



Università degli Studi di Cagliari
Corsi di Laurea in Ingegneria Chimica e Ingegneria Meccanica

FONDAMENTI DI INFORMATICA

`http://people.unica.it/gianlucamarcialis`

A.A. 2018/2019

Docente: **Gian Luca Marcialis**

SISTEMI OPERATIVI

Sommario

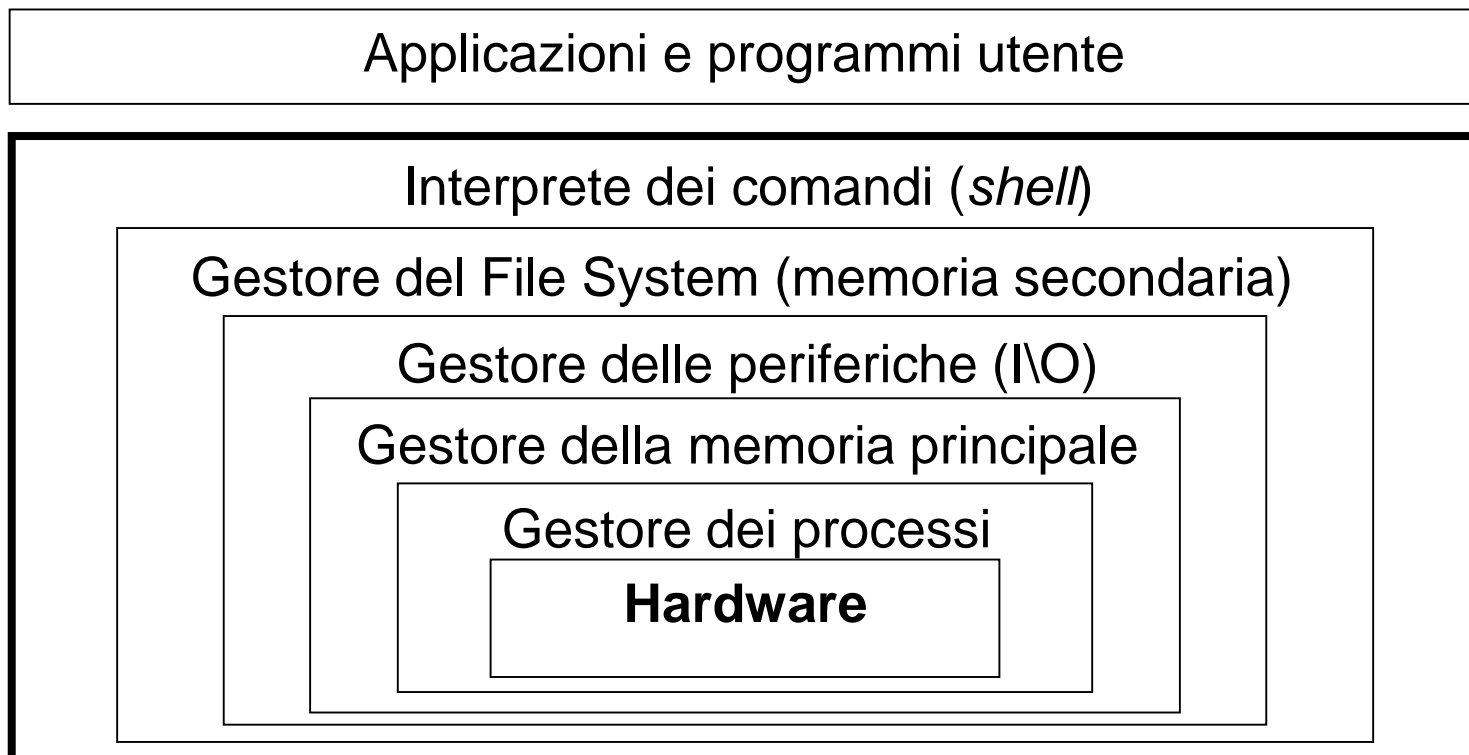
- Architettura
- Gestione dei processi
- Gestione della memoria
- Driver
- Shell dei comandi
- Interfacce

Il Sistema Operativo (SO)

- Insieme di programmi (software) con le seguenti funzioni
 - gestire le *risorse* (hardware) del calcolatore in modo **corretto, efficiente e sicuro**
 - fornire agli utenti e ai programmatori un'*interfaccia* semplificata con l'hardware
- Risorse del calcolatore
 - tutti i componenti hardware necessari per l'esecuzione dei programmi:
 - memoria centrale, processore, periferiche, ecc.
- Interfaccia utente\hardware
 - insieme di funzionalità software che "nascondono" agli utenti i dettagli dell'hardware (ad es., I/O su memoria secondaria)

Schema dell'architettura a livelli del SO

- Gli utenti possono interagire solo con la *shell* del SO
 - ad es.: lanciare un programma, visualizzare il contenuto di una directory, ecc.



