

LEZIONE 1: Cenni su hardware e Sistemi Operativi

1. Componenti dell'architettura di un elaboratore

- CPU (Central Processing Unit): È composta principalmente da tre unità:
 - ALU (Arithmetic Logic Unit): si occupa delle operazioni matematiche e logiche.
 - Control Unit (CU): coordina il flusso di istruzioni e dati tra le varie componenti.
 - Registri: sono piccole aree di memoria ultra-veloci situate all'interno della CPU, usate per memorizzare temporaneamente dati e istruzioni. La velocità della CPU è misurata in GHz (gigahertz), che indica il numero di cicli di clock al secondo.

◦ RAM (Random Access Memory): Oltre ad essere volatile, la RAM è anche dinamica (DRAM) o statica (SRAM). La DRAM è più economica ma più lenta, mentre la SRAM è più veloce ma più costosa. La RAM è divisa in heap (memoria per variabili dinamiche) e stack (memoria per variabili locali).

◦ Memoria su disco (HDD/SSD):

▪ HDD: In un hard disk, i dati sono memorizzati su dischi magnetici rotanti. La velocità di lettura/scrittura è limitata dal movimento fisico delle testine, ma gli HDD sono ideali per archiviazioni a lungo termine grazie al loro costo contenuto.

▪ SSD: Gli SSD sono basati su memoria flash NAND, quindi non hanno parti mobili, rendendoli molto più veloci e resistenti agli urti. Hanno un ciclo di vita limitato in termini di scritture, ma sono sempre più usati per il loro miglioramento nelle prestazioni.

2. Sistema operativo

◦ Il sistema operativo fornisce un'interfaccia tra l'hardware e l'utente, gestendo l'allocazione delle risorse e la sicurezza. Ogni sistema operativo ha un kernel, che è la parte fondamentale responsabile della gestione delle risorse hardware e dell'esecuzione dei processi. Il kernel può essere monolitico (tutto in un unico blocco) o microkernel (separato in moduli).

3. Comandi del sistema operativo

◦ ls: Visualizza il contenuto della directory corrente. L'uso di opzioni come ls -a mostra anche i file nascosti (quelli che iniziano con un punto).

◦ cd: Usato per navigare tra le directory. Esempio: cd .. ti porta alla directory superiore.

◦ touch: Crea un file vuoto o aggiorna la data di modifica di un file esistente. È utile per creare file di log temporanei.

◦ mkdir: Serve per creare directory, importante per strutturare progetti o file system.

◦ rm: Rimuove file o directory. È importante essere cauti con il comando, specialmente con rm -rf, che rimuove in modo ricorsivo e forzato tutte le directory e i loro contenuti.

4. Creazione di una struttura standard di un progetto Python

◦ Una struttura di progetto ben definita in Python include file per il codice (solitamente una cartella src o project_name), test (cartella tests), e file di configurazione (requirements.txt, setup.py).

◦ Fondamentale avere un file README.md per documentare il progetto e un file .gitignore per escludere file non necessari dal controllo versione.

=====

LEZIONE 7: Reti di Calcolatori – Fondamenti e Principi

1. Cos'è una rete di calcolatori

◦ Le reti di calcolatori sono composte da due o più dispositivi (computer, server, router, etc.) interconnessi che permettono la trasmissione di dati. Le reti possono essere cablate, wireless o un mix di entrambi, con la possibilità di essere collegate tra loro tramite modem, switch, router, o hub.

- I principali modi di trasmissione includono la trasmissione seriale (un bit alla volta) e la trasmissione parallela (più bit contemporaneamente).

2. Utilizzi delle reti

- Le reti sono essenziali per garantire la collaborazione remota, l'accesso a risorse condivise, e la comunicazione globale. Le moderne reti sono progettate per essere scalabili, permettendo la connessione di milioni di dispositivi in tutto il mondo, come nel caso delle reti sociali, delle applicazioni cloud e delle reti aziendali.

3. Comunicazione point-to-point e broadcast

- La comunicazione point-to-point avviene tra due dispositivi con una connessione diretta (ad esempio, un collegamento tra un computer e una stampante). Questo tipo di comunicazione è affidabile, ma non scalabile a livello di rete globale.

- La comunicazione broadcast è utilizzata per inviare un messaggio a tutti i dispositivi su una rete, come nel caso di un router che invia pacchetti a tutti i dispositivi in una rete locale.

4. Topologia

- Topologia ad albero: è una combinazione della topologia a stella e della topologia a bus. Viene utilizzata in reti di grandi dimensioni in cui ci sono più livelli di dispositivi.

- Topologia a maglia completa: ogni dispositivo è connesso direttamente a tutti gli altri dispositivi della rete. Garantisce alta resilienza e ridondanza, ma è costosa da implementare.

5. Modello Client-Server

- Nel modello Client-Server, il server è il nodo centrale che offre risorse o servizi (come file, applicazioni, database), mentre i client sono i dispositivi che accedono a queste risorse. I client inviano richieste HTTP o altre richieste di rete al server, che elabora la richiesta e restituisce una risposta. Questo modello è usato per applicazioni web, email, e servizi di streaming.

