

Προγραμματισμός Υπολογιστών



Αναπαράσταση Πληροφορίας

Κ. Βασιλάκης, ΣΤΕΦ, ΤΕΙ Κρήτης



Δεδομένα και πληροφορία

- **Δεδομένα** είναι ένα σύνολο διακριτών στοιχείων σχετικά με ένα συμβάν ή μια διαδικασία χωρίς κάποια άμεση σημασία.
 - Τα δεδομένα είναι ουσιαστικά οντότητες και ιδιότητες που προκύπτουν από παρατηρήσεις, καταγραφές ή υπολογισμούς και μπορεί να είναι χαρακτήρες, κείμενο, αριθμοί, σύμβολα, σχήματα, εικόνες κλπ και γενικότερα ιδιότητες που προκύπτουν από παρατηρήσεις, πειραματισμούς ή υπολογισμούς.
- **Ψηφιακά δεδομένα (data)** είναι δεδομένα που έχουν αναπαρασταθεί με κάποιο τρόπο (κωδικοποίηση), είναι αποθηκευμένα σε υπολογιστή και είναι δυνατόν να επεξεργαστούν.
 - Τα ψηφιακά δεδομένα αποθηκεύονται σε αρχεία δηλ. συλλογές δεδομένων που έχουν κάποια σχέση μεταξύ τους (κείμενο, εγγραφές, βάσεις δεδομένων κλπ).
- Η **πληροφορία** παράγεται από την επεξεργασία ή/και την ερμηνεία των ψηφιακών δεδομένων και μας βοηθά στη λήψη αποφάσεων. Πρόκειται για επεξεργασμένα δεδομένα στα οποία έχει δοθεί κάποιο νόημα.
 - Η επεξεργασία δεδομένων αφορά συνήθως σε συλλογή, αναζήτηση, ομαδοποίηση, ταξινόμηση, σύγκριση, επιλογή δεδομένων και σε εκτέλεση αριθμητικών/λογικών υπολογισμών.



Δεδομένα & πληροφορία (παράδειγμα)

- Έστω οι αριθμοί 8, 5, 7, 6, 7
- Αυτοί οι αριθμοί 8, 5, 7, 6, 7 σαν δεδομένα δεν έχουν κάποιο νόημα.
- Αν πρόκειται για βαθμούς εργασιών ενός φοιτητή σε ένα μάθημα αποκτούν σημασία, διότι ο μέσος όρος αυτών μπορεί να μας οδηγήσει στην απόφαση αν ο φοιτητής περνά το μάθημα ή όχι.
- Αν όλες οι εργασίες μετρούν το ίδιο τότε ο μέσος όρος της βαθμολογίας μας δίνει τη πληροφορία που θέλουμε (περνά ή όχι το μάθημα).
- Ο υπολογισμός του μέσου όρου είναι η επεξεργασία δεδομένων.
- Αν ο μέσος όρος είναι ≥ 5 τότε έχουμε τη πληροφορία ότι ο φοιτητής περνάει το μάθημα.

Δεδομένα + επεξεργασία / ερμηνεία --> πληροφορία



Η αναπαράσταση στους υπολογιστές

- Οι υπολογιστές είναι σε θέση να αναγνωρίζουν 2 διακριτές καταστάσεις (αν περνά ή όχι ρεύμα).
- Το **bit** είναι μικρότερη μονάδα μέτρησης και εκφράζει αυτές τις 2 διακριτές καταστάσεις (τιμές 0 και 1).
- Τα πάντα μέσα σ' ένα υπολογιστικό σύστημα *κωδικοποιούνται* με αυτές τις 2 καταστάσεις (0/1):
 - οι εντολές που εκτελούνται και
 - τα ψηφιακά δεδομένα που επεξεργάζονται (κείμενο, αριθμοί, αναλογικό σήμα, εικόνες, video κλπ).
- Συνεπώς, οι κωδικοποιήσεις βασίζονται στο δυαδικό σύστημα και συνιστούν τρόπους αναπαράστασης των δεδομένων στους υπολογιστές.
- Οι κωδικοποιήσεις λαμβάνουν υπόψη τους το είδος των δεδομένων που πρόκειται να αναπαρασταθούν.
- Αφιερώνουν συγκεκριμένο αριθμό από bits για κάθε στοιχείο των δεδομένων που πρόκειται ν' αναπαραστήσουν.