Architettura a livelli del SO

- Gestore dei processi (programmi *in esecuzione*)
 - condivisione della CPU tra i vari processi (*concorrenza*)
 - gestione delle fasi di attesa per operazioni di I/O
- Gestore della memoria principale
 - allocazione della memoria tra i diversi processi
- Gestore delle periferiche
 - insieme di programmi (*driver*) che gestiscono le operazioni di I/O con le periferiche
 - Anch'esse risultano come macchine “dedicate”
 - I dettagli HW-SW sono mascherati agli utenti, che si trovano a comunicare con esse attraverso primitive ad alto livello (leggi/scrivi)

Architettura a livelli del SO (cont.)

➤ Gestore del file system

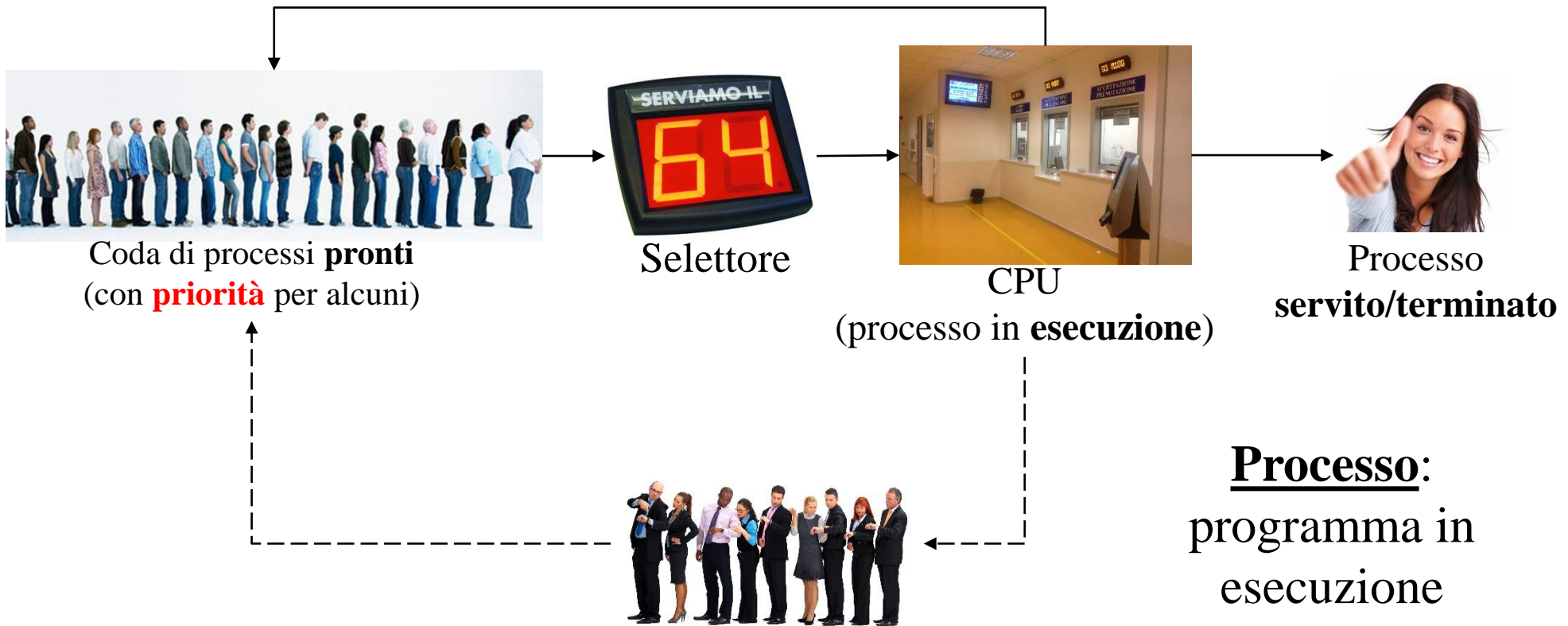
- organizza la memoria di massa (secondaria) gerarchicamente in *file* e *directory*
- gestisce le operazioni sui file (creazione, scrittura, lettura, ecc.)

