ΔΥΑΔΙΚΑ ΑΡΧΕΙΑ

- Τα δυαδικά αρχεία σχετίζονται με τις δυαδικές ροές. Σε ένα δυαδικό αρχείο τα δεδομένα γράφονται όπως ακριβώς αναπαρίστανται στη μνήμη του υπολογιστή. Σε αντίθεση με τα αρχεία κειμένου, δεν υπάρχουν χαρακτήρες τέλους γραμμής που χωρίζουν τα δεδομένα σε γραμμές.
- Το άνοιγμα ενός δυαδικού αρχείου γίνεται πάλι με την fopen() αλλά με την προσθήκη του χαρακτήρα 'b' αμέσως μετά τον χαρακτήρα που υποδηλώνει αν ανοίγουμε το αρχείο για διάβασμα, γράψιμο ή προσάρτηση:

```
fp = fopen("file.bin","rb"); /* άνοιγμα για διάβασμα */
fp = fopen("file.bin","wb"); /* άνοιγμα για γράψιμο */
fp = fopen("file.bin","ab"); /* άνοιγμα για προσάρτηση */
```

  όπου fp είναι δείκτης σε αρχείο.
- Το κλείσιμο του αρχείου γίνεται πάλι με την fclose().

ΔΥΑΔΙΚΑ ΑΡΧΕΙΑ

- Η ανάγνωση ή εγγραφή δεδομένων από/σε δυαδικά αρχεία γίνεται με άμεσο τρόπο, δηλαδή ολόκληρα μπλοκ δεδομένων διαβάζονται, μονομιάς, από το δίσκο στη μνήμη ή γράφονται από τη μνήμη στο δίσκο. Για παράδειγμα, μπορούμε να γράψουμε έναν ολόκληρο πίνακα δεδομένων στο δίσκο με μια μόνο κλήση της συνάρτησης άμεσης εξόδου fwrite() και να τον διαβάσουμε με μια κλήση της συνάρτησης άμεσης εισόδου δεδομένων fread().
- Η συνάρτηση βιβλιοθήκης fread() έχει τέσσερα ορίσματα και χρησιμοποιείται για το διάβασμα ενός μπλοκ δεδομένων από ένα δυαδικό αρχείο. Το πρώτο όρισμα είναι δείκτης στην περιοχή μνήμης που έχει δεσμευθεί για τα δεδομένα. Το δεύτερο όρισμα είναι το μέγεθος του κάθε στοιχείου που θα διαβαστεί σε bytes. Το τρίτο όρισμα καθορίζει το πλήθος των στοιχείων που θα διαβαστούν και το τέταρτο όρισμα είναι ο δείκτης στο αρχείο. Η fread() επιστρέφει το πλήθος των στοιχείων που διαβάστηκαν.

```
int fread(void *data, int nr_of_bytes, int N, FILE *fp)
```

  *Σημείωση: μέγεθος μπλοκ (σε bytes) = N * nr_of_bytes*

ΔΥΑΔΙΚΑ ΑΡΧΕΙΑ

- Η συνάρτηση βιβλιοθήκης fwrite() έχει παρόμοια σύνταξη με την fread(), (έχουν τα ίδια ορίσματα) και χρησιμοποιείται για την εγγραφή ενός μπλοκ δεδομένων σε ένα δυαδικό αρχείο. Η fwrite() επιστρέφει το πλήθος των στοιχείων που γράφτηκαν με επιτυχία.

```
int fwrite(void *data, int nr_of_bytes, int N, FILE *fp)
```

- Παράδειγμα:

```
float pin[100];
FILE *fp1, *fp2;

fp1 = fopen("infile.bin","rb");
fp2 = fopen("outfile.bin","wb");
. . .
fread(pin,sizeof(float),100,fp1);   /* block of 400 bytes */
/* ή ισοδύναμα:   fread(pin,100*sizeof(float),1,fp1); */
/* ή ισοδύναμα:   fread(pin,50*sizeof(float),2,fp1);   κ.λπ. */
. . .
fwrite(pin,100*sizeof(float),1,fp2);
```

Σημείωση: Είτε διαβάσουμε 100 φορές από 1 float αριθμό τη φορά είτε 50 φορές από 2 floats τη φορά … είτε 2 φορές από 50 floats τη φορά είτε και τους 100 floats μονομιάς είναι το ίδιο και το αυτό γιατί οι αριθμοί είναι αποθηκευμένοι στο αρχείο ο ένας μετά τον άλλον.

