

# Informatica per le discipline umanistiche

## Lezione 8 – File

`cristiano.longo@unict.it`



# Riepilogo

**1966** Unix (Ritchie et al.)

**1946** ACE (Turing)

**1935**  $\lambda$ -calculus (Church)

**1949** EDVAC (von Neumann et al.)

**1936** Macchina di Turing

**1928** Entscheidungsproblem (Hilbert)

**1879** Logica del Primo Ordine (Frege)

**1854** Logica dei Predicati (Boole)

**1837** Macchina Analitica (Babbage, Lovelace)

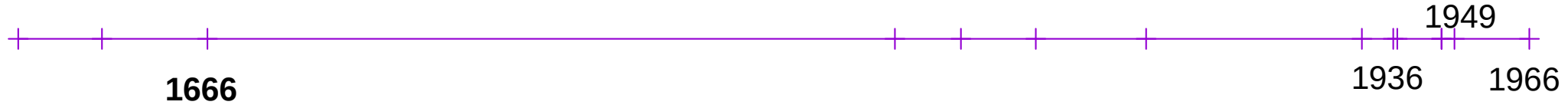
**1822** Macchina Differenziale (Babbage)

**1666** De arte combinatoria (Leibniz)

**1642** Pascalina (Pascal)

**1623** Orologio calcolante (Schickard)

# Riepilogo

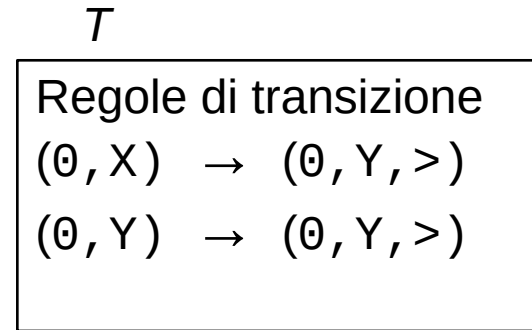
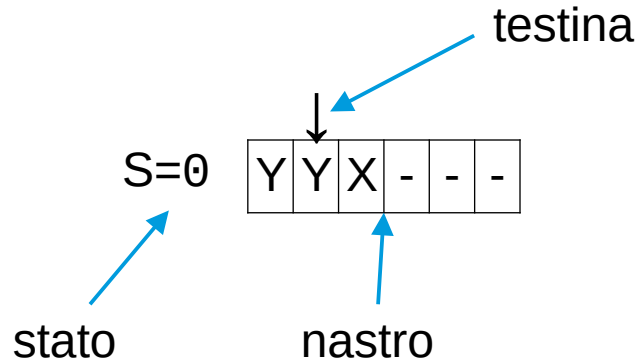
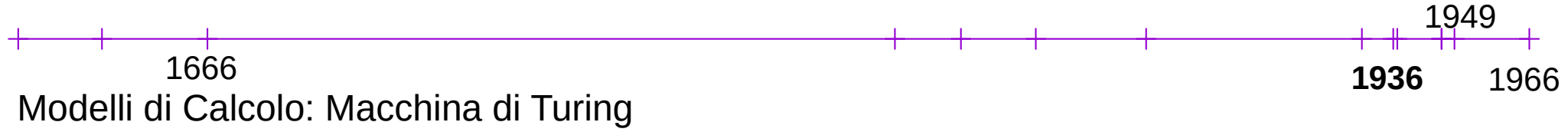


*“[...] quando sorga una controversia, non ci sarà più necessità di discussione tra due filosofi di quella che c'è tra due calcolatori. Sarà sufficiente prendere una penna, sedersi al tavolo e dirsi l'un l'altro: **calculemus!**”*

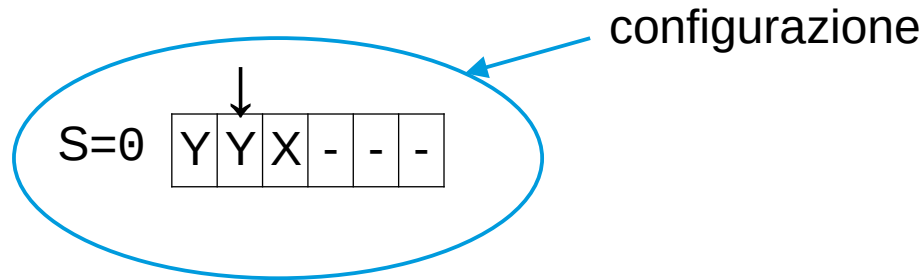
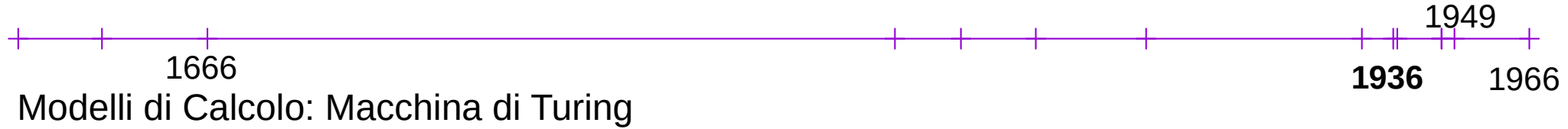
Dissertatio de arte combinatoria  
(Leibniz, 1666)



