(Κεφάλαιο 2.7 και 12)

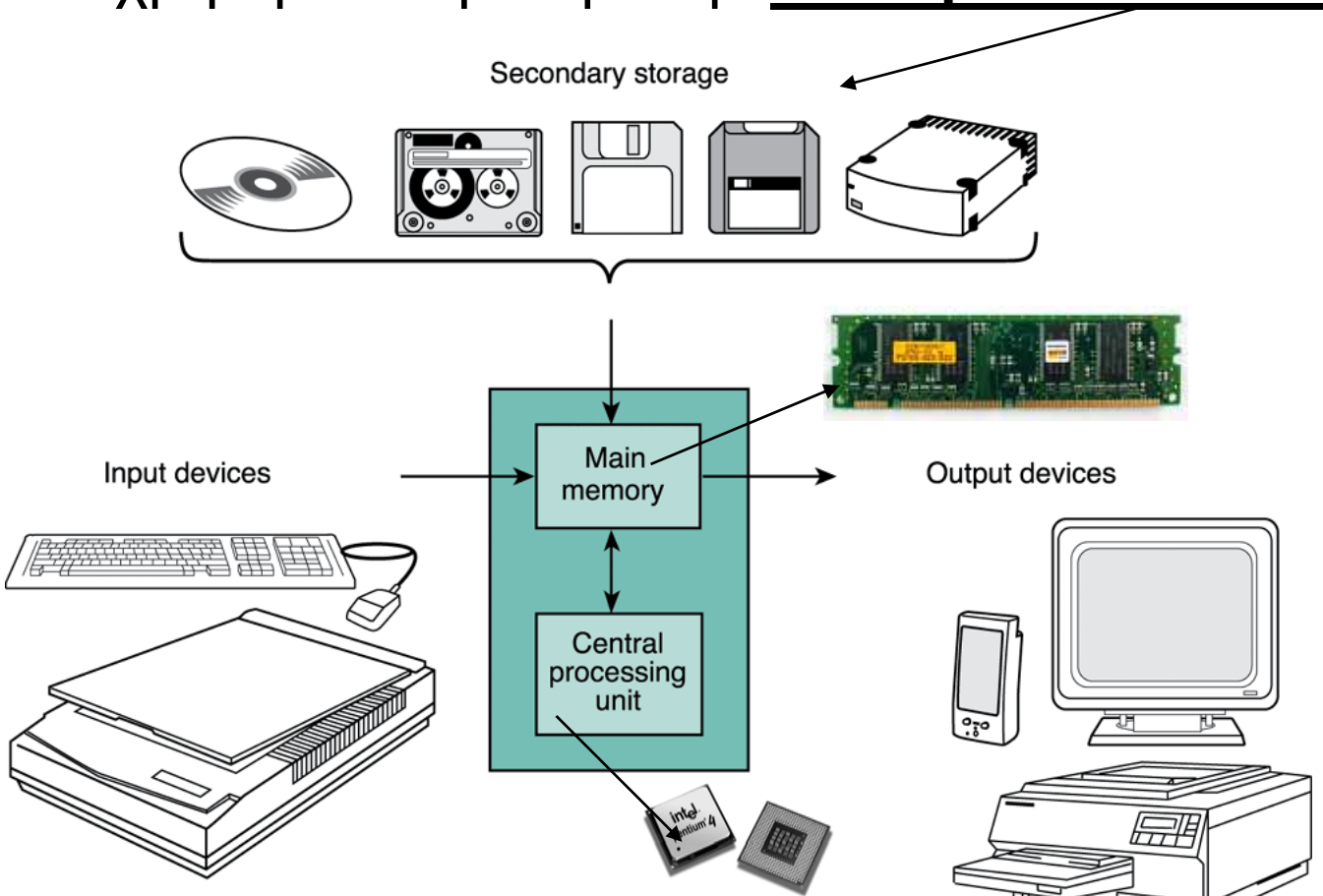
Αρχεία στην C

(Διάλεξη 13)

