

La nascita dell'Informatica

L. Morelli - Articoli riadattati per scopi didattici da R. Gallo

13 febbraio 2011

Si ringraziano gli alunni della 3bInf - 4bInf a.s. 2010/2011

Sommario

Le pagine di seguito, di carattere storico e divulgativo, fanno parte di una serie di articoli pubblicati dalla rivista Dev (Infomedia) che purtroppo ha smesso le pubblicazioni. Dopo molti anni rimane in ogni caso un riferimento al quale attingere con fiducia.

1 Informatica come scienza non definita

Informatica è un termine relativamente nuovo nella lingua italiana e, forse, solo parzialmente corretto etimologicamente: esso deriva, infatti, dalla corruzione dei termini **informazione** (la semantica del dato) e **automatica** (la scienza che studia l'avviamento e l'andamento dei processi)¹. E, infatti, pensiero comune oggi associare l'Informatica ai computer: l'associazione è lecita, ma non completa (condizione necessaria ma non sufficiente, si direbbe scientificamente). Come ogni massiccia ed avveniristica costruzione necessita di fondamenta robuste e resistenti, così la Scienza dell'Informazione ha seguito un percorso di sviluppo particolarmente vario, affinando le tecniche di ricerca, produzione, ma soprattutto di analisi dei problemi nel tempo. E come ogni Scienza che si rispetti ha avuto i propri momenti bui, come le visioni irrealizzate di Babbage e della Lovelace, come la scoperta del calcolo quinario da parte di Wilhelm Leibniz, dimenticata per decenni, come la presunzione che un circuito meccanico fosse più efficiente di uno elettrico o che fosse logicamente impossibile determinare un protocollo universale sul quale basare ogni tipo di calcolo. Informatica come scienza dell'informazione, dunque, ma anche come razionalizzazione di concetti privi di ordine, come studio dei fenomeni logici che sottostanno al calcolo e come riproduzione artificiale di quegli stessi processi di calcolo.

¹Oggi si parla più ampiamente di ICT: Information e Communication Technology dato l'irrompere da un po' di anni delle tecnologie per lo scambio delle informazioni a livello globale come Internet

1.1 Tempi antichi

Il lungo cammino percorso dalla Scienza dell'Informazione ha inizio nientemeno che intorno al 3000 a.C. Siamo sotto la dinastia dell'Imperatore cinese Fou-Hi e tra le immagini classiche alle quali siamo abituati fa capolino un simbolo magico, un elemento composto da otto trigrammi. Curiosamente ciascun trigramma è composto di tre gruppi di linee, alcune lunghe ed altre spezzate a metà. Ad un occhio allenato appare evidente che se le righe lunghe corrispondono ad un **1** e quelle brevi ad uno **0**, tali simboli giungono a codificare nientemeno che i primi otto numeri naturali, attraverso un omologo del sistema binario. Occorre ricordare che l'abaco ed il pallottoliere appaiono in medio oriente solo venticinque secoli dopo, pressappoco mentre - intorno al 300 a.C. - Aristotele fonda i principi della Logica. Ma torniamo al nostro ottagonio di trigrammi: occorre attendere il genio di Gottfried Wilhelm Leibniz per analizzarli a fondo e mettere a punto l'aritmetica binaria che tutti noi conosciamo. Tale scoperta gli sarà particolarmente utile quando presenterà, nel 1694, una macchina calcolatrice più perfezionata dell'allora famosa Pascalina, uno strumento inventato dall'allora giovane Pascal per aiutare il padre, un esattore delle tasse, ad eseguire somme e sottrazioni. La macchina calcolatrice di Wilhelm Leibniz era in grado di eseguire in **automatico** (attraverso ruote dentate ed ingranaggi inizialmente in legno) anche moltiplicazioni e divisioni, utilizzando gli studi di Moreland sulle moltiplicazioni presentate come sequenza di addizioni successive. Ma se la scoperta di un sistema di codifica numerica diverso dal decimale può apparire già notevole di per sé, è opportuno ricordare che centocinquanta anni prima Bacon aveva messo a punto un **codice biletterale** attraverso il quale era in grado di codificare con cinque soli simboli differenti tra loro l'intero alfabeto inglese usato in quel tempo. Si trattava, come accennato all'inizio, di una codifica **quinaria**, cioè in base 5, che permetteva 25 diverse combinazioni attraverso l'uso di cinque lettere su due posizioni. Certo, è mancato il salto di genio nell'applicare tale sistema di codifica ai numeri, tuttavia credo che sia giusto annoverare anche questa scoperta nell'ambito delle infrastrutture dell'Informatica attuale. Sinora abbiamo assistito allo sviluppo degli strumenti di base per la velocizzazione del calcolo: ricordiamo che l'Uomo ha iniziato ad utilizzare e domare il fuoco quando sentì l'esigenza di scaldarsi ed a costruire dimore sempre più comode e sofisticate al termine del periodo di nomadismo transumante. In quest'ottica possiamo inserire la spinta matematica verso sistemi di calcolo che, di volta in volta, soddisfacessero le esigenze imposte dalle situazioni contingenti. Il secolo dei lumi ha visto, invece, un tentativo di generalizzazione e categorizzazione del pensiero scientifico (con l'unica, ma ovvia eccezione della Logica di Aristotele) nel tentativo di trarne il massimo beneficio pratico dalle scoperte. Falcon, ad esempio, ebbe per primo l'idea di utilizzare schede perforate per **programmare** il lavoro dei telai tessili. Su tale concetto nel XIX secolo Charles Babbage e Lady Ada Lovelace

(figlia del poeta inglese Byron) producono una serie di specifiche relative alla **meccanizzazione** del sistema di calcolo. Mentre, infatti, sino a quel momento ciascuna macchina calcolatrice era in grado di eseguire esclusivamente un tipo di task, il concetto proposto da Babbage è quello che diventerà portante nell'analisi di funzionamento di un moderno calcolatore elettronico: un sistema, cioè, composto di unità aritmetica, memoria, registri e immissione dati costituiti da schede perforate. Purtroppo il perfezionismo di Babbage e gli investimenti non lungimiranti non permisero alla coppia di condividere il successo che avrebbero meritato, lasciandoli **visionari** del tempo. Alla Lovelace è stato intitolato un linguaggio di programmazione (Ada, appunto), a lei si deve la definizione di **algoritmo** come processo logico di esecuzione di un programma, in onore di Abu Ja'far Muhammad ibn Musa Al-Khwarizmi (780-850 dC.), famoso matematico persiano. Intanto, nel 1770 il progresso si riflette anche nella tecnologia del trattamento dei materiali: Hahn perfeziona i cilindri dentati di Leibniz e produce la prima macchina calcolatrice in grado di eseguire direttamente le quattro operazioni. Cinquant'anni dopo, Charles-Xavier Thomas de Colmar produce ulteriori migliorie sul progetto e crea l'Aritmometro, una macchina talmente versatile e leggera da essere prodotta in ben 1500 esemplari in 30 anni e da meritare la medaglia d'oro all'Esposizione di Parigi del 1855.

1.2 L'era meccanica

Se nei secoli precedenti si è cercato di cavalcare l'onda della prima rivoluzione industriale, nel XX secolo è l'elettronica a farla da padrona. Il secolo si apre con la scoperta del tubo a vuoto (la cosiddetta **valvola termoionica**): John Ambrose Fleming, un ingegnere inglese che compì numerose scoperte nei campi dell'elettronica, della fotometria, delle misure elettriche e della telegrafia senza fili e Lee De Forest, un fisico e inventore americano, concluse la sua vita nel 1961 con oltre 300 brevetti. L'importanza di queste due figure, assurde agli onori della cronaca proprio durante il passaggio del secolo, è incentrata nell'invenzione rispettivamente di diodo e triodo. Sull'onda di queste invenzioni, nel 1919 viene sviluppato il primo oscillatore elettronico bistabile, alla base degli attuali circuiti elettronici di memorizzazione. Per restare in ambito oramai noto, Hermann Hollerith cambia nome alla sua società, Hollerith aveva messo in pratica quei principi mutuati da Falcon e da Babbage ed in occasione del Censimento statunitense del 1890 produsse un sistema di catalogazione a schede perforate basato su un'unità aritmetico-logica, un sistema di memorie e registri, un sottosistema di memorie ed un ingegnoso codificatore di dati su schede di cartone, attraverso il quale era possibile inserire le informazioni da catalogare; sostanzialmente non si era più di fronte ad un apparecchio che produceva un unico risultato aritmetico attraverso un meccanismo dopo la fornitura di precisi dati in ingresso; piuttosto, l'laboratore in questione consentiva di regolarizzare ed organizzare

un flusso di dati caotico traendone informazioni utili ed in modo quanto mai flessibile e veloce, Bene, per produrre e distribuire questa macchina analitica **Hollerith** creó la Tabulating Machine Corporation, una società che verso la fine degli anni Venti venne ribattezzata International Business Machine, familiarmente **IBM**. E mentre gruppi di ricercatori continuavano a portare avanti i propri studi sui sistemi di calcolo elettromeccanici analogici (tra questi ricordiamo Vannevar Bush del MIT, che divenne consulente scientifico personale del presidente degli Stati Uniti), seguendo l'onda dei ricercatori americani, IBM produce il modello 601 (va ricordato che 601 è il codice di blocco del computer scelto da Michael Crichton nel suo **Andromeda Strain**), un modello a relais di cui commercializza 1500 esemplari. Ma l'era digitale incalza. Siamo nel 1937 quando George Stibitz produce il primo circuito binario, un sommatore chiamato Modello K (in cui K è l'iniziale di Kitchen, ad indicare che tale esemplare era stato interamente assemblato nella sua cucina). Il 1937 è un anno che verrà ricordato soprattutto per la famosissima opera di **Alan Turing** sulla logica computazionale: appena di seguito al **Teorema di Gódel** ed in concomitanza con gli studi di Alonzo Church, apre le porte alla teoria della calcolabilità degli algoritmi e produce quel gioiello d'ingegno che è la **Macchina di Turing**, il primo vero computer virtuale dotato di elaboratore, dati e programma. L'anno successivo Shannon discute una tesi in cui per la prima volta vengono messi a confronto l'algebra Booleana ed i circuiti elettrici; sua la definizione di bit, contrazione di binary digit. Nello stesso anno Konrad Zuse progetta e costruisce il Versuchmodell 1 (o Z1), il primo calcolatore reale basato su logica binaria (ancorché meccanico) e l'anno successivo produce lo Z2, presentato all'istituto di ricerche aeronautiche della Germania, utilizzando sezioni elettromeccaniche al posto di alcune parti meccaniche, Contemporaneamente John Atanasoff e Clifford Berry produrranno una sommatrice a 16 bit binaria, il primo calcolatore costruito con tubi a vuoto². George Stibitz (quello del **modello K**) assieme a Samuel Williams, costruisce una calcolatrice adottando la codifica BCD (Binary Codec Decimal). Composto di 450 relais, la macchina era di una semplicità d'uso disarmante, in grado di aggiungere due grandi numeri in meno di un minuto (un grande passo avanti rispetto all'IBM 601 di cinque anni prima, in grado di eseguire una sola operazione al secondo).

1.3 La guerra

Durante la Seconda Guerra Mondiale ciascuno Stato coinvolto cerca in tutte le maniere di ottenere vantaggi decisivi sull'avversario: l'Intelligence giocherà in questa partita un ruolo decisivo. Come ricorderete, i Tedeschi disponevano di notevoli risorse, provenienti dalle loro acciaierie e dalle fabbriche di riconversione: anche i cervelli che ancora non avevano deciso di fuggire all'estero

²Valvole termoioniche che riproducono il funzionamento del diodo prima che questi venga inventato e realizzato in silicio

lavoravano compatti; note al grande pubblico sono le bombe razzo V1 e V2 e gli aerei caccia a reazione prodotti dall'Aeronautica, ma non molti sono a conoscenza dell'enorme vantaggio strategico guadagnato dalla Germania attraverso il loro sistema di cifratura dei dati. Essi avevano, infatti, messo in opera **Enigma**, una macchina elettromeccanica a rotori in grado di modificare il valore delle lettere di un messaggio utilizzando un algoritmo predefinito con una rapidità notevole: la codifica pressoché impenetrabile dei messaggi permise loro il blocco, imposto ai rifornimenti diretti alle truppe alleate, attraverso i sommergibili. La storia ci racconta che per decifrare i messaggi in codice dell'esercito tedesco, gli inglesi misero a punto - presso Bletchley Park - i calcolatori **Robinson** e **Colossus** sotto la direzione del matematico Alan Turing. Con notevole genio anticipatore vennero costruite le prime macchine che integrano i concetti di aritmetica binaria, clock interno, memoria cache, lettori di nastro, operatori booleani, sottoprogrammi e stampanti. Il progetto resterà coperto dal **Segreto di stato** sino al 1975, anno in cui verrà finalmente dichiarato e guadagnerà la fama meritata. Sempre intorno al 1940 la Bell installa due telescriventi a Dartmouth College, collegato attraverso linee telegrafiche ad un Modell a Manhattan durante un congresso dell'American Mathematical Society. In tale occasione due ricercatori, Norbert Wiener e John Mauchly, danno dimostrazione delle potenzialità del calcolo a distanza. L'anno successivo John Atanasoff e Clifford Berry costruiscono ABC, un calcolatore binario dotato di memoria e circuiti logici: la memoria era costituita da due tamburi rotanti e poteva mantenere 60 word da 50 bit; l'elaboratore aveva una frequenza di 60 Hz e poteva eseguire una somma al secondo. Frattanto, in Germania, le macchine progettate da Konrad Zuse (i primi calcolatori a tutti gli effetti, in quanto disponevano di un programma registrato che poteva esser letto attraverso un nastro) avevano la capacità di eseguire ben 4 addizioni o una moltiplicazione al secondo. Infine, ad Harvard (1943) IBM appoggia lo sviluppo e la produzione di ASCC **Mark I**. Costruito secondo i canoni della macchina analitica di Babbage, disponeva di due unità di immissione a banda perforata rispettivamente per dati e istruzioni (architettura di Von Neumann), era costituito di 3000 relais e di ben 800 km di cavo elettrico ed era in grado di effettuare 3 operazioni su 23 cifre per secondo.

1.4 Il dopoguerra e l'avvento dell'informatica

Nel 1945 Vannevar Bush, già noto per l'analizzatore differenziale, pubblica **As we may think**, dove descrive una sorta di macchina immaginaria chiamata **Memex** in grado di aiutare un individuo a ricercare e trovare diversi generi d'informazione dalle caratteristiche simili attraverso legami ed associazioni tra gli elementi del documento. Ci troviamo di fronte alla descrizione formale dell'ipertesto. La fine della Guerra lascia tutti frastornati: i vincitori consci di non dover imporre sanzioni troppo pesanti, i perdenti

presi dalla necessità di ricostruire, anzi di costruire un presente differente. Una nuova riconversione ha luogo nell'industria e le macchine - un tempo appannaggio unico del Ministero della Guerra - iniziano ad essere prese in considerazione per un utilizzo commerciale. A tal proposito, quasi a voler sdrammatizzare una situazione grave ma in fase di lenta e costante ripresa, ricorderò un episodio avvenuto proprio su quel Mark I utilizzato dagli Stati Uniti: un insetto, caduto per errore in un circuito, blocca il funzionamento della macchina. La matematica Grace Murray Hopps deciderà che tutto ciò che arresterà il corretto funzionamento di un elaboratore prenderà il nome di **bug** (insetto). Pare che l'espressione sia rimasta, sebbene a ricerche più approfondite sia emerso che lo stesso Edison avesse impiegato tale termine durante la descrizione di un suo esperimento. In quel periodo John Von Neumann raccoglie il gruppo di studiosi che lavorò a suo tempo su **ENIAC** e pubblica il primo rapporto in cui viene descritto in dettaglio un calcolatore elettronico con programma registrato, che chiamerà EDVAC (Electronic Discrete Variable Automatic Computer). I computer che seguiranno le specifiche di questo documento, si dirà, lavoreranno secondo una **architettura di Von Neumann**. Come tutti sappiamo, i computer che da 50 anni a questa parte sono stati costruiti e che lavorano sulle nostre scrivanie (salvo rarissime eccezioni) osservano ancora queste specifiche, celebrando tuttora il suo genio visionario.