6. Modello Peer-to-Peer (P2P)

- In una rete P2P, i nodi sono simmetrici e tutti i dispositivi possono fungere sia da client che da server. Le risorse (file, video, ecc.) vengono distribuite direttamente tra gli utenti. I client P2P sono autonomi, il che significa che non c'è un server centrale che gestisce le connessioni.

- Le applicazioni P2P sono particolarmente utili per sharing di file (es. BitTorrent), comunicazioni vocali e video (es. Skype), e in blockchain (dove la rete distribuisce il controllo e la gestione dei dati tra i vari nodi).

7. Classificazione su scala geografica

- Le WAN (Wide Area Network) possono essere pubbliche (internet) o private (come le VPN aziendali). Le MAN (Metropolitan Area Network) sono utili per connettere edifici in una città o università, e le LAN (Local Area Network) sono utilizzate per connettere dispositivi all'interno di una singola struttura come un ufficio o una casa.
-
-
-

LEZIONE 10: Architettura – Gestione delle Informazioni

1. Gestione delle informazioni

- La gestione delle informazioni riguarda anche il concetto di Database Management System

(DBMS), che è un software che gestisce la creazione, l'archiviazione e il recupero delle informazioni in un database.

◦ DBMS si occupano di operazioni come query (ricerca), inserimento, aggiornamento e eliminazione dei dati.

2. Codice Binario

◦ Oltre a essere fondamentale per la rappresentazione numerica, il codice binario viene usato in sistemi digitali, dove ogni stato (acceso o spento) è rappresentato da 1 o 0. Le porte logiche (AND, OR, NOT) sono utilizzate per costruire circuiti elettronici che elaborano i dati in binario.

3. Codice ASCII

◦ ASCII è anche utilizzato per rappresentare simboli di controllo come il carriage return (ritorno a capo), line feed (avanzamento linea), e tabulazione. Ogni codice ASCII occupa un byte (8 bit), e i caratteri non visibili sono utilizzati per il controllo del flusso di dati nei terminali.

4. Dimensioni dei dati

◦ Le dimensioni dei dati vengono misurate in bit, byte e word (parola), che rappresentano unità di memorizzazione in un computer. La dimensione di una parola varia a seconda dell'architettura del computer, di solito 16, 32 o 64 bit.

◦ Il termine bitrate si riferisce alla quantità di dati che possono essere trasferiti in un dato periodo di tempo (ad esempio, 10 Mbps indica 10 megabit al secondo).

5. Proprietà del binario

◦ Il sistema binario ha numerosi vantaggi per la realizzazione di circuiti elettronici, come la semplicità nel design di porte logiche e la facilità con cui possono essere rappresentati stati elettronici (on/off, 1/0).

6. Conversione binario-decimale

◦ Il metodo della divisione per 2 funziona dividendo ripetutamente un numero decimale per 2 e registrando i resti, che poi vengono letti al contrario.

◦ Esempio, per il numero 13:

- $13 \div 2 = 6$ resto 1
- $6 \div 2 = 3$ resto 0
- $3 \div 2 = 1$ resto 1
- $1 \div 2 = 0$ resto 1 Il numero binario di 13 è 1101.

7. Conversione decimale-binario

◦ Il metodo della divisione per 2 è simile al metodo di conversione inversa. Si divide il numero decimale per 2, annotando il resto ad ogni passaggio. Il risultato finale sarà la rappresentazione binaria del numero.

8. Addizione e sottrazione binaria

◦ Addizione binaria: simile all'addizione decimale, ma con solo 0 e 1. Quando si somma 1 + 1 si ottiene 10, quindi un carry (riporto) deve essere aggiunto al bit successivo.

◦ Sottrazione binaria: quando si sottrae un numero binario più grande da uno più piccolo, si utilizza il complemento a due, che è un modo per rappresentare i numeri negativi in binario.

9. Algebra Booleana

◦ L'algebra booleana è essenziale nella progettazione di circuiti digitali, per ottimizzare le operazioni logiche. Le equazioni booleane possono essere semplificate usando le leggi dell'algebra booleana per costruire circuiti più efficienti e risparmiare risorse.

10. Modello di Von Neumann

- Il modello di Von Neumann prevede che la memoria condivida lo spazio sia per i dati che per il programma in esecuzione. Questo modello ha portato all'idea dei registri di memoria e delle istruzioni di fetch per ottenere dati dalla memoria.
-
-
-

LEZIONE 11: Von Neumann Model e Protocolli di rete

1. Architettura degli Elaboratori – CPU, Memoria e I/O

1. Il Modello di Von Neumann: definisce una struttura base del computer, composta da tre blocchi principali:

- CPU (Central Processing Unit)
 - Memoria
 - Dispositivi di I/O (Input/Output)
- Questi elementi sono collegati tramite il Bus di Sistema, che si suddivide in:
- Bus Dati: Trasmette i dati tra i componenti (istruzioni, risultati, ecc.).
 - Bus Indirizzi: Specifica le locazioni di memoria o periferiche I/O coinvolte nel trasferimento.
 - Bus di Controllo: Trasporta segnali di controllo e sincronizzazione (es. segnali di lettura/scrittura, interrupt).

