Informatica Generale

Introduzione

Scopi del corso

- Aspetti fondazionali
  - Cos'è un elaboratore
  - Cos'è un linguaggio di programmazione
  - Cos'è un algoritmo
- Aspetti pratici
  - Compilazione
  - Programmazione

Parte I: Hardware

- Codifica dell'informazione
- Architettura dei sistemi informatici
  - Struttura dell'elaboratore
  - Linguaggio macchina
- Sistemi operativi
  - Gestione dei processi e della memoria
  - Come usare un sistema operativo

Parte II:
Introduzione alla Programmazione

- Compilazione ed esecuzione
- Costrutti di programmazione
- Strutture dati semplici
- Metodologia di programmazione

Introduzione all'informatica

Cos'è l'informatica?

- Scienza della ***rappresentazione*** e dell'***elaborazione*** dell'***informazione***

ovvero

- Studio degli ***algoritmi*** che ***descrivono*** e ***trasformano*** l'informazione

Nozione di Algoritmo

- Sequenza di passi per risolvere un determinato problema
- Calcolatore = Esecutore di algoritmi
- Gli algoritmi sono descritti tramite programmi scritti in linguaggi ad ***alto livello*** e poi tradotti in ***linguaggio macchina***

Criteri di valutazione

- *Correttezza*
  - l'algoritmo risolve il problema in modo completo (spesso occorre provare la correttezza manualmente usando tecniche matematiche)
- *Efficienza*
  - lo risolve nel modo più veloce possibile (esistono criteri matematici di valutazione)

Esempio: elevamento a potenza

- *Problema*: Calcolare a elevato alla n
  - Utilizziamo le variabili N e Ris
  - Inizialmente Ris=1 e N=n
- *Algoritmo*:
  - Fino a che N>0

    Calcola Ris * a e memorizzalo in Ris

    Decrementa N
- *Correttezza*:
  - Al termine Ris=a elevato alla n

Linguaggi di Programmazione

- Scopo: *descrivere in maniera rigorosa un algoritmo*
- Classi di linguaggi:
  - Linguaggio macchina
    - Dipendono dall'hardware
  - Linguaggio ad alto livello
    - C, C++, Java, Virtual Basic

Esempio in Pseudo Pascal

```
Program potenza;
    Integer Ris, N, A;
    Read(N);Read(A);
    Ris=1;
    While (N>0) do
       Ris=Ris*A;
       N=N-1;
    Print(Ris);
```

Esempio

- Il precedente programma va tradotto in linguaggio macchina (comprensibile all'elaboratore) cioè viene compilato in sequenze di istruzioni
- Quando le istruzioni vengono eseguite il programma prende dati in ingresso (valori iniziali di N e A) attraverso la tastiera (*input*) e poi stampa il risultato sul video (valore finale di Ris) (*output*)
- In generale un programma può essere visto infatti come una funzione da *input* ad *output*.

Utilizzo di un elaboratore

- Come utente:
  - Uso software applicativo esistente per creare documenti e interfacce grafiche, effettuare calcoli, navigare in rete
- Come sviluppatore:
  - Creo nuovi programmi sullo strato del software esistente
    - Nuovi programmi applicativi
    - Nuovi programmi di sistema (cioè che fanno funzionae il calcolatore)

Parte I: Hardware

Architettura dei Sistemi Informatici

- *Sistemi informatici* PC, terminali e reti
- *Architettura* insieme delle componenti del sistema, descrizione delle loro funzionalità e della loro interazione
- Suddivisione principale *hardware e software*

Hardware

- Unità di Elaborazione (Processore o CPU):
  - Svolge le elaborazioni
  - Coordina il trasferimento dei dati
  - Cioè esegue i programmi
- Memoria Centrale
  - Memorizza dati e programmi per l'elaborazione
  - Volatile
  - Accesso rapido
  - Capacità limitata

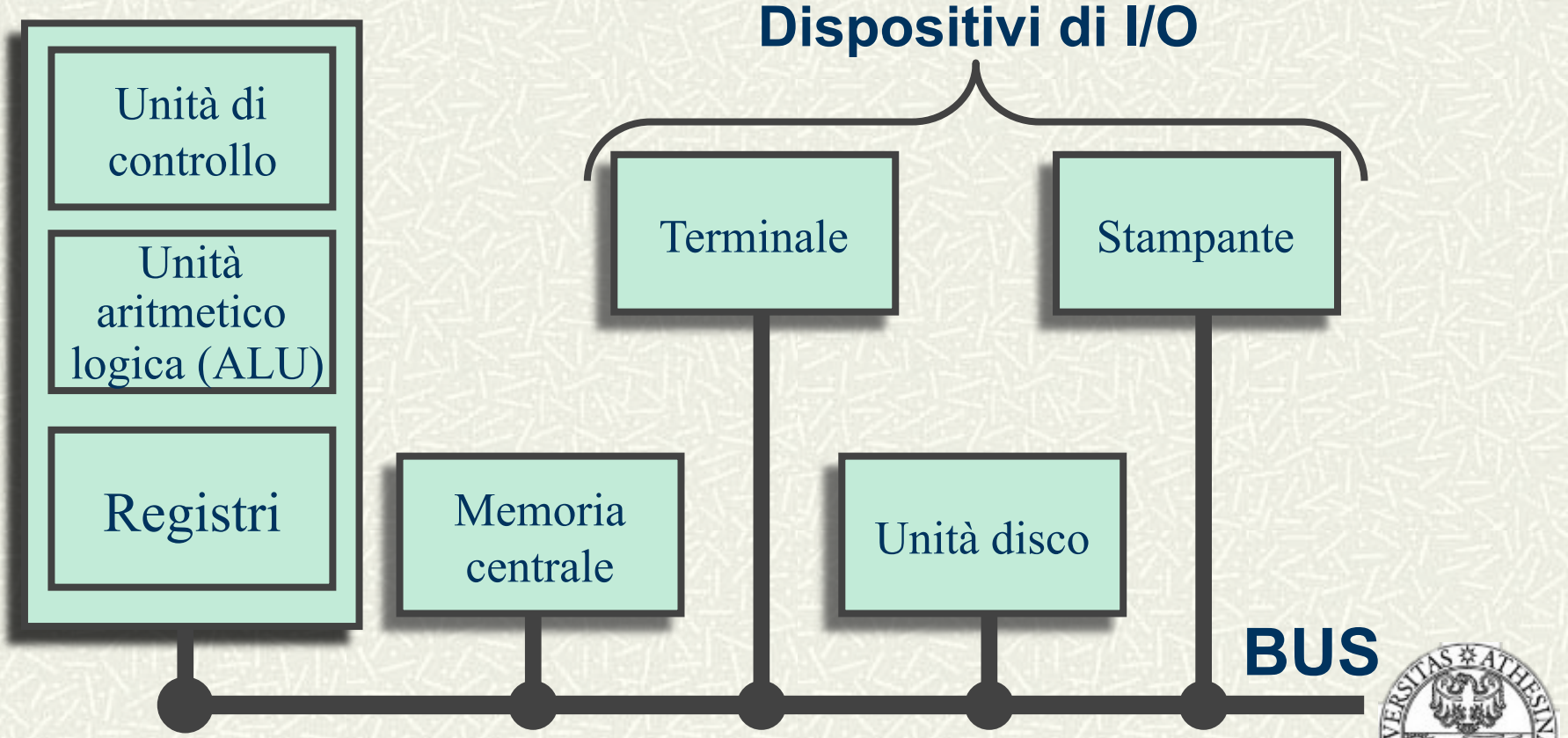
Hardware

- Memoria Secondaria (DVD, harddisk, floppy)
  - Grande capacità
  - Persistente
  - Accesso piu lento della RAM
- Unità Periferiche
  - Interfaccia verso l'esterno
  - Terminali (tastiera, video)
  - Stampanti

