

## INFORMACIJA S PODRUČJA AUTOMATIZACIJE

*Zelimir SEISSEL — Zagreb*

Posljednjih pedesetak godina započela je nova era u djelatnosti čovjeka, era u kojoj je ostvarena mogućnost da se ljudski misaoni procesi i rutinski intelektualni rad zamjeni radom strojeva. U čitavoj ljudskoj povijesti unatrag tri i više tisućljeća sva nastojanja i ostvarenja ljudske znanstvene i tehničke misli bila su usmjerena na to da se pomoću strojeva zamijeni i mnogostruko poveća fizička snaga čovjeka. A sada usporedo teku dva nezadrživa nastojanja, jedno staro tritisućljeća ili više, i drugo, mlado tek pedesetak godina u kojima su stvorena elektronska računala.

Elektronsko računalo je vrlo kompliciran stroj koji zahvaljujući svojoj konstrukciji može ne samo u vrlo kratkom vremenu izračunati golemi broj računskih operacija i memorirati rezultate, nego može i donositi odluke obavljanjem jednostavnih logičkih operacija. Zbog toga je u stanju uspješno rješavati zadatke, od jednostavnog množenja do kompliciranih matematičkih operacija, kao i vođenje najzamršenijih tehničkih procesa uključujući ovamo i tako tešku operaciju kao što je čovjekov lijet na Mjesec.

Jedna od najvažnijih karakteristika elektronskih računala je brzina rada. Moderna elektronska računala obavljaju sto tisuća do tri milijuna operacija u sekundi. Takva brzina rada omogućila je da se izvede oko osamdeset milijardi proračuna dnevno potrebnih u svakom danu lijeta kompozicije Saturn—Apollo na relaciji Zemlja—Mjesec.

Kako u naše vrijeme elektronska računala prodiru u sve grane tehničke djelatnosti, to će biti cilj ovog članka da čitaoca informira pomoću jednog jednostavnog primjera o načinu kontaktiranja čovjeka i elektronskog računala.

Kao i čovjek, računalo ima svoj jezik na kome mu treba izdati naređenja da bi ono moglo razumjeti i dati rješenja postavljenih mu zadataka. Jezik je računalu u njegov »mozak« usadio konstruktor računala, a tko god želi razgovarati sa računalom mora taj jezik naučiti. Danas su najpoznatiji jezici računala Asembler, Algol, Cobol i Fortran. Primjer koji će biti prikazan dan je u jednom obliku Fortrana, jezika prilagođenog za obradu naučnih i tehničkih podataka.

Svaka operacija na elektronskom računalu sastoji se od dva odvojena dijela. Prvi je davanje računalu niza naredbi po kojima će ono u drugom dijelu operacije primiti podatke (recimo mjerenja) i obraditi prema instrukcijama danim u naredbama. Prva operacija zahtijeva stvaranje programa elektronskog računala. Program je na jeziku računala napisan niz naredbi i obješnjenja potrebnih za izvršenje željenog zadatka. Program ubušen u papirnu traku ili kartice računalo će pročitati i zapamtiti u svojoj memoriji i o tome nas obavijestiti. Ono će nas čak upozoriti i na to da li smo učinili kakvu gramatičku pogrešku na njegovom jeziku pa nas stoga ne razumije.

Ovdje će bit prikazan i objašnjen jedan jednostavan program na Fortran jeziku. U primjeru je zbog pojednostavljenja odabrana forma koju bi izabrao neki početnik programer, ali koja bi ipak dala rezultate postavljenog zadatka.

Primjer se sastoji od zadataka da elektronsko računalo izračuna udaljenost između niza točaka u ravnini koje su zadane svojim koordinatama. Matematička formulacija zadataka je dakle ovakva

$$DY = YZ - YP \quad DX = XZ - XP \quad D = \sqrt{DY^2 + DX^2}$$

U Fortran jeziku simboli za zbrajanje i odbijanje su jednaki kao i u matematici, dakle + i -. Simbol za kvadriranje su dvije zvjezdice iza kojih je napisana potencija (npr.  $A^2$  u Fortranu glasi  $A^{**2}$ ). Drugi korjen se na Fortranu označava kraticom engleskog jezika SQRT, a iza te kratice stoji izraz pod korjenom, u zgradi.

Prema tome formule za izračunavanje udaljenosti između dvije točke u ravnini zadane koordinatama na Fortran jeziku glase

$$DY = YZ - YP \quad DX = XZ - XP \quad D = \text{SQRT}(DY^{**2} + DX^{**2})$$

Prijevod matematičke formule konkretnog primjera na Fortran jeziku bio je dakle jednostavan. Međutim da bi računalo znalo što će učiniti s tim formulama treba mu to i riječima objasniti, treba mu dakle dati naređenje. Svaka naredba piše se u programu u jedan redak a naredbe se međusobno povezuju brojevima koji se pišu na početku svake naredbe. U primjeru će biti upotreb- ljeno sedam naredbi koje će najprije biti objašnjenje. To su uglavnom riječi engleskog jezika kako slijede.

