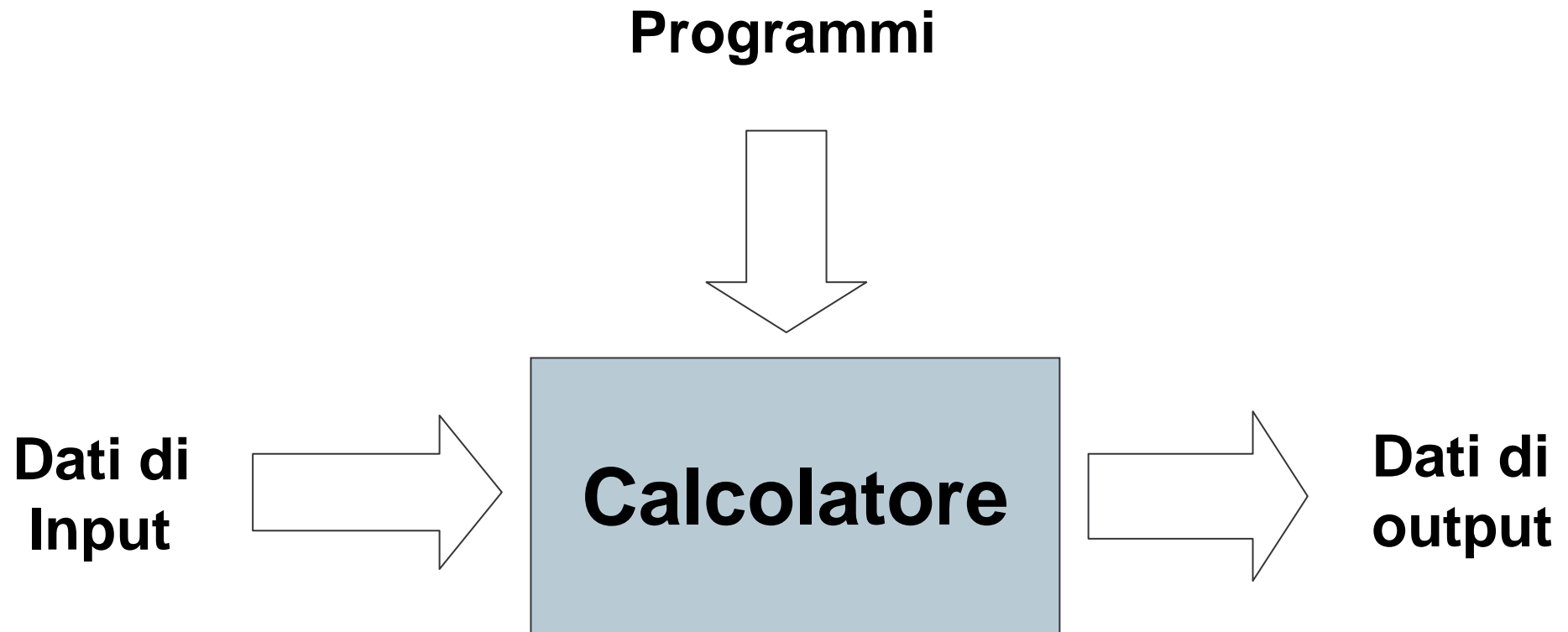


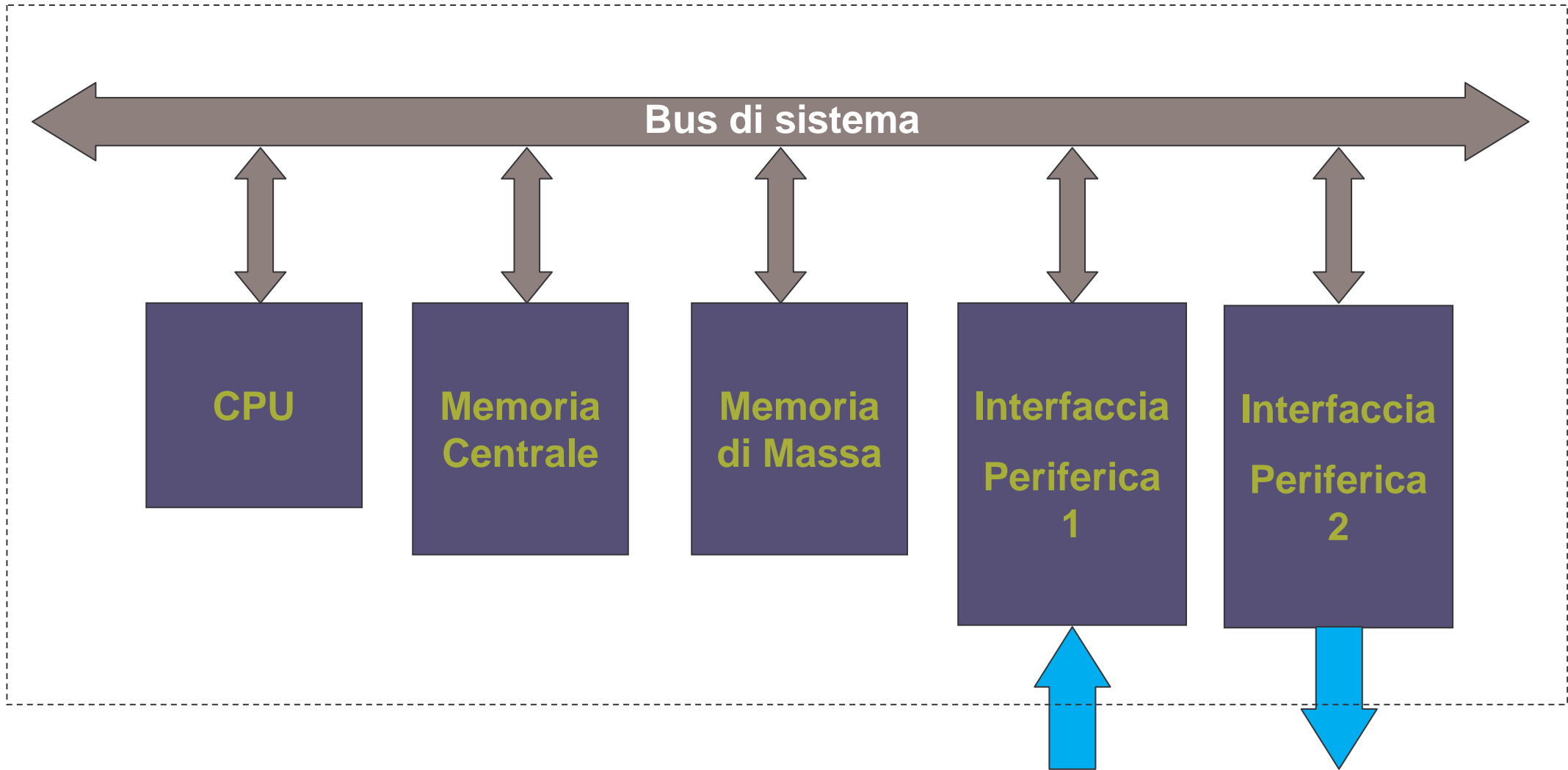
Sistemi Operativi

Modulo 2

Schema di un Sistema di Calcolo



Modello di von Neumann



Complessità dell'hardware

Il modello di Von Neumann è uno schema di principio. Attualmente in commercio esistono:

- diversi dispositivi di memorizzazione di massa, anche molto diversi tra loro;
- molti tipi di periferiche diverse.

