

Jurassic News

Commodore Club: Turbo Tape

Il racconto: La guerra dei cloni

Darwin: Il libro dei perché del PC IBM

Prova hardware: Next Computer - CUBE

Come eravamo: Un Sinclair QL italiano

TAMC: L'indecidibilità

Laboratorio: Si fa presto a dire gioco

Retro linguaggi: SNOBOL

Retrocomputing: Specializzazione museale

Retrocomputer Magazine

Anno 10 - Numero 56 - Novembre 2015



Jurassic News

Rivista aperiodica di Retrocomputer

Coordinatore editoriale:

Tullio Nicolussi [Tn]

Redazione:

redazione@jurassicnews.com

Hanno collaborato a questo numero:

Lorenzo [L2]

Salvatore Macomer [Sm]

Sonicher [Sn]

Besdelsec [Bs]

Lorenzo Paolini [Lp]

Giovanni [jb72]

Antonio Tierno

Damiano Cavicchio

Dino Baldi

Riccardo Franch

Diffusione:

Lettura on-line sul sito o attraverso il servizio Issuu.com; il download è disponibile per gli utenti registrati.

Sito Web:

www.jurassicnews.com

Contatti:

info@jurassicnews.com

Copyright:

I marchi citati sono di copyrights dei rispettivi proprietari.

La riproduzione con qualsiasi mezzo di illustrazioni e di articoli pubblicati sulla rivista, nonché la loro traduzione, è riservata e non può avvenire senza espressa autorizzazione.

Jurassic News

promuove la libera circolazione delle idee

Jurassic News

E' una fanzine dedicata al retro-computing nella più ampia accezione del termine. Gli articoli trattano in generale dell'informatica a partire dai primi anni '80 e si spingono fino ...all'altro ieri.

La pubblicazione ha carattere puramente amatoriale e didattico, tutte le informazioni sono tratte da materiale originale dell'epoca o raccolte su Internet.

La redazione e gli autori degli articoli non si assumono nessuna responsabilità in merito alla correttezza delle informazioni riportate o nei confronti di eventuali danni derivanti dall'applicazione di quanto appreso sulla rivista.

Il contenuto degli articoli è frutto delle conoscenze, esperienze personali e opinioni dei singoli autori; possono pertanto essere talvolta non precise o differire da fonti "ufficiose" come Wikipedia e siti Web specializzati.

Sono gradite segnalazioni di errori, imprecisioni o errate informazioni che possono, a discrezione della redazione, essere oggetto di errata-corrige in fascicoli successivi.

*Scrivere a:
redazione@jurassicnews.com
dettagliando il più possibile l'argomentazione.*

Per un attimo...

“Per un attimo ripenso al Commodore 64 e a tutti i sogni che era capace di regalare. E mi dispiace per chi, pur satollo di modernissimi gadget e fabulistiche opportunità hi-tech, non abbia conosciuto quelle emozioni.”

Con questa frase si conclude un articolo apparso tempo fa sul Fatto Quotidiano, a firma Umberto Rapetto.

E' un articolo interessante che vi consiglio senz'altro di leggere (riferimenti in fondo alla pagina) perché, al di là della nostalgia del passato, l'autore propone una considerazione su quello che lui identifica come “Vero digital-divide”.

Infatti, sostiene Umberto, il vero gap tecnologico di oggi non è più fra chi può permettersi e chi invece non ha accesso a determinate tecnologie, ma piuttosto fra la disponibilità di esse e l'uso che se ne fa.

Quante risorse sprecate!

Basta pensare a quanto se ne sta in idle un sistema operativo moderno, rispetto alla percentuale di CPU effettivamente impegnata a fare qualcosa di concreto. Queste sono risorse/energia vera che se ne va in fumo (in calore, per essere precisi) senza nulla produrre di utile.

E non si tratta tanto del fatto che vorremmo che il nostro apparecchio elettronico, sia esso un computer o un tablet o un telefonino, stia la maggior parte del tempo a calcolare dati o anche semplicemente a muovere uno sprite sullo schermo per farci rilassare, ma si tratta di constatare quanto poco consapevoli siano le nuove generazioni rispetto alla potenzialità che tengono sul palmo della mano ogni giorno.

Giocare va bene, comunicare anche, ma quando constatiamo che i vari store di app sono pieni di assolute nullità, non possiamo che farci venire il magone, proprio come Umberto.

Buona lettura.

[rif: <http://www.ilfattoquotidiano.it/2012/12/10/dannato-commodore-64/441524/>]

4 *Per un attimo...*

Retro linguaggi

12 *SNOBOL (parte 3) - dati e strutture*

Come eravamo

20 *Un Sinclair QL italiano*

Prova hardware

26 *Next Computer - CUBE*

Come eravamo

38 *Prodest?... Di certo non giova a Olivetti*

Celebrazioni

54 *I vent'anni di Windows95*

Retrocomputing

Specializzazione museale 6

Il racconto

Automatik (25) - La guerra dei cloni 8

Darwin

Il libro dei perché del PC IBM (parte 3) 14

Laboratorio

Si fa presto a dire gioco... 18

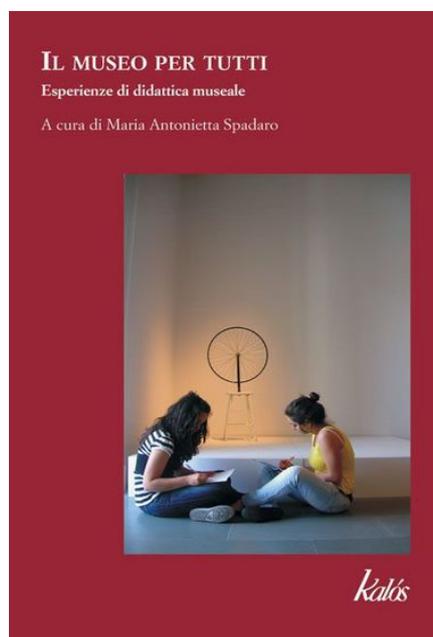
Commodore Club

TurboTape: mettiamo il turbo alle cassette 42

TAMC

L'indcidibilità (parte 1) 50

Specializzazione museale



di **Tullio Nicolussi**

Una delle aspirazioni più comuni di chi si dedica alla passione del retro computing è quella di aprire un museo. Per fare questo si capisce subito che sono necessarie tre condizioni imprescindibili:

1. una collezione di “reperti” significativa;
2. una locazione adatta;
3. una affidata organizzazione volontaria.

Per quanto riguarda la locazione c'è poco da fare: o si trova uno sponsor pubblico, possibilmente affidabile, oppure ci si autofinanzia ma è una strada quantomeno improba dati i costi in gioco.

Per quanto riguarda il terzo punto è evidente che senza una schiera più o meno nutrita di amici che condividono la stessa passione e che si impegnano a fornire la propria opera gratuitamente, non si va da nessuna parte! Sì, si può pensare di esporre una propria collezione in locali privati, visitabile su appuntamento o in rare occasioni annuali, ma certo non si può parlare di “museo pub-

blico”.

E veniamo al primo punto, cioè alla consistenza della collezione. Il discorso vale per una struttura permanente, il classico museo da visitare, ma può essere trasposta anche per una esposizione temporanea: le classiche esposizioni da collezionisti.

Qui entrano in gioco le diverse tipologie di esposizione che si possono realizzare. Ad esempio se si intende aprire un museo monomarca, poniamo Commodore (per non parlare della solita Apple), è evidente che bisogna disporre di “quasi” tutto ciò che è stato prodotto dalla casa canadese e possibilmente di qualche rarità, cioè di modelli poco diffusi, oserei dire “millesimati” come il famoso modello “Gold”. Poiché il visitatore sarà presumibilmente un cultore della marca, troverà la propria soddisfazione nell'ammirare dal vivo i modelli meno diffusi, le curiosità e così via.

Se si opta invece per un percorso generale sulla storia del calcolo automatico, si potranno esporre pezzi comuni e la completezza del

percorso sarà realizzato dalle testimonianze significative di una certa epoca/mercato/cultura senza pretendere l'eshaustività.

Questo tipo di percorso espositivo è da una parte più facile e dall'altra più impegnativo nella realizzazione. Più facile perché si potrà utilizzare meglio il proprio posseduto, anche se non di particolare valore, mentre è particolarmente difficile far percepire l'evoluzione del calcolo automatico esponendo solo una serie di modelli. È facile accostare un mainframe, diciamo un IBM 360 ad un Amiga e confrontarne le dimensioni riportate alla mera potenza di calcolo. Ma non sarebbe corretto: sono due sistemi molto lontani come età, scopo, soluzioni tecnologiche ed utilizzo.

In questo tipo di esposizione la scelta dei reperti da inserire deve essere accurata in modo che non vi siano inutili ripetizioni o una mancanza di equilibrio nella sequenza esposta. Ad esempio un Amiga 500 ha lo stesso valore espositivo di un Amiga 600, un Apple II equivale ad un Apple //e. Lo so che tutte le macchine sono diverse per progetto, epoca, diffusione, etc... Stiamo parlando di una sequenza storica, non di una rassegna esaustiva!

Pensiamo anche allo scopo dell'allestimento, sia esso temporaneo che permanente. Se ad esempio si vuole mostrare ad una scolaresca la storia dell'informatica, saranno significativi i supporti e le periferiche. Per un collezionista specializzato il pezzo forte sarà quel certo modello raro, firmato dal progettista, visto in precedenza solo in fotografia...

Un altro aspetto da esaminare è la tipologia di museo che si vuole realizzare. Da qualche anno si sta affermando una scuola di pensiero che vede i musei non come mere esposizioni statiche, ma luoghi di ricerca, recupero ed educazione. I musei più visitati al giorno d'oggi devono creare degli eventi, cioè occasioni di coinvolgimento per i visitatori, siano essi delle scolaresche di vario

grado e sia occasionali ospiti come le famiglie.

In questo campo l'informatica ha molto da dire perché è relativamente facile disporre di oggetti funzionanti visto che non stiamo parlando di secoli fa! Non è necessario che tutti i pezzi in esposizione funzionino; si possono creare degli angoli se non intere sale, dove il visitatore può interagire con i vecchi sistemi, magari con l'assistenza di una guida che ha lo scopo di spiegare ed invitare a provare qualche percorso.

È facilissimo ottenere il coinvolgimento dei più giovani predisponendo delle postazioni gioco. I ragazzi e perfino i bambini sotto i dieci anni sono naturalmente attratti dai videogiochi e lo stimolo di provare non è difficile da far nascere. Certo il valore dell'esperienza sarà maggiore se il soggetto riuscirà a riflettere, per quanto le sue esperienze gli consentono, sull'evoluzione di quel comparto; cioè farà dei confronti fra la sua esperienza recente con le varie console super-grafiche e la povertà della grafica che il museo gli propone.

In Inghilterra il Museo Nazionale di Informatica riesce ad organizzare delle vere e proprie sessioni full-immersion con i ragazzi, spiegando prima come sono codificate le informazioni ed i comandi su un particolare sistema e poi invitandoli ad eseguire il proprio codice direttamente su una macchina reale.

Certo, questo tipo di esperienza non può sostituire un vero e moderno percorso di istruzione che porti i ragazzi a diventare dei coder. Esperienze come i gruppi CodeDojo sono strategiche ed efficaci. Però, dopo aver programmato in Scratch, un breve approccio ai vecchi metodi lo vedrei come un tassello importante nella preparazione di una professione futura per coloro che la vorranno intraprendere.

(=)

Automatik (26)

La guerra dei cloni



Di Lorenzo Paolini

Mi è stato chiesto di parlare di quello che era l'ambiente dei noleggiatori di apparecchi per l'intrattenimento nell'epoca d'oro delle sale giochi. Ne posso parlare per la mia limitata esperienza e per il periodo e il territorio che conosco. Posso cioè descrivere una finestra di un mondo che è molto più grande di quanto io ho potuto esplorare ed inoltre non posso attribuire nessun carattere statistico e/o scientifico a questa analisi.

Quando mi inserii in questo ambiente lavorativo non avevo nessuna esperienza a riguardo, se si esclude la frequentazione dei bar di paese e dei corrispondenti flipper, scenario di qualche sfida fra amici e di una discreta, seppur non eccelsa, pratica di successo. Più recentemente ero stato incuriosito dalla presenza nei bar dei primi videogiochi e in particolare di Space Invaders, che avevo frequentato, ma quasi esclusivamente da spettatore, all'epoca del militare (1980-

1981). Per la verità Space Invaders non mi piaceva granché con la sua musica incalzante e sincopata e l'inevitabilità del soccombere ben presto alle orde di invasori. Trovavo che erano soldi buttati quelli inseriti nella gettoniera di quelle macchinette. Con il flipper non avevo la stessa relazione, nel senso che ci giocavo abbastanza e quindi di soldini ne investivo... ma era diverso, almeno mi sembrava: con il flipper hai un rapporto fisico che ti porta ad acquisire competenza come si trattasse di imparare un mestiere oppure di imparare a ballare: sono conquiste tecniche che si raffinano con l'assiduità d'uso, cioè più pratici e più diventi bravo. Senza contare che i più bravi potevano far durare la partita moltissimo con l'uso esclusivo della loro tecnica e ovviamente con un pizzico di fortuna. C'era con noi in compagnia un tizio con qualche anno più di me che aveva veramente una tecnica sopraffina. Era uno spettacolo

vederlo giocare e spesso si formavano alle sue spalle dei veri e propri crocchi di gente che tifava per lui.

Con *Space Invaders* non mi sembrava potesse essere la stessa cosa. Immagino che l'uso affinasse l'abilità di manovra con il joystick e le capacità di coordinazione, ma era inevitabile che si soccombessero e nemmeno i più bravi andavano oltre pochi livelli. Quello che ho capito dopo era la diversa "anima" dei due giochi: il flipper rimaneva uguale a se stesso indefinitivamente, mentre *Space Invaders* aumentava la difficoltà livello dopo livello.

Mai avrei pensato nel 1980 che mi sarei impiegato qualche anno dopo come tecnico riparatore proprio di videogiochi e flipper!

La vita è strana, ma forse le strade sono tracciate per noi e non facciamo che seguirne l'andamento, magari con poche possibilità di arbitraria scelta quando ci troviamo davanti ad un bivio.

La verità è che fra il 1980 e il 1985 ci fu una vera e propria *excalation* nei videogiochi "da bar" (chiamiamoli così o arcade, se preferite) con decine di titoli nuovi arrivati dagli States e dal Giappone ogni mese. I noleggiatori, cioè coloro che affittavano (il mio titolare diceva "mettere giù") un gioco in un certo locale si trovarono spiazzati sotto l'aspetto tecnico-organizzativo. Erano prima di tutto dei traslocatori (con quel loro correre su e giù per il territorio a ritirare un gioco, ripararne un'altro ed avviare un nuovo posto), discreti elettricisti, geniali meccanici di precisione (il jukebox è una serie di leve e servo meccanismi elettromeccanici) ma erano un disastro sull'elettronica! Giocoforza era così: questi avevano imparato il mestiere da buoni artigiani praticando "le botteghe" e creandosi sul campo quella che oggi si chiamerebbe *knowledge base*. Erano persone di bassa scolarità ai quali mancava una base teorica per sperare di poter leg-

gere un qualsivoglia manuale in inglese (e nemmeno in italiano a volte...).

Ma i videogiochi da bar si guastavano, principalmente perché la gente si terminava manovrando il joystick o si faceva saltare le cervella nella frenesia del pigia-pigia sui tasti. E così giù pugni sulla plancia e schiaffi cattivi al joystick quando la famigerata scritta "GAME OVER" mostrava tutta la sua irriverenza sul monitor. Finché si trattava di cambiare un pulsante, sostituire un joystick o l'alimentatore e il video in un cabinet, non c'erano problemi ma le cose si facevano via via più complicate: non c'era speranza di mettere mano alle piastre elettroniche e quando si guastava un flipper bisognava cambiarlo subito nel locale, senza avere peraltro grandi speranze di una rapida e soprattutto economica riparazione.

Un flipper moderno, cioè non elettromeccanico, funziona sotto la supervisione di un programma eseguito da un microprocessore; il resto sono device che manovrano luci o relè con una abbondante schiera di triac collegati in pull-up. I triac che pilotano le bobine dei bump (quella sorta di fungo che scatta quando la pallina lo colpisce) devono sopportare dei picchi di corrente notevoli e perciò sono soggetti ad una certa percentuale di guasti. Prima che queste aziende si rendessero conto di non poter fare a meno della specializzazione di un tecnico elettronico, erano dissanguate da quei pochi laboratori o colleghi più lungimiranti che si facevano pagare le riparazioni a suon di quattrini! Far riparare una scheda Bally, uno dei più diffusi produttori di flipper, poteva costare dal mezzo milione in su (un flipper completo ne costava circa cinque). Quello che era più fastidioso e sicuramente più costoso era il fermo macchina. Infatti un flipper posizionato in un locale giusto poteva portare a casa un incasso netto mensile dalle cinquecentomila Lire in su!

Romano, il titolare della ditta dove lavoravo, aveva un principio: -"un gioco doveva pagarsi nel giro di un anno, altrimenti l'investimento era in perdita".

I flipper però avevano una vita lunga, e penso che anche oggi, pur non praticando da anni (decenni in effetti) l'ambiente, si possano tranquillamente far lavorare per un decennio... Diverso il fenomeno dei videogiochi: duravano come novità un annetto, massimo due, poi diventavano di seconda scelta in un declino rapido che già dopo tre anni li rendeva non rimpiazzabili nei posti profittevoli. In aggiunta di questo la proliferazione di titoli, spesso pochissimo diversi uno dall'altro, alimentava il desiderio del giocatore di affrontare nuove sfide in piattaforme diverse che sembravano (e per buona parte lo erano) lontane anni luce per grafica, suono e divertimento dai titoli di appena un anno prima. Guai a sottovalutare l'aspetto novità: il videogiocatore è diverso dal giocatore tradizionale del flipper. Quest'ultimo è un affezionato cliente del locale e gioca perché si trova con gli amici nel posto che fa parte del suo vissuto relazionale; il primo invece è capace di cambiare locale solo perché ci trova un gioco più nuovo.

La conseguenza era che si doveva acquistare nuovi titoli con sempre maggiore frequenza. Questo lo volevano i giocatori ma anche i proprietari dei locali che "marcavano stretto" il titolare del noleggiatore ma anche i collaboratori che gestivano le installazioni. Ricordo in particolare un tipaccio (non c'è altro termine se non forse "avanzo di galera") che ci minacciava se non gli avessimo portato un "gioco da far soldi", come sbrigativamente bofonchiava. Anche i titoli più famosi, come ad esempio Pac-Man, in realtà nel business del noleggiatore non resistevano molto più a lungo degli altri. Erano conosciuti perché in qualche modo la stampa e la televisione ne parlava e quindi più o meno la gente ci giocava,

ma senza quell'insano meccanismo ludopatico (ora so che si chiama così) che portava il giocatore a svuotarsi letteralmente le tasche in una frenesia inarrestabile di nuove partite. I vari Pac-Man, Defender, Space Invaders, si conservavano bene nelle sale giochi ma non nei locali dove il posto era singolo o al massimo in coppia. Nelle sale giochi, vuoi per affezione e vuoi per variare la minestra, i clienti continuavano a giocare ma gli incassi non si potevano paragonare alle performance sensazionali di qualche titolo particolarmente azzecato.

