

4SI - TPSIT

UDA1 - Sistemi operativi

1. POST (Test di accensione)

Il POST è un programma della Scheda Madre.

(slide 36)

POST (Power-On Self Test) è il programma di autodiagnostica che viene eseguito automaticamente all'accensione di ogni componente hardware.

Appena un dispositivo (come la scheda madre, la RAM, la scheda video, ecc.) riceve corrente, effettua una serie di test interni per verificare che tutto funzioni correttamente.

Se tutti i controlli danno esito positivo, il POST della scheda madre segnala il successo e il processo di bootstrap può proseguire.

In caso contrario, il computer emetterà segnali d'allarme (beep particolari o messaggi di errore sullo schermo) indicanti l'hardware guasto.

In base alla sequenza di beep, interpretandoli con il manuale della scheda madre, si capisce dove sta il problema.

2. Sistema Operativo

Sistema Operativo:

dirige le operazioni e gestisce le risorse hardware

Processo: istanza del programma

Il sistema operativo svolge due compiti fondamentali.

Primo: funge da gestore delle risorse hardware: si assicura che CPU, memoria e periferiche vengano assegnate e utilizzate correttamente dai vari programmi, evitando conflitti e ottimizzando le prestazioni.

Secondo: fornisce all'utente (e ai programmi applicativi) un ambiente di uso comodo, mettendo a disposizione comandi e interfacce per interagire con il sistema.

In altre parole, il SO da un lato è un amministratore che orchestra l'uso di hardware tra i vari processi, dall'altro è un interprete che traduce i comandi dell'utente in azioni hardware.

quando accendiamo un programma, il SO dedica il 100% delle risorse della CPU; questo viene limitato se ci sono diversi processi attivi

Senza il SO ci sarebbe solo l'hw: no UX, no multitasking

il prompt dei comandi permette l'accesso diretto all'hardware

Nel mondo del software distinguiamo due grandi categorie: software di base e software applicativo.

SOFTWARE BASE E SOFTWARE APPLICATIVO

Il software di base (es: Sistema Operativo) comprende tutti i programmi che servono al funzionamento essenziale del computer e a supportare lo sviluppo o l'esecuzione di altri programmi. Fanno parte del software di base, ad esempio: il sistema operativo stesso, i programmi di utilità come editor di testo semplici, compilatori, linker, loader e debugger (strumenti usati per creare ed eseguire altri software).

Il software applicativo, invece, è costituito dai programmi che l'utente utilizza per svolgere compiti specifici o lavori pratici (videoscrittura, giochi, browser web, grafica, ecc.). In breve, il software di sistema serve alla macchina per funzionare, mentre il software applicativo serve all'utente per lavorare.

Gli aggiornamenti software vengono fatti principalmente per temi di sicurezza.

KERNEL DEL SISTEMA OPERATIVO

Il kernel (in italiano "nucleo") è la parte centrale e più importante del sistema operativo.

È il "cuore" del SO, ossia quel programma che rimane sempre caricato in memoria RAM e gestisce le operazioni di base del sistema.

Quando accendiamo il PC, dopo il bootstrap viene caricato in RAM proprio il kernel del sistema operativo, che da quel momento resta attivo a gestire processi, memoria e dispositivi a basso livello.

Il kernel opera a stretto contatto con l'hardware: gestisce l'accesso alla CPU, la memoria fisica, i driver delle periferiche, e offre servizi di base su cui si appoggiano tutti gli altri software.

COS'è IL SISTEMA OPERATIVO

Quando accendi il PC, una piccola parte “fissa” della macchina (firmware BIOS/UEFI) esegue i controlli, trova il disco di avvio e carica in RAM il cuore del sistema operativo: il kernel.

Da quel momento il SO è sempre attivo e governa tutto, fino allo spegnimento. Il SO è, in poche parole, un insieme di programmi che fanno da intermediario tra L'UTENTE e L'HARDWARE.

NUCLEO = KERNEL

Gestione dei processi

La gestione dei processi coordina l'esecuzione dei programmi nel sistema.

il multitasking è un vero multitasking

Quando ho i processi attivi e non modifico nulla (non scrivo, non modifico le finestre, etc) i processi utilizzano solo la RAM; quando ad esempio scrivo, si attiva anche la CPU perché sto modificando qualcosa.

Il sistema operativo divide la memoria centrale tra i processi in esecuzione (es. caricando in RAM il codice e i dati di un programma attivo) e, in presenza di molte applicazioni, può ricorrere alla memoria virtuale (swapping su disco) per estendere le capacità della RAM.

-> il S.O. ricolloca i processi meno usati dall'utente nel disco, assegnando più RAM possibile al processo principale che l'utente richiede

Makefile

Serve ad automatizzare script ripetitivi (lanciare script, etc)

è un file di testo che contiene regole su come costruire o aggiornare un programma.

nomeFile.make

Struttura:

target: [file1] [file 2]

[Comando 1]

[Comando 2]

NB: importante il “tab” prima dei comandi (stile python)

Git

Git è un version control system, che consente di gestire un progetto in condivisione con altri utenti, lavorando in locale e caricando in un secondo momento il programma con le modifiche, facendo un merge con le altre versioni caricate online.

Comandi da terminale git:

\$ git status → serve per controllare lo stato dei file nel branch

\$ git add. → Carica tutti i file

\$ git commit -m "Aggiunto 2 file" → per commentare le modifiche fatte

\$ git push → pubblica i file che sono stati caricati

\$ git pull → prendo i file da git in locale