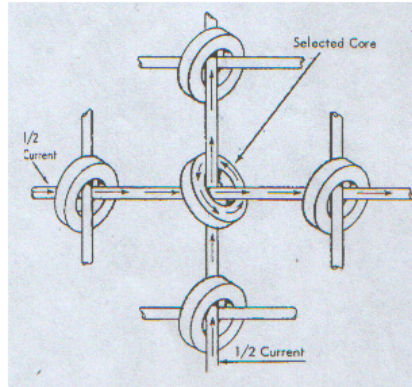


Figura 1: Memoria a Nuclei di ferrite

2 E l'hardware??

Hardware: ferramenta, ferraglia. Con questo termine viene definita la parte **rigida** di un elaboratore, sia esso elettronico, elettromeccanico, meccanico; forse con l'avvento dei computer quantistici o nanotecnologici tale termine potrà subire un netto cambiamento (*ricordiamo che nelle reti neurali il dato è spesso indistinguibile dal programma che lo elabora e che gli elaboratori a parallelismo massiccio non seguono il paradigma di Von Neumann*) o forse, nonostante la stridente contraddizione semantica, continueremo a definire hardware ciò che **giace nella scatola**, limitandoci al massimo a definire il tutto come **firmware** od **orgware** indistinto.

2.1 La legge di Moore

Si sente spesso parlare della cosiddetta **Legge di Moore**, secondo la quale **ogni chip ha capacità circa doppia rispetto al suo predecessore e ogni 18-24 mesi nasce una nuova generazione di chip**; allo stesso modo sono apparsi il **Corollario di Machrone** (il computer che desideri o che ti dicono che devi assolutamente comprare, costa sempre la stessa cifra, nonostante gli anni passino) ed il **Corollario di Rock** (l'ammontare degli investimenti che vanno impiegati per costruire semiconduttori raddoppia ogni quattro anni). Quest'ultimo, in particolare, nasconde un dilemma assai insidioso, che difficilmente viene riportato dai media; anzi, per quanto ne sappia, nessuno lo ha mai formulato esplicitamente: **Se la potenza di un chip raddoppia ogni 2 anni, mentre gli investimenti raddoppiano ogni quattro e il computer che desideri costa sempre la stessa cifra,**

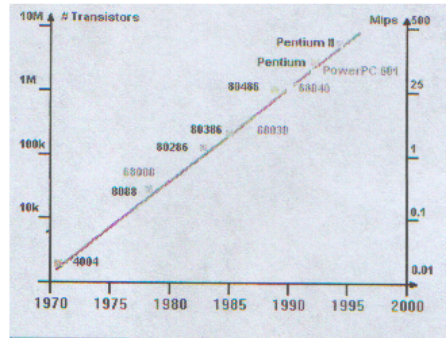


Figura 2: Previsioni date dalle Legge di Moore

allora la produzione di chip raddoppia ogni anno, quadruplicando la potenza di calcolo o (il che è lo stesso) redistribuendola al doppio della popolazione. Ciò significa che, tenendo conto dell'intrinseca limitazione tanto della crescita della tecnologia (velocità della luce, limiti di Planck) quanto della crescita della popolazione in grado di acquistare un computer, oramai vicina alla **crescita-zero**, questi numeri porteranno in un tempo ben delimitato da equazioni alla rovina degli imprenditori del silicio. Bene, ora facciamo un bel respiro e andiamo avanti. Gordon E. Moore, Chairman Emeritus alla Intel Corporation, espresse la legge che porta il suo nome nel lontano (informaticamente parlando) 1965. Come tutti sanno, la Intel fu da lui fondata nel 1968 e diretta dal 1979 al 1987; ciò che non tutti sanno è che nel 1961 venne prodotto il primo circuito integrato planare e quindi la **legge** era allora considerata una sorta di approssimazione. Come risulta visibile in Figura 3, invece, l'approssimazione divenne una vera e propria legge, con un suo trend costante. Ma si sa: fatta la legge, trovato l'inganno...

2.2 Potenza di calcolo

La legge di Moore è valida per i microprocessori, non per i computer in genere: l'elaboratore, come abbiamo visto nella scorsa puntata, è nato meccanico, si è trasformato in elettromeccanico, passando per l'elettronica valvolare, per i transistor e solo alla fine concedendosi al microprocessore. Possiamo affermare che approssimativamente tra il 1780 ed il 1850 hanno avuto il loro momento di grazia gli elaboratori meccanici, dal 1850 al 1900 gli elaboratori elettromeccanici, tra il 1900 ed il 1940 i calcolatori a valvole, tra il 1940 ed il

1970 quelli a transistor, tra il 1970 ed il 2000 quelli a microprocessore; siamo quindi giunti sicuramente ad un'epoca di transizione, in cui il passaggio ad una tecnologia totalmente differente (e piú efficiente) è da un lato richiesta dalla Fisica stessa della materia, dall'altra invece fortemente osteggiata dalle lobby economiche, operanti tra l'altro in modo simile anche nel bloccare lo sviluppo di motori a scoppio non basati sul petrolio (Corollario di Rock sul ritorno dell'investimento). La legge di Moore è quindi destinata a cessare la propria validità o a modificarsi con l'avvento di nuove tecnologie produttive. E prima, come andavano le cose? Sulle orme del Documento di Von Neumann, l'EDVAC, vennero costruiti i primi elaboratori elettronici **veri**, diretti al calcolo numerico ed alle transazioni (ENIAC ed EDSAC, seguiti nel 1949 da BINAC il primo computer biprocessore). La Z4 di Konrad Zuse viene rimontata al Politecnico di Zurigo e migliorata: è ora la prima calcolatrice in grado di eseguire salti condizionati e dotata di una pipeline a due istruzioni. Sempre nel 1951 viene prodotto il primo tamburo rotante come memoria di massa, mentre P. Eckert e J. Mauchly, dopo aver progettato ENIAC, producono UNIVAC (UNIVersal Automatic Computer), il primo elaboratore commerciale della storia. Siamo giunti ad una capacità elaborativa di ben 8333 addizioni o 555 moltiplicazioni al secondo! L'anno successivo IBM produce il Modello 701 per il Ministero della Difesa. Con una memoria di 2048 o 4096 word da 36 bit, poteva realizzare 16000 addizioni o 2200 moltiplicazioni al secondo. Rispetto all'architettura di UNIVAC si nota un microprogramma indirizzato all'esecuzione piú efficace delle moltiplicazioni rispetto alle addizioni di un fattore 2. Il primo computer commerciale IBM, il Modello 650, nasce invece a luglio del 1953; la sua peculiarità consiste nella compatibilità con gli elaboratori a schede perforate della stessa marca. E siamo giunti al 1955: Gene Amdahl, che successivamente fonderà una società per la produzione di computer di classe mainframe compatibili IBM, sviluppa per IBM il Modello 704, la prima macchina commerciale dotata di coprocessore matematico. Potenza: 5kFLOPS (per fare un paragone, un Pentium 90 risulterebbe 6.400 volte piú potente). Memoria di 32768 word da 36 bit, risulta 3 volte piú veloce del Modello 701, prodotto solo 3 anni prima, con una memoria 8 volte maggiore. Già in questo paragone possiamo vedere come la legge di Moore non fosse applicabile allora, in quanto la crescita di potenza risultava doppia di quella prevista dalla legge stessa.

2.3 Informatica Personale

Negli anni Cinquanta troviamo diverse interessanti novità: la nascita dei primi elaboratori elettronici ibridi (a valvole e transistor) e totalmente a transistor da parte di Olivetti (Elea) e Beh (Tradic). IBM produce i primi strumenti per la memorizzazione di dati su disco; chiamati RAMAC 305 (Random Access Memory for Accounting and Control), sono composti da 50 dischi da 61 cm, in grado di memorizzare ben 5 Megaword. La WhirlWind

produce TXO, il progenitore dei mini computer, per testare il funzionamento di memorie a nuclei di ferrite e transistor. Le sue caratteristiche: processore a 18 bit, 3.500 transistor, 83.000 istruzioni al secondo, memoria di 65.536 word. Nel 1957 viene prodotto il primo elaboratore elettronico transistorizzato sovietico: ciò significa che i calcoli per il progetto e la messa in orbita dello Sputnik vennero eseguiti a mano o con elaboratori a valvole. E proprio in seguito al lancio dello Sputnik, il presidente Dwight D. Eisenhower richiede la creazione di ARPA, l'antesignana di Internet. Nell'anno 1958 Bull produce il primo supercalcolatore dotato di supporto hardware per il multi-thread. Nello stesso anno viene lanciato il CDC 1604, un calcolatore costruito da un certo Seymour Cray, che diverrà noto per la produzione di calcolatori elettronici dalle prestazioni assolutamente superiori. Ancora nel 1958 Texas Instruments produce il primo circuito integrato, Bell il primo modem, John McCarthy (matematico del MIT) creerà il LISP su un IBM 7090, Willy Higinbotham il primo videogioco della storia (era simile al Pong distribuito da Atari nel 1972). Nel 1959 viene prodotto il PDP-1 da parte di Digital e l'ATLAS I: l'uso di memoria virtuale, multiprogrammazione, pipelining ed operazioni in virgola mobile la rendono avveniristica, mentre a potenza di calcolo, 200 kFLOPS, risulta di ben quaranta volte maggiore di quella dell'IBM 704 di quattro anni prima... Negli anni Sessanta Fairchild inizia la commercializzazione dei circuiti integrati, IBM mantiene il 62,5

2.4 I Personal Computer

Frattanto IBM continua a perdere quote di mercato. Nel 1968 un gruppo di ricercatori proveniente dalla Stanford University esegue la prima dimostrazione di interazione con un ambiente a finestre, mouse ed ipertesti; Burroughs lancia i primi elaboratori costruiti con circuiti integrati; Hewlett Packard presenta la prima calcolatrice programmabile da ufficio. Niklaus Wirth crea il linguaggio di programmazione Pascal (ne parleremo nella puntata dedicata al software), Kernighan, Ritchie e Thompson ultimano UNIX. L'anno successivo vede la produzione del CDC 7600 di Mr. Cray e la creazione delle specifiche di connessione seriali RS232. Digital Equipment Corporation presenta il PDP-11, che con un processore a 16 bit offre maneggevolezza, interazione, potenza e compatibilità tra macchine diverse. Al PARC (Palo Alto Research Center) creato da Xerox presso la Stanford University, viene approfondita e sviluppata la **metafora della scrivania**, un sistema di gestione grafico a finestre. Ma i tempi sono maturi per una nuova rivoluzione. A gennaio del 1971 Don Hoefler parla per la prima volta di Silicon Valley; ad aprile Arpanet conta 23 computer collegati da 15 differenti siti; Bili Fernandez e Steve Wozniak assemblano un piccolo elaboratore casalingo utilizzando pezzi di scarto di altre società d'informatica; a fine anno Intel mette in vendita il primo microprocessore, concepito da Marcian Hoff, Stanley Mazor e dal nostro Federico Faggin. Si trattava di un vero e proprio **computer su**

un chip, con processore a 4 bit cloccato a 108 kHz, in grado di indirizzare 640 byte di memoria e di eseguire 60.000 operazioni al secondo. Ben 2.300 transistor assemblati in tecnologia a 10 micron. Si racconta che sul primo chip Intel 4004 siano state iscritte le iniziali F.F. Ma perché 4004? Il primo 4 si riferisce ai 4 bit del processore, mentre il resto è dovuto al progetto dell'MCS4, il primo microcomputer commercializzato da Intel (sebbene si vociferi che i piani progettuali appartenessero inizialmente ai francesi o ai russi): tale microcomputer disponeva di una ROM, una RAM ed uno shift register, rispettivamente Intel 4001, Intel 4002 ed Intel 4003. A meno di 5 mesi di distanza viene prodotto il chip Intel 8008, fratello maggiore a 8 bit e 200 kHz, 60.000 istruzioni (a 8 bit) al secondo, 3,500 transistor. Ma il 1972 vede diversi successi: Bushnell e Dabney creano l'ATARI portando all'esordio Pong; Alan Kay crea SmallTalk, il primo linguaggio ad oggetti; inizia la proliferazione di mailing-list su Arpanet (sintomatico notare come una delle prime sia sf-lovers); appaiono i primi floppy da 5,25 pollici; Hewlett Packard presenta la calcolatrice programmabile tascabile HP 65, che verrà portata nello spazio durante la missione ApoIlo-Soyuz; Bili Gates e Paul Allen fondano la Traf-O-Data, produttrice di un sistema basato sull'Intel 8008 per la misura del traffico automobilistico; inizia lo studio di un documento per le specifiche del TCP/IP. Nell'anno successivo appare per la prima volta il termine *microcomputer* in un articolo della stampa americana, Ken Thompson riscrive UNIX in C, Gary Kildall scrive il primo sistema operativo per microcomputer, il CP/M (Central Program for Microcomputers), Bob Metcalfe mette a punto l'interfaccia diretta Ethernet presso gli stabilimenti Xerox, IBM inventa il disco Winchester, iniziano ad apparire i primi microcomputer commerciali basati su microprocessore. Aprile 1975: Ed Roberts costruisce Altair, il primo microcomputer basato sul processore Intel 8080: clock a 2 MHz, 64 KB di memoria indirizzabile, 640.000 istruzioni al secondo prezzo: inferiore a 400 dollari in scatola di montaggio. Pare che il nome della macchina non derivi dal nome della stella protagonista nel film **Il pianeta proibito**, bensì dall'episodio **Viaggio ad Altair** della serie televisiva Star Trek. Nello stesso periodo la MOS Technology mette in vendita il microprocessore MC6502 (il futuro cuore di Apple e Commodore 64 e Vic 20) a 25 dollari, mentre il prezzo di un 8080 era fisso a 150 dollari. A Settembre 1975 esce il primo numero di **Byte magazine**. Marzo-Aprile 1976: Steve Jobs e Steve Wozniak, rispettivamente 21 e 26 anni, costruiscono il loro calcolatore che chiamano Apple Computer. Fondano la Apple Computers il 1 Aprile 1976 ed il loro computer verrà venduto da Byte Shop a 666,66 dollari. Nel mese di giugno Texas Instruments mette in commercio il primo processore 16 bit, il TMS 9900; a luglio la Zilog, fondata da Federico Faggin nel 1974, produce lo Z80, processore a 8 bit secondo alcuni clonato dall'Intel 8080; ad agosto Wozniak inizia a progettare l'Apple II, presentato a Dicembre dello stesso anno. Quasi contemporaneamente viene presentato il Cray I, primo supercomputer vettoriale di una lunga e blasonata serie di elaboratori

vettoriali dalle prestazioni esaltanti.

2.5 Gli anni ottanta

Ciò che segue cessa d'essere storia e diviene cronaca. Da un lato i microcomputer ad 8 e 16 bit che si trasformano lentamente in console da videogioco o in personal, dall'altro un processo lento ed apparentemente irreversibile che incombe sui mamrnut, il crollo dell'importanza dei mainframe, il downsizing ed il rightsizing, la crescita della Rete, l'interconnessione, la multimedialità ed una tonnellata di neologismi dalla pronuncia difficile e dalla vita breve all'inseguimento di un sogno oramai dissolto costituito dal pionierismo informatico, quando chi sapeva dove mettere le mani era un vero **genio del male**, quando il debug veniva impostato in modalità hardware, quando i sistemi di elaborazione richiedevano raffreddamento ad acqua o elio liquido e si utilizzavano gruppi di continuità costituiti da motori marini di transatlantici. Altri settori da ricordare riguardano gli sviluppi avuti dalle periferiche di salvataggio dati (SASI, poi SCSI, nacque nel 1979), dalla larghezza ed efficienza dei bus di trasmissione (la Hayes produsse per Apple un modem a 110/300 baud), dalle caratteristiche delle interfacce grafiche (basti pensare che nel 1968 Benoit Mandelbrot studiava gli insiemi frattali che presero il suo nome su di un insulso monitor monocromatico); la rivincita dei minielaboratori, grazie a UNIX in diversi sapori, come workstation grafiche in rete. Ciò che invece manca su Internet é la percezione del genio e della follia di quegli anni gloriosi, il cambiamento di indirizzo del concetto di elaboratore elettronico nato come sistema di aiuto alla burocrazia durante il censimento, divenuto sistema di elaborazione numerica spinta, trasformato in sistema di gestione della rete ed infine assunto ai meriti di **giocattolo tecnologico** per la maggior parte degli utenti. E nel frattempo il microprocessore, che tanta parte ha avuto nell'avanzamento del livello di diffusione del computer, sta vivendo una seconda giovinezza nei laboratori di genetica, dove vengono forgiate nuove molecole per consentire un trasferimento di informazione piú efficiente ed una logica piú **umana**.