Voglio svelare un segreto del mestiere: mai mettere giù due giochi nello stesso locale, a meno che il passaggio di persone sia notevole, tipo nelle stazioni. Se lo fate uno dei due giochi farà in pratica da tappezzeria. Questo principio i vari gestori dei bar non lo capivano: pensavano che se un gioco incassa cento, due giochi incasseranno duecento... completamente sbagliato!

Nella frenesia della sostituzione delle macchine le ditte di noleggio vissero una crisi di crescita. Si chiama così quando il mercato potenziale è molto più grande di quello che puoi coprire e cercando di approfittare della situazione inneschi un meccanismo poco virtuoso che porta a spese sempre crescenti con una immobilizzazione sempre meno sopportabile del capitale investito. Anche la ditta Automatik dove andai a lavorare, corse questo rischio ma Romano, il titolare, era persona accorta e abbastanza intelligente da capire che era necessario intervenire su vari fronti per garantire una crescita lenta ma costante dei profitti della ditta. Questi fronti erano in buona sintesi tre: incremento del personale "di manovalanza", dotarsi di una figura specializzata nelle riparazioni elettroniche e investire capitale nel rinnovo continuo della dotazione di apparecchi. Il personale di bassa specializzazione migrava i giochi da un locale all'altro, eseguiva piccole

riparazioni e raccoglieva gli incassi; il tecnico specializzato doveva occuparsi del laboratorio e delle riparazioni più cospicue in modo da minimizzare il tempo di quiescenza degli apparecchi. Era poi necessaria una adeguata organizzazione di marketing per gli acquisti e per rinnovare o mantenere il parco clienti ed infine una precisa gestione amministrativa-finanziaria.

Il piano era buono, peccato che non ci fossero le condizioni per realizzarlo: in pratica Romano era un ottimo artigiano ma la sua visione del business era da paraocchi e come ogni artigiano abituato a guadagnare dal suo lavoro giorno per giorno, pensò che anche nell'ambizioso sviluppo della sua azienda si potesse risparmiare stiracchiando la coperta di qua e di là. Così il personale "logistico" venne raddoppiato (da uno si passò a due) ma il nuovo acquisto non si rivelò autonomo. Un po' per la giovane età e un po' per mancanza di iniziativa, questo giovane apprendista rimase apprendista oltre ogni ragionevole limite ipotizzabile per un lavoro tanto semplice: in fondo si trattava di girare con un furgoncino per movimentare i cabinet e fare piccole riparazioni; attività che si rivelarono sopra le forze di questo giovane di belle speranze e così Daniele, che era il vecchio tuttofare della ditta, finì per portarlo appresso tutto il giorno con poco costruito dal punto di vista dello sviluppo del mercato aziendale.

Sul tecnico (che fui io) Romano sbagliò un'altra volta. Diciamo che più che sbagliare persona (non voglio buttarmi giù più di quanto meriti) sbagliò target. Nella sua ingenuità condita da prudenza contadina, Romano pensò di risparmiare e così prese me che ero meno scolarizzato e preparato degli altri ingegneri che si presentarono al colloquio di lavoro. Mi poteva pagare di meno e invece del "mega laboratorio super attrezzato" si convinse che bastava un locale decentemente ampio con un banchetto da bricolage in un angolo, qualche pinza e cacciavite, un salda-

tore a stagno e un tester. Era tutto!

Inizialmente sono convinto che si pavoneggiò della sua pensata al risparmio con quel suo parente, ingegnere elettronico che lavorava in una multinazionale a Milano, il quale gli aveva stilato il progetto del laboratorio. Conobbi questa persona durante il mio colloquio di lavoro e mi restituì subito l'impressione di una persona preparata che sapeva il fatto suo.

Il locale "laboratorio" prese vita nella trasformazione del garage doppio che si trovava addossato alla casa di proprietà di Romano e sede della prestigiosa ditta Automatik. La dotazione tecnica consigliata dall'ingegnere sortì il misero elenco di attrezzatura elencato prima e questo fu tutto!

Venne una volta quell'ingegnere di Milano (mi spiace non ricordo assolutamente il suo nome) e la sua unica reazione fu uno scuotimento del capo, molto più espressivo di tanti discorsi.

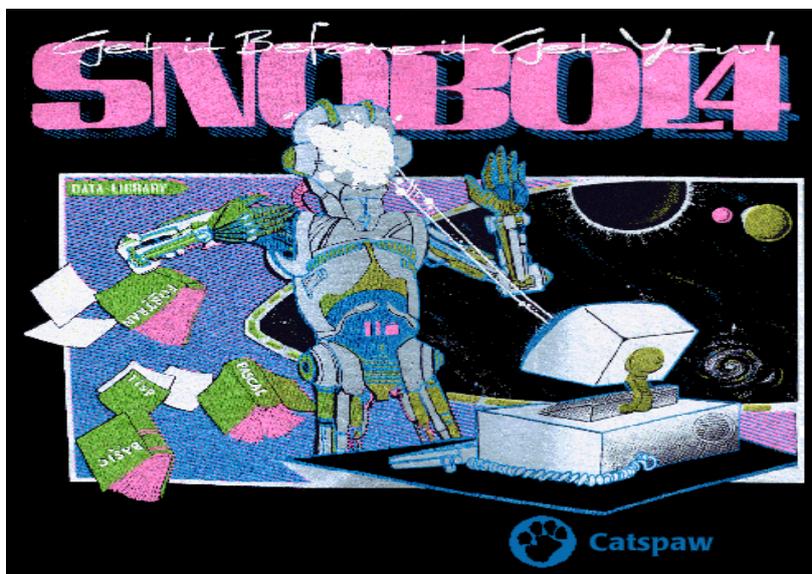
Il bello è che all'inizio gli effetti positivi delle riparazioni che potei mettere in atto dall'alto (dal basso, sarebbe meglio dire...) della mia limitatissima esperienza si videro: i flipper non rimanevano più inoperosi per mesi in attesa che arrivasse la scheda riparata, la pleora di monitor video guasti si ridusse di gran lunga e anche qualche riparazione "sfiziosa" sulle piastre dei giochi giapponesi furono eseguite con successo.

Poi però la frenesia iniziale scemò in conseguenza dell'andamento non proprio nella direzione sognata da Romano per il suo business. Io me ne andai nel giro di due anni, il settore marketing (che faceva Romano in persona) non diede prova di grande performance e l'amministrazione, affidata alla moglie del titolare, era da sei meno meno...

Ma questa è un'altra storia...

(=)

SNOBOL (Parte 3)



di Salvatore Macomer

Quando si esaminano i linguaggi di programmazione più antichi (per modo di dire...), a volte si scoprono particolarità e soluzioni modernissime che pensiamo siano state introdotte molto più tardi nell'ingegneria del software, ma che appunto si scopre erano già presenti. Secondo un mio collega insegnante, questo deriva dal fatto che i linguaggi di alto livello sono stati ideati e spesso realizzati da una singola persona o da un team molto ristretto. Nella progressione evolutiva della programmazione dei calcolatori, alcune istruzioni vengono dimenticate per poi ricomparire come delle novità due o tre generazioni avanti!

Un singolo individuo, ma anche una “scuola di pensiero”, spesso fa delle scelte personali. Ad esempio basta che un certo modo di programmare, diciamo una istruzione, non sia “simpatica” ed ecco che viene stralciata dal progetto del nuovo linguaggio in fase di definizione. Altre volte succede viceversa,

cioè l'utilizzo di certi paradigmi che l'autore trova confortevoli, vengono traslati par-paro nel nuovo linguaggio, anche se possono inquinare la purezza stilistica.

Nel caso dello Snobol ad esempio troviamo una costruzione che forse non ci aspettavamo di scoprire: l'uso dei puntatori.

L'istruzione di assegnazione come:

```
MESE = 'GENNAIO'
```

crea in memoria una struttura stringa che ha un certo valore, cioè la sequenza di caratteri che formano la parola GENNAIO.

\$MESE è il puntatore alla struttura che la contiene e lo statement:

```
$MESE = 'FEBBRAIO'
```

equivale a:

```
GENNAIO = 'FEBBRAIO'
```

Questo concetto dei puntatori in SNOBOL

forse crea un po' di confusione a chi è abituato al loro uso in altri linguaggi, come ad esempio nel C.

Nel C l'assegnazione:

```
Nome = 'pippo' ;
```

crea anche il puntatore alla variabile Nome e si può usare il suo indirizzo di memoria nel modo seguente:

```
&Nome = 'topolino' ;
```

Nota: so che la sintassi non è corretta e che bisognerebbe assegnare il valore tramite un `memcpy()`; è solo per far capire il concetto!

Mentre il puntatore indiretto:

```
Nome* = 'topolino' ;
```

equivale al concetto di puntatore nello SNOBOL, cioè modifica l'area di memoria puntata indirettamente dal valore contenuto ad un certo indirizzo. Quindi in prima analisi è questo il parallelo con lo SNOBOL, cioè un indirizzamento indiretto, non un vero puntatore alla memoria, anche se il linguaggio lo chiama "puntatore".

Dato che lo SNOBOL nasce con l'intento di lavorare sulle sequenze di caratteri e loro combinazioni (quello che noi chiamiamo comunemente "stringhe"), è evidente che il tipo di dato principale del linguaggio siano proprio esse. Il concetto di stringa è però più complesso rispetto alla sequenza di caratteri in memoria, assomiglia al concetto di classe nei moderni linguaggi ad oggetti, anche se nello SNOBOL il paradigma non è così sofisticato.

In effetti il tipo base dello SNOBOL non è la stringa, come noi la intendiamo, ma piuttosto il "pattern" che possiamo pensare sia equivalente al concetto matematico di "insieme".

Il linguaggio si traduce in pratica in una sequenza di valutazioni che hanno lo scopo di costruire pattern partendo da altri pattern. Vediamo quindi alcuni esempi che ci aiuteranno a capire meglio.

Costruiamo un pattern:

```
MESE = 'GEN' | 'FEB' | 'MAR' | 'APR'
```

MESE è un pattern costruito da quattro pattern definiti (in pratica delle costanti) che sono i valori delle sigle dei primi quattro mesi dell'anno. L'operatore '|' equivale all'unione e la sequenza può essere arbitrariamente lunga, ad esempio per comprendere tutti e dodici i mesi dell'anno.

Costruire un pattern da altri pattern è semplicemente mettere in sequenza i pattern sorgente:

```
PODIO = PRIMO SECONDO TERZO
```

Dove PRIMO, SECONDO e TERZO sono tre pattern precedentemente definiti.

Il linguaggio SNOBOL è ricco di funzioni che lavorano su pattern (o su stringhe, se volete pensare il tutto in maniera tradizionale). Alcune di queste funzioni sono "tradizionali", nel senso che ogni linguaggio le implementa. Parliamo di funzioni come il calcolo della lunghezza di una stringa ad esempio. Altre funzioni sono più sofisticate o quantomeno differiscono dai linguaggi tradizionali.

Ecco come si fa la comparazione di due predicati (i predicati sono il risultato della valutazione di pattern o, in ultima analisi, sono le tradizionali stringhe).

Esempio sia:

```
X = 3
```

```
Y = '3'
```

L'operatore EQ verifica se i suoi argomenti hanno lo stesso valore:

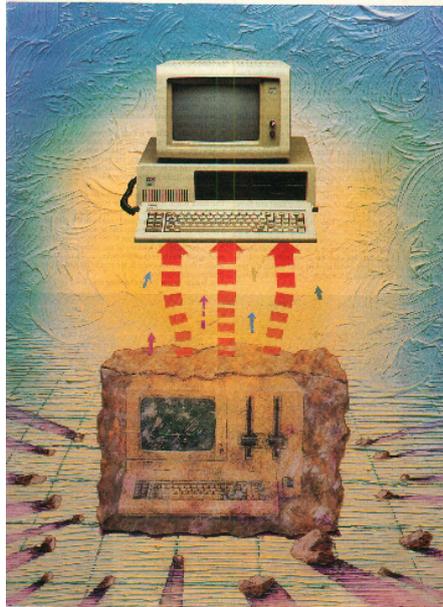
EQ(X, Y) restituisce true, mentre IDENT(X, Y) restituisce false perché pur avendo lo stesso valore numerico, i due pattern sono di tipo differente.

Diciamo che per certi versi lo SNOBOL ha anticipato alcuni dei concetti alla base della programmazione ad oggetti.

(...continua...)

Il libro dei perché del PC IBM

(parte 3)



di Riccardo Franch

Una domanda si sono fatti tutti su un particolare del progetto del PC IBM: - “Perché l’8088 e non l’8086?”

David Bradley, un programmatore che partecipò all’avventura della nascita di questo sistema che volendo o no è l’assoluto riferimento per l’informatica personale dal 1981 ad oggi, da questa come tutte le altre risposte: -”Semplicemente questione di costi; il bus a 8 bit del fratellino minore rispetto al 16 bit full dell’8086, permetteva un risparmio sensibile di costo non tanto per il prezzo del microprocessore ma per la semplificazione che ne derivò per la parte di bus e memoria”.

Il coprocessore matematico 8087 era previsto nel progetto ma semplicemente al momento dell’uscita non era disponibile da Intel in quantità industriali. La piastra madre del PC venne progettata “alla cieca”,

cioè con il socket pronto ma senza un vero collaudo del coprocessore matematico.

Sulla qualità del firmware la IBM puntò molto. Da una parte doveva inglobare funzioni complesse senza delegarle al programmatore. Una delle routine più “toste” è quella che decodifica i tasti premuti sulla tastiera e ne fa la decodifica. Anche la diagnostica fu spinta molto. La procedura di POST venne criticata perché di fatto rendeva lo start-up della macchina particolarmente lento. Ma era necessaria: le componenti non erano così affidabili come oggi e in particolare la memoria che andava opportunamente testata ad ogni avvio. La IBM puntava molto sul mantenere anche per il prodotto a basso costo (si fa per dire) la sua proverbiale qualità dell’assistenza.

Un particolare apparentemente stride con la filosofia conservativa e prudentiale della real casa: la scelta di Microsoft come part-

ner per il sistema operativo e per il BASIC. E' ben nota la defianza di Gary Kildall della Digital Research Incorporated che preferì non rinunciare ad un volo sul suo aereo privato per recarsi a discutere con IBM sull'adozione del suo sistema operativo per la nuova macchina. Vanno date tutte le attenuanti a Kildall ma va anche riconosciuto al management di IBM l'aver rischiato sulla nuova azienda (la Microsoft come azienda per azioni, diremo in Italia, fu fondata solo due mesi prima che uscisse il PC nel 1981).

Diciamo una cosa: il DOS fa abbastanza schifo (e lo faceva di più nel 1981)!

Chi lo afferma, specificando che si sta riferendo dal punto di vista della Computer Science, è Thomas Haigh uno storico della tecnologia che opera alla University of Wisconsin-Milwaukee. "Le pesanti limitazioni tecniche" - dice Haigh - "potevano essere accettabili nei primi anni di vita del prodotto, ma divennero insopportabili nel prosieguo".

Ricorda Haigh che il DOS nelle versioni fino al 1995 non era nemmeno compatibile al 100% con il BIOS e la cosa si fece evidente con l'uscita di Windows 95. Per superare le limitazioni sulla memoria vennero inventati vari "quasi standard" come il LIM (Lotus - Intel - Microsoft) che permetteva lo sfruttamento dello spazio di memoria superiore al megabyte da parte di alcuni programmi avidi di spazio come lo spreadsheet Lotus 1-2-3. Memoria e pseudo-multitasking furono presi in carico da altri oggetti che di fatto "vestivano" il DOS di un abito di decenza. Pensiamo ad esempio al prodotto Quarterdeck Extended

Memory Manager che tutti gli utilizzatori "seri" installavano sulle loro macchine 386 e seguenti.

Altri prodotti addirittura si riscrivevano le routines di base. WordPerfect conteneva i driver per pilotare la stampante senza passare per le limitazioni del DOS.

Il PC aveva un'arma che nessun altro possedeva: una capillare rete di vendita in tutto il mondo. Così ogni store IBM spingeva la macchina a cominciare dai possessori delle varie classi di mainframe che avrebbero potuto (sostenevano i venditori) disporre di programmi avanzati come il foglio di calcolo e la video scrittura, impossibili (o consapevolmente trascurate) sui sistemi "alti" dell'azienda.

Il successo del prodotto sorprese IBM ma anche la stimolò sulla strada del miglioramento. Una delle idee era cambiare il sistema operativo e trasportare un sistema operativo serio e relativo software dai sistemi superiori alla macchina base. Questo lavoro era stato abbandonato inizialmente perché non c'era materialmente il tempo e IBM voleva uscire in fretta. Ma anche il secondo tentativo fallì e questa volta la causa furo-





no le stringenti limitazioni dell'architettura della macchina. Si sarebbe dovuto cambiare tutto o quasi, cosa che IBM tenterà (terzo tentativo) con i sistemi PS/2 e il sistema operativo OS/2.

E i cloni? Vogliamo parlare del fenomeno della copiatura della macchina perpetrata dai Giapponesi prima e dai Tailandesi subito dopo?

La IBM aveva creato una macchina aperta. L'idea era che le terze parti avrebbero dovuto sviluppare hardware e soprattutto software e contribuire così al successo del sistema in un circolo win-win.

I "cinesi" capirono che qui c'era da mangiare: l'architettura era semplice, le componenti reperibili e a basso costo, il margine incredibilmente alto... insomma c'era spazio! Così invece che costruire pezzi aggiuntivi si costruirono macchine intere. L'unica cosa che IBM riuscì a proteggere fu il BIOS. Era un prodotto sviluppato in casa e poteva essere protetto contro la copia. Potevano "i cinesi" comprare la licenza d'uso? Certo, ma a che prezzo! Così gli intraprendenti manager dell'elettronica d'oriente cavarono il coniglio dal cappello: clonarono il BIOS ma non copiandone il codice, ricostruendo per intero passo dopo passo tutte le funziona-

lità ma con un codice diverso. Ovviamente l'operazione venne contrastata sul piano del diritto ma mentre i giudici si perdevano in perizie su perizie e gli avvocati prosperavano con i loro lauti onorari, la gente comprava i cloni cinesi che costavano la metà e funzionavano esattamente alla stessa ma-

niera.

Nei primi anni il clone si comprava senza sistema operativo (ufficiale), poi nelle lusinghe della Microsoft di Bill Gates capì che gialli o a strisce e stelle, una licenza d'uso portava dollari. Nella sua visione futuristica-imprenditoriale o semplicemente per una gran botta di fortuna, il contratto di fornitura del DOS alla IBM non conteneva la clausola di esclusività. La Microsoft poteva venderlo a chiunque e lo fece!