Διδάσκων: Δημήτρης Ζεϊναλιπούρ

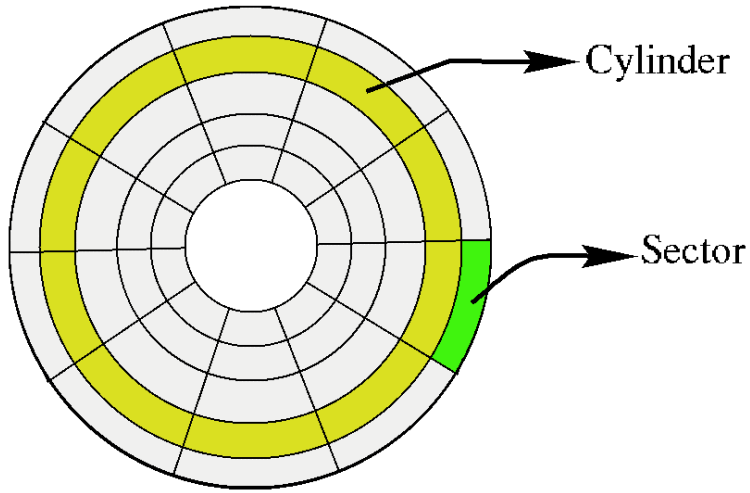
Επανάληψη στην Αποθήκευση (Storage)

Για να αποθηκεύσουμε δεδομένα από ένα πρόγραμμα, πρέπει να χρησιμοποιήσουμε την **Δευτερεύουσα Μνήμη**



Επανάληψη στην Αποθήκευση (Storage)

- Η πιο συνηθισμένη μορφή δευτερεύουσας μνήμη η οποία αξιοποιείτε από ένα πρόγραμμα είναι ο Μαγνητικός Δίσκος.
- **Μαγνητικός Δίσκος:** Αποτελείτε από ένα ή περισσότερους δίσκους με μαγνητική επικάλυψη
- Τα δεδομένα αποθηκεύονται σε τομείς (**sectors**).
- Οι δίσκοι διαθέτουν την δυνατότητα **σειριακής** αλλά και **άμεσης** πρόσβασης στα δεδομένα.

