibilita' globale delle informazioni (in formato elettronico) con costi nettamente minori rispetto al passato
- Abbattimento del costo della diffusione delle informazioni

L'importanza della gestione digitale delle informazioni e' tale da accentuare il divario esistente tra chi puo' accedere ai nuovi mezzi e chi ne e' escluso, problema indicato generalmente con il termine Digital Divide.

Analogico e Digitale

Grandezze analogiche:

Sono grandezze **continue**: hanno infiniti *stadi intermedi*. Si pensi al movimento dell'ombra dell'asta di una meridiana in contrapposizione alle posizioni finite di un orologio digitale, o alle infinite posizioni che puo' assumere il potenziometro di un amplificatore (ovviamente un amplificatore *analogico!*). Tensione e corrente elettriche sono grandezze analogiche perche' rappresentabili con funzioni continue.

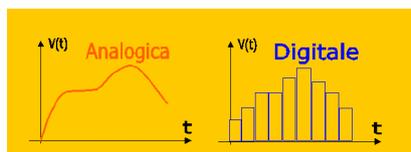
Grandezze digitali:

Sono *discrete*: assumono un numero *finito* di stadi intermedi. Ad esempio l'interruttore per accendere una lampada. Quando i valori sono solo due si parla di valori *binari*, caso che riscontriamo in tutte le tecnologie basate sul silicio come i microprocessori o sui vecchi transistor. Si potrebbe pensare ai componenti di un computer come ad enorme numero di lampadine che possono essere solo accese o spente: l'evoluzione tecnologica passando dai transistor ai microprocessori ha raggiunto un livello di miniaturizzazione dell'ordine dei ~30 nanometri.

La miniaturizzazione presenta i seguenti vantaggi:

- Minor costo di produzione (rapporto tra il numero di transistor e superficie del *wafers* sul quale si stampano i microprocessori)
- Minore distanza tra gli elementi = minore *attrito* prodotto dal passaggio della corrente = minore dispersione termica = minore calore. Questo si traduce in un minore consumo elettrico, ulteriormente accentuato dal corrispettivo risparmio sulle unita' di raffreddamento.

In informatica, con digitale ci si riferisce a tutto cio' che viene rappresentato con numeri o che opera manipolando numeri. Il termine deriva dall'inglese digit, che significa cifra, che a sua volta deriva dal latino digitus, che significa dito.



Una grandezza, tensione o corrente, si dice analogica quando al variare del tempo puo' essere rappresentata mediante una funzione continua, senza punti di discontinuita'.

Una grandezza si dice digitale quando viene rappresentata mediante un sistema binario costituito da valori 0 ed 1.

Links: <http://www.culturadigitale.net/?p=4>

In informatica quindi si pone costantemente il problema della rappresentazione delle grandezze analogiche (tipiche del mondo *reale*) in valori digitali (discreti, e quindi approssimativi) **A/D - D/A**

Sistema Binario

I *computer* lavorano solo con grandezze finite, e si basano su sistemi cosiddetti *binari*, dato che i microprocessori sono transistor miniaturizzati, che a loro volta possono essere visti come insiemi di lampadine che per definizione possono essere solo *accese* o *spente*.

Potremmo considerare quindi i due stati come:

- Spento = 0
- Acceso = 1

Entita' *discrete* come numeri interi o caratteri possono essere espressi facilmente con la sola combinazione dei valori 0 e 1, mentre valori *complessi* riconducibili a numeri reali (pensiamo alle infinite sfumature di un colore o ai suoni) tramite un'approssimazione basata su un numero di bit arbitrario. Ad esempio il carattere *a* in formato *ASCII* si puo' esprimere con: 01100001 , un'immagine puo' essere rappresentata da una griglia di punti (pixel) piu' o meno estesa (risoluzione) con una certa *profondita'* di colore espressa in bit,

E cercare di comporre valori *complessi* (approssimazione di numeri reali tramite un numero di bit arbitrario), con la sola combinazione dei valori 0 e 1, mentre entita' *discrete* il carattere *a* in formato *ASCII* si puo' esprimere con: 01100001 .

Schema sulla trasformazione da binario a decimale:

0	0	0	0	0	0	0	0	decimali = 0
0	0	0	0	0	0	0	1	decimali = 1
0	0	0	0	0	0	1	0	decimali = 2
0	0	0	0	0	0	1	1	decimali = 3
0	0	0	0	0	1	0	0	decimali = 4
0	0	0	0	0	1	0	1	decimali = 5
0	0	0	0	0	1	1	0	decimali = 6
...	
1	1	1	1	1	1	1	1	decimali = 255

Si lavora in base 2 (cioe' si moltiplica sempre per 2), quindi si ha a che fare con sequenze numeriche del tipo:

2 4 8 16 32 64 128 256 512 1024 ...

C'e' una ripetizione di questi *patterns* numerici dovuti alle **parole** usate dai processori. Se l'unita' minima dell'informazione e' il *bit*, l'unita' minima *funzionale* e' il **Byte**: con 256 valori possibili possiamo esprimere efficacemente caratteri, colori ed altre informazioni percepibili dai nostri sensi.

In informatica, *word* (dall'inglese parola) e' un termine che identifica la dimensione nativa dei dati usati da un computer. Una *word* e' semplicemente un gruppo di bit di una determinata dimensione che sono gestiti come unita' da un microprocessore. La dimensione (o lunghezza) della *word* e' un'importante caratteristica dell'architettura di un computer.

Links aggiuntivi:

- <http://www.culturadigitale.net/?p=8>
- <http://doc.studenti.it/appunti/introduzione-allinformatica/2/introduzione-informatica.html>

Unita' di misura

Se diciamo che una singola *lampadina* sia un **bit**, nel nostro lavoro ci troviamo a gestire miliardi di lampadine. Cerchiamo di definire dei multipli del bit per rendere la quotidianita' piu' semplice.

bit

Una singola informazione, l'unita' minima. Si indica con la *b minuscola*. E' l'unita' di misura di riferimento per il *trasferimento* dei dati su reti: una scheda di rete *10/100* sposta circa 12MB (Bytes) al secondo.