2. Bus di Sistema

◦ Il Bus di Sistema è il canale di comunicazione che collega la CPU, la memoria e le periferiche, permettendo:

- Il trasferimento di dati, indirizzi e segnali di controllo tra le componenti.

3. CPU – Unità di Elaborazione Centrale

La CPU è il "cervello" del computer, esegue istruzioni, elabora dati e coordina il funzionamento generale del sistema.

a. Processing Unit – Unità di Elaborazione

- ALU (Arithmetic Logic Unit)
 - Esegue operazioni aritmetiche (somma, sottrazione, ecc.) e logiche (AND, OR, NOT, XOR).
- Registri Temporanei (TEMP)
 - Memorie veloci per immagazzinare dati intermedi durante l'esecuzione delle istruzioni.

b. Control Unit – Unità di Controllo

- PC (Program Counter)
 - Contiene l'indirizzo della prossima istruzione da eseguire.
- IR (Instruction Register)
 - Memorizza temporaneamente l'istruzione corrente da eseguire.

4. Ciclo di Istruzione (Instruction Cycle)

Le fasi dell'esecuzione di un'istruzione sono:

- FETCH
 - La CPU preleva l'istruzione dalla memoria (usando il PC).
 - L'istruzione viene caricata nell'IR.
- DECODE
 - L'unità di controllo interpreta l'istruzione e identifica l'operazione da eseguire.
- EXECUTE
 - L'operazione viene eseguita (es. calcolo ALU, trasferimento dati, ecc.).

5. Memoria

- La memoria centrale (solitamente RAM) è usata per memorizzare temporaneamente:
 - Le istruzioni del programma in esecuzione
 - I dati da elaborare
- Organizzazione della Memoria: composta da celle da 1 byte (8 bit). Ogni cella ha:
 - Un valore (il contenuto, es. 10100010)
 - Un indirizzo (es. 0001, 0010, ecc.)
- Operazioni Fondamentali
 - FETCH: recupera un valore da una locazione di memoria.
 - STORE: memorizza un valore in una locazione specifica.

6. Registri MDR e MAR

- MDR (Memory Data Register)
 - Registra temporaneamente i dati da leggere/scrivere in memoria.
 - Funziona come buffer tra la CPU e la memoria.
- MAR (Memory Address Register)
 - Contiene l'indirizzo della cella di memoria da leggere/scrivere.
 - Viene utilizzato durante le operazioni FETCH e STORE.

7. Operazioni di FETCH e STORE

- FETCH (lettura da una locazione A):
 - MAR \leftarrow A
 - Invio segnale di lettura
 - MDR \leftarrow Dato letto
- STORE (scrittura del valore X nella locazione A):
 - MDR \leftarrow X
 - MAR \leftarrow A
 - Invio segnale di scrittura

8. Dispositivi di Input e Output

Permettono l'interazione del computer con l'esterno.

a. Input:

Funzione: Inviano dati al computer.

Esempi: Tastiera, mouse, scanner, microfono, webcam.

Funzionamento: Convertono segnali esterni in dati digitali.

b. Output:

Funzione: Mostrano i risultati dell'elaborazione.

Esempi: Monitor, stampanti, altoparlanti, proiettori.

Funzionamento: Convertono segnali digitali in forma percepibile (suono, immagine, stampa).

9. Dispositivi di Input/Output

Alcuni dispositivi possono sia ricevere sia inviare dati:

- Monitor Touchscreen
- Input: tocco dell'utente
- Output: visualizzazione su schermo
- Hard Disk Esterno
- Input: scrittura dati
- Output: lettura dati
- Modem
- Input: ricezione dati da Internet
- Output: invio dati verso Internet

2. Computer Networks – Protocols

Internet:

- Internet (Internet): insieme di reti diverse collegate tramite gateway.
- Ogni rete può avere hardware e protocolli differenti.
- protocolli permettono la comunicazione tra dispositivi all'interno e tra le reti.

Protocols

- Protocollo: insieme di regole predefinite che gestiscono la comunicazione tra dispositivi elettronici.
- Inizialmente centrati sull'hardware, oggi il focus è sul software.

Central Role of Software

- Funzionalità moderne:
 - Gestione dati, sicurezza, performance, cloud.
 - Software sofisticato = alta adattabilità.
- Sicurezza:
 - Software implementa firewall, crittografia, IDS.
- Compatibilità:
 - Permette l'interoperabilità tra hardware diversi.

Stack of Layers (Struttura a strati)

I protocolli sono organizzati in livelli (es. OSI Model).

Ogni livello (n) comunica col suo pari sull'altra macchina attraverso il protocollo di livello n.

