Fundamentele programării
Curs 1 – Introducere

Lefkovits Sziddónia

sites.umfst.ro / lefkovits-Sziddónia

utilizator: FP
parolă: Curs2022FP

Frecvența   1 GHz

K. kilo, M – mega, G – giga

$1 K UM = 1000 UM$ $1 KHz = 1000 Hz$

$1 M UM = 1000^2 UM$ $1 MHz = 1 \cdot 10^6 Hz$

$1 G UM = 1000^3 UM$ $1 GHz = 10^9 Hz$

$1 Hz = 1$ bătaie de inimă

TACT pe secunda

$1 m = 80$ bătăi de inimă

.000.000.0000

flop – floating point operation
16 – 32 tacte / 1.

1102 biliarde de operații float

aprox 16 – 32 tacte de frecvența $= 1$ op. float
???

1 bit   0 - 1 tranzistor stins/oprit

1b   1 -   aprins / pornit

$7\ 6\ 5\ 4\ 3\ 2\ 1\ 0$ 1B

| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1byte

$7\ 6\ 5\ 4\ 3\ 2\ 1\ 0$

$0\ 0\ 0\ 0.\ 0\ 0\ 0\ 0 \rightarrow 0$

$\vdots$

$1\ 1\ 1\ 1.\ 1\ 1\ 1\ 1 \rightarrow 255$

1 KB

$$\boxed{500\ Mb/1}$$

500 MB/1

$\square < \begin{smallmatrix}0\\1\end{smallmatrix}$ 2 variante

8 pozitii 2^8 nr. de variante

$2^8 = 256$

1 bit

1 Byte = 1B = 8 biti   Byte = octet

1 kB = 1024 = 2^{10} Bytes

~~1000~~

$2^9 = 512$

$2^{10} = 1024$

1 MB = 1024 kB = 2^{10} kB = 2^{20} B

1 GB = 1024 MB = 2^{10} MB = 2^{20} kB = 2^{30} B.

1 TB = 2^{10} GB = 2^{20} MB = 2^{30} kB = 2^{40} B.

$= 2^{40} \cdot 8\ b$

1 PetaByte = 1PB = 2^{50} B

HDD: 2-4 rms (milisec)

RAM: 30-60 ns (nanosec)

L1 cache: level 1   0.5 ns.

2 ms $\longrightarrow$ 20 ns.

milisec $\longrightarrow$ 20 nanos.

1 UM = 1000 mili = m

1 m UM = 1000 μ UM

1 m UM = 10^6 nUM = nano

$\underset{?}{\dfrac{2 \text{ ms}}{}}$ HDD RAM = 20 ns.

$\dfrac{2 \cdot 10^{-3} s}{10^{-4}} = 2 \cdot 10 \cdot 10^6 s$

$2 \cdot 10^{-3} \cdot 10^7 s = 2 \cdot 10^7 s.$

Software: programme uzuale

SO: Sistem de operare,

$\underbrace{}$

$\xrightarrow{}$ are acces direct la componentele hardware

Descriere: Să se obțină sol. ec. de gr I. $ax^2 + bx + c = 0.$

Analiza: Date de intrare: a, b, c citire / dă de la tastatura

pași: calc. Δ_1
$\Delta > 0$
$\Delta = 0$
$\Delta < 0$

calc. x_1, x_2

implementare în C

verificare / testare

Primul program in C

```
#include <iostream>      fișiere antet
#include <conio.h>       header files
                    → _getch()
                    ↳ cod afișare C++

#include <windows.h>
                    ↳ system

using namespace std;
    ↳ intreg
int main ()         ✗        int - integer = intreg
{  ✗   → acolada
        deschisa   → (nu) se pune ;

cout << "Text" ;
_getch(); // getch(); asteapta apasarea unei taste
system("pause"); opreste programul ↗ Press any....

return 0; // 0 program terminat corect
}  ✗
   ↑ acolada inchisa
        Lumi: compilare ↗ build ↗ exe
```