

2 Calcolatore e Software

2.1 Informazioni per programmare

2.1.1 Uso del Calcolatore

- Compilare ed eseguire i programmi da terminale per imparare e vedere gli errori che si commettono nello scrivere codice.
- provare e testare le proprie soluzioni a casa in modo da arrivare alla lezione successiva avendo capito gli algoritmi risolutivi.
- Importante saper googlare e filtrare le informazioni trovare (affidarsi a siti noti come StackOverflow).

2.1.2 Software ausiliari

- Editor di testo: è consigliato l'uso di IDE che non siano a completamento automatico in modo da imparare a scrivere codice, uno valido potrebbe essere Notepad++, Atom o VSCode.
- Compilatore: nel caso utilizzeremo GCC ANSI (specifico C*89).
- Ambiente: l'ambiente di sviluppo nel caso di Windows User è Ubuntu.

2.2 Architettura dell'elaboratore

2.2.1 Modello di Von Neumann

L'architettura di Von Neumann è una tipologia di architettura hardware per computer digitali programmabili a programma memorizzato la quale condivide i dati del programma e le istruzioni del programma nello stesso spazio di memoria, contrapponendosi all'architettura Harvard nella quale invece i dati del programma e le istruzioni del programma sono memorizzati in spazi di memoria distinti.

È necessario un mezzo di comunicazione fra la CPU e la memoria di lavoro (RAM), il modello di Von Neumann fornisce solo un modello di riferimento ad alto livello.

La comunicazione può essere di diversi tipi, ad esempio punto a punto o a Bus. La tipologia di comunicazione scelta per essere applicata nei calcolatori è quella a Bus perché è flessibile (scalabile facilmente), le periferiche guaste non creano problemi (solo il bus è fondamentale) e non è eccessivamente costosa. Come prestazioni il fatto che il canale non può supportare più periferiche per volta non è un grosso problema perché tanto la CPU ne svolge una per volta.

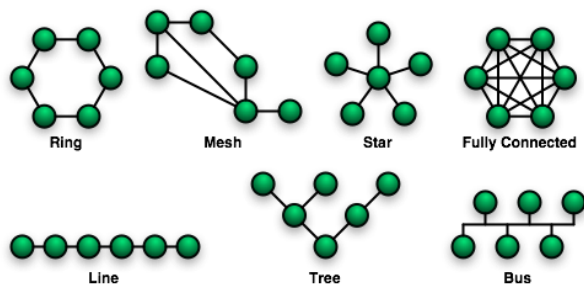


Figure 1: Tipologie della comunicazione

2.2.2 Bus di sistema

Il Bus di sistema è composto da tre livelli:

1. Bus di indirizzi: transitano gli indirizzi di memoria dell'informazione che viene scambiata.
2. Bus di dati: transita l'informazione, quindi o viene letta la memoria o viene sovrascritta la memoria.

3. Bus di controllo: segnale di controllo (ACK) che avvisa quando è disponibile l'informazione sul bus e coordina le attività dando il via libera alla successiva operazione.

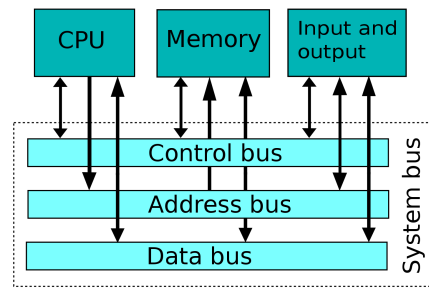


Figure 2: Bus di sistema

La CPU sceglie gli indirizzi che andranno ad interagire sul Bus indirizzi, il dispositivo fa riferimento all'indirizzo sul Bus impostato dalla CPU, infatti:

- il bus indirizzi è monodirezionale dalla CPU alla memoria
- il bus dati è bidirezionale, potremmo esserci periferiche che possono sia leggere che scrivere in memoria (stampante-scanner), altre che svolgono solo output (schermo monitor) e alcune solo input (tastiera e mouse).