Hardware

- Bus di Sistema
  - Collega le altre componenti
    - RAM
    - Memorie Secondarie
    - Periferiche
  - Insieme di collegamenti di vario tipo

Organizzazione di un calcolatore "bus oriented"



Esempi: Personal Computer (PC)

- *Contenitore* con
  - CPU, RAM
  - Memoria Centrale
    - Fisso
    - Unità per Dischetti/CD
- Monitor
- Tastiera

Alcuni accessori per PC

- (Lettore Floppy), CD, DVD
- Chiavette USB
- Modem
- Mouse
- Stampante
- Scanner
- Joystick






Altri Sistemi Informatici

- *Workstation*
  - Calcolatore con elevate prestazioni
- *Server*
  - *Calcolatore* con elevate prestazioni che offre servizi
- *Main-frame*
  - Grandi Server (per reti di terminali con centinaia di utenti)
- *Notebook/laptop, smartphone e palmari*
  - Elaboratori portatili



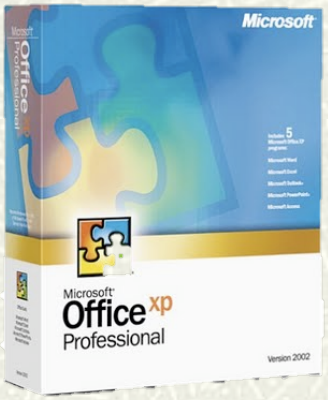
Altri Sistemi Informatici

- Reti di Calcolatori
  - *Reti Locali*
    - collegano terminali vicini tra loro (ad es. il nostro laboratorio)
  - *Reti Geografiche*
    - collegano dei calcolatori a medio-grandi distanze (ad es. Internet)

Software

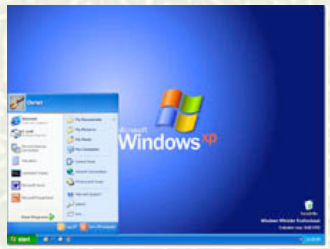
- *Software di base:*
  - Dedicato alla gestione dell'elaboratore
  - Esempio: **sistema operativo**

- *Software applicativo*:
  - Dedicato alla realizzazione di specifiche applicative
  - Esempio:
    - programmi per scrittura,
    - gestione aziendale,
    - navigazione su internet, ...



Sistema Operativo

- Rende la componente hardware facile da usare
- Fornisce funzionalità ad alto livello agli utenti
- Ad esempio:
  - organizza la memoria di massa
  - gestisce comandi immessi dall'utente:
    - Esegui un programma! Mostra i dati su video!



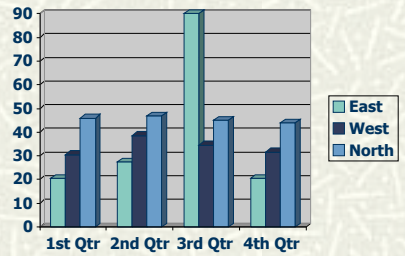
- Se il sistema è multi-utente deve gestire le risorse disponibili cercando di soddisfare tutti gli utenti
- Esempi: MS DOS, OS X, Windows, Unix, Linux

Software Applicativo

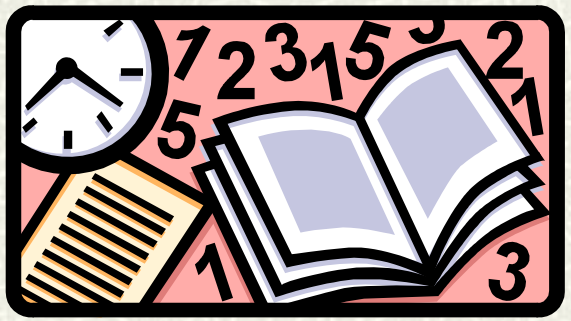
- Video Scrittura
  - per costruire e testi e definire formati di stampa
- Agende elettroniche
  - indirizzario, calendari
- Posta Elettronica
  - per comunicazione
- Fogli elettronici
  - per elaborazioni contabili
- Database
  - sistemi per la gestione di dati

Applicazioni

- Calcolo Numerico: statistiche, ecc
- Gestione Aziendale: banche, assicurazioni,
- Telematica: bancomat, ecc
- Automazione industriale:, robotica, ecc
- Internet: commercio virtuale, ecc



Rappresentazione della Informazione



Codifica dell'informazione

- Il calcolatore memorizza ed elabora vari tipi di informazioni
  - Numeri, testi, immagini, suoni
- Occorre rappresentare tale informazione in formato facilmente manipolabile dall'elaboratore
- Si utilizza una rappresentazione digitale

Codifica digitale

- L'unità minimale di rappresentazione è il ***bit*** (binary digit – cifra digitale): **0** o **1**
- Informazioni complesse si memorizzano come sequenze di bit
- Una sequenza di **8 bit** viene chiamata ***Byte***
  - 0 0 0 0 0 0 0 0
  - 0 0 0 0 0 0 0 1
  - .....

Codifica dell'informazione

- Per codificare in nomi delle province liguri mi bastano 2 bit
- Ad esempio:
  - 0 0 per rappresentare Genova
  - 0 1 per rappresentare La Spezia
  - 1 0 per rappresentare Imperia
  - 1 1 per rappresentare Savona

- In generale su **N** bit si possono codificare $\mathbf{2^N}$ informazioni (tutte le possibili combinazioni di 0 e 1 su N posizioni)

- Con un byte si possono codificare quindi $\mathbf{2^8 = 256}$ possibili informazioni

Altre unità di misura

- **KiloByte (KB), MegaByte (MB), GigaByte (GB)**

- Per ragioni storiche in informatica Kilo, Mega, e Giga indicano però le ***potenze di 2*** che più si avvicinano alle corrispondenti potenze di 10
- Più precisamente
  - 1 KB = 1024 x 1 byte = $2^{10} \sim 10^3$ byte
  - 1 MB = 1024 x 1 KB = $2^{20} \sim 10^6$ byte
  - 1 GB = 1024 x 1 MB = $2^{30} \sim 10^9$ byte
  - ...
- I multipli del byte vengono utilizzati come unità di misura per la capacità della memoria di un elaboratore

La Codifica dei Caratteri

A B ... a b .... & % $ ...

