الجامعة السورية الخاصة
SYRIAN PRIVATE UNIVERSITY

وحدة متطلبات الجامعة

مهارات الحاسوب
Computer Skills
2019

المحاضرة السابعة

أنظمة العد
(Numbering Systems)

مقدمة

- قبل اختراع الكتابة كان القدماء يقومون بالعد على أصابع اليد فكانت اليد تمثل الأعداد وعند انتهاء الأصابع كانوا يحتاجون إلى شخص آخر ليقوم بالعد فكان الأول يمثل الآحاد و الثانى يمثل العشرات.
- عند اختراع الكتابة اجتهد العلماء لاختراع منظومة أعداد بدلاً من الاصابع.
- كانت هذة الأعداد هي الأعداد الأساسية وهي (من ٠ حتى ٩) وباضافة الواحد إلى الصفر يتكون العدد ١٠.
- اعتمد العلماء على أنّ كل الأعداد بعد ٩ مزيج من عددين أو أكثر مثل ١٠ و ١٠٠ و ٦٧٣٥.
- وضع قواعد هذه الأنظمة العالم العربي :الخوارزمي
- تقوم فكرة أي نظام عد على مبدأين أساسين:
  - أساس النظام (Base) وهو عدد صحيح موجب.
  - عدد رموز أو مفردات هذا النظام.

تعريف نظام العد:

- **هو طريقة للتعامل مع رسوم الأرقام للتعبيرعن قيمتها وكيفية تطبيق العمليات الحسابية عليها.**

أو

- هو شفرة لها مجموعة من الرموز تستخدم للتعبير عن الأعداد.

أنظمة العد (Numbering systems)

١- النظام العشري (Decimal System)

- يعتبر النظام العشري أكثر أنظمة العد استعمالاً من قبل الإنسان، وقد سمي بالعشري لأنه يتكون من عشرة أرقام هي (٠، ١، ٢، ...، ٩) و التي بدورها تشكل أساس نظام العد العشري.
- بشكل عام، يمكن القول أن أساس أي نظام عد يساوي عدد الأرقام المستعملة لتمثيل الأعداد فيه، وهو يساوي كذلك أكبر رقم في النظام مضافاً إليه واحد.
- تمثل الأعداد في النظام العشري بواسطة قوى الأساس ١٠ وهذه تسمي بدورها **أوزان خانات العدد**، أي

$$10^n,\ldots,10^3,10^2,10^1,10^0\,.\,10^{-1},10^{-2},10^{-3},\ldots,10^{-n}$$

أمثلة: النظام العشري (Decimal System)

الرقم ٩٨٧٦

		آحاد	عشرات	مئات	ألوف
الرقم	=	٦	٧	٨	٩
أوزان المواقع	=	10^0	10^1	10^2	10^3
القيمة	=	١	١٠	١٠٠	١٠٠٠

قيمة العدد = (٦×١) + (٧×١٠) + (٨×١٠٠) + (٩×١٠٠٠)

= ٦ + ٧٠ + ٨٠٠ + ٩٠٠٠

= ٩٨٧٦

يمكن كتابة العدد العشري N=7129.45 على النحو الآتي:

$$N= 7x10^{3}+1x10^{2}+ 2x10^{1}+ 9 \times 10^{0}+ 4x10^{-1}+ 5x10^{-2}$$

٢- النظام الثنائي (Binary System)

- إن الأساس المستعمل في النظام الثنائي هو ٢ ويتكون هذا النظام من رقمين فقط هما ٠ و ١ ويسمى كل منهما رقماً ثنائياً (Binary Digit).
- كما يمكن استخدام أي حالتين مثل "صح" و"خطأ" أو "تشغيل" و "إطفاء".
- لتمثيل كل من الرقمين ٠ و ١ يلزم خانة واحدة فقط، ولهذا السبب أصبح من الشائع إطلاق اسم بت (Bit) على الخانة التي يحتلها الرقم داخل العدد الثنائي.

