



LETNA ŠOLA

ZBORNIK

1987



Vodnik po razstavi

5
GEMINI
MULTIPROCESORSKI RAČUNALNIŠKI SISTEM

6
TERMINALI
EMULACIJE
INFORMACIJSKA ORODJA
APLIKACIJSKI GENERATORJI
AVTOMATIZACIJA PISARNIŠKEGA
POSLOVANJA

7
**SPECIALI-
ZIRANE
DELOVNE
POSTAJE**
SISTEMI ZA ZAJEM PODATKOV
ŠALTERSKA DELOVNA MESTA
MALI POSLOVNI SISTEMI

8
TRIGLAV
INFORMACIJSKA ORODJA
APLIKACIJSKI GENERATORJI
POSLOVNE APLIKACIJE

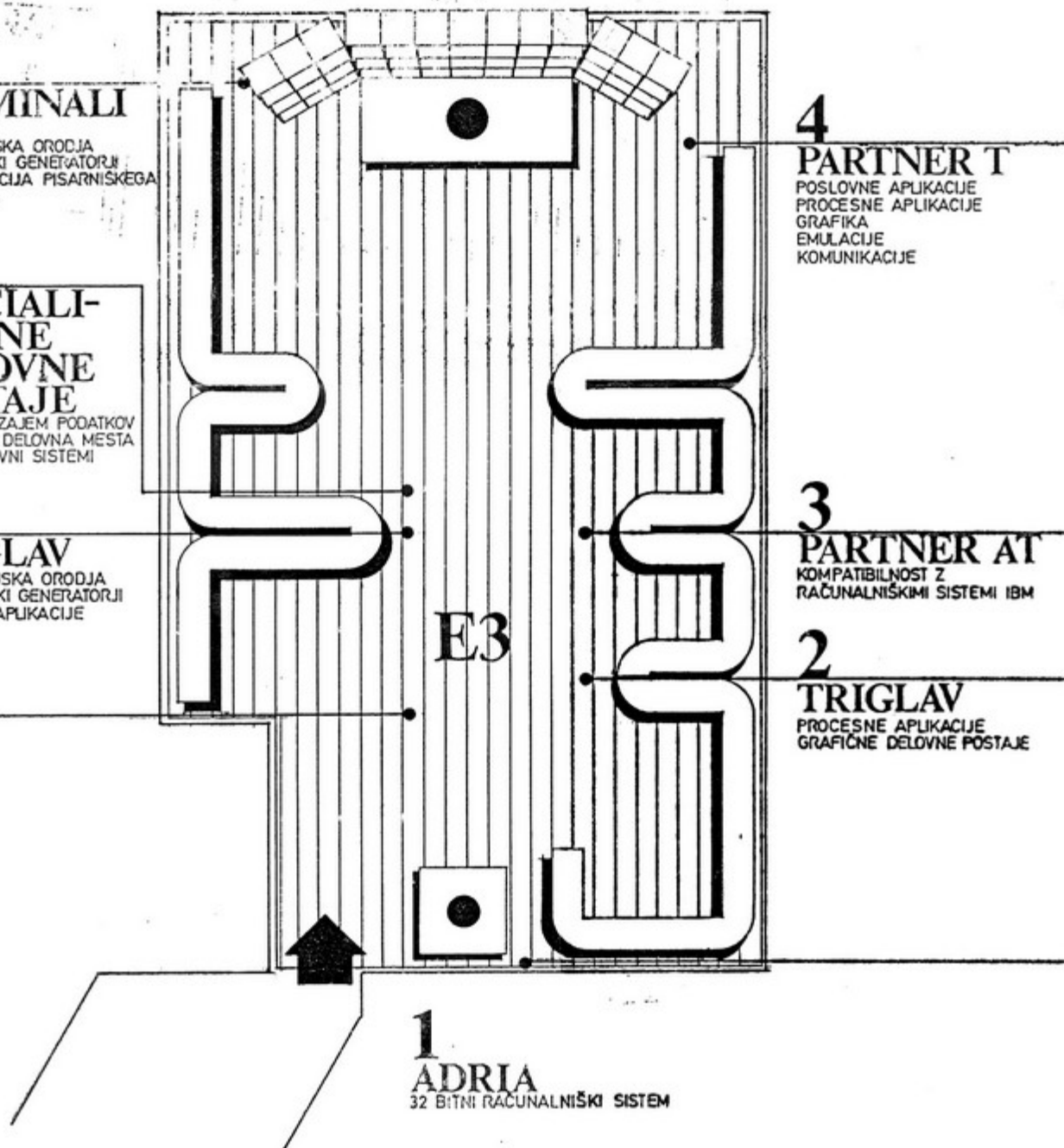
4
PARTNER T
POSLOVNE APLIKACIJE
PROCESNE APLIKACIJE
GRAFIKA
EMULACIJE
KOMUNIKACIJE

3
PARTNER AT
KOMPATIBILNOST Z
RAČUNALNIŠKIMI SISTEMI IBM

2
TRIGLAV
PROCESNE APLIKACIJE
GRAFIČNE DELOVNE POSTAJE

E3

1
ADRIA
32 BITNI RAČUNALNIŠKI SISTEM



LETNA SOLA ISKRA DELTA

Seminar: AIC 04

**PRODOR V VISOKE INFORMACIJSKE
TEHNOLOGIJE**

ALGORITMI
IZSTAVLJENI
V AIC 04

Z B O R N I K R E F E R A T O V



35 BILM IZVODI
ADRIA

bidc
LETNA SOLA

Ljubljana, maj 1987

UVODNA BESEDA

S temo letošnje LETNE SOLE '87 ISKRE DELTE "Prodor v visoke tehnologije informatike" smo želeli opozoriti našo javnost, da imamo v Iskri Delti svojo lastno tehnologijo in da gradimo naše sisteme po lastni ISKRA DELTA ARHITEKTURI, ki je rezultat naše strategije in izvira iz postavljenih globalnih ciljev.

Res je, da se oziramo po vzorih in dogajanjih v svetu, po svetovnih standardih, pa naj so uradni ali tržni, vendar trdimo, da imamo originalno lastne produkte, sad našega dela - vse od razvoja, proizvodnje, do trga. Nimamo licenc za proizvodnjo računalnikov in jih tudi ne rabimo. Na jugoslovanskem in svetovnem trgu kupujemo mnogo komponent, sklopov in enot, ki jih vgrajujemo v naše sisteme. Vendar trdimo: rezultat je naš. Ne proizvajamo ravno vseh gradbenih materialov - vendar hiša je naša. Zato trdimo, čeprav imamo DEC in IBM kompatibilne rešitve, so to ISKRA DELTA rešitve.

Veseli smo, če v nas uporabniki iščejo Iskra Delta tehnologijo, če nam postavljajo naloge in cilje, neprijetno pa nam je, če od nas želijo samo pravni naslov za uvoz te ali one tuje opreme. Zato lahko vsakemu damo izjavo, da Iskra Delta ne proizvaja ne tega ne onega tipa DEC, IBM, Siemens, itd. računalnika, kajti mi proizvajamo Iskra Delta računalnike. Trdimo, da je večina problemov naših uporabnikov rešljivih z domačo opremo, in da lastno tehnologijo v celoti obvladujemo. Drugi tega ne morejo trditi.

S seminarji v sklopu letne sole - Prodor v visoke tehnologije informatike - želimo poudariti, da smo uspeli prodreti v področje mikroročunalnikov in miniračunalnikov, da je v letošnjem letu pred nami preboj v svet supermikroročunalnikov in superminiračunalnikov in napovedujemo: vizija distribuiranih, večprocesorskih in paralelnih računalnikov se v Iskri Delti uresničuje. Verjetno ne bomo prvi na svetu dosegli tega globalnega cilja, vendar dosegli ga bomo pravočasno.

Prodor v visoke tehnologije informatike pomeni ne samo prodor Iskre Delte na področju strojne opreme, temveč tudi prodor na drugih področjih informatike (operacijskih sistemih, podatkovnih strukturah, komunikacijah).

O naši strategiji bi vam radi povedali še več. A kaj ko stari izrek Sun Tsuja pravi:

Vsako vidi taktiko, s katero osvajam, nihče
pa ne vidi strategije, s katero zmagam.

Zato vas raje vabimo na razstavo Letne Sole Iskre Delte '87, kjer bomo tako kot vsako leto prikazali nove proizvode:

Iz palete terminalov PAKA 3090, PAKA 3100, PAKA 5000, PAKA 7800



moramo posebej omeniti PAKO 3090, našo 9" izvedbo monitorja za salterske terminale, za pisarniško poslovanje, itd. in PAKO 5000, s katero sledimo DEC VT-220 standardu.

Tudi družina PARTNER ne gre v odpis, kot si želi naša konkurenca, temveč se še celo pomlaja kot PARTNER/T. Poleg izboljšane grafike in LAN-a je bistveno izboljšana uporabniška prijaznost z novo menujsko tehniko in novimi programskimi paketi. Na voljo sta dve izvedbi, z 9" in 12" monitorjem. PARTNER/T tudi kot inteligentni terminal pridobiva na moči.

Dosedanji emulaciji DEC VT-100 se pridružujejo še DEC VT-220, IBM 3276 BSC in IBM 3276 SDLC, in seveda emulaciji PAKA 3100 in PAKA 5000. Pa se druge so v teku. Kaj več o tem vam bomo povedali v sklopu delovnega seminarja KOMUNIKACIJE.

Nekateri "strokovnjaki" so prisegali, da brez MS-DOS ne morejo rešiti svojih problemov. Čeprav smo bili in se vedno smo prepričani, da tudi CP/M PLUS zadošča, smo sprejeli izziv trga in naredili PARTNER/AT. Upamo, da bodo tudi drugi izpolnili svoje obljube.

V družini TRIGLAV je cel kup novosti, od stolpne izvedbe, laboratorijske, industrijske za vgradnjo in seveda že znane namizne. Nova je distribucija za operacijski sistem OS-9 in izboljšave obstoječih DELTA/M, RSX-11/M, UNIPLUS, XENIX. Pod Xenixom vas čaka presenečenje: spet možnost MS-DOS. O komunikacijah pa - ni problemov - kaj več na seminarju. Vzpostavili smo tudi most med družinama sistemov Triglav in DIPS ter tako povezali procesno informatiko v integralni informacijski sistem.

Novi presenečenji sta še sistema ADRIA in prvi rezultat projekta GEMINI, za katerega se vedno iščemo pravo ime. ADRIA je naš mali DELTA 4850. cenovno in po moči prilagojen zahtevnejši grafiki, CAD/CAM, poslovnemu podsistemu in še kaj. Bistveno pa je: operacijski sistem DELTA/V je na ADRIJI in DELTI 4860 enak.

Projekt GEMINI je delovno ime za realizacijo enega od naših globalnih ciljev: distribuirane informatike. Prvi rezultat je že omenjeni novi računalniški sistem, ki predstavlja integracijo naših hardverskih in softverskih znanj s področja superminiračunalnikov, operacijskih sistemov, IDA informacijskih orodij, komunikacij, itd. Sistem in družina, ki ji pripada še nimata svojega pravega marketinskega imena, predstavljata pa začetek nove generacije računalnikov.

In kaj naj rečemo za zaključek? Morda isto kot lani: želimo, da bi bilo tudi letošnje srečanje vsaj tako uspešno kot pretekla leta, nove ideje, izmenjane izkušnje in sodelovanje pa vzpodbuda za nadaljnje delo. Vse kar vam bomo pokazali in povedali, je naš skupni prispevek k uresničenju strategije tehnološkega razvoja SFR Jugoslavije.

URNIK SEMINARJA

"PRODOR V VISOKE INFORMACIJSKE TEHNOLOGIJE"

Torek, 19. maj

- 9.00 Otvoritev seminarja
- 9.30 Strategija tehnološkega razvoja Jugoslavije in informatika
- 10.30 Pregled pomembnejših računalniških sistemov v Jugoslaviji
- 11.30 Računalniške arhitekture in umetna inteligenca
- 12.30 Gdmor za kosilo
- 15.00 Večprocesorske arhitekture
- 16.00 Paralelno procesiranje
- 17.00 Svetovni standardi na področju računalniških vodil
- 18.00 Coctail v prostorih Cankarjevega doma

Sreda, 20. maj

- 9.00 Nadaljnji razvoj računalniških sistemov Triglav
- 10.00 Unix operacijski sistemi
- 11.00 Računalnik v industrijskem okolju
- 12.00 Oblikovanje OEM tržišča v Jugoslaviji
- 13.00 Odmor za kosilo
- 15.00 Okrogla miza na temo "Strategija tehnološkega razvoja Jugoslavije in informatika"

Vzporedno z okroglo mizo poteka v dvorani E/2 v času od 15.00 do 18.00 "workshop" na temo RACUNALNISKE KOMUNIKACIJE z naslednjim programom:

- 1. Uvod (M. Komunjer, dipl.ing.)
 - 1.1 Proizvodni program Komunikacije v odnosu na ostale proizvode Iskre Delte
 - 1.2 Strategija kratkoročnega razvoja
- 2. Inteligentni terminal (mag. D. Perhoč, dipl.ing.)
 - 2.1 Asinhrona povezave
 - 2.2 Sinhrona povezave
 - 2.3 Kompleksne delovne postaje
- 3. Računalniške mreže (B. Sostarič, dipl.ing.)
 - 3.1 Lokalne mreže
 - 3.2 Privatne mreže
 - 3.3 Javne mreže
- 4. Najava novih proizvodov (mag. S. Seljan, dipl.ing)
- 5. Diskusija

Cetrtek, 21. maj

- 9.00 Večprocesorski računalniški sistemi IDC
- 10.00 Razvoj informacijskih orodij IDA
- 11.00 Problemi računalniških jezikov in razvoja aplikacij
- 12.00 Novi pristopi IDC do uporabnikov
- 13.00 Odmor za kosilo

Od 15.00 do 18.00 poteka v dvorani E/2 "workshop" na temo TOVARNA PRIHODNOSTI z naslednjim programom:

1. Uvod: Procesi v proizvodnih organizacijah
2. Nove tehnologije v proizvodnih procesih:
 - 2.1 Robotizirane delovne linije
 - 2.2 Računalniški inženiring v strojništvu
 - 2.3 Vloga orodij za avtomatizacijo pisarn v proizvodnih organizacijah
 - 2.4 Sistemi za podporo odločanju in upravljanju
3. Prototipni pristop k računalniški podpori: Prikaz primera z AGP
4. Distribuirana obdelava podatkov: Kritična analiza
5. Končna diskusija



**STRATEGIJA TEHNOLOSKEGA
RAZVOJA JUGOSLAVIJE
IN INFORMATIKA**

Avtor referata: mag. Vanja Bufon, dipl.ing.



Ljubljana, maj 1987



1. UVOD

Casovno obdobje oblikovanja in spremljanja strategije tehnološkega razvoja Jugoslavije je nedvomno primeren čas za strateško razmišljanje in oblikovanje lastne razvojne strategije. Gre za obojestranski proces, saj s svojim razmišljanjem in stališči želimo sooblikovati družbene procese, hkrati pa iščemo družbeno potrditev lastne prehojene poti in oceno usklajenosti lastnih strateških ciljev z družbenimi.

Malo tehnologij pa ima tako izrazit infrastrukturni karakter kot informacijska tehnologija. Te proizvode ne oblikuje samo proizvajalec temveč predvsem končni uporabnik, ki informacijsko tehnologijo vgradi v lastne tehnološke rešitve. Poudarjamo, da vez med proizvajalcem in uporabnikom ne obstaja samo v času življenske dobe proizvoda ampak meji skoraj na rodbinsko razmerje. Strateška zasnova informacijskega sistema že odloča njegovo bodočo rast in dograjevanje. Zamenjava proizvajalca in/ali slaba usklajenost strategij obeh partnerjev pomeni resne motnje in dodatne težave pri njegovi gradnji. Neopredeljenost in nedoslednost, ki se kaže v odsotnosti, odstopanju ali neprestanem menjavanju strategije (praktično v tem primeru strategija ne obstaja), pa vodi v katastrofo. Prav zaradi te soodvisnosti mislimo v Iskri Delti, da so nam poleg lastnih sodelavcev najbližji naši uporabniki. To je naša družina, to je okolje, v katerem želimo dograjevati našo lastno strategijo, se vključevati v jugoslovansko informatiko in v skladu s strategijo tehnološkega razvoja Jugoslavije v enakopravno mednarodno delitev dela.

2. NAŠI POGLEDI NA STRATEGIJO TEHNOLOŠKEGA RAZVOJA JUGOSLAVIJE

2.1 OSNOVNA IZHODISCA

Strategija tehnološkega razvoja Jugoslavije se se zlasti v segmentu, ki obravnava računalništvo in informatiko, vse preveč podrobno ukvarja s posameznimi konkretnimi tehnikami in rešitvami, medtem ko pušča nekatere bistvene cilje nedorečene in ne razrešuje dokončno nekaterih bistvenih dilem:

- * kaj so globalni cilji in kakšna je vloga informatike znotraj strategije tehnološkega razvoja Jugoslavije,
- * kakšni so kriteriji, s katerimi bomo merili in vrednotili realizacijo postavljene strategije,
- * po katerih kriterijih se bodo kvalificirali nosilci posameznih nalog in po katerih kriterijih bomo opredeljevali odgovornosti ter možnosti ukrepanja,
- * kakšno bo okolje za realizacijo strategije.



2.2 ODRAZ TRGA IN PROIZVODNJE V ZNANSTVENO RAZISKOVALNEM DELU

Vsebina strategije tehnološkega razvoja Jugoslavije je prevec zreducirana na razciscevanje in ukrepe znotraj raziskovalnega podrocja oziroma na ukrepe, ki vzpodbujajo razvoj tega podrocja in izboljšujejo predvsem položaj raziskovalnih organizacij. V marsicem je prevec podrobna. Nedorečena pa ostaja v opredelitvi in definiciji inovacijskega procesa, ki sega globlje vse od ideje preko proizvoda do trga. Kritični so spoji razvoj/proizvodnja in proizvodnja/trg. Definicija razvoja je preozka in jo moramo razširiti tako, da bo poleg osnovnih raziskav in tržno usmerjenega razvoja v ožjem smislu, določevala tudi inovacije proizvodnega procesa kot pomembno razvojno funkcijo. Izredno kratki časi za realizacijo ideje na trgu zahtevajo ne samo nove tehnologije temvec tudi novo organizacijo /3,7,8,9/.

Kritična je ideja o skladu za pospeševanje tehnološkega razvoja, ki ni dovolj precizirana in obstaja bojazen, da se bo izrodila v svoje nasprotje:

- * v sklad za privilegirane organizacije
- * ali v sklad za razvojne institucije brez odgovornosti do realizacije na trgu.

Potrebno je aktivirati vse razpoložljive razvojno-raziskovalne kapacitete od univerz, institutov do raziskovalnih enot v organizacijah združenega dela za strateske naloge, vendar pod pogojem, da je razvoj končan sele po realizaciji na trgu. Dosedanja praksa na področju informatike kaže, da se marsikatera razvojno-raziskovalna institucija ponaša s prototipi in rešitvami, ki pa nikoli ne pridejo na trg. Odgovor, da so za to odgovorne proizvodne delovne organizacije, je samo izgovor in beg pred odgovornostjo, kajti pravilno zastavljen razvoj je pogojen s trgom in s tehnologijo bodoče proizvodnje. Učinkovitost znanstveno-raziskovalnega dela bi morali meriti s trgom, ki tudi regenerira sredstva za nove raziskave.

V celoti je premalo poudarjena vloga trznega gospodarstva. Samo v pogojih skupnega gospodarskega prostora in doslednega uveljavljanja ekonomske prisile bo namreč enoten jugoslovanski trg, odprt tudi navzven, deloval spodbujevalno in obenem objektivno pri uveljavljanju novih tehnoloških rešitev /3/.

Država mora sistemsko motivirati in vzpodbujati organizacije združenega dela za hitrejši tehnološki razvoj ter oblikovati osnove za enotne tehnološke rešitve v velikih infrastrukturnih sistemih. Vzpodbuditi je potrebno razvoj znanosti in tehnologije, njeno racionalnejšo organizacijo in ustvarjanje kritičnih velikosti razvojnih potencialov. Ti ukrepi morajo biti usklajeni z ostalo zakonodajo in morajo veljati se zlasti glede enotnega gospodarskega prostora, uveljavljanja trznih zakonitosti ter ekonomske logike za tisto raziskovalno dejavnost, ki je neposredno namenjena razvoju tehnologij, izdelkov in infrastrukturnih sistemov. Poudariti zelimo, da si vsak



gospodarski subjekt, tudi informatika ni izjema, želi poslovati na osnovi tržnega gospodarstva. Dejstvo pa je, da danes takšen trg ne eksistira. Nepremišljeni hitri gospodarski ukrepi spreminjajo tržne pogoje iz enega ekstrema v drug in s tem razvrednotijo strateško ukrepanje organizacij združenega dela ter ga preusmerjajo v iskanje trenutnih tržnih možnosti. To je bližje strategiji spekulacij kot strategiji zasledovanja globalnih ciljev. Tako sta se se zlasti v zadnjem obdobju razmahnili "jugoslovanska proizvodnja" osebnih in mikroročunalnikov ter uvoz rabljenih računalnikov. Uvozu botrujejo bodisi trgovske organizacije, bodisi razne oblike privatne iniciative od posameznikov do raznih zadrug, z enim samim kratkoročnim ciljem - želje za profitom. Običajno ne rešujejo niti vzdrževanja prodanih izdelkov, da o kakšni dolgoročni strategiji sploh ne govorimo. Zato jih običajno tudi ne zanima sodelovanje z domačimi proizvajalci informatike niti prodaja domačih izdelkov.

2.3 KADRI

Pri strategiji tehnološkega razvoja Jugoslavije imajo ključni pomen kadri. Bolj ali manj je razumljivo, da potrebujemo strokovno usposobljene strokovnjake v skladu z mednarodnimi merili. S primerno kadrovsko politiko in solanjem tudi v tujini je moč doseči željeno raven in primeren raziskovalni potencial /12/. Padamo pa na osveščenosti kadrov; na motiviranosti in zaupanju v lastno znanje in v sposobnost, da smo zmožni izpeljati zastavljene cilje. Navduševanje nad vsem kar je tuje, je rezultat psihološke vojne, ki dobro obrodi v ekonomsko problematičnem okolju. Se zlasti če dopustimo sirjenje nerealnih marketinških informacij, ki povečujejo proizvode zahodne tehnologije, domače uspehe pa ovrednotijo po principu že videno. Pojavlja se fatalizem, neučinkovitost, kritizerstvo in odpor, pa tudi odhajanje v tujino. Temu pogosto sledi vračanje v vlogi tujega prodajalca in svetovalca, kot akterja nadaljnje psihološke vojne. Na področju visokih tehnologij je ta indokrinacija že dosegla kritično točko in so pod njenim vplivom tudi nekateri univerzitetni kadri.

Tujcu vse preveč zaupamo le na besedo, da bo proizvod naredil, domačim pa ne verjamemo, niti ko proizvod vidimo, in tako se revež dokazuje, dokler proizvod tehnološko res ne zastara.

Naša pogosta napaka je tudi priseganje, da bi problem rešili, če bi imeli najnovejšo tehnologijo. Običajno pa so te "modne muhe informatike" samo skrivanje nesposobnosti, kajti dimenzija našega problema takšne tehnologije sploh ne zahteva. Pa tudi če bi bila naša informatika in njeni proizvodi na nivoju sedemdesetih let, bi si človek želel, da bi naša današnja stopnja gospodarstva ustrezala visokorazvitim v tistem času. V sklopu svojih predavanj o japonski tehnologiji je dr. Shigeo Shinga jasno povedal, da pretirana in prehitra robotizacija brez predhodne faze inoviranja proizvodnega procesa vodi v ekonomsko katastrofo. Kot rešitev je na prvo mesto postavil delavca-inovatorja, ki preko poznavanja proizvodnega procesa s postopno vedno višjo stopnjo mehanizacije in avtomatizacije gradi komercialno robotiko /13/.



Zato mislimo, da je ključna naloga strategije tehnološkega razvoja Jugoslavije zgraditi osveščenega delavca-inovatorja.

2.4 MEDNARODNI TRG IN VPLIV TUJIH STRATEGIJ

Strategija tehnološkega razvoja Jugoslavije obravnava primarno jugoslovanski prostor, potrebno pa bo poseči tudi v mednarodne odnose. Precizirati je treba pogoje in načine vključevanja v mednarodno delitev dela, mednarodni tehnološki transfer, skupne naložbe v tehnološko intenzivne produkcije ter vključevanje organizacij združenega dela v mednarodne znanstvene in tehnološke programe. Pogrešamo tudi vgraditev obrambnih mehanizmov proti negativnim učinkom strategij drugih držav in transnacionalk /10,11/. V svetovnem merilu se odvija med obema političnima blokoma ekonomska in tehnološka vojna kot posebna oblika vojne. Vendar so trenja tudi znotraj držav istega bloka. Poučen primer je trgovinski spor visokorazvitih držav ZDA in Japonske na področju informatike, ki mu je sledila uvedba zaščite ameriškega trga. ZDA so identificirale japonsko strategijo sele po njenih učinkih na trgu. Neodgovorjeno ostaja vprašanje, v kolikšni meri bodo ukrepi ZDA učinkoviti.

Negativni gospodarski trendi v Jugoslaviji, neusklajenost, včasih celo za tuje firme ugodnejši predpisi, pogojevanje mednarodnih pogodb in ne nazadnje indoktriniranje naših kadrov s tujo tehnologijo kaže, da smo pod stalnim pritiskom tujih strategij. Zal moramo ugotoviti, da ekonomski in tehnološki vojni ter drugim pritiskom iz tujine nismo kos. Prav zato morajo vsebovati strategija tehnološkega razvoja Jugoslavije in drugi dokumenti mere, ki bodo obvladovale okolje za realizacijo naše strategije.

