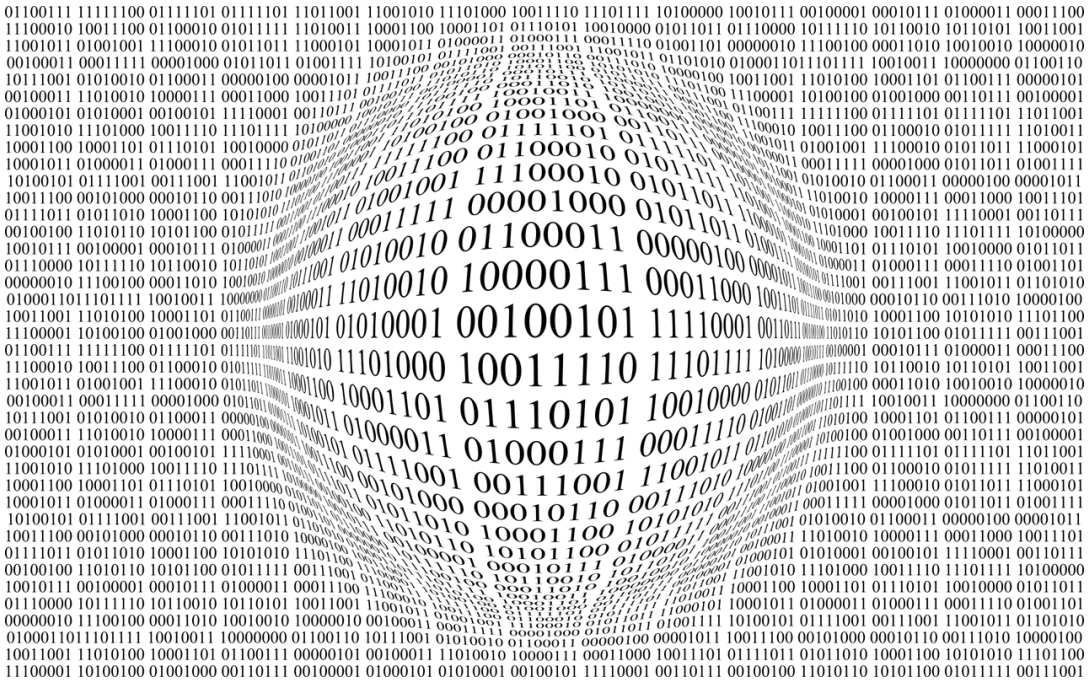
Programski jezici

Programski jezici

- Mašinski
- Visokog nivoa
- Objektno-orijentisani
- Funkcionalni

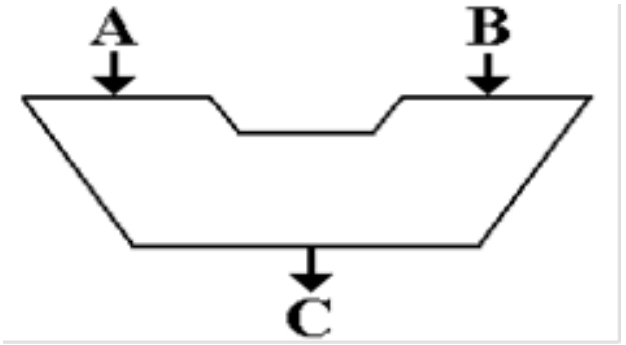
Mašinski jezik: binarni brojevi

0000 – 0
0001 – 1
0010 – 2
0011 – 3
0100 – 4
0101 – 5
0110 – 6
0111 – 7
1000 – 8
1001 – 9
1010 – 10
1011 – 11
...