Πόσα bits χρειάζομαι;

- Για τα σημεία του ορίζοντα (βορράς, νότος, ανατολή, δύση);
 - Απαιτούνται 4 διακριτές καταστάσεις.
 - με 1 bit έχουμε 2 διακριτές καταστάσεις: 0 και 1 – δεν επαρκούν!
 - με 2 bits έχουμε 4 διαφορετικές καταστάσεις – επαρκούν! Αυτές είναι: 00, 01, 10 και 11.
- Για τα 24 κεφαλαία γράμματα της αλφαβήτου;
 - με 1 bit έχουμε 2 διαφορετικές καταστάσεις – δεν επαρκούν!
 - με 2 bits έχουμε 4 διαφορετικές καταστάσεις – δεν επαρκούν!
 - με 3 bits έχουμε 8 διαφορετικές καταστάσεις – δεν επαρκούν!
 - με 4 bits έχουμε 16 διαφορετικές καταστάσεις – δεν επαρκούν!
 - με 5 bits έχουμε 32 διαφορετικές καταστάσεις – επαρκούν για τα 24 γράμματα (περισσεύουν 8).
 - A: 00000, B:00001, C:00010,, X:10110, Y:10111, Z:11000



Πόσα bits χρειάζομαι;

- Γενικότερα, με **k bits** μπορούμε ν' αναπαραστήσουμε:
 $N=2^k$ διαφορετικές καταστάσεις ($k=\log_2 N$).
- Το τελευταίο δεξιά ψηφίο είναι το **λιγότερο σημαντικό ψηφίο** (Least Significant Bit - **LSB**) της δυαδικής αναπαράστασης.
- Το πρώτο αριστερά ψηφίο είναι το **περισσότερο σημαντικό ψηφίο** (Most Significant Bit – **MSB**) της δυαδικής αναπαράστασης.



Αναπαράσταση κειμένου

- Αφορά σε χαρακτήρες που μπορούν να εκτυπωθούν:
*A-Z, a-z, 0-9, !, @, #, \$, %, ^, &, *, (,), _, -, +, /, κλπ*
- Επίσης αφορά και σε κάποιους χαρακτήρες ελέγχου που δεν εκτυπώνονται:
αλλαγή γραμμής, <enter>, <backspace> κλπ
- Σε κάθε χαρακτήρα αντιστοιχεί ένας συγκεκριμένος συνδυασμός από bits (δηλ. ένας δυαδικός αριθμός).
- Σε κάθε κωδικοποίηση χρησιμοποιείται ένας συγκεκριμένος αριθμός από bits, ανάλογα με το αριθμό των χαρακτήρων που πρόκειται να αναπαρασταθούν.
- Υπάρχουν πολλές αναπαραστάσεις χαρακτήρων κειμένου.
- Οι πιο γνωστές:

ASCII, ISO & UNICODE



Κωδικοποίηση κατά ASCII

- ASCII (American standard Code for Information Interchange).
- Πρόκειται για τη πρώτη κοινή αναπαράσταση χαρακτήρων κειμένου για όλους τους κατασκευαστές υπολογιστών.
- Χρησιμοποιεί 7 bits για την κωδικοποίηση και ένα όγδοο bit για λόγους έλεγχου στη μετάδοση των bits.
- Συνεπώς, υπάρχουν 128 (2^7) διαφορετικοί συνδυασμοί των 7 bits.
- Αρχικά περιελάμβανε μόνο το λατινικό αλφάβητο.
- Οι υπόλοιποι 128 ($2^8=256$ συνολικά) συνδυασμοί χρησιμοποιήθηκαν αργότερα για ειδικούς γραφικούς χαρακτήρες ή για χαρακτήρες άλλων αλφάβητων από τον ISO (International Standardization Organization - Διεθνής Οργανισμός Τυποποίησης).
- Χρησιμοποιήθηκε για πρώτη φορά το 1963 και ευρέως από το 1968 και μετά.
- Έχει επικρατήσει σαν ορολογία περιγραφής αρχείων κειμένου.