Ker so mere usmerjene proti tujim strategijam, bodo sprožale protinapade z zahtevami po korekturah. Nikakor pa to ne bodo direktni napadi temveč poizkusi ustvarjanja "javnega mnenja" in psihoz. V našo klasiko sodi metoda ustvarjanja psihoz o monopolnem položaju neke jugo-grupacije na jugoslovanskem trgu, ki ji sledi šok terapija zdravljenja z interventnimi zakoni in pogosto interventnimi uvozi, ki pa samo slabijo položaj združenega dela in uničujejo že tako nizko akumulacijo. Če se bodo ti trendi nadaljevali, bomo kmalu imeli samo se en monopol - monopol nad krediti.

2.5 KRITERIJI

Ne borimo se za privilegiran položaj, temveč za načine in kriterije, preko katerih bo družba motivirala organizacije združenega dela za hitrejši tehnološki razvoj ter selekcionirala soustvarjalce strategije za enotne tehnološke rešitve. Ali kratko: kot se delavec kvalificira z rezultati dela, naj se tudi organizacija združenega dela kvalificira z rezultati realizacije strategije tehnološkega razvoja Jugoslavije.



Uveljavitev tržno orientiranega gospodarstva sama po sebi se ni dovolj, kajti vzpostavitev trga mora biti posledica naše in ne neke tuje strategije. Zato so poleg opredelitve globalnih ciljev najpomembnejši del strategije tehnološkega razvoja Jugoslavije kriteriji. Predloženi so kriteriji:

- * kakovosti tehnologije,
- * aspiracije (osvojenost in osvojljivost) in
- * kriteriji cilja uporabe tehnologije.

V razpravah se je izoblikovalo stališče, da je te kriterije potrebno dopolniti še s kriteriji produktivnosti, konkurenčnosti, čistoče, energetske varčnosti, zaposlitvene primernosti, surovinske neodvisnosti ter obrambe in samozaščite.

Podrobneje nima smisla razčlenjevati posameznih kriterijev, saj obstoja o tem več dokumentov in razprav.

Pomembno je, da ti kriteriji veljajo brez izjeme na celotnem področju Jugoslavije. Vsako odstopanje pomeni negiranje in razpad koncepta selekcije /1,2,3/.

3. INFORMATIKA IN ISKRA DELTA

Naše gospodarstvo se je začelo srečevati z računalništvom in informatiko v šestdesetih letih, intenzivneje pa v začetku sedemdesetih let preko zunanjetrgovinskih podjetij, ki so zastopala praktično vse pomembnejše tuje proizvajalce. V tem obdobju segajo korenine Iskre Delte v dva segmenta: zastopstvo CDC preko Iskre Commerca in zastopstvo DEC preko Elektrotehne.

V tem času so se na različnih mestih v Sloveniji in Jugoslaviji začeli tudi poizkusi postopnega osvajanja računalniške proizvodnje. Večina razvojno-proizvodnih poizkusov je bila in ostala pod direktnim vplivom tujih transnacionalk, preko zastopniških podjetij, vezanih z licenčnimi in kooperacijskimi pogodbami. Iskra Delta je svojo razvojno pot oprla na lastne noge s ciljem, da čim več izdelava sama, ostalo pa kupi na jugoslovanskem ali na mednarodnem trgu in integrira v svojo ponudbo. Vsa leta do danes je bila zvesta globalnim ciljem:

- * osamosvojiti se in zmanjšati enostransko odvisnost od DEC ter organizirati lastni razvojno-proizvodni program z vsemi oblikami podpore uporabnika, s čim nižjo uvozno komponento;
- * s ključnimi proizvodi prodreti na svetovni trg in uravnotežiti devizno plačilno bilanco;
- * postaviti takšna izhodišča razvojno-proizvodnega programa, ki bodo omogočila postopno tehnološko integracijo jugoslovanskih proizvajalcev računalništva in informatike.



Od svojega nastanka leta 1978, preko obdobja zakona o obveznem združevanju računalniških proizvodnih in razvojnih kapacitet v SR Sloveniji do danes, je uspela realizirati razvoj in proizvodnjo več družin proizvodov: terminale Paka, mikroračunalnike Partner in Triglav, miniračunalnike Delta 800, Delta 4850, IDA informacijska orodja, IDA komunikacije, projekt procesnih modulov DIPS, številne aplikativne rešitve za različne gospodarske panoge itd. Gre za rastoče žive družine proizvodov. V letu 1987 se jim priključujejo novi proizvodi kot PARTNER/AT, TRIGLAV XEN-32, TRIGLAV UNX-32, ADRIA, GEMINI, PAKA 5000 in drugi. V obdobju 1987-91 gremo razvojno gledano v prodor na področje superminiračunalnikov, paralelnih računalnikov, distribuiranih sistemov itd., paralelno pa nadaljujemo prodore na tuje trge zahodne Evrope, SEV-a ter neuvršenih dežel.

V zadnjem obdobju smo zgradili zelo močne izobraževalne in servisne zmogljivosti Iskre Delte na področju cele Jugoslavije. Glavni servisni center v Ljubljani vključuje tudi center za obnovo sistemov in popravilo modulov. Glavni izobraževalni center je v Novi Gorici. Zgrajena je tudi servisno prodajna ter aplikacijsko softwarska razvojna struktura v vseh važnejših centrih republik in pokrajin. Iskra Delta je zasledovala tudi policentrični razvoj po republikah. Tako ima na primer enote v Mariboru, Novem mestu, Novi Gorici, Kranju, Titovem Velenju, Ptujju, Celju, Zagrebu, Rijeki, Splitu, Beogradu, Novem Sadu, Skopju, Titogradu, Sarajevu, Slovenj Gradcu, Kopru, Bledu, Celju, Pristini, Nisu in Banja Luki.

Hitra rast, številni čedalje bolj zahtevni proizvodi, predvsem pa kvaliteta izdelka, zahtevajo tudi modernizacijo proizvodnje. Tako je v gradnji nova tovarna na lokaciji Stegne - Ljubljana, ki bo zagotovila pogoje dela za racionalen kompleksen razvoj, proizvodnjo in servis. S tem bodo dani vsi pogoji za pravo industrijo informatike. Seveda predstavlja ta investicija precejšnje breme, vendar vemo, da bomo uspeli narediti tudi ta korak iz maloserijske v serijsko proizvodnjo. Pri tem računamo tudi na pomoč naših uporabnikov.

V oktobru 1986 je prišlo do pobude SOZD Gorenja, da se njihove temeljne organizacije iz Titovega Velenja in Ptujja, ki delajo na področju računalništva, vključijo v DO Iskra Delta. S tem so bile združene vse realne proizvodne in razvojne kapacitete iz te dejavnosti iz SR Slovenije v Iskri Delti, zaposlenost pa je dosegla 2100 delavcev, od česar več kot polovica z visoko in višjo izobrazbo. Iskra Delta je ves čas veliko vlagala v razvoj novih kadrov, tako že zaposlenih v DO, kot tudi novih na fakultetah in srednjih solah. V tem trenutku stipendiramo v Jugoslaviji preko 370 dijakov in studentov za potrebe združenega računalništva.



Kot pomemben dosežek, ki je gotovo izjemen v siceršnji praksi jugoslovanskega združenega dela, velja omeniti, da se je zaradi lastnih vsakoletnih razvojnih dosežkov in z osvajanjem proizvodnje posameznih sklopov, sistemov in softwera, vsakoletnemu več kot 100% fizičnem povečanju realizacije in proizvodnje, uvozna odvisnost Iskre Delte zmanjševala.

Koncem leta 1985 je Iskra Delta začela pripravljati plasman svojih proizvodov v izvoz. V ta namen je oddvojila zelo pomembne kadrovske in materialne resurse in uspela realizirati postavitev lastne proizvodne firme IDC v Avstriji ter prodajno-tehničnih centrov v Frankfurtu in New Yorku. S tem se je tudi začel realen izvoz računalnikov v zahodno Evropo in ZDA. Izvozni projekt, ki se odvija v Avstriji, je bil podprt s strani avstrijskih koroskih deželnih in zveznih oblasti. Ta vrsta financiranja poteka brez garancij jugoslovanskih bank.

Dosedanji tržni rezultati in raziskave trga nam dajejo osnovo za oceno, da je možno v naslednjih petih letih realizirati izvoz na Zahod in v tretje dežele v obsegu 100 mio \$.

Stanje, ko se je večina ostalih jugoslovanskih potencialnih proizvajalcev v svoji strategiji odločila za naslonitev na eno od svetovnih firm, je onemogočilo resnično razvojno proizvodno povezovanje v jugoslovanskem prostoru, ker so to preprečevale in preprečujejo tuje firme, na katere so naslonjeni jugoslovanski partnerji in dejstvo, da jugoslovanske DO ne posedujejo lastne tehnologije in proizvodov. Akumulacija, ki jo je konjunktorni računalniški trg omogočal, se ni usmerjala v krepitev lastnih sposobnosti za razvoj računalnikov. Z zveznimi zakoni uvedena administrativna kontrola uvoza računalnikov ni bila podprta z jasno strategijo, kar je povzročilo spopad za delitev trga preko komisije za odobravanje uvoza računalnikov.

Za doseganje strateških ciljev je bila Iskra Delti obljubljena ustrezna finančna in devizna pomoč, ki ni bila realizirana. Tako je morala sama, brez družbene pomoči, realizirati zahtevno poenotenje heterogenega razvoja in proizvodnje računalniške tehnologije ter ustrezno prestrukturirati kadre. To ji je uspelo z velikimi naporji vseh zaposlenih, z usmerjanjem akumulacije v lastno tehnologijo ter z zaupanjem uporabnikov, ki vidijo v programu Iskre Delte dolgoročno strateško možnost opremljanja ter izgradnje svojih informacijskih sistemov.

Iskra Delta je v svojem razvojnem konceptu in naporih stalno zasledovala cilj, da izdela in razvije vse potrebne elektronske in softwarske elemente ter kompletne rešitve za potrebe trga. Razvoj ni bil samemu sebi namen, temveč smo se vseskozi zavedali pomembnosti infrastrukturnega značaja naših proizvodov. V zaostreni gospodarski situaciji se vse bolj kaže, da smo imeli prav in da jugoslovansko gospodarstvo brez informatizacije in avtomatizacije ne bo imelo možnosti za kolikor toliko enakopravno vključevanje v mednarodno delitev dela. Zavedali smo se, da je informatika vsaj toliko pomembna kot energetika, kar pa širša družba do danes še ni spoznala. Smatramo, da podobno kot je na



prelomu iz prejšnjega stoletja v zdajšnje prinesla revolucionarne spremembe elektrika, take in še večje prinaša s seboj, ob prelomu v naslednje stoletje, informatika.

Prepričani smo, da bodo gospodarski subjekti v Jugoslaviji spoznali pomembnost domače računalniške infrastrukture, če se bodo hoteli v proizvodnji novih izdelkov in v ekonomičnejši proizvodnji obstoječih pojavljati na zunanjih tržišcih.

Tudi miselnost, da se lahko brez razvojno proizvodne tržne tradicije in izkušenj, ki so si jih nabrali v desetih letih delavci Iskre Delte, začne z nekim najsodobnejšim razvojem, kot ga predvideva strategija tehnološkega razvoja Jugoslavije za področje računalništva, ni realna. Kajti le na podlagi izkušenj, dosežene stopnje organiziranosti in različnih tehnoloških znanj je možno načrtovati nadaljnjo izgradnjo še bolj zahtevnih generacij računalniških sistemov in softwarea.

Iskra Delta se je uspela razvojno in proizvodno osamosvojiti od vpliva ameriških in drugih transnacionalnih firm, ustvariti lastne perspektive za naprej in je začela vzpostavljati tudi druge kvalitetne odnose v povezovanju s tujino. Do danes se zaradi naše tehnološke podrejenosti in premalo znanja tuje firme niso pogovarjale z nami na enakopravni osnovi. Iskra Delta pa je s svojim konceptom, tehnologijo in akumuliranim znanjem dosegla, da se te firme, ki razpolagajo z visoko tehnologijo, začenjajo z nami pogovarjati na enakopravni bazi in to predvsem, da nas priznavajo za partnerje v izmenjavi znanje za znanje - to je, da naš izdelek vključujejo v svoj originalni izdelek, ter da mi njihov originalni izdelek vključimo v svoj končni originalni izdelek. Tako je Iskra Delta že začela razvoj paralelnega računalniškega sistema, ki vključuje umetno inteligenco in govorni vmesnik (interface), kar predstavlja naslednjo generacijo računalnikov in kjer so nas kot partnerje v izmenjavi znanja za znanje že priznali ustrezni instituti in univerze iz ZDA in Japonske.

Iskra Delta je letos soočena tudi z različnimi neugodnimi sistemskimi in družbenimi razmerami, kot na primer:

- * v februarju je bil ukinjen slovenski zakon, ki je določal razvoj računalništva kot dejavnost posebnega družbenega pomena;
- * prav tako je bil v mesecu februarju sprejet zakon o prepovedi razpolaganja z družbenimi sredstvi za negospodarske in neproizvodne investicije, ki določa nakup računalnikov (domačih) kot negospodarsko oz. neproizvodno investicijo. Iskra Delta je v tej zvezi že sprožila postopek za spremembo zakona;
- * veliki uporabniki računalnikov v Jugoslaviji, ki kupujejo tuje računalnike s koriščenjem mednarodnih kreditov, najetih preko mednarodnih finančnih institucij, niso omejeni pri zelo spretno izoblikovanih nakupih in najemanju računalnikov iz tuje proizvodnje;



- * kljub strogim omejitvam deviznega odliva in uvoza, kar ima za posledico kontingentiranje uvoza repromateriala za potrebe domače računalniške industrije, se je razvil obsežen uvoz rabljenih in zastarelih računalnikov;
- * pod vplivom več kot desetletne aktivnosti transnacionalnih in njihovih zastopnikov se je vcepjalo prepričanje, da mali in zaostali narodi niso sposobni lastnega razvoja na področju računalništva. Psihološki kompleks inferiornosti je zato problem, s katerim smo soočeni pri različnih strukturah družbenega odločanja;
- * povečana aktivnost zastopniških firm novači kreativne strokovnjake direktno za delo s tujimi transnacionalnimi firmami ali v obliki malih firm in zasebnega dela, ki ne nosijo bremen systemskega in aplikativnega razvoja;

Le z usklajenimi stališči in realizacijo strategije tehnološkega razvoja Jugoslavije bomo lahko uspešno nadalje razvijali tako pomemben infrastrukturni element kot je računalništvo in to na področju razvoja, proizvodnje, trženja, servisiranja in šolanja. V nadaljnjem razvoju Iskre Delte želimo čim kreativnejše vključevanje razvojnih inštitutov univerz in malih fleksibilnih firm in zasebnikov v funkcijo domačih tehnoloških, zlasti proizvodnih in tržnih potreb, optimalno vključevanje tujih tehnoloških dosežkov kot tudi komplementiranje domače proizvodnje in tržne ponudbe ter spremljanje svetovnih tehnoloških tokov. S tem pa bi bilo prav tako omogočeno hitrejše, samostojnejše informatiziranje gospodarstva in celotne družbe, zmanjševanje tehnološkega zaostajanja za razvitimi deželami, enakopravnejše vključevanje v kvalitetno mednarodno menjavo znanja in dolgoročno utrjevanje konkurenčnega položaja našega izvoza.

4. ZAKLJUČEK

O drugih jugoslovanskih znanstveno-raziskovalnih, proizvodnih in tržnih organizacijah ne bomo govorili. Raje smo in bomo predstavili nove dosežke Iskre Delte in vsaj delno odkrili tancico, kaj nameravamo razvijati v prihodnosti.

Javnosti pa prepuščamo sodbo, da bodisi s svojimi merili, bodisi z merili strategije tehnološkega razvoja Jugoslavije izmeri, ovrednoti in rangira jugoslovanske dosežke na področju informatike. Strokovni kadri, ki delajo na področju informatike, pa naj se opredelijo ali potrebujemo domačo računalniško industrijo ali pa samo uporabo uvoženih računalnikov. Kajti čas ekonomsko-tehnološke vojne to opredelitev zahteva. Opredelitvi pa mora slediti akcija.

5. LITERATURA

- /1/ Strategija tehnološkega razvoja Socialistične federativne republike Jugoslavije in Predlog programa ukrepov za podpiranje razvoja znanosti in tehnologije - 1 del, Poročevalec Skupščine SR Slovenije in Skupščine SFR Jugoslavije za delegacije in delegate, letnik XIII, št. 3, Ljubljana, 27.01.1987.
- /2/ Strategija tehnološkega razvoja Jugoslavije (predlog), Savez inženjera i tehničara Jugoslavije, Beograd, januar 1987.
- /3/ Materiali 33. seje Republiškega družbenega sveta za gospodarski razvoj in ekonomsko politiko, Ljubljana, 23.2.1987.
- /4/ Dosedanji razvoj domače industrije računalništva in predlogi ukrepov za uresničevanje strateških ciljev na področju računalniške in informacijske tehnologije, Iskra Delta, DS, maj 1987.
- /5/ Erik Vrenko, Poteze, ki so kot kamen okoli vratu (intervju), Delo, Ljubljana, 31.3.1987.
- /6/ Erik Vrenko, Smiselnost naložb v znanje, Delo, Ljubljana, 14.4.1987.
- /7/ Erik Vrenko, O naši zamisli inovacijske družbe in o ovirah na poti do nje (intervju), revija RR 3, Ljubljana, december 1985.
- /8/ Andrej Skarabot, Integralni sistem razvoja in osvajanja izdelkov, Revija RR 2, Ljubljana, 1986/8.
- /9/ Naš razvoj in znanstvenotehnološka revolucija, Teorija in praksa, Ljubljana, 1-3, 1986, LET XXIII.
- /10/ IBM's Strategic Thinking, Strategic Incorporated, Kalifornija, avgust 1985.
- /11/ International Seminar on the Management of Technology, International Management Institute, Zeneva/Svica, april 1986
- /12/ dr. Predrag Radivojević, Naučno-tehnološki potenciali Jugoslavije, Privredni pregled, Beograd 1986.
- /13/ dr. Shigeo Shing, Proizvodnja brez napak, seminar "Kako povečati produktivnost", Ljubljana, april 1987.



**PREGLED POMEMBNEJSIH
RACUNALNISKIH SISTEMOV
V JUGOSLAVIJI**

Avtor referata: Tit Turensek, dipl. ing., ISKRA DELTA



Ljubljana, maj 1987



1. UVOD

V referatu je podan kratek pregled stanja in trendov računalniške opreme. Zavedamo se, da je v omejenem obsegu nemogoče podati izčrpen pregled celotnega tržišča in obravnavati vse njegove segmente in aspekte. Vsled tega smo se omejili le na tiste, ki so zanimivi s stališča stanja računalništva v Jugoslaviji in ki vplivajo na dogajanja na našem trgu.

2. KATEGORIZIRANJE IN DEFINICIJE

Da bi lažje med seboj primerjali računalniške sisteme in ugotavljali tržne trende na področju informacijskih tehnologij, je smotno računalniške sisteme klasificirati oziroma kategorizirati v karakteristične grupe produktov (= računalniških sistemov).

Vsesplošno sta se uveljavila dva principa kategorizacije računalniških sistemov:

- 1 - Kategorizacija na osnovi strojnih, performančnih in cenovnih karakteristik
- 2 - Kategorizacija na osnovi aplikativnega področja, načina uporabe sistemov in nivoja organiziranosti

Po prvi kategorizaciji dobimo sledečo taxonomijo računalniških sistemov:

- * LARGE MAIN FRAME
- * SUPER MINI
- * MINI
- * MULTI USER MICRO / SUPER MICRO
- * SINGLE USER MICRO / PC
- * (HOME COMPUTER)

Glede na nivo organiziranosti bi lahko razvrstili računalniške sisteme v sledeče kategorije:

- * INDIVIDUALNA DELOVNA POSTAJA
(PERSONAL WORK STATION)
- * VECUPORABNIŠKA DELOVNA POSTAJA
(TEAM WORK STATION)
- * SISTEMI NA NIVOJU ORGANIZACIJSKE ENOTE
(DEPARTMENT COMPUTER)
- * SISTEMI NA NIVOJU VEČJE ORGANIZACIJE
(CORPORATE COMPUTER)



Kategorizacija je slična IBM-ovi 4 nivojski hierarhiji, pri čemer sta poudarjena predvsem prvi in zadnji nivo.

Mislim, da takšna razvrstitev z današnjega stanja in trendov razvoja arhitekture informacijskih sistemov ni več upravičena, saj sugerira hierarhično strukturo sistemov, pri čemer naj bi bili na spodnjem koncu osebni računalniki in na zgornjem veliki centralni računalniki (large mainframe).

Današnji HW in sistemski SW vse bolj omogočata realizacijo maxime:

"Računalniška oprema (HW, SW) na posameznih lokacijah naj bo prilagojena obsegu podatkov in obdelav, ki podpirajo funkcije na dotični lokaciji. Sistemi na različnih lokacijah si izmenjujejo podatke, kadar je to potrebno in v obsegu, kot je to potrebno." Nivo upravljanja večje organizacije (nivo SOZD-a, korporacije) običajno potrebuje le manjši obseg agregiranih podatkov.

Večkrat srečujemo kategorizacijo glede na aplikativno področje:

- * SISTEMI ZA AVTOMATIZACIJO PISARNISKEGA POSLOVANJA
(OFFICE AUTOMATION SYSTEMS)
- * MAJHNI POSLOVNI SISTEMI
(SMALL BUSINESS COMPUTERS)
- * DELOVNE POSTAJE ZA RAZLIČNE POTREBE
(CAD / CAM WORKSTATION, I.T.D.)
- * SISTEMI ZA VODENJE PROIZVODNJE
(FACTORY COMPUTERS)
- * ZNANSTVENO - INŽENIRSKI SISTEMI
(SCIENTIFIC ENGINEERING SYSTEMS)
I.T.D.

Analiza posameznih aplikativnih področij bi morala poleg strojne opreme nujno zajeti tudi aplikativne programske rešitve, pokazala pa bi zelo zanimive trende.

Kot ilustracijo vzemimo sisteme za pisarniško poslovanje (OFFICE SYSTEMS). Na osnovi naslednjih treh tabel, ki odražajo stanje v tem segmentu trga v ZDA, je lahko razbrati:

- * da prevladuje uporaba miniračunalnikov
- * da v kolikor se uporabljajo PC-ji, je jasno izražena potreba po povezavi PC-jev v mreže in ne posamezni PC-ji



Tab.1 INTEGRIRANI PISARNISKI SISTEMI

| | |
|-------------------------------|------|
| Mini na nivoju oddelkov | 55 % |
| "Mainframe" centralni sistemi | 22 % |
| Mreza PC-jev | 15 % |
| Samostojni PC-ji | 8 % |

Tab.2 MINIRAČUNALNIKI V PISARNISKIH SISTEMIH

(Delež v totalu US \$)

| | 1985 | 1986 |
|------|--------|--------|
| DEC | 27,7 % | 35,3 % |
| IBM | 24,3 % | 17,9 % |
| WANG | 8 % | 9 % |

Tab.3 INTEGRIRANI PISARNISKI SISTEMI (USA 1986)

| | 1985 | 1986 |
|--------|--------|--------|
| DEC | 22,9 % | 26,0 % |
| IBM | 23,9 % | 20,3 % |
| WANG | 11,3 % | 10,9 % |
| HP | 5 % | 5,4 % |
| Ostali | 25,9 % | 25,6 % |

V nadaljnji obravnavi se bomo oslonili na taxonomijo po karakteristikah opreme, pri čemer pa ne bomo obravnavali sicer zanimivega in pisanega trga hišnih računalnikov, pač pa se bomo omejili na profesionalne sisteme.

3. PROFESIONALNI MIKORARAČUNALNIŠKI SISTEMI

Mikroraračunalniške sisteme je smotrno razdeliti v dve grupi:

1. Enouporabniški mikroraračunalniški sistemi, ki jih popularno imenujemo osebni računalniki ali PC-ji;
2. Večuporabniški mikroraračunalniški sistemi ali supermikroraračunalniki (supermicrocomputers datapro).



Kategorijo enouporabniških mikroračunalniških sistemov opredeljujejo sledeče karakteristike:

Običajno so v uporabi 8, 16 ali 16/32 bitni procesorji, zasnovani na znanih mikroprocesorskih familijah: Zilog Z80, Intel 8088, 8086, 80C18, 80286, Motorola 68000, 68010, itd. ali pa na proizvajalčevih lastnih procesorjih.

Takt procesorja je tipično med 4 in 10 MHz, spomin od 128 K do 1MB (tipično 512 K).

Sistemi so opremljeni z 1 ali 2 pogonoma gibkih diskov, pogosto tudi disk kapacitete 10 - 30 MB.

V tem področju so se uveljavili operacijski sistemi, ki predstavljajo svetovno priznane tržne standarde, v prvi vrsti sta to MS-DOS in CP/M (različne verzije) ter drugi operacijski sistemi softverskih hiš PC/DOS, DOS Microsoft, itn. Cenovni razred teh sistemov leži med 2000 US \$ do 7000 US \$.

Večkrat se PC-ji te kategorije s primernim operacijskim sistemom uporabijo kot večuporabniški sistemi z do 4-imi delovnimi mesti. Pri tem gre zelo pogosto za posebno namensko aplikacijo. Tak sistem je npr. DELTA 400B kot "sistem za zajem podatkov" s štirimi delovnimi mesti, ali pa kot bančna salterska delovna postaja.

Dominanten v tem segmentu trga je IBM s svojima PC/XT in PC/AT računalnikoma, ki v tem segmentu se vedno držita v ZDA 54 % trga, čeprav je njegov delež vseh mikroračunalnikov v svetu padel od 71 % v letu 1983 na cca 33 % v letu 1986.

Tab.4 TRZISCE PC-jev V ZDA

| | 1985 | 1986 |
|--------|------|------|
| IBM | 54 % | 54 % |
| COMPAQ | 5 % | 7 % |
| APPLE | 8 % | 5 % |
| AT&T | 3 % | 4 % |
| ZENITH | 3 % | 4 % |
| Ostali | 26 % | 26 % |



Deleži posameznih proizvajalcev se zelo razlikujejo od države do države, saj so vse nacionalne politike, z izjemo Jugoslavije, naravnane na podporo in zaščito domače proizvodnje (npr. uvedba zaščitnih carin na japonske proizvode mikroracunalske in elektronske industrije s strani ZDA). Tako je na tržišču ZRN delež evropskih proizvajalcev že 20 % in še narašča.

