

# Sistemi di Elaborazione in Rete

Presentazione del corso

# Caratteristiche essenziali del corso

## ◆ Obiettivi:

- Illustrare i concetti di base relativi ai Sistemi Operativi ed alle Reti di Calcolatori
- Descrivere il funzionamento della rete Internet, accennando allo sviluppo di applicazioni concorrenti e distribuite

## ◆ Valore formativo:

- 5 CREDITI

## ◆ Prerequisiti:

- Fondamenti di informatica
- Conoscenza di Java e dei rudimenti della programmazione object-oriented
- Conoscenze di base sull'architettura dei calcolatori

# Programma (1)

## Parte I: Sistemi Operativi

### ■ Introduzione ai Sistemi Operativi

- Concetti fondamentali, Funzioni e servizi tipicamente implementati, Architettura di un sistema operativo

### ■ Gestione dei Processi

- Gestione di Processi e Thread, Scheduling, Comunicazione fra processi

### ■ Gestione della Memoria Principale

- Tecniche per la gestione della memoria, Swapping, Paginazione e Segmentazione, Memoria Virtuale

### ■ File System e Dispositivi di I/O

# Programma (2)

## Parte II: Reti di Calcolatori

### ■ Introduzione alle reti di calcolatori

- Tipi di reti, topologie
- Architettura delle reti di calcolatori
- Strutturazione a livelli, Concetto di protocollo
- Servizi Connection-less e Connection-oriented

### ■ Il modello di riferimento ISO OSI

- Livelli Physical, Data Link, Network, Transport, Session, Presentation, Application

# Programma (3)

## Parte III: La rete Internet

### ■ Protocolli dei livelli inferiori

- Protocollo di rete IP
- Protocolli di trasporto UDP e TCP

### ■ Architettura di Internet

- Indirizzamento, Nomi e Domini, DNS

### ■ La posta elettronica

- Formato degli indirizzi e dei messaggi, Protocollo di trasporto (SMTP), Protocollo di consegna (POPv3/IMAPv4)

### ■ Il WEB

- Tecnologie e protocolli di base (URL, HTTP, HTML/XML)
- Motori di ricerca
- Il web dinamico (pagine JSP)
- I Web Services (?)

# Programma (4)

## ◆ Esercitazioni:

- Processi e thread in Java
- Internet e protocollo TCP/IP
- Programmazione di rete in Java
  - Socket
- Applicazioni distribuite su Web
  - Pagine JSP e Servlet
  - Web Services (?)

# Materiale Didattico

## ◆ Libri di Testo

- A. Silberschatz, P. Galvin, G. Gagne: *Sistemi Operativi*, Addison-Wesley, 6a ed., 2002.
- A. Tanenbaum, Reti di calcolatori, 4a ed., Prentice Hall, 2003

## ◆ Testi Complementari

- P. Ancilotti, M. Boari, A. Ciampolini, G. Lipari: *Sistemi Operativi*, McGraw-Hill, 2004
- M. L. Liu, Distributed Computing: Principles and Applications, Prentice Hall, 2003

## ◆ Altro materiale didattico

- Lucidi delle lezioni

# Esame

## ◆ Prova scritta

- esercizi di programmazione Java e domande “teoriche”

## ◆ Progetto:

- presentazione di una relazione su un tema (concordato con il docente) inerente i contenuti del corso
- Sviluppo di un'applicazione

## ◆ Prova orale:

- verifica della conoscenza di argomenti trattati a lezione, oppure

## ◆ Propedeuticità:

- Esame di Fondamenti di Informatica



# Orario

	LUNEDI'	MARTEDI'	MERCOLEDI'	GIOVEDI'	VENERDI'
9.00					
10.00					
11.00					
12.00					
15.00	Esercitazione				
16.00	Esercitazione		Lezione		
17.00	Esercitazione		Lezione		
18.00			Lezione		

**Totale ore = 48**  
(30 lezione + 18 esercitazione)

# Altre informazioni

## ◆ Docente: Luigi Pontieri

- ufficio: ICAR-CNR (c/o DEIS), cubo 41C, I° piano
- e-mail: [pontieri@icar.cnr.it](mailto:pontieri@icar.cnr.it)

## ◆ Tutor: Luigi Labonia

- e-mail: [luigi.labonia@giustizia.it](mailto:luigi.labonia@giustizia.it)

## ◆ Sito web del corso:

<http://www.icar.cnr.it/pontieri/didattica/SER.html>

# Introduzione ai Sistemi Operativi

**Definizione di Sistema Operativo**

**Classificazione dei Sistemi di Elaborazione**

# Argomenti

---

- ◆ Cosa è un Sistema Operativo?
  - Il Sistema Operativo nel contesto di un calcolatore
- ◆ Architettura di base del calcolatore
  - La macchina di Von Neumann
- ◆ Classificazione dei sistemi di elaborazione

# Cosa è un Sistema Operativo?

## ◆ Sistema Operativo:

- programma che agisce come intermediario tra l'utente e l'hardware di un Sistema di Elaborazione.
- Il primo sistema operativo è stato realizzato negli anni '50

## ◆ Sistema di Elaborazione

- è uno strumento programmabile per la rappresentazione, la memorizzazione e l'elaborazione delle informazioni
- detto anche “computer”, “calcolatore” o “elaboratore”

# Hardware e Software

- ◆ Due tipi di componenti in un Sistema di Elaborazione
  - Hardware: la struttura fisica del calcolatore
  - Software: l'insieme dei programmi che consentono di impiegare l'hardware per svolgere compiti utili all'utente

# Software e macchine virtuali

## ◆ Hardware: la macchina reale

- sistema di dispositivi di varia natura (elettronici, elettromagnetici, elettromeccanici, ottici ...)
- le operazioni (**istruzioni**) che l'hardware sa eseguire direttamente rappresentano le frasi del **linguaggio macchina** del calcolatore
- le istruzioni del linguaggio macchina sono molto semplici e il calcolatore può eseguirle in modo molto efficiente

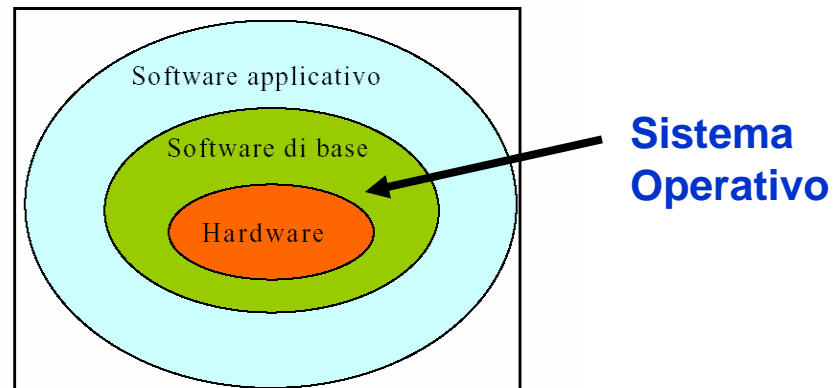
## ◆ Software:

