

Μνήμη – Διευθύνσεις – Δείκτες

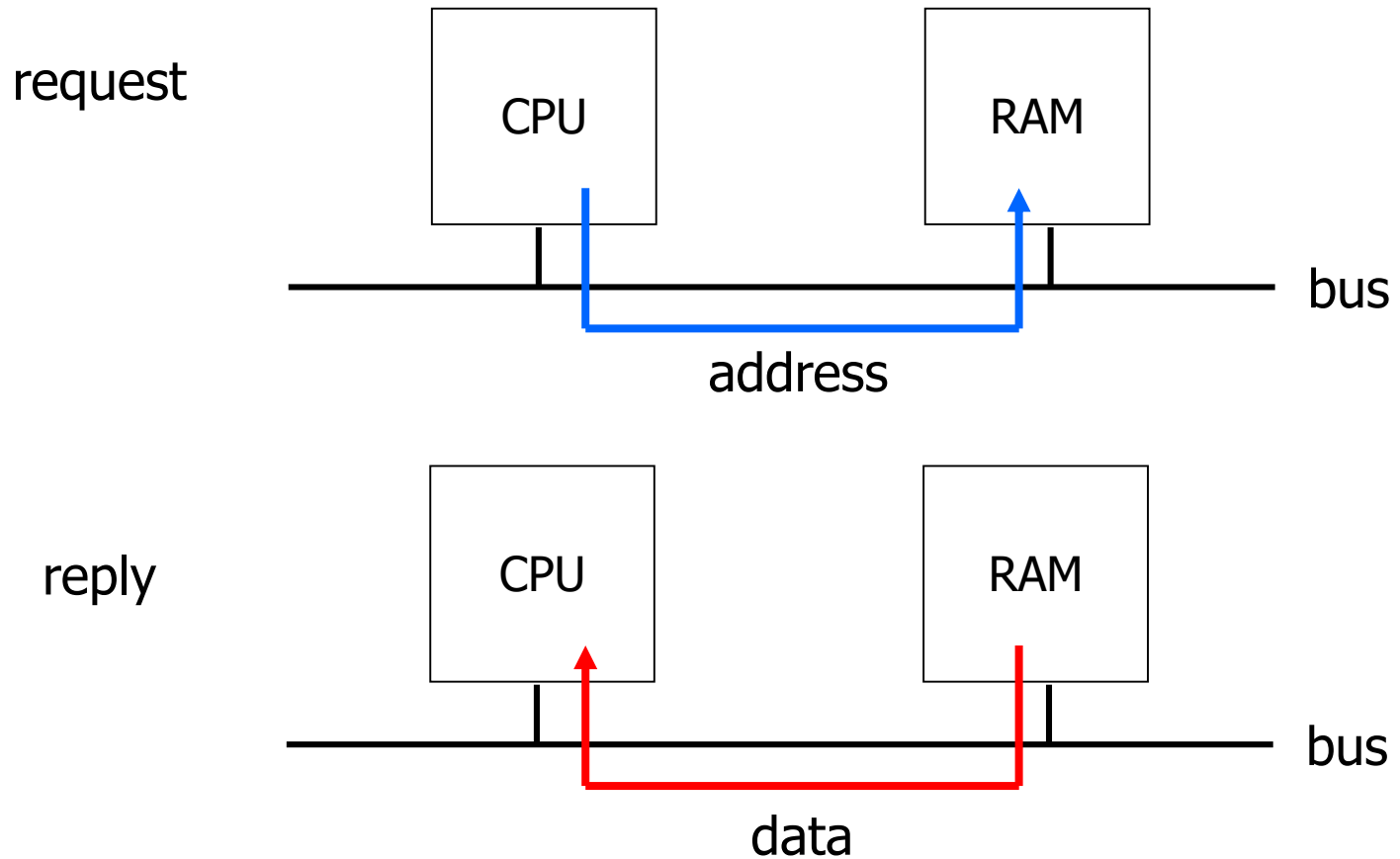
Μνήμη Η/Υ

- Ειδικό υλικό, **ξεχωριστό** από τον επεξεργαστή
- Χρησιμεύει για την αποθήκευση δεδομένων και της κατάστασης εκτέλεσης του προγράμματος
- Η μνήμη του Η/Υ **δεν** είναι «μόνιμη»
- Αν σταματήσει η παροχή ρεύματος, τα δεδομένα που βρίσκονται στη μνήμη του Η/Υ **χάνονται**
- Αν θέλουμε τα δεδομένα να «επιζήσουν» και να εξακολουθούν να υφίσταται μετά από μια επανεκκίνηση, πρέπει να αποθηκευτούν στον δίσκο
 - βλέπε σχετική ενότητα προς το τέλος του μαθήματος