2.2.3 Memoria di lavoro RAM

Nomenclatura:

- Indirizzo di memoria: cella specifica nella memoria
- Parola di memoria: contenuto di una cella di memoria
- Dimensione della parola: dimensione della cella di memoria in numero di bit, gli attuali calcolatori hanno almeno 32 bit (in generale per l'aritmetica binaria è comodo un multiplo di 8).

Si può interagire con la RAM in due modi:

- Modalità lettura: si copia la parola ad un certo indirizzo e si mette su bus (possono essere necessari anche più giri del bus)
- Modalità scrittura: si copia dal bus e si incolla in un indirizzo.

Se indichiamo con k il numero di bit degli indirizzi, il numero di celle descrivibili sono esattamente 2^k , e questo viene detto spazio di indirizzamento.

La memoria di lavoro viene anche chiamata RAM (Random Access Memory) perché il tempo di accesso a qualsiasi cella è costante e indipendente dalla posizione della cella in memoria (memoria tabellare), quindi si ha un tempo di accesso sequenziale.

La caratteristica principale della RAM è che è una memoria volatile (non permanente), ovvero una volta che non è più alimentata perde il contenuto e all'accensione non si ha un contenuto deterministico (0,1 casuali).

La CPU interagisce direttamente con la RAM.

2.2.4 CPU

La CPU, o unità centrale di elaborazione o processore centrale (central processing unit), nell'architettura di von Neumann di un calcolatore indica un'unità fondamentale (o sottosistema) logico e fisico che sovrintende alle funzionalità logiche di elaborazione principali di un computer.

Il Bus è l'insieme di linee a tensione alta o bassa che corrispondono all'1 e 0 che collegano i moduli di un sistema di elaborazione.

La CPU ha il ritmo scandito dal clock che funge da metronomo, la velocità di clock misura il numero di cicli eseguiti dalla CPU ogni secondo (misurati in GHz gigahertz).

Un registro, in informatica e nell'architettura dei calcolatori, è una piccola parte di memoria utilizzata per velocizzare l'esecuzione dei programmi fornendo un accesso rapido ai valori usati più frequentemente:

- **ALU** (arithmetic-logic unit) [U1]:
Una unità aritmetica e logica (arithmetic and logic unit) è una tipologia particolare di processore digitale che si contraddistingue per essere preposta all'esecuzione di operazioni aritmetiche o logiche.
L'ALU recupera i dati dai registri del processore (2 ingressi e il selettore), processa i dati nell'accumulatore e provvede a salvare il risultato nel registro di uscita (1 risultato).
- **CU** (control unit) [U2]:
L'unità di controllo è un componente delle CPU che ha il compito di coordinare tutte le azioni necessarie per l'esecuzione di una istruzione e di insiemi di istruzioni. Le azioni che coordinano i vari settori della CPU.
- **PC** (program counter) [R1]:
Il program counter è un registro della CPU la cui funzione è quella di conservare l'indirizzo di memoria della prossima istruzione (in linguaggio macchina) da eseguire. È un registro puntatore cioè punta a un dato che si trova in memoria all'indirizzo corrispondente al valore contenuto nel registro stesso.
- **IR** (instruction register) [R2]:
Il registro istruzione (instruction register) è un registro della CPU che immagazzina l'istruzione in fase di elaborazione. Ogni istruzione viene caricata dentro il registro istruzione che la deposita mentre viene decodificata, la prepara per l'esecuzione e quindi la elabora.
- **SP** (stack pointer) [R3]:
Lo stack pointer (ESP) è un registro dedicato alla CPU che contiene l'indirizzo della locazione di memoria occupata dal top dello stack. Lo stack viene allocato e deallocato continuamente quindi può esser facile perder traccia della sua "testa".
- **RF** (register file) [R4]: Il register file o banco dei registri è la memoria più vicina al centro di elaborazione in quanto essa sta proprio dentro la CPU.
- **PSW** (process status word) [R5]:
Il registro di stato (status register o flag register) è un insieme di flag presenti nella CPU che indicano lo stato di diversi risultati di operazioni matematiche. Ci sono diversi flag, un esempio è OF overflow flag.
- **MAR** (memory address register) [R6]:
Il memory address register (MAR) è un registro temporaneo (buffer) della CPU collegato al Bus indirizzi contenente l'indirizzo della locazione di memoria RAM in cui si andrà a leggere o scrivere un dato. In altre parole, il MAR contiene l'indirizzo di memoria del dato a cui la CPU dovrà accedere. Tutti i risultati di un'elaborazione che devono essere immagazzinati in memoria transitano prima per il registro MDR e da esso raggiungono poi la esatta locazione di memoria.
- **MDR** (memory data register) [R7]:
Il memory data register (MDR) è un registro a cui la unità aritmetica e logica (ALU) ha accesso diretto e che contiene momentaneamente i dati da/per la CPU. L'MDR, insieme al memory address register (MAR), interfaccia quindi la CPU con la memoria centrale (MC), tutti i dati e le istruzioni che dalla memoria devono essere elaborati nel processore, transitano per il registro MDR e successivamente, da questo, raggiungono gli opportuni registri per l'elaborazione vera e propria.

