

2. Kompjuteri, programi, programski jezici

Pošto je cilj ovog predmeta izučavanje kako se projektuju programi jezicima treće generacije (3GL), može biti od koristi podsećanje na neke opšte kompjuterske termine koji će biti korišćeni. Na primer, šta ćemo podrazumevati kad kažemo kompjuter, šta je kompjuterski program, šta su to generacije programskih jezika?

Nakon izučavanja ove lekcije bićete u stanju da:

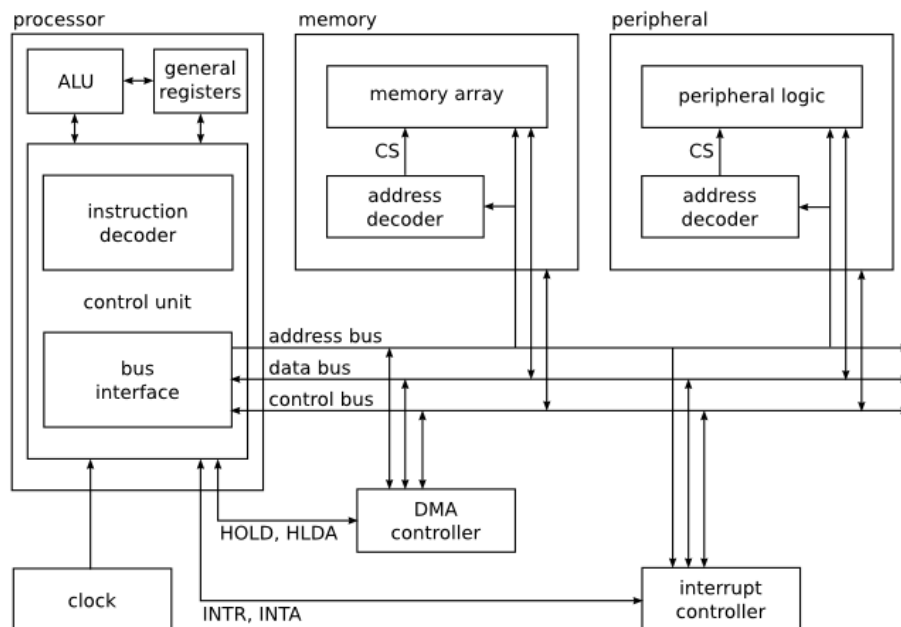
1. definišete šta je to kompjuter
2. definišete šta je to kompjuterski program
3. opišete razvoj kompjuterskih (programskih) jezika
4. skicirati jednostavan model kompjuterskog programa i njegovog okruženja
5. objasniti kako se podaci razlikuju od informacija

Osnovni elementi kompjuterskog sistema: hardver, softver, komunikacije

Kompjuteri su, verovatno, najsloženije mašine koje je čovek do sada stvorio. Ako bi hteli da definišemo kompjutere mogli bi smo reći:

- Kompjuter ima ulazne i izlazne uređaje.
- Kompjuter ima centralnu procesorsku jedinicu (poznatu kao CPU) za izvršavanje aritmetičkih i logičkih operacija.
- Kompjuter ima memoriju za smeštanje programa i podataka.
- Kompjuter može da izvršava nizove instrukcija (naredbi, komandi).

