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
```

```
    signed int
    unsigned short int
    signed short int
    unsigned char
    signed char
```

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

```
int n,m,k;

k = n / m; // czesc calkowita z dzielenia n przez m
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

```
char s[] = "Wstep do informatyki";
char a;

a = 'X';
```

- znaki specjalne

```
'\n', '\t', '\b', '\r', '\f',

'\\', '\'', '\"', '\0'
```

- specjalne znaczenie biblioteki

```
#include <string.h>

char M[81];

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:

$$\begin{array}{r|l} 2715 & 3 \\ 339 & 3 \\ 42 & 2 \\ 5 & \end{array} \qquad \begin{array}{lr} \text{bo} & 2715 = 339 * 8 + 3 \\ \text{bo} & 339 = 42 * 8 + 3 \\ \text{bo} & 42 = 5 * 8 + 2 \end{array}$$

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.*

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

Sprawdzamy:

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

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

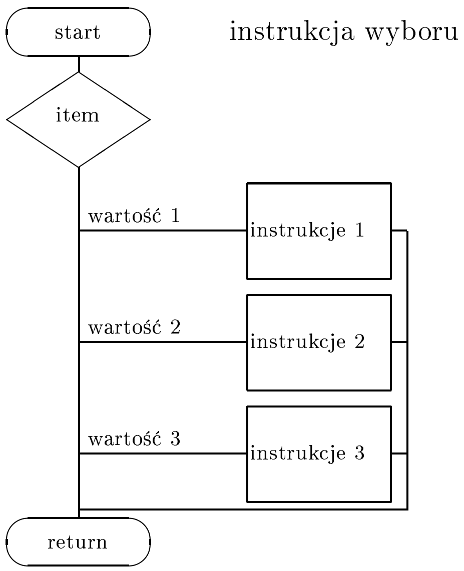
$$43 \times 13 = 559$$

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

. . . .

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

```
int main(int argc,char *argv[])
{
    int c;
    c = getopt(argc,argv,"hi:");
    if( c==-1 )
        return -1;
    switch(c)
    {
        case 'h':
            wyswietl_opcje();
         break;
        case 'i':
            typ_int(optarg);
         break;
        case '?':
            printf("ERROR: \"%c\" opcja !?\n",optopt);
         break;
```

```
        default:
            puts("ERROR: upss ... !?");
        break;
    }
}
```

Funkcje własne w C

- deklaracja – przekazanie argumentów

```
const int N=20;

void pomnoz(int *nC, int *mC, double C[][N],\
            int  nA, int  mA, double A[][N],\
            int  nB, int  mB, double B[][N])
{
    int i,j,k;
    for(i=0 ; i<nA ; i=i+1)
    {
        for(j=0 ; j<mB; j=j+1)
        {
            C[i][j] = 0.0;
            for(k=0; k<mA ; k=k+1)
                C[i][j] = C[i][j]+A[i][k]*B[k][j];
        }
    }
    *nC = nA;
    *mC = mB;
}
```

- użycie

```
int main(int argc,char *argv[])
{
    double A[N][N],B[N][N],C[N][N];
    int i,nA,mA,nB,mB,nC,mC;

// jakies instrukcje poprzedzajace

    pomnoz(&nC, &mC, C, nA, mA, A, nB, mB, B);

// dalsze instrukcje

    return 0;
}
```

Program sprawdzający zakres liczb

Plik: `tpz.c`

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