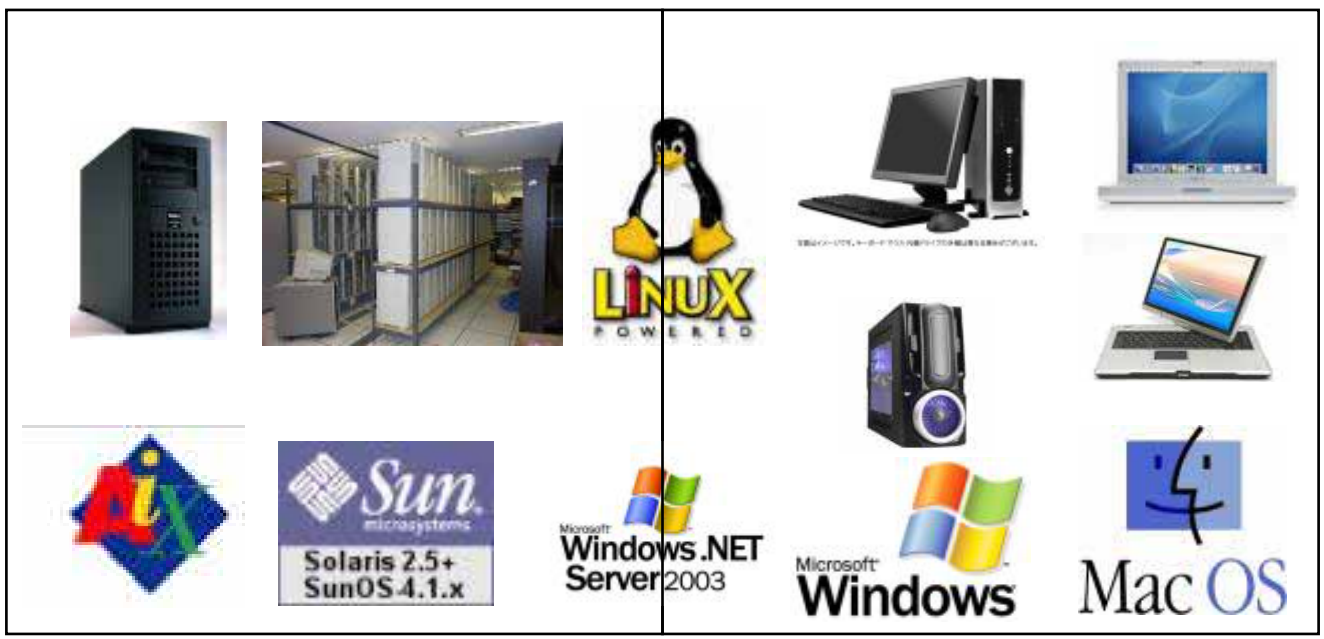


Επανάληψη στα Λειτουργικά Συστήματα (Operating Systems)

- Ποιος Διαχειρίζεται τους Πόρους (Μνήμη, Δίσκο, Επεξεργαστή, κτλ) σε ένα Υπολογιστικό Σύστημα?

ΤΟ ΛΕΙΤΟΥΡΓΙΚΟ ΣΥΣΤΗΜΑ

(Operating System)



Λειτουργικά Συστήματα και Συστήματα Διαχείρισης Αρχείων

- Κάθε γλώσσα προγραμματισμού έχει κάποια βιβλιοθήκη η οποία προσφέρει λειτουργίες πρόσβασης σε αρχεία αποθηκευμένα σε μονάδες Δευτερεύουσας Μνήμης.

- Στην C, οι λειτουργίες αυτές προσφέρονται από την βιβλιοθήκη <stdio.h>.

- Η Βιβλιοθήκη επικοινωνεί με το **υποσύστημα διαχείρισης αρχείων (File System)** του **λειτουργικού συστήματος** το οποίο στην συνέχεια διεκπεραιώνει τις λειτουργίες που ζητά ο προγραμματιστής. (το filesystem στα Windows είναι το FAT32, NTFS)

- Ας δούμε τι περιέχουν εσωτερικά τα αρχεία (δηλαδή πως αποθηκεύουν τις πληροφορίες)…

Αρχεία Κειμένου

- Θυμηθείτε ότι το **αρχείο** είναι μία ακολουθία bytes.
  π.χ. 01100011 01100001 01110010…

  ascii:99 ascii:97 ascii:114

  c a r
- Οι χαρακτήρες αυτοί είναι αποθηκευμένοι σειριακά στο αρχείο και διαχωρίζονται με διάφορους ειδικούς χαρακτήρες

Αρχείο όπως το βλέπει ο χρήστης

```
Car
Test   hello
```

Αρχείο στην Πραγματικότητα

Car\nTest\thelloEOF

Τέλος αρχείου ascii: -1

Αρχεία Κειμένου

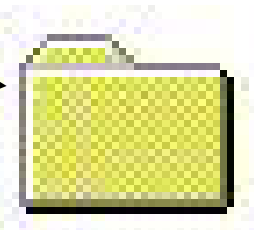
Ο Πίνακας ASCII – Αρκετοί χαρακτήρες είναι κρυμμένοι – δηλαδή δεν φαίνονται στην οθόνη και στα αρχεία

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Spa	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DE

Δυαδικά Αρχεία

Καλά, όλα τα αρχεία περιέχουν Χαρακτήρες ASCII? **ΟΧΙ**

Ας δούμε πως **είναι κωδικοποιημένο** ένα αρχείο εικόνας GIF (Graphics Interchange Format), συγκεκριμένα την γνωστή εικόνα του **folder στα Windows**