Aritmetičko logička jedinica

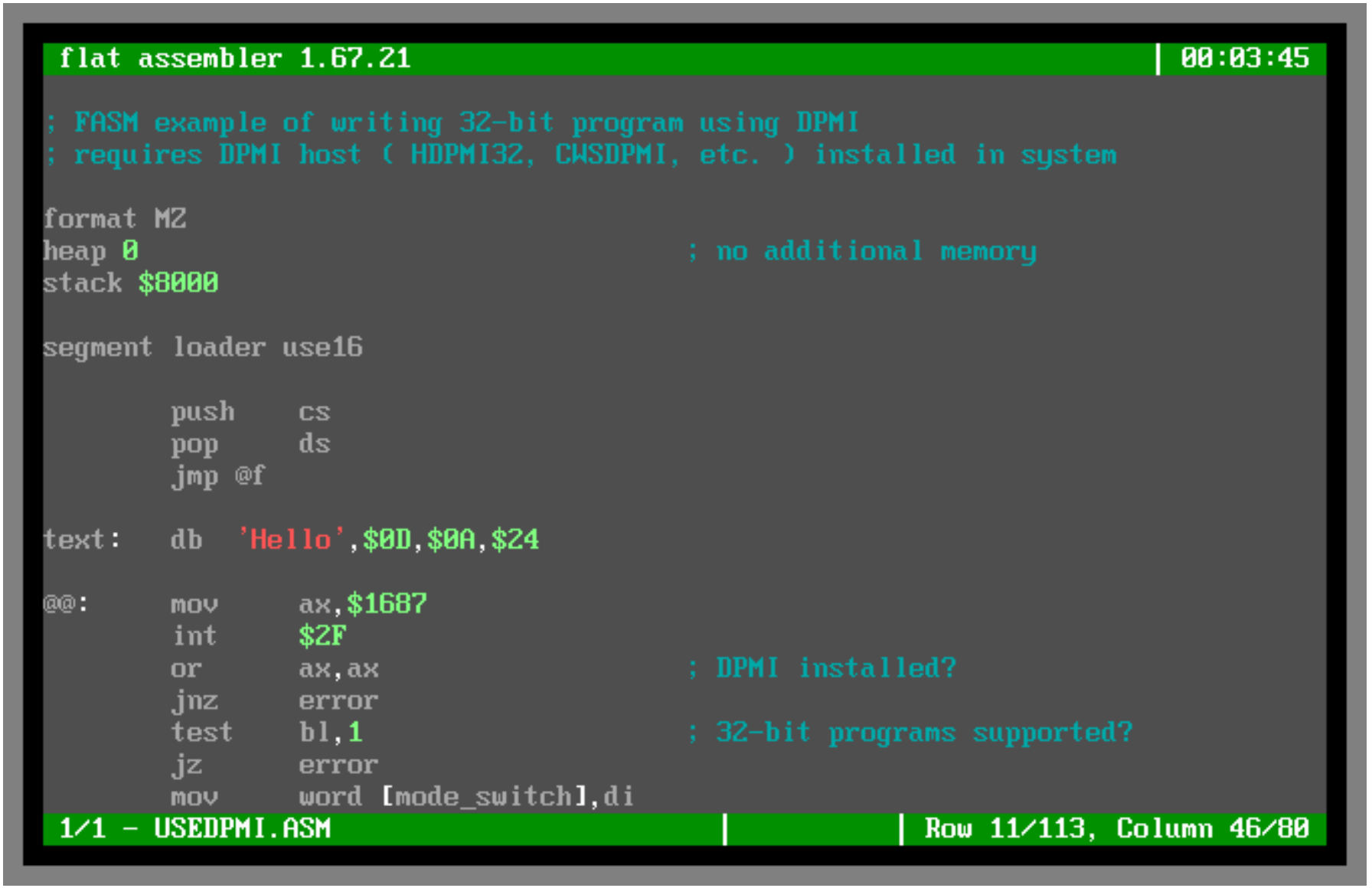
- Operandi: A i B
- Rezultat: C
- Operacije:
  add, sub, mul, div, cmp, etc.



Assembler

```
MOV AX, BX;      AX := BX
ADD DX, CX;      DX := DX + CX
SUB DX, AX;      DX := DX – AX
INC AX;          AX := AX + 1
NOP
```

Assembler


flat assembler 1.67.21 | 00:03:45
; FASM example of writing 32-bit program using DPMI
; requires DPMI host (HDPMI32, CWSDPMI, etc.) installed in system
format MZ
heap 0 ; no additional memory
stack $8000
segment loader use16
push cs
pop ds
jmp @f
text: db 'Hello',$0D,$0A,$24
@@: mov ax,$1687
int $2F
or ax,ax ; DPMI installed?
jnz error
test bl,1 ; 32-bit programs supported?
jz error
mov word [mode_switch],di
1/1 - USEDPMI.ASM | Row 11/113, Column 46/80

Programski jezici visokog nivoa

- Bliski ljudskom načinu razmišljanja
- Mogu se koristiti reči koje ljudi koriste u svakodnevnom životu
- Mogu se praviti procedure i funkcije
- Primeri: C, Pascal, Cobol, Fortran

Programski prevodioci

- Prevode programski jezik višeg nivoa (uključujući objektno-orijentisane) u mašinski jezik
- Moguće je parcijalno prevođenje izvornih fajlova

Tok izvršavanja naredbi procesora

- Procesor iz memorije uzima naredbu sa adrese PC (program counter) i izvršava je
- Nakon naredbe, izvršava sledeću
  - Ukoliko je naredba bila naredba skoka, sledeća naredba se nalazi na adresi koja je data prethodnom narebom;
  - Inače, sledeća naredba je odmah iza trenutne.
- Proverava se da li je došlo do prekida

Promenljive

- Naziv za određenu memorijsku lokaciju
- Mogu biti različitih prostih tipova:
  - int
  - float
  - boolean
  - string
- Mogu biti složenih tipova:
  - Nizovi
  - Strukture