3 La guerra dei Sistemi Operativi

L'Informatica come scienza indipendente prende piede vista la crescita imperiosa nelle prestazioni degli elaboratori elettronici dal 1945 ad oggi, resa possibile da un continuo lavoro di ricerca e sviluppo e dall'offerta di un numero sempre crescente di esemplari, che garantivano una economia di scala in grado di coordinare gli studi necessari per affrontare i nuovi problemi legati alla progettazione di sistemi di produzione sempre piú avanzati.

3.1 Primi passi

I primi monolitici elaboratori elettronici erano un groviglio di cavi e valvole termoioniche. Il tecnico responsabile del sistema ne era allo stesso tempo progettista, sistemista, programmatore ed operatore: ovvio quindi che ciascun nuovo progetto presupponesse un pool di tecnici differente, gestito da colui o coloro che di volta in volta avevano il controllo dello sviluppo. Fu IBM a rendersi conto che anche l'economia di gestione di un elaboratore elettronico poteva sottostare alle ferree leggi di Von Neumann: era sufficiente infatti studiare il flusso dei dati in ingresso ed in uscita come un'astrazione ed utilizzare l'oggetto **flusso** come un substrato sul quale attivare di volta in volta gli impulsi necessari per la gestione del programma; se tali impulsi fossero stati codificati in maniera logica anziché fisica, il livello dei segnali gestito dall'hardware non avrebbe dovuto essere modificato dal progettista durante la programmazione, ma avrebbe interagito con l'hardware in maniera trasparente. In questo modo il progettista avrebbe assunto il ruolo di programmatore senza doversi sobbarcare anche il ruolo di ingegnere elettrico: la codifica logica dei segnali avrebbe prodotto gli effetti desiderati qualunque fosse la macchina sulla quale si stava lavorando. Era il 1953, data di nascita dell'IBM 650, il primo computer commerciale di IBM il cui flusso di schede perforate era compatibile con i suoi successori ed il primo ad essere costruito in un migliaio di esemplari per ragioni di economia di scala, per dotare le Università statunitensi di un sistema di elaborazione univoco. Due anni prima Grace Murray Hopper aveva creato il primo compilatore, l'AO, che consentiva di creare un eseguibile a partire da un codice sorgente. Nel 1955-57 sull'IBM 704 creato da Gene Amdahl verrà invece sviluppato il FORTRAN.

3.2 Computer Utilizzabili

L'idea di riutilizzare parti comuni della ricerca sugli elaboratori si stava facendo strada: oltre ai linguaggi di programmazione fioriti nella seconda metà degli anni cinquanta si sentiva la necessità di accrescere l'interazione tra operatore e la macchina. IBM, nel 1959, annuncia il modello 1401: 12.000 esemplari sviluppati per l'amministrazione e la contabilità e gestiti da un unico sistema elettromeccanico di immissione dati a schede perforate. Ma il

vero punto di forza di tale sistema era una stampante veloce in grado di produrre sino a 600 linee di stampa al minuto. Sempre nel 1959 Digital sviluppa il sistema PDP-1, il primo elaboratore elettronico interattivo. Prodotto da geni dell'information technology di allora, il PDP-1 rappresentava per la prima volta nella storia un elaboratore **divertente da usare**: non per nulla su di esso S. Russel, J.M. Graetz e W. Wiitanen (studenti del MIT) svilupperanno nel 1960 Space War, il primo videogioco interattivo della storia! E Digital, nel frattempo trasformatasi in DEC, rendendosi conto dell'impatto pubblicitario provocato con l'annuncio di questa nuova frontiera dell'Informatica **ludica**, offrirà gratuitamente una copia di Space War su ogni PDP-1 venduto. Coloro che affermano che l'attuale sistema operativo di Microsoft ha successo grazie al suo aspetto simile ad un gioco non sono certamente in torto.., sono solo in ritardo di 40 anni sulla Storia. E per aggiungere un particolare divertente a questa sezione, ricordiamo che uno dei piú assidui giocatori di Space War era uno studente dell'Università dello Utah, un certo Nolan Bushnell, che in seguito fonderà la società Atari.

3.3 Periferiche

Nel 1963 viene lanciata la Teletype Inktronic, la prima stampante a getto d'inchiostro. Dotata di connessione a 1200 baud, era in grado di disegnare caratteri alfabetici su 80 colonne; nello stesso anno Ivan Sutherland del MIT produce il primo computer grafico interattivo in grado di utilizzare una penna ottica. E sempre in campo grafico, nel 1965 Ted Nelson produce una serie di specifiche per l'interconnessione di documenti elettronici, coniato i termini ipertesto ed ipermedia. Il 1966 vede la nascita della prima console per videogiochi, la Magnavox Odyssey I da parte di Ralph Baer. Tra il 1967 ed il 1968 nascerà invece la Evans&Sutherland, la prima società specializzata nelle applicazioni di digitalizzazione e riproduzione grafica della realtà, mentre IBM produrrà il primo lettore di floppy disk. Ma l'evento cardine del 1968 risulterà lo studio di Douglas C. Engelbart dello Stanford Research Institute, nel quale verrà data dimostrazione di un ambiente grafico con finestre gestibile attraverso un mouse: in tale ambiente Engelbart mostrerà come interagire con un documento di testo, con un sistema basato su ipertesto e con una scrivania elettronica interattiva pensata per il lavoro collaborativo.

3.4 Interoperabilità

Frattanto un gruppo di studenti di informatica, guidati da Ken Thompson e Dennis Ritchie iniziarono a riflettere su di un nuovo sistema di elaborazione a partizione di tempi, ma nessuno nel management dell'istituto sembrò interessarsi al progetto; costoro trovarono pertanto un vecchio PDP-7 inutilizzato (evoluzione oramai obsoleta del glorioso PDP-1), recuperato inizialmente da

Thompson per produrre un gioco interattivo, Space Travel ed iniziarono a lavorare. Alcune idee vennero inizialmente mutuate dal progetto MUL-TICS (nozioni di processo, albero delle directory, interprete a linea di comando che gira come un programma a se stante), altre vennero sviluppate da zero (ad esempio, l'idea di utilizzare l'output standard di un programma come fonte dati di un processo successivo). In breve tempo il PDP-7 si trovò ad ospitare un kernel primitivo, alcuni programmi di utilità, un editor ed un assembler; fu Brian Kernighan infine a trovare un nome valido per il sistema appena nato: Unix, in raffronto al ben più corposo e mai terminato Multics. Ken Thompson, infine, pensando che un sistema del genere non sarebbe completo senza un linguaggio di programmazione ad alto livello, inizia il porting di un compilatore Fortran sul PDP 7, ma poi cambia idea ed opta per il linguaggio interpretato B (in onore al BCPL al quale si ispira). Tra gennaio e marzo del 1971, con la scusa di sviluppare un nuovo elaboratore di testi (dopo l'insuccesso Multics, il termine **sistema operativo** era vietato nei Bell Labs) Thompson e Ritchie ottengono in prestito un PDP 11/20 ed eseguono di nascosto il porting del loro Unix: nei 24 KB di memoria del computer, 16 erano occupati da Unix, gli 8 restanti erano a disposizione degli utenti, il disco poteva contenere 512 KB ed i file una grandezza massima di 64 KB. L'utility *roff* venne portata sul sistema per consentire a tre utenti del servizio brevetti di lavorare sulla macchina mentre Thompson e Ritchie continuavano a lavorare allo sviluppo del sistema. Il successo di tale esperienza ha provato l'utilità di Unix e ne ha consentito un porting anche sul successivo PDP 11/45. E a questo punto che Dennis Ritchie riprende il linguaggio B, scritto da Thompson per il PDP 7 e decide di adattarlo al PDP 11 sul quale Unix è appena stato portato. Più che in adattamento si tratterà di una riscrittura che renderà il C in grado di creare codice macchina per il PDP 11. A questo punto il passo successivo sarà quello di riscrivere Unix in C per renderlo portabile su tutte le macchine che dispongano almeno di un compilatore C standard.

3.5 Fuochi d'artificio

Nel 1971 Gary Starkweather mette a punto la prima stampante laser nei laboratori Xerox di Palo Alto; dagli stessi laboratori viene prodotto l'anno successivo SmallTalk, il primo linguaggio ad oggetti, da parte di Alan Kay. A marzo del 1973 viene presentata Xerox Alto, una stazione di lavoro con schermo grafico che utilizza il linguaggio SmallTalk, una interfaccia grafica, un mouse ed una scheda di rete Ethernet. Nel mese successivo Dick Shoup, sempre di Xerox, mette a punto una stazione dotata di scheda grafica a colori in grado di gestire una risoluzione di 640x486 punti a 256 colori e di digitalizzare un segnale video. Realizzerà anche Superpaint, che risulta allo stesso tempo un programma di disegno a colori ed un prodotto per la digitalizzazione elettronica di immagini video. Poiché tale progetto non rientrava

negli obiettivi della Xerox, venne annullato. Dick Shoup si dimise due anni dopo e fondó Aurora System, una società specializzata nel trattamento di immagini digitali e di mappe meteorologiche per la televisione. Nel frattempo viene fondata la società Atari e in meno di due mesi vengono venduti 10.000 **Pong** in altrettanti locali. Appaiono i primi floppy disk da 5 pollici e un quarto. Gary Kildall sviluppa CP/M, il primo sistema operativo per microcomputer. Tale sistema si diffuse su tutti i sistemi micro professionali, tanto che Kildall fondó la Intergalactic Digital Research Inc. (poi rinominata semplicemente Digital Research Inc.), ma subí uno stop definitivo quando i microcomputer vennero dotati di un interprete BASIC (grazie a Paul Allen ed a Bill Gates). Occorre a questo punto notare che mentre gli home microcomputer disponevano in genere di un interprete BASIC in ROM che veniva lanciato automaticamente all'accensione della macchina, gli elaboratori piú professionali avevano bisogno di caricare un programma per la gestione delle periferiche e del sistema (che in genere era ben piú complesso); tale sistema trovava posto su di un floppy disk caricato al boot-strap esattamente come avviene oggi, solo che il boot caricava un interprete dei comandi relativamente leggero e poco versatile. Il CP/M venne dotato successivamente di supporto per la multiutenza (il CP/M con MP/M) e diversi uffici vennero dotati di tali sistemi informativi. Nel febbraio del 1975 appare il primo sistema WYSIWYG: Bravo, sviluppato da Charles Simonyi nei laboratori Xerox di Palo Alto, mentre Michael Shroyer produce Electric Pencil, il primo programma di elaborazione testi per microcomputer, sul proprio Altair. Sempre a Palo Alto, nel 1979, viene invitato un gruppo di sviluppatori della Apple, tra i quali Steve Jobs. Durante una dimostrazione della workstation grafica Alto (della quale abbiamo già citato le caratteristiche), Jobs ha una visione che lo porterà alla modifica del progetto Lisa ed al MacIntosh. Nel luglio del 1980 IBM è alla ricerca di un sistema operativo per i propri nuovi **Personal Computer** (IBM PC). Il loro interesse risulta rivolto al CP/M di Gary Kildall, che già consente il funzionamento di differenti sistemi utilizzando la medesima interfaccia di comando, ma il giorno dell'appuntamento Kildall non è presente; si racconta che girasse sull'aeroporto con il proprio aereo privato aspettando che IBM accettasse la sua offerta economica. I funzionari IBM, invece, indispettiti dall'essere trattati in quel modo da un ultimo arrivato e desiderando rendere pan per focaccia, accolsero l'offerta che un ragazzino lentigginoso aveva fatto loro pervenire tramite la propria madre, un avvocato di grido. Fu così che Microsoft ottenne l'esclusiva per la fornitura di un sistema operativo ad IBM ed al loro neonato PC. E' da sottolineare il fatto che la stessa Microsoft era stata artefice del successo di Kildall, in quanto produttrice della **Soft Card** che aveva permesso la diffusione del CP/M su Apple. Il mese successivo vede Microsoft pronta a distribuire Xenix, un porting di Unix per macchine basate su processori Intel x86 e Motorola M6800. Contemporaneamente Tim Patterson, della società Seattle Computer Products, sviluppa in due mesi la versione 0.10

di QDOS (acronimo di Quick and Dirty Operating System), un clone del CP/M riprogrammato per processori a 16 bit. Un altro mese occorre per fornire al QDOS un editor di linea, chiamato poco fantasiosamente Edlin. A settembre Potterson presenterà il tutto sotto il nome di 86-DOS e ad ottobre Microsoft alla disperata ricerca di un sistema operativo per macchine a 16 bit per rispettare l'accordo con IBM, acquista per 50.000 dollari i diritti dell'86-DOS. Il resto è storia nota...

4 Programmare che passione

L'avvento dei personal Computer, porta con se la nascita di DOS e Windows. Quest'ultimo, nato come sistema di supporto grafico al DOS, è divenuto via via sempre più indipendente da quest'ultimo sino ad assurgere alla fama oggi conquistata (lasciando la discussione sui suoi meriti ai posteri). Abbiamo volutamente trascurato la polemica che portò alla rottura della collaborazione tra IBM e Microsoft nello sviluppo del sistema operativo OS/2 (inizialmente voluto da IBM come sistema operativo per gli allora rivoluzionari PS/2), a causa della necessità di Microsoft di mantenere la compatibilità con il mondo delle applicazioni a 8/16 bit: la storia racconta oramai che pochi mesi dopo la rottura, durante lo sviluppo di OS/2 2.0, la casa di Redmond lanciò il rivoluzionario (rispetto alle versioni precedenti) Windows 3.0, prontamente seguito dal 3.1 bug-free, dall'interfaccia grafica piacevolmente migliorata rispetto al 3.0 e ad OS/2 2.0 uscito in fretta e furia pochi mesi dopo. Allo stesso modo eviteremo di parlare della migrazione di quella materia grigia che produsse il VMS (l'importante sistema operativo implementato dalla Digital sui propri VAX) verso altri lidi, al fine di gestire ed ottimizzare il kernel di NT. Ci occuperemo, invece, di Big Blue e dei suoi sistemi operativi, per poi occuparci di qualcosa di completamente diverso.

4.1 Ibm, mainframe e sistemi operativi

Tra gli anni Sessanta ed i primi anni Ottanta del XX secolo si è assistito ad un incremento prestazionale dei mainframe dovuto all'introduzione sempre più spinta di circuiti elettronici di tipo solid-state ed alla centralizzazione dell'informazione da parte dei maggiori enti governativi. Un fenomeno del genere provocò un aumento quadratico della quantità di dati da trattare con efficienza e con l'aumentare delle informazioni occorreva mantenere sotto controllo il livello qualitativo del servizio, evitando che dati sensibili si perdessero. L'introduzione dei Sistemi 360 e 370 consentì ad IBM di ottenere una posizione di spicco: si narrava che i suoi sistemi operativi fossero dotati di una istruzione di controllo per ogni due istruzioni esecutive e di un sistema di controllo elettronico di ridondanza pur mantenendo la capacità di gestire e monitorare migliaia di transazioni remote al secondo. L'architettura S/370 venne mantenuta sui mainframe di classe 43XX, 93XX e 30XX e poteva essere descritta come una serie di sottosistemi coordinati tra loro a più livelli da un supervisore: ad un sistema di gestione delle periferiche di output e dei canali chiamato DOS/VSE fece subito riscontro il nuovo sistema operativo MVS (Multiple Virtual System) con i sottosistemi per il job entry (Jes, Jes2), il time sharing remoto (TSO), la gestione delle comunicazioni remote (BTAM/VTAM) ed un monitor transazionale interfacciabile con diversi linguaggi di programmazione, in grado di consentire l'accesso tramite coassiale e l'interazione tra molti utenti remoti contemporaneamente su di

un p001 di dati condivisi e mantenuti in memoria virtuale. Oggi un sistema del genere può essere gestito senza soverchie preoccupazioni da un server Intel di classe Pentium-II, ma trent'anni fa era una vera meraviglia della tecnica. In più, per chi volesse il massimo, era disponibile VM (Virtual Machine), una sorta di shell che consentiva a ciascun utente evoluto di un sistema mainframe di caricare la propria configurazione di sistema operativo contemporaneamente, fornendo in sostanza diverse istanze di un prodotto o diverse versioni di sistema operativo per controllarne il comportamento. Di nuovo ci viene in mente l'interfaccia a finestre o VMware, ma occorre anche ricordare che allora con 64 MB di memoria RAM si gestivano 2400 utenti online contemporanei ed un sistema di dischi fissi con cache permetteva al massimo velocità di accesso di 24 millisecondi... Meraviglie d'altri tempi.

