



I Lincei per la Scuola
Fondazione



UNIVERSITÀ
DEGLI STUDI DI TRIESTE



I Lincei per una nuova didattica nella scuola: una rete nazionale
Polo di Trieste

CULTURA DIGITALE

A.S. 2021/22

Agostino Accardo (accardo@units.it)



uniTS

Introduzione all'Informatica: terminologia, architetture e linguaggi di programmazione

Cosa vedremo:

- Terminologia
- Breve storia dell'Informatica
- Macchina di von Neumann / memorie/ Dispositivi periferici
- Codifica delle informazioni
- Algoritmi e linguaggi di programmazione
- Cenni di programmazione
- Software di base e applicativi
- Sistemi Operativi



INFORMAZIONE (alcune definizioni, Treccani):

notizia, dato o elemento che consente di avere conoscenza più o meno esatta di fatti, situazioni, modi di essere, ecc.

il contenuto di novità e d'imprevedibilità di un messaggio intercorrente fra sistemi in relazione

SCIENZA DELL'INFORMAZIONE, fa parte di **SCIENZE MATEMATICHE** => **INFORMATICA TEORICA**

INGEGNERIA DELL'INFORMAZIONE, deriva da **INGEGNERIA ELETTRONICA** suddivisa in: **ING. ELETTRONICA**, **ING. AUTOMATICA**, **ING. TELECOMUNICAZIONI**, **ING. BIOMEDICA**, **ING. INFORMATICA**, **ING. GESTIONALE**
da cui: **ICT=INFORMATION & COMMUNICATIONS TECHNOLOGY**

INFORMATICA: scienza che si occupa del trattamento-elaborazione dell'INFORMAZIONE => deriva da....



INFORMATICA: «Gestione automatica delle informazioni» (dal francese *informat(ion) (automat)ique* 1962)

COMPUTER SCIENCE: implica un legame con la ‘scienza dello studio del *calcolatore*’

COMPUTER ⇔ **CALCOLATORE:** che fa calcoli o
ELABORATORE: che ‘elabora’ informazioni/dati secondo precisi comandi/ordini/istruzioni

ORDINATEUR: che dà/segue ordini

In ogni caso è uno STRUMENTO per eseguire molto rapidamente delle istruzioni

HARDWARE (macchina/computer) + **SOFTWARE** (programma)

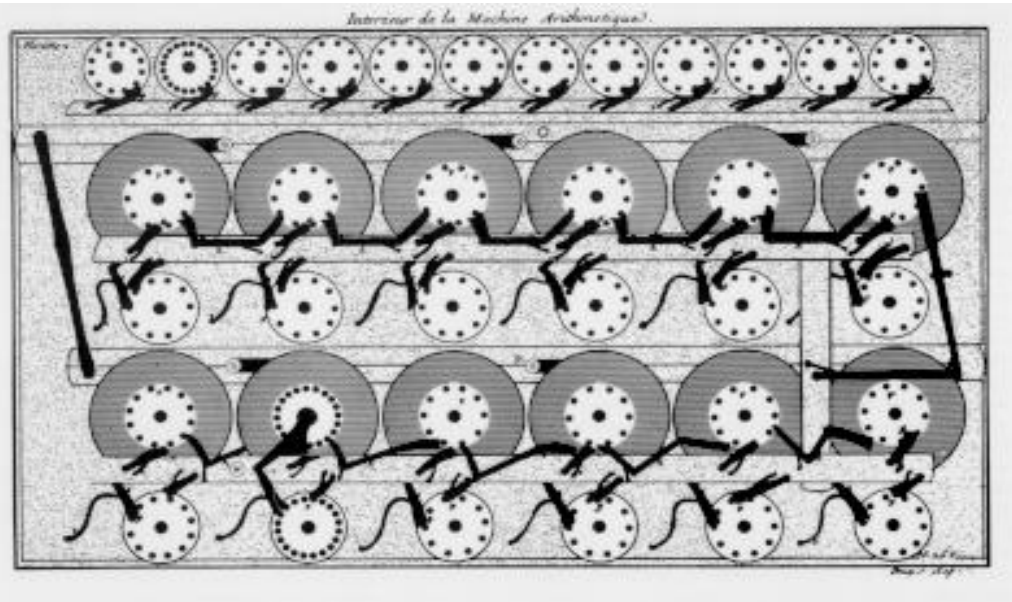
ALGORITMO: sequenza di istruzioni per eseguire un compito

Dalle macchine di calcolo di tipo meccanico

(1642 **Pascalina**)



Blaise Pascal, 1623-1662



1671 Leibnitz

1822-32 Babbage (Macchina differenziale e Macchina analitica)

1850 Boole

1892 Steiger (Millionaire, macchina a moltiplicazione diretta)

1899 Burroughs (Calcolatrice a moltiplicazione diretta)

1890 Hollerith (Macchina per il censimento americano del 1890, su scheda perforata)



.... a quelle teoriche

1936 Macchina di Turing (Alan Mathison Turing 1912-1954)
=> primo modello teorico di elaboratore programmabile a istruzioni memorizzate in grado di risolvere qualsiasi problema di logica simbolica in un numero finito di passi.

.... a quelle elettromeccaniche (che usano nastri perforati)....

1936 Zuse (calcolatori elettromeccanici: Z1, Z2, Z3 e Z4)

1937 Shannon descrive l'impiego della logica binaria per semplificare i circuiti elettrici a relé

1937 (progetto) 1944 (realizzazione) Aiken + IBM (International Business Machines): Mark1 o Bessie (Automatic Sequence Controlled Calculator)

.... a quelle a valvole elettroniche....

1943 Università Pennsylvania per l'esercito (calcoli balistici ad alta velocità):

ENIAC (Electronic Numerical Integrator And Computer)

Macchina decimale
impiegava 18.000 valvole
pesava 30 tonnellate
occupava 180 mq
eseguiva 300 moltiplicazioni al secondo.
Programmazione hardware!



1945 Macchina di Von Neumann

teorica => programmabile tramite software



... e dal 1948 a **transistor** (1952 **EDVAC**, EDSAC, MADM, UNIVAC, SEAC, MANIAC...)

MAIN FRAME --- MINICOMPUTER

1976 Cray-I, primo supercalcolatore, così potente che il Pentagono decise di venderlo solamente a pochi clienti fedeli agli USA

Inizio anni '70: nascono i microprocessori

1976: nasce il primo **PERSONAL COMPUTER (PC)**

OGGI: **COMPUTER PARALLELI**

Progetto Blue Gene: architettura progettata per realizzare supercomputer a parallelismo massivo (no sequenziale e quindi no di von Neumann) con potenze di calcolo dalle decine di teraflops (10^{12}) al petaflops (10^{15}).

Cinque progetti Blue Gene, tra i quali il Blue Gene/L, il Blue Gene/C, il Blue Gene/P e il Blue Gene/Q.