# Riepilogo – Macchina di Turing



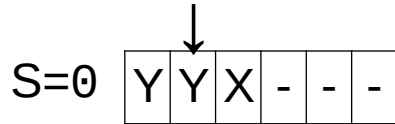
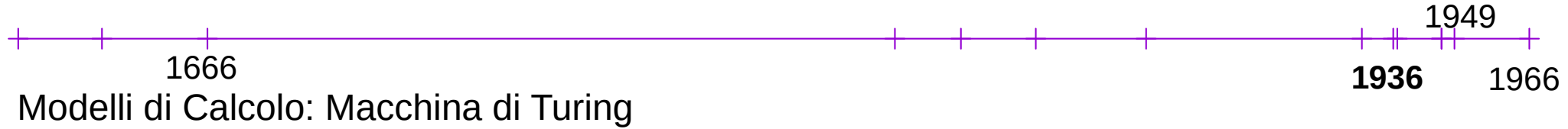
# Riepilogo – Macchina di Turing



*T*

Regole di transizione	
$(0, X)$	$\rightarrow (0, Y, >)$
$(0, Y)$	$\rightarrow (0, Y, >)$

# Riepilogo – Macchina di Turing



**Y $\emptyset$ YX**

configurazione in notazione compatta

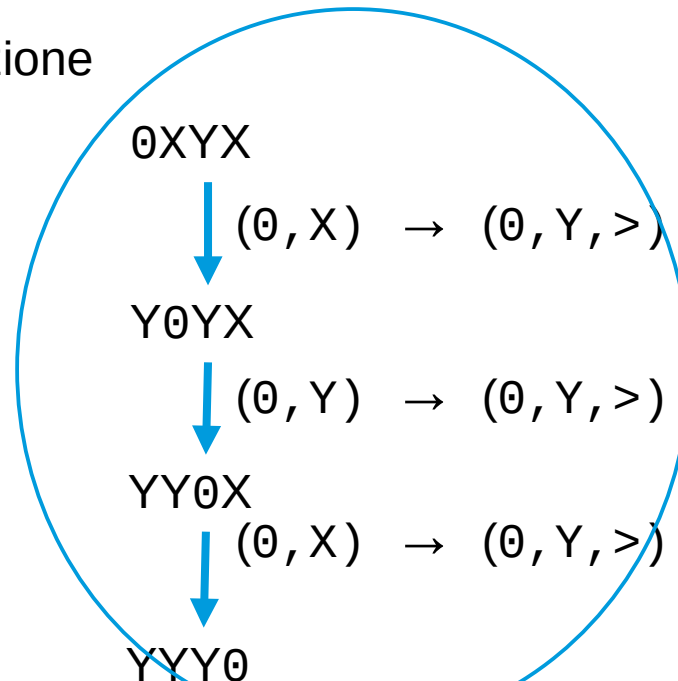
$T$

Regole di transizione
$(\emptyset, X) \rightarrow (\emptyset, Y, >)$
$(\emptyset, Y) \rightarrow (\emptyset, Y, >)$

# Riepilogo – Macchina di Turing



Computazione  
 $T(\theta XYX)$



$T$

Regole di transizione
$(\theta, X) \rightarrow (\theta, Y, >)$
$(\theta, Y) \rightarrow (\theta, Y, >)$

# Riepilogo – Macchina di Turing



Modelli di Calcolo: Macchina Universale

$U(T, \theta X Y X)$

$(\theta, X) \rightarrow (\theta, Y, >)$	$(\theta, Y) \rightarrow (\theta, Y, >)$	-	$\theta$	X	Y	X
------------------------------------------	------------------------------------------	---	----------	---	---	---



$(\theta, X) \rightarrow (\theta, Y, >)$	$(\theta, Y) \rightarrow (\theta, Y, >)$	-	Y	$\theta$	Y	X
------------------------------------------	------------------------------------------	---	---	----------	---	---



$(\theta, X) \rightarrow (\theta, Y, >)$	$(\theta, Y) \rightarrow (\theta, Y, >)$	-	Y	Y	$\theta$	X
------------------------------------------	------------------------------------------	---	---	---	----------	---

