

## Una breve e sintetica storia dell'informatica

### Da Eniac a Linux

*Un percorso militare, scientifico e infine industriale*

#### **QUARTA PUNTATA: LE SCHEDE DI CONTROLLO E L'ULTIMA FASE DEI COMPUTER DI SECONDA GENERAZIONE**

Nel 1960, nell'anno della realizzazione del progetto NORAD, i computer venivano utilizzati per ben trecento funzioni differenti, quindi esistevano altrettante tipologie di applicazioni.

In quello stesso anno il numero di computer in funzione supera le seimila unità, mentre nel 1955 se ne potevano contare appena 250.

Stiamo descrivendo di una rivoluzione, anche sociale, giacché se dietro i computer di prima generazione gli addetti erano appena qualche centinaio, ora il numero di addetti si avvicina alle ventimila – trentamila unità.

Questo sviluppo fu dovuto all'elaborazione di linguaggi di programmazione e di assembleri, quanto all'applicazione di alcune intuizioni di un matematico ungherese emigrato in America, John Von Neumann.

#### La casualità di Von Neumann

L'idea di Von Neumann è semplice: perché non fornire all'elaboratore una memoria stabile e residente capace di inizializzare il sistema, in maniera neutrale, indipendente dall'applicativo da eseguire e in funzione del quale lo si accende?

Perché non fare in modo che il computer abbia un suo 'essere iniziale', uno stato iniziale dal quale e sul quale partire?

Questo semplificherebbe notevolmente le cose, giacché la inizializzazione della macchina, tutta la fase in cui, cioè, gli operatori lavorano sui circuiti per farli riconoscere come insieme integrato e investiti di senso logico, è una fase lunga e laboriosa. Se pensiamo che il passaggio da un'applicazione all'altra richiedeva il riavvio del sistema e una riassegnazione delle sue operazioni, il gap cronologico e produttivo era notevole.

Alla fine di quelle attività squisitamente umane, e solo allora, il computer riconosceva sé stesso e si disponeva in uno stato funzionale a una nuova elaborazione di dati, ma ci voleva questa preliminare elaborazione, appunto, in base alla quale la macchina si autonominava 'elaboratore di dati'.

Insomma perché non scrivere un programma che faccia giungere la macchina a ciò che oggi potremmo dire il 'prompt dei comandi'.

Ma c'era dell'altro. Il computer, in fase esecutiva, operava in perfetta sequenzialità: operazione uno, operazione due e via scorrendo. Ognuna di queste operazioni occupava, anche per tempi infinitesimali, le intere risorse del sistema, l'intero hardware.

Insomma la macchina, priva di disciplina interna, si faceva monopolizzare da ogni singola istruzione.

Ora se si dotava l'elaboratore di un sistema di auto riconoscimento si creavano i presupposti per una economizzazione delle sue risorse: rompere la sequenzialità e interrogare i singoli job, lavori da compiere.

Tutte queste cose fu la RAM: Random Access Memory, memoria ad accesso casuale.

Questa intuizione notevole fu messa in opera prestissimo: già nel 1948 il Selective Sequence Electronic Calculator della IBM adotta un programma memorizzato residente nel computer.

Questo programma memorizzato e standard non è uno standard, e infatti non tutte le macchine lo utilizzano, e non è, soprattutto, standardizzato e cioè se ne scrive uno proprio per ogni modello ed è, dunque, qualcosa di commercialmente diverso dalla RAM attuale, ma compie, concettualmente, lo

stesso lavoro e, per certi versi, anticipa l'idea del kernel, di un sistema operativo di base, fatte ovviamente tutte le necessarie distinzioni.

### Un programma di calcolo

Il software era costituito da compilatori, assembleri e codici di programmazione per l'uso di stampanti e periferiche.

Questo, se da una parte aveva precisato la relazione tra operatore e macchina e l'aveva resa meno estemporanea e più formalizzata, dall'altra parte aveva complicato la lavorazione. Sul vecchio ENIAC le cose sarebbero state più semplici.

Proviamo a descrivere una sessione di lavoro su un computer di seconda generazione nella quale bisogna creare un programma di calcolo compilato in FORTRAN, eseguirlo e stamparne i risultati.