Nel 2019 il più veloce sviluppava una potenza di 478 teraflops

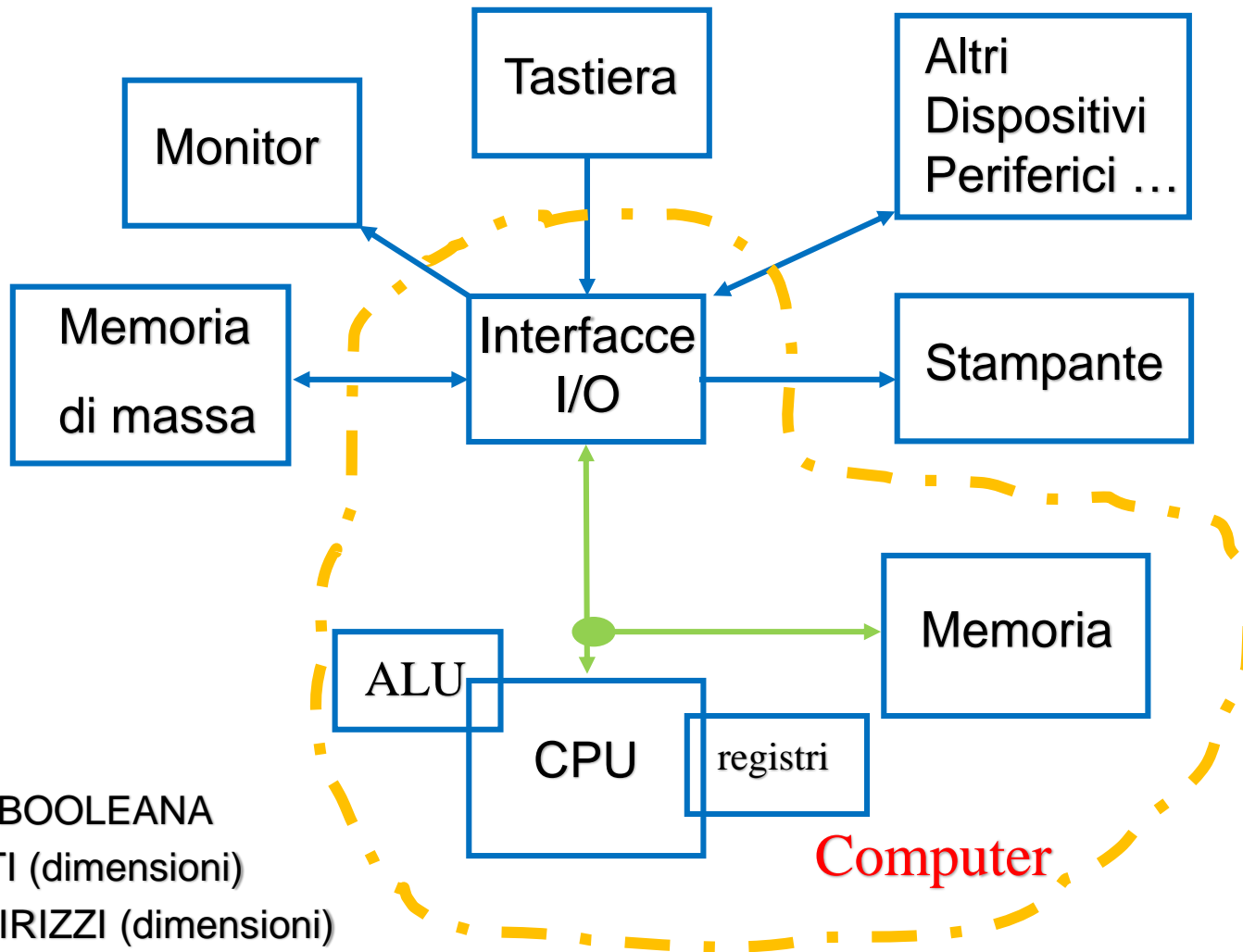


... DOMANI... **COMPUTER QUANTISTICI**

MACCHINA DI VON NEUMANN



I Lincei per la Scuola
Fondazione



BIT

BYTE

LOGICA BOOLEANA

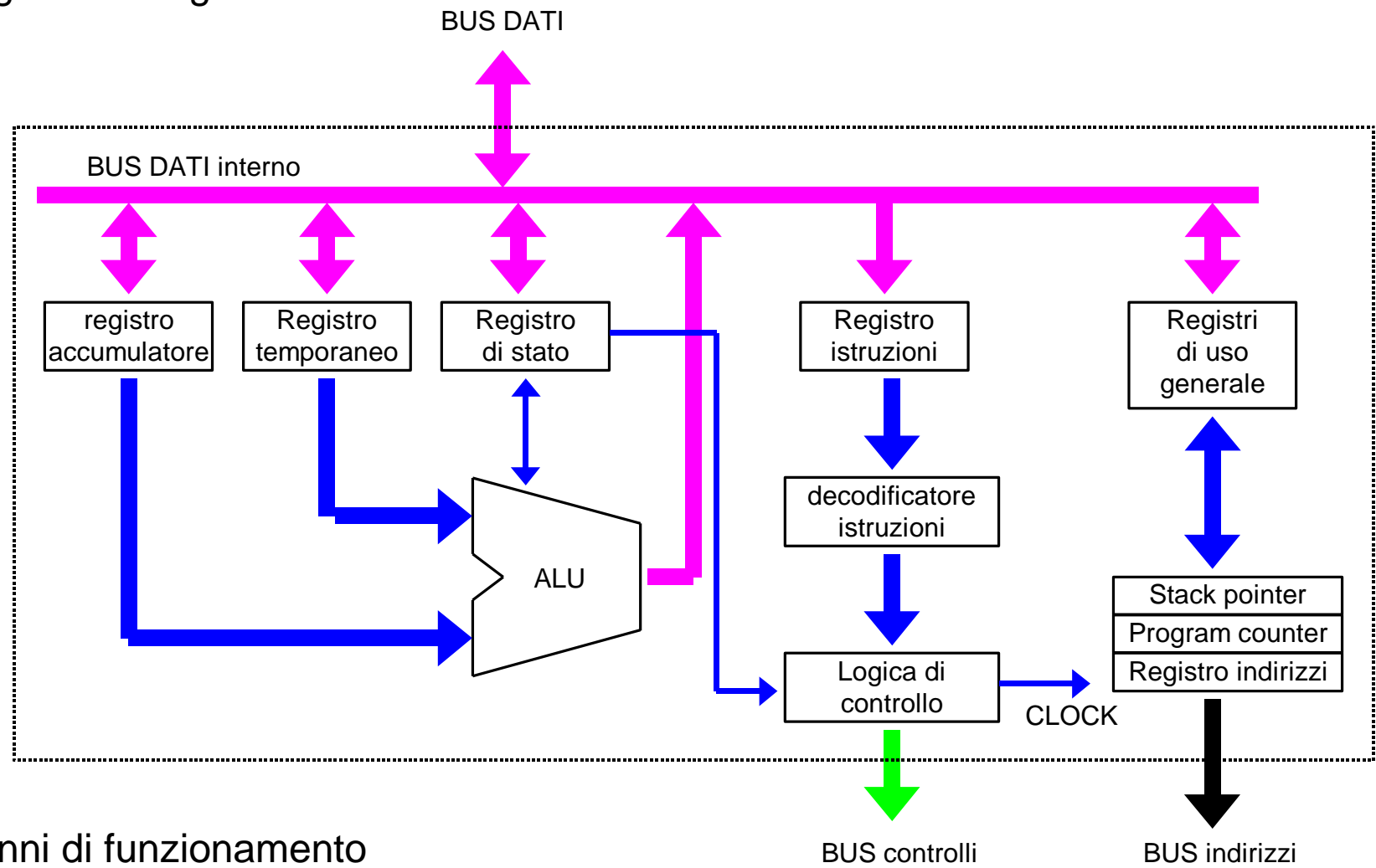
BUS DATI (dimensioni)

BUS INDIRIZZI (dimensioni)

BUS CONTROLLO

MACCHINA DI VON NEUMANN – HW CPU

il microprocessore (CPU) contiene la logica di controllo, l'unità aritmetico-logica ed i registri interni



Cenni di funzionamento



ISTRUZIONI/COMANDI

Ogni CPU dispone di un gruppo specifico di comandi che viene chiamato set di istruzioni.