*Naredba ACCEPT TAPE* — Naredba se odnosi na učitavanje trake koju će računalo naknadno primiti a u našem primjeru na njoj će biti ubušene koordinate točaka između kojih će se računati udaljenosti. Treba dakle napisati:  
ACCEPT TAPE YP, XP, YZ, XZ

*Naredba FORMAT* — Ova naredba informira računalo da li će primiti cijele ili decimalne brojeve i u kojem obliku. Kako je dakle naredba objašnjenje naredbi ACCEPT TAPE one se moraju povezati, a to se čini na taj način da se iza naredbe ACCEPT TAPE stavi neki broj, a isti taj broj stavi se ispred naredbe FORMAT. Spajanjem ovih dvaju naredbi:

1 ACCEPT TAPE 2, YP, XP, YZ, XZ    2 FORMAT (4 F 10.2)

stavlja se računalu do znanja da treba učitati traku na kojoj su ubušene koordinate dvaju točaka i da su one decimalni brojevi sa deset znamenaka od kojih su dvije decimalne (izraz u zgradi iza naredbe FORMAT).

*Naredba IF (YP)* — Naređenje računalu da ispita da li je izraz u zgradi negativan, nula ili pozitivan. Ako se napiše

IF (YP) 11, 11, 3

pa ako računalo ustanovi da je YP negativan ili nula ono će preskočiti u računanju sve druge naredbe i izvršiti naredbu koja nosi broj 11 i sve koje iza nje slijede. Ako međutim računalo nađe da je YP pozitivan ono će ići na naredbu broj 3 i izvršiti sve naredbe koje iza nje slijede.

*Naredba PRINT* — Ova naredba aktivira štampač računala i ono će odštampati veličine koje su napisane iza riječi PRINT. Kako treba izvijestiti računalo u kojem obliku želimo da se rezultati štampaju to se naredba PRINT mora povezati s novom naredbom FORMAT. (kao i kod naredbe ACCEPT TAPE).

*Naredba GO TO* — Po ovoj naredbi računalo će nastaviti rad na naredbi koja je označena istim brojem koji je stavljen iza naredbe GO TO. (na pr. GO TO 1 znači: idi na naredbu broj jedan izvrši nju i sve operacije koje iza nje slijede).

*Naredba STOP* — Daje računalu na znanje da je učitavanje programa završeno.

*Naredba END* — Kada računalo učita ovu naredbu prekine radom. Iz ovih sedam naredbi dade se sastaviti slijedeći program za računanje udaljenosti između dvije točke u ravnini. Prema napisima naredbama računalo će u neprekidnom nizu računanja izračunavati udaljenosti i nakon što izračuna zadani broj udaljenosti samo će prekinuti rad na izvršavanju programa.

```
C   PROGRAM ZA RAČUNANJE UDALJENOSTI
C   FORTAN PROGRAM
C
1   ACCEPT TAPE 2, YP, XP, YZ, XZ
2   FORMAT (4 F 10.2)
3   IF (YP) 11, 11, 4
4   DY = YZ — YP
5   DY = XZ — XP
6   D = SQRT (DY**2 + DX**2)
7   PRINT 8, D
8   FORMAT (F 6.2)
9   GO TO 1
10  STOP
11  END
```

Ovaj program ubušen u traku računalo će učitati i čekati da mu dademo traku podataka na kojoj će biti ubušene koordinate između kojih treba računati dužine. Dobivanjem trake podataka računalo počinje izvršavanjem programa.

Prema naredbi broj 1 računalo će sa trake podataka učitati koordinate prvih dvaju točaka (YP, XP, YZ, XZ).

Prema naredbi broj 3 računalo će naći da je YP pozitivan broj i ići dalje na naredbu broj 4 i 5 tj. izračunati će koordinatne razlike i zatim po naredbi 6 udaljenost između dvije učitanе točke.

Izvršavajući naredbu broj 7 računalo će odštampati izračunatu udaljenost u obliku (F 6.2). (decimalni broj sa šest znakova od kojih su dvije decimale).

Nakon toga naredba broj 9 vraća računalo natrag na naredbu broj 1 te ono po njoj učitava koordinate drugog para točaka sa trake podataka. Tako se čitav taj ciklus računanja ponavlja onoliko puta koliko dužina želimo izračunati. Ako smo nakon svih parova točaka između kojih treba izračunati dužinu staviti jednu fiktivnu točku sa koordinatama nula, dolazi do izražaja naredba broj 3. Po njoj će računalo naći da je  $YP = \emptyset$  i ići na naredbu 11 END i prekinuti rad.

Iz opisa programa vidi se da je tok računanja potpuno automatiziran. Brzina dobivanja rezultata određena je mogućnošću štampanja štampača. Kako je ta brzina prosječno trinaest znakova u sekundi to bi se za jedan sat rada na izvršavanju ovoga programa dobilo oko sedam tisuća dužina izračunatih iz koordinata. Uzevši u obzir i vrijeme koje je potrebno za ubušivanje koordinata u traku ili kartice te vrijeme potrebno za kolacioniranje upisa, za računanje sedam tisuća dužina dolazi se do nekih dvijestopedeset radnih sati za izvršenje zadatka a to znači da je potrebno uložiti po najboljoj procjeni cca 60% manje vremena nego bi to bilo potrebno u slučaju računanja klasičnim načinom uz upotrebu stolnog računskog stroja.

Ovakva ušteda na vremenu objašnjava potrebu i velik stupanj kompjutorizacije u tehnički naprednim zemljama svijeta.