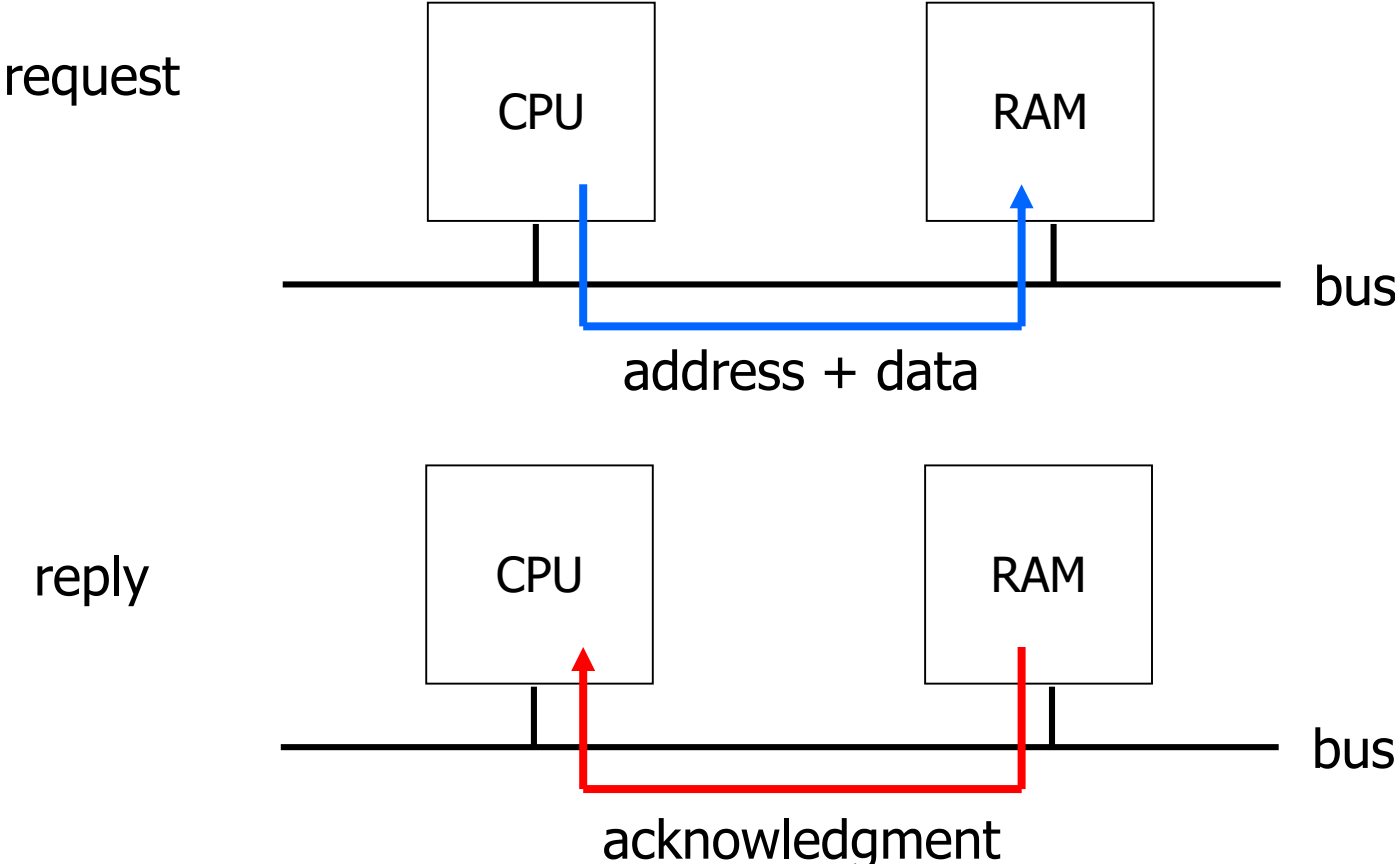
Random access memory (RAM)

- Μνήμη που μπορεί να προσπελαστεί άμεσα
 - σε αντίθεση με σειριακή προσπέλαση – βλέπε αρχεία
- Η προσπέλαση οποιουδήποτε σημείου της μνήμης επιφέρει το ίδιο κόστος / καθυστέρηση
- Σε επίπεδο υλικού, η μνήμη υλοποιείται από ένα ή περισσότερα memory chips
- Ο επεξεργαστής προσπελάζει την μνήμη μέσω ενός πρωτοκόλλου αίτησης/απάντησης, που λαμβάνει χώρα πάνω από ένα ειδικό δίαυλο επικοινωνίας