CISC=Complex Instruction Set Computer

o **RISC=Reduced Instruction Set Computer**

Il programmatore può utilizzare le istruzioni usando i linguaggi di programmazione:
macchina/assembler, C++, Java, ...

Istruzioni: per la macchina sono sequenze di cifre binarie, per un linguaggio sono sequenze di caratteri che andranno ‘tradotti’ nel linguaggio della macchina

Le istruzioni svolgono operazioni elementari:

Istruzioni di trasferimento dati

Istruzioni aritmetiche

Istruzioni logiche

Istruzioni di salto

Istruzioni di rotazione e di shift

Istruzioni di Input/Output

Istruzioni d’interruzione

Centinaia di famiglie di CPU ‘single Core’, fino ai sistemi a 64bit all’inizio del 2000:

INTEL:

Architettura P6: Pentium3 (1999-2003, 450 MHz / 1.4 GHz)

Architettura P7: Pentium4 (2000-2008, 1.3 GHz / 3.8 GHz)

AMD:

Athlon 64 (dal 2003, 1.0 GHz / 3.2 GHz)

MOTOROLA per MAC:

Power PC 970FX (2004-2006, 1.6 GHz / 2.7 GHz)

Causa impossibilità di incremento velocità del clock a causa dell’eccessivo calore sviluppato dai transistors e dei conseguenti problemi di dissipazione termica, si è passati ad ottimizzare le prestazioni incrementando l’efficienza dell’architettura e la parallelizzazione delle operazioni => Sistemi Multi-Core



da

INTEL: CORE 2 (dal 2006 al 2011, 1.06 GHz / 3.33 GHz)

AMD: ATHLON 64 X2 (dal 2005, 1.9 GHz / 3.2 GHz)

MOTOROLA: POWERPC 970MP (2005-2006, 2.0-2.5GHz)

Architettura per portatili desktop e workstation

Due unità operative e cache su singolo Chip



a

INTEL: CORE i9 (dal 2019, 8 Core, 14nm, 2.1 GHz)

AMD: ZEN+ (dal 2018, 6 Core, 12nm, 3.1-4.2 GHz)

INTEL x MAC: COFFEE LAKE (dal 2018, 6 Core, 14nm, 2.2-2.9 GHz)

Dim. dei *gate* dei transistor=12-14 nm, 3-7 Gtransistor

Istruzioni X86 a 64 bit, 1 istruzione a 128-256 bit in un ciclo





AMD Athlon™ 64 FX Dual-Core Processor Architecture (Socket 939)

AMD64 Technology Core

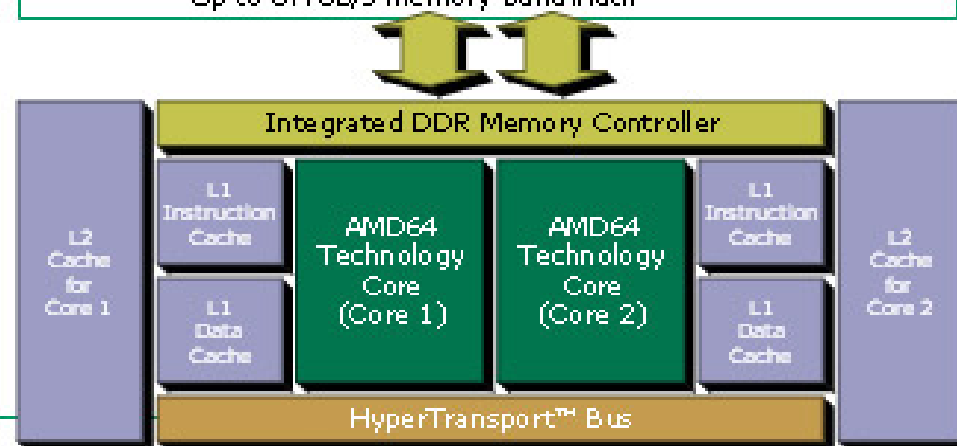
- Leading-edge performance for 32-bit and 64-bit computing
- Additional 64-bit internal registers for computing performance
- Addressability beyond 4GB for new capabilities
- Enhanced Virus Protection*

Integrated DDR Memory Controller

- High-performance, high-bandwidth
- Designed to reduce DRAM latency
- Boosts performance for many applications, especially memory intensive applications
- PC-3200 (DDR-400), PC-2700 (DDR-333), PC-2100 (DDR-266) or PC-1600 (DDR-200)
 - Unbuffered DIMMs
 - 128-bit interface
 - Up to 6.4GB/s memory bandwidth

Cache

- Large on-die cache memory
 - 64KB L1 instruction cache per core
 - 64KB L1 data cache per core
 - 1024KB L2 cache per core
 - 1152KB total effective cache per core
 - 2304KB total effective cache per processor
 - Designed to improve performance for many applications, especially multi-threaded applications



HyperTransport™ Bus

- A system bus that uses HyperTransport technology for high-speed I/O communication
 - Up to 8GB/s of available system bandwidth

RAM, Random Access Memory= memoria elettronica, volatile

ROM, Read Only Memory = memoria permanente

EPROM (Erasable Programmable ROM)

EAROM (Electrically Alterable ROM)

EEPROM (Electrically Erasable Programmable ROM)

....

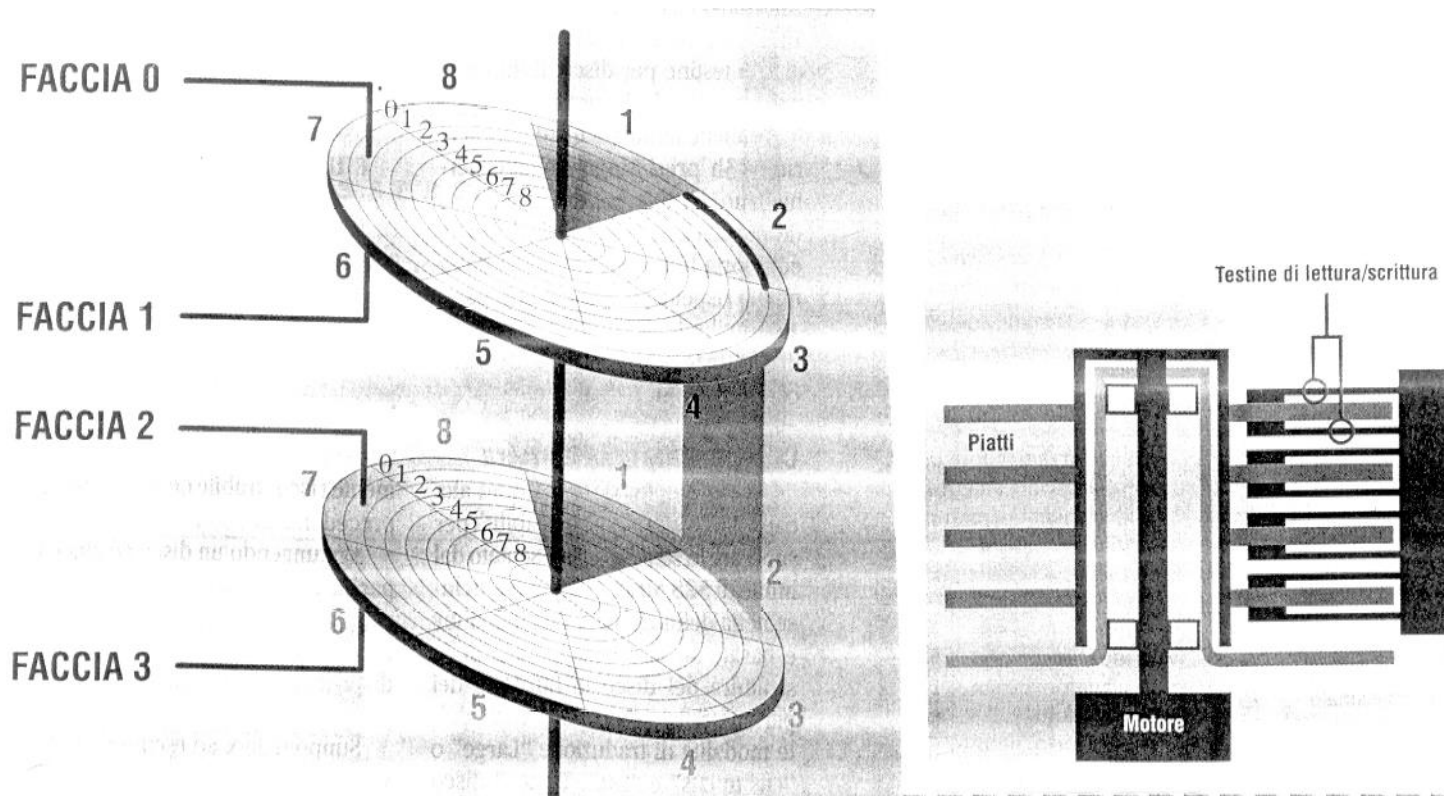
Memoria Periferica (o di massa) di LETTURA / SCRITTURA:

nastri (bobine), dischi (Floppy – Hard), penne USB, unità SSD (impropriamente dette ‘dischi allo stato solido’), CD, DVD, ...

dispositivi magnetici, elettronici, ottici => memorie permanenti

MEMORIE – DISPOSITIVI DI ARCHIVIAZIONE - HD

L'HD è ormato da più piatti magnetici rotanti. Ogni piatto diviso in tracce. Ogni traccia divisa in settori, costituiscono la più piccola unità logica accessibile sul disco. La scrittura e lettura dei dati avviene cambiando la polarizzazione magnetica delle particelle che ricoprono la superficie dei piatti tramite apposite testine di R/W



MEMORIE – DISPOSITIVI DI ARCHIVIAZIONE - HD

capacità : GB-TB

velocità di rotazione dei piatti (maggiore velocità => maggiore quantità di dati che passa sotto la testina magnetica) 5.400 - 10.000 giri/min (RPM).

Tempo medio di ricerca (seek) della traccia e di accesso al settore desiderato per recuperare il dato: pochi ms

MEMORIE – DISPOSITIVI DI ARCHIVIAZIONE - CD

Introdotti nel 1979 da SONY e PHILIPS
disco di resina termoplastica trasparente

12 cm diametro

Meno per i mini CD

ricoperto da un sottile foglio di materiale metallico (Ag, leghe Ag, Au)

le informazioni sono memorizzate come successioni di

"*pits*" - "buchi" - 1 e

"*lands*" - "terre" - 0

Letti e scritti con LASER (quindi dischi ottici)

Unica traccia (solco guida del laser) a spirale: parte dal centro

Si massimizza la velocità nell'accesso sequenziale

Si penalizza l'accesso diretto

Derivano da CD musicali e vanno letti in modo da produrre un flusso di dati costante

Variano la velocità di rotazione dall'interno 500rpm all'esterno 200rpm

MEMORIE – DISPOSITIVI DI ARCHIVIAZIONE - DVD

Nascono per volontà del DVD (Digital Video/Versatile Disk) Forum formato da Philips, Sony, Matsushita, Hitachi, Warner, Toshiba, JVC, Thomson, Pioneer si è incaricato di redigere le specifiche del nuovo supporto

Il *DVD forum* individua 3 principali campi d'applicazione per il DVD:

DVD-Video, destinato a contenere film, in sostituzione della videocassetta.

DVD-Audio, pensato per sostituire il CD Audio grazie a una maggiore fedeltà.

DVD-ROM, destinato a sostituire il CD-ROM

FORMATI:

DVD-R/-RW e il DVD+R/+RW (più affidabile, con un migliore controllo degli errori) usano una tecnica di registrazione elicoidale

Il DVD-RAM usa una tecnica di registrazione a cerchi concentrici, simile a un HardDisk, al quale è assimilabile nell'uso.

DVD+R DL Double Layer e DVD-R DL, DVD a doppio strato con una capienza di circa 9 GB

DISPOSITIVI PERIFERICI

I dispositivi vengono collegati alle interfacce connesse ai Bus e possono essere gestiti in 3 modi :

- **via software**: l'intervento è asincrono per cui non è possibile prevedere quando la periferica viene servita, può venire impiegato per la lettura di strumenti di misura secondo una sequenza prefissata (p.es. stampanti in background)
- con richiesta di **interruzione** da parte della periferica: la CPU interviene quando la periferica lo richiede mediante l'attivazione di una linea di interrupt che sospende l'esecuzione del programma ed esegue la routine di gestione dell'interrupt. Per individuare quale dispositivo ha richiesto l'interruzione si usano tecniche di Polling o di vettorizzazione (p.es. mouse, tastiera)
- con accesso diretto alla memoria (**DMA**): per trasferire dati dall'esterno alla RAM ad alta velocità; la CPU 'cede' i bus al dispositivo periferico che li gestisce direttamente (p.es. HD, DVD, ecc,)

CODIFICA DELLE INFORMAZIONI - Caratteri

CODIFICA DELLE INFORMAZIONI

CARATTERI (ASCII) = 1 BYTE

•ASCII = American Standard Code for Information Interchange

codice a (7) 8 bit -ASCII esteso

•Primi 32 caratteri con valori decimali da 00 a 31: caratteri di controllo (^...);

•da 32₁₀ a 64₁₀ si hanno caratteri speciali come la spazio, il \$, le parentesi (), le cifre da 0 a 9 e la @ ;

•da 65₁₀ a 90₁₀ lettere maiuscole A-Z;

•poi ancora caratteri speciali come le parentesi. {};

•da 97₁₀ a 122₁₀ lettere minuscole a-z;

•poi ancora caratteri speciali e simboli grafici.