العدد الثنائي ١٠

الأرقام ٠ ١

أوزان المواقع = 2^0 2^1

قيم الأوزان = ١ ٢

قيمة العدد = (٠×١) + (١×٢)

= ٠ + ٢

= ٢

٢-١- التحويل من النظام الثنائي إلى النظام العشري

- لتحويل أي عدد ثنائي إلى مكافئه العشري فإنه يجب علينا استعمال قانون التمثيل الموضعي للأعداد. و ينطبق هذا القانون عندما يكون الرقم الثنائي صحيحاً أو كسراً مع مراعاة أن أساس نظام العد هنا هو ٢.

$$N = a_n R^n + a_{n-1} R^{n-1} + \cdots + a_1 R^1 + a_0 R^0 + a_{-1} R^{-1} + \cdots + a_{-m} R^{-m}$$

مثال ١ : حول العدد الثنائي التالي إلى مكافئه العشري: $(110011010)_2 = (??)_{10}$

110011010 =

$= 0 * 2^0 + 1 * 2^1 + 0 * 2^2 + 1 * 2^3 + 1 * 2^4 +$

$0 * 2^5 + 0 * 2^6 + 1 * 2^7 + 1 * 2^8$

$= 0 + 2 + 0 + 8 + 16 + 0 + 0 + 128 + 256$

$= 410$

مثال ٢: تحويل العدد الثنائي إلى مكافئه العشري:

100111

$$2^0 \times 1 = 1$$
$$2^1 \times 1 = 2$$
$$2^2 \times 1 = 4$$
$$2^3 \times 0 = 0$$
$$2^4 \times 0 = 0$$
$$2^5 \times 1 = 32$$

\+

39 ← عشري

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

Examples:

$10101_2 = 1\times2^4+0\times2^3+1\times2^2+0\times2^1+1\times2^0 = 16+4+1= 21$ (Decimal)

$10111_2 = 1\times2^4+0\times2^3+1\times2^2+1\times2^1+1\times2^0 = 16+4+2+1= 23$ (Decimal)

$100011_2 = 1\times2^5+0\times2^4+0\times2^3+0\times2^2+1\times2^1+1\times2^0 =32+2+1= 35$ (Decimal)

٢-٢- التحويل من النظام العشري إلى النظام الثنائي

تحويل الأعداد العشرية الصحيحة الموجبة :

لتحويل أي عدد صحيح موجب من النظام العشري إلى الثنائي نستعمل طريقة الباقي (Remainder Method) الموضحة كالآتي:

١. أقسم العدد العشري على الأساس ٢.
٢. أحسب باقي القسمة الذي يكون إما ١ أو ٠.
٣. أقسم ناتج القسمة السابق على الأساس ٢ كما في الخطوة (١)
٤. أحسب باقي القسمة كما في الخطوة (٢)
٥. استمر في عملية القسمة وتحديد الباقي حتى يصبح ناتج القسمة الصحيح صفراً.
٦. العدد الثنائي المطلوب يتكون من أرقام الباقي مقروءة من الباقي الأخير إلى الباقي الأول (لاحظ أن الباقي الأول يمثل LSD (least significant digit) بينما يمثل الباقي الأخير MSD (most significant digit)

أمثلة: التحويل من النظام العشري إلى النظام الثنائي.

مثال ١: حول الرقم ١٢ من النظام العشري إلى النظام الثنائي.

	الباقي	ناتج القسمة	
الخانة الأدنى منزلة LSD	0	$12 \div 2 = 6$	1
	0	$6 \div 2 = 3$	2
	1	$3 \div 2 = 1$	3
الخانة الأعلى منزلة MSD	1	$1 \div 2 = 0$	4
	إنهاء القسمة		

فيكون الناتج (من أسفل إلى أعلى ومن اليسار إلى اليمين):

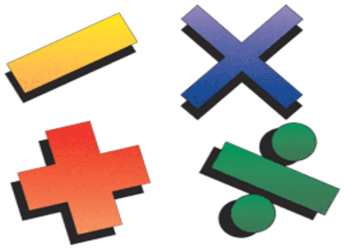
$$(12)_{10} = (1100)_2$$

مثال ٢: حول الرقم ١٥٠من النظام العشري إلى النظام الثنائي.