(Primerjaj tabelo 4 in tabelo 5.)

Tab.5 TRŽIŠČE MIKRORACUNALNIKOV V ZRN

| | 1986 |
|---------------|--------|
| IBM | 29,0 % |
| OLIVETTI | 11,0 % |
| SIEMENS | 6,3 % |
| COMMODORE | 5,8 % |
| APPLE | 5,0 % |
| VICTOR | 4,8 % |
| TRIUMPH ADLER | 4,3 % |
| COMPAQ | 3,7 % |
| HP | 2,1 % |
| Ostali | 19,0 % |

Čim bolj je tržišče razvito, toliko bolj se opaža tendenca, da postajajo tehnične karakteristike proizvodov vse manj pomembne.

Med najpomembnejše kriterije se uvrščajo:

- * povezljivost z ostalimi sistemi v organizacijsko kompleksnejšo mrežo računalnikov različnih zmogljivosti
- * cena na delovno mesto
- * razpoložljivost aplikativne programske opreme
- * razširjenost proizvoda
- * sposobnost povezovanja v LAN
- * tržna trdnost proizvajalca

V reputaciji proizvoda in oceni proizvajalca so zajeti servis in ostale storitve, ki predstavljajo "infrastrukturo" proizvoda. Uporabniki vse bolj prihajajo do spoznanja, da je PC oziroma enouporabniška delovna postaja sestavni del informacijskega sistema, ki predstavlja ogrodje organizacije. Nabava in uporaba PC-jev postaja sestavni del strateškega načrtovanja razvoja INFORMACIJSKEGA SISTEMA, s tem pa stopa v ospredje povezava micro-host sistem, ki določa uporabne karakteristike PC-ja kot delovne postaje.

V tem segmentu nudi ISKRA DELTA družino PARTNER v različnih variantah z operacijskim sistemom CP/M PLUS, že omenjeni sistem DELTA 400B z operacijskim sistemom CP/M PLUS in MP/M ter PARTNER/AT z operacijskim sistemom MS-DOS.

Poleg tega so na našem trgu prisotni še sledeči sistemi: IBM PC/XT in PC/AT, COMMODORE, OLIVETTI M2x, HEWLET PACKARD HP 150 M in VECTRA, TANDY 1000 in 3000 HD, pojavljajo pa se tudi firme kot so: ELING, VICTOR, EPSON, SPERRY, itd.

Trg osebnih računalnikov se je ravno v zadnjem času pričel bistveno menjati. V letu 1985 so pričeli sceno agresivno obvladovati CLONE-i, ki so izkoristili standarde (operacijski sistem MS-DOS). Na trgu se je pojavilo niz PC/XT in PC/AT kompatibilnih sistemov in IBM znatneje izgublja svoj delež trga. Posebno zmedo na trgu povzročajo tako imenovani "NO-NAME" producenti, ki imajo svoje izvire na Daljnem vzhodu. Z neprofesionalno, kvalitetno dvomljivo opremo, so pričeli potiskati cene navzdol.

Tab.6 PADEC CEN PC-JEV V ZRN V LETU 1986

| | |
|-----------|------|
| COMMODORE | 10 % |
| PHILLIPS | 14 % |
| SIEMENS | 16 % |
| LANDY | 20 % |
| OLIVETTI | 26 % |
| HP | 29 % |
| IBM | 30 % |

Padec cen gre seveda na račun servisa in ostalih storitev. Takemu vdoru nekvalitetnih CLONE-ov je podvržena predvsem Evropa, prav posebej pa Jugoslavija, kjer taki "NO-NAME" proizvodi dobivajo celo lokalno družbeno podporo. Cena je lahko še toliko nižja, ker gre za določen tip trgovine, brez vlaganja v razvoj in proizvodnjo.



IBM je razumljivo hitro potegnil nauk in tako se je v aprilu pojavila na trgu nova serija PERSONAL SYSTEM/2, ki bazira na INTEL-ovih mikroprocesorjih 8086, 80286 in 80386, z operacijskima sistemoma PC-DOS 3.3 in OS/2, z množico IBM lastnih chipov in novimi 3.5" disketami.

Mimogrede rečeno, IBM opušta dosedanje modele, odpadna tehnologija pa je primeren artikel za nerazvite latinsko-ameriške in afriške države in pa morda tudi Jugoslavijo.

Prihodnost enouporabniških mikroročunalnikov vidimo predvsem v funkciji inteligentnih terminalov.

Zanimivejšo skupino predstavljajo VEČUPORABNIŠKI MIKROČUNALNIŠKI SISTEMI, ali kot jih često imenujejo SUPERMICRO COMPUTERS (dataproc). Definiramo jih kot sisteme, osnovane na mikroprocesorju, ki podpirajo najmanj stiri uporabnike. Mikroprocesor je 16/32 ali 32 bitni s taktom od 5 MHz navzgor. Najpogosteje se uporablja MOTOROLA 68000, 68010, 68020, INTEL 80286, 80386, DEC J11 in MICROVAX ter drugi, pogosto pa tudi firmam lastni procesorji. Nadaljna karakteristika supermicročunalnikov je "memory management", ki omogoča CPU-ju, da hitro ugotavlja, kje so zahtevani podatki v memoriji ali na disku ter podpira funkcijo virtualne memorije.

Običajno so sistemi zgrajeni okoli enega ali več vodil, kot je npr. VME vodilo, MULTIBAS, itd. Najpogostejši operacijski sistemi so UNIX s svojimi izvedbami, PIC, MP/M, UCSD p-System, OASIS, mikroVMS, RSX-11. Vseplošno je opaziti trend k UNIX-u, čeprav ne tako niter, kot je to bilo prvotno pričakovati. Skoraj vsi večji proizvajalci nudijo lastne portacije UNIX-a, kot primarni ali pa kot sekundarni operacijski sistem.

Tab.7 RAZSIRJENOST UNIX-A V USA

| KOT PRIMARNI O.S. | | | |
|---------------------|--------|--------|--------|
| | 1984 | 1985 | 1986 |
| Planirano | 13,8 % | 12,2 % | 12,6 % |
| Instalirano | 7,6 % | 7,8 % | 8,0 % |
| KOT SEKUNDARNI O.S. | | | |
| | 1984 | 1985 | 1986 |
| Planirano | 18,2 % | 15,8 % | 15,0 % |
| Instalirano | 3,5 % | 3,9 % | 5,1 % |

V tej kategoriji sistemov ima ISKRA DELTA družino TRIGLAV, predstavnik firme IBM je PC RT, poleg IBM-a je na tem segmentu dominanten DEC z delovnimi postajami, osnovanimi na mikroVAX procesorju ter mikro PDP-11 sistemi.

Vsak dan bo težje razmejiti to grupo sistemov od mini računalnikov na spodnjem koncu, saj se po performancah močno prekrivajo. Morda bi kot prednostno lastnost mini računalnikov označili praviloma večje možnosti širitev.

Z aplikativnega stališča so področje supermikroročunalnikov mali poslovni sistemi, poslovne in tehnične delovne postaje, CAD/CAM postaje z zahtevnejšo grafiko ter sistemi za procesno upravljanje.

Vse bolj pomembna, tako za supermikroročunalnike, kot tudi za mikroročunalnike, je njihova povezanost z drugimi sistemi v mreže (recimo LAN) ter njihova povezanost na zmogljivejši sistem. Pri tem ne mislimo toliko na fizično možnost komuniciranja, kot pa na integriranost v aplikacijo in njeno vertikalno distribucijo. S tem se odpre problem "Link micro-host", ne v fizičnem, pač pa tudi v softverskem in konceptualnem pogledu, ki po našem mnenju se vedno ni v celoti zadovoljivo rešen.

Poleg strojne komunikacijske opreme in ustreznega protokola (prvi in drugi nivo) mora biti implementirana tudi oprema, ki omogoča sledeče funkcije oziroma servise: prenos datotek in komunikacijo program-program, aplikacijski generator, tekst procesor z integrirano poslovno grafiko, elektronsko pošto, tele-konferenco, ekstrahiranje podatkovne baze, računalniške razpredelnice, itd.

Mikroročunalnik se integrira v aplikacijo v funkciji "front-end" obdelave (data entry, z različnimi vhodnimi kontrolami) ter "back-end" obdelave (formatiranje tiskanih izpisov in ekranskih prikazov, generacija poročil).

Odločno je opaziti težnjo po načrtovanem vklapljanju mikroročunalnikov v integrirane sisteme.

4. MINI RAČUNALNIKI

Miniračunalniške sisteme smo kategorizirali na mini in supermini sisteme. Predikcija mnogih analitikov, da bodo supermikroročunalniki na enem in supermini računalniki na drugem koncu izpodrinili mini računalnike, saj po svojih performancah supermikro sistemi vse bolj posegajo tudi na področje supermini računalnikov, se do sedaj še ni uresničila. Razmejitve med sistemi z oznakami supermikro, mini, supermini, je tvegano podati. Po eni strani je res, da so mikroprocesorji pogosto hitrejši ali enako sposobni kot procesorji miniračunalnikov, vendar to samo po sebi se ne zagotavlja performanc sistema. Po drugi strani pa supermini računalniki, ki sedaj v celoti obvladujejo sceno srednjih računalnikov, vse bolj prodirajo na področje velikih mainframe sistemov.



Nasproti mikroročunalnikom se miniračunalniki odlikujejo praviloma, poleg večjega števila terminalov, po večji hitrosti, večjih diskovnih kapacitetah, znatno večjih komunikacijskih možnostih ter možnostih širitve.

Trdoživost mini sistemov je podprta z obsežnim softwareom in vrsto perifernih enot. Ključni faktor pa je in bo tudi v bodoče razmerje cena/performance. Miniračunalniki izgubljajo na pomenu kot računalniki za splošne potrebe, pač pa dobivajo pomen kot računalniki za posebne namene.

Tako je npr. pri DEC-u, kot enem od vodilnih proizvajalcev mini računalnikov v svetu, obseg posla na popularni seriji PDP-11 večji kot firme Nacional semiconductor ali Data General v celoti. Samo v Evropi je bilo v letu 1986 prodanih cca 7000 sistemov. Tipični predstavniki te vrste računalnikov, ki so pri nas bolj razširjeni, so DEC-ova familija PDP-11, D800 Iskre Delte, zasnovana na isti osnovni arhitekturi, Honeywell DPS 6/16 bitni sistemi, IBM S/36 ter HP 1000 in HP 300 (modeli 37, 58) in HP 9000 (model 200).

5. SUPERMINI RAČUNALNIKI

Supermini računalniki so običajno 32 bitni, srečamo pa tudi drugačne dolžine besed, npr. 48 bitni ali pa celo 64 bitne sisteme. Tipično podpirajo do 250 delovnih postaj. Osnovna cena na spodnjem koncu znaša od 50.000 US \$ do 100.000 US \$ in na zgornjem koncu preko 1 Mio US \$. Moč procesorja je tipično med 0,5 do 10 MIPS. Na zgornjem koncu supermini računalniki vse bolj prevzemajo vlogo velikih mainframe sistemov.

V tem razredu računalnikov se pojavljajo skoraj v vsaki družini tudi dvojno procesorski sistemi, kot npr. VAX 8800, ali pa IBM 4381 MG 14. Namen dvoprocorskih sistemov je bolj v povečevanju performanc kot pa v povečanju varnosti v primeru izpada, čeprav je nekaj proizvajalcev usmerjenih v sisteme, neobčutljive na izpade (fault-tolerant systems). Eden vodilnih na tem področju je TANDEM, pa tudi STRATUS. Zanimivo je, da taki sistemi pri nas niso našli popularnosti, pač pa se problemi okoli izpadov večkrat rešujejo s podvojitvijo sistemov. Korak dalje v to smer je odšel DEC s CLUSTER konceptom, pa tudi ISKRA DELTA s konceptom GEMINI.

Predstavniki te skupine računalnikov na našem trgu so DEC-ova VAX familija, IBM-ove familije S38, 4300, 9370, S38, HP 9000 in HP 3000 serija (predvsem modeli SPECTRUM). ISKRA DELTA je v tem področju zastopana s familijo DELTA/V (ADRIA, D 4860 in GEMINI.) Očitna je strategija vseh proizvajalcev, dati na trg družine, ki performančno pokrivajo čim širše področje.



Do teh sistemov so, glede na aplikativna področja, postavljene sledeče zahteve:

- * enakopravna komunikacija med posameznimi sistemi
- * zrela komunikacijska arhitektura in povezava na sisteme drugih proizvajalcev;
- * čim širše performančno področje
- * integriran aplikativni sistem
- * integracija mikroročunalnikov
- * enostavnost uporabe
- * javne mreže (OSI standardi)
- * D.B.software in informacijska orodja

Na zgornjem koncu, že v področju mainframe sistemov, najdemo IBM 4381 ter VAX 8700, 8800, 8974 in 8978. Pri ISKRA DELTA v tem delu segmenta trga ne tekmujemo. Potrebe tržišča rešujemo s konceptom distribuirane obdelave, torej s homogeno mrežo naših sistemov v mreže ali pa se navezujemo na druge sisteme.

Problem širokega performančnega področja je rešen s premišljeno komunikacijsko arhitekturo DELTA-Net in informacijsko arhitekturo IDA, ki daje uporabniku v celotnem performančnem spektru od sistema TRIGLAV pa do GEMINI homogeno delovno okolje. V to smer gre tudi nadaljni razvoj v smeri sekundarnega operacijskega sistema, osnovanega na UNIX-u. Mišljenja smo, da je večina uporabniških problemov rešljiva s sistemi v rangu sistemov ISKRA DELTA in problem sam ne opravičuje prevelikih investicij v mainframe. Ti sistemi so, ne samo v nabavi, pač pa tudi v eksploataciji (rezervni deli, servis). No, tudi mi v laboratorijih pripravljamo rešitve v smeri večjih sistemov.

Dejstvo je, da sta dobro premišljena arhitektura računalnika in operacijskega sistema daleč pred tehnologijo ter kaže tendenco razvoja, tako v smeri mikroprocesorjev (chip VAX) kot tudi v smeri mainframe sistemov. Napor v to smer je opazen pri vseh pomembnejših proizvajalcih računalniške opreme (DEC, Honeywell, IBM).

Na kratko si pogledjmo njihove produktne strategije, kot jih je mogoče razpoznati na trgu.



Pri HONEYWELL-ovi družini DPS 6 srečamo širok spekter 12-ih 16 bitnih sistemov, od mikro sistema 6/10 do DPS 6/78, katerih moč procesorja sega do 0,68 MIPS-a ter 32 bitni spekter 6-ih sistemov od DPS 6/85 do DPS 6/95-2 z močjo procesorja do 1,8 MIPS-a. Na celotni družini deluje operacijski sistem GIKOS 6. Družina je zanimiva vsled možnosti upgrade-a iz 16 bitnega procesorja na 32 bitni. Upgrade med vsemi procesorji družine ni mogoč. Za nas je ta družina pomembna, ker se nekateri procesorji proizvajajo na licenčni osnovi v tovarni EI Niš. V tem hipu ni popolnoma jasno, kateri procesorji so oziroma bodo vzeti iz proizvodnje.

Firmi DIGITAL EQUIPMENT je ravno širok razpon, po enotni arhitekturi, z enotnim operacijskim sistemom, koncipirane družine VAX, omogočil proboj na drugo mesto med svetovnimi proizvajalci računalnikov. Razpon družine sega od namiznih delovnih postaj, osnovanih na mikro VAX-u do velikih mainframe sistemov VAX 8974 in VAX 8978, ki sta primerljiva z IBM famijlo 3090-180 do 3090-400. Po drugi strani pa je dal DEC na trg v zadnjem letu preveliko stevilo sistemov. Postavlja se vprašanje, ali je to upravičeno, poleg tega pa povzroča probleme pri vzdrževanju.

IBM je v lanskem oktobru najavil serijo 9370 in s tem se povečal konfuzijo v področju srednje velikih računalnikov te firme, saj je na tem področju kar 7 procesorjev: SYSTEM1, S/88, 8100, S/36, S/38, 4300 in 9370, več ali manj med seboj nekompatibilnih.

Očitno gre pri sistemih 9370 za zamenjavo sistemov 4341, 4361. Model 9377-90 naj bi po njihovih navedbah dosegel zmogljivosti modela 4381-1. Tako lahko pričakujemo v naslednjem obdobju dodatne modele na spodnjem koncu (namizno verzijo, zasnovano kot delovna postaja) in tudi še močnejše modele na zgornjem koncu.

Operacijski sistemi so prav gotovo najšibkejša točka. So ne samo "požrešni" kar se tiče uporabe računalniških resourcov, pač pa so tudi relativno komplicirani za uporabo. Pod sloganom "...zadovoljiti vse potrebe uporabnikov..." se skriva neenotnost SW koncepta in njegova neprilagojenost možnostim HW. Gre se vedno za batch orientiran SW, na katerega so dodatno prilepljene time-sharing lastnosti, torej za zasnovo iz 60-ih let!

IBM je in ostaja dobavitelj velikih mainframe sistemov arhitekture S/370, za katere je objavil, kot "strategic product of direction" operacijski sistem MVS/XA (extended architecture), ki pa na sistemih 9370 ni razpoložljiv.

Pisano ponudbo svojih operacijskih sistemov opravičuje: "...Ne moremo priporočiti iste rešitve za vse uporabnikove potrebe..."



Seveda je pri tem zamolčano dejstvo, da uporabnikove potrebe niso statične, pač pa se z rastjo uporabnika spreminjajo in uporabnik ne more kupiti enega sistema za današnje in drugega za jutrišnje potrebe. Težava je še toliko večja, ker je migracija aplikacij med sistemi zelo zamuden posel, medsebojna povezljivost pa tudi ni enostavna. Kar uporabnik res potrebuje, je enotna informacijska arhitektura, ki mu omogoča postopno kontinuirano rast.

Poleg IBM-ovih mainframe sistemov, med katere sodijo sistemi 3481, 3080 in 3090 familije, prodira v to področje DEC, tako s Cluster konceptom, kot tudi z VAX 8974 in 8978. Na jugoslovanskem tržišču je v tem segmentu poleg IBM-a dominanten tudi Sperry in Burroughs. Po združitvi obeh firm pa je daljna perspektiva vsakega od obeh programov vsaj zaenkrat nejasna.

6. ZAKLJUČEK

Stojimo na stališču, da je v končni konsekvenci, s stališča uporabnika, to je s stališča gospodarstva, odločujoča rešitev nekega problema po določeni optimalni ceni. To rešitev pa sestavlja tako strojna oprema, sistemski software, aplikativni software in končno vključitev vsega tega v organizacijo kot celoto delovne organizacije in to zadnje je tudi najšibkejša točka računalništva pri nas. Mi postavljamo svoj koncept, zgrajen na principih distribuirane obdelave, kjer se posamezne funkcionalne enote vključujejo po svojih manjših sistemih in se le-ti dalje povezujejo v mreže. Mislimo, da je to sodoben koncept, dobro prilagojen našim tehničnim, kot tudi ekonomskim razmeram.

7. LITERATURA

Pri izdelavi tega referata smo uporabljali sledeče vire:

- * DATAMATION
- * COMPUTERWORLD
- * MARKET & TECHNIK
- * DATAPRO REPORTS ON COMMUNICATIONS
- * DATAPRO REPORTS ON MINICOMPUTERS
- * DEC SALES UPDATE.

**RACUNALNISKE ARHITEKTURE IN
UMETNA INTELEGENCA**

Avtor referata: prof.dr. Anton P. Zeleznikar, dipl.ing., ID



Ljubljana, maj 1987



V tem spisu obravnavana problematika opisuje nekatere nove pa tudi standardne perspektive razvoja računalniških arhitektur in njihove inteligentne uporabe. Poudarek se daje paralelnim arhitekturam in novim usmeritvam t. i. tehnološke inteligence. V tej zvezi prihaja v ospredje potreba po novi opredelitvi informacije, ki bi lahko postala podlaga tudi za tehnološko inteligenco v novi perspektivi. Današnje tehnološko in uporabnostno razpotje med starim in novim je prikazano v kritični perspektivi in v perspektivi krize. Naslovi tega spisa so: novo pojmovanje informacije; paralelnost in inteligenca kot razvojna atributa računalništva in informatike; paralelne računalniške arhitekture; kriza umetne inteligence; nove računalniške tehnologije in njihova uporaba; raziskave in razvoj paralelnih sistemov in informacije v Iskri Delti; in nazadnje se nadaljevanje razvoja standardnih računalniških sistemov in njihovih orodij.

UVOD

Računalniška tehnologija in njena uporaba sta se približali razpotju: prva pot vodi v smeri že osvojene, racionalistično-trde uporabe računalniških orodij in za to pot je značilna današnja znanstveno, tehnično (inženirsko) ali aplikativno (administrativno) trda osnova; druga pot vodi v smeri inteligentne, vobče informacijsko-mehke uporabe prihodnjih računalniških orodij in za to pot je danes značilno iskanje netradicionalnih, informacijskih temeljev. Lahko bi rekli, da se računalniška tehnologija sooča s svojo kar dolgotrajno razvojno slepoto, ki sega od njenih strojnih začetkov v štiridesetih letih, prek enostavnih simbolnih pripomočkov v petdesetih letih, do formalno-jezikovne uporabnosti današnjih dni. Ta slepota doživi svoja prva metodološka prelamljanja s pojavom umetne inteligence v sedemdesetih letih, ko postane jasno, česar tedanji in prihodnji računalniki ne bodo več zmogli. Računalniki prve razvojne poti ne bodo dosegli zmogljivosti, ki bi bile primerljive s človekovo inteligenco.

Pojem inteligence, ki se pojavi v povezavi z umetno inteligenco, sprva le kot ne-inteligenca oziroma kot bistvena metodološka izboljšava brez dejanske inteligence, začne sčasoma na svoji poti odkrivati nepremostljive ovire strojnega in programskega inteligentnega delovanja. Čedalje bolj jasno postaja, da zamisli inteligence kot dinamičnega informacijskega pojava ne ustreza niti statična računalniška arhitektura niti dobro-strukturirani programi formaliziranih jezikov. Statična strojna arhitektura in programirna tehnika namreč ne premoreta osnovne informacijske pojavnosti, ki se imenuje informacijska nastajalnost. Prej ko slej postane namreč očitno, da je mogoče inteligenco kot funkcionalno zmogljivost strojev in programov vložiti ali vgnezditi le v nastajajočo informacijo. Strojno nastajalnost je mogoče doseči le z dinamično spremenljivo strojno arhitekturo med delovanjem stroja, programsko nastajalnost pa z možnostjo neomejenega spreminjanja programov med njihovim izvajanjem.



V drugem primeru se današnja programirna tehnika s strukturiranim programiranjem izkaže kot izrazito neprimerna, če ne že kar neinteligentna.

Pojem inteligence pripelje končno do nakazanega razpotja: druga pot ostaja iskanje možnosti realizacije informacijskih strojev in informacijskih programov, v katere je mogoče vložiti inteligenco. Razpotje je razvojno, evolucijsko: današnji računalniški sistemi prve poti imajo svoje razvojno nadaljevanje, ki je tehnološko (npr. v okviru fizike trdne snovi) in metodološko značilno (npr. v okviru metodologije racionalističnih znanosti in tehnik). Ta smer ima svojo lastno perspektivo, ki je utemeljena z racionalistično tradicijo in se nadaljuje v prihodnost kot povsem primerna za določena področja tehnoloških in administrativnih aplikacij. Druga smer je informacijska in cilj te smeri je doseganje višjih informacijskih funkcij, kakršna je tudi inteligenca. Ta smer se razvija bistveno drugače kot prva smer, in sicer tehnološko (npr. uporaba bioloških substanc, nevrlnih vezij) in metodološko (npr. uporaba procesov oziroma pojavov različnih fenomenologij, od fizikalnih, bioloških, nevrofizioloških, biokemičnih, do kompleksnih korteksnih procesov).

Sodobna računalniška industrija mora danes pozorno spremljati obe razvojni smeri: tradicionalno, tehnološko-trdo in novo, informacijsko-mehko pot. Prav zaradi tega ni več nenavadno, da imajo veliki industrijski koncerni od Japonske do ZDA tudi ustrezno opremljene in delujoče laboratorije za raziskave bioloških (zlasti celičnih) pojavov in njihovo uporabo v novih, visokofunkcionalnih informacijskih (npr. nevrlnih) vezjih. V okviru tega spisa bomo pregledali le nekatere značilne usmeritve novogeneracijskega računalniškega razvoja in razvoja tistega dela umetne inteligence, ki se začinja utemeljevati na novih informacijskih osnovah.



1. NOVO POJMOVANJE INFORMACIJE

Pojem 'informacija' uporabljamo kot danost, ki je za vselej enoumno razumljiva. Zlasti v množinski obliki se nam ta pojem pojavlja v pomenu podatka, sporočila, torej kot opredeljena, statična informacijska entiteta. Kadar razmišljamo o informacijskih procesih v možganih, se to razmišljanje največkrat reducira na predstavo, da drugi, neinformacijski možganski procesi premetavajo in preoblikujejo nekakšno informacijo. Le redko razmišljamo v smeri, da bi lahko bila v možganih prav informacija tista, ki informacijo oblikuje, torej informacija-proces oziroma informacija kot proces.

Če razumevamo informacijo kot obliko (lat. forma, gr. eidos, morfe) in kot proces (lat. informare, gr. poiesis), postane pojem informacije dinamičen. Informacija o informaciji nam lahko pomeni informacijo o procesu, ki je tudi sam informacijski proces. S takšnim pojmovanjem informacije je mogoče razumevati tudi spontano, torej vnapraj neopredeljeno nastajanje informacije, seveda tudi kot procesne nastajalnosti. Glagol 'informirati' dobi v tej zvezi moč nastajanja informacije in seveda tudi nastajanja informiranja samega. V vsem tem se informacija očitno utemeljuje z informacijo. Lahko rečemo: novo informacijo je mogoče vložiti oblikovno in procesno v obstoječo informacijo. In prav to je pojav, ki ga živo bitje - nizko in visoko razvito - ves čas dejansko doživlja. Nova informacija nastaja iz obstoječe informacije in se v obstoječo informacijo strukturno-pomensko in organizacijsko-pomensko, vlaga, torej prav z obstoječo informacijo pomensko utemeljuje in pomensko polašča.