L'operatore doveva, innanzitutto, caricare il compilatore FORTRAN sul computer; poi scriveva il programma di calcolo rispettando, ovviamente, la sintassi del linguaggio di programmazione.

Dopo di ciò copiava questo suo elaborato in FORTRAN (una sorta di programma sorgente) su un nastro o altro supporto; a questo punto faceva eseguire il compilatore che produceva, così, un file in assembler (una sorta di programma oggetto).

Eseguiva, allora, questo file creando il programma e rendendolo, finalmente, residente sul computer (installandolo si direbbe oggi); poi collegava il programma generato a tutte le librerie necessarie alla sua esecuzione e all'emissione dell'output (scrittura su disco, scrittura su nastro, scrittura sulle stampanti).

I tempi di messa in opera del programma erano eccezionalmente elevati rispetto a quelli della sua esecuzione. Ogni volta, inoltre, un programma andava reinstallato e bisognava ricreare dal sorgente l'assembler e installarlo nuovamente nel computer.

Anche, dunque, nel caso di una seconda esecuzione del programma di calcolo i tempi di preparazione erano notevoli.

Ma c'era di più, se si errava in qualche compilazione, nell'esecuzione dell'assembler o nel collegamento con le periferiche, il programma non si eseguiva e non c'era assolutamente modo di capirne il motivo: bisognava, semplicemente (e si fa per dire) rifare, con pazienza, tutto da capo.

Ci voleva una soluzione.

### Programmatori e operatori

Un primo passo verso la semplificazione della operatività della macchina venne compiuto con l'invenzione di routine residenti sul computer per le funzioni comuni: l'operatore non doveva più ricompilare i codici per le periferiche, si limitava a caricarli ed eseguirli.

Questa routine operativa, che non poteva essere univoca ma che era mirata su ogni singolo modello, venne estesa a tutte le macchine: vennero, cioè, elaborate delle schede di controllo capaci di automatizzare la relazione tra compilatore e assembler e la lettura dei dispositivi di input e output.

Si trattava di programmi di utilità, ognuno specifico per ogni modello, capaci di interpretare, controllare e guidare il processo.

Siamo, più o meno, alla fine degli anni cinquanta.

Una volta che l'operatore aveva individuato il nastro giusto, il programma sorgente adeguato, e lo aveva inserito nel lettore adeguato, con poche operazioni manuali avviava il caricamento del programma e la scheda di controllo o 'monitor residente', come si diceva allora, eseguiva l'assembler e la connessione alle periferiche.

È una novità sconvolgente sotto molti punti di vista.

Innanzitutto aumenta gli spazi di autonomia della macchina dall'intervento costante dell'uomo per la sua lavorazione: il computer, attraverso questi primi programmi di controllo operativo, acquisisce una sua identità e un'indipendenza operativa.

In secondo luogo l'elaborazione dei sistemi operativi introduce una netta separazione tra chi usa il computer e chi scrive programmi per lui: tra operatore e programmatore, cioè, figure che fino ad allora si confondevano.

Si delinea, cioè, una differenza tra chi opera 'brutalmente' su tastiere o schede perforate e chi elabora un sistema operativo e gli applicativi a quello adeguati.

Il primo non può più vedere né toccare i programmi ma deve limitarsi a caricarli e farli eseguire. Ogni suo intervento può essere foriero di danni e guasti. Per qualsiasi problema interpellerà il programmatore.

Cambia, dunque, l'organizzazione del lavoro intorno al computer: nasce l'idea attuale di 'utente informatico'.

Ma c'è un terzo punto: attraverso questa rivoluzione informatica il programmatore acquisisce una competenza e responsabilità sull'opera informatica, ne diventa l'unico conoscitore e l'autentico proprietario.

D'ora innanzi l'elaborazione e la scrittura di programmi sorgente sarà il suo repertorio.

È questo il momento d'oro della programmazione, in un mondo dove quasi ogni macchina offriva specificità notevoli e non era possibile immaginare una omologazione dei sistemi operativi.

L'informatica, con questo suo passo, si avvia verso due direzioni che sono secondo alcuni divergenti ma secondo altri parallele: verso, cioè, l'individuazione di una sua specificità linguistica e scientifica, una autonoma dignità intellettuale e verso una privatizzazione del software e un oscuramento delle sue logiche e del suo linguaggio agli utenti, anche evoluti.