- insieme dei programmi che possono essere eseguiti dal calcolatore
- mostra il calcolatore agli utenti come una **macchina virtuale** più semplice da usare rispetto all'hardware sottostante

# Software e macchine virtuali

## ◆ Macchine virtuali

- Semplificano la comunicazione fra utenti e hardware
- Le varie macchine (insiemi di operazioni) sono via via più astratti:
  - più vicini alla logica dell'utente e più lontani dalla logica del calcolatore come dispositivo elettronico
- L'unico responsabile dell'esecuzione del SW è l'HW sottostante



## ◆ Il software di base

➔ macchina virtuale più semplice da gestire e programmare

## ◆ Il software applicativo

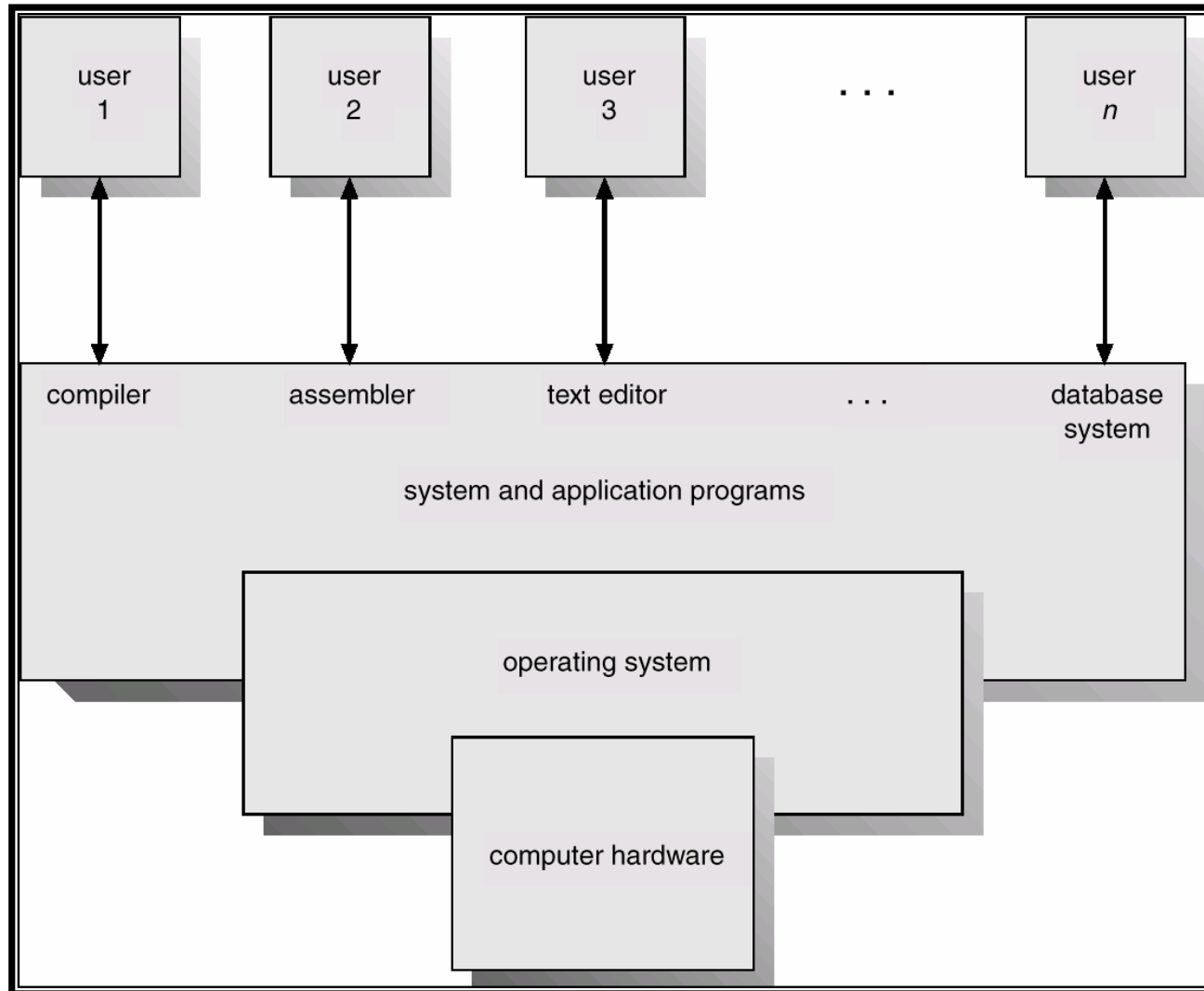
➔ macchina virtuale utilizzabile per soddisfare esigenze specifiche degli utenti



# Componenti di un sistema di elaborazione

- ◆ Utenti (persone, macchine, altri computer).
- ◆ Hardware
  - (CPU, Bus, memoria, dispositivi di I/O)
- ◆ Programmi di sistema e applicazioni
  - Specificano i modi in cui le risorse del sistema possono essere usate per risolvere un problema (compilatori, database, video games, fogli elettronici)
- ◆ Sistema Operativo
  - Permette di eseguire i programmi utente, aiutando gli utenti a risolvere i loro problemi tramite il computer
  - Semplifica e ottimizza l'uso dell'hardware

# Componenti di un sistema di elaborazione



# Funzioni principali di un SO

- ◆ Semplificare lo sviluppo e l'esecuzione dei programmi
  - Fornire servizi per usare le risorse HW (system calls)
  - Garantire l'indipendenza del SW applicativo dall'HW
- ◆ Realizzare politiche di gestione delle risorse hw
  - Evitare conflitti di accesso alle risorse, garantendo la corretta esecuzione dei programmi
  - Definire criteri con cui assegnare una risorsa a fronte di più richieste contemporanee

# Diversi ruoli del SO

## ◆ Nucleo (kernel)

- il solo programma sempre caricato in memoria

## ◆ Allocatore di Risorse

- gestisce e alloca risorse in modo efficiente
- funziona come mediatore tra risorse in conflitto

## ◆ Programma di controllo

- accetta e soddisfa le richieste degli utenti/programmi
- controlla l'esecuzione dei programmi e delle operazioni di I/O

## ◆ Macchina estesa

- simula una macchina virtuale più facile da programmare, astruendo dai dettagli tecnologici dell'hardware
- costituisce una base sulla quale è possibile scrivere programmi applicativi

# Architettura dell'hardware (cenni)

# Macchina di Von Neumann

## ◆ Modello di riferimento di base

- utile per descrivere e analizzare un sistema di elaborazione
- Von Neumann progettò, nel 1945, il primo calcolatore con programmi memorizzabili anziché codificati mediante cavi e interruttori