4.2 Algoritmi e linguaggi

Sembra oramai dimostrato che il cammino culturale e tecnologico di qualsiasi entità vivente non possa prescindere da una solida infrastruttura di comunicazione e di linguaggio, nella quale ad una base semantica e concettuale viene collegato un articolato sistema di morfologia e sintassi. Senza giungere necessariamente all'esempio banale dell'uomo, possiamo infatti osservare che le comunità degli **insetti sociali** come api, termiti e formiche sfruttano un complesso sistema di segnali a favore della comunità; anche tra i mammiferi, i primati ed i cetacei hanno sviluppato un sistema di articolazione dell'apparato fonatorio e della gestualità tale da collocarli all'estremo superiore della scala dell'intelligenza animale (se si prescinde dall'essere umano). Appare quindi evidente che per elaborare e trasportare dati convertendoli in informazioni significative non si possa prescindere da una struttura organizzativa complessa. Se ora analizziamo il cammino percorso dai linguaggi di programmazione, ci rendiamo conto che la struttura sottostante è giunta a maturazione lentamente, per passi successivi e solo da poco siamo in grado di creare sistemi assiomatici ristretti dotati di una qualche completezza formale. Il nostro viaggio inizia, come ricorderete, intorno all'820 d.C. con lo studioso persiano Al-Khuwarizmi: questi, infatti presenta in tale data a Baghdad un testo intitolato **La scienza dell'eliminazione e della riduzione**, un trattato logico matematico che avrà grande riscontro nell'Europa occidentale durante l'invasione araba e costituirà una solida base per le successive elaborazioni (Algebra di Boole). Ada Lovelace conià in suo onore il neologismo **algoritmo**, europeizzando il nome del matematico del passato, per indicare il principio basato sulla esecuzione di una operazione attraverso la successione di iterazioni elementari su di un cammino definito. Nel 1854 George Boole (1815-1864), matematico inglese, crea un'algebra che prenderà il suo nome. Ricordiamo che un'algebra è un insieme che soddisfa le strutture di anello e uno spazio vettoriale con proprietà ed assiomi speciali; l'algebra Booleana è un'algebra nella quale le operazioni binarie vengono scelte per

gestire operazioni di unione ed intersezione nella Teoria degli Insiemi. In particolare, per ogni insieme A , il sottoinsieme di A forma un'algebra Booleana sotto le operazioni di unione, intersezione e complementazione. In particolare dimostra che qualsiasi processo logico può essere decomposto in una serie di operazioni logiche collegate dai connettivi **e**, **o**, **non** applicati su due differenti stati (zero-uno, si-no, vero-falso, aperto-chiuso). Su questi studi verranno teorizzate e realizzate differenti applicazioni, dai circuiti elettrici ed elettronici ai sistemi idraulici, dall'assiomatizzazione dei costrutti logici all'approfondimento della teoria degli insiemi. Ma per vedere i primi veri linguaggi di programmazione è necessario attendere ancora un secolo: è infatti del 1950 il primo assembler creato da Maurice V. Wilkes dell'Università di Cambridge; prima di lui la programmazione di un elaboratore elettronico veniva eseguita interamente in binario. Wilkes, già progettista di EDSAC e autore, con Wheeler e Gui di **Preparation of Programs for Electronic Digital Computers**, pubblicato nel 1951, ha meritato il Turing Award della ACM. Il primo compilatore, invece, vede la luce l'anno successivo: si tratta di AO, ideato da Grace Murray Hopper (quella che inventò il termine **bug**) che, anziché sostituire codici mnemonici ad operazioni binarie come l'assembler, creava le istruzioni in binario a partire da un codice sorgente ad **alto livello**, in pratica disgiunto dalla particolare sintassi dell'elaboratore. Il primo linguaggio di programmazione universale, invece, viene creato nel 1957 da John Backus, dell'IBM. Il nome prescelto è FORTRAN, contrazione di FORmula TRANslator: si tratta di un compilatore sviluppato per fini prettamente matematici, ottimizzato e dalla notevole fortuna (ne esistono implementazioni attuali con blocchi di istruzioni e thread paralleli ed il FORTRAN/95 risulta tuttora più efficiente del C per alcuni scopi). A seguire viene creato ALGOL58 (ALGOrithmic Language), difficile parto seguito ad un incontro tra europei e nordamericani. Sempre nel 1958 John Mc Carthy del MIT, fondatore del dipartimento di intelligenza artificiale, crea il linguaggio di programmazione LISP (LISt Processor). LISP modifica la prospettiva della programmazione, spostandosi da un sistema (imperativo) di comandi ed istruzioni ad uno (dichiarativo) di oggetti e relazioni.

4.3 Presente e Futuro

Ovvero COBOL.³ Sviluppato nel 1960, il COBOL (COmmon Business-Oriented Language) risulta tuttora, ad oltre quarant'anni dalla sua presentazione, croce e delizia (poca) di un enorme novero di programmatori. Implementato inizialmente sugli elaboratori di classe mainframe, tale linguaggio subì infatti diverse modifiche nel 1974 e nel 1985, senza tuttavia perdere la compatibilità verso le release precedenti, divenendo così lo strumento di sviluppo preferito dalle grandi aziende, dalle banche e dalle assicurazioni grazie alla portabilità su piattaforme differenti, alla stabilità, all'interazione con monitor transazionali, database ed altri linguaggi ed anche alla strettissima collaborazione con IBM. Secondo linguaggio universale dopo il FORTRAN, provoca il rapido declino dell'ALGOL. Nel 1962 il canadese Kenneth Iverson crea il linguaggio APL (A Programming Language), ma la vera rivoluzione giunge due anni dopo: mentre IBM produce il linguaggio PL/1 (Programming Language - 1) che diverrà uno strumento di sviluppo delle serie 36 e 38, Thomas Kurtz e John Kemeney creano per i propri studenti del Dartmouth College il BASIC (Beginners All-purpose Symbolic Instruction Code). Iniziano anche le prime **guerre di religione**: mentre viene creato il codice ASCII (American Standard Code for Information Interchange) per facilitare la codifica delle informazioni tra differenti computer (normalizzata dall'ISO nel 1966), IBM continua ad utilizzare il proprio standard di codifica dati denominato EBCDIC (Extended Binary Coded Decimal Interchange Code). Nel 1966 viene creato LOGO, un linguaggio di istruzioni grafiche basato sul movimento di una tartaruga pilotata sullo schermo. Accolto tiepidamente negli ambienti scolastici, implementato su alcuni home computer, non ebbe mai uno sviluppo vasto negli ambienti accademici come invece il PASCAL, creato nel 1968 dall'austriaco Niklaus Wirth. Il PASCAL era un linguaggio in grado di utilizzare una sintassi a blocchi tipica delle forme BNF e coerente con il teorema di Iacopini-Bohm. Negli anni Settanta Thompson e Ritchie sviluppano il linguaggio B (derivato dal BCPL del Digital PDP) e nel 1978 Brian Kernighan scrive sul Linguaggio C (sviluppato da Ritchie sulla base del B) un testo che diverrà famoso come il **Kernighan&Ritchie**. Le caratteristiche del C sono una estrema efficienza del compilatore e la porta-

³Il panorama dei linguaggi di programmazione presenti oggi è vastissimo, ricordiamo PROLOG come successore di LISP Scheme come via di mezzo tra LISP e SMALLTALK, Eiffel altro linguaggio Object Oriented. Dai vari dialetti di Basic molto diffuso per la sua semplicità. PASCAL che ha avuto enorme successo negli anni ottanta come linguaggio didattico ad arrivare all'Object Pascal e Delphi-Kylix. In ultimo la numerosissima schiera di linguaggi script come PERL, Python, Javascript sotto la spinta dirompente data dalla presenza del WEB. I linguaggi sono molti ma la semantica dietro ad essi è rimasta la stessa oramai da molti anni e la sintassi generale è facilmente comprensibile da chi ne conosca almeno uno. Dal punto di vista delle metodologie di progettazione, dato oramai per obsoleto la rappresentazione tramite Diagramma di Flusso, usato come strumento didattico, si deve citare UML un sistema sviluppato da Rational per garantire un processo di sviluppo e documentazione coerente e completo

bilità dei sorgenti su piattaforme diverse. Nel 1972 gli stabilimenti Xerox di Palo Alto sviluppano l'ennesima meraviglia tecnologica: Alan Kay presenta SmallTalk, il primo linguaggio interamente ad oggetti della storia, creando un importantissimo precedente ed una piattaforma semantica sulla quale verranno sviluppati non solo altri linguaggi ad oggetti, ma anche una completa metodologia di progettazione del software. Nel 1979 il Pentagono sceglie ADA, un linguaggio sviluppato dal francese Jean Ichbiah per la Honeywell-Bull come unico linguaggio standard sostitutivo della pleora di linguaggi allora in uso. Nel 1983 Bjarne Stroustrup sviluppa una estensione ad oggetti per il linguaggio C, che chiama intuitivamente C++. Nonostante il timido tentativo di Objective-C, il C++ risulta ad oggi il linguaggio ad oggetti preferito dagli sviluppatori di progetti orientati all'oggetto; Stroustrup vince il Grace Myurray Hopper Award, proposto dalla Association for Computing Machinery (ACM), nel 1993.

5 Reti telematiche

All'interno di un computer esiste un sistema di comunicazione interno detto *bus*. Ma cosa accade nel momento in cui i computer hanno la necessità di scambiare informazioni tra loro? Bene, un esempio può rendere benissimo l'idea dell'argomento del quale si vuole parlare: se due individui hanno necessità di comunicare e non c'è aria tra loro (o la distanza è tale che le onde sonore verrebbero smorzate ben prima di raggiungere il ricevente) è sempre possibile frapporre una sorta di **amplificatore** o **ripetitore** del segnale che ne consenta la corretta trasmissione a destinazione. Lo stesso discorso è possibile applicarlo ai computer, ma occorrerà tener presente una serie di fattori che contribuiranno alla degradazione del segnale e dell'informazione. Ma andiamo con ordine, e vediamo come si inizia a parlare di reti telematiche.

5.1 Gli albori della trasmissione dati

L'antesignano degli strumenti per la trasmissione dati (se si eccettua il megafono di origine greca) è il telegrafo ottico. Sviluppato nel 1792 da un'idea dei fratelli Chappe, era costituito da una rete di proiettori lungo un percorso predefinito, che si scambiavano messaggi luminosi da stazione a stazione; ciascun proiettore era montato su di un supporto basculante, e poteva inviare messaggi in diverse direzioni, a seconda di dove era posizionata di volta in volta la stazione ricevente. Il sistema era arcaico e poco più valido dei segnali di fumo escogitati dai Pellerossa americani, ma rappresenta comunque una pietra miliare. Infatti tra il 1836 ed il 1838 gli inglesi Edward Davy, William Looke e Charles Wheatstone riescono a costruire un primo prototipo di telegrafo, mentre l'americano Samuel Finley Breese Morse, parimenti interessato al nuovo argomento dell'elettricità ed alla pittura di miniature, crea il codice di trasmissione telegrafica basato su linee e punti che ancora porta il suo nome. Ed è proprio Samuel Morse che, il 24 maggio del 1844, esegue il primo esperimento di trasmissione coronato da successo: l'invio del messaggio **What hath God wrought?** attraverso una distanza di 60 chilometri, tra Philadelphia e Washington. L'incremento delle reti telegrafiche procede spedito, ed in una decina d'anni verranno installati ben 37.000 chilometri di linee. Nel 1858 un transatlantico si occupò di tirare un cavo per le trasmissioni telegrafiche tra gli Stati Uniti e l'Europa. Dopo il collegamento, effettuato con successo, la linea darà modo di trasmettere per qualche giorno a fasi alterne, per poi cessare definitivamente di funzionare. Ma il secondo cavo di trasmissione transoceanica, installato nel 1866 con l'esperienza del precedente fallimento, continuerà a funzionare per oltre cento anni. Nell'ultimo quarto di secolo viene inventato il telefono. Sorvoliamo in questa sede i litigi per la concessione del brevetto di sfruttamento tra Meucci e Bell, ricordando solo che nell'occasione Bell fonderà i laboratori che ancor oggi tengono alto il suo nome: la Bell Telephone Company. Elettricità e

trasmissione telegrafica e telefonica accolgono quindi il nuovo secolo. Mentre matematici di alto livello come Nyquist e Shannon iniziano a studiare le funzioni di trasferimento dell'informazione ed i livelli di disturbo massimo accettabile per ciascuna tipologia di dato, la fine della Seconda Guerra Mondiale vede apparire il primo computer degno di questo nome.

5.2 Terminale: questo sconosciuto

Douglas Hofstadter, nel proprio libro **Gödel, Escher, Bach**, gioca un poco con la classica affermazione di autocoscienza cartesiana **Penso, dunque sono** (cogito, ergo sum), modificandola in **Penso dunque non ho accesso al livello in cui sommo**, indicando con tale affermazione che le funzioni neurologiche di calcolo logico matematico sono sviscerate dalla semantica del calcolo stesso, e non ho necessità di sapere quali neuroni siano attivi mentre eseguo una addizione. In altre parole potremmo dire oggi modernamente che ci presenta come progetto di un sistema operativo completamente astratto dall'hardware sul quale lavora. Ma una volta non era così: il computer era considerato un grande calcolatore di dati e funzioni, quanto più flessibile e programmabile, tanto meglio. Persino quando venne presentato il progetto di BINAC, il primo computer biprocessore, i problemi di condivisione di memoria e di bus di comunicazione interprocessore erano stati per il momento accantonati. Oggi invece il **cogito** cartesiano viene espresso con un bel **Penso dunque sono... connesso** da Derrick de Kerckhove, esprimendo da un lato implicitamente l'impossibilità per il singolo di riportare una visione realistica ed oggettiva dell'esistenza, dall'altra la necessità di percepire la realtà attraverso gli altri in una sorta di estremo baluardo contro le varie posizioni nichiliste. Curiosamente (ma neanche tanto, dal momento che comunque rappresenta l'evoluzione dinamica dell'espressione del pensiero umano) anche il computer attraversa queste fasi **filosofiche**, passando del concetto di **macchine per contare** a quello di **doganiere elettronico** a quello di **cellula** necessaria ma non sufficiente per acquisire autocoscienza. Vediamo come. Nel 1955 l'IBM realizza SABRE (Semi Automated Business Related Environment), un sistema di 1.200 telescriventi interconnesse attraverso linee telefoniche ed utilizzate all'interno degli Stati Uniti per la prenotazione voli della compagnia aerea American Airlines. SABRE, con tutti i successivi ammodernamenti, è perfettamente funzionante anche oggi, ed accoglie al proprio interno numerosissime altre compagnie aeree che possono così contare su un rodato e robusto sistema di prenotazione telematico già pronto. Due anni dopo, nel 1957, viene lanciato lo Sputnik, il primo satellite artificiale. I Sovietici avevano vinto la prima tappa della corsa allo Spazio, e gli Statunitensi ne erano rimasti sconvolti. Per questa ragione l'allora presidente Dwight D. Eisenhower crea ARPA (Advanced Research Project Agency) in seno al Department of Defense per definire un certo numero di progetti ed assicurare agli statunitensi una superiorità tecnica e scientifica nei confronti

dei propri avversari. Nel luglio del 1958 viene prodotto il primo bunker della rete di difesa SAGE. Un computer AN/FSQ7 (derivato dal WhirlWind del 1951) in ciascun bunker, è in grado di gestire 400 aerei simultaneamente. Il progetto ebbe lunga vita, e l'ultimo bunker SAGE verrà smantellato nel 1984. Nello stesso anno i laboratori Beil producono il primo apparato modulatore/demodulatore di dati binari su linea telefonica (modem). Ma i tempi sono maturi per uno sviluppo sempre più rapido: luglio 1961, Leonard Kleinrock del MIT presenta un primo studio teorico per la trasmissione dei dati a commutazione di pacchetto; ottobre 1962, J.C.R. Licklider viene nominato responsabile delle ricerche presso ARPA, allo scopo di consentire un migliore uso militare dell'informatica, grazie ad un suo articolo pubblicato ad agosto e relativo ad una **rete galattica** alla quale chiunque potesse collegarsi a programmi od informazioni, ovunque si trovasse. Tale concetto direte aperta verrà trasmesso ai suoi successori, Ivan Sutherland, Bob Taylor et Lawrence G. Roberts del MIT. 1964: Leonard Kleinrock del MIT cerca di convincere Lawrence G. Roberts della convenienza di utilizzare il sistema a commutazione di pacchetto anziché le linee dedicate per il progetto della rete. Tale convinzione giunge nel 1965, quando Lawrence G. Roberts e Thomas Merrill collegano tra loro un computer TX-2 in Massachusetts con un Q-32 in California utilizzando una linea telefonica ed un protocollo a commutazione di pacchetto. Due anni dopo Roberts, giunto ai vertici del progetto, pubblica uno studio intitolato Piani per la rete ARPANET. Nello stesso tempo Donald Davies e Robert Scantlebury del NPL e Paul Baran della RAND pubblicano anch'essi due documenti sull'importanza della trasmissione a commutazione di pacchetto per trasmissioni sicure in voce e per l'elevata resistenza e ridondanza del sistema contro disastri e catastrofi naturali ed artificiali (guerra nucleare su tutto). È da notare come gruppi differenti di scienziati che portano avanti progetti paralleli senza nulla sapere l'uno dell'altro giungano alle medesime conclusioni. Tra l'altro è a questo punto che nasce la leggenda metropolitana secondo la quale il gruppo di progetti ARPA era nato per garantire le comunicazioni in caso di distruzione da guerra nucleare.