Per gestire diversi dispositivi è necessaria una conoscenza approfondita del loro funzionamento in quanto:

- è troppo complesso da gestire;
- offre dei servizi di livello estremamente basso;
- richiede conoscenze estremamente specialistiche;
- l'architettura hardware può variare molto da computer a computer.

Complessità del software

Un qualunque programma dovrebbe:

- Implementare l'algoritmo per cui è stato scritto;
- essere in grado di gestire tutte le periferiche per la memorizzazione e l'I/O.

Problema della **RIDONDANZA**:

- Se ogni programma contenesse le istruzioni per la gestione delle periferiche ci sarebbe una enorme ridondanza di software, del tutto inutile;

Complessità del software

Il software viene diviso generalmente in due categorie:

- **Software di sistema:** è costituito da programmi per la gestione delle varie componenti di un sistema di calcolo (hard disk, CD-ROM, monitor, stampanti, ecc);
- **Software applicativi:** sono programmi che devono eseguire compiti molto specifici: browser, posta elettronica, programmi di simulazione, riproduzione di musica, elaborazione di immagini, ecc.

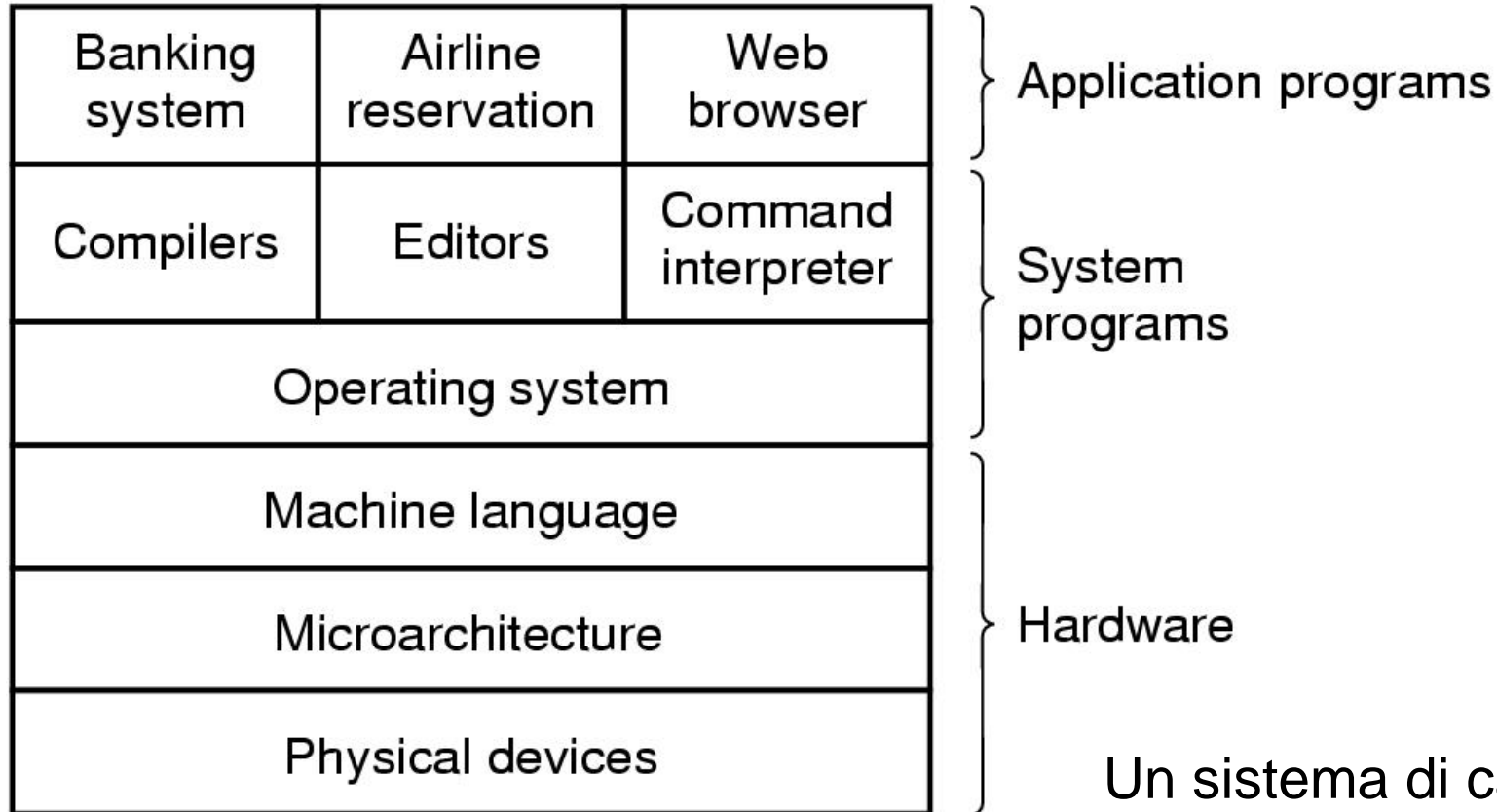
Complessità e Stratificazione

- In linea di principio, un sistema di calcolo può essere schematizzato in maniera molto semplice:

SISTEMA DI CALCOLO = HARDWARE + SOFTWARE

- In pratica un tale sistema si presenta molto complesso, pertanto nasce l'esigenza di affrontare e gestire tale complessità;
- La soluzione scelta per la gestione di un tale livello di complessità è stata quella di dividere in strati sia il software che l'hardware.
- La stratificazione consente di trattare e gestire in maniera separata ogni strato rispetto agli altri. Quello che bisogna definire è solo l'interazione fra i vari strati.

Il Sistema Operativo (Operating System)



Un sistema di calcolo è costituito da:

- Hardware
- Programmi di sistema
- Programmi applicativi

Il Sistema Operativo

Il Sistema Operativo è uno strato software che:

- opera direttamente sull'hardware;
- isola dai dettagli dell'architettura hardware;
- fornisce un insieme di funzionalità di alto livello;

Obiettivi

- Convenienza

Rende l'utilizzo del computer più semplice ed intuitivo

- Efficienza

Permette di impiegare le risorse del sistema in maniera più efficiente

- Capacità di evoluzione

Permette l'introduzione di nuove funzionalità e/o il miglioramento delle risorse hardware senza interferire con il servizio svolto

Definizione di Sistema Operativo

- **E' una macchina estesa**
 - Cela la complessità della macchina
 - Mostra all'utente una **Virtual Machine** più facile da usare

- **E' un gestore di risorse**
 - Gestisce le risorse hardware disponibili
 - Distribuisce ai programmi tempo di CPU

Il S.O. come macchina estesa

- La gestione delle periferiche richiede un'approfondita conoscenza del dispositivo che si vuole utilizzare.
- Lo sviluppo di programmi applicativi non deve occuparsi dei dettagli relativi alla gestione delle periferiche: ciò richiederebbe una notevole aggiunta di complessità al programma da sviluppare.
- E' il S.O. che nasconde agli applicativi la complessità dell'hardware delle periferiche, fornendo una semplice interfaccia: ad es. il concetto di **FILE**.
- Il S.O. può essere visto come una **macchina** su cui far girare gli applicativi.

