

#### Firmware/1

Il firmware è un tipo di software incorporato all'interno di un componente elettronico. In questo contesto, il componente elettronico si riferisce a qualsiasi circuito integrato intelligente presente in vari dispositivi, inclusi processori, schede grafiche, schede audio, schede di interfaccia di rete e dispositivi periferici come stampanti e monitor.

La funzione principale del firmware è inizializzare il componente e facilitare la sua interazione con altri componenti all'interno del dispositivo. Essenzialmente, il firmware fornisce al componente un insieme di interfacce e protocolli di comunicazione, permettendogli di utilizzare un linguaggio comune (che include sintassi e semantica) per comunicare efficacemente con altri componenti.

#### Firmware/2

Il termine firmware è una combinazione di 'firm' (fermo) e 'ware' (software) che indica che il firmware non è facilmente modificabile dall'utente. Funziona come un ponte tra hardware e software, consentendo ai componenti hardware di comunicare e funzionare secondo le istruzioni del software.

Il BIOS (Basic Input/Output System) è un esempio ben noto di firmware nei computer. Risiede sulla scheda madre ed è cruciale per gestire il processo di avvio iniziale del computer, inclusa l'inizializzazione dell'hardware e l'avvio del sistema operativo.

All'interno di ogni dispositivo, che sia un PC desktop, un laptop, un computer di bordo di un'auto o un computer interno di un aereo, ci sono componenti fondamentali simili, come le schede madri. Questi componenti possono differire per dimensioni e specifiche tecniche, ma svolgono funzioni analoghe.

Il firmware è tipicamente memorizzato in una memoria non volatile all'interno del componente o del dispositivo elettronico, il che significa che la memoria mantiene i suoi contenuti anche quando l'alimentazione è spenta.

#### Memoria non volatile

I tipi comuni di memoria non volatile utilizzati per memorizzare il firmware includono:

- ROM (Read-Only Memory): Tradizionalmente usata per la memorizzazione del firmware, una volta scritta non può essere modificata.
- EPROM (Erasable Programmable Read-Only Memory): Può essere cancellata e riprogrammata utilizzando la luce UV, ma questo processo non è così conveniente come le alternative moderne.
- EEPROM (Electrically Erasable Programmable Read-Only Memory): Può essere cancellata e riprogrammata usando cariche elettriche, rendendola più flessibile rispetto all'EPROM.
- Memoria Flash: Un tipo di EEPROM che può essere cancellata e riscritta in blocchi, comunemente utilizzata nei dispositivi moderni per la sua flessibilità e maggiore capacità di archiviazione.

Questi tipi di memoria sono integrati nel dispositivo, garantendo che il firmware sia prontamente accessibile per inizializzare e controllare i componenti del dispositivo quando è acceso.

#### Scheda madre/1

Una scheda madre è la principale scheda a circuiti stampati (PCB) in un computer o in un altro dispositivo elettronico che ospita e permette la comunicazione tra molti dei componenti elettronici cruciali.

- Hub centrale: Funziona come l'hub centrale che collega tutte le parti del computer, consentendo loro di comunicare e lavorare insieme.
- Socket CPU: La scheda madre contiene un socket che ospita la central processing unit (CPU). Questo è dove viene installata la CPU.
- Slot di memoria: Ha slot per moduli di memoria (RAM), permettendo al computer di memorizzare temporaneamente e accedere rapidamente ai dati.
- BIOS/UEFI: Il firmware del Basic Input/Output System (BIOS) o Unified Extensible Firmware Interface (UEFI) è memorizzato sulla scheda madre, fornendo le istruzioni necessarie per il processo di avvio iniziale e l'inizializzazione dell'hardware.

#### Scheda madre/2

- Slot di espansione: Questi slot consentono di aggiungere schede aggiuntive, come schede grafiche, schede audio, schede di rete e altri periferiche.
  - Connettori di archiviazione: La scheda madre ha connettori per dispositivi di archiviazione, come hard disk e SSD.
  - Connettori di alimentazione: Include connettori per fornire energia alla scheda madre e ai suoi componenti dall'alimentatore (PSU).
  - Connettori periferici: Porte e connettori per periferiche come dispositivi USB, jack audio, cavi di rete e connessioni video (HDMI, DisplayPort).
- In sintesi, la scheda madre è la spina dorsale di un computer, integrando tutti i componenti critici e assicurando che possano comunicare e funzionare insieme in modo efficace.

