



OLIVETTI MOS: il sistema operativo del PC 128 S

1ª parte

In questo articolo parleremo del MOS nei suoi caratteri più ampi, cercando di spiegarne i meccanismi fondamentali e l'interconnessione con l'intero sistema

In tutti i computer più moderni, parte del software residente è dedicata al controllo e all'uso di componenti hardware. Nei micro-computer, questo software è spesso combinato con un linguaggio interprete, rendendo così difficile l'accesso al programmatore. Il PC 128S in ciò si differenzia: ha infatti 35Kbytes di ROM nota anche come "Machine Operating System" o MOS. L'interprete del Basic del PC 128S, lascia il controllo dell'hardware al MOS e ne fa uso attraverso le routine previste nello stesso. Questa separazione fra MOS e linguaggio ha numerosi vantaggi:

- in un sistema con più di un linguaggio, non è necessaria la duplicazione del software per supportare l'hardware;

- i linguaggi hanno la sola necessità di tradurre i comandi di gestione dell'hardware nei corrispondenti comandi MOS, pertanto essi risulteranno più compatti e più facili da scrivere;

- il funzionamento dei comandi di gestione dell'hardware è costante nei vari linguaggi;

- i programmi in assembly, possono usare le routine del MOS

(che saranno sempre presenti a differenza delle routines dei linguaggi), riducendo così il formato dei codici, lo sforzo per scriverli, e il tempo necessario alle correzioni;

- i programmi in assembly, agendo in un secondo processore, possono usare le routine del MOS per accedere alle locazioni (ex. memory-mapped devices) nel processore dell'I/O.

Il ruolo del MOS

Il MOS è una vasta e complessa parte di software in codice macchina, che provvede a supportare la vasta gamma dei componenti hardware del computer. Molte delle sofisticate caratteristiche del computer (come la gestione del suono, la gestione della tastiera, i tasti funzione, le primitive grafiche, ecc.) non sono caratteristiche dell'hardware o dei linguaggi, ma delle routine del MOS, che sono usate per interfacciare l'hardware.

Il MOS (diversamente dai filing system e dai linguaggi che possono essere selezionati o esclusi) si

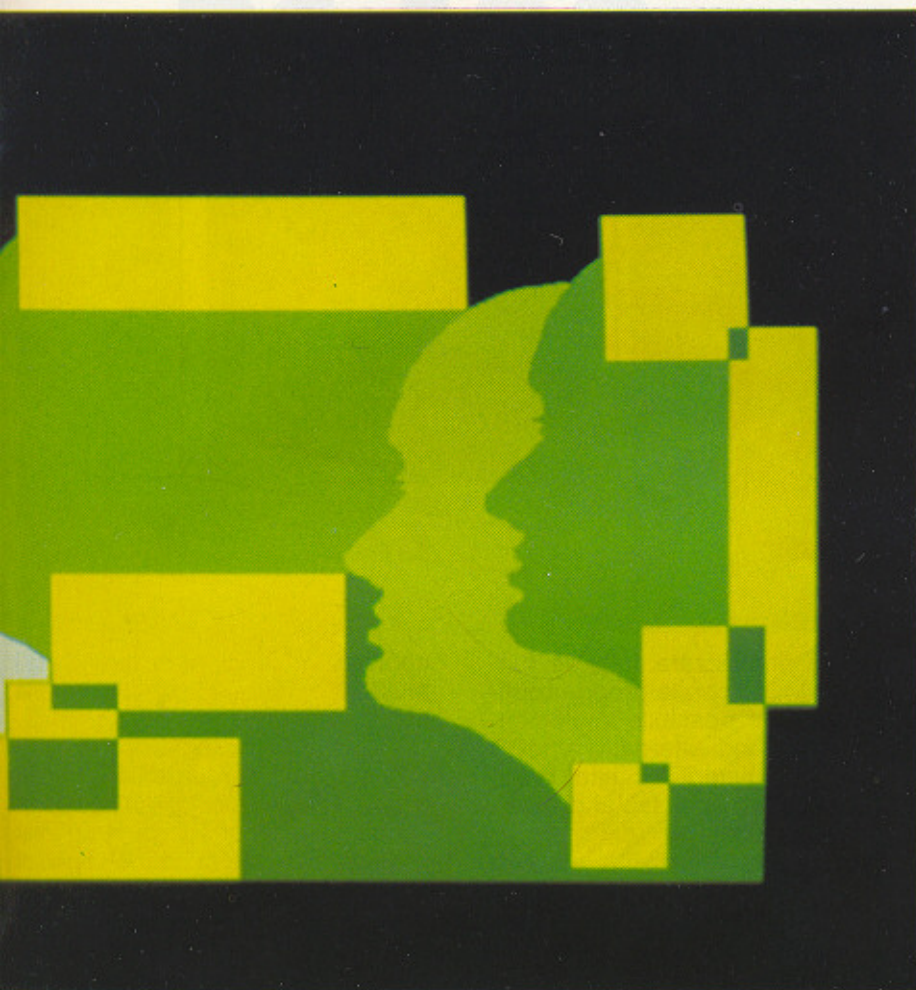


comporta come un "filtro" fra altro software ed l'hardware del computer. I linguaggi, i filing system ed i buoni programmi in assembly, fanno un uso estensivo delle routine MOS, come una più semplice ed efficiente alternativa per il controllo diretto dell'hardware.

Le routine del MOS hanno quattro principali tipi di funzione:

— Integrazione del sistema

Il MOS prende il controllo quando il computer è attivato o resettato; sorveglia l'azione reciproca dei componenti software del sistema (filing system e linguaggi); controlla gli interrupt e provvede ai processi di gestione fra il sistema e le periferiche. In tal modo, fa da "collante" fra le varie parti del sistema, facendole lavorare assieme.