## ◆ La macchina di Von Neumann è un modello semplificato

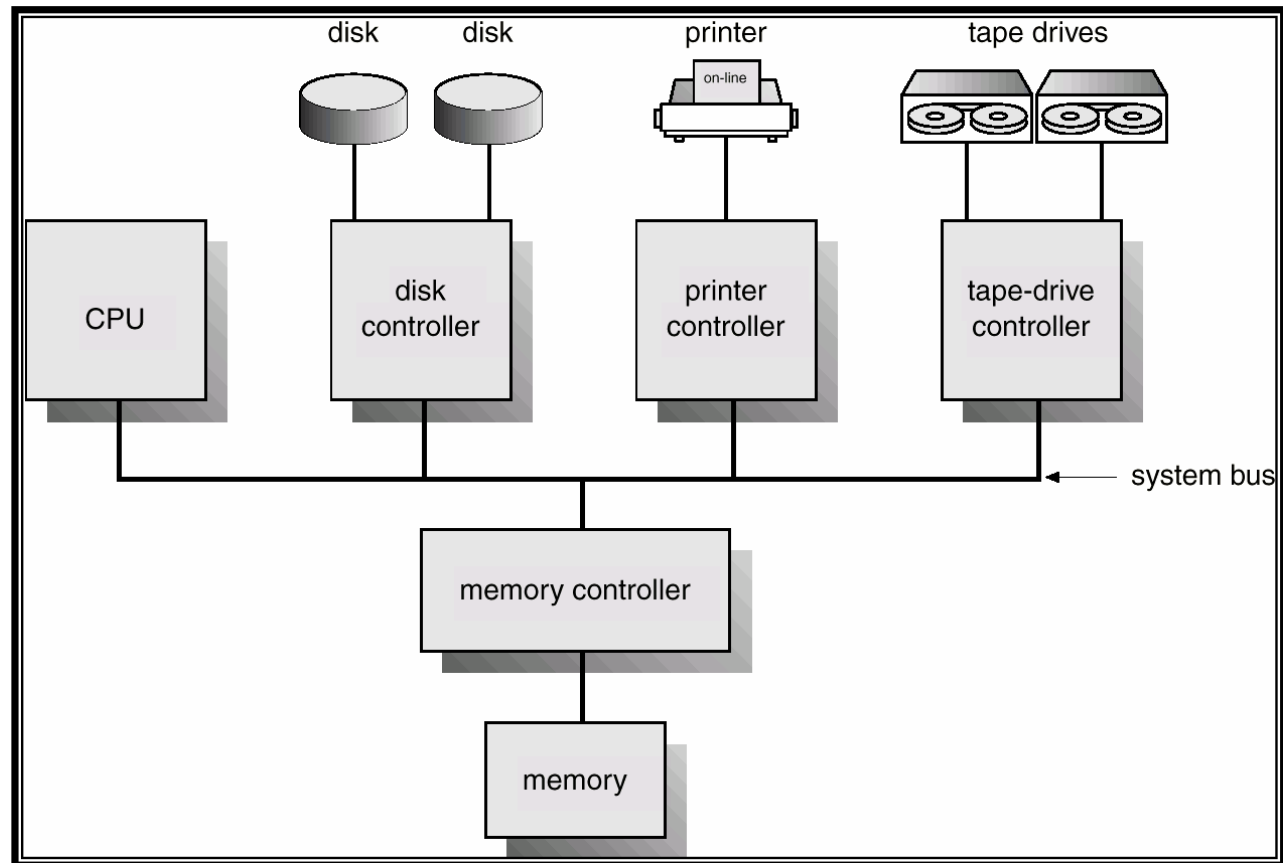
- L'architettura dell'hardware di un calcolatore reale è molto complessa
- esistono varie architetture per un sistema di calcolo, che possono discostarsi anche notevolmente dallo schema della macchina di Von Neumann

# Macchina di Von Neumann:

## 4 tipi di componenti funzionali

- ◆ unità centrale di elaborazione (CPU)
  - esegue istruzioni per l'elaborazione dei dati
  - svolge anche funzioni di controllo
- ◆ memoria centrale
  - memorizza e fornisce l'accesso a dati e programmi in esecuzione
- ◆ interfacce di ingresso e uscita
  - componenti di collegamento con le periferiche del calcolatore
- ◆ bus
  - svolge funzioni di trasferimento di dati e di informazioni di controllo tra le varie componenti funzionali

# Architettura dell'hardware

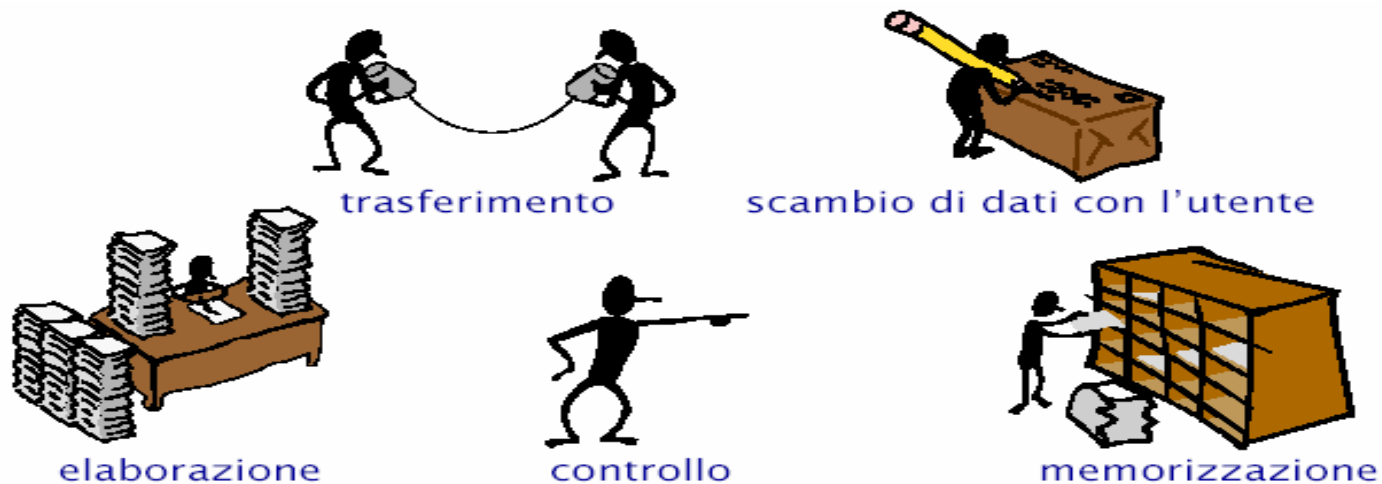




# Macchina di Von Neumann

Il funzionamento di un calcolatore è descrivibile in termini di poche **componenti** (macro-unità) **funzionali**

- ogni componente è specializzata nello svolgimento di una tipologia omogenea di funzionalità
- *eccezione*: l'unità centrale di elaborazione svolge sia funzionalità di elaborazione sia di controllo



# Struttura della Memoria

## ◆ Memoria Centrale

- celle di memoria accessibili direttamente dalla CPU in cui sono mantenuti dati e programmi in esecuzione