Codici per i simboli dell'alfabeto

- Per rappresentare i simboli dell'alfabeto anglosassone (0 1 2 ... A B ... A b ...) bastano 7 bit
  - Nota: *B* e *b* sono simboli diversi
- Per l'alfabeto esteso con simboli quali &, %, $, ... bastano 8 bit come nella codifica accettata universalmente chiamata ASCII
- Per manipolare un numero maggiore di simboli la Microsoft ha introdotto la codifica UNICODE a 32 bit (2^{32} caratteri)

Codifica ASCII

- La codifica ASCII (American Standard Code for Information Interchange) utilizza codici su 8 bit
- Ad esempio
  - 0 1 0 0 0 0 0 1 rappresenta A
  - 0 1 0 0 0 0 1 0 rappresenta B
  - 0 1 0 0 0 0 1 1 rappresenta C

- Le parole si codificano utilizzando sequenze di byte
  - 01000010 01000001 01000010 01000001
    B A B A

Tabella ASCII

Dec	Hx	Chr	Dec	Hx	Chr	Dec	Hx	Chr	Dec	Hx	Chr	Dec	Hx	Chr	Dec	Hx	Chr
32	20	**SPACE**	48	30	**0**	64	40	**@**	80	50	**P**	96	60	**`**	112	70	**p**
33	21	**!**	49	31	**1**	65	41	**A**	81	51	**Q**	97	61	**a**	113	71	**q**
34	22	**”**	50	32	**2**	66	42	**B**	82	52	**R**	98	62	**b**	114	72	**r**
35	23	**#**	51	33	**3**	67	43	**C**	83	53	**S**	99	63	**c**	115	73	**s**
36	24	**$**	52	34	**4**	68	44	**D**	84	54	**T**	100	64	**d**	116	74	**t**
37	25	**%**	53	35	**5**	69	45	**E**	85	55	**U**	101	65	**e**	117	75	**u**
38	26	**&**	54	36	**6**	70	46	**F**	86	56	**V**	102	66	**f**	118	76	**v**
39	27	**’**	55	37	**7**	71	47	**G**	87	57	**W**	103	67	**g**	119	77	**w**
40	28	**(**	56	38	**8**	72	48	**H**	88	58	**X**	104	68	**h**	120	78	**x**
41	29	**)**	57	39	**9**	73	49	**I**	89	59	**Y**	105	69	**i**	121	79	**y**
42	2A	*	58	3A	**:**	74	4A	**J**	90	5A	**Z**	106	6A	**j**	122	7A	**z**
43	2B	**+**	59	3B	**;**	75	4B	**K**	91	5B	**[**	107	6B	**k**	123	7B	**{**
44	2C	**,**	60	3C	**<**	76	4C	**L**	92	5C	****	108	6C	**l**	124	7C	**|**
45	2D	**-**	61	3D	**=**	77	4D	**M**	93	5D	**]**	109	6D	**m**	125	7D	**}**
46	2E	**.**	62	3E	**>**	78	4E	**N**	94	5E	**^**	110	6E	**n**	126	7E	**~**
47	2F	**/**	63	3F	**?**	79	4F	**O**	95	5F	**_**	111	6F	**o**	127	7F	**DEL**

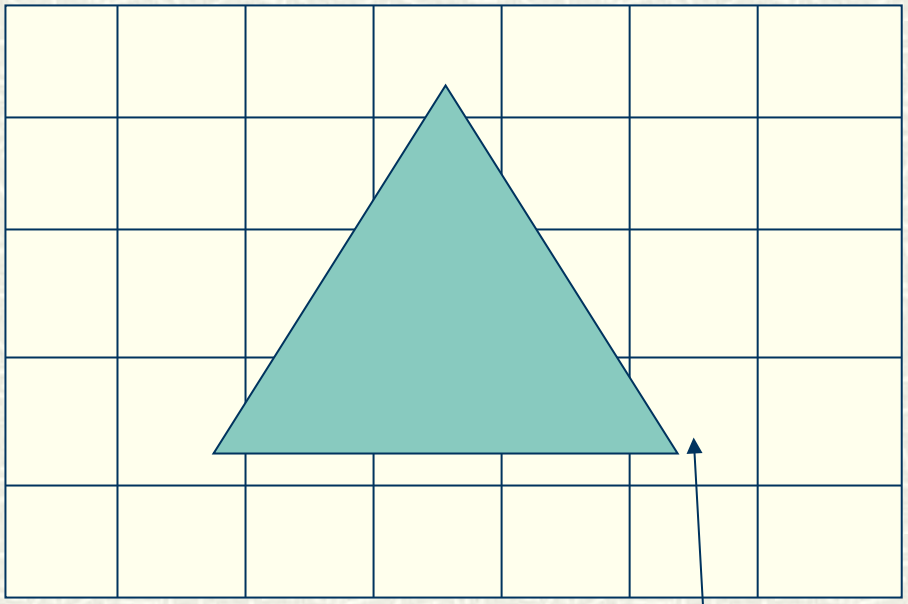
Nota: il valore numerico di una cifra può essere calcolato come differenza del suo codice ASCII rispetto al codice ASCII della cifra 0 (es. ‘5’-‘0’ = 53-48 = 5)

Codifica di immagini

Pixel – Picture element

- Le immagini vengono scomposte in griglie
- Le caselle di una griglia vengono chiamate *pixel*
- La risoluzione indica il numero di pixel in cui è suddivisa un'immagine
  - Risoluzione tipica di uno schermo video 800 x 600, 1024 x 768, 1280 x 800, 1440 x 900

Codifica di un'immagine



Pixel = 1

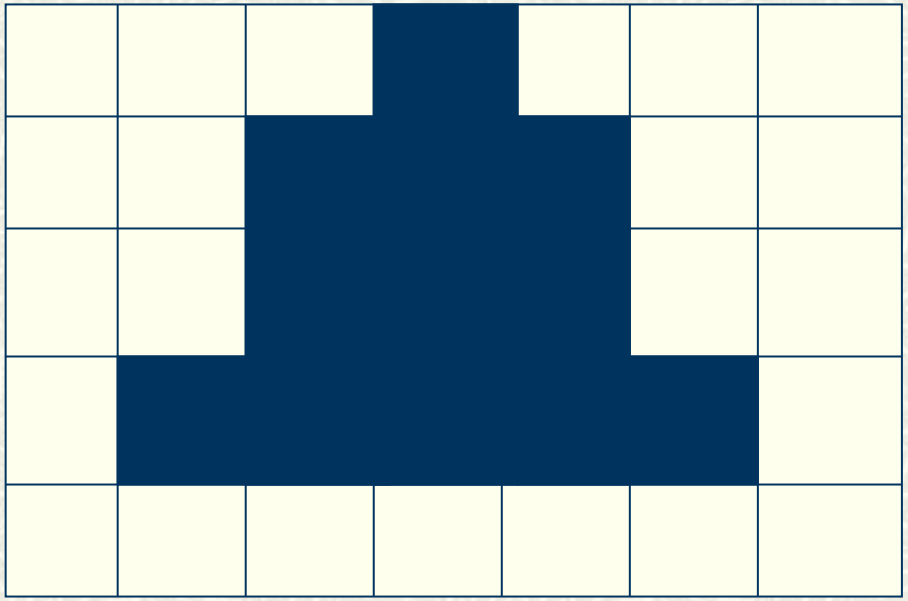
0 0 0 1 0 0 0
0 0 1 1 1 0 0
0 0 1 1 1 0 0
0 1 1 1 1 1 0
0 0 0 0 0 0 0