5.3 ARPANET

Ad Agosto 1968 sono definite le specifiche di progetto della rete, viene lanciata la richiesta di finanziamento e cercate le compagnie che si occuperanno della produzione del commutatore di pacchetti IMP. Quattro mesi dopo iniziano i lavori: BBN installa il primo sistema direte basato su un mini-computer Honeywell 516 (da 12 Kbytes di RAM) presso l'UCLA, ed il primo computer (XDS SIGMA 7) vi è connesso. Un altro computer (XDS 940) presso lo Stanford Research Institute viene connesso su di una linea a 50 kbit/sec. Si tratta dei primi esemplari della rete ARPANET, subito seguiti da un IBM 360/75 all'Università di Santa Barbara ed un DEC PDP-10 presso la Utah University a Salt Lake City. Le cronache narrano che durante

la prima prova di connessione, l'utente doveva eseguire un login remoto: i due computer erano collegati attraverso i rispettivi IMP, mentre gli operatori erano al telefono per confermare ciascuna operazione; praticamente dopo l'handshake telefonico di L ed O, all'arrivo della G il sistema ricevente andò in crash. Gli anni successivi videro la creazione dell'interfaccia RS232 e dell'NCP (Network Control Program), tramite il quale numerosi elaboratori poterono collegarsi ad ARPANET. Nel marzo 1972 Ray Tomlinson della BBN realizza un primo sistema di posta elettronica per ARPANET, migliorato pochi mesi dopo da Roberts. Ad ottobre dello stesso anno, in occasione di una conferenza sulle comunicazioni a Washington, 40 terminali vennero posti online su di un IMP. Tale sfoggio di tecnologia non passò inosservato, e venne sottolineata la necessità di un comune protocollo di comunicazione; per approfondire il problema venne creato l'InterNetwork Working Group (INWG) con a capo Vinton Cerf. L'ARPA diviene DARPA e passa sotto il controllo della Difesa. Il successo del programma di posta elettronica genera la necessità di produrre mailing list elettroniche. Una delle prime mailing list apparse su DARPA fu SFLOVERS, una lista dedicata alla fantascienza... Gennaio 1973: 35 macchine sono collegate sotto DARPA, mentre viene sperimentato un primo collegamento via satellite per collegare sulla rete l'Università delle Hawaii. Bob Metcalfe, della Xerox, pubblica le specifiche per la connessione diretta Ethernet. 1974: la BBN lancia Telenet, la prima rete a commutazione di pacchetto commerciale. Due anni dopo AT&T e Bell sviluppano UUCP (Unix to Unix Copy Program), un protocollo di scambio dati largamente utilizzato prima dell'avvento del TCP/IP, allora allo studio del DoD. Darpa conta ormai 111 elaboratori. È in questo periodo che viene presentata X25, una normativa del CCITT (Comité Consultatif International Télégraphique et Téléphonique) che descrive l'interfacciamento di terminali su di una rete di comunicazione per pacchetti; normativa emessa con rapidità sorprendente onde evitare l'imposizione del sistema SNA (Systems Network Architecture) di IBM. Febbraio 1978: Ward Christanson e Randy Suess distribuiscono il primo programma di BBS, mentre Christanson sviluppa il protocollo Xmodem per il trasferimento di file binari. Il CCITT definisce la pila OSI. 1979: Bob Metcalfe lascia la Xerox e fonda la 3Com per commercializzare le schede di rete Ethernet; Comuserve lancia il suo primo servizio in linea per gli amanti della nuova microinformatica: MicroNET; Hayes produce un modem a 110/300 baud per l'Apple II, venduto a 380 dollari; al termine dell'anno nasce il primo sistema di news USENET grazie agli sforzi di Steve Bellovin, Tom Truscott, Jim Ellis, Dennis Rockwell e Stephen Daniels, ciascuno incaricato di un differente aspetto del progetto (peraltro puramente personale). Uno dei primi gruppi che videro la luce fu net.chess

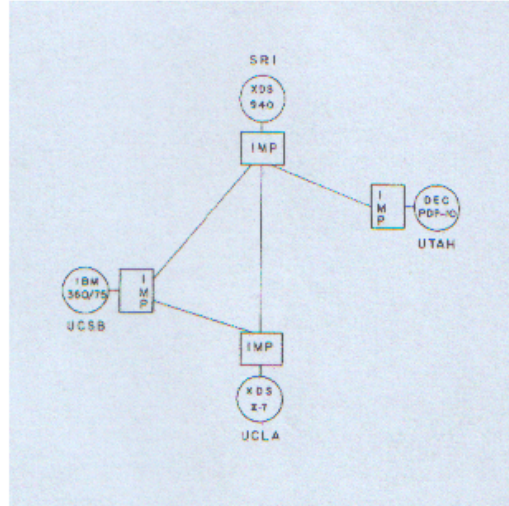


Figura 3: La prima ARPANET nel 1969

5.4 L'avvento di Internet

Agosto 1980: Vinton Cerf, dell'INWG, ora scienziato del DARPA, propone un piano di interconnessione (inter-network connection) tra le reti CSNET e ARPANET tramite protocollo TCP/IP; tale migrazione da NCP a TCP/IP sarà compiuta il primo gennaio 1983. È il punto di partenza della rete Internet come noi oggi la conosciamo. L'anno successivo vedrà 213 macchine collegate su Internet, che diverranno 562 ad agosto 1983. Sempre nel 1982 viene messa in opera la rete EUNET, che collegherà diversi sistemi europei e consentirà il trasferimento di email e di news anche sulle macchine del vecchio continente, i primi stati collegati: Olanda, Danimarca, Svezia ed Inghilterra. Giugno 1984: Tom Jennings produce il software FidoBBS, in grado di gestire una BBS sul proprio personal computer. In meno di sei mesi il programma viene distribuito ovunque, e decine di punti di connessione creano la rete mondiale Fidonet. Notare che tale rete si basa semplicemente sulla connessione tra computer attraverso linea telefonica, e nulla ha a che vedere con Internet, per il momento.

6 SuperComputers

Il concetto di supercomputer ha da sempre affascinato la fantasia degli esseri umani. Non è raro che durante la prima analisi di un progetto informatico il responsabile amministrativo, sovente digiuno di informatica, richieda **un programma tale che io preme un tasto e mi venga proposta la risposta migliore**, come se il computer stesso fosse dotato delle capacità di discernimento, astrazione e fantasia necessarie per un simile task. Di supercomputer è anche piena la storia della fantascienza: nomi del calibro di Fredric Brown, D.F. Jones, Harlan Ellison ed il citato Douglas Adams si sono almeno una volta invaghiti dell'idea proponendo un possibile scenario in cui un sistema acquisisce autocoscienza e prende il controllo di tutto ciò che lo circonda. Ma qual è la situazione attuale del calcolo automatico massiccio, e come si è giunti a questo stato?

6.1 Velocità e potenza

Innanzitutto tenteremo di precisare alcuni concetti sui quali regna sovrana una mistificazione non voluta: la velocità di calcolo, la potenza, l'efficienza e l'efficacia di un elaboratore. Un motore diesel è in grado di spostare una massa enorme, come quella di un transatlantico, mentre un motore a benzina molto meno potente riesce a portare una Ferrari in vetta al Campionato del Mondo di Formula 1, ed un motore elettrico rappresenta un esempio di rendimento meccanico senza confronti: quale è da preferire? La risposta è quanto mai scontata: dipende dal tipo di lavoro da eseguire! Allo stesso modo, abbiamo computer diesel, turbocompressi ed iperefficienti presi rispettivamente tra i mainframe, i computer paralleli e le reti di calcolo distribuite; cercheremo di illustrare la nascita e lo sviluppo di ciascuna di queste categorie, e mostriamo che spesso la differenza di efficienza di un sistema dipende anche dalla coppia interattiva composta dalla progettazione dell'algoritmo e dalla piattaforma sulla quale tale algoritmo viene implementato. In altre parole, mentre l'uso dell'Assembly al posto del Cobol può offrire una velocizzazione del 500% di un programma, la scelta di un algoritmo più efficace permette un guadagno anche di diecimila volte in termini di tempi di esecuzione. Nessuno mai affermerebbe che un Commodore 64 sia stato un supercomputer, né tantomeno che lo sia oggi, eppure un C64 che implementasse correttamente gli algoritmi di calcolo numerico sviluppati nell'ultimo quarto di secolo sarebbe almeno cinque volte più veloce dell'ultimo Pentium IV che utilizzasse i metodi di calcolo tradizionali. Per tornare al parallelo automobilistico, non è solo il motore che conta, occorre valutare anche le capacità del pilota. Poiché tuttavia una storia completa degli algoritmi di calcolo e del loro sviluppo nel tempo (e nello spazio) ci porterebbe forse a scrivere almeno una decina di puntate, ci si limiterà in questo caso a mostrare le possibili estensioni del

concetto di velocità, limitandoci a raccontare la storia dell'hardware creato appositamente per il calcolo veloce.

6.2 L'informatica di Lamarck

Jean-Baptiste Lamarck fu uno scienziato vissuto tra il 1744 ed il 1829; ebbe il grande vantaggio di nascere in Francia e vivere in pieno Illuminismo, assimilando ed approfondendo le numerose nuove idee che circolavano in quel periodo. Profondamente interessato alla Natura, divenne in breve tempo un esperto nel campo della tassonomia e dell'evoluzione animale, studiando tutto lo scibile dell'epoca ed approfondendo settori trascurati come gli invertebrati. Una delle affermazioni che gli valsero notorietà dichiarò che il tempo e le condizioni favorevoli sono i due mezzi principali attraverso i quali la natura si è industriata per dare vita alle proprie creazioni. Ciò lo portò ad annunciare che, spinto da un ambiente favorevole, dal tempo e da valide necessità interne, un organismo vivente produce in sé le mutazioni necessarie per avvantaggiarsi, trasmettendole poi ai propri discendenti. Mentre tale teoria doveva scontrarsi violentemente nel 1859 con l'ipotesi evoluzionistica di Charles Darwin, possiamo notare come essa si riconosca pienamente nel mondo dei supercomputer: tali macchine infatti, sviluppate soprattutto per eseguire calcoli scientifici intensivi (ambiente favorevole), cercano di incorporare le capacità migliori di ciascun settore (tempo e necessità interne): abbiamo i più veloci, i più avanzati tecnologicamente, i più costosi... quelli che suscitano maggiormente l'interesse e la curiosità! E quel che è più importante, ciascuna di queste macchine, nate normali e modificate per esigenze dell'ambiente circostante, trasmette poi ciascuna delle nuove caratteristiche assunte ai propri discendenti.

6.3 Dal biprocessore alla serie CDC

Due gusti sono meglio di uno, recitava una nota pubblicità. Tale affermazione doveva essere presente anche nelle menti di P. Eckert e J. Mauchly: come abbiamo visto il mese scorso, infatti, fu loro il progetto e la creazione nel 1949 del primo computer biprocessore, nato sul progetto EDVAC di von Neumann e fantasiosamente chiamato BINAC. Il progetto nacque su commessa della Marina Militare statunitense; ciascuno dei due processori eseguiva la medesima operazione in parallelo, e più che un raddoppio di velocità si tentò allora di raddoppiare l'affidabilità dei dati: un altro esempio in cui velocità e potenza non vanno di pari passo. D'altronde anche oggi, nei nostri personal computer, il raddoppio del numero dei processori non comporta necessariamente un raddoppio della velocità di esecuzione, in quanto il sistema operativo deve gestire un overhead per decidere a quale processore affidare una determinata istruzione, e anche per tale scelta è necessario tempo macchina... Saltiamo sei anni, ora. È il 1955 e Gene Amdahl sviluppa per

IBM il Modello 704, la prima macchina commerciale dotata di coprocessore matematico. Già, poiché sino ad allora i computer lavoravano su numeri interi, moduli e resti, mentre il Modello 704 è in grado di gestire sino a 5000 operazioni in virgola mobile al secondo. Un'altra notevole caratteristica di tale supercomputer è l'affidabilità: aveva un tempo medio tra due down di ben una settimana (meglio di molti PC attualmente utilizzati 50 anni dopo). Infine, una buona piattaforma consente un miglioramento degli algoritmi, e non è un caso che proprio sul 704 sia stato sviluppato il compilatore Fortran. Come Gene Amdahl, anche Seymour Cray lascerà il proprio nome inciso negli annali dell'informatica: mentre il primo creerà una società specializzata nella produzione di doni efficienti ed economici dei mainframe, la Amdahl appunto, il secondo produce nel 1958 il primo calcolatore commerciale interamente transistorizzato: un aumento quindi dell'efficienza, una diminuzione di calore prodotto, l'eliminazione di delicate valvole termoioniche, quindi la possibilità di lavorare su tensioni, correnti e frequenze molto più alte. Ad onor del vero occorre ricordare che il primo progetto di elaboratore interamente transistorizzato fu italiano, e prodotto nei laboratori ELEA-Olivetti, ma a questa storia (ed all'Informatica Italiana) dedicheremo una puntata a parte. Il 1958 vede anche la nascita del BULL Gamma 60, il primo supercalcolatore francese dedito al calcolo intensivo e dotato di supporto per il multi-thread in hardware. Multi-thread, ovvero la capacità di spezzettare un processo di elaborazione in diversi frammenti che possono venire eseguiti in parallelo; non si tratta, come per BINAC, di ridondanza di un medesimo processo, né della possibilità di lanciare programmi diversi su processori diversi: con il multi-thread viene radicalmente modificata l'architettura delle code di attesa dei programmi sia in ingresso che in uscita, ottimizzando l'uso delle diverse risorse di cui il computer dispone. Non un aumento netto di velocità nell'esecuzione di un algoritmo, bensì un aumento dell'efficienza globale nell'utilizzo del **tempo-macchina** e, in ultima analisi, un tempo inferiore per il completamento del processo. E a proposito di architetture, nel 1959 viene sviluppato ATLAS I, un sistema decisamente innovativo che introdurrà i concetti di memoria virtuale, multiprogrammazione e pipelining sulle unità di calcolo per interi e in virgola mobile. Con i suoi 200 kFLOPs (FLOP = FLoating point OPERations) di picco, supera di quaranta volte il Modello 704 di IBM. Appare un aumento di potenza pari ad un fattore dieci per ciascun anno trascorso. Per tutta risposta IBM progetta e produce nei laboratori di Los Alamos, il Modello 7030 Stretch (= allungato), che dispone di un processore due volte più rapido del Modello 704, ma soprattutto di un accesso asincrono alla memoria che diminuirà di ben sei volte i tempi di accesso. 1964: Seymour Cray colpisce ancora, e produce il CDC 6600, un supercomputer destinato ad un grande successo. Potenza: 3 MIPS (milioni di istruzioni per secondo), pari all'incirca ad un Intel 80386 prodotto vent'anni dopo. Preoccupata dalla produzione di Cray, IBM annuncia il proprio supercomputer IBM 90, un progetto inesistente nato esclusivamente per au-

mentare le aspettative dei propri clienti e sottrarre quote di mercato a Cray. La scorrettezza, come sappiamo, sarà ritentata al lancio del CDC 7600 da 20,83 MFLOPs nel 1969, ma stavolta verrà condannata. A titolo informativo possiamo dire che le macchine sviluppate da Cray, oltre ad avere una potenza di calcolo straordinaria, avevano anche un aspetto particolarmente piacevole e avveniristico. Il CDC 7600 in particolare sviluppa ed approfondisce i concetti di pipeline delle istruzioni, un sistema di gestione del microcodice che consente di riorganizzare ed elaborare più istruzioni nel medesimo ciclo di clock, affidandole a differenti unità di elaborazione (contatori di programma, gestori di salto, unità aritmetica intera, unità a virgola mobile, registri di cache): dotato di 9 CPU, 16 unità di controllo per periferiche, clock a 36 MHz, 180 Km di cavi ed un prezzo di 5,1 milioni di dollari USA di allora, rappresentava un oggetto da fantascienza. Tale architettura complessa ed efficiente venne per la prima volta inserita nel mondo dei processori Intel con l'introduzione della **programmazione in modalità protetta** del famoso 80286 (il cuore dell'IBM AT) e sviluppata completamente sui processori successivi: un ottimo esempio di ricaduta tecnologica e di ereditarietà delle modifiche strutturali sui sistemi informatici.