Z nastajanjem informacije se pojavlja nova informacija, ki se pomensko navezuje na obstoječo informacijo. To lastnost informacije lahko poimenujemo s protiinformacijo. Protiinformacija nastaja z opazovanjem, raziskovanjem in spoznavanjem obstoječe informacije, torej iz dane informacije, kot kritično-spoznavni odnos do same informacije, kot informacija izven obstoječe informacije, kot informacijsko nasprotje. Tako lahko rečemo: z informacijo nastaja njena protiinformacija, ki se vlaga v obstoječo informacijo. Informacija je očitno krožna informacijska lastnost, ki je informacijsko paralelna, serijska in rekurzivna. Tako pridemo do naslednje, skrajno zgoščene in že na določen način formalizirane formulacije informacije:

Informacija s protiinformacijo je procesnost informiranja, vlaganja, nastajanja in protiinformiranja informacije z lastnostjo krožnosti (cirkularnosti), ki je rekurenčna (rekurzivna, kompleksno in nastajalno ponavljajoča), paralelna in serijska.



S to formulacijo nismo povedali ničesar bistveno novega, kar ne bi že poznali iz uporabe današnjih računalniških sistemov. Nase dosedanje razumevanje informacije smo le nekoliko posplošili. Ali niso računalniški programi informacija, s katero se procesira druga informacija? Računalniški program je načrt procesa za pridobivanje različnih podatkov, pri čemer je potek tega procesa seveda lahko odvisen od podatkovne vsebine. Današnji računalniški sistem pa je še vedno informacijsko statičen, saj je nastajanje njegovih programov mogoče le z zunanjimi (sistemskimi ali programirnimi) posegi in izvajajoči programi in arhitektura sistema vobče niso podvrženi nastajalnim spremembam.

2. PARALELNOST IN INTELIGENCA KOT RAZVOJNA ATRIBUTA RAČUNALNIŠTVA IN INFORMATIKE

V tem besedilu pojmujeemo računalništvo kot tehnološko in metodološko osnovo računalnikov, informatiko pa kot raznovrstna področja uporabe računalnikov. V osemdesetih letih sta se filozofsko in tehnološko izoblikovali dve medsebojno tesno povezani in soodvisni razvojni usmeritvi računalništva in informatike, in sicer paralelna arhitektura in inteligenčnost novogeneracijskega računalniškega sistema kot tehnološkega proizvoda in uporabniškega pripomočka (orodja). Arhitekturna paralelnost ima svoje začetke že v sistemih iz šestdesetih let in se kasneje nadaljuje v različnih oblikah v multiprocesorske in superračunalniške sisteme. Pojem sistemske inteligence se pojavi z umetno inteligenco tudi v šestdesetih letih. V začetku osemdesetih let pa sprožijo Japonci svetovni razvojni izziv, ko ustanovijo in objavijo vladni projekt razvoja računalnikov t. i. pete generacije. V okviru tega projekta se prav paralelnost in inteligenca razglašata kot bistveni zmogljivosti tega, kar poimenujejo Japonci kot novogeneracijsko.

Kako je zakrita problematičnost lastnosti paralelnosti in inteligenčnosti novogeneracijskega sistema? Kje se skrivajo tehnološke, razvojne in uporabniške pasti in zaenkrat malodane nepremostljive ovire? Razvijmo najprej vpraševanje o računalniški (sistemski) in uporabniški paralelnosti. Živi organizmi so nazorni primerki paralelne procesnosti. Kot vsakdanje izkustvo doživljamo razmišljanje o več stvareh hkrati, pa tudi hkratno razmišljanje o več podrobnostih enega samega problema ali procesa. Človekovi možgani omogočajo paralelno razmišljanje s svojo strukturo, organizacijo in s svojo regularno (umsko, priučeno, izkustveno) procesnostjo. Beseda 'hkrati' nakazuje v teh primerih paralelnost. To paralelno delovanje (razmišljanje, obnašanje) pa je mogoče le tedaj, če je celota (problem, misel, program) ustrezno razbita, tako da je omogočeno ločeno procesiranje njenih delov z ločenimi procesorji in ponovno sestavljanje ločeno procesiranih delov v celoto. Z dilemo razstavljanja, ločenega (hkratnega, paralelnega) procesiranja in sestavljanja smo v bistvu že nakazali jedro problematičnosti vsakega umetnega (tehnološkega, metodološkega) paralelnega procesiranja.



Paralelnost je tehnološki (strojni) in uporabniški (informacijski) problem. Paralelna arhitektura mora razreševati probleme učinkovitega medsebojnega komuniciranja med ločenimi procesorji, pomnilniki in drugimi strojnimi viri, uporabniška paralelnost pa mora razstavljati probleme v paralelne procese in jih zopet sestavljati. Pri tem je uporabniška paralelnost seveda odvisna, oziroma pogojena s paralelno strojno arhitekturo. Za določeno strojno arhitekturo so potrebni programski pripomočki (operacijski sistemi, prevajalniki, komunikacijski programi itd.), ki omogočajo učinkovito uporabo paralelnega računalniškega sistema.

Uporabniška paralelnost, s katero bi računalniški sistem avtomatično izpeljal t. i. problemsko dekompozicijo v paralelne procese, izvršil obdelavo posameznih paralelnih procesov na razpoložljivih procesorjih in naposled sestavil rezultat vhodnega problema, je zaenkrat realizirana le na ravni nekaterih klasičnih programirnih jezikov (fortran, pascal itd.). Paralelnost ostaja tako prek ko slej tista smer razvoja računalniške arhitekture in uporabe, ki konceptualno se ni dovolj prečiščena, učinkovita in standarizirana. Čedalje bolj jasno postaja tudi, da bodo paralelne arhitekture zahtevale nove programirne jezike, s katerimi bo lahko tudi uporabnik izražal svoje posebne zamisli paralelnega procesiranja.

Inteligenca tehnoloških sistemov, ki bi bila primerljiva z inteligenco živih bitij, bo mogoče tehnološko realizirati šele s koncepti dinamičnih (nevronskih) računalniških arhitektur in na novih temeljih umetne inteligence. T. i. peta računalniška generacija (citiram T. Winograda in F. Floresa) uporablja pojem 'inteligentno' le v sibirskem pomenu fraz kot so 'inteligentni terminali'. V tem pomenu se omenjajo 'inteligentne povezovalne funkcije', 'inteligentna sistematizacijska podpora', 'inteligentno programiranje', 'inteligentno načrtovanje integriranih vezij', 'inteligentni uslužnostni sistem', 'inteligentna orodja za programiranje' in 'inteligentni komunikacijski sistem'. Vse to pa so le reklamna besedila za nekoliko boljše in bolj dodelane izvedbe današnjih proizvodov. Današnja umetna inteligenca bo našla svojo pot predvsem v različnih posebnih uporabah, ne bo pa imela osrednjega vpliva na splošno uporabnost strojev. Nekateri zelo ambiciozni cilji umetne inteligence, kot sta splošno (univerzalno) strojno prevajanje in razumevanje realnega naravnega jezika, pa bodo preprosto izginili.

Seveda pa ostaja inteligenca prej ko slej bistveni atribut prihodnjega tehnološkega razvoja. Ta tehnološka inteligenca pa ne bo temeljila na današnji racionalistični usmeritvi umetne inteligence, temveč na novem pojmovanju informacije.



3. PARALELNE RACUNALNIŠKE ARHITEKTURE IN NJIHOVA UPORABA

Današnje potrebe inženirskega in znanstvenega dela usmerjajo računalniško industrijo v novo obdobje, v katerem naj bi se dokončno uveljavilo paralelno procesiranje. Usmeritev v paralelno procesiranje, ki je brzkone edina pot za zadovoljevanje nenasitnih potreb tehniških uporabnikov po numeričnih obdelavah, predstavlja verjetno najbistvenejši napredek v računalniški tehnologiji po letu 1947, ko je John von Neumann izumil serijsko (posledično) arhitekturo. Raznovrstne nove arhitekture si že utirajo pota iz industrijskih laboratorijev. Te radikalno različne oblike ne bodo spremenile le načinov tehničnega reševanja težavnih problemov, temveč tudi razumevanje teh problemov. Nove možnosti uporabe takih sistemov pa prinašajo tudi nove izzive. Novi stroji bodo podobno kot njihovi von neumannski predhodniki odvisni od programske opreme.

Pa tudi uporabniki novih strojev bodo morali imeti več znanja o strojni opremi kot so jo imeli doslej. Podobno bodo morali načrtovalci novih strojev vedeti precej več o namenskih uporabah paralelnih računalnikov. Napačna izbira paralelnega sistema na proizvajalski ali uporabniški strani bo lahko imela za posledico občutno, tudi usodno kapitalsko izgubo.

Uporabniki, ki bodo sami pisali programe, se bodo srečevali z nepredvidljivimi težavami preizkušanja in popravljanja programov zaradi pomanjkanja programirnih orodij. Uporabniki bodo zaradi tega morali tudi sami spemljati dogajanja pri razvoju teh orodij. Vendar bodo paralelni računalniki kljub vsemu postali nepogrešljivi pripomočki inženirjev prihodnosti. Ko bodo načrtovalni problemi postali težje rešljivi, bo za njihovo zanesljivo reševanje potrebna tudi večja računalniška moč. Ko se bodo integrirana vezja zmanjševala, jih ne bo mogoče več modelirati z diskretnimi elementi. Potrebno bo reševati enačbe polja najprej v dveh, kasneje pa tudi v treh dimenzijah za integrirana vezja. Računalniška moč načrtovalnih delovnih postaj se bo se naprej povečevala. Vendar pa bo zahtevana procesirna zmogljivost pri današnjih tehnologijah kmalu dosegla svojo gornjo mejo. Z galijevim arsenidom bo mogoče preklopne hitrosti povečati od petkrat do desetkrat, seveda le z dovolj visoko ceno. Vendar pa bo neglede na hitrost procesorjev mogoče pošiljati signale s hitrostjo svetlobe skozi procesorsko kocko s stranico 30 cm v časih, ki ne bodo krajši od ene nanosekunde. Torej bo edina možnost za povečevanje podatkovnega pretoka razširjeni, večkanalni, paralelni pretok.

Druga pot v povečevanju zmogljivosti s paralelnostjo bo uporaba dolgih ukazov, npr. ukazov z dolžino 96 bitov (podjetje Culler Scientific) ali celo 1000 bitov v superračunalnikih (podjetje Multiflow Computer). Izredno dolgi ukazi lahko učinkovito krmilijo procesorska (paralelna) polja z uporabo drobnozrnatega paralelizma. Se večkrat pa bo več mikroprocesorjev lahko povečalo podatkovni pretok ob nizki ceni.



Programi, ki so danes uspešni na serijskih strojih, ne bodo mogli biti uporabljeni na paralelnih strojih. Programske spremembe bodo bistveno odvisne od arhitekture in topologije večprocesorskega sistema. Ze zgodaj se bo potrebno razvojno odločati med različnimi arhitekturnimi in operacijskimi možnostmi, kot so npr. enkratni ukaz in mnogokratni podatki (SIMD), mnogokratni ukaz in mnogokratni podatki (MIMD), sporočilno in paketno preklapljanje, deljeni in porazdeljeni pomnilnik. Kadar program zahteva pogostno izmenjavo podatkov, je npr. deljeni pomnilnik najbolj primeren.

V današnjih paralelnih računalnikih so procesorji dostikrat povezani z vodilom, in sicer tako, da je mogoče procesorje dodajati po potrebi. Najpogostejši operacijski sistem je modificirani Unix z dodatkom za izvajanje v realnem času. Ti stroji uporabljajo tudi optimizacijske prevajalnike, ki razdelijo uporabniški (vhodni) program avtomatično v procese, ki se izvajajo paralelno. Tako uporabnik ni obremenjen s problemom paralelnega strukturiranja svojih programov.

Ti prevajalniki so navadno prirejeni za uporabo jezika Fortran (veliki numerični programi) ali jezika C. Uporaba jezika Ada je bolj problematična zaradi njegovega rendezvous (srečevalnega) mehanizma pri medprocesorski komunikaciji. Tako so Fortran, C in Ada prav gotovo jeziki za današnjo uporabo multiprocesorskih sistemov.

Seveda pa prevajalniki ne morejo izboljševati samih programskih algoritmov (spreminjanje pomena programa). Dodatki za jezik Fortran omogočajo, da računalniški sistem sam modelira uporabo čim večjega števila paralelnih procesorjev. Povečanje hitrosti za 80 do 90% je mogoče že pri uporabi sistemov s 4 do 10 paralelnimi procesorji. Težava je v tem, da dobljeni programi niso prenosljivi na stroje z različnimi arhitekturami. Glavna ovira je v različni zrnatosti paralelizma (npr. drobna, srednja, ali groba zrnatost). Paralelni sistemi s skupnim vodilom delujejo večinoma grobozrnato, npr. s subrutinami.

Srednjezrnati paralelizem, ki obsega npr. zanke ali celo posamezne ukaze, je sicer izvedljiv, vendar je mogoče pri sto ali tisoč paralelnih procesorjih izkoristiti drobnozrnatost posameznih operacij ali celo paralelizem v okviru samih operacij.

Naslednjo skupino strojev za splošno uporabo sestavljajo t. i. drevesni stroji. Znani primer takega stroja je Dado, ki je bil razvit na univerzi Columbia in se trži kot računalnik pete generacije. Dado uporablja posebno razvrstitev procesorjev v binarnem drevesu, t. i. H-drevo, ki ga je mogoče domala neomejeno siriti. Dado začena z osmimi procesorji tipa 68020, razsiriti pa ga je mogoče na sistem z 8192 procesorji, kjer se doseže hitrost 14 milijonov ukazov na sekundo. Pri demonstraciji govorne analize je Dado s 512 procesorji procesiral trisekundni stavek v realnem času, docim je VAX 780 porabil za to nalogo kar 12 minut.



Slaba stran drevesne arhitekture je, da morajo podatki včasih potovati navzdol do drevesnih korenin. To pomanjkljivost bi bilo mogoče nadomestiti s križnimi povezavami. Inmosov trasputer ima že sam stiri vgrajena komunikacijska vrata, kar omogoča konstrukcijo mreže procesorjev. Tudi evropski projekt Parsifal uporablja mrežno arhitekturo. Hiperkocka ali binarna n-kocka lahko raste domala neomejeno brez izgube svoje zmogljivosti. Doslej je bila realizirana hiperkocka reda 16, v kateri je 65536 procesorskih vozlišč, vsako vozlišče pa ima 16 sosedov (je povezano neposredno s šestnajstimi sosednjimi vozlišči).

Druga uporabna lastnost hiperkocke je njena možnost posnemanja drugih topologij. Tako je mogoče npr. hiperkocko uspešno uporabiti pri realizaciji metuljčne (butterfly) metode za hitro Fourierjevo transformacijo.

Učinkovitost paralelnega sistema je bistveno odvisna od uporabljenih algoritmov v aplikativnih programih. Pri računanju vrednosti 'pi' s serijskim strojem se npr. lahko uporabi aproksimativna poligonska metoda, s katero se v vsaki iteraciji podvoji število oglišč in se tako približuje obliki kroga. S paralelnim strojem bi bila integracija arctan-krivulje (z uporabo pravila n-točkovne pravokotniške kvadrature) veliko hitrejša, ker je mogoče hkrati izračunavati več pravokotnikov.

Izbira algoritma je odvisna tudi od velikosti hiperkocke in od velikosti problema. Pri reševanju eliptičnih parcialnih diferencialnih enačb se uporablja npr. implicitna metoda alternativnih smeri. Ocenjevalni postopki (benchmarks) kažejo, da so pri uporabi serijske ciklične redukcije na eni strani in Gaussove eliminacije na drugi strani primernejše velike hiperkocke. Uporaba paralelnih sistemov oziroma njihova primernost je tako vobče odvisna tako od njihove velikosti kot od uporabniškega problema.



4. KRIZA UMETNE INTELEGENCE

Kriza umetne inteligence se pojavlja kot zavest o razkoraku (o razkroju, razpadu) med obljubo (vizijo, avantgardo) in dejanskim stanjem (zmogljivostjo, metodologijo), med cilji umetne inteligence (željami, utvarami) in njenimi dejanskimi možnostmi (tehnologijo, uporabo). V tej zvezi je v območju umetne inteligence problematična prav inteligenca, ki se ji očita, da je v okviru umetne inteligence neinteligentna. Četudi bi računalnik vselej zmagoval v šahovski igri, bi Winograd in Flores trdila, da ta program ne predstavlja inteligence (šah je naposled le deterministična igra).

Kritika usmeritve umetne inteligence je stara in se resneje pojavlja že s H. L. Dreyfusovo knjigo "Česa računalniki ne zmorejo" (1972). Ta kritika doseže svoj idejni vrh s knjigo T. Winograda in F. Floresa "Razumevanje računalnikov in spoznavanja," objavljeno v prejšnjem letu. Ta knjiga izzove intenzivno reakcijo v obliki njenih ocen v časopisu Artificial Intelligence letos; protinapad izhaja iz doslej pravovernih umetnointeligentnih skupin (med njimi je tudi glavni urednik časopisa Artificial Intelligence). Inteligenca v okviru umetne inteligence preprosto ni dosegljiva s temeljev in položajev trde racionalistične tradicije, ki je utemeljena z programirljivo in matematizirano metodologijo, za katero so značilni formalizirani vzorci, algoritmi, jeziki s formalnim instrumentarijem sklepanja in formalne abstraktne transformacije.

Potreba po novi opredelitvi umetne inteligence postaja tako nujna. Izjavljanje, da umetna inteligenca ni inteligenca v filozofskem in psihološkem pomenu, ne prispeva k razjasnjevanju. Tu se pojavlja vprašanje, ali je morda potrebno celo inteligenco opredeliti drugače, kot je bila tradicionalno (psihološko in filozofsko) opredeljena doslej (Binet, Hobhouse, Thorndike, Heidegger, Turing, Gilford, Eysenck itd.). Vprašanje o strukturi in organizaciji inteligence odpre vprašanje o njenih temeljih. Ali so temelji inteligence informacijski? Kaj je informacija v živem kot oblika in proces življenja? Ali je mogoče inteligenco utemeljiti kot funkcijo informacije, kot funkcijo informacijskih oblik in informacijskih procesov?

Zakaj naj bi se umetna inteligenca po svojih začetnih neuspehih odrekla funkcionalnosti, ki je podobna naravni inteligenci oziroma inteligenci živega?

Očitno bo potrebno inteligenco utemeljiti nekoliko drugače in bolj inovativno in temeljito kot doslej, da bo mogoča njena konstruktivna vključitev v okvire umetne inteligence. Heidegrski pojem inteligence je orisan z vrženostjo bitja v svet, v tej vrženosti pa sta tako bitje kot svet nepredvidljiva, vnaprej neopredeljiva. Narava inteligence je bila opredeljena kot posebna informacijska oblika in informacijski proces (glej Zeleznikar: Information Determinations I).

Inteligenca (tudi umetna inteligenca) je specifično strukturirana, organizirana in usmerjena informacija, ki zadeva različne oblike in procese živega, dejavnosti in kulture. Inteligenca je informacijsko prepletena struktura, ki je problemsko (vpaševalno) povezana s posebno, usmerjeno informacijo, ki informira (informatitira) to informacijo na poseben, rezultativen način, ko producira t. i. trenutne rešitve danega (inteligentnega) problema. Inteligenca je posebna informacija, ki zadeva posebno problemsko domeno in daje informacijske rezultate v obliki posebnih informacijskih oblik in posebnih informacijskih procesov. Na ta način je vsaka inteligenca povezana s svojo lastno slepoto in s svojimi lastnimi prelomi (te slepote). Slepota vsakršne inteligence izvira iz njene posebnosti, značilnosti in usmerjenosti glede na celotno, širše informacijsko okolje. Inteligentni prelomi prekinjajo njeno slepoto in omogočajo njen razvoj skozi novo, inteligentno-bistveno informacijo, ki je nastala kot inteligentna raziskava v okviru same inteligence.

Inteligenca je informacijski proces ali določena polastitev inteligentne strukture, organizacije in usmeritve. Inteligenca informatizira polastitveno, bistveno in sebi primerno glede na svoje problemsko območje. To območje se pojavlja v inteligenci kot objektna informacija, ki je informatizirana s to informacijo in s samo inteligenco. Na ta način nastaja problemska informacija, ki je odvisna od globine problemskoreševalne inteligence. Inteligenca je tako tudi informacija o problemskoreševalni stopnji, globini, strukturi, organizaciji in usmeritvi. Taka inteligentna polastitev povzroča inteligentno slepoto toda tudi inteligentne prelome (te slepote). Na ta način je vsaka inteligenca inteligentno relativna, ko vsebuje sebi značilno slepoto (intelligentno utvaro ali intelligentno neumnost) in ko hkrati informatizira svoje prelome (inteligentno bistvena dejstva, ki prihajajo na površino samo v posebnih primerih).

Inteligenca je kot informacija vselej vstavljena (vgnezdena) v širši informacijski kontekst. Inteligenca nastaja v informacijsko odvisnem sistemu, njena oblika in procesnost pa sta odvisni od strukture in organizacije informacije v sistemu. Inteligenca se cirkularno oblikuje iz inteligence in druge informacije, ki je inteligentno in sistemsko odvisna. Inteligenca razvija sama sebe in raste na višje in različne informacijske ravni.

Tu je pomembna ugotovitev, kako sta naraščajoča in razvijajoča informacijska prepletenost na eni strani in pojavljajoča inteligenca na drugi strani tesno povezana procesa. Zaradi tega inteligence ni mogoče popolnoma jasno ločevati od splošnega in širšega informacijskega ozadja. Tu se kaže, kot da je inteligenca le ena od posebnih informacijskih oblik, ki nastaja iz informacijskega ozadja. V okviru tega ozadja pa ima inteligenca svojo lastno strukturo, organizacijo in razvojno usmeritev, ki določajo njeno "inteligentnost."



Inteligenca se informacijsko vselej nanaša na določeno področje, problem ali položaj. V vsakem primeru je mogoče govoriti le o inteligenci, ki je več ali manj specifično usmerjena, ki zadeva specifične načine, dejavnosti in oblike. Podobno kot pri informaciji je tudi pri inteligenci mogoče opazovati splošen princip njenega nastajanja iz raziskovanja, vpraševanja in protiinformatiziranja. Inteligenca je vobče informacijska oblika z značilnim problemskousmerjenim, učinkovitim in pritiinformacijskim raziskovanjem, ko se išče posebna rešitev za posebno problemsko situacijo. Inteligenca je podobna vpraševalnemu sistemu, kjer je problemsko vpraševanje glavni informacijski proces in problemsko odgovarjanje glavni trenutni rezultat. Tako postaja jasno, kako je inteligenca cirkularen informacijski mehanizem s raziskovalnim krogom vprašanja, vpraševanja in odgovarjanja. Ta krožni proces je živa, nastajajoča in razvijajoča oblika in usmeritev, ki deluje dinamično na inteligenco samo.

Inteligenca je naravna informacijska lastnost, ki doslej tehnološko se ni bila realizirana. Ta lastnost obstaja vselej v enostavnih in zapletenih živih sistemih. Namen (naloga, skrb) inteligence v živem je preživetje vrste. Inteligenca na ravni zavesti je lahko vdomljena tudi izven preživetvenih principov kot oblika kulturne dejavnosti. S tega vidika je očitno, da umetna inteligenca zaenkrat se ne obstaja kot inteligenca.

5. NOVE RAČUNALNIŠKE TEHNOLOGIJE IN NJIHOVA UPORABA

Uporaba določene računalniške tehnologije je vezana na številna in raznovrstna orodja, ki so za proizvodnjo tehnologije potrebna. Standardne tehnologije razpolagajo s svojim načrtovalnim in proizvodnim orodjem, ki omogoča ekonomsko izdelavo tehnoloških elementov in sistemov. Nove računalniške tehnologije se nimajo zadostnega orodja za njihovo načrtovanje in izdelavo in so tako tudi bistveno omejene. Preučevanje uporabe neke nove tehnologije mora biti zato vselej podrejeno utemeljenosti izbire neke nestandardne tehnologije, ekonomski in tehnološki upravičenosti posebne tehnološke izbire.

Današnja standardna in perspektivna tehnologija temelji na polprevodniških substratih CMOS in GaAs. Z razvojem živčnih vezij prihajajo v poštev tudi biološki substrati in njihovi spoji s kovino. V integriranih vezjih se praktično ne uporabljajo dinamična elektromagnetna vezja in Josephsonovi spoji (supraprevodniki). To seveda ne pomeni, da v naslednjih letih ne bo bistvenih premikov v smeri novih tehnologij. Očitno bomo morali tudi v lastnih integriranih vezjih prej ko slej uporabiti substrate tipa CMOS in seveda GaAs, ko se bomo vključevali s svojimi izdelki v svetovna računalniška tržišča.



6. RAZISKAVE IN RAZVOJ PARALELNIH SISTEMOV IN INFORMACIJE V ISKRA DELTI

Pobuda za razvoj t. i. novogeneracijskih računalniških produktov Iskre Delte je prišla iz vodstvenega strateškega koncepta Iskre Delte na osnovi studij dveh obiskov na vodilnih japonskih univerzah (Tokyo, Tsukuba in Sapporo), institutu RIKEN in vladnem institutu (ICOT) za razvoj t. i. japonske pete računalniške generacije (glej npr. v seznamu slovtva, A. P. Zeleznikar: Od Sappora do Tokia nazaj v Ljubljano). Po obisku Fakultete za računalniške znanosti Tokijske univerze v juniju 1986 je postalo tudi vse bolj jasno, da so t. i. uradni (vladni) projekti novogeneracijskega računalništva razvojno vprašljivi. Profesor Eiichi Goto na Tokijski univerzi se je lotil npr. raziskav in razvoja konceptualno in tehnološko povsem neobičajnega (novega, tehnološko in konceptualno inovativnega) superračunalnika, ki daleč presega zmogljivosti današnjih, standardnih (uradnih japonskih) tehnoloških zamisli. O tem drugem obisku na Japonskem ni bilo objavljeno nobeno poročilo.