Un aspetto particolare riguarda l'editoria periodica: la diffusione del PC provocò una crescita quasi incontrollata di riviste; nascevano continuamente nuove testate dedicate al computer e al PC in particolare in tutti i suoi aspetti hardware e software. Nel 1985 si contarono 55 nuove pubblicazioni solo negli Stati Uniti! Una situazione over-size che spingeva gli editori ad annunciare esclusive eclatanti che spesso si rivelavano dei flop. Ci fu una testata che celebrò come prodotto dell'anno un software annunciato ma che non vide mai la luce! Da questi episodi nacque il termine "vaporware" come aggettivo per gli annunci senza seguito pratico.

Termina qui questa "storia del PC IBM" scritta partendo dalle testimonianze di chi ha partecipato di persona al progetto o ha vissuto l'ambiente nel quale si è sviluppato.

Che dire d'altro? L'informatica, ma la società stessa, sarebbe la stessa oggi se nel 1981 la IBM non avesse preso la decisione di costruire il PC?

Chissà...

(=)

Riferimenti bibliografici.

[1] - Edward Bride - *The IBM Personal Computer: A Software-Driven Market* - in *COMPUTER, COVER FEATURE*, Published by the IEEE Computer Society 0018-9162/11 © 2011 IEEE;

[2] - <http://www-03.ibm.com/ibm/history/ibm100/us/en/icons/>;

[3] - David J. Bradley - *The creation of the IBM PC* - in *BYTE* Settembre 1990;

[4] - E. Bride, "The Media Are the Message: 'The Influencers'," *IEEE Annals of the History of Computing*, Oct.-Dec. 2006

Riferimenti alle immagini.

[1] - *BYTE* - Settembre 1990; come byte rappresenta la nascita del PC IBM dal prototipo grezzo del System/23 DataMaster;

[2] - "IBM PC 5150". Con licenza CC

BY-SA 3.0 tramite
Wikimedia Commons

- http://commons.wikimedia.org/wiki/File:IBM_PC_5150.jpg#/media/File:IBM_PC_5150.jpg;

[3-4] - <http://oldcomputers.net/ibm5150.html>;

[5] - <http://abccresellers.com/store/product101.html>



Si fa presto a dire gioco...



Di Lorenzo Paolini

Ciao a tutti.

Qualcuno già mi conosce per i miei racconti della serie *Automatik*, che gli amici di *Jurassic News* mi dicono siano molto apprezzati. Grazie.

Del mio periodo di lavoro nella ditta di video-noleggio nei primi anni '80 mi è rimasto ben poco di concreto: qualche schema o manuale, vari chip e un paio di schede che avevo portato a casa per ripararle a tempo perso ma che poi non ho più consegnato sia per dimenticanza che per la maniera, diciamo "brusca", con la quale ho chiuso la mia esperienza nel settore. D'altra parte non è che in ditta si brillasse per ordine e nessuno sapeva quanto materiale c'era in giro nei vari magazzini e nemmeno quanti giochi erano effettivamente in produzione.

Come ho già avuto modo di raccontare, quando ho iniziato la mia avventura nel mondo dell'intrattenimento, non mi aspettavo che l'elettronica digitale fosse ad un livello molto più alto rispetto al comparto "elettronica digitale" che praticavo a livello amatoriale dilettrandomi con i progetti a base di microprocessori che le riviste proponevano in quell'epoca.

Infatti è facile pensare che se uno Z80 a 4 MHz comanda una macchina abbastanza sofisticata come un computer MSX (la cui definizione dello standard è del 1983), dentro un videogioco cosa può esserci di più complicato?

Invece, ad onta del fatto che in fondo un videogioco da bar sempre lo stesso programma deve far girare, non ha le complicazioni della memoria di massa da gestire, etc... ci si ritrova con schemi elettrici complicatissimi e una ricchezza di componenti insospettabili.

E' il caso di questo gioco della Atari-Namco (non ho mai capito chi fosse il vero costruttore...), **Pole Position**, uscito per le sale gioco nel 1982 e del quale possiedo la raccolta di schemi elettrici che possono rendere l'idea di cosa stiamo parlando.

Credo che chiunque abbia frequentato una sala giochi dal 1982 al 1985 circa l'abbia visto. In un cabinet "stile monoposto" ci si sedeva come su una vera macchina: freno e acceleratore erano i due pedali, il volante ovviamente era il comando principale e per il cambio... non mi ricordo affatto come funzionasse... basterebbe spulciare lo schema

ovviamente, ma non è questo l'importante adesso.

Per essere precisi ne esisteva anche una versione up-side, cioè verticale dove si guidava in piedi; ma vuoi mettere lo stare seduti dentro una monoposto?

Vedendo le immagini dello schermo e i video presenti in Rete, si vede che la grafica non era poi eccezionale (ma stiamo parlando del 1982, ragazzi!). Quindi è spontaneo stupirsi quando si scopre che l'elettronica prevede non uno ma ben tre microprocessori: due Zilog Z8002 e uno Zilog Z80.

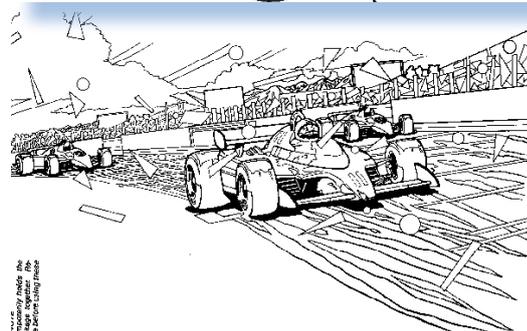
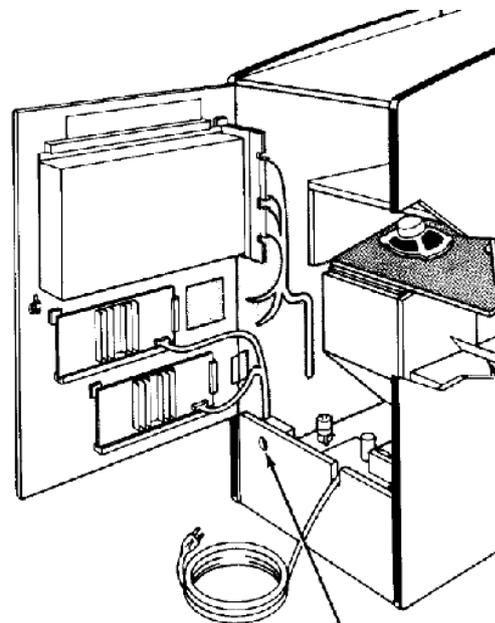
Oltre a questo non si contano i chip custom ospitati sulle schede e la dotazione di ROM che è formata da ben 14 chip 2764. La RAM è invece minima, appena un K, dovendo servire praticamente solo per tenere i punteggi.

Si accede alla sezione elettronica da un pannello posto sul retro del cabinet, praticamente sulla schiena del giocatore, dove sono posizionati anche due generosi altoparlanti. Il suono, venendo da dietro, rende molto bene l'atmosfera del motore e la parte sonora in generale è molto curata. Se ne occupa lo Z80 coadiuvato da parecchio materiale campionato che è registrato nelle ROM.

I due Z8002 si occupano invece della grafica (sempre con generosa mole di dati in ROM per le texture) e l'altro di tutto la logica video, anche questa complicatissima mancando di un processore grafico specializzato.

Chi volesse cimentarsi nella decifrazione degli schemi elettrici li può trovare sul sito <https://www.mikesarcade.com/arcade/manuals.html>;

Le immagini sono tratte dal sito: http://www.arcade-museum.com/game_detail.php?game_id=9063



THE ABOVE INFORMATION IS FOR INFORMATIONAL PURPOSES ONLY AND IS NOT TO BE USED FOR REPAIR OR MAINTENANCE.

Schematic Package Supplement to



Operation, Maintenance, and Service Manual

*Pole Position is engineered and designed by Namco Ltd. Manufactured under license by Atari, Inc.

1M

© ATARI, INC., 1982

SP-219 Sheet 1A
2/1/82

(=)

Un Sinclair QL italiano



*di Damiano Cavicchio
con la collaborazione di Dino Baldi*

Tutti conoscono il computer Sinclair QL, ma per chi non ricordasse esattamente di cosa si trattava ricordiamo qui le note più importanti. QL è un computer nato agli inizi degli anni 80 dall'evoluzione dei progetti "home" di Sir Clive Sinclair (papà dello ZX80-81 e poi dello Spectrum) che forse per il suo contenuto tecnologico avrebbe meritato più fortuna.

Alcune delle sue caratteristiche erano assolutamente originali per una macchina di quella classe: come ad esempio il suo Sistema Operativo.

Sfortunatamente però il suo arrivo sul mercato non fu privo di errori, ritardi e imperfezioni dovute ad una certa fretta (o meglio ad un annuncio prematuro della disponibilità della macchina). Inoltre anche le altre case costruttrici si davano da fare e al momento della disponibilità agli utenti il Sinclair QL non apparve più così rivoluzionario.

Ad esempio nella memoria di massa Sinclair scelse di equipaggiare il QL con due microdrive, che nel tempo si sono rivelati poco affidabili e con prestazioni troppo distanti

da quelle di un vero disco magnetico. 100 Kb per unità non sono molti e anche all'epoca poco rivaleggiava con la capienza dei floppy disk che stava crescendo rapidamente.

Infine la mania del risparmio dotò la macchina di una tastiera non certo all'altezza di una macchina professionale.

Forse l'errore è stato proprio quello di puntare verso il mercato "office" piuttosto che sfruttare la potenza del sistema per il comparto "home computing".

La potenza del processore 68008, la grande quantità di memoria indirizzabile e la relativa facilità di gestione dei segnali sulla "user port", unita alle ottime caratteristiche dell'interprete BASIC fornito in dotazione (il Super-BASIC, per essere precisi), indusse qualcuno ad usare la "base" per progetti di automazione semi-industriale.

Anche in Italia molti appassionati ed estimatori del QL con grande fantasia creativa e competenze informatiche, sono riusciti a modificare il QL adattandolo alle loro esigenze, come è possibile vedere dalla foto di apertura e da quella della pagina a fronte.



Oltre alle realizzazioni “private” con tanto di contenitore in truciolato di legno, magari funzionali ma certo non catalogabili nel comparto dei VAR (Value Added Reseller), esistono tracce di utilizzo della macchina Sinclair QL “modata”, cioè modificata in qualche parte per adattarvi periferiche o altro che la rendessero adatta a certe funzioni.

Ad esempio la SPEM di Torino, del cui lavoro rimane pochissimo, si è cimentata. L'amico Dino Baldi è riuscito a “mettere le mani” sopra un esemplare e ne ha descritto le caratteristiche nel suo seguitissimo blog (riferimenti in fondo all'articolo).

Seguiamo la storia della “scoperta”, così come raccontata da Dino:

- “Durante la magica domenica di Brusaporto Retrocomputing, si avvicina al nostro stand un simpatico signore. Dopo quattro chiacchiere mi chiede se sono interessato a un QL, vista la nostra sfilza di

Sinclair.

Una breve trattativa ci porta a concordare un appuntamento a chiusura della mostra, presso la sua casa nei dintorni di Dalmine.”

Il tutto si presenta perfettamente conservato, con scatola originale, guida ad anelli totalmente in italiano, alimentatore e una notevole dotazione a corredo:



Oltre alla stampante Sinclair QL 1000 c'è il notevole monitor Prism QL14 ed un cavo RGB-scart.

Purtroppo, come correttamente dichiarato all'atto dell'acquisto, la tastiera risulta non funzionante, impedendo così di verificare lo stato dei delicati microdrive.

Il bello però doveva ancora venire. Il nostro ospite ci guida in un locale adibito a cantina, su un tavolo spicca una strana macchina in lamiera, a prima vista auto-costruita.



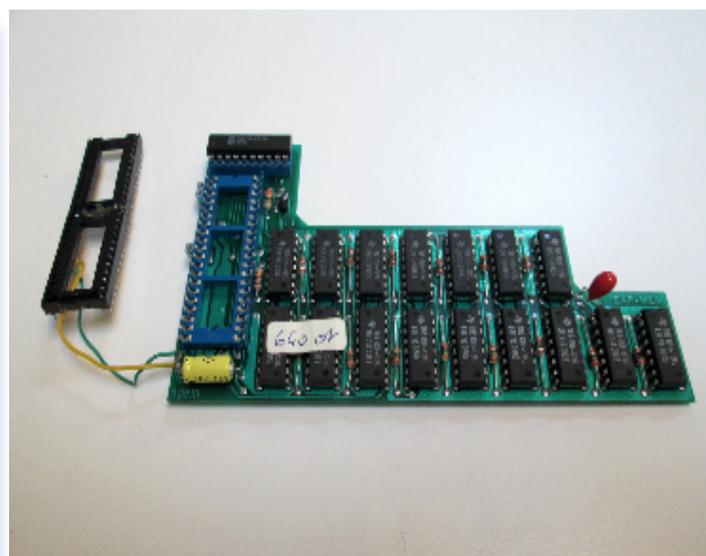
Con grande sorpresa scopriamo che l'esemplare in oggetto è un interessantissimo Spem System2, ossia un Sinclair QL, montato in un case metallico, equipaggiato con controller floppy e due drive da 3"1/2. Una grossa ventola, alimentata a parte, spunta in modo un po' buffo dalla carcassa. A corredo alcune scatole contenente documentazione e qualche libro. Una stampante Epson LX-400 ad aghi completa il tutto.

Prima di procedere a "sezionare" questo inaspettato acquisto, vale la pena spendere due parole sulla azienda che l'ha prodotto.

La SPEM nasce a Torino nei primi anni

'80 sotto la direzione di Masoero Guido (chi l'ha conosciuto lo descrive come un pioniere dell'informatica personale in Italia) e si occupa di micro computer importando nel nostro Paese i prodotti che all'estero avevano un certo successo e per i quali non esisteva una esclusiva all'importazione. Ad esempio importò l'ACORN Archimedes e il QL Sinclair.

Progettò espansioni per i sistemi che importava; in particolare è nota questa scheda di espansione da 512 Kb per il QL:



Ancora una realizzazione della SPEM per il QL; si tratta dell'interfaccia per floppy disk da 3,5":



ESPANDETE IL VOSTRO QL A 512 KBYTES

Si!!! Avete letto bene. Proprio il Vostro QL può essere espanso a 512 kb con un **costo molto basso** e con possibilità di usare i FLOPPY disk contemporaneamente. Questo è possibile con il nuovo sistema messo a punto dalla S. P. E. M. di Torino che senza usare la porta di espansione vi monta all'interno del computer le nuove RAM da 256 kbit e il relativo circuito di commutazione.

Con un regalo a scelta tra: Un programma per **CREARE una RAM DISK**
Una scheda con zoccolo **PORTA EPROM per QL**

Vi costa solo **420.000** lire.
Scheda porta EPROM con zoccolo lire 15.500
Kit di espansione per esperti £ 260.000

Telefona per informazioni allo 011/856519 di Torino
SPEM di MASOERO GUIDO - Via Ponchielli, 26 C - 10154 TORINO

AGGIORNATE IL VOSTRO QL A JS

Si!!! Avete letto bene. È una nuova iniziativa della SPEM. A tutti i possessori di QL inglesi che fanno montare l'espansione interna di memoria, la SPEM dà in omaggio una coppia di ROM versione JS. Per i QL italiani in regalo un programma su EPROM.

Listino prezzi IVA esclusa:

Espansione interna a 512 Kb totali
montaggio compreso L. 340.000
Espansione interna a 512 Kb totali
in Kit di montaggio L. 200.000
Scheda espansione interna da 512 Kb
senza saldature L. 360.000

In tutti i casi si può usare
il floppy disk con l'espansione

ROM vers JS di ricambio
con istruzioni di montaggio L. 68.000
Interfaccia per floppy disk
SPEM con TOOLKIT L. 199.000

SPEM di Guido Masoero
Via Ponchielli, 26C - 10154 Torino - Tel. 011/856519

Singolo floppy disk 3,5"
da 720 Kb formattati Panasonic L. 299.000
Doppio floppy disk 3,5"
da 720 Kb formattati Panasonic L. 550.000
Scheda porta EPROM con Eprom 27128
da programmare L. 20.000
Programmatore di Eprom 2764-27128
della CAMEL per QL L. 250.000
Programmatore di Eprom seriale RS 232
da 2716 a 27513 L. 550.000
Espansione di memoria da 512 Kb
per ATARI 520 ST L. 200.000

Sconti per quantità ai sigg. rivenditori.

Le novità del QL della SPEM DI TORINO

PRODOTTI PER SINCLAIR QL.
QL inglese 128 Kb programmi PSION + manuale in italiano L. 330.000
QL inglese 640 programmi PSION + manuale in italiano L. 600.000
QL italiano 128 Kb programmi PSION + manuale in italiano L. 390.000

DIGITALIZZATORE video con programma di LOOK & SHOW su disco o MDV
Permette di ottenere immagini da una telecamera da videoregistratore.
Regolazione di contrasto, di luminosità e dei singoli colori.
Il programma stampa in multiback con scala dei grigi o cerchi perfetti su EPSON PANASONIC o Matheeman TALLY. L. 200.000
Telecamera BN TV circuito chiuso adattata al digitalizzatore L. 300.000

TASTIERA QL stile PC At completa di tastiera numerica, 10 tasti funzione esterni al solito posto, cavo da 60 cm, compatibilità totale con tutti i comandi e programmi, si collega al posto della tastiera QL. L. 179.000

CABINET stile Pc per QL da usare con tastiera SPEM - spazio previsto per due microdrive, 2 floppy, 1 hard disk CEPRUM L. 49.000
Scheda CEPRUM per 3 EPROM da 32 o 64 Kb sino a 192 Kb totali, si inserisce nella porta per floppy e riporta il connettore. Usa la memoria tra 00000 e FFFFH mai usata sino a nessuna interferenza. L. 93.000

Eprom 27256 programmata con iandisk e toolkit 2 per CEPRUM L. 23.000
Eprom 27512 programmata con uno dei programmi PSION 2.30 L. 39.000

ESPANSIONE INTERNA di MEMORIA per QL tipo QLRAM19 senza saldature.
Questa espansione viene inserita sullo zoccolo della ULA e attraverso lo zoccolo prende tensione ed i segnali necessari al suo funzionamento accelera i programmi PSION del 30-40%. Montaggio alla portata di tutti: memoria totale 640 Kb. Le complete ed esaurienti spiegazioni permettono il montaggio a chiunque in pochi minuti.

Permette il multiback con 412 Kb di Ram per un totale di 640 Kb L. 329.000
Completo di eprom DUNTERMAN, cassetta e manuale L. 349.000

Superboard interfaccia ad innesto con espansione a 640 Kb, interfaccia floppy, interfaccia per stampante Centronics, toolkit 2.08 L. 449.000
Mouse per SuperBoard SANDY inglese con programma OFRAM L. 95.000

TUTTI i tipi di stampanti EPSON e PANASONIC con ottimi prezzi TELEFONARE
STAMPANTE EPSON 1760 serie L con cavo QL52 e 80 colonne L. 200.000
STAMPANTE EPSON LX 800 811 80 colonne 150 CPS NLD compatibile QL e PC FG/LD L. 1.339.000
SINGOLA e CONTINUA INTERFACCIA CENTRINIC L. 560.000
STAMPANTE PANASONIC KX-1081 80 colonne 120 cps compatibile EPSON ultima con tutti i programmi PSION anche in grafica L. 520.000

Richiedete catalogo gratuito per i prodotti Sinclair.