6.4 Gli ultimi supercomputers

Continuando il discorso sui supercomputer, dal primo bi-processore sino alla creazione di un'architettura che, valida per operazioni general purpose, consentiva guadagni notevoli in termini di potenza elaborativa. In questa prospettiva abbiamo potuto notare come, pur restando saldi in linea di massima, i concetti espressi da Von Neumann attraverso le specifiche EDVAC iniziano a mostrare i propri limiti. Durante questo cammino, infatti, il concetto di serializzazione delle informazioni è messo alla prova: memorie che fungono da tampone ad altre memorie, programmi e dati non sempre facilmente separabili, sistemi di predizione dei salti che ottimizzano il codice da eseguire rielaborando la sequenza di istruzioni prevista dal compilatore. Ma, come sempre, cerchiamo di andare con ordine. Per seguire al meglio la questione, tuttavia, abbiamo lasciato un po' in disparte i comprimari della corsa all'ultimo Hertz; ce ne occuperemo ora. Il 1965 vede l'uscita dell'elaboratore BESM6. Prodotto dalla società ITMiVT sotto il controllo di Sergei Alexeevich Lebedev. Questa macchina, che dispone di un processore a 48 bit che lavora a 9 MHz di frequenza di clock e di 192 Kbyte di memoria a nuclei di ferrite, è in grado di sviluppare la potenza di calcolo di 1 MIPS. Il BESM-6 sarà costruito in ben 350 esemplari, sino ai primi anni Ottanta, mentre l'ultimo esemplare verrà definitivamente smontato nel 1992. Un vero successo durato 37 anni. Nello stesso anno Burrough produrrà Illiac IV, il primo supercalcolatore ad architettura vettoriale che unisce un'architettura parallela e una pipeline composta di 64 processori (il progetto iniziale ne prevedeva ben 256). Potenza di calcolo di progetto: 200 MIPS! Purtroppo, trattandosi

della prima macchina vettoriale di rilievo a discostarsi decisamente dal sistema di calcolo basato sull'elaborazione seriale delle informazioni, Iliac IV ebbe notevoli problemi di messa a punto: nato da un progetto iniziato nel 1964-65, il primo elaboratore della serie venne installato presso la NASA nel 1972 ed iniziò a funzionare a regime solo nel 1975. Attorno al 1973 viene messo a punto Mio, un supercomputer sovietico nato sotto la direzione di Mikhail Kartsev. Si tratta di un sistema multiprocessore in grado di sviluppare una potenza di 20 o 30 MIPS. I dati relativi a questo sistema, con finalità militari e rimasto a lungo segreto, sono circondati dal mistero. Utilizzato all'interno della rete SPRN dedicata al controllo del lancio di missili nemici, per l'analisi dei dati satellitari e per calcolarne e seguirne le traiettorie via radar, è stato prodotto in decine di esemplari su un arco di 15 anni. Molti MIO sono tuttora funzionanti.

6.5 CDC cambia nome

Nel 1976, Cray Research Inc. presenta il capostipite di una lunga e famosa schiera di supercalcolatori ad architettura vettoriale: il Cray I. Il prototipo, che come al solito venne installato presso i laboratori di Los Alamos, era dotato di una potenza di picco di 133 MFLOP/s. Tre anni dopo dai sovietici viene prodotto il modello M13. L'ultima creatura di Mikhail Kartsev è basata su circuiti integrati di tipo LSI (Large Scale of Integration) e sviluppa una potenza tra i 50 ed i 200 MIPS. La corsa verso la potenza continua ed escono i modelli CYBER 203 e, di seguito, CYBER 205, che con memoria a 32 Mbyte e 200 MFLOP/s risulta il supercomputer più veloce della sua epoca. Almeno sino all'agosto del 1982, quando la Cray Research Inc. sviluppa il suo Cray X-MP, il primo supercomputer Cray multiprocessore. Questo mostro era in grado di lavorare con due o quattro processori alla velocità di 105 MHz, ciascuno dei quali era in grado di erogare una potenza di ben 235 MFLOP/s. A tale macchina erano associabili due dischi SSD, Solid State Disk, in realtà composti di 1 Gbyte di memoria RAM con una banda di 150 Mbyte/s rappresenterà il Cray più venduto: 189 esemplari distribuiti sino al 1988. Sempre Cray è il primo supercomputer a superare la potenza di un GFLOP/s (un miliardo di operazioni in virgola mobile al secondo); il Cray 2 possedeva 4 processori a 250 MHz ed era in grado di indirizzare sino a 4 Gbyte di memoria. Il sistema operativo che veniva montato sul Cray 2 era Unicos, una versione customizzata di UNIX System V. Un programma di moltiplicazione matriciale scritto per operare sui quattro processori faceva raggiungere al supercomputer la potenza di 1,7 GFLOP/s. L'unità centrale del Cray 2 era immersa in liquido conduttore di calore ed isolante elettrico (Fluorinert) che ne garantiva un corretto raffreddamento. Nel 1986 la Thinking Machines commercializza il primo supercomputer a parallelismo massiccio progettato su nuovi concetti: la Connection Machine CM-1 composta da 65536 processori. La macchina viene concepita un poco come

il cervello umano dal momento che ciascun processore effettua un lavoro ridotto. Il modo nel quale ciascun processore parla con gli altri è geniale: la macchina, infatti, per risolvere un problema dato riconfigura le connessioni interne tra i processori, proprio come accade con le sinapsi. Ovviamente, la contropartita è costituita dalla enorme difficoltà di programmazione, soprattutto riguardo all'ottimizzazione degli algoritmi: mentre infatti esiste oramai una quarantennale esperienza sulla stesura di algoritmi seriali, poco o nulla sinora è stato studiato per ottimizzare i problemi attraverso un approccio massicciamente parallelo. L'ultimo nato della famiglia Cray è il T932, una macchina vettoriale a 32 processori in grado di lavorare con un throughput di circa 56 GFLOP/s. Una menzione a parte merita il progetto APE, gestito dal fisico italiano Nicola Cabibbo per l'Istituto Nazionale di Fisica Nucleare nella prima metà degli anni Novanta: si tratta di un progetto nato per studiare i fenomeni di QCD (cromodinamica quantistica) per la simulazione di eventi caotici, movimenti atmosferici, fluidodinamica e così via. La macchina era topologicamente progettata come un ipercubo nel quale ogni processore era connesso direttamente con i suoi vicini; fisicamente composta di schede con 8 processori, ciascuna in grado di erogare la potenza di circa 40 MFLOP/s, la macchina era estremamente modulare e consentiva, anche in fase di costruzione, di effettuare i primi calcoli. La sua alta modularità le ha permesso, sia in fase di progetto che in fase di esecuzione, di raggiungere una potenza sino a 100 volte maggiore del più potente Cray dell'epoca. Mentre da una parte si studiano a fondo le possibili architetture hardware alternative per creare mostri di potenza sempre crescente in grado di sfruttare al meglio i sistemi a parallelismo massiccio, dall'altra iniziano ad apparire soluzioni assolutamente geniali dal punto di vista dell'ingegneria del software. Alcuni problemi vengono definiti **paralleli** perché non sono descritti da un algoritmo univoco, che abbia un solo punto di ingresso ed uno di uscita che fornisca la risposta: la scomposizione di un numero in fattori mediante il metodo delle curve ellittiche, ad esempio, prevede che un dato numero venga sottoposto ad una serie di operazioni fissate; se dopo essere stato controllato con 200 valori casuali diversi il numero non mostra fattori, si può affermare che tale numero non ha fattori inferiori ad un limite preimpostato. Allo stesso modo, esiste un metodo per calcolare esattamente la cifra binaria prescelta dall'espansione decimale di π attraverso la somma di una serie di valori calcolati in precedenza (algoritmo BBP, utilizzato nel progetto PiHex per calcolare la quadrilionesima cifra binaria di π). In questi casi il problema può facilmente venire scomposto in sottoproblemi semplici e dipendenti da un algoritmo efficiente; perché dunque non utilizzare il proprio PC e quello degli amici per trovare soluzioni a problemi ritenuti sinora difficili? Su questo concetto sono nati i primi esempi di calcolo distribuito su Internet: tale concetto consente di considerare ciascun PC come un processore collegato via rete a numerosissimi altri processori, gestiti magari da un server che centralizzi ed elabori i risultati di ciascuno. Il progetto GIMPS,

ad esempio, focalizzato sulla ricerca di numeri primi di Mersenne, possiede attualmente la potenza continua di oltre 105 Cray T932, pari a 5820 miliardi di operazioni in virgola mobile al secondo (GFLOP/s). Se invece non avete intenzione di mettervi alla prova con un sistema distribuito su Internet, niente paura: esiste oramai la possibilità di creare veri e propri supercomputer casalinghi. Basta un minimo di perizia con l'hardware, qualche capacità a livello di software di sistema ed il gioco è fatto! Un esempio per tutti è dato dal Prime Monster, presentato come capostipite di una nutrita schiera di Monster Farms : Prime Monster è composto di 12 Pentium IV 1,6 GHz con overclock a 2,3/2,4 GHz montati su altrettante motherboard e gestite diskless attraverso una scheda di rete tramite un server Linux. Potenza di lavoro equivalente: oltre 26 GHz, L per complessivi 30 GFLOP/s. Qualora siate più portati al risparmio, potreste sempre crearvi una piccola farm dedicata al calcolo parallelo basata su Athion: ogni nodo aggiunto verrebbe a costare meno di 250 euro, e con 5 nodi è già possibile raggiungere e superare i 10 GFLOP/s...

7 I computers della NASA

Le capacità scientifiche e narrative di Arthur C. Clarke rappresentano un indiscusso baluardo contro il quale, da decenni, si scontrano inutilmente generazioni di scrittori della New Age. Il genio visionario di Stanley Kubrick ne rappresenta un valido contraltare sotto il profilo eidetico-narrativo. Per questa ragione, l'incontro e l'unione delle capacità di costoro nella stesura della sceneggiatura e nella produzione del film citato nell'occhiello sono additate dalla massa della critica come il massimo evento del secolo relativo al cinema di fantascienza (e non). Il computer paranoide che parla con voce asettica e sicura rimane uno dei punti chiave della pellicola e nell'immaginario collettivo costituisce una sorta di pietra miliare, un termine di paragone relativo ad ogni altra rappresentazione del tipo. Leggende metropolitane riportano che Clarke e Kubrick abbiano scelto il nome HAL (che nella sceneggiatura costituisce acronimo di Heuristic Agoritmical) poiché risultava esattamente IBM traslato all'indietro di una lettera (one step ahead IBM): la verità era un'altra. Ma HAL 9000 era realmente così avanti per i propri tempi, o la NASA è riuscita in qualche modo a ricreare una parvenza del film nelle proprie astronavi?

7.1 La corsa allo spazio

Come alcuni ricorderanno, gli Anni Cinquanta e Sessanta del XX secolo hanno rappresentato un periodo particolarmente critico nella storia dell'Umanità: la Guerra Fredda contrapponeva due blocchi politici che attraverso campagne di disinformazione e propaganda tendevano a descrivere all'interno del proprio Paese le nefandezze dell'altro onde giustificare le scorrettezze politiche ed economiche compiute dalle rispettive Diplomazie. Praticamente la stessa situazione di oggi, solo che allora il mezzo per dimostrare superiorità sull'avversario non era l'informazione bensì il predominio scientifico e tecnologico, nell'ambito degli armamenti e dello spazio. Fu per questo che quando l'allora Unione Sovietica conquistò il primato lanciando in orbita lo Sputnik, gli Stati Uniti, seppur delusi, non furono presi del tutto in contropiede: l'Aeronautica aveva, infatti, già iniziato gli esperimenti con gli aerei orbitali, progetto mutuato dalla Germania nazista ed esportato in America da Von Braun. In contrapposizione ai vari F80, F86 (fighter), B25, B52 (Bomber) dell'Aviazione statunitense, un ufficio speciale iniziò, infatti, a progettare e produrre una serie di aerei-razzo la cui sigla iniziava per X (il progetto segreto) e che vide prototipi come X1, X3 ed alla fine X15 infrangere ad uno ad uno tutti i precedenti record di velocità e di altitudine. Ma nel 1958 il testimone della corsa allo spazio venne passato nelle mani della neonata NASA (non senza nervosismi in campo militare). Allora l'Ente spaziale americano aveva in dotazione UNIVAC, un prototipo di computer mainframe utilizzato soprattutto per eseguire elaborazione e con-

trollo di calcoli in batch. Ciò significava che alte pile di schede perforate, corrispondenti alle righe di codice di un programma, venivano inserite nell'elaboratore che le analizzava e nell'arco di qualche ora (o di qualche giorno) sputava fuori le risposte. Ma il sistema su quel pachiderma imponeva tempi di attesa notevoli, soprattutto perché, durante l'elaborazione, le periferiche di ingresso ed uscita non potevano funzionare; la NASA invece aveva bisogno di un sistema di controllo real-time per le proprie capsule spaziali e UNIVAC andava bene solo per calcolare accelerazioni, traiettorie ed orbite di rientro. Per questa ragione le prime capsule statunitensi che lasciarono l'atmosfera terrestre, nel progetto Mercury, non erano dotate di alcun computer. Il progetto Mercury, tuttavia, rappresentò per gli Stati Uniti il primo progetto spaziale di esplorazione del cosmo con esseri umani, seguito dai progetti Gemini, Apollo, Skylab e Shuttle. A ben vedere, le Mercury progettate dalla McDonnell-Douglas non erano dotate di nulla: avevano al massimo lo spazio per un passeggero e per i comandi principali di navigazione (per lo più razzi di assetto e radio), ma i voli erano stati programmati a terra dall'inizio alla fine, dipendevano in tutto e per tutto dall'accuratezza del sistema di guida del razzo vettore Atlas, e la presenza umana (senza nulla togliere all'eroismo di quei pionieri del cosmo) era sostanzialmente equiparabile a quella di un automa che esegue operazioni preordinate secondo una tabella di marcia dettata dal centro di controllo. Il rientro veniva calcolato in base ai computer a terra, gli impulsi per l'accensione dei razzi frenanti e di assetto erano inviati dal centro controllo alla capsula in volo, e non si ravvisava alcuna necessità di avere un computer a bordo.