Αρχή

```
47 49 46 38 39 61 0D 00  0B 00 C4 00 00 FF FB FF  GIF89a....Δ..□ù□
00 00 FF 63 65 00 FF FF  00 E5 E4 6F FF FF 9C E4  ..□ce.□□.εδo□□□δ
E1 77 9C 9A 63 E8 E1 77  E5 DF 77 E5 E0 77 E8 E0  αw□□cθαwεíwεûwθû
77 E4 DB 77 E4 D9 77 E4  D7 76 E4 D6 77 E4 D7 79  wδŸwδΩwδXvδΦwδXy
FF CF 00 CE 9A 00 00 00  00 FF FF FF 00 00 00 00  □O.Σ□....□□□....
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00  00 00 00 00 00 21 F9 04  .............!ω.
```

Το σχήμα δείχνει ότι οι εικόνες (όπως και ΟΛΑ τα αρχεία) είναι ουσιαστικά μια σειρά από bytes (στο σχήμα κωδικοποιημένα στο δεκαεξαδικό σύστημα για να κερδίσουμε χώρο) $47_{16}=1000111_2$

Εισαγωγή στα Αρχεία

- Τώρα θα επικεντρωθούμε στις λειτουργίες εγγραφής και ανάγνωσης σε αρχεία της βιβλιοθήκης <stdio.h>.
- **Η επεξεργασία αρχείων γίνεται κατά τον ακόλουθο τρόπο:**

Α) Συσχετίζουμε μία οντότητα της C με το αρχείο (*άνοιγμα* του αρχείου)

Β) Πραγματοποιούμε τις λειτουργίες ανάγνωσης και εγγραφής δεδομένων

C) Κλείνουμε το αρχείο

Α) Άνοιγμα Αρχείων

Συνάρτηση fopen (stdio.h)

FILE *fopen(char *filename, char *mode);

- Η παράμετρος *filename* υποδεικνύει το όνομα του αρχείου που επιθυμούμε να ανοίξουμε
- Το Mode υποδεικνύει το είδος της πράξης (π.χ. Read, write, read-write, etc)

```
main() { FILE *fp;
  fp = fopen("myfile.txt", "r");
  …. }
```

Τύπος Ανοίγματος (εδώ READ)

Όνομα Αρχείου

Οντότητα Διαχείρισης Αρχείου.

Α) Άνοιγμα Αρχείων

Η παράμετρος mode υποδεικνύει τον τρόπο με τον οποίο θέλουμε να προσπελάσουμε το αρχείο

Mode	Σημασία
r	Μόνο **Ανάγνωση**. Αν το αρχείο δεν υπάρχει, επιστρέφεται NULL
w	Μόνο **Εγγραφή**. Αν το αρχείο δεν υπάρχει δημιουργείται, αν υπάρχει τα περιεχόμενά του καταστρέφονται.
a	**Προσθήκη**. Αν το αρχείο δεν υπάρχει, δημιουργείται.
r+	**Ανάγνωση** και **εγγραφή**. Αν το αρχείο δεν υπάρχει, επιστρέφεται NULL.
w+	**Ανάγνωση** και **εγγραφή**. Αν το αρχείο δεν υπάρχει δημιουργείται, αν υπάρχει τα περιεχόμενά του καταστρέφονται.
a+	**Προσθήκη** και **ανάγνωση**. Αν το αρχείο δεν υπάρχει, δημιουργείται.

C) Κλείσιμο Αρχείων

Συνάρτηση **fclose (stdio.h)**

int fclose(FILE *fp);

Η παράμετρος *fp* υποδεικνύει την **Οντότητα Διαχείρισης Αρχείου.**

```
main() { FILE *fp;
        fp = fopen("myfile.txt", "r");
        fclose(fp);
…. }
```

Β) Άνοιγμα/ Εγγραφή/ Κλείσιμο Αρχείων

Γράψετε ένα απλό πρόγραμμα που προσθέτει την λέξη "Hello" σε ένα αρχείο.

```
#include <stdio.h>
int main() {
        FILE *fp;
        fp = fopen("myfile.txt", "a"); // άνοιγμα αρχείου
        fprintf(fp, "Hello");          // εκτύπωση σε αρχείο
        fclose(fp);                    // κλείσιμο αρχείου
        return 0;
}
```

