

Sistemi Elettronici e informatici in ambito
Radiologico // Informatica Medica – parte b
A.A. 2014-15

Agostino ACCARDO

Dipartimento di Ingegneria e Architettura 040-5587148

accardo@units.it

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
0	0	000	-	NUL
1	1	001	-	SOH
2	2	002	-	STX
3	3	003	-	ETX
4	4	004	-	EOT
5	5	005	-	ENQ
6	6	006	-	ACK
7	7	007	-	BEL
8	8	010	-	BS
9	9	011	-	TAB
10	A	012	-	LF
11	B	013	-	VT
12	C	014	-	FF
13	D	015	-	CR
14	E	016	-	SO
15	F	017	-	SI
16	10	020	-	DLE
17	11	021	-	DC1
18	12	022	-	DC2
19	13	023	-	DC3
20	14	024	-	DC4
21	15	025	-	NAK
22	16	026	-	SYN
23	17	027	-	ETB
24	18	030	-	CAN
25	19	031	-	EM
26	1A	032	-	SUB
27	1B	033	-	ESC
28	1C	034	-	FS
29	1D	035	-	GS
30	1E	036	-	RS
31	1F	037	-	US
32	20	040	 	Space
33	21	041	!	!
34	22	042	"	"
35	23	043	#	#
36	24	044	$	\$
37	25	045	%	%
38	26	046	&	&
39	27	047	'	'
40	28	050	((
41	29	051))
42	2A	052	*	*
43	2B	053	+	+
44	2C	054	,	,
45	2D	055	-	-
46	2E	056	.	.
47	2F	057	/	/
48	30	060	0	0
49	31	061	1	1
50	32	062	2	2
51	33	063	3	3
52	34	064	4	4
53	35	065	5	5
54	36	066	6	6
55	37	067	7	7
56	38	070	8	8
57	39	071	9	9
58	3A	072	:	:
59	3B	073	;	;
60	3C	074	<	<
61	3D	075	=	=
62	3E	076	>	>
63	3F	077	?	?

DEC	HEX	OCT	HTML	CHR
64	40	100	@	@
65	41	101	A	A
66	42	102	B	B
67	43	103	C	C
68	44	104	D	D
69	45	105	E	E
70	46	106	F	F
71	47	107	G	G
72	48	110	H	H
73	49	111	I	I
74	4A	112	J	J
75	4B	113	K	K
76	4C	114	L	L
77	4D	115	M	M
78	4E	116	N	N
79	4F	117	O	O
80	50	120	P	P
81	51	121	Q	Q
82	52	122	R	R
83	53	123	S	S
84	54	124	T	T
85	55	125	U	U
86	56	126	V	V
87	57	127	W	W
88	58	130	X	X
89	59	131	Y	Y
90	5A	132	Z	Z
91	5B	133	[[
92	5C	134	\	\
93	5D	135]]
94	5E	136	^	^
95	5F	137	_	_
96	60	140	`	`
97	61	141	a	a
98	62	142	b	b
99	63	143	c	c
100	64	144	d	d
101	65	145	e	e
102	66	146	f	f
103	67	147	g	g
104	68	150	h	h
105	69	151	i	i
106	6A	152	j	j
107	6B	153	k	k
108	6C	154	l	l
109	6D	155	m	m
110	6E	156	n	n
111	6F	157	o	o
112	70	160	p	p
113	71	161	q	q
114	72	162	r	r
115	73	163	s	s
116	74	164	t	t
117	75	165	u	u
118	76	166	v	v
119	77	167	w	w
120	78	170	x	x
121	79	171	y	y
122	7A	172	z	z
123	7B	173	{	{
124	7C	174	|	
125	7D	175	}	}
126	7E	176	~	~
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	ˆ	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	š	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	ı	205	Í	249	ù
162	ϕ	206	Î	250	ú
163	£	207	Ï	251	û
164	¤	208	Ð	252	ü
165	¥	209	Ñ	253	ý
166	ı	210	Ò	254	þ
167	§	211	Ó	255	ÿ
168	ˆ	212	Ô		
169	©	213	Õ		
170	ª	214	Ö		
171	«	215	×		