Sicuramente l'invenzione dei sistemi operativi fu foriera di molteplici conseguenze nel mondo dell'editoria informatica, ovvero nacque quella che potrebbe dirsi editoria informatica: scrivere e produrre programmi poteva essere oggetto di interessanti iniziative imprenditoriali e la divisione del lavoro introdotta tra operatore e programmatore rendeva possibile la creazione di un mercato del software.

Il sistema operativo, infatti, era costitutivamente un 'mediatore' di operazioni e programmi informatici dentro la macchina e sarebbe divenuto un mediatore, una comune misura, tra macchine diverse; insomma il sistema operativo permette di immaginare la serializzazione della produzione degli applicativi.

### Prima e dopo Aristotele

E qui mi pare giusto aprire un inciso e avviare un'ipotesi.

Fino a questi anni il computer appare vincolato a una procedura produttiva artigianale: si costruiscono elaboratori, macchine e programmi di inizializzazione mirati verso un determinato contesto operativo.

E non se ne producono pochi, stiamo parlando di 6.000 esemplari. È come se l'apparato industriale si metta al servizio di una produzione e di stilemi di gestione tipicamente artigianali. È un paradosso, ma la storia si costruisce su questi paradossi.

Non che non si avanzino standard produttivi, soprattutto nel campo dell'hardware (schede perforate, lettori per schede perforate, stampanti e via discorrendo), ma questi standard non sono sufficientemente importanti per improntare l'intero ciclo produttivo del calcolatore e l'organizzazione del lavoro su quello.

La commistione tra operatore e programmatore rappresenta in modo paradigmatico questo stato artigianale dell'informatica: il processo di lavoro non è ancora differenziato e parcellizzato e, soprattutto, caratteristica tipica dell'attività artigianale, non esiste una definita divisione tra lavoro di analisi e progetto e intervento operativo, tra impegno intellettuale e manuale: intellettualità e manualità confluiscono sul medesimo sapere e contribuiscono a costruirlo direttamente, senza mediazioni.

Per questi primi quindici anni dello sviluppo dell'elettronica digitale, ci troviamo di fronte alla

cooperazione indifferenziata, intorno al computer, degli individui.

Pur avendo acquisito, attraverso l'elaborazione di compilatori, assembleri, interpreti di comandi e routine per le funzioni comuni, un certo grado di astrattezza dei singoli processi produttivi, il lavoro di programmazione non si qualifica come intervento a sé e slegato dalla specificità della macchina. Si tratta solo del fatto che l'informatica, attraverso questo impegno, ha acquisito una sua particolarità e dignità linguistica, nulla di più.

L'elaborazione delle prime schede di controllo, dei primordiali sistemi operativi, cioè, cambia radicalmente il quadro e lo trasforma a favore di una radicale e definitiva industrializzazione del processo produttivo informatico.

L'autonomia che il computer ha guadagnato rispetto all'utente rende visibile l'autonomia del programmatore nei confronti del singolo contesto operativo e si riflette in una indifferenza al caso singolo in un'assolutezza, scioglimento e astrazione dalla concreta realtà operativa delle macchine informatiche.

Solo a queste condizioni era possibile costruire un'industria elettronica e un mercato dell'informatica e le condizioni possono essere riassunte così: assolutezza del lavoro intellettuale e contemporanea astrazione e standardizzazione dei contesti operativi.

Ma soprattutto un altro fattore: autonomizzazione del sapere dall'operare, il sapere come terreno privilegiato dal mercato.

Non sappiamo se IBM e Hewlett Packard conoscessero Aristotele ma sicuramente conoscevano Henry Ford e la sua sconvolgente idea di un carburatore standard, perfettamente intercambiabile da un modello automobilistico a un altro, un 'carburatore astratto' dunque.

L'informatica è nata, seppur sotto cappelli ingombranti (sistemi di difesa militare, multinazionali telefoniche, grossi centri di calcolo e contabilità) grazie a un'orizzontale collaborazione tra soggetti diversi, dotati di diverse competenze, e questi individui manifestano una assoluta inconsapevolezza delle potenzialità commerciali del loro lavoro.