2.2.5 Memoria di massa

Serve per memorizzare in memoria dei dati che devono essere conservati, infatti essa è una memoria non volatile, quindi se non alimentata non perde il contenuto.

Svolge la funzione di stoccaggio (magazzino) per tutte le informazioni che non servono nell'immediato.

Come pro ha che è poco costosa e come dimensioni è diversi ordini di grandezza maggiore della RAM, come contro ha che i tempi di accesso sono elevati.

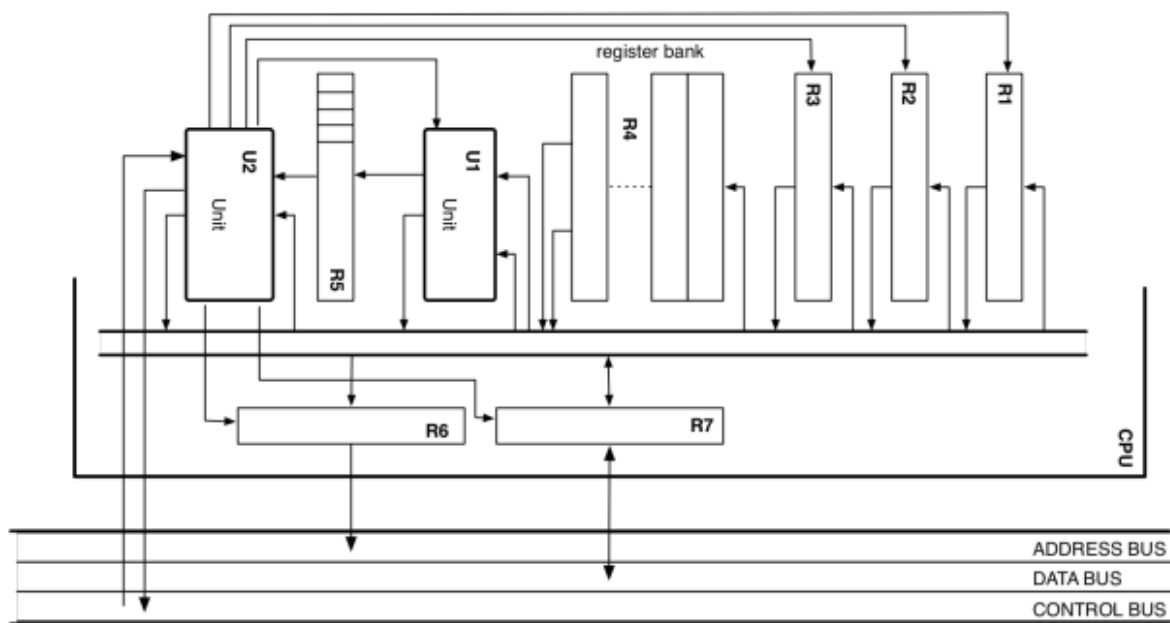


Figure 3: Componenti della CPU

Prima venivano usati gli Hard Disk, ora si usano i dischi a stato solido (SSD) che sono un compromesso fra efficienza e prezzo.