150	2	0
75	2	1
37	2	1
18	2	0
9	2	1
4	2	0
2	2	0
1		1

$$(150)_{10} = (10010110)_2$$

إجراء العمليات الحسابية على الأعداد الثنائية الموجبة



يمكن إجراء العمليات الحسابية من جمع وطرح وضرب وقسمة كما هو الحال في النظام العشري مع مراعاة أن أساس النظام المستعمل هنا هو 2.

عملية الجمع :

لو أخذنا عددين ثنائيين A ,B وكان كل منهما يتكون من خانة واحدة فقط (Bit)، وبما أن كل خانة يمكن أن تكون أما 0 أو 1 فإنه يوجد للعددين معاً أربع احتمالات كالآتي:

A	B	المجموع	الفيض carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

أما إذا كانت الأعداد الثنائية مكونة من أكثر من خانة واحدة فإن عملية الجمع تنفذ بنفس طريقة الجمع في النظام العشري مع مراعاة أن أساس النظام العد المستعمل هو ٢

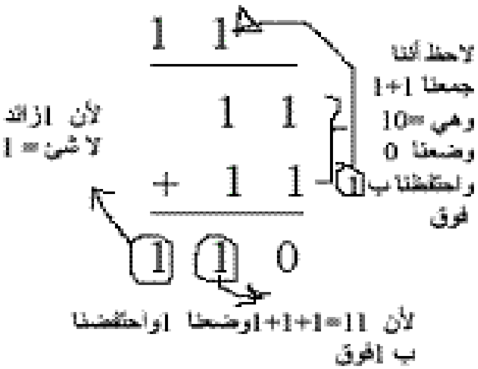
عملية الجمع :

مثال ١ : اجمع العددين الثنائيين: $(101)_2 + (011)_2 = (?)_2$

المحمول	1 1 1
العدد الأول	101
	+
العدد الثاني	011
	1000

الناتج: $(101)_2 + (011)_2 = (1000)_2$

مثال ٢ : اجمع العددين الثنائيين: $(11)_2 + (11)_2 = (?)_2$



الناتج: $(11)_2 + (11)_2 = (110)_2$

مثال ٣ : اجمع العددين الثنائيين: $(101101)_2 + (1011)_2 = (?)_2$

$$\begin{array}{r} {}^{1} \\ 101101 \\ +\ 001011 \\ \hline 111000 \end{array}$$

الناتج: $(101101)_2 + (1011)_2 = (111000)_2$

عملية الضرب :

تشبه عملية الضرب في النظام الثنائي عملية الضرب في النظام العشري

مع مراعاة القواعد الأساسية التالية:

$$0 \times 0 = 0$$
$$0 \times 1 = 0$$
$$1 \times 0 = 0$$
$$1 \times 1 = 1$$

مثال ١: ما هو ناتج ضرب العددين الثنائيين:

$$(101)_2 \times (10)_2 = (?)_2$$

الناتج هو:

$$(101)_2 \times (10)_2 = (1010)_2$$

$$\begin{array}{r} 101 \\ \times \quad 10 \\ \hline 000 \\ 101 \\ \hline 1010 \end{array}$$

مثال ٢: ما هو ناتج ضرب العددين الثنائيين:

$$(101)_2 \times (111)_2 = (?)_2$$

الناتج هو:

$$(101)_2 \times (111)_2 = (100011)_2$$

$$\begin{array}{r} 1\ 0\ 1 \\ \times \qquad 1\ 1\ 1 \\ \hline 1\ 0\ 1 \\ 1\ 0\ 1 \\ +\ 1\ 0\ 1 \\ \hline 1\ 0\ 0\ 0\ 1\ 1 \end{array}$$

٣- النظام الثماني (Octal System)

- نظام الأعداد الثماني هو نظام الأعداد الذي أساسه العدد (٨).
- مكونات هذا النظام ستكون: (٠، ١، ٢، ٣، ٤، ٥، ٦، ٧).
- لا يستخدم نظام الأعداد الثماني بكثرةٍ في أيامنا هذه، واقتصرت استخداماته على البدايات الأولى لظهور الحواسيب.
- **مثال**: العدد $١٧٥٣_{٨}$ هو: $1\times8^3+7\times8^2+5\times8^1+3\times8^0=(1003)_{10}$
- يستخدم نظام الثماني في الحوسبة في بعض الاحيان كما في ترميز UTF-8 (نظام ترميز Unicode)
- العلاقة بين كل من نظام الأعداد الثماني ونظام الأعداد الثنائي:

٠	1	2	3	4	5	6	7
000	001	010	011	100	101	110	١١١

٤- نظام الأعداد الستة عشري (Hexadecimal Number System).