DEC	HEX	OCT	HTML	CHR	DEC	HEX	OCT	HTML	CHR
0	0	000	-	NUL	64	40	100	@	@
1	1	001	-	SOH	65	41	101	A	A
2	2	002	-	STX	66	42	102	B	B
3	3	003	-	ETX	67	43	103	C	C
4	4	004	-	EOT	68	44	104	D	D
5	5	005	-	ENQ	69	45	105	E	E
6	6	006	-	ACK	70	46	106	F	F
7	7	007	-	BEL	71	47	107	G	G
8	8	010	-	BS	72	48	110	H	H
9	9	011	-	TAB	73	49	111	I	I
10	A	012	-	LF	74	4A	112	J	J
11	B	013	-	VT	75	4B	113	K	K
12	C	014	-	FF	76	4C	114	L	L
13	D	015	-	CR	77	4D	115	M	M
14	E	016	-	SO	78	4E	116	N	N
15	F	017	-	SI	79	4F	117	O	O
16	10	020	-	DLE	80	50	120	P	P
17	11	021	-	DC1	81	51	121	Q	Q
18	12	022	-	DC2	82	52	122	R	R
19	13	023	-	DC3	83	53	123	S	S
20	14	024	-	DC4	84	54	124	T	T
21	15	025	-	NAK	85	55	125	U	U
22	16	026	-	SYN	86	56	126	V	V
23	17	027	-	ETB	87	57	127	W	W
24	18	030	-	CAN	88	58	130	X	X
25	19	031	-	EM	89	59	131	Y	Y
26	1A	032	-	SUB	90	5A	132	Z	Z
27	1B	033	-	ESC	91	5B	133	[[
28	1C	034	-	FS	92	5C	134	\	\
29	1D	035	-	GS	93	5D	135]]
30	1E	036	-	RS	94	5E	136	^	^
31	1F	037	-	US	95	5F	137	_	_
32	20	040	 	Space	96	60	140	`	`
33	21	041	!	!	97	61	141	a	a
34	22	042	"	"	98	62	142	b	b
35	23	043	#	#	99	63	143	c	c
36	24	044	$	\$	100	64	144	d	d
37	25	045	%	%	101	65	145	e	e
38	26	046	&	&	102	66	146	f	f
39	27	047	'	'	103	67	147	g	g
40	28	050	((104	68	150	h	h
41	29	051))	105	69	151	i	i
42	2A	052	*	*	106	6A	152	j	j
43	2B	053	+	+	107	6B	153	k	k
44	2C	054	,	,	108	6C	154	l	l
45	2D	055	-	-	109	6D	155	m	m
46	2E	056	.	.	110	6E	156	n	n
47	2F	057	/	/	111	6F	157	o	o
48	30	060	0	0	112	70	160	p	p
49	31	061	1	1	113	71	161	q	q
50	32	062	2	2	114	72	162	r	r
51	33	063	3	3	115	73	163	s	s
52	34	064	4	4	116	74	164	t	t
53	35	065	5	5	117	75	165	u	u
54	36	066	6	6	118	76	166	v	v
55	37	067	7	7	119	77	167	w	w
56	38	070	8	8	120	78	170	x	x
57	39	071	9	9	121	79	171	y	y
58	3A	072	:	:	122	7A	172	z	z
59	3B	073	;	;	123	7B	173	{	{
60	3C	074	<	<	124	7C	174	|	
61	3D	075	=	=	125	7D	175	}	}
62	3E	076	>	>	126	7E	176	~	~
63	3F	077	?	?	127	7F	177		DEL

DEC	CHR	DEC	CHR	DEC	CHR
128	€	172	↵	216	∅
129	□	173		217	Ù
130	,	174	@	218	Ú
131	f	175	—	219	Û
132	..	176	°	220	Ü
133	...	177	±	221	Ý
134	†	178	²	222	Þ
135	‡	179	³	223	Ë
136	~	180	´	224	à
137	%	181	µ	225	á
138	Š	182	¶	226	â
139	<	183	·	227	ã
140	Œ	184	,	228	ä
141	□	185	'	229	å
142	Ž	186	°	230	æ
143	□	187	>	231	ç
144	□	188	¼	232	è
145	'	189	½	233	é
146	'	190	¾	234	ê
147	"	191	¿	235	ë
148	"	192	À	236	ì
149	•	193	Á	237	í
150	—	194	Â	238	î
151	—	195	Ã	239	ï
152	~	196	Ä	240	ð
153	™	197	Å	241	ñ
154	§	198	Æ	242	ò
155	>	199	Ç	243	ó
156	œ	200	È	244	ô
157	□	201	É	245	õ
158	ž	202	Ê	246	ö
159	ÿ	203	Ë	247	+
160		204	Ì	248	ø
161	¡	205	Í	249	ù
162	¢	206	Î	250	ú
163	£	207	Ï	251	û
164	¤	208	Ð	252	ü
165	¥	209	Ñ	253	ý
166	¦	210	Ò	254	þ
167	§	211	Ó	255	ÿ
168	¨	212	Ô		
169	©	213	Õ		
170	ª	214	Ö		
171	«	215	×		

CODIFICA DELLE INFORMAZIONI - Caratteri

Creata centinaia di sistemi di codifica, ognuno abbinava i numeri ai caratteri in modo differente

Nessuna codifica comprendeva un numero di caratteri sufficiente per tutte le circostanze, anche per una sola lingua

Qualsiasi elaboratore, e a maggior ragione un server di rete ed in particolare il **server di posta** (!), correva il rischio di non riuscire a decodificare correttamente le informazioni

=> Anni '80 e segg: standardizzazione ISO 8859-n ('parte' n da 1 a 15, ciascuna parte a 8bit)

Nonostante ciò la conversione tra codifiche era spesso complicata e la possibilità di usare più bit a costi esigui ha sempre più facilitato la diffusione di UNICODE, così

⇒ Nel 2004 si decide di lasciare l'ISO 8859 e di passare a UNICODE Sistema base (basic multilingual plane) a 16 bit (65535 simboli diversi)

+

Altri 16 sistemi (plan) supplementari. Ad oggi include oltre 100.000 simboli

CODIFICA DELLE INFORMAZIONI – Numeri Interi

Codifica posizionale => stesso valore = diverse rappresentazioni
base 10 (decimale), base 2 (binaria), base 16 (esadecimale)

Trasformazione binaria-decimale: somma dei prodotti tra i bit e le corrispondenti potenze di 2

Es. $0001\ 0101\ 0111 = 2^8+2^6+2^4+2^2+2^1+2^0 = 343_{10} = 1 \cdot 16^2 + 5 \cdot 16^1 + 7 \cdot 16^0$
Simboli in base 16: 0,1..9,A,B,C,D,E,F

Rappresentazione e corrispondenza esadecimale (**offset binario**):

Binario	Esad.	Dec.	Binario	Esad.	Dec.
0000	0	0	0001	1	1
....	1001	9	9
1010	A	10	1011	B	11
1100	C	12	1101	D	13
1110	E	14	1111	F	15

Intervallo dei numeri rappresentabili con n cifre binarie in offset binario (solo numeri senza segno): da 0 a $2^{n-1}-1$

Esempi di trasformazione decimale-binaria

CODIFICA DELLE INFORMAZIONI – Numeri Interi

Rappresentazione e corrispondenza esadecimale (**complemento a 2**) con 4 cifre binarie:

Binario	Esad.	Dec.	Binario	Esad.	Dec.
0000	0	0	1111	F	-1
0001	1	1	1110	E	-2
....
0111	7	7	1000	8	-8

CODIFICA NECESSARIA PER AVERE I NUMERI CON SEGNO

OPERAZIONE DI COMPLEMENTO A 2:

0011 0101 ($35_{16}=53_{10}$) \Rightarrow 1100 1011 ($CB_{16} = -53_{10}$) la cui somma fa 0!