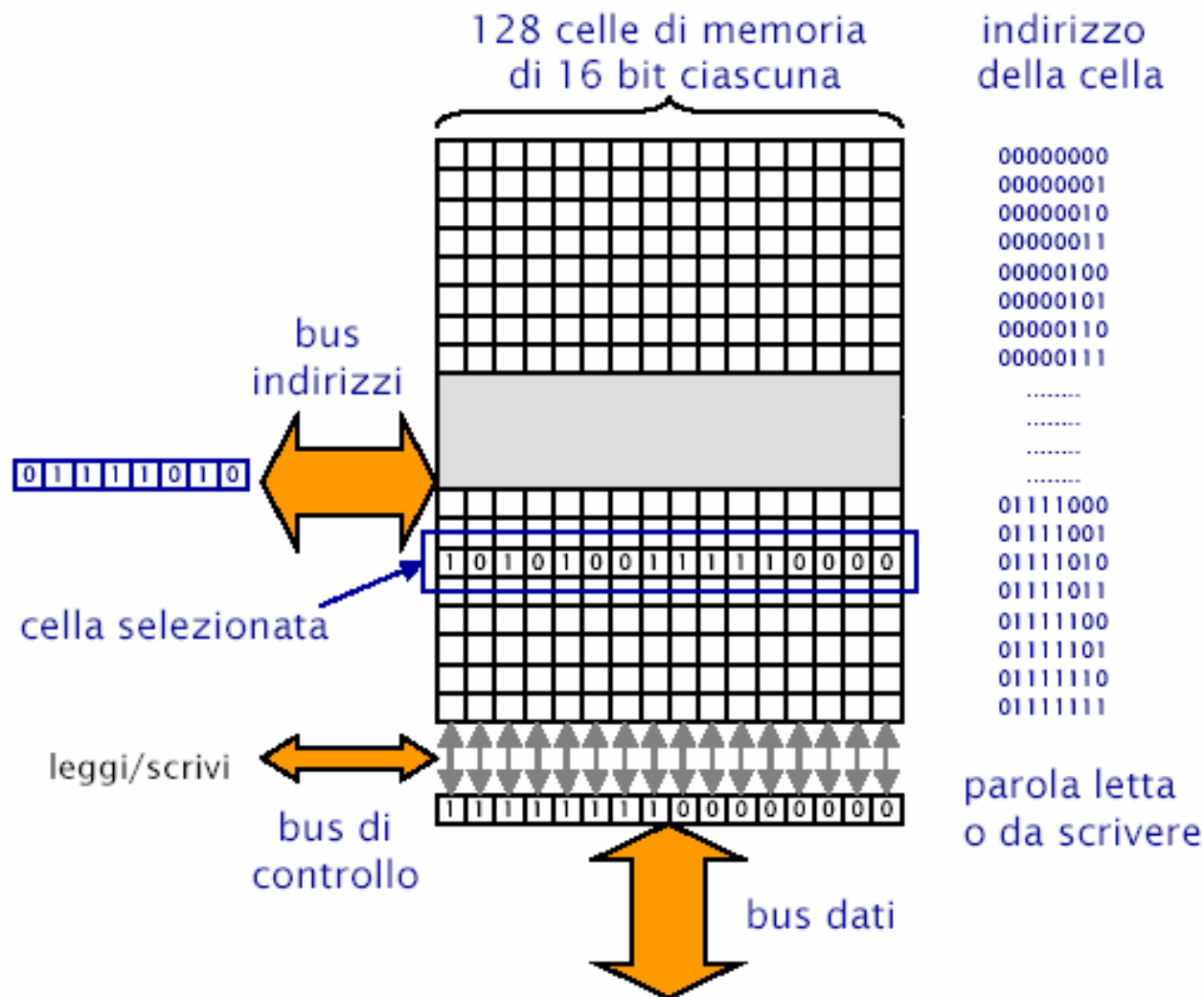
## ◆ Memoria Secondaria

- estensione della memoria centrale di grandi dimensioni e non volatile
- serve come archivio permanente
- Varie tecnologie: dischi magnetici/ottici, nastri, flash
- Hard disk (fisso) vs. unità esterne (rimovibili)

# Memoria centrale

- ◆ E' costituita da **celle** (o **locazioni**)
  - ogni cella può contenere una quantità fissata di memoria (numero di bit), detta **parola**
- ◆ Ogni cella è caratterizzata da
  - **indirizzo**, un numero che identifica la cella per accedervi
  - **valore**, la sequenza di bit memorizzata in essa
- ◆ La memoria fornisce le operazioni di:
  - **lettura**: consultazione del valore di una cella con un dato indirizzo
  - **scrittura**: modifica del valore di una cella con un dato indirizzo

# Memoria centrale



# CPU

---

# Unità centrale di elaborazione (CPU)

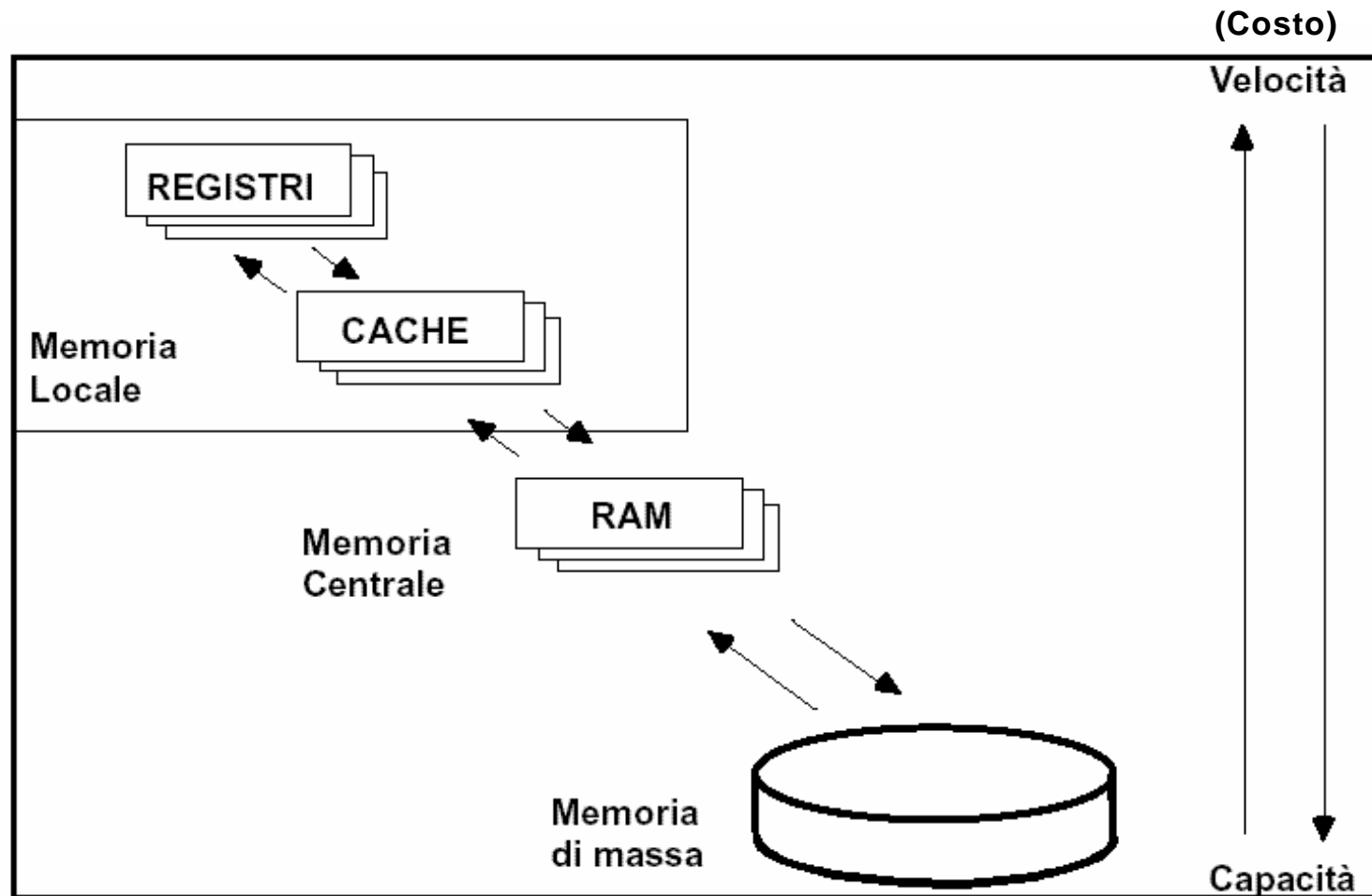
- ◆ L'**unità centrale di elaborazione** (CPU) o **processore** presiede all'esecuzione di un programma
  - il programma è memorizzato in memoria centrale come sequenza di istruzioni del linguaggio macchina
- ◆ La CPU è composta da:
  - Unità di Controllo (CU)
  - Unità Logico-Aritmetica (ALU)
  - Registri: celle di memoria (veloci) usati dalle due unità funzionali (CU e ALU)
  - Clock

