Wykład 05 w dniu 29.X.08r

Liczby w komputerze:

News

- Ariane 5 flight 501 „On 4 June 1996, the maiden flight of the Ariane 5 launcher ended in a failure. Only about 40 seconds after initiation of the flight sequence, at an altitude of about 3700 m, the launcher veered off its flight path, broke up and exploded."

- Patriot Missile On February 25, 1991, during the Gulf War, an American Patriot Missile battery in Dharan, Saudi Arabia, failed to track and intercept an incoming Iraqi Scud missile. The Scud struck an American Army barracks, killing 28 soldiers and injuring around 100 other people.

- Pentium II Math Bug? I received email from "Dan"who asked if I could reproduce what he thought was a bug in the Pentium Pro processor. I wrote an assembly language program that checked into the problem. I also ran the test on a Pentium-II processor that I had recently bought at Fry's Electronics, an Intel Pentium Processor (P54C), Intel Pentium Processor with MMX Technology (P55C), and an AMD K6. Sure enough, I came to the same conclusion as Dan: it looks like a bug to me.

Podstawowe typu numeryczne

- rzeczywiste

  — w `C`

```
long double
double
float
```

  — standard w domowych komputerach: **IEEE 754**

  — czy

$$(a+b)+c = a+(b+c) \quad !?$$

  — deklaracja

```
double A[n][m]

// uzycie
b = A[i][j];
```

  odpowiada matematycznemu zapisowi

$$b = A_{ij}, \quad i = 0, \ldots, n-1,\ j = 0, \ldots, m-1$$

- całkowite

  — rodzaje

```
    unsigned long long int
    signed long long int
    unsigned long int
    signed long int
    unsigned int
```

```
6     signed int
7     unsigned short int
8     signed short int
9     unsigned char
10    signed char
```

— tablice – tak samo jak dla liczb rzeczywistych
— operacje arytmetyczne oraz przejście między liczbami **całkowitymi** i **rzeczywistymi**

```
1  int n,m,k;
2
3  k = n / m; // czesc calkowita z dzielenia n przez m
4  k = n % m; // reszta z dzielenia
```

Specjalne znaczenie typu `char`

- zakres: $1, \ldots, 127$
- ASCII (American Standard Code for Information Interchange)
- początek: Samuel Morse 1837 – elektryczny telegraf
  Recording of Morse's 1844 Baltimore to Washington demonstration.
- 1 byte (1B, oktet) = 8 bitów (8b)
- SI: 1kB – 10^3, 1MB – 10^6, 1GB – 10^9

Tablica `char` – **string**i

- przedstawienie tekstu

```
1  char s[] = "Wstep do informatyki";
2  char a;
3
4  a = 'X';
```

- znaki specjalne

```
1  '\n', '\t', '\b', '\r', '\f',
2
3  '\\', '\'', '\"', '\0'
```

- specjalne znaczenie biblioteki

```
1  #include <string.h>
2
3  char M[81];
4
5  strcpy(M,"matematyka");
```

Systemy zapisu liczb całkowitych

- cyfra, podstawa, pozycja

  — dziesiętny: $\{0,1,2,3,4,5,6,7,8,9\}$ – cyfry, 10 – podstawa

$$5 \times 10^3 + 0 \times 10^2 + 0 \times 10^1 + 4 \times 10^0 = 5004_{10}$$

— szesnastkowy: $\{0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f\}$ – cyfry, 16 – podstawa

$$\begin{aligned} & 0xa9b = \\ = \quad & a9b_{16} = a \times 16^2 + 9 \times 16^1 + b \times 16^0 = 10 * 256 + 9 * 16 + 11 = 2715_{10} = 2715 \end{aligned}$$

— ósemkowy: $\{0,1,2,3,4,5,6,7\}$ – cyfry, 8 – podstawa
— dwójkowy (binarny): $\{0,1\}$ – cyfry, 2 – podstawa

- przejście między systemami

Zadanie: *Liczbę* 2715 *przedstawić w systemie ósemkowym.*
Algorytm:

2715	3	bo	$2715 = 339 * 8 + 3$
339	3	bo	$339 = 42 * 8 + 3$
42	2	bo	$42 = 5 * 8 + 2$
5			

Wynik: $2715 = 5233_8 = 05233$
Sprawdzamy:

$$\begin{aligned} & 05233 = \\ = \quad & 5 \times 8^3 + 2 \times 8^2 + 3 \times 8^1 + 3 \times 8^0 = 5 * 512 + 2 * 64 + 3 * 8 + 3 = \\ = \quad & 2560 + 128 + 24 + 3 = 2715 \end{aligned}$$

