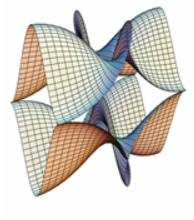


Prirodoslovno-matematički fakultet
Matematički odsjek
Sveučilište u Zagrebu

PROGRAMIRANJE 1

Predavanje 02 - Algoritmi. Osnovna građa računala.

5. listopada 2020.

Sastavio: Zvonimir Bujanović
(skraćena verzija predavanja prof. Singera)

Algoritmi

Algoritam

- Metoda, pravilo ili postupak za rješavanje nekog problema ili dostizanje nekog cilja.
- Sastoji se od konačnog niza koraka.

Jedan od ciljeva (ovog i drugih) kolegija:

1. Naučiti koristiti i implementirati postojeće algoritme za rješavanje raznih poznatih problema.
2. Naučiti kako analizirati algoritam i odrediti njegovu efikasnost.
3. Naučiti kako samostalno osmisliti algoritme za nepoznate probleme.

Algoritmi u stvarnom životu

Primjeri algoritama:

- Kulinarski recepti.
- Upute za rukovanje raznim strojevima ili uređajima.
- Upute za sastavljanje namještaja.

Algoritmi se sastoje od niza koraka.

- Svaki korak je neka instrukcija ili naredba ili akcija koju treba izvršiti.
- Koraci mogu uključivati i:
  - Grananje:
    "Ako vrijedi to-i-to, onda napravi to-i-to, a u protivnom to-i-to".
  - Ponavljanje:
    "Ponovi 5 puta sljedeću instrukciju",
    "Sve dok je ispunjen taj-i-taj uvjet, ponavljaj sljedeću akciju".

Algoritmi - Svojstva

Algoritam izvodi neke operacije nad nekim podacima, u obliku niza koraka, i daje neko rješenje.

Osnovna svojstva algoritma su:

- ima (ili nema) ulazne podatke,
- ima izlazne podatke,
- završava u konačnom vremenu,
- uvijek je nedvosmisleno definiran.

Redovito je cilj osigurati da je algoritam efikasan, tj. da završi u razumnom vremenu.

Algoritmi - Što je jedan korak?

Uočimo da jedan korak algoritma ovisi o “moći” izvršitelja.

Algoritam za kupovinu novog ormara:

1. Naruči ormar s web-stranice trgovine namještaja.
2. U dogovorenom terminu preuzmi paket s ormarom na kućnoj adresi.
3. Sastavi ormar (nazovi susjeda Juru).
4. Rasporedi stvari u ormar.

Algoritmi - Što je jedan korak?

Uočimo da jedan korak algoritma ovisi o “moći” izvršitelja.

Algoritam za kupovinu novog ormara:

1. Naruči ormar s web-stranice trgovine namještaja.
2. U dogovorenom terminu preuzmi paket s ormarom na kućnoj adresi.
3. Sastavi ormar:
    1. Otvori paket.
    2. Izvadi najveću dasku, te 4 vijka broj 37263.
    3. Zašerafi 4 vijka u kuteve najveće daske.
    4. ...
4. Rasporedi stvari u ormar.

Algoritmi - Što je jedan korak?

Dodavanje novog elementa `x` na prvo mjesto u polju `polje` (Python):

```
polje.insert(0, x);
```

Dodavanje novog elementa `x` na prvo mjesto u polju `polje` (C):

```
for(i = n-1; i >= 0; --i)
    polje[i+1] = polje[i];
polje[0] = x;
```

Algoritmi - Što je jedan korak?

Dodavanje novog elementa `x` na prvo mjesto u polju `polje` (ASM):

```
    jmp .L2
.L3:
    movl    -44(%rbp), %eax
    leal    1(%rax), %ecx
    movl    -44(%rbp), %eax
    cltq
    movl    -32(%rbp,%rax,4), %edx
    movslq  %ecx, %rax
    movl    %edx, -32(%rbp,%rax,4)
    subl    $1, -44(%rbp)
.L2:
    cmpl    $0, -44(%rbp)
    jns .L3
    movl    -36(%rbp), %eax
    movl    %eax, -32(%rbp)
    movl    $0, %eax
    movq    -8(%rbp), %rsi
    xorq    %fs:40, %rsi
```

Zašto programski jezik C?