Byte:

Un Byte e' formato da 8 bit. Si indica con la *B Maiuscola* . La si puo' considerare l'unita' *funzionale* minima nell'informatica, dato che un numero minore di bit difficilmente puo' rappresentare qualcosa di significativo. E' l'unita' di misura di riferimento per lo storgaggio dati: le dimensioni dei files e dei supporti di storgaggio si esprimono in Bytes.

Multipli del Bit (TABELLE DI CONVERSIONE):

- Bit
- Byte (B): 8 bit
- Kilobyte (KB): 1024 B

- Megabyte (MB): 1024 KB, 1048576 B
- Gigabyte (GB): 1024 MB, 1048576 KB, 1073741824 B
- Terabyte (TB): 1024 GB, 1048576 MB, 1073741824 KB, 1099511627776 B

Link disponibile: <http://it.wikipedia.org/wiki/Byte>

Nota:

Si tenga conto che i produttori di Hard Disk generalmente indicano la capienza di detti supporti in migliaia di megabyte e non 1024*: quindi per formattare 1 hd di 500 MB si considera 500/1000 (e non 500/1024).

Trasmissione dati su reti

Il termine velocità di trasmissione (o di trasferimento) viene solitamente utilizzato a proposito di scambi di informazioni tra computer o comunque dispositivi elettronici. Siccome su questi dispositivi l'informazione viene memorizzata e viaggia in forma digitale, ovvero è sostanzialmente una sequenza di bit, è naturale che tale velocità venga misurata in bit per secondo (e da qui il termine equivalente inglese *bitrate*).

Ad esempio, se in una linea ADSL abbiamo un trasferimento dati di 4 Mb/s, cioè 4.000.000 b/s, avremo (4.000.000/8) B/s equivalenti a 500 kB/s.

Quindi per calcolare la velocità di trasferimento di un file le cui dimensioni sono espresse in BYTES su una rete bisogna **dividere per 8**

Ad es:

- ADSL 20 Mb (*venti mega bit*) ha una velocità equivalente di circa ~1.8MB (*mega Bytes*) per secondo
- Rete ethernet 10/100 Mb ha una velocità di trasferimento di circa ~12MB per secondo.

C'è uno scarto tra la velocità nominale di trasferimento dati su una rete e l'effettivo trasferimento dei dati tra i due punti. Tipicamente questo è dovuto a forme di interferenza del supporto trasmissivo (pensiamo a una rete wifi su onde radio), a imprecisioni dell'hardware di rete, conflitti tra pacchetti, problemi ambientali.

Velocità di clock

Usiamo l'*Hertz* come unità di misura per le frequenze quali quelle di clock della CPU (es 2.0GHz numero di cicli del processore in un secondo) o di un BUS di per il trasferimento dati (es 33Mhz). Un hertz corrisponde alla distanza tra due creste di un'onda.

In teoria, la **velocità o frequenza di clock** è il numero di commutazioni (tra i due livelli logici "0" e "1") che circuiti logici (interni ad un'unità di calcolo o di un microprocessore) sono in grado di eseguire in un minuto secondo (preso come unità di tempo); essa espressa in cicli/secondo (Hertz) ed i suoi multipli.

Unità di misura per le frequenze:

- 1 hekt o hertz (simbolo hHz) = 10² Hz = 100 Hz
- 1 kilohertz (simbolo kHz) = 10³ Hz = 1 000 Hz
- 1 megahertz (simbolo MHz) = 10⁶ Hz = 1 000 000 Hz
- 1 gigahertz (simbolo GHz) = 10⁹ Hz = 1 000 000 000 Hz

Ricordiamo che i prefissi 'kilo', 'mega', 'giga', 'tera' derivano dal sistema numerico decimale (a base 10) e servono per esprimere le quantità in 'ordini di grandezza':

- 1 KILO = mille = 10³
- 1 MEGA = 1 milione = 10⁶
- 1 GIGA = 1 miliardo = 10⁹
- 1 TERA = mille miliardi = 10¹²

La velocità o frequenza di clock è il numero di commutazioni tra i due livelli logici "0" e "1" che circuiti logici interni ad un'unità di calcolo o di un microprocessore sono in grado di eseguire nell'unità di tempo di un minuto secondo, ed è espressa in cicli al secondo, o hertz, e suoi multipli; normalmente per eseguire un'istruzione o una semplice somma sono necessari più cicli di clock.

In queste moderne CPU essendo la velocità di clock così elevata, viene generata da un minuscolo oscillatore al quarzo posto all'interno della stessa CPU, regolabile tramite il BIOS. Gli informatici esperti sono in grado di aumentare le prestazioni dei computer aumentando la velocità di clock nominale del processore, ma si tratta di un'operazione rischiosa che può portare al surriscaldamento della CPU. Vi sono comunque alcuni processori di particolare efficienza e robustezza che sono in grado di supportare bene velocità di clock maggiori di quella nominale, senza subire danni. Nel gergo informatico questa operazione viene definita *_overclocking*, ed è utilizzata anche dall'industria per testare l'affidabilità delle CPU prodotte.

Tipicamente la velocità di clock consigliata per una CPU viene stabilita da un valore medio statistico ottenuto considerando vari esemplari del wafer da cui proviene la CPU, quindi è possibile che la nostra CPU sia *stabile* anche a una velocità di clock **leggermente** superiore a quella indicata. All'opposto, quando un elaboratore non necessita della velocità massima della CPU è possibile fare il *downclock* della stessa, in modo da ridurre il consumo e il riscaldamento, a vantaggio di una maggiore stabilità del sistema e minore usura del sistema di raffreddamento.

Le CPU più moderne, in particolare quelle montate sui sistemi portatili, possono variare dinamicamente la frequenza di lavoro a seconda del carico di lavoro da sostenere, in modo da ridurre il consumo medio e la rumorosità (causata dalle ventole di raffreddamento). Ne è un esempio la tecnologia [SpeedStep](#) di Intel, ed equivalenti di altri produttori.