AMSTRAD
AMSTRAD PC 1640 con 1 floppy e monitor bianco/verde L. 999.000
AMSTRAD PC 1640 con 1 floppy e monitor COLORI L. 1.349.000
AMSTRAD PC 1640 con 1 floppy e scheda EGA 16 COLORI L. 1.339.000
SUPPLEMENTO per 3 floppy 5-1/4 360 Kb L. 250.000
SUPPLEMENTO HARD DISK 20 Mbytes al posto del FLOPPY L. 750.000
Kit Floppy da 3,5" 720 kb da montare sul Pc compatibile o su AMSTRAD PC - 1640 con DOS configurato, istruzioni L. 250.000

I FLOPPY DISK da 3,5 pollici per QL.
GDISK Floppy Disk Driver 3,5" con DRIVE CITIZEN interfaccia SPEM 1.16
Alimentatore nel contenitore metallico nero con interruttore ON/OFF.
Capacità 1640 settori pari a 720 Kbytes e 450 FILES 80 tracce 2 facce.
Manuale in italiano, disco DMAGGIO con i 4 programmi PSION 2.30 per l'ippr, il copiatore PLAGIO, DISKCOPY, ICE, TOOLKIT 2 e varie utility per convertire i programmi da MDV a FCP.

Floppy Singolo da 3,5" 720 Kb formattati interfaccia SPEM L. 499.000
Doppio floppy da 3,5" 720 Kb formattati interfaccia SPEM L. 799.000
Nuovo floppy economico 3,5" 720 Kb con interfaccia SPEM L. 399.000

Interfaccia Floppy SPEM o un'estensione dei comandi residenti e alta velocità trasferimento DATI 15 Kb al secondo. Emulazione del microdrive.
Completo disco PSION, Plagio, Utility e manuale in italiano L. 179.000

PLOTTER GRAPHTEC MP3100 Eprom 40 cm/sec HP-QL SERIALE PARALLELA adatto al programma SPEM per disegni di CIRCUITI STAMPATI L. 1.099.000

Cavo Convertitore seriale-parallelo adatto a tutte le stampanti o Plotter con interfaccia Parallela CENTRONICS CAVO 1,5 metri L. 79.000 3 metri L. 82.000
Monitor Philips 7902 da 12" formati video con cavo QL L. 149.000
Monitor Philips 8630 da 14" Colore RGB alta risoluzione con cavo QL, usabile con videoregistratore e sintonizzatore a Archimede L. 699.000
ADATTATORE (per tipo ATARI) su QL L. 14.000
TWIN EXPANSION per collegare 2 interfacce sulla porta floppy tipo interfaccia floppy e denominazione di parole antizipate L. 79.000
Sintetizzatore vocale QL con software su MDV e altoparlante L. 199.000

DUALROM scheda con EPROM ICE e TOOLKIT 202 per passare da ICE a TOOLKIT senza reset. Completo di manuali e con RAM DISK EPROM HP Aggiornamento del FLOPPY MICRO PERIPHERAL L. 49.000
Programma MANDELBROT QUICKLY su disco 3,5 per QL 640 L. 49.000

Scheda da 4 Microdrive Line 22.000 - Scheda da 20 L. 100.000
Scheda da 10 dischi 3,5" bulk Line 35.000 per 50 pezzi caduno L. 3.000
per 100 dischi caduno Line 2.500 per 500 dischi caduno L. 2.400
Dischi 5-1/4 BULK 2 DO 360K per PC confezione da 50 pezzi L. 99.000

PROGRAMMA GCS-CAD nuova edizione PROFESSIONALE su DISCO. Traccia le pose su plotter di CIRCUITI STAMPATI sino a 8 formati. Con 16 tipi di pape, 18 spidi p-zoccolo, CORSO DI ISTRUZIONI presso la SPEM Torino.

Prezzo completo QL 640 doppio drive PLOTTER monitor colori L. 3.500.000 + IVA

Il nuovo Archimede della AFORN è disponibile con 1 Mega RAM, 512K Rom, 1 floppy 3 1/2, mouse, emulatore MS/DOS L. 2.201.000 + IVA

Spese postali comprese in ITALIA per importi superiori a 100.000 lire
SPEM di Guido Masoero Via Ponchielli, 26C - 10154 TORINO
☎ 011/856519

NUOVA SEDE Via Ansa, 86 - 10154 TORINO ☎ 011/85.79.24

Publicità per l'espansione di memoria per il QL costruita dalla ditta SPEM di Torino. [MC-Micro-computer - n. 47 - Dicembre 1985].

Al centro: ulteriori espansioni disponibili in una pubblicità del 1986 [MC-Micro-computer - n. 52 - Maggio 1986].

Sotto: Siamo a Dicembre 1987 (MC-Microcomputer n. 69) e le espansioni si sono moltiplicate. Ora la SPEM vende anche per Amstrad CPC, stampanti, plotter, programmi di CAD...

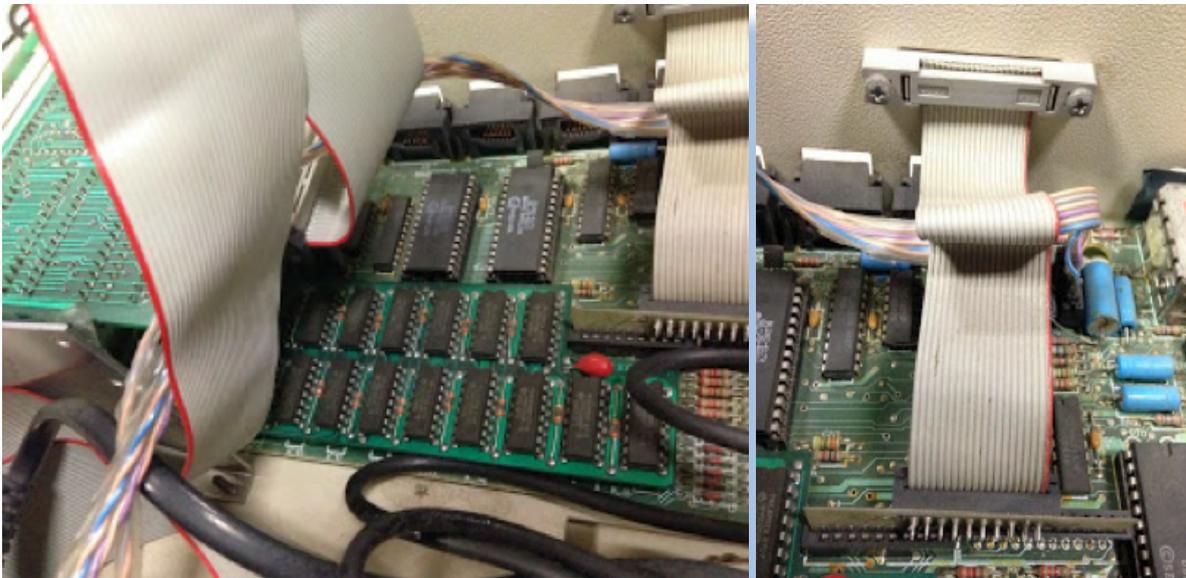
E' disponibile anche il "nuovo Archimede della AFORN..." notate il refuso.

La sede è cambiata, segno che la società si sta espandendo e che il mercato è in pieno fermento.



L'interno dello Spem è fantastico nella sua semplicità: l'alimentatore fissato sottosopra con del bi-adesivo, i microdrive sono stati letteralmente segati da un case originale, la mainboard è posta verso il fondo, inserita in un bus di espansione con la scheda controller ed un po' di ram aggiuntiva.

Notevole è la soluzione per ovviare al terribile problema della tastiera a membrana: un pettine si inserisce al posto dei flat, per raggiungere un connettore posto sul retro e a questo si connette la tastiera, sempre dotata di cavo flat.



Conclusion.

La storia dell'informatica riserva sempre delle sorprese. Quando si pensa che noi abbiamo quasi del tutto ereditato da oltreoceano, si viene prontamente smentiti dai ritrovamenti di soluzioni ideate e realizzate da appassionati per loro stessi o per risolvere problemi concreti.

Ne abbiamo già incontrate aziende piccole o piccolissime che hanno avuto il loro momento di gloria, inesorabilmente poi oscurato dalla piega che ha preso il mercato e l'industria dell'informazione in Italia e nel mondo.

(=)



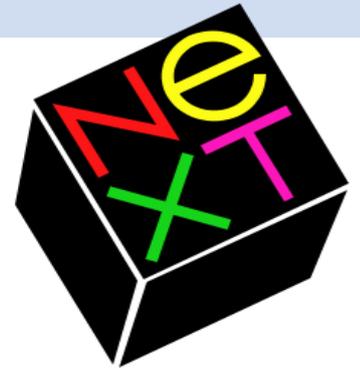
Riferimenti:

- <http://computer.blogspot.it/2013/10/spem-system2-un-sinclair-ql-italiano.html>
- http://www.rwapadventures.com/ql_wiki/index.php?title=Sinclair%20QL%20Home%20Computer&lang=en
- <http://www.digitanto.it/mc-online>
- <http://quantum.altervista.org/qitaly.htm>

Immagini:

- <http://computer.blogspot.it/>
- <http://www.nightfallcrew.com/18/04/2014/memory-expansion-spem-exp-mem-512k-for-sinclair-ql/>
- <http://www.qlforum.co.uk/viewtopic.php?f=2&t=1061>
- Le immagini di apertura dell'articolo con il QL nel cabinet di legno sono di Damiano Cavicchio.

Next Computer - Cube



di Tullio Nicolussi

Nel 1990 Steve Jobs si rivolgeva ai potenziali acquirenti del suo nuovo sistema con queste parole: *“Nel 1980 il personal computer ha rivoluzionato il modo di lavorare, ora nel 1990 il mondo ha bisogno di un nuovo computer per rendere possibile un ulteriore passo rivoluzionario. Il prossimo computer si chiama Next”*. [ndr - traduzione non letterale].

Ma vediamo cosa era successo negli anni precedenti per arrivare a questo punto della storia informatica.

Contesto storico.

Nel 1985 la Apple Computer non navigava in buonissime acque: i MacIntosh e in particolare il Lisa erano costosi e non avevano sfondato il mercato come era nelle speranze di Steve Jobs. La Apple rimaneva ancorata al suo modello //, visto che nemmeno il sistema siglato /// era riuscito per varie

ragioni a scalzare la tradizione. Ma a Jobs non stava bene: il mondo andava in una certa direzione, lui ne era sicuro, e la Apple doveva farsi trovare pronta, un passo avanti alla concorrenza.

Jobs venne estromesso da quel manager che lui stesso aveva voluto: John Sculley e mai decisione fu più sbagliata!

Uscito da Apple con un cospicuo capitale Jobs si guardò intorno e decise di puntare su una azienda che aveva un progetto sfidante: una macchina Unix con interfaccia grafica. Mancavano ancora parecchi passi per arrivare alla produzione e più di tutto mancava un manager visionario, cosa che Steve Jobs era sicuramente.

Il Next Cube viene annunciato nel 1989 ma la sua disponibilità effettiva è datata 1990. Nasce con l'intento di aprire una nuova epoca nella storia della giovane scienza informatica e le potenzialità per questo passo c'erano tutte. Purtroppo anche questa volta il prodotto, da tutti riconosciuto come ecce-

zionale, non poteva essere diffuso alla massa di utilizzatori: quelli professionali erano troppo pochi e gli hobbisti, popolo che ha fatto la fortuna di Apple, poco propensi ad un investimento tanto cospicuo.

Che non si trattasse di un personal, come lo si intendeva allora, ma di una macchina di classe decisamente superiore, lo testimonia anche il fatto che la rivista di informatica maggiormente diffusa in Italia e cioè MC microcomputer, ignora bellamente questo prodotto, perfino nella sua famosa tabella "Guida Computer" che non riporta traccia dell'azienda e dei suoi prodotti. Solo all'uscita della versione "Color" due anni dopo, MC troverà l'estro per farne una recensione, così riusciamo a scoprire che per portarsi a casa un NextStation Color con monitor e hard disk ci vogliono quindici milioni di Lire (nel 1992).

La Next Computer produsse tre modelli: il Cube (1989), la Next Station (1991) e la Next Station Color che è del 1993.

Più che un computer la Next spinge per vendere "un sistema": periferiche di input (scanner, videocamere, etc...) vengono veicolati dal sistema centrale verso l'output. Nel 1990 l'output era prevalentemente una stampa ed ecco le stampanti laser per una qualità professionale. Ma anche il multimedia comincia a farsi sentire e allora telecamere, foto, suoni,... tutti acquisiti dalle versatili interfacce della macchina ed elaborate dal software, diventano "file multimediali" che hanno come scopo organizzare, elaborare e rendere più completa l'esperienza di ricerca e condivisione della conoscenza (con la macchina viene regalata una enciclopedia con tutta l'opera di Shakespeare).

Con la versione Color il sistema ideato da Next raggiunse il massimo sviluppo ma fu anche il canto del cigno. Le cronache narrano di un business insostenibile, nonostante il vice presidente della società, Dan'l Lewin,

sostenesse fin dalla presentazione della società nel 1988, che la vendita di grandi volumi non era il loro obiettivo. Dichiarazione parzialmente corretta da Jobs che precisò che le vendite in volume non era "per ora" l'obiettivo, ma che non vedeva ragione alcuna per cui la cosa non potesse diventare reale nel futuro.

Abbandonando l'hardware, la Next Computer effettuò uno switch verso lo sviluppo del sistema operativo NextStep con supporto per altri processori diversi dal Motorola 68000 sul quale era nato. Si stimano [fonte ComputerWorld] un totale di 50.000 sistemi commercializzati nei tre modelli, prima che la Next terminasse la produzione.

Nel 1996 Apple e Next Computer si fondono (in realtà Apple compra Next per 429 milioni di dollari); Steve Jobs torna a ricoprire la carica di CEO nella compagnia che aveva fondato e NextStep diventa il basamento sul quale poggia il nuovo sistema operativo dei sistemi Apple.





Fino a qui il riassunto in poche frasi di quella che è una favola spesso romanzata e della quale forse si considerano particolari da un verso irrilevanti e dall'altro totalmente fasulli o almeno dubbi.

Non corriamo il rischio di ripetere cose strane e di dover scegliere una versione piuttosto che un'altra. Occupiamoci delle macchine.

Primo approccio.

Parleremo del Cube, il primo hardware commercializzato, visto che si tratta di un sistema innovativo che intende rompere qualche schema nel campo dell'informatica personale e professionale. Come ogni piattaforma innovativa anche il Cube non è esente da difetti, anzi! Con il senno di poi molti lo giudicano una macchina "sbagliata". Si badi bene, non perché non sia un sistema innovativo, estremamente accattivante esteticamente e con soluzioni funzionali, ma perché va a collocarsi in un settore, quello delle workstation, presidiato da aziende solide, difficilmente scalzabili nella loro posizione leader.

Il Cube si presenta esternamente come... un cubo! Beh, non poteva essere altrimenti! Il materiale con cui è costruito è il magnesio che rende leggero e contemporaneamente robusta l'unità centrale. Una curiosità: il magnesio è infiammabile per cui è possibile bruciare letteralmente il cabinet, a patto di innescare le fiamme ad una certa temperatura o riducendolo in polvere (ma non c'è pericolo che si incendi per surriscaldamento).

Due inserti "rigati" movimentano la super-



ficie del cabinet e rendono meno pesante quello che poteva essere un contenitore anonimo. Accanto all'unità centrale una ampia tastiera dalla quale esce il filo del mouse (poteva mai mancare in una macchina di Steve?) e un monitor degno di questo nome, appoggiato su una base basculante regolabile, completano il sistema. Il colore è nero-antracite per tutti gli elementi, fatta eccezione per il logo colorato, un quadrato obliquo con la scritta Next stilizzata che completa l'idea di design: una macchina subito riconoscibile che deve spiccare accanto a tante tastiere e monitor beige.

Le due griglie poste al centro della faccia anteriore del Cube nascondono due bay da 5,25" per altrettante unità. Uno contiene l'unità magneto-ottica, una delle innovazioni più eclatanti della nuova macchina di Steve, l'altro è riservato ad una eventuale espansione che potrebbe essere ragionevolmente un hard-disk.

Molto particolare il supporto del monitor che permette una regolazione in altezza ed è pure dotato di due rotelline di gomma che dovrebbero facilitare il posizionamento avanti-indietro della periferica, senza lasciare troppi segni sul piano della scrivania.

Lo stesso corpo del monitor, che richiama il design squadrato e le grigliature dell'unità centrale, è sagomato in modo che dia l'impressione di essere meno massiccio rispetto alle sue dimensioni che per l'epoca sono ragguardevoli: 17", peraltro in linea con la classe di sistemi dove si posiziona.

Minima anche la quantità di cavi, che pur ci sono, ad ingombrare il piano di lavoro. Il posizionamento delle porte di I/O lateralmente e non sul retro del cubo, contribuiscono a tenere ordinato il tutto e ad evitare di avere il classico mazzo di cavi che si aggroviglia sul retro del piano di lavoro.



Hardware.

Jobs sapeva bene che andava a proporre un sistema in una fascia di mercato presidiata verso l'alto dalle workstation Unix e dai mini-computer e verso il basso da home computer che si stavano evolvendo e confluivano nei sistemi da ufficio, fascia peraltro saldamente presidiata dalla architettura IBM.