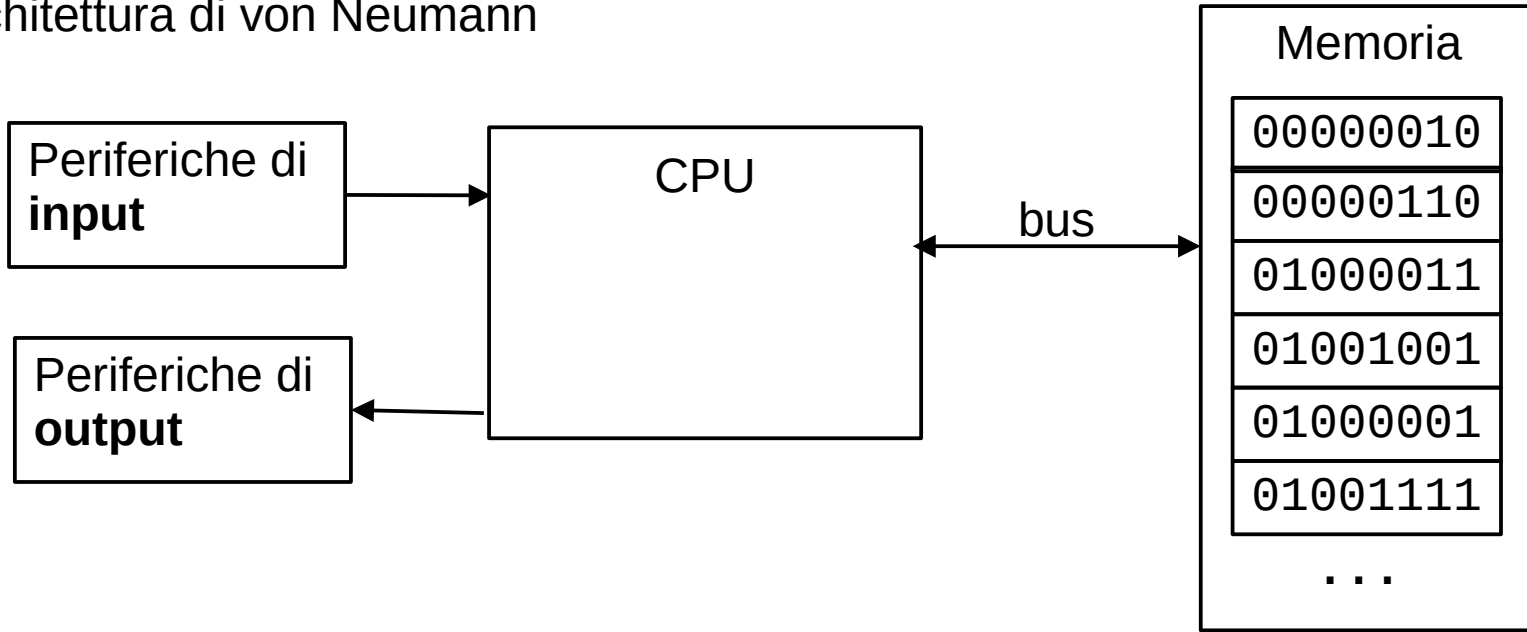


$(\theta, X) \rightarrow (\theta, Y, >)$	$(\theta, Y) \rightarrow (\theta, Y, >)$	-	Y	Y	Y	$\theta$
------------------------------------------	------------------------------------------	---	---	---	---	----------

$T$

Regole di transizione  
 $(\theta, X) \rightarrow (\theta, Y, >)$   
 $(\theta, Y) \rightarrow (\theta, Y, >)$