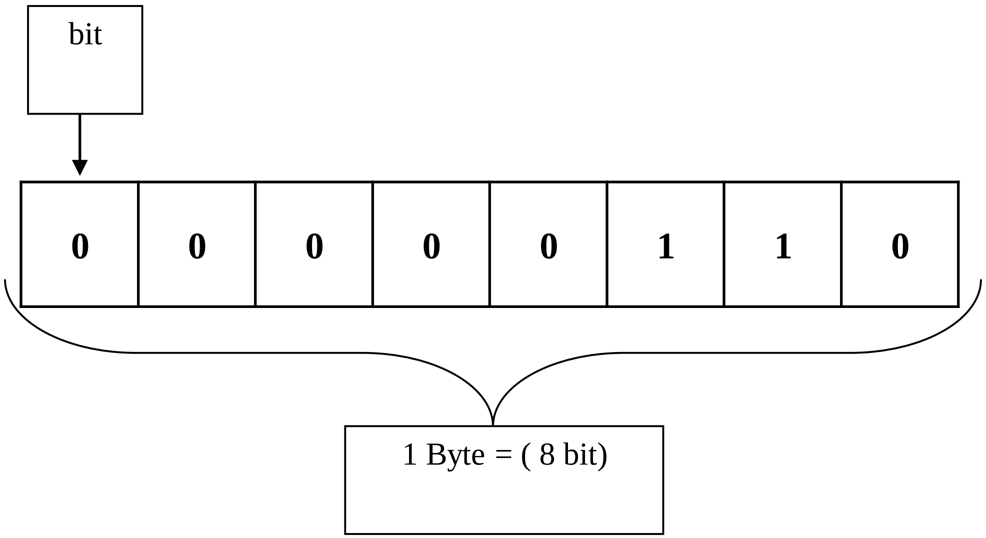
- يمتلك نظام الأعداد الستة عشري أساساً هو العدد (١٦).
- مكونات هذا النظام هي الأعداد التالية: (٠، ١، ٢، ٣، ٤، ٥، ٦، ٧، ٨، ٩، A، B، C، D، E، F).
- بهدف التمييز بينه وبين النظام العشري، تم ترميز الأرقام من ١٠ وحتى ١٥ في النظام الستة عشري بالأحرف التالية: (A، B، C، D، E، F) حيث تشير هذه الأحرف للأرقام (١٠، ١١، ١٢، ١٣، ١٤، ١٥) على التوالي.
- يستخدم نظام السداسي عشري في ترميز عناوين ذاكرة الحاسوب.

مثال: الرقم التالي : 3AD1 يكتب كالتالي:

$$3\times16^3+10\times16^2+13\times16^1+1\times16^0=(1003)_{10}$$

٥- تمثيل البيانات داخل الحاسوب.

- بت (Bit): أصغر وحدة تخزين في الحاسوب بداخل الذاكرة هي البت (Bit) وهي خلية ثنائية تستوعب فقط اما 0 أو 1 ويعني off 0 و 1 on وهناك مضاعفات لهذه الوحدة.
- بايت (Byte): وهي تساوي 8 بت ويستخدم البت لتمثيل رقم أو حرف أو رمز.
- الكلمة Word: وهي عبارة عن مجموعة من البتات المتجاورة تعامل كوحدة واحدة ويعتمد عدد البتات في الكلمة على الحاسوب وهو عنصر مهم في تصميم الحاسوب واصغر كلمة تساوي 8 بت وهناك حواسيب باطوال كلمة تساوي 16 بت و 32 بت و 64 و 128بت.



٥-١- المتمم الثنائي (Two's Complement) و كيفية الحصول عليه

- يستخدم المتمم الثنائي من أجل تمثيل الأعداد السالبة في الحاسب في النظام الثنائي.
- لتمثيل عدد سالب نتبع الخطوات التالية:
  - نكتب العدد بالنظام الثنائي.
  - نقلب الأصفار واحدات و الواحدات أصفاراً.
  - نضيف واحد إلى الرقم الناتج.