- działania

Zadanie: *Należy obliczyć iloczyn* 101011_2 *przez* 1101_2 *i wynik zapisać w systemie szesnastkowym.*

$$\begin{array}{rr} & 10\ 1011 \\ \times & 1101 \\ \hline & 10\ 1011 \\ & 1010\ 11 \\ & 1\ 0101\ 1 \\ \hline & 10\ 0010\ 1111 \end{array}$$

Sprawdzamy:

$$10\ 1011_2 \times 1101_2 = 10\ 0010\ 1111_2$$

$$2b_{16} \times d_{16} = 22f_{16}$$

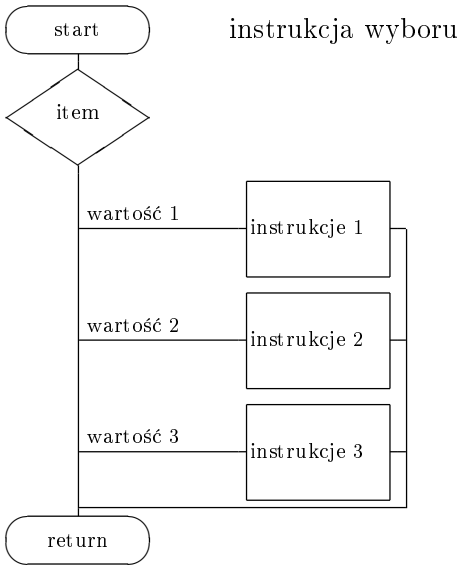
$$43 \times 13 = 559$$

$$53_8 \times 15_8 = 1057_8$$

. . . . . .

Instrukcja wyboru w C

- schemat blokowy



- pseudokod

```
switch item do
    case wartość 1
     | instrukcje 1; break;
    end
    case wartość 2
     | instrukcje 2; break;
    end
    case wartość 3
     | instrukcje 3; break;
    end
end
```

- realizacja

```
1  int main(int argc,char *argv[])
2  {
3      int c;
4      c = getopt(argc,argv,"hi:");
5      if( c==-1 )
6          return -1;
7      switch(c)
8      {
9          case 'h':
10             wyswietl_opcje();
11         break;
12         case 'i':
13             typ_int(optarg);
14         break;
15         case '?':
16             printf("ERROR: \"%c\" opcja !?\n",optopt);
17         break;
```

```
18          default:
19              puts("ERROR: upss ... !?");
20           break;
21      }
22  }
```

. . . . . .

Funkcje własne w C

- deklaracja – przekazanie argumentów

```
1  const int N=20;
2
3  void pomnoz(int *nC, int *mC, double C[][N],\
4              int  nA, int  mA, double A[][N],\
5              int  nB, int  mB, double B[][N])
6  {
7      int i,j,k;
8      for(i=0 ; i<nA ; i=i+1)
9      {
10         for(j=0 ; j<mB; j=j+1)
11         {
12             C[i][j] = 0.0;
13             for(k=0; k<mA ; k=k+1)
14                 C[i][j] = C[i][j]+A[i][k]*B[k][j];
15         }
16     }
17     *nC = nA;
18     *mC = mB;
19 }
```

- użycie

```
1  int main(int argc,char *argv[])
2  {
3      double A[N][N],B[N][N],C[N][N];
4      int i,nA,mA,nB,mB,nC,mC;
5
6  // jakies instrukcje poprzedzajace
7
8      pomnoz(&nC, &mC, C, nA, mA, A, nB, mB, B);
9
10 // dalsze instrukcje
11
12     return 0;
13 }
```

. . . . . .