- I livelli servono:
 - a semplificare il design,
 - a nascondere i dettagli tecnici ai livelli superiori.

Network Communication (Analogia e Funzionamento)

Analogia:

- Mittente → messaggio tradotto e trasportato → destinatario.
- Traduzioni = trasformazioni dei dati (es. crittografia).
- Ogni passaggio = livello del protocollo (es. traduttore = livello applicazione).
- Meccanismo:
 - Sorgente (top-down): dati scendono i livelli con header/trailer aggiunti.
 - Destinazione (bottom-up): ogni livello rimuove il proprio header/trailer.
 - Comunicazione avviene grazie ai servizi del livello inferiore.
- Esempio tecnico (Livelli 5 → 1):
 - Layer 5 (Applicazione): invia messaggio M.
 - Layer 4 (Trasporto): aggiunge header con seq. num, ACK, etc.
 - Layer 3 (Rete): divide in pacchetti, header con IP.
 - Layer 2 (Collegamento dati): frame con MAC + trailer (errore).
 - Layer 1 (Fisico): invio su cavo o mezzo.

Connection-Oriented Services

- Caratteristiche:
 - Affidabili (ACK, ritrasmissione, ordine garantito).
 - Controllo di flusso.
 - Handshake iniziale e chiusura.
 - Usati in: email, banking.
- Handshake:

- Richiesta ($A \rightarrow B$)
 - Risposta ($B \rightarrow A$)
 - Conferma ($A \rightarrow B$)
- ACK Mechanism:
 - $A \rightarrow$ pacchetto $\rightarrow B$
 - $B \rightarrow$ ACK $\rightarrow A$
 - Ritrasmissione se ACK non ricevuto

Connectionless Services

- Caratteristiche:
 - Nessun handshake, alta velocità.
 - Meno affidabili (pacchetti persi o fuori ordine).
 - Ogni pacchetto è autonomo.
 - Usati in: streaming, gaming.
-
-
-

LEZIONE 12: Fondamenti di Sistemi Operativi, Modello ISO/OSI, Livello Fisico, Livello Data Link

3. Architetture dei Computer Fondamenti di Sistemi Operativi

Firmware

Il firmware è un tipo di software incorporato all'interno di un componente elettronico. In questo contesto, per componente elettronico si intende qualsiasi circuito integrato intelligente presente in vari dispositivi, inclusi processori, schede grafiche, schede audio, schede di rete e periferiche 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. In pratica, il firmware fornisce al componente un set di interfacce e protocolli di comunicazione, permettendo di usare un linguaggio comune (comprendente sintassi e semantica) per comunicare efficacemente con gli altri componenti.

Firmware/1

Il termine firmware deriva dalla combinazione di "firm" e "ware", indicando che il firmware non è facilmente modificabile dall'utente. Serve da ponte tra hardware e software, permettendo ai componenti hardware di comunicare e funzionare secondo le istruzioni del software.

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

Ogni dispositivo, che sia un PC desktop, un laptop, il computer di bordo di un'auto o il sistema interno di un aereo, contiene componenti fondamentali simili come la scheda madre. Questi componenti possono variare in dimensioni e specifiche tecniche, ma svolgono funzioni analoghe.

Il firmware è solitamente memorizzato in una memoria non volatile all'interno del componente elettronico o del dispositivo, cioè una memoria che mantiene i dati anche quando il dispositivo è spento.

Memoria Non Volatile

Tipi comuni di memoria non volatile usati per memorizzare il firmware includono:

ROM (Read-Only Memory): Usata tradizionalmente per il firmware, ma una volta scritta non può essere modificata.

EPROM (Erasable Programmable Read-Only Memory): Può essere cancellata e riprogrammata usando luce UV, ma questo processo non è comodo come le alternative moderne.

EEPROM (Electrically Erasable Programmable Read-Only Memory): Può essere cancellata e

riprogrammata usando una carica elettrica, risultando più flessibile rispetto all'EPROM.

Memoria Flash: Un tipo di EEPROM che può essere cancellata e riscritta a blocchi, comunemente usata nei dispositivi moderni grazie alla sua flessibilità e maggiore capacità di archiviazione.

Questi tipi di memoria sono integrati nel dispositivo, garantendo che il firmware sia sempre accessibile per inizializzare e controllare i componenti al momento dell'accensione.

Scheda Madre (Motherboard):

La scheda madre è la principale scheda a circuito stampato (PCB) in un computer o altro dispositivo elettronico che ospita e consente la comunicazione tra molti componenti elettronici cruciali.

Hub Centrale: Funziona come hub centrale che collega tutte le parti del computer, permettendo loro di comunicare e lavorare insieme.

Socket CPU: Contiene la sede dove si installa il processore (CPU).

Slot di Memoria: Slot per moduli di memoria RAM, consentendo al computer di memorizzare temporaneamente dati per un accesso rapido

BIOS/UEFI: Il firmware Basic Input/Output System (BIOS) o Unified Extensible Firmware Interface (UEFI) è memorizzato sulla scheda madre e fornisce le istruzioni necessarie per il processo di avvio iniziale e l'inizializzazione hardware.

