Sistemi Operativi Batch

martedì 7 marzo 2023 14:59

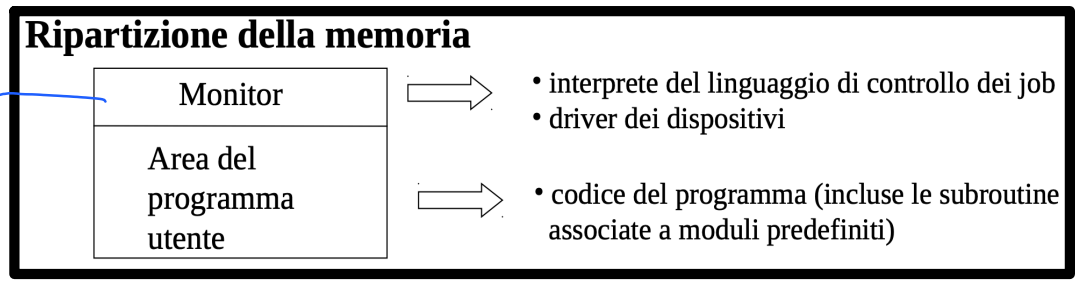
SISTEMI OPERATIVI BATCH

ANNI 50 - 60

LE PRIME VERSIONI dei SISTEMI OPERATIVI che ERANO STATE mandate IN ESERCIZIO IN quell'epoca ERANO denominate SISTEMI OPERATIVI BATCH.

QUALI SONO LE CARATTERISTICHE DI UN S.O BATCH?

Questa architettura di memoria, veniva ad essere suddivise in due zone separate:



SOFTWARE del MONITOR

LA ZONA SUPERIORE ERA RISERVATA PER mantenere IL SOFTWARE di UN COMPONENTE denominato MONITOR.

IL MONITOR AVEVA AL SUO INTERNO delle CAPACITÀ che nell'ELABORAZIONE SERIALE ERA IN MANO agli utilizzatori: Load, IL caricamento di qualche cosa che NOI dobbiamo mandare IN esecuzione all'INTERNO della memoria del NOSTRO SISTEMA di calcolo. ALL'INTERNO del SISTEMA OPERATIVO BATCH lo PORTA AVANTI IL MONITOR.
IL MONITOR HA AL SUO INTERNO una logica che funziona da INTERPRETE del LINGUAGGIO di CONTROLLO dei JOB. Quindi c'è del software all'INTERNO di questo MONITOR che CONTROLLA se eventualmente CI SONO dei JOB da mandare IN esecuzione e IL CONTROLLO si fa effettuando l'INTERROGAZIONE che FORNISCE dei dati IN INPUT e che permette di caricare alcuni dati all'INTERNO della MEMORIA di LAVORO.
Se questa cosa è VERA, IL MONITOR NON fa altro che attivare questo DISPOSITIVO per caricare all'INTERNO dell'altra zona della memoria AREA DEL PROGRAMMA UTENTE, IL JOB da eseguire.
IL caricamento lo fa IL MONITOR, MA esso ha anche altre COMPONENTI SOFTWARE, Tra le tante cose c'è anche L'INSIEME dei driver dei dispositivi.

Essi sono l'insieme dei moduli SOFTWARE che ci permettono di pilotare dei COMPONENTI che sono all'INTERNO dell'hardware, quindi PER esempio dei DISPOSITIVI.

Quindi se noi SCRIVIAMO un'applicazione che deve andare IN ESERCIZIO sul SISTEMA OPERATIVO BATCH, questa applicazione deve poter usare i DISPOSITIVI, NON c'è la necessità che NOI all'INTERNO dell'applicazione andiamo ad IMPORTARE IL SOFTWARE che esegua l'INTERAZIONE CON IL DISPOSITIVO.

PERCHÈ? Perché questo SOFTWARE è già PRESENTE nel MONITOR, e quando l'applicazione VIENE mandata IN ESERCIZIO per andare ad INTERAGIRE CON un DISPOSITIVO NON può fare altro che INVOCARE una funzione all'INTERNO del MONITOR.

TUTTE LE ATTIVITÀ AD USO DEGLI UTILIZZATORI NELL'ELABORAZIONE SERIALE, VENIVANO AD ESSERE AUTOMATIZZATE PERCHÈ codificate ALL'INTERNO di MODULI SOFTWARE DI QUESTO MONITOR.

QUESTO MONITOR È RESIDENTE IN MEMORIA, OSSIA L'INSIEME dei moduli SOFTWARE che COSTITUISCANO IL MONITOR NON VENGONO MAI AD ESSERE scaricati dalla MEMORIA.
quando accendiamo IL SISTEMA di calcolo vengono caricati IN memoria, quando lo SPEGNIAMO IL CONTENUTO della MEMORIA VIENE ad ESSERE PERSO.
MA MENTRE IL NOSTRO SISTEMA di calcolo è attivo, IL MONITOR È RESIDENTE IN MANIERA COSTANTE nel tempo all'INTERNO della memoria, quindi è sempre UTILIZZABILE e DISPONIBILE IN Tutte le attività che RIGUARDANO la gestione dei JOB che devono essere eseguiti.