Na osnovi lastnega strateškega ocenjevanja je postajalo čedalje bolj jasno, da bo potrebno kritično preučiti predvsem dva usmerjevalna vidika, in sicer računalnik kot paralelni sistem in njegovo prihodnjo uporabo skozi možnosti, ki jo nakazujejo nove, netradicionalne (ameriške, neevropske) usmeritve umetne inteligence (Center for the Study of Language and Information, Stanford University). Današnja umetna inteligenca je kot programirna dejavnost usmerjena predvsem v dokaj standardne programirne naloge in ji ni mogoče pripisati prilastka inteligence (glej npr. J. Haugeland; Winograd, Flores; in Zeleznikar: Artificial Intelligence Experiences Its Own Blindness). Skozi to kritično perspektivo je bilo potrebno preučiti informacijo z novih, sodobnejših in realnejših vidikov kot informacijsko pojavnost in kot posledice te pojavnosti, ki so bistvene za oblikovanje prihodnjih inteligentnih strojev, inteligentnih programov in inteligentnih razvojnih usmeritev.

Iskra Delta se je odločila za projekt razvoja paralelnega računalniškega sistema (in RISC procesorja) v poletju leta 1986. Projekt se je začel kadrovsko in vsebinsko oblikovati v jeseni leta 1986. Prva, idejna zasnova paralelnega računalniškega sistema je bila izgotovljena pred koncem leta 1986 (glej P. Brajak, seznam slovtva).

S kasnejšimi iteracijami projekta, seminarji skupine raziskovalcev Iskre Delte na oddelkih za računalniške znanosti Massachusetts Institute of Technology (Laboratory for Computer Science, prof. Robert H. Halstead), Newyorske univerze (Ultra Computer Research Laboratory, Courant Institute of Mathematical Sciences, prof. Malvin H. Kalos in prof. Gotlieb) in Univerze v Arizoni (Computer Engineering Research Laboratory, prof. Ralph Martinez in prof. Branko Souček) in s strokovnimi razpravami na vrsti uglednih ameriških univerz (Columbia University, Carnegie Mellon University, Stanford University, UC Berkeley, UCLA, California Institute of Technology, Purdue University,

University of Illinois in Thomas J. Watson Research Center, IBM) je projekt dobil svojo končno obliko in prečiščeno realizacijsko vsebino. Na osnovi prečiščenega koncepta se je lahko začela realizacija konkretne systemske računalniške opreme.

Paralelno z identifikacijo prihodnjega paralelnega računalniškega sistema (Parasys) so bile izvršene tudi uvodne raziskave s področja filozofije, teorije in tehnologije informacije (Železnikar), katerih bistvo je bilo iskanje vprašanj in odgovorov, ki zadevajo prihodnjo usmeritev informacijskih procesov in strojev. Ta del projekta se je tako soočil s prihodnjo uporabo paralelnih računalniških oziroma informacijskih sistemov, s problematiko t. i. informacijskega programiranja in načrtovanja informacijskih strojev.

7. NADALJEVANJE RAZVOJA STANDARDNIH RAČUNALNIŠKIH SISTEMOV IN NJIHOVIH ORODIJ

Oris možnosti razvoja novih tehnologij, paralelnih računalniških sistemov in umetne inteligence seveda ne pomeni, da se ne nadaljuje dosedanji razvoj standardnih računalniških sistemov četrte generacije z vso intenzivnostjo. Iskra Delta razvija in proizvaja skladno s svojo poslovno in tehnološko strategijo mikroročunalniške sisteme (8-, 16- in 32-bitne) z različnimi mikroprocesorji (Zilog, Intel, Motorola, DEC itn.) kot posamične sisteme, delovne postaje in večuporabniške sisteme povezane v mrezo, miniračunalniške sisteme do razreda t. i. supermini sistemov s svojimi lastnimi miniprocesorji in paralelnih sistemov z več procesorskimi enotami. Iskra Delta razvija in proizvaja tudi raznovrstno systemsko in aplikativno programsko opremo.

Pri razvojnem delu sodeluje Iskra Delta stalno in občasno z vrsto domačih in tujih univerz in institucij v ZDA (MIT, NYU, Carnegie Mellon University, Stanford, UCLA, CIT, University of Arizona, Purdue University, T. Watson RC - IBM) in Japonski (ICOT, Tokyo University, Tsukuba University, RIKEN, Sapporo University). To sodelovanje razvija zlasti na področju paralelnih računalniških sistemov, umetne inteligence in novih bazičnih tehnologij (GaAs).

Iskra Delta ima lastno strategijo tehnološkega razvoja, ki vključuje tudi posodabljanje njene proizvodnje, in sicer avtomatizacije, spremljanja proizvodnje in trženja, CIM in CIC in poslovno-tehnološkega informacijskega sistema. Vse to usklajuje z razvejanim sistemom branž, ki so tehnološko odprte v različne dejavnosti računalniške in informacijske uporabe.



Slovstvo

P. Brajak: Designing a Reconfigurable Intelligent Memory Module (RIMM) for Performance Enhancement to Large Scale, General Purpose Parallel Processor. *Informatica* 11 (1987). No. 1, 19-53.

H. L. Dreyfus: *What Computers Can't Do*. Harper and Row, New York, 1972.

H. R. Maturana, F. J. Varela: *Autopoiesis and Cognition: The Realization of the Living*. D. Reidel Publ. Co., Dordrecht, Holland, 1980.

J. Haugeland, Ed.: *Mind Design: Philosophy, Psychology, Artificial Intelligence*. The MIT Press, Cambridge, 1985.

T. Winograd, F. Flores: *Understanding Computers and Cognition: A New Foundation for Design*. Ablex Publ. Corp., NJ, 1986.

A. P. Zeleznikar: Umetna inteligenca: polemični zapis. *Informatica* 9 (1985). St. 3, 87-89.

A. P. Zeleznikar: Od Sappora do Tokia nazaj v Ljubljano. *Informatica* 10 (1986). St. 2, 68-74.

A. P. Zeleznikar: Na poti k informaciji (On the Way to Information). *Informatica* 11 (1987). St. 1, 4-18.

A. P. Zeleznikar: Information Determinations I. *Informatica* 11 (1987). No. 2, 3-17.

A. P. Zeleznikar: Raziskave računalnikov in informacije v naslednjem desetletju. *Informatica* 11 (1987). St. 2, 57-59.

A. P. Zeleznikar: Artificial Intelligence Experiences Its Own Blindness. Proceedings of MIPRO '87, New Generation Computers. Opatija, May 20-22, 1987. *Informatica* 11 (1987). No 3 (in print).

A. P. Zeleznikar: Research of Computers and Information for the Next Decade. Proceedings of MIPRO '87, New Generation Computers. Opatija, May 20-22, 1987. *Informatica* 11 (1987). No. 3 (in print).



A. P. Zeleznikar: Principles of Information. Informatica 11
(1987). No. 3 (in print).

The Research Program at Center for the Study of Language and
Information. Ventura Hall. Stanford University. Stanford, Ca.



VEĆPROCESORSKE ARHITEKTURE

Avtorji referata: mag. Drago Novak, dipl.ing.
mag. Borut Kastelic, dipl.ing.
Dušan Salehar, dipl. ing.
vsi ISKRA DELTA



Ljubljana, maj 1987



V prispevku so opisane klasične večprocesorske arhitekture in programska oprema, ki omogoča uporabo takih sistemov. Podana je klasifikacija in navedeni so cilji pri načrtovanju. Predstavljene so osnovne topološke strukture kot so: skupno vodilo, navzkrižno stikalo in večstopenjske mreže. Programska oprema se deli na tisto, ki omogoča pravilno in konsistentno delovanje sistema in na tisto, ki programerju olajšuje zahtevno delo programiranja takih sistemov. Na koncu je opis nekaj večprocesorskih sistemov, ki jih je mogoče kupiti na trgu. Ta pregled potrjuje, da so večprocesorske arhitekture prešle stopnjo testnih prototipov v laboratorijskih raziskovalnih ustanov in si utrle pot na trg.

1. UVOD

Večprocesorski sistem (multiprocesorski sistem) je sistem, v katerem več procesorjev sodeluje pri reševanju skupnega problema in si pri tem deli skupne vire. Tipični skupni viri so vhodno-izhodne naprave in pomnilnik. Delo procesorjev koordinira skupni operacijski sistem, ki je lahko centraliziran ali porazdeljen.

Procesorji med seboj komunicirajo in se sinhronizirajo pri doseganju skupnih virov. Obseg komunikacije je odvisen od vrste aplikacij, ki jih sistem izvaja, oz. za katere je namenjen. Aplikacije oz. obseg komunikacij pogojujejo arhitekturo sistema. V grobem delimo večprocesorske sisteme v tesno povezane in ohlapno povezane.

- * **Tesno povezani sistemi.** Procesorji imajo dostop do skupnega pomnilnika, ki lahko služi za komunikacijo ali pa celo vsebuje vse programe, ki jih procesorji izvajajo. Pri teh sistemih je obseg komunikacije zelo velik, izvajanje programov pa nadzoruje centralni operacijski sistem. Takim sistemom pravimo tudi pravi multiprocesorji.
- * **Ohlapno povezni sistemi.** Pri teh sistemih imajo posamezne procesne enote se lokalni oz. privatni pomnilnik v katerem imajo večino programov, ki jih izvajajo. Porazdelitev funkcij posamezne aplikacije, ki jo taki sistemi izvajajo je statična. Obseg komunikacij med posameznimi enotami (par procesor - pomnilnik) je manjši. Procesorji komunicirajo med seboj s pošiljanjem sporočil preko klasičnih komunikacijskih kanalov.

Cilji pri načrtovanju

Večprocesorske sisteme gradimo zato, da povečamo zmogljivost, zanesljivost in/ali fleksibilnost. V prvem primeru povečujemo odzivni čas sistema. Zmogljivost se veča z dodajanjem novih procesorjev. Zanesljivost oz. integriteta sistema označuje stopnjo neobčutljivosti na izpade in napake v sistemu. Osnovni pogoj za povečanje zanesljivosti sistema je redundantnost (več elementov iste vrste).



V večprocesorskih sistemih, ki jih ponavadi gradimo s povezovanjem več procesorskih, pomnilniških in vhodno-izhodnih modulov, je stopnja redundantnosti na vsak način višja kot pri monoprocessorskih sistemih. Zato lahko pri večprocesorskih sistemih dosežemo boljšo zanesljivost delovanja, če si to postavimo za cilj. Fleksibilnost sistema se odraža v možnostih modificiranja in širjenja sistema. Pri širjenju sistema gre za dodajanje novih elementov, ki ima za posledico povečanje zmogljivosti sistema pri istem naboru funkcij ali povečanje števila funkcij. Fleksibilnost se odraža na tem, kako lahko prilagodimo sistem raznim aplikacijam brez drastičnih posegov v arhitekturo sistema. Pri večprocesorskih sistemih je ta fleksibilnost lahko zelo velika.

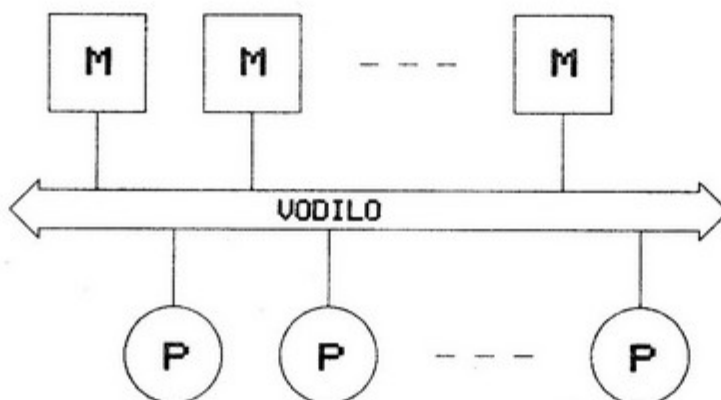
2. TOPOLOGIJE

Večprocesorske sisteme klasificiramo po tem, kako so povezani procesorski, pomnilniški in vhodno-izhodni moduli. Ta podsistem povezav (topologija) neposredno vpliva na zmogljivost, fleksibilnost in integriteto. Najpogostejše topologije so:

- * skupno vodilo
- * navzkrižno stikalo (crossbar)
- * večstopenjske povezave

2.1. SKUPNO VODILO

Skupno vodilo je najenostavnejši način povezave procesorskih in pomnilniških modulov v večprocesorski sistem (slika 1). Vodilo sestavljajo: podatkovno, naslovno in kontrolno vodilo. Samo vodilo in pomnilniški moduli predstavljajo pasivni del sistema. Aktivni so procesorski moduli, ki si delijo pasivni del.



M = pomnilnik
P = procesor

Slika 1.

Procesorji torej tekmujejo med seboj za dostop do skupnega vodila. V vsakem trenutku lahko namreč uporablja skupno vodilo in s tem ostali pasivni del sistema samo en procesor. Dostop do vodila urejuje kontroler oz. arbiter, ki je lahko centraliziran (poseben modul) ali porazdeljen (na procesorskih modulih).

S širjenjem sistema, dodajanjem novih procesorskih modulov, se večja število hkratnih zahtev po vodilu in vsak posamezni procesor dalj časa čaka na vodilo.

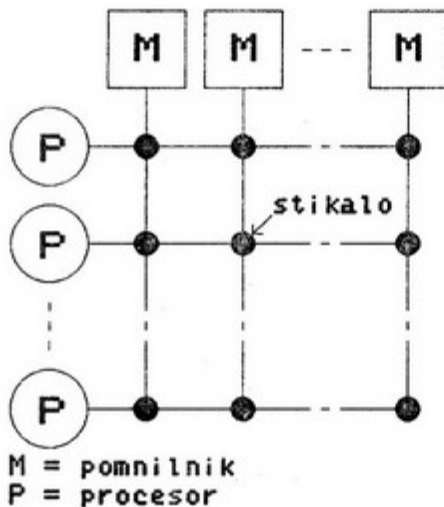
S stališča procesorja se to odraža kot podaljševanje časa dostopa do pomnilnika.

Vodilo nas torej omejuje pri širjenju sistema, saj z dodajanjem novih procesorjev upočasnjujemo delovanje posameznih procesorjev v sistemu. Skupno vodilo je primerno za manjše sisteme. Kljub temu pa je zaradi enostavnosti, nizke cene in zanesljivosti ena najčešče uporabljenih arhitektur.

Ozko grlo, ki ga predstavlja skupno vodilo, omilimo z dodajanjem novih vodil (sistemi z več skupnimi vodili) ali z uvedbo privatnih cache pomnilnikov.

2.2. NAVZKRIŽNO STIKALO

Z dodajanjem novih vodil v sistem, uvajamo paralelne poti in s tem zmanjšujemo število konfliktov. Navzkrižno stikalo je mreža vodil, ki vodijo do vseh pomnilnikov in procesorjev (slika 2).



Slika 2,

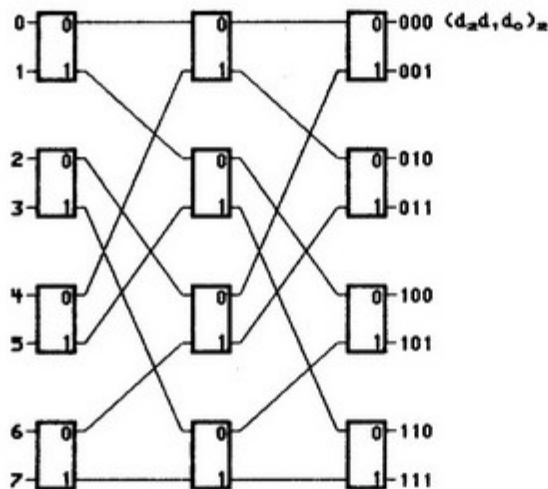


Na presečiščih so stikala. Ta stikala so dokaj kompleksni elementi, saj preklaplajo cela vodila in se rešujejo konfliktna situacije, ko je več zahtev za dostop do istega pomnilnika.

Velika prednost stikala pred skupnim vodilom je visoka propustnost. Matrika stikal omogoča paralelne poti med pari procesor - pomnilnik. Z dodajanjem procesorjev se torej hitrost delovanja posameznega procesorja znotraj sistema praktično ne zmanjša. Sistem se da siriti do dimenzij matrike na zelo enostaven način. Če porazdelimo arbitražo, ki je sicer implementirana v stikalih matrike po pomnilniških modulih, dobimo novo arhitekturo oz. sisteme s pomnilniki z več vrati (multiport memories).

2.3. VEČSTOPENJSKE POVEZAVE

Navzkrižno stikalo postane pri večjem številu procesorjev zelo kompleksno in drago. Če popolnoma povezano strukturo stikala nadomestimo z delno povezano (nekatera stikala spustimo) in gradimo povezovalni sistem večstopenjsko in modularno, dobimo naslednjo tipično topologijo, ki je primerna za sisteme z velikim številom procesorjev. Ena izmed takih povezav je "banyan" mreža (slika 3). Tako mrežo sestavljajo majhna navzkrižna stikala 2×2 ali podobno, ki so v več nivojih med seboj povezana tako, da obstoja točno ena pot med poljubnim parom procesor - pomnilnik. V Banyan mreži raste število stikal počasneje kot pri navzkrižnem stikalu (pri $n \times n$ dimenziji $n \log n$ za banyan in n^2 za navzkrižno stikalo).



Slika 3.

2.4. CACHE POMNILNIK

Cache pomnilnik je zelo hiter pomnilnik, ponavadi manjše kapacitete, v katerem so shranjeni ukazi in podatki, ki so jih procesorji nazadnje uporabljali oz. izvajali.

Ker so v programih pogosto zanke in ker programi procesirajo v posameznem delu samo del podatkov, je verjetnost, da bomo lahko uporabili podatke ali ukaze iz cache pomnilnika zelo velika. Taki ponovni uporabi rečemo zadetek.

V sistemih s skupnim vodilom zelo zmanjšamo promet na vodilu, če procesorskim modulom dodamo cache pomnilnik.

Procesorji pogosto najdejo podatek oz. ukaz v svojem privatnem cache pomnilniku (zadetek) in zato takrat ne obremenjujejo skupnega vodila. Ker lahko ima več procesorjev shranjen isti podatek v lokalnem cache pomnilniku je potrebno zagotoviti koherenco oz. konsistenco podatka v sistemu, ko en procesor spremeni ta podatek.

Moderni cache kontrolerji, ki so implementirani v VLSI tehnologiji zagotavljajo tudi koherenco cache pomnilnikov. Cache pomnilniki so uporabni tudi v drugih arhitekturah v glavnem zato, da omilijo počasnost pomnilniških modulov.

3. VEĆPROCESORSKI OPERACIJSKI SISTEMI IN JEZIKI

Razlika med večprocesorskimi operacijskimi sistemi in enoprocessorskimi operacijskimi sistemi, ki podpirajo multiprogramiranje, je v dodatni kompleksnosti, ki zahteva več simultano delujočih procesorjev. Operacijski sistemi za multiprogramirane enoprocessorske sisteme zahtevajo regularno uporabo skupnih virov, zaščito pomnilnika in podatkov, preprečevanje mrtvih točk (dead lock) in nenormalne zaključitve procesov. Većprocesorski operacijski sistemi, ki podpirajo več asinhronih sočasno izvajajočih se procesov, pa morajo poleg omenjenega se zagotavljati učinkovito izkoriščanje skupnih virov.

3.1. KLASIFIKACIJA VEĆPROCESORSKIH OPERACIJSKIH SISTEMOV

Prisotnost več tesno povezanih procesorskih enot v računalniškem sistemu pogojuje novo dimenzijo v izvedbi operacijskih sistemov. V grobem delimo operacijske sisteme na sisteme s centralizirano in sisteme s porazdeljeno kontrolo.

Centralizirana kontrola je taksna kontrola, pri kateri odloča o vsakem skupnem viru en sam objekt. Status sistema je dostopen vsem procesorjem in je vedno ažuren.



Porazdeljena kontrola predstavlja ravno obraten pristop. Vsak skupni vir kontrolira več objektov. V sistemu ni globalne baze podatkov. Na več mestih se formirajo lokalne baze, ki so bolj ali manj neazurne. Končne odločitve se formirajo po večinskem ali kakšnem drugem principu.

Realizirani večprocesorski operacijski sistemi delujejo v glavnem na principu centralizirane kontrole. Delimo jih na tri temeljne konfiguracije: gospodar-suženj, ločeni izvršniki in plavajoči izvršniki.

- * V sistemih gospodar-suženj (master-slave) je izvršnik vedno v istem procesorju. Ostali procesorji so sužnji, ki prejemajo naloge od gospodarja ter ga kličejo za vsako uslugo. Take sisteme je najenostavnejše implementirati, vendar pa so s stališča integritete sistema najneugodnejši. Izpad procesorja, ki izvaja izvršnik, povzroči izpad celotnega sistema.
- * V sistemih z ločenimi izvršniki (separate supervisor) ima vsak procesor svoj izvršnik, ki servisira lokalne zahteve. Tak sistem je razbit na bolj ali manj avtonomne enote s procesorjem pomnilnikom in V/I napravami. Seveda obstaja tudi interakcija med procesorji. Vsak izvršnik ima svoj lokalni nabor tabel, nekaj tabel pa je skupnih in so dostopne celotnemu sistemu. Ta sistem je v primerjavi s sistemom gospodar-suženj mnogo manj občutljiv na napake.
- * Sistemi s plavajočim izvršnikom (floating supervisor) tretirajo vse procesorje kot anonimno množico virov. Izvršnik ni fizično razpršen po sistemu, niti ga ne izvaja en sam vnaprej določeni procesor. Procesorji si med seboj podajajo vlogo gospodarja, ki jo prevzamejo, ko izvajajo kakšno vlogo izvršnika. Večina rutin izvršnika mora omogočati ponovni vstop, tako da jo lahko izvaja več procesorjev hkrati. Ta organizacija vsebuje prednosti obeh prej naštetih in je zelo ugodna s stališča integritete sistema, ker ob izpadu posameznega procesorja, sistem še vedno deluje.

3.2. UNIX V VECPROCESORSKIH SISTEMIH

Klasična zasnova operacijskega sistema UNIX sloni na enoprocorski arhitekturi, ki jo sestavljajo centralna procesna enota, pomnilnik in periferija. Večprocesorska arhitektura vsebuje dve ali več procesnih enot, ki si dodeljujejo skupni pomnilnik in periferijo. Potencialno je tak računalnik zmogljivejši, ker tečejo procesi sočasno na različnih procesorjih.



Pri prenosu operacijskega sistema UNIX na večprocesorsko arhitekturo, se pojavijo problemi v zvezi z zaščito podatkov in sinhronizacijskimi mehanizmi. Na izbiro sta dve možnosti:

- * izdelava popolnoma novega operacijskega sistema v obstoječem ali novem jeziku ali
- * adaptacija obstoječega operacijskega sistema za novo arhitekturo sistema.

Operacijski sistem UNIX in njegovo okolje delimo na več plasti: materialna oprema, jedro, sistemski klici, školjka, podporni programi in uporabniški programi.

Ker ne želimo adaptirati obstoječih podpornih in uporabniških programov pri prenosu na večprocesorske sisteme, moramo ohraniti enake sistemske klice. Zato so izvedene vse spremembe in dopolnitve na jedru operacijskega sistema.

Najenostavnejši je prenos UNIXa na konfiguracijo gospodar-suženj, kjer izvaja jedro samo procesor gospodar. V tem primeru se pojavi nova kritična sekcija v večprocesorskih sistemih le v razvrščevalniku, ker le-ta nima zaščite proti možnosti izvajanja enega procesa na dveh procesorjih hkrati. Naprimer v večprocesorskem sistemu z gospodarjem in dvema sužnjema je možno, da najdeta oba sužnja istočasno isti proces, ki je pripravljen za izvajanje. Ta problem je rešljiv na dva načina: gospodar lahko določi kateri suženj bo izvajal določeni proces ali pa omogoči istočasno le enemu sužnju, da izvaja razvrščevalno zanko.

Druga metoda prenosa operacijskega sistema UNIX na večprocesorsko konfiguracijo je omogočenje izvajanja jedra na vseh procesorjih. V jedru je potrebno definirati kritične sekcije in jih zaščititi, tako da lahko izvaja kritično sekcijo samo en procesor istočasno. Zaščita kritičnih sekcij se izvede s semaforji. Dejansko ni problem zamenjave zaščitnih mehanizmov, temveč odkrivanje kritičnih sekcij jedra, ki se pojavijo v večprocesorskih sistemih.

Omenjene implementacije multiprocesorskih UNIX sistemov so posplošene za poljubno število procesorjev, toda zavedati se moramo, da se zmogljivost sistema ne povečuje linearno s številom procesorjev. Zaradi dodeljevanja skupnih virov, zaradi povečevanja verjetnosti, da procesor naleti na blokiran semafor ali zaradi tega, ker izvaja jedro samo en procesor, so zmogljivosti večprocesorskih sistemov dokaj omejene. Realizirane rešitve večprocesorskih UNIX sistemov niso prinesle najboljših rezultatov.

3.3. PRISTOPI K IZRABI PARALELIZMA

Paralelni program za večprocesorski sistem sestoji iz dveh ali več interaktivnih procesov. Proces je sekvenčni program, ki se izvaja sočasno z ostalimi procesi. Za izdelavo paralelnih programov poznamo dva pristopa: eksplicitni in implicitni.

Eksplicitni pristop omogoča programerju, da sam določi paralelne sekvence z uporabo določenih jezikovnih konstruktov. V drugem primeru imamo implicitno sočasnost. Tukaj določi prevajalnik kaj se lahko izvaja sočasno. Ta primer je predvsem primeren za "data-flow" računalnike.

Ena možnost označitve sočasnosti v eksplicitnem načinu je z uporabo FORK in JOIN stavkov. FORK stavek kreira nov proces in JOIN stavek čaka, da se predhodno kreirani proces zaključi. S kombinacijo teh stavkov lahko kreiramo poljubno število sočasnih procesov in izvedemo sinhronizacijo med njimi.