Programski jezik C daje srednji put:

- Dovoljno je visokog nivoa da se algoritmi mogu implementirati kratkim i čitkim programima.
- Dovoljno je niskog nivoa da ne skriva detalje od programera.
- Daje vrlo dobru bazu za razumijevanje složenosti algoritama i učenje o algoritmima.
- Daje vrlo dobru bazu za razumijevanje kako se odvijaju "stvari u pozadini" drugih programskih jezika i aplikacija općenito.
- Po uzoru na C je razvijena sintaksa mnogih drugih suvremenih jezika: C++, C#, Java, JavaScript, PHP, itd.

Točka 2 $\rightsquigarrow$ moramo malo znati o građi i načinu funkcioniranja računala prije programiranja u C-u.

Osnovna građa računala

Računalo:

- Stroj za izvršavanje algoritama.
- Zna izvršavati “osnovne strojne instrukcije” – vidi slajd s programom u ASM i slajd 13.
- Zna raditi s “osnovnim podacima” – nizovima nula i jedinica.

Ulazni i izlazni podaci se nalaze na nekom mediju/uređaju (disk, tipkovnica, ekran, ...).

Algoritam na računalu transformira ulazne podatke.
Međurezultati se spremaju ih u memoriju.

Instrukcije koji čine algoritam:

- Spremljene su na nekom mediju (disk).
- Prilikom pokretanja programa se učitavaju u memoriju.
- Procesor redom izvršava instrukcije iz memorije.

Memorija i logički sklopovi

Memorija računala sastoji se od elemenata koje zovemo bistabili.
U bistabil možemo spremiti jedan bit – binarnu znamenku 0 ili 1.

Logički sklopovi u računalu:

- Omogućavaju izvršavanje logičkih operatora poput NOT, OR, AND nad pojedinim bitovima.

Realizacija aritmetičkih operacija u računalu:

- Bitovi se organiziraju u veće cjeline (npr. 32 bita) i interpretiraju kao brojevi (cijeli, realni).
- Spajanjem više logičkih sklopova u složenije sklopove se mogu implementirati operacije zbrajanja, množenja, itd. (vježbe, §2.5)

Svi ovi sklopovi čine tzv. aritmetičko-logičku jedinicu procesora.

Sve operacije u računalu su realizirane osnovnim logičkim sklopovima koji rade s bitovima iz memorije!

Organizacija memorije

Riječ

- Svi bitovi u memoriji su grupirani u tzv. riječi (eng. word).
- Jedna riječ se tipično sastoji od jednog byte-a, tj. 8 bitova.
- Riječi su u memoriji numerirane linearno.
- Redni broj riječi u memoriji zovemo adresa te riječi.

riječ_0	riječ_1	riječ_2	riječ_3	...	riječ_{n-2}	riječ_{n-1}

Svaki podatak u memoriji:

- Ima adresu – riječ(i) u koje je spremljen.
- Ima vrijednost – niz nula i jedinica koji su spremljeni u te riječi.

Računalo vidi podatke kao “sadržaj spremljen na toj-i-toj adresi”.
Kažemo: “adresa pokazuje na podatak u memoriji”.

Adresa se podacima obično dodjeljuje automatski, tj. ne možemo utjecati na to gdje će se podatak spremiti.