Il S.O. come gestore delle risorse

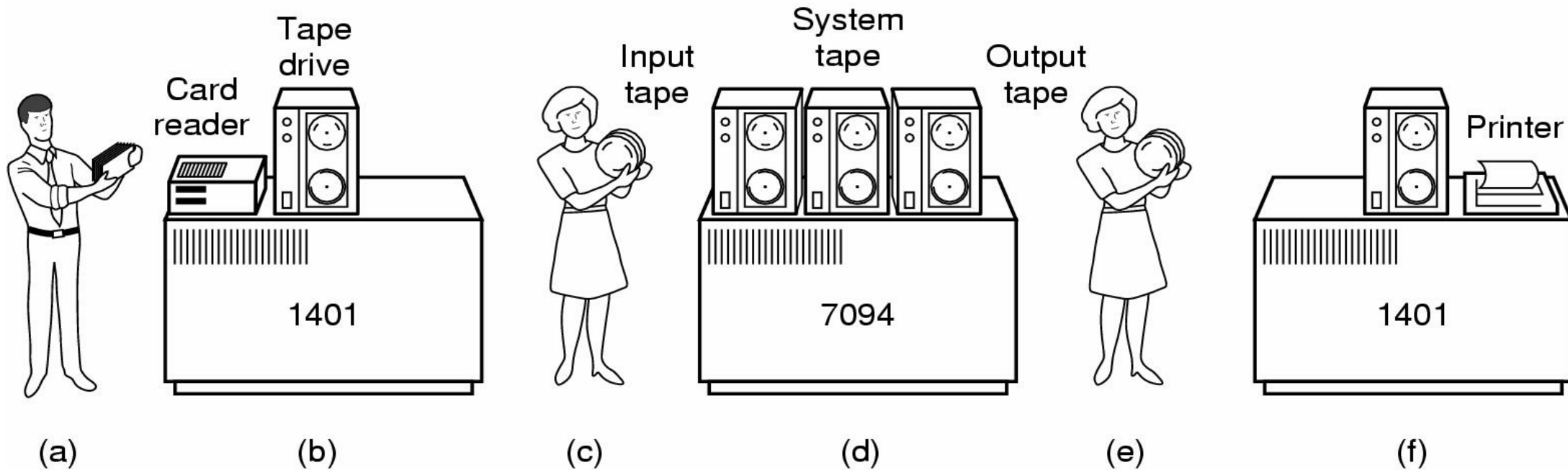
Il S.O. può essere visto anche come quella parte del sistema di calcolo che si occupa della gestione delle risorse disponibili:

- Memoria
- Tempo di calcolo
- Periferiche (dischi, cd-rom, ecc)

Breve storia dei S.O.

- First generation 1945 - 1955
 - vacuum tubes, plug boards
- Second generation 1955 - 1965
 - transistors, batch systems
- Third generation 1965 – 1980
 - Multiprogramming and time sharing
- Fourth generation 1980 – present
 - personal computers

Breve storia dei S.O.: la 2^a generazione



batch system

- bring cards to 1401
- read cards to tape
- put tape on 7094 which does computing
- put tape on 1401 which prints output

Il 7094 (IBM) aveva un solo tipo di periferica: il nastro magnetico.

Breve storia dei S.O.: la 3^a generazione

I S.O. di terza generazione hanno rappresentato un grosso passo avanti rispetto a quelli della generazione precedente:

- Gestione di più periferiche (nastri, monitor, stampanti, tastiera);
- Esecuzione di più programmi (multiprogramming);
- Gestione di più utenti (time sharing);

Il Multiprogramming

Gestire l'esecuzione di più programmi non è un'operazione banale:

- Bisogna memorizzare tutti i dati relativi ai programmi che si vuole gestire;
- Bisogna gestire la memoria in maniera tale che ogni programma possa accedere solo alla zona di memoria che gli è stata assegnata.

Il Time Sharing

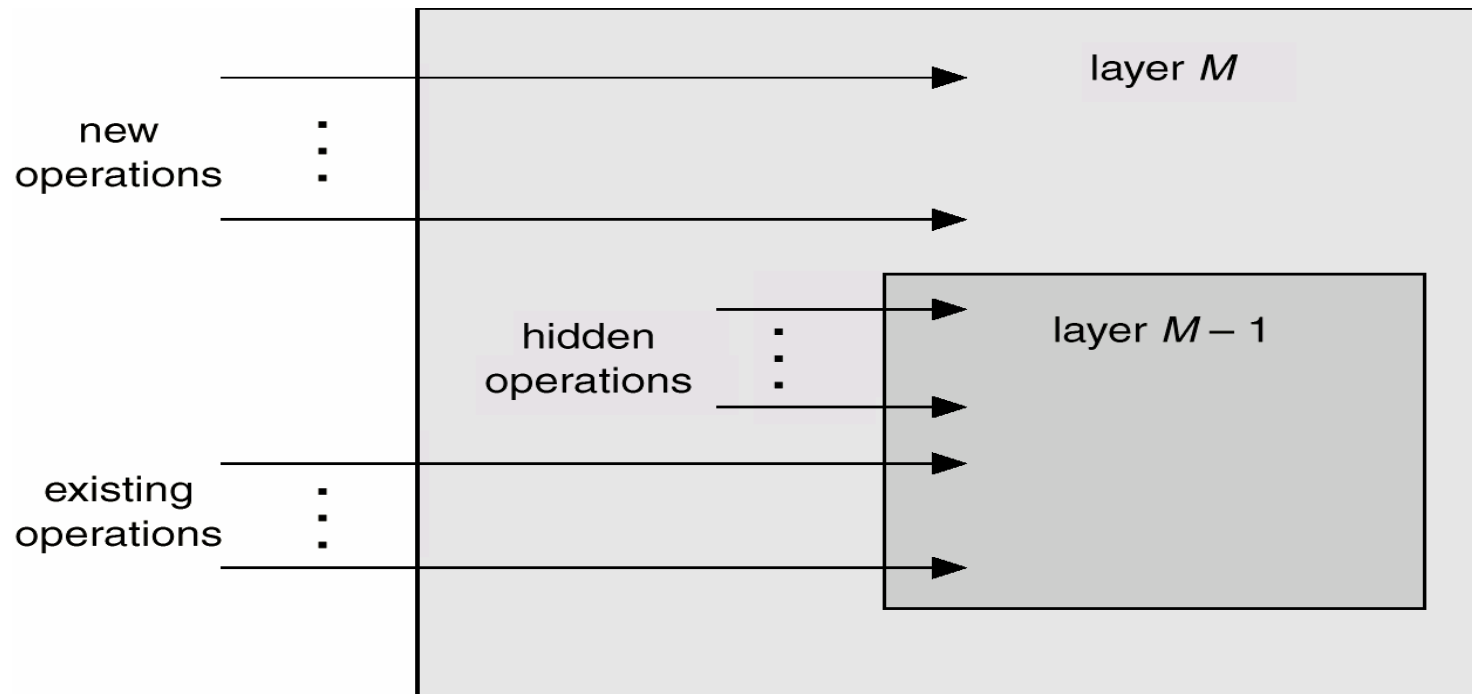
- Il time sharing consente a più utenti di usare la stessa risorsa di calcolo (CPU) contemporaneamente e indipendentemente l'uno dall'altro.
- Il time sharing è stato reso possibile dal fatto che la velocità di calcolo della CPU è stata fin dagli inizi molto maggiore dei tempi di reazione dell'uomo.
- La problematica principale del time sharing è data dalla suddivisione del tempo di calcolo tra i vari utenti.