RANGE DEI NUMERI A n CIFRE BINARIE in complemento a 2:

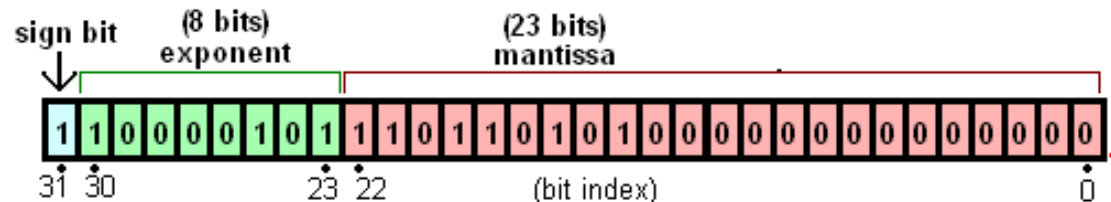
$[-2^{n-1}, +2^{n-1}-1]$

CODIFICA DELLE INFORMAZIONI - Numeri decimali

NUMERI CON LA VIRGOLA (FLOATING POINT):

MANTISSA, ESPONENTE E SEGNO

- es. con 32 bit di cui 1 bit per S, 8 bit per E, 23 bit per M (ma esistono diversi 'standard'):
- $(+/-)10^{-38} - (+/-) 10^{+38}$
- $10^{+38} \approx (2^{127})$;



- Con un Sistema posizionale a base r ($r > 0$) la rappresentazione di un numero positivo N con n cifre intere e m frazionarie è':
- $N = d_{n-1} d_{n-2} \dots d_1 d_0 \cdot d_{-1} d_{-2} \dots d_{-m}$
- con $d_i \in \{0, 1, 2, \dots, r-1\}$
- mentre il suo valore è':
- Valore $V = \sum_{i=0}^{n-1} d_i \cdot r^i + \sum_{i=1}^m d_{-i} \cdot r^{-i}$ $N = (-1)^s \cdot M \cdot r^E$ dove r è la base; per noi $r=2$
- il numero con la virgola resta definito dalla tripla di numeri (S, M, E)
- Bisogna fissare il posto standard del punto decimale che, in **forma normalizzata**, si stabilisce stia a sinistra della prima cifra significativa diversa da zero:
- $9.08_{10} = (-1)^0 \cdot 0.5675_{10} \cdot 2^4$
- $S \in \{0, 1\} \Rightarrow$ Segno del numero, dato da $(-1)^s$;
- M = valore assoluto;
- E = Esponente di r con segno.
- $9.08 \Rightarrow S=0; M=0.5675; E=+4 \Rightarrow (0, 0.5675, +4)$

CODIFICA DELLE INFORMAZIONI - Floating Point

Per trasformare la parte decimale si moltiplica ripetutamente x 2 la parte decimale residua (colonna a sin) e si considera l'unità risultante, sottraendo 1 ogni volta che si supera l'unità:

	sottraggo 1 quando supera l'unità	cifra binaria=cifra a sin della virgola
0,5675		
1,135	0,135	1
0,27		0
0,54		0
1,08	0,08	1
0,16		0
0,32		0
0,64		0
1,28	0,28	1
0,56		0
1,12	0,12	1
0,24		0
0,48		0
0,96		0
1,92		1
...		...

Sequenza binaria corrispondente al decimale 0,5675:
10010001010001.... e avanti così finchè ci sono bit disponibili..

Così 9.08 diventa:

0 0100 10010001010001...

S E M

1 bit per S, 8 bit per E, 23 bit per M

CODIFICA DELLE INFORMAZIONI - Immagini

- Formati per immagini (.jpg, .tif, .raw, .gif, ecc.), compressione con perdita e senza perdita

Dimensioni (in pixel) e profondità (#bit per pixel)

- B/N
- Colori

- Formati per sequenze di immagini/filmati (.mov, .mp4, .avi, ecc.)

ALGORITMI E LINGUAGGI DI PROGRAMMAZIONE

ALGORITMO

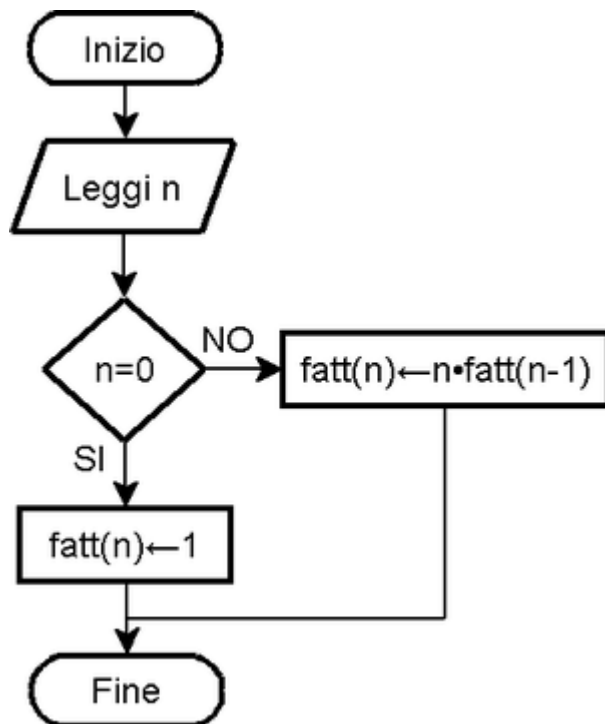
da Al-Kuwarizmi, matematico Persiano (800 d.C.)

- Sequenza di istruzioni non ambigue per la risoluzione di un problema in un numero di passi finito => 'ricetta'
- Un algoritmo scritto per l'elaboratore è un programma
- Rappresentazione mediante:
 - diagrammi di flusso- - flow chart
 - Pseudo-linguaggi

... E traduzione in un linguaggio comprensibile per una CPU

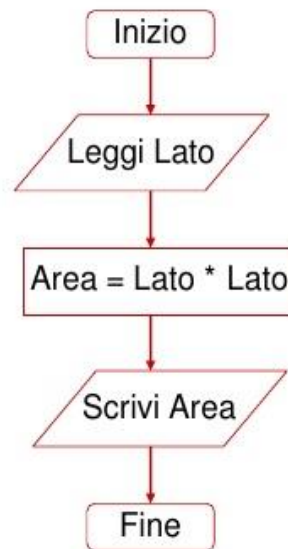
ALGORITMI E LINGUAGGI DI PROGRAMMAZIONE

Diagramma di flusso



Esempio di pseudo linguaggio

- Calcolare l'area di un quadrato dato il lato.



PSEUDO LINGUAGGIO

```
PROGRAMMA Area
INIZIO
  Leggi (Lato)
  Area ← Lato*Lato
  Scrivi (Area)
FINE
```

ALGORITMI E LINGUAGGI DI PROGRAMMAZIONE

A basso livello ('codice macchina')

- **MACCHINA**
- **ASSEMBLER**

Ad alto livello (Teoria dei linguaggi formali – linguaggi non ambigui):

- **FORTRAN** (traduttore di formule) 1957 (IBM)
- **COBOL** (Common Business Oriented language) (1959)
- **BASIC (1964)**
- **Linguaggi 'strutturati':**
 - **PASCAL (1970)**
 - **C (1972), C++ (1983, ultimo standard del 2017) 'ad oggetti'**
 - **JAVA (1995)**
 - **HTML (1993)**
 - **MATLAB (1984)**
 - => (producono 'codice sorgente')

ALGORITMI E LINGUAGGI DI PROGRAMMAZIONE

INTERPRETE (per linguaggi interpretati, BASIC):

Verifica la correttezza sintattica di una riga alla volta, la traduce in linguaggio macchina e la esegue

Lenta nell'esecuzione delle ripetizioni

PER LINGUAGGI COMPILATI (FORTRAN, PASCAL, C, C++,...)