➤ Interprete dei comandi (shell)

- è l'interfaccia tra il calcolatore e gli utenti (*a linea di comando* o *grafica*)
- mette a disposizione un insieme di **comandi**, consentendo ad es. di lanciare l'esecuzione di programmi, "navigare" nel file system, ecc.
- Nei Sistemi Operativi è realizzato attraverso *interfacce grafiche* (**Graphic User Interface, GUI**), con le quali si semplifica il concetto dell'operazione da svolgere
 - Es. *spostare un file nel cestino* equivale a scrivere e far eseguire da riga di comando un'istruzione del tipo "*cancella il file*"

Gestione dei processi: un esempio

Reimmissione nella lista pronti in caso di
«quanto» insufficiente



Eventuale messa in coda di **attesa** di evento interno/esterno (interrupt) prima di reimmissione nella coda dei processi pronti

Processo:
programma in
esecuzione

Gestione della memoria centrale

- Ogni processo occupa una parte della memoria centrale
- Problemi
 - la memoria principale ha dimensioni ridotte: può contenere pochi processi
 - bisogna evitare che un processo acceda a celle di memoria allocate ad altri processi (di altri utenti o del SO!)
 - il programma associato ad ogni processo deve poter essere caricato a partire da un indirizzo qualsiasi, ma i programmi al loro interno devono fare riferimento ad indirizzi precisi (es.: operandi delle istruzioni)...

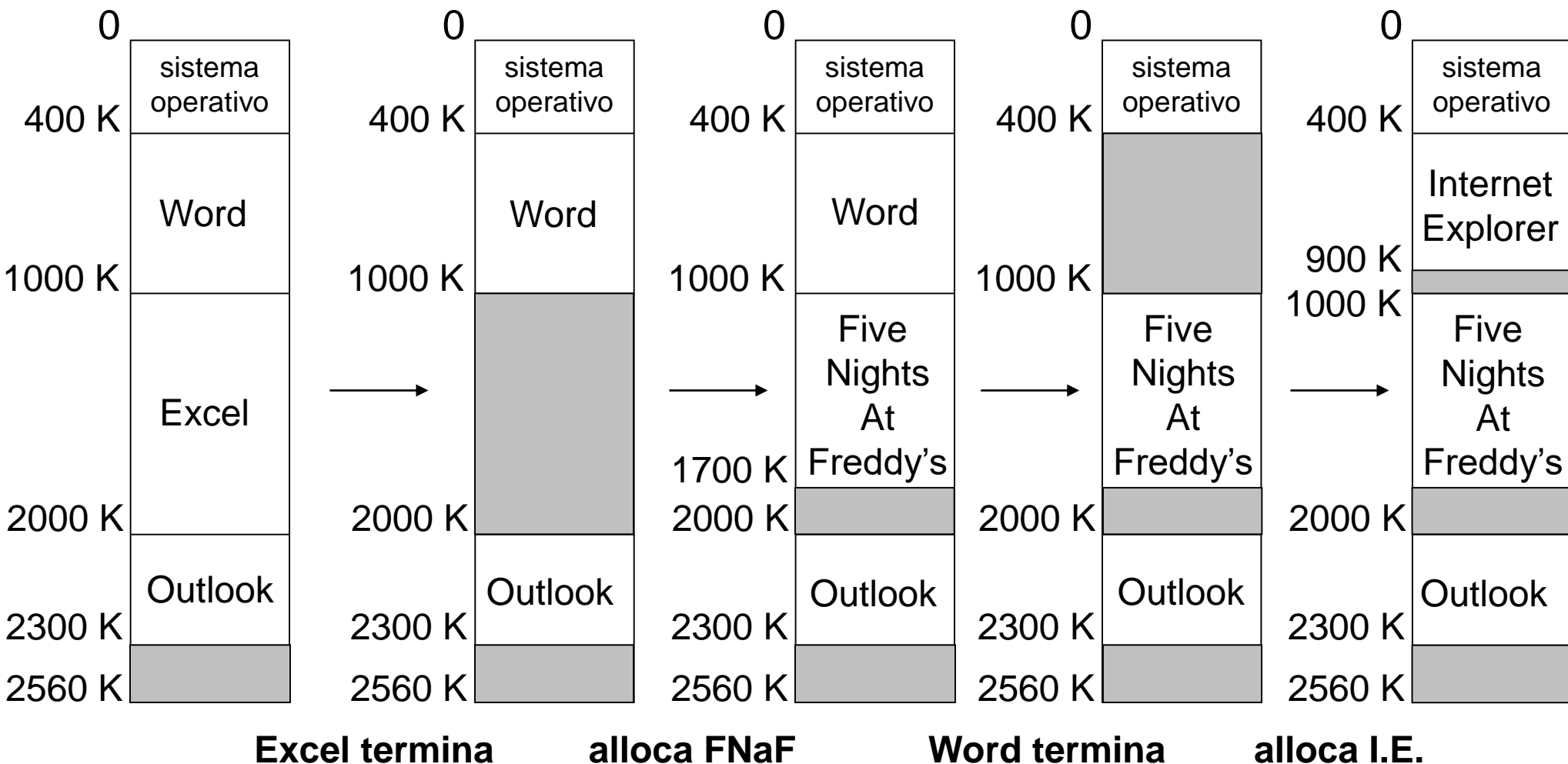
Tecniche di allocazione della memoria: allocazione contigua