2.2.6 Insieme di istruzioni della CPU

Le istruzioni supportate dalla CPU sono quelle basilari, che combinate fra di loro sono in grado di formare istruzioni complesse. Le principali sono:

- Aritmetiche
- Logiche
- Sul flusso di controllo (es: cicli) che infatti iniziano con 'j' che sta per 'Jump'

Nel caso del 'Jump' l'incremento del contatore dell'operazione corrente fa un'operazione inutile, ma tutto sommato è più efficiente così perché si cerca di ottimizzare il caso medio (che spesso è quello che accade più di frequente) ovvero l'andamento lineare del programma (senza cicli o salti). nel caso di salto si sovrascrive l'indirizzo (gestito dal controllo sul PSW).

Il linguaggio ASSEMBLY è l'unico linguaggio interpretato dal calcolatore ed è un insieme di istruzioni composte da due parti:

- Codice operazioni (OP:CODE)
- Operando/i

2.2.7 Gerarchia di memoria

Le diverse memorie di un calcolatore dalla più veloce alla più lenta:

1. Registro:

Un registro (in inglese: processor register), in informatica e nell'architettura dei calcolatori, è una piccola parte di memoria utilizzata per velocizzare l'esecuzione dei programmi fornendo un accesso rapido ai valori usati più frequentemente e/o tipicamente, i valori correntemente in uso in una determinata parte di un calcolo.

2. Cache:

La memoria cache (in inglese cache memory, memory cache o CPU cache) è una memoria veloce (rispetto alla memoria principale), relativamente piccola, non visibile al software e completamente gestita dall'hardware, che memorizza i dati più recentemente usati della memoria principale (MM - Main Memory) o memoria di lavoro del sistema.

La funzione della memoria cache è di velocizzare gli accessi alla memoria principale aumentando le prestazioni del sistema. Nei sistemi multiprocessori con memoria condivisa permette inoltre di ridurre il traffico del bus di sistema e della memoria principale, che rappresenta uno dei maggiori colli di bottiglia di questi sistemi. La memoria cache fa uso della tecnologia veloce SRAM, contro una più lenta DRAM della memoria principale, connessa direttamente al processore.

Ha un tempo di accesso che aumenta in base alla grandezza della memoria, quindi si deve trovare un balace tra grandezza e prestazioni.

3. RAM: spiegata nel capitoletto prima

4. Memoria di massa: spiegata nel capitoletto prima

Livelli della gerarchia di memoria

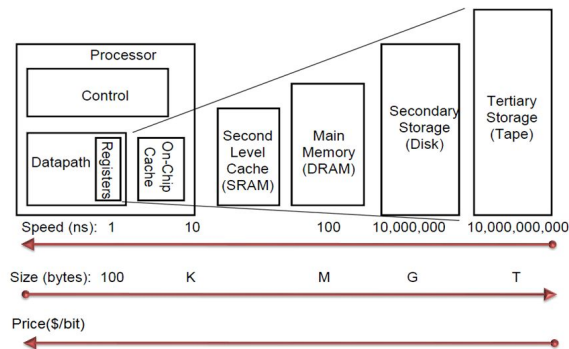


Figure 4: Piramide della gerarchia della memoria

2.3 Organizzazione del sistema operativo

È presente un'organizzazione a livelli:

1. Applicazione (tipo programma in C)
2. Applicazioni che lavorano a basso livello (compilatori, interpreti di comandi, software di sistema)
3. Sistema operativo:
 - (a) Gestore dei file
 - (b) Gestore delle periferiche
 - (c) Gestore della memoria lavoro RAM
 - (d) Kernel
4. Hardware

La visione dell'organizzazione del sistema operativo viene definita **semplificata ed estesa**:

- **Semplificata:**
maschera la complessità dei dettagli delle operazioni richiedendo solo delle istruzioni generali che poi saranno applicate a basso livello con comandi specifici non visibili dall'esterno (nasconde ai programmatori i dettagli dell'hardware fornendogli una API conveniente e facile da usare). Ad esempio per stampare non vengono richieste tutte le specifiche della stampante, basta dire l'operazione richiesta e poi la gestione delle singole operazioni di basso livello vengono svolte

- **Estesa:**
il calcolatore non si preoccupa di vedere e valutare la disponibilità delle risorse, sarà poi il kernel che dovrà gestire la CPU per far sì che vengano eseguiti tutti i programmi.