La struttura del Sistema Operativo

A causa della loro complessità, i Sistemi Operativi sono di solito strutturati come una serie di livelli (architettura a *buccia di cipolla*).

Ogni livello realizza un certo sottoinsieme di funzioni attraverso una *macchina virtuale* che:

- nasconde i meccanismi implementativi
- offre un insieme ben definito di funzionalità ai livelli superiori.



La struttura del Sistema Operativo

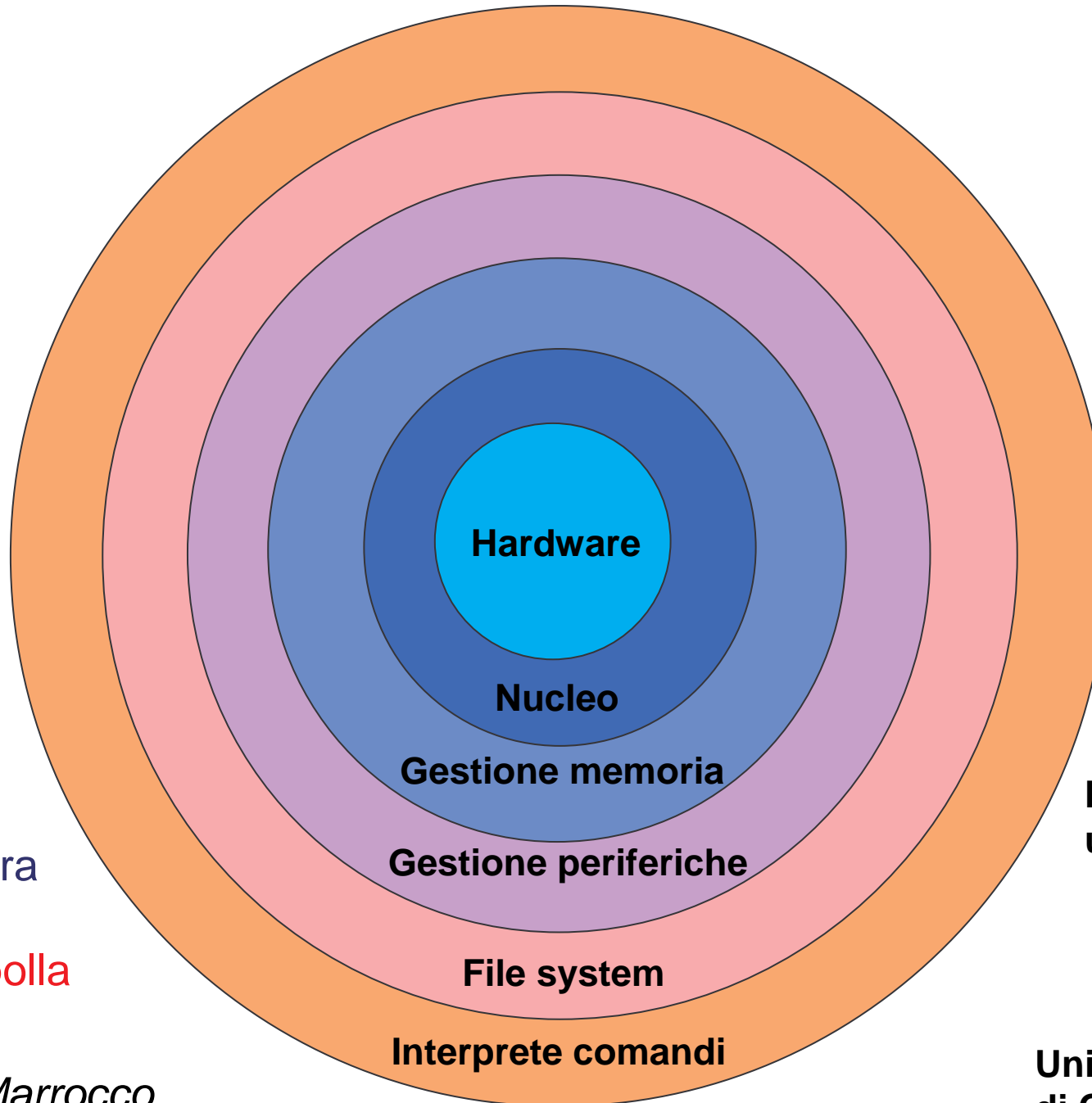
- Ogni livello del sistema operativo interagisce con gli altri livelli attraverso un'*interfaccia*.
- Un'interfaccia di una qualunque entità può essere definita in maniera del tutto generale come quella parte dell'entità che la mette in comunicazione con l'esterno.
- In un SO l'interfaccia di un livello è rappresentato dall'insieme di comandi che una determinata entità è in grado di eseguire.
- I diversi livelli comunicano tra loro attraverso le rispettive interfacce.

La struttura del Sistema Operativo

La suddivisione di un S.O. in livelli che interagiscono tra loro attraverso delle interfacce ben definite ha delle implicazioni molto importanti:

- Per accedere ai servizi forniti da una parte del S.O. non devo conoscere come è fatta o i suoi dettagli di funzionamento; basta conoscerne l'interfaccia.
- Se modifico, un livello senza modificarne l'interfaccia, le modifiche non sono visibili dall'esterno e quindi non devo modificare quelle parti del sistema che interagiscono con quel livello.

La struttura del Sistema Operativo



Questa struttura
è detta:
a buccia di cipolla

C. Marrocco

**Università degli Studi
di Cassino**

Le chiamate di sistema

- L'interfaccia fra S.O. e i programmi utenti avviene attraverso delle **chiamate di sistema**;
- Le chiamate di sistema rappresentano un set di istruzioni fornite dal sistema operativo che estende il set di istruzioni fornite dal processore (linguaggio macchina);
- L'insieme delle chiamate di sistema è l'interfaccia del sistema operativo verso l'esterno;
- Il S.O. fornisce tutti i suoi servizi solo attraverso le chiamate di sistema.

Il linguaggio macchina esteso

Il linguaggio macchina di un eseguibile contiene due tipi di istruzioni (binarie):

- Istruzioni nel linguaggio macchina della CPU, che sono eseguite direttamente dalla CPU;
- Istruzioni che rappresentano chiamate di sistema per il S.O.
 - Queste istruzioni vengono eseguite richiamando quella parte del S.O. chiamato interprete dei comandi, che poi comunica con la parte del S.O. interessata dallo specifico servizio richiesto da quella istruzione.

Esempi di Sistemi Operativi

UNIX: sistema operativo creato da Ken Thompson e Dennis Ritchie nel 1969. Inizialmente progettato per permettere a diversi programmatori di sfruttare simultaneamente le risorse di un minicomputer, ben presto si sviluppò in mainframe dagli usi più disparati, nelle università come nelle aziende, diventando uno dei migliori sistemi operativi di sempre. Attualmente la maggior parte dei server che collegano la rete Internet hanno UNIX come S.O

Linux: sistema operativo derivato da UNIX, che ha preso il nome dal suo creatore Linus Torvalds. Nel 1994 viene rilasciata la versione 1.0 di Linux. Questo sistema è trasferibile sul proprio computer tramite ftp da Internet oppure è disponibile nei CD-ROM allegati a molte riviste di informatica. Le più note distribuzioni Linux sono: Red-Hat, Debian, Ubuntu, Suse, Mandrake.