# Riepilogo – Architettura di von Neumann



# Riepilogo – Architettura di von Neumann

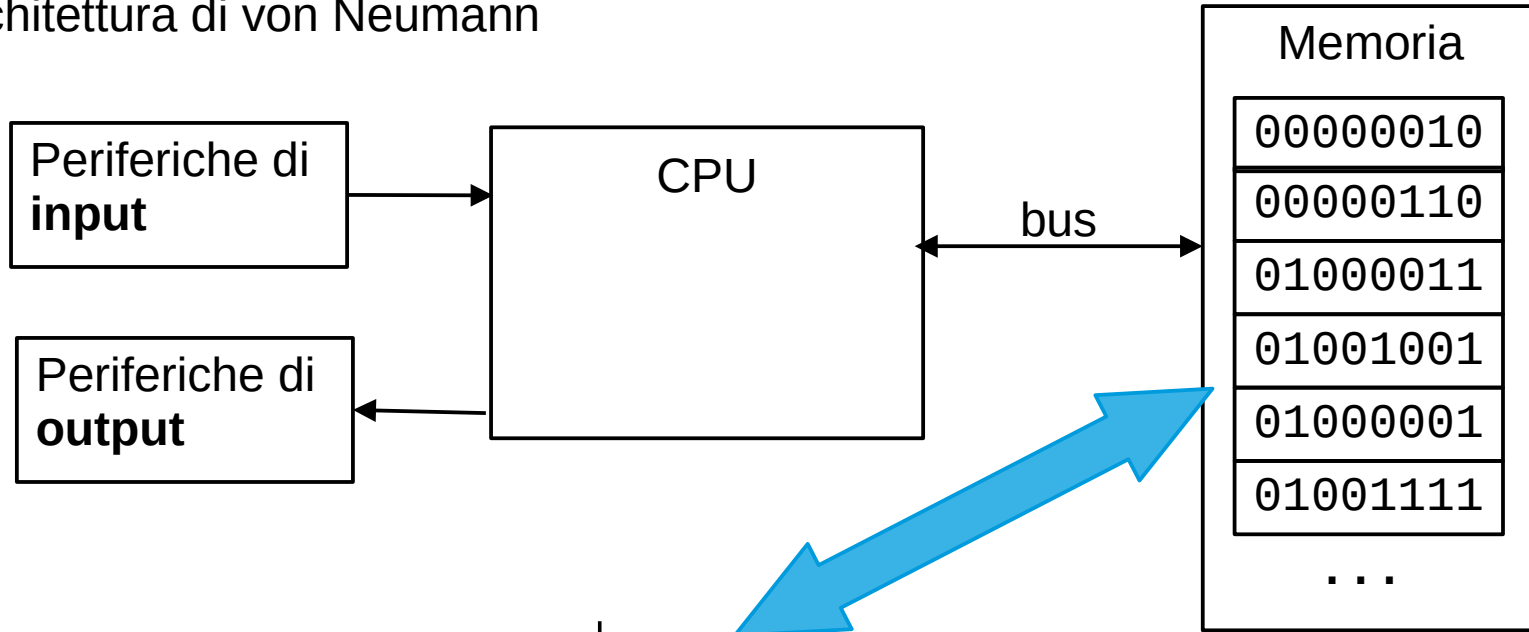
1949

1666

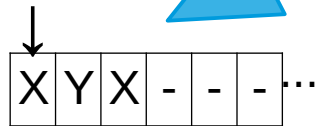
1936

1966

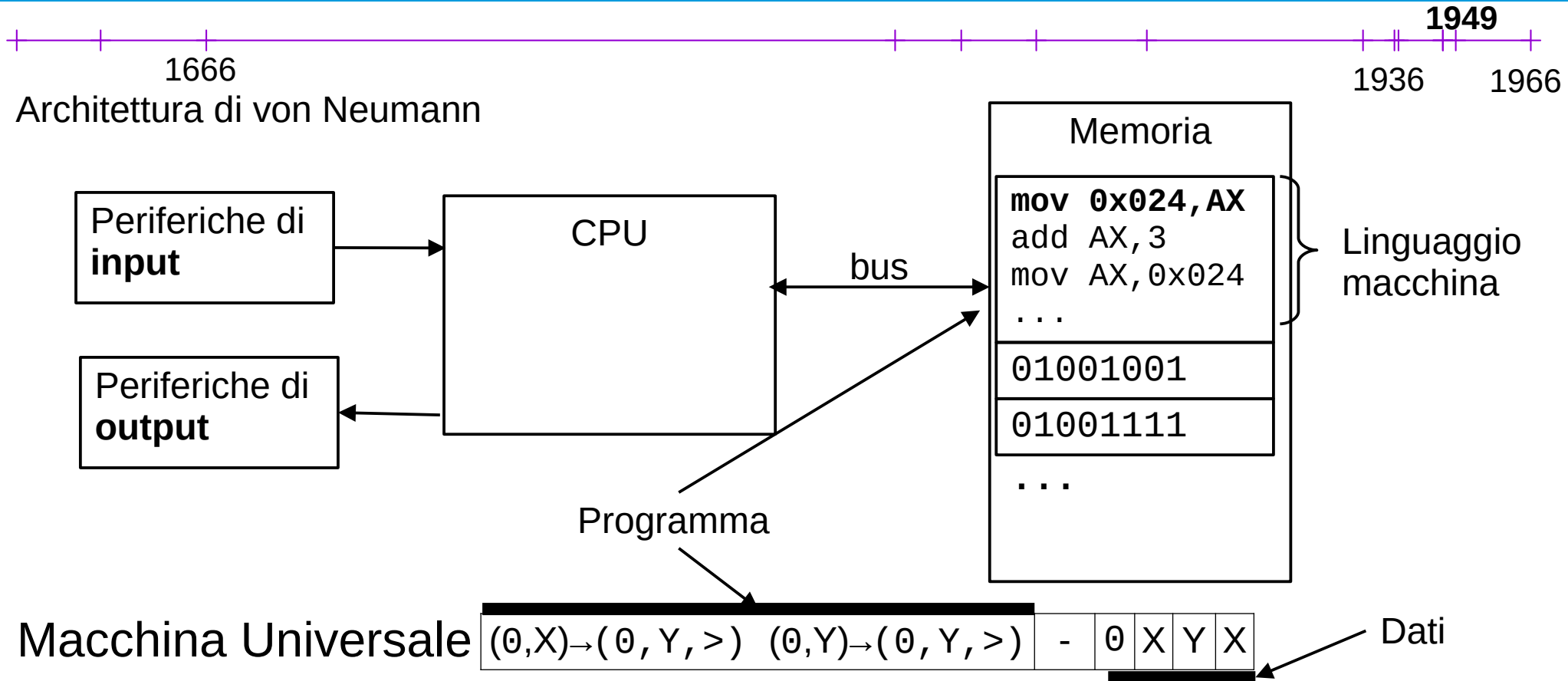
Architettura di von Neumann



