LEZIONE 1

* Cos’è un SO

Un SO è un insieme di programmi che servono a interporre il computer con l’uso di altri programmi veri e propri (word, excel, ecc); quindi in altre parole il SO deve assegnare le risorse necessarie al programma per poterlo rendere eseguibile.

* [Detto in maniera generale: un SO è un insieme di programmi che funge da intermediario tra un utente e gli elementi fisici di un calcolatore (ovvero l’hardware).
* Gli obiettivi di un SO

Eseguire programmi d’applicazione (ovvero i programmi veri e propri lanciati dall’utente) e deve inoltre rendere possibile l’esecuzione di tali programmi e rendere il sistema conveniente da utilizzare.

Tutte queste procedure devono garantire un utilizzo efficiente dei dispositivi questo perchè noi in realtà potremo senz’altro lanciare ad esempio l’esecuzione di un filmato sul nostro computer ma ovviamente se tale esecuzione impedisce che si possano fare parallelamente altre operazioni il sistema non risulterà essere efficiente.

Quindi con uso EFFICIENTE si intende proprio che il dispositivo sia conveniente.

* Componenti del nostro sistema di calcolo

Bisogna valutare 4 punti principali (I quali sono disposti a strati uno sopra l’altro):

1. DISPOSITIVI FISICI (HARWARE): strato più basso; forniscono le risorse fondamentali per tutto il resto dell’elaborazione (CPU, memoria, I/O)
2. SISTEMA OPERATIVO: deve controllare e coordinare l’uso dei dispositivi da parte dei programmi d’applicazione per gli utenti.
3. PROGRAMMI D’APPLICAZIONE: definiscono il modo in cui sono utilizzate le risorse del sistema per risolvere i problemi degli utenti, cioè sono i programmi veri e propri che lancia l’utente (compilatore, videogiochi, applicativi per il business, ecc).
4. UTENZA: strato più elevato del sistema di calcolo. Possono essere persone oppure anche altre macchine o calcolatori.

Ma perché l’utente di una macchina può essere un altro sistema?

Immaginiamo due programmi che comunicano tra loro in maniera automatica senza l’intervento diretto dell’utente, quindi anche la richiesta di un altro programma che deve accedere ad un nostro programma sulla nostra macchina è un comparabile a quello che potrebbe essere il singolo utente e lanciare una singola applicazione.

Tutto questo appena detto si può schematizzare con questa struttura stratificata

Utente 4

Utente 2

Utente 1

Utente 3

Compilatore Assemblatore Editor di testo Basi di dati

Programmi d’applicazione

Sistema operativo

Dispositivi fisici

Lv più basso

Da questa struttura si vede che il sistema operativo sia un elemento che va ad interporsi tra i programmi d’applicazione e i dispositivi fisici , di conseguenza è giusto dire che è un intermediario tra programmi d’applicazionde e dispositivi fisici.

ESEMPIO

Immaginiamo di usare word (programma d’applicazione). Quando word deve scrivere non utilizza lui in maniera diretta i dispositivi fisici, ma farà delle chiamate al sistema operativo il quale avrà il compito di metterci a disposizione le risorse che richiediamo.

Definizione di SO

Il SO è visto come:

* ASSEGNATORE DI RISORSE: visto come un sistema che gestisce e assegna risorse, questo perché avendo il computer più risorse (più CPU, più stampanti, ecc), il sistema deve comprendere quali di queste risorse disponibili sia più conveniente assegnare.
* PROGRAMMA DI CONTROLLO: dato che questo controlla l’esecuzione dei programmi, in quanto tali programmi una volta che sono eseguiti nella macchina non vengono lasciati lì in esecuzione, ma il sistema conteggia l’esecuzione. Questo per dire che un programma durante l’esecuzione non deve bloccare il resto della macchina.
* Tutte queste operazioni vengono svolte perché all’interno del computer ci sta qualcosa che permette l’operatività del sistema operativo ovvero il **KERNEL**, cioè il nucleo. All’interno del nostro computer viene caricato questo programma ( ovvero il SO), e questo è caricato valutandone soprattutto le richieste d’uso.

ESEMPIO

Un sistema operativo molto grande può avere diversi moduli per la comunicazione, stessa cosa per gestione della stampa ecc; non viene caricato tutto il SO ma ciò che viene caricato sicuramente nel SO è sempre una parte minimale, e man mano che servono saranno caricati altri moduli volta per volta. La parte che viene sempre e comunque caricata è appunto il KERNEL, ovvero l’unico programma che funziona sempre, mentre il resto è visto come programma d’attivazione.