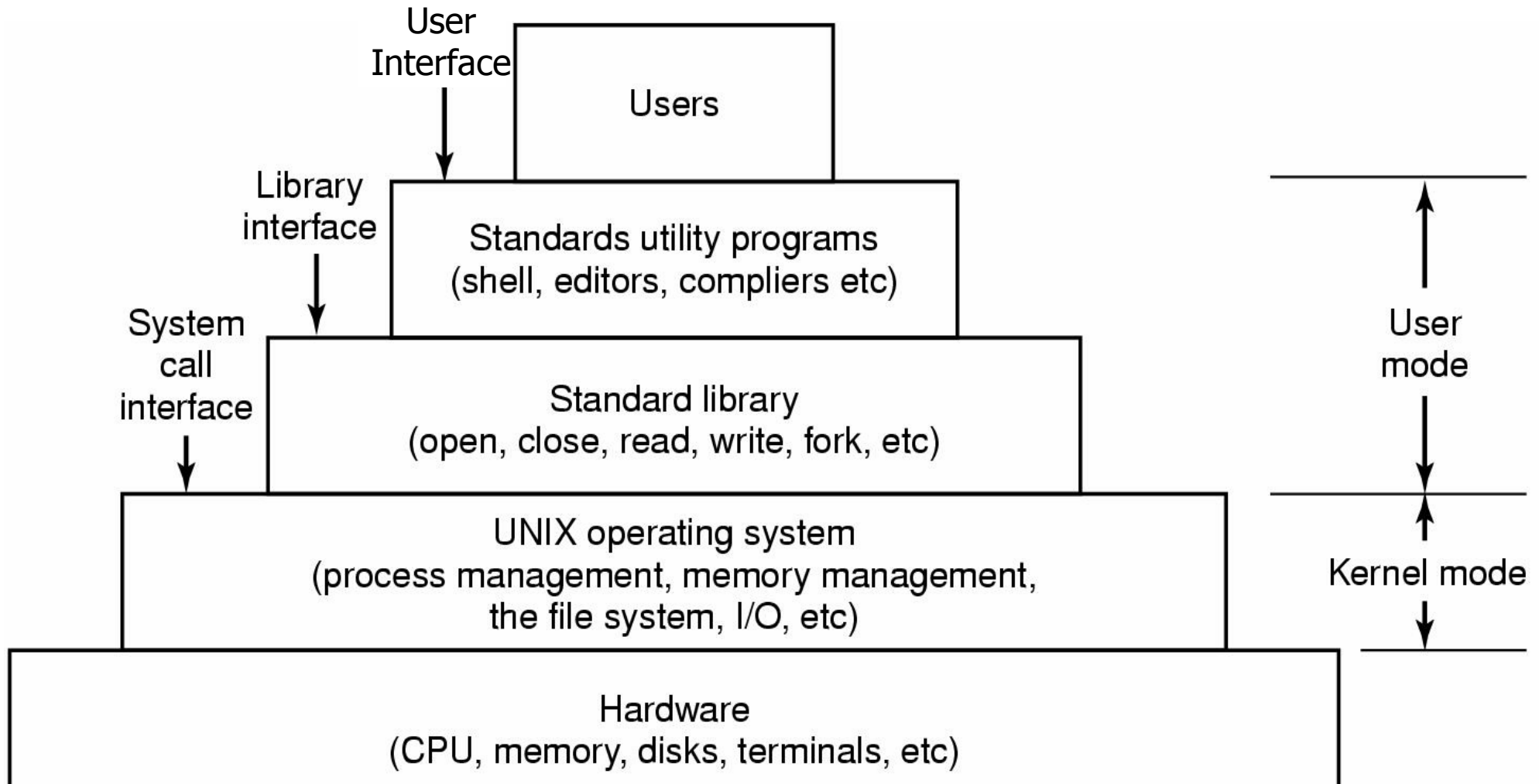
Esempi di Sistemi Operativi

MS-DOS: nel 1980 Tim Paterson e Bill Gates crearono il loro sistema operativo, l'MS-DOS (*Microsoft Disk Operating System*) inizialmente per i microprocessori della Intel a 8 e 16 bit. Successivamente è divenuto il sistema operativo standard dei computer IBM e compatibili.

Windows: termine che identifica la famiglia del ben noto S.O. di Microsoft. Inizialmente la prima versione era solo un'interfaccia grafica di MS-DOS, poi con la serie 9x è diventato il S.O. più diffuso nel mondo.

Macintosh (MAC): tratto dal nome di una qualità di mele californiane, il Macintosh di Apple è un computer nato dall'esperienza di Steve Jobs, Steve Wozniak e dai suoi collaboratori all'interno della Xerox. Jobs inizialmente lavorò su un computer prototipo denominato LISA. Nel gennaio 1984 lanciò il Mac, la prima piattaforma grafica di Apple, con riscontri significativi nel business dei PC.