— Routine di I/O

Il MOS contiene delle routine per controllare i dati di input e di output dal e per l'hardware. Queste sono usate dai linguaggi, dai filing system e dai programmi in codice macchina. Il sistema hardware di I/O più importante, interfacciato in questo modo, è rappresentato dalla tastiera, dal monitor, dalla stampante, dall'output sonoro, dall'analogo input, e dall'I/O RS423. L'I/O passando del filing system, ne è largamente controllato dal suo software.

— Configurazione del MOS

Il MOS provvede ad una vasta gamma di comandi atti ad alterare e controllare la configurazione dell'hardware e delle routine MOS usate appunto per utilizzarlo. Que-

sti comandi possono essere usati per cambiare alcune caratteristiche, quali la velocità dell'auto-repeat, la gestione del video o il tipo di stampante corrente. Essi sono usati dai linguaggi e dal filing system, ma possono anche essere eseguiti direttamente nel "command mode" dei linguaggi.

— I comandi del filing system

Molti comandi del filing system (ex. LOAD, SAVE, CAT) che sono comuni ad un certo numero di filing system, sono tradotti dal MOS in una forma più semplice prima di essere introdotti nel corrente filing system per l'esecuzione. Ciò rende i codici più compatti e permette al MOS di estendere le agevolazioni offerte dal filing system.

Notare che il MOS ha il controllo

del computer solo per brevi periodi: durante l'attivazione, il resettaggio e quando controlla gli interrupt. Altrimenti, è semplicemente usato dai linguaggi o dalle applicazioni software, come una collezione di routine che da loro accesso al sistema hardware.

L'uso del MOS

Quando il computer viene acceso, si attiva anche il MOS che esegue le funzioni appena descritte. Il MOS è anche costantemente usato dal linguaggio corrente, dal programma in codice macchina o dal filing system, per provvedere ai dati di I/O. In questo modo si usa quasi costantemente il MOS.

Il MOS può anche essere usato di proposito per le proprie necessità, come:

- cambiare il comportamento del sistema (ex. alterare la velocità d'auto-repeat dei tasti della tastiera, chiudere un collegamento esterno ecc.)

- richiedere dati al MOS (ex. l'ora e la data dal cmos clock/calendar, ecc.)

- I/O di dati (ex. stampare sullo schermo, leggere i dati provenienti dalla RS423 o dalla tastiera, ecc.)

Ci sono tre modi per fare quanto detto:

- eseguendo i comandi MOS; sia che questi siano contenuti in un programma che digitati in modo diretto;

- usando le routine MOS nei linguaggi che prevedono accessi all'assembler;

- mandando i codici VDU allo schermo sia che questi siano contenuti in un programma che digitati in modo diretto.

Il MOS può essere usato solamente da programmi in assembly o perlomeno da linguaggi di programmazione che prevedono interconnessioni con l'assembly (ex. il comando CALL del BASIC e la fun-

- usando il MOS la gestione del computer è più equilibrata, pertanto non ci sarà necessità di comprendere l'interazione dei propri



tori di terminale, non prevedono un accesso diretto alle routine del MOS. Questi programmi possono eliminare alcuni o tutti i codici di controllo VDU e possono limitare l'uso dei comandi MOS, per contro essi provvederanno all'accesso a queste agevolazioni tramite propri comandi.

I tre metodi sopra descritti, non hanno lo stesso scopo. I comandi MOS sono usati principalmente per cambiare il comportamento delle routine MOS e dei componenti dell'hardware, selezionando la configurazione desiderata dal gruppo disponibile. I codici di controllo VDU sono usati per controllare il video display hardware, periferiche grafiche e il text display. Un diretto accesso alle routine MOS attraverso una programmazione in assembly, permette di eseguire le stesse funzioni dei comandi MOS e dei codici di controllo VDU. Dal MOS, si possono sia ottenere informazioni, sia delle agevolazioni addizionali (come un I/O), che cambiare drasticamente il suo comportamento, rimpiazzando molte delle sue routine con le proprie.

I vantaggi del MOS

Usando l'assembly è virtualmente



possibile rimpiazzare tutte le funzioni del MOS, con le proprie routine. È sempre possibile implementare dei comandi alternativi al MOS tramite i linguaggi ad alto livello (ex. per diretto accesso ai dispositivi programmabili).

Sebbene la flessibilità sia una delle maggiori caratteristiche di questo microcomputer, tale potenzialità va usata con attenzione. Usando i comandi del MOS, si hanno molti vantaggi:

comandi con gli altri codici del MOS, di cercare spazio di memoria per le routine ecc.

- nel controllo delle periferiche, le routine adoperate saranno più corte e più facili da scrivere, diminuiscono i pericoli di "bug" nei programmi e non ci sarà bisogno di approfondire, più di tanto, la conoscenza delle periferiche.

- usando correttamente le routine del MOS si potrà essere certi di poter far funzionare i propri programmi, non solo sulla propria configurazione ma, tramite il Tube, anche in un co-processor, con le ultime versioni del MOS e con gli altri dispositivi compatibili.

- usando il MOS si otterranno dei programmi più veloci, i quali potranno essere implementati in programmi in linguaggi d'alto livello, quale il BASIC ecc.

Il MOS è stato progettato con molta attenzione, per semplificare l'uso dell'hardware, per arricchire le sue capacità attraverso l'uso intelligente degli interrupt e del buffering e per provvedere ad un ampio campo d'usi possibili attraverso il proprio software. Fino a quando le agevolazioni del MOS sono disponibili, si raccomanda di usarle il più possibile, piuttosto di sforzarsi a controllare l'hardware in modo diretto.