SISTEMI MAINFRAME

Sono sistemi main, come dice la parola stessa, quindi principali, di grande volume non inteso in senso saziale, ma logico. Spesso possiamo identificarli come un centro di calcolo con tante macchine.

* L’obiettivo di un sistema mainframe è quello di:
* ridurre i tempi di elaborazione raggruppando insieme lavori con requisiti simili, questo perché se riusciamo a fare questo raggruppamento abbiamo l’opportunità di trattarli in maniera omogenea.
* Deve essere in grado di fare trasferimenti automatici da un lavoro (job) al successivo. Quindi un mainframe non assolve solo un lavoro e poi si ferma, ma deve essere capace di accodare una serie di processi, quindi sapere come far partire e fermare un processo per poter far cominciare il processo successivo.
* Un’altra peculiarità deve essere quella relativa ad un **MONITOR** **RESIDENTE,** cioè ad una capacità del sistema mainframe di realizzare un pacchetto di funzionalità che rendano in maniera residente, ovvero che rendano sempre attiva la capacità di tener sotto controllo

l’intero sistema.

Configurazione della memoria per un sistema Batch (a lotti)

SISTEMA OPERATIVO

Area per programmi utenti

Questi 2 blocchi stanno a simboleggiare che

all’interno del nostro sistema abbiamo il

blocco del SO quindi tutto concentrato in

un’unica parte; un altro blocco che è quello

dell’area per programmi degli utenti. Questa

è una visione che ci facilita la configurazione

della nostra memoria in un sistema in cui da

una parte (quella alta) ci sta tutto il SO quindi

tutte le procedure, mentre in quella bassa

l’area dei programmi utenti.

Sistemi multiprogrammati

Il SO tiene contemporaneamente nella memoria centrale diversi lavori, e la CPU non rimane mai inattiva.

0

STEMA OPERATIVO

LAVORO 1

LAVORO 2

LAVORO 3

LAVORO 4

Parte privilegiata

Immaginiamo questo oggetto come uno scaffale nel quale andiamo ad inserire dei dati. Quando il SO dovrà memorizzare i vari job nei punti liberi della memoria, dovrà trovarsi di fronte ad una serie di scelte da effettuare per scegliere probabilmente il posto migliore per il job a seconda delle esigenze di una certa circostanza.

512 KB

Caratteristiche di un sistema multiprogramamto

* Tutti i job che entrano nel sistema sono mantenuti in un gruppo che consiste di tutti i processi d’elaborazione presenti nel disco che attendono il caricamento nella maniera centrale.
* Gestione della memoria: il sistema deve essere capace di ripartire la memoria tra tutti i job che si trovano in entrata.
* *Scheduling* *della* CPU: consiste nell’assegnare a certi job anziché ad altri la CPU; questa non è una scelta basata semplicemente sulla turnazione, ma basata su molti altri parametri i quali permettono di effettuare una scelta più adeguata e corretta ai fini della disponibilità di un programma, dell’urgenza; in altre parole bisogna definire la **PRIORITA’** di ogni processo.
* Allocazione delle risorse.

Sistemi a partizione del tempo di elaborazione

Affinchè la CPU possa essere distribuita tra tutti i processi che ne fanno richiesta bisognerà adottare qualche metodo. Il metodo adottato è quello **TIME**-**SHARING** (ripartizione del tempo); la CPU viene assegnata ad un solo processo per volta ed esso deve essere presente nella memoria centrale; quindi la CPU non fa nient’altro che commutare la sua esecuzione tra quei processi che si trovano già nella memoria centrale.

* Comunicazione diretta tra utente e sistema: il sistema passa rapidamente da un utente all’altro, quindi ogni utente ha l’impressione di disporre dell’intero calcolatore. Significa che il time-sharing è una metodologia che viene eseguita in maniera molto rapida e consente al sistema di avere un’apparenza di esclusività.
* Ciascun utente dispone di almeno un proprio programma nella memoria.

Sistemi da scrivania

* Personal computer: calcolatori utilizzati da un singolo utente.
* Dispositivi I/O: tastiera, mouse, schermo, piccole stampanti.
* Comodità e prontezza d’uso per l’utente.
* Possibilità di adottare le tecnologie sviluppate per i grandi sistemi.
* Possono funzionare con sistemi operativi diversi (Windows, MacOS, UNIX, Linux).

