IV. SOFTVER

Programi i programski paketi se jednom riječju nazivaju softver (engl. software) i ovom se riječju generalno obuhvataju sve vrste računarskih programa.

I na ovom se mjestu ističe da računar predstavlja jedinstvo hardvera i softvera.

1. RAZVOJ SOFTVERA

Razvoj softvera glavni je faktor u razvoju informacionih tehnologija. Primjena softvera se raširila na sva područja ljudskog djelovanja. Samo na američkom tržištu, prodaja poslovnog softvera za personalne računare iznosila je 1981. godine oko 500 miliona USD, dok se prodaja u 1986. godini kretala oko 4,6 milijarde USD. Danas ove brojke ostvaruju pojedini proizvođači.

Mogućnosti razvoja softvera, pored postignutih rezultata, i dalje su neslućene. Razvoj programskih jezika, a posebno jezika četvrte generacije, objektno orijentiranog programiranja i pokušaja stvaranja univerzalnih programskih alata, dovodi do toga da se softver sve više približava korisniku i postaje jednostavniji za upotrebu. Zbog mnoštva gotovih programskih paketa za najrazličitije namjene, danas najčešće nije racionalno razvijati svoj softver. Ali, ostaje potreba za razvojem specifičnih softverskih rješenja za pojedina područja, kao i dodatnih programa, koji predstavljaju dogradnju već kupljenih softveskih paketa.

2. VRSTE SOFTVERA

Nema jasne granice između pojedinih vrsta softvera. Na primjer, dijelove programskog paketa za obradu teksta imaju mnogi programski paketi, ali se po ostalim svojim karakteristikama svrstavaju u druge vrste softvera.

Jednostavno rečeno, softver se dijeli na:

- sistemski i
- aplikativni,

uz napomenu da tu nema izričito definisane jasne granice razdvajanja ove dvije grupacije softvera.

U osnovi se softver može podijeliti i na slijedeći način:

- sistemski,
- uslužni (paketski), } aplikativni softver
- aplikativni i
- programski softver.

3. SISTEMSKI SOFTVER

Sistemski softver obuhvata:

- softver postavljen u ROM čipovima, koji služi za inicijalizaciju računara i
- operativni sistem.

Sistemski softver se sastoji od računarskih programa koji kontrolišu i podržavaju rad računarskog sistema i njegove aktivnosti na obradi podataka.

Računar se ne može pokrenuti, niti može uraditi bilo kakav zadatak bez sistemskog softvera.

Ranije je sistemski softver, pored operativnog sistema, obuhvatao i: komunikacijski softver, programe prevodioce, pomoćne (servisne) programe i programe za upravljanje bazama podataka (DBMS). Danas se taj dio softvera posebno nabavlja. Osim toga, mogu se nabaviti i različiti operativni sistemi. Kao dio sistemskog softvera može se pojaviti i emulacijski softver.

3.1 Operativni sistem

Operativni sistem predstavlja interfejs između korisnika i računarskog sistema.

Predstavlja skup programa koji upravljaju radom rašunarskog sistema, odnosno omogućava:

- izvršavanje aplikativnih programa i koordinira njihov rad,
- provodi kontrolu i upravlja centralnim procesorom, centralnom memorijom, perifernim uređajima, pohranjivanjem podataka i informacija.

Svi ostali zadaci su prepušteni aplikativnom softveru. Uobičajeno se isporučuje zajedno s računarom.

Može se reći i da predstavlja skup programa koji služe za upravljanje resursima računara (pri čemu pod osnovnim resursima

podrazumijevamo: CPU (procesor), operativna memorija, ulazno-izlazni uređaji i fajlovi).

Osim navedenih, OS vrši i neke pomoćne funkcije, kao što su:

- dijagnostika otkaza hardvera,
- evidencija softverskih grešaka,
- administrativne funkcije (vođenje evidencije o upotrijebljenim resursima).

Primjenom operativnog sistema, postupak koordinacije i sinhronizacije rada elektronskih i mehaničkih uređaja računara se automatizuje.

S obzirom na broj korisnika koji OS opslužuju, isti se dijele na:

- jednoskorisničke i
- višekorisničke operativne sisteme.