CODIFICA DELLE INFORMAZIONI

- Esistono centinaia di sistemi di codifica, preesistenti a Unicode, e ognuno di questi abbina i numeri ai caratteri in modo differente. Nessuna di queste codifiche comprende un numero di caratteri sufficiente per tutte le circostanze. Per le sole lingue dell'Unione Europea, ad esempio, è necessario utilizzare parecchi sistemi di codifica distinti. Anche considerando una sola lingua, come l'italiano, non esiste una codifica unica che comprenda tutte le lettere e tutti i segni di punteggiatura e simboli tecnici di uso comune.
- Questi sistemi di codifica, inoltre, sono in contraddizione l'uno con l'altro. Succede che due codifiche utilizzino lo stesso numero per due caratteri *diversi* o che, viceversa, adottino numeri diversi per lo *stesso* carattere. Qualsiasi elaboratore, e a maggior ragione un server di rete, ha bisogno di utilizzare codifiche diverse. Il problema è che, quando i dati passano da una codifica a un'altra, o da una piattaforma a un'altra, si corre il serio rischio di non riuscire a decodificare le informazioni.
- Dal 2004 si decide di lasciare l'ISO 8859 e si passa a
UNICODE

UNICODE

- Unicode assegna un numero (16bit=2Byte, ora a 32bit) univoco a ogni carattere, indipendentemente dalla piattaforma, indipendentemente dall'applicazione, indipendentemente dalla lingua.
- Inizialmente Unicode conteneva gli alfabeti principali delle maggiori lingue vive, compresi gli ideogrammi poi anche alfabeti meno conosciuti
- La versione 5.0 di Unicode (luglio 2006), include oltre 99.000 simboli tra cui anche una gran varietà di simboli matematici, più che sufficienti a permettere la trascrizione di complesse formule.
- Unicode è alla base di molti moderni standard, come XML, Java, ECMAScript (JavaScript), LDAP, CORBA 3.0, WML ...
- costituisce l'implementazione ufficiale dello standard internazionale ISO/IEC 10646. Unicode è supportato da molti sistemi operativi e da tutti i più moderni web browser

CODIFICA DELLE INFORMAZIONI

NUMERI INTERI:

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

FIXED POINT

SENZA SEGNO (IN COMPLEMENTO A 2; IN OFFSET BINARIO)

NUMERO BYTE PER CODIFICARE IL NUMERO LEGATO ALLE CIFRE
SIGNIFICATIVE

ESEMPI: DI COMPLEMENTO A 1 E A 2

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

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

Trasformazione decimale-binaria e viceversa

- Conversione di interi **da base 2 a base 10**: somma dei prodotti tra i bit e le corrispondenti potenze di 2;
- Es. $0001\ 0101\ 0111_2 = 2^8+2^6+2^4+2^2+2^1+2^0 = 343_{10} = 1 \cdot 16^2 + 5 \cdot 16^1 + 7 \cdot 16^0$
- **Simboli in base 16: 0,1..9,A,B,C,D,E,F**
- Rappresentazione e corrispondenza esadecimale:
- | 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 |
- $\mathbf{FF}_{16} = 15 \cdot 16^1 + 15 \cdot 16^0 = 255_{10} = \text{MAX INTERO IN UN BYTE} = 1 \cdot 16^2 - 1 = 256_{10} - 1 = 100_{16} - 1.$

CODIFICA DELLE INFORMAZIONI

OPERAZIONI (SOMMA-PRODOTTO -SHIFT), esempi... + esempio che mostra errori dovuti al nr limitato di bit utilizzati per la codifica)

esempio con 3 cifre significative:

(1) 12,7 +

(2) 3,28 +

(3) 4,05 =

se (1)+(2) e poi + (3) = 19,9; se (2)+(3) e poi + (1) = 20,0

OVERFLOW

1011+

0110=

10001

CODIFICA DELLE INFORMAZIONI

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:
- $(+/-)10^{-38} - (+/-) 10^{+38}$
- $10^{+38} \approx (2^{127})$;
- Con un Sistema posizionale a base r ($r > 0$) la rappresentazione di un numero positivo N con n cifre intere e m frazionarie e':
- $N = d_{n-1} d_{n-2} \dots d_1 d_0 \cdot d_{-1} d_{-2} \dots d_{-m}$
- con $d_i \in \{0, 1, 2, \dots, r-1\}$
- mentre il suo valore e':
- $$\text{Valore } N = \sum_{i=0}^{n-1} d_i \cdot r^i + \sum_{i=1}^m d_{-i} \cdot r^{-i} \quad N = (-1)^s \cdot M \cdot r^E \quad \text{dove } r \text{ e' la base; per noi } r=2$$
- 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.
- $9.08 \Rightarrow S=0; M=0.5675; E=+4 \Rightarrow (0, 0.5675, +1)$

Codifica floating-point normalizzata

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
...		...

