

# Informatica Generale

## Andrea Corradini

---

### **09 - Gestione della memoria, File System e periferiche**

# Quali sono le parti di un SO ?

lato  
utente

servizi richiesti dagli utenti

Interfaccia grafica  
(desktop)

nucleo del SO (kernel)

Gestore dei  
processi

Gestore dei  
processori

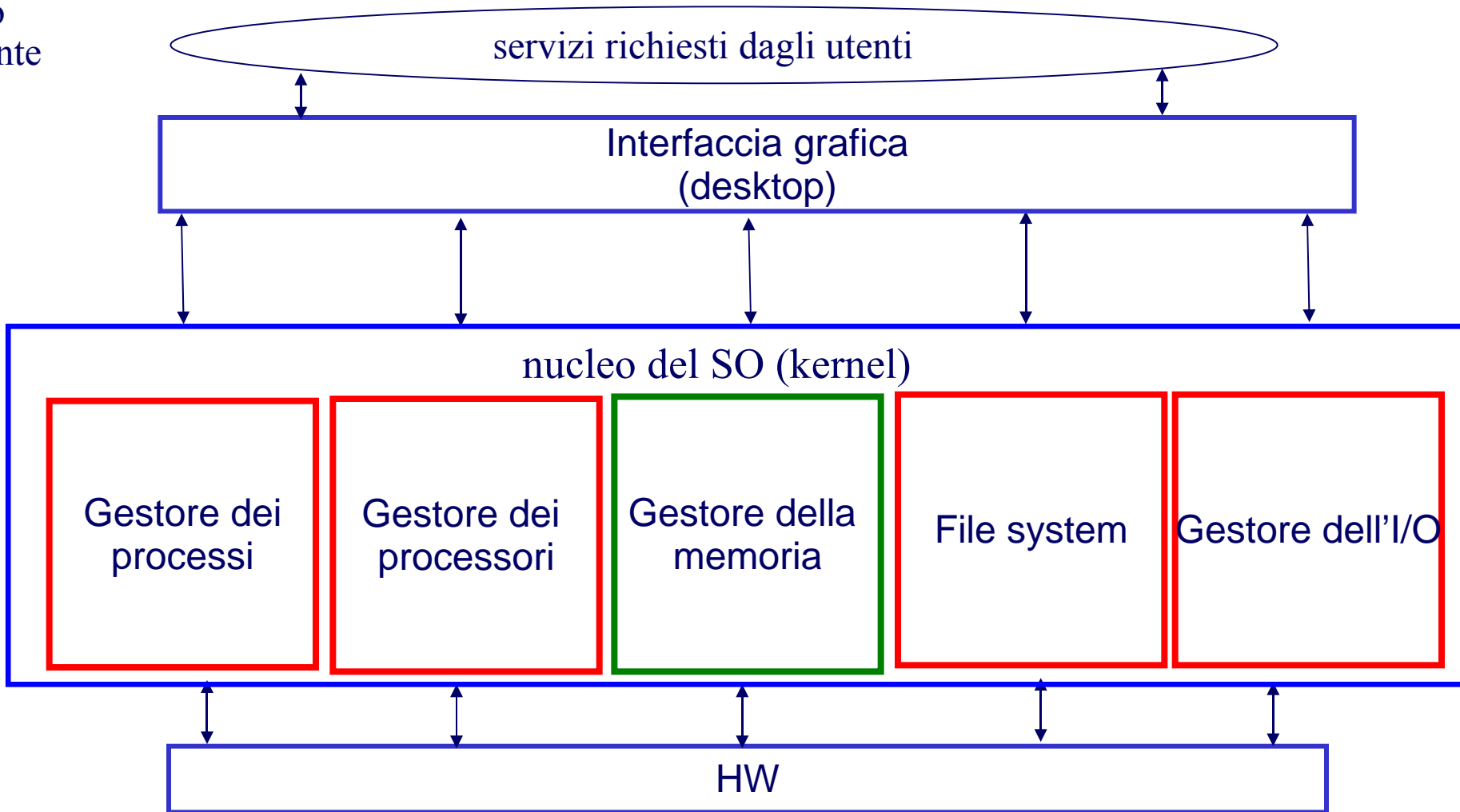
Gestore della  
memoria

File system

Gestore dell'I/O

HW

S  
I  
S  
T  
E  
M  
A  
  
O  
P  
E  
R  
A  
T  
I  
V  
O



# Il gestore della memoria

- Tutti i programmi che compongono il SO ed i programmi applicativi attivi usano contemporaneamente la RAM
- Il *gestore della memoria* si preoccupa di fare condividere la RAM ai vari processi in esecuzione in modo che :
  - ogni processo abbia il suo spazio *privato* distinto dagli altri (e inaccessibile agli altri)
  - ogni processo abbia abbastanza memoria per eseguire le proprie istruzioni e raccogliere i propri dati

# Il gestore della memoria 2

## ■ Gestione statica

- ricopiare interamente lo spazio di indirizzamento di un programma da memoria secondaria a RAM all'inizio dell'esecuzione

AmpiezzaRAM - 1

Una possibile  
organizzazione della RAM  
con più programmi attivi  
contemporaneamente



Area riservata, non accessibile  
in modalità utente

# Il gestore della memoria 3

- **Problemi della gestione statica**
  - non posso eseguire programmi con spazio di indirizzamento più grande della **RAM** !
  - frammentazione della memoria (ho “buchi” nella **RAM**)
- **Soluzione: memoria virtuale**
  - ciascun processo ha **uno spazio di indirizzamento virtuale** in cui lavorare, che può essere superiore alla memoria fisica presente nel calcolatore
  - solo parte dello spazio di indirizzamento è in **RAM**: quella acceduta di recente

# Swapping

- Quando la memoria centrale non ha dimensioni tali da contenere tutti i programmi da eseguire contemporaneamente effettua un trasferimento (**swapping**) del contenuto da un'area della memoria centrale in un'altra della memoria di massa (**area di swap**)
- Quindi i processi in che non sono in esecuzione possono essere in RAM o in memoria di massa: il diagramma degli stati è più complesso.