```

   KU PRZESTRODZE !!!
*/
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>
/*----------------------------------------*/

void wyswietl_opcje()
{
   puts("h\tinformacja\ni <n>\tcalkowite\nr <n>\trzeczywiste\n");
   exit(0);
}
/*----------------------------------------*/

void typ_int(char *arg)  //  typy liczb calkowitych
{
   unsigned long long int uLL;
   signed long long int sLL;
   unsigned long int uL;
   signed long int sL;
   unsigned int uN;
   signed int sN;
   unsigned short int uS;
   signed short int sS;
   unsigned char uC;
   signed char sC;
   int i,n;

   printf("int: %s\n",arg);
   n = atoi(arg);
   uLL = 1;
   sLL = 1;
   uL  = 1;
   sL  = 1;
   uN  = 1;
   sN  = 1;
   uS  = 1;
   sS  = 1;
   uC  = 1;
   sC  = 1;
   for(i=2 ; i<=n ; i=i+1)
   {
      uLL = uLL * (unsigned long long int)i;
      sLL = sLL * (signed long long int)i;
      uL  = uL  * (unsigned long int)i;
      sL  = sL  * (signed long int)i;
      uN  = uN  * (unsigned int)i;
      sN  = sN  * (signed int)i;
      uS  = uS  * (unsigned short int)i;
      sS  = sS  * (signed short int)i;
      uC  = uC  * (unsigned char)i;
      sC  = sC  * (signed char)i;
   }
   printf("%i!\n",n);
   printf("unsigned long long int(%i):\t%llu\n", sizeof(uLL), uLL);
   printf("signed long long int(%i):\t%lli\n",   sizeof(sLL), sLL);
   printf("unsigned long int(%i):\t\t%lu\n",     sizeof(uL) , uL);
```

```
    printf("signed long int(%i):\t\t%li\n",        sizeof(sL) , sL);
    printf("unsigned int(%i):\t\t%u\n",            sizeof(uN) , uN);
    printf("signed int(%i):\t\t\t%i\n",            sizeof(sN) , sN);
    printf("unsigned short int(%i):\t\t%hu\n",     sizeof(uS) , uS);
    printf("signed short int(%i):\t\t%hi\n",       sizeof(sS) , sS);
    printf("unsigned char(%i):\t\t%hhu\n",         sizeof(uC) , uC);
    printf("signed char(%i):\t\t\t%hhi\n",         sizeof(sC) , sC);
    exit(0);
}
/*------------------------------------------------*/

void typ_real(char *arg)  //  typy liczb rzeczywistych
{
    long double rLD,a,b,r1,r2;
    double rD;
    float rF;
    long int i,n;

    printf("real: %s\n",arg);
    n = atol(arg);
    rLD = 1.0L;
    rD  = 1.0l;
    rF  = 1.0;
    for(i=0 ; i<n ; i=i+1)
    {
        rLD = rLD* (long double)1000.0;
        rD  = rD*  (double)1000.0;
        rF  = rF*  (float)1000.0;
    }
    printf("10^%li\n",3*n);
    printf("long double(%i):\t%12.2Le\n",sizeof(rLD),rLD);
    printf("double(%i):\t\t%12.2le\n",sizeof(rD),rD);
    printf("float(%i):\t\t%12.2e\n\n",sizeof(rF),rF);
    a =  rLD;               //  tu sprawdzamy lacznosc dodawania
    b = -rLD;
    r1 = 1.0L + (a + b);
    printf("1.0+(a-a):\t\t%16.8Lf\n",r1);
    r2 = (1.0L + a) + b;
    printf("(1.0+a)-a:\t\t%16.8Lf\n",r2);
    if( r1!=r2 )
    {
        puts("Upss... !?");
    }
    exit(0);
}
/*------------------------------------------------*/

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

/* brak dodatkowych komunikatow
    opterr = 0;
*/
    c = getopt(argc,argv,"hi:r:");
    if( c==-1 )
    {
        puts("ERROR: brak zaznaczonych opcji !?");
```

```c
        return -1;
    }
    switch(c)
    {
        case 'h':
            wyswietl_opcje();
         break;
        case 'i':
            typ_int(optarg);
         break;
        case 'r':
            typ_real(optarg);
         break;
        case '?':
            printf("ERROR: \"%c\" opcja !?\n",optopt);
         break;
        default:
            puts("ERROR: upss... !?");
         break;
    }
    return 0;
}
/*-----------------------------------------*/
```

. . . .

Program pokazyjący operacje na stringach – tekście :

Plik: `str.c`