- Ogni programma occupa una sequenza contigua di celle di memoria (area di memoria)
- Un nuovo programma viene allocato in una delle aree libere, se possibile
- Es.: si supponga che i primi 400 KB siano occupati dal SO, e che sia richiesta l'esecuzione di 5 processi utente, nell'ordine:

Processo	Memoria	Tempo (ore)
Word	600 KB	10
Excel	1000 KB	5
Outlook	300 KB	20
FNaF	700 KB	8
Int. Explorer	500 KB	15

(cont.)

Esempio di allocazione contigua



Problemi con allocazione contigua

- **Frammentazione:** possono esistere tante aree di memoria libere, nessuna in grado di contenere un intero processo

- *Compattamento*

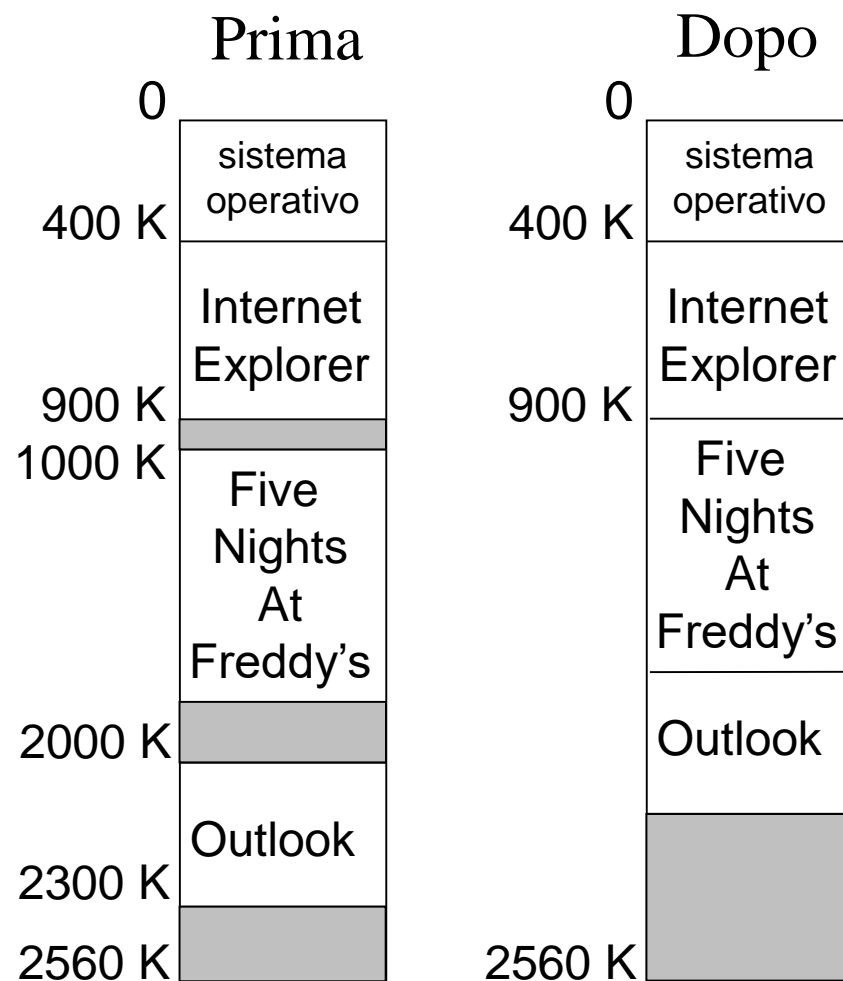
- Raggruppare le aree libere in un'unica più ampia potrebbe richiedere molto tempo