Supporti di storage (dati)

Supporti **hardware** adibiti a contenere le informazioni che trattiamo. In ambito informatico con il termine *storage* si identificano i dispositivi hardware, i supporti per la memorizzazione, le infrastrutture ed i software dedicati alla memorizzazione non volatile di grandi quantità di informazioni in formato elettronico.

Tipici esempi di supporto di storage sono:

- Hard disks
- CD rom
- Unità a nastro
- ...

Supporti fisici per la memorizzazione non volatile: <http://it.wikipedia.org/wiki/Storage>

In ambito informatico con il termine STORAGE (STORAGGIO, contenitore che conserva info) si identifica tutto l'hardware (dispositivi/supporti/infrastrutture) dedicato alla memorizzazione non volatile di grandi quantità di informazioni in *formato elettronico*. Tipici supporti storage sono: hard disks, CD-ROM, unità a nastro, floppy, chiavette USB.

Con il termine memorizzazione non volatile si intende la possibilità di immagazzinare delle informazioni in maniera persistente con una ragionevole probabilità che l'informazione rimanga inalterata per un ragionevole lasso di tempo.

Con il termine memorizzazione NON VOLATILE si intende la possibilità di immagazzinare delle informazioni IN MANIERA PERSISTENTE CON UNA RAGIONEVOLE PROBABILITÀ CHE L'INFORMAZIONE RIMANGA INALTERATA PER UN RAGIONEVOLE LASSO DI TEMPO, e per farlo in formato digitale binario e' necessaria avere un supporto fisico con le seguenti caratteristiche:

- sul supporto possono essere *scritte almeno una volta* sequenze di bit;
- le sequenze di bit scritte rimangono *inalterate a meno di una specifica operazione di modifica*;
- le sequenze di bit possono essere *lette un numero elevato di volte* senza alterarle.

La memoria RAM di un computer è la tipica memoria volatile, infatti è sufficiente che venga a mancare la corrente elettrica perché tutte le informazioni in essa immagazzinate vadano perdute.

Esistono molti supporti fisici che rispondono a queste caratteristiche, ma nella storia dell'informatica solo alcuni si sono affermati e sono tuttora utilizzati:

- Schede perforate (decisamente obsolete);
- Nastri magnetici;
- Dischi magnetici (hard disk e floppy disk);

- Dischi ottici;

Supporti fisici per la memorizzazione non volatile: <http://it.wikipedia.org/wiki/Storage>

Hard Disks

Link: http://it.wikipedia.org/wiki/Hard_disk

Il disco rigido o hard disk (anche chiamato disco fisso) e' un dispositivo utilizzato per la memorizzazione a lungo termine dei dati in un computer. Il disco rigido e' il **principale supporto di storage** dei computer per via della sua capienza (~300 GB) e della sua velocita' di trasferimento dati (~25MB per secondo) .

Memorie Flash

I supporti di storage *USB* o comunque di tipo *NAND* o *SSD* sono supporti removibili usati alla stregua dei vecchi floppy disk. La caratteristica peculiare e' che non hanno parti in movimento e sono quindi meno delicati rispetto ai supporti magnetici con testine in movimento su dischi (come gli hard disks) o a quelli a nastro (come nei dispositivi a cartucce quali DAT e similari). Sono meno fragili dei supporti ottici (ala CD-Rom) e meno sensibili all'esposizione solare diretta, piu' compatti fino ad essere tascabili nei tagli piu' piccoli.

Link: http://it.wikipedia.org/wiki/Chiave_USB

Propieta' dei supporti di storage

<i>Supporti di Storage</i>				
<i>Tipo di dispositivo</i>	<i>capienza</i>	<i>lettura</i>	<i>scrittura</i>	<i>re-scrittura[1]</i>
Hard disk	~300GB	vero	vero	vero
Floppy disk	1.44MB	vero	dipende [2]	dipende
CD ROM	~640MB	vero	falso	falso
CD R	~640MB	vero	vero	falso
CD RW	~640MB	vero	vero	vero
DVD ROM	~4GB	vero	falso	falso
Blue Ray	~35GB
USB Key	~1/128GB	vero	vero	vero

Nota:

per DVD e Blue RAY valgono gli stessi ragionamenti di scrittura (R) e re-scrittura (RW) fatti per i CD, solo con *tante sigle in piu'*.

Tutti supporti di storage hanno vita limitata (aspettativa di vita), influenzata fortemente dalla nostra modalita' di utilizzo e da fattori/accidenti esterni; sono quindi soggetti ad alcune norme di conservazione specifiche:

- HD: mantiene dati per 2-5 anni
- SUPPORTI MAGNETICI: evitare l'esposizione a rapide variazioni termiche e campi magnetici.
- SUPPORTI OTTICI: tenerlo lontano da fonti calore, non depositarlo senza la custodia dedicata, non sottoporlo a nessun tipo di sollecitazione meccanica (piegarlo, torcerlo) .

HARD DISK

Il disco rigido o hard disk (anche chiamato disco fisso) e' un dispositivo utilizzato per la memorizzazione a lungo termine dei dati in un computer. E' 1 supporto magnetico con 1 testina che si sposta per leggere dati su uno o piu' dischi.

Caratteristiche fondamentali: capiente (200-500 MB fino TB) + Trasn.dat (I/O) + riscrivibile (R,W,RW).

E' quindi il **principale supporto di storage** per via della sua capienza (~300 GB), ma rappresenta un limite ('collo di bottiglia') per le *prestazioni* dell'intero pc perche' e' molto lento nel trasferire dati (25-30

MB/s) rispetto alla cpu (< 3 GHZ). Negli ultimi anni le prestazioni della cpu sono raddoppiate circa ogni 18 mesi (Legge di [Moore](#)), mentre i dischi rigidi sono rimasti relativamente statici dal punto di vista delle prestazioni nello scambio dati.

