Ανάπτυξη και Σχεδίαση Λογισμικού

Η γλώσσα προγραμματισμού C

Γεώργιος Δημητρίου

Αλφαριθμητικά και Αρχεία

- Αλφαριθμητικά (strings)

- Αρχεία (files)
  - τα βασικά στοιχεία

Αλφαριθμητικά της C

- Συμβολοσειρές (= ακολουθίες χαρακτήρων)
  - Όχι ξεχωριστός τύπος
  - Πίνακες τύπου char
    - προσοχή στο χαρακτήρα τερματισμού!
  - Ιδιαίτερη μεταχείριση
  - Υποστήριξη βιβλιοθήκης

Δήλωση Αλφαριθμητικών

- Όπως οι απλοί πίνακες
- Διαφοροποίηση στην αρχικοποίηση

```
char message[20];
char x[] = “this is a string!”;
char y[10] = {‘s’, ‘t’, ‘r’, ‘i’, ‘n’, ‘g’};
char z[][10] = {“number1”, “number2”};
```

Αποθήκευση

- Κάθε χαρακτήρας ένα byte
  - εφόσον έχουμε ASCII χαρακτήρες

- Ειδικός χαρακτήρας τερματισμού: '\0'
  - Υπονοείται σε έκφραση διπλών εισαγωγικών

    "this is a string!" ← 18 χαρακτήρες
  - Δεν υπονοείται σε έκφραση απλών χαρακτήρων

    {'s', 't', 'r', 'i', 'n', 'g'} ← 6 χαρακτήρες

    {'s', 't', 'r', 'i', 'n', 'g', '\0'} 7 χαρακτήρες

Συνηθισμένο Λάθος

- Όχι χώρος για το χαρακτήρα τερματισμού!
  - Μπορεί να οδηγήσει σε απρόβλεπτα λάθη…

```
char x[6] = “string”;
printf(“%s”,x);
```

→ stringh-a▒▒▒(

```
char y[] = {‘e’, ‘r’, ‘r’, ‘o’, ‘r’};
printf(“%s”,y);
```

→ errorstringh-a▒▒▒(

Απρόβλεπτο Λάθος

```
int x = -10;
char y[] = {‘s’,‘o’,‘m’,‘e’,‘t’,‘h’,‘i’,‘n’,‘g’};
printf (“y=\"%s\"\nx=%d\n”,y,x);
y[9] = ‘\0’;
printf (“y=\"%s\"\nx=%d\n”,y,x);
```

Τι θα τυπωθεί;

```
y="something▒▒▒▒▒▒h-a▒▒▒("
x=-10
y="something"
x=-256
```

Πίνακες και Δείκτες

- Αλφαριθμητικό ως πίνακας

  char x[] = “string”;

- Αλφαριθμητικό ως δείκτης

  char *y = “string”;

- Ίδιος τρόπος αναφοράς, αλλά διαφορετική αποθήκευση

  x[2], *(x+1), y[0], *(y+3)

  y = “other string”;

  x = “you cannot do this”;

Είσοδος / Έξοδος

- Μορφότυπος %s για τις scanf() και printf()
- Συναρτήσεις gets() και puts()

  char * gets(char*); ← ανάγνωση από πληκτρολόγιο

  int puts(const char*);

  ← εκτύπωση στην οθόνη
- Συναρτήσεις sscanf() και sprintf()
  - Όμοιες με τις scanf() και printf() με επιπλέον πρώτο όρισμα ένα αλφαριθμητικό από όπου διαβάζουμε ή στο οποίο γράφουμε

Παράδειγμα

```
char x[30];
printf("what is your name?\n");
gets(x);
```

Τι γίνεται αν ο χρήστης είναι Ινδός και έχει μακρύ όνομα;

Κίνδυνος υπερχείλισης και σφάλματος!
Γενικά η gets() θεωρείται επικίνδυνη συνάρτηση και αποφεύγεται

Παράδειγμα

```
char *record[2] = {“Costas 2314 30 m”, “Eleni 2718 26 f”};
char name[10], all[100], gen;
int i, AM, age;
for (i = 0; i < 2; i++) {
        sscanf(record[i], “%s %d %d %c”, name, &AM, &age,
                &gen);
        if (gen == ‘m’)
                sprintf(all, “His name is %s, AM is %d, age is %d.”,
                        name, AM, age);
        else
                sprintf(all, “Her name is %s, AM is %d, age is %d.”,
                        name, AM, age);
        printf(“%s\n”,all);
}
```

Βιβλιοθήκη string

```
size_t strlen(const char *);
char *strcpy(char *, const char *);
char *strncpy(char *, const char *, size_t);
char *strcat(char *, const char *);
char *strncat(char *, const char *, size_t);
int strcmp(const char *, const char *);
int strncmp(const char *, const char *, size_t);
char *strchr(const char *str, int c);
…
```