Scheda Madre/2

Slot di Espansione: Permettono l'aggiunta di schede aggiuntive, come schede grafiche, audio, di rete o altre periferiche.

Connettori di Memoria: Connettori per dispositivi di memorizzazione, come hard disk e SSD.

Connettori di Alimentazione: Connessioni per fornire energia alla scheda madre e ai suoi componenti dall'alimentatore.

Connettori Periferici: Porte e connettori per periferiche come USB, jack audio, cavi di rete, e uscite video (HDMI, DisplayPort).

In sintesi, la scheda madre è la spina dorsale del computer, integrando tutti i componenti critici e assicurando che possano comunicare e funzionare efficacemente insieme.

BIOS (Basic Input/Output System)

Il BIOS è un insieme di routine software che fornisce la struttura software di base permettendo al sistema operativo di interfacciarsi con l'hardware del dispositivo.

Il BIOS è memorizzato in un chip di memoria non volatile, quindi mantiene i dati anche senza alimentazione. Oggi, il BIOS è solitamente memorizzato su memoria Flash o EEPROM saldata sulla scheda madre, che può essere riscritta tramite procedure di aggiornamento.

BIOS – Fase di Avvio (Bootstrap)

Quando il dispositivo si avvia, prima che il sistema operativo venga caricato, il BIOS esegue varie funzioni chiave:

Mostra la schermata di avvio, che permette all'utente di configurare alcune impostazioni basiliari del BIOS.

Inizia la routine di boot per caricare il sistema operativo, caricando progressivamente le parti essenziali del sistema operativo dalla partizione di avvio per portare il sistema a uno stato operativo stabile.

Esegue i test POST (Power On Self Test) per verificare il corretto funzionamento dei componenti come scheda madre, processore e memoria.

Configura le impostazioni hardware se non vengono rilevati errori gravi durante i test.

Cerca il Master Boot Record (MBR) nel disco rigido o altro supporto di memoria dove è memorizzato il sistema operativo.

Legge il contenuto del MBR e passa il controllo al sistema operativo.

Se la tabella MBR è danneggiata, il disco potrebbe non avviarsi.

UEFI (Unified Extensible Firmware Interface)

UEFI è un'interfaccia firmware moderna progettata per inizializzare l'hardware e caricare il sistema operativo all'avvio.

Vantaggi rispetto al BIOS:

Interfaccia grafica (GUI) con supporto mouse, più user-friendly.

Boot Manager integrato per scegliere quale sistema operativo o boot loader eseguire.

Supporto per driver caricati direttamente, migliorando compatibilità e prestazioni.

Sicurezza migliorata con funzioni come Secure Boot.

Modularità e possibilità di aggiornamenti o estensioni nel tempo.

GPT (GUID Partition Table)

GPT è un sistema moderno per la gestione delle partizioni su dischi rigidi o SSD, parte dello standard UEFI e successore del vecchio MBR.

Usa GUID (Globally Unique Identifier), identificatori univoci globali per ogni partizione.

Include header primario, voci di partizione e header secondario (backup).

Offre maggiore sicurezza e flessibilità rispetto a MBR.

Driver

I driver sono programmi software che permettono al sistema operativo e ad altri software di comunicare con l'hardware.

Funzionano da intermediari, traducendo comandi di alto livello del sistema operativo in azioni comprensibili dall'hardware.

Sono installati nel sistema operativo e possono essere aggiornati frequentemente.

Sistema Operativo (OS)

Il sistema operativo è il software fondamentale che gestisce tutte le operazioni del dispositivo, inclusi applicazioni, programmi, servizi e interazioni con l'utente.

Posizionato tra l'hardware (incluso il firmware) e il livello applicativo, il sistema operativo astrae la complessità del computer per l'applicazione.

Compiti principali:

Gestire i componenti hardware

Eseguire applicazioni, programmi e servizi

Interfacciarsi con le periferiche

Facilitare l'interazione con l'utente

Esempi: Windows, GNU/Linux, macOS, Android, iOS.

Classificazione dei Sistemi Operativi

In base al tipo di elaborazione:

Batch OS: esegue programmi senza interazione utente.

Time-Sharing OS: esegue più programmi simultaneamente con interazione.

Real-Time OS: garantisce esecuzione in tempo reale senza ritardi.

Embedded OS: integrato in hardware dedicato.

Hypervisor OS: permette virtualizzazione, eseguendo più OS insieme.

In base alle funzionalità:

Monolitici (Kernel grande, tutto integrato)

Microkernel (Kernel minimale, servizi separati)

Reti di Computer

Architettura di Rete

Definisce struttura e funzionamento di una rete tramite livelli e protocolli.

Proprietà di rete:

Proprietà privata (design indipendente dal produttore)

Standard de facto (es. TCP/IP)

Standard de iure (es. ISO/OSI)

Modello ISO/OSI

Creata nel 1984 da ISO, standard ufficiale per descrivere le reti a 7 livelli.