Κωδικοποιήσεις ISO και UNICODE

- Το ISO 8859 είναι μια σειρά από 8-bits τυποποιήσεις χαρακτήρων (8859-1, 8859-2, κλπ).
- Η κωδικοποίηση ASCII αναπαριστά ικανοποιητικά τους λατινικούς χαρακτήρες της αγγλικής γλώσσας, αλλά δεν συμπεριλαμβάνει εθνικούς χαρακτήρες άλλων χωρών.
- Το ISO 8859 είναι μια 8-bit επέκταση του ASCII που χρησιμοποιεί και τα 8 bits.
- Για τους ελληνικούς χαρακτήρες (μονοτονικό σύστημα) υπάρχει το ISO 8859-7.
- Η κωδικοποίηση UNICODE χρησιμοποιεί 8, 16 ή και 32 bits για την αναπαράσταση χαρακτήρων (2^8 , 2^{16} και 2^{32} διαφορετικούς χαρακτήρες).
- Η κωδικοποίηση UNICODE περιλαμβάνει 2 κατηγορίες αναπαραστάσεων:
 - UTF (Unicode Transformation Format) και
 - UCS (Universal Character Set)



Κωδικοποίηση ISO 8859

Bits	b8b7b6b5	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
b4b3b2b1	HEX	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0			SP	0	@	P	`	p			NBSP	°	ı	Π	ü	π
0001	1			!	1	A	Q	a	q			`	±	Α	Ρ	α	ρ
0010	2			"	2	B	R	b	r			'	²	Β		β	ς
0011	3			#	3	C	S	c	s			£	³	Γ	Σ	γ	σ
0100	4			\$	4	D	T	d	t				'	Δ	Τ	δ	τ
0101	5			%	5	E	U	e	u				ˆ	Ε	Υ	ε	υ
0110	6			&	6	F	V	f	v			ı	À	Z	Φ	ζ	φ
0111	7			'	7	G	W	g	w			§	·	H	X	η	χ
1000	8			(8	H	X	h	x			ˆ	É	Θ	Ψ	θ	ψ
1001	9)	9	I	Y	i	y			©	Ĥ	I	Ω	ι	ω
1010	A			*	:	J	Z	j	z				ı	K	İ	κ	ı
1011	B			+	;	K	[k	{			«	»	Λ	ÿ	λ	ü
1100	C			,	<	L	\	l				¬	Œ	M	ά	μ	ό
1101	D			-	=	M]	m	}			SHY	½	N	έ	ν	ύ
1110	E			.	>	N	^	n	-			®	Υ	Ξ	ή	ξ	ώ
1111	F			/	?	O	-	o	DEL			¯	Ω	Ο	ί	ο	



Αναπαράσταση ακεραίων αριθμών

- Όλοι οι φυσικοί με πρόσημο και το μηδέν
- Εύρος: $-\infty$ 0 $+\infty$
- Φυσικά δεν υπάρχει υπολογιστικό σύστημα που μπορεί να αναπαραστήσει όλους τους ακεραίους.
- Όπως έχουμε δει πριν, ο αριθμός των bits που χρησιμοποιείται για την αναπαράσταση των ακεραίων καθορίζει και το εύρος τους (μικρότερος – μεγαλύτερος).
- Υπάρχουν διαφορετικές αναπαραστάσεις για:
 - Ακεραίους χωρίς πρόσημο
(μη προσημασμένοι: από 0 έως ∞).
 - Ακεραίους με πρόσημο
(προσημασμένοι: από $-\infty$ έως $+\infty$).