# Paginazione

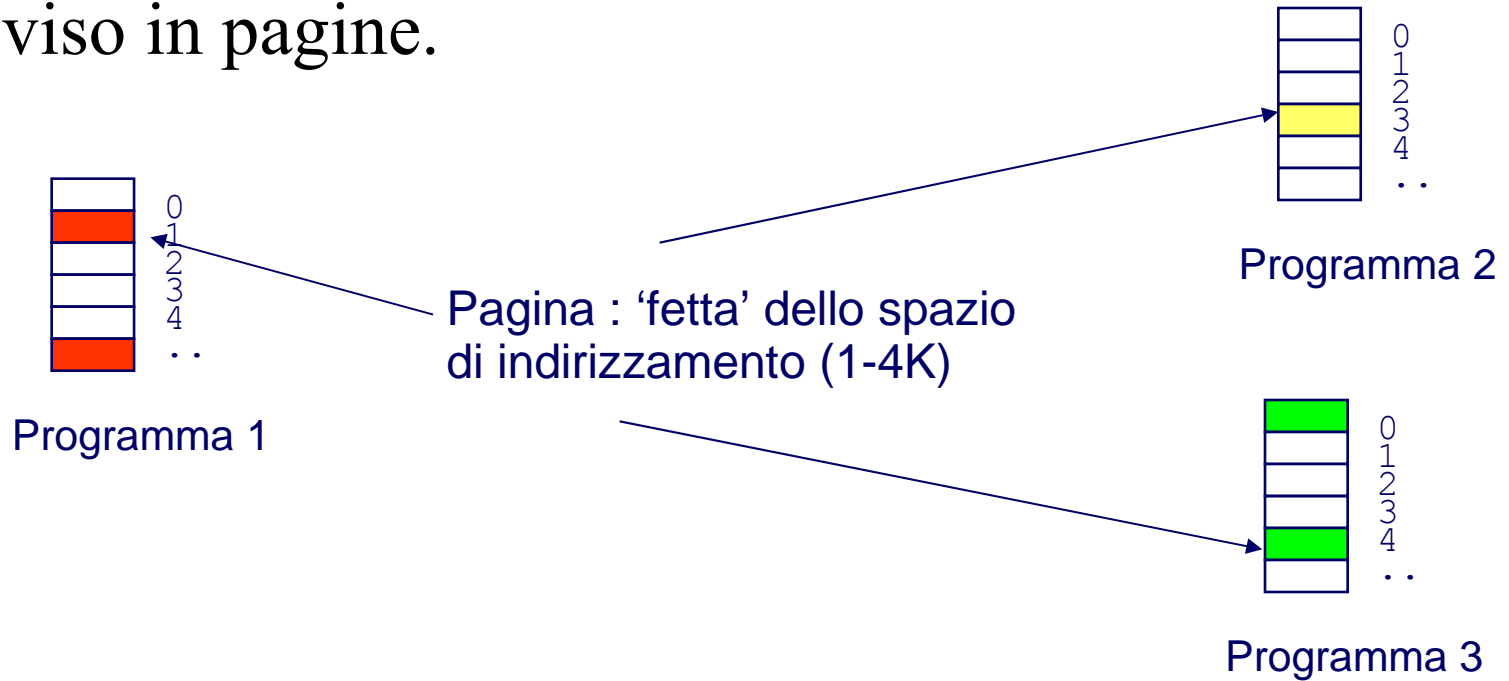
## **Gestione dinamica** della memoria:

- Lo spazio logico di indirizzamento del processo è suddiviso in ‘fette’, tutte della stessa ampiezza (**pagine logiche**)
- Lo spazio fisico di indirizzamento disponibile sul calcolatore è suddiviso in **pagine fisiche**, della stessa dimensione delle pagine logiche
- ad ogni istante carico in RAM solo i pezzi che mi servono per l’esecuzione corrente



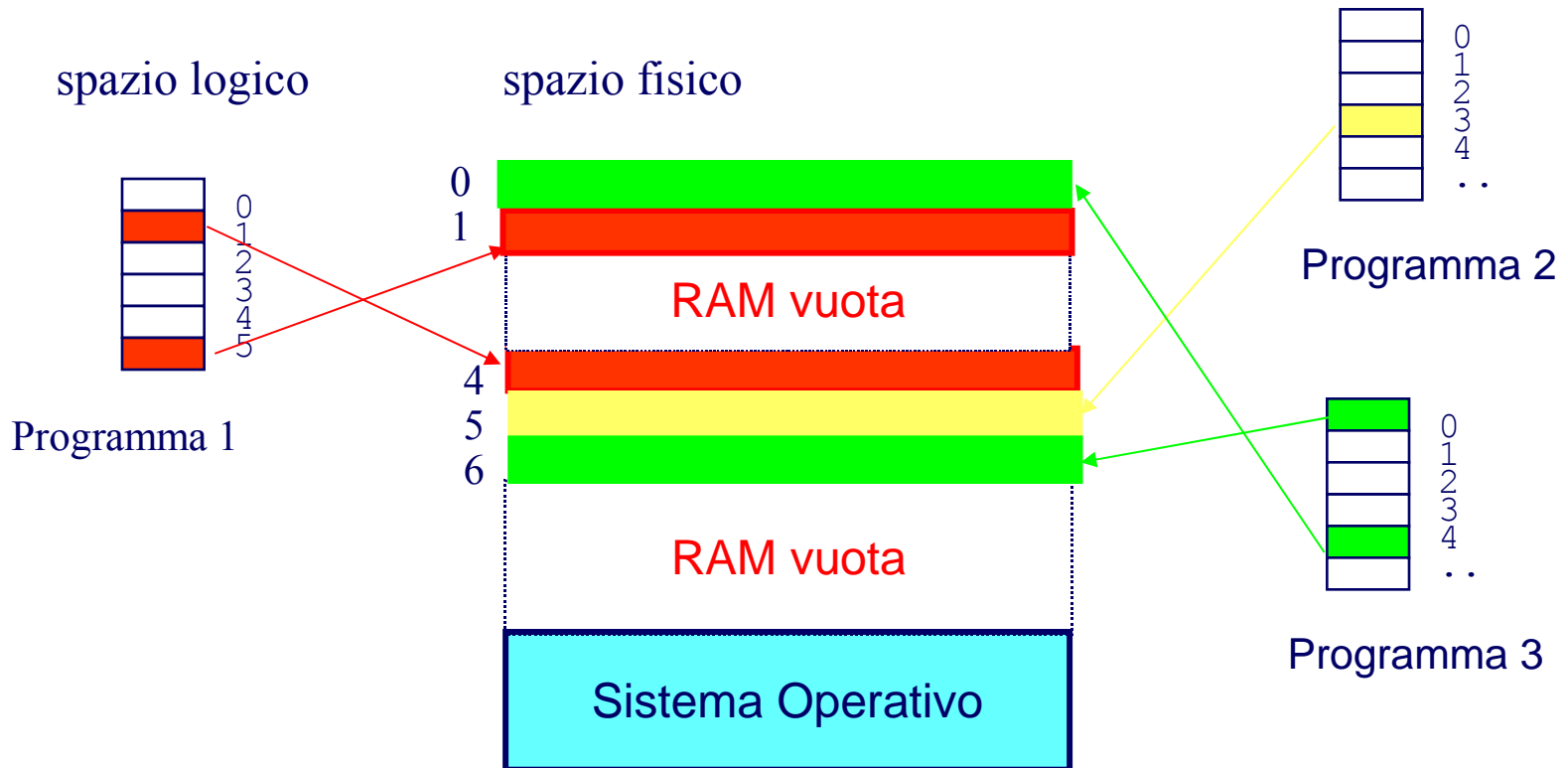
# Paginazione

Lo spazio di indirizzamento dei processi è diviso in pagine.