Program sprawdzający zakres liczb

Plik: `tpz.c`

```
1  /* WdI w IM UWr.
2     wyklad 06 w dniu 22.XI.05r
3     liczbowe typy proste i zlozone,
4     wstep do wskaznikow, funkcje
```

```
5
6       KU PRZESTRODZE !!!
7   */
8   #include <unistd.h>
9   #include <stdlib.h>
10  #include <stdio.h>
11  /*----------------------------------------*/
12
13  void wyswietl_opcje()
14  {
15      puts("h\tinformacja\ni <n>\tcalkowite\nr <n>\trzeczywiste\n");
16      exit(0);
17  }
18  /*----------------------------------------*/
19
20  void typ_int(char *arg)  //  typy liczb calkowitych
21  {
22      unsigned long long int uLL;
23      signed long long int sLL;
24      unsigned long int uL;
25      signed long int sL;
26      unsigned int uN;
27      signed int sN;
28      unsigned short int uS;
29      signed short int sS;
30      unsigned char uC;
31      signed char sC;
32      int i,n;
33
34      printf("int: %s\n",arg);
35      n = atoi(arg);
36      uLL = 1;
37      sLL = 1;
38      uL  = 1;
39      sL  = 1;
40      uN  = 1;
41      sN  = 1;
42      uS  = 1;
43      sS  = 1;
44      uC  = 1;
45      sC  = 1;
46      for(i=2 ; i<=n ; i=i+1)
47      {
48          uLL = uLL * (unsigned long long int)i;
49          sLL = sLL * (signed long long int)i;
50          uL  = uL  * (unsigned long int)i;
51          sL  = sL  * (signed long int)i;
52          uN  = uN  * (unsigned int)i;
53          sN  = sN  * (signed int)i;
54          uS  = uS  * (unsigned short int)i;
55          sS  = sS  * (signed short int)i;
56          uC  = uC  * (unsigned char)i;
57          sC  = sC  * (signed char)i;
58      }
59      printf("%i!\n",n);
60      printf("unsigned long long int(%i):\t%llu\n", sizeof(uLL), uLL);
61      printf("signed long long int(%i):\t%lli\n",   sizeof(sLL), sLL);
62      printf("unsigned long int(%i):\t\t%lu\n",     sizeof(uL) , uL);
```

```
63      printf("signed long int(%i):\t\t%li\n",        sizeof(sL) , sL);
64      printf("unsigned int(%i):\t\t%u\n",            sizeof(uN) , uN);
65      printf("signed int(%i):\t\t\t%i\n",            sizeof(sN) , sN);
66      printf("unsigned short int(%i):\t\t%hu\n",     sizeof(uS) , uS);
67      printf("signed short int(%i):\t\t%hi\n",       sizeof(sS) , sS);
68      printf("unsigned char(%i):\t\t%hhu\n",         sizeof(uC) , uC);
69      printf("signed char(%i):\t\t\t%hhi\n",         sizeof(sC) , sC);
70      exit(0);
71  }
72  /*-----------------------------------------*/
73
74  void typ_real(char *arg)  //  typy liczb rzeczywistych
75  {
76      long double rLD,a,b,r1,r2;
77      double rD;
78      float rF;
79      long int i,n;
80
81      printf("real: %s\n",arg);
82      n = atol(arg);
83      rLD = 1.0L;
84      rD  = 1.0l;
85      rF  = 1.0;
86      for(i=0 ; i<n ; i=i+1)
87      {
88          rLD = rLD* (long double)1000.0;
89          rD  = rD*  (double)1000.0;
90          rF  = rF*  (float)1000.0;
91      }
92      printf("10^%li\n",3*n);
93      printf("long double(%i):\t%12.2Le\n",sizeof(rLD),rLD);
94      printf("double(%i):\t\t%12.2le\n",sizeof(rD),rD);
95      printf("float(%i):\t\t%12.2e\n\n",sizeof(rF),rF);
96      a =  rLD;               //  tu sprawdzamy lacznosc dodawania
97      b = -rLD;
98      r1 = 1.0L + (a + b);
99      printf("1.0+(a-a):\t\t%16.8Lf\n",r1);
100     r2 = (1.0L + a) + b;
101     printf("(1.0+a)-a:\t\t%16.8Lf\n",r2);
102     if( r1!=r2 )
103     {
104         puts("Upss... !?");
105     }
106     exit(0);
107 }
108 /*-----------------------------------------*/
109
110 int main(int argc,char *argv[])
111 {
112     int c;
113
114 /* brak dodatkowych komunikatow
115     opterr = 0;
116 */
117     c = getopt(argc,argv,"hi:r:");
118     if( c==-1 )
119     {
120         puts("ERROR: brak zaznaczonych opcji !?");
```

```
121         return -1;
122     }
123     switch(c)
124     {
125         case 'h':
126             wyswietl_opcje();
127          break;
128         case 'i':
129             typ_int(optarg);
130          break;
131         case 'r':
132             typ_real(optarg);
133          break;
134         case '?':
135             printf("ERROR: \"%c\" opcja !?\n",optopt);
136          break;
137         default:
138             puts("ERROR: upss... !?");
139          break;
140     }
141     return 0;
142 }
143 /*---------------------------------------*/
```

.....

