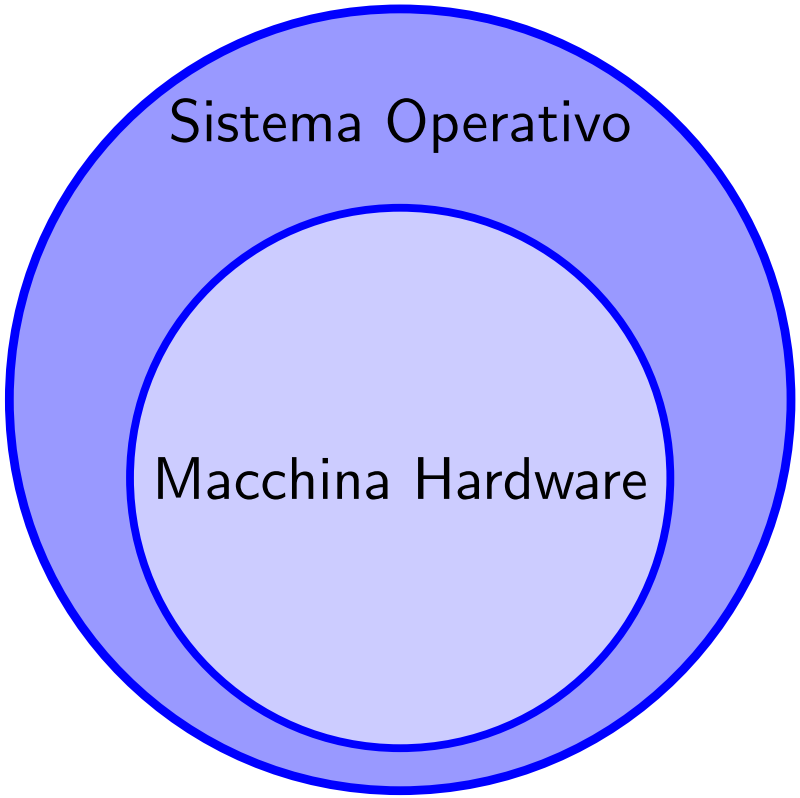
Informatica e Bioinformatica: Sistemi Operativi

Mauro Conti

27 marzo 2013

Macchina Hardware/Software



- La macchina hardware corrisponde alle componenti fisiche del calcolatore (quelle viste nella lezione precedente).
- Un sistema operativo è un programma che funge da interfaccia tra l'utente (e/o i programmi applicativi) e l'hardware del calcolatore.

Sistemi Operativi

Il software può essere suddiviso in due grandi classi:

- programmi applicativi, che risolvono i problemi degli utenti: editor di testi, browser, visualizzatore di immagini...
- programmi di sistema, che controllano e coordinano l'utilizzo delle risorse hardware da parte dei programmi applicativi dell'utente.

L'insieme dei programmi di sistema forma il sistema operativo (SO)

Sistemi Operativi

Esistono vari sistemi operativi:

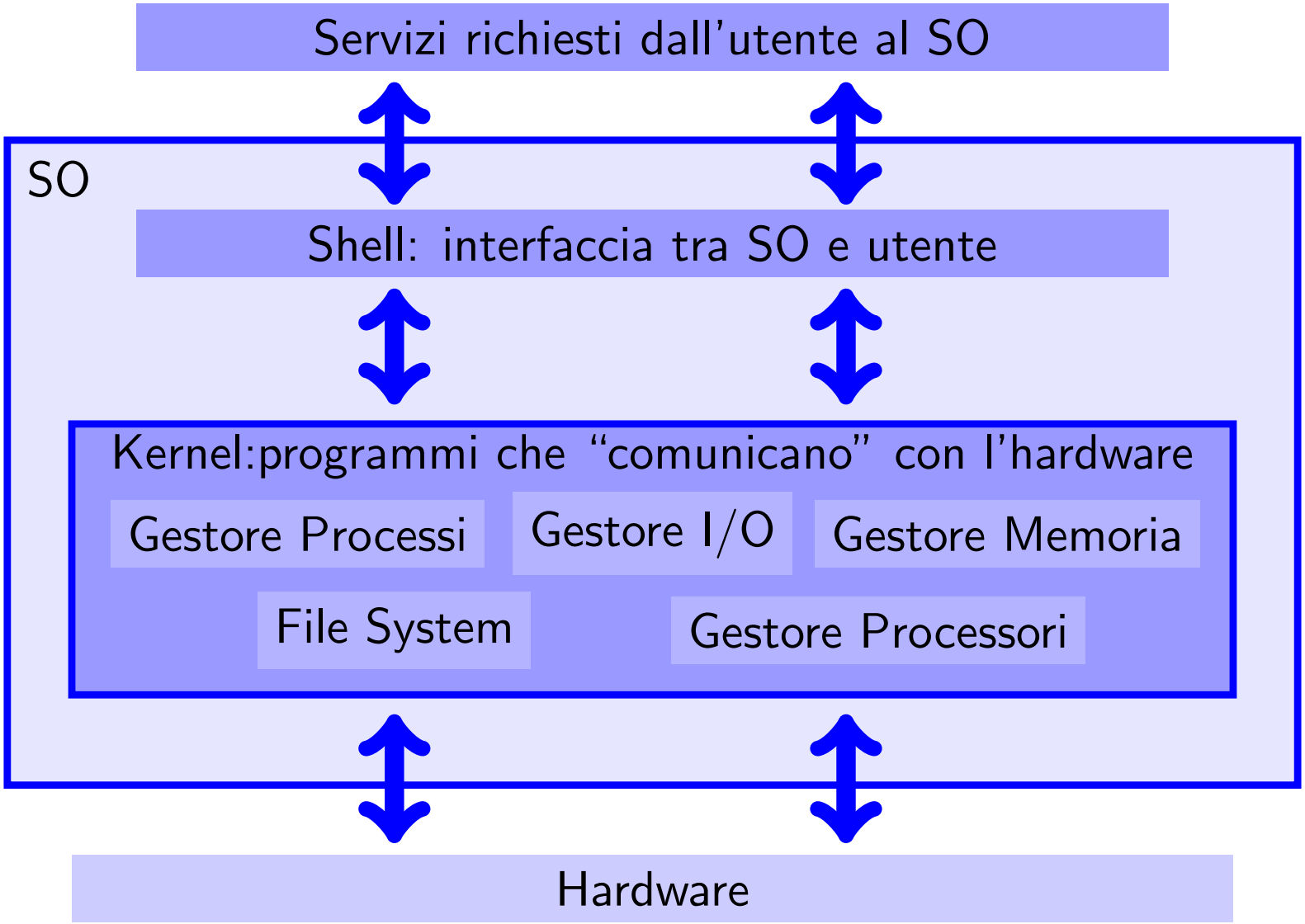
- Unix (Sviluppato inizialmente negli AT&T Labs)
- DOS (Microsoft), basato su unix, in disuso
- Windows (Microsoft)
- Mac OS (Apple), basato su unix
- Linux, open source, basato su unix
- Android (google), basato da linux