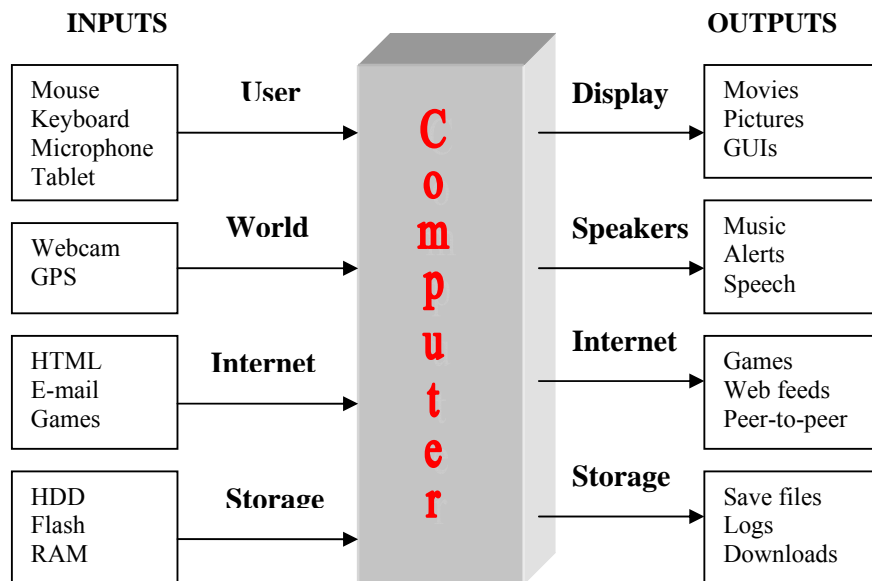
Na sledećoj slici prikazana je takozvana osnovna arhitektura kompjuterskog sistema onako kako je proučavaju hardverski profesionalci.



Slika 2.1 Osnovna arhitektura kompjutera

Na slici se uočavaju gore pomenuti delovi kompjutea – procesor, memorija, periferni uređaji kao i spojni putevi (magistrale) koji povezuju delove u jedinstvenu celinu. Preko magistrala se odvija sva komunikacija među delovima kompjutera, a magistrale (ili bus-ovi) služe za prenos podataka, adresa i kontrolnih komandi. Gornja arhitektura prikazuje takozvani stand-alone kompjuter koji nije povezan sa drugim kompjuterima. Danas je, međutim, skoro uvek slučaj da je kompjuter na kome radimo u lokalnoj ili globalnoj (Internet) mreži.

Za potrebe programera i softver inženjera pogodnija je malo uprošćenija ahitektura, koja u sebi sadrži i mrežni deo prikazana koja jeprikazana na sledećoj slici.



Slika 2.2 Programerski pogled na kompjuter

Šta je kompjuterski program?

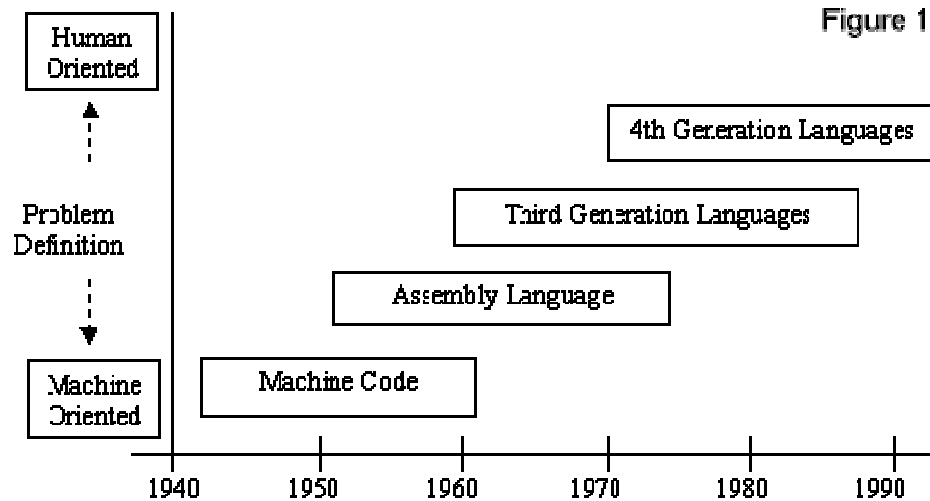
Ako pogledate definiciju koju smo dali za kompjuter, ali obrnutim redosledom počev od “Kompjuter može da izvršava nizove instrukcija” - dobićete definiciju kompjuterskog programa - to je niz (sekvenca) instrukcija koju CPU interpretira kao aritmetičke ili logičke operacije nad podacima iz memorije. Kompjuterski program je smešten u memoriji i takođe se sastoji od podataka i instrukcija. Ulazno-izlazni uređaji (I/O) omogućavaju mašini da prenosi podatke između nje i spoljnjeg sveta.