Sistemi paralleli

* Sistemi con più unità d’elaborazione, con più CPU in stretta collaborazione.

Un sistema parallelo è un sistema sul quale concorrono diversi processi in modo parallelo; ma questi processi posso concorrere in parallelo quando sono indipendenti tra di loro.

* Sistemi strettamente connessi: i processi condividono la memoria e i temporizzatori dei cicli di macchina e di clock; la comunicazione di solito avviene attraverso la memoria condivisa.

Vantaggi dei sistemi paralleli

1. Maggior produttività (troughput).
2. Economia di scala.
3. Degradazione controllata.
4. Tolleranza ai guasti.

Questo perché ci sta una ridondanza delle risorse.

Differenza tra SMP e AMP (nei sistemi paralleli)

* MULTIELABORAZIONE SIMMETRICA(SMP): Ciascuna unità d’elaborazione esegue un’identica copia del SO.

Si possono eseguire molti processi contemporaneamente senza causare un rilevante calo delle prestazioni.

La maggior parte dei moderni sistemi operativi supporta la SMP.

* MULTIELABORAZIONE ASIMMETRICA(AMP): A ogni unità d’elaborazione si assegna un compito specifico.

Un’unità d’elaborazione principale controlla il sistema, le attività attendono istruzioni dall’unità principale.

Più comune nei sistemi di grandi dimensioni.

ARCHITTETTURA PER LA SMP

Unità d’elaborazione

Unità d’elaborazione

Unità d’elaborazione

memoria

La memoria è condivisa ma le unità tra di loro possono eseguire copie identiche del sistema senza che venga a degradarsi la prestazione dell’intero sistema.

Sistemi distribuiti

* I sistemi distribuiti si basano sulle reti per realizzare le proprie funzioni, sfruttano le capacità di comunicazione per cooperare nella soluzione dei problemi di calcolo e per fornire agli utenti un ricco insieme di funzioni.
* **Sistema distribuito:** ciascuna unità d’elaborazione ha la propria memoria locale e comunica con le altre per mezzo di linee di comunicazione di vario genere, ad esempio bus ad alta velocità o linee telefoniche.
* Vantaggi:

1. Condivisione delle risorse.
2. Maggiore velocità.
3. Affidabilità.
4. Comunicazione.

* Richiedono un’infrastruttura di rete.
* Local Area Networks (LAN) o Wide Area Networks (WAN) , Metropolitan Area Networks (MAN).
* Possono essere sia sistemi client-server, sia sistemi peer-to-peer.

Struttura generale di un sistema client-server

client

client

client

client

……….

rete

server

Batterie di sistema (system cluster)

* Le **batterie di sistemi** sono basate sull’uso congiunto di più unità d’elaborazione riunite per svolgere attività d’elaborazione comuni.
* Forniscono un’elevata disponibilità.
* **Batterie asimmetriche:** un calcolatore rimane nello stato di **attesa attiva** mentre l’altro esegue le applicazioni.
* **Batterie simmetriche:** due o più calcolatorieseguono le applicazioni e allo stesso tempo si controllano reciprocamente.

**Sistemi d’elaborazione in tempo reale**

* Spesso impiegati nella gestione dei dispositivi di controllo per applicazioni specifiche (controllo di esperimenti scientifici, rappresentazione d’immagini in medicina…).
* Presenta vincoli di tempi fissati e ben definiti entro i quali si deve effettuare l’elaborazione.
* Esistono due tipi di sistemi d’elaborazione in tempo reale:

1. **Sistemi d’elaborazione in tempo reale stretto:** si ha quando il nostro sistema dovrà dare le risposte in tempo estremamente ridotto in modo più rapido possibile.
2. **Sistemi d’elaborazione in tempo reale debole:** sistemi nei quali è richiesto un tempo di elaborazione rapido però a differenza dei primi non è cosi critico se in qualche modo la risposta non arriva nei tempi assegnati.

**Sistemi palmari**

* Personal digital assistent (PDA).
* Telefoni cellulari
* A causa delle piccole dimensioni, la maggior parte dei dispositivi palmari dispone di:
* Memoria limitata.
* Unità d’elaborazione lente.
* Schermi piccoli.