Razširitev FORK-JOIN koncepta je bločno-strukturiran jezik, ki ga je predlagal Dijkstra. Vsak proces iz nabora procesov S_1, S_2, \dots, S_n se lahko izvaja sočasno z uporabo naslednjega "cobegin-coend" konstrukta:

```
begin
  S0;
  cobegin S1; S2;... Sn; coend
  Sn+1;
end
```

V tem primeru se izvaja blok stavkov med "cobegin" in "coend" sočasno sele po izvršitvi stavka S_0 . Stavek S_{n+1} se izvede sele potem, ko se zaključijo vsi stavki S_1, S_2, \dots, S_n .

4. KOMERCIALNI VEČPROCESORSKI SISTEMI REALIZIRANI V MIKROPROCESORSKI TEHNOLOGIJI

4.1. ALIANT COMPUTER SYSTEMS CORPORATIONFX/8

FX/8 je posebna izvedenka večprocesorskih sistemov. Ima vektorski ukazni nabor, pri čemer so ukazi za paralelno procesiranje že vsebovani v strojni opremi. Njegov fortran prevajalnik avtomatično določa konverzijo obstoječe fortranske kode v kodo, ki se izvaja paralelno.

Do 12 interaktivnih procesorjev (IP) izvaja programe na osnovi en program en procesor, medtem ko se lahko vseh 8 računskih procesorjev (computational elements = CE) združi v delu nad enim programom.

Interaktivni procesorji so standardni Motorolini MC68012, medtem ko so računski procesorji lastna 64 bitna "gate array" vezja.



Hrbtenica sistema FX/8 je lastno vodilo, ki omogoča teoretični prenos do 376 Mzlogov na sekundo. Vsi procesorji si delijo globalni pomnilnik, preko skupnega "cache" sistema in vodila. "Cache" koherenca je prozorna.

Najpomembnejša lastnost sistema je, da prehaja izvajanje programa iz zaporednega načina v paralelni povsem samodejno, s pomočjo enega samega ukaza. Paralelno se izvajajo deli programa, ki se ne morejo izvajati v vektorskem načinu. Odkrivanje paralelizma v programu je delo prevajalnika, nadzor izvajanja brez odvečnega "overheada" pa stvar sistema.

V maksimalni konfiguraciji zmore FX/8 do 36 MIPS in 94 MFLOPS. Bolj relevantni so rezultati LINPACK primerjalnega testa, ki podajajo 1.78 MFLOPS z enim procesorjem in 8.4 MFLOPS pri polni konfiguraciji.

Sistem omogoča programiranje v C, pascalu in fortranu, pri čemer je fortran edini, ki avtomatično prevaja program v kodo, ki se lahko paralelno izvaja. Fortran prevajalnik je enak DECovemu VAX/VMS fortranu, tako da se lahko obstoječi programi samo prevedejo.

Operacijski sistem je predelan UNIX 4.2, imenovan tudi Concentrix.

4.2. ELXI : SISTEM 6400

Sistem 6400 je vektorski procesni sistem sestavljen iz dvanajstih 64-bitnih procesorjev.

Napovedana zmogljivost je 72 MIPS in 79.2 MFLOPS pri maksimalni konfiguraciji. Podatkovno vodilo omogoča 320 Mzlogov na sekundo. Glavni pomnilnik pa obsega do 192 Mzlogov.

Operacijski sistem je predelan osnovni kernel UNIX System V.2 operacijskega sistema, pri čemer trije operacijski sistemi (UNIX 4.3, UNIX System V.2 in EMBOS) paralelno delujejo. Sistem bazira na izmenjavi sporočil.

Vsak procesni element ima vgrajene razvrševalne funkcije in funkcije za delo s sporočili v strojno-programski opremi.

Arhitektura omogoča, da se samostojni sekvenčni procesi pogovarjajo med seboj s pomočjo sporočil. Tak pristop je zelo različen od drugih večprocesorskih arhitektur, ki delujejo na osnovi semaforjev pri medprocesorski komunikaciji.



4.3. FLEXIBLE COMPUTER CORPORATION: FLEX/32

Posebne² v večprocesorski arhitekturi je tudi Flex/32 sistem. Procesorji so med seboj lahko tesno ali pa ohlapno povezani.

Druge večprocesorske arhitekture uporabljajo skupni pomnilnik za medprocesorske komunikacije in tudi za izvajanje programa. Flex/32 pa uporablja lokalni pomnilnik za program, skupni pomnilnik in podatkovna vodila pa predvsem za medprocesorske komunikacije.

Sistem je namenjen za uporabo že obstoječih programov, brez spreminjanja kode. Kljub temu pa Flex/32 nudi dokaj velik nabor orodij: UNIX System V za razvoj programske opreme, simulator paralelizma za testiranje paralelnih aplikacij, paralelni C, fortran itd.

V Flex/32 se lahko vgradi do 20 procesorjev tipa National NS32032 in Motorola MC 68020. Vsak ima svoj lokalni pomnilnik in vhodno/izhodno naprave, povezani pa so med seboj s sinhronim podatkovnim vodilom. Posamezni kabineti so lahko med seboj povezani tudi preko VME podatkovnega vodila.

4.4. SEQUENT COMPUTER CORPORATION: BALANCE 8000, BALANCE 21000

Večprocesorska sistema Balance 8000 in 21000 lahko izvajata obstoječe UNIX aplikacije brez sprememb. Prav tako pa se jih lahko preprogramira v paralelno izvajanje. Obstajajo programska orodja za testiranje paralelnih aplikacij, prav tako tudi druga orodja kot na primer paralelna implementacija Ada prevajalnika.

Kot pri večini sistemov je UNIX jedro spremenjeno. Pri balance sistemih je jedro UNIX 4.2 ali UNIX System V kompatibilno.

V sistem je lahko vgrajenih do 12 32032 procesorjev (Balance 8000) ali pa do 30 32032 (Balance 21000).

4.5. ENCORE COMPUTER CORPORATION: MULTIMAX

Večprocesorski sistem Multimax ima dve podatkovni vodili. Nanobus (100 Mzlogov na sekundo) povezuje procesorske pare in globalni pomnilnik. Drugo vodilo je lokalno in povezuje med seboj po dva procesorja in 32 Kzložni cache pomnilnik.

Tesno povezana arhitektura Multimaxa omogoča vsem procesorjem in programom dostop do globalnega pomnilnika in vhodno/izhodni periferiji. Skupni pomnilnik je dodeljen procesom, ne pa procesorjem.

Operacijski sistem UMAX je kompatibilen z Unix System V in UNIX 4.2. Zaradi Unix kompatibilnosti lahko večina obstoječih aplikacij dela brez sprememb v izvornem programu.



4.6. CONCURRENT COMPUTER SYSTEMS: 3200MP

Večprocesorski sistem 3200MPS ohranja shemo gospodar-suženj. Sistem je sestavljen iz lastne 3250XP procesne enote, ki je gospodar in do devet pomožnih procesnih enot.

Imajo lasten operacijski sistem OS/32, katerega particije se nahajajo v pomožnih procesnih enotah, sam gospodar pa ima celo kopijo operacijskega sistema.

Naloga gospodarja je, da razporeja čakajoče vrste, vzdržuje sistem, in razbremenjuje preobremenjene pomožne procesne enote. Pomožne procesne enote razvrščajo procese in skrbijo za vhodno-izhodno obdelavo.

Polna konfiguracija omogoča 22 MIPS in 60 MWhetstone ukazov.

5. ZAKLJUČEK

Področje večprocesorskih sistemov je bilo zadnjih petnajst let zelo atraktivno raziskovalno področje. Razvoj mikroprocesorske tehnologije in zahtevnost aplikacij pa sta poskrbela, da se je preselilo v razvojne laboratorije. Kompleksnost, ki je bila doslej vedno opravičilo, zakaj določena arhitektura ni implementirana, ne predstavlja več take omejitve. Zato lahko pričakujemo, da bomo na trgu vse česče srečevali večprocesorske sisteme, ki pa bodo seveda še vedno omejeni na specialne aplikacije.

6. LITERATURA

- * Philip H.Enslow jr., Multiprocessor Organization - A Survey, Computing Surveys, Vol.9, No.1,1977
- * Kai Hwang, Faye A.Briggs, Computer Architecture and Parallel Processing, McGraw Hill 1985
- * Drago Novak, Sistemi z več procesorji, Informatica, st.3, 1980



**PARALELNO PROCESIRANJE -
PARSYS PROJEKT**

Avtorji referata: mag. Peter Brajak, dipl.ing.
dr. Saša Prešern, dipl.ing.
mag. Lojze Vogel, dipl.ing.
Peter Hiti
vsi ISKRA DELTA



Ljubljana, maj 1987

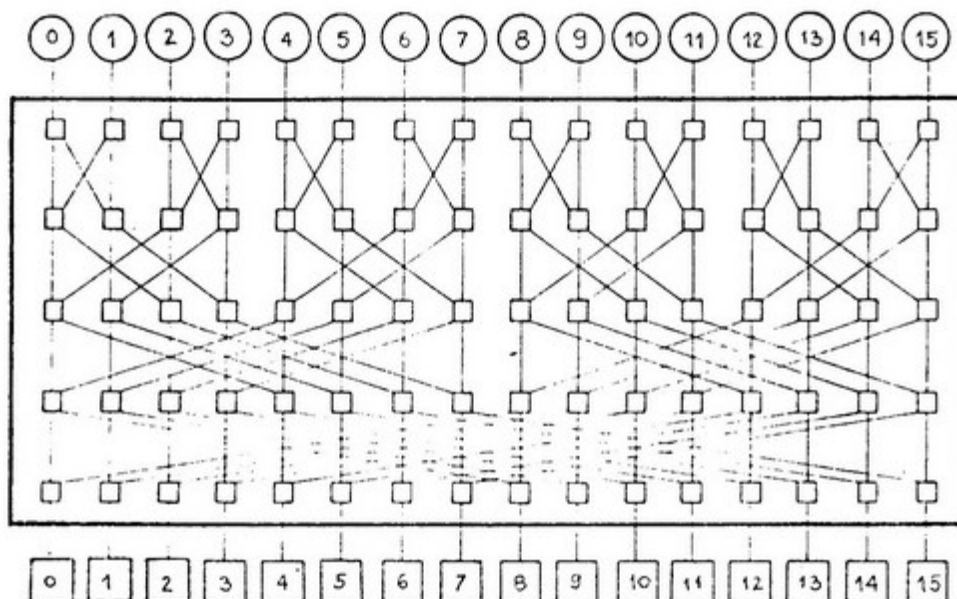


SAZETAK: PARSYS je istraživačko razvojni projekt na Iskra Delti s ciljem razvoja potpuno spregnutoga MIMD tipa paralelnog sistema. Prototipna arhitektura sistema je zasnovana na principu zajedničke memorije i sastojati će se iz 64 procesorska modula i 256 memorijskih jedinica povezanih preko mreže samostalno razvijenih inteligentnih sklopova ISMM (Intelligent Shared Memory Module) u topološkoj konfiguraciji šest dimenzijske kocke. Pored toga da ISMM-i omogućavaju brzi dostup do zajedničke memorije, oni izvršavaju funkcije koje rješavaju odnosno minimiziraju probleme karakteristične za paralelno procesiranje.

U ovom radu, prikazani su koncepti PARSYS projekta, odnosno nacrt ISMM elementa, kao najvažnijeg dijela sistema. Prikazano je kako ISMM-i omogućavaju većem broju procesora čitanje iste adrese u jednom memorijskom ciklusu, kako su minimizirani memorijski sudari i kako ISMM-i uspostavljaju listu blokiranih procesa u samo jednom memorijskom ciklusu.

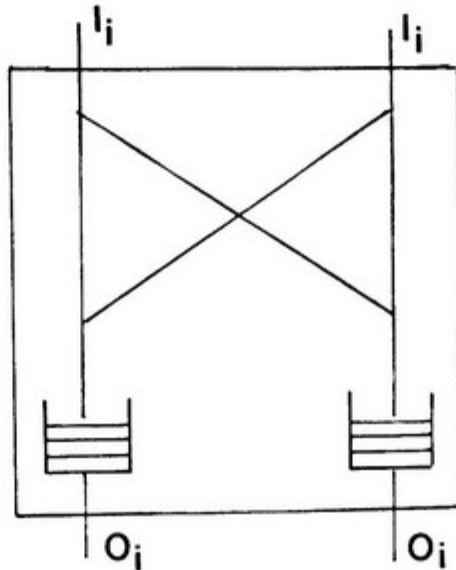
1. ARHITEKTURA PARSYS SISTEMA

Rezultati zadnjih svjetskih istraživanja na području paralelnog procesiranja /BBN86, IBM85, IBM86, NYU82, CED86/, kao i predviđanja da će postojeći računarski sistemi dostići svoj limit u smislu količine procesne snage, ohrabрили su istraživačko razvojni rad MIMD višeprocorskog sistema sa zajedničkom memorijom u Iskri Delti. Koncepti i filozofsko opredjeljenje PARSYS sistema je prikazano u /BRA87a/. Sistem se izvorno sastojao od 64 procesora i 64 memorijskih modula povezanih sa višestepenom mrežom tipa Banyan (slika 1) i povezujućim sklopovima kao na slici 2.



slika 1





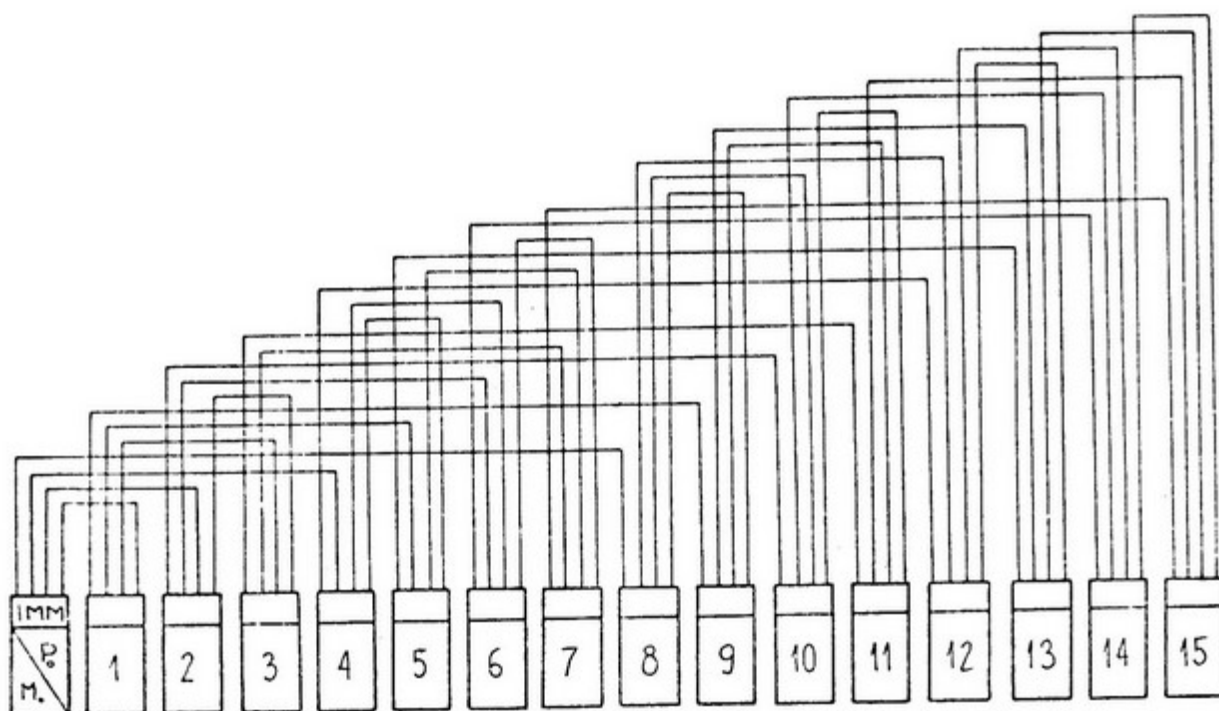
slika 2

Pokazalo se je, da takav tip mreže, odnosno tip sklopova ima četiri problema:

1. broj ulazno/izlaznih zica po sklopu je prevelik (osam ulazno/izlaznih vrata),
2. broj ulazno/izlaznih zica po tiskanoj ploči je prevelik,
3. organizacija sistema je kompleksna budući da su procesorski i memorijski moduli odvojeni.
4. protok paketa (procesorskih zahtijeva) kroz sklop je premalen i može prouzročiti usko grlo i zastoje.

Zbog toga je prelaz višestepene mreže u višestepenu kocku predstavljao originalnu ali i neminovnu transformaciju (slika 3). S takvom transformacijom su sklopovi postali RU (Routing Unit), svaki od njih sa samo šest ulazno/izlaznih vrata.





slika 3

RUi ne služe samo kao povezujući elementi između procesora i memorijskih modula, već izvode funkcije za poboljšanje karakteristika paralelnog procesiranja kao što su:

- * višeprocorsko čitanje/pisanje iste adrese u jednom ciklusu,
- * višeprocorsko čitanje/pisanje različitih adresa u jednom ciklusu,
- * brzi prijenos velikog broja memorijskih zahtjeva kroz mrežu,
- * brzu sinhronizaciju procesa,
- * brzu izmjenu procesa na procesorima (context switch).

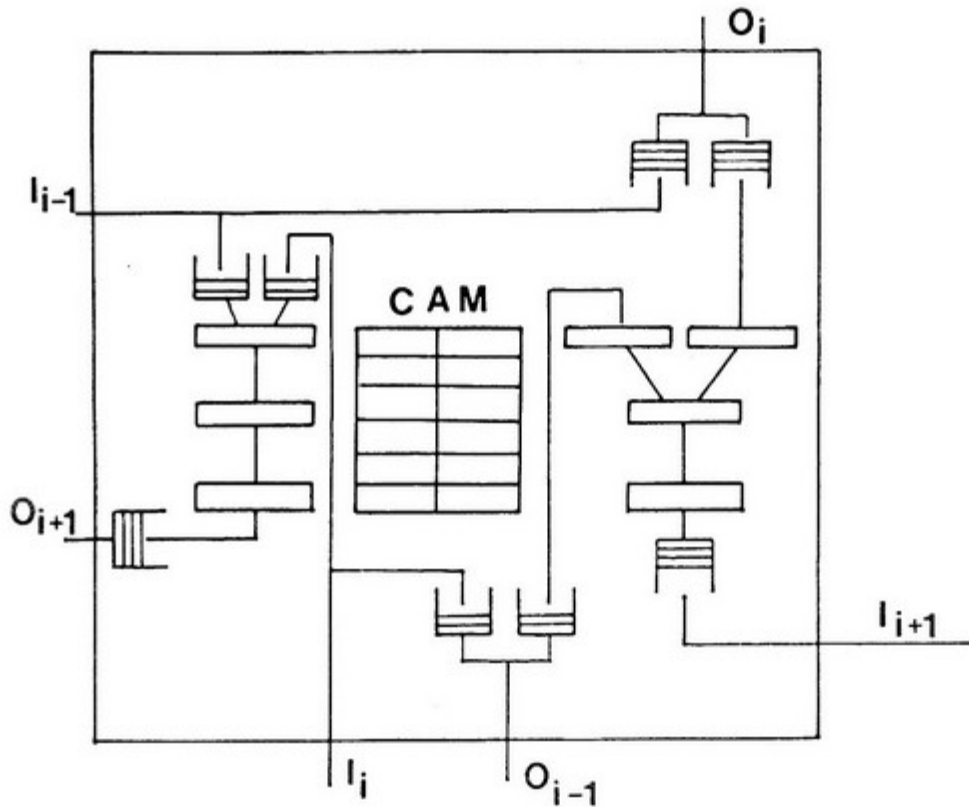
2. ISMM FUNKCIJE (PARSYS PRISTUP)

ISMM funkcije se mogu podijeliti u PreRU (pred RU), RU i PostRU (poslije RU) funkcije. RU funkcije su najznačajnije i rješavaju probleme 1,3,4,5. U 64 procesorskom prototipu svaki ISMM ima 6 RU.



2.1 RU ARHITEKTURA

Arhitektura RU je distribuirana i paralelna (pipeline) tako da omogućava brzi prijenos velikog broja zahtjeva kroz mrežu (slika 4). Na slici se vidi da su putevi zahtjeva (I_i, I_{i-1}) i putevi odgovora (I_{i+1}) odvojeni. Svaki RU ima CAM (Content Addressable Memory) i osam FIFO modula koji omogućavaju da paketi asinhrono i brzo prolaze kroz RU. Pokazano je da takva RU organizacija omogućava 4-5 puta bolju iskoristivost nego kod sklopova na slici 2.

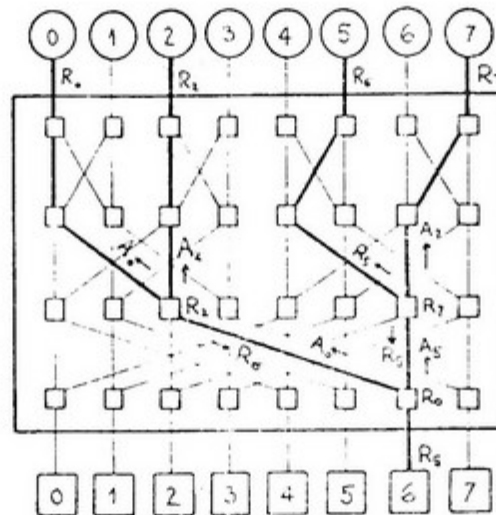


slika 4



2.2 MEMORIJSKI KONFLIKTI - ISTA LOKACIJA

Jedna od najvećih karakteristika PARSYS sistema je da omogućava dva ili više procesora simultano čitanje iste memorijske adrese. To je moguće zbog toga jer se putevi kroz koje prolaze dva memorijska zahtijeva križaju u jednom RU. Ako dva procesora zatraže isti podatak istovremeno, dva zahtijeva se sretnu u jednom RU. Prvi zahtijev se pohrani u CAM, i proslijedi naprijed. Drugi zahtijev ostane u RU. Na povratku, kada se odgovor vrati u taj isti RU, logika RU pošalje odgovor k oba procesora. Shematski prikaz simultanog čitanja je dan na slici 5. Slovo R prestavlja memorijski zahtijev (request), slovo A prestavlja odgovor (answer).



slika 5

2.3 MEMORIJSKI KONFLIKTI - ISTI MEMORIJSKI MODUL

Dva su oblika preslikavanja logičnih, programsko definiranih adresa u fizične adrese kod sistema sa većim brojem memorijskih modula:

1. jedan na jedan preslikavanje

Prva adresa je prva lokacija u prvom memorijskom modulu, druga adresa je druga lokacija u prvom memorijskom modulu, N-ta adresa je N-ta lokacija u prvom modulu, N+1 adresa je prva lokacija u drugom modulu, itd.

2. "interleaving"

Prva adresa je prva lokacija u prvom modulu, druga adresa je prva lokacija u drugom modulu, N-ta adresa je prva lokacija u N-tom modulu, N+1 adresa je druga lokacija u prvom modulu, itd.



Simulacijski rezultati su pokazali da oba preslikavanja adresa nisu prikladna za paralelno procesiranje. Konkretno, kada N procesora istovremeno zahtijeva N različitih memorijskih adresa, 40% memorijskih modula je neaktivno, a 60% memorijskih modula zbog toga ima konflikte. Kod sistema sa "cache" memorijama, kao što je PARSYS, taj odnos je još slabiji: 50% memorijskih modula je neaktivno.

Kod PARSYS projekta, je po prvi put u arhitekturi računara primjenjeno potpuno promješano preslikavanje adresa: adresa jedan je npr. treća lokacija drugog modula, adresa dva je stoprva lokacija dvadesetoga modula, itd. U primjeru takvog preslikavanja adresa, moguće je izračunati vjerojatnoću iskoristivosti memorijskih modula. Tako, vjerojatnoća $P(k_1=K_1, \dots, k_n=K_n)$, je vjerojatnoća koja kaže koliko memorijskih modula ima 1 zahtijev, koliko memorijskih modula ima 2 zahtijeva, itd., a definirana je s izrazom (i).

$$P(k_1=K_1, k_2=K_2, \dots) = \frac{\prod_{i=1}^{i=S} \binom{M - \sum_{j=1}^{i-1} K_j}{K_i} \prod_{r=0}^{K_i-1} \binom{N - \sum_{j=1}^{i-1} jK_j - ir}{i}}{M^N}$$

izraz (i)

Ohrabrujući rezultati se pojavljuju kod $M=4N$, odnosno kad je broj memorijskih modula 4 puta veći od broja procesora. Vjerojatnoća da više od dva zahtijeva dođu u isti memorijski modul je 0,002. Također, 98% memorijskih modula ima 1 zahtijev, 2% imaju dva zahtijeva, što u pravom smislu predstavlja bezkonfliktni memorijski sistem.

Implementacija potpuno promješanoga preslikavanja je matematički vrlo kompleksan problem, u PARSYS projektu je to riješeno na vrlo originalan, jednostavan i brz način što je i prikazano u /BRA87b/.

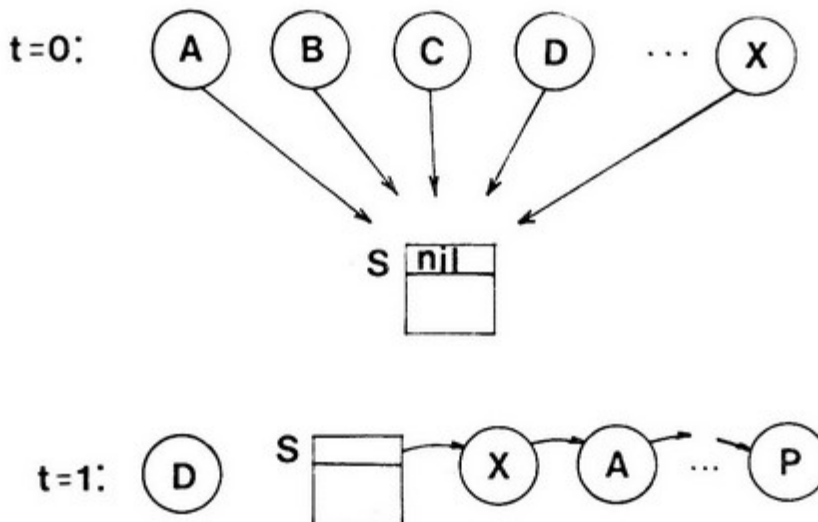


2.4 SINHRONIZACIJA PROCESA

Sinhronizacija procesa predstavlja jedan od najvažnijih koncepta paralelnoga procesiranja. Pitanje je kako uskladiti procesno komuniciranje i omogućiti pravilan rad višeprocorskog sistema.