Struttura di un Sistema Operativo

Elementi principali di un SO:

- Shell (guscio), l'interfaccia tra SO e utente
- Kernel (nucleo), l'insieme di programmi che realizzano le funzioni di base di un calcolatore: gestione della memoria principale, gestione di più programmi in esecuzione contemporaneamente, gestione della memoria di massa, ...

Struttura del Sistema Operativo

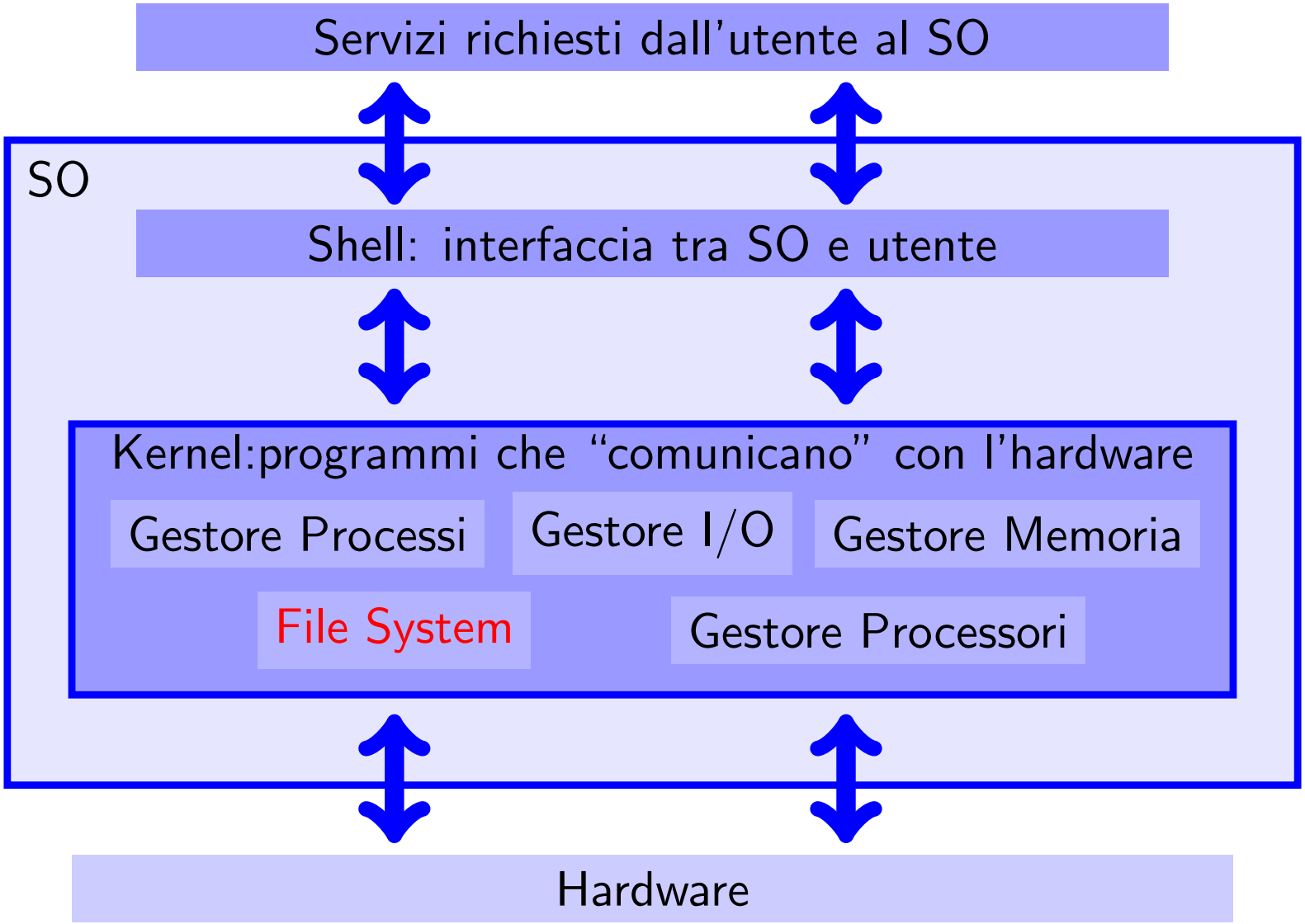


Sistema Operativo: Shell

La Shell è l'interfaccia tra SO e utente:

- è il programma che permette agli utenti di comunicare con il sistema e di avviare programmi applicativi.
- La shell può essere grafica oppure testuale
  - All'avvio di Windows, Mac OS, Linux, Android, viene caricata la shell grafica
  - DOS aveva solo una shell testuale, Mac OS e soprattutto linux hanno anche una potente shell testuale

Struttura del Sistema Operativo



La gestione della memoria secondaria

Poichè la memoria principale è volatile e troppo piccola per contenere tutti i dati e tutti i programmi in modo permanente, il computer è dotato di memoria secondaria (hard disk)

- Il SO garantisce una visione logica uniforme del processo di memorizzazione:
  - Nasconde le caratteristiche fisiche della memoria di massa:
    - Esempio: abbiamo creato un'immagine con un programma di fotoritocco e clicchiamo su "Salva" per immagazzinarla nell'hard disk
    - A questo punto il SO invoca il gestore della memoria di massa, che sceglie dove mettere fisicamente l'immagine sul disco, inizia a far ruotare il disco, sposta la testina ecc. . .
    - Se tali comandi facessero parte del programma di fotoritocco, esso dovrebbe incorporare le istruzioni dettagliate per ogni tipo di memoria di massa sul mercato. Ancora peggio, un secondo programma applicativo dovrebbe duplicare tutte queste istruzioni!
    - La soluzione più semplice è delegare al SO il compito di conoscere tutte le istruzioni per salvare i dati in ogni tipo di memoria di massa.