(Editor) Sorgenti

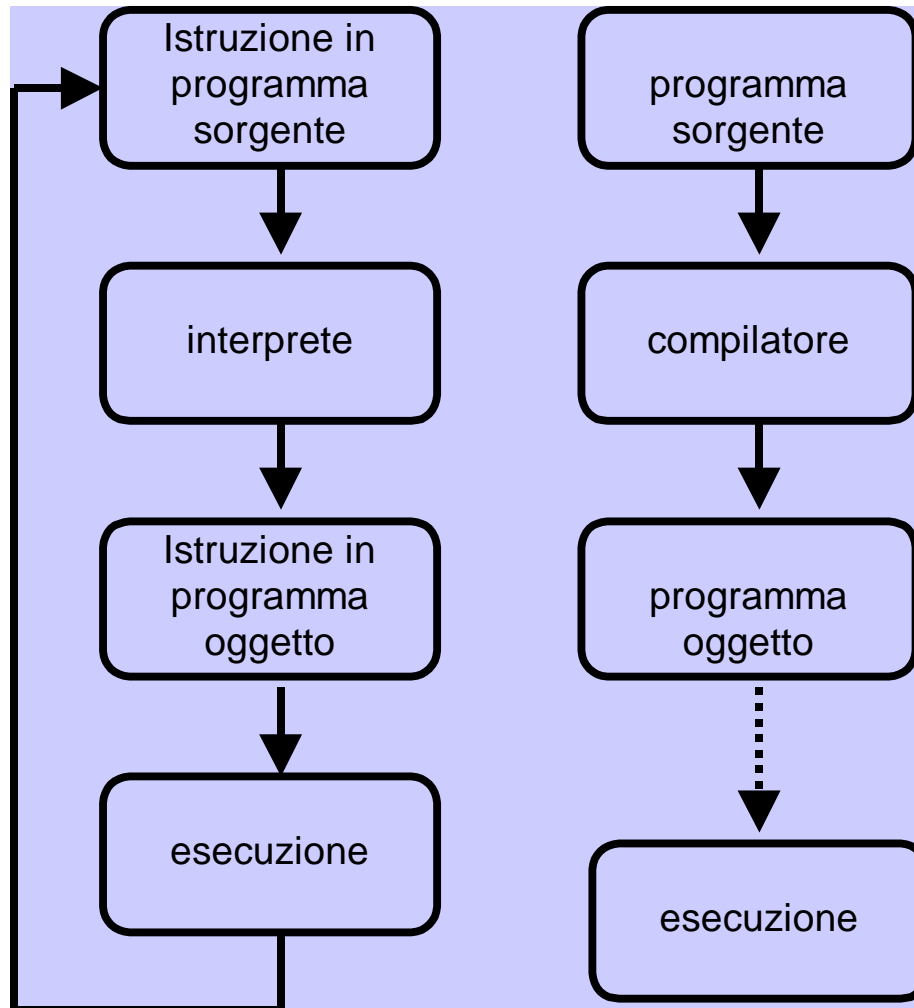
(Compiler) Oggetti + Librerie compilate

=> (Linker) Eseguibili

Il compilatore verifica dapprima la correttezza sintattica e poi traduce tutto il codice sorgente in oggetto

Alcuni linguaggi come java, traducono in un metalinguaggio (bytecode) indipendente dalla CPU e poi una virtual machine lo traduce ed esegue

ALGORITMI E LINGUAGGI DI PROGRAMMAZIONE

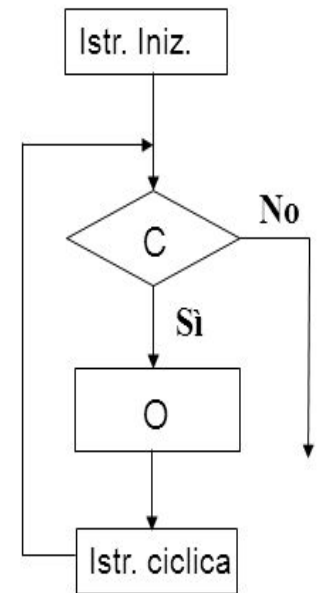
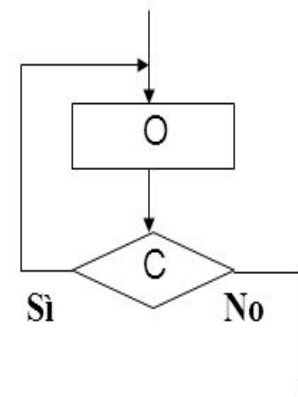
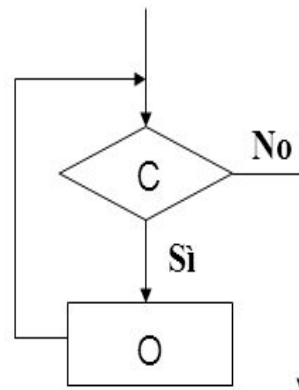
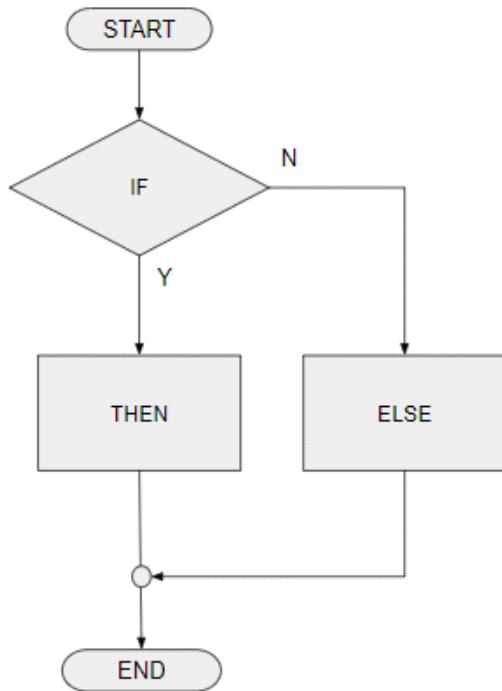


CENNI DI PROGRAMMAZIONE

PROGRAMMA = ISTRUZIONI + DATI

STRUTTURE DI PROGRAMMA = ISTRUZIONI:

Assegnazione, condizione, ripetizione



chiamate a sottoprogrammi (funzioni/subroutine)

CENNI DI PROGRAMMAZIONE

PROGRAMMA = ISTRUZIONI + DATI

STRUTTURE DATI ('VARIABILI' E 'COSTANTI'):

semplici: numeri interi, decimali, caratteri

strutturate:

array (insieme indicizzato di dati della medesima struttura)

record (insieme di dati di strutture non necessariamente identiche)

puntatori (indirizzi di aree di memoria utili per realizzare strutture come alberi, liste, e anche più complesse)

LE VARIABILI POSSONO ESSERE statiche o dinamiche

Espressioni logiche/numeriche, funzioni,...

Semplice esempio in MATLAB...

SOFTWARE DI BASE E APPLICATIVI

IL SOFTWARE DI BASE è composto da programmi di utilità di cui tipici esempi sono:

- editor;
- interpreti;
- compilatori;
- assemblatori

da Strumenti Software indispensabili per costruire programmi eseguibili (ambienti di sviluppo)

e dal Sistema Operativo

IL SOFTWARE APPLICATIVO è costituito da qualsiasi programma che non faccia parte del SW di BASE, p.es. i Browser, OFFICE, MATLAB, ...

Semplice esempio in EXCEL o in WORD....