SE NOI ABBIAMO CARICATO un PROGRAMMA e gli abbiamo dato IL CONTROLLO TRAMITE IL MONITOR, IL PROGRAMMA che è IN ESERCIZIO RESTITUIRÀ IL CONTROLLO al MONITOR IN CASO di:

- TERMINAZIONE
- ERRORE
- INTERAZIONE CON I DISPOSITIVI

NON C'È LA POSSIBILITÀ PER IL MONITOR, di RIPRENDERE IL CONTROLLO quando LUI VUOLE.
L'Applicazione può chiamare uno o più moduli del MONITOR per eseguire delle INTERAZIONI con i DISPOSITIVI.
IL MONITOR HA AL SUO INTERNO I DRIVER DEI DISPOSITIVI e l'applicazione potrebbe voler

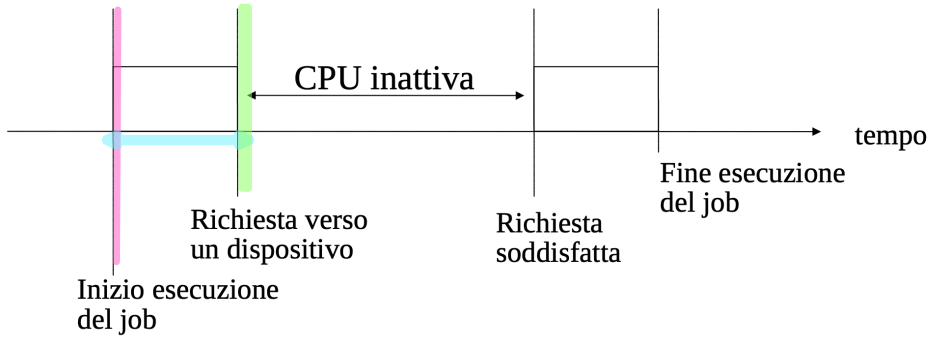
MA NON CI SONO ALTRI TIPI DI INTERAZIONE, E QUESTO È UN PROBLEMA.

- basati sull'utilizzo di un **monitor (set di moduli software)**
  1. Il programma e' reso disponibile tramite un dispositivo di input
  2. Il monitor effettua il caricamento del programma in memoria e lo avvia
  3. Il programma restituisce il controllo al monitor in caso di terminazione o errore, ed in caso di interazione coi dispositivi
  4. Il monitor e' residente in memoria di lavoro
  5. I moduli richiesti per uno specifico programma vengono caricati dal monitor come subroutine del programma stesso

RELAZIONI TRA IL SOFTWARE DEL MONITOR E IL SOFTWARE DI UN'APPLICAZIONE

- APPLICAZIONE CHE INTERAGISCE CON IL MONITOR PERCHÈ MAGARI VUOLE INTERAGIRE CON I DISPOSITIVI.

LO FA INVOCANDO PRATICAMENTE IL SOFTWARE del MONITOR



SUPPONIAMO DI AVERE una linea del tempo IN CUI NOI RAPPRESENTIAMO quello che STA SUCCEDENDO all'INTERNO del NOSTRO SISTEMA di calcolo.

Supponiamo che, durante lo svolgimento di questa linea del tempo, NOI ABBIAMO dato IL CONTROLLO al software del JOB che DOBBIAMO andare ad eseguire.

Quindi IL MONITOR ha caricato IN memoria IL PROGRAMMA da eseguire e gli ha passato IL CONTROLLO e chiaramente IN questa fase IL NOSTRO PROGRAMMA STA utilizzando la CPU, CHE STA ESEGUENDO LE SUE ISTRUZIONI MACCHINA.

Supponiamo che A questo ISTANTE di tempo, NOI ABBIAMO che IL PROGRAMMA APPLICATIVO, QUINDI IL JOB che STIAMO andando ad ESEGUIRE, VADA AD effettuare una RICHIESTA VERSO UN DISPOSITIVO, VADO AD INVOCARE un modulo del MONITOR e questo modulo fa si che SI ATTIVI un dispositivo PER qualche OPERAZIONE.

ORA LA CPU RIMANE INATTIVA: NON significa che NON STA FACENDO NULLA, ogni CPU FA SEMPRE QUALCOSA.

NON LA STIAMO UTILIZZANDO PER PORTARE AVANTI IN MANIERA CORRETTA E FRUTTUOSA delle ISTRUZIONI MACCHINA.

EVENTUALMENTE la STIAMO utilizzando PER BUSY WAITING, SEMPLICEMENTE TESTANDO LO STATO DEL DISPOSITIVO

PROBLEMA: POSSIBILE sottoutilizzo della CPU DOVUTO alla lentezza dei dispositivi

PER OVVIARE A QUESTO PROBLEMA ABBIAMO AVUTO DEGLI SVILUPPI IN PARTICOLAR modo NEI SISTEMI OPERATIVI BATCH, e QUESTI SVILUPPI SONO BASATI SU UNA SERIE di TECNICHE che SONO STATE POI STORICAMENTE INTRODOTTE.

UNA DI QUESTE SI CHIAMA SPOOLING.