مثال: مثّل العدد ٣٠- بالنظام الثنائي عن طريق المتمم الثنائي:

الخطوة الأولى: نوجد المكافئ الثنائي للعدد الصحيح: $(30)_{10} = (00011110)_2$

الخطوة الثانية: نقلب كل ٠ إلى ١ وكل ١ إلى ٠: 1110 0001

الخطوة الثالثة: نضيف (نجمع) ١ إلى العدد الثنائي الناتج في الخطوة الثانية، أي:

$$\left.\begin{array}{r} 1110\ 0001 \\ + \\ 0000\ 0001 \end{array}\right\} = 1110\ 0010$$

٥-٢- تمثيل الأعداد داخل الحاسب باستخدام التشفير الثنائي الصريح.

التشفير الثنائي الصريح يتطلب تخزين الأعداد في ذاكرة الحاسب كعدد ثابت من البتات بصيغة **الكلمة**.

- **الكلمة (word)** هي سلسلة من البتات تعامل كوحدة.
- طول الكلمة هو عدد البتات في الكلمة.
- نعتبر أن الحاسوب يستخدم كلمات ذات طول 32بت.

تمثيل الأعداد الصحيحة

يمثل العدد الصحيح الموجب في ذاكرة الحاسب بصيغته الثنائية، وبمكمل الثنائيات إذا كان سالباً.

مثال: يمثل العدد ٤٢٣ في ذاكرة حاسب من ٣٢ بت كالآتي:

✓ نوجد المكافئ الثنائي للعدد المذكور: $(423)_{10} = (1\,1010\,0111)_2$

✓ نكتب العدد الثنائي الحاصل بصيغة ٣٢ بت كالآتي:

خانة الإشارة

0	0	0	0	0	0	0	0	0	...	0	0	1	1	0	1	0	0	1	1	1

تابع المثال: تمثيل العدد (−٤٢٣) في ذاكرة حاسب من ٣٢ بت كالآتي:

- ✓ نوجد المكافئ الثنائي للعدد الموجب ٤٢٣: $(423)_{10} = (1\,1010\,0111)_2$
- ✓ نكتب العدد الحاصل بصيغة ٣٢ بت: $[00\cdots(1\,1010\,0111)]_{32}$
- ✓ نقلب كل ٠ إلى ١ وكل ١ إلى ٠. $[00\cdots(1\,1010\,0111)]_{32} \to [11\cdots(0\,0101\,1000)]_{32}$
- ✓ نضيف (نجمع) ١ إلى العدد الثنائي الناتج: $[11\cdots(0\,0101\,1000)]_{32} + 1 = [11\cdots(0\,0101\,1001)]_{32}$
- ✓ الناتج بصيغة ٣٢ بت:

خانة الإشارة

1	1	1	1	1	1	1	1	1	1	...	1	1	0	0	1	0	1	1	0	0	1

- يستطع الحاسب معرفة هل الرقم قي الذاكرة موجب أم سالب وذلك بالنظر إلى البت في الخانة الأولى من أقصى اليسار: إذا كان البت الأول 0 فإن العدد موجب وإذا كان 1 فإن العدد سالب.
- بالتالي، فإن أكبر عدد صحيح (موجب) يمكن تخزينه في ذاكرة الحاسب ذي ٣٢ بت هو:

0	1	1	1	1	1	1	1	1	1	1	1	1	...	1	1	1	1	1	1	1	1	1

- أي ٣١ واحد، أو $2^{31} - 1$ ويساوي تقريباً ٢ مليار عدد موجب.
- بالمثل فإن أصغر عدد (سالب) يمكن تخزينه في ذاكرة الحاسب الحاسب ذي ٣٢ بت هو: تقريباً ٢ مليار عدد سالب.