- *Deframmentazione*

- Le aree occupate vengono riallocate in modo contiguo

- **Alternativa:**

- Allocazione **non** contigua



Paginazione e segmentazione

➤ Tecniche di allocazione **non** contigua

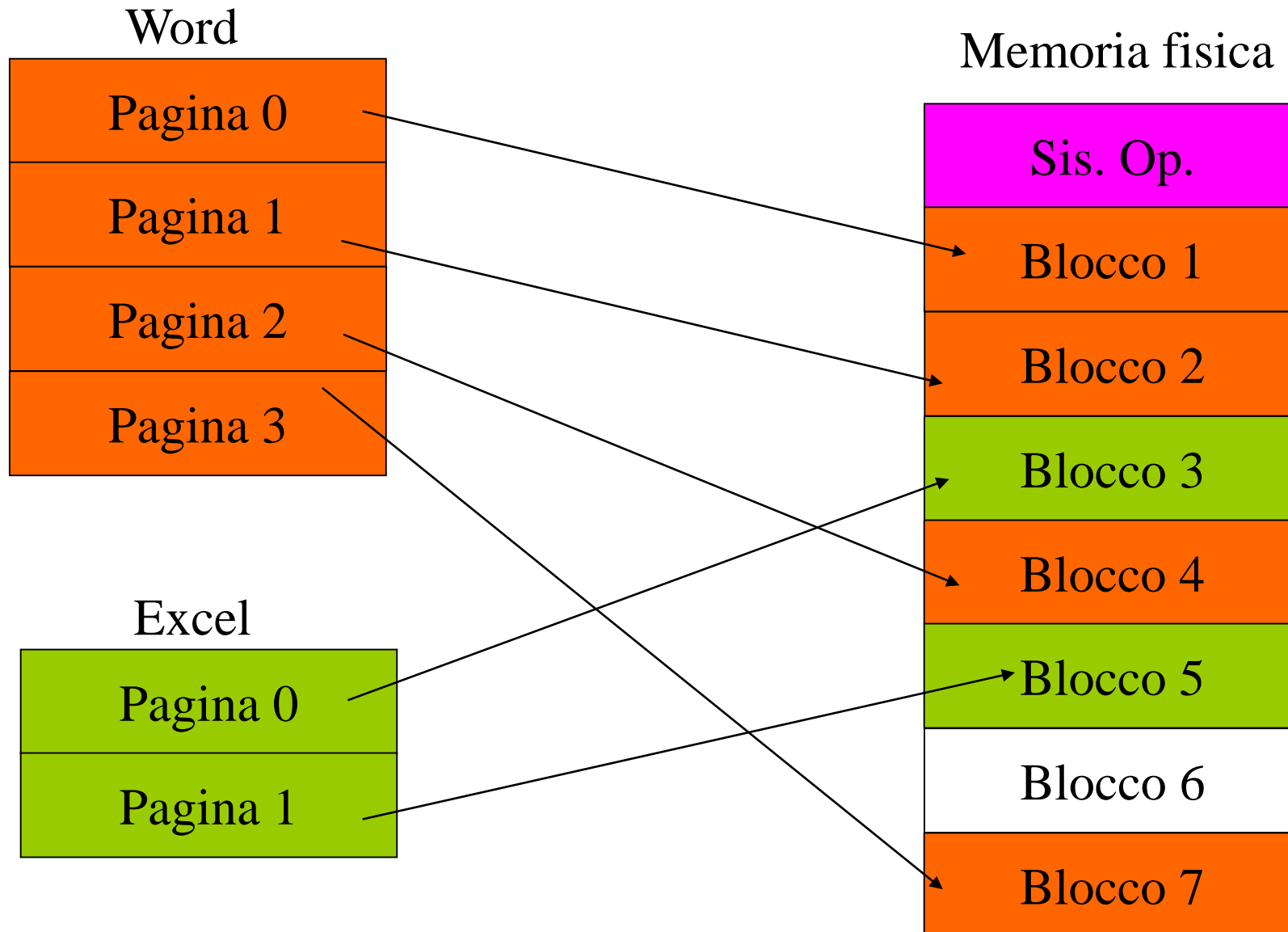
➤ **Paginazione**

- la memoria è divisa in *blocchi*
- i programmi sono divisi in *pagine*
- pagine e blocchi hanno tutti uguale dimensione
- le pagine di uno stesso programma possono occupare blocchi di memoria non contigui
- Il Sistema Operativo tiene traccia delle pagine fisiche allocate attraverso una “Page Map Table” assegnata ad ogni processo