Ora tutto cambia: l'autonomia del sapere informatico, la sua astrattezza e, in soldoni, l'acquisizione di standard informatici generali creano i presupposti per una radicale separazione tra chi usa il computer e chi lo progetta. Esattamente come, qualche decennio prima, Ford aveva introdotto il divario tra chi usa e costruisce la sua Ford T e chi ne progetta le componenti.

Quasi subito nascerà una 'contestazione interna' a questo nuovo ed emergente assetto, una contestazione professionale pragmatica e non ideologica: la pirateria o hacking (con temine moderno e negativo).

Bisognerà attendere pochi anni, al massimo cinque o sei, e ottimi programmatori si mettono a lavorare 'in proprio' per mettere a completo frutto le possibilità di questo processo di standardizzazione informatica.

Da una parte, sotto il punto di vista delle aziende di elettronica, la standardizzazione e astrazione dei processi informatici era il risultato di una razionalizzazione imposta dal mercato all'informatica e l'autonomizzazione dei saperi in questo campo il prodotto di una gestione 'taylorista' dei processi produttivi e ideativi, insomma un naturale portato del mercato, dall'altra parte, il punto di vista di molti tecnici e programmatori, tutti questi fattori erano, al contrario, il prodotto di un'immensa impresa intellettuale e scientifica volta a migliorare in maniera disinteressata i processi informativi e come tale non poteva avere proprietari. La matematica ha proprietari e diritti di autore?

Ma ne riparleremo.

Neanche dieci anni dopo la realizzazione dei primi sistemi operativi, verrà generato un programma di gestione standardizzato e, per certi versi, 'universale': il sistema UNIX, sistema capace di dialogare e installarsi su molte macchine e di lavorare con differenti CPU.

Non verrà realizzato, e crediamo non per caso, da un pericoloso hacker informatico, ma niente po' po' di meno che nei laboratori della AT&T, colosso multinazionale del rame e della telefonia; la standardizzazione era il cuore del mercato informatico anche se, attraverso i primi programmatori indipendenti, poteva essere il centro e l'impulso di una libertà informatica e informativa inimmaginabile prima.

Ma ci occuperemo con calma e più avanti di questa doppia faccia di UNIX.

L'hardware non sta a guardare: monitor, mouse e circuiti integrati

L'hardware non starà a guardare, assolutamente.

Nel 1961 vengono sperimentati i primi circuiti integrati, capaci di sincronizzare il lavoro di molti transistor: le capacità della macchina vengono moltiplicate in forma esponenziale e le sue dimensioni possono essere diminuite nel medesimo modo.

L'elettronica molecolare produce i suoi effetti e il mondo della televisione inizia a guardare con interesse quello informatico e non più viceversa.

Attraverso i circuiti integrati le prestazioni del sistema aumentano vertiginosamente.

L'adozione su larga scala e in forma esclusiva di questi circuiti determina il passaggio convenzionale tra la seconda e terza generazione dei computer. Per ora i nuovi componenti vengono adottati in maniera episodica e in forme coordinate con vecchie tecnologie a transistor, ma fanno sentire l'odore del nuovo.

E profumo di novità si sentiva ovunque, soprattutto nel campo della semplificazione e 'brutalizzazione', per quanto scritto sopra, del lavoro degli operatori.

Vengono adottati insieme con le prime tastiere anche i primi monitor.

L'analogia veniva strappata al digitale e contemporaneamente si instaurava un dialogo diretto e, dunque, elettronico tra operatore e macchina. Il computer riceveva e trasmetteva secondo segnali analogici, vicini al linguaggio naturale.

La specializzazione del lavoro informatico, la sua parcellizzazione, comportava, necessariamente, una semplificazione di alcune operazioni.

### L'impero digitale dell' IBM

IBM aveva sponsorizzato il progetto MARK degli anni quaranta e aveva iniziato a costruire in proprio elaboratori digitali dotati di novità e innovazioni, come il già menzionato Selective Sequence Electronic Calculator del 1948, che era dotato di una embrionale memoria residente e inizializzante e di una sequenza non seriale nel lavoro del processore.

Era, stata, inoltre IBM a progettare il primo hard disk (1951) e a serializzarne e standardizzarne la produzione e le caratteristiche tra il 1956 e 1957.