Append mode

Ας δούμε τώρα τις εντολές fprintf, fclose

σε περισσότερο βάθος

Β) Άνοιγμα/ Εγγραφή/ Κλείσιμο Αρχείων

```
#include <stdio.h>

main() { FILE *fp;
  // Try to open the file
  if ((fp = fopen("myfile.txt", “a")) == NULL)  {
        printf("Error opening file\n");
        exit(1);
  }
  // Write to the file “Hello 5 6”
  fprintf(fp, "Hello%d %d“, 5,6);
  fclose(fp);
}
```

Το ίδιο πρόγραμμα με συνθήκη έλεγχου που ελέγχει αν άνοιξε το αρχείο

Π.χ. Αν γέμισε ο δίσκος τότε το fopen θα αποτύχει, και ο χρήστης του προγράμματος μπορεί να ειδοποιηθεί με μήνυμα λάθους

Β) Άνοιγμα / Ανάγνωση Ακέραιου / Κλείσιμο Αρχείων

```
#include <stdio.h>
main() { FILE *fp;
        int a;
        if ((fp = fopen("myfile.txt", “r")) == NULL)     {
                printf("Error opening file\n");
                exit(1);
        }
        // Διάβασε ένα αριθμό από το αρχείο
        fscanf(fp, “%d“, &a);

        // Close the file
        fclose(fp);
}
```

Mode Ανάγνωσης

Ανάγνωση από αρχείο ενός ακεραίου

Β) Άνοιγμα / Ανάγνωση Ακέραιου / Εκτύπωση / Κλείσιμο Αρχείων

```
#include <stdio.h>
main() { FILE *fp;
        int a;
        if ((fp = fopen("myfile.txt", “r")) == NULL)     {
                printf("Error opening file\n");
                exit(1);
        }
        // Διάβασε ένα αριθμό από το αρχείο
        fscanf(fp, “%d“, &a);
        // Εκτύπωσε τον αριθμό
        printf(“%d“, &a);
        // Close the file
        fclose(fp);
}
```

Ανάγνωση από αρχείο ενός ακεραίου και εκτύπωση στην οθόνη

Β) Άνοιγμα / Ανάγνωση Ακέραιου + Εκτύπωση x 2 / Κλείσιμο

```
#include <stdio.h>
main() { FILE *fp;
        int a;
        if ((fp = fopen("myfile.txt", “r")) == NULL)     {
                printf("Error opening file\n"); exit(1);
        }
        fscanf(fp, “%d“, &a);
        printf(“%d“, &a);

        fscanf(fp, “%d“, &a);
        printf(“%d“, &a);

        fclose(fp);
}
```

Ανάγνωση από αρχείο δυο ακέραιων και εκτύπωση στην οθόνη

Πρόγραμμα Ανάγνωσης Πολλών Αριθμών

Κάποιος σας δίδει ένα μεγάλο αρχείο αριθμών. Σας ζητά να βρείτε τον μέγιστο αριθμό στο αρχείο.

Myfile.txt

```
5
6
6 4     5       5       53
34      4       4       100
54
54
543
6
```

Πρόγραμμα Ανάγνωσης Πολλών Αριθμών

```
#include <stdio.h>
main() { FILE *fp;
        int a; int max=-1;
        if ((fp = fopen("myfile.txt", “r")) == NULL)          {
                printf("Error opening file\n"); exit(1);
        }
        while (fscanf(fp, "%d", &a) != EOF)     {
                //printf("%d\n", a);
                if (a>max) { max = a; }
        }
        printf(“max:%d”, max);    // Print the MAX

        fclose(fp);                // Close the file
}
```

Άσκηση

Τι ακριβώς κάνει το πιο κάτω πρόγραμμα?

```
#include <stdio.h>
#include <stdlib.h>
main() { FILE *fp;
        int i; int max=-1;
        srand(time(NULL));

        if ((fp = fopen("myfile.txt", "w")) == NULL)        {
                printf("Error opening file\n"); exit(1);
        }
        for (i=0;i<10;i++) {
                fprintf(fp, "%d\n", rand() % 1000);
        }
        // Close the file
        fclose(fp);
}
```