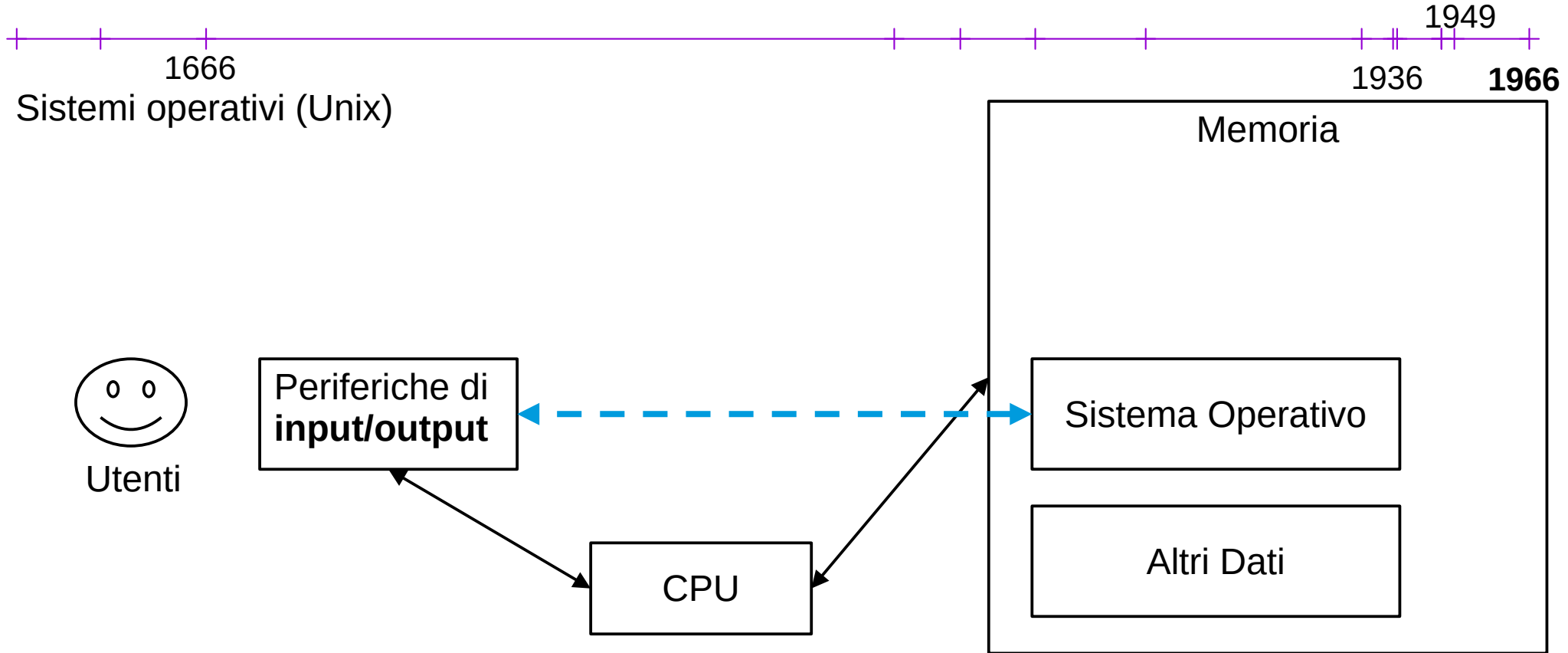
Macchina Universale



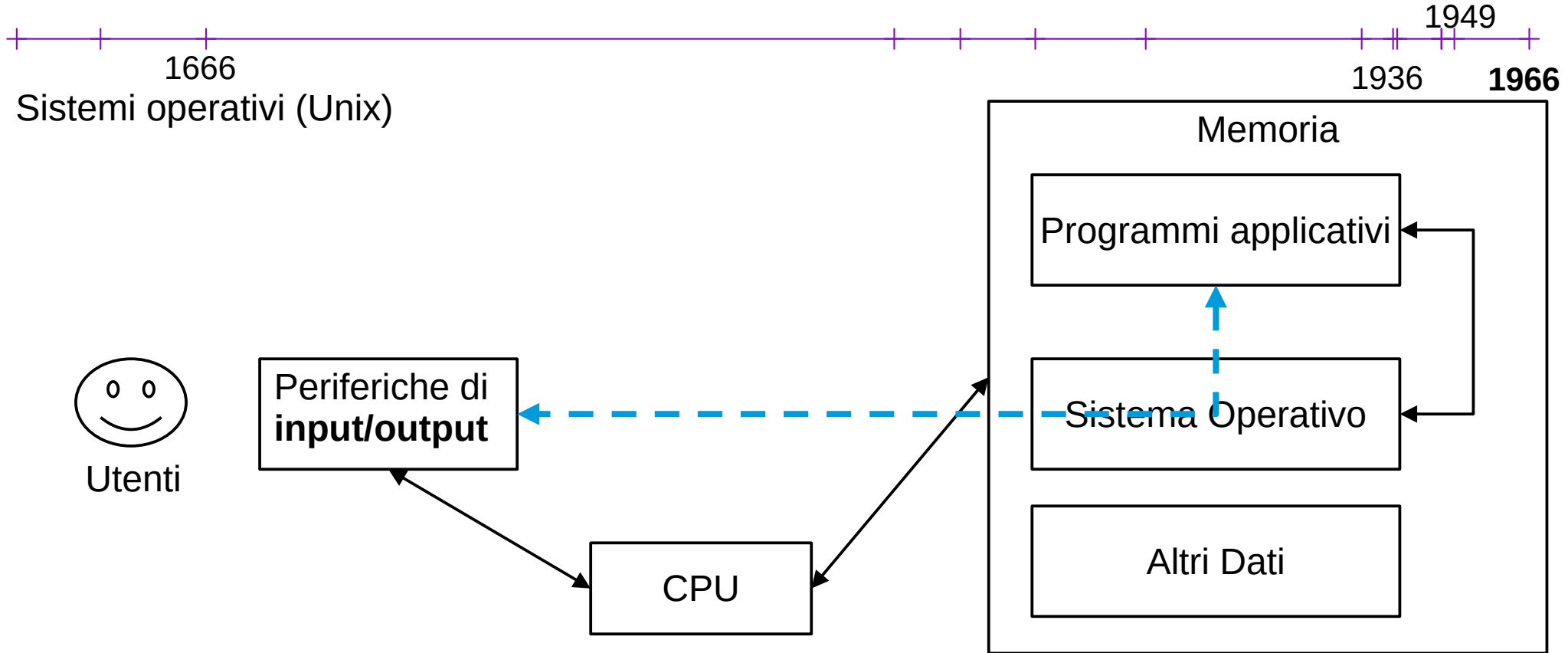
# Riepilogo – Architettura di von Neumann



# Riepilogo – Architettura di von Neumann



# Riepilogo – Architettura di von Neumann



# File

Un file è una sequenza finita di Byte.

