

# Keyboard

Ασσιούρας Ιωάννης 5593

Βούκας Ιωάννης 5001

Πρωτονοτάριος Ιωάννης 6072

# The PC keyboard

- Η κύρια συσκευή εισόδου του συστήματος.
- 101 πλήκτρα (τουλάχιστον).
- IBM PC/AT compatible.
- Περιλαμβάνουν αλφαριθμητικά πλήκτρα, σημεία στίξης και διάφορα λειτουργικά πλήκτρα.
- Περισσότερο χρησιμοποιούμενη διάταξη η QWERTY.

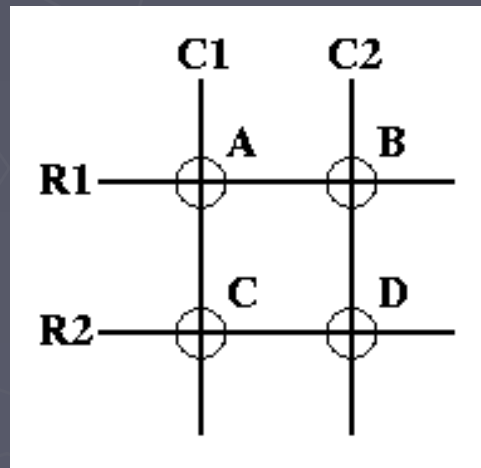
# Keyboard Hardware

- 8042 Microcontroller

Ελέγχει συνεχώς αν κάποιο πλήκτρο είναι πατημένο παράλληλα με τις κανονικές λειτουργίες του επεξεργαστή.

- Key matrix

Σειρές και στήλες καλωδίων, με κάθε κουμπί να λειτουργεί σαν διακόπτης.



# Keyboard Hardware(2)

## ► Keybounce

Οι μηχανικές επαφές δεν κλείνουν το κύκλωμα άμεσα, αλλά ταλαντώνονται για κάποιο χρόνο.

## • Λύση

Ο έλεγχος της επαφής γίνεται κάθε 10ms.

# Scan Codes

- ▶ Είναι ο binary κωδικός που στέλνει ο 8042 στον on-board controller.
- ▶ Ορισμένα πλήκτρα όπως το shift, ctrl και alt έχουν νόημα μόνο όταν είναι πατημένα.
- ▶ Για κάθε πλήκτρο υπάρχουν 2 scan codes:
  - down code
  - up code

# Scan Codes(2)

**Table 72: PC Keyboard Scan Codes (in hex)**

Key	Down	Up	Key	Down	Up	Key	Down	Up	Key	Down	Up
Esc	1	81	[{	1A	9A	, <	33	B3	<i>center</i>	4C	CC
1 !	2	82	]}	1B	9B	. >	34	B4	<i>right</i>	4D	CD
2 @	3	83	Enter	1C	9C	/ ?	35	B5	+	4E	CE
3 #	4	84	Ctrl	1D	9D	R shift	36	B6	<i>end</i>	4F	CF
4 \$	5	85	A	1E	9E	* PrtSc	37	B7	<i>down</i>	50	D0
5 %	6	86	S	1F	9F	alt	38	B8	<i>pgdn</i>	51	D1
6 ^	7	87	D	20	A0	space	39	B9	<i>ins</i>	52	D2
7 &	8	88	F	21	A1	CAPS	3A	BA	<i>del</i>	53	D3
8 *	9	89	G	22	A2	F1	3B	BB	/	E0 35	B5
9 (	0A	8A	H	23	A3	F2	3C	BC	<i>enter</i>	E0 1C	9C
0 )	0B	8B	J	24	A4	F3	3D	BD	F11	57	D7
- _	0C	8C	K	25	A5	F4	3E	BE	F12	58	D8
= +	0D	8D	L	26	A6	F5	3F	BF	ins	E0 52	D2
Bksp	0E	8E	::	27	A7	F6	40	C0	del	E0 53	D3
Tab	0F	8F	“	28	A8	F7	41	C1	home	E0 47	C7
Q	10	90	~	29	A9	F8	42	C2	end	E0 4F	CF
W	11	91	L shift	2A	AA	F9	43	C3	pgup	E0 49	C9
E	12	92	\	2B	AB	F10	44	C4	pgdn	E0 51	D1

# On-board controller

- ▶ Παίρνει τα scan codes από το πληκτρολόγιο και:
  - Δημιουργεί interrupt (int 9) στον επεξεργαστή.
  - Βάζει το scan code στη θύρα 60h.
- Scan code conversion.
  - Στην πραγματικότητα τα σήματα μεταξύ των 2 controllers (on-board και 8042) δεν είναι τα scan codes που φτάνουν στην ISR.

# Interrupt Service Routine

- ▶ Διαβάζει το scan code που της δίνει ο on-board controller.
- ▶ Επεξεργάζεται το scan code και με βάση την κατάσταση των modifier keys επιστρέφει τον πραγματικό κωδικό του πλήκτρου που πατήθηκε.