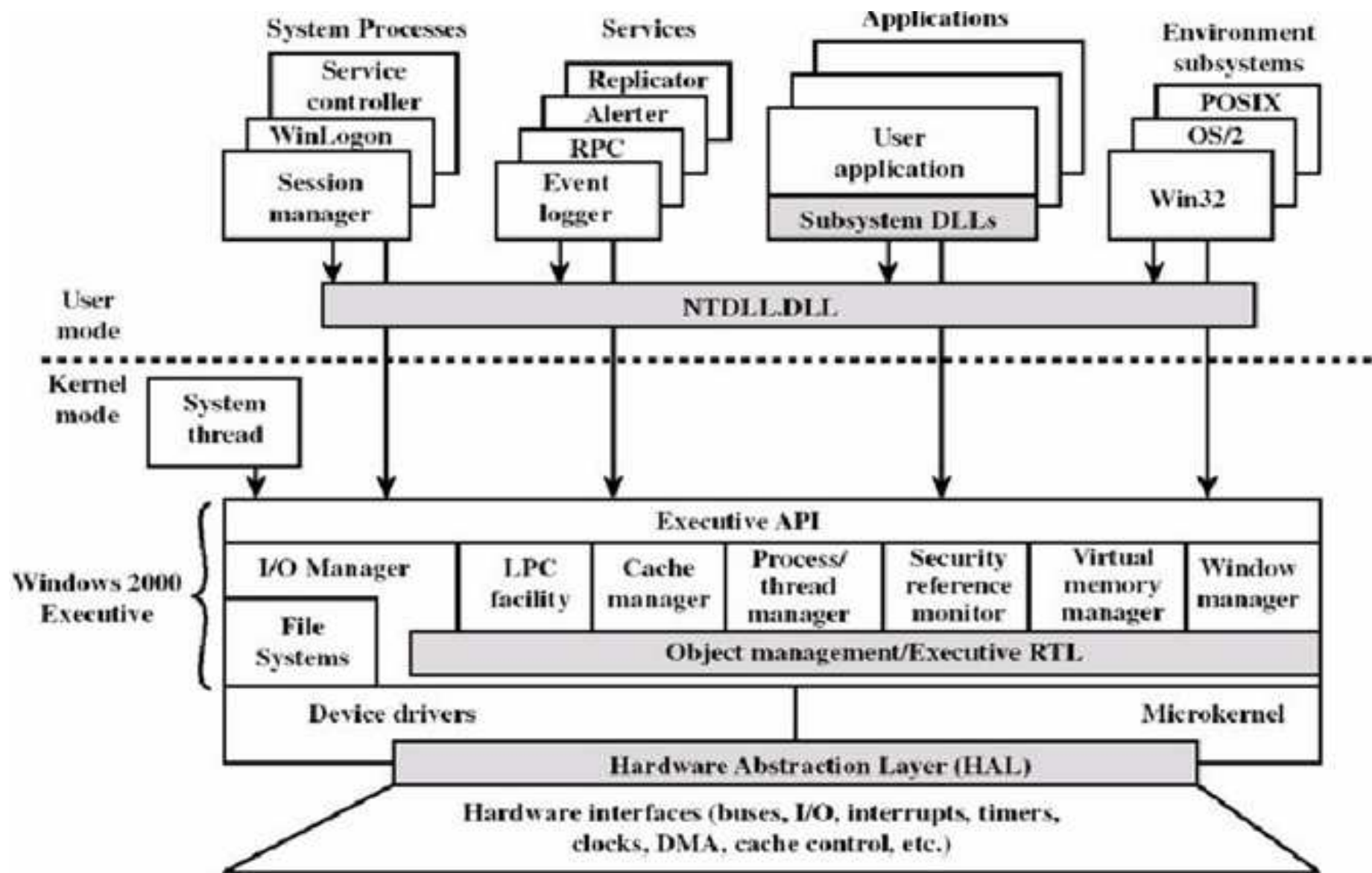
La Famiglia Unix



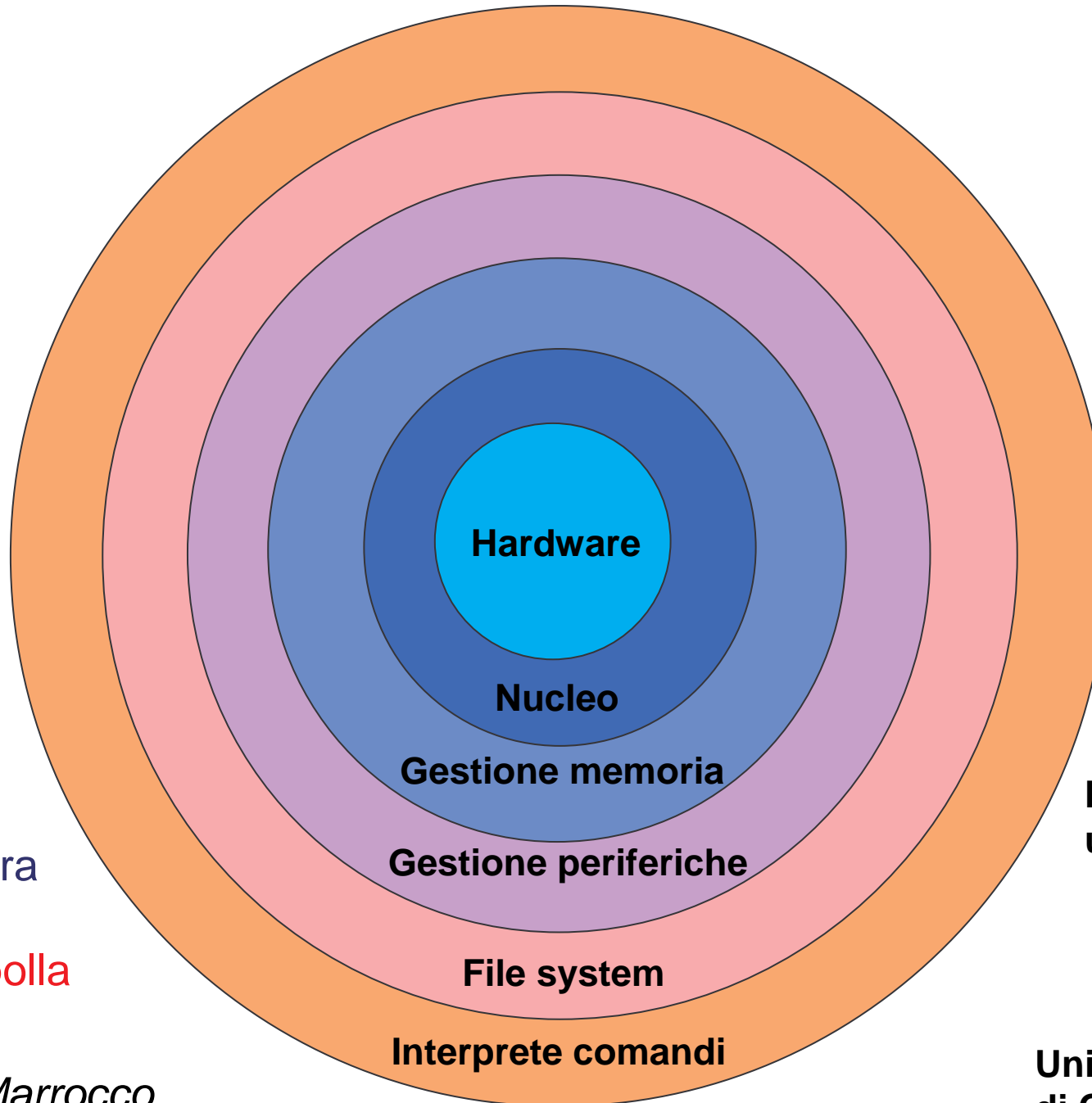
La struttura della famiglia Unix

System calls					Interrupts and traps			
Terminal handing		Sockets		File naming	Map- ping	Page faults	Signal handling	Process creation and termination
Raw tty	Cooked tty	Network protocols		File systems	Virtual memory			
	Line disciplines	Routing		Buffer cache	Page cache	Process scheduling		
Character devices		Network device drivers		Disk device drivers		Process dispatching		
Hardware								

La struttura di Windows NT/2000



La struttura del Sistema Operativo



Questa struttura
è detta:
a buccia di cipolla

C. Marrocco

**Università degli Studi
di Cassino**

Le funzioni principali di un S.O.

- Gestione dei processi.
- Gestione della memoria.
- Gestione delle periferiche.
- File system.
- Gestione utenti.

La Gestione dei Processi

Che cos'è un processo?

Un processo è un **programma in esecuzione** caratterizzato da una coppia di elementi:

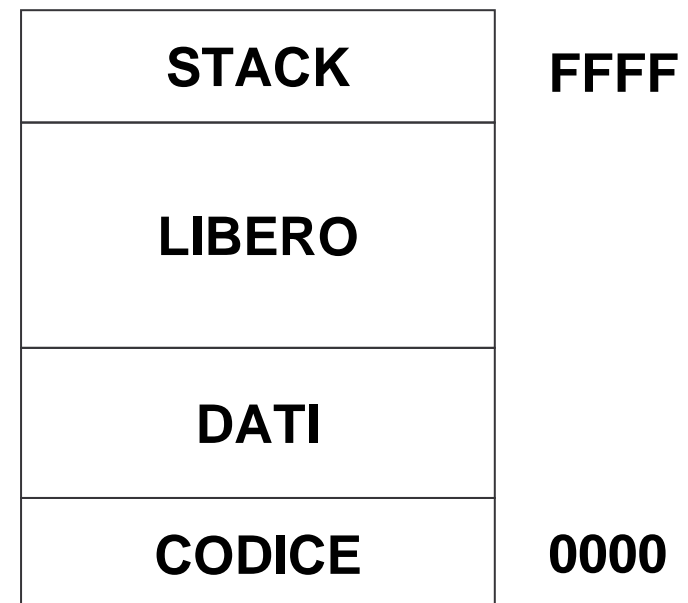
- Il **codice eseguibile** ovvero un adeguato spazio di **indirizzamento**.
- Lo **stato del processo** ovvero l'insieme dei valori contenuti nella memoria centrale, nell'insieme di registri del processore (tra cui program counter e instruction register), i thread, i descrittori dei file.