# Tipi di memoria in un calcolatore

---

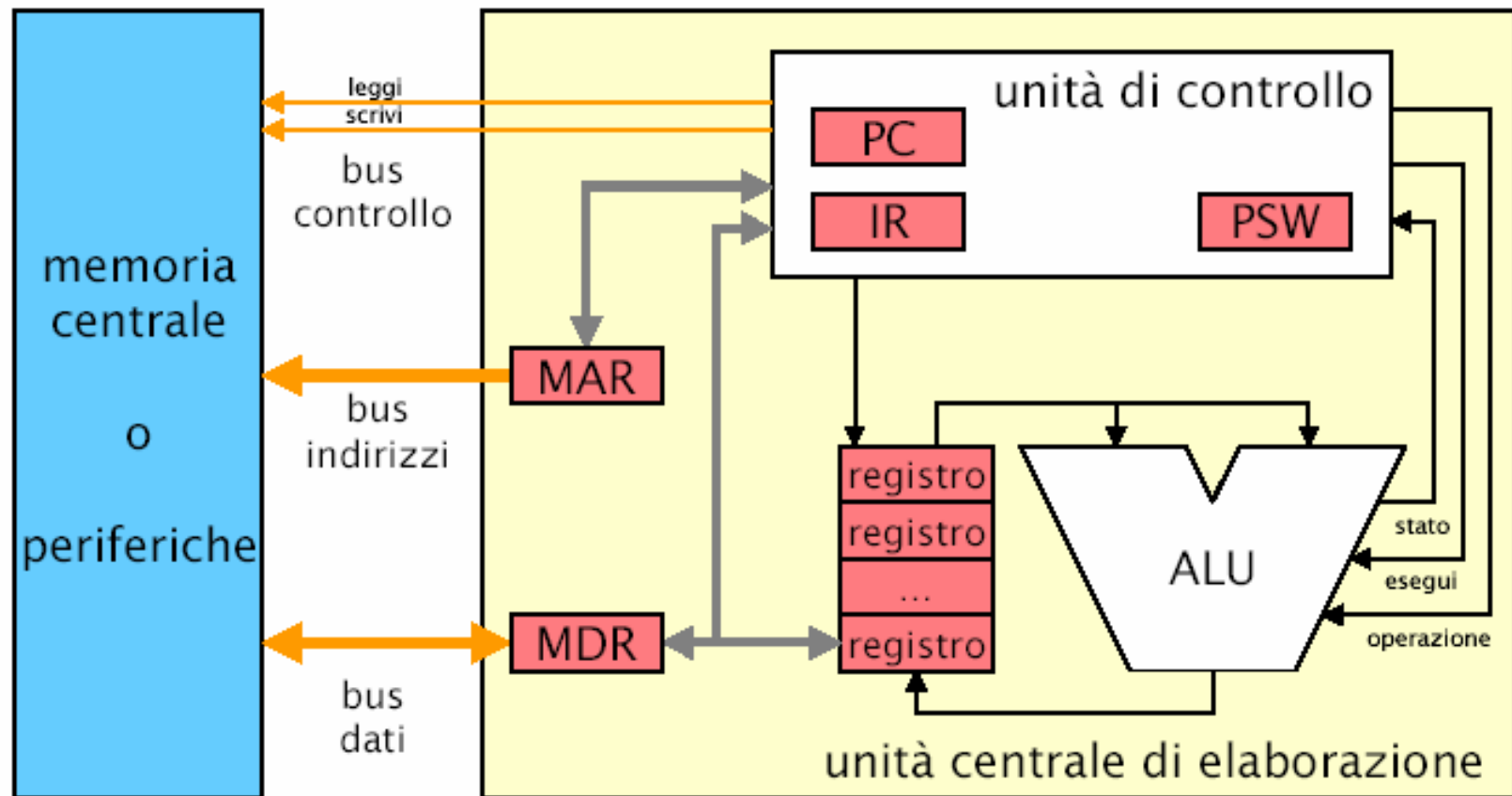
- ◆ Memoria centrale
- ◆ Memorie di massa
- ◆ Registri della CPU
- ◆ Memoria cache
  - Livello di memoria intermedio fra registri e memoria centrale
  - Memoria ad alta velocità per tenere una copia delle celle della memoria centrale più frequentemente accedute
  - richiede una politica di cache management e di gestione della consistenza