codifica

Decodifica

0 0 0 1 0 0 0
0 0 1 1 1 0 0
0 0 1 1 1 0 0
0 1 1 1 1 1 0
0 0 0 0 0 0 0

Codifica



Immagine

Immagini in toni di grigio

- Se si assegna un solo bit a ogni pixel si rappresentano immagini in bianco e nero
  - 0 = bianco 1 = nero
- Per poter rappresentare immagini più complesse
  - si codificano i toni di grigio
  - Si associa una codifica di un tono di grigio ad ogni pixel

Immagini a colori

- Nella codifica RGB si utilizzano tre colori rosso
  - (Red), verde (Green) e blu (Blue):
- Ad ogni colore si associa un certo numero di sfumature codificate su N bit (2^N possibili sfumature)
- Ad esempio
  - se si utilizzano 2 bit per colore si ottengono 4 sfumature per colore
  - ogni pixel ha un codice di 6 bit
- Con 8 bit si ottengono 256 sfumature e 256^3 (16 milioni) possibili colori

Bitmap

- La rappresentazione di un'immagine mediante la codifica a pixel viene chiamata *bitmap*
- Il numero di byte richiesti per memorizzare una bitmap dipende dalla risoluzione e dal numero di colori
- Es. se la risoluzione è 640x480 con 256 colori occorrono 2.457.600 bit = 300 KB
- I formati bitmap più consciuti sono BITMAP (.bmp), GIF (.gif), JPEG (.jpg)
- In tali formati si utilizzano metodi di *compressione* per ridurre lo spazio di memorizzazione

Rappresentazione dei suoni

- Si effettuano dei campionamenti su dati analogici
- Si rappresentano i valori campionati con valori digitali
- La frequenza del campionamento determina la fedeltà della riproduzione del suono

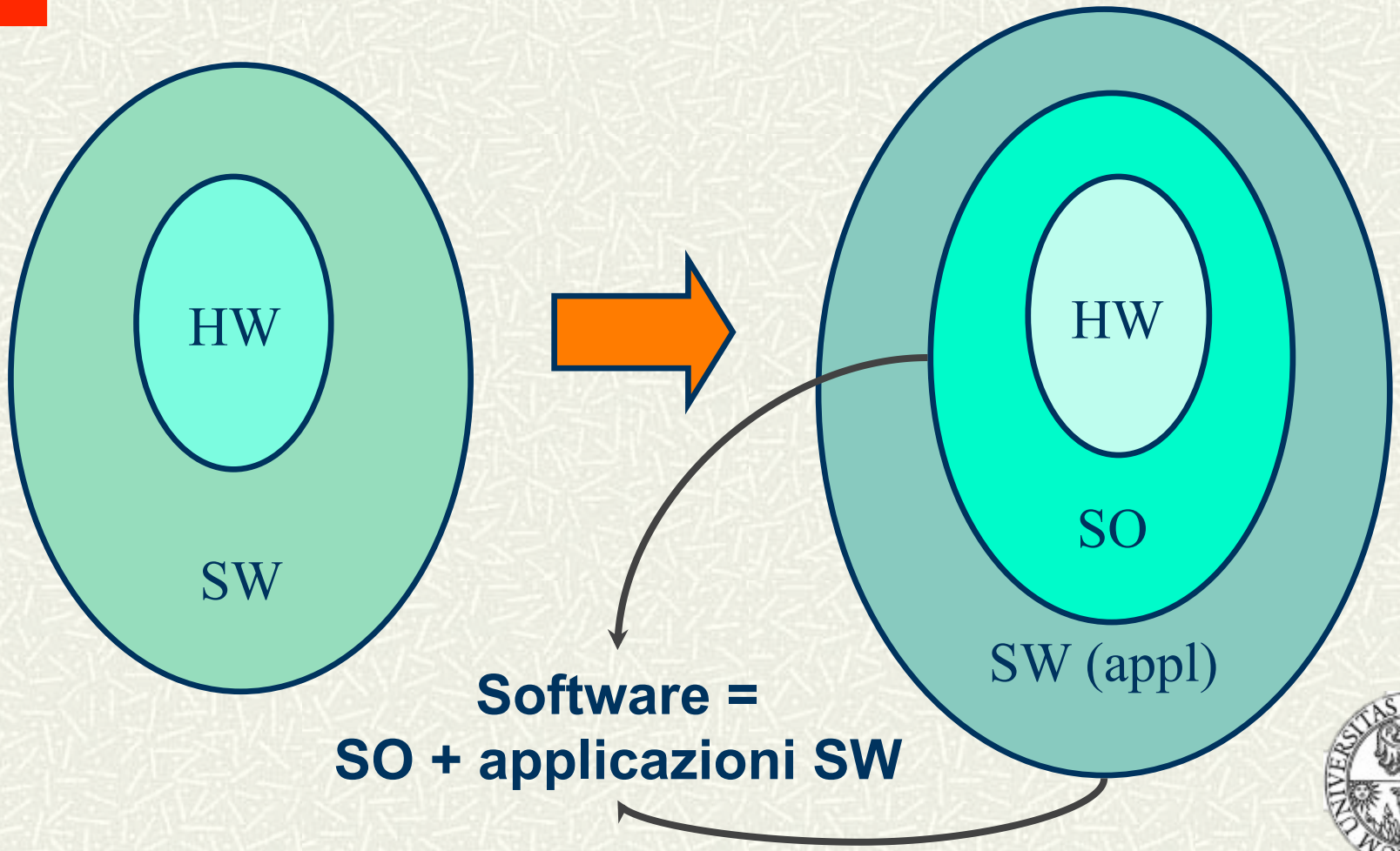
I Sistemi Operativi (cenni)

- I sistemi operativi permettono di **gestire le risorse** efficientemente
  - tengono traccia di chi accede alle risorse
  - accettano e soddisfano le richieste di uso di risorse
  - risolvono i conflitti tra più risorse
- Possono essere visti come una **macchina di calcolo estesa**
  - rappresentano la base su cui è possibile scrivere programmi applicativi in modo più semplice che utilizzando direttamente l'HW.