Read transaction

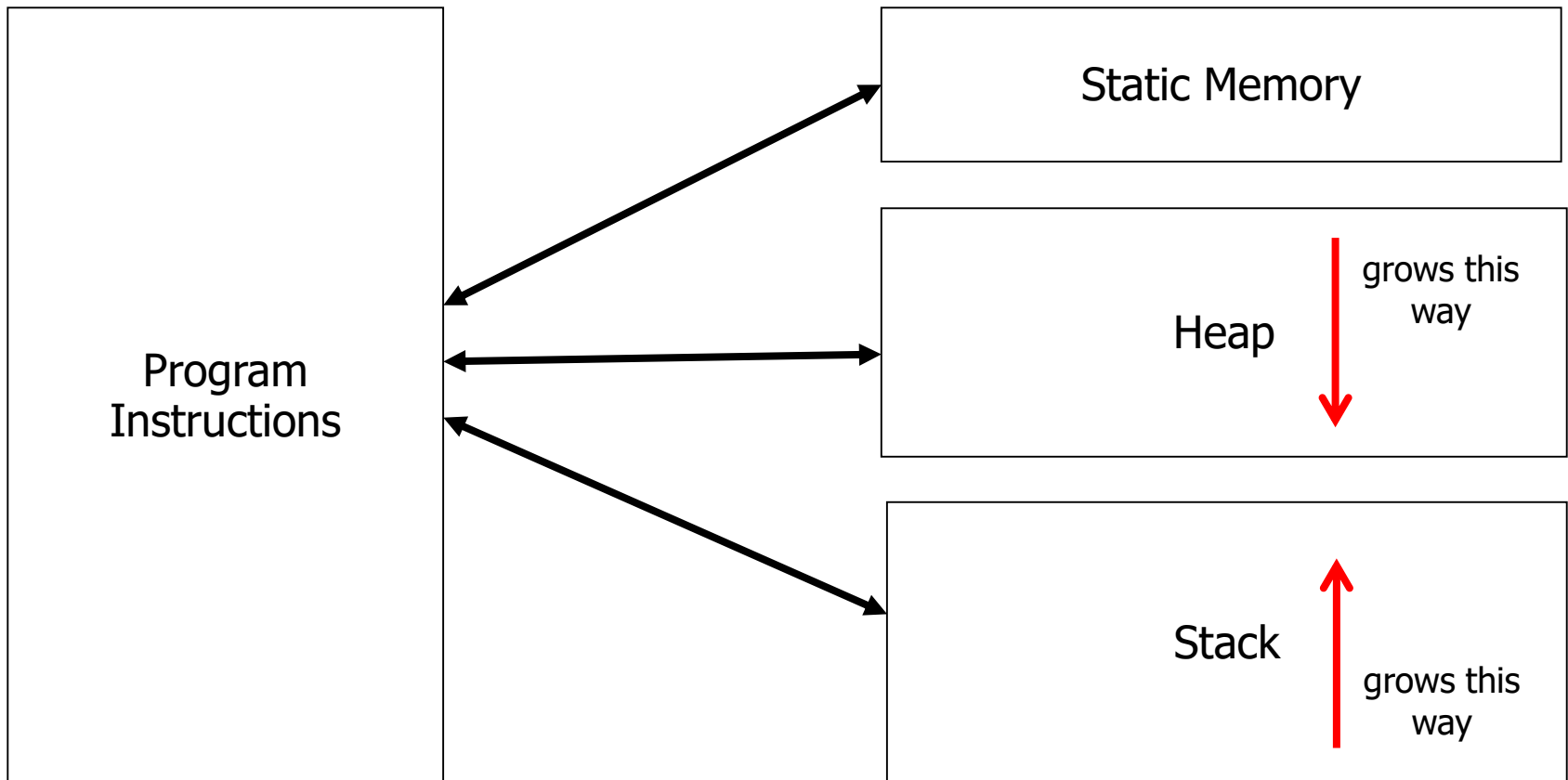


Write transaction



Μνήμη προγράμματος

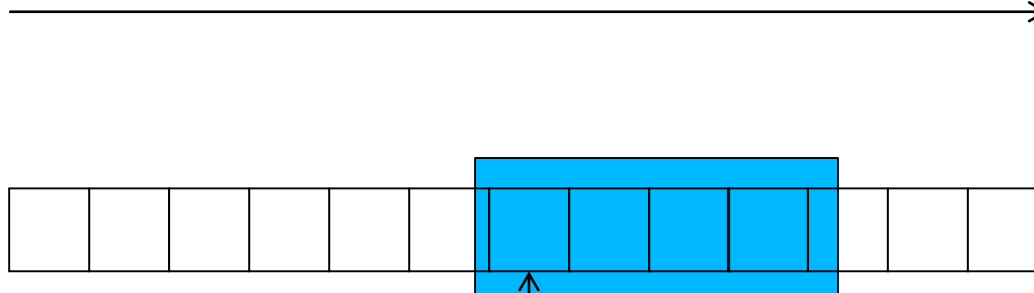
- Στατική μνήμη (static memory)
 - δεσμεύεται στατικά, κατά την έναρξη του προγράμματος
 - καθολικές & στατικές μεταβλητές (globals, static)
- Στοιίβα (stack)
 - αυξομειώνεται κατά την κλήση συναρτήσεων
 - τοπικές μεταβλητές συναρτήσεων
- Δυναμική μνήμη / σωρός (dynamic memory / heap)
 - δεσμεύεται/αποδεσμεύεται δυναμικά, κατά την εκτέλεση
 - στην C, με βάση ρητές εντολές του προγράμματος



Μεταβλητές

- Τα περιεχόμενα της μνήμης προσπελάζονται μέσω των αντίστοιχων διευθύνσεων (addresses)
- Οι μεταβλητές ενός προγράμματος είναι εύχρηστα ονόματα που αντιστοιχίζονται (αυτόματα) στην **διεύθυνση** της μνήμης που χρησιμοποιείται για να αποθηκευτούν τα περιεχόμενα της μεταβλητής
 - ποιος υλοποιεί αυτή την αντιστοίχιση;
- Ο αριθμός των bytes που χρησιμοποιούνται για την αποθήκευση των περιεχομένων μιας μεταβλητής εξαρτάται από τον **τύπο** της

address
space



address of k
notation: $\&k$

contents of k
notation: k

#bytes used to store k
notation: `sizeof(k)`

Ερμηνεία δεδομένων

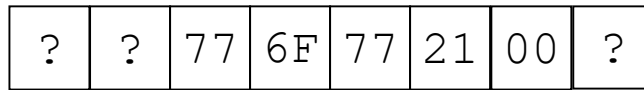
- Η τιμή ενός ή περισσότερων bytes μπορεί να **ερμηνευθεί** με **διαφορετικούς** τρόπους
- Η ερμηνεία των περιεχομένων των μεταβλητών (bytes στις αντίστοιχες θέσεις μνήμης) γίνεται **αυτόματα** με βάση τον **τύπο** τους
