Wykład 08 w dniu 19.XI.08r

Znaki pisarskie, ciągi znaków w C:

Jednobajtowy typ liczb całkowitych `char` służy w C do przedstawiania *znaków pisarskich* a tablice utworzone z tego typu mogą być interpretowane jako *napisy* (stringi).

Plik: `w08-1.c`

```c
#include <stdio.h>

int main()
{
   char c,s[15];
   int si;

   si   = sizeof(char);
   c    = 'M';
   s[0] = c;
   s[1] = 0;
// s[1] = '\0';
   printf("si: %i\n c: %c\n s: %s\n\n",si,c,s);
   si   = sizeof(s);
   printf("s1: %i\n",si);
   si   = sizeof(char *);
   printf("s2: %i\n",si);
   return 0;
}
```

działanie:

```
02:10:46 chaos:~/OUT$ gcc -Wall w08-1.c -o a
02:10:51 chaos:~/OUT$ ./a
si: 1
 c: M
 s: M

s1: 15
s2: 4
02:10:53 chaos:~/OUT$
```

Kodowanie ASCII (American Standard Code for Information Interchange) opis np. na `http://pl.wikipedia.org/wiki/ASCII` oraz *„stałe łańcuchowe”*.
Plik: `w08-2.c`

```c
#include <stdio.h>

const char s[] = "a\tb\nA\tB\n";

int main()
{
   int i,n;

   n = sizeof(s);
   printf("n: %i\n",n);
   printf("%s",s);
```

```c
    for(i=0 ; i<n ; i=i+1)
    {
        printf("%i: 0x%02hhx %2hhi\n",i,s[i],s[i]);
    }
    return 0;
}
```

<u>działanie:</u>

```
03:11:22 chaos:~/OUT$ gcc -Wall w08-2.c -o a
03:11:25 chaos:~/OUT$ ./a
n: 9
a       b
A       B
0: 0x61 97
1: 0x09  9
2: 0x62 98
3: 0x0a 10
4: 0x41 65
5: 0x09  9
6: 0x42 66
7: 0x0a 10
8: 0x00  0
03:11:27 chaos:~/OUT$
```

. . . .

<u>Operacje na łańcuchach znaków (stringach) w C:</u>

Patrz np.: `http://pl.wikibooks.org/wiki/C/Napisy`

<u>Plik: `w08-3.c`</u>

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

int main()
{
    char M[81];
    int i,n;

    strcpy(M,"a\tb\nA\tB\n");
    n = sizeof(M);
    printf("n: %i\n",n);
    printf("%s",M);
    i = -1;
    do {
            i = i+1;
            printf("%i: 0x%02hhx %2hhi\n",i,M[i],M[i]);
        } while(M[i]!='\0');
    n = strlen(M);
    printf("\nn: %i\n\n",n);
    for(i=0 ; i<n ; i=i+1)
    {
        printf("%i: 0x%02hhx %2hhi\n",i,M[i],M[i]);
    }
```

```
    return 0;
}
```

działanie:

```
04:09:14 chaos:~/OUT$ gcc -Wall w08-3.c -o a
04:09:17 chaos:~/OUT$ ./a
n: 81
a       b
A       B
0: 0x61 97
1: 0x09  9
2: 0x62 98
3: 0x0a 10
4: 0x41 65
5: 0x09  9
6: 0x42 66
7: 0x0a 10
8: 0x00  0

n: 8

0: 0x61 97
1: 0x09  9
2: 0x62 98
3: 0x0a 10
4: 0x41 65
5: 0x09  9
6: 0x42 66
7: 0x0a 10
04:09:19 chaos:~/OUT$
```

Informacje o dostępnych funkcjach można uzyskać przez:

```
04:37:40 chaos:~$ man string
Reformatting string(3), please wait...
04:37:52 chaos:~$ man 3 strcpy
Reformatting strcpy(3), please wait...
04:38:12 chaos:~$
```

Deklaracja `strcpy` w skrócie:

```
char *strcpy(char *dest, const char *src);
```

Plik: `w08-4.c`
Dopisywanie łańcucha.

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

int main()
```

```
{
   char M[81];

   strcpy(M,"abc ");
   strcat(M,"xyz\n");
   printf("%s",M);
// to samo w jednym kroku
   strcat(strcpy(M,"abc "),"xyz\n");
   printf("%s",M);
   return 0;
}
```

działanie:

```
04:54:46 chaos:~/OUT$ gcc -Wall w08-4.c -o a
04:55:01 chaos:~/OUT$ ./a
abc xyz
abc xyz
04:55:03 chaos:~/OUT$
```