#### Sistema di Input/Output di Base (BIOS)

Il Basic Input/Output System (BIOS) è un insieme di routine software che fornisce la struttura software fondamentale permettendo al sistema operativo di interfacciarsi con l'hardware del dispositivo. Il BIOS è memorizzato in un chip di memoria non volatile, il che significa che conserva i dati anche senza alimentazione. Oggigiorno, il BIOS è tipicamente memorizzato su memoria FLASH o EEPROM saldata direttamente sulla scheda madre, che può essere riscritta utilizzando una procedura di aggiornamento appropriata.

#### BIOS - BOOTSTRAP/1

Quando un dispositivo viene avviato, prima che il sistema operativo venga caricato, il BIOS esegue diverse operazioni chiave:

**Schermata di Avvio:** Visualizza una schermata che consente all'utente di configurare e gestire alcune impostazioni di base del BIOS.

**Routine di Avvio:** Il BIOS avvia la routine di avvio per caricare il sistema operativo. Il processo di avvio è incrementale, caricando le parti essenziali del sistema operativo dal settore di avvio per portare il sistema a uno stato operativo stabile.

#### BIOS - BOOTSTRAP/2

**Individuazione del MBR:** Dopo aver eseguito le routine di base del BIOS, la prima istruzione del BIOS è localizzare il "punto di partenza" del disco rigido (o altro supporto di memorizzazione di massa) dove è memorizzato il sistema operativo. In questo punto si trova il Master Boot Record (MBR).

**Passaggio di Controllo al Sistema Operativo:** Il BIOS legge il contenuto del MBR e trasferisce il controllo al sistema operativo (da questo punto, la gestione del dispositivo è completamente gestita dal sistema operativo).

Il MBR contiene informazioni sulla posizione del SETTORE DI AVVIO sul disco rigido (o altro supporto di memorizzazione di massa che ospita il sistema operativo).

Il SETTORE DI AVVIO è il punto di ingresso per l'avvio del sistema operativo.

#### Interfaccia Firmware Estensibile Unificata (UEFI)/1

L'Unified Extensible Firmware Interface (UEFI) è un'interfaccia firmware moderna progettata per inizializzare i componenti hardware e caricare il sistema operativo all'avvio di un dispositivo. Offre un processo di avvio simile al BIOS ma con diversi vantaggi:

**Interfaccia Grafica Utente (GUI):** UEFI può offrire un'interfaccia grafica più intuitiva con supporto per il mouse, rendendo più semplice navigare e configurare le impostazioni.

**Gestore di Avvio:** Un programma integrato in UEFI che determina quale sistema operativo o boot loader eseguire, semplificando il processo di avvio e offrendo maggiore flessibilità rispetto al BIOS.

**Supporto per Driver:** UEFI può caricare direttamente i driver, migliorando la compatibilità hardware e le prestazioni durante il processo di avvio.

Sicurezza Avanzata: UEFI include funzionalità come Secure Boot, che aiuta a proteggere il sistema assicurando che solo software affidabile venga caricato durante il processo di avvio.

Estendibilità e Aggiornamenti: UEFI è modulare e può essere aggiornato o esteso con nuove funzionalità, fornendo un migliore supporto per nuove tecnologie e una maggiore stabilità del sistema nel tempo.

## Interfaccia Firmware Estensibile Unificata (UEFI)/2: VEDI SLIDE 12

### Driver

I driver sono programmi software che permettono al sistema operativo e ad altri software di comunicare con i dispositivi hardware. Agiscono come intermediari, traducendo i comandi di alto livello del sistema operativo in comandi di basso livello che l'hardware può comprendere. Gestiscono l'interazione tra il sistema operativo e l'hardware.

### Caratteristiche:

Installati sul sistema operativo.

Facilitano la comunicazione tra OS e hardware.

Possono essere aggiornati o sostituiti più frequentemente.