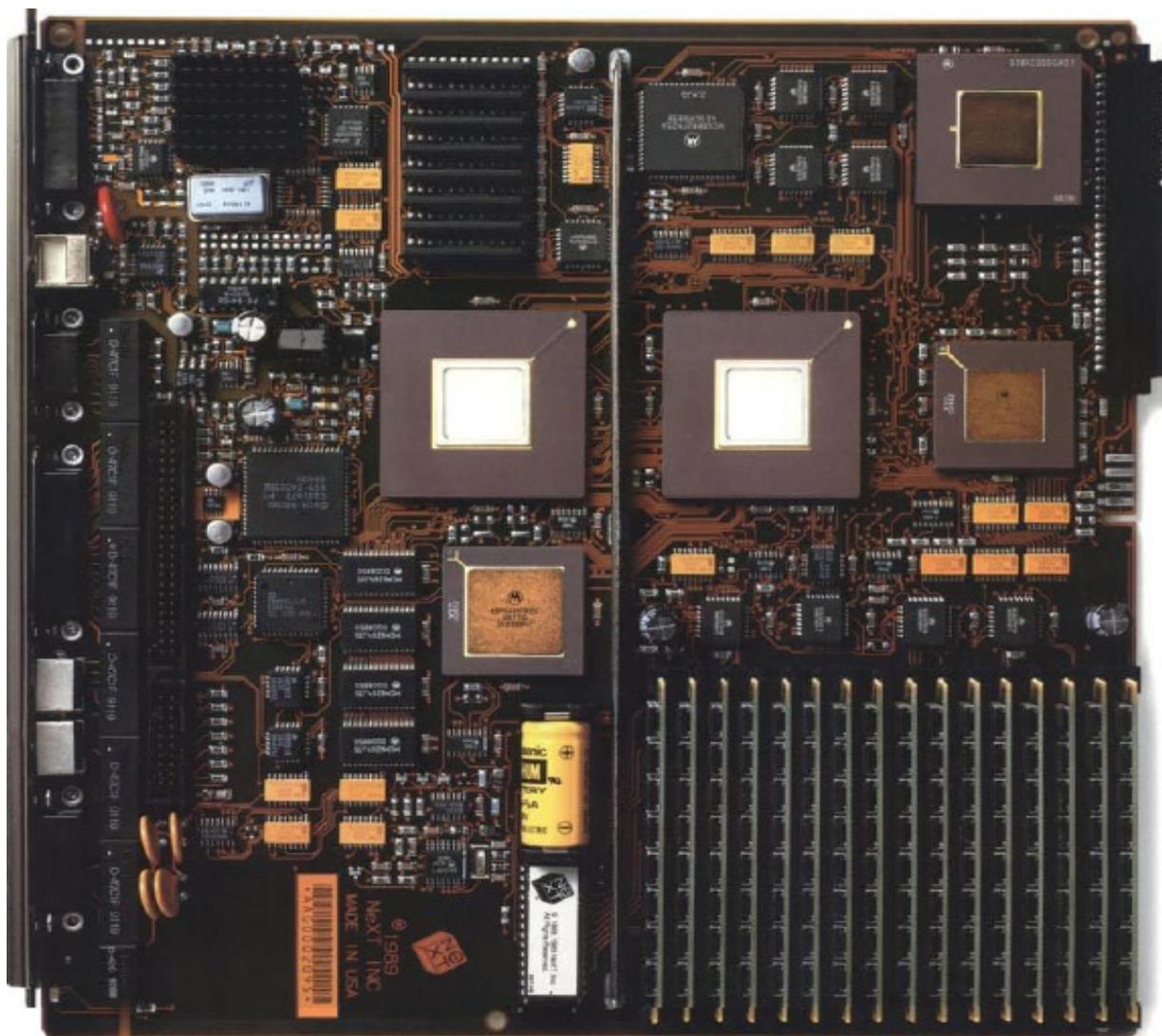
Il target fu individuato nel settore della ricerca accademica, dove macchine Unix come le workstation SUN la facevano da padrone e dove la spesa per un prodotto innovativo in grado di aiutare la ricerca, non era un problema primario.

Il suo sistema doveva far esclamare "wow" per essere desiderato e così ci mise dentro tutto il meglio che si poteva avere all'epoca in termini di tecnologia hardware e di strumenti software. Se si esclude la grafica a colori e la scelta di una unità di storage prematura, il suo intento si può dire riuscito.

Attorno ad un processore Motorola 68030 funzionante a 25 MHz fanno corona una unità aritmetica floating-point (Motorola 68882) che però è opzionale e il gestore di memoria MMU; la piastra madre ospita anche un DSP (Digital Signal Processor), 8 Mb

La piastra madre ordinatissima e razionale. Quasi un quadro d'arte...

Nella pagina a fianco: il Cube brucia...



di RAM e una serie completa di interfacce per l'I/O, porta Ethernet e porta SCSI comprese. Due "chipponi" custom si occupano rispettivamente dell'unità disco e della gestione dei 12 canali DMA. L'uso della modalità Direct Memory Access è molto usata nel progetto al punto da essere la base portante dell'intera architettura.

Il DSP (Digital Signal Processor) 56001 si occupa della componente suono oltre che di processare i segnali analogici provenienti dall'esterno attraverso l'apposita porta di I/O. Questa è anch'essa una novità rispetto alle architetture dei calcolatori personali preesistenti. Attraverso questo hardware è

possibile elaborare segnali analogici attraverso una campionatura programmabile.

La densità della componentistica è impressionante: la Next affermava che in ogni singolo chip del Cube fossero presenti fino a 10 volte i transistor presenti in un intero Mac II.

Un solo cavo di alimentazione è sufficiente per tutto il sistema che vede la presenza di connessioni "a cascata", nel senso che la tastiera si attacca dietro al monitor e il mouse alla tastiera...

Questo arrangiamento delle connessioni non è solo una questione di riduzione di cavi, ma le periferiche sono attive sullo stesso bus.

E' possibile ad esempio comandare la luminosità del monitor o il volume degli altoparlanti (che sono nel corpo del monitor stesso) operando sulla tastiera. Il sistema non prevede un interruttore di accensione, cosa che lascia interdetto chi si avvicina per la prima volta alla macchina, ma c'è un tasto sulla tastiera che fa questa funzione di accensione. Lo spegnimento avviene poi via software quando si decide di chiudere il sistema operativo.

Aprondo il cabinet dal pannello posteriore si accede alla disposizione ordinatissima della componentistica. In basso trova posto l'alimentatore, flessibile al massimo in termini di tensione e frequenza, sul fondo la piastra nu-bus full 32 bit, sulla quale si innestano le schede, una di queste è ovviamente la piastra madre che viene installata a destra e che porta in uscita i segnali di interfaccia attraverso una nutrita serie di connettori.

Il Cube arriva privo di drive per floppy, anche se terze parti e la Next stessa si sono poi affrettate ad offrire delle soluzioni esterne. La sua strategia di storage si basa sull'unità magneto-ottica e sulla presenza della porta Ethernet di tipo coassiale (Thin Ethernet, veniva definito questo connettore) in standard a 10 Mbit/sec.

Non è presente la classica ROM con i driver di base e le funzioni di inizializzazione. Una semplice PROM contiene il minimo bootstrap per caricare il software IPL dall'unità di storage.

L'unità magneto-ottica da 256 Mb di produzione Canon è sicuramente la componente del Cube che ha attirato maggiore attenzione su di sé. Nel 1990 un floppy portava un MegaByte circa di

dati e un hard-disk da 100 Mb era un lusso che pochi si potevano permettere. Disporre di una capacità così alta e per di più removibile e quindi virtualmente di una memoria di storage illimitata, è stata una delle chiavi del successo del primo computer Next.

Una cartridge costava circa 50 dollari, un prezzo non popolare ma nemmeno troppo esoso da non poter essere preso in considerazione da chi intendeva disporre di un magazzino dati di una certa dimensione. Le critiche all'unità magneto-ottica non furono rivolte al suo costo, ma piuttosto alle prestazioni dell'unità che si rivelarono non proprio all'altezza e piuttosto lontane da quelle di un hard-disk.

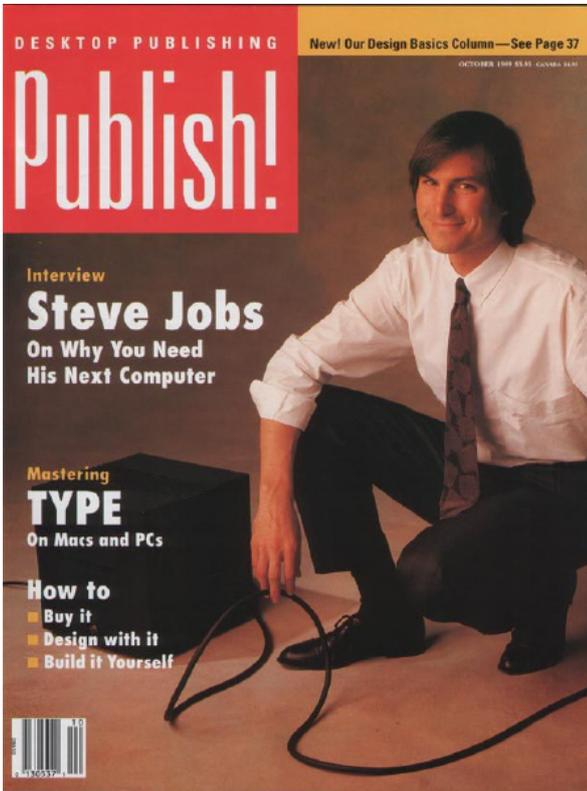
Il problema è tecnologico e determina una certa macchinosità nella scrittura dei dati, con la necessità di operare più di un passaggio: prima per cancellare un'area e poi per scriverla. Next ha fatto di tutto per rendere meno palese il gap prestazionale con le unità hard-disk, dotando il sistema di un generoso



La rivista "Publish!", specializzata in tecnologie per l'editoria annuncia una nuova opportunità per gli editori ed autori.

Sotto: la rivista Byte recensisce la macchina sul numero di Novembre del 1988.

A fianco: la postazione di lavoro completa per l'editor.



buffer di RAM. D'altra parte l'alternativa era un hard-disk di pari capacità, ma avrebbe alzato il costo del sistema di altri tremila dollari all'incirca!

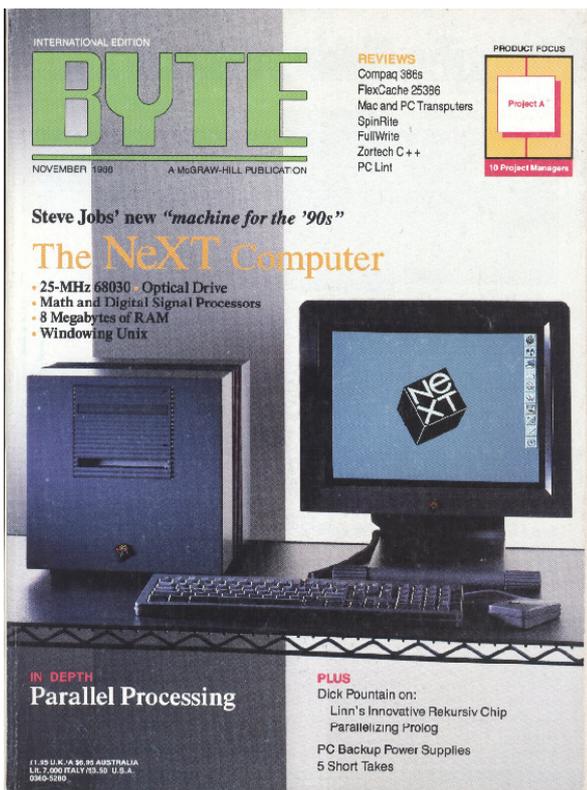
Nella seconda versione del Cube, offerta nel 1991, l'unità hard-disk era disponibile (addirittura da 340 Mb), ma il prezzo dell'unità (considerando il raddoppio della RAM, portata a 16 Mb, dell'interfaccia video migliorata e del nuovo processore MC68040) salirà oltre i 9.000 dollari.

Lo schema a blocchi dimostra un disegno molto pulito raccolto attorno ad un bus da 32 bit multiplexato (indirizzi e dati viaggiano sulle stesse linee). Questo tipo di condivisione delle linee sarà abbandonato a breve preferendo separare decisamente le due "strade", ma allora era tollerabile (induce un certo ritardo nell'esecuzione). Il clock a 25 MHz è al top di gamma nel 1990 ma non si era ancora capito come eliminare le interferenze radio fra le piste, così era necessario controllare l'afflusso dei dati/indirizzi.

Il Cube presenta nel campo dell'indirizzamento la presenza di un chip MMU/DMA di controllo per l'accesso alla memoria. Non è una novità assoluta ma induce una certa complessità nel software di base. Non per nulla i bug più difficili da fissare anche per il Lisa di Apple (altra macchina della stessa classe tecnologica, anche se più vecchia di qualche anno) che si avvaleva di una analogo soluzione.

La sezione video si avvale di 256 KByte di RAM statica connessa attraverso una tecnica che consente un doppio accesso: uno riservato alla CPU e l'altro al circuito video, indipendenti uno dall'altro.

La mappa video prevede 1120x832 pixel che è una misura assolutamente fuori standard e che va a sfruttare al massimo la risoluzione del monitor che è di 96 dot per inch (punti per pollice). Ogni pixel della mappa video è co-



dificato in due bit, il che consente di avere quattro livelli di grigio (dal bianco al nero e due intermedi). Sembrerebbe una capacità povera ma l'interfaccia grafica è in grado di visualizzare in maniera piacevole ed efficace gli elementi. L'alimentazione +/- 12 Volt, viene ricavata dall'unità centrale che provvede poi a ricavare le tensioni necessarie al funzionamento della parte logica e del circuito di pilotaggio del CRT.

Entrambe le periferiche di output, cioè display e stampante, usano il Postscript come protocollo di descrizione della pagina. Significa che le unità devono contenere una parte logica di decodifica che partendo da uno stream PostScript consenta di avere il risultato finale da mostrare all'utente. Il software DPS (Display PostScript) si pone fra il sistema operativo che elabora una descrizione della pagina in formato PDL (Page Description Language) e la periferica di output. La tecnica consente al sistema operativo di essere indipendente sia dall'hardware delle periferiche che dal gestire delle finestre video. Dovrebbe quindi favorire il porting del NextStep verso altre piattaforme, a patto che Adobe, proprietaria della tecnologia PDL, crei i necessari componenti per le piattaforme software, come ad esempio per Windows nel settore PC o X-Window per le macchine Unix.

Nell'hardware si deve includere quasi obbligatoriamente la stampante laser in B/W che lavora in PostScript. Una scelta professionale ma il suo costo, di quasi duemila dollari, la rende fuori portata dell'hobbista.

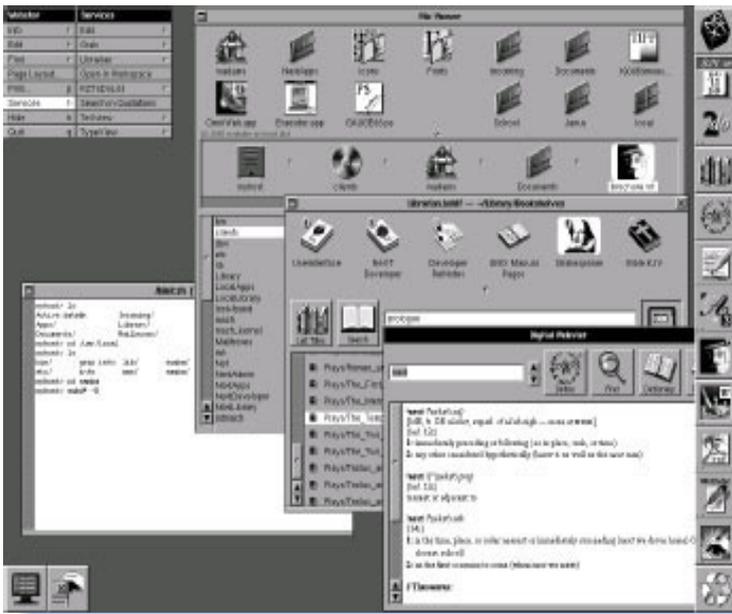


Il software.

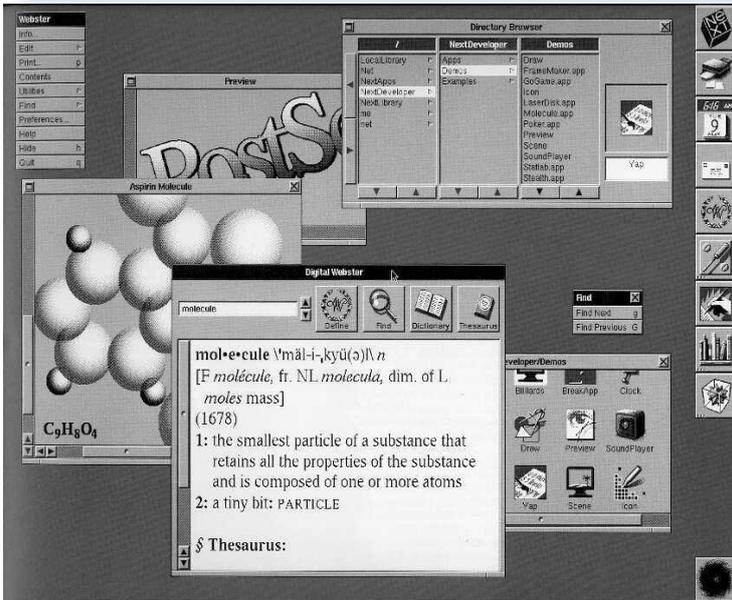
Al pari dell'hardware il software in dotazione al sistema Next è una riuscita combinazione delle migliori conoscenze allora disponibili.

Due sono le componenti principali che fanno del Cube una macchina perfetta per lo sviluppo del software: prima di tutto Unix, un sistema multitasking in time-sharing che nel Cube vede la sua implementazione basata su un micro-kernel Mach e su una interfaccia grafica a finestre, ovviamente interagibile con il mouse. Il secondo componente è l'SDK, un kit di sviluppo basato sulla programmazione ad oggetti.

NextStep come sistema operativo è uno Unix basato sul micro-kernel Mach sviluppato alla Carnegie-Mellon University. Rispetto al kernel BSD 4.3 dal quale deriva e che mantiene come compatibilità all'indietro, NextStep presenta delle particolarità che hanno lo scopo di aumentare le prestazioni e



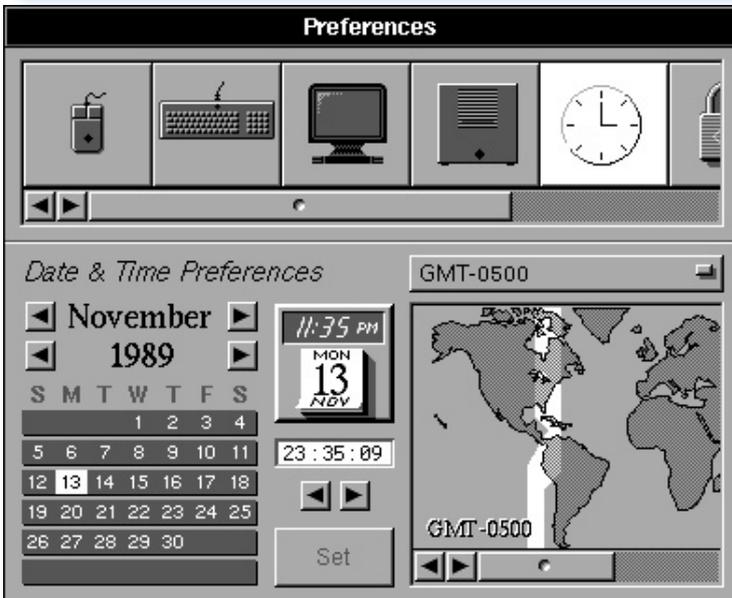
Immagini del desktop del sistema operativo nelle prime versioni; nella pagina a fianco OpenStep, porting su piattaforme diverse e a colori.



la multitasking. In particolare Mach prevede meccanismi di condivisione della memoria fra processi e una tecnologia di comunicazione fra processi IPC (Inter-process communication) e il supporto per la programmazione a threads. In pratica un task può istanziare porzioni di memoria che poi vengono condivise con i processi figli.