Principi:

- Ogni livello con astrazione e funzione chiara
- Minima informazione tra livelli
- Non troppi livelli o funzioni in uno

Livello Fisico

Gestisce la trasmissione di bit su supporti fisici:

- Cavi rame (Ethernet, coassiale)
- Fibra ottica
- Radiofrequenze (Wi-Fi, Bluetooth)

Wi-Fi e Bluetooth

Wi-Fi: frequenze 2.4/5/6 GHz, copertura ~100 m, LAN wireless.

Bluetooth: 2.4 GHz, ~10-100 m, PAN (personal area network).

Vantaggi: niente cavi, mobilità e facilità di connessione.

Frequenza e Banda

Frequenza = oscillazioni al secondo (Hz).

Banda (bandwidth) = intervallo di frequenze usate da un segnale.

Banda definisce quanti dati si possono trasmettere al secondo (bps).

Prodotto Banda-Ritardo

Quantità massima di dati in transito in rete.

Banda = "larghezza tubo", Ritardo = "lunghezza tubo", prodotto = volume dati in viaggio.

Livello Data Link

Trasferisce dati tra nodi vicini con controllo errori e gestione flusso.

Funzioni:

- Framing (incapsulamento dati in frame)
- Rilevamento e correzione errori (es. CRC)
- Controllo flusso
- Indirizzamento MAC (indirizzi fisici)

LLC e MAC (Sotto-livelli Data Link)

LLC: gestione errori e controllo flusso.

MAC: controllo accesso al mezzo trasmissivo, evita collisioni.

Metodi di Controllo Flusso LLC

Stop-and-Wait: invia un frame, aspetta conferma. Semplice ma lento.

Sliding Window: invia più frame prima di aspettare conferme. Più efficiente ma complesso.

MAC e Controllo Accesso

CSMA/CD: usato in Ethernet, rileva collisioni e ritrasmette.

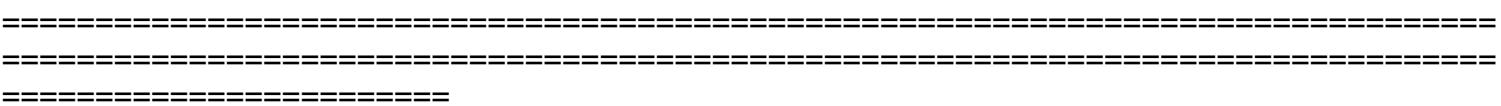
CSMA/CA: usato in Wi-Fi, evita collisioni aspettando canale libero.

Switch di Rete

Dispositivo che smista dati tra dispositivi in LAN.

Legge indirizzo MAC destinazione, inoltra frame alla porta corretta.

Costruisce dinamicamente una tabella MAC con gli indirizzi e le porte.



LEZIONE 16: Livelli di Rete, Trasporto, Sessione, Presentazione, Applicazione

Livello di Rete (Network Layer)

È il terzo livello del modello ISO/OSI e si occupa di inviare i dati tra dispositivi su reti diverse.

Funzioni principali:

Routing: trova il percorso migliore da mittente a destinatario

Forwarding: inoltra i pacchetti da un router all'altro

Indirizzamento logico: usa gli indirizzi IP per identificare i dispositivi

Frammentazione: divide pacchetti grandi in parti più piccole e li ricomponere alla destinazione

Routing

Path Determination: i router scelgono il percorso migliore usando tabelle e algoritmi (es. Dijkstra)

Packet Switching: i pacchetti passano da un router all'altro

Next-Hop Forwarding: ogni router invia il pacchetto al router successivo fino alla destinazione finale

Commutazione di pacchetto (Packet Switching)

Store-and-Forward: il router riceve tutto il pacchetto, lo controlla per errori e poi lo inoltra (più sicuro, ma più lento)

Cut-Through: il router inoltra il pacchetto appena legge l'indirizzo di destinazione (più veloce, ma non controlla errori)

Indirizzamento Logico - IP

IPv4: indirizzi a 32 bit (es. 192.168.1.1)

Classi di indirizzi:

A: grandi reti (es. 10.0.0.1)

B: medie reti (es. 172.16.0.1)

C: piccole reti (es. 192.168.1.1)

D: multicast

E: riservati

Subnet Mask: divide indirizzo IP in parte di rete e parte host (es. 255.255.255.0)

NAT (Network Address Translation)

Permette a più dispositivi con indirizzi privati di accedere a Internet con un unico indirizzo pubblico (es. il router di casa).

Livello di Trasporto (Transport Layer)

Gestisce la comunicazione end-to-end tra dispositivi:

Suddivide i dati in segmenti

Controlla errori e flusso dei dati

Gestisce la connessione (apertura, mantenimento, chiusura)

Socket

Interfaccia che combina indirizzo IP + porta per identificare una comunicazione specifica (es. 192.168.1.1:80).