Παράδειγμα

```
// replace each x with y, counting the replacements
int replace_x_with_y (char *str, char x, char y)
{
   int i = -1, count = 0;
   while (str[++i] != '\0') {
      if (str[i] != x) continue;
      str[i] = y;
      count++;
   }
   return count;
}
```

Παράδειγμα

```
// replace each x with y, counting the replacements
int replace_x_with_y (char *str, char x, char y)
{
   int i, count, n = strlen(str);
   for (count = 0, i = 0; i < n; i++) {
      if (str[i] != x) continue;
      str[i] = y;
      count++;
   }
   return count;
}
```

Παράδειγμα

```
// replace each x with y, counting the replacements
int replace_x_with_y (char *str, char x, char y)
{
   int count = 0;
   while (str = strchr(str, x)) {
      *str = y;
      count++;
   }
   return count;
}
```

Άλλες Συναρτήσεις

- Από βιβλιοθήκη stdlib

```
int atoi(const char *);
double atof(const char *);
long strtol(const char *, char **, int);
unsigned long strtoul(const char *, char **, int);
double strtod(const char *, char **);
```

- Από βιβλιοθήκη ctype

```
int isalnum(int);
int isalpha(int);
int isdigit(int);
…
```

Αρχεία της C

- Δομές με τις οποίες μπορούμε να ανοίξουμε, να κλείσουμε, να διαβάσουμε και να γράψουμε ένα αρχείο του δίσκου
  - Ο τύπος των δομών αυτών ονομάζεται FILE και ορίζεται στη βιβλιοθήκη stdio
  - Η μορφή μιας δομής FILE δε μας απασχολεί
- Διαχωρισμός σε αρχεία κειμένου και δυαδικά αρχεία
  - Προς το παρόν θα δούμε μόνο αρχεία κειμένου

Άνοιγμα Αρχείου

- Θέλουμε το όνομα του αρχείου
- Καλούμε τη συνάρτηση fopen():

  FILE * fopen(const char *, const char *);

  - Η δεύτερη παράμετρος είναι ο τύπος προσπέλασης του αρχείου
    - "r" για ανάγνωση, "w" για εγγραφή
- Αν το αποτέλεσμα είναι NULL, τότε συνέβη κάποιο σφάλμα στο άνοιγμα του αρχείου
  - Διαφορετικά κρατάμε το δείκτη στη δομή FILE και ξεχνάμε το όνομα του αρχείου

Είσοδος / Έξοδος

- Συναρτήσεις fscanf() και fprintf()
  - Όμοιες με τις scanf() και printf() με επιπλέον πρώτο όρισμα ένα δείκτη σε τύπο FILE από όπου διαβάζουμε ή όπου γράφουμε
  - Η πρώτη επιστρέφει το πλήθος στοιχείων που διαβάστηκαν επιτυχώς ή τη σταθερά EOF σε περίπτωση σφάλματος
  - Η δεύτερη επιστρέφει το πλήθος χαρακτήρων που εκτυπώθηκαν ή κάποιο αρνητικό αριθμό σε περίπτωση σφάλματος

Είσοδος / Έξοδος

- Συναρτήσεις fgetc() και fputc() για ανάγνωση και εγγραφή χαρακτήρα

  int fgetc(FILE *);

  int fputc(int, FILE *);

- Συναρτήσεις fgets() και fputs() για ανάγνωση και εγγραφή αλφαριθμητικού
  - Ανάγνωση με μέγιστο πλήθος χαρακτήρων

    char * fgets(char *, int, FILE *);

    int fputs(char *, FILE *);

Άλλες Συναρτήσεις Ε/Ε

- Κλείσιμο αρχείου

  int fclose(FILE *);

- Άδειασμα δομής εγγράψιμου αρχείου

  int fflush(FILE *);

- Έλεγχος τέλους αρχείου

  int feof(FILE *);

- Έλεγχος σφάλματος στην τελευταία προσπέλαση του αρχείου

  int ferror(FILE *);

Σειρά Προσπέλασης

- Σειριακή
  - Η δομή αρχείου διατηρεί έναν δείκτη που προχωράει σειριακά από την αρχή προς το τέλος του αρχείου
  - Με τη συνάρτηση rewind() επαναφέρουμε το δείκτη στην αρχή του αρχείου

    void rewind(FILE *);

- Τυχαία προσπέλαση…

Τι Μάθαμε Σήμερα

- Αλφαριθμητικά της C
  - Δήλωση και αρχικοποίηση
  - Αποθήκευση
  - Αναφορά ως πίνακας και ως δείκτης
  - Σφάλματα με αλφαριθμητικά
  - Συναρτήσεις χειρισμού αλφαριθμητικών
- Βασικά στοιχεία αρχείων της C
  - Κύριες λειτουργίες και βασικές συναρτήσεις προσπέλασης αρχείων κειμένου