La parte network del sistema utilizza lo standard di SUN chiamato NSF (network File System) per l'accesso alle risorse remote e il protocollo TCP/IP per quanto riguarda il trasporto dei pacchetti.

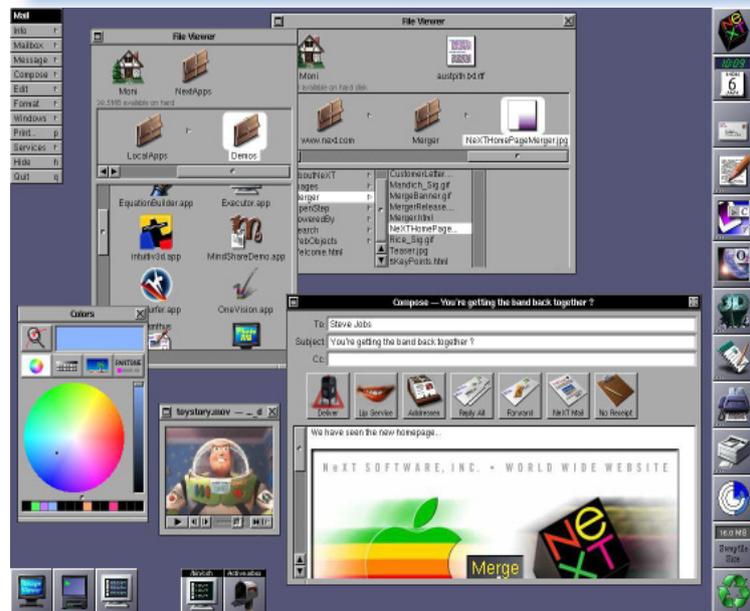
La gestione grafica del sistema si realizza attorno ad un gestore di finestre chiamato Workspace Manager che provvede alla funzionalità di base dell'interfaccia. Nella realizzazione della sua interfaccia la Next ha sfruttato le prerogative già appannaggio delle macchine Mac (e non poteva che essere così, vista la presenza di Jobs). Ad esempio è disponibile un gestore del file system molto simile al Finder di Mac con la possibilità di esplorare l'albero delle directory così come oggi è normale ma che per un operatore Unix un tempo si trattava di operare con comandi testuali.



Il Software Developer Kit è l'altra componente sulla quale Next punta molto per invogliare l'acquisto del suo sistema. Attorno ad un linguaggio ad oggetti (Objective C) e ad un compilatore standard C, vengono rese disponibili allo sviluppatore una serie di librerie per semplificare l'utilizzo dell'interfaccia grafica e degli altri elementi del sistema. È noto che proprio sul suo Next Cube Tim Berners Lee diede vita presso il CERN al primo Web Server.

Da questa esperienza nacque WebObject, un tool di programmazione che anticipa le idee legate al Web dinamico, cioè al processing delle pagine HTML sul server prima che vengano inviate al client; idea che poi Microsoft copiò (e ti pareva!) nelle specifiche ISAPI (Internet Server API).

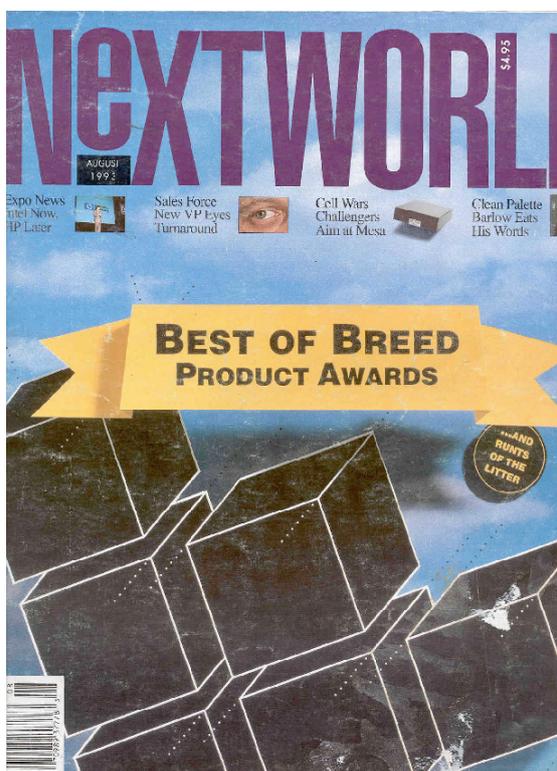
NextStep era un prodotto talmente buono che dopo il 1993, quando la Next Computer pensò di chiudere l'esperienza hardware, divenne il prodotto di punta dell'azienda. Il porting su altre piattaforme, in particolare Unix ma non solo, diede vita ad un mercato di nicchia ma abbastanza attivo. Il nome venne cambiato in OpenStep per sottolineare l'universalità del porting. Esistono versioni anche per Windows e OS/2, cioè anche



Il successo dei sistemi, pur di nicchia, crea occasioni di business anche per l'editoria: viene pubblicata una rivista specializzata NextWorld per tutti i fortunati possessori.

per piattaforme di elezione "personale". IBM, che si comprò la licenza di OpenStep per farne il sistema operativo che sostituisse OS/2, rinunciò al progetto quando Steve Jobs portò OpenStep in Apple per farlo diventare OSX. Si ignorano le cause che indussero IBM alla rinuncia; la versione più probabile è che non voleva confondersi sul mercato con i sistemi Apple...

Oltre al software di base e al SDK, Next Cube viene venduto con interessanti pacchetti: un word processor, chiamato WriteNow, il dizionario elettronico Webster e il programma Mathematica della allora non troppo famosa Wolfram Research.



Conclusione.

Molto ricercato fra i retro collezionisti per il suo design e per il significato storico, e alimentata dal numero relativamente basso di pezzi prodotti, il "cubo" rimane una specie di mantra per gli appassionati e non solo per quelli che si dedicano alla "mela" e ai suoi derivati. Pur superato in eleganza e ovviamente in funzionalità dai suoi "fratellini più piccoli", cioè dai due modelli "slim" (o pizza-box, come chiamano gli inglesi il fattore di forma piatto) commercializzati negli anni seguenti, è uno di quegli oggetti che si annoverano all'unanimità fra le pietre miliari sulla strada dell'evoluzione dell'informatica personale.

Se si va a considerare ogni aspetto si troveranno anche alcune critiche e non solo rivolte all'adozione dell'unità magneto-ottica e alla mancanza di un hard-disk, ma sulla robustezza in generale. Per qualche strana ragione che appare inspiegabile a prima vista, il Next Cube ha sofferto di una percentuale di guasti "importante". Che le macchine di Sinclair potessero avere una certa quantità di rientri era considerato se non proprio accettabile, almeno sopportabile, visto il prezzo di qualche centinaio di dollari; che la cosa capitasse con una macchina da 6.000 dollari e rotti è decisamente meno digeribile!

In generale le critiche mosse dai primi recensori sulle riviste specializzate dell'epoca, si sono rivelate fondate e alla lunga la macchina non è stata accolta come Jobs e compagnia speravano.

Peraltro sembra il destino delle macchine che si propongono come un salto generazionale rispetto alla produzione da cui provengono. E' il caso dell'Apple ///, del Lisa, del QL di Sinclair, dell'Amiga 4000, dell'Osborne one,... e se ne possono elencare molte altre!

(=)



Immagine suggestiva tratta da una brochure pubblicitaria della Next Computer. Notate come l'immagine veicola l'idea di un ampio respiro di spazio e di estremo rigore (i cavi sono stati tolti, ovviamente).

Bibliografia e reference.

"The Next Computer"; Tom Thomson, Nick baran; in *First Impressions*; *BYTE*, Novembre 1988.

">1977 - 1987 - Quando il computer divenne personal"; Paolo Cognetti; 2014; <http://www.retrocomputer.it/1977-1987/index.asp>

"Digital Retro: The Evolution and Design of the Personal Computer"; Gordon Laing; 2004; ISBN-10: 1904705391, ISBN-13: 978-1904705390; <http://www.digitalretro.co.uk>

InfoWorld 12 Aug 1991

Machines that count: the NeXT cube
This week, computer historian John Pratt looks at the NeXT Cube

John Pratt on 23 August, 2007 22:00

<http://www.prepressure.com/prepress/history/events-1988>

<http://allaboutstevejobs.com/blog/tag/next/>

Prodest?...

Di certo non giova a Olivetti

OLIVETTI PRODEST PC 128 S.
OGGI IMPARI L'INGLESE,
LA MUSICA,
L'INFORMATICA.

660.000* Lire
SISTEMA COMPLETO
CON DUE PROGRAMMI
A SCELTA.

olivetti PRODEST
HOME OFFICE COMPUTER

* MONITOR MONOCROMATO IN ECCELLENZA

di Jb72

Verso la fine degli anni Ottanta Olivetti tenta un'incursione maldestra e sostanzialmente disorganizzata nel mercato consumer, forse uno dei tanti tentativi che da lì in avanti l'azienda tenterà per immettere un po' di ossigeno nelle proprie finanze; si tratta di una linea di prodotti di bassissimo profilo introdotti attraverso una sciagurata operazione di marketing. Operazione probabilmente fondata su valutazioni del tutto distorte sulla realtà del mercato e sulle tendenze che questo aveva in seno: attraverso delle campagne pubblicitarie tutto sommato convincenti e di forte presa sul pubblico, immette una linea di prodotti denominata "Prodest".

Olivetti al vertice della propria espansione multinazionale aveva proceduto ad importanti acquisizioni in Europa, come Triumph-Adler in Germania, Thomson in Francia e

Acorn nel Regno Unito. Proprio da queste ultime due proviene il materiale per imbastire quella fallimentare impresa nel tentativo di penetrare il mercato consumer, cioè quello che fino ad allora era stato dei cosiddetti "home computers". In realtà un mercato sempre più esigente in cui pullulavano macchine dalle grandi pretese e non senza soluzioni originali e potenti: si pensi ad Amiga, la serie ST di Atari oppure gli Archimedes di Acorn con processori RISC!

La linea prodest di Olivetti si compone di tre piccoli home computer che si aggiungono progressivamente a partire dal 1987 e che sono del tutto diversi e assolutamente incompatibili tra loro nonostante l'ingannevole trovata di identificarli attraverso un nome comune. Tra l'altro, e questa la cosa decisamente più rattristante, non si tratta nemmeno di novità tecnologiche ma piuttosto

sto, di piccoli restyling di macchine già esistenti. Non solo, ma si tratta pure di piccoli computers già abbastanza diffusi e pure di discreto successo, purtroppo però, in mercati di altre nazioni! La linea prodest aveva una diffusione nazionale, proprio in un paese come l'Italia in cui la tecnologia telematica era particolarmente precaria e svantaggiata; in quel periodo in cui la diffusione di informazioni e software era estremamente limitata, non vi erano tante possibilità di fare affidamento su una base di software esistente e nemmeno la possibilità di sapere che quei computers avevano avuto un discreto successo in altre paesi europei.

Al top della linea si pose, nel 1988, il PC1: un compatibile MS-DOS di "seconda generazione" (floppy tre e mezzo e hard disk 20 Mega opzionale, un discreto processore NEC-V40, grafica multistandard CGA). Per la precisione però si tratta di un tentativo discutibile e già tentato da molti altri di proporre un personal computer MS-DOS

dentro un contenitore "all-in-one" estremamente limitante (seppure molto carino nella versione Olivetti Prodest), ma con capacità hardware ormai improponibili e superate anche per gli utenti con meno pretese, compresa la possibilità di non essere obbligati a comprare un monitor dedicato.

Prodotto intermedio della linea Prodest è il PC128S, probabilmente il migliore e più originale dei tre: introdotto nel 1987 in contemporanea alla versione del suo omologo britannico: si tratta infatti dell' Acorn BBC Compact, praticamente l'ultimissima incarnazione della serie BBC di Acorn che nel frattempo stava mettendo a punto l'Archimedes proprio perché gli 8 bit erano troppo pochi anche per questo settore non professionale. Si tratta di macchine di larga diffusione in Gran Bretagna e dalle ottime qualità, dotate in patria di un parco software di qualità e di una diffusa conoscenza da parte di utenti appassionati. Un personal però che apparteneva decisamente ad una



generazione precedente a quella in cui venne proposto in Italia da Olivetti. Proprio per le sue qualità esso poteva anche non sfigurare completamente e, con tutti i suoi notevoli limiti, attrarre l'attenzione di alcuni appassionati. Basato come i suoi predecessori sul mitico MOS6502, perfezionato soprattutto dal punto di vista del sistema operativo proprietario e di dotazioni hardware di livello (come i floppy disk da 3 pollici e mezzo) poteva vantare una grafica abbastanza importante per quella categoria di macchine (640x256x2 punti), una interfaccia grafica un po' posticcia (in realtà molto intrigante per i veri "smanettoni") e l'ultima evoluzione dell'ottimo BBC Basic. In realtà sarebbe stato un ottimo pc fino a qualche anno prima ma sicuramente non era un prodotto da dare in pasto al generico consumatore, ma bensì una macchina per veri appassionati. La stessa Acorn, proprietà di Olivetti, era passata ormai ad Archimedes abbandonando questo personal della generazione precedente e anche per Olivetti si rivelò un buco nell'acqua... seppure di classe!

Infine il piccoletto della famiglia con cui era stata presentata la linea Prodest nel 1986 era in realtà un'altra macchinetta "riciclata" da altri mercati (nello specifico quello francese) dove aveva riscosso negli anni precedenti un discreto successo. Rigorosamente del tutto incompatibile con i fratelli della serie, denominato in maniera ancora più ingannevole PC128 (senza la "S" finale ma le differenze reali sono incommensurabili): si tratta del Thomson MC5. In questo caso di un home-computer ancora più limitato dal punto di vista hardware; proposto come vero low-cost, era in realtà un "plasticone" che si avvaleva di un registratore a cassette integrato, dotato fin nella dotazione iniziale di una "penna ottica" che funzionava anche su un comune televisore, di alcune originalità nel dialetto del linguaggio Microsoft BASIC implemen-

tato che lo rendevano molto facile da programmare con diletto e infine, di una ROM da 64 kappi piuttosto ben fornita di software di configurazione. Tutte caratteristiche però che ne avevano decretato un successo in patria parecchi anni prima, ma del tutto insignificanti al momento della sua riproposizione sul mercato italiano.

Ovviamente a queste "invenzioni" si affiancò un parco software riadattato e del tutto sgangherato che si inseriva malissimo nella realtà di mercato. Il tutto fece perdere praticamente da subito ogni credibilità in questa operazione che si rivelò disastrosa anche per l'immagine di Olivetti stessa.

(=)



TurboTape: mettiamo il turbo alle cassette



di Antonio Tierno

La schermata di apertura è familiare a molti di noi, vero? Quante ore, giorni, settimane... in attesa davanti a questa schermata? E se il caricamento non andava a buon fine e bisognava ricominciare daccapo? Quando ci penso mi viene da sorridere!

Quando vidi per la prima volta, a casa di un mio amico, il floppy drive per Commodore che permetteva di caricare i videogames in pochi secondi, rimasi sbalordito. Soprattutto mi chiedevo come fosse possibile che lo stesso videogame a me richiedeva minuti e minuti di attesa per essere caricato da cassetta mentre lui dopo pochi secondi poteva già giocare!

Purtroppo ricordo anche che il floppy drive aveva un prezzo spropositato: adesso non rammento esattamente l'anno ma ricordo che mi sarebbe costato quanto il Commodore 64 stesso. Inutile dire che non riuscii a convincere i miei a comprarmelo!

Per fortuna però, poco tempo dopo, il mio

principale fornitore di software mi procurò una cassetta contenente decine e decine di giochi e tutti richiedevano soltanto una manciata di secondi per il caricamento e per caricarli era necessario utilizzare un comando diverso dal solito "LOAD" o dalla combinazione di tasti "SHIFT+RUN STOP", ovvero dovevo digitare "<-L + RETURN". E' così che venni a conoscenza di una soluzione che permetteva di velocizzare di molto i tempi di caricamento del mio amato Commodore 64: il TurboTape.

Il TurboTape appartiene alla categoria dei fast loader. Un fast loader è un software per home computer che permette di accelerare la velocità di caricamento dei file da floppy disk o supporto a nastro.

Ma come è possibile ciò? Vediamone insieme il funzionamento.

Il TurboTape è una utility creata nel 1983 dal programmatore tedesco Stephan Senz per il Commodore 64. In pratica, ottimizzando il metodo di registrazione di file su nastro, permetteva di memorizzare e legge-

Fig. 1
Tipica
schermata di
caricamento
dal Datas-
sette.

re i dati quasi alla stessa velocità del floppy drive Commodore 1541. Ciò era possibile, anche se a discapito dell'affidabilità, come vedremo tra breve.

Per la memorizzazione dei dati su nastro, molti computer utilizzano un formato analogico: ogni byte da memorizzare è diviso in bit che sono poi convertiti in toni audio. Due toni distinti rappresentano i due stati possibili di un bit (0/1).

I computer Commodore invece usano un formato digitale: piuttosto che registrare su nastro una particolare frequenza, scrivono un pattern di onde quadre (nella letteratura Commodore tecnicamente si parla di dipoli). I due poli vengono creati registrando alternativamente un certo segnale oppure un periodo (= lunghezza) uguale di assenza di segnale.

In particolare nei computer Commodore vengono usati pattern di onde quadre di 3 periodi diversi: breve, media, lunga.

Formato del loader della ROM

Ogni byte è preceduto da un marker che consiste in una onda quadra lunga seguita da una media. Un bit 0 è rappresentato con un'onda quadra breve seguita da una media, mentre un bit 1 è l'esatto opposto: un'onda media seguita da una breve. Ogni byte termina quindi con un bit di parità che può essere 0 o 1 in modo tale da rendere dispari il numero totale di bit 1 presenti nel byte.

Quindi i primi bit di un byte memorizzato su nastro potrebbero essere rappresentati come in Figura 3.