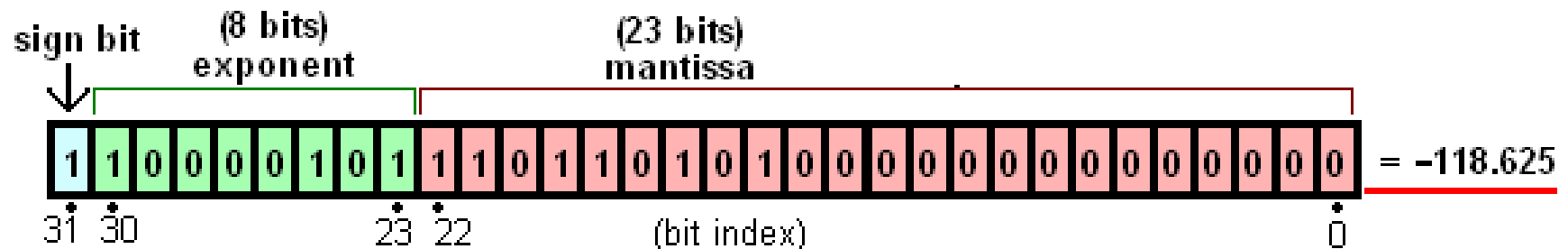
Sequenza binaria
corrispondente al decimale
0,5675:
10010001010001.... e
avanti così finchè ci sono
bit disponibili..

Codifica floating-point normalizzata (altro esempio)

- tutto in binario con: 1 bit riservato a S, X bit riservati a M, e Y bit riservati a E
(**esponente della base 2**) esprimibile in modulo e segno
- Es.: $-1/8$ (= -0.125)
- $S=1$, $M=0.125_{10} = 0*2^{-1}+0*2^{-2}+1*2^{-3}+0*2^{-4} +0*2^{-5} +0*2^{-6}$ ecc. $=1000_2 * 2^{-2}$.
- $E= -2 = 1\dots110_2$
- Es 8 bit: 4 bit per M, 3 bit per E ed 1 per S
- $1\ 1000\ 110_2$
 - Trasformazione parte decimale => binario: $1/8=0.125_{10}=0.001_2$
Si moltiplica ripetutamente x 2 la parte decimale e si considera l'unità risultante:
 - $0.125*2=>0.250*2=>0.500*2=>1.000$
- Nei Computer viene anche implementata la notazione secondo lo standard IEEE 754
- 32 bit: 1 S 8 E 23 M
 - l'esponente non e` in modulo e segno ma sempre positivo con offset 127 la mantissa non e` data dalle cifre dopo lo zero virgola ma dopo l'uno virgola: il primo uno e` implicito e cosi` si rappresentano mantisse a 24 bit con soli 23
- Lo standard specifica anche i formati con numero di bit diversi (es. 64 bit)

ESEMPIO

- Codificare -118.625 con IEEE 754.
- È negativo: segno "1"
- Il numero senza segno in binario è: $1110110,101$ (da 2^6 a 2^{-3})
- Sposto la virgola lasciando solo un "1" a sinistra della virgola:
 $1110110.101 = 1.110110101 \times 2^6$. ottenendo un numero a virgola mobile
- La mantissa è a destra della virgola (uso 23 bit riempiendo di "0":
 11011010100000000000000).
- L'esponente 6 va convertito in offset binario ovvero codifico
 $6+127=133$, in binario: $10000101 = 128+5 = 133$.



CODIFICA DELLE INFORMAZIONI

Immagini

- Formati per immagini (.jpg, .tif, .raw, .gif, ecc.)
Dimensioni (in pixel) e profondità (#bit per pixel)
- B/N
- Colori
- Radiologiche (DICOM)
- “commerciali”

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
- Un algoritmo scritto per l'elaboratore e` un programma (non necessariamente viceversa).
- Rappresentazione mediante:
 - diagrammi di flusso- - flow chart
 - Pseudo-linguaggi ... esempi vari ...