Μη προσημασμένοι ακέραιοι

- Ο αριθμός N των bits που χρησιμοποιείται για την αναπαράσταση καθορίζει και τον μεγαλύτερο αριθμό που μπορεί να αναπαρασταθεί.
- Παραδείγματα:
 - 4 bits: ο μεγαλύτερος ακέραιος είναι $1111_2 = 15_{10} = (2^4 - 1)$.
Εύρος: από 0 έως 15.
 - 8 bits: ο μεγαλύτερος ακέραιος είναι $11111111_2 = 255_{10} = 2^8 - 1$.
Εύρος: από 0 έως 255.
- Γενικά, για N bits το εύρος των τιμών που έχουμε για ακεραίους χωρίς πρόσημο είναι:
από 0 έως $2^N - 1$.
- Συνεπώς, για n' αποθηκευτεί σε υπολογιστή ένας μη προσημασμένος ακέραιος αριθμός, απλά μετατρέπεται στο δυαδικό σύστημα.



Παραδείγματα μη προσημασμένων

- Έστω N ο αριθμός των bits που χρησιμοποιείται για την αναπαράσταση:
 - Αν $N=8$, ο αριθμός 12_{10} (1100_2) αποθηκεύεται ως 00001100.
 - Αν $N=16$, ο αριθμός 12_{10} (1100_2) αποθηκεύεται ως 00000000000001100.
 - Αν $N=8$, ο αριθμός 260_{10} (100000100_2) δεν μπορεί ν' αποθηκευτεί. Αν προσπαθήσουμε να τον αποθηκεύσουμε σε 8 bits τότε έχουμε το φαινόμενο της **υπερχείλισης** (overflow).
 - Αν $N=16$, ο αριθμός 260_{10} (100000100_2) αποθηκεύεται ως 0000000100000100.
 - Αν $N=32$, ο αριθμός 260_{10} (100000100_2) αποθηκεύεται ως 0000000000000000000000000100000100.



Προσημασμένοι ακέραιοι

- Απαιτείται ένδειξη για το πρόσημο του ακεραίου.
- Αφιερώνεται λοιπόν ένα bit για την ένδειξη του πρόσημου:
0 --> θετικός, 1 --> αρνητικός
- Τα υπόλοιπα bits χρησιμοποιούνται για τη παράσταση του μέτρου του ακεραίου (απόλυτη τιμή).
- Προφανώς με ένα bit λιγότερο (αυτό του πρόσημου) το μέτρο των αριθμών υποδιπλασιάζεται.
- Η αναπαράσταση των θετικών ακεραίων είναι η ίδια σε όλες τις κωδικοποιήσεις που θα εξετάσουμε στην συνέχεια.
- Αλλάζει μόνο ο τρόπος αναπαράστασης των αρνητικών αριθμών.
- Θα εξεταστούν 3 κωδικοποιήσεις αρνητικών αριθμών:
 - Πρόσημο και μέτρο
 - Συμπλήρωμα ως προς 1
 - Συμπλήρωμα ως προς 2



Κωδικοποίηση Πρόσημου-Μέτρου

- Το τελευταίο αριστερά bit (Most Significant Bit – MSB) χρησιμοποιείται για την ένδειξη του πρόσημου.
- Τα εναπομείναντα ψηφία χρησιμοποιούνται για το μέτρο του αριθμού:
 - +12 --> 00001100 (8 bits αναπαράσταση)
 - -12 --> 10001100 (8 bits αναπαράσταση)
 - +260 --> 0000000100000100 (16 bits αναπαράσταση)
 - -260 --> 1000000100000100 (16 bits αναπαράσταση)
- Ο μέγιστος προσημασμένος ακέραιος που μπορεί να παρασταθεί με N bits και κωδικοποίηση πρόσημου-μέτρου είναι:
$$011111\dots1111_2 = +(2^{N-1} - 1)_{10}$$
- Ο ελάχιστος προσημασμένος ακέραιος που μπορεί να παρασταθεί με N bits και κωδικοποίηση πρόσημου-μέτρου είναι:
$$111111\dots1111_2 = -(2^{N-1} - 1)_{10}$$
- Γενικά, με N bits, το εύρος τιμών που έχουμε με κωδικοποίηση πρόσημου-μέτρου είναι: από $-2^{(N-1)} - 1$ έως $+2^{(N-1)} - 1$
- Παραδείγματα:
 - Με 8 bits το εύρος τιμών είναι: -2^7-1 έως $+2^7-1$ (-127 έως +127).
 - Με 16 bits το εύρος τιμών είναι: $-2^{15}-1$ έως $+2^{15}-1$ (-32767 έως +32767)
- Έχουμε όμως 2 παραστάσεις για το μηδέν.
 - 000... ..000 (+0) και
 - 100... ..000 (-0)