```c
/* WdI w IM UWr.
    wyklad 06 w dniu 22.XI.05r
    specjalne znaczenie liczb
    calkowitych unsigned char

    chary i stringi
*/
#include <unistd.h>
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
/*-----------------------------------------*/
const char WdI[] = "Wstep do informatyki";
/*-----------------------------------------*/

void wyswietl_opcje()
{
    puts("h\tinformacja\na\tkody ASCII\n\
L <s>\tliczba znakow\nz\tznaki specjalne\n\
t\tstala tablica znakow\nw <s>\twystapienia znaku \'a\'\n\
k <s>\tkopiowanie");
}
/*-----------------------------------------*/

void znaki_ASCII()
{
    unsigned int i;
    char c;

```

```
    puts("kody ASCII:");
    for(i=32 ; i<127 ; i=i+1)
    {
        c = (char)i;
        printf("%3u %c 0x%02x 0%03o\n",i,c,i,i);
        if( (i+1)%16==0 )
        {
            puts("klawisz: Enter || ^C");
            getchar();
        }
    }
    puts("KONIEC");
}
/*--------------------------------------------------------*/

void dlugosc_str(char arg[])
{
    int i,n;

    printf("dlugosc: \"%s\"",arg);
    for(i=0 ; arg[i]!='\0' ; i=i+1)
    {
        ;
    }
    n = strlen(arg);
    printf("\ni: %i\nn: %i\n",i,n);
}
/*--------------------------------------------------------*/

void znaki_specjalne()
{
    char c;

    c = '\n';
    printf("0x%02hhx - nowy wiersz\n",c);
    c = '\t';
    printf("0x%02hhx - tabulator poziomy\n",c);
    c = '\b';
    printf("0x%02hhx - cofacz\n",c);
    c = '\r';
    printf("0x%02hhx - powrot karetki\n",c);
    c = '\f';
    printf("0x%02hhx - nowa strona\n",c);
    c = '\\';
    printf("0x%02hhx - ukosnik\n",c);
    c = '\'';
    printf("0x%02hhx - apostrof\n",c);
    c = '\"';
    printf("0x%02hhx - znak cala\n",c);
    c = '\0';
    printf("0x%02hhx - ogranicznik ciagu znakow\n",c);
}
/*--------------------------------------------------------*/

void tablica_znakow()
{
    int i,n;

```

```
    i = -1;
    do {
            i = i+1;
            printf("[%3i] 0x%02hhx %c\n",i,WdI[i],WdI[i]);
        } while( WdI[i]!= '\0');
    n = strlen(WdI);
    printf("\ni: %i\nn: %i\n",i,n);
}
/*--------------------------------------------------------------*/

void wystapienia_znaku_a(char arg[])
{
    int i,n;

    printf("wystapienia \'a\' w stringu: \"%s\"\n",arg);
    n = 0;
    for(i=0 ; arg[i]!='\0' ; i=i+1)
    {
        if( arg[i]=='a' )
        {
            if( n==0 )
            {
                printf("Pozycja: %i",i);
            }
            else
            {
                printf(", %i",i);
            }
            n = n+1;
        }
    }
    if( n!=0 )
    {
        printf("\nIlosc wystapien: %i\n",n);
    }
    else
    {
        puts("Brak !?");
    }
}
/*--------------------------------------------------------------*/

void kopiowanie(char arg[])
{
    char M1[128],M2[128];
    int i;