Τα ορίσματα της main

Κατά την εκτέλεση ενός προγράμματος της C από τη γραμμή εντολών, αμέσως μετά το όνομα του εκτελέσιμου αρχείου, μπορούμε να περνάμε πληροφορίες (τιμές, ονόματα αρχείων, κ.λπ.) με παραμετρικό τρόπο στο πρόγραμμα μέσω των ορισμάτων της main. Τα ορίσματα της main είναι τα argc και argv και ορίζονται ως εξής:

```
int main(int argc, char *argv[])
```

Το argc αναφέρεται στο συνολικό πλήθος των παραμέτρων στη γραμμή εντολών (μετράει το όνομα του εκτελέσιμου αρχείου καθώς και τις παραμέτρους που ακολουθούν – στην ίδια γραμμή και χωρίς παρενθέσεις) και το argv είναι ένας πίνακας με δείκτες στο όνομα του εκτελέσιμου και στις παραμέτρους που περνάμε στη main.

Παράδειγμα: Έστω ότι θέλουμε ένα πρόγραμμα (το cube) που να υψώνει έναν ακέραιο αριθμό (num) στον κύβο και η κλήση να γίνεται από τη γραμμή εντολής ως

```
>A:\cube num
```

Το παρακάτω πρόγραμμα κάνει ακριβώς αυτό.

```c
#include <stdio.h>
#include <stdlib.h>

#define CUBE(X) ((X)*(X)*(X))

int main(int argc, char *argv[])
{
  int num;

  if(argc != 2)
  {
    printf("Please type: cube number\n");
    exit(-1);
  }
  num = atoi(argv[1]);

  printf("%d\n",CUBE(num));

  return 0;
}
```

Παράδειγμα:

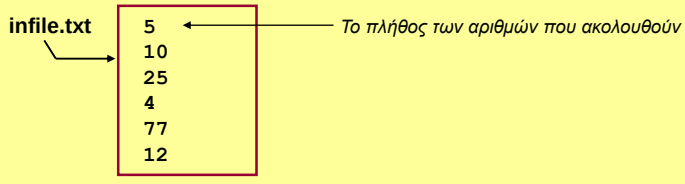
```
A:\>cube 25
15625
```

Να γραφεί πρόγραμμα mat_pow.c που να υψώνει τα στοιχεία του πίνακα που βρίσκει στο αρχείο του 1ου ορίσματος στη δύναμη που δηλώνεται στο 2ο όρισμα και να τοποθετεί τον νέο πίνακα στο αρχείο του 3ου ορίσματος. Ας θεωρήσουμε ότι το πρώτο αρχείο είναι αρχείο κειμένου, το δεύτερο είναι δυαδικό και ότι ο πρώτος αριθμός (ακέραιος) στο κάθε αρχείο είναι η διάσταση του πίνακα.