La gestione della memoria secondaria

Poichè la memoria principale è volatile e troppo piccola per contenere tutti i dati e tutti i programmi in modo permanente, il computer è dotato di memoria secondaria (hard disk)

- Il SO garantisce una visione logica uniforme del processo di memorizzazione:
  - Nasconde le caratteristiche fisiche della memoria di massa.
  - Fornisce all'utente e a programmi applicativi un modello astratto secondo il quale i dati sono organizzati: il File System
    - I componenti base del modello sono i file e le cartelle
    - Un file è una sequenza di byte che contiene informazioni tipicamente "omogenee", ad esempio un programma, un testo, un'immagine, una canzone, un video, ...
    - una cartella (directory) è un contenitore di file (e cartelle)

File

- Un file è una sequenza di byte che contiene informazioni tipicamente "omogenee"
  - ad esempio un programma, un testo, un'immagine, una canzone, un video, ...
- Per ogni file vengono memorizzate varie ulteriori informazioni
  - il nome, solitamente nella forma nomefile.estensione; ad esempio sistemioperativi.pdf, mare.jpg, elio.mp3, ...
  - data di creazione e ultima modifica
  - dimensione
  - diritti di accesso
  - posizione effettiva dei dati nella memoria di massa

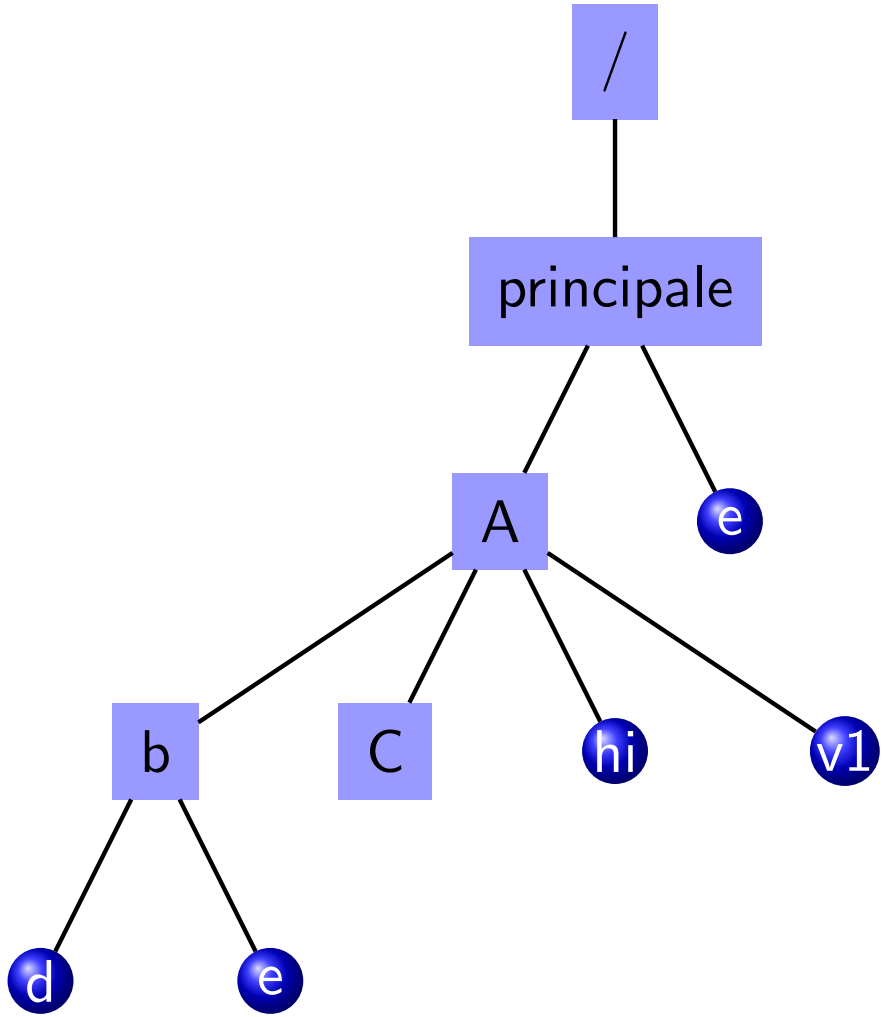
File System

- Il file è l'astrazione informatica di un archivio di dati. Il concetto di file è indipendente dal mezzo sul quale viene memorizzato (che ha caratteristiche proprie e una propria organizzazione fisica)
- Un file system è composto da un insieme di file
- Il SO è responsabile delle seguenti attività riguardanti la gestione del file system:
  - Creazione e cancellazione di file
  - Creazione e cancellazione di directory
  - Manipolazione di file e directory
  - Codifica del file system sulla memoria secondaria