➤ **Segmentazione**

- i programmi vengono divisi in *segmenti* di lunghezza anche diversa, che possono essere caricati in aree di memoria non contigue

Un esempio di paginazione



Gestione del file system

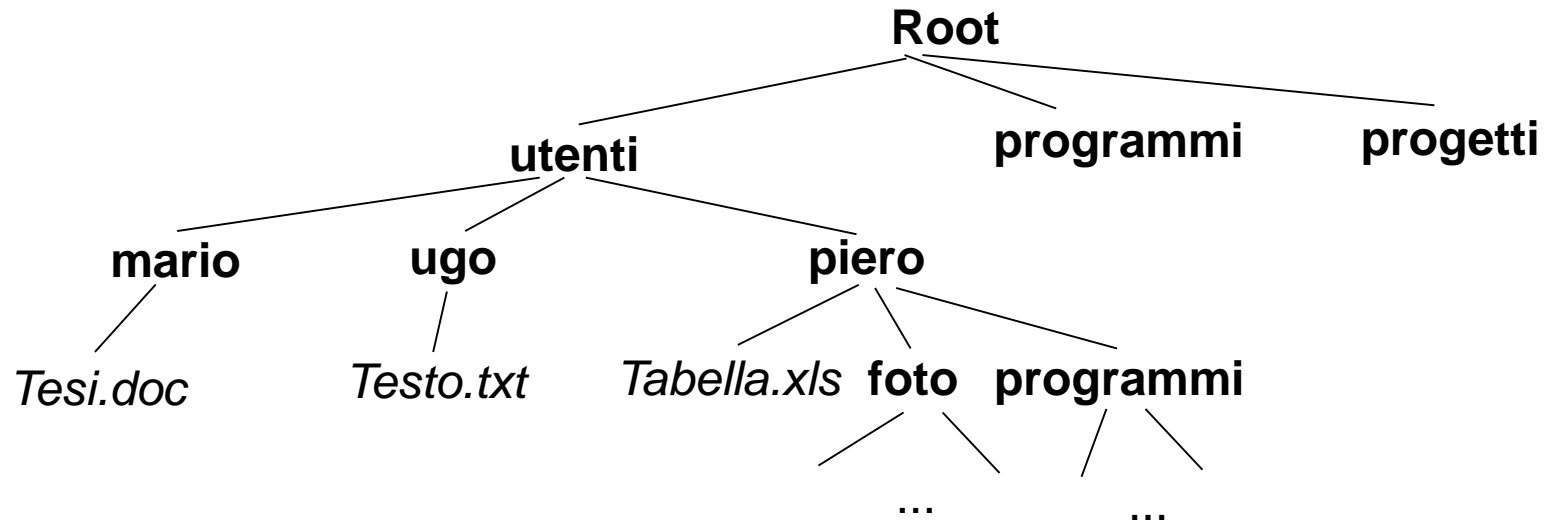
- Organizzazione di dati e programmi in memoria secondaria:
 - indipendente dal dispositivo fisico (nastri, dischi magnetici e ottici, ecc.)
 - unità di memorizzazione: **file**
 - i file sono organizzati in strutture gerarchiche ad albero: **directory**
- File
 - qualsiasi dato o programma si trova all'interno di un file
 - proprietà dei file:
 - nome (lo distingue dai file all'interno della stessa directory)
 - attributi (proprietario, permessi di accesso, tipo, dimensioni, data di creazione e modifica, ecc.)

Gestione del file system (cont.)

➤ Directory

- struttura che può contenere un numero arbitrario di file e altre directory
- la directory a livello più alto è detta **root** (radice)
- due tipi:
 - directory *di sistema*: contengono i programmi e i dati del SO
 - directory utente: contengono applicazioni e dati degli utenti
- proprietà:
 - Nome . Estensione
 - Nome: 8 caratteri; Estensione: 3 caratteri
 - Vincoli non presenti nei SO più moderni (XP, MacOSX)
 - Attributi (proprietario, permessi di accesso, ecc.)