Link: http://it.wikipedia.org/wiki/Hard_disk

- I vari formati dei supporti ottici DVD: http://it.wikipedia.org/wiki/Dvd#Formati_DVD
- I vari formati dei supporti ottici Blue-Ray Disk: http://it.wikipedia.org/wiki/Blu-ray_Disc

Classi di elaboratori

Cerchiamo di distinguere gli elaboratori per caratteristiche fisiche e funzionali, considerando la loro architettura, ambiente di utilizzo, target di utenza a cui sono rivolti.

Stabiliamo alcuni criteri per distinguere le varie classi:

- dimensioni
- mobilita'
- input | output
- capacita' di calcolo
- numero di utenti
- completezza dell'interfaccia utente

Calcolatori tascabili



Sono le macchine piu' semplici, generalmente poco flessibili, tendono ad essere piu' sofisticate nel caso delle calcolatrici scientifiche. Queste ultime sono spesso programmabili e hanno display comunque di dimensioni ridotte. L'interfaccia utente e' quindi molto semplificata.

Palmari e smart phones



Device caratterizzati dalle piccole dimensioni e dalla portabilita' (dimensioni *tascabili* e autonomia nell'ordine delle decine di ore), possono essere relativamente potenti e flessibili, soprattutto quando vengono equipaggiati con sistemi operativi analoghi a quelli dei personal computers. Le loro prestazioni in alcuni casi sono paragonabili a PC fissi di 4/5 anni precedenti alla loro introduzione sul mercato.

La capacita' di storage e' in genere molto limitata, ma e' possibile espanderla con schede di memoria. I modelli piu' recenti possono appoggiarsi a file-system di rete.

Autonomia: da 4 a 25 ore. Capacita' di storage: centinaia di MB

Ultraportabili (netbook)



Sono sostanzialmente l'anello intermedio tra i palmari e i portatili. Di prezzo intermedio tra i due, hanno in genere una dotazione hardware semplificata e ridotte capacita' di espansione rispetto a un portatile. Dotati di monitor e tastiere *standard* possono far *girare* gli stessi sistemi operativi dei computer fissi e quasi tutti i software con essi compatibili senza dover intervenire sulle interfacce utente.

Autonomia: da 1 a 8 ore. Capacita' di storage: da 4GB (stato solido) a 80GB (hard disk magnetici)

Portatili



Versioni portatili o per lo meno *trasportabili* dei personal computers. Hanno generalmente una tastiera completa ma di dimensioni ridotte, un display di ~ 15 pollici, processori ottimizzati al risparmio energetico, capacita' di stoccaggio limitate. L'espandibilita' e' molto limitata. I portatili hanno una batteria che ne consente l'utilizzo senza accesso alla rete elettrica fissa per alcune ore. I cosiddetti *trasportabili* hanno un'autonomia senza rete elettrica irrisoria. Visto il progressivo abbassamento dei prezzi stanno raggiungendo i computer fissi per diffusione.

Autonomia: da 2 a 10 ore. Capacita' di stoccaggio: da 250Gb fino a 1TB (Anche in RAID!)

Trasportabili

I cosiddetti *TRASPORTABILI* sono sostanzialmente delle work-station che si possono spostare da una luogo all'altro. Data l'autonomia irrisoria della batteria (anche 30/45 minuti) sono praticamente utilizzabili solo connessi alla rete elettrica.

Tipicamente sono piu' tozzi e pesanti dei portatili veri e propri, ma sono in genere piu' economici. Ad eccezione dei modelli particolarmente lussuosi con display da 17 o piu' pollici, tipicamente dedicati al multimedia.

Ultraportatili

Sacrificano le prestazioni per migliorare la portabilita': molto leggeri, grande autonomia (6 o piu' ore), display da 12 pollici o inferiore, tipicamente molto costosi.

Personal Computer



Sono le normali macchine fisse usati dagli utenti per gli usi piu' disparati. Sono le macchine piu' *comuni*, utilizzate per lavori di office automation e intrattenimento, dotate di processori che privilegiano le prestazioni rispetto al consumo (e al costo), e sono generalmente il risultato dell'assemblaggio di parti sfuse (schede madri, processori, periferiche) la cui combinazione puo' portare a risultati variegati. A differenza delle classi precedentemente descritte sono aggiornabili (posto che ci sia compatibilita' tra i singoli componenti).

Work Station

Sono postazioni singole ad alte performance per gli usi piu' impegnativi (grafica, calcolo). Fino ad alcuni anni fa erano generalmente caratterizzate da architetture multiprocessore, ma oggi si trovano *palmas* con piu' di una CPU quindi questo elemento non e' piu' determinante. Generalmente il costo e' di molte migliaia di euro. Questo termine sta diventando pero' obsoleto, data la difficolta' di caratterizzare queste macchine, per scopi promozionali / commerciali viene ormai usato per quasi tutti i sistemi in vendita...

Server



Un server e' un elaboratore che tipicamente eroga servizi a molti *utenti* (clients) e offre continuita' del servizio: un server tipicamente resta acceso 24/7 mentre una workstation otto ore al giorno. L'hardware di un server (in particolare le parti in movimento piu' soggette a usura: alimentatore, ventole, hard disks) dovrebbero quindi essere in grado di funzionare tutto il giorno tutti i giorni, dato che l'interruzione, seppur temporanea, dell'erogazione dei servizi puo' compromettere la produttivita' di molti *clients*. Componenti

come alimentatore o hard disk hanno comunque una vita limitata (anche se di *buona* qualita') e in previsione di un loro malfunzionamento e' possibile implementare sistemi di *ridondanza* relativamente economici come RAID e alimentazione ridondante. Puo' essere possibile a seconda del tipo di hardware sostituire *al volo* (hot swap) i componenti guasti senza essere costretti a spegnere la macchina per l'intervento di manutenzione.