File System

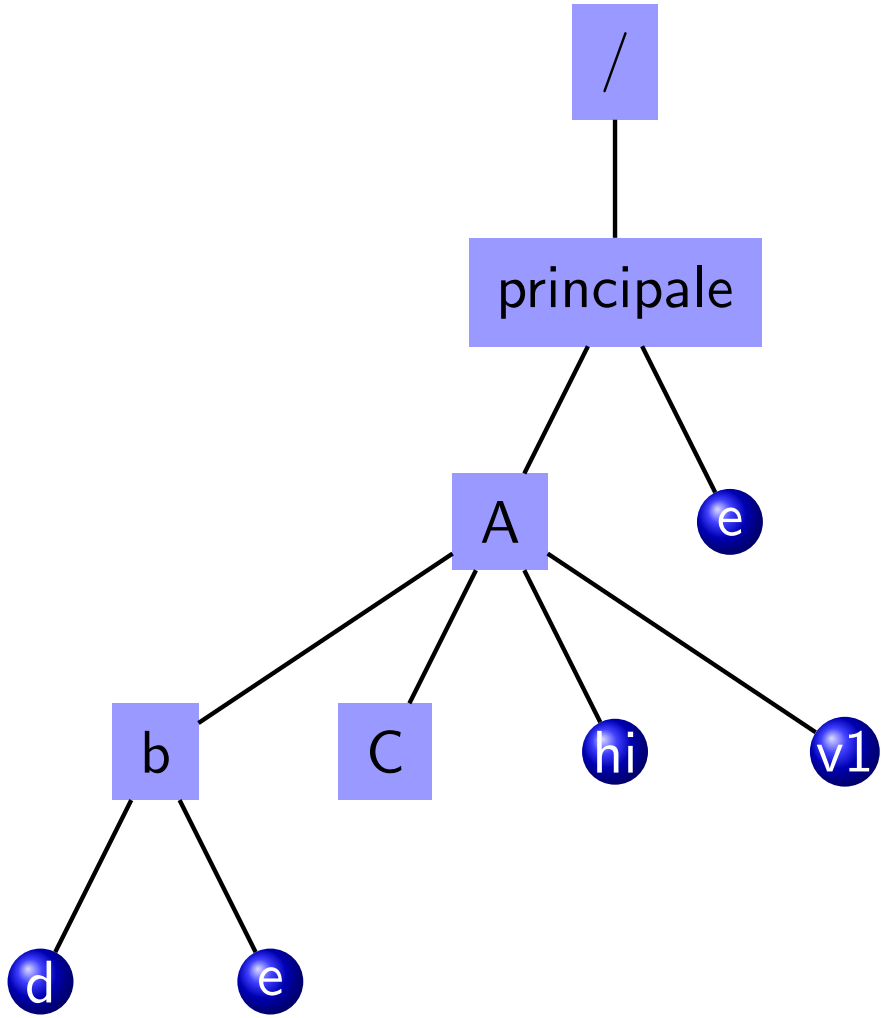
Quasi tutti i SO utilizzano un'organizzazione gerarchica del file system ad albero (o a grafo in alcuni casi).

- Nella figura i nodi rettangolari rappresentano le cartelle mentre i nodi circolari i file
- Un file è identificato dal percorso dalla radice dell'albero: il percorso del file "hi" è "/principale/A/hi"
- Possono esistere file con lo stesso nome se sono in cartelle diverse (ad esempio i file "e")
- Nel SO Linux la radice, ed il carattere per separare le cartelle, sono indicati con /;



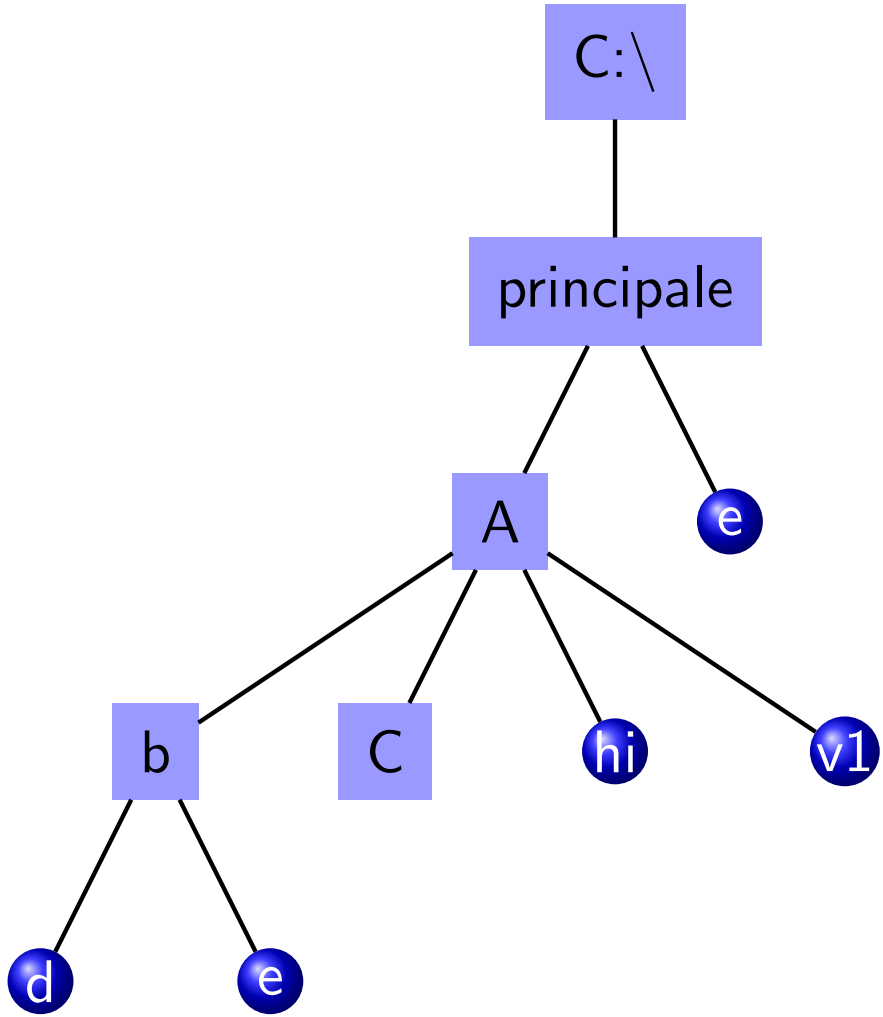
File System

- È possibile navigare all'interno del file system percorrendo gli archi in un qualsiasi verso. Ad esempio, se mi trovo nella cartella "A" e sto utilizzando una shell grafica, cliccando su "b" mi sposto nella cartella "b"
- Per riferirmi ad un file posso indicare
  - il percorso dalla radice dell'albero (percorso assoluto)
  - il percorso a partire dalla mia posizione corrente (percorso relativo). Ad esempio, se mi trovo nella cartella "A" posso riferirmi a "d" semplicemente come "b/d"