Instrukcije i programi zahtevaju neki redosled kojim će ih kompjuter izvršavati, a to je zapravo element za projektovanje. Projektovanjem programa određujemo koje su programske instrukcije i kakva struktura podataka je potrebna da bi mašina-kompjuter obavila neki željeni zadatak. Najznačajnije je to što projektant programa određuje niz (sekvenca), tj. redosled kojim će se instrukcije izvršavati u cilju uspešnog odvijanja i završetka programa.

Sada kada imamo jednu jasniju predstavu šta je kompjuter i šta je kompjuterski program, hajde da pogledamo šta su generacije jezika?

Generacije kompjuterskih jezika

Pojam generacije proistekao je iz razvoja koji je prikazan na sledećoj slici koja prikazuje razvoj kompjuterskih jezika od 1940-tih godina do danas.



Slika 2.3 Razvoj programskih jezika

Cilj ove slike nije da bude potpuno precizna, već da da ideju o vremenskim periodima i različitim generacijama koje se pojavljivale i međupovezanost jezika raznih generacija sa složenosću problema koji se rešavaju primenom kompjutera.

Vertikalna osa predstavlja definiciju problema, tj. pokazuje kako mi projektujemo (pišemo) kompjuterske programe. Na dnu ove skale se nalazi mašinski orjentisan način definisanja programa. To znači da se kompjuterski program izražava u jeziku koji je bliži mašini nego programeru-čoveku, kao što smo vi i ja. Na vrhu ove ose je ljudski orjentisan način izražavanja programa. To znači da je definicija problema - kompjuterski program -izražen na način bliži jeziku projektanta-čoveka nego jeziku mašine. I to je naravno poželjniji i bolji način za programiranje.

Mašina razume jezik izražen pomoću 1 i 0 (binarni kod), na primer:

101011110001010001000101111010010101010

dok čovek više voli da isti iskaz prikaže ovako:

Saberi total iz ove nedelje sa totalom iz prethodne nedelje

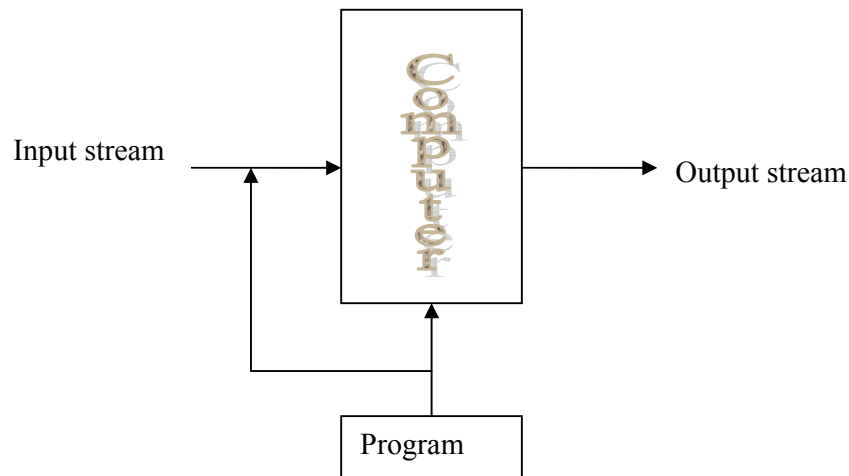
Horizontalna osa na slici prikazuje približno vreme kada se određena generacija jezika pojavila i ušla u upotrebu.