I requisiti di capacita' di calcolo e memoria avranno da calcolare in base ai requisiti dei servizi in esecuzione sul server in relazione al numero dei client collegati e effettivo utilizzo di questi. Non e' raro trovare server con quantitativi di RAM o CPU ben piu' limitati dei clients che servono quando i servizi sono poco esosi di risorse o le richieste relativamente frequenti e soprattutto senza picchi di utilizzo. Alcuni servizi fondamentali come DNS, routing o firewall possono saturare la banda disponibile del server ben prima delle altre risorse di sistema. Capita ritrovare vecchie work-station in disuso, ormai sostituite da macchine piu' recenti e prestanti, svolgere decorosamente queste funzioni. Il che' rappresenta un'interessante opzione di riciclo, per quanto il consumo elettrico sia il piu' delle volte sproporzionato al lavoro svolto (tenendo come riferimento macchine a bassissimo costo di derivazione embedded o NAS di produzione recente, ma il discorso si farebbe complicato...) .

Tipicamente i server hanno capacita' di *collegamento* (hardware di rete) superiori ai singoli client (in proporzione alle macchine che devono servire), ma dato che spesso client e server condividono la stessa architettura hardware non e' semplice fare in modo che i server siano in grado di gestire input/output in quantita' direttamente proporzionale al numero dei clients. Si tenga conto che molti servizi non prevedono un flusso costante di dati ma solo chiamate sporadiche: questo permette di poter pianificare la ampiezza di banda necessaria al server sul valore massimo *stimato* di connessioni che si potrebbero verificare in un singolo momento piuttosto che sul reale numero di clients.

Spesso i server non hanno periferiche di input dedicate (mouse e tastiera), e preferibilmente montano hardware a basso consumo (anche per limitare di conseguenza l'usura dei sistemi di raffreddamento, e per permettere la concentrazione in singoli armadi o *server farm*).

Mainframe



Tipo di macchine "desuete" (a parte i casi in cui non se ne puo' fare a meno, ma la tendenza attuale e' di sostituirle con uno o piu' server), i mainframe sono fundamentalmente caratterizzate da una superiore capacita' nella gestione dell'input/output, e nel garantire la continuita' del servizio (molti mainframe permettono la sostituzione a caldo anche dei processori).

I mainframe o Sistemi Centrali sono computer utilizzati per applicazioni critiche soprattutto da grandi aziende e istituzioni, tipicamente per elaborare con alte prestazioni ed alta affidabilita' grandi moli di dati, come quelle in gioco nelle transazioni finanziarie, nei censimenti, nelle statistiche di industrie e clienti, nelle applicazioni ERP(Enterprise Resource Planning), etc

Cluster



soluzione *economica* per raggiungere alte prestazioni utilizzando piu' macchine *semplici* in parallelo per lo stesso scopo. Google ad esempio e' un cluster costituito da macchine x86 facilmente reperibili sul mercato. L'hardware e' quindi relativamente economico come gli interventi su di esso, e data la possibilita' di garantire il funzionamento del cluster a prescindere dal *fallimento* di un singolo membro si puo' tollerare hardware meno sofisticato.

Supercomputer



I supercomputer sono macchine che raggiungono le massime prestazioni possibili secondo la tecnologia disponibile per singola unita'. La loro architettura e' altamente specializzata e costruita su misura, sono generalmente molto costosi e poco aggiornabili.

La tendenza attuale e' di sostituirli con clusters di macchine piu' semplici da gestire: nelle _classifiche dei "super computer" molti sistema sono clusters.

Linguaggi di programmazione

I computer, come detto in precedenza, registrano e elaborano dei dati. Quest'ultima parte viene svolta con dei programmi che sono sostanzialmente delle sequenze di istruzioni, scritte in un linguaggio comprensibile all'elaboratore, ripetute piu' volte.

links: http://it.wikipedia.org/wiki/Linguaggio_di_programmazione

Sappiamo che i microprocessori elaborano sequenze binarie formate dai valori 1|0, ma sarebbe decisamente scomodo per gli sviluppatori scrivere i programmi in questo formato. Si *preferisce* scrivere i software in linguaggi formali piu' simili alla lingua umana (inglese), che oltre a essere piu' *significativi* per gli sviluppatori permettono di scrivere sequenze di comandi che potranno poi essere adattate automaticamente alle varie CPU.

La trasformazione del *codice sorgente* scritto dal programmatore in un linguaggio di piu' *alto livello* rispetto al *codice oggetto* direttamente eseguibile dal microprocessore viene svolta da un compilatore o da un interprete.

Esistono probabilmente centinaia di linguaggi di programmazione, caratterizzabili in base a:

- Distanza dall'hardware: linguaggi di basso livello come l'*assembler* sono molto piu' a contatto con le caratteristiche proprie della CPU rispetto a linguaggi di alto livello come *Python / Ruby / PHP*. La possibilita' di interagire in modo piu' stretto con l'hardware puo produrre codice piu' prestante.
- Semplicita': linguaggi di *alto livello*, soprattutto se interpretati, risultano molto piu' semplici da scrivere e mantenere, abbassando di conseguenza costi e tempi di sviluppo. Vista la attuale capacita' di calcolo dei microprocessori l'efficienza massima del codice tende oggi a non essere prioritaria rispetto al tempo di sviluppo (almeno in molti casi).

http://it.wikipedia.org/wiki/Linguaggio_di_programmazione#Linguaggi_compilati_e_linguaggi_interpretati

Compilatori

In informatica, un compilatore e' un programma che traduce una serie di istruzioni scritte in un determinato linguaggio di programmazione (codice sorgente) in istruzioni di un altro linguaggio (codice oggetto). Questo processo di traduzione si chiama compilazione.

Dal codice sorgente viene quindi generato un codice oggetto direttamente eseguibile dalla CPU, senza bisogno che al momento dell'esecuzione sia disponibile lo stesso linguaggio o componenti dell'ambiente si sviluppo. Dato che il compilatore ha la possibilita' di fare un'analisi generale del codice prima di cominciare a convertirlo il codice risultante e' in genere piu' ottimizzato rispetto al codice eseguito da un interprete.