File System Windows

- Il SO Windows utilizza una rappresentazione ad albero per ogni memoria di massa.
- La radice di ciascun albero è indicata con una lettera seguita da due punti e \, ad esempio C:\oppure D:\
- Il carattere per separare le cartelle è \
- il percorso assoluto del file “hi” è “C:\principale\A\hi”



SO Mono/Multitasking

SO mono/multitasking

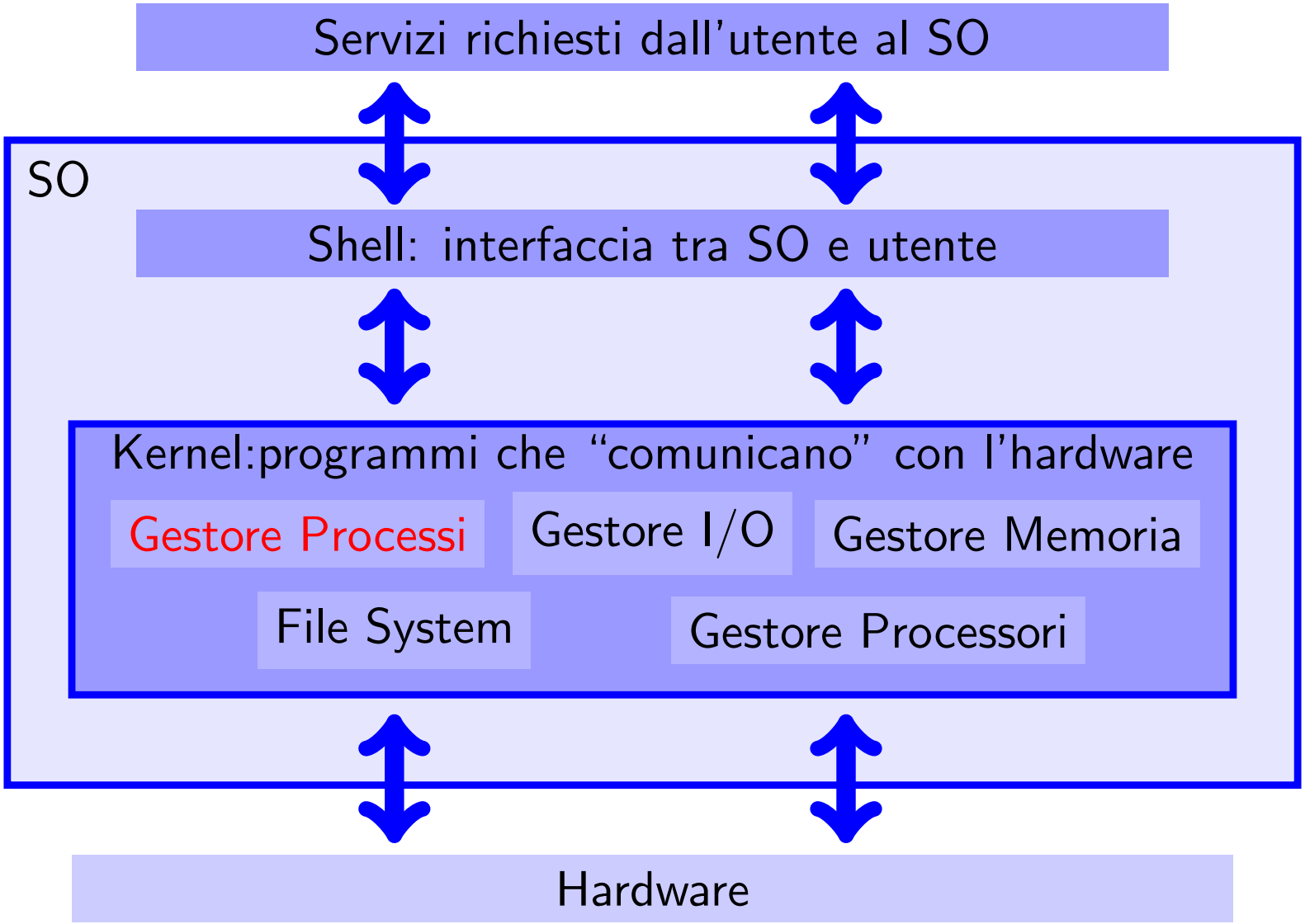
- I primi SO erano monotasking (ad esempio il DOS), ovvero in grado di gestire l'esecuzione di un solo programma per volta
- Tutti i moderni SO sono multitasking: Windows, Linux, Mac OS
- Ma se il calcolatore ha una sola CPU, come fa ad eseguire più programmi contemporaneamente?