Organizacija memorije

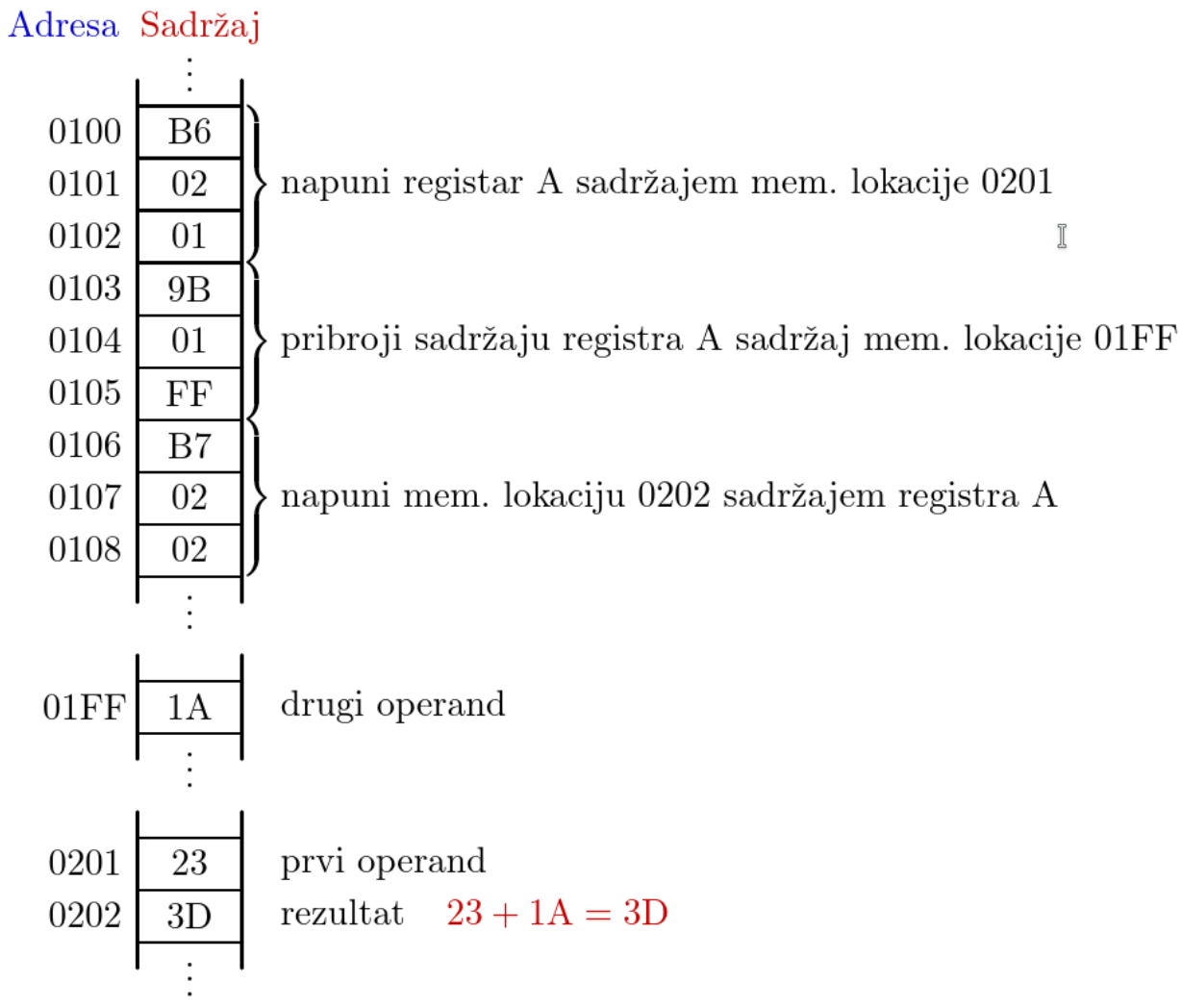
Da bismo nešto spremili ili pročitali kao sadržaj lokacije u memoriji, moramo imati dvije osnovne instrukcije:

- Spremi podatak na “tu-i-tu” adresu.
- Pročitaj podatak s “te-i-te” adrese.

Netrivijalna posljedica:

- Memorijske adrese su također podaci.
- Strojne instrukcije računala moraju sadržavati memorijske adrese podataka s kojima rade.

Zapis u memoriji: programski kod i podaci



Adrese 0100-0108 sadrže strojne instrukcije za zbrajanje dvaju brojeva. Jedan od tih brojeva je na adresi 0201. Drugi broj je na adresi 01FF.

Rezultat se zapisuje na adresu 0202.

Organizacija memorije

Memorijske adrese standarno zapisujemo u heksadecimalnom sustavu, npr: 4FA281BC.

Današnja računala za spremanje memorijske adrese obično koriste 64 bita.