Livello di Sessione (Session Layer)

Gestisce le sessioni tra applicazioni:

Mantiene e termina le connessioni

Controlla il dialogo in modalità half-duplex (una direzione per volta) o full-duplex (due direzioni simultanee)

Livello di Presentazione (Presentation Layer)

Traduce i dati tra formato applicativo e rete

Cripta/decripta per la sicurezza
Comprimi i dati per ottimizzare la trasmissione

Livello di Applicazione (Application Layer)
Fornisce servizi di rete alle applicazioni, come:
Trasferimento file
Email
Login remoto
Accesso a risorse di rete

LEZIONE 17 - Servizi e Funzioni del Sistema Operativo

4. Architetture dei Computer Servizi e Funzioni del Sistema Operativo Gestione dei Processi

Un processo è un'istanza di un programma in esecuzione. Su un sistema operativo multitasking, più processi possono girare contemporaneamente.

Il sistema operativo si occupa di:

- Creare e terminare processi
- Sospendere e riprendere processi
- Sincronizzare e far comunicare i processi
- Gestire situazioni di deadlock (blocchi)

I processi possono fare system call (chiamate di sistema) per:

- Eseguire altri processi (exec)
- Creare copie di se stessi (fork)
- Inviare segnali tra processi (wait/signal)
- Terminare processi (kill)

Gerarchia dei processi:

Il primo processo (es. systemd o init) è il padre di tutti gli altri, formando un albero padre-figlio.
Ogni processo ha un PID (Process ID), un PPID (Parent PID), e un USERID (utente che lo esegue).

Stati di un Processo

- Init: Processo appena caricato in memoria.
- Ready: Processo pronto, in attesa della CPU.
- Running: Processo in esecuzione.
- Waiting: Processo sospeso, in attesa di un evento (es. input da dispositivo).
- Swapped: Processo spostato in memoria virtuale (disco) in attesa di tornare in RAM.
- Zombie: Processo terminato ma ancora in memoria, in attesa che il padre lo "liberi".
- Terminated: Processo finito, memoria liberata.

Gestione della Memoria Principale (RAM)

- Assegna e libera memoria ai processi.
- Isola lo spazio di memoria di ogni processo per sicurezza e stabilità.
- Mappa gli indirizzi logici usati dai processi agli indirizzi fisici in RAM.
- Gestisce la memoria virtuale e il paging (spostamento di pagine tra RAM e disco).

Isolamento e Mappatura della Memoria

- Ogni processo ha un indirizzo virtuale che sembra continuo e privato.

Il sistema operativo traduce gli indirizzi virtuali in indirizzi fisici reali.

Vantaggi:

Sicurezza: i processi non possono accedere alla memoria degli altri.

Flessibilità: spazio virtuale più grande della RAM fisica disponibile.

Efficienza: il sistema ottimizza l'uso della memoria fisica.

Memoria Virtuale e Paging

La memoria virtuale usa RAM + disco (swap) per simulare più memoria.

La memoria è divisa in pagine.

Se una pagina non è in RAM (page fault), viene caricata dal disco.

Quando la RAM è piena, pagine meno usate vengono spostate su disco (swap out) per far spazio a pagine attive (swap in).

Gestione del File System

Il file system è un livello astratto che organizza i dati su memoria secondaria (dischi, SSD).

I file sono sequenze di byte permanenti, organizzati in cartelle (directory).

Le cartelle sono file speciali che puntano ad altri file (struttura ad albero).

Esempi di percorsi file:

Windows: C:\Tom\Data\one.txt (non case-sensitive)

Linux/Unix: ~Tom/src/minimumSpanningTree.c (case-sensitive, usa ~ per cartella home)

I processi possono:

Creare, aprire, leggere, scrivere, chiudere, cancellare file

Cambiare attributi (es. sola lettura)

Il file system protegge i file con permessi di accesso (lettura, scrittura, esecuzione) e gestisce le code di accesso.

Gestione dei Dispositivi Periferici

L'OS gestisce la comunicazione con periferiche (tastiere, stampanti, monitor).

Usa una memoria buffer per "parcheggiare" temporaneamente i dati in transito.

Per ogni periferica esiste un driver che traduce le richieste dell'OS in comandi per l'hardware.

Gestisce la coda di richieste per evitare conflitti di accesso.

Gestione della Memoria Secondaria

Memoria secondaria = dischi, SSD, memorie persistenti (dati non volatili).

Accesso più lento rispetto alla RAM.

L'OS gestisce:

Allocazione e deallocazione dello spazio

Gestione dello spazio libero

Ottimizzazione e serializzazione delle operazioni

Gerarchia della Memoria

Registri CPU

Cache

Memoria principale (RAM)

Memoria secondaria (dischi, SSD)

Memoria esterna (dispositivi rimovibili)

Protezione e Sicurezza

L'OS garantisce privacy e sicurezza per utenti e risorse.

Elementi chiave:

Autenticazione: verifica dell'identità (login con username e password).