SO Mono/Multitasking

SO mono/multitasking

- I primi SO erano monotasking (ad esempio il DOS), ovvero in grado di gestire l'esecuzione di un solo programma per volta
- Tutti i moderni SO sono multitasking: Windows, Linux, Mac OS
- Ma se il calcolatore ha una sola CPU, come fa ad eseguire più programmi contemporaneamente?
- Affidandosi al gestore processi

Struttura del Sistema Operativo



Gestore Processi

- Un SO consente il caricamento in memoria e l'esecuzione di più programmi che si alternano nell'uso della CPU.
- Per far ciò un programma può essere eseguito, sospeso e fatto ripartire più volte.
- Programma: insieme statico di istruzioni
- Processo: programma + stato dell'esecuzione. Lo stato dell'esecuzione consiste nell'insieme delle informazioni che sono necessarie per far ripartire il programma dal punto esatto in cui è stato sospeso:
  - contenuto dei registri della CPU
  - valori delle celle della Memoria principale assegnate al programma
  - prossima istruzione da eseguire

Gestore dei Processi

Il sistema operativo è responsabile, tra le altre cose, delle seguenti attività riguardanti la gestione dei processi:

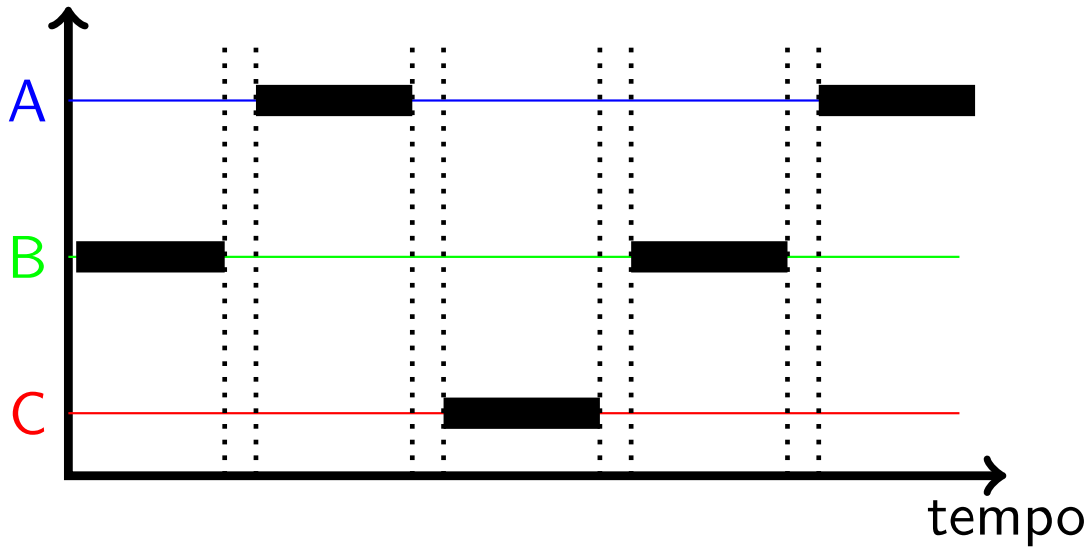
- creazione e terminazione dei processi
- sospensione e riattivazione dei processi
- comunicazione tra processi

Il gestore dei processi "realizza" una macchina virtuale in cui ciascun programma opera come se avesse a disposizione un'unità di elaborazione dedicata: se sto eseguendo 3 programmi contemporaneamente, il gestore ci fa credere di avere 3 CPU.

Ripartizione del tempo della CPU tra i processi

Esistono varie politiche per decidere quando e per quanto tempo assegnare la (le) CPU ad un processo:

- nei sistemi time-sharing ad ogni processo è assegnata la stessa quantità di tempo.

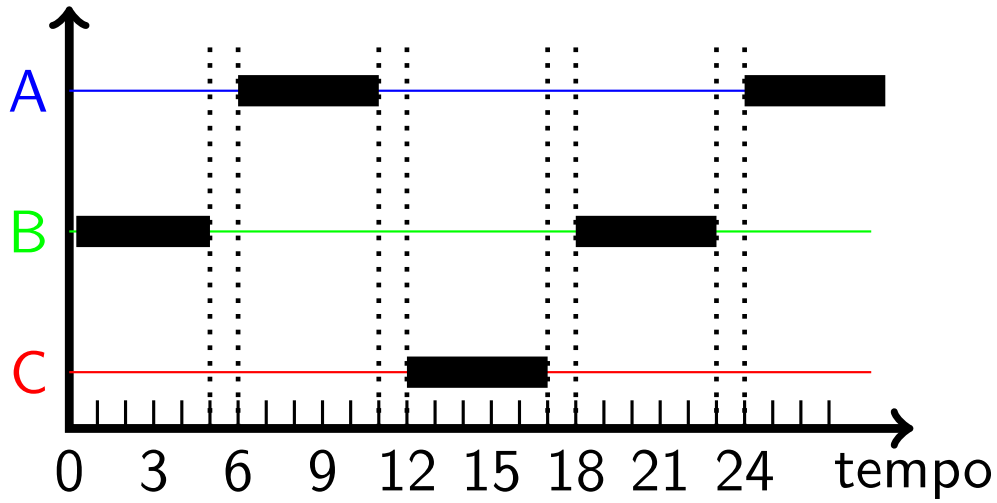


- Nel grafico i rettangoli neri indicano il tempo che la CPU ha dedicato ai 3 processi A,B,C. Lo spazio tra le linee tratteggiate indica il tempo impiegato per la sospensione del processo in esecuzione ed il ripristino del successivo.

Ripartizione del tempo della CPU tra i processi

- Chi progetta un gestore dei processi, come sceglie la quantità di tempo da assegnare ad un processo?
- In generale il gestore dei processi deve:

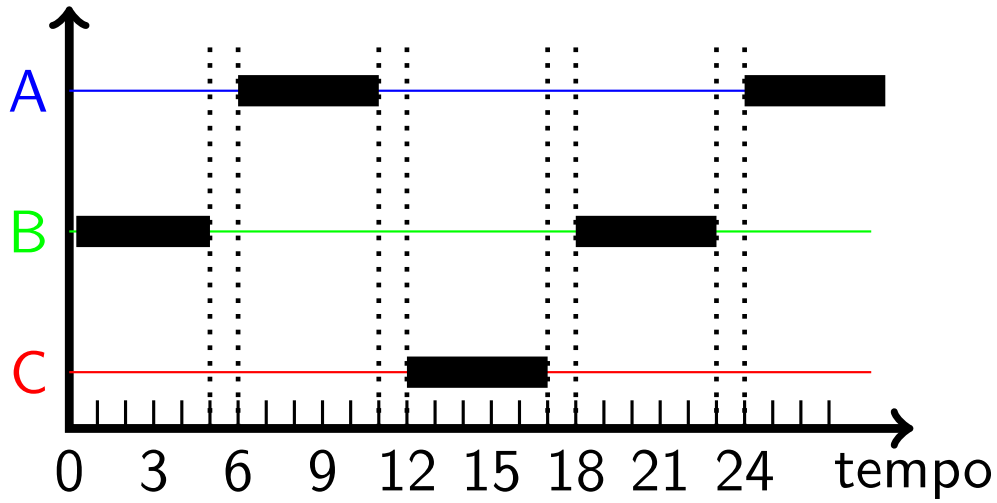
- assicurare che ogni processo non debba aspettare eccessivamente per ottenere la CPU (altrimenti l'utente si spazientisce!); in altri termini il tempo tra la sospensione e il ripristino di un processo non deve essere troppo lungo (nella figura sotto B viene sospeso all'istante 5 e riavviato all'istante 18 (13 unità di tempo di attesa)