٦- شيفرة الحاسب الآلي (Computer codes)

- يتشكل كل رمز أو حرف أو رقم أو إشارة من ٨ بتات (8 Bit = 1 Byte)، وتكون إما ٠ أو ١.
- أو البايت (Byte) هو عبارة عن حرف أو رمز أو إشارة ويساوي ٨ بت(أي مجموعة من ثمان بتات). ويعتبر البايت وحدة خزن للبيانات في الحاسوب.
- قام المبرمجون بوضع ترميز قياسي (Standard Code) لترميز الحروف الأبجدية والأرقام والرموز فيما يماثله من هذا النظام الثنائي، وعليه فقد استخدم علم الترميز (التشفير) في وضع شفرة لأبجديات اللغة المستعملة (اللغة الإنجليزية مثلاً) وقد تعددت هذه الشيفرات.
- مثلاً حرف a يشفر بنظام أسكي (01100001) ASCII وعلامة (=) (٠٠١١١١٠١) وهكذا.
-

أهم الشيفرات العالمية

١. نظام بي سي دي BCD (Binary Coded Decimal) الذي وضعته شركة IBM ولكن لم يدم طويلاً وذلك نظراً لإقتصار هذا النظام على سداسية البتات والذي يؤدي إلى عدم الترميز (التشفير) لكل الحروف.

٢. نظام تشفير إبسيديك EBCDIC (Extended Binary-Coded Decimal Interchange Code) و عبارة عن ٨ بتات (٤ المنطقة و ٤ الرقم)، ويكون مسبوق بالرمز ١١ في الأحرف أو ١١١ أو ١١١١ للأرقام، ويستطيع ترميز ٢٥٦ حرف.

٣. نظام تشفير أسكي ASCII (American Standard Code for Information Interchange) وعملت هذه التشفيرة في الوقت الذي تطورت فيه الحواسيب الصغيرة والذي مكن مؤسسة المعايير القياسية الأمريكية (American National Standards Institute) من تحديد المعايير االقياسية للحواسيب. وهو نظام سباعي الخانات أضيف له الصفر في الخانة الثامنة اليسرى.

٤. تعتبر مجموعة ترميز ASCII حالياً أكثر المجموعات شهرة ، وقد تم ترميز أبجديات اللغة في الشيفرتين السابقتين أستخدمتا ثمان خانات لتشفير الأبجديات والأرقام ، فمثلاً رمز الحرف M بتشفير أسكي هو ٠١١٠١١٠١ وهكذا.

أهم الشيفرات العالمية

٥. الترميز العالمي Unicode ويسمى المعيار القياسي العالمي للحروف (Unicode Worldwide Character Standard) .

- هو إصدار قياسي جديد يمثل حجم الكلمة ببايتين (٢ Bytes) بدلاً من بايت واحد، أي أن ١٦ بت لكل حرف ولذا فإن إحتمال وجود الحرف من بين ٦٥٥٣٦ = ٢١٦ حرفاً أو رمزاً حقيقياً.
- هذا يكفي لتغطي كافة لغات العالم أي أنها تمثل أي رمز في العالم ويمكن ضم الحروف الهجائية الصينية واليابانية والكورية وتلك الموجودة في النصوص الكلاسيكية والتاريخية المعروفة، بما فيها العربية .

ASCII	SYMBOL	EBCDIC
00110000	0	11110000
00110001	1	11110001
00110010	2	11110010
00110011	3	11110011
00110100	4	11110100
00110101	5	11110101
00110110	6	11110110
00110111	7	11110111
00111000	8	11111000
00111001	9	11111001
01000001	A	11000001
01000010	B	11000010
01000011	C	11000011
01000100	D	11000100
01000101	E	11000101
01000110	F	11000110
01000111	G	11000111
01001000	H	11001000
01001001	I	11001001
01001010	J	11010001
01001011	K	11010010
01001100	L	11010011
01001101	M	11010100
01001110	N	11010101
01001111	O	11010110
01010000	P	11010111
01010001	Q	11011000
01010010	R	11011001
01010011	S	11100010
01010100	T	11100011
01010101	U	11100100
01010110	V	11100101
01010111	W	11100110
01011000	X	11100111
01011001	Y	11101000
01011010	Z	11101001
00100001	!	01011010
00100010	“	01111111
00100011	#	01111011
00100100	$	01011011
00100101	%	01101100
00100110	&	01010000
00101000	(	01001101
00101001	)	01011101
00101010	*	01011100
00101011	+	01001110

نهاية المحاضرة السابعة