# Tipi di memoria in un calcolatore





# Struttura della CPU

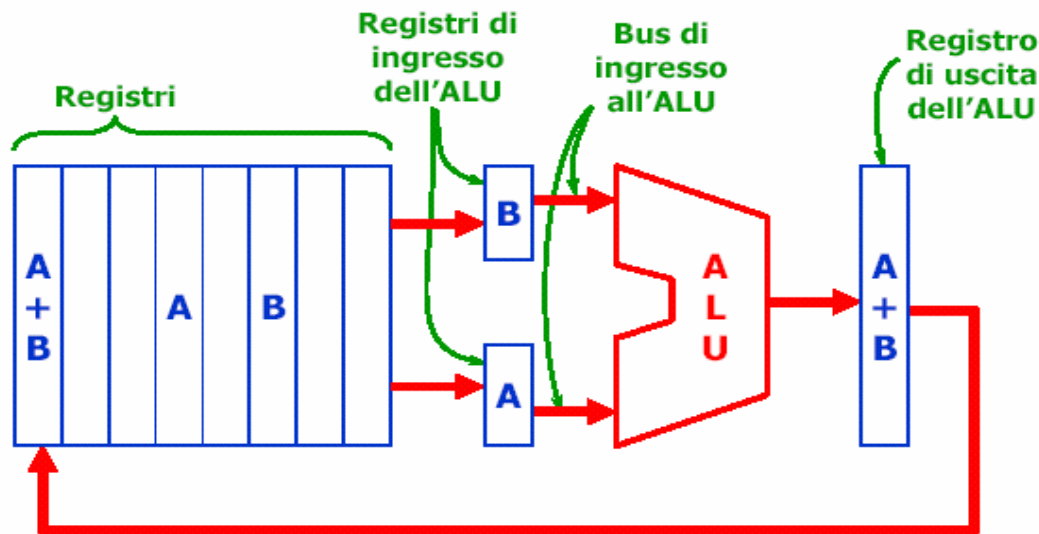


# Registri

- ◆ PC: contatore delle istruzioni (program counter)
  - contiene l'indirizzo della prossima istruzione da eseguire
- ◆ IR: registro delle istruzioni (instruction register)
  - contiene l'istruzione che deve essere eseguita
- ◆ PSW: parola di stato del processore
  - contiene informazioni, opportunamente codificate, sull'esito dell'ultima istruzione che è stata eseguita
- ◆ MAR: registro indirizzi della memoria
  - indirizzo della cella che deve essere acceduta
- ◆ MDR: registro dati della memoria
  - dato che è stato acceduto o che deve essere memorizzato
- ◆ registri generali
  - memorizza operandi e risultati delle operazioni

# Unità Logico-Aritmetica

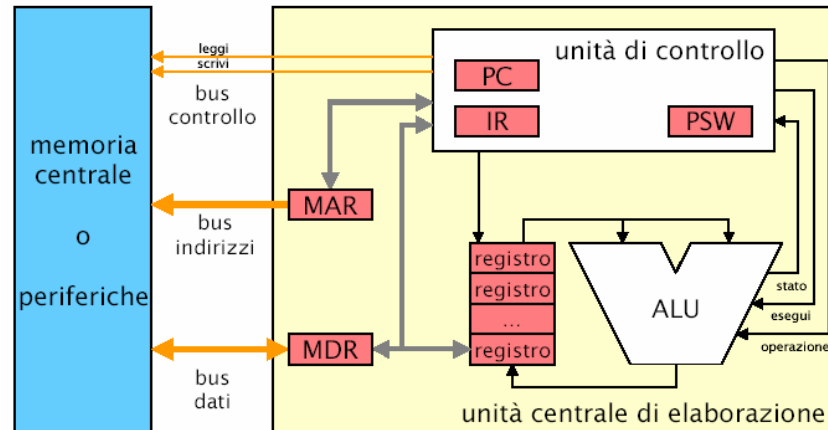
- ◆ L'**Unità Logico-Aritmetica (ALU)** è costituita da un insieme di circuiti in grado di svolgere le operazioni di tipo aritmetico e logico
- ◆ La ALU legge i valori presenti in alcuni registri, esegue le operazioni e memorizza il risultato in un altro registro



# Unità di controllo: ciclo di esecuzione delle istruzioni

- ◆ L'unità di controllo esegue un'istruzione mediante le tre seguenti operazioni di base:
  - **fetch**: preleva una nuova istruzione da eseguire dalla memoria centrale
  - **decode**: interpreta l'istruzione decodificandone la rappresentazione binaria
  - **execute**: esegue l'operazione richiesta
- ◆ Un programma è eseguito applicando ad ogni istruzione la sequenza fetch-decode-execute
  - ciclo di esecuzione dell'istruzione o ciclo macchina

# Ciclo fetch-decode-execute



## ◆ FETCH

- si accede alla prossima istruzione, riferita dal registro contatore dell'istruzione (PC)
- si copia l'istruzione nel Registro Istruzioni (IR)

## ◆ DECODE

- si decodifica l'istruzione, ed eventualmente si trasferiscono alcuni operandi (dati) in opportuni registri generali

## ◆ EXECUTE

- si incrementa il registro contatore dell'istruzione (PC)
- si comanda l'esecuzione delle operazioni necessarie alle altre componenti HW coinvolte (ALU, interfacce I/O, memoria, bus) 37

# Istruzioni del linguaggio macchina

## ◆ Istruzioni per l'elaborazione dei dati

- aritmetiche
- logiche (AND, OR, NOT)
- relazionali (maggiore, minore, uguale, ...)

## ◆ Istruzioni per il controllo del flusso di esecuzione

- Selezione (*If*)
- Ciclo (*While*, *For*)
- Si possono tutte ottenere con istruzioni di salto (*Jump*) e salto condizionato (*JumpZ*, *JumpGZ*)

## ◆ Istruzioni per il trasferimento di informazioni

- dati ed istruzioni fra CPU e memoria
- dati fra CPU e dispositivi di I/O (tramite interfacce)

# Esempio di esecuzione di un programma in linguaggio macchina

## Registri CPU

1000	Load	3568	R1
1001	Add	R1	R2
1002	Store	R1	3568
1003	Jump	1000	

3568	44
------	----

R1

R2

PC

IR

MAR

MDR

# Esecuzione istruzione **1000**: fetch

**1000**

<b>Load</b>	<b>3568</b>	<b>R1</b>
<b>Add</b>	<b>R1</b>	<b>R2</b>
<b>Store</b>	<b>R1</b>	<b>3568</b>
<b>Jump</b>	<b>1000</b>	