Imajte na umu da slika prikazuje samo aproksimativno, i da to što su jezici treće generacije završeni krajem 80-tih godina prošlog veka, to ne znači da oni nisu i danas u upotrebi. U praksi ne postoje jasne granice upotrebe jezika, pa nije redak slučaj da se kod nekih obimnijih projekta pojavi više generacija jezika u istom projektu. Tako se mogu naći projekti koji u sebi sadrže asemblerske programe kombinovane sa programima treće i četvrte generacije.

Poslednje generacije kompjuterskih jezika omogućavaju projektantima i programerima da pišu programe na jeziku vrlo sličnom prirodnom jeziku.

Model kompjuterskog programa

Pošto je kompjuterski program niz instrukcija koje “obrađuju” podatke, možemo to prikazati kroz sledeći prosti model:



Slika 2.4 Program transformiše ulaz i generiše izlaz

Model pokazuje da program procesira podatke koji dolaze sa ulaza – izvora podataka, a rezultati obrade idu na izlaz. Iako to nije obavezno slučaj, kod većih programa i programskih sistema ulaz i izlaz podataka su obično van samog programa. Ulaz je obično neki ulazni ili memorijski uređaj kao što je tastatura, disk, skener, traka, a izlaz, obično neki izlazni uređaj ili memorija kao što je ekran, štampač, disk, traka, crtač itd.

Sam program je zapravo niz instrukcija zapisanih u memoriji kompjutera kojim se precizno određuje na koji način će se transformisati ulazni podaci da bi se dobio željeni izlaz. No da bi takav pogodan program dospao u memoriju računara, potrebno je da bude najpre sačinjen u nekom programskom jeziku koji je razumljiv za čoveka-programera a potom preveden na jezik razumljiv procesoru kompjutera. Taj proces stvaranja novih programa i njihovo prevođenje na kompjuterski jezik jeste glavni predmet našeg razmatranja.

Podaci i informacije

Poslednja tema u ovom delu je kratko objašnjenje termina podatak i informacija. Ponekad se ovi termini koriste kao sinonimi, iako ne bi trebalo da bude tako. Podatak predstavlja neku vrednost (niz simbola, znakova) van konteksta i bez značenja. Informacija je podatak sa značenjem. Na primer, recimo da imamo sledeću listu brojeva:

24.56, 21.94, 27.23, 30.61

Šta oni znače? Značenje ovih brojeva zavisi od konteksta u kojem se koriste. Oni mogu predstavljati prosečne temperature tokom nekog perioda, ali isto tako mogu predstavljati i cenu nekog proizvoda. Takvi brojevi mogu biti smešteni na disk i možemo napisati kompjuterski program koji čita te brojeve sa diska - izvora podataka – i koji ih tretira kao temperature. Neki drugi program ih može takođe čitati sa diska i tretirati ih kao cene. Tako, zapravo sam program određuje kontekst podataka.

Šiže predavanja

U ovom predavanju upoznali smo se sa osnovnim konceptima koji će nam pomoći da bolje razumemo naredna predavanja. Sada ste u stanju da definišete šta se podrazumeva pod kompjuterom i kompjuterskim programom, imate jasniju ideju kako su se razvijali programski jezici, kao i jasniju predstavu o sličnostima i razlikama pojmova podatak i informacija.

O podacima i informacijama biće više reči u narednom predavanju.

Pitanja

1. Šta je to *instrukcija* u generalnom smislu?
2. Šta je to *mašina* u generalnom smislu?
3. Dajte jedan primer mašine i način na koji dajete instrukcije toj mašini. To ne mora biti kompjuter (recimo automat za koka-kolu)
4. Navedite četiri osnovne karakteristike kompjutera. Možete li da navedete još neke?
5. Šta se podrazumeva pod izrazom - *niz* (sekvenca)?
6. Šta znači *izvršavanje* instrukcije?
7. Šta je *kompjuterski program*?
8. *Mašinski kod* koristi prirodan jezik računara – šta to zapravo znači?
9. Šta je *ulaz podataka*?
10. Šta je *izlaz podataka*?