Firmware vs Driver

Differenze Chiave:

#### Posizione:

Firmware: Integrato nell'hardware.

Driver: Installato sul sistema operativo.

#### Funzione:

Firmware: Gestisce e controlla l'hardware a un livello fondamentale.

Driver: Traduce i comandi dell'OS in azioni eseguite dall'hardware.

#### Frequenza di Aggiornamento:

Firmware: Aggiornato meno frequentemente, spesso attraverso processi di aggiornamento specializzati.

Driver: Può essere aggiornato regolarmente per migliorare le prestazioni o aggiungere nuove funzionalità.

### Sistema Operativo/1

Il sistema operativo è il software fondamentale responsabile della gestione delle operazioni dell'intero dispositivo, compresa l'esecuzione di applicazioni, programmi e servizi, nonché l'interazione con l'utente. Posizionato tra il layer hardware (incluso il firmware) e il layer delle applicazioni (applicazioni, programmi, servizi), il ruolo chiave del sistema operativo è astrarre le complessità del computer per il layer delle applicazioni.

In particolare, il sistema operativo gestisce quattro compiti principali:

Gestione dei componenti hardware.

Esecuzione di applicazioni, programmi e servizi.

Interfacciamento con dispositivi periferici.

Facilitazione dell'interazione con l'utente.

### Sistema Operativo/2

In generale, il sistema operativo definisce due aspetti principali di un dispositivo:

Modalità Operativa del Dispositivo: Questa modalità determina lo scopo e l'ambito del dispositivo, spiegando perché viene utilizzato un dispositivo specifico.

Interazione con l'Utente: Questa modalità definisce come gli utenti possono interagire e operare con il dispositivo.

Esempi di sistemi operativi includono:

Microsoft Windows

GNU/Linux

macOS  
Android  
iOS  
...e molti altri.

#### Classificazione Basata su Caratteristiche di Elaborazione

**Sistemi Operativi Batch:** Questi sistemi operativi eseguono programmi in modo non interattivo. Gli utenti caricano programmi e dati e ricevono i risultati solo dopo il completamento dell'elaborazione.

**Sistemi Operativi Interattivi a Tempo Condiviso:** Questi sistemi operativi eseguono più programmi contemporaneamente, condividendo le risorse hardware (come il tempo della CPU e la memoria) tra loro. Gli utenti possono interagire con i programmi durante l'esecuzione, fornendo input e ricevendo output durante tutto il processo.

**Sistemi Operativi in Tempo Reale:** Questi sistemi operativi garantiscono che i programmi vengano eseguiti in tempo reale (o entro un intervallo di tempo fisso) senza ritardi, latenze o degradazione delle prestazioni. Sono tipicamente utilizzati in processi industriali o robotica applicata.

**Sistemi Operativi Integrati:** Questi sistemi operativi sono integrati in hardware dedicato, spesso presenti in dispositivi specifici come automobili o aeroplani.

**Sistemi Operativi Hypervisor:** Questi sistemi operativi permettono la suddivisione delle risorse hardware in diverse macchine virtuali, consentendo l'esecuzione simultanea di più OS, programmi e modalità operative.

#### Classificazione Basata su Caratteristiche Funzionali

**Sistemi Operativi Mono-Task:** Questi sistemi operativi consentono l'esecuzione di un solo programma (ovvero compito) alla volta. Un nuovo programma può iniziare solo quando quello attuale è terminato.

**Sistemi Operativi Multi-Task:** Questi sistemi operativi permettono l'esecuzione parallela di più programmi. Le risorse hardware (e.g., tempo della CPU, memoria) sono condivise tra vari programmi, consentendo loro di essere eseguiti contemporaneamente.

**Sistemi Operativi Multi-Threading:** In questi sistemi operativi, ogni singolo programma è diviso in più sotto-programmi (ovvero thread). Ogni thread viene eseguito sulla CPU (o su un core), migliorando le prestazioni del dispositivo.

**Sistemi Operativi Mono-Utente:** Questi sistemi operativi sono progettati per interagire e eseguire programmi per un solo utente alla volta.