# Interrupt Service Routine(2)

- ▶ Μεταφράζει δύο ειδών πλήκτρα σε 16-bit κωδικούς:
  - Πλήκτρα με αντιστοιχία σε ASCII κωδικό.
    - ▶ Low order byte--→ASCII
    - ▶ High order byte--→scan code
  - Πλήκτρα χωρίς αντιστοιχία σε ASCII κωδικό (extended).
    - ▶ Low order byte--→0
    - ▶ High order byte--→extended code

# DOS Services

- ▶ Προσέγγιση εργαστηρίου.
- ▶ Έτοιμες συναρτήσεις για χρήση περιφερειακών συσκευών.
- ▶ Int 21h , Ah=07h → direct console input without echo.
  - Περιμένει για είσοδο του keyboard μέχρι να πατηθεί ένα πλήκτρο

# Int 21,07 (μειονεκτήματα)

- ▶ Είναι γραμμένη για real mode και δε μπορούμε να τη χρησιμοποιήσουμε μέσω protected mode.
- ▶ Busy waiting.
- ▶ Εναλλακτικά χρησιμοποιήσαμε ISR για υλοποίηση της ίδιας λειτουργικότητας άμεσα από protected mode.

# Υλοποίηση

- ▶ Προέκταση της εργαστηριακής άσκησης με τον editor.
- ▶ Αλλαγές/Προσθήκες που έγιναν στην άσκηση:
  - IDT
  - Keyboard ISR
  - Conversion table
  - Scheduler
  - Parser

# IDT

Κατασκευάζουμε Interrupt Gate Descriptor στη θέση 9 του IDT ώστε να εξυπηρετηθεί η Int 9 του πληκτρολογίου (offset `keyboard_isr`, selector `CODE_SEL`).

# Keyboard ISR

- ▶ Παίρνει το scan code του πληκτρολογίου από τη διεύθυνση 60h.
- ▶ Καλεί τη συνάρτηση PutInBuffer η οποία:
  - Ελέγχει αν το scan code αντιστοιχεί σε πάτημα κάποιου από τα L/R Shift, L/R Ctrl, CapsLock, NumLock.
  - Για κάθε άλλο πλήκτρο καλεί την Convert.
    - ▶ Convert: Ελέγχει τα KbdFlags και ανάλογα με το ποιο modifier key είναι πατημένο μετακινείται στην κατάλληλη γραμμή και στήλη του πίνακα conversion table.
  - Αγνοεί up codes από πλήκτρα εκτός των modifiers.

# Conversion Table

- Το Conversion Table έχει την εξής μορφή:

	<b>Norm</b>	<b>shft</b>	<b>ctrl</b>	<b>alt</b>	<b>num</b>	<b>caps</b>	<b>shcap</b>	<b>shnum</b>	
<b>word</b>	1071h,	1051h,	1011h,	1000h,	1071h,	1051h,	1051h,	1071h	;Q
<b>word</b>	1177h,	1057h,	1017h,	1100h,	1077h,	1057h,	1057h,	1077h	;W
<b>word</b>	1265h,	1245h,	1205h,	1200h,	1265h,	1245h,	1245h,	1265h	;E
<b>word</b>	1372h,	1352h,	1312h,	1300h,	1272h,	1252h,	1252h,	1272h	;R
<b>word</b>	1474h,	1454h,	1414h,	1400h,	1474h,	1454h,	1454h,	1474h	;T
<b>word</b>	1579h,	1559h,	1519h,	1500h,	1579h,	1559h,	1579h,	1559h	;Y
<b>word</b>	1675h,	1655h,	1615h,	1600h,	1675h,	1655h,	1675h,	1655h	;U
<b>word</b>	1769h,	1749h,	1709h,	1700h,	1769h,	1749h,	1769h,	1749h	;I

# Conversion Table(2)

- ▶ Πολλαπλασιάζει επί 8 το scan code για να μετακινηθεί στην κατάλληλη γραμμή του πίνακα.
- ▶ Προσθέτει ένα index ανάλογα με το modifier key που έχει πατηθεί και επιστρέφει την τιμή στη μεταβλητή key.
- ▶ Κάνει set την Keyboard\_data ώστε να ενημερωθεί ο scheduler ότι υπάρχει πλήκτρο για ανάγνωση και να περάσει τον έλεγχο στον parser.

Μένει να στείλουμε EOI στο 8259 για να μπορούμε να δεχτούμε επιπλέον διακοπές από το πληκτρολόγιο.



# Scheduler

- ▶ Ο scheduler στο κύριο loop του ελέγχει εάν υπάρχει data από το πληκτρολόγιο.
  - Αν ναι δίνει τον έλεγχο στον parser.
  - Αν όχι καλεί συνεχώς το display.
- ▶ Ο scheduler σταματάει εάν οι δύο μεταβλητές active1, active2 γίνουν 0 (πατηθεί ESC).
- ▶ Η εφαρμογή επιστρέφει σε DOS καλώντας μέσω του TASK\_RM την real-mode συνάρτηση exit\_point.

# Parser

- ▶ Φορτώνει το αρχείο μέσω real mode
- ▶ Στο main loop του που περνάει μόνο εφόσον του έχει δοθεί ο έλεγχος από τον scheduler.
- ▶ Ελέγχει τη μεταβλητή key στην οποία έχει τοποθετηθεί ο κωδικός του πλήκτρου από την ISR.
  - Αν το Low Order Byte = 0 ,σημαίνει ότι η ISR επέστρεψε extended κωδικό (arrow keys,F2).
  - Αν το Low Order Byte != 0 ,τότε το low order byte είναι ο κωδικός ASCII του πλήκτρου που πατήθηκε (escape,characters,backspace).
- ▶ Κάνει reset την keyboard\_data και δίνει τον έλεγχο στον scheduler.