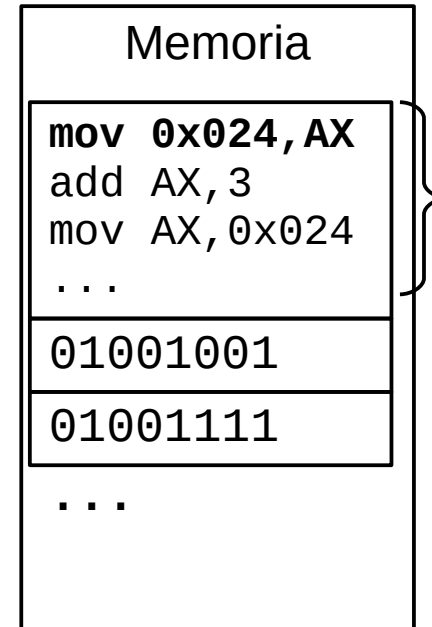
01000011	01001001	01000001	01001111
----------	----------	----------	----------

# File eseguibili

Un file è una sequenza finita di Byte.

01000011	01001001	01000001	01001111
----------	----------	----------	----------

Una sequenza di istruzioni in linguaggio macchina è un **file eseguibile**.



# File System

Le memorie possono essere viste come una sequenza di **locazioni** contenenti ciascuna un byte.

Il sistema operativo *organizza* i file all'interno delle memorie in **File System**.

Un file system è una struttura che permette di gestire molteplici file in una memoria.

Ad esempio dati due file *f1* ed *f2*.

Indirizzo	Contenuto
00000000	00000011
00000001	00000110
00000010	00000000
00000011	01001001
00000100	01000001
00000101	01001111
00000110	01001000
00000111	01000101
00001000	01001100
00001001	01001100
00001010	01001111

# File System

Le memorie possono essere viste come una sequenza di **locazioni** contenenti ciascuna un byte.

Il sistema operativo *organizza* i file all'interno delle memorie in **File System**.

Un file system è una struttura che permette di gestire molteplici file in una memoria.

Ad esempio dati due file *f1* ed *f2*:

- *f1* inizia alla locazione **00000011**,

Indirizzo	Contenuto
00000000	<b>00000011</b>
00000001	00000110
00000010	00000000
<b>00000011</b>	01001001
00000100	01000001
00000101	01001111
00000110	01001000
00000111	01000101
00001000	01001100
00001001	01001100
00001010	01001111

# File System

Le memorie possono essere viste come una sequenza di **locazioni** contenenti ciascuna un byte.

Il sistema operativo *organizza* i file all'interno delle memorie in **File System**.

Un file system è una struttura che permette di gestire molteplici file in una memoria.

Ad esempio dati due file *f1* ed *f2*:

- *f1* inizia alla locazione **00000011**,
- *f2* inizia alla locazione **00000110**.

Indirizzo	Contenuto
00000000	00000011
00000001	<b>00000110</b>
00000010	00000000
00000011	01001001
00000100	01000001
00000101	01001111
<b>00000110</b>	01001000
00000111	01000101
00001000	01001100
00001001	01001100
00001010	01001111

# File System

Le memorie possono essere viste come una sequenza di **locazioni** contenenti ciascuna un byte.

Il sistema operativo *organizza* i file all'interno delle memorie in **File System**.

Un file system è una struttura che permette di gestire molteplici file in una memoria.

Ad esempio dati due file *f1* ed *f2*:

- *f1* inizia alla locazione **00000011**,
- *f2* inizia alla locazione **00000110**.

Indirizzo	Contenuto
00000000	00000011
00000001	00000110
00000010	00000000
00000011	01001001
00000100	01001001
00000101	01001111
00000110	01001000
00000111	01000101
00001000	01001100
00001001	01001100
00001010	01001111

# Formattazione

Formattare una memoria significa installarci un file system vuoto.

Indirizzo	Contenuto
00000000	<b>00000000</b>
00000001	00000110
00000010	01000011
00000011	01001001
00000100	01000001
00000101	01001111
00000110	01001000
00000111	01000101
00001000	01001100
00001001	01001100
00001010	01001111

# Formattazione

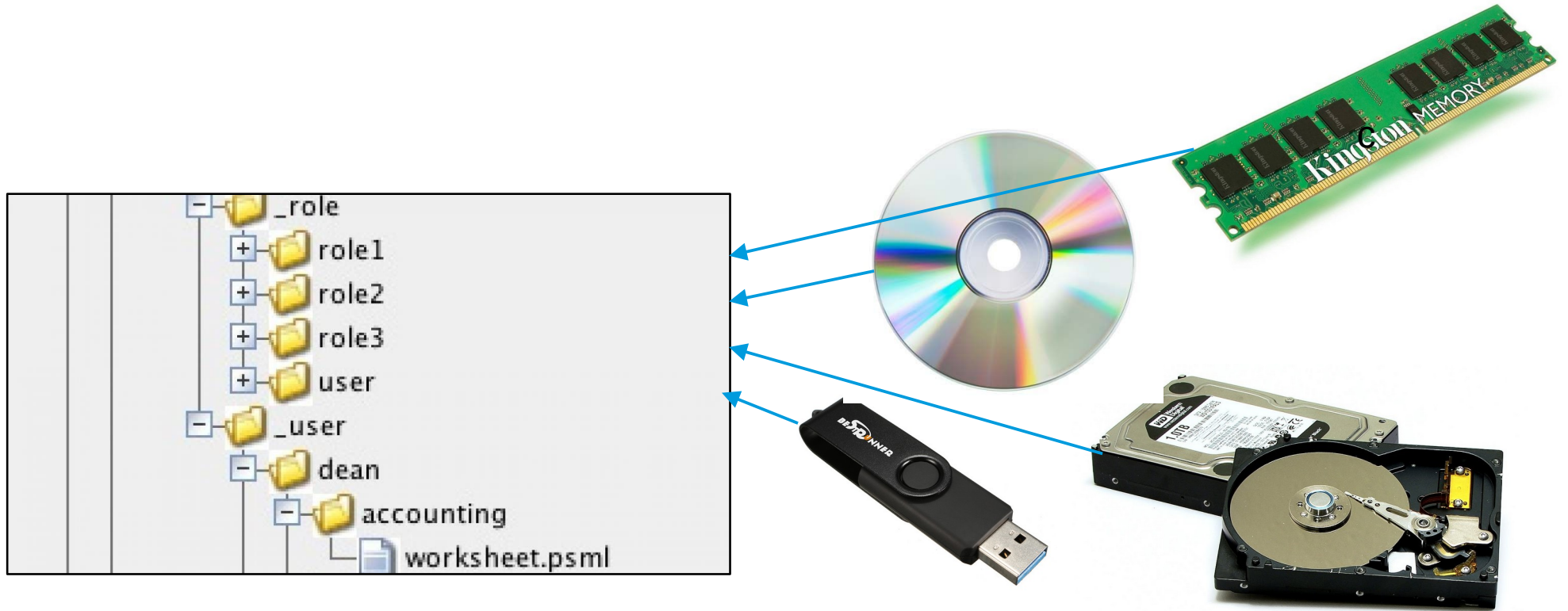
Formattare una memoria significa installarci un file system vuoto.