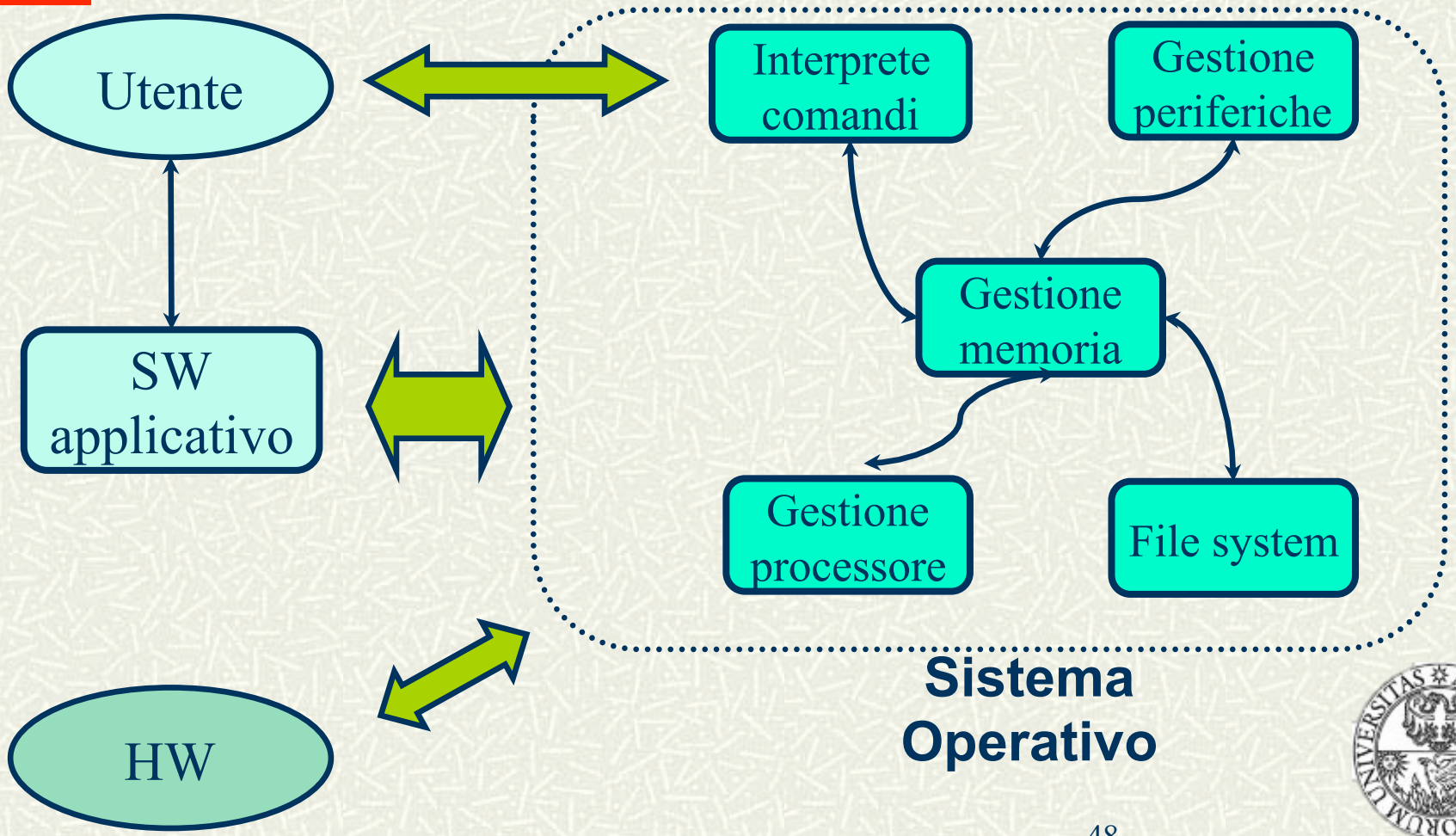
Vantaggi

- I sistemi operativi permettono definire uno standard per interfacciare i dispositivi fisici, per cui:
  - lo sviluppo dei programmi risulti più semplice ed indipendente dal calcolatore che si utilizza
  - l'aggiornamento del SW di base e dell'HW sia trasparente all'utente ed alle applicazioni.

Il SO come intermediario tra HW e SW



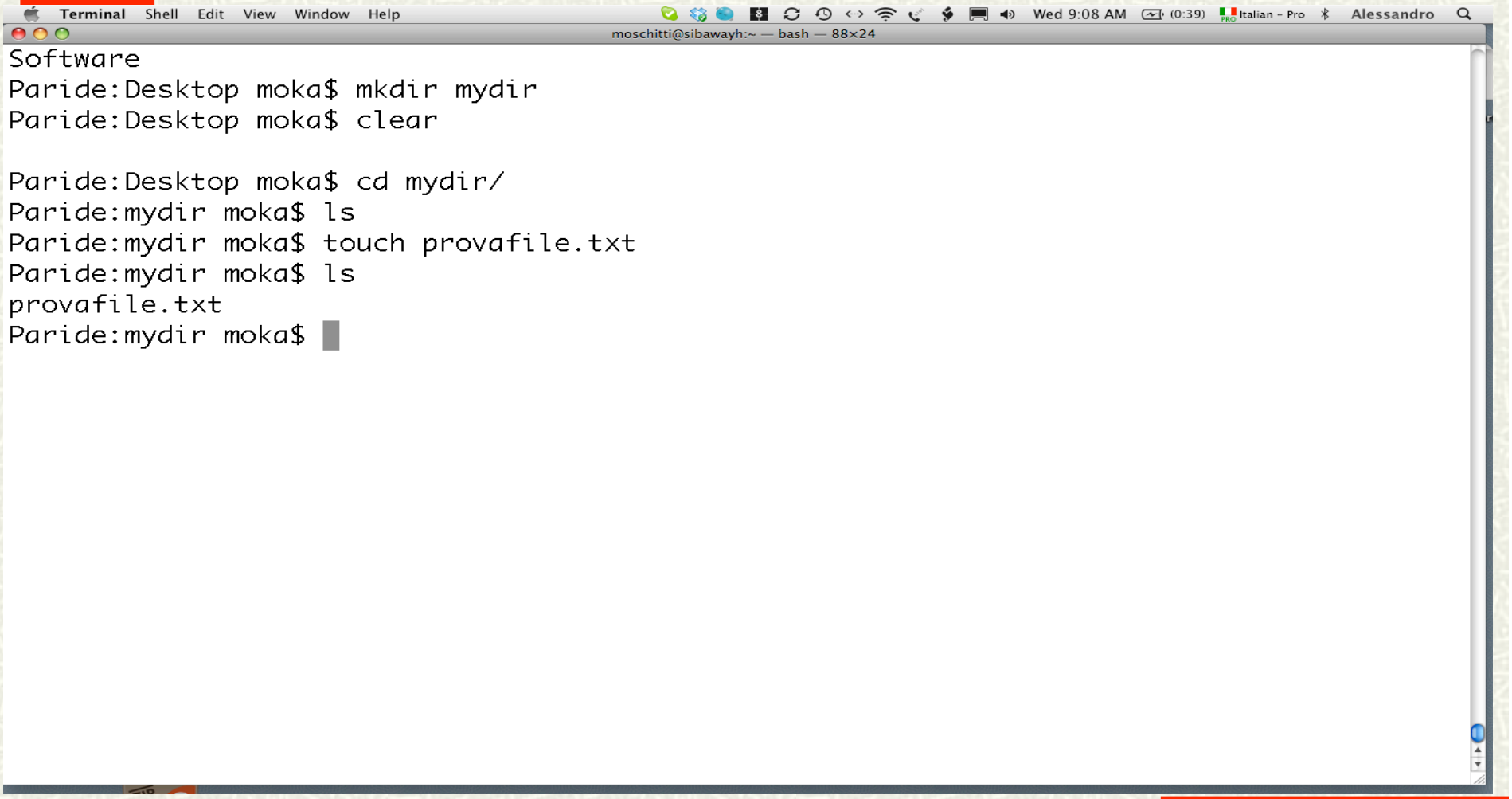
Composizione di un SO



Il file system e shell di Unix

Caratteristiche generali dei FS comunemente usati da Unix/Linux

Shell (linux/mac osx/unix)