Program pokazyjący operacje na stringach – tekście :

Plik: `str.c`

```
1  /* WdI w IM UWr.
2     wyklad 06 w dniu 22.XI.05r
3     specjalne znaczenie liczb
4     calkowitych unsigned char
5
6     chary i stringi
7  */
8  #include <unistd.h>
9  #include <stdlib.h>
10 #include <string.h>
11 #include <stdio.h>
12 /*---------------------------------------*/
13 const char WdI[] = "Wstep do informatyki";
14 /*---------------------------------------*/
15
16 void wyswietl_opcje()
17 {
18     puts("h\tinformacja\na\tkody ASCII\n\
19 L <s>\tliczba znakow\nz\tznaki specjalne\n\
20 t\tstala tablica znakow\nw <s>\twystapienia znaku \'a\'\n\
21 k <s>\tkopiowanie");
22 }
23 /*---------------------------------------*/
24
25 void znaki_ASCII()
26 {
27     unsigned int i;
28     char c;
29
```

```
30      puts("kody ASCII:");
31      for(i=32 ; i<127 ; i=i+1)
32      {
33          c = (char)i;
34          printf("%3u %c 0x%02x 0%03o\n",i,c,i,i);
35          if( (i+1)%16==0 )
36          {
37              puts("klawisz: Enter || ^C");
38              getchar();
39          }
40      }
41      puts("KONIEC");
42  }
43  /*----------------------------------------*/
44
45  void dlugosc_str(char arg[])
46  {
47      int i,n;
48
49      printf("dlugosc: \"%s\"",arg);
50      for(i=0 ; arg[i]!='\0' ; i=i+1)
51      {
52          ;
53      }
54      n = strlen(arg);
55      printf("\ni: %i\nn: %i\n",i,n);
56  }
57  /*----------------------------------------*/
58
59  void znaki_specjalne()
60  {
61      char c;
62
63      c = '\n';
64      printf("0x%02hhx - nowy wiersz\n",c);
65      c = '\t';
66      printf("0x%02hhx - tabulator poziomy\n",c);
67      c = '\b';
68      printf("0x%02hhx - cofacz\n",c);
69      c = '\r';
70      printf("0x%02hhx - powrot karetki\n",c);
71      c = '\f';
72      printf("0x%02hhx - nowa strona\n",c);
73      c = '\\';
74      printf("0x%02hhx - ukosnik\n",c);
75      c = '\'';
76      printf("0x%02hhx - apostrof\n",c);
77      c = '\"';
78      printf("0x%02hhx - znak cala\n",c);
79      c = '\0';
80      printf("0x%02hhx - ogranicznik ciagu znakow\n",c);
81  }
82  /*----------------------------------------*/
83
84  void tablica_znakow()
85  {
86      int i,n;
87
```

```
88      i = -1;
89      do {
90              i = i+1;
91              printf("[%3i] 0x%02hhx %c\n",i,WdI[i],WdI[i]);
92          } while( WdI[i]!= '\0');
93      n = strlen(WdI);
94      printf("\ni: %i\nn: %i\n",i,n);
95  }
96  /*-----------------------------------------*/
97
98  void wystapienia_znaku_a(char arg[])
99  {
100     int i,n;
101
102     printf("wystapienia '\'a\'' w stringu: \"%s\"\n",arg);
103     n = 0;
104     for(i=0 ; arg[i]!='\0' ; i=i+1)
105     {
106         if( arg[i]=='a' )
107         {
108             if( n==0 )
109             {
110                 printf("Pozycja: %i",i);
111             }
112             else
113             {
114                 printf(", %i",i);
115             }
116             n = n+1;
117         }
118     }
119     if( n!=0 )
120     {
121         printf("\nIlosc wystapien: %i\n",n);
122     }
123     else
124     {
125         puts("Brak !?");
126     }
127 }
128 /*-----------------------------------------*/
129
130 void kopiowanie(char arg[])
131 {
132     char M1[128],M2[128];
133     int i;
134
135     printf("string: \"%s\"\n",arg);
136     i = -1;
137     do {
138             i = i+1;
139             M1[i] = arg[i];
140         } while(arg[i]!='\0');
141     strcpy(M2,arg);
142     printf("Wynik:\nM1\t\"%s\"\nM2\t\"%s\"\n",M1,M2);
143 }
144 /*-----------------------------------------*/
145
```

```
146 int main(int argc,char *argv[])
147 {
148    int c;
149 
150    c = getopt(argc,argv,"haL:ztw:k:");
151    if( c==-1 )
152    {
153       puts("ERROR: brak zaznaczonych opcji !?");
154       return -1;
155    }
156    switch(c)
157    {
158       case 'h':
159          wyswietl_opcje();
160        break;
161       case 'a':
162          znaki_ASCII();
163        break;
164       case 'L':
165          dlugosc_str(optarg);
166        break;
167       case 'z':
168          znaki_specjalne();
169        break;
170       case 't':
171          tablica_znakow();
172        break;
173       case 'w':
174          wystapienia_znaku_a(optarg);
175        break;
176       case 'k':
177          kopiowanie(optarg);
178        break;
179       case '?':
180          printf("ERROR: \"%c\" opcja !?\n",optopt);
181        break;
182       default:
183          puts("ERROR: upss... !?");
184        break;
185    }
186    return 0;
187 }
188 /*-------------------------------------------*/
```

.....