SISTEMA OPERATIVO

- è un insieme di programmi che **governa** il funzionamento di un calcolatore rendendo la gestione delle sue risorse **trasparente per l'utilizzatore** mettendo a disposizione dell'utilizzatore una macchina **virtuale**, semplice da usare (friendly) in quanto risponde direttamente ai comandi-utente agendo da intermediario tra l'utente e le risorse del computer
- fornisce l'ambiente di lavoro e l'interfaccia utente per inviare i comandi al software di sistema
- gestisce tutte le risorse disponibili:
 - ✓ l'accesso ai files (file system) su disco o da/per qualsiasi dispositivo periferico
 - ✓ l'accesso dei processi/task alla CPU (scheduler)
 - ✓ la memoria centrale

SISTEMA OPERATIVO - STRATIFICAZIONE

- (Hardware)
- **kernel**: è il nucleo del sistema operativo. Interfaccia verso l'hardware. Esegue le funzioni di base
- **File system**: gestisce l'archiviazione/trasferimento dei file
- **Shell**: conchiglia, guscio che interfaccia l'utente; è l'interprete di comandi, eventualmente sotto forma grafica
- **Tramite l'interfaccia utente è possibile 'lanciare' / eseguire:**
 - programmi di utilità di base
 - programmi applicativi

SISTEMI OPERATIVI – famiglie/evoluzione

Tutte le macchine della serie 360 di IBM (fino a 1MByte, 5.1MHz, 1.900.000\$), dal 1965 al 1978, utilizzavano lo stesso software e lo stesso sistema operativo (**DOS/360**) consentendo l'evoluzione dei sistemi verso soluzioni più potenti senza cambiare il software

DOS = Disk Operating System, SO residente su disco, da trasferire in memoria

UNIX dal 1970 => **LINUX** dal 1991 a righe di comando....

Sistemi monoutente e monoprogramma (batch) ora.... multi utente e multi task (interattivi). La CPU lavora sempre in maniera sequenziale ma gestisce i processi in time-sharing (i processi sembrano in parallelo)

SO più usati per computer desktop (a finestre), multi tasking:

Microsoft Windows, MacOS, UNIX/Linux

Sistemi operativi “real_time”: **QNX**, con tempo di risposta garantito per gestire eventi esterni

SO per dispositivi mobile: iOS, Android, Windows Phone,

SISTEMI OPERATIVI – Processi/Tasks

In esecuzione (stato attivo) il processo è caratterizzato dalla transizione tra 3 stati (ciclici)

- READY : il processo è pronto per essere attivato ma essendoci più processi che processori deve attendere il suo turno
 - RUN : il processo è assegnato al processore ed è in esecuzione
 - WAIT : il processo sta aspettando un evento
 - READY
 - COMPLETE : Il processo è stato eseguito
-
- I processi hanno una gerarchia (padre figlio)
 - Ogni processo interferisce con altri processi
 - intenzionalmente (scambio di dati)
 - non intenzionalmente (competizione per le risorse)
 - I processi devono essere identificabili ed hanno un nome, una relazione con altri processi (padre figlio), hanno un proprietario, un gruppo ed una modalità di esecuzione.....

Esempio: Cntr-Alt-Canc

SISTEMI OPERATIVI – Bootstrapping

- All' accensione: RAM è 'vuota'
- La ROM contiene:
 - il "caricatore" (BOOTSTRAP),
 - i programmi diagnostici per la verifica dell' Hardware e delle periferiche di I/O (BIOS= Basic Input Output System)
- Il contenuto della ROM viene caricato nella RAM
- il BOOTSTRAP (che ora è in RAM) può caricare il DOS da un'area prescelta del disco nella zona convenzionale della RAM
- *NB. un programma per essere **eseguito** deve stare in RAM !*

ESERCIZI

- Convertire numeri interi da una base ad un'altra
- Convertire numeri decimali con la virgola in binario
- Sviluppare algoritmi per risolvere semplici problemi:
 - sommare la serie $1 - 1/2 + 1/3 - \dots + 1/9999 - 1/10000$ in 4 modi diversi e confrontare i risultati
 - Calcolare il 35-esimo elemento della Serie di Fibonacci
 - Calcolare la posizione dell'elemento della serie di Fibonacci > 30000
 - sviluppare un simulatore di una calcolatrice tascabile
 - fornire la soluzione (approssimata) di equazioni polinomiali di grado n col metodo di Newton
 - ecc.

APPROFONDIMENTO: LA MACCHINA DI TURING

La macchina di Turing è un particolare tipo di automa formato da:

- 1- Una memoria costituita da un nastro illimitato diviso in celle**
- 2- Una testina mobile di lettura-scrittura/cancellazione che può spostarsi di una cella nelle due direzioni e scrivere 1 o 0 nella cella sottostante**
- 3- Un numero finito di 'stati' interni**
- 4- Una tabella (costituente il programma o algoritmo) mediante la quale, data una coppia 'stato interno - simbolo letto' viene o cambiato lo stato o eseguita un'azione elaborativa (di lettura/scrittura)**

L'idea di base su cui si fonda la macchina è che essa deve leggere i dati in ingresso, e produrne in uscita degli altri. Va pertanto scelto un insieme finito di simboli $A=[a_1, a_2, \dots, a_n]$, che costituisce l'alfabeto della macchina.

La testina legge il simbolo presente nella cella e successivamente lo cancella sostituendolo con un nuovo simbolo, (eventualmente lo stesso). Essa è inoltre dotata di una memoria interna costituita da un certo numero finito di stati interni S tra i quali si possono individuare lo stato iniziale, che è convenzionalmente quello in cui si trova la macchina di Turing quando viene avviata, e un sottoinsieme F di S costituito dai cosiddetti stati finali o di accettazione, in corrispondenza di uno dei quali si arresterà e la stringa presente sul nastro in quel momento è il risultato del calcolo.

APPROFONDIMENTO: ELEMENTI DELLA CPU

Program counter (PC) scandisce la sequenzialità delle operazioni, ha dimensioni pari al bus indirizzi, contiene l'indirizzo di memoria della prossima istruzione da eseguire, viene aggiornato durante l'esecuzione dell'istruzione. Il PC viene salvato nello stack quando si ha un salto (subroutine).

Stack Pointer (SP) contiene l'indirizzo di memoria di una cella dello stack che è una zona di memoria organizzata come una pila (LIFO Last In First Out) dove vengono salvati gli indirizzi delle istruzioni in caso di salto o interruzione.

Registro Istruzioni (IR) memorizza l'istruzione da eseguire.

Registro indirizzi (AR) mantiene l'indirizzo dell'istruzione fino alla fine dell'esecuzione dell'istruzione stessa.

Registro di stato (FR=Flag Register) è composto da celle (flags) che forniscono informazioni sullo stato del MicroP: carry, zero, overflow, segno, interrupt.

ALU (unità logico-aritmetica), somme, rotazioni e scorrimenti, op. logiche.

Logica di controllo fornisce al micro tutti i segnali necessari al funzionamento del micro e all'esecuzione della particolare istruzione. Controlla anche il registro di stato e genera il clock interno.

I registri di uso generale possono, ad esempio, servire a velocizzare operazioni complesse

APPROFONDIMENTO: POLLING

Il ***Polling*** è una tecnica sincrona di gestione delle periferiche

Dopo aver riconosciuto la presenza dell'interrupt la CPU controlla ciclicamente (ciclo di polling) quale dispositivo ha richiesto l'interrupt e lo serve

Il ciclo di polling introduce dei tempi morti
Se ho più richieste contemporanee queste vengono servite sequenzialmente secondo la scansione del ciclo