Per essere precisi, le tre onde quadre – o, se vogliamo, i tre impulsi – hanno la seguente durata:

Impulso breve (S): 348us

Impulso medio (M): 510 us

Impulso lungo (L): 672 us

Dunque per rappresentare un bit sono ne-

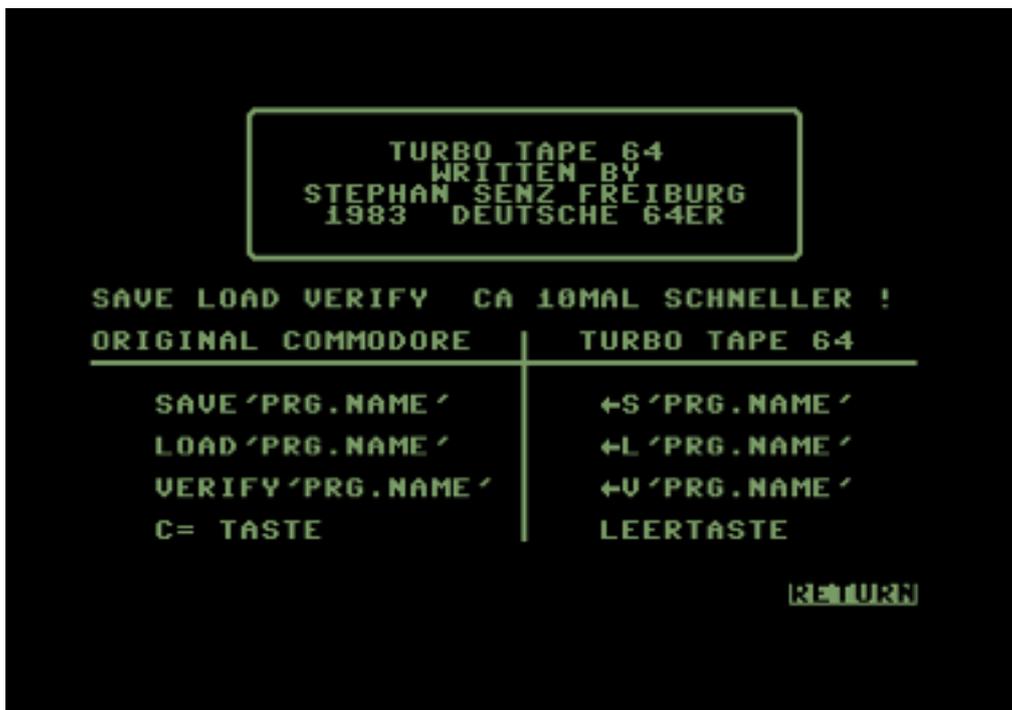


Fig. 2

Schermata iniziale del programma..

Fig. 3

Classico pattern di onde quadre per rappresentare i bit su nastro.



cessari i seguenti tempi:

$$0 \text{ bit} = S + M = 858\mu\text{s}$$

$$1 \text{ bit} = M + S = 858\mu\text{s}$$

E di conseguenza per memorizzare 1 byte abbiamo bisogno di:

$$8 \text{ data bits} = 8 * 858\mu\text{s} +$$

$$1 \text{ check bit} = 858\mu\text{s} +$$

1 data marker = $L + M$ (se nel blocco successivo ci sono nuovi dati) = 1182 μs

$$L + S \text{ (per la fine dei dati)} = 1020\mu\text{s}$$

Per un totale di 8904 μs per ogni byte!

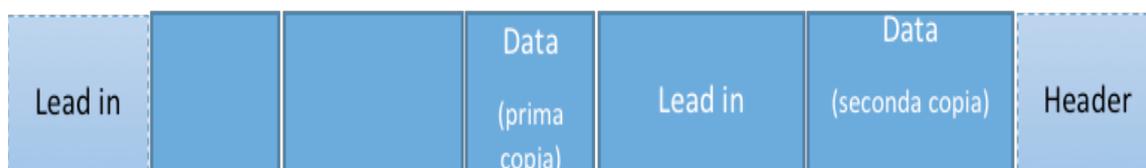
Vediamo la struttura dell'intero file:

- LEAD IN (per l'header): $6800 \times S$ = 9,4 s + 8 sync bytes
- HEADER: 4 bytes per gli indirizzi di inizio e fine + 16 bytes per il nome del file + circa 200 bytes di dati in coda
- LEAD IN (per i dati): $1500 \times S$ = 1,9 s + 8 sync bytes
- DATA: variabile

Inoltre, quando si salva un programma su nastro, il Commodore lo memorizza automaticamente due volte. In tal modo se c'è un errore nella memorizzazione (e ad esempio in un determinato byte il numero di bit 1 sommato al bit di parità non risulta dispari) il Commodore "ricorda" dove è avvenuto l'errore e lo corregge con i dati della seconda registrazione. Per questo la memorizzazione su nastro del Commodore avrà una struttura come quella mostrata in Figura 4.

Fig. 4

Struttura dei dati nella memorizzazione su nastro in un computer Commodore.



Nel caso in cui l'errore non può essere corretto o ci sono più di 30 errori nella prima memorizzazione, vedremo apparire su schermo un messaggio del genere:

?LOAD ERROR

Grazie al suo sistema di rilevamento e correzione degli errori, il formato Commodore di memorizzazione su nastro viene definito affidabile.

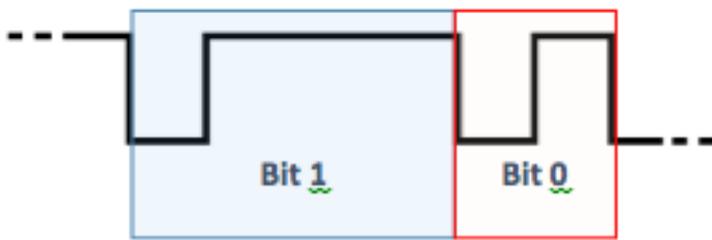


Fig. 5

Pattern dei dipoli del TurboTape.

Formato del loader TurboTape

Anche il TurboTape utilizza dei pattern di onde quadre per la memorizzazione dei dati su nastro, però utilizza un solo impulso per la rappresentazione di ogni bit.

La durata dei due impulsi differisce di quel tanto che basta per permettere alla routine di caricamento di distinguerli. TurboTape li registra su nastro allo stesso modo della routine SAVE

Fin tanto che i tasti RECORD e PLAY del registratore sono pigiati, la linea "write" (che viene dal bit 3 della porta input/output del processore 6510) controlla il segnale che viene scritto su nastro. Quando tale linea è attiva, la testina di scrittura del Datasette genera un pattern magnetico su nastro. Quando la linea viene disattivata, rimane su nastro solo un'area vuota (blank).

Il dipolo del TurboTape inizia alla transizione da 5 Volts (ON) a 0 Volts (OFF) di tale linea "write" della cassetta. Nel cosiddetto TurboSave, l'onda quadra ha sempre la stessa lunghezza, che si tratti di registrare un bit 0 oppure 1 (in realtà non sono delle vere e proprie onde quadre). I bit vengono distinti in base alla lunghezza del segnale di 5V seguente. Un segnale di 5V più breve indica un bit 0, uno più lungo indica un bit 1 (vedi figura 5).

Quindi dopo il primo segnale di 5V, il primo periodo di silenzio è costante, dopodiché

la linea "write" viene di nuovo attivata e la durata del segnale di write determina il valore del bit da memorizzare. Per riassumere, possiamo dire che per il TurboTape la durata del bit 1 è circa tre volte quella di un bit 0.

0 bit: Impulso breve (S): 211us

1 bit: Impulso lungo (L): 324us

Dunque la lunghezza di 1 byte è variabile in quanto dipende da quanti 0 e da quanti 1 contiene.

$\text{min} = \$00 = 00000000 = 8 \times 211\text{us} = 1688\text{us}$

$\text{MAX} = \$FF = 11111111 = 8 \times 324\text{us} = 2592\text{us}$

$\text{Average} = 01010101 = (4 \times 211 + 4 \times 324)\text{us} = 2140\text{us}$

Vediamo anche in questo caso la struttura dell'intero file:

- LEAD IN (per l'header): $(256 \times 4 + 247) \times \$02 = 2,3\text{ s} + 8\text{ sync bytes}$
- HEADER: 4 bytes per gli indirizzi di inizio e fine + 16 bytes per il nome del file + qualche byte di dati in coda
- LEAD IN (per i dati): $(256 + 247) \times \$02 = 0,9\text{ s} + 8\text{ sync bytes}$
- DATA: variabile
- CHECKSUM

Dunque la memorizzazione su nastro del TurboTape avrà una struttura come quella mostrata in Figura 6.

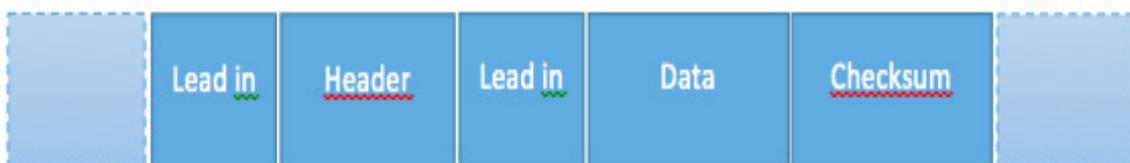


Fig. 6

Struttura dei dati nella memorizzazione di TurboTape.

TurboSave: scrittura su nastro

Il formato del TurboTape è il metodo di memorizzazione di dati su nastro più compatto possibile. Però si comprende facilmente che senza rilevazione e correzione degli errori non sia un metodo molto affidabile. Basta infatti perdere un bit nella fase di LOAD per rovinare tutto il caricamento. Per ovviare a ciò il TurboTape precede i dati memorizzati su nastro con una serie di bit di sincronizzazione: un byte di valore 2 ripetuto 256 volte, a cui segue un countdown di 9, 8, 7, 6, 5, 4, 3, 2, 1. Durante il caricamento TurboTape va a cercare questi bytes, sapendo che c'è una lunga serie di byte posti a 2 e che prima o poi troverà il countdown. In sostanza ha 256 possibilità di sincronizzarsi, quindi parte il countdown e alla fine è a conoscenza di quando iniziano i dati veri e propri. In questo caso la sincronizzazione può fallire solo se la sequenza di countdown è rovinata, il che "disorienta" il TurboTape.

Seguendo quindi la cosiddetta "synchronization leader", la routine di TurboSave scrive gli indirizzi di inizio e di fine del programma. Sono memorizzati nei primi 4 bytes dei dati salvati. Dopo memorizza i dati per poi aggiungere una checksum. Raggiunto l'indirizzo di fine programma la checksum viene scritta come ultimo byte.

Inoltre ogni programma salvato con TurboTape viene preceduto da un programma

di bootstrap memorizzato utilizzando il formato normale della routine SAVE. Questo bootstrap contiene il loader in modo tale che non sia necessario utilizzare il TurboTape per caricare un programma salvato con TurboTape.

Ma come fa un normale comando LOAD a diventare un TurboLOAD?

La porzione di bootstrap è abbastanza contenuta (10 bytes nel Commodore64). I dati vengono salvati in un formato non rilocabile, quindi iniziano sempre alla locazione di memoria 812 (\$032C). In questo modo ogni volta che viene caricata quella locazione il normale LOAD viene automaticamente convertito in TurboLOAD.

Quando viene eseguita una standard LOAD, uno degli ultimi step che il C64 effettua consiste nel richiamare una subroutine chiamata CLALL (Close ALL files) contenuta in ROM. La CLALL passa attraverso un vettore agli indirizzi 812-813 (\$32C-32D), ma tali indirizzi saranno stati cambiati dal programma bootstrap in modo che il flusso di esecuzione venga passato all'inizio della routine TurboLoad all'indirizzo 814 (\$32E). In ogni caso i byte che iniziano a tale indirizzo non sono abbastanza per decifrare i dati contenuti nel nastro e infatti la maggior parte della routine è contenuta nel buffer della cassetta.

Come questa arrivi lì è un'altra storia interessante.

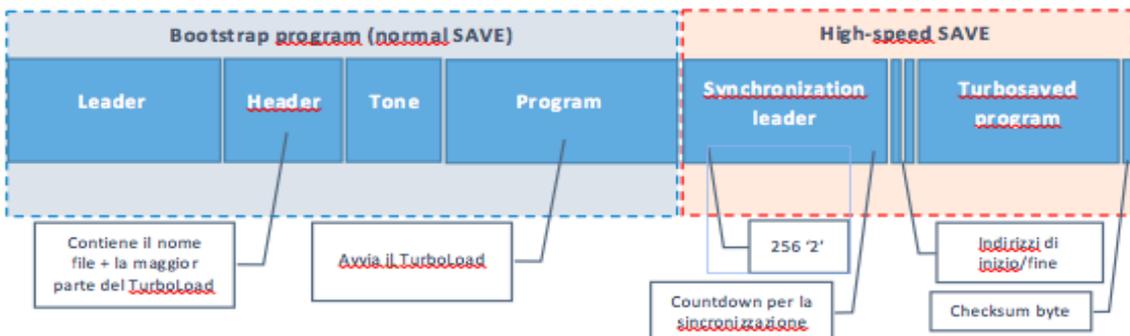


Fig. 7

Layout completo del salvataggio di un programma con TurboSave.

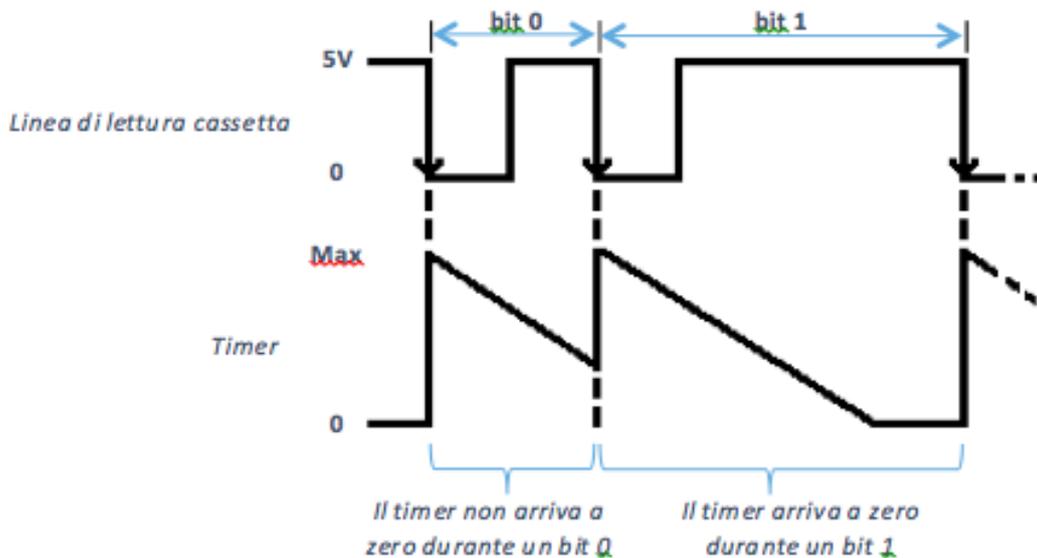


Fig. 8

Procedimen-
to di lettura
del bit con
TurboTape.

Ogni programma memorizzato su nastro ha un nome lungo 187 caratteri. E ogni programma scritto su nastro dalla routine SAVE "normale" è preceduta da un header di 192 caratteri; tale lunghezza corrisponde ai 192 bytes del buffer della cassetta (locazioni di memori 828-1019). I primi 5 bytes dell'header sono utilizzati per: un identificatore (1 byte), un indirizzo di partenza per il programma memorizzato (2 bytes) e un indirizzo di fine (2 bytes). I rimanenti 187 bytes sono disponibili per il nome del file, sebbene di solito vengano utilizzati solo i primi 16.

La routine TurboSave sfrutta questo spazio che di solito è sempre disponibile riempiendo le locazioni successive al sedicesimo byte del nome file (cioè a partire dalla locazione 849) con il resto del linguaggio macchina del TurboLoad, dove è scritto come parte del nome del file durante il processo di caricamento. Quando il nome viene rilevato durante il processo di LOAD, tutti i dati dell'intestazione vengono caricati nel buffer cassetta.

Quindi i pochi bytes salvati regolarmente servono solo a trasferire il controllo alla restante parte della routine contenuta nel buffer.

TurboLoad: lettura da nastro

Per leggere un bit, il TurboTape sfrutta le peculiarità dei chip di interfaccia periferica: il CIA (Complex Interface Adapter) sul C64, oppure il VIA (Versatile Interface Adapter) sul VIC20. Ciascuno di questi chip ha una linea (chiamata FLAG sul CIA e CAI sul VIA) che riesce a rilevare la transizione del segnale da HIGH a LOW, ovvero l'inizio di un dipolo, che viene usata per la linea di lettura del nastro da Datassette.

Per rilevare l'inizio di un dipolo, la routine TurboLoad va a monitorare il bit 4 della locazione di memoria 56333 (\$DCOD) sul C64, oppure il bit 1 della locazione 37165 (\$912D) sul VIC20. Tali bit saranno impostati a 1 quando il segnale che viene letto da nastro passa da 5 volts a 0 volts (fronte di discesa del dipolo).

Per determinare se un bit che viene letto è uno 0 o un 1, la routine TurboLoad fa partire un timer quando viene rilevato l'inizio del dipolo. Ciascun chip di interfaccia possiede 2 timer a 16 bit. Sul C64 viene impiegato il Timer 2 del CIA 2, la versione per VIC20 impiega il Timer 1 del VIA 1.

I timer sono fissati per il tempo desiderato e lasciati andare fino alla scadenza del

tempo (fino a quando non arrivano a 0). Lo schema è quello di impostare i timer per un periodo che è più lungo della durata di un dipolo bit 0, ma inferiore alla durata del dipolo bit 1. Poi, quando viene rilevato il fronte di discesa successivo, lo stato del timer viene controllato. Se il timer arriva a 0 prima dell'inizio del dipolo successivo, allora il tempo per il bit letto era maggiore rispetto al conteggio del timer e quindi era un 1 bit. Se il timer sta ancora contando quando il dipolo successivo inizia, allora la lunghezza del dipolo che doveva essere letto era più corto del conteggio del timer specificato, e quindi era un 0 bit.

Lo stato del timer può essere determinato andando a controllare il bit 1 della locazione 56589 (\$DDOD) sul C64, oppure il bit 6 della locazione 37149 (\$911D) sul VIC20. Questi saranno 0 se i timer stanno ancora contando, oppure 1 se i contatori hanno già raggiunto il valore 0 che corrisponde al valore letto da nastro. Il procedimento è illustrato in Figura 8.

La routine TurboVerify opera andando a leggere da nastro il programma di bootstrap per il programma da verificare salvato con TurboSave e poi modificando parte del codice della routine TurboLoad. Sovrascrive infatti un'istruzione di memorizzazione con un'istruzione di "compare and branch". Così, quando la routine TurboLoad subentra, i dati letti dal nastro vengono confrontati solo con i dati già in memoria, invece di essere caricati sui dati esistenti.

Esempio

Supponiamo di voler memorizzare un programma che occupa circa 10 Kbyte. Riassumiamo nella tabella seguente la situazione, tenendo conto di:

- Numero di bit utilizzati nel format dello schema di memorizzazione
- Controllo di parità
- Struttura dei bytes
- Numero di copie dei byte dati memorizzati
- Durata di bit e byte
- Regione di sincronizzazione e sequenza di countdown
- Il listato del programma in linguaggio assembly ha una struttura fissa.

Si può vedere chiaramente come il TurboTape velocizzi - e non di poco - la memorizzazione e la lettura da nastro.

Fig. 9
Tabella di comparazione fra il metodo nativo C64 e il TurboTape.

