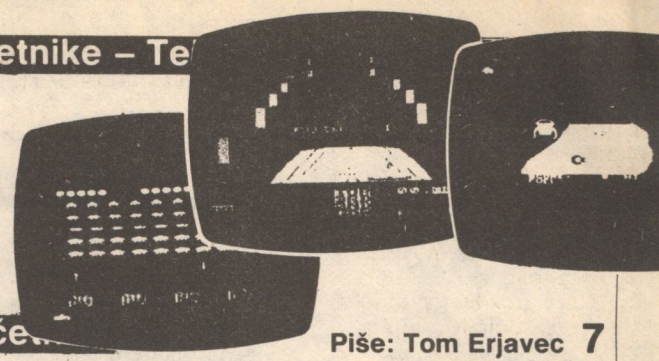
OSNOVE RAČUNALNIŠTVA



Teleksov tečaj za začetnike – Teleksov tečaj za zače

Piše: Tom Erjavec 7

MEMO: Zadnjič smo govorili o sintaksi zbirnega jezika, o pisanju programov v zbirnem jeziku in spoznali smo, kako zbirnik prevaja zbirni jezik v strojno kodo. Ogledali smo si primere programov in njihovih prevodov. Omenili smo razlike v strojni kodi, ki jo generirata zbirnik in prevajalnik visokega zbirnega jezika. Zato danes povejmo kaj še o višjih programskih jezikih in strukturiranem programiranju.

Višji programski jeziki

Doslej smo se že lahko prepričali o tem, da en sam stavek zbirnega jezika kaj malo pove. Na primer »seštej dve števili« ali pa »skoči na naslov ta-in-ta« je dokaj malo vsebine za veliko pisanja. Posebno se zaplete zadeva, če želimo obdelovati kompleksne podatkovne strukture. Ker so programerji pogosto rabili za opisovanje kompleksnih programov bolj izrazno orodje, obenem pa so želeli svoje programe hitreje sestavljati, so prišli na dan višji programski jeziki. Za nekatere ste gotovo že slišali: BASIC, FORTRAN, COBOL, ALGOL, PASCAL, FORTH...

Komplikatorji in interpretorji

Višji programski jezik ima prevajalnik, ki prevede program, napisan v tem jeziku, v strojno kodo. Namesto prevajalnika ima lahko poseben program, ki stavke jezika »interpretira« enega za drugim, sproti in opravlja želene funkcije. Vidimo torej, da obstajata dva načina priprave programov v višjih jezikih. Prevajanje in poznejše izvajanje ali pa sprotno prevajanje in sprotno izvajanje. Prvemu načinu pravijo kompiliranje (prevajanje celotnega programa v strojno kodo). Programu, ki izvede prevajanje, pravijo prevajalnik (kompilator), v angleščini COMPILER. Strojna koda takega programa je pravzaprav precej podobna kodi, kakršno bi generiral zbirnik ob prevodu programa v zbirnem jeziku, ki bi opravljal enako nalogo.

Drugi način je interpretiranje programa. Namesto da bi ga v višjem jeziku prevedli v strojno kodo, ga predamo kakemu drugemu progamu, ki bere njegove stavke ter jih izpolnjuje enega za drugim v skladu s pravili, zapisanimi za vsak možen programski stavek posebej v obliki manjših podprogramov. Takemu programu, ki interpretira uporabnikov program v višjem jeziku, pravimo interpretor (angl. INTERPRETER). Vsakdo že pozna BASIC, ki ga uporabljajo popularni hišni računalniki (npr. Sinclair). BASIC je najbolj razširjen interpretor. Značilno pa je še, da ima skoraj vsak BASIC povsem svoj »dialekt« in da se med seboj tako razlikujejo, da programov z enega računalnika ni mogoče uporabiti na drugem.

Glavni razliki med obema načinoma sta hitrost in način izvajanja. Izvajanje kompiliranih programov je neprimerno hitrejše kot interpretiranih. Stvar je preprosta. Prevajanje je dolgotrajno (relativno). Npr. kompiliranje srednje dolgega programa (1000 stavkov) traja nekaj minut, zato pa program potem izredno hitro poteka, saj stroj že natanko ve, kaj mora delati, ko čita strojno kodo.

Pri interpretaciji programov pa mora stroj najprej razvozlati, kaj mora v naslednjem stavku narediti in nato še izpeljati zahtevano. Zato traja izvajanje mnogo dalj časa. Prednost pa je v tem, da lahko program izmenično popravljamo in nato takoj izvajamo, ali pa izvedemo samo en del, namesto da bi vsakič za vsako malenkostno spremembo prevajali ves program, kot bi to morali storiti s kompilatorjem.