La compilazione richiede una certa quantita' di tempo, ma questo accade **solo in fase di compilazione**. Una volta creato il codice oggetto questo sara' immediatamente eseguibile su tutte le macchine compatibili, con un evidente vantaggio di prestazioni rispetto ai linguaggi interpretati che devono procedere alla *traduzione* a ogni esecuzione del programma.

La necessita' di dover ricompilare ogni volta che si introduce un cambiamento del codice viene pero' visto come uno svantaggio rispetto ai linguaggi interpretati, dal punto di vista dello sviluppatore.

Poiche' in informatica non esistono *proiettili d'argento* (silver bullets, una soluzione per tutti i problemi) abbiamo a disposizione tanti linguaggi diversi, per supportare approcci piu' o meno strutturati a problematiche piu' o meno complesse: linguaggi piu' o meno semplici, piu' o meno efficienti nelle prestazioni, ottimizzati per fare specifiche compiti, etc. L'idea di base sarebbe di cercare di elaborare soluzioni semplici con gli strumenti piu' adeguati (i linguaggi di programmazione) alla natura del problema.

Esempio di codice in linguaggio C

```
#include <stdio.h>

int main()
{
    printf("Hello, World!");
    return 0;
}
```

links: [http://it.wikipedia.org/wiki/C_\(linguaggio\)](http://it.wikipedia.org/wiki/C_(linguaggio))

Analisi codice: [http://it.wikipedia.org/wiki/C_\(linguaggio\)#Hello_world.21](http://it.wikipedia.org/wiki/C_(linguaggio)#Hello_world.21)

Interpreti

[http://it.wikipedia.org/wiki/Interprete_\(informatica\)](http://it.wikipedia.org/wiki/Interprete_(informatica))

Un linguaggio interpretato e' un linguaggio di programmazione i cui programmi vengono eseguiti da un interprete. Tale approccio si distingue da quello dei linguaggi compilati: a differenza di un interprete, un compilatore non esegue il programma che riceve in ingresso, ma lo traduce in linguaggio macchina (memorizzando su file il codice oggetto pronto per l'esecuzione diretta da parte del processore).

L'approccio interpretato comporta una minore efficienza a run-time; un programma interpretato, in esecuzione, richiede piu' memoria ed e' meno veloce, a causa dello overhead introdotto dall'interprete stesso. Durante l'esecuzione, l'interprete deve infatti analizzare le istruzioni a partire dal livello sintattico, identificare le azioni da eseguire (eventualmente trasformando i nomi simbolici delle variabili coinvolte nei corrispondenti indirizzo di memoria), ed eseguirle; mentre le istruzioni del codice compilato, gia' in linguaggio macchina, vengono caricate e istantaneamente eseguite dal processore.

Esempio in python:

```
print "hello world!"
```

Analisi: scrive hello world a schermo !

Java e simili

link http://it.wikipedia.org/wiki/Macchina_virtuale_Java

La macchina virtuale Java, detta anche Java Virtual Machine o JVM, e' la macchina virtuale che esegue i programmi in linguaggio Java bytecode, ovvero i prodotti della compilazione dei sorgenti Java. La JVM e' formalmente una specifica, mantenuta da Sun Microsystems. Qualsiasi sistema che si comporti in modo coerente con tale specifica sara' quindi da considerarsi una particolare implementazione della JVM. Esistono implementazioni software per praticamente tutti i sistemi operativi moderni, sia gratuite che commerciali. Inoltre, esistono implementazioni speciali per particolari ambienti hardware/software (per esempio telefoni cellulari e palmari), e persino implementazioni hardware.

La disponibilita' di implementazioni della macchina virtuale Java per diversi ambienti operativi e' la chiave della portabilita' di Java, proclamata nello slogan write once, run everywhere ("scrivi una volta, esegui dappertutto"). La macchina virtuale realizza infatti un ambiente di esecuzione omogeneo, che nasconde al software Java (e quindi al programmatore) qualsiasi specificita' del sistema operativo sottostante:

```
=====
| Software applicativo Java |
=====
| Java Virtual Machine |
|=====
| Sistema Operativo |
|=====
```

Sostanzialmente il codice Java viene scritto per essere eseguito da una Java Virtual Machine (JVM), della quale viene rilasciata una versione per ogni diverso sistema operativo. In questo modo modo lo sviluppatore puo' scrivere un codice che sia indipendente dal sistema in cui verra' eseguito, dato che la

JVM (Java Virtual Machine) dovrebbe comportarsi allo stesso modo su qualunque sistema venga installato.

Licenze software

La licenza in ambito informatico e' il contratto che puo' accompagnare un prodotto software. Tale contratto specifica le modalita' con cui l'utente puo' usare tale prodotto, garantendo dei diritti ed imponendo obblighi.

La licenza e' imposta da chi detiene il copyright sul prodotto software; la sua validita' dipende dalla presenza del diritto d'autore, questo perche' solo chi detiene il copyright ha il diritto di far rispettare in ogni sede la licenza stessa.

In vari casi l'autore puo' rilasciare un prodotto software sotto licenze differenti, lasciando all'utente la possibilita' di scegliere quella preferita.

Tipi di licenze

Publico dominio

Il pubblico dominio non e' propriamente una licenza, per quanto sia determinante nei rapporti tra l'opera e i fruitori. L'essenza sta proprio nel non sottoporre l'opera a meccanismi di tutela della proprieta' dell'opera. L'opera e' patrimonio dell'umanita'. Tutte le opere delle quale il copyright e' scaduto per termini di leggi (a seconda del paese possono volerci fino a un centinaio di anni) diventano di pubblico dominio e quindi liberamente fruibili. Ad es. non dovete chiedere il permesso a nessuno per stampare e diffondere l'Odissea di Omero, posto che non contravveniate ad altre leggi vigenti nel farlo.

WWW, TCP/IP, [sqlite](#) sono sotto pubblico dominio

Publico dominio = cessazione di tutti i diritti di esclusiva: http://it.wikipedia.org/wiki/Publico_dominio

Freeware

<http://it.wikipedia.org/wiki/Freeware>

Software *a gratis*. Nel senso che non si paga per utilizzarlo. Gratis come in "birra gratis per tutti".