Kod višekorisničkih sistema nužan je mehanizam zaštite korisnika i podataka od neautorizovanog korištenja, kao i zaštite OS od samih korisnika. Redoslijed izvršavanja programa može se regulirati dodjelom različitih stepena prioriteta korisnicima. Za ove poslove s operativnim sistemom u pravilu je zadužen sistem-administrator.

Poznati operativni sistemi su: MS DOS, OS/2, VMS, **UNIX, Windows, Linux**.

Budućnost rada na PC računarima po svemu sudeći je u velikoj mjeri vezana za Windowse. Osnovna računarska pismenost podrazumijeva i poznavanje Windowsa. Zastupljenost Windows aplikacija sve je veća. Prema nekim procjenama, u 1994. godini 99% svih sjevernoameričkih korporacija koristilo se barem jednom od Windows aplikacija. U 1993. godini taj postotak je iznosio samo 83%.

4. APLIKATIVNI SOFTVER

Aplikativni softver se sastoji od programa koji upravljaju računarskim sistemom da bi se izvršile konkretne aktivnosti obrade podataka za koorisničke potrebe. Dakle, računar se koristi za rješenje specifičnog problema ili konkretnog posla za korisnika.

Aplikativni softver korisnik može kupiti na tržištu kao gotov, ili se koristiti adekvatnim softverom u javnom vlasništvu, a može i naručiti izradu softvera, ili ga izraditi u vlastitoj režiji.

U ovu kategoriju spadaju:

- aplikativni programi opće namjene (tekst-procesori, spredšit alati za tablične proračune, programi za grafiku itd.),
- poslovni aplikativni programi (knjigovodstvo finansija, marketing, proizvodnja),
- naučni aplikativni programi (naučne analize, inženjerski dizajn),
- te ostali aplikativni programi (obrazovni softver, igre, muzika i umjetnost).

4.1 Kupljeni (gotov) softver

Količina gotovog softvera na tržištu raste geometrijskom progresijom. Neke od najznačajnijih grupacija ovog softvera su:

- antivirus softver i softver za zaštitu sistema,
- backup softver, za zaštitu podataka i programa,
- CAD/CAE/CAM (Computer Aided Design / Computer Aided Engineering / Computer Aided Manufacturing) softver. Namijenjen je za potrebe računarom podržanog inženjerskog dizajna i razvoja proizvoda, za inženjerske analize i istraživanja, te za proizvodnju;
- database softver, za rad s bazama podataka,
- softver za tehničke (PDMS – Product Data Management System) i poslovne informacione sisteme (ERP – Enterprise Resource Planning),
- softver za obradu tabela i analizu podataka,
- softver za potrebe statistike,
- grafički softver, za obradu slika i crteža, te za prividnu stvarnost (virtual reality software),
- softver za vođenje projekata,
- softver za programiranje i razvoj softvera,
- softver za računarske mreže,
- multimedijski i prezentacijski softver,
- edukacijski softver,

- softver za igru, zabavu i kućnu upotrebu.

5. PROGRAMSKI JEZICI – PROGRAMSKI SOFTVER

Programski jezik je vještačka tvorevina i sastoji se od simbola grupisanih u riječi. Po svojoj strukturi sličan je prirodnom, govornom jeziku čovjeka. Služi za izradu programa (softvera) koji treba da formalizuju određene algoritme s ciljem rješavanja problema uz podršku računarom.

Programski jezici se dijele u više kategorija.

Svaki programski jezik mora zadovoljiti dva osnovna zahtjeva:

- da je razumljiv za čovjeka i
- da ima mogućnost automatskog prevođenja u oblik razumljiv računaru.

Do sada je definisano nekoliko stotina programskih jezika, od kojih je samo desetina u praktičnoj upotrebi.