 - ο προγραμματιστής **δεν** χρειάζεται να γνωρίζει την εσωτερική αναπαράσταση στην μνήμη
 - αυτό είναι πολύ σημαντικό – γιατί;
- Ο προγραμματιστής μπορεί να «εκβιάσει», με δική του ευθύνη) μια διαφορετική ερμηνεία των περιεχομένων της μνήμης – με **type casting**

```
char str[5] = "wow!";
char *c;
int i;
int *k;

k = (int *) str;
printf("print as int: %d (%x)\n", *k, *k);

c = (char *) k;
for (i=0; i<sizeof(int); i++) {
    printf("print byte %d of k as char %c\n", i, *(c+i));
}
```

—————→ addresses get higher



τιμή bytes στην μνήμη

pointer

ερμηνεία ως char 'w'

ερμηνεία ως string "wow!"

ερμηνεία ως int
(little endian) 561475447 dec
 21776F77 hex

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Δείκτες (pointers)

- Ο δείκτης είναι ένας **μετα-τύπος** που μπορεί να εφαρμοστεί σε οποιοδήποτε τύπο δεδομένων
- Η τιμή ενός δείκτη είναι μια **διεύθυνση μνήμης**
- Μπορεί να χρησιμοποιηθεί για να προσπελαστούν τα περιεχόμενα της αντίστοιχης θέσης της μνήμης, **ερμηνεύοντας** τα με βάση **τον τύπο του δείκτη**
- Στην C, επιτρέπεται αριθμητική με δείκτες
 - +1: αύξηση της τιμής του δείκτη (διεύθυνσης) κατά τον αριθμό των bytes που έχουν τα αντικείμενα του τύπου
 - -1: μείωση της τιμής του δείκτη (διεύθυνσης) κατά τον αριθμό των bytes που έχουν τα αντικείμενα του τύπου

little endian

```
char str[5] = "wow!";
```

```
char *c = &str[1];
```

```
char *c2 = c + 1;
```

```
char **c3 = &c2;
```

pointer size = 2 bytes
(just as an example)