Il file system di Unix

- Il file system è la parte del SO che si occupa di mantenere i dati/programmi in modo persistente
- Astrazioni fornite :
  - *File* : unità di informazione memorizzata in modo persistente
  - *Directory* : astrazione che permette di raggruppare assieme più file

I file di Unix

- Tipi di file Unix :
  - *regular* (-): collezione di byte non strutturata
  - *directory* (d) : directory
  - … e altri più complessi

Attributi di un file Unix

- File = nome + dati + attributi
- Alcuni attributi dei file unix :
  - es. `ls -l pippo.c`

```
-rw-r--r-- 1 susanna users 1064  Feb 6 2002 pippo.c
```

Tipo del file
(regolare, -)

Attributi di un file Unix (2)

- File = nome + dati + attributi
- Alcuni attributi dei file unix :
  - es. `ls -l pippo.c`

```
-rw-r--r--  1 susanna users 1064  Feb 6 2002 pippo.c
```

Protezione
r - permesso di lettura (directory, listing)
w- permesso di scrittura (directory, aggiungere file)
x - permesso di esecuzione (directory, accesso)

Attributi di un file Unix (3)

- File = nome + dati + attributi
- Alcuni attributi dei file unix :
  - es. `ls -l pippo.c`

```
-rw-r--r-- 1 susanna users 1064 Feb 6 2002 pippo.c
```

Proprietario del file

Gruppo

Data ultima modifica

Attributi di un file Unix (4)

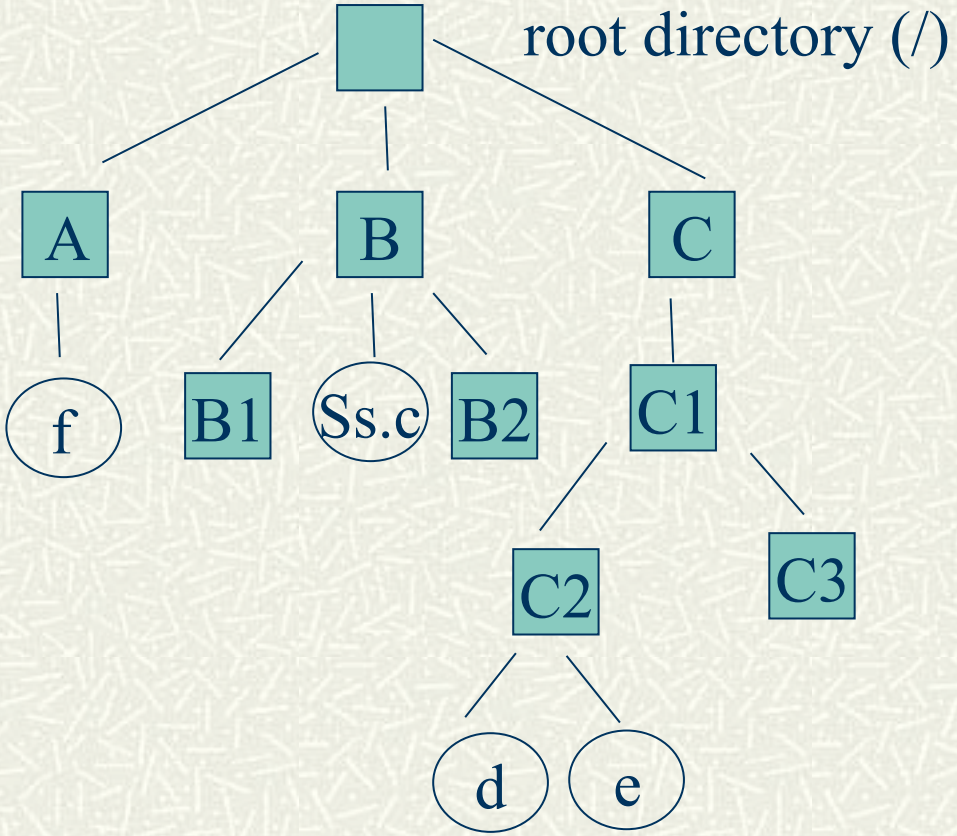
- File = nome + dati + attributi
- Alcuni attributi dei file unix :
  - es. `ls -l pippo.c`

```
-rw-r--r-- 1 susanna users 1064 Feb 6 2002 pippo.c
```