Generacije programskih jezika su:

- mašinski jezici, koji se nazivaju i programski jezici prve generacije (1GL, 1950-1954),
- asembleri i makro-asembleri su jezici druge generacije (2GL, 1955-1959) i nalaze se između mašinskih jezika i viših programskih jezika,
- jezici treće generacije (3GL) su viši programski jezici, koji su proceduralni,
- jezici četvrte generacije (4GL), u koje spadaju: SQL, HTML, PHP, ASP, tj. neproceduralni jezici, sa usko specijaliziranom namjenom, prema kojoj mogu biti:
  - opisni (služe opisivanju dokumenata - PostScript, HTML),
  - upitni (za generiranje podskupova iz baza podataka i formiranje izvještajâ - SQL),
  - grafički (LabView, G – jezici za programiranje virtualnih instrumenata).

5.1 Niži programski jezici ili programski jezici niskog nivoa

U ovu grupu spadaju mašinski jezici i simbolički mašinski jezici – asembleri i makro-asembleri. Niži programski jezici su platformski, okrenuti računaru (instrukcije se opisuju simbolički).

Sve hardverske komponente računara na najnižem nivou razumiju jedino jezik koji se sastoji od binarnih jedinica i nula. Prilikom projektovanja računara, CPU se projektuje tako da interpretira skup instrukcija koje se nazivaju instrukcijski skup. Svaka instrukcija u ovom skupu ima jedinstven binarni kod koji CPU može da interpretira direktno. Ovaj binarni kod se zove mašinski kod instrukcije, a skup svih mašinskih kodova instrukcija se zove mašinski jezik. Program u mašinskom jeziku često se naziva izvršni program.

U današnje vrijeme, programiranje u mašinskom jeziku je težak i neefikasan način programiranja.

5.2 Viši programski jezici ili proceduralni programski jezici

Ideja je bila da se programski jezik približi čovjekovom načinu izražavanja (instrukcije su obično izvedene iz riječi engleskog jezika).

Zbog problema programiranja u mašinskom jeziku, pokazalo se da računar ne može naći širu primjenu ukoliko se ne poboljša komunikacija korisnika sa sistemom. Razvoj raznih vrsta sistemskog softvera obezbijedio je ove pretpostavke. Tu se prije svega misli na razvoj tzv. programskih jezika višeg nivoa, koji su bliži i razumljiviji korisniku.

Pošto računar može da izvršava samo programe u formi mašinskog jezika, programi pisani u višem programskom jeziku (izvorni - source programi) se prevode na mašinski jezik, a za prevođenje se koriste posebni programi, tzv. programi prevodioci, u žargonu nazvani kompajlerima (iako je pravilnije zvati ih kompilatorima).

Viši programski jezici su donijeli slijedeće prednosti:

- olakšano programiranje (bliži čovjekovom prirodnom jeziku),
- skraćeno vrijeme obuke u programiranju,
- nije nužno detaljno poznavanje hardvera računara,
- prenosivost programa (program se može prevesti na drugom tipu računara i izvršiti bez ikakvih izmjena u programu).

Danas postoji veliki broj viših programskih jezika. Oni se mogu klasifikovati na osnovu raznih kriterija. Najčešća klasifikacija je po namjeni.

U kategoriju opće namjene (općenamjenski jezici-algoritamski orijentirani jezici) ubrajaju se jezici koji služe za rješavanje numeričkih problema u oblasti tehnike i drugih disciplina koje koriste matematičke modele, kao i za rješavanje numeričkih problema vezanih za poslovnu obradu podataka i informacione sisteme. Najpopularniji jezici u ovoj klasi su: FORTRAN, BASIC, PASCAL, COBOL, C, C++, C# i drugi.

Jezici vještačke inteligencije čine drugu grupu viših programskih jezika. Ovi programski jezici se primjenjuju u oblastima simboličkog računanja, mašinskog prepoznavanja i zaključivanja, robotike i ekspertnih sistema. Najpopularniji su LISP i PROLOG.

Treću grupu čine jezici za sistemsko i konkurentno programiranje. Namijenjeni su za pisanje sistemskog softvera. To su: ADA, BLISS, MODULA-2, OCCAM i drugi jezici, a danas se u ovu svrhu najviše koriste jezici C++ i C#.

5.3 Objektno orijentirani programski jezici

Neki smatraju da ovi jezici potkategorija proceduralnih programskih jezika.