    printf("string: \"%s\"\n",arg);
    i = -1;
    do {
            i = i+1;
            M1[i] = arg[i];
        } while(arg[i]!='\0');
    strcpy(M2,arg);
    printf("Wynik:\nM1\t\"%s\"\nM2\t\"%s\"\n",M1,M2);
}
/*--------------------------------------------------------------*/

```

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

   c = getopt(argc,argv,"haL:ztw:k:");
   if( c==-1 )
   {
      puts("ERROR: brak zaznaczonych opcji !?");
      return -1;
   }
   switch(c)
   {
      case 'h':
         wyswietl_opcje();
       break;
      case 'a':
         znaki_ASCII();
       break;
      case 'L':
         dlugosc_str(optarg);
       break;
      case 'z':
         znaki_specjalne();
       break;
      case 't':
         tablica_znakow();
       break;
      case 'w':
         wystapienia_znaku_a(optarg);
       break;
      case 'k':
         kopiowanie(optarg);
       break;
      case '?':
         printf("ERROR: \"%c\" opcja !?\n",optopt);
       break;
      default:
         puts("ERROR: upss... !?");
       break;
   }
   return 0;
}
/*---------------------------------------*/
```

....

Mnożenie macierzy – programowanie proceduralne

Plik: `mm.c`

```c
/* WdI w IM UWr.
   wyklad 06 w dniu 22.XI.05r

   mnozenie macierzy
*/
#include <stdlib.h>
#include <stdio.h>

/*---------------------------------------*/
```

```
const int N=20;
/*----------------------------------------------------*/

void wczytaj(char *naF, int *n, int *m, double Mac[][N])
{
    FILE *fIN;
    int i,j;

    fIN = fopen(naF,"r");
    if( fIN==NULL)
    {
        printf("ERROR: \"%s\" upss.. !?\n",naF);
        exit(-1);
    }
    printf("Wprowadzam: %s\n",naF);
    fscanf(fIN,"%i",n);
    fscanf(fIN,"%i",m);
    for(i=0; i<*n ; i=i+1)
    {
        for(j=0 ; j<*m ; j=j+1)
        {
            fscanf(fIN,"%lf",&Mac[i][j]);
        }
    }
    fclose(fIN);
}
/*----------------------------------------------------*/

void wyswietl(int n, int m, double Mac[][N])
{
    int i,j;

    for(i=0 ; i<n ; i=i+1)
    {
        for(j=0 ; j<m ; j=j+1)
        {
            printf("%6.2lf ",Mac[i][j]);
        }
        putchar('\n');
    }
}
/*----------------------------------------------------*/

void pomnoz(int *nC, int *mC, double C[][N],\
            int  nA, int  mA, double A[][N],\
            int  nB, int  mB, double B[][N])
{
    int i,j,k;

    if( mA!=nB )
    {
        puts("ERROR: zla ilosc kolumn i wierszy !?");
        exit(-2);
    }
    for(i=0 ; i<nA ; i=i+1)
    {
        for(j=0 ; j<mB; j=j+1)
        {
```

```
            C[i][j] = 0.0;
            for(k=0; k<mA ; k=k+1)
            {
                C[i][j] = C[i][j]+A[i][k]*B[k][j];
            }
        }
    }
    *nC = nA;
    *mC = mB;
}
/*-------------------------------------------------------*/

int main(int argc,char *argv[])
{
    double A[N][N],B[N][N],C[N][N];
    int i,nA,mA,nB,mB,nC,mC;

    if( argc!=3 )
    {
        puts("ERROR: zla ilosc parametrow !?");
        for(i=0 ; i<argc ; i=i+1)
        {
            puts(argv[i]);
        }
        return -3;
    }
    wczytaj(argv[1], &nA, &mA, A);
    wczytaj(argv[2], &nB, &mB, B);
    pomnoz(&nC, &mC, C, nA, mA, A, nB, mB, B);
    puts("Wynik:");
    wyswietl(nA, mA, A);
    puts("X");
    wyswietl(nB, mB, B);
    puts("=");
    wyswietl(nC, mC, C);
    return 0;
}
/*-------------------------------------------------------*/
```

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