Esempio di organizzazione del file system

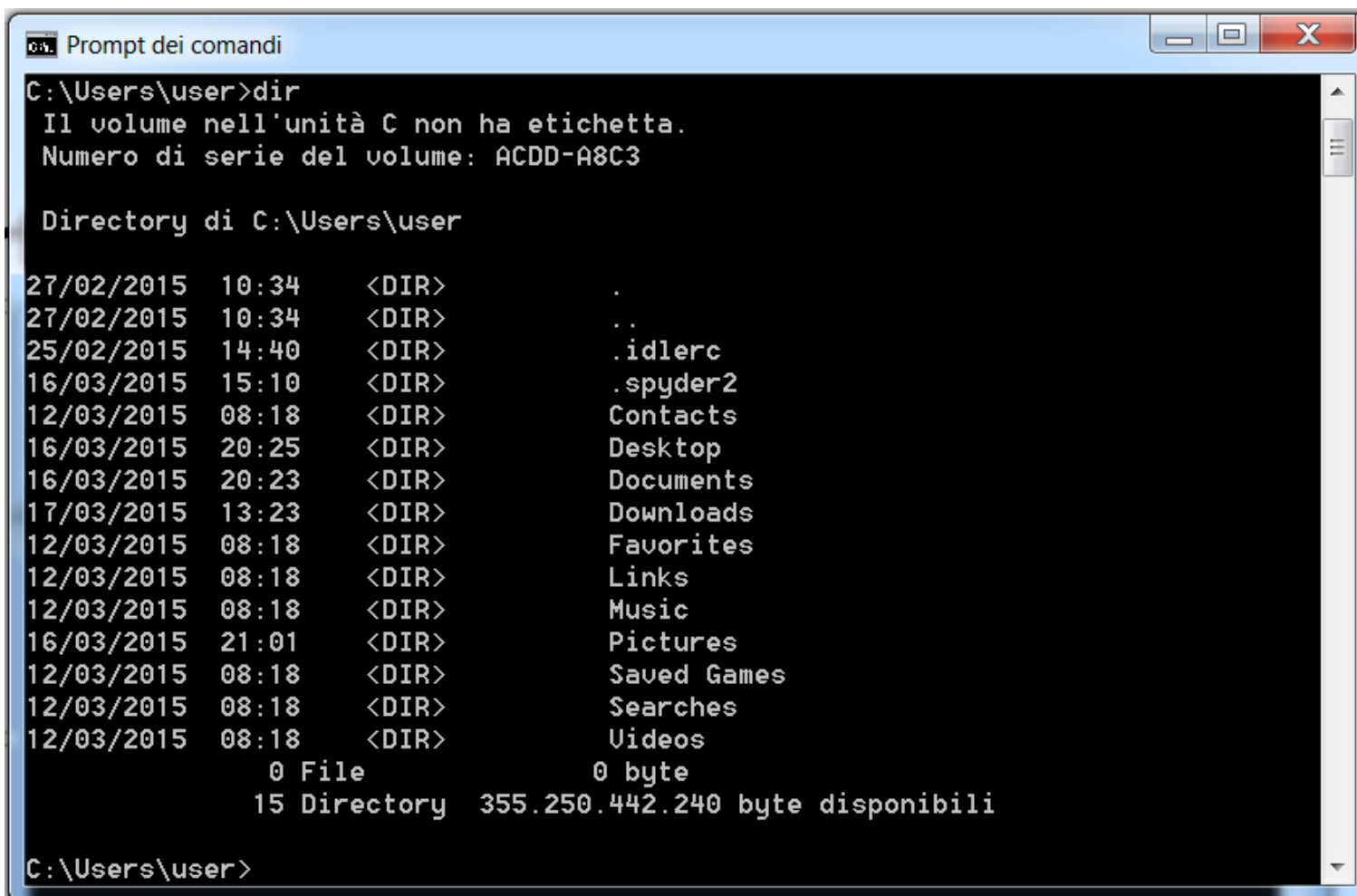


- Ogni file e directory è identificato univocamente dal **path name**, che include l'intero cammino dalla directory radice. Es. (notazione Windows):
 - \Root\utenti\ugo\Testo.txt
 - \Root\utenti\mario\Tesi.doc
- File e directory contenuti in directory diverse possono avere lo stesso nome
- In Windows la cartella "root" corrisponde alla cartella "Desktop"