Autorizzazione: verifica dei permessi per accedere a risorse.

Processi ereditano i permessi dell'utente associato.

Utenti possono essere raggruppati in gruppi per semplificare i permessi.

File e risorse hanno regole di accesso (lettura, scrittura, esecuzione).

Interfacce Utente (HCI)

Interfaccia Alfanumerica: terminale che mostra testo e riceve input da tastiera.

Interfaccia Grafica (GUI): finestre, icone, input da mouse o touchscreen.

=====

=====

=====

LEZIONE 19: Modello TCP/IP

Computer Networks: TCP/IP Model

1. Introduzione al TCP/IP Model

TCP/IP è un modello a 4 livelli usato per standardizzare la comunicazione dati tra reti diverse, inclusa Internet.

È il modello di fatto per la comunicazione in rete, noto per la sua robustezza e flessibilità.

2. Confronto tra ISO/OSI Model e TCP/IP Model

Livello TCP/IP Livello ISO/OSI Funzione principale

Application Application, Presentation, Session Fornisce servizi di rete direttamente alle applicazioni utente. Gestisce i protocolli applicativi.

Transport Transport Gestisce la trasmissione dati affidabile o non affidabile tra host.

Internet Network Gestisce l'indirizzamento e l'instradamento dei pacchetti dati.

Network Access Data Link, Physical Gestisce l'accesso alla rete fisica, l'indirizzamento hardware, e la trasmissione fisica.

3. Application Layer

Funzioni:

Fornisce servizi di rete alle applicazioni utente tramite vari protocolli.

Principali protocolli e funzioni:

DNS: Risoluzione dei nomi di dominio (es. www.cisco.com → 198.133.219.25)

HTTP/HTTPS: Navigazione web, con HTTPS che aggiunge sicurezza tramite crittografia

FTP: Trasferimento file client-server

SMTP/POP3/IMAP: Invio e ricezione email

DHCP: Assegnazione automatica di indirizzi IP

DNS (Domain Name System) – Come funziona:

Utente inserisce un dominio nel browser.

Query inviata a un resolver DNS.

Resolver contatta il root server per trovare il server TLD (.com, .org).

Resolver contatta il server TLD per trovare il DNS autoritativo del dominio.

Il server autoritativo fornisce l'indirizzo IP.

Il resolver restituisce l'indirizzo IP al browser, che si connette al server web.

HTTP / HTTPS

HTTP: Protocollo per il trasferimento di pagine web.

HTTPS: Estensione sicura di HTTP, cifra i dati scambiati.

Processo tipico di navigazione:

Inserimento URL → 2. Risoluzione DNS → 3. Connessione al server →

Invio richiesta GET → 5. Ricezione risposta (HTML) → 6. Rendering pagina.

FTP (File Transfer Protocol)

Usa due connessioni: controllo (comandi) e dati (file).

Supporta autenticazione con username/password o accesso anonimo.

Email Protocols

SMTP: Invia email da client a server o tra server.

POP3: Scarica email dal server al client e spesso cancella dal server.

IMAP: Gestisce email direttamente sul server, sincronizzando più dispositivi.

DHCP (Dynamic Host Configuration Protocol)

Assegna automaticamente indirizzi IP e altre configurazioni di rete.

Processo: DHCPDISCOVER → DHCPOFFER → DHCPREQUEST → DHCPACK.

4. Transport Layer

Funzioni:

Gestisce la trasmissione dati affidabile o non affidabile tra host.

Protocolli principali:

TCP (Transmission Control Protocol): Connessione orientata, garantisce consegna senza errori e in ordine.

UDP (User Datagram Protocol): Connessione senza garanzie, più veloce, usato per streaming o giochi.

TCP Header (campi principali)

Source/Destination Port: identifica app mittente/destinataria

Sequence Number: posizione dati

Acknowledgment Number: conferma ricezione

Flags: SYN (inizio connessione), ACK (conferma), FIN (fine connessione)

Checksum: controllo errori

UDP Header (campi principali)

Source/Destination Port

Length (header + dati)

Checksum

5. Internet Layer

Funzioni:

Indirizzamento logico e routing dei pacchetti dati.

Protocolli principali:

IP (Internet Protocol): instradamento dati (IPv4, IPv6).

ICMP (Internet Control Message Protocol): messaggi di errore e diagnostica (es. ping).

ARP (Address Resolution Protocol): associa indirizzi IP a indirizzi MAC.

ICMP

Usato per errori e diagnostica.

Tipi comuni: Echo Request/Reply (ping), Destination Unreachable, Time Exceeded, Redirect.

ARP

Mappa un IP a un indirizzo MAC nella rete locale.

Funziona tramite richieste broadcast e risposte unicast.

Mantiene una cache con mappature temporanee.

6. Network Access Layer

Funzioni:

Gestisce il livello fisico e di collegamento dati: trasmissione fisica, framing, indirizzamento hardware.

Tecnologie comuni:

Ethernet (LAN cablata)

Wi-Fi (wireless)