7.2 Programmare da terra

Il progetto Gemini costituì invece un iniziale test di fattibilità per quanto riguardava le missioni lunari, sviluppando sofisticate tecniche di rendezvous improponibili se progettate da Terra. Le navicelle spaziali del progetto Mercury e Gemini sono molto simili, se si eccettua il fatto che la Gemini era biposto. L'unica altra differenza era un sistema di manovra orbitale posto nella parte posteriore della capsula, che poteva essere utilizzato dal nuovo paio di mani rappresentato dal secondo passeggero. Le capacità di manovra della Gemini permisero di imparare a padroneggiare una serie di manovre utili all'avvicinamento di oggetti posti in orbita, come Agena, un versatile veicolo spaziale della Lockheed nella fattispecie riempito di propellente liquido e posto in orbita in precedenza da un vettore Atlas. Ma un rendezvous valido richiedeva un accurato inserimento in orbita, una complessa manovra di inseguimento, movimenti precisi e capacità di pilotaggio una volta raggiunto il modulo Agena. Per raggiungere tali obiettivi venne progettato il computer digitale della Gemini. Tale sistema aveva sei funzionalità (programmi): prelancio, ascesa, inserimento, avvicinamento, rendezvous e rientro; venne prodotto proprio da IBM. Nel prelancio si occupava di eseguire i control-

li di self-check, e negli ultimi 150 minuti prima del lancio poteva ricevere i parametri necessari per il volo. Durante la salita teneva sotto controllo i valori di velocità e rotta prodotti dal razzo vettore per poter eventualmente prendere il controllo in caso di avaria del computer del Titan; anche se tali parametri non fossero stati utili per la guida, sarebbero risultati utili per il calcolo della velocità addizionale necessaria per raggiungere l'orbita al termine della spinta del secondo stadio del Titan. Ma le operazioni di rendezvous richiedevano un computer a bordo anche per altre ragioni. La rete di posizionamento terrestre, infatti, non riusciva allora a coprire tutti i settori percorsi dalla Gemini, ed in caso di blackout o di mancanza del footprint necessario sarebbe stato impossibile governare da terra la manovra di avvicinamento. Nella missione Gemini XI avvenne infatti l'avvicinamento al di fuori del controllo di telemetria da Terra, e nella medesima missione venne tentato un rientro completamente controllato da computer, che mancò il bersaglio di meno di tre miglia. Ma come si presentava il computer della Gemini? Innanzi tutto aveva tre sistemi d'interfaccia: i controlli del computer stesso, la Manual Data Insertion Unit (MDIU) e l'Incremental Velocity Indicator (IVI). I controlli consistevano in un deviatore a sette posizioni che permetteva di scegliere il tipo di programma in memoria, un tasto di start che lo metteva in funzione, una spia di malfunzionamento, una spia di attività che segnalava l'esecuzione del programma prescelto ed un interruttore di reset. La MDIU era costituita da una tastiera numerica a 10 tasti ed un registro con display a 7 cifre simile ad un odometro: le prime due cifre servivano ad indicare la locazione di memoria (erano disponibili 99 posizioni) mentre le ultime cinque cifre indicavano il dato. Un errore era rappresentato da una fila di zeri, mentre un numero negativo da un 9 iniziale. L'IVI infine mostrava gli incrementi di velocità richiesti o risultanti per una manovra di accelerazione: aveva un display a tre cifre tarato in piedi al secondo per ciascun asse di movimento. Niente a che vedere quindi con i moderni computer dotati di monitor o tastiera alfanumerica... ma la vera sorpresa deve ancora arrivare!

7.3 Progetto GEMINI

IBM riceve il contratto di fornitura del computer della Gemini, dell'ammontare di 26,6 milioni di dollari di allora, il 19 aprile 1962, e rilascia il primo prototipo il 31 agosto 1963. Lo spazio disponibile nella capsula è rappresentato da un vano dell'ampiezza di 48x36,83x32,38 cm mentre il peso previsto si aggira intorno ai 27 Kg. Il tempo di ciclo per una istruzione di addizione è di 0,14 secondi, il triplo per una moltiplicazione e quasi un secondo per una divisione; tutta l'aritmetica viene eseguita in virgola fissa. Alcuni computer a valvole (come il Whirlwind) erano più veloci, ma le valvole erano componenti fragili mentre le specifiche di progetto richiedevano la massima affidabilità. Per questo motivo si risolse di utilizzare nuclei di ferrite per

la memoria principale, poiché non volatile: il programma caricato restava in memoria anche a computer spento. Il sistema di memorizzazione della Gemini era costituito da 39 piani composti da una matrice di nuclei 64x64, corrispondente a 4,096 celle di memoria da 39 bit. Ciascuna word di 39 bit era suddivisa in 3 sillabe da 13 bit ed il piano di memoria in 18 settori. Un indirizzo completo doveva specificare settore e sillaba. Le istruzioni erano composte da 13 bit rilocabili, i dati dai rimanenti 26, contenuti esclusivamente nelle sillabe O ed 1. Le schede con i circuiti aritmetici e logici costituivano la maggior parte del computer della Gemini. A partire dalla missione Gemini VIII (missione che avrebbe quasi ucciso Nell' Armstrong, l'uomo destinato a muovere il primo passo sulla Luna), a causa dell'espansione dei programmi per i sistemi di navigazione e guida, venne aggiunto un sistema di memorizzazione secondario basato su nastro magnetico, sviluppato da IBM, sistema che ebbe successo sino alle missioni Skylab nel 1980: aveva una capacità di 1.170.000 bit che espandeva di sette volte e mezzo la capacità della memoria fisica. Il caricamento dei programmi presenti su nastro richiedeva sei minuti. Tale sistema veniva utilizzato per conservare i dati ed i programmi necessari per la seconda parte del volo. Poiché la NASA aveva particolarmente a cuore l'affidabilità (reliability) di ciascuna missione, IBM portò il tasso di errore di lettura da nastro, che era di 1 su 100.000, ad 1 su 1.000.000.000: triplicava ciascun bit da passare alla memoria, analizzava ciascuna tripletta di bit letti attraverso un circuito di votazione ed inviava alla memoria il risultato del voto di maggioranza, sistema utilizzato in seguito persino sullo Shuttle. Onde minimizzare problemi legati alla vibrazione dell'abitacolo vennero aggiunti pesi di zavorra sulle bobine del nastro. La preparazione in camera a vuoto eliminò infine i problemi verificatisi nella produzione di cortocircuiti tra i transistor, dovuti alle particelle di polvere invisibili che si spostavano ad alta velocità durante le accelerazioni improvvise. Nonostante l'aspetto da battaglia, questo hardware servì a 12 missioni in 20 mesi tra il 1965 ed il 1966 senza alcun inconveniente, se si eccettua un problema alla 48ma rivoluzione della Gemini IV nel quale l'astronauta James McDiwitt non riuscì a riprogrammare il computer per il rientro, optando per la manovra manuale. IBM peraltro non riuscì mai a riprodurre il problema a terra. Il software utilizzato presso il Centro di Controllo a terra era una derivazione di SAGE (Semi-Automatic Ground Environment) a sua volta mutuato dal sistema operativo IBM 360, ma come si programmavano i 150.000 bit disponibili sulla Gemini? Ovviamente in puro assembler, poiché l'unica alternativa possibile di allora, il FORTRAN, produceva codice poco ottimizzato per le necessità stringenti della NASA. Gli esperti svilupparono un ristretto set di 16 istruzioni appositamente per il computer della Gemini. In quel periodo la programmazione era considerata un'arte più che una scienza, ed argomentazioni di documentazione e metodologia sarebbero apparse solo dopo il 1968, quando i programmatori avevano oramai accumulato 15 anni di esperienza, sudore e lacrime. Ad ogni modo il software prodotto doveva essere modu-

lare nello sviluppo, in modo da poter essere integrato e riutilizzato in piú di una missione, ma monolitico nel caricamento. Nel progetto Gemini le richieste di modifica ed i tentativi di ridurre i programmi all'interno dello spazio richiesto provocarono numerose successive versioni del software; tali versioni presero il nome di **Gemini Math Flow**. Ad esempio il Math Flow One racchiudeva le procedure di ascesa, avvicinamento, rendezvous e rientro; la versione due inserí anche navigazione orbitale e inizializzazione per il rientro, ma superó i limiti di capacità di memoria e il reparto programmazione del progetto cancelló l'aggiunta. Nel Math Flow Four venne aggiunta con successo l'inizializzazione del rientro, ma il programma utilizzava 12.150 delle 12.288 word disponibili. Tale prodotto sarebbe dovuto partire con la Gemini IV, ma la decisione della NASA di optare per un angolo di rientro costante anziché proporzionale cambió la logica di guida. Math Flow Ave incorporó tale modifica, ma riempí nuovamente la memoria e venne rimpiazzato da un Math Flow Three modificato sui voli III e IV. La versione finale, Math Flow Seven, venne utilizzata con successo sulle navicelle dalla VIII alla XII... ma solo perché erano state fornite di memoria di massa a nastro; era composto di sei sottoprogrammi con nove modalità operative: Executor, Prelaunch, Ascent, Catch-Up, Rendezvous e Reentry. Tali programmi vennero ampiamente testati sia attraverso simulatori scritti in FORTRAN che attraverso esemplari di computer di preserie a Terra. Le procedure di documentazione e testing vennero assemblate ed assimilate man mano che se ne presentava l'occasione, creando un valido precedente per le successive missioni, Apollo su tutte.

7.4 Progetto Apollo

La Luna, il satellite naturale della Terra, orbita ad una distanza media di 384.000 Km dalla superficie del nostro pianeta. Complice incolpevole di migliaia di storie d'amore, ha rappresentato da sempre il luogo piú remoto ed inaccessibile dell'immaginario collettivo: fu puntando ad essa che l'ex-presidente degli Stati Uniti John Kennedy volle dimostrare la supremazia americana nella corsa allo spazio, una corsa iniziata come tutti sanno al secondo posto dietro ai Sovietici. Raggiungere la Luna con un equipaggio umano rappresentava infatti una svolta storica nel gran libro delle esplorazioni, ed al tempo della dichiarazione di quest'intento, nessuno o quasi osava anche solo pensare al suo compimento. Sbarcare sulla Luna con i mezzi di allora fu un'impresa eroica e quasi impossibile, tanto che diversi detrattori dubitarono del fatto. Bili Kaysing, ex direttore delle pubblicazioni tecniche presso i laboratori della Rocketdyne Research, la ditta che progettó e costruí i motori dei razzi che apparentemente portarono le navicelle Apollo sulla Luna, ha scritto un libro il cui emblematico titolo recita **Non siamo mai andati sulla Luna** e riporta come prove una serie di informazioni significative basate sulle immagini fotografiche prodotte dagli astronauti delle prime missioni

Apollo. Se tutto ciò appare strano e fantascientifico (chi non torna con la memoria al 1978, con il film di Peter Hyams intitolato **Capricorn One**) occorre ricordare le immense pressioni politiche ed economiche alle quali era sottoposto l'Ente Spaziale Americano, la burocrazia strisciante in ciascuna gara di appalto, l'urgenza di attuare la promessa di Kennedy entro il decennio, la corsa contro i russi... Ad ogni modo occorre ricordare che sul suolo lunare è stato deposto uno strumento necessario per controllare l'effettiva distanza Terra-Luna attraverso un fascio laser, e che la NASA ha messo a disposizione del pubblico una documentazione della missione, completa di allegati relativi ai diversi sistemi di guida e navigazione. Ed è proprio di questa documentazione che parleremo nella puntata odierna di Storia dell'informatica. Una delle prime domande che ci si pone pensando alla missione Apollo è la seguente: perché mai non si è effettuato un lancio diretto dalla Terra alla Luna, scegliendo invece il sistema delle orbite di parcheggio? La risposta è semplice: non esisteva una tecnologia in grado di erogare la potenza necessaria per un lancio diretto, né la capacità di gestire un lancio verso un punto di atterraggio posto su di un bersaglio a quasi 400.000 chilometri di distanza ed in movimento; si preferì invece lanciare un modulo di comando e di servizio verso la Luna sfruttando la minima attrazione gravitazionale ed immetterlo in orbita (manovra che consente ampie correzioni anche durante il tragitto), lasciando al modulo lunare il compito di procedere poi all'allunaggio. Per riuscire a gestire al meglio questo progetto, si decise di dotare il modulo di comando (CM) ed il modulo di allunaggio (LEM) di sistemi di elaborazione identici, chiamati Primary Guidance, Navigation and Control Systems o, più familiarmente, PGNCS (pronunciato pings), dotati di software differente per ciascun modulo. Il modulo lunare aveva un sistema di elaborazione aggiuntivo chiamato AGS (Abort Guidance System) che consentiva di portare in salvo l'equipaggio in caso di malfunzionamento del sistema principale. I sistemi di controllo NASA avevano un sistema di backup del PGNCS per poter intervenire rapidamente in caso di malfunzionamento del modulo di comando attraverso un sistema di telemetria gestito da Terra. In caso di necessità, una volta abortita la missione, il PGNCS del LEM era in grado di controllare le operazioni svolte dal PGNCS del modulo di comando. Una rete di computer creata dalla NASA intorno al globo (NASCOM) consentiva di gestire telecomunicazioni e telemetria con la navicella; la rete era composta di linee dedicate, canali satellitari e stazioni radio ad alta frequenza. I dati venivano controllati ogni frazione di secondo, ed eventualmente veniva richiesto un uplink verso l'astronave in UHF alla velocità di 1200 bit al secondo mentre Houston raccoglieva i dati dei diversi centri radio della rete così costituita a 2.400 bit al secondo. Il Goddard Space Flight Center ed il Mission Control Center erano dotati di un collegamento per trasmissione dati a 50.000 bit al secondo duplicato per ragioni di ridondanza e prevenzione di overflow. Erano disponibili infine per le comunicazioni due satelliti Intelsat posizionati sull'Atlantico e sul Pacifico. Un simile sistema di elaborazione e

comunicazione avrebbe così garantito diverse funzioni: assicurare la qualità delle linee di trasmissione, verificare l'accuratezza dei messaggi attraverso controlli ridondanti, mantenere aggiornati i sistemi di monitoraggio del volo. I sensori inseriti nell'astronave tenevano sotto controllo la temperatura della cabina, la pressione, le informazioni fisiche relative agli astronauti come battito cardiaco e respirazione: tali dati erano trasmessi alle stazioni di terra al ritmo di 51.200 bit al secondo. I computer del Mission Control Center, infine, riscontravano e selezionavano modifiche o deviazioni, confrontandole con i programmi a disposizione, ed indicavano le aree di problema o i dati necessari ai controllori di volo; inoltre distribuivano le schermate informative al personale di missione, assemblavano i dati elaborati in formati leggibili, registravano i dati su nastro magnetico per i controllori di volo e mantenevano allineate le temporizzazioni di sistema.