U literaturi postoje različiti sinhronizacijski mehanizmi, od jednostavnih Test and Set mehanizama, preko P i V semafora, do vrlo kompleksnih Fetch and Add. Svi ti mehanizmi pokušavaju uspostaviti rep čekanja i koordinirati rad više procesa. Drugim riječima, ako N procesa istovremeno želi dostup do određene zajedničke strukture, jedan od N procesa dobije pravo korištenja te strukture, ostali se postavljaju u rep čekanja. Kod mehanizama kao što su P i V semafori, Test and Set, je problem brzina, odnosno, N procesa se u najboljem primjeru postavi u vrstu čekanja u N memorijskih ciklusa. Na drugoj strani, Fetch and Add omogućava uspostavu repa čekanja u jednom memorijskom ciklusu, ali je zato implementacijski teško izvediv, zahtijeva dodatni hardware i što je najveći problem, radi samo kad su ulazni zahtjevi u RU istovremeni što nije slučaj u realnom procesiranju.

U sklopu PARSYS projekta razvijen je novi sinhronizacijski mehanizam: Get and Link. Ideja Get and Link-a je da se N procesa poveže u listu čekanja u samo jednom memorijskom ciklusu. Implementacijski, Get&Link je jednostavan kao i P i V semafori, koncepcijski je brz kao Fetch and Add. Na slici 6 vidimo procese A,B,C,...,Z koji napadaju skupni objekt kontroliran sa semaforom S. U samo jednom memorijskom ciklusu, proces D prvi dolazi do semafora S, osvaja zajednički objekt, a ostali procesi dobivaju vezu (pointer) na proces u vrsti čekanja ispred njega.



slika 6



Get & Link sinhronizacijski mehanizam je implementiran koristeći CAM element u RU. Njegove prednosti u odnosu na Fetch and Add su slijedeće:

- 1) Get & Link je jednostavniji implementirati,
- 2) Get & Link je dinamičan,
- 3) Get & Link ne zahtijeva istovremene ulaze u RU.

2.4 IZMJENA PROCESA ("CONTEXT SWITCH")

Izmjena procesa na procesoru je usko povezana sa sinhronizacijskim problemima. Do izmjene dolazi kada proces nije sposoban nastaviti izvođenje svog zadatka, jer mu to sinhronizacijski mehanizam ne dopušta.

Izmjena procesa kod PARSYS sistema je implementirana s pomoću PreRU logike. Svaki PreRU promatra listu spremnih procesa, neodvisno od rada procesora, odnosno procesa na tom procesoru. PreRU dogradi (donese) dinamički dio prvog spremnog procesa (PC, SP, registri, itd.) u svoju lokalnu memoriju. Kada dođe do izmjene procesa, procesoru nije potrebno više puta ići kroz mrežu da dogradi dinamičan dio nekog spremnog procesa, već je taj dio u njegovom lokalnom memorijskom modulu.

3. ZAKLJUČAK

Tri su pristupa studirana kao mogućnost PARSYS projekta, a predstavljaju karakteristične arhitekture paralelnog procesiranja:

- * masivno paralelni VLSI specifični sistemi /CM85, DAD87, CH81/,
- * data-flow /DF87/,
- * MIMD sa zajedničkom memorijom /BBN87, IBM85, IBM86, CED83, NYU83/.

VLSI specifični sistemi nisu privlačni zbog svoje procesorske kompleksnosti (posebni procesori) i visoke cijene.

Data-flow sistemi predstavljaju koncepcijski najbolje paralelne mašine, ali implementacijski su neizvedivi s današnjom tehnologijom.

PARSYS je MIMD sistem s zajedničkom memorijom, a rezultati drugih, i naši originalni koncepti ulijevaju nadu u njegov uspjeh.

Smatramo da je ISMM arhitektura superiornija od drugih sistema sa sklopovima i da će svijetski sve intezivnijim svjetskim razvojem programske opreme (kompilatori, programski paralelni jezici, nove programske okoline) ISMM arhitektura imati još većeg značaja na području paralelnog procesiranja.



4. LITERATURA

/BRA87a/ Brajak P., Designing a Reconfigurable Intelligent Memory Module (RIMM) for Performance Enhancement to Large Scale, General Purpose Parallel Processor, Informatica, Vol 11, 1987-No.1, pp. 19-53.

/BRA87b/ Brajak P., Prešern S., Zeleznikar A., Rationale and Concepts for the PARSYS Parallel Processor Architecture, MIT Seminar, Apr. 15. 1987, Laboratory for Computer Science, MIT

/BBN86/ Rettberg R., Thomas R., Contention is no Obstacle to Shared-Memory Multiprocessing, Comm. of the ACM, Vol. 29, no 12, pp. 1202-1212, Dec. 1986

/IBM86/ Pfister G.F. et al., An Introduction to the IBM research Parallel Processor Prototype (RP3), IBM T.J. Watson RC, Yorktown Heights, RC 11948 (#53834), June 1986

/IBM85/ Beetem J., Monty D., Weingarten D., The GF11 Supercomputer, 12-th International Conference on Computer Architecture, 1985, pp. 108-115

/NYU82/ Gottlieb A. et al., The NYU ULTRA Computer - Designing a MIMD Shared Memory Parallel Computer, (-th International Conference on Computer Architecture, 1982, pp. 27-42

/CED86/ Yew P.C., Architecture of the CEDAR Parallel Supercomputer, August 1986, Centar for Supercomputing Research and Development, University of Illinois, Urbana, CSR D Rpt. No. 609

/DF87/ Silc J., Robić B., The Review of Some Data Flow Computer Architectures, Informatica, Vol. 11. No 1., pp. 61-66

/CM85/ Hillis D., The Connection Machine, The MIT Press, 1985

/DAD87/ Pilpel S., A Balanced Scalable Parallel Processor, VLSI Systems Design, March 1987, pp. 80-88

/CHI81/ Snyder L., Programming Processor Interconnection Structures, Tech. Rep. CDS-TR-381, Purdue University



**SVETOVNI STANDARDI NA PODROČJU
RAČUNALNISKIH VODIL**

Avtorji referata: Marko Kovačević, dipl. ing.
Tomaž Hribar, dipl. ing.
Mihajlo Knežević, dipl. ing.
vsi ISKRA DELTA



Ljubljana, maj 1987



1. UVOD

V prvem delu so opisani glavni mehanizmi in problemi univerzalnih računalniških vodil. Z enostavnimi besedami je opisano kako delujejo univerzalna vodila mehansko in električno.

Posebej se obravnava problem arbitraže (dodeljevanja) vodila neki aktivni napravi, ki ima v določenem trenutku kontrolo nad vsemi napravami na vodilu.

O protokolu vodila govorimo kot o načinu ugotavljanja veljavnosti signalov na vodilu v različnih fazah prenosa podatkov med dvema ali več napravama, priključenima na vodilo. V zvezi s protokolom govorimo o časovnem zamiku kot o razliki med najhitrejšo in najpočasnejšo linijo na vodilu, kar je izrednega pomena za časovni protokol vodila.

O problemu metastabilnosti, ki se pojavlja pri vseh vrstah vodil govorimo kot o problemu, ki mora biti zelo skrbno obravnavan pri dizajniranju vodil in naprav glede na hude posledice na zanesljivost sistemov tam, kjer se temu problemu ne daje pravega pomena.

Efikasnost univerzalnih vodil obravnavamo kot ključno lastnost vodil, ki ima bistven pomen za efikasnost sistemov, ki jih na osnovi univerzalnih vodil gradimo. Predlaga se več različnih načinov povečevanja efikasnosti univerzalnih vodil.

Prvi del zaključimo z obravnavo zanesljivosti in tolerance napak na univerzalnih vodilih.

Drugi del podaja pregledno primerjalno tabelo sodobnih 32-bitnih vodil VMEbus, Futurebus, Multibus II, BI. Tabela vsebuje informacije o omenjenih vodilih razdeljene na več področij: splošno (izvor, support, performanse...), protokol, cena in kompleksnost, zanesljivost, podpora multiprocesiranja ter fizične lastnosti.

2. SPLOŠNI POJMI O RAČUNALNIŠKIH VODILIH

Računalniška vodila so komunikacijske poti med različnimi napravami računalniškega sistema. Večina računalnikov, tudi najmanjši, vsebujejo več različnih vodil optimiziranih za različne vrste komunikacij. Večina vodil je skrita znotraj integriranih vezij ali na tiskaninah računalniških modulov. Druga vodila se pojavljajo na dostopnih mestih računalniških sistemov in so običajno opisana v priloženi dokumentaciji računalniškega sistema, kar omogoča modularno širitev sistema.

Nekatera vodila niso samo enostavno dostopna temveč celo ustrezajo široko sprejetim standardom. To omogoča enostavno komunikacijo med računalniškimi napravami različnih proizvajalcev.



Termin "vodilo" označuje možnost komuniciranja med več kot dvema napravama. Vodilo običajno implicira paralelne povezave med napravami.

Sodobna vodila so narejena iz električnih prevodnikov, večinoma bakrenih žic, bakrenih linijskih vzorcev na tiskanem vezju ali aluminijastih vzorcev v integriranih vezjih. Optične komunikacije se niso ekonomične za vodila zaradi težav pri distribuciji signalov na več naprav ter visoke cene pretvorbe svetlobnih signalov v električne.

Slika 1 ponazarja delovanje vodil.

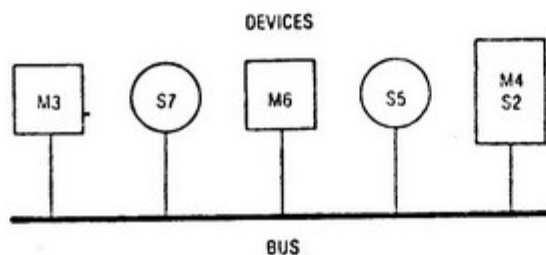
Ce naprava M3 želi komunicirati z napravo S5, M3 pošlje signale po vodilu, ki povzročijo odgovor s strani S5. Signali, ki selektirajo S5 - in ne kake druge naprave na vodilu - se imenujejo NASLOV.

Ce S3 pošilja podatke za S5 pravimo, da M3 VPISUJE v S5. Ce S5 pošilja podatke za M3 pravimo, da M3 ČITA iz S5. S3 določa vpisovanje ali čitanje tako da generira kontrolni signal dodatno k naslovu.

Napravo, ki inicira in kontrolira potek komunikacije imenujemo GOSPODAR (master). Napravo, ki odgovori imenujemo SUŽENJ (slave). Nekatero napravo na vodilu lahko deluje kot gospodar in kot suženj (M4/S2).

Ce neka naprava želi komunicirati po vodilu mora le-to biti nezasedeno. Ce pa dve (ali več) naprave želita istočasno uporabiti vodilo v trenutku, ko le-to ni zasedeno mora obstajati mehanizem odločanja katera od teh naprav bo zasedla vodilo. Temu mehanizmu pravimo ARBITRAŽA. Arbitraža je možna na več načinov, kar je eden izmed bistvenih elementov po katerem se vodila razlikujejo.

Pri računalniških vodilih opažamo različne metode ugotavljanja veljavnosti signalov. Celotnemu postopku ugotavljanja veljavnosti signalov na vodilu pravimo PROTOKOL VODILA. Operacijo naslavljanja naprav na vodilu imenujemo NASLOVNI CIKEL. Operacijo prenosa podatkov po vodilu med dvema napravama imenujemo PODATKOVNI CIKEL.



Slika 1 : Primer računalniškega vodila



Razlikujemo dva protokola vodil - SINHRONI in ASINHRONI.

- * Pri sinhronem protokolu ugotavljamo veljavnost signalov glede na uro vodila.
- * Pri asinhronem protokolu ugotavljamo veljavnost signalov na vodilu na osnovi specialnih potrditvenih signalov.

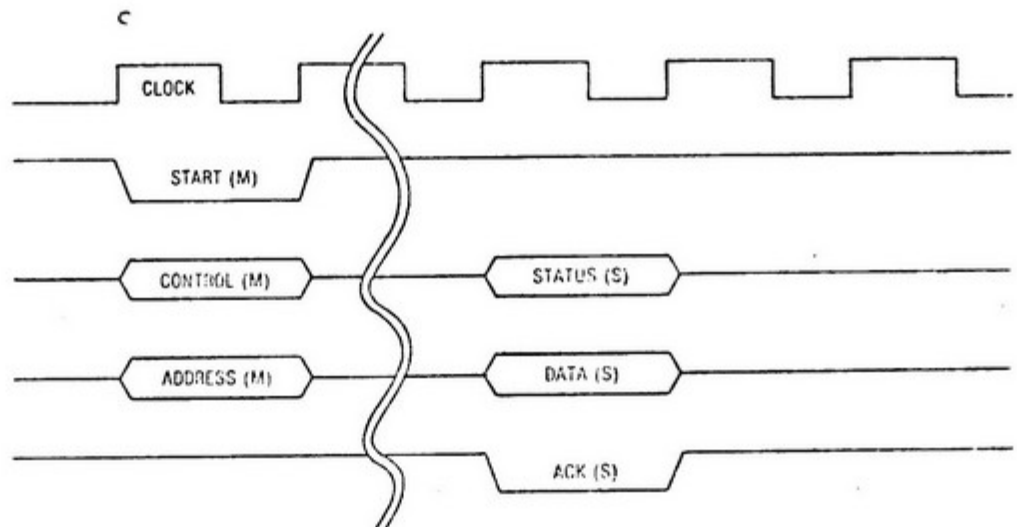
Zanesljivost računalniških vodil bistveno vpliva na zanesljivost sistemov zgrajenih nad temi vodili. Za izboljševanje zanesljivosti vodil pogosto prenašamo redundantne informacije po vodilu, kar omogoča kontrolo pravilnosti prenosa (pariteta, ECC). Pri časovno kritičnih operacijah na vodilu uporabljamo mehanizme kot je TIME-OUT.

Vodilo s svojo geometrijo in elektromehanskimi lastnostmi vpliva na namembnost, zanesljivost in ceno sistemov.

3. PROTOKOLI

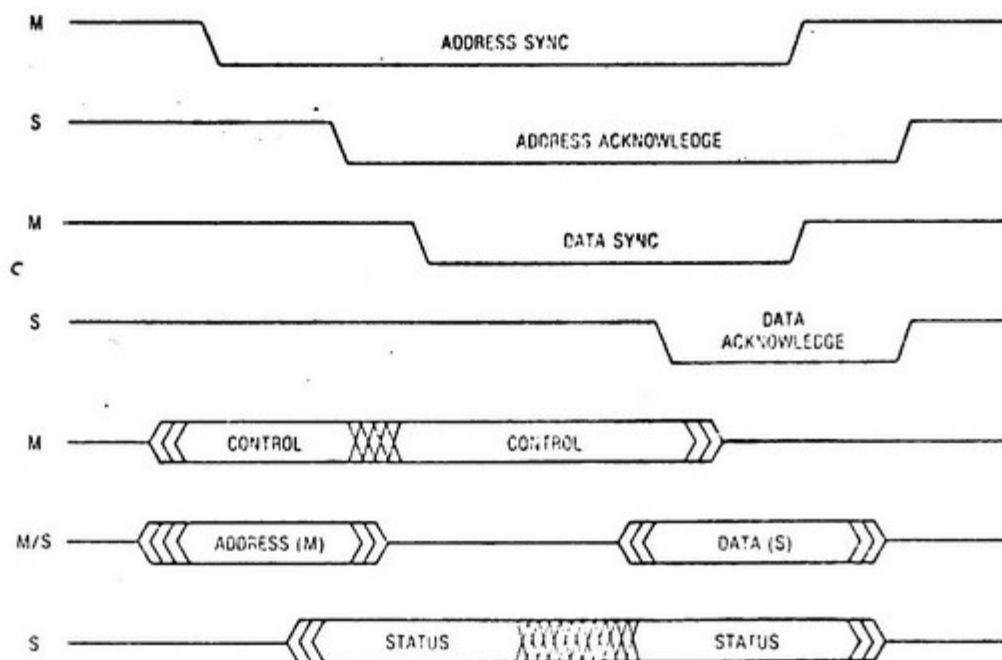
Protokol je način ugotavljanja veljavnosti signalov na vodilu. Poznamo dve skupini protokolov in to sinhroni in asinhroni.

Pri sinhronih protokolih so vsi signali veljavni glede na uro na vodilu. Pri asinhronih protokolih pa vsaka postavitev signalov zahteva potrditev in označitev kdaj je zahtevana operacija končana. Sinhroni protokoli so v splošnem lažje razumljivi, lažje se implementirajo in testirajo, vendar pa so vezani na določeno hitrost ure.



Slika 2: Sinhroni protokol vodila





Slika 3: Asinhroni protokol vodila

Pri asinhronih protokolih je težje načrtovati in testirati vmesnik na vodilo, se posebej, če želimo vmesnik realizirati v VLSI tehnologiji. Zato se asinhroni protokol izvaja z maksimalno hitrostjo, ki jo dopušča določena tehnologija. Če se izboljša sama tehnologija to dovoljuje tudi večje hitrosti pri protokolu.

O časovnem multipleksiranju linij govorimo kadar se v različnih fazah prenosa na linijah pojavljajo različne informacije. Časovno multipleksirana so lahko tako sinhrona kot asinhrona vodila. Običajno se na multipleksiranih linijah pojavljajo adresse in podatki, pri nekaterih vodilih po istih linijah poteka tudi arbitražna (BI vodilo).

Časovno multipleksiranje linij zmanjša število pinov tako na konektorju kot tudi na vmesniku na vodilo. Zanj je se posebej pomembno, če je vmesnik izveden v VLSI tehnologiji.

Časovno prekrivanje operacij (pipelining) se lahko pojavi le pri vodilih, ki imajo multipleksirane adresne, podatkovne in kontrolne linije. Časovno prekrivanje operacij omogoča, da se prekrivata faza zahteve trenutnega cikla prenosa in faza odgovora prejšnjega cikla prenosa. To omogoča boljše iskorisčenost pasovne širine vodila.



Prekinitvene strukture v grobem lahko razdelimo na hardverske prekinitve in na virtualne (inteligentne) prekinitve. Preko takih prekinitvenih struktur lahko naprave na vodilu sinhronizirajo svojo dejavnost ali pa si pošiljajo sporočila o stanju določene naloge, ki jo opravlja več naprav. Pri hardverskih prekinitvenih strukturah si ena ali pa več naprav deli posebno linijo na vodilu, preko katere pošilja svoje prekinitvene zahteve.

Pri virtualnih prekinitvenih strukturah pa takih linij ni na voljo, ampak se prekinitvene zahteve pošiljajo preko posebnih paketov podatkov (sporočil).

Večina sodobnih vodil omogoča prenos 8-, 16-, 24- ali 32-bitnih podatkov. Omejitve, ki se postavljajo so večinoma pri prenosih blokov podatkov, ko 24 bitna širina ni dovoljena. Važno je tudi da so podatki poravnani na ustrezno adresu.

Prenose na vodilu lahko delimo glede na število naprav, ki sodelujejo v prenosu, glede na število prenosov in glede na način, kako naprave sodelujejo v protokolu prenosa.

Naprava na vodilu, ki zahteva prenos, lahko naslovi eno samo napravo, lahko pa tudi več. Če je naslovljenih več naprav in se podatki v te naprave vpisujejo, potem se izvaja broadcast operacija. Broadcast operacija pa se izvede, kadar se podatki čitajo iz več kot ene naprave. Vsa vodila omogočajo enostavne enojne operacije vpisovanja ali čitanja, nekatera pa omogočajo tudi prenos blokov podatkov. Prenos bloka podatkov je sestavljen iz enega cikla, ki generira začetno addresso in, ki mu sledi več podatkovnih ciklov. Kontrolne linije lahko pri tem označujejo operacijo vpisa ali branja, vendar pa se v prenosu enega bloka ti dve operaciji običajno ne smeta mešati.

Naprava, ki je prenos zahtevala, kontrolira ves čas prenosa naslov in kontrolne linije ter v primeru napake poskrbi za smiselno zaključitev prenosa, medtem ko naslovljena naprava le sprejme ali pošlje podatke in konča svoj del protokola.

Naprava, ki prenos zahteva pa lahko sestavi tudi sporočilo določene oblike, ki vsebuje vse potrebne podatke (tip sporočila oz. ukaza, naslov sprejemnika in pošiljatelja ...) in morebitne podatke. Nato naprava tako sporočilo pošlje na vodilo. Naprava, ki je naslovljena tako sporočilo sprejme, ga obdela in pošlje prvi napravi ustrezno sporočilo. V tem načinu prenosa govorimo o prenašanju sporočil (message passing).

Če si hoče neka naprava zagotoviti izključni dostop do določenih podatkov v naslovljeni napravi, uporabi zaklenitveno operacijo, ki drugim napravam onemogoča dostop do podatkov v naslovljeni napravi.



Nekatera vodila imajo že vgrajene funkcije (možnosti) za ohranitev koherence CACHE spominov. Pri implementaciji ukaznega cache spomina se ne srečamo s problemom koherence ker se kode ukazov ne spreminjajo. Implementacija podatkovnega cache spomina (ali kombinacije ukazni/podatkovni) je veliko bolj učinkovita vendar pa tudi veliko bolj kompleksna zaradi problematike koherence podatkov v delovnem in cache spominu.

Na voljo sta dva načina ohranjanja konsistence cache spomina. Pri načinu "vpisovanje skozi" se vse, kar se vpiše v cache vpiše tudi v glavni pomnilnik. Pri načinu "vpisovanje nazaj" se podatki vpičejo najprej v cache, v glavni pomnilnik pa po potrebi kasneje. Ta način omogoča boljso iskoriščenost glavnega vodila, zato pa je implementacija zahtevnejša. Problem ohranjanja konsistence cache spomina se še zaostri pri multiprocesorskih sistemih.

4. ARBITRAŽA

Na vodilo je običajno priključenih več naprav, ki lahko v kateremkoli trenutku zahtevajo dostop do vodila. Zato mora obstajati mehanizam, ki določa, kdaj naprava lahko dobi dovoljenje za uporabo vodila. Ta mehanizem se imenuje arbitražo. Arbitražo lahko delimo na dve osnovni vrsti in sicer na centralizirano in distribuirano arbitražo.

- * O centralizirani arbitraži govorimo, kadar imamo na vodilu en sam arbitrator, ki določa, katera naprava bo naslednja imela pravico dostopa do vodila. Dva primera centralizirane arbitraže sta verižna in zvezdasta. Verižna povezava je izvedena tako, da so vse naprave povezane na linijo za pošiljanje zahtev. Linija preko katere arbitrator pošlje signal dovoljenja uporabe vodila, pa je prekinjena. Vsaka naprava na vodilu ima vhod ter izhod za signal dovoljenja, kakor tudi vezje, ki na osnovi zahteve lokalne naprave za vodilo odloči ali bo le-ta zasedla vodilo ali pa bo signal dovoljenja posredovalo na izhodu za dovoljenje in tako omogočilo naslednji napravi v verigi dostop do vodila. V povezavi v zvezdo pa ima vsaka naprava svojo linijo za zahtevo in svojo linijo za sprejem signala dovoljenja.
- * Distribuirana arbitražo nima centralnega arbitratorskega vezja, ampak se vsaka naprava zase odloča, kdaj lahko zasede vodilo. Pri distribuirani arbitraži ima lahko vsaka naprava svojo linijo preko katere pošilja zahteve za vodilo, lahko pa ima tudi eno ali več kod, ki določa prioritetni nivo naprave. Ko naprava želi arbitrirati, postavi kodo na vodilo. V prvem primeru porabimo več pinov na konektorju, obstaja več linij na vodilu, zato pa je arbitratorsko vezje preprostejše in arbitražo je hitrejša.



Ce zelimo učinkovito izrabiti vodilo, moramo arbitracijski cikel izvesti preden se konča prejšnji podatkovni prenos. Princip "fairness" zagotavlja, da naprava, ki je prej poslala zahtevo dobi vodilo prej kot višje prioriteta naprava, ki pa začne arbitražo kasneje.

V primeru, ko naprava konča prenos podatkov, hkrati pa ni druge naprave, ki bi želela dostop do vodila lahko prejšnja naprava obdrži kontrolo nad vodilom dokler druga naprava ne zahteva vodila. Ta način ohranitve kontrole imenujemo "parking".

5. ZANESLJIVOST

Zanesljivost delovanja vodila je eden od pomembnih faktorjev, ki vpliva na zanesljivost celotnega sistema. Zato je potrebno v definiciji vodila predvideti določene funkcije, ki omogočajo, sprotno preverjanje pravilnost signalov in prenosa podatkov na vodilu.

Običajno se za detektiranje napak pri prenosu podatkov skupini signalov dodeli paritetni bit. ECC koda se uporablja predvsem v velikih sistemih, ki za majhno število kontroliranih bitov zahteva razmeroma veliko število ECC bitov. Zato se pri manjših sistemih uporablja pariteta za zaščito podatkov na vodilu.

Kadar pride do napake pri prenosu podatkov mora obstajati način, da se celoten prenos ponovi. Prenos ponovi tista naprava, ki ga je zahtevala, medtem ko veljavnost podatkov lahko kontrolirajo vse naprave, ki sodelujejo pri prenosu. Število ponavljanj operacij mora biti določeno v specifikaciji vodila. Do ponavljanj operacij na vodilu lahko pride tudi, če je naslovljena naprava zaposlena s kakšno drugo operacijo in v določenem času ne more odgovoriti na zahtevo ali pa, če pri naslovljeni napravi pride do take napake, da ne more pravočasno in pravilno odgovoriti.