**ATTENZIONE: durante una formattazione i dati preesistenti non vengono effettivamente cancellati!**

Indirizzo	Contenuto
00000000	<b>00000000</b>
00000001	00000110
00000010	01000011
00000011	01001001
00000100	01000001
00000101	01001111
00000110	01001000
00000111	01000101
00001000	01001100
00001001	01001100
00001010	01001111

# File Manager

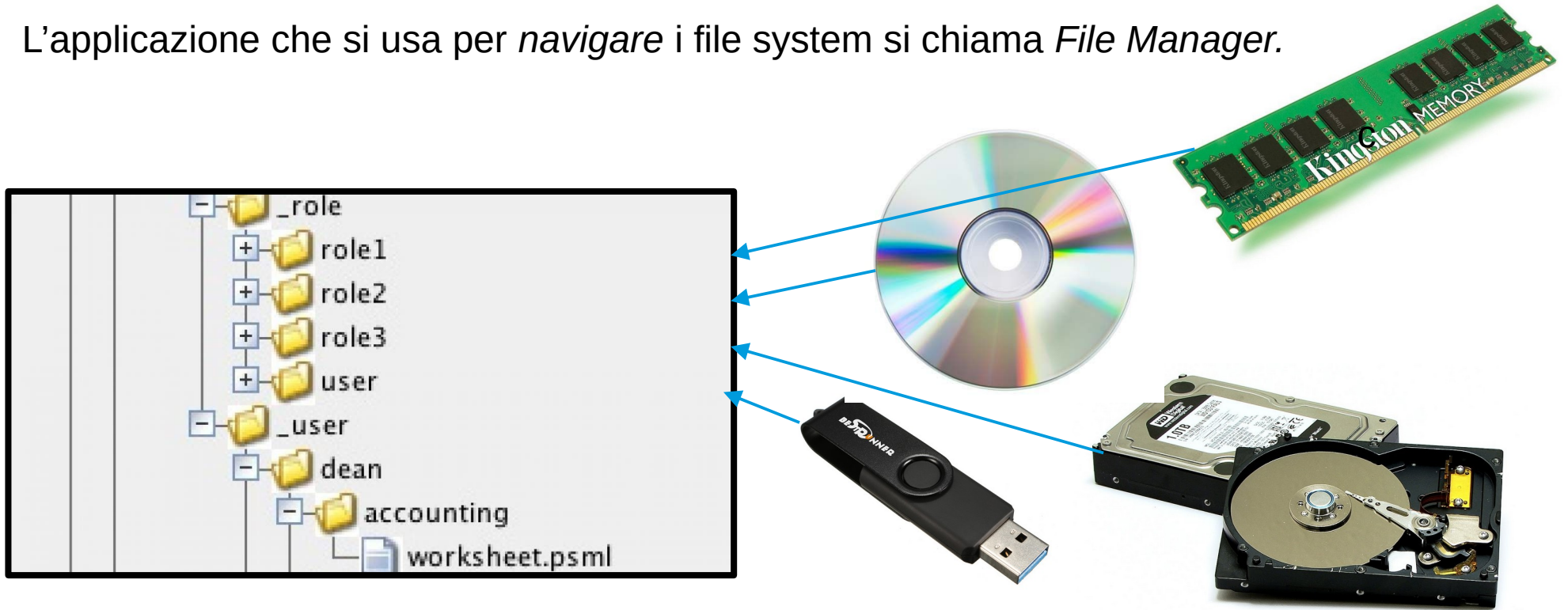
Il file system offre una *interfaccia* univoca indipendentemente dal tipo di memoria.



# File Manager

Il file system offre una *interfaccia* univoca indipendentemente dal tipo di memoria.

L'applicazione che si usa per *navigare* i file system si chiama *File Manager*.



# Directory

I file system normalmente permettono di raggruppare file in directory (cartelle).

- **Lezioni**
  - lezione1.pdf
  - lezione2.pdf
- **Saluti**
  - ciao.txt
  - hello.txt

# Directory

I file system normalmente permettono di raggruppare file in directory (cartelle).

