Τμήμα Μηχανικών Η/Υ & Πληροφορικής
Πανεπιστήμιο Πατρών

Εισαγωγή στον Προγραμματισμό

Αρχεία

Επεξεργασία Αρχείων στη C

- Με τη χρήση `FILE *` αναπαριστάνουμε δείκτη σε ένα αρχείο.
- `fopen` χρησιμοποιείται για να ανοίγει ένα αρχείο. Επιστρέφει την ειδική τιμή `NULL` για να δείξει ότι δεν μπορεί να ανοίξει ένα αρχείο.

```
FILE *fptr;
char filename[]= "file2.dat";
fptr= fopen (filename,"w");
if (fptr == NULL) {
    fprintf (stderr, "ERROR");
     /* DO SOMETHING */
}
```

Αρχεία

- **αρχεία κειμένου (text files) και δυαδικά αρχεία (binary files)**
- Τα αρχεία κειμένου αποτελούνται από μία η περισσότερες γραμμές, οι οποίες περιέχουν χαρακτήρες σε μορφή αναγνώσιμου (κυρίως ASCII κώδικα)
- Ένα δυαδικό αρχείο δεν περιέχει απαραίτητα αναγνώσιμους χαρακτήρες
- Στα αρχεία κειμένου χρησιμοποιούμε κυρίως τις συναρτήσεις fscanf(), fprintf(), fgetc()/getc(), fputc()/putc(), fgets(), fputs()
- Στα δυαδικά αρχεία κειμένου χρησιμοποιούμε κυρίως τις συναρτήσεις fread(), fwrite(), σε συνδυασμό με την fseek()
- Τα αρχεία κειμένου αποθηκεύουν τις τιμές των μεταβλητών σε μορφή χαρακτήρων (συνεπώς έχουμε μετατροπή αριθμών σε γράμματα), ενώ στα δυαδικά αρχεία η αποθήκευση διατηρεί την δυαδική αναπαράσταση των τιμών στη μνήμη του υπολογιστή. Τα δυαδικά αρχεία είναι χρήσιμα κυρίως για αναπαράσταση πινάκων εγγραφών, ενώ τα αρχεία κειμένου για εφαρμογές φυσικής γλώσσας.

Τρόποι Ανοίγματος Αρχείου

- FILE * fopen(const char * filename, const char * mode)
- Το δεύτερο όρισμα της `fopen` καθορίζει τον *τρόπο* με τον οποίο ανοίγουμε ένα αρχείο.

  Υπάρχουν οι εξής επιλογές:
- "r" ανοίγει ένα αρχείο για διάβασμα
- "w" ανοίγει ένα αρχείο για εγγραφή
- "a" ανοίγει ένα αρχείο για προσθήκη / εγγραφή
- “b” δυαδικά αρχείο (binary)
- rb, wb, ab, r+, w+, a+, rb+,wb+,ab+

Διαχείριση Αρχείων

- Κλείσιμο αρχείων
  int *fclose* (FILE * fp);
- Είσοδος – έξοδος χαρακτήρων
  int *putc*(int c, FILE * fp);
  (ισοδύναμα *fputc*())
  int *getc*(FILE * fp);
  (ισοδύναμα *fgetc*())
- Είσοδος – έξοδος συμβολοσειρών
  int *fputs* (const char * s, FILE * fp);
  char * *fgets* (char * s, int n, FILE * fp);

- int *feof(FILE* *fp)

  Not 0: end of file

- void *rewind* (FILE *fp);
- int *fflush(FILE* *fp)

  0: success, EOF: error

- int *fseek*(FILE *fp, long int apostasi, int thesi)

  thesi=SEEK_SET (0), SEEK_CUR (1), SEEK_END (2)

  int *ftell*(FILE *fp)

- int fread(void *buffer, int arbytes, int fores, FILE *fp)

  Διάβασε fores στοιχεία, μεγέθους arbytes το καθένα, στο buffer array

- int fwrite(void *buffer, int arbytes, int fores, FILE *fp)