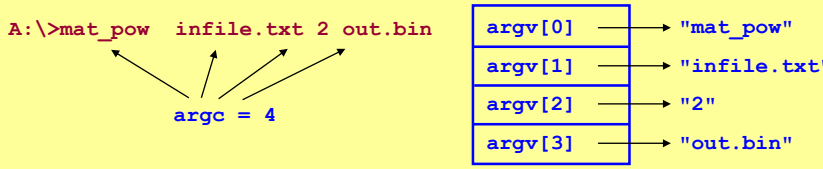
```c
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
  int i,dim,n,m,r;
  int *data;
  FILE *ifp,*ofp;

  if(argc != 4)
  { printf("mat_pow   in_txt_file   power   out_bin_file\n");
    exit(-1); }

  if((n = atoi(argv[2])) <= 0)   /* n holds the power:  n > 0 */
  { printf("The power must be a positive integer.\n");
    exit(-1); }

  if((ifp = fopen(argv[1],"r")) == NULL)
  { printf("Can\'t open file %s for reading...\n",argv[1]);
    exit(-1); }
```

```
  fscanf(ifp,"%d\n",&dim);
  data = (int *) malloc(dim*sizeof(int));
  for(i=0; i<dim; i++)
  {
    fscanf(ifp,"%d\n",data+i); /* Read next integer number */
    m = n;
    r = *(data+i);
    while( --m > 0)
      *(data+i) *= r; /* Now data[i] is raised to the power of n */
  }
  fclose(ifp);

  if((ofp = fopen(argv[3],"wb")) == NULL)
  {
    printf("Can\'t open binary file %s for writing...\n",argv[3]);
    exit(-1);
  }
  fwrite(&dim,sizeof(int),1,ofp);
  fwrite(data,dim*sizeof(int),1,ofp);
  fclose(ofp);
  free(data);
  return 0;
}
```

Παράδειγμα αρχείου εισόδου



Παράδειγμα εκτέλεσης προγράμματος από τη γραμμή εντολών:



Επισκόπηση δυαδικού αρχείου εξόδου out.bin με hex viewer

```
000000h: 05 00 00 00 64 00 00 00 71 02 00 00 10 00 00 00 ; ....d...q.......
000010h: 29 17 00 00 90 00 00 00                         ; )...□...
```

4 bytes/int

α/α 1ου byte γραμμής

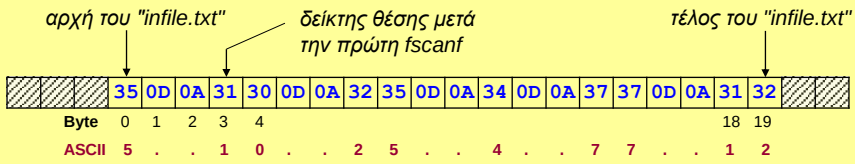
χαρακτήρες που αντιστοιχούν στα bytes

Σειρά αποθήκευσης των bytes: LSB → MSB (little endian).

Π.χ. $29\ 17\ 00\ 00_{16} \rightarrow 00\ 00\ 17\ 29_{16} = 1x16^3+7x16^2+2x16^1+9 = 5929_{10} = 77^2$

ΣΕΙΡΙΑΚΗ / ΤΥΧΑΙΑ ΠΡΟΣΠΕΛΑΣΗ ΑΡΧΕΙΩΝ

- Οι συναρτήσεις που έχουμε συναντήσει μέχρι στιγμής για το διάβασμα ή την εγγραφή δεδομένων σε αρχεία χαρακτήρων ή σε δυαδικά αρχεία διαβάζουν ή αποθηκεύουν δεδομένα με τη σειρά, ξεκινώντας από την αρχή του αρχείου (αν πρόκειται για διάβασμα ή για εγγραφή σε νέο αρχείο) ή ξεκινώντας από το τέλος του αρχείου αν πρόκειται για προσάρτηση σε προϋπάρχον αρχείο.
- Όλα τα παραδείγματα που έχουμε δει μέχρι τώρα είτε με αρχεία κειμένου είτε με δυαδικά αρχεία αφορούν τη λεγόμενη **σειριακή προσπέλαση αρχείων**. Στη σειριακή προσπέλαση, ο **δείκτης θέσης** του αρχείου καθορίζει ποιο θα είναι το επόμενο byte που θα προσπελαστεί (για διάβασμα ή γράψιμο) με την αρχή του αρχείου (πρώτο byte) να έχει δείκτη θέσης 0. Π.χ. κατάσταση μετά την πρώτη fscanf του mat_pow.c που διαβάζει τον πρώτο ακέραιο και προσπερνάει τους επόμενους λευκούς χαρακτήρες:



ΣΕΙΡΙΑΚΗ / ΤΥΧΑΙΑ ΠΡΟΣΠΕΛΑΣΗ ΑΡΧΕΙΩΝ

- Στην περίπτωση **τυχαίας προσπέλασης** παρέχεται η δυνατότητα α) μεταφοράς του δείκτη θέσης σε συγκεκριμένο σημείο (byte) του αρχείου και β) προσπέλασης των δεδομένων (για εγγραφή ή ανάγνωση) ξεκινώντας από αυτό το σημείο.
- Ο τρόπος που γίνεται η τυχαία προσπέλαση αρχείων στη C είναι μέσω της συνάρτησης βιβλιοθήκης fseek()

```
int  fseek(FILE *fp, long offset, int origin)
```

όπου το origin δηλώνει τη θέση αναφοράς (αρχή αρχείου, τρέχουσα θέση, τέλος αρχείου), το offset δηλώνει την απόσταση από το origin σε bytes (θετικός ή αρνητικός αριθμός ανάλογα με την περίσταση) και fp προσδιορίζει το αρχείο. Οι τρεις τιμές της θέσης αναφοράς είναι κωδικοποιημένες ως εξής (βλ. stdio.h):

`SEEK_SET`	ή	`0`	Αναζήτηση αναφορικά με την αρχή του αρχείου
`SEEK_CUR`	ή	`1`	Αναζήτηση αναφορικά με την τρέχουσα θέση
`SEEK_END`	ή	`2`	Αναζήτηση αναφορικά με το τέλος του αρχείου

Η fseek() επιστρέφει 0 αν είναι επιτυχής και μη-μηδενική τιμή σε περίπτωση αποτυχίας.

Άλλες χρήσιμες συναρτήσεις

- Η συνάρτηση βιβλιοθήκης rewind() τοποθετεί τον δείκτη θέσης στην αρχή του αρχείου:

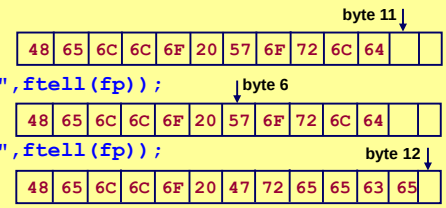
```
void rewind(FILE *fp)
```

- Η συνάρτηση ftell() επιστρέφει την τρέχουσα θέση του δείκτη θέσης. Σε περίπτωση λάθους επιστρέφει το -1:

```
long ftell( FILE *fp)
```

Παράδειγμα χρήσης των ftell() και fseek().

```
#include <stdio.h>
int main()
{
  FILE *fp;

  fp = fopen("hello.txt","w");
  fputs("Hello World",fp);

  printf("Position at byte: %d\n",ftell(fp));
  fseek(fp,6,SEEK_SET);

  printf("Position at byte: %d\n",ftell(fp));
  fputs("Greece",fp);
  fclose(fp);
  return 0;
}
```



Άλλες χρήσιμες συναρτήσεις

- Η συνάρτηση βιβλιοθήκης rewind() τοποθετεί τον δείκτη θέσης στην αρχή του αρχείου:

```
void rewind(FILE *fp)
```

- Η συνάρτηση ftell() επιστρέφει την τρέχουσα θέση του δείκτη θέσης. Σε περίπτωση λάθους επιστρέφει το -1:

```
long ftell( FILE *fp)
```

Παράδειγμα χρήσης των ftell() και fseek().

```
#include <stdio.h>
int main()
{
  FILE *fp;

  fp = fopen("hello.txt","w");
  fputs("Hello World",fp);

  printf("Position at byte: %d\n",ftell(fp));
  fseek(fp,6,SEEK_SET);

  printf("Position at byte: %d\n",ftell(fp));
  fputs("Greece",fp);
  fclose(fp);
  return 0;
}
```