Lo spazio di indirizzamento

Lo spazio di indirizzamento è una lista di locazioni di memoria comprese tra due valori (un minimo ed un massimo) che vengono usate per la lettura e la scrittura dei dati.

Contiene:

- il programma eseguibile;
- i dati del programma;
- il suo stack.



I thread

- Il concetto di processo è associato a quello di **thread** con cui si intende l'unità granulare in cui un processo può essere suddiviso.
- Un processo ha sempre almeno un thread (sè stesso), ma in alcuni casi un processo può avere più thread che vengono eseguiti in parallelo.
- Mentre i processi sono fra loro indipendenti (utilizzano diverse aree di memoria ed interagiscono solo mediante appositi meccanismi di comunicazione) i thread condividono le medesime **informazioni di stato**, la memoria ed altre **risorse di sistema**.
- Mentre la creazione di un nuovo processo è sempre onerosa per il sistema in quanto devono essere allocate le risorse necessarie alla sua esecuzione, il thread è parte del processo, e quindi una sua nuova attivazione viene effettuata in tempi ridottissimi.

La gestione dei processi

- Molti programmi prevedono un elevato scambio di informazioni con le periferiche di I/O.
- Quando un programma chiede un'operazione di I/O la CPU è ferma in attesa del completamento dell'operazione.
- In alcuni casi, il tempo di attesa di I/O può arrivare anche al 80% o al 90% del tempo totale di esecuzione di un programma.
- Si ha un notevole spreco della risorsa CPU (**tempo macchina**).

La gestione dei processi

- Il problema della gestione dei processi è stato affrontato fin dalla 3^a generazione di SO (1965-1980).
- Si parla di **multiprogrammazione**.
- La multiprogrammazione implica la gestione delle principali risorse disponibili:
 - Il tempo macchina.
 - La memoria.
 - La gestione delle periferiche.

La gestione dei processi

- Il problema della gestione dei processi si presenta ai giorni nostri in maniera diversa.
- Lo stesso sistema operativo è un insieme di programmi che devono essere eseguiti allo stesso tempo.
- Ma generalmente si ha a disposizione un'unica CPU che lavora in maniera sequenziale (modello di Von Neumann): come è possibile fare in modo che questi processi vengano eseguiti insieme?
- L'unico modo è quello di simulare il **parallelismo**.

Il tempo macchina

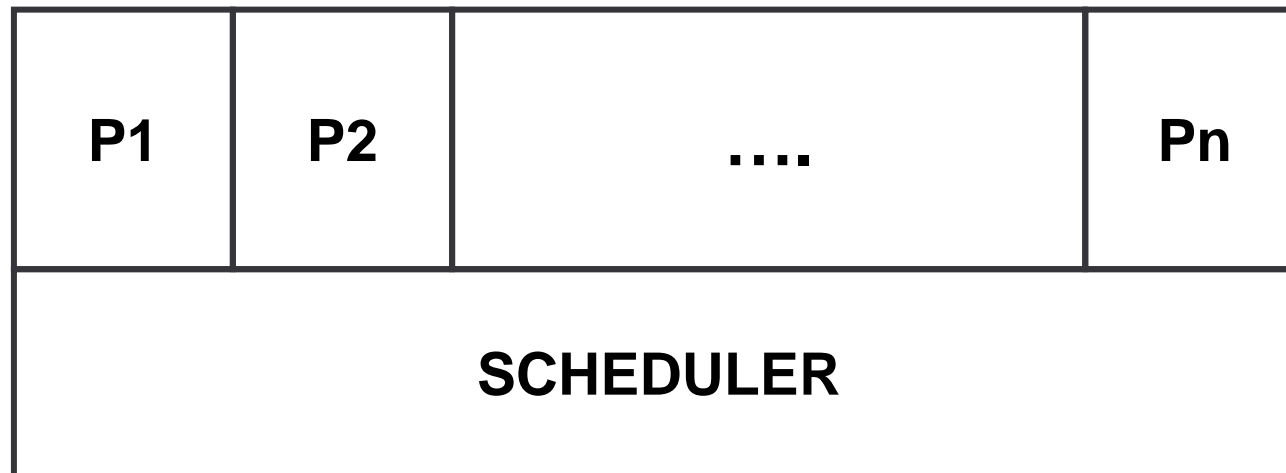
- I processi devono sincronizzarsi ovvero coordinare le loro attività.
- Il tempo macchina della CPU viene quantizzato.
- In pratica, la risorsa tempo viene suddivisa in parti uguali chiamate: ticks (istanti).
- Ogni processo viene eseguito per un certo numero di ticks e poi messo in stato di attesa.

Lo Scheduling

La simulazione di questo parallelismo viene chiamata **scheduling**

La parte del SO che si occupa della scheduling è lo **scheduler**

Lo scheduler fa parte del kernel del S.O.



Lo Scheduling

- Il processo in esecuzione non deve accorgersi in nessun modo della sua interruzione.
- Lo scheduling dei processi può essere effettuato solo salvando tutti i dati relativi al processo: il contenuto di tutti i suoi registri nel momento in cui viene sospeso.
- Il contenuto dei registri di un processo rappresenta una “istantanea” del processo in esecuzione al momento della sua interruzione.
- Quest’istantanea deve essere ripristinata all’atto della ripresa di quello specifico processo.

Lo Scheduling

- Lo scheduler gestisce una particolare struttura dati chiamata **tabella dei processi**.
- Nella tabella dei processi sono memorizzati tutti i dati relativi ai processi in esecuzione.
- La transizione tra i processi è possibile tramite un meccanismo hardware chiamato **interruzione (interrupt)**.
- Il meccanismo degli interrupt è per l'appunto quello che consente di interrompere un processo in esecuzione per l'avvio di un altro processo.

Le interruzioni

Quando arriva un'interruzione la CPU salva in maniera automatica le parti essenziali del programma in esecuzione e fa partire lo scheduler che poi porta a termine il lavoro.