# Paginazione

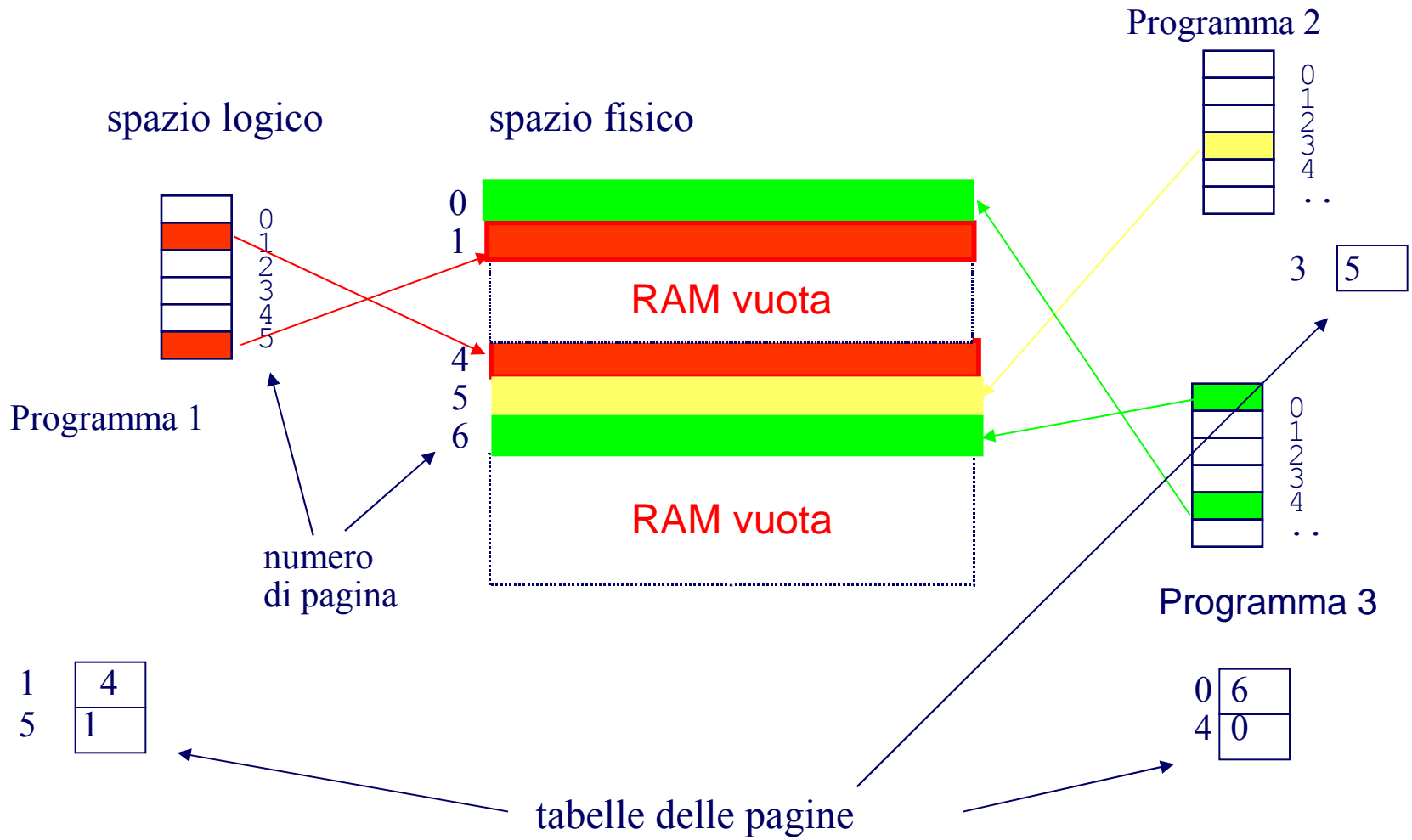
- ad ogni istante solo le pagine necessarie sono caricate in memoria (**località!**)



# Paginazione

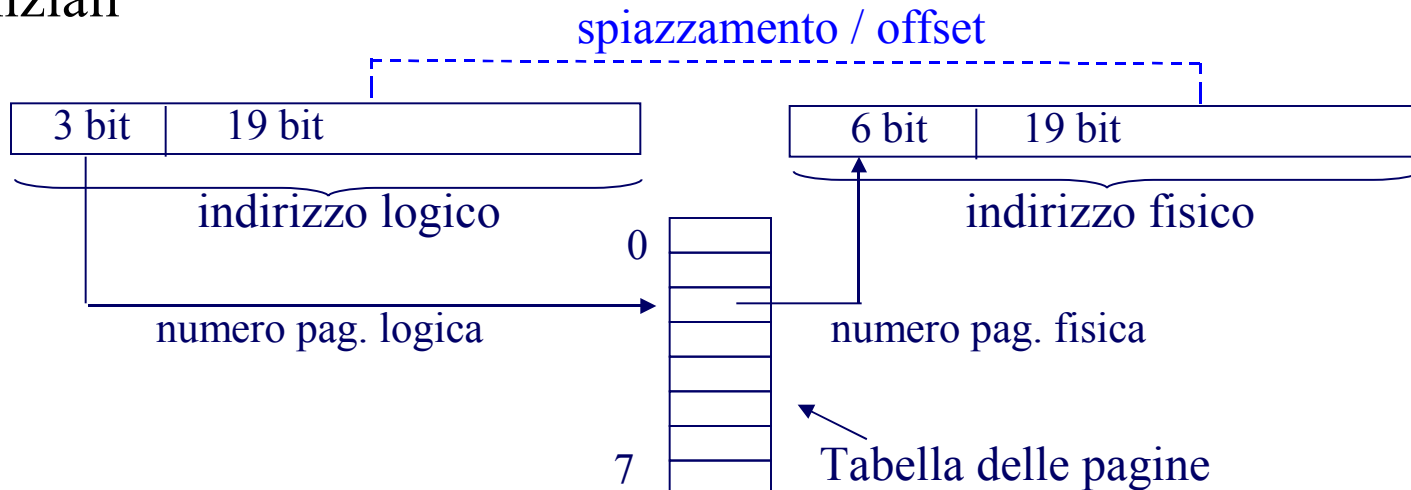
- Serve un dispositivo hardware aggiuntivo in grado di convertire gli indirizzi logici cui fa riferimento il programma nei corrispondenti indirizzi fisici: **Memory Management Unit**.
- La MMU utilizza una **tabella delle pagine** che:
  - mantiene la relazione tra ogni pagina logica e l'indirizzo della pagina fisica corrispondente