Κωδικοποίηση συμπληρώματος ως προς 1

- Και εδώ το MSB χρησιμοποιείται για την ένδειξη του πρόσημου (0 -> θετικός, 1-> αρνητικός).
- Για τη παράσταση των αρνητικών ακεραίων χρησιμοποιούμε το συμπλήρωμα της δυαδικής παράστασης του αριθμού ως προς 1.
- Το συμπλήρωμα ως προς 1 ενός δυαδικού αριθμού βρίσκεται εύκολα αν αντικατασταθούν όλα τα 1 του αριθμού με 0 και όλα τα 0 με 1.
- Παράδειγμα:
 - ο αριθμός $+12_{10}$ (1100_2) θα παρασταθεί σε 8 bits ως: **00001100** (ίδια με την κωδικοποίηση πρόσημου-μέτρου).
 - ο αριθμός -12_{10} θα παρασταθεί ως: **11110011**
- Μέγιστος θετικός: **01111... ..1111** = $+(2^{(N-1)} - 1)_{10}$
- Ελάχιστος αρνητικός: **100000... ..0000** = $-(2^{(N-1)} - 1)_{10}$
- Και πάλι το μηδέν αποθηκεύεται με 2 τρόπους:
 - 000000.... ..000000 και
 - 11111.... ..11111 (συμπλήρωμα του 000... ..000 ως προς 1)



Κωδικοποίηση συμπληρώματος ως προς 2

- Και πάλι το MSB χρησιμοποιείται για την ένδειξη του πρόσημου (0 -> θετικός, 1->αρνητικός).
- Για τη παράσταση των αρνητικών ακεραίων χρησιμοποιούμε το συμπλήρωμα της δυαδικής παράστασης του αριθμού ως προς 2.
- Το συμπλήρωμα ως προς 2 ενός δυαδικού αριθμού βρίσκεται εύκολα αν αντικατασταθούν όλα τα 1 του αριθμού με 0 και όλα τα 0 με 1 και στη συνέχεια προσθέσουμε 1. Δηλαδή αν προσθέσουμε 1 στο συμπλήρωμα του.
- Παράδειγμα:
 - ο αριθμός $+12_{10}$ (1100_2) θα παρασταθεί σε 8 bits ως: 00001100 (ίδια με την κωδικοποίηση πρόσημου-μέτρου).
 - ο αριθμός -12_{10} θα παρασταθεί ως: $11110011+1= 11110100$
- Έχουμε ένα (1) μόνο μηδέν: 0000... .. 000
- Το εύρος τιμών που έχουμε με κωδικοποίηση συμπληρώματος ως προς 2: από $-2^{(N-1)}$ έως $+2^{(N-1)} - 1$
- Πρόκειται για τον τρόπο αναπαράστασης ακεραίων με πρόσημο.



Παράδειγμα παράστασης με 4 bits

	Π-Μ	Συμπ. - 1	Συμπ. - 2
7	0111	0111	0111
6	0110	0110	0110
5	0101	0101	0101
4	0100	0100	0100
3	0011	0011	0011
2	0010	0010	0010
1	0001	0001	0001
0	0000 1000	0000 1111	0000
-1	1001	1110	1111
-2	1010	1101	1110
-3	1011	1100	1101
-4	1100	1011	1100
-5	1101	1010	1011
-6	1110	1001	1010
-7	1111	1000	1001
-8	-	-	1000



Παράδειγμα παράστασης με 8 bits

	Π-Μ	Συμπ. - 1	Συμπ. - 2
128			
127	01111111	01111111	01111111
126	01111110	01111110	01111110
...
...
2	00000010	00000010	00000010
1	00000001	00000001	00000001
0	00000000 10000000	00000000 11111111	00000000
-1	10000001	11111110	11111111
-2	10000010	11111101	11111110
...
...
-126	11111110	10000001	10000010
-127	11111111	10000000	10000001
-128	-	-	10000000