1001

1002

1003

3568

44

## Registri CPU

R1   
R2

PC

IR

MAR

MDR

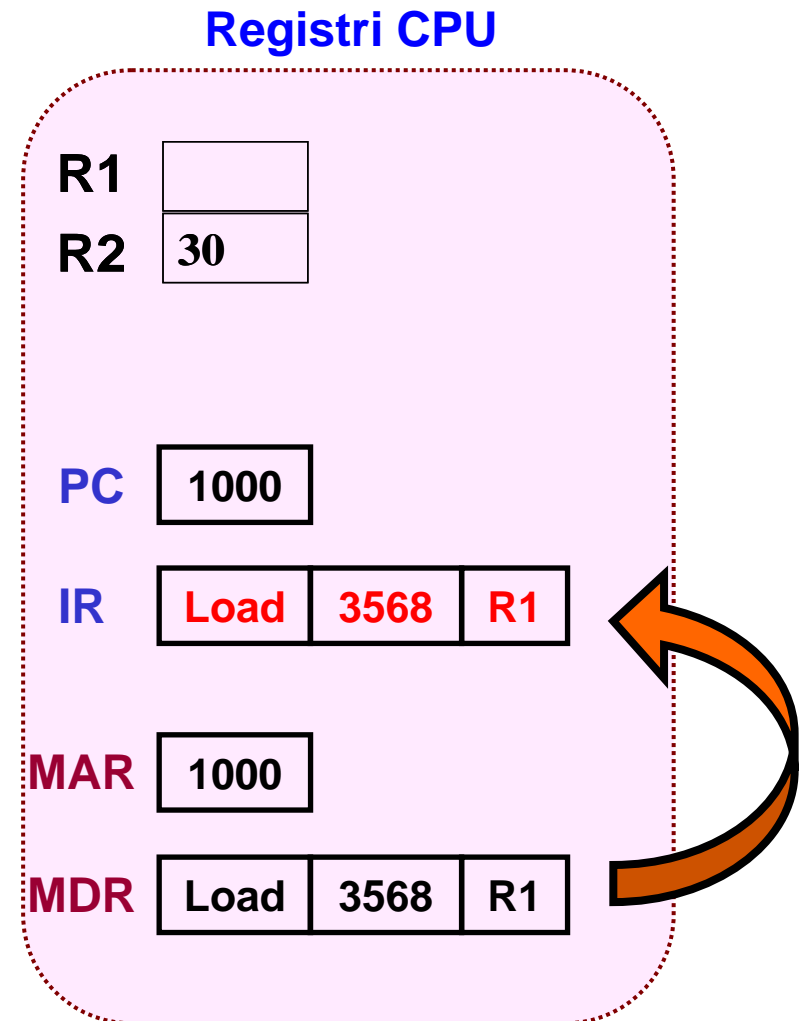


# Esecuzione istruzione *1000* (2): fetch

1000	Load	3568	R1
1001	Add	R1	R2
1002	Store	R1	3568
1003	Jump	1000	

3568	44
------	----



# Esecuzione istruzione **1000** (3): decode + execute

1000	<b>Load</b>	3568	R1
1001	<b>Add</b>	R1	R2
1002	<b>Store</b>	R1	3568
1003	<b>Jump</b>	1000	

3568	44
------	----

## Registri CPU

R1 **44**  
R2 30

PC **1001**

IR Load 3568 R1

MAR 1000

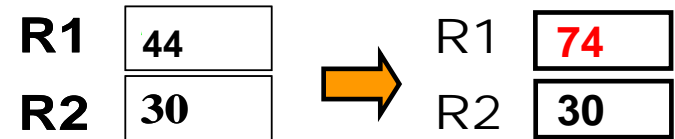
MDR Load 3568 R1

# Esecuzione istruzione **1001**

1000	<b>Load</b>	3568	R1
1001	<b>Add</b>	R1	R2
1002	<b>Store</b>	R1	3568
1003	<b>Jump</b>	1000	

3568	44
------	----



*NB.:*

E' stata attivata la ALU

$$R1 = 44 + 30 = 74$$

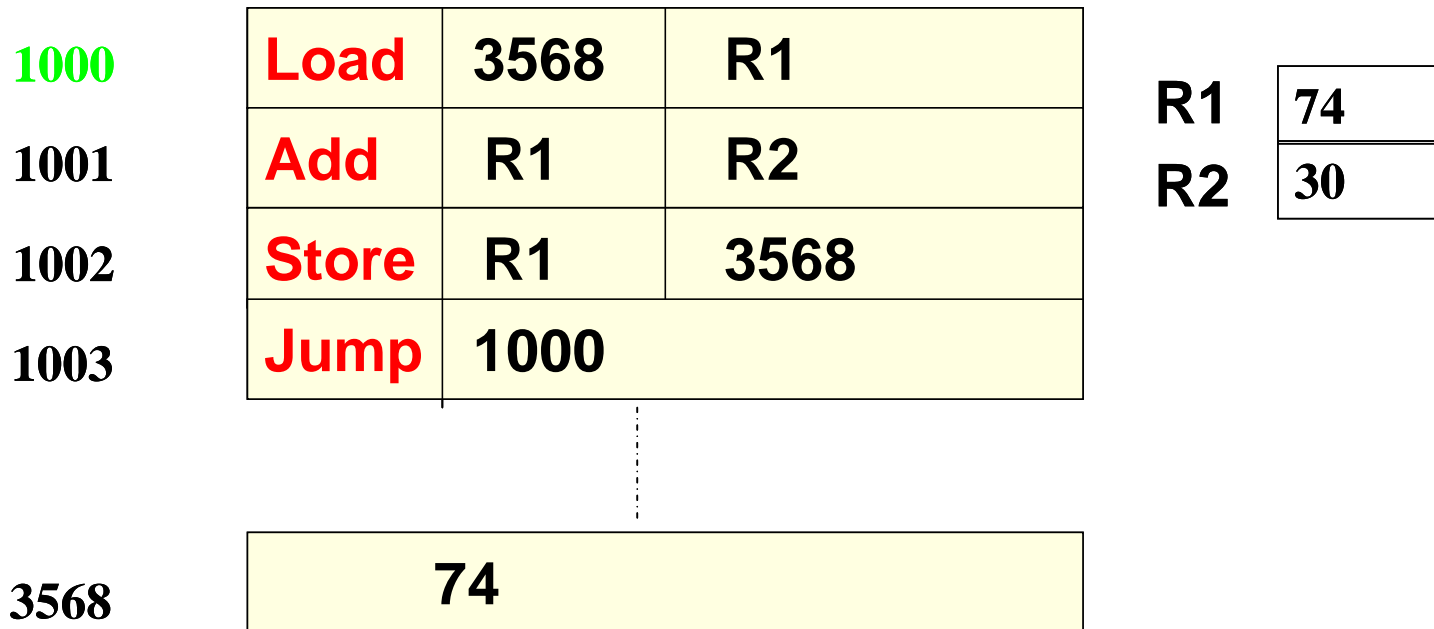
# Esecuzione istruzione *1002*

1000	Load	3568	R1	R1	74
1001	Add	R1	R2	R2	30
1002	Store	R1	3568		
1003	Jump	1000			

3568	74
------	----

# Esecuzione istruzione *1003*

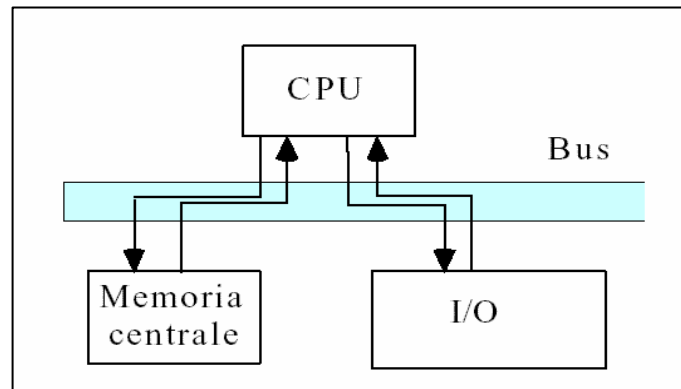


# Bus di sistema

---

# Bus di sistema

- ◆ Interconnette le componenti interne del calcolatore
  - CPU, memoria ed interfacce a periferiche (I/O, memoria di massa, ...)
- ◆ Collega due unità funzionali alla volta
  - una trasmette e l'altra riceve
- ◆ Funzionamento master/slave
  - solitamente la CPU (master) seleziona la connessione da attivare e ordina il trasferimento dei dati