Funkcionalni jezici

- Sve su funkcije
- Primeri: Lisp, Prolog, ML
- Primer Prolog koda:

```
parent(charles1, james1).
parent(elizabeth, james1).
parent(charles2, charles1).
parent(catherine, charles1).
parent(james2, charles1).
parent(sophia, elizabeth).
parent(george1, sophia).

Query: parent(charles1, george1).
```

Naredbe

C++ program:

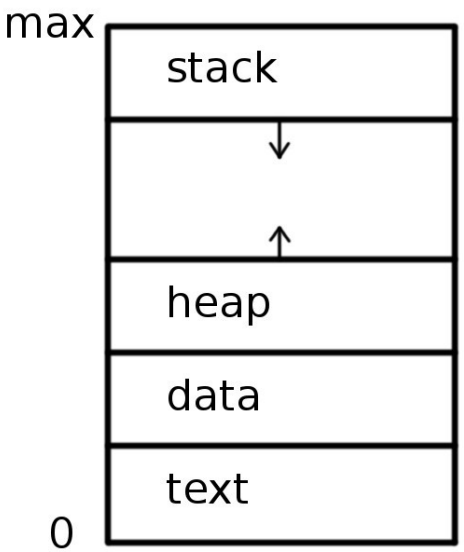
```
int main(){
    int a, b, c;
    cin >> a;
    cin >> b;
    c = (a + b) * 1.2;
    cout << c << endl;
}
```

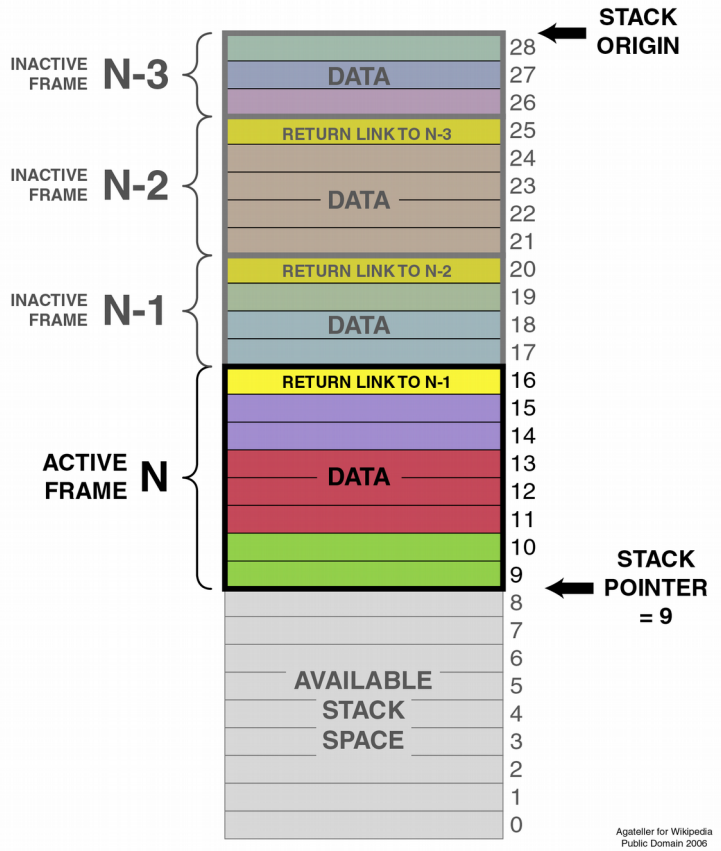
Funkcije

- Kod koji se poziva iz glavnog programa

  ili drugih funkcija
- Može imati argumente sa kojima se poziva
- Može imati povratnu vrednost
- Primer poziva:
  c = f(a, b);

Stek memorija





Objektno orijentisano programiranje

- Jezici još bliži ljudskom načinu razmišljanja
- Omogućavaju enkapsuliranje informacija i funkcija pojedinih objekata
- Moguće je nasleđivanje funkcionalnosti (primer: vozilo → automobil i vozilo → kamion)

Objektno orijentisano programiranje

- Definišu se klase
  - Klase mogu imati funkcije (metode)
  - Klase mogu imati promenljive (polja)
- Promenljive tipa klasa se nazivaju objektima

```
class Student {       // Klasa
    int godina;  // promenljiva
    string ime;  // promenljiva
};
```

Objektno orijentisani jezici

- Neki od najčešće korišćenih programskih jezika su:
  - C++
  - Java
  - Python
- Svaki definiše svoju sintaksu i funkcionalnosti

Funkcionalni jezici

- Sve su funkcije
- Primeri: Lisp, Prolog, ML
- Primer Prolog koda:

```
parent(charles1, james1).
parent(elizabeth, james1).
parent(charles2, charles1).
parent(catherine, charles1).
parent(james2, charles1).
parent(sophia, elizabeth).
parent(george1, sophia).

Query: parent(charles1, george1).
```