In ragione della sua tendenza innata alla standardizzazione e taylorizzazione della produzione intellettuale che ruota intorno al nuovo prodotto informativo, IBM assunse, nei fatti, una politica monopolizzante.

E' attraverso IBM che l'informatica assume una facies industriale anche se, lo ribadiamo, altri attori (società di telefonia, AT&T, Bell Company, Hewlett Packard e il governo americano) parteciparono all'avventura dell'informatica sotto il profilo commerciale e imprenditoriale.

Non parlare di IBM e dei suoi prodotti tra fine anni quaranta e seconda metà degli anni sessanta significherebbe amputare la trattazione intorno ai computer di seconda generazione.

IBM fu la prima a vendere, oltre che affittare, centri di calcolo e a modulare la sua offerta informatica in settori di mercato 'business' e 'consumer' e questo già da questi anni, segnatamente tra 1955 e 1959.

### IBM 701

Nel 1952 IBM costruì il modello 701, non ancora basato sulla recentissima tecnologia dell'Hard Disk, che pure la società aveva sperimentato, ma su nuove valvole o diodi tubulari.

La velocità del calcolatore si innalzò fino a 20.000 operazioni la secondo, superando quella di qualsiasi macchina di prima generazione (ENIAC, UNIVAC, EDVAC).

Era dotato di un lettore di nastro magnetico capace di 8 MB che costituiva l'autentica memoria di

massa del calcolatore.

IBM 701 aveva un costo esorbitante e fu costruito in funzione del suo noleggio. Alla fine IBM ne fabbricò ben 19 esemplari, tutti rivolti a un mercato ristrettissimo e business.

### RAMAC 305

Dobbiamo ulteriormente ricordare per IBM il progetto – standard RAMAC, acronimo che stava per Random Access Method for Accounting and Control, che fu il primo computer a usare una testina rotante su un disco di memoria e che fu immediatamente utilizzato nell'industria automobilistica e dall'esercito americano, già nel 1957.

Inoltre il nuovo disco RAMAC poteva non solo essere utilizzato dal modello corrispondente ma anche essere collegato e reso operativo insieme con altri modelli precedenti dell'IBM, quali il 701 e il 650. Fu davvero una rivoluzione.

Si presentava come un enorme armadio alto circa due metri dove era contenuto il disco rigido e con un ingombro di un metro e mezzo al quadrato, una console della grandezza di una scrivania e una CPU che corrispondeva a un secondo armadio di notevoli dimensioni. Il RAMAC 305 possedeva anche un disco aggiuntivo, una seconda unità rigida, che ne faceva raggiungere l'ingombro complessivo fino a un'area di 30 - 40 metri quadrati.

RAMAC, inoltre, aveva in dotazione una stampante di notevoli dimensioni.

La programmazione della macchina avveniva con un metodo misto, da una parte utilizzando da console un linguaggio di programmazione e dall'altro, sempre da console, riposizionando i cablaggi su di un patch panel di controllo della quale era dotata, sul modello dei centralini telefonici.

Il disco era formato da una serie di corone metalliche, tamburi (da qui il nome di drum memory), sulle quali lavoravano moltissime testine rotanti.

### IBM 7090 e 7030

Nel 1958 la società texana realizzò un altro prodotto di fascia business, il 7090, con una novità importantissima: l'introduzione nella CPU accanto alle valvole tubolari di 50.000 transistor e naturalmente l'adozione della tecnologia RAMAC per la memoria di massa.

IBM 7090, proprio perché studiato per garantire prestazioni e velocità di calcolo massime, era davvero mastodontico e il suo affitto costava circa 60.000 dollari al mese.

La velocità di calcolo del processore si avvicinava al mezzo milione di operazioni al secondo (500 KH).

Alla memoria RAMAC si associava, per estenderla, una memoria su nastro.

Nel 1961 IBM presentò una macchina simile, il 7030, costruita interamente sui transistor, dotata di una tastiera di input e soprattutto capace di coordinare il lavoro del processore con quello delle memorie attraverso strumenti di 'buffer', tamponamento dei dati.