# Paginazione



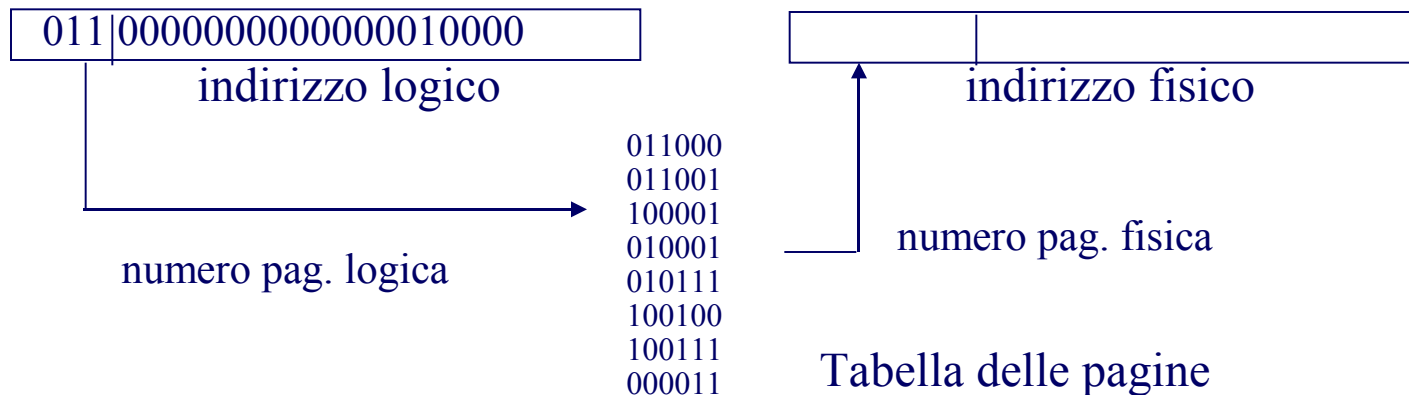
# Esempio di paginazione

- Dimensioni:
  - memoria fisica di 32MByte (indirizzata con 25 bit)
  - memoria logica di 4 MByte (indirizzo di 22 bit)
  - pagine lunghe 512 KByte (indirizzo 19 bit)
- I primi 3 dei 22 bit dell'indirizzo logico selezionano una delle  $2^3 = 8$  righe della tabella delle pagine, il cui contenuto rappresenta l'indirizzo della pagina fisica corrispondente, mentre i restanti 19 bit identificano lo spiazzamento (offset) relativo all'inizio della pagina specificata da 3 bit iniziali



# Paginazione: esercizio

- Dimensioni (come prima):
  - memoria fisica di 32MByte (indirizzata con 25 bit)
  - memoria logica di 4 MByte (indirizzo di 22 bit)
  - pagine lunghe 512 KByte (indirizzo 19 bit)
- Scrivere l'indirizzo fisico corrispondente all'indirizzo logico 011000000000000000010000



# Quali sono le parti di un SO ?

lato  
utente

servizi richiesti dagli utenti

Interfaccia grafica  
(desktop)

nucleo del SO (kernel)

Gestore dei  
processi

Gestore dei  
processori

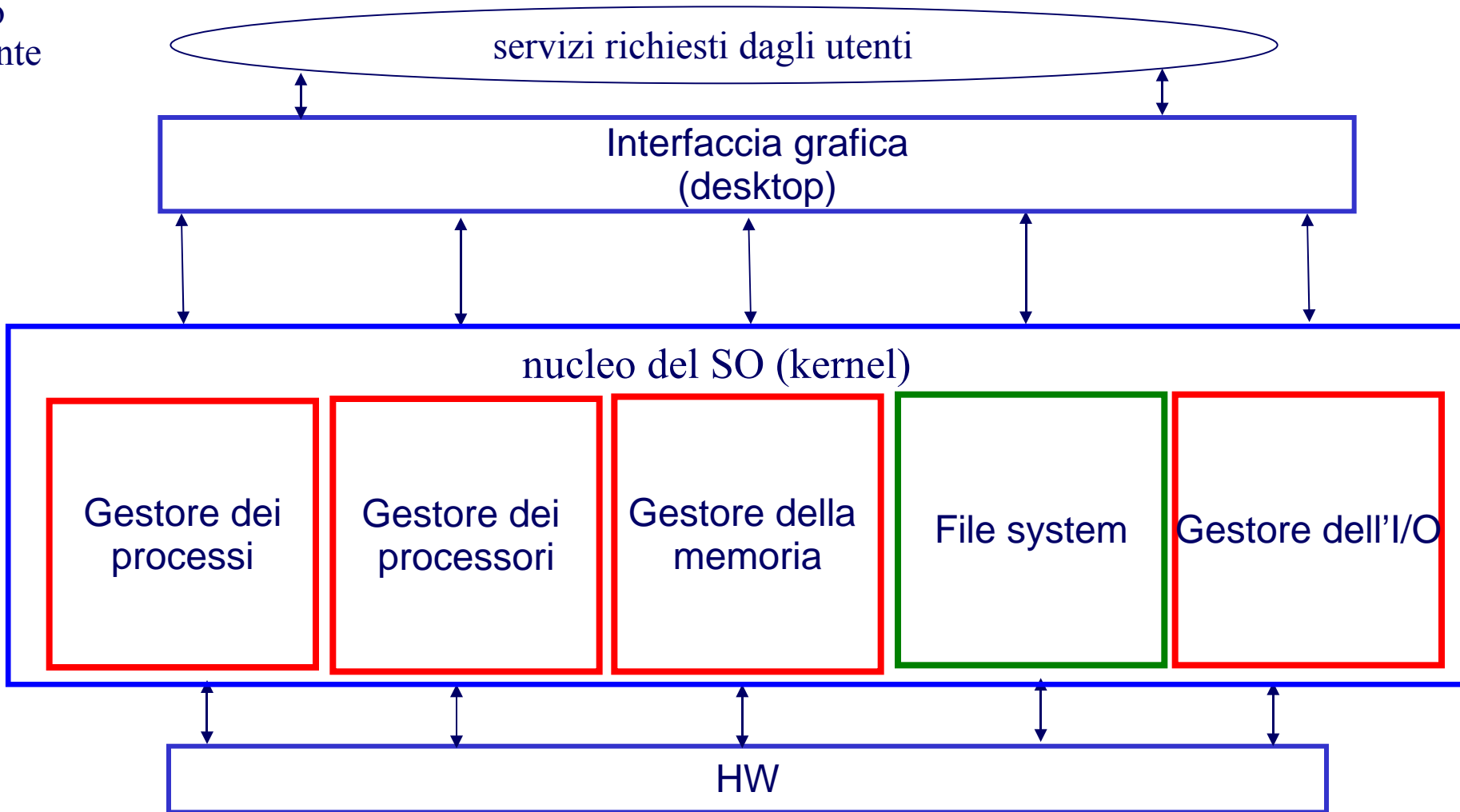
Gestore della  
memoria

File system

Gestore dell'I/O

HW

S  
I  
S  
T  
E  
M  
A  
  
O  
P  
E  
R  
A  
T  
I  
V  
O



# File System

---

- È la parte del SO che
  - permette di memorizzare dati e programmi in modo persistente
  - permette di organizzare dati e programmi in modo da renderne agevole la localizzazione da parte dell'utente umano
  - può essere modificato per cancellare dati obsoleti, aggiornare l'organizzazione etc ..



