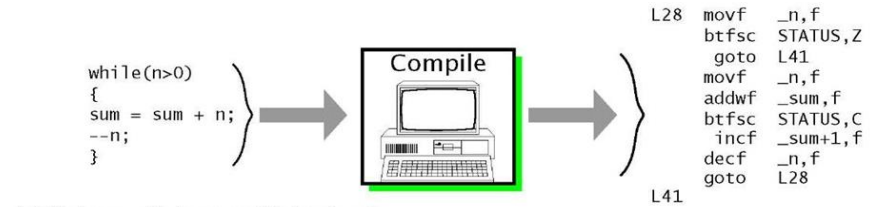
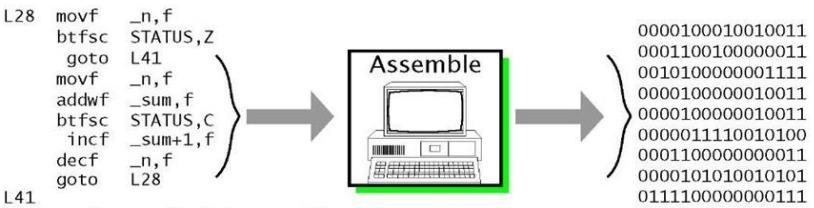
العلاقة بين البرمجيات والمكونات المادية لجهاز الحاسب

البرمجيات Software عبارة عن سلسلة من التعليمات statements التي تخبر الاجهزة المادية Hardware بانجاز مهمة معينة عادةً ما تكون هذه التعليمات مكتوبة بلغة برمجة عالية المستوى((Java, C++, Visual basic, Python, …) يمكن للإنسان التعامل معها بسهولة. ان التعليمات المكتوبة باللغات العليا تعرف بشفرة المصدر Source code ، تعليمات شفرة المصدر لايفهمها الحاسوب، لذا يتم تحويلها الى لغة ابسط تعرف بلغة التجميع (Assembly language) ذات المستوى المتوسط بواسطة برنامج خاص يعرف بالمترجم Compiler. لغة التجميع تتالف من ايعازات مبسطة Instructions يتم تحويل ايعازات لغة التجميع الى لغة الالة Machine language باستخدام برنامج المجمع assembler . لغة الالة هي اللغة التي يفهمها الحاسوب ، وتنفذ ايعازاتها مباشرة في المعالج CPU.تتمثل لغة الالة بصيغة الاعداد الثنائية فقط.



(a) First, compile to assembly-level code.

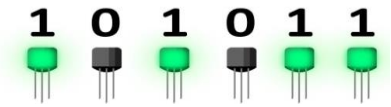


(b) Second, assemble-link to machine code.

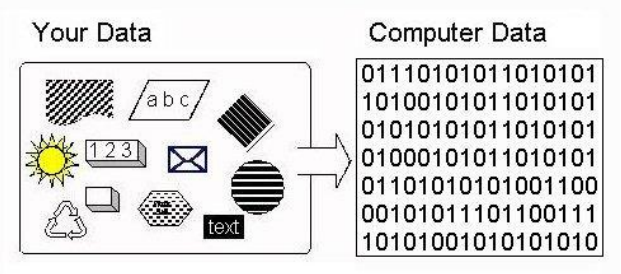
تمثل البيانات في الحاسوب

الحاسوب هو الة رقمية تستخدم لتخزين البيانات و معالجتها، وتتكون من مجموعة كبيرة من الدوائر الالكترونية

- هناك حالتان للدوائرة الالكترونية
  - 0:الدائرة مفتوحة, التيار الكهربائي لا يمر
  - 1: الدائرة مغلقة, التيار الكهربائي يمر



- جميع البيانات data (النصوص والارقام والصور والاصوات ومقاطع الفيديو....) يتم تخزينها بالحاسوب على شكل **ارقام**.



- جميع الارقام تكتب بالصيغة الثنائية (binary numbers)
- تتكون الارقام الثنائية من سلسلة من الخانات bits التي تأخذ اما 0 او1.
- كل ثمان ثنائيات تعرف بالبايت Byte (1 Byte=8-bit)

- الجدول ادناه يمثل وحدات التخزين ومضاعفاتها

Bit	0 or 1	
Byte	8	Bit
KB	1024	Byte
MB	1024	KB
GB	1024	MB
TB	1024	GB

- في النظام الثنائي كل مرتبة لها وزن من مضاعفات 2. حيث المرتبة الاولى وزنها 2^0 اما المرتبة الثانية فوزنها 2^1، الثالثة 2^2 وهكذا.