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

Software di base

- E' composto da programmi di utilità di cui tipici esempi sono:
- editor;
- interpreti;
- compilatori;
- assembleri
-
- **Strumenti Software** indispensabili per costruire programmi eseguibili (sviluppo)

LINGUAGGI

A basso livello

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

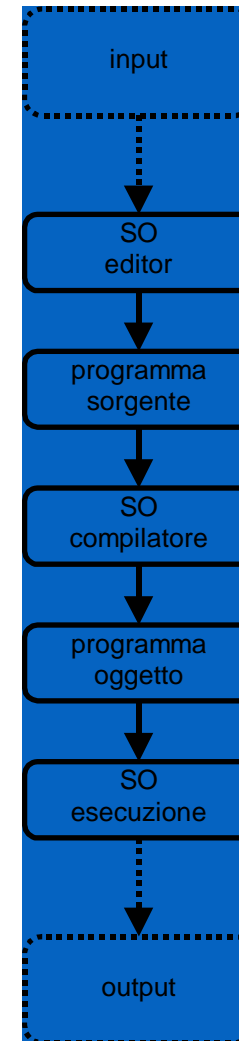
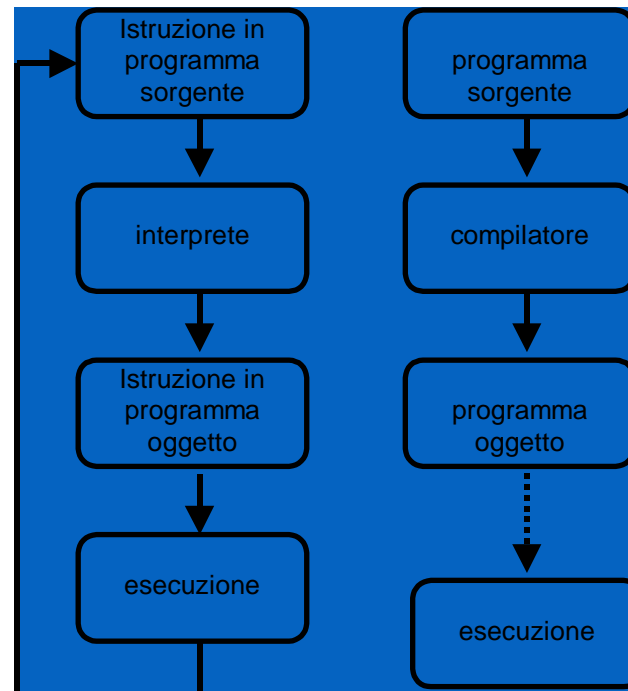
Ad alto livello:

- **FORTRAN** (traduttore di formule) 1957 (IBM)
- **COBOL** (Common Business Oriented language) (anni 60)
- PASCAL
- C
- JAVA
- MATLAB
-

LINGUAGGI AD ALTO LIVELLO

- Compilati
- Interpretati
- (Editor) Sorgenti – Oggetti – Linker - Eseguibili

- Per la macchina: solo codice binario (codice binario, oggetto - eseguibile)
- Il programmatore scrive un programma in un linguaggio di programmazione e produce un codice sorgente (normalmente ASCII) - Assembler, Basic, Fortran, Cobol, Pascal, C, Java, html, matlab, pv-wave, spss,.....
- Traduzione sorgente-binario Interprete o compilatore (o entrambi interpiler)
- l'interprete è sempre residente in memoria e quindi rende oltremodo inefficiente l'esecuzione.



S.O.

- Tutte le macchine della serie 360 di IBM utilizzavano lo stesso software e lo stesso sistema operativo (OS/360)
- consentiva di far evolvere i sistemi verso soluzioni più potenti senza cambiare il software

- Sistemi operativi multi-utente
- CTSS (1965)
- Multics (1969) General Electric, MIT, Bell Lab.
- In competizione con Multics il sottogruppo dei Bell lab. crea UNIX (gioco di parole per prendere in giro Multics)
 - Thompson-Ritchie +Kernighan (creatore del C '70)
- CMAS (Cambridge Multiple Access System)
 - realizzato in quegli anni in Europa

- Necessità di standardizzare.
 - 1974 il ministero della difesa americano spendeva 3 miliardi di dollari l'anno in software che veniva realizzato con 400 diversi linguaggi di programmazione
- PASCAL, C, C++, ecc.
- Nel 1982 compaiono i linguaggi modulari di alto livello **Modula-2**

Sistema Operativo

- e' un insieme di programmi specializzato nel **governare** il funzionamento di un calcolatore rendendo la gestione delle sue risorse trasparente per l'utilizzatore;
- mette cosi' a disposizione dell'utilizzatore una macchina **virtuale** non esistente, ma più semplice da usare (friendly) in quanto risponde ai comandi-utente;
- il processo di virtualizzazione si propaga ad ogni strato aggiuntivo di software.
- è un programma che controlla il computer, migliora l'efficienza e diminuisce i costi dell'uso del computer
- agisce da intermediario tra il computer e l'utente o i programmi
- un utente generalmente sottopone un lavoro JOB al SO
- un JOB può essere suddiviso in unità sequenziali o parallele
- il processo (TASK) è un'unità che può essere svolta simultaneamente ad altri processi
- Il SO gestisce i processi con lo Scheduler
- l'ADDRESS SPACE è l'insieme di programmi dati e I/O accessibili ad un processo