Operazioni su file e directory

- Il SO mette a disposizione degli utenti (attraverso l'interprete dei comandi) e delle applicazioni (chiamate di sistema) un insieme di operazioni su file e directory
 - creazione di file e directory nella directory corrente
 - elenco del contenuto della directory corrente
 - cambiamento della directory corrente
 - copia, cancellazione e ridenominazione
 - apertura di un file (esecuzione o visualizzazione)
 - modifica dei permessi
- Ogni operazione può essere eseguita solo se si dispone dei permessi necessari

Interfacce utente-macchina: la shell



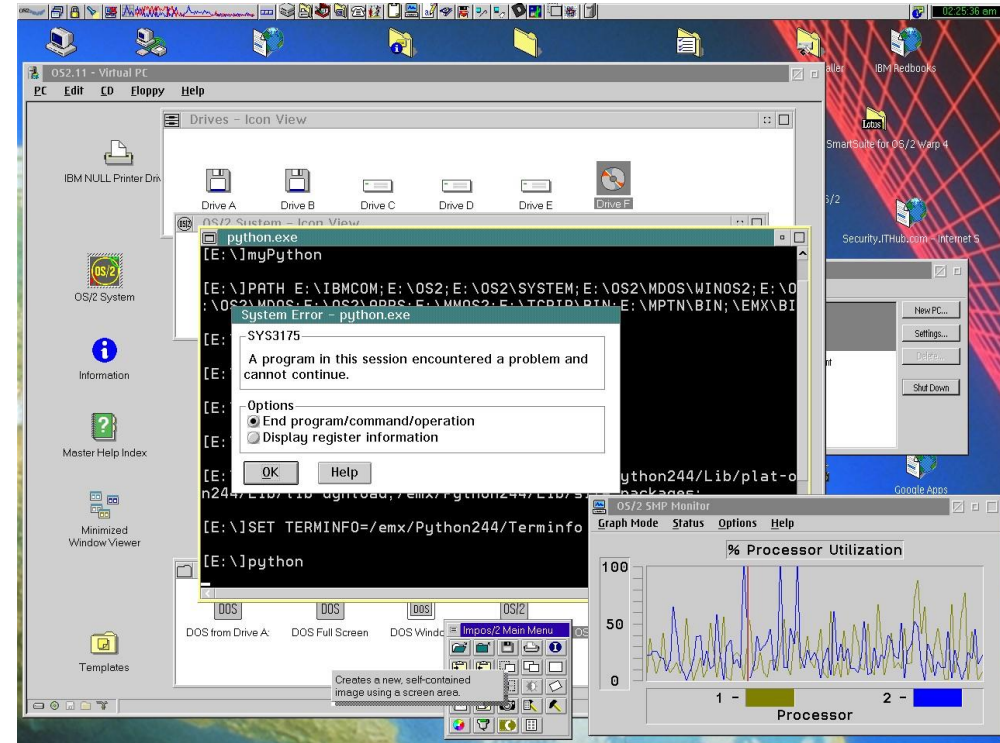
```
Prompt dei comandi
C:\Users\user>dir
Il volume nell'unit  C non ha etichetta.
Numero di serie del volume: ACDD-A8C3

Directory di C:\Users\user

27/02/2015  10:34    <DIR>          .
27/02/2015  10:34    <DIR>          ..
25/02/2015  14:40    <DIR>          .idlerc
16/03/2015  15:10    <DIR>          .spyder2
12/03/2015  08:18    <DIR>          Contacts
16/03/2015  20:25    <DIR>          Desktop
16/03/2015  20:23    <DIR>          Documents
17/03/2015  13:23    <DIR>          Downloads
12/03/2015  08:18    <DIR>          Favorites
12/03/2015  08:18    <DIR>          Links
12/03/2015  08:18    <DIR>          Music
16/03/2015  21:01    <DIR>          Pictures
12/03/2015  08:18    <DIR>          Saved Games
12/03/2015  08:18    <DIR>          Searches
12/03/2015  08:18    <DIR>          Videos
                0 File             0 byte
                15 Directory  355.250.442.240 byte disponibili

C:\Users\user>
```

Interfacce grafiche



Per saperne di più...

- Il Sistema Operativo
 - Ceri, et al., Capitolo 13