  Γράψε fores στοιχεία, μεγέθους arbytes το καθένα, από το buffer array

Παράδειγμα (1)

```
#define getchar()  getc(stdin)
#define putchar(c) putc((c), stdout)

char *fgets(char *s, int n, FILE *iop)
 { register int c; register char *cs; cs=s;
    while (--n>0 && (c=getc(iop))!=EOF)
                if ((*cs++=c)==‘\n’) break;
    *cs=‘\0’; return (c==EOF && cs==s)?NULL:s;
}


int fputs(char *s, FILE *iop)
{ int c;
  while (c=*s++) putc(c,iop);
  return ferror(iop)?EOF:0;
}
```

Γράφοντας σε Αρχείο με *fprintf*

- Η *fprintf* δουλεύει όπως η *printf* και η *sprintf* με την διαφορά ότι το πρώτο όρισμα είναι ένας δείκτης σε αρχείο.

```
FILE *fptr;
fptr= fopen ("file.dat","w");
/* Check it's open */
fprintf (fptr,"Hello World!\n");
```

- Θα μπορούσαμε επίσης να διαβάσουμε αριθμούς από ένα αρχείο χρησιμοποιώντας *fscanf* – αλλά υπάρχει και καλύτερος τρόπος.

Διαβάζοντας από Αρχείο με *fgets*

- *fgets* είναι ένας καλύτερος τρόπος ανάγνωσης από αρχείο
- Μπορούμε να διαβάσουμε από αρχείο με *fgets*

```
FILE *fptr;
char line [1000];
/* Open file and check it is open */
while (fgets(line,1000,fptr) != NULL) {
   printf ("Read line %s",line);
```

fgets έχει 3 ορίσματα, μία συμβολοσειρά, ένα μέγιστο αριθμό χαρακτήρων προς ανάγνωση και ένα δείκτη αρχείου. It returns NULL if there is an error (such as EOF)

fgets(char*s, int n, FILE *fstream); (διαβάζει το πολύ n-1 χαρακτήρες σταματώντας αν συναντήσει χαρακτήρα νέας γραμμής)

Κλείνοντας ένα Αρχείο

- Μπορούμε να κλείσουμε ένα αρχείο απλώς χρησιμοποιώντας *fclose* και το δείκτη αρχείου. Ακολουθεί παράδειγμα "hello files".

```
FILE *fptr;
char filename[]= "myfile.dat";
fptr= fopen (filename,"w");
if (fptr == NULL) {
    printf ("Cannot open file to write!\n");
    exit(-1);
}
fprintf (fptr,"Hello World of filing!\n");
fclose (fptr);
```

Παράδειγμα (3)

Να υλοποιηθούν και να χρησιμοποιηθούν οι ακόλουθες συναρτήσεις:

1. int **createRandFile**(char *fileName,int N): Δημιουργεί ένα αρχείο με όνομα fileName στο οποίο γράφει N τυχαίους αριθμούς. Επιστρέφει 1 σε επιτυχία και -1 σε αποτυχία
2. int ***loadData**(char *fileName,int N): Διαβάζει από το αρχείο fileName N τυχαίους αριθμούς και τους επιστρέφει σε ένα δείκτη ακεραίων.
3. int **writeArray2File**(char *fileName,int *array,int N): Γράφει τους αριθμούς του διανύσματος array στο αρχείο fileName. Επιστρέφει 1 σε επιτυχία και -1 σε αποτυχία.

```
int createRandFile(char *fileName,int N)
    {
        FILE *fp;
        int i;

        fp = fopen(fileName,"w");
        if (fp)
        {
            srand((unsigned)time(NULL));
            for (i=0;i<N;i++)
            {
                fprintf(fp,"%d\n",(MAX*rand())/RAND_MAX);
            }
            fclose(fp);
        return 1;
        }
        else
            return -1;
    }
```

stdlib.h → srand, rand

time.h -> time()

```
int *loadData(char *fileName,int N)
    {
        int *array = (int*)malloc(sizeof(int)*N);
        FILE *fp;
        int i;

        fp = fopen(fileName,"r");
        if (fp)
        {
            for (i=0;i<N;i++)
            {
                fscanf(fp,"%d",&array[i]);
            }
            fclose(fp);
            return array;
        }
        else
            return NULL;
    }
```

```
int writeArray2File(char *fileName,int *array,int N)
    {
        FILE *fp;
        int i;

        fp = fopen(fileName,"w");
        if (fp)
        {
            for (i=0;i<N;i++)
            {
                fprintf(fp,"%d\n",array[i]);
            }
            fclose(fp);
            return 1;
        }
        else
        return -1;
    }
```