Numero di blocchi su disco utilizzati

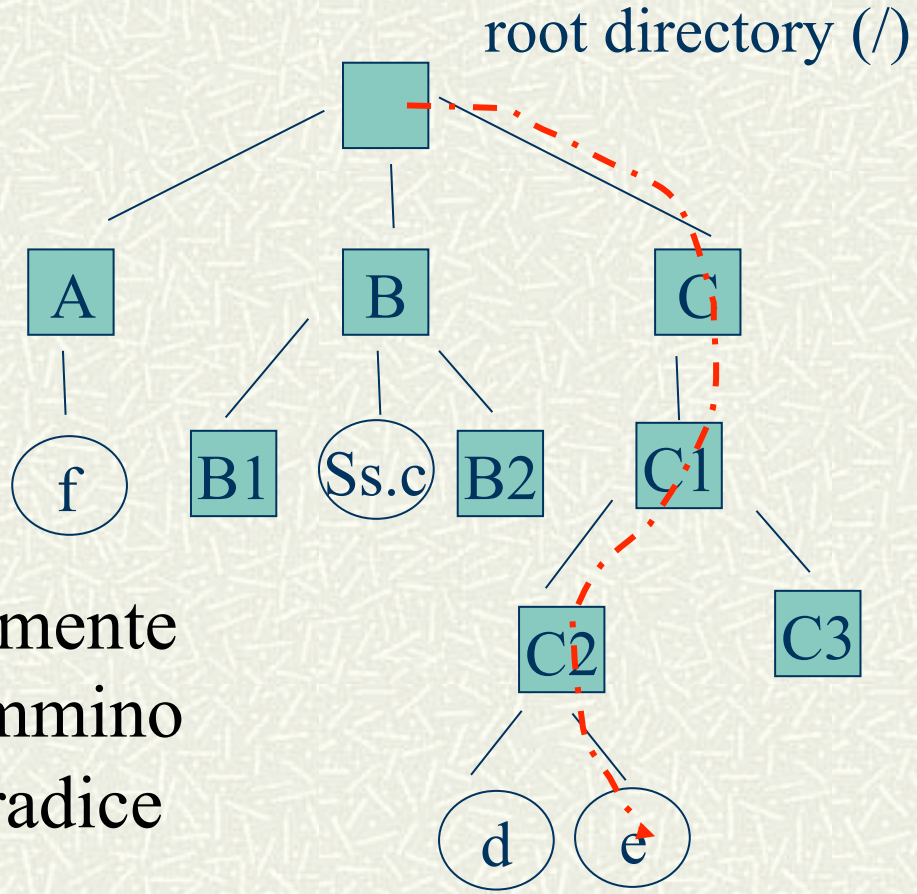
Lunghezza in byte del file

Il FS di Unix è gerarchico



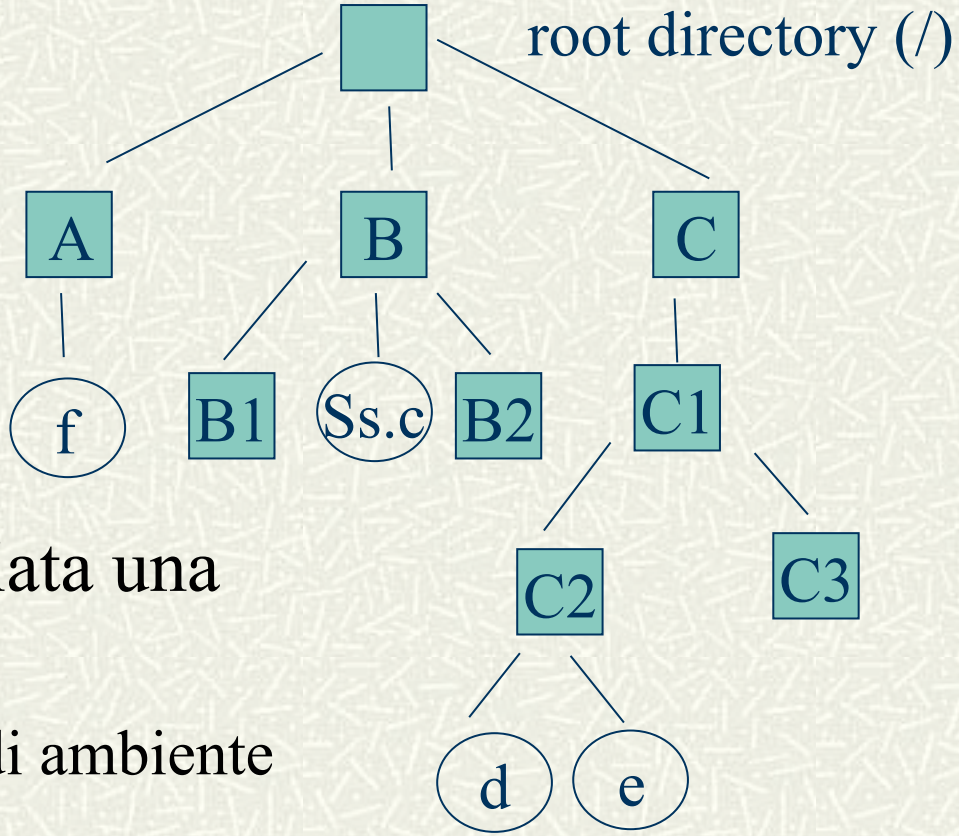
- Esempio di FS

Path name assoluto



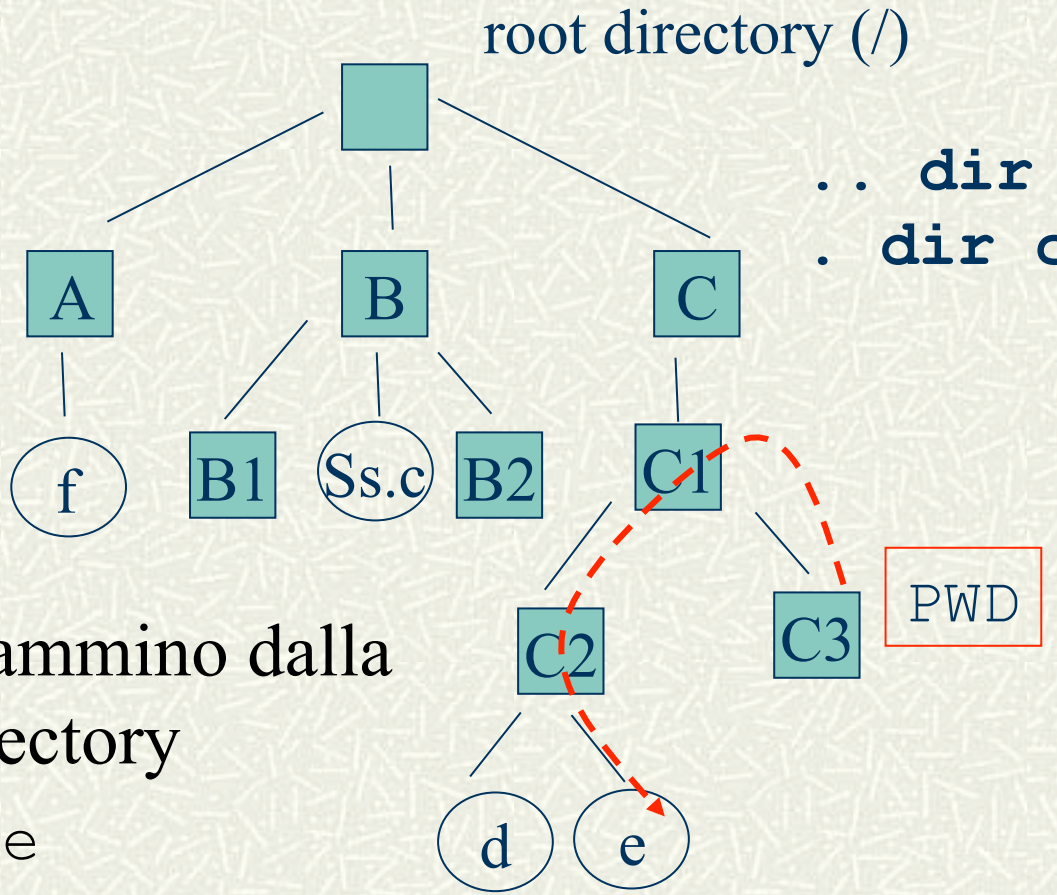
- Ogni file è univocamente determinato dal cammino che lo collega alla radice
  - `/C/C1/C2/e`

Path name relativo



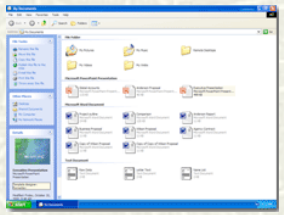
- Ogni shell ha associata una *working directory*
  - è indicata nella var di ambiente `PWD`
  - si cambia con `cd`