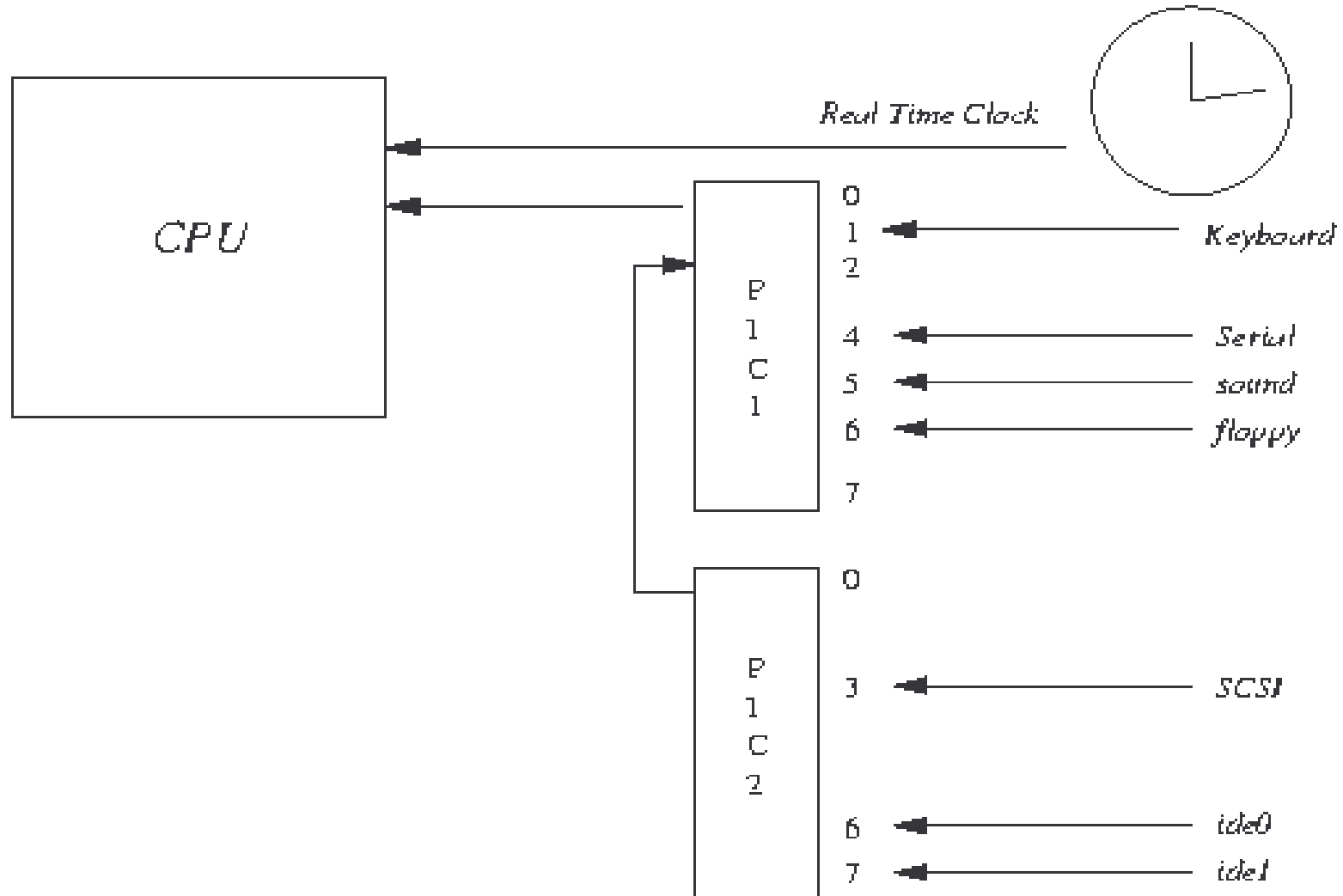
La sospensione deve avvenire in modo che il processo possa **ripristinare il contesto** in cui si era interrotto.

Interruzioni interne: viene generata internamente al processore in maniera sincrona (ad es. se un processo richiede l'esecuzione di un'operazione di I/O).

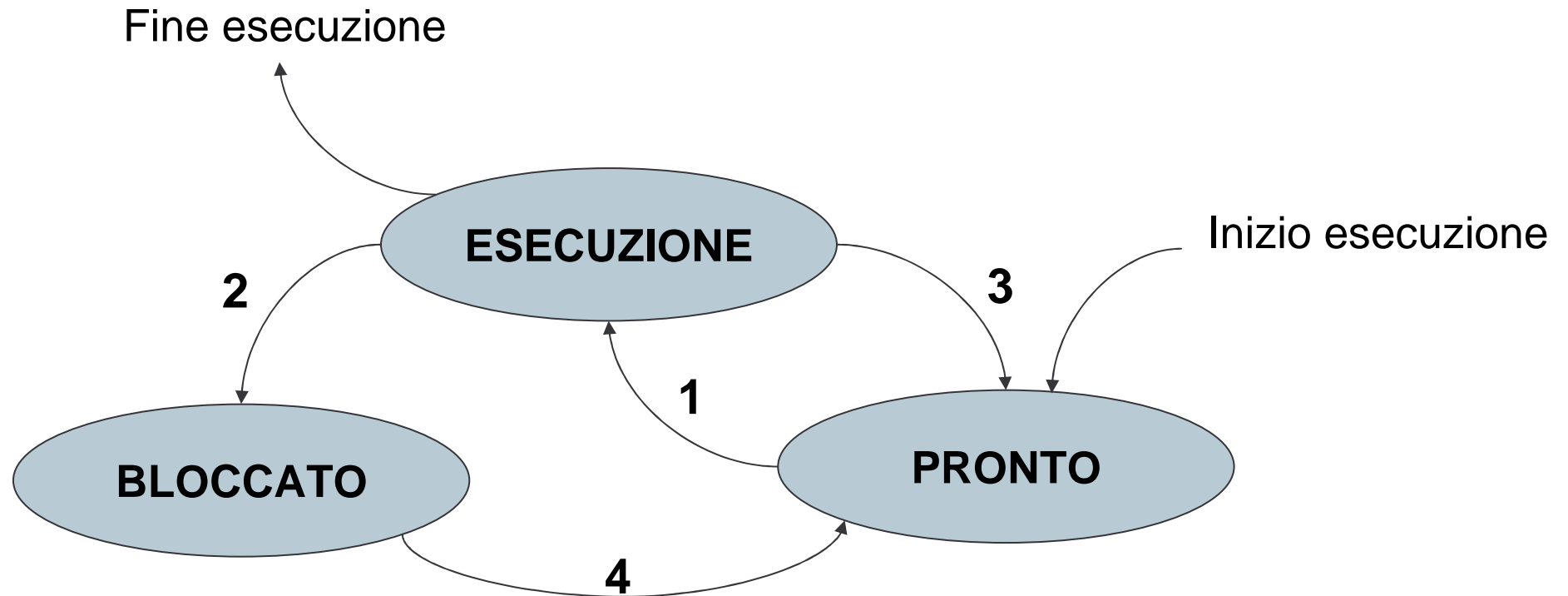
Interruzioni esterne: è un evento asincrono in cui una periferica esterna comunica al processore che ha terminato l'operazione richiesta (ad es. termina la lettura di dati dalla memoria centrale e il processo che era stato interrotto può riprendere la sua esecuzione).

Le interruzioni

Il meccanismo delle interruzioni è implementato in hardware.



Gli stati di un processo



1. Il processo viene scelto.
2. Blocco in attesa di I/O (Interruzione interna).
3. Viene scelto un altro processo (Interruzione esterna).
4. L'attesa per I/O è terminata.

NOTA: Se non ci sono processi in esecuzione il S.O. esegue un processo chiamato **IDLE**, il quale ha il solo compito di tenere impegnata la CPU