Le prestazioni del 7030 si avvicinavano a 1 MHz di clock del processore e malgrado nel 1964 fu ritirato dal mercato delle vendite per una forbice tra le rese tecniche e le rese economiche del modello, l'IBM 7030 fu una macchina rivoluzionaria giacché completamente polifunzionale (fu utilizzata in meteorologia, balistica, nel progetto Apollo) e capace di coordinare in maniera trasparente i flussi elettronici.

### La fascia consumer di IBM

650 e 1401 furono gli oggetti che tra la fine degli anni cinquanta e i primi sessanta dominarono la fascia 'economica' del mercato dell'informatica.

Il noleggio del 650 costava appena 3.000 dollari al mese e ne furono prodotti e venduti ben duemila esemplari.

La tecnologia montata sul 650 non era, ovviamente, a transistor ma a valvole tubolari ma anche questo modello poteva usufruire del collegamento con un disco RAMAC e dunque aumentare la

velocità di calcolo e l'ampiezza della memoria.

Pochi anni dopo IBM emise un nuovo modello super economico, il 1401.

IBM 1401 era formato da quattro moduli (CPU, Unità di input per schede e nastri, console di controllo e stampante) che complessivamente occupavano appena tre o quattro metri quadri.

Ma portava con sé due novità sensazionali: aveva acquisito la tecnologia buffer del 7030 ed era dotata di un sistema operativo standard che rendeva obsoleto il mantenimento di programmatori e analisti nelle società ospiti.

Insomma il 1401 giungeva al cliente già programmato e 'chiavi in mano', inoltre al contratto di acquisto e noleggio era associato un diretto supporto tecnico di IBM.

Il 1401 fu davvero il primo computer di massa, anche se ovviamente, questa massa si estendeva al mercato e alle esigenze e potenzialità di medie e piccole imprese.

Dopo il successo del 1401 IBM finì di avere seri rivali sul mercato dell'informatica.

### Curiosità ultime

L'hardware non stava a guardare ma inevitabilmente perdeva il passo. E questo divario tra prestazioni dell'hardware e software interno e delle periferiche si sarebbe approfondito e aumentato. Le periferiche, per loro natura, hanno struttura elettromeccanica mentre, al contrario, il resto della macchina ne possiede una squisitamente elettronica.

Che accade? Accade che sempre più spesso la macchina attende il dispositivo di input e output e quanto più veloce è la macchina tanto più aumentano i tempi di attesa, i tempi in cui la CPU è inattiva.

Alcuni gustosi esempi.

Fase di stampa o cioè di un output: ENIAC impiegava, con una velocità di circa 10 bit al secondo, due ore a mandare in stampa un documento di due pagine. Un personal computer degli anni novanta, capace di 166 milioni di operazioni al secondo ci impiega circa 10 secondi.

Se si fanno delle proporzioni la relazione è assolutamente favorevole a ENIAC. ENIAC utilizzava, in relazione a quel particolare dispositivo di output, la sua rudimentale CPU in modo cinquanta volte più efficace.

Non parliamo di una nuovissima macchina che vada a 2 GHz e che stampi le due pagine, poniamo, in 6 secondi. Quella sarà 300 volte meno efficiente di ENIAC rispetto ai dispositivi.

Sarebbe come comprare una Ferrari 3.000 Turbo e usarla su un sentiero, rigidamente in prima. Non ci sarebbe da stupirsi se un nostro ipotetico amico ci chiedesse. "Ma perché hai comprato questa macchina?"

Ora la maggior parte dei dispositivi di output e input (stampanti, scanner, tastiere e mouse) sono sentieri sui quali si sviluppa una potenza da formula uno, ma lo sono anche gli applicativi più usati dagli utenti e per i quali oggi nel 95% dei casi si acquista un computer e cioè fogli di calcolo e programmi di videoscrittura.

Per la buona resa di quel genere di applicazioni potrebbe bastare un sistema operativo minimale e una potenza di calcolo di qualche milione di operazioni al secondo; paradossalmente le uniche applicazioni capaci di impegnare seriamente le moderne CPU sono quelle grafiche e tra quelle i videogiochi, certamente non le applicazioni di office automation per le quali anche le aziende mettono in piedi reti di PC: migliaia di Ferrari a percorrere un sentiero.

Questo divario, davvero insensato, è stato generato dall'autonomizzazione dei saperi informatici descritta prima e dalla sua utilizzazione commerciale.