Path name relativo (2)



- Il PNR è il cammino dalla Working Directory
  - `./../C2/e`
  - (il '.' iniziale si può omettere)

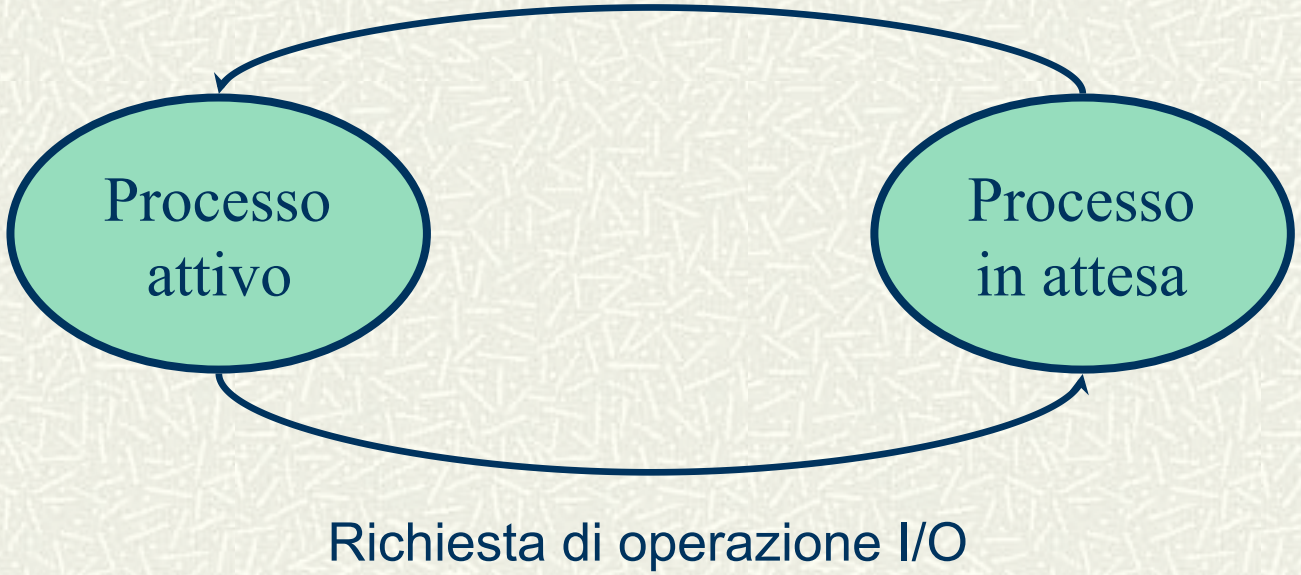
Fine Introduzione



Processi e programmi

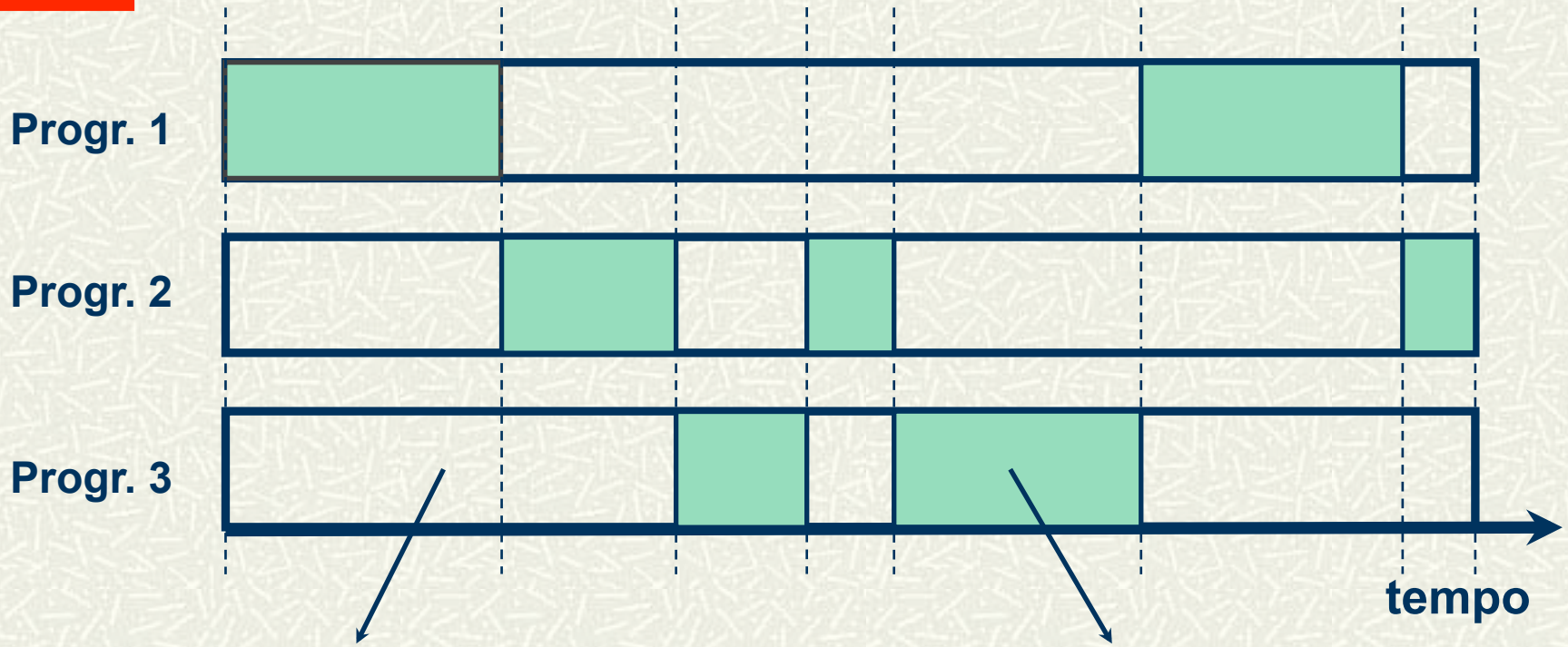
- Un **programma** è una entità statica composta dal codice eseguibile del processore.
- Un **processo** è una entità dinamica relativa al programma *in esecuzione,* ed è composto da:
  - codice del programma
  - dati necessari all'esecuzione del programma
  - stato dell'esecuzione

Esecuzione di un processo



Ogni operazione di I/O consiste in una chiamata al SO e successiva sospensione del processo utente per attendere l'esecuzione dell'operazione di I/O

La Multiprogrammazione



Time Sharing

- È possibile condividere la CPU tra più processi interattivi, **suddividendo** il tempo di esecuzione del processore tra più utenti
- Ogni processo utilizza periodicamente un intervallo di tempo prestabilito **(quanto)**
- Durante il quanto di esecuzione di un processo, tutti gli altri processi sono sospesi
- Al termine di ogni quanto **(context switch),** il processo in esecuzione viene sospeso e si assegna la CPU ad un altro processo.