# File System 2

- Obiettivo:
  - presentare all'utente l'organizzazione logica dei dati e le operazioni che è possibile compiere su di essi.
- Operazioni di base di un file system
  - **recupero** di dati precedentemente memorizzati
  - **eliminazione** (cancellazione) di dati obsoleti
  - **modifica/aggiornamento** dei dati preesistenti
  - **copia** di dati (es. da HD a USB) per backup o per il trasferimento;
- I servizi vengono forniti sia ai programmi applicativi sia direttamente agli utenti

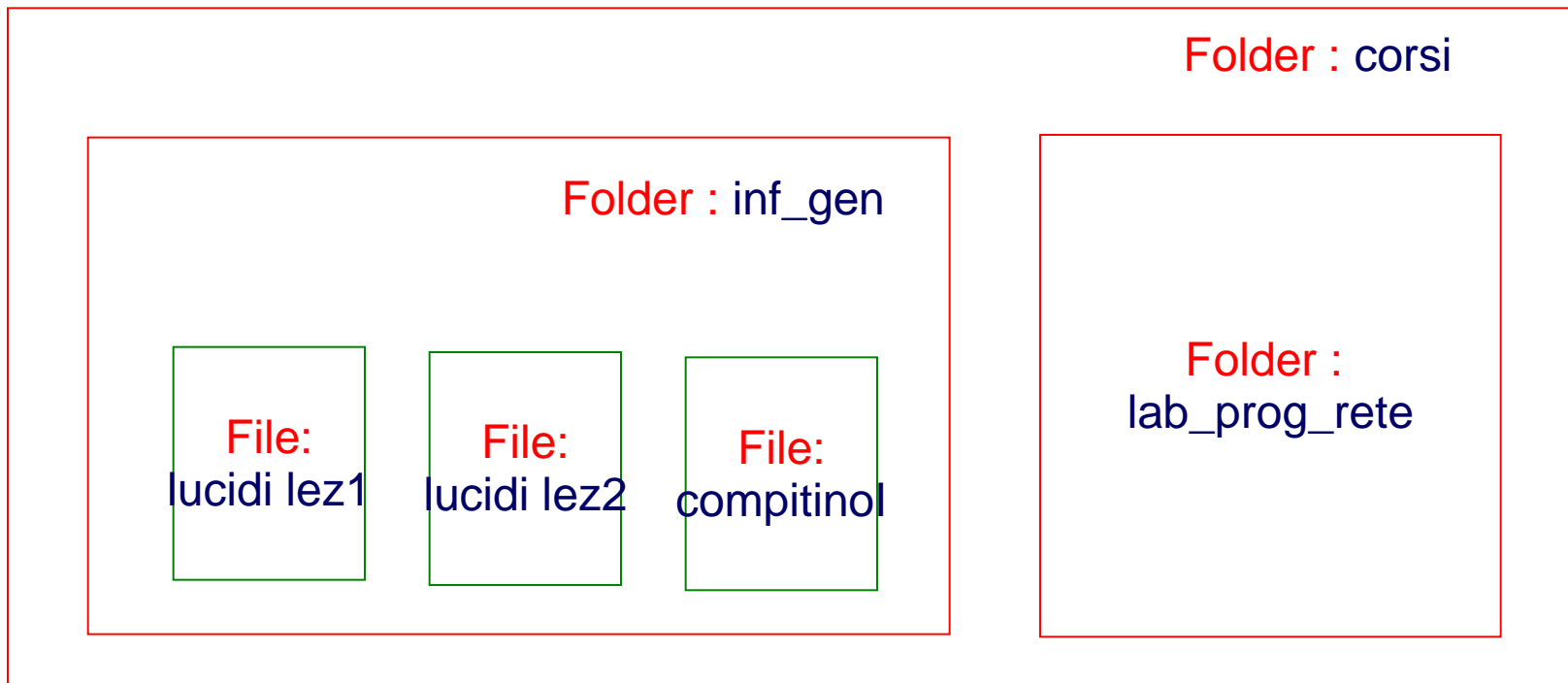
# File System 3

---

- *I contenitori* logici di informazioni sono :
  - il file (o archivio) : contiene dati o programmi rappresentati con una opportuna codifica binaria
  - il folder/directory (o cartella) : astrazione che permette di collezionare insieme più file e/o folder

# File System 4

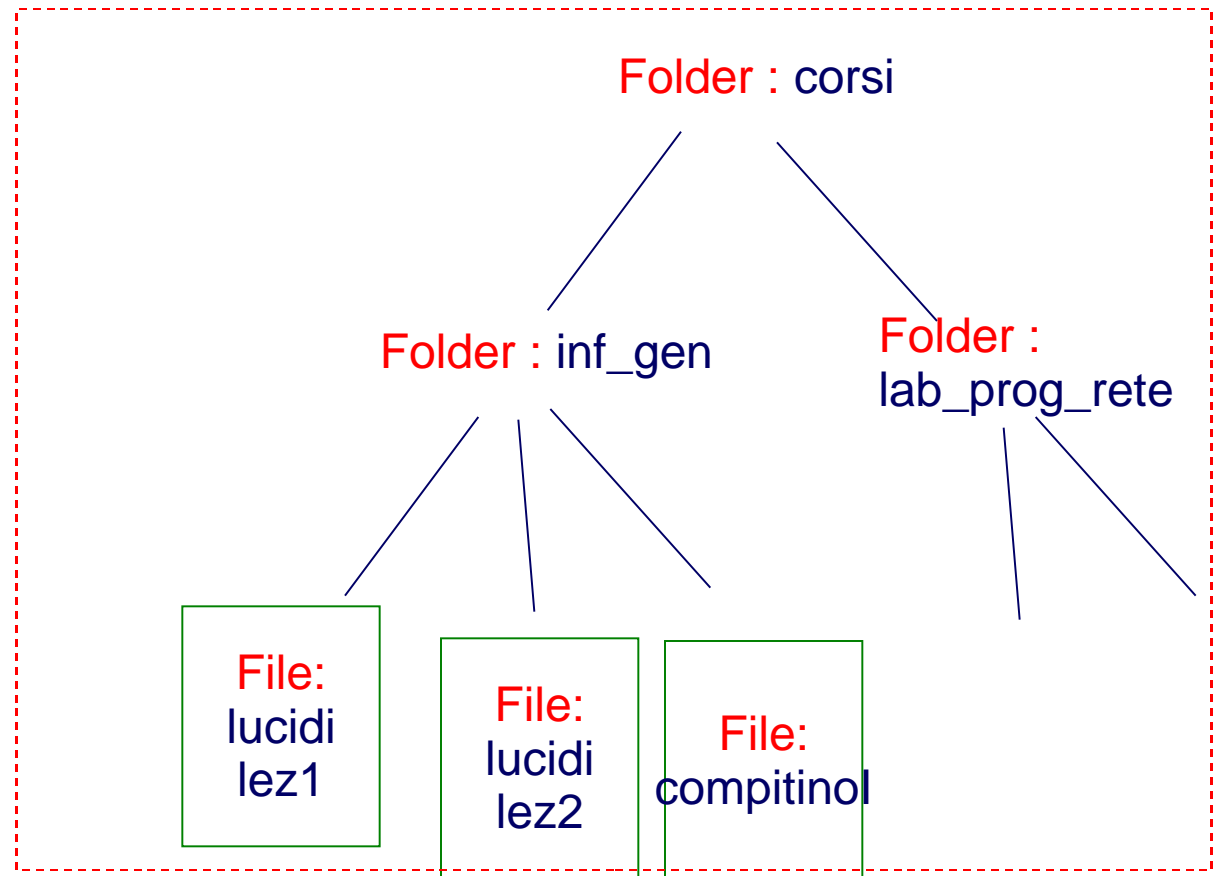
- Esempio : una possibile organizzazione dei file relativi ai corsi che insegno