SO

- Software che consente di gestire la macchina :
- gestisce il flusso di dati ed avvia e termina processi.
- consente al software di arrivare all'HW ed al SW della macchina.
- fornisce l'ambiente di lavoro (comandi) e l'interfaccia (utente) per inviare i comandi al software di sistema (altre interfacce sono l'interfaccia HW e quella di programmazione).
- Gestisce la CPU, gli accessi ai dischi, la memoria, i dispositivi di I/O, fornisce sottoprogrammi di utilità, gestisce la condivisione delle risorse.

Sistema operativo

- **Facilità d'uso**
- Il SO crea un sistema virtuale con modalità d'uso più semplici di quelle del sistema reale.
- **Ottimizzazioni delle prestazioni**
- massimizzare il lavoro complessivo che il sistema è in grado di fare in un certo tempo (Throughput: quantità di lavoro (n. programmi) fatta nell'unità di tempo). Utilizzata principalmente per lavori in BATCH
- effettuare un lavoro nel minor tempo possibile. Utilizzato principalmente per lavori in TIME-SHARING
- Il programma di controllo della macchina virtuale si connette all'HW e gestisce la multiprogrammazione fornendo a livello superiore più macchine virtuali.

Stratificazione SW

- (Hardware)
- **S.O.: kernel:** e' il nucleo del sistema operativo. Interfaccia verso l'hardware. Esegue le funzioni di base
- **S.O.: File system:** gestisce l'archiviazione dei file e l'organizzazione logica dei dischi
- **S.O.: Shell:** conchiglia, guscio che interfaccia l'utente; e' sostanzialmente un interprete di comandi
 - con eventuale interfaccia grafica;
- S.O.: interfaccia utente
- programmi di utilità di base
- programmi applicativi

Alcuni (famiglie di) S.O.

- DOS = Disk Operating System
- Windows
- SO per Apple / Macintosh – MacOS
- Linux

- Unix
- VMS x macchine Digital

Evoluzione dei S.O.

- Da Sistemi *Batch* a *interattivi*;
- mono-utente mono-programmato; CP/M, DOS
- “ “ multi-programmato; Windows 3.x
- molti utenti contemporanei;
multi_programmato o multi_tasking
Windows NT,...
MacOs...
Unix ...VMS
- Sistemi operativi “real_time” (QNX)
 - tempo di risposta garantito per gestire eventi esterni.

S.O. mono & multiprogrammato:

- I Sistemi Operativi sono mono o multi task cioè tali da poter eseguire un solo (mono) o più programmi in **apparente** contemporaneità.
- S.O. multi task sono suddivisi in multiutenti e monoutente
- I S.O. multi task e monoutente danno all'utente la possibilità di effettuare più funzioni;
- Nei S.O. multi task e multiutenti il meccanismo può essere applicato a + utenti in **apparente** contemporaneità.
 - La CPU lavora sempre in maniera sequenziale ma gestisce i processi in time-sharing (i processi sembrano in parallelo)

S.O. multi - utente e multi_tasking

- Funzioni:
 - eseguire i comandi utenti;
 - gestire Input/output e file;
 - segnalare errori;
 - assegnare la CPU ad ogni programma-utente per un tempuscolo con apparente contemporaneità di esecuzione (Scheduler);
 - fornire i servizi richiesti dalla rete;
 - assicurare sicurezza e privacy
 - informazioni del singolo utente e del sistema ...

Processi- Task

- 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

Processi - Task

- I processi hanno una gerarchia (padre figlio)
- Ogni processo interferisce con altri processi
 - intenzionalmente (scambio di dati)
 - non intenzionalmente (competizione per le risorse)
- Il SO deve tenere sotto controllo l'attività di tutti i processi per :
 - condizionarne gli scambi
 - regolare la competizione nell'uso delle risorse
- I processi si scambiano segnali fra loro e con il SO, mentre gli I/O mandano IRQ
- 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.....