**Sistemi Operativi Multi-Utente:** Questi sistemi operativi possono interagire e eseguire programmi per più utenti simultaneamente.

#### Struttura del Sistema Operativo

Un sistema operativo può essere strutturato come segue:

- Gestione dei Programmi e Interfaccia Utente
- Gestione della Protezione e Sicurezza
- Gestione della Memoria Secondaria
- Gestione delle Unità Periferiche
- Gestione del File System
- Gestione della Memoria Principale
- Gestione dei Processi (Scheduler)

#### Kernel del Sistema Operativo

Il kernel è il componente centrale del sistema operativo. Funziona come un ponte tra le applicazioni e l'hardware fisico, garantendo un'esecuzione coordinata e fluida dei compiti.

Basati sulla loro struttura interna e sul design del kernel, i sistemi operativi possono essere classificati in tre tipi:

- Sistema Operativo con Kernel Monolitico
- Sistema Operativo con Microkernel
- Sistema Operativo con Kernel Ibrido

Sistema Operativo con Kernel Monolitico

Questo sistema implementa un'astrazione completa del computer all'interno del kernel, che esegue il sistema operativo stesso.

- Kernel Monolitico: Include tutte le funzionalità di base necessarie per il sistema operativo per gestire l'hardware e i processi di sistema nello spazio del kernel (Kernel Space). Questo include driver di dispositivi, gestione della memoria, gestione del file system e chiamate di sistema, tutto eseguito in un unico spazio di indirizzi. Le applicazioni e le librerie vengono eseguite nello spazio utente (User Space).

- Performance Elevata: Eseguendo tutti i servizi essenziali nello spazio del kernel, i kernel monolitici possono raggiungere alte prestazioni grazie a un numero minimo di cambi di contesto tra la modalità utente e la modalità kernel, riducendo l'overhead e migliorando l'efficienza.

- Rischio di Instabilità: Poiché tutti i componenti funzionano nello stesso spazio di indirizzi, un guasto o un bug in una parte del kernel può potenzialmente far crashare l'intero sistema.

- Modularità Limitata: Il kernel monolitico è meno modulare rispetto al microkernel. Aggiungere nuove funzionalità o driver spesso richiede di modificare e ricompilare l'intero kernel, il che può essere complicato e limitare la flessibilità.

#### Sistema Operativo con Microkernel

Questo sistema implementa solo le funzioni essenziali del sistema operativo nel kernel. Altre funzionalità sono fornite da programmi di sistema separati come driver di dispositivi e server.

- Design Modulare: Il microkernel è progettato per essere modulare. Mantenendo solo funzioni essenziali come la comunicazione tra processi, la schedulazione di base e un'astrazione hardware minima nello spazio del kernel (Kernel Space), altre funzionalità vengono gestite da processi separati nello spazio utente (User Space). Questo design migliora la modularità e consente aggiornamenti e manutenzioni più facili.

- Maggiore Stabilità e Sicurezza: Isolando i driver e i servizi di sistema dal kernel, il microkernel può aumentare la stabilità e la sicurezza del sistema. I guasti o i bug in un servizio non necessariamente faranno crashare l'intero sistema, come accadrebbe in un kernel monolitico.

#### Sistema Operativo con Kernel Ibrido

Questo sistema implementa varie funzionalità del sistema operativo all'interno del kernel ma può caricare moduli aggiuntivi al momento dell'avvio per migliorare e completare le caratteristiche del kernel.

- Approccio Flessibile: Il kernel ibrido fornisce un equilibrio tra le architetture monolitiche e microkernel. Includendo funzionalità essenziali all'interno del kernel e permettendo di caricare moduli aggiuntivi al momento dell'avvio, offrono un approccio flessibile al design del sistema.

- Benefici di Prestazioni e Modularità: Questa architettura mira a raggiungere i benefici di prestazioni dei kernel monolitici mantenendo un grado di modularità.

- Esempi di Sistemi Operativi: I sistemi operativi come le versioni moderne di Windows e macOS utilizzano un approccio a kernel ibrido. Questi sistemi dimostrano l'implementazione pratica e i vantaggi dei kernel ibridi in scenari reali.