# File System 5

- La struttura è gerarchica ad *albero* (i file possono essere solo foglie)

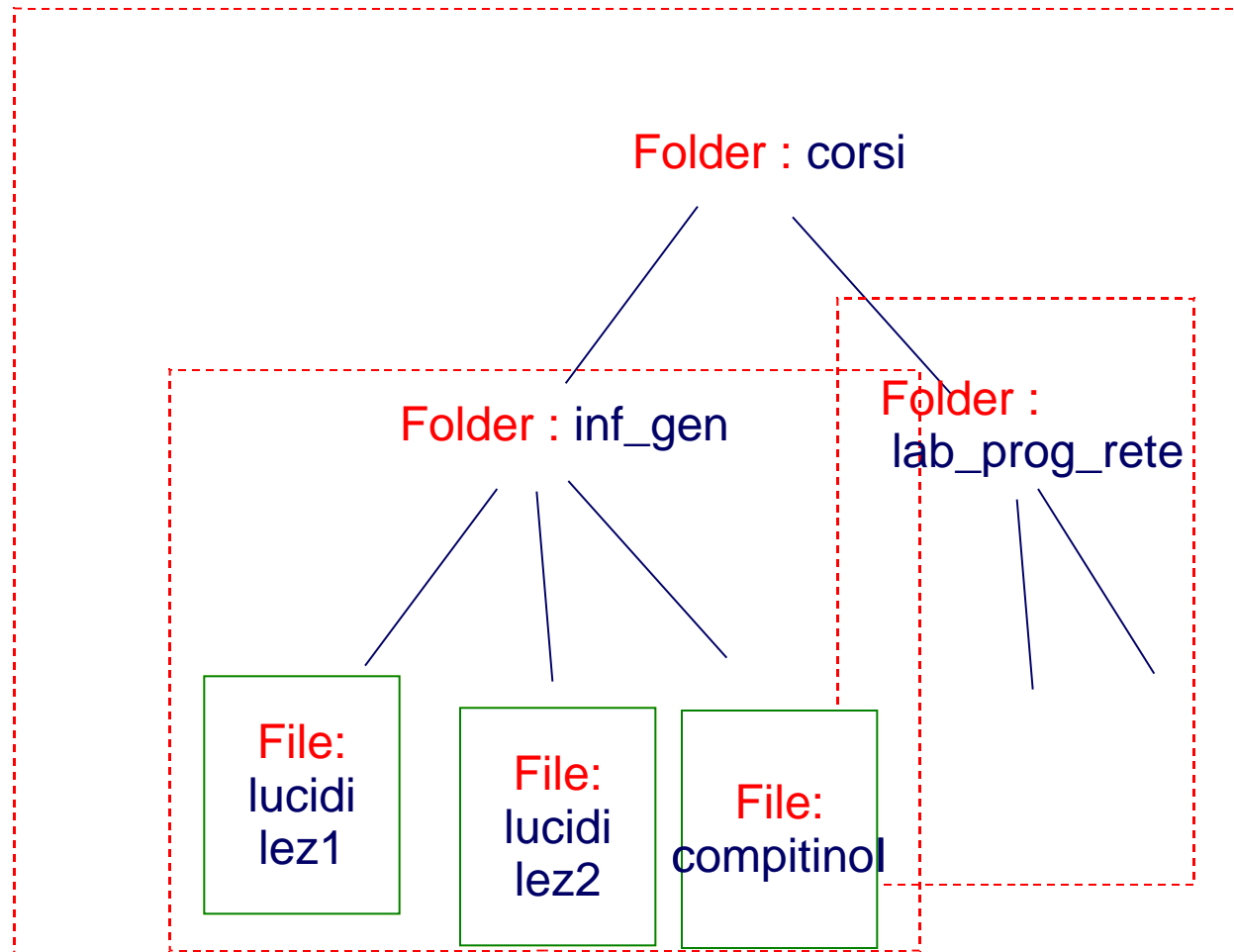
Contenuto del  
folder 'corsi'



# File System 6

- La struttura è gerarchica (ad *albero*)

Contenuti dei  
due folder  
inf\_gen e lab\_prog\_rete



# File

- Nome di un file
  - ogni file ha un nome (es **lez1.doc**, **lez2.ppt** etc)
  - il nome deve essere unico nel folder che contiene il file
  - la parte dopo il punto (.) si chiama ‘estensione’
  - di solito l’estensione da informazione sul formato del file e/o sull’applicazione utilizzata per crearlo

**lez2.ppt**

estensione

# File 2

- Nome di un file (cont.)
  - alcuni SO (come Windows) non mostrano normalmente l'estensione
    - **lez1.doc** viene mostrato come '**lez1**'
  - tuttavia l'estensione è sempre presente, e serve al sistema per capire di che tipo di file si tratta, e quale applicazione attivare quando desideriamo 'leggerlo'
    - es. un doppio click su un file **lez1** attiva automaticamente Word per visualizzarlo

# File 3

- Fisicamente, un file è una sequenza di byte che contiene dati o programmi rappresentati con una opportuna codifica binaria
- *Organizzazione dei file* :
  - **byte-stream** (successione sequenziale di byte)
  - **sequenziale** (sequenza di record di lunghezza fissa o variabile)
  - **diretto** (sequenza di record di lunghezza fissa)
  - **a indice** (sequenza dei record, ciascuno indicizzato)
- *Metodo di accesso* :
  - **sequenziale** (byte-stream, sequenziale, secondo la chiave del record)
  - **diretto** (diretto, secondo l'indice)