Metodo	Bit per la rappresentazione	Bit di parità	Struttura dei byte (in bit)	Copie presenti	Durata del byte (ms)	Byte per secondo	Tempo per 10 kb
ROM loader	3	si	20	2	20,4	49	7'
Turbotape	2	no	8	1	5,44	183	55"

Il prezzo da pagare

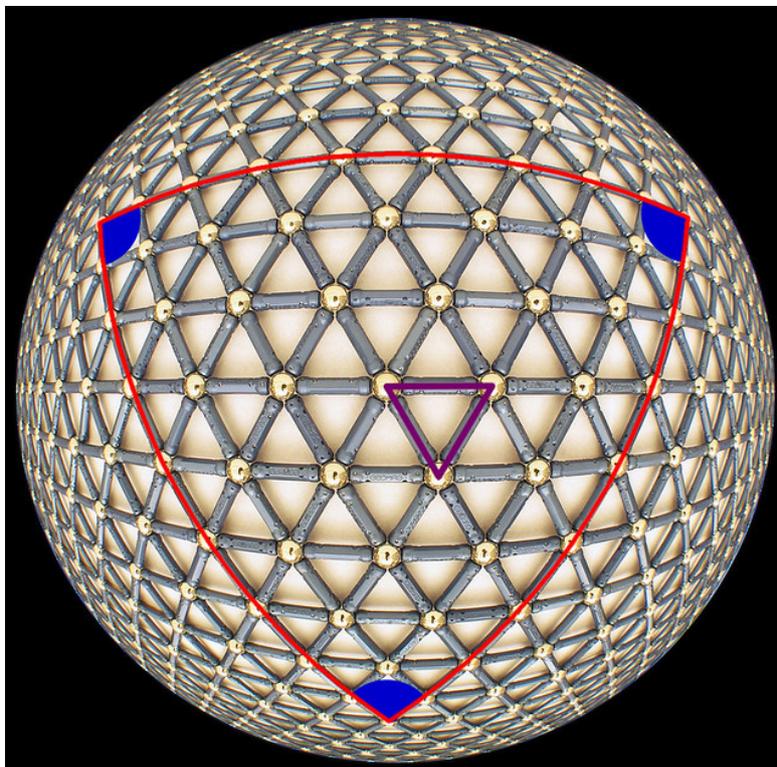
Come detto in precedenza, la checksum finale è l'unico controllo fatto sui dati memorizzati col formato TurboTape. E non è detto che anche se la checksum sia corretta non ci sia un problema nel caricamento, visto che c'è più di un modo per ottenere una certa somma.

Insomma, l'incremento di velocità del TurboTape non avviene senza un prezzo da pagare. E tale prezzo da pagare è l'affidabilità.

Inoltre le routines di lettura presenti nella porzione di bootstrap sono specifiche al chip CIA per il C64 e al VIA per il VIC20, visto che chip differenti devono essere acceduti attraverso differenti locazioni di memoria. Non c'è abbastanza spazio nel buffer della cassetta per utilizzare un TurboTape universale che funzioni su tutte le macchine. Questo si traduce col fatto che programmi salvati con TurboTape su un VIC20 non funzioneranno su un Commodore64 e viceversa.

(=)

L'indecidibilità



di Salvatore Macomer

Incalcolatori sono stati ideati e costruiti dai matematici e in particolare da coloro che approfondirono quella branca della matematica che si chiama oggi giorno Logica Matematica, appunto.

Ok, è più o meno una banalità; lo sanno tutti: Babbage, Pascal, Boole, Turing,.. tutti logici-matematici e tutti a sognare l'automazione del calcolo, ma non solo; pensavano anche all'automazione del pensiero umano, a cominciare dalla rappresentazione del ragionamento e poi via via complicando, con la speranza di scoprire la pietra filosofale dell'era moderna: l'Intelligenza Artificiale.

Ma se il ragionamento umano, attraverso la sua massima rappresentazione nota fin dagli antichi filosofi greci, cioè il linguaggio, contiene delle trappole (i paradossi), anche la trasposizione della logica nell'am-

bito del calcolo automatico si porta dietro questa tara.

Il più famoso è il paradosso del mentitore. La frase: "Io sto mentendo" ha arrovellato il cervello delle migliori menti nella ricerca di una inquadratura formale. La frase è vera o falsa? Questo è il dilemma! Infatti se è vera significa che sto mentendo e quindi sto affermando il falso, d'altra parte se è falsa allora sto dicendo il vero e quindi entro in contraddizione. Questa costruzione è apparsa come un'eresia belle e buona ai vecchi padri della filosofia: contraddiceva i principi del grande Archimede che avevano stabilito due principi fondamentali: la non contraddizione e il terzo escluso. Il primo afferma che se una cosa (frase) è vera, allora non può essere falsa; il secondo esclude che ci sia una terza proprietà oltre al vero e al falso.

Purtroppo nel linguaggio naturale (quel-

lo che usiamo tutti i giorni per descrivere il mondo) esistono tre tipi di frasi che si possono classificare secondo il loro valore di verità: quelle vere, quelle false e quelle, come il paradosso del mentitore, che non sono né vere né false.

Si dice allora che il linguaggio naturale è indecidibile, cioè contiene delle frasi per le quali non si può decidere se siano vere o meno. Nella matematica al posto delle frasi ci sono le “frasi matematiche”, cioè i teoremi. Quando parliamo di teoremi non ci riferiamo solo ai famosi teoremi di geometria che tutti abbiamo studiato al liceo, tipo il Teorema di Pitagora; piuttosto ci si riferisce a tutte le formule matematiche che possiamo scrivere con le regole del linguaggio matematico.

Ad esempio $A^2 + B^2 = C^2$ che poi è il Teorema di Pitagora. Questa formula fa parte dei teoremi dimostrabili, cioè si può sapere che è vera per una certa combinazione dei numeri interi A , B e C e si possono trovare tutte le combinazioni possibili.

Non è stato così per lungo tempo per un'altra serie di formule, ad esempio per formula:

$$A^3 + B^3 = C^3$$

Che è conosciuto come “Ultimo Teorema di Fermat” perché postulato dal matematico Fermat il quale lo congetturò nel 1637 dicendo che l'equazione $A^n + B^n = C^n$ non ha soluzioni intere se $n > 2$. Per secoli i matematici hanno provato a dimostrare questa affermazione, visto che Fermat diceva di averlo fatto ma di non avere spazio dove scriverla sul margine del libro che stava computando. Poiché alla dimostrazione non approdava nessuno, che si pensò, con buona ragione, che in realtà Fermat avesse preso un abbaglio e che la sua presunta dimostrazione sia stata errata. Comunque ora il teorema è stato dimostrato ed è vero, anche se la dimostrazione, oltremodo arzi-

gogolata, lascia perplessi più di un matematico moderno.

Di queste formule “strane” ne esistono altre, al punto che possiamo senz'altro affermare che nella matematica esistono frasi (teoremi) per i quali non si può dire se sono veri o sono falsi. La matematica è perciò un sistema formale non completo o in altre parole è indecidibile.

A questa definitiva conclusione è giunto il logico Gödel che ha dimostrato, nei primi del novecento che tutti i sistemi formali del secondo ordine (in pratica quelli generali, che comprendono tutti gli elementi di un dato contesto) sono indecidibile, proprio come la matematica.

Non spaventiamoci però: mica tutto è indecidibile. Ad esempio certi sottoinsiemi della matematica sono completi, cioè si può decidere senza ambiguità per tutti i suoi elementi se sono veri o falsi. Uno di questi è la Geometria Euclidea, quella presentata nei suo famoso tomo “Elementi” e che tutti abbiamo studiato al liceo. Un'altro, più vicino a noi informatici è l'Algebra di Boole. Infatti non esiste una “Tavola di Verità” per le variabili logiche dell'algebra booleana per la quale non si possa dedurre la verità o la confutazione. Magari i calcoli possono essere lunghi, ma si possono fare, o almeno dimostrare che se anche la tavola contiene infiniti elementi è vera o falsa.

Bene. Fino a qui, sperando di non avere stravolto nessuno, una brevissima e se volete anche approssimata, desamina delle idee che riguardano i sistemi formali e le loro regole.

Prendiamo in considerazione adesso un insieme di linguaggi non liberi da contesto, cioè le cui parole e combinazioni di parole hanno un significato non ambiguo. I linguaggi di programmazione dei calcolatori. Non ha importanza se scegliamo il BASIC

piuttosto che il COBOL o Java o altro che desiderate; con ognuno di essi si possono scrivere programmi, cioè descrivere algoritmi che risolvano determinati problemi. Attenzione che non ci interessa se un linguaggio sia più o meno potente, più o meno astratto piuttosto che vicino o lontano dall'Assembly; quello che ci interessa è che sia sufficientemente ricco da implementare in qualche modo un programma per elaboratore.

Per capirci, non stiamo parlando di linguaggi poveri di istruzioni con i quali si possano scrivere solo programmi elementari. Ad esempio: supponiamo di aver inventato un nostro linguaggio di programmazione che contiene un solo statement: PRINT con il quale si possa scrivere sul video una frase fissa.

Ecco un programma di tre istruzioni scritto con questo linguaggio:

```
PRINT "X"  
PRINT "HELLO WORLD"  
PRINT "CIAO MARTINA!"
```

Si capisce benissimo cosa fa questo programma e quando esso termina. Non ci sono IF, non ci sono LOOP, o GOTO o FOR... non c'è nulla di nulla. Linguaggio poverissimo ma "completo"! Cioè non si possono scrivere programmi con questo linguaggio per i quali dato un input non si possa dedurre l'output e non si possa sapere se ad un certo punto si ferma oppure no.

L'insieme di tutti i programmi scritti con questo linguaggio è completo!

Ok, ma un linguaggio più generalista, cioè un linguaggio utile all'analisi di algoritmi matematici nella più ampia accezione del termine, può generare insieme non completi?

La risposta non è:- "Sì, è possibile", la risposta è:- "**Sì, è inevitabile**".

Cioè come conseguenza del teorema di

Gödel, nell'insieme di tutti i programmi per calcolatore, scritti con qualsiasi linguaggio sufficientemente ricco, esistono dei programmi per i quali non è possibile dato un input sapere quale sia o se ci sarà un output.

Sconvolti? Niente panico: i programmi per il calcolo della pensione o della busta paga producono output e l'output è corretto (nei limiti dell'errore umano) :-)

Bene, ora che abbiamo buttato l'amo vogliamo prenderlo un pesce! Andiamo a dimostrare che nell'insieme dei programmi esiste almeno un programma indecidibile.

Suspance...

Trovare un simile programma può non essere affatto facile. Diciamo che dovremmo spremere la zucca provando a scrivere il nostro codice e meravigliandoci del suo comportamento imprevedibile qualora si verifici l'anomalia che andiamo cercando.

Dobbiamo subito eliminare un fraintendimento che ho rilevato spesso. Quando si dice: -"Costruisci un programma che dato un input non si possa dedurre l'output" molti trovano la possibile soluzione nei numeri casuali. Se il mio programma si limita a stampare un numero casuale, allora effettivamente non potrò prevedere quale è il suo output. Questo però non è vero nel contesto che stiamo trattando, nel senso che la casualità nei computer è solo apparente: il generatore di numeri casuali è solamente un pseudo-generatore, cioè sembra che faccia proprio quello, ma in realtà pesca da un bacino di numeri grande quanto si vuole ma ripetibile.

Un programma costituito così:

```
: INIZIO  
S = INPUT_NUMERO_INIZIALE  
: RIPETI  
N = GENERA_NUMERO_CASUALE(S)  
SE N = 1
```

```

ALLORA VAI A :TERMINA
ALTRIMENTI VAI A :RIPETI
:TERMINA `fine programma
    
```

Questo programma si ferma ad un certo punto, oppure no?

Dal momento che la funzione `GENERA_NUMERO_CASUALE()` non fa veramente quello che dice il suo nome, allora possiamo dire che sì, dato come input il “seme” nella variabile `S`, dopo un certo numero di passi il numero casuale diventa 1 e il programma si ferma. Se il numero fosse veramente casuale invece non si potrebbe dire questa cosa perché è ben vero che la statistica ci dice che prima o poi l'1 esce, ma potrebbe anche uscire all'infinito o comunque non saremo in grado di dire in quanti passi il programma si ferma.

Questo programma (nel caso di veri valori casuali) è un esempio di elemento dell'insieme che dimostra l'incompletezza del sistema formale rappresentato da tutte le combinazioni di frasi generabili dal linguaggio

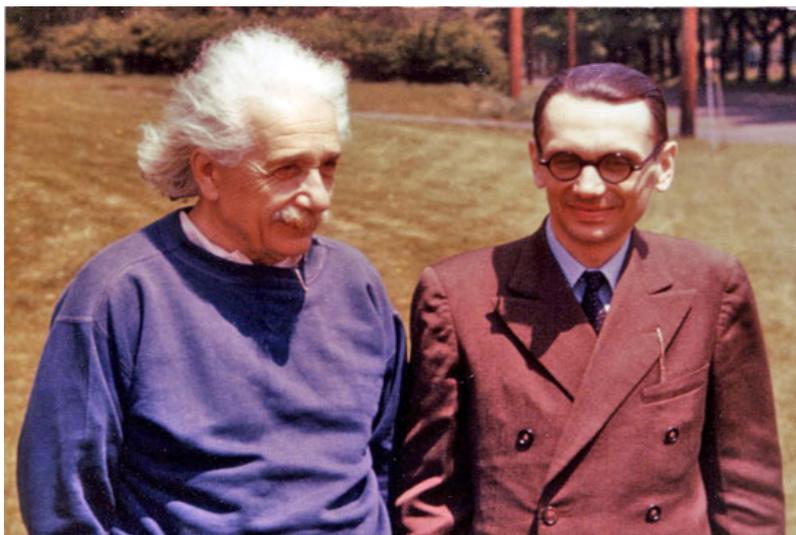
di programmazione.

Ma c'è un altro esempio più “intrigante” e più divertente (divertente per i logici-matematici si intende), oltre che essere del tutto generale. Pensate a cosa succede nel linguaggio naturale: i paradossi sono i teoremi indimostrabili. Esistono paradossi nell'insieme dei programmi? Sì, esistono!

Suspance n. 2...

L'argomento si fa un tantino più ostico e pertanto lo affrontiamo con calma nella prossima puntata. Intanto, se avete voglia, riflettete sul fatto che i paradossi nel linguaggio nascono quando il linguaggio pretende di essere interprete di se stesso, cioè in una frase vogliamo parlare della stessa frase.

(...continua...)



Due grandissimi scienziati (Albert Einstein e Kurt Gödel) ritratti assieme a Princeton. Una foto ad alta concentrazione neuronale, nulla da dire!

I vent'anni di Windows 95



di Sonicher

Chi non ricorda le “finestre volanti”?

Era l'anno 1995 e vedevamo per la prima volta lo screen saver di default del “mirabolante” nuovo sistema operativo per PC: Windows 95, appunto!

Con il senno di poi si capisce il messaggio che la casa di software, allora ancora saldamente diretta da Bill Gates, aveva voluto veicolare: “state usando un sistema che fa volare i vostri programmi!”.

Certo, con il senno di poi si possono elencare mille difetti di questo pseudo-sistema operativo. Non ultima la considerazione, un po' eccessiva ma non molto distante dalla verità, che si tratta di una interfaccia grafica con il paradigma della scrivania virtuale, del mai morto DOS e delle idee realizzate in precedenza con Windows 3.11.

Già, la scrivania virtuale: vera icona della modernità! E' strano come le idee innovative che poi si impongono e rimangono

uguali a se stesse nel corso di anni e anni, nascano e si appoggino sulla concretezza della percezione collettiva.

Pensiamo ad altri oggetti più recenti: la messaggistica istantanea, la fotografia digitale, i social network,... se ne possono elencare parecchi di questi bisogni e comportamenti declinati in chiave moderna. Con tutte le conseguenze positive e negative del caso, ovviamente.

In fondo la scrivania cos'è? Semplicemente un piano dove appoggiare fogli e strumenti di lavoro e il computer, sostituendo gli oggetti concreti con i corrispondenti virtuali, doveva ad un certo punto assomigliare alla nostra scrivania!

L'idea non era nemmeno così originale: ci avevano provato IBM, HP, Novell e forse altri in passato se vogliamo annoverare fra di essi anche GEM e i suoi cloni. Questi avevano implementato quello che prima si

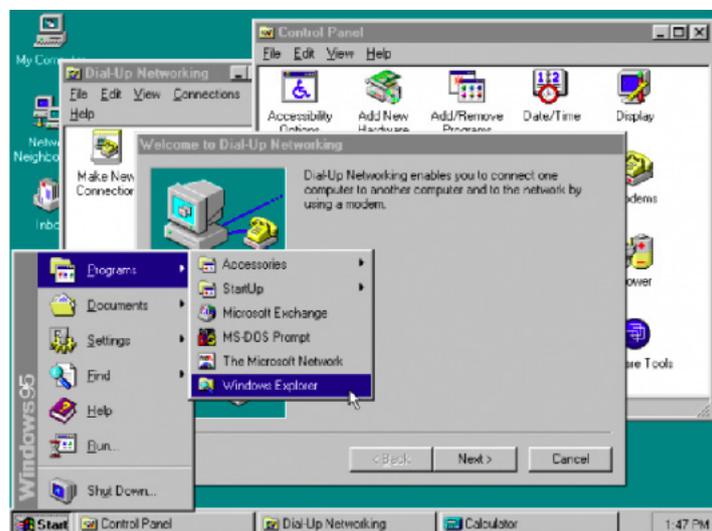
chiamava "workspace". Notate come si sia partiti da lontano prima di realizzare che quella sul computer era una vera e propria scrivania, seppur virtualizzata.

Con Windows 95 nasce anche il famigerato bottone START, quello che nemmeno Microsoft è più riuscita a far morire. Da fastidio alla real casa di Redmond quella presenza, forse perché è la porta verso una gestione dei programmi che sa di antico, di un pseudo DOS che non vuole morire...

Eppure, tolto il paradigma della scrivania, idea che ha fatto innamorare molti del computer, questo Win95 è così simile al precedente Win 3.11 da risultare addirittura imbarazzante. Le finestre sono uguali, fin dai font e dai particolari grafici. La struttura del file system è rimasta la stessa e così la gestione delle risorse macchina, cioè del Pannello di Controllo.

Windows 95 è stato solo un ponte verso il ben più corazzato Windows 98 che ha resistito poi a tutti i tentativi di de-tronizzazione, riuscita solo con l'avvento del compianto e mai dimenticato Windows XP.

Windows 95 ha goduto di una massiccia campagna pubblicitaria da parte di Microsoft con due testimonial di eccezione: Jennifer Aniston e Matthew Perry, due protagonisti della famosa serie TV "Friends", trasmessa con successo anche in Italia. Lei soprattutto era popolarissima e anche tutt'oggi fa la parte della reginetta sulle



riviste di gossip d'oltreoceano!

A Windows 95 va sicuramente riconosciuto il merito di aver sdoganato l'utilizzo del PC come macchina home e non solo come sistema da ufficio. Nessuno l'ha rimpianto quando tre anni più tardi è uscito il 98, ma è comunque parte della nostra storia.

(=)





Ci sono computer che sono diventati “cult” senza nemmeno essere stati venduti, oppure venduti in pochi esemplari. Ma se per quelli poco diffusi si può capire la rarità e quindi il loro valore per il collezionista, per le macchine progettate solo sulla carta o esistite solo come prototipo, la cosa è più sottile. Si possono considerare calcolatori a tutti gli effetti, anche se non hanno mai calcolato nulla o quasi?

Quel simpaticone di Clive Sinclair, guardate cosa voleva combinare...