L'utente non pu' modificarlo in alcun modo, a meno che non espressamente previsto.

Il termine freeware indica un software che viene distribuito in modo gratuito.

Il freeware e' distribuito indifferentemente con o senza codice sorgente, a totale discrezione dell'autore e senza alcun obbligo al riguardo. È sottoposto esplicitamente ad una licenza che ne permette la redistribuzione gratuita. Il software freeware viene concesso in uso senza alcun corrispettivo, ed e' liberamente duplicabile e distribuibile, con pochissime eccezioni.

Shareware

Shareware e' una tipologia di licenza software molto popolare sin dai primi anni Novanta. Vengono distribuiti sotto tale licenza in genere programmi facilmente scaricabili via Internet o contenuti in CD e DVD quasi sempre allegati alle riviste di Informatica in vendita in edicola.

Il software sotto tale licenza puo' essere liberamente ridistribuito, e puo' essere utilizzato per un periodo di tempo di prova variabile (generalmente 30 giorni). Scaduti questi termini, per continuare ad utilizzare il software e' necessario registrarlo presso la casa produttrice, pagandone l'importo. All'avvio dell'applicazione shareware generalmente un Nag Screen informa l'utente su come effettuare la registrazione e sulle condizioni di utilizzo.

Licenze per software liberi

http://it.wikipedia.org/wiki/Software_libero

Il Software libero e' una questione di liberta', non di prezzo. Per capire il concetto, bisognerebbe pensare alla liberta' di parola e non alla birra gratis [NdT: il termine free in inglese significa sia gratuito che libero, in italiano il problema non esiste].

Cosa sia, o meglio quanto debba essere libero un software per poter essere definito tecnicamente *libero* e' motivo di discussione sia tra sviluppatori che tra i non addetti ai lavori, data l'applicazione del concetto di copyleft ad altre discipline diverse dall'informatica.

Generalmente con software libero si intende quel codice rilasciato sotto licenza GPL, per quanto altre licenze vengano generalmente accettate come *libere*. Stabiliti dei criteri, almeno per quanto riguarda il software, si puo' procedere a una comparazione tra le varie licenze alternative alla GPL:

- http://it.wikipedia.org/wiki/Comparazione_di_licenze_libere
- <http://www.gnu.org/philosophy/categories.it.html>

Dal sito del progetto Gnu possiamo *copiare* la seguente definizione:

Definizione di Software Libero

Questa definizione ha lo scopo di chiarire quali sono i requisiti che un certo programma deve soddisfare perche' lo si possa considerare "software libero".

Il Software libero e' una questione di liberta', non di prezzo. Per capire il concetto, bisognerebbe pensare alla liberta' di parola e non alla birra gratis [NdT: il termine free in inglese significa sia gratuito che libero, in italiano il problema non esiste].

L'espressione "software libero" si riferisce alla liberta' dell'utente di eseguire, copiare, distribuire, studiare, cambiare e migliorare il software. Piu' precisamente, esso si riferisce a quattro tipi di liberta' per gli utenti del software: Liberta' di eseguire il programma, per qualsiasi scopo (liberta' 0). Liberta' di studiare come funziona il programma e adattarlo alle proprie necessita' (liberta' 1). L'accesso al codice sorgente ne e' un prerequisito. Liberta' di ridistribuire copie in modo da aiutare il prossimo (liberta' 2). Liberta' di migliorare il programma e distribuirne pubblicamente i miglioramenti, in modo tale che tutta la comunita' ne tragga beneficio (liberta' 3). L'accesso al codice sorgente ne e' un prerequisito.

Un programma e' software libero se l'utente ha tutte queste liberta'. In particolare, se e' libero di ridistribuire copie, con o senza modifiche, gratis o addebitando delle spese di distribuzione a chiunque ed ovunque. Essere liberi di fare queste cose significa (tra l'altro) che non bisogna chiedere o pagare nessun permesso.

Bisogna anche avere la liberta' di fare modifiche e usarle privatamente nel proprio lavoro o divertimento senza doverlo dire a nessuno. Se si pubblicano le proprie modifiche, non si deve essere tenuti a comunicarlo a qualcuno in particolare o in qualche modo particolare.

La liberta' di usare un programma significa liberta' per qualsiasi tipo di persona od organizzazione di utilizzarlo su qualsiasi tipo di sistema informatico, per qualsiasi tipo di attivita' e senza dover successivamente comunicare con lo sviluppatore o con qualche altra entita' specifica. Quello che conta per questa liberta' e' lo scopo dell'utente, non dello sviluppatore; come utenti potete eseguire il programma per i vostri scopi; se lo ridistribuite a qualcun altro, egli e' libero di eseguirlo per i propri scopi, ma non potete imporgli i vostri scopi.

La liberta' di ridistribuire copie deve includere le forme binarie o eseguibili del programma e anche il codice sorgente, sia per le versioni modificate che non modificate (distribuire programmi in formato eseguibile e' comodo per avere sistemi operativi liberi facili da installare). E' legittimo anche se non c'e' alcun modo di produrre una forma binaria o eseguibile (dal momento che alcuni linguaggi non supportano questa caratteristica), ma si deve avere la liberta' di ridistribuire tali forme nel caso si trovi o si sviluppi un modo per farlo.

Affinche' le liberta' di fare modifiche e di pubblicare versioni migliorate abbiano senso, si deve avere accesso al codice sorgente del programma. Percio', l'accessibilita' al codice sorgente e' una condizione necessaria per il software libero.

Un importante modo di modificare un programma e' quello di includervi funzioni e moduli liberi gia' esistenti. Se la licenza del programma prevede che non si possano includere moduli gia' esistenti, ad esempio se richiede che voi possiate aggiungere solo codice di cui detenete il copyright, allora la licenza e' troppo restrittiva per essere considerata libera.

Queste liberta' per essere reali devono essere irrevocabili fin tanto che non si fa qualcosa di sbagliato: se lo sviluppatore del software ha il potere di revocare la licenza anche senza che l'utente sia causa di tale revoca, il software non e' libero.

