Διαδικαστικός Προγραμματισμός

Βασίλης Παλιουράς
paliuras@ece.upatras.gr

Συναρτήσεις βασικής βιβλιοθήκης για αλφαριθμητικά

- πρότυπα στο `string.h`
- `char *strcpy (char *, const char *) ;`
- `int strcmp (const char *, const char *) ;`
- `char *strcat (char *, const char *) ;`
- `char *strchr (const char *, char) ;`
- `size_t strlen (const char *) ;`
- και άλλες…

- `size_t` τύπος μη αρνητικού ακέραιου
- `NULL` σταθερή τιμή μηδενικού δείκτη, null pointer constant, `(void *) 0`

char * strtok(**char** *, **const char** *)

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

int main(void) {
        char names[] = "Giannis,Kostas Maria:Thanassis";
        char delims[] = ", :";
        char * ch_ptr;

        printf ("names before: %s\n", names);
        ch_ptr = strtok(names, delims) ;
        printf("%s\n", ch_ptr);
        while ((ch_ptr = strtok(NULL, delims)) != NULL){
                printf("%s\n", ch_ptr);
        }
        printf("names after: %s\n", names);
        { int i;
                for(i=0; i<40; i++) {
                        printf("%c (%d) : ", names[i] , names[i]);
                }
        }

        return 0;
}
```

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

int main( void ) {
        char names[] = "Giannis,Kostas Maria:Thanassis";
        char delims[] = ", :";
        char * ch_ptr;

        printf ("names before: %s\n", names);
        ch_ptr = strtok(names, delims) ;
        while (ch_ptr != NULL){
                printf("%s\n", ch_ptr);
                ch_ptr = strtok(NULL, delims);
        }
        printf("names after: %s\n", names);
        { int i;
                for(i=0; i<40; i++) {
                        printf("%c (%d) : ", names[i] , names[i]);
                }
        }

        return 0;
}
```

```c
#include <stdio.h>
#include <string.h>
#define SIZE 40

int main(void ) {
        char names[] = "Giannis,Kostas Maria:Thanassis";
        char delims[] = ", :";
        char * ch_ptr;
        char * arg;

        printf ("names before: %s\n", names);
        arg = names;
        while ((ch_ptr = strtok(arg, delims))!= NULL){
                printf("%s\n", ch_ptr);
                arg = NULL;
        }
        printf("names after: %s\n", names);
        { int i;
                for (i=0; i<SIZE; i++) {
                        printf("%c (%d) : ", names[i] , names[i]);
                }
        }

        return 0;
}
```

char * strcat(**char** *, **const char** *)

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

int main(void) {
 char word[10] = "more ";
 char base[100] = "";

 int i;
 for (i=0; i<5; i++) {
   strcat(base, word);
   printf("%s %d\n", base, strlen(base));
 }

 return 0;
}
```

char * strchr(const char *, int)

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

int main(void) {
        char word[10] = "abcVcba";
        char ch = 'V';
        char * ch_ptr ;

        ch_ptr = strchr(word, ch);
        if (ch_ptr != NULL) {
                printf("%c at %p, (character %d)\n",
                        *ch_ptr , ch_ptr, ch_ptr-word);
                }
        else {
                printf("not existing");
             }
        return 0;
}
```

size_t strcspn (**const char** * str1, **const char** * str2);

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

int main(void) {
        char str1[ ] = "testabcabc";
        int i;

        i = strcspn(str1, "abc");

        printf("%d %c\n",i, str1[i]);

        return 0;
}
```

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

int teststrcspn(char *, char *);

int main(void) {
        char str1[ ] = "testabcabc";

        teststrcspn(str1,"abc");
        teststrcspn(str1,"fgh");

        return 0;
}

int teststrcspn(char *s, char *chars) {
        int i;

        i = strcspn(s, chars);

        printf("%2d %c %d\n",i, s[i], strlen(s));

        return 0;
}
```

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

int main(void) {
 char str[ ] = "testabcabc";
 char chars[] = "ac";
 char * str_i;
 int i;
 int loc = 0;

 str_i = str;                    /* addresses only ! */

 for (i=strcspn(str_i,chars); i < strlen(str_i);
                 i = strcspn(str_i, chars)) {
  loc = loc + i ;
  printf("%2d %c %d\n", loc, str_i[i], strlen(str_i));
  str_i = str_i + i + 1;
  loc ++;
 }

 return 0;
}
```

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

int main(void) {
        char str[ ] = "testabcabc";
        char chars[] = "ac";
        char * str_i;
        int i;
        int loc = 0;

        str_i = str;   /* addresses only ! */

        for (; (i = strcspn(str_i,chars)) < strlen(str_i); ){
                loc += i ;
                printf("%2d %c %d\n", loc, str_i[i], strlen(str_i));
                str_i += i + 1;
                loc ++;
                }

        return 0;
}
```

char * strpbrk (**const char** *, **const char** *);

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

int main(void) {
 char str[ ] = "testabcabc";
 char chars[] = "ac";
 char * str_i;

 str_i = strpbrk(str, chars);
 while (str_i!=NULL) {
  printf("%2d %c %d\n", str_i-str, *str_i, strlen(str_i));
  str_i = strpbrk(str_i+1, chars);
 }

 return 0;
}
```

char * strstr(const char *, const char *);

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

int main(void) {
        char str[ ] = "testabcabc";
        char * ch_ptr ;

        printf("%s\n", str);

        ch_ptr = strstr(str, "abc");
        strncpy(ch_ptr, "FGH", 3);

        printf("%s\n", str);

        return 0;
}
```

Buffer overflow!

```c
#include <stdio.h>
#include <string.h>
#define N 64

int main(void) {

        char str2[] = "abcdefghijklmaaaaaaaa" ;
        char str1[10] = "copy!";
        char word2[] = "aaa";

        printf("%s %s %p %p\n", str2, str1, str2, str1);

        strcpy(str1, str2);

        printf("%s %s\n", str2, str1);

        return 0;
}
```

```c
#include <stdio.h>
#include <string.h>
#define N 64

int main(void) {

        char str2[] = "abcdefghijklmaaaaaaaa" ;
        char str1[10] = "copy!";
        char word2[] = "aaa";

        printf("%s %s %p %p\n", str2, str1, str2, str1);

        strncpy(str1, str2, sizeof str1);
        str1[9] = '\0';
        printf("%s %s\n", str2, str1);

        return 0;
}
```

char * strncpy(char *, const char *, size_t);

```c
#include <stdio.h>
#include <string.h>
#define N 64

int main(void) {

        char str[N] ;
        char word[] = "copy!";
        char word2[] = "aaa";

        strcpy(str, word);
        printf("%s\n", str);

        strncpy(str+1, word, 2);

        printf("%s\n", str);

        strncpy(str, word2, 4);
        printf("%s\n", str);

        return 0;
}
```