0	1	1	0	0	1
32	16	8	4	2	1
2^5	2^4	2^3	2^2	2^1	2^0

- لمعرفة العدد المكتوب بالنظام الثنائي يتم جمع الاوزان المقابلة للمراتب التي قيمتها 1 فقط.

0	1	1	0	0	1
32	16	8	4	2	1

16 + 8 + 1 = 25

مثال: حول العدد الثنائي $(100101)_2$ الى الصيغة العشرية (الاعتيادية)

$$100101_2 = [(1) \times 2^5] + [(0) \times 2^4] + [(0) \times 2^3] + [(1) \times 2^2] + [(0) \times 2^1] + [(1) \times 2^0]$$

$$100101_2 = [1 \times 32] + [0 \times 16] + [0 \times 8] + [1 \times 4] + [0 \times 2] + [1 \times 1]$$

$$\mathbf{100101_2 = 37_{10}}$$

- اذا كان عدد الخانات n يمكن بواسطتها ترميز 2^n من الاعداد.التي تتراوح قيمها بين 0-2^{n}-1
- الجدول التالي يوضح ترميز الاعداد المكونة من 4 مراتب في النظام الثنائي.

Decimal	Binary
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

نظام الترميز الموحد أسكي ASCII

نتعامل مع الحاسوب باستعمال الأحرف و الأرقام و الرموز التي تعوّدنا عليها سواء بإدخالها عبر لوحة المفاتيح أو عند عرضها في الشاشة أو الطابعة. يقوم الحاسوب بتخزين كل المعطيات على شكل أعداد صحيحة في النظام الثنائي. ولذلك فمن المهم أن تتفق كافة أجهزة الكمبيوتر على الأرقام التي تمثل فيها الحروف. هذا ما يستدعي استعمال نظام ترميز خاص.استحدث نظام ASCII للحرف اللاتيني الذي يضع لكل حرف او رقم أو رمز إملاءعدد خاص من 32 إلى 127 . كما ان العدد 32 مخصص للـ فراغ Space والعدد 127 مخصص للحذف Delete هو موضح في الجدول ادناه اما الاعداد من 0 الى 31 فهي مخصصة الى رموز التحكم كعرض النص على الشاشة او الطباعة ..... الخ

DEC	BIN	Symbol	DEC	BIN	Symbol	DEC	BIN	Symbol
32	100000		64	1000000	@	97	1100001	a
33	100001	!	65	1000001	A	98	1100010	b
34	100010	"	66	1000010	B	99	1100011	c
35	100011	#	67	1000011	C	100	1100100	d
36	100100	$	68	1000100	D	101	1100101	e
37	100101	%	69	1000101	E	102	1100110	f
38	100110	&	70	1000110	F	103	1100111	g
39	100111	'	71	1000111	G	104	1101000	h
40	101000	(	72	1001000	H	105	1101001	i
41	101001	)	73	1001001	I	106	1101010	j
42	101010	*	74	1001010	J	107	1101011	k
43	101011	+	75	1001011	K	108	1101100	l
44	101100	,	76	1001100	L	109	1101101	m
45	101101	-	77	1001101	M	110	1101110	n
46	101110	.	78	1001110	N	111	1101111	o
47	101111	/	79	1001111	O	112	1110000	p
48	110000	0	80	1010000	P	113	1110001	q
49	110001	1	81	1010001	Q	114	1110010	r
50	110010	2	82	1010010	R	115	1110011	s
51	110011	3	83	1010011	S	116	1110100	t
52	110100	4	84	1010100	T	117	1110101	u
53	110101	5	85	1010101	U	118	1110110	v
54	110110	6	86	1010110	V	119	1110111	w
55	110111	7	87	1010111	W	120	1111000	x
56	111000	8	88	1011000	X	121	1111001	y
57	111001	9	89	1011001	Y	122	1111010	z
58	111010	:	90	1011010	Z	123	1111011	{
59	111011	;	91	1011011	[	124	1111100	|
60	111100	<	92	1011100	\	125	1111101	}
61	111101	=	93	1011101	]	126	1111110	~
62	111110	>	94	1011110	^	127	1111111	
63	111111	?	95	1011111	_			
64	1000000	@	96	1100000	`			