Programeri su decenijama pokušavali da odgovore na jednostavno pitanje: "Kako pisati dobre programe?" Pokazalo se da su dobri programi modularni, lako shvatljivi, da se mogu primjenjivati i u situacijama za koje nisu specijalno pisani, da se mogu srazmjerno lako mijenjati i dopunjavati. Sedamdesetih godina vodile su se žustre debate oko tzv. struktuiranog, odnosno proceduralnog programiranja kao metoda za pisanje programa, a epilog danas znamo: proceduralno programiranje je ušlo u arsenal svakog programera koji iole drži do sebe. Uporedo, mada bez pompe, evoluirale su i druge programske metode. Takozvano objektno orijentirano programiranje (skraćeno OOP) predstavlja najznačajniji pomak u tom pravcu, a OO jezici doživjeli su krajem osamdesetih veliku popularnost.

Najveći broj računarskih jezika podržava model izračunavanja koji se može sažeto iskazati na slijedeći način: aktivne procedure djeluju nad pasivnim podacima. Svi najpoznatiji programski jezici su u ovoj grupi: pascal, fortran, algol, kobol, bejzik i drugi. U njima postoji oštra razlika između podataka i informacija koje se obrađuju i procedura (potprograma, funkcija) koje ih obrađuju. Procedure i podaci su kod ovih jezika različiti entiteti, pa se programiranje svodi na kreiranje programskih modula i njihovo pozivanje u toku izvršavanja programa. Svaki jezik definiše na svoj način prenošenje podataka u procedure, kao i u kojim se sve programskim blokovima može doći do jedne varijable.

Objektno orijentirani jezici zasnivaju se na pretpostavci da jedan jedini entitet – objekat – treba da obuhvati i podatke i procedure. Podaci u objektu mogu biti promijenjeni samo preko procedura (često se nazivaju metodama) koje pripadaju tom objektu. Izračunavanja se izvode slanjem poruka (zahtjeva) objektima u programu. Objekti odgovaraju na poruke tako što stvaraju nove objekte i vraćaju ih kao rezultate. Svaki objekat je poput nezavisnog procesora čije se ponašanje može dokučiti samo na osnovu odgovora koje odaje. Na primjeru običnog sabiranja dva broja može se shvatiti velika razlika između ova dva pristipa.

Pretpostavka je da treba sabrati dva broja: 2 i 4. U konvencionalnim programskim jezicima ovo izračunavanje bi se objasnilo na slijedeći način: Naredba sabiranja "+" primjenjuje se na dva cjelobrojna argumenta 2 i 4, i vraća rezultat 6. U OO jeziku rezultat će također biti 6, ali način na koji se do njega dolazi je drugačiji. Poruka "+4" poslata je cjelobrojnom objektu 2. Objekat koji prima poruku (u ovom slučaju 2) samostalno odlučuje kako će se naredba izvršiti, dok u jezicima tipa paskala aritmetički operator "+" komanduje izvršenjem. Komunikacija sa objektima uvijek je dvosmjerna pa se tako rezultat koji je i sam objekat (u našem primjeru cjelobrojni objekat 6), vraća kada se poruka pošalje objektu.

Programiranje u OO jezicima se u krajnjoj liniji svodi na svega dvije operacije:

1. prepoznavanje objekata i poruka u problemu za koji želimo da pišemo program;
2. određivanje odgovarajuće strukture podataka koja podatke interno predstavlja u okviru objekta, uz razvoj algoritama koji odgovaraju na poruku (tj. izvršavaju naredbu).

Drugim riječima, potrebno je prepoznati spoljašnji i unutrašnji vid objekta. Spoljašnji nas obavještava šta dati objekat radi, a unutrašnji precizira kako se to postiže. Ne postoji način da se promijeni stanje objekta osim slanjem poruka. Takva komunikacija je prirodna i preuzeta je iz svakodnevnog života. O tome šta neka druga osoba misli može se suditi samo po porukama koje prima ili odaje, a svaka individua zna svoje unutrašnje stanje i kako reaguje na primljene poruke. Prema većini objekata u realnom životu ponašamo se na isti način – slanjem i primanjem poruka, tako da je objektno orijentirano programiranje prirodan nastavak uobičajenog ljudskog ponašanja. Ustvari, u OO jezicima svaki program je simulacija nekog realnog procesa, i baš zato je programiranje na ovim jezicima najproduktivniji vid programiranja, takoreći – dječija igra. Ovaj termin nije slučajno odabran, jer prvi OO jezik "Smalltalk" nastao je upravo iz želje da se napravi snažan računar za djecu.