Plik: `w08-5.c`
Znajdywanie wystąpienia danego znaku w łańcuchu. Funkcja `strchr` o deklaracji:

```
char *strchr(const char *s, int c);
```

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

int main()
{
   char *ws,M[81];

   strcpy(M,"Matematyka\n");
   ws = strchr(M,'e');
   if( ws!=NULL )
   {
      printf("1: %s",ws);
   }
   else
   {
      printf("1: Upss...!?\n");
   }
   ws = strchr(M,'x');
   if( ws!=NULL )
   {
      printf("2: %s",ws);
   }
   else
   {
      printf("2: Upss...!?\n");
   }
   return 0;
}
```

działanie:

```
05:10:03 chaos:~/OUT$ gcc -Wall w08-5.c -o a
```

```
05:10:25 chaos:~/OUT$ ./a
1: ematyka
2: Upss...!?
05:10:26 chaos:~/OUT$
```

<u>Plik: `w08-6.c`</u>
Porównanie dwóch łańcuchów. Funkcja `strcmp`

```
int strcmp(const char *s1, const char *s2);
```

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

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

    if(argc!=3)
    {
        puts("ERROR: Zla ilosc argumentow !?");
        return -1;
    }
    r = strcmp(argv[1],argv[2]);
    if( r<0 )
    {
        printf("%s < %s\n",argv[1],argv[2]);
        return 0;
    }
    if( r==0 )
    {
        printf("%s = %s\n",argv[1],argv[2]);
        return 0;
    }
    printf("%s > %s\n",argv[1],argv[2]);
    return 0;
}
```

<u>działanie:</u>

```
05:28:44 chaos:~/OUT$ gcc -Wall w08-6.c -o a
05:29:00 chaos:~/OUT$ ./a abc acd
abc < acd
05:29:21 chaos:~/OUT$ ./a acd abc
acd > abc
05:29:33 chaos:~/OUT$ ./a abc abc
abc = abc
05:29:43 chaos:~/OUT$
```

<u>UWAGA – przepełnienie bufora:</u>

<u>Plik: `w08-7.c`</u>
Funkcje:

```
char *strncpy(char *dest, const char *src, size_t n);
char *strncat(char *dest, const char *src, size_t n);
```

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

int main()
{
   char M[3];

   strcpy(M,"1                                      2");
   puts(M);
   return 0;
}
```

działanie:

```
05:38:40 chaos:~/OUT$ gcc -Wall w08-7.c -o a
05:38:56 chaos:~/OUT$ ./a
1                                      2
Naruszenie ochrony pamięci
05:38:59 chaos:~/OUT$
```

. . . .

Bezpieczne wczytanie pliku tekstowego:

Dane są w pliku `w08-8.txt`:

```
w 1
w 2
wiersz trzeci
w 4
w 5
```

Plik: `w08-8.c`
Funkcje:

```
FILE *fopen(const char *path, const char *mode);
int fclose(FILE *stream);

char *fgets(char *s, int size, FILE *stream);
```

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

//const char EOF=0x1a;
const int N=25;
const char naF[]="w08-8.txt";

int main()
{
   char txt[N][7],M[7],*ws;
   FILE *fF;
   int i,n;

```

```
    fF = fopen(naF,"r");
    if( fF==NULL )
    {
       puts("ERROR: problem z fopen() !?");
       return -1;
    }
    n = 0;
    do {
          ws = fgets(M,7,fF);
          if( ws==NULL || strchr(M,EOF)!=NULL)
          {
             break;
          }
          strcpy(txt[n],M);
          n = n+1;
       } while(n<N);
    fclose(fF);
// wyswietlamy to co zostalo wczytane
    printf("n: %i\n",n);
    for(i=0 ; i<n ; i=i+1)
    {
       if( strchr(txt[i],'\n')==NULL)
       {
          printf("%2i| %s\n",i,txt[i]);
       }
       else
       {
          printf("%2i: %s",i,txt[i]);
       }
    }
    return 0;
}
```

<u>działanie:</u>

```
06:36:00 chaos:~/OUT$ ls
w08-8.c  w08-8.txt
06:36:02 chaos:~/OUT$ gcc -Wall w08-8.c -o a
06:36:07 chaos:~/OUT$ ./a
n: 7
 0: w 1
 1: w 2
 2| wiersz
 3|  trzec
 4: i
 5: w 4
 6: w 5
06:36:12 chaos:~/OUT$
```

. . . .

Wrocław, dnia 13/11/2008