Ripartizione del tempo della CPU tra i processi

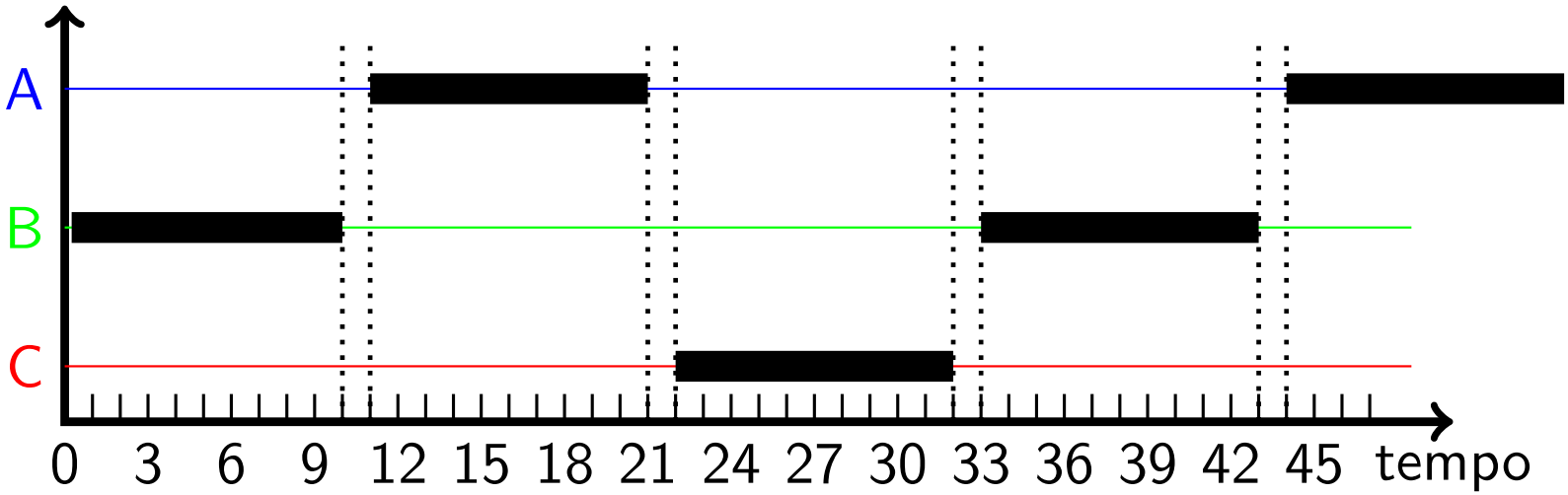
- Chi progetta un gestore dei processi, come sceglie la quantità di tempo da assegnare ad un processo?
- In generale il gestore dei processi deve:

- massimizzare il tempo in cui la CPU esegue programmi e minimizzare il tempo speso a sospendere,ripristinare processi (la CPU deve eseguire programmi, non perdere tempo a fare altro!); in altri termini vogliamo mantenere piccolo la somma dei tempi tra le linee verticali tratteggiate (nell'esempio tale somma è 4 nei primi 25 istanti)



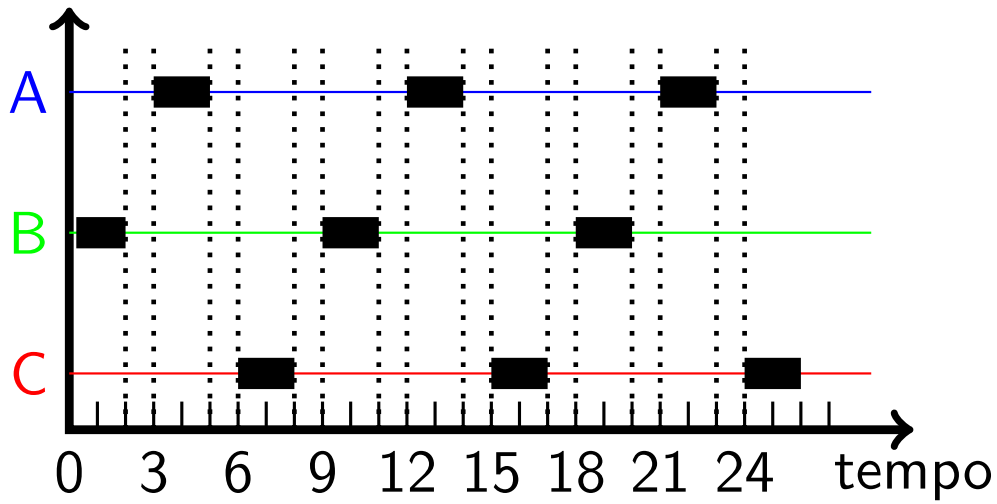
Ripartizione del tempo della CPU tra i processi

- Chi progetta un gestore dei processi, come sceglie la quantità di tempo da assegnare ad un processo (la lunghezza di ciascun rettangolo nero nel grafico)?
- Analizziamo i due casi limite:
  - se si aumenta smisuratamente il tempo da dedicare ad un processo (se i rettangoli neri sono molto lunghi) diminuisce il tempo dedicato alla sospensione/ripristino dei processi (da 4 a 2 in 25 istanti) ma aumenta il tempo tra una sospensione ed una riattivazione di un processo (B viene sospeso a 10 e riattivato a 33, ovvero 23 istanti di attesa)