Tuttavia, certi tipi di regole sul come distribuire il software libero sono accettabili quando non entrano in conflitto con le liberta' principali. Per esempio, il copyleft, noto anche impropriamente come "permesso d'autore", e' (detto poche parole) la regola per cui, quando il programma e' ridistribuito, non e' possibile aggiungere restrizioni per negare ad altre persone le liberta' principali. Questa regola non entra in conflitto con le liberta' principali, anzi le protegge.

Indipendentemente dal fatto che si siano ottenute copie di software GNU a pagamento o gratuitamente, si ha sempre la liberta' di copiare e cambiare il software, e anche di venderne copie.

Software libero non vuol dire non-commerciale. Un programma libero deve essere disponibile per uso commerciale, sviluppo commerciale e distribuzione commerciale. Lo sviluppo commerciale di software libero non e' piu' inusuale: questo software commerciale libero e' molto importante.

Copyleft != Copyright

Il copyleft e' un metodo generico per rendere un programma libero ed imporre che tutte le modifiche e versioni estese del programma siano anch'esse software libero.

L'espressione inglese copyleft, lett. "(diritto di) copiatura (esclusiva) abbandonato" , e' gioco di parole su copyright (lett. "diritto (esclusivo) di copiatura"), in cui la seconda parola del composto, "right" e' scambiata con "left" individua un modello alternativo di gestione dei diritti d'autore basato su un sistema di licenze attraverso le quali l'autore (in quanto detentore originario dei diritti sull'opera) indica ai fruitori dell'opera che essa puo' essere utilizzata, diffusa e spesso anche modificata liberamente, pur nel rispetto di alcune condizioni essenziali. Nella versione pura e originaria del copyleft (cioe' quella riferita all'ambito informatico) la condizione principale obbliga i fruitori dell'opera a rilasciare eventuali modifiche apportate all'opera a loro volta sotto lo stesso regime giuridico (e generalmente sotto la stessa licenza). In questo modo, il regime di copyleft e tutto l'insieme di liberta' da esso derivanti sono sempre garantiti.

Il modo piu' semplice per rendere un programma, o altro lavoro, libero e' dichiararlo di dominio pubblico, privo di copyright. [NdT: in Italia questo concetto a rigore non esiste, perche' l'autore non puo' rinunciare alla paternita' dell'opera; effetti simili si possono ottenere con l'uso di una licenza permissiva, come la licenza di X del MIT]. Cio' permette a chiunque di condividere il programma e i suoi miglioramenti. Tuttavia permette anche a chi non vuol cooperare di convertire il programma in software proprietario. Costoro possono infatti apportare modifiche, che siano significative o meno, e distribuire il risultato come prodotto proprietario. Chi riceve il programma nella forma modificata non ha la liberta' data dall'autore originario; l'intermediario l'avra' rimossa.

Segue su: <http://www.gnu.org/copyleft/copyleft.it.html>

Link alla Licenza GPL (e DSFG):

- http://it.wikipedia.org/wiki/GNU_GPL
- <http://www.gnu.org/copyleft/gpl.html>
- FAQ sulla GPL: <http://www.fsf.org/licensing/licenses/gpl.html>
- DSFG (Le Linee Guida Debian per il Software Libero (Debian Free Software Guidelines - DSFG): http://www.debian.org/social_contract#guidelines

Non ci dovrebbe essere nessuna differenza tecnica tra "free software" e "open source".

Licenza BSD

Le licenze BSD sono una famiglia di licenze permissive per software. Molte sono considerate libere ed open source. Il loro nome deriva dal fatto che la licenza BSD originale (detta anche licenza BSD con 4 clausole) fu usata originariamente per distribuire il sistema operativo Unix Berkeley Software Distribution (BSD), una revisione libera di UNIX sviluppata presso l'Universita' di Berkeley.

Links:

- http://it.wikipedia.org/wiki/Licenze_BSD
- <http://www.opensource.org/licenses/bsd-license.php>

Licenza Apache

In ambito informatico, con Licenza Apache (ingl:Apache License) si intende una precisa licenza, che similmente alla licenza BSD, permette di utilizzare il codice dentro progetti non open source e senza obbligo di rilascio dei sorgenti modificati.

La Licenza Apache venne scritta dalla Apache Software Foundation (ASF) per il piu' usato server web, Apache. È una licenza per software libero, ma non compatibile con la GNU GPL.

Questa licenza assume particolare importanza in quanto usata per molti altri software rilasciati dalla Apache Software Foundation ([ASF](#)) e altri progetti,

Esistono le versioni 1.0, 1.1 e 2.0.

Eula

EULA o End User License Agreement (accordo di licenza con l'utente finale) e' il contratto tra il fornitore di un programma software e l'utente finale. Tale contratto assegna la licenza d'uso del programma all'utente nei termini stabiliti dal contratto stesso.

Segue su: <http://it.wikipedia.org/wiki/EULA>

- Multilicenze (enterprise)
- Licenze singole
- OEM : legate all'hardware, non possono essere spostata su una macchina diversa

Link alla licenza EULA di MS Windows XP:

- <http://alice.andreamanni.com/eula.txt>
- <http://www.microsoft.com/windowsxp/eula/home.mspx>

-
- 1 Un floppy disk e' scrivibile se la targhetta e' posizionata in modalita' di scrittura, e all'opposto puo' diventare in sola lettura
 - 2 Con *re-scrittura* si intende la possibilita' di modificare un dato piu' di una volta. Ad esempio su un supporto ottico *Recordable* si puo' scrivere una sola volta (**WORM**), dopodiche i dati scritti non sono ulteriormente modificabili (ma possono essere aggiunte successive *sessioni* di scrittura se previsto). Questa caratteristica li rende indicati come supporto di back-up. Supporti *Rewritable* possono invece essere scritti piu' volte (la posizione dei cristalli pit/lane interni puo' essere modificata piu' volte, quindi i dati possono essere modificati / eliminati).