Una directory può contenere altre directory.

- Lezioni
  - lezione1.pdf
  - lezione2.pdf
  - lezione3.pdf
- Saluti
  - **it**
    - ciao.txt
    - Buongiorno.txt
  - hello.txt

# Directory

I file system normalmente permettono di raggruppare file in directory (cartelle).


Una directory può contenere altre directory.

Un file system ha sempre una directory radice, che contiene tutti gli elementi della gerarchia.

- /
  - Lezioni
    - lezione1.pdf
    - lezione2.pdf
  - Saluti
    - **it**
      - ciao.txt
      - Buongiorno.txt
    - hello.txt

# Link


Un link è un *collegamento simbolico* ad un altro elemento (file o directory) nel file system.

- dir1
    - **file1.txt**
    - file2.txt
  - dir2
    - **ltofile1.txt**
    - file3.txt
- 

# Link

Un link è un *collegamento simbolico* ad un altro elemento (file o directory) nel file system.

Un file e tutti i link verso di lui risultano identici nei contenuti.

- dir1
    - **file1.txt**
    - file2.txt
  - dir2
    - **ltofile1.txt**
    - file3.txt
- 

**file1.txt**

Questo è il  
file1.

**ltofile1.txt**

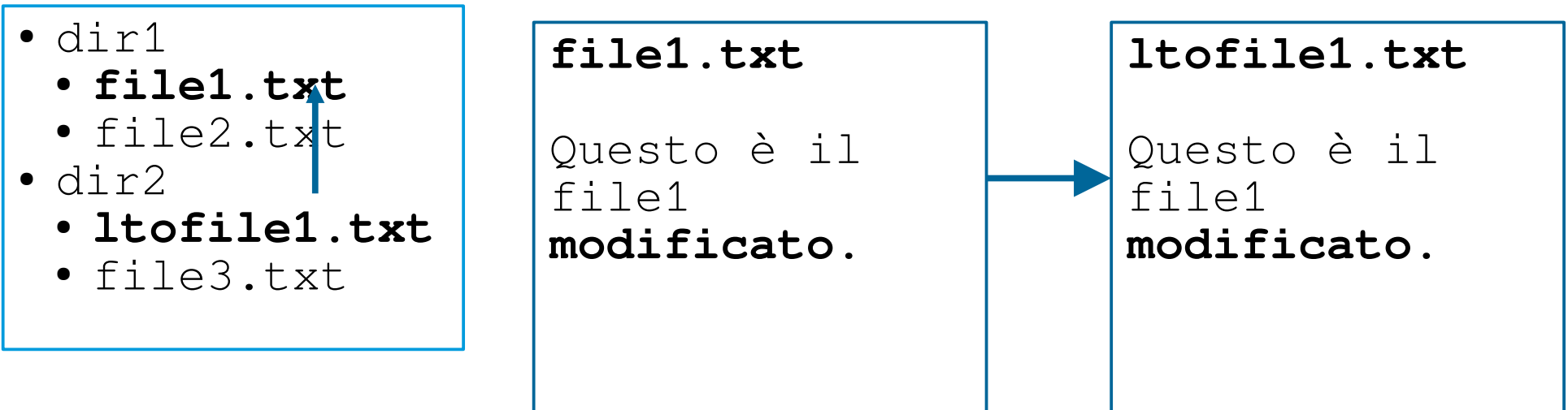
Questo è il  
file1.

# Link

Un link è un *collegamento simbolico* ad un altro elemento (file o directory) nel file system.

Un file e tutti i link verso di lui risultano identici nei contenuti.

Se modifico il file originale o uno dei suoi link, il file e tutti i link verso di lui risultano modificati allo stesso modo.



# Link

Un link è un *collegamento simbolico* ad un altro elemento (file o directory) nel file system.

Un file e tutti i link verso di lui risultano identici nei contenuti.

Se modifico il file originale o uno dei suoi link, il file e tutti i link verso di lui risultano modificati allo stesso modo.

Se elimino il link, il file originale resta immutato.

- dir1
  - **file1.txt**
  - file2.txt
- dir2
  - file3.txt

**file1.txt**

Questo è il  
file1.

~~**ltofile1.txt**~~

~~Questo è il  
file1.~~

# Link

Un link è un *collegamento simbolico* ad un altro elemento (file o directory) nel file system.

Un file e tutti i link verso di lui risultano identici nei contenuti.

Se modifico il file originale o uno dei suoi link, il file e tutti i link verso di lui risultano modificati allo stesso modo.

Se elimino il link, il file originale resta immutato. Se invece elimino il file originale, i link verso di lui perdono significato.

- dir1
  - ~~file1.txt~~
  - file2.txt
- dir2
  - **ltofile1.txt**
  - file3.txt

~~file1.txt~~  
Questo è il  
file1.

**ltofile1.txt**

~~Questo è il  
file1.~~

# Partizioni

Una memoria può essere **partizionata** per accogliere più di un file system. L

**C:**

- **C:/**
  - lezione1.pdf
  - lezione2.pdf
  - lezione3.pdf

**F:**

- **F:/**
  - **it**
    - ciao.txt
    - Buongiorno.txt
  - hello.txt

# Partizioni

Una memoria può essere **partizionata** per accogliere più di un file system.  
Le partizioni possono essere formattate separatamente.

> format F:

**C:**

- **C:/**
  - lezione1.pdf
  - lezione2.pdf
  - lezione3.pdf

**F:**

- **F:/**