Z višjimi programskimi jeziki smo dobili stavčne konstrukte, ki so omogočali priročne odločitve, ponavljanja pod določenimi pogoji, izračunavanje matematičnih funkcij, definiranje in obravnavanje kompleksnih podatkovnih struktur ipd. Uveljavilo se je tudi strukturirano programiranje, ki si je zastavilo za cilj, da bi bili programi grajeni iz manjših zaključenih enot, da bi bili logični, razumljivi in da v njih ne bi bilo dolgih preskokov na drûge dele programa. Najhujši nasprotnik strukturiranega programiranja je stavek GOTO, ki se pojavlja v mnogih jezikih, a ga moderni programerji skušajo izločiti, ker povzroča skakanje na poljubno mesto v programu zaradi česar lahko nastane silna nepreglednost. Starejši višji jeziki niso bili strukturirani, čeprav se je v njih vseeno dalo držati pravil strukturirane-

Nova knjiga: ABC računalništva 1

Ljubiteljem mikroračunalništva in hišnih računalnikov se obeta nov vir informacij. Zveza organizacij za tehnično kulturo Slovenije je v sodelovanju z avtorjem Ivanom Gerličem pripravila knjigo ABC računalništva. Trenutno tečejo zadnje priprave pred izdajo in imeli smo priložnost, da si ogledamo besedilo med vnašanjem v računalnik. Knjiga pa ne bo samo pisala o računalnikih, ampak jo bo računalnik tudi oblikoval. Celotna oblika knjige z vsem vnašanjem besedila vred bo narejena v treh dneh. Ko berete te vrstice, je knjiga že v tisku. Izšla bo 25. maja.

In vsebina? Zelo zanimiva. Po uvodnih razmišljanjih nam avtor predstavi zgodovino računalništva od Pascalovega računskega strojčka do najnovejših mikro tehnologij in Josephsonovega spoja. Sledi poglavje o anatomiji računalnika, kjer so predstavljeni glavni sklopi računalnika, kot so pomnilnik, procesna enota, vhodno-izhodne enote itd. V poglavju o mikroprocesorjih in mikroračunalnikih so podani glavni gradniki mikrojev in nekaj dodatne periferne opreme. Opisani so tudi hišni računalniki. Avtor nato omeni najvažnejše programske jezike in opiše najbolj pogoste stavčne konstrukte. V zadnjem, a obsežnem poglavju, je predstavljen jezik BASIC, ki je tako priljubljen med lastniki hišnih računalnikov. Podanih je tudi precej primerov kratkih programov. Na koncu sta še dva dodatka. V prvem je seznam računalniških ukazov (BASIC), v drugem pa še izčrpen slovarček računalniških pojmov.

Knjiga naj bi v prodaji stala okrog 400 din, kar je gotovo izredno ugodna cena za bogato vsebino.

T. E.

Vaše vprašanje, strokovnjakov odgovor

V naši rubriki bomo nekaj prostora namenili tudi vprašanjem bralcev. Pišite nam na naslov **Uredništvo Teleksa, Titova 35, 61000 Ljubljana (z oznako: Za osnove računalništva),** izbrali bomo najzanimivejša vprašanja, takšna, ki bi utegnila zanimati zlasti najširši krog začetnikov. Vprašanja so lahko teoretična, skušali pa bomo posredovati tudi druge informacije, npr. naslove klubov, specializiranih revij.

Zanima me programski jezik PROLOG. Prijatelj trdi, da je prolog jezik najnovejše generacije inteligentnih računalnikov.

B. Č.
Ljubljana

Računalniška inteligenca je še dokaj nejasna zadeva, ki ni povsem enolično definirana. Lahko rečemo, da je to sposobnost računalnika, da posnema logično razmišljanje in odločanje človeka. To pomeni, da se mora znajti v novih in nepredvidenih situacijah. Na osnovi podobnih situacij in znanih dejstev se mora nato odločiti za ustrezno ukrepanje. Prolog je programski jezik, ki temelji na matematični logiki, in omogoča programiranje računalnika za reševanje problemov, kot so pomoč zdravniku pri diagnostiki bolezni, analiza biokemičnih struktur, simbolično reševanje enačb in podobno.

Prolog se že v zasnovi razlikuje od jezikov kot so pascal, fortran ali basic, kjer programiramo na podlagi točno določenega algoritma, ki bo iz vhodnih podatkov izračunal rezultate. V prologu opisujemo dejstva in relacijo med objekti, ki se pojavljajo v določenem problemu. Ne opisujemo pa načina, kako bo računalnik problem rešil. Način reševanja je pogojen s semantiko (to je veda o pomenu besed) prologa, dalje s tem, kakšna nova dejstva lahko prolog »potegne« iz že znanih in le deloma z zahtevami programerja.

V prologu ločimo tri osnovne stavke. To so: deklaracija določenih dejstev o objektih in njihovih medsebojnih relacijah, definiranje pravil in postavljanje vprašanj. Na primer: Če bi hoteli v prologu povedati, da ima Jana knjigo, Janez pa denar, bi to zapisali takole:

ima (jana, knjiga).
ima (janez, denar).

Besedica »ima« pomeni relacijo, »jana«, »knjiga«, »janez« in »denar« pa so objekti, ki jih relacija opisuje. S tema stavkoma smo opisali podatkovno bazo. Zdaj lahko postavimo vprašanja. Na primer:

?– ima (jana, knjiga).
?– ima (janez, knjiga).

Prvo vprašanje pomeni »ali ima jana knjigo?« Prolog pogleda v bazo podatkov in odgovori »yes«. Na drugo vprašanje »ali ima janez knjigo?«, pa bi odgovorili z »no«. Zakaj pa to, saj nismo nikjer dejali, da Janez nima knjige? Ugotovimo torej eno od pravil prologa: velja samo tisto, kar smo opisali v podatkovni