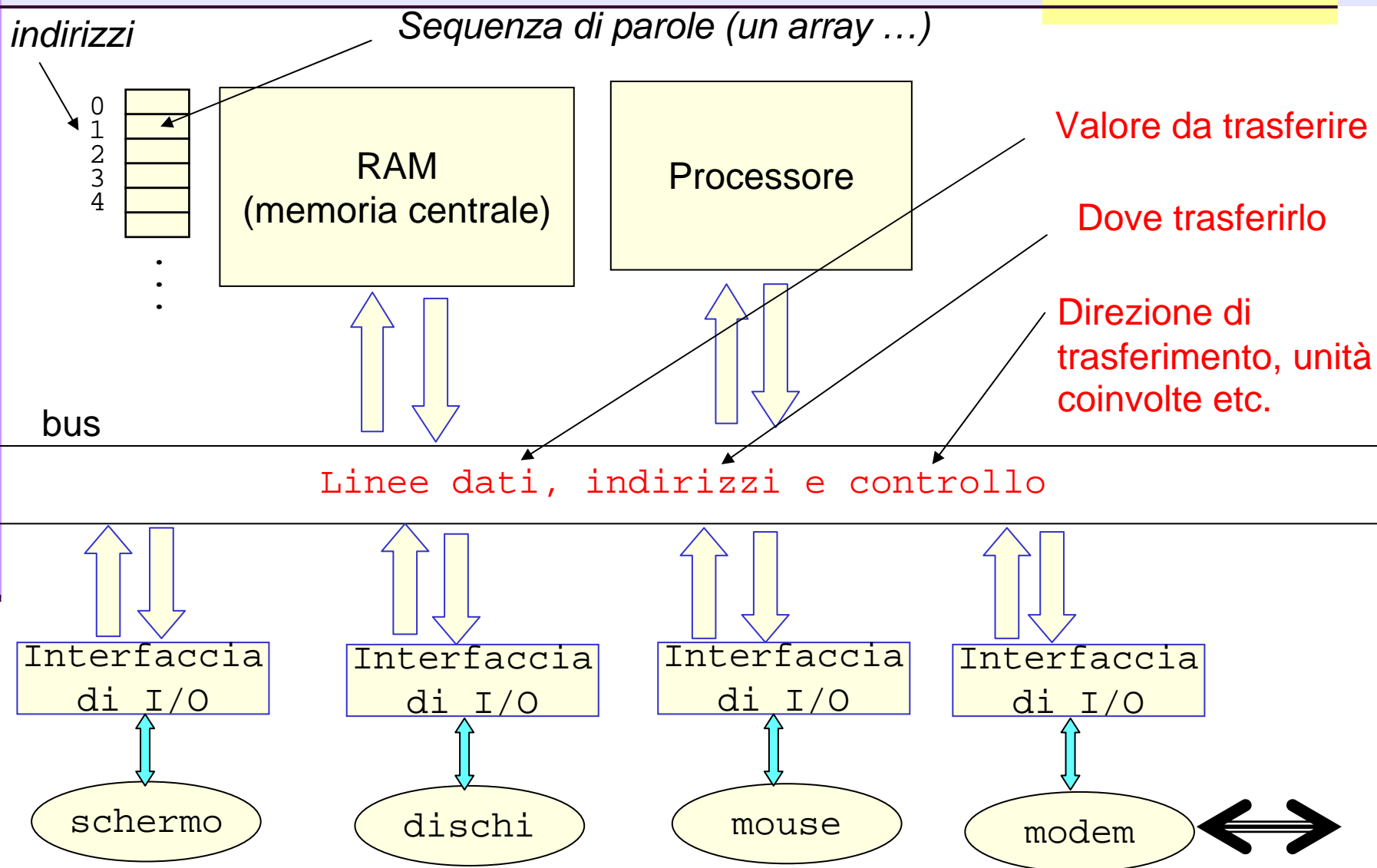


# Bus di sistema

- ◆ Il bus trasporta dati, indirizzi e comandi
- ◆ **Bus dati** (data bus)
  - Serve per trasferire dati tra
    - memoria centrale e registro dati (MDR) della CPU
    - periferiche e CPU (o memoria centrale)
  - Bidirezionale
- ◆ **Bus indirizzi** (address bus)
  - Serve per trasmettere il contenuto del registro indirizzi (MAR) alla memoria (o a una periferica)
  - si seleziona una cella per successive operazioni di lettura o scrittura
  - Unidirezionale
- ◆ **Bus comandi** (command bus)
  - Serve per inviare comandi
    - verso la memoria (es: lettura o scrittura)
    - o verso una periferica (es.: stampa)
  - Unidirezionale



# Bus di sistema



# Periferiche di I/O

---

# Periferiche di ingresso-uscita (I/O)

- ◆ Un calcolatore può essere collegato a vari dispositivi di ingresso e/o uscita (**periferiche**)
  - esempi: tastiera, mouse, schermo, stampanti, modem
  - anche le memorie di massa (es., unità disco e lettore di CD-ROM) sono considerate periferiche
- ◆ Nella macchina di Von Neumann, le periferiche non fanno parte del calcolatore
  - ogni periferica è controllata da un'**interfaccia (controller)**
  - una interfaccia ha il compito di tradurre i segnali interni del calcolatore in un formato comprensibile alla periferica stessa, e viceversa

# Interazione fra CPU e dispositivi

- ◆ I dispositivi (device) di I/O e la CPU possono lavorare in parallelo
- ◆ Ogni controller (interfaccia) ha un buffer locale
  - L'I/O avviene dal dispositivo al buffer locale del controller
- ◆ Sincronizzazione fra CPU e dispositivi di I/O:
  - La CPU guida lo spostamento dei dati tra la memoria centrale ed il buffer locale del controller
  - La CPU ordina l'avvio dell'operazione al controller
  - Il controller segnala che ha completato la sua operazione tramite un segnale (***interruzione***)
  - La CPU viene usata per eseguire una procedura di gestione dell'interruzione
- ◆ Segnali di controllo e dati viaggiano sul bus di sistema

# Direct Memory Access (DMA)

---

- ◆ Alternativa all'interazione mediante interrupt usata nel trasferimento di regioni contigue di memoria
- ◆ Un microprocessore dedicato (DMA controller) trasferisce direttamente i dati in memoria
- ◆ CPU e DMA controller si alternano nel controllo del bus