Da preprečimo, da neka naprava zasede vodilo za več kot določen čas uporabljamo TIMEOUT mehanizem. Po določenem času se vodilo vrne v regularno stanje. Do timeouta pride, če naprava, ki je prenos zahtevala, naslavlja napravo, ki je ni v sistemu, ali pa, če naslovljena naprava predolgo ne vrne odgovora.

Metastabilnost se nanaša na obnašanje logičnih elementov kot so flip-flopi, kadar so podvrženi vhodnim signalom, ki so časovno drugačni kot to zahteva specifikacija logičnega elementa. V tem primeru se izhod takega elementa postavi v naključno izbrano stanje, pred postavitvijo v končno stanje lahko izhod tudi oscilira. Metastabilnost je problem, ki se pojavlja pri vseh vrstah vodil in na katerega moramo biti zelo pozorni, če zelimo, da bo sistem zanesljivo deloval.



Mehansko mora biti vodilo tako zasnovano, da bo zanesljivo delovalo tudi kadar se konektorji ne bodo povsem ujemali in ko plošče ne bodo povsem ravne (določene morajo biti meje dopustnih toleranc). Konektorji morajo biti tudi dovolj odporni tako na mehanske kot tudi na atmosferske vplive.

Nekatere specifikacije tudi določajo maksimalno električno porabo sistema. Sistem lahko hladimo bodisi z naravno konvekcijo ali pa s forsiranim hlajenjem. V obeh primerih pa moramo zagotoviti, da so vsi elementi dovolj hlajeni in da ne prihaja do lokalnega dviga temperature zaradi neustreznega hlajenja.

6. ELEKTROMEHANSKE LASTNOSTI

Električni del vodila razen zadnje plošče sestavljajo oddajniki in sprejemniki. Oddajniki so vezja, ki spreminjajo napetostne nivoje na signalnih linijah vodila. Oddajnik mora biti sposoben krmiliti linijo na visok napetostni nivo, na nizek napetostni nivo, obstajati pa mora tudi stanje, ko je oddajnik na vodilu neaktiven.

En način, da to dosežemo je uporaba vezij s tristanjskimi izhodi.

Drugi način pa je priključitev linije na napetostni vir preko upora, oddajnik pa uporabimo za krmiljenje linije na določen napetostni nivo. Glede na tehnologijo, ki jo pri drugem načinu uporabljamo, ločimo oddajnike s prostim kolektorjem (TTL), oddajnike s prostim emitorjem (ECL) in oddajnike s prostim ponorom (MOS).

Sprejemniki so vezja, ki primerjajo vhodno napetost z vrednostjo, ki je vnaprej določena, in nato generirajo ustrezni logični nivo, ki ga uporablja ostala logika na modulu.

Oddajniki in sprejemniki so električno del vodila, čeprav se običajno fizično nahajajo na modulih. Ko računamo zanesljivost posameznega vodila moramo upoštevati tudi zanesljivost oddajnikov in sprejemnikov.

Povezave na zadnji plošči (backplane) moramo obravnavati kot prenosne linije, ki imajo svojo karakteristično impedanco. Zato morajo biti vse linije na zadnji plošči na obeh koncih vodila ustrezno zaključene, ker v nasprotnem primeru lahko pride do odbojev na linijah.

Casovni zamik je največja možna razlika v propagacijskem času najhitrejšega in najpočasnejšega signala na liniji. Casovni zamik se pojavlja pri vseh vrstah vodil. V primeru sinhronih vodil, vpliva na maksimalno, se varno, frekvenco ure na vodilu in ga kasneje ni treba več upoštevati v dizajnu, medtem ko je pri asinhronih vodilih potrebno časovni zamik vpoštevati pri vsaki potrditvi veljavnosti kakega signala.



Elektromagnetne motnje so pomemben faktor, ki ga upoštevamo pri gradnji računalniških vodil. Da se pojavijo težave zaradi elektromagnetnih motenj morajo biti izpolnjeni trije pogoji:

- * obstajati mora vir elektromagnetnih motenj, ki je lahko praznjenje elektrostaticnega naboja, motnje električne napetosti na katero so priključeni usmerniki, drugi računalniki, oddajniki ...
- * obstajati mora medij preko katerega se te motnje prenašajo na napravo, ki je na take motnje občutljiva. Motnje se lahko prenašajo s prevajanjem ali s sevanjem.
- * obstajati morajo elementi vodila, ki so na motnje občutljivi (linije, oddajniki, sprejemniki...).

Problem elektromagnetnih motenj odstranimo ali vsaj zmanjšamo pod dopustno mejo z odstranitvijo ali dušitvijo enega ali več zgornjih pogojev.

Vodila pogosto uporabljajo standardne velikosti plošč tiskanega vezja (npr. Evropa format pri VME vodilu, podaljšan Evropski format pri Multibus II vodilu), medtem ko imajo nekatera vodila definirane lastne formate plošč, ki tipično varirajo med 8x9 colami in 14x15 colami.

Priključitveni konektorji na vodilo so lahko robni konektorji, 64- ali 96- pinski DIN konektorji ali ZIF (Zero Insertion Force) konektorji. Priključitveni konektor mora zagotavljati zanesljiv stik med ploščo in vodilom skozi celotno življensko dobo sistema. Signali na konektorjih morajo biti tako razporejeni, da omogočajo čim krajše priključitvene linije na oddajnike in sprejemnike. Običajno definicija vodila predpiše maksimalno dolžino teh povezav, ki je tipično okoli 5 cm.

7. PRIMERJALNA TABELA ZNANIH SVETOVNIH VODIL

V nadaljevanju podajamo primerjalno tabelo znanih svetovnih računalniških vodil, ki lahko pomaga strokovnjakom pri izbiri vodila za svoje bodoče projekte, kakor tudi pri oceni o primernosti nekaterih vodil za posamezne računalniške konfiguracije in aplikacijska okolja. Primerjalna tabela podaja vrsto podatkov za popularna vodila VMEbus, MULTIBUS II, BI, Futurebus.

Vodilo VMEbus smo izbrali za primerjavo zaradi njegove popularnosti v svetu in velikega deleža na tržišču (se posebej OEM tržišču), njegove vsestranske uporabnosti v različnih okoljih (poslovnem, industrijskem ...) kakor tudi prisotnosti tega vodila v jugoslovanskem prostoru (Iskra Delta sistemi Triglav).



MULTIBUS II smo izbrali kot primer sinhronega vodila s zelo razdelano strukturo kompleksnega vodila, ki ima podvodila za enostavno delitev sistemskih funkcij, kar ima za cilj razbremenitev glavnega sistema. Omembna vredna je zelo močna VLSI potpora temu vodilu, ki jo ponuja podjetje Intel (USA) kot predlagatelj in lastnik pravic.

Vodilo BI smo izbrali kot primer vodila, ki ga uporabljajo za gradnjo zmogljivih mini računalnikov VAX (DEC), kakor tudi novih Delta 8000 računalnikov.

Futurebus je primer sodobno zasnovanega hitrega vodila, ki rešuje večino problemov enoprocesorskih in večprocesorskih računalnikov, ki jih gradimo na osnovi sodobnih VLSI gradnikov.

Primerjalna tabela je razdeljena na podtabele, ki grupirajo značilne informacije o omenjenih vodilih po naslednjih ključih:

- * Splošne značilnosti
- * Značilnosti protokola
- * Kompleksnost
- * Podpora večprocesorskih sistemov



SPLOSNE ZNACILNOSTI

| | BI vodilo | VME vodilo | MULTIBUS II vodilo | FUTUREBUS vodilo |
|---------------------------------------|-------------------------|---|-------------------------|-------------------------|
| Standard | / | IE ³ P1014 Draft 1.2 IEC 821 | IE ³ P1296 | IEEEP896.1 |
| Primary Sponsor | DEC | Motorola | Intel | IEEEP896 |
| VLSI support | da | ne | da | - |
| Hitrost prenosa (bandwidth) v Mb/s | 13.3 | 57 | 40 | 117 |
| Pipelinig | ne | | ni uporabljen | ne |
| Prenos sporočil | ne | ne | da | bo definiran |
| Arbitrazni algoritem | Fairness, prioriteta | RWD, ROR | Fairness, prioriteta | Fairness, prioriteta |

ZNACILNOSTI PROTOKOLA

| | BI vodilo | VME vodilo | MULTIBUS II vodilo | FUTUREBUS vodilo |
|--------------------------------|---------------------------|------------------------------------|---|---------------------|
| Protokol | Sinhroni 5 MHz | Asinhroni | Sinhroni 10 MHz | Asinhroni |
| Sirina podatkovne poti | 32 bitov | 32,24,16, 8 bitov | 32,24,16,8 | 32,24,16,8 |
| Multipleksiranost | ADR,DATA, ARB | nemultip: | ADR,DATA | ADR,DATA |
| Velikost adresnih prostorov | 2 ³⁰ (1 Gb) | 2 ²⁴ 2 ³² | 2 ³² ,2 ²⁴ , 2 ¹⁶ ,2 ⁸ | 2 ³² |
| Broadcast | da | ne | da | da |
| Zaklepanje | da | da | da | da |
| BIST | da | ne | ne | da |



KOMPLEKSNOST

| | BI vodilo | VME vodilo | MULTIBUS II vodilo | FUTUREBUS vodilo |
|--|--------------|---------------|-----------------------|---------------------|
| Število adresnih prostorov | 1 | 64 | 4 | 1 |
| Število konektorjev | 2 | 2 | 1 | 1 |
| Število pinov | 120 | 128 | 96 | 96 |
| Število aktivnih signalov | 52 | 107 | 67 | 67 |
| Število linij za zahtevo prekinitve | / | 7 | / | / |
| Pariteta | da | ne | da | opcija |

PODPORA VECPROCESORSKIH SISTEMOV

| | BI vodilo | VME vodilo | MULTIBUS II vodilo | FUTUREBUS vodilo |
|------------------------------------|--------------|-----------------|--------------------------|---------------------|
| Geografsko naslavljanje | slot ID | ne | LATCH n pin | 5-bitni slot ID |
| Avtokonfiguriranje | da | ne | da (ob dvigu sistema) | da |
| Virtualne zahteve za prekinitve | da | ni specific. | da | da |
| Prenasanje sporočil | ne | ne | da | da |
| Cache write-through | da | omejeno | omejeno | da |

8. ZAKLJUČEK

Tako kot vsa druga področja računalništva in mikroelektronike se tudi na področju računalniških vodil pojavlja marsikaj novega, kar ponuja veliko večje možnosti za gradnjo kompleksnejših in inteligentnejših računalniških sistemov. Vse večje so možnosti za komponiranje kompleksnejših računalniških sistemov na osnovi inteligentnih računalniških podcelot od katerih vsaka predstavlja cel računalnik s procesorji, spomini in periferijo in ki so vse povezane preko vodil z vgrajenimi mehanizmi, ki rešujejo celo paleto tehničnih problemov, ki se tukaj pojavljajo. Računalniška vodila postajajo že sama po sebi inteligentna tako da prevzemajo funkcije, ki smo jih do sedaj srečevali samo pri najbolj kompleksnih sistemih. Take funkcije omogočajo paralelizme v delovanju različnih računalniških podsistemih ali v multiprocesorskih sistemih, omogočajo hitre prenose blokov podatkov po vodilih, izmenjavo kompleksnih sporočil med napravami sistema in se mnogo drugega. Počasi se klasična predstava o računalniških vodilih kot o nekemu številu paralelnih prenosnih linij izgublja, pojavljajo se pa inteligentne strukture za komuniciranje med posameznimi napravami sistema, kar bo v prihodnosti omogočilo dejansko paralelno procesiranje.

Več o problematiki sodobnih računalniških vodil lahko najdete v naslednjih publikacijah:

- * David B. Gustavson, Stanford Linear Accelerator Center: Computer Buses - A Tutorial, 1984 IEEE.
- * Rob Wilson : The Physics of Driving Backplane Buses, EDN March 1986
- * Roger Edwards : Futurebus - the Independent Standard for 32-bit Systems, EDN March 1986
- * Simon Peyton Jones : Using Futurebus in a Fifth-Generation Computer, EDN March 1986
- * Multibus II Bus Architecture Specification Handbook, Pub. No. 146077-B, Intel Corporation
- * NuBus Specification, Pub. No. TI-2242825-0001, Texas Instruments Incorporated, Dallas, TX
- * The VMEbus Specification Rev.C.1, Motorola, October 1985

**NADALJNI RAZVOJ
RACUNALNISKIH SISTEMOV TRIGLAV**

Avtorji referata: Marko Rogač, dipl.ing.
mag. Damjan Hafner, dipl.ing.
Marjan Bohnec, dipl.ing.
vsi ISKRA DELTA



Ljubljana, maj 1987



1. PREGLED OBSTOJEČEGA STANJA NA APARATURNI IN PROGRAMSKI OPREMI RAČUNALNIŠKIH SISTEMOV TRIGLAV

1.1. Arhitektura aparaturne opreme

Arhitektura aparaturne opreme sistemov temelji na VME vodilu. VME vodilo predstavlja svetovni industrijski standard, ki ga podpira 200 proizvajalcev z 2000 različnimi moduli. Modularnost in možnost različnega konfiguriranja (razširitve pomnilnika, procesne moči itd.) sistemov Triglav, daje široko uporabo tako v poslovnih kot procesnih sistemih.

VME vodilo specificira povezavo med obdelujočimi podatki, pomnilnimi podatki in perifernimi enotami v povezujočem sistemu. Zadnja verzija VME vodila je C.1 in je standardizirano kot IEC 821 vodilo in IEEE P1014/D1.2 vodilo.

VITA (VMEbus International Trade Association) je mednarodna organizacija, ki stalno spremlja vodilo in izdaja dodatne specifikacije, kot so preizkusni predpisi in potrjuje, da moduli ustrezajo VME specifikacijam.

Uvedeno je se serijsko VMS in razširitveno VSB vodilo.

1.2. Moduli računalniških sistemov Triglav

Moduli računalniških sistemov Triglav ustrezajo specifikacijam VME vodila verzija C.1 in jih lahko združujemo oz. konfiguriramo z moduli drugih proizvajalcev.

Vsak modul predstavlja zaključeno funkcijsko celoto in tehnološko zadovoljuje visoko integracijo elementov na večslojnih tiskanih vezjih.

Trije procesorski VME moduli omogočajo tri različne računalniške sisteme, ostali periferni moduli so enaki za vse sisteme.

Procesorski moduli so:

- * CPU 286 modul z mikroprocesorjem iAPX80286, pomnilnikom 512KB RAM in 64KB EPROM, dvema serijskima kanaloma (RS-232), uro realnega časa itd.
- * CPU J11 modul z mikroprocesorjem DEC J11, pomnilnikom 512KB in 8KB EPROM, serijskim (RS-232) in paralelnim (Centronics) kanalom, uro realnega časa, itd.
- * CPU 68010 modul z mikroprocesorjem M 68010, pomnilnikom 1MB RAM in 64KB EPROM, dvema serijskima (RS-232) in paralelnim (Centronics) kanalom, uro realnega časa, itd.
- * pomnilnik RAM razširjamo z DRAM modulom kapacitete 2 MB in pariteto



- * periferne enote gibki disk in winchester disk podpira krmilnik FD/WD z vmesnim pomnilnikom 8KB in DMA prenosom
- * ICC modul krmili šest serijskih kanalov (RS-232 oz RS-422) in LAN mrežo. Modul krmili procesor Z80 s 64KB pomnilnikom, kar omogoča emulacijo protokolov na samem modulu
- * GRAF modul krmili VT-100 alfanumerični terminal in barvno grafiko resolucije 640 x 480 x 16
- * napajalnik, ki ustreza vsem zahtevam VME vodila tako dimenzijsko kot tehnično

Novejši moduli so:

- * AD 16 modul je 12-bitni analogno digitalni pretvornik s šestnajstimi analognimi oz. osmimi diferencialnimi vhodi. Vhodne veličine so: bipolarni 12,5V, 15V, 110V
unipolarni 0-5V, 0-10V
- * VME/DIPS modul je vmesnik, ki omogoča povezavo med VME in DIPS vodilom. Modul tako omogoča uporabo vseh perifernih modulov mikroročunalniškega sistema DIPS
- * VME/Ramtek modul povezuje VME vodilo na Ramtek grafični terminal. Povezava je realizirana z DMA kanalom in tako omogoča, da postane računalnik Triglav zelo visoko zmogljiva delovna postaja
- * VME/EGA modul daje kompletno EGA grafiko na VME vodilu in PC kompatibilnost

1.3. Konfiguracije računalniških sistemov Triglav

Konfiguracije računalniških sistemov Triglav so osnovane na posameznih procesorskih moduli s krmilnikom diskov oz. gibkih diskov.

Kapaciteta winchester diska je 40MB ali 80MB gibkega diska 1MB in hitrega traku (streamer) kapacitete 20MB ali 40MB.

Z dodatnimi moduli razširjamo sisteme Triglav s pomnilnikom, serijskimi kanali, grafiko, itd.



1.4. Sistemska programska oprema računalniških sistemov Triglav

1.4.1. Triglav DEL-16 s procesorsko glavo DEC J11

Triglav DEL-16 s procesorsko glavo DEC J11 teče z operacijskim sistemom DELTA/M. Vsebuje vse potrebne kanalske programe za winchester diske, gibke diske, hitri trak, tiskalnik, terminale, itd. Z operacijskim sistemom dajemo tudi grafično knjižnico TGLIB.

Od višje nivojskih jezikov nudimo Cobol, Fortran, Pascal, Basic, itd.

1.4.2. Triglav UNX-16 s procesorsko glavo M68010

Triglav UNX-16 s procesorsko glavo M68010 teče z operacijskim sistemom Uniplus, ki je AT&T sistem UNIX V.2.

Vsebuje vse potrebne kanalske programe za winchester diske, gibke diske, hitri trak, tiskalnik, terminale, itd.

Z operacijskim sistemom Uniplus dajemo tudi grafično knjižnico TGLIB, ki zajema vse osnovne grafične rutine (inicializacija, izbira palete barv, risanje premice, pravokotnika, loka, kroga, itd.).

Od višje nivojskih jezikov nudimo (razen C previjalnika, zbirnika in Fortran 77, ki je že vključen v operacijski sistem Uniplus) še RM-Cobol, SVS-Pascal in SVS-Fortran.

1.4.3. Triglav UNX-16 s procesorsko glavo M68010

Triglav UNX-16 s procesorsko glavo M68010 za vodenje procesnih aplikacij v realnem času nudimo operacijski sistem OS9, ki vsebuje vse potrebne kanalske programe.

Od višjih jezikov nudimo Pascal in Basic.

1.4.4. Triglav XEN-16 s procesorsko glavo iAPX80286

Triglav XEN-16 s procesorsko glavo iAPX80286 teče z operacijskim sistemom XENIX, ki je AT&T sistem UNIX V.2.

Vsebuje vse potrebne kanalske programe za winchester diske, gibke diske, hitri trak, tiskalnik, terminale, itd.

Z operacijskim sistemom XENIX dajemo tudi grafično knjižnico TGLIB.

Od višje nivojskih jezikov nudimo se MS-Cobol, MS-Pascal in MS-Fortran.



1.4.5. Triglav XEN-16 z operacijskim sistemom MS-DOS

Na Triglavu XEN-16 uvajamo tudi MS-DOS operacijski sistem, ki bo z modulom VME/EGA tudi po grafični strani kompatibilen s PC računalniki.

Poudarjamo, da je računalnik v osnovi na VME vodilu z MS-DOS operacijskim sistemom.

1.4.6. Komunikacijska programska oprema

- * Triglav DEL-16 s procesorsko glavo DEC J11 ima izdelano DELTANET protokol, ki omogoča mešano povezavo vseh sistemov Delta/M in Delta/V. Izdelan je tudi program za distribucijo datotek (PFT/RMT). V izdelavi je lokalna mreža na osnovi "token bus" arhitekture.
- * Triglav UNX-16 in Triglav XEN-16 vključujeta v operacijskem sistemu UNIX povezavo datotek (uucp), sistema v funkciji terminala (CU), izvajanje programa (UUX) in pošiljanje sporočil (mail). V izdelavi je program za prenos datotek z Delta/M in Delta/V operacijskega sistema (RMT) in program za posredovanje datotek RMT uporabnikom (PFT).

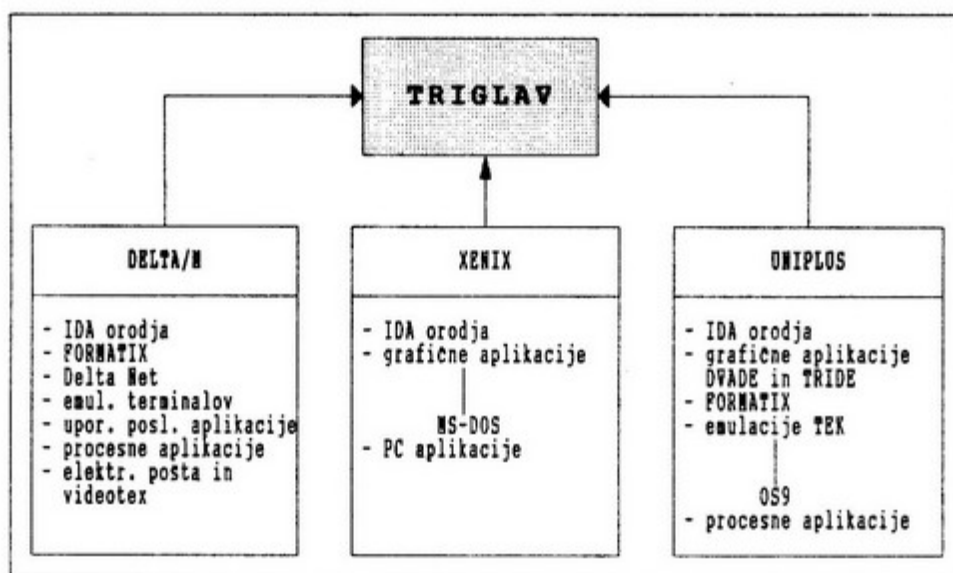
1.5. Aplikativna programska oprema računalniških sistemov Triglav

- * Vsa aplikativna programska oprema, ki teče na Delta/M operacijskem sistemu na večjih računalnikih Delta (Delta 800, Delta 4850, itd.) teče tudi na sistemu Triglav DEL-16.
- * Aplikativne programske opreme na operacijskem sistemu Unix (Uniplus oz. Xenix) je zaradi uvajanja novega operacijskega sistema se relativno malo, vendar se intenzivno dela na pisanju aplikacij. Upoštevana je prednost Unix operacijskega sistema zaradi možnosti prenosa aplikacij na različne računalniške sisteme.

Izdelane so emulacije grafičnega terminala TEK 4107, PLOT 10 TCS knjižnice za terminale serije 4010 in 4107 in tehnično risanje DWAVE in TRIDE.

Za operacijski sistem UNIX je pripravljen tudi programski paket FORMATIX in od IDA orodij je že uvedena IDA-Paza, IDA-Leksikon, IDA-Ekran in IDA-Query.





1.6. Ohišja računalniških sistemov Triglav so pripravljena tako za poslovno kot procesno uporabo

- * Poslovna varianta sistema Triglav je v modernem ergonomsko oblikovanem namiznem ohišju. Novo je računalniško kabinetno ohišje z zadosti prostora za periferijo in module.

Izdelano je tudi ohišje za dodatno periferijo k sistemu. Novo je namizno ohišje v stilu PC z dodatno periferijo.

- * Za procesno vodenje nudimo 19'' laboratorijsko ohišje za namizno izvedbo in vgradnjo v omaro z ustrezno opremo za vgradnjo modulov in periferije.

1.7. OEM prodaja modulov in sistemov Triglav

OEM prodaja modulov in sistemov Triglav omogoča uporabnikom gradnjo lastnih sistemov oz. podsklopov. Module dajemo z ustrezno programsko opremo in ustreznimi priročniki, ki omogočajo uporabnikom enostavno uporabo modulov in sistemov.

2. RAZVOJ 32-BITNIH RAČUNALNIŠKIH SISTEMOV TRIGLAV

- 2.1. Za nalogo smo si zadali izdelati 32-bitni mikroračunalniški sistem, modularne zgradbe primeren za poslovne in procesne aplikacije. V poslovnih aplikacijah kot večuporabniški sistem z učinkovitim delom 32 uporabnikov in kot visoko zmogljiva grafična postaja. V procesnih aplikacijah kot osnova za robotske sisteme, procesne računalnike, itd.

Aparaturna oprema in arhitektura sistema temelji na VME vodilu, vsak modul je funkcionalno zaključena celota.

Programska oprema mora odgovarjati svetovnim standardom.

- 2.2. 32-bitni mikroprocesorji so razen napredka v visoki integraciji elementov tudi računačniško prešli v novo družino. Če so se pri 16-bitnih mikroprocesorjih proizvajalci zgedovali po miniračunalnikih so pri 32-bitnih mikroprocesorjih uveljavili vse prednosti, ki so jih mikroračunalniki že imeli.

Razen 32-bitnih podatkovnih in naslovnih registrov, več skladovnih kazalcev, direktnega naslavljanja 4 Gigabytes, različnih naslavljanj, so dodali še instrukcijski in podatkovni cache pomnilnik oz. logiko za zunanji cache pomnilnik, virtualni pomnilnik, koprocesorsko logiko, delitev pomnilnika, itd.

Nova VLSI tehnologija 32-bitnih procesorjev dosega 30MHz (pri 16-bitnih 12MHz), kar zahteva za izkoriščanje polne moči (brez čakalnih zank pomnilnika) visoke zahteve tudi ostalih elementov (pomnilniki, vmesna vezja za naslovne podatke, dekodiranje, itd.) Design takih modulov je zahtevnejši in zahteva zmogljivejšo razvojno opremo.

Kompatibilnost programske opreme je tako na izvorni kot objectni kodi med 16-bitnimi in 32-bitnimi mikroprocesorji. Pomeni, da bo vsa izdelana programska oprema 16-bitnih računalniških sistemov Triglav tekla tudi na novih 32-bitnih sistemih. Programski nabor instrukcij je razširjen tako, da je za večjo procesno moč potrebno optimizirati zlasti sistemsko programsko opremo.

Na trgu so tudi nova vezja, namensko izdelana za podporo VME vodila (MVS2400, SCB68155, itd.)