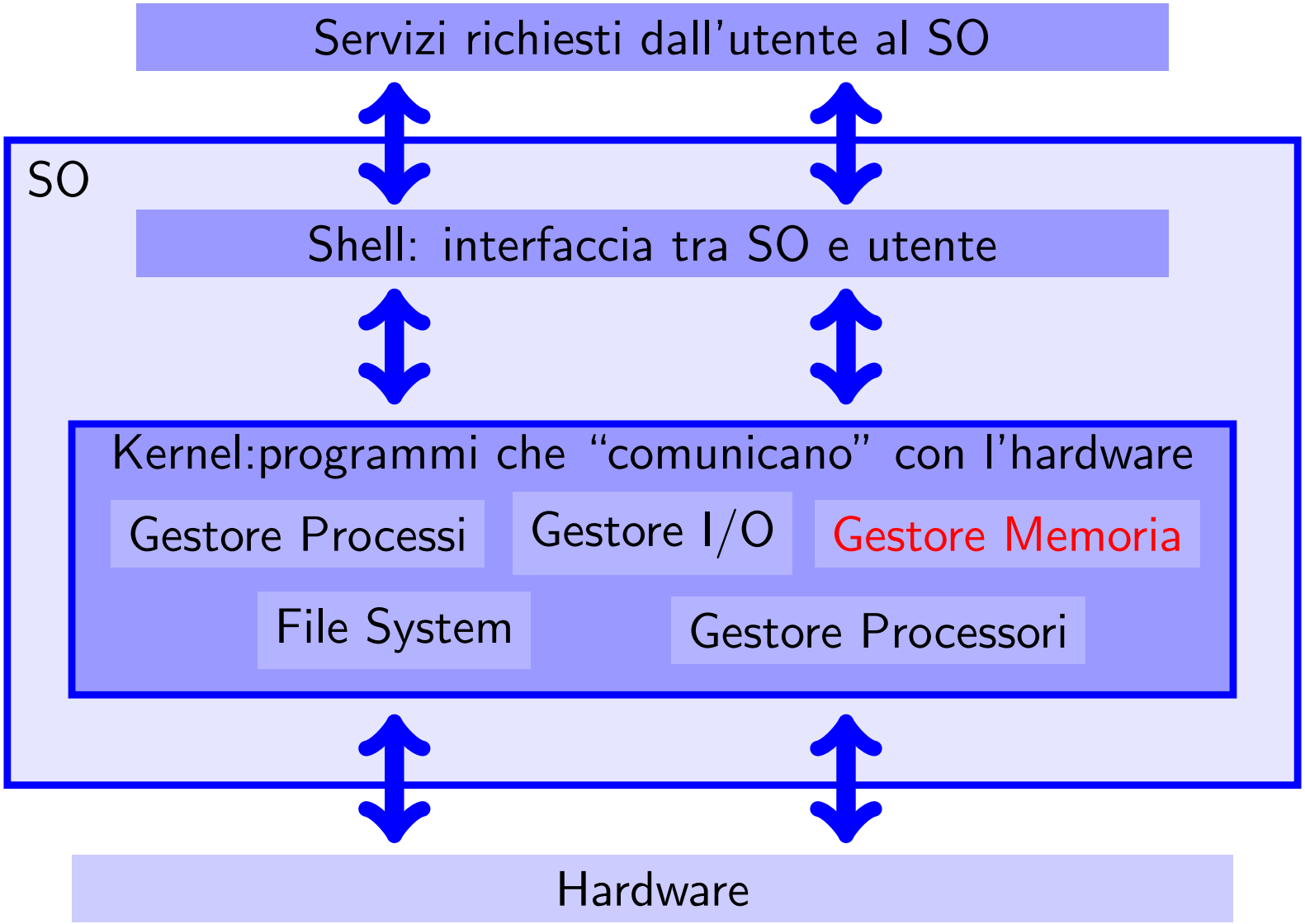


Ripartizione del tempo della CPU tra i processi

- Chi progetta un gestore dei processi, come sceglie la quantità di tempo da assegnare ad un processo (la lunghezza di ciascun rettangolo nero nel grafico)?
- Analizziamo i due casi limite:
  - se si diminuisce troppo il tempo da dedicare ad un processo (se i rettangoli neri sono molto corti) diminuisce il tempo tra una sospensione ed una riattivazione di un processo (B viene sospeso a 2 e riattivato a 9, ovvero 7 istanti di attesa) ma aumenta il tempo totale dedicato alla sospensione/ripristino dei processi (da 4 a 8 in 25 istanti)



Struttura del Sistema Operativo



Gestore della Memoria

Il gestore della memoria si occupa di:

- Tenere traccia di quali parti della memoria sono usate e da chi
- Allocare e deallocare lo spazio di memoria quando necessario
- Decidere quali processi caricare quando diventa disponibile spazio in memoria

Il gestore della memoria "realizza" una macchina virtuale in cui ciascun programma opera come se avesse a disposizione una memoria dedicata

Memoria Virtuale

- Nei SO multitasking, più programmi possono essere caricati contemporaneamente in memoria
- Spesso la memoria non è sufficiente per contenere completamente tutto il codice dei processi, ma nel modello FDE si assume che un programma risieda nella memoria principale per essere eseguito!
- Il gestore della memoria ci permette di barare realizzando una memoria "virtuale" molto più grande di quella fisica
  - teniamo, per ogni programma, solo un sottoinsieme dei suoi dati e delle sue istruzioni. Il resto le parcheggiamo nella memoria di massa (tecnicamente nella cosiddetta area di swap)
  - quando ci viene chiesto un dato o un'istruzione non in memoria, si recupera in fretta dalla memoria di massa, si mette in quella principale, e poi si invia alla CPU.