---

# Gestore delle periferiche di I/O

# Gestore delle periferiche I/O

- Comunicazione tra l'ambiente CPU-RAM e i dispositivi esterni
  - Asincronicità tra ambiente e calcolatore
  - Gestione dell'accesso contemporaneo al calcolatore da parte di diverse periferiche
- Mascherare ai processi l'esistenza di un numero limitato di risorse
  - Es. Stampa di uno o più processi (un browser e un programma di posta) su un'unica stampante
- Mascherare ai processi la differenza tra risorse dello stesso tipo (o di tipo simile)
  - Es. Stampante laser da un plotter e da una terminale video

# Driver e controller

Ogni periferica è gestita da due entità cooperanti, il controller e il driver

- **il controller :**

- un dispositivo hardware che colloquia direttamente con la parte elettrico/meccanica della periferica (operazioni di trasferimento dei dati)
- dipende dalle caratteristiche fisiche delle periferiche che gestisce
- l'interfaccia per la gestione di un mouse è sicuramente diversa da quella usata per gestire un lettore CD-ROM

# Driver e controller 2

Ogni periferica è gestita da due entità cooperanti, il controller e il driver

- **il driver :**

- un programma software che fa parte del sistema operativo e che conosce le caratteristiche specifiche della periferica (ed è capace di mascherarle) e colloquia con il controller
- fornisce un insieme di primitive ad alto livello per la gestione delle operazioni di I/O utilizzabili dai programmi applicativi e dagli utenti

# Driver e controller 3

- Solitamente il driver non è sviluppato da chi costruisce il SO (es. Microsoft) ma da chi costruisce il dispositivo
- I SO comprendono i driver per la gestione delle periferiche più comuni:
  - tastiera, video, mouse, etc.
  - stampanti, scanner, ...
- Quando viene acquistata una nuova periferica, il driver corrispondente deve essere inserito nel SO (**installazione**)
  - operazione rischiosa, perché se il nuovo driver è difettoso può rovinare il vostro sistema (il driver essendo nel SO ha accesso a tutta la macchina senza restrizioni!....)

# Driver e controller 4

- Quando si installa un nuovo driver bisogna fornirgli delle informazioni a basso livello per permettergli di colloquiare correttamente con il resto del sistema (**configurazione**)
  - es: stabilire quali linee del bus utilizzare per inviare una interruzione
- Questa operazione non è sempre facile!

# Plug&Play

---

- I sistemi operativi più recenti sono dotati di funzioni **plug&play** che permettono la configurazione automatica dei driver:
  - all'attivazione il SO scandisce e esamina tutte le periferiche collegate al sistema;
  - le periferiche si fanno riconoscere specificando quali driver servono
  - il sistema operativo installa i driver opportuni per la loro gestione

# Plug&Play 2

- Un sistema plug&play consente di aggiungere (**plug**) nuove periferiche al sistema che possono essere utilizzate (**play**) senza necessità di intervento da parte dell'utente per la selezione e l'installazione dei driver
- Una periferica plug&play è quella per cui la configurazione può essere fatta automaticamente dal sistema
  - sono generalmente tutte così nei PC odierni



# Spooling

- Lo **spooling** è un processo che consente di svincolare la stampa di uno o più file dal resto dell'elaborazione, senza attese inutili.
  - Quando un processo desidera stampare un file, invece di inviarlo direttamente alla stampante copia una versione per la stampa del documento
  - Il SO crea un processo di spooling che lo mette in un'opportuna *coda di spooling* e nei momenti di inattività della CPU fa procedere la stampa (svincolata dal programma che l'ha richiesta)
- Il processo di spooling può essere anche un processo *remoto* (cioè che risiede su un'altra macchina accessibile da una rete) che ha accesso a una o più stampanti.

# Quali sono le parti di un SO ?

lato  
utente

servizi richiesti dagli utenti

Interfaccia grafica  
(desktop)

nucleo del SO (kernel)