Kernel

- Kernel Mode modalità di esecuzione del codice da parte di un microprocessore in cui tutta la memoria è accessibile e tutte le istruzioni della CPU possono venire eseguite. Poiché il SO ha una priorità superiore a quello degli applicativi, il microprocessore riesce a tenere sotto controllo tutti i processi attivi. Se infatti il SO potesse venire interrotto l'applicazione che guadagnasse il controllo di una risorsa potrebbe non la rilasciarla più in quanto all'interno del sistema non esiste un altro componente in grado di coordinare le temporizzazioni.

Fasi di bootstrapping

- All' accensione: RAM e' '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 !*

Organizzazione dei dischi (DOS)

- I file sono alla base dell'organizzazione dei dischi, sono contenitori di byte.
- Struttura ad albero con directory e sottodirectory (cartelle o folder) *path*
- directory principale *root*, in posizione e di dimensioni fisse (in FAT, 1.44 FD, 222 entries; HD 512 entries)
- I file hanno un nome (8 caratteri maiuscoli) ed una estensione (di 3 caratteri) che convenzionalmente ne definisce il tipo. Sono eseguibili dal sistema operativo solo i file con estensioni .exe, .com, .bat.
 - Sicurezza??
- I file hanno degli attributi :
 - (1 byte, 6 bit usati)
- ed altre proprietà :
 - data
 - dimensione

Attribute	Bit Code
Read-Only	00000001
Hidden	00000010
System	00000100
Volume Label	00001000
Directory	00010000
Archive	00100000

Organizzazione dei dischi.

- **FAT** (File allocation Table) anche detta FAT16, contiene le informazioni sull'uso dei cluster
- Inizialmente solo dischi piccoli (32 MB) poi con modifiche 128 MB ed infine 2GB
 - Evolve in FAT32 sotto Windows 95.
- Il file system si appoggia su un certo numero di aree speciali, funzionali alla propria organizzazione, che vengono riservate all'atto della formattazione del disco : (in sequenza, dall'inizio del disco)
 - master boot record
 - partition table
 - file allocation table
 - root directory

Gestione della memoria - Paginazione

- È una tecnica che consente ad una macchina (PC) di comportarsi come se avesse più RAM di quanto non sia fisicamente presente nel sistema. Si ottiene combinando la RAM con una parte del HD. Lo spazio del disco impiegato in questo modo prende il nome di **memoria virtuale**. (costa molto meno un giga di disco che non di RAM).
- La paginazione è una tecnica comune a molti sistemi operativi e si fonda sulla disponibilità di un file di swap che funge da “serbatoio dei dati”.
- La memoria (reale+virtuale) viene divisa in pagine (blocchi di 4 KB) alcuni dei quali saranno residenti in memoria RAM altri su disco. Ognuno di questi blocchi è una pagina logica. I blocchi in RAM sono (anche) pagine fisiche. Il sistema di paginazione gestisce le associazioni fra pagine fisiche e logiche. E quando un programma legge e scrive dalla memoria ha a disposizione tutti i blocchi logici. Nella memoria, oltre al SO, vi è una tabella che serve a sapere dove si trova una pagina logica (mediante un bit (su 32) detto *entry*)

UNIX

- **Unix**
- Prima versione 1969 per PDP-7 e poi per PDP-11.
- Bell-lab poi Università e più tardi diffusione esterna
- Nella terza versione il SO risulta scritto quasi completamente in C risultando così **portabile** (cosa non possibile in assembler). Sono a disposizione i sorgenti del SO, facile correzione ed espansione.

Shell

- E `un programma che esegue comandi
 - script: file di comandi per la shell
- esistono molte shell
- molte delle loro funzioni derivano dalla shell Bourne /bin/sh
- Linux usa una versione migliorata
- Bourne-again: bash
- La finestra della shell può essere definita finestra di terminale, cioè la parte di interfaccia utente che accetta la pressione dei tasti dal sistema e disegna i caratteri sullo schermo e che agisce da intermediario fra l'utente e la shell.

Comandi base

- Ls
- cp
- mv
- touch
 - crea un file o aggiorna la marcatura temporale
- rm
- echo
- cd
- mkdir
- rmdir
- grep
- more e less
 - less simile a more ma più potente less --help
- pwd
- diff
- file
- find
- head e tail
- sort

Autorizzazioni sui file

- Tipo aut. Utente aut. Gruppo altre aut.

- _ _ _ _

- `chmod g+r file`

- `chmod o+r file`

- `chmod go+r file`

- -r per eliminare

- con numeri

- `chmod 751 file rwx r_x __x`