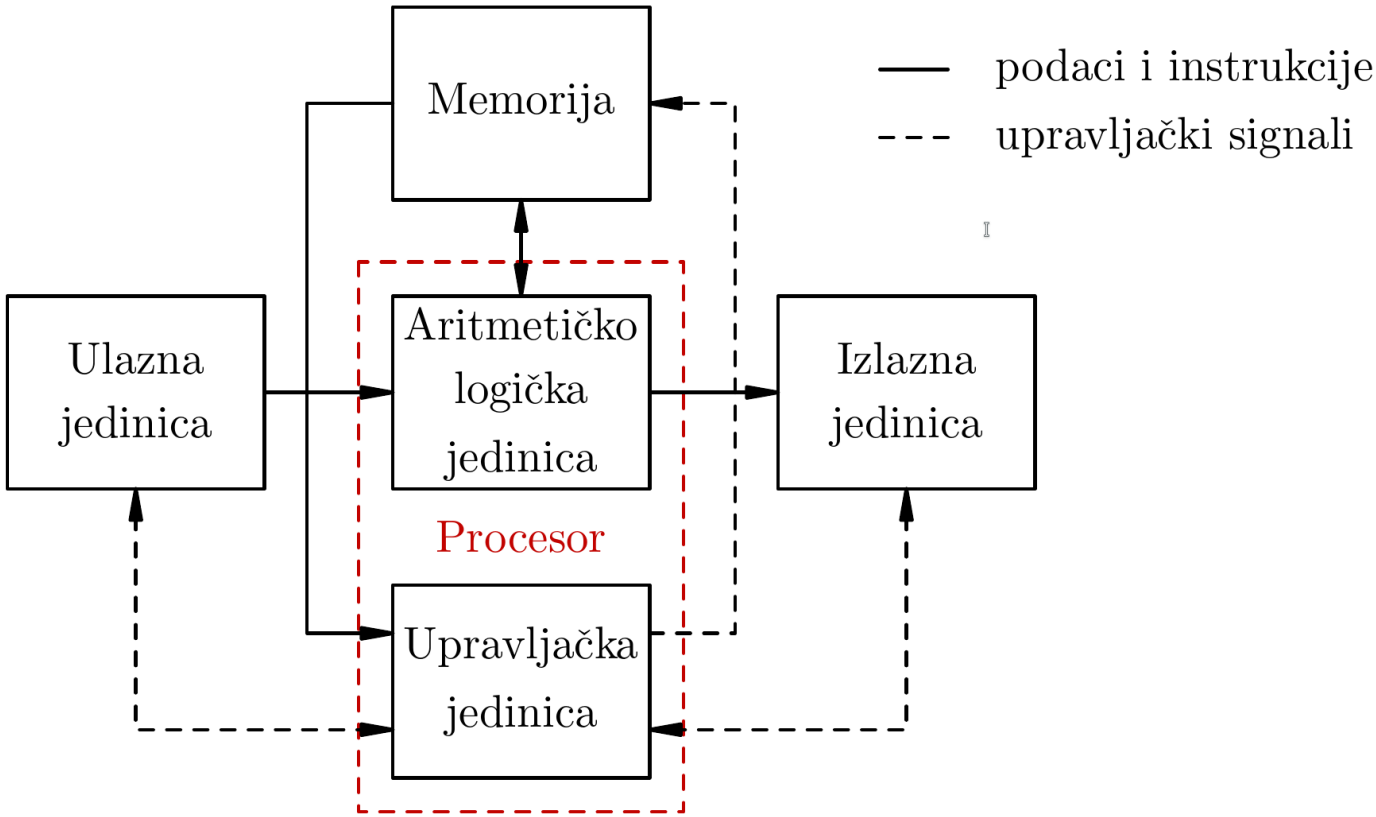
- Dakle, moguće je imati max. $2^{64} \approx 18.4 \cdot 10^{18}$ riječi (byte-ova) memorije, tj. 18.4 milijardi GiB.

Ranija 32-bitna računala su mogla adresirati max. 2^{32} byte-ova, tj. 4GiB memorije.

Napomena:

- 1kiB = 1024 byte-ova.
- 1MiB = 1024×1024 = 1 048 576 byte-ova.
- 1GiB = 1024×1024×1024 = 1 073 741 824 byte-ova.

von Neumannov model računala



Upravljačka jedinica - svi poslovi vezane za instrukcije:

- dohvat instrukcija iz memorije (engl. fetch),
- njihovu interpretaciju (engl. decode),
- njihovo izvršavanje (engl. execute).