8 Evoluzione dell'hardware

Il progetto Apollo diede ai progettisti di hardware della NASA quella che spesso viene chiamata una seconda possibilità. Nato inizialmente come evoluzione del sistema di guida dei missili Polaris, l'hardware si evolse in fretta passando dal cosiddetto vecchio Block I al Block II. Harry i. Goett propose alla NASA un computer general purpose dal volume inferiore ai sette litri, con word di 24 bit e 80 watt di consumo, ma tale macchina, dotata di una logica a transistor, aveva diversi svantaggi: l'impossibilità di eseguire complementazioni, la lettura distruttiva della memoria ed il memory-cycle time non fisso. Nel 1962 venne quindi eseguito uno studio sul possibile utilizzo dei circuiti integrati, tecnologia che allora aveva compiuto appena tre anni, H team del MIT delegato all'analisi scelse di utilizzare porte NOR DCTL (direct-coupled transistor logic) dotate di elementi a tre input consistenti in tre transistor e quattro resistenze. Con cinquemila elementi del genere venne costruito il computer della missione Apollo. Certo, utilizzare una maggiore varietà di elementi avrebbe semplificato il progetto diminuendo grandemente il numero dei componenti, ma utilizzando i soli NOR venne preferita la semplicità e l'affidabilità dell'insieme. In tal modo anche il tempo di ciclo della macchina venne fissato ad 11,7 millisecondi. Il MIT scelse i circuiti integrati di tipo NOR nell'autunno del 1962, e nell'estate del 1963 il 60% della produzione statunitense di tali microcircuiti era utilizzata nel progetto Apollo. Era così stato definito il disegno di base del computer, dal consumo di 70 watt a 28 volt, e ne erano state predisposte due unità: una per il modulo di comando ed una per il LEM. L'equipaggio poteva comunicare con il computer attraverso unità display e tastiera (keyboard) da cui il nomignolo DSKY (pronunciato disk). A dimostrare l'importanza di tale componente, tanto da dare a DSKY il titolo di **quarto membro dell'equipaggio**, basti pensare che per completare una missione lunare sarebbe stato necessario premere ben

10.500 tasti. Il display era dotato di 10 spie, una segnalazione di computer busy, un PROGRAM display, VERB e NOUN display, tre indicatori numerici a cinque cifre piú segno e 19 tasti con le cifre 0-9 e le funzioni VERB, NOUN, CLEAR, KEY RELEASE, PROCEED, RESET, ENTER, PLUS, MINUS. Vi erano due DSKY nel modulo di comando ed uno nel LEM. Ad un'idea iniziale di parola a 24 bit ne seguí una di parola a 16 bit per questioni di rapiditá di calcolo e di efficienza. La singola precisione era data da 14 bit, con gli ulteriori due bit utilizzati per il segno e per il controllo della paritá, mentre due o tre word garantivano doppia o tripla precisione. Uno schema di **bank register** permetteva di gestire la memoria (chi ricorda il Commodore 128?). L'utilizzo di bank register consentiva l'indirizzamento di tutta la memoria della macchina. La storia della memoria dell'Apollo è costellata di modifiche ed accrescimenti in corso d'opera. Le richieste iniziali del MIT erano di 4K word di memoria fissa e 256 word di memoria dinamica cancellabile, ma già a giugno del 1963 erano richiesti 10K di memoria fissa ed 1K di memoria cancellabile; le richieste crebbero ulteriormente, sino ad assestarsi a 36K word per la memoria fissa e 2K word per quella cancellabile. Ovviamente, se i progettisti hardware avessero conosciuto in anticipo le richieste dei produttori del software, avrebbero scelto anche una word da 24 bit in grado di indirizzare tranquillamente tutta la memoria disponibile senza problemi di bank register shifting; occorre tuttavia sottolineare che in quel caso la storia insegnó qualcosa al team, creando i presupposti per lo sviluppo di metodologie utili all'analisi, alla creazione ed alla gestione di progetti. La composizione finale della memoria risultó standard per il componente riscrivibile, in quanto vennero utilizzati i classici nuclei di ferrite simili a quelli usati nel Progetto Gemini, ma unica per quanto riguarda la memoria fissa. Per quest'ultima venne scelto il sistema core rope, un tipo di memoria di sola lettura ad alta densitá che utilizzava i nuclei di ferrite su di un progetto completamente differente. Il MIT utilizzó tale sistema nell'analisi del computer della prima sonda per Marte, e lo trasferí sull'Apollo. La memoria core rope ha il vantaggio di poter mantenere piú informazioni in uno spazio piú ristretto, ma è particolarmente difficile da produrre, ed i dati inseriti in fabbrica non possono piú essere modificati.

8.1 ...e il software

L'uso di memoria core rope fu una croce per gli sviluppatori del software della NASA. Il software da codificare nella core rope doveva infatti essere consegnato mesi prima della partenza della missione per consentire la corretta creazione della memoria e soprattutto il testing necessario. Purtroppo, una volta prodotta, la memoria non era modificabile in modo semplice, poiché ciascun modulo sigillato richiedeva di essere completamente ricablato anche solo per cambiare un bit. Il software doveva non solo venire consegnato per tempo, ma anche essere del tutto esente da banchi. Sebbene il buon senso

mostri come sia vantaggioso completare per tempo qualcosa di così complesso ed importante come il software relativo al computer di una missione spaziale molto prima dell'inizio della missione stessa per consentirne l'uso nei simulatori, difficilmente il software risulta consegnato nei termini o perfettamente funzionante. Fortunatamente (per il Programma Apollo), la natura della core rope mise sotto pressione i programmatori del MIT, tanto da far loro consegnare tutto subito e per bene. È evidente che il concetto di software bug-free era lontano dalla concezione di quei tempi: la programmazione era un procedimento iterativo di rimozione degli errori, e nonostante ciò diversi banchi giungevano felicemente a destinazione nel prodotto finale a causa di tecniche di testing assolutamente non all'altezza del loro compito. Nel caso specifico si riuscì a produrre core rope esenti da difetti utilizzando come test un nastro magnetico contenente la procedura e le risposte previste. La NASA era comunque al corrente delle continue preoccupazioni relative allo sviluppo del software per la missione Apollo. Per tale ragione il 18 dicembre 1967 venne istituita la Guidance Software Task Force, alla ricerca di nuovi sistemi per migliorare sviluppo e verifica dei programmi. Un caricamento classico del software comprendeva una quarantina di programmi e trenta routine simultanee. Cambi di programma e richieste diverse richiedevano l'uso della procedura verb-noun. Chi è abituato alla tastiera odierna noterà senz'ombra di dubbio la completa mancanza di tasti alfabetici: mentre la maggior parte dei comandi per computer vengono inseriti digitandone il nome, la lista dei comandi del computer dell'Apollo specifica coppie sostantivo-verbo. Erano disponibili 100 numeri a 2 cifre per programma, verbo e sostantivo, e la maggior parte di essi veniva utilizzata ad ogni missione. Esempi di coppie verb-noun sono **display velocity** e **load angle**. Se ad esempio l'equipaggio aveva intenzione di eseguire il rendezvous targeting program, occorreva premere il tasto VERB seguito dalle cifre 3 e 7, quindi il tasto ENTER per inviare al computer una richiesta di program change, quindi premere 3, 1, Enter per richiedere il programma P31. Dal programma l'equipaggio poteva poi richiedere gli angoli di manovra (VERB 50, NOUN 18), monitorare i cambiamenti durante la manovra (VERB 06, NOUN 18), o richiedere il cambio di velocità necessario per la manovra successiva (VERB 06, NOUN 84). Difficile? Niente affatto: per ogni missione venivano fornite diverse serie di raccoglitori ad anelli con schede che permettevano di controllare tutte le possibili combinazioni, oltre al catalogo stellare, alla lista dei VERB, dei NOUN, ai codici di allarme, alla gestione e risoluzione degli errori. Nonostante le 100 coppie VERB-NOUN, la settantina di programmi differenti e l'interfaccia utente molto limitata che alternava decimale ed ottale e lampeggiava per richiedere l'attenzione, secondo la maggior parte degli astronauti il computer di Apollo era semplice da usare; come per altri aspetti delle missioni spaziali, ore di simulatore permettono di lavorare sul computer di bordo in modo naturale, reagendo all'errore ancor prima di aver terminato una sequenza errata. Ciò non toglie che a gran voce venne

richiesta (ed ottenuta) una mini-lista di comandi e programmi che potevano essere utilizzati senza dover ogni volta confermare l'immissione

9 Il progetto SHUTTLE

Lo Space Shuttle, la navetta spaziale ideata tenendo a mente i criteri di economicità e riusabilità tanto cari alla NASA è l'idea di un orbiter riutilizzabile già presente in alcuni progetti tedeschi datati intorno alla fine della Seconda Guerra Mondiale che mostrano un veicolo portato in quota da un bombardiere che parte con propulsione a razzo e torna alla base dopo aver compiuto il proprio compito. Ciò che la navetta tedesca non prevedeva era invece il sistema integrato e ridondante di computer di cui lo Shuttle venne dotato. Oggi la navetta spaziale fa bella mostra di sé e crea una tessera di notevole importanza nella storia della conquista dello spazio e nell'immaginario collettivo: numerosi film di fantascienza attuali ne prevedono la presenza, da Armageddon a Space vampires, da The Astronaut's wife al visionario 2001: A space odyssey, sino a Space cowboys. In quest'ultimo, in particolare, vengono mostrate sequenze interessanti in cui si parla dell'avionica dello Shuttle, mutuata da quella di B52 e di F16, e della caratteristica di richiedere un allenamento per lancio e permanenza nello spazio drasticamente minore rispetto al Progetto Apollo: si avvicina in sostanza l'epoca in cui lo Shuttle verrà utilizzato come mezzo di trasporto orbitale alla stessa stregua di un normale aviogetto di linea, o quasi.

9.1 L'hardware dello SHUTTLE: Ridondanza

La ricerca per il progetto STS iniziò nei tardi Anni Sessanta, prima che il primo uomo scendesse sulla Luna. La NASA cercava uno strumento in cui i componenti costosi (motori, razzi a propellente solido e l'orbiter) potessero essere riutilizzati, mentre il componente sacrificabile era rappresentato dal serbatoio esterno di carburante. L'orbiter avrebbe dovuto planare durante il rientro senza l'ausilio di motori. Allo stesso modo il sistema di computer subì diverse modifiche, e venne alla fine affidato al team Rockwell/IBM, con parte dei tecnici al lavoro dotati dell'esperienza acquisita durante le missioni Gemini ed Apollo. Esistevano due aspetti del problema di progettazione del computer: quello funzionale e quello dei componenti. I progetti precedenti utilizzavano il computer esclusivamente per guida, navigazione e controllo di assetto, ma numerosi fattori nella progettazione dell'astronave portarono a crescere il numero delle funzioni richieste. Gli studi effettuati suggerivano tre differenti approcci alla risoluzione del problema: il primo assegnava un piccolo computer specializzato a ciascuna funzione, distribuendo i processi in modo che il malfunzionamento di un solo computer non mettesse in pericolo gli altri sistemi di controllo della nave; il secondo proponeva un computer

centralizzato con capacità di time-sharing, estendendo i concetti già sperimentati per Gemini ed Apollo. L'ultima variante prevedeva diversi processori utilizzanti una memoria comune, praticamente la fusione delle due idee precedenti. È inutile sottolineare che nel 1971 erano già stati vagliati quattro sistemi multiprocessore per quest'ultima variante. Per quanto riguardava la capacità elaborativa, si valutò di progettare un sistema con un processore almeno del 50% - 100% più potente rispetto alle esigenze attuali, consci della crescita esponenziale avuta dai progetti precedenti nello sviluppo del software. Una ulteriore modifica rispetto ai progetti precedenti imponeva l'utilizzo di aritmetica floating-point. I computer precedenti infatti adottavano una aritmetica a virgola fissa, per cui lo scaling dei calcoli doveva essere scritta nel software, ed era stato appurato che i computer di Apollo avevano richiesto almeno il 30

9.2 Vecchia, vecchissima, anzi attualissima

Oramai rimpiazzata da memorie a semiconduttore su circuiti integrati, la memoria a nuclei di ferrite rappresentava ancora il non plus ultra nelle applicazioni militari ed aeronautiche a causa del tempo di caricamento nullo dei programmi. In più iniziò a farsi luce negli ambienti di progettazione il concetto di bus, per lo spostamento di grandi quantità di dati su apparecchi militari sempre più dipendenti dall'elettronica. In aggiunta occorre ricordare che, dal momento che la ridondanza era una delle parole chiave del progetto, era necessario consentire a più computer presenti sulla navetta di parlare tra loro, consentendo in questa maniera un sistema di votazione il più rapido ed efficiente possibile. I primi studi per lo Shuttle prevedevano il concetto di fail operational/fail safe: un problema e la missione poteva continuare, due problemi e la missione doveva considerarsi abortita, dal momento che un eventuale terzo problema avrebbe ridotto la ridondanza a sole tre macchine, il minimo necessario per la procedura di votazione. In un progetto del 1970 un computer special-purpose gestiva le funzioni di controllo di volo (il fly-by-wire) mentre un altro computer general-purpose si occupava di guida, navigazione e funzioni di gestione dei dati. Questi due computer avevano gemelli e l'intero gruppo di quattro era duplicato per offrire il sistema di ridondanza richiesto. Nel 1971 vennero proposti due nuovi progetti: uno più economico, l'altro più ambizioso. Il primo utilizzava due gruppi di AGC (Automatic Guidance Control) contenenti 32K di memoria cancellabile e memoria di massa su nastro magnetico anziché le **core ropes** dell'Apollo, mentre l'altro, ambizioso, proponeva un sistema di computer **collaborativi** che offrivano scarsa potenza elaborativa individuale. Il cuore del sistema era composto da un sistema multiprocessing relativamente ampio con processori locali a livello di sottosistema per isolare il computer centrale dalle eventuali modifiche dei sottosistemi.

9.3 Il sistema definitivo

Contrariamente ai progetti Gemini ed Apollo, la NASA desiderava un computer completo per lo Shuttle. Cinque sistemi con qualifica **space** superiore alle specifiche militari erano a quel tempo disponibili: l'IBM 4Pi, l'Autonetics D232, l'Alpha della Control Data Corporation, il Raytheon RAC-251 e l'Honeywell HDC-701. Il profilo minimo per i sistemi computerizzati si era evoluto includendo lunghezza di parola a 32 bit per calcoli piú accurati, un minimo di 64K di memoria e possibilità di microprogrammazione. I **micro-programmi** venivano chiamati firmware, e contenevano software di controllo altrimenti realizzato in hardware; il firmware aveva una maggiore versatilità in caso di modifica delle specifiche di progetto. Alla fine fu scelto il sistema IBM 4Pi a causa della sua storia ed architettura: già utilizzato nelle applicazioni aeronautiche, aveva a che fare con i sistemi montati sullo Skylab e faceva parte della famiglia architetturale dei mainframe IBM 360. A causa della similitudine del set di istruzioni di AP-1 e del Sistema 360, i programmatori esperti del 360 erano praticamente pronti a lavorare sul progetto a costo di un retraining minimo. In piú erano già disponibili numerosi tool di sviluppo per l'AP-1 sul 360. Infine, dal momento che lo Shuttle non avrebbe portato a bordo alcun tool di sviluppo né compilatore, era piú conveniente eseguire i test del software a terra, dove esistevano simulatori AP-1 che giravano sui 360. Una ulteriore caratteristica distintiva relativa ai computer del Progetto Shuttle rispetto a Gemini ed Apollo risiede nel linguaggio di programmazione: mentre infatti nelle precedenti missioni venne scelto di codificare i programmi in Assembly per avere sotto controllo la gestione di processore e registri in modo efficiente, per una questione di rapidità di sviluppo venne commissionato un linguaggio di programmazione ad alto livello che portò i costi di produzione del software ad un calo del 10%-15%. Curiosamente, il linguaggio di programmazione ad alto livello prodotto venne sviluppato dalla Intermetrics di Cambridge e chiamato HAL/S: esso supportava aritmetica vettoriale e task schedulati secondo un sistema di priorità definito dall'utente. Non è tuttora chiara la ragione della scelta del nome del linguaggio: il film di Kubrik, **2001: odissea nello spazio** era stato presentato nei cinema piú o meno nel periodo di definizione delle specifiche di progetto. Ovviamente la NASA, come abbiamo visto in un'altra puntata, nega qualsiasi relazione tra il nome del compilatore e HAL 9000 presentato nel film; John R. Garman, del Johnson Space Center, uno dei principali responsabili dello sviluppo di HAL/S, dichiarò una volta che l'origine del nome possa essere stata mutuata da un individuo incaricato del primo sviluppo, il cui nome era per l'appunto HaI. Altri suggeriscono che sia l'acronimo di Higher Avionics Language (gli Stati Uniti sono la terra degli acronimi, si narra che per evitare inutili e umoristiche esagerazioni abbiano sviluppato l'AAAA, Association Against Acronym Abuse...).

10 Uomini e macchine

La proposta di utilizzare HAL incontró una notevole opposizione da parte di coloro che, oramai avvezzi a programmare in linguaggio macchina, ritenevano piú lento, poco ottimizzato (quando non addirittura immorale) il codice prodotto automaticamente da una macchina. Oltre, naturalmente, ad essere prони ad errori di progettazione del sistema di codifica. In breve i compilatori non producevano codice bene come gli esseri umani: lo facevano semplicemente in modo piú veloce ed accurato, ma senza quei guizzi di genio caratteristici della codifica manuale. Per arginare eventuali critiche, Richard Parten, responsabile della Divisione Software dell'astronave, richiese una serie di benchmark: da una parte mise i migliori programmatori assembler di IBM, dall'altra chiese di produrre le medesime funzioni attraverso una codifica in HAL; fece poi girare le due batterie di programmi l'una contro l'altra, ottenendo una differenza di efficienza del 10% - 15%. La storia però non ci tramanda quale sezione fosse risultata maggiormente efficace...