Gestore dei  
processi

Gestore dei  
processori

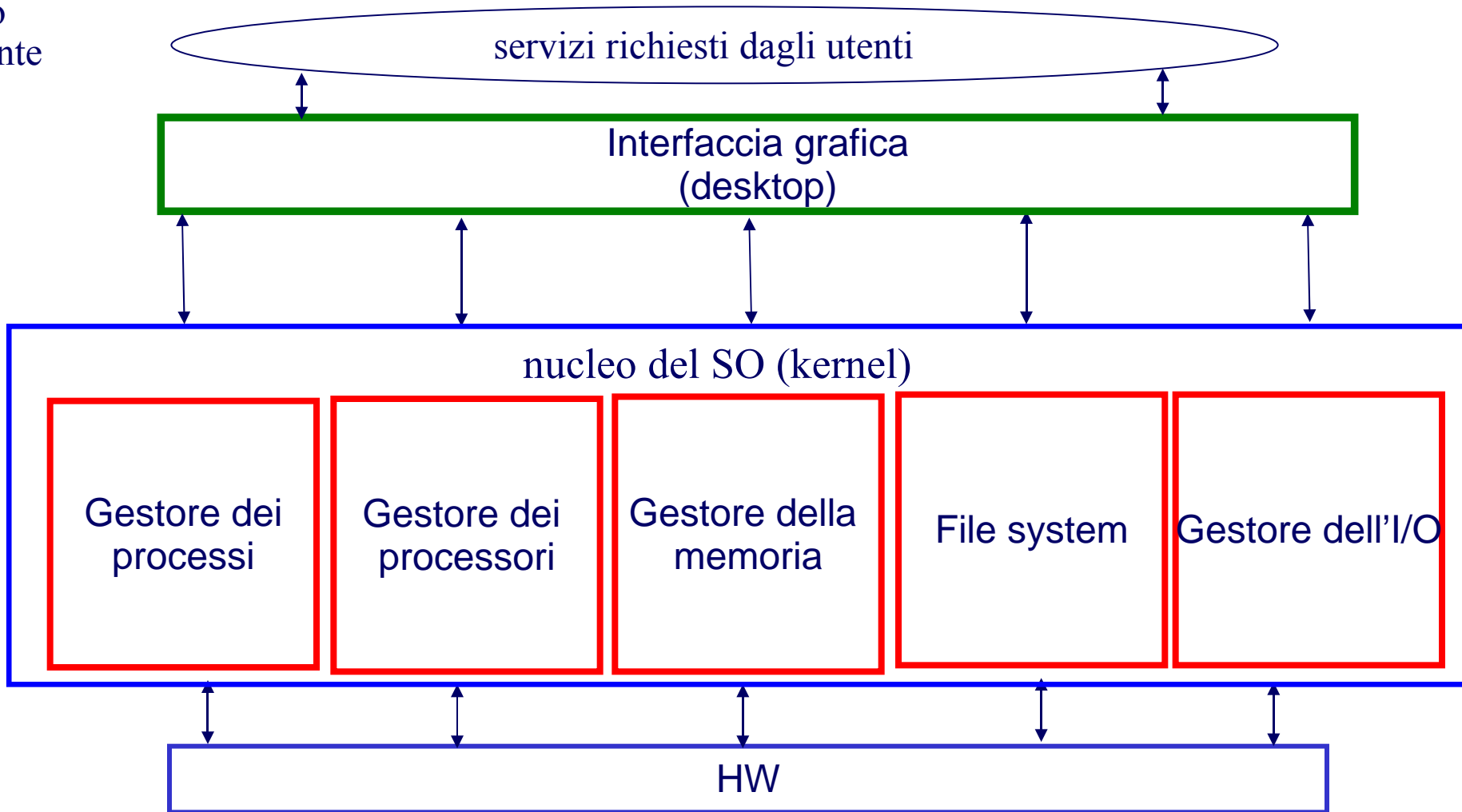
Gestore della  
memoria

File system

Gestore dell'I/O

HW

S  
I  
S  
T  
E  
M  
A  
  
O  
P  
E  
R  
A  
T  
I  
V  
O



# Interprete dei comandi e interfaccia

- L'interprete dei comandi (*shell*) fornisce l'interfaccia tra utente e macchina (funzionalità per mandare comandi al SO)
  - es. operazioni di lettura, copia file, esecuzione di un programma
- L'interfaccia dell'interprete può essere molto diversa
  - testuale (MS-DOS e shell di Unix/Linux): linea di comando tramite tastiera
  - grafica (Windows, Linux, MAC OS): desktop (icone e finestre) accessibile tramite mouse

# Esempi di SO: Windows

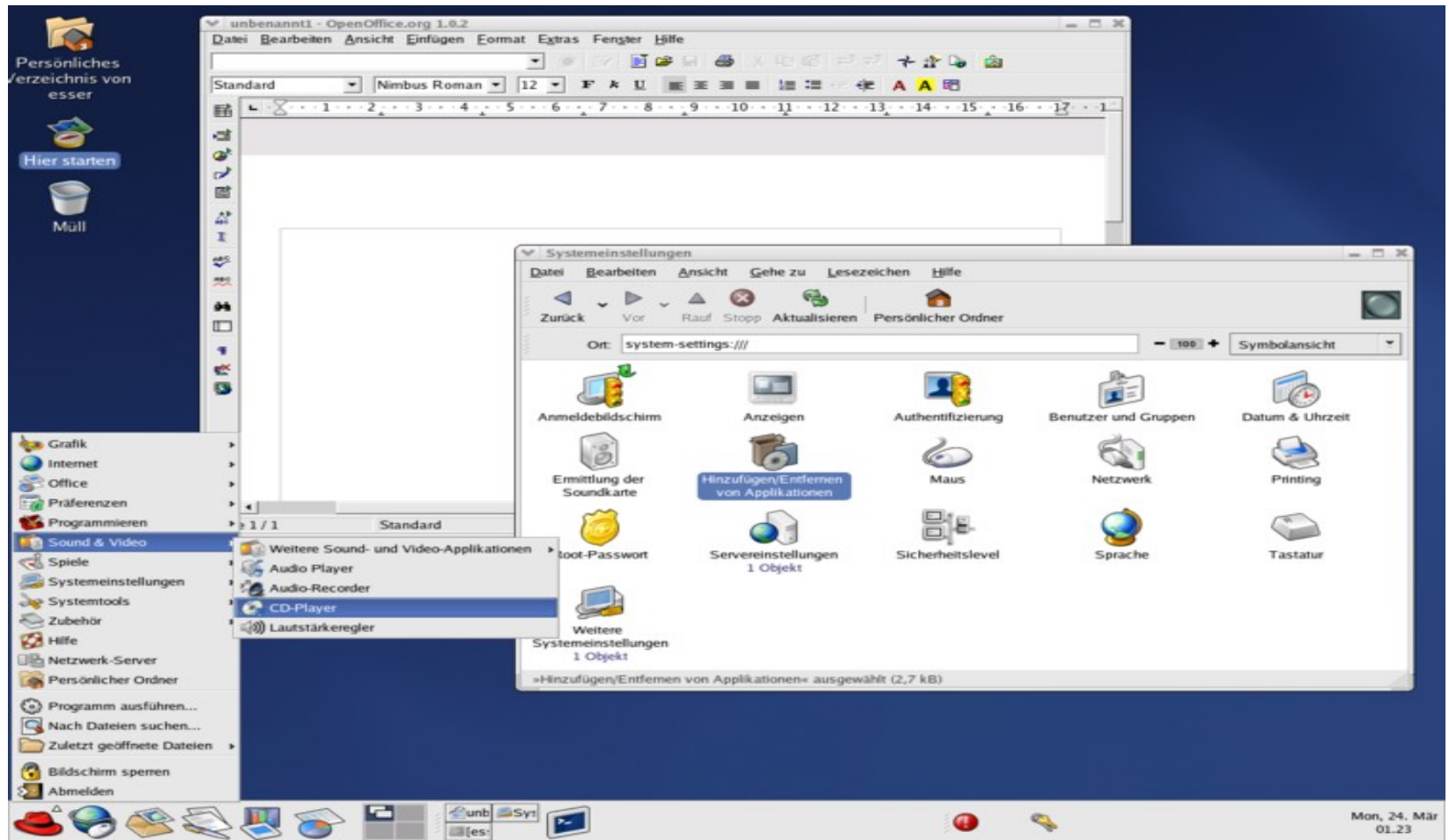
---

- **Microsoft Windows** è una famiglia di SO prodotta da Microsoft a partire dal 1985.
- Caratteristica: l'interfaccia grafica (desktop).
- Dimensioni delle parole:
  - Windows 95: 16 bit
  - Windows NT: 32 bit
  - Windows XP: 64 bit

# Esempi di SO: Linux

- E' un sistema operativo *libero* (codici sorgenti accessibili a tutti).
- Documentazione dettagliata (per installazione SO e applicativi) e guide disponibili gratuitamente sul web
- Interfaccia:
  - inizialmente solo testuale
  - successivamente grafica (molto simile a Mac OS e Windows), grazie all'integrazione di ambienti desktop come Gnome e KDE.
- Esistono varie distribuzioni a secondo degli scopi (Debian, Red Hat, Ubuntu, etc.)
- Software applicativo:
  - inizialmente mancavano alternative ai prodotti commerciale
  - attualmente ci sono varie possibilità: OpenOffice (come MS Office), gimp (come photo-shop), ...

# Una schermata di Red Hat



# Una schermata di Ubuntu