5.4 Programski jezici četvrte generacije

Jezici četvrte generacije (Fort-Generation Languages) su neproceduralni viši programski jezici, ili jezici vrlo visokog nivoa. Omogućavaju korisnicima da izvrše kompleksne procedure, koristeći se sa relativno manjim brojem komandi. Kako se manji broj komandi lakše uči, to omogućava da korisnik lakše napiše svoj program. Program pisan u jeziku četvrte generacije razvija se interaktivno. To znači da se naredbe u programu uređuju (logički i jezično testiraju) u toku pisanja programa, dakle prije njegovog završetka.

Upitni jezici (Query Language) četvrte generacije postali su dio sistema za upravljanje bazama podataka. To su također neproceduralni jezici, koje mogu lako učiti i osobe koje nisu programeri.
Upitni jezici omogućavaju krajnjim korisnicima da putem samo nekoliko komandi pristupe sistemu i dobiju informaciju iz njega, a da pri tome ne trebaju stvarati poseban program.
Jedan od takvih jezika je i IBMov **SQL** (Structured Query Language – struktuirani upitni jezik) koji omogućuje korisnicima baze podataka da brzo dobiju željeni izvještaj. Naravno, za kreiranje izvještaja treba poznavati strukturu baze podataka, te definirati oblik izlaznih podataka i uz nekoliko naredbi dobiti željeni izvještaj.

5.5 Programski alati

Osim mogućnosti koje se razvijaju kod upitnih jezika, krajnjim korisnicima dostupni su i tzv. generatori izvještaja, kod kojih je dovoljno utvrditi šta se želi dobiti kao krajnji rezultat, a da se ne definira procedura.

Pored generatora izvještaja, postoje i generatori aplikacija (Application Generators) koji omogućavaju korisniku razvoj čitave aplikacije pomoću jednostavnih naredbi i odgovora na upite.

5.6 Prevodioci i interpreteri

Već je rečeno da računar može da izvršava samo izvršne programe, dobivene prevođenjem izvornih programa na mašinski jezik.

Prema načinu prevođenja instrukcija izvornog programa u mašinski jezik, prevodioci se dijele na:

- nterpretere i
- kompajlere (compiler).

Kod interpretera se u RAM memoriji računara nalazi izvorni kod programa, koji se mora prevoditi svaki put kad se starta izvršavanje programa.

Kompajler prevodi kompletan izvorni kod i snima ga u datoteku koja predstavlja izvršni program. Kod pokretanja programa, u RAM memoriji računara se ne nalazi izvorni kod, nego izvršni program, čije

se mašinske instrukcije jedna po jedna uvode u procesor računara i tu izvršavaju.

6. SOFTVER ZA INŽENJERSKI DIZAJN (KONSTRUIRANJE), INŽENJERING I PROIZVODNJU UZ PODRŠKU RAČUNARA

6.1 Softver za geometrijsko modeliranje – CAD softver

CAD (Computer Aided Design) softver je namijenjen za konstruiranje, odnosno inženjerski dizajn podržan računarom.

6.2 Softver za inženjerske analize – CAE softver

CAE (Computer Aided Engineering) softver omogućava realizaciju različitih inženjerskih analiza, od kojih izdvajamo:

- analize napona i deformacija,
- hidrodinamičke analize strujanja radnog fluida, ili fluida koji opstrujava analiziranu strukturu,
- analize interaktivnog djelovanja fluida i konstrukcije.

Ovdje se ističu programski alati za rješavanje problema iz mehanike kontinuuma, pri čemu se može napisati da je:

CCM = CFD + CSM , odnosno

Computational Continium Mechanics = Computational Fluid Dynamics + Computational Solid Mechanics

6.3 Softver za proizvodnju podržanu računarom – CAM softver

CAM (Computer Aided Manufacturing)