Mnożenie macierzy – programowanie proceduralne

Plik: `mm.c`

```
1 /* WdI w IM UWr.
2    wyklad 06 w dniu 22.XI.05r
3 
4    mnozenie macierzy
5 */
6 #include <stdlib.h>
7 #include <stdio.h>
8 
9 /*-------------------------------------------*/
```

```
10 const int N=20;
11 /*----------------------------------------*/
12
13 void wczytaj(char *naF, int *n, int *m, double Mac[][N])
14 {
15    FILE *fIN;
16    int i,j;
17
18    fIN = fopen(naF,"r");
19    if( fIN==NULL)
20    {
21       printf("ERROR: \"%s\" upss.. !?\n",naF);
22       exit(-1);
23    }
24    printf("Wprowadzam: %s\n",naF);
25    fscanf(fIN,"%i",n);
26    fscanf(fIN,"%i",m);
27    for(i=0; i<*n ; i=i+1)
28    {
29       for(j=0 ; j<*m ; j=j+1)
30       {
31          fscanf(fIN,"%lf",&Mac[i][j]);
32       }
33    }
34    fclose(fIN);
35 }
36 /*----------------------------------------*/
37
38 void wyswietl(int n, int m, double Mac[][N])
39 {
40    int i,j;
41
42    for(i=0 ; i<n ; i=i+1)
43    {
44       for(j=0 ; j<m ; j=j+1)
45       {
46          printf("%6.2lf ",Mac[i][j]);
47       }
48       putchar('\n');
49    }
50 }
51 /*----------------------------------------*/
52
53 void pomnoz(int *nC, int *mC, double C[][N],\
54             int  nA, int  mA, double A[][N],\
55             int  nB, int  mB, double B[][N])
56 {
57    int i,j,k;
58
59    if( mA!=nB )
60    {
61       puts("ERROR: zla ilosc kolumn i wierszy !?");
62       exit(-2);
63    }
64    for(i=0 ; i<nA ; i=i+1)
65    {
66       for(j=0 ; j<mB; j=j+1)
67       {
```

```
68              C[i][j] = 0.0;
69              for(k=0; k<mA ; k=k+1)
70              {
71                  C[i][j] = C[i][j]+A[i][k]*B[k][j];
72              }
73          }
74      }
75      *nC = nA;
76      *mC = mB;
77  }
78  /*----------------------------------------*/
79
80  int main(int argc,char *argv[])
81  {
82      double A[N][N],B[N][N],C[N][N];
83      int i,nA,mA,nB,mB,nC,mC;
84
85      if( argc!=3 )
86      {
87          puts("ERROR: zla ilosc parametrow !?");
88          for(i=0 ; i<argc ; i=i+1)
89          {
90              puts(argv[i]);
91          }
92          return -3;
93      }
94      wczytaj(argv[1], &nA, &mA, A);
95      wczytaj(argv[2], &nB, &mB, B);
96      pomnoz(&nC, &mC, C, nA, mA, A, nB, mB, B);
97      puts("Wynik:");
98      wyswietl(nA, mA, A);
99      puts("X");
100     wyswietl(nB, mB, B);
101     puts("=");
102     wyswietl(nC, mC, C);
103     return 0;
104 }
105 /*----------------------------------------*/
```

Dane:

Plik: `a.dat`

```
2 3
1.0 -1.0 0.0
2.0  0.0 1.0
```

Plik: `b.dat`

```
3 2
3.0  0.0
0.0 -1.0
1.0  1.0
```

Wynik:

```
19:53:48 chaos:~/OUT$ gcc -Wall mm.c -o mm
19:53:58 chaos:~/OUT$ ./mm a.dat b.dat
Wprowadzam: a.dat
Wprowadzam: b.dat
Wynik:
  1.00  -1.00   0.00
  2.00   0.00   1.00
X
  3.00   0.00
  0.00  -1.00
  1.00   1.00
=
  3.00   1.00
  7.00   1.00
```

. . . . . .

Literatura

[1] Ragen, A, *Leksykon języka C,* WNT, Warszawa 1990

Wrocław, dnia 29/10/2008