2.3.1 Kernel

Separazione tra **meccanismi e politiche**:

Con l'espressione separazione tra meccanismi e politiche nell'ambito dei sistemi operativi si intende la distinzione fra come eseguire qualcosa (meccanismo) e che cosa si debba fare, ovvero quali scelte operare, in risposta ad un certo evento (politica).

Ad esempio nel caso del Kernel il meccanismo è la divisione del tempo della CPU (come gestire la risorsa), mentre la politica è la strategia su come ottimizzare il processo (round robbin).

Spiegazione del Kernel:

C'è uno strato del sistema operativo (più vicino alla macchina fisica) chiamato Kernel che gestisce la CPU. Esso organizza ed esegue i programmi offrendo una visione migliore di quello che la macchina effettivamente ha disponibile al momento, come se rendesse disponibile un processore per ogni processo.

Definizione di processo:

Definition 2.1 (Processo). *un programma quando è in esecuzione viene definito processo, ovvero se il suo stato è attivo e ha dei dati su cui svolgere delle operazioni.*

Chiaramente un calcolatore necessita che vengano eseguiti più programmi in parallelo (o almeno così deve sembrare), quindi viene adottata la politica del **Round Robbin**:

- **Divisione del tempo di elaborazione:**
solo una applicazione alla volta può essere attiva, quindi ogni processo ha a disposizione un pò di tempo di esecuzione sulla CPU.
- **Stato ready:**
Quando un processo è attivo ed è pronto per essere eseguito è nello stato Ready, ma è in attesa di andare in esecuzione.
- **Stato run:**
Il processo è in esecuzione sulla CPU perchè è il suo turno e resta in esecuzione per un quantitativo di tempo limitato.
- **Stato wait:**
Se al processo serve un dato derivante da un evento esterno non ancora accaduto passa allo stato wait (di attesa) e una volta fornitogli il dato torna in coda come ready.
- Quando il processo finisce, o finisce il tempo a sua disposizione sulla CPU (finisce il suo quanto di tempo) passa in lista di attesa come ready to be executed.

La politica di Round Robbin viene definita ad assegnamento ciclico di quanti di tempo, chiaramente non è l'unica politica, ne esistono altre che gestiscono l'assegnamento diversamente (es: priorità ai processi brevi).

Grazie al Round Robbin l'utente avrà l'impressione che più processi siano in esecuzione in parallelo, anche se in realtà il Kernel fa alternare i processi facendoli eseguire parzialmente e un pò ciascuno, ma sempre uno solo per volta in esecuzione (stato run).

2.3.2 Gestione della memoria di lavoro (MMU)

La Memory Management Unit (MMU), o unità di gestione della memoria, indica una classe di componenti hardware che gestisce le richieste di accesso alla memoria generate dalla CPU.

Aiuta a rendere disponibile ad un processo dei dati nella memoria che hanno nome, posizione e effettiva cella di memoria del dato disaccoppiati.

Infatti si sistemano gli indirizzi di memoria rilocando (aggiornando) gli indirizzi (ad esempio quando finiscono dei processi e si libera dello spazio).

la memoria può essere suddivisa in diversi modi:

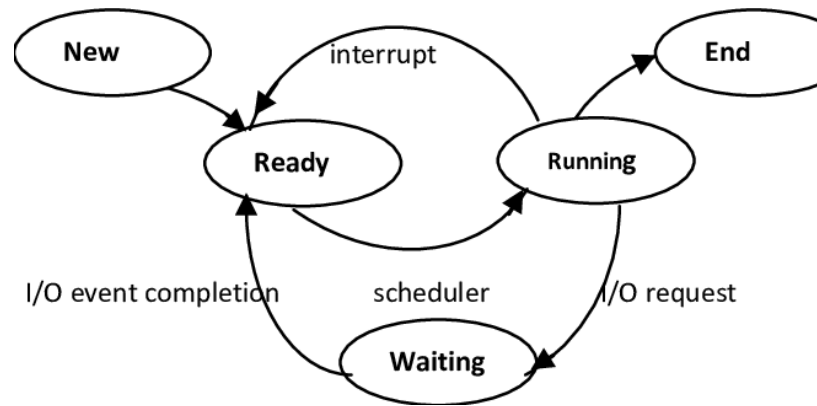


Figure 5: Diagramma dei processi

- Memoria suddivisa a pagine:
La memoria è suddivisa in pagine di egual dimensione, quindi si ha solo una frammentazione della memoria interna alla pagina.
- Segmentazione della memoria:
La memoria dedicata ai processi viene assegnata in base alla loro dimensione uno dopo l'altro, quindi una volta finiti dei processi si creano dei buchi che vengono riempiti da nuovi processi. Si ha una frammentazione della memoria esterna.

Entrambi i metodi hanno dei pro e dei contro e vengono utilizzati entrambi.

Swapping della memoria:

Definition 2.2 (Swapping della memoria). *passare dal memoria di massa (memoria virtuale) a RAM e viceversa.*

Quando la RAM si riempie perchè ci sono troppi processi pending (programmi in stato ready o wait), ciò che non ci sta nella RAM viene salvato nella memoria virtuale che è una parte di memoria ricavata dalla memoria di massa. Ora con le SSD lo swapping è molto rapido, prima con gli Hard Disk si notava il rallentamento casuato dalla ricerca nella memoria di massa.

2.3.3 Gestore delle periferiche

I driver hanno informazioni su come gestire il dispositivo della periferica (dipendono dal dispositivo fisico e delle sue singole specifiche).

I driver svolgono principalmente questi due compiti:

- Gestire microoperazioni a basso livello
- Fornire primitive di accesso semplificate (API)

Ci sono anche meccanismi per periferiche che creano una cosa e gestiscono le politiche di accesso (sempre con una visione semplificata ed estesa).

Ogni periferica interagisce con sistemi di calcolo con politiche differenti in base al tipo di dato fornito / richiesto:

- Polling:
Controllo periodico della periferica con una frequenza significativa, quindi serve un buffer per memorizzare informazioni non ancora raccolte
(es: ogni tot la prof chiede se ci sono domande).
- Interrupt:
Flag per sollevare un 'interrupt' e segnalare una richiesta di interazione interrompendo il flusso dell'evento che è avvenuto. Viene salvato lo stato e poi mandata la risposta ad interrupt, infine si ripristina lo stato e si prosegue con l'operazione di prima.

(es: se hai domanda chiedi alla prof, ma solo se sono poche se no non riesce a spiegare e fare un discorso).

Per gestire le richieste delle periferiche di accesso alla memoria c'è il **DMA (Direct Memory Access)** che è un coprocessore programmato dalla CPU e viene usato per spostare dati di grosse dimensioni (sposta dalla cella x alla cella y, un numero n di celle).

Il DMA rende il processo molto efficiente perchè non rallenta la CPU, esso lavora solamente nei ritagli di tempo liberi del Bus e così la CPU può fare altro e vengono anche riempiti i tempi morti del Bus.

2.3.4 File System

Il File System è il gestore della memoria di massa e mostra una visione semplificata e logica dei file salvati in memoria permanente (potrebbe anche essere una memoria condivisa).

L'organizzazione è logica ad albero, dove ogni nodo è:

- File e i suoi attributi.
- Directory (folder) che è un elemento logico che contiene o file o altre folder.

Di conseguenza dobbiamo avere tutto il percorso (path) per riuscire ad aprire un file salvato nel File System.

Attributi dei file:

Gli attributi di file sono proprietà specifiche di un file, ad esempio:

- A Attributo di file di archivio
- R Attributo di file di sola lettura
- S Attributo di file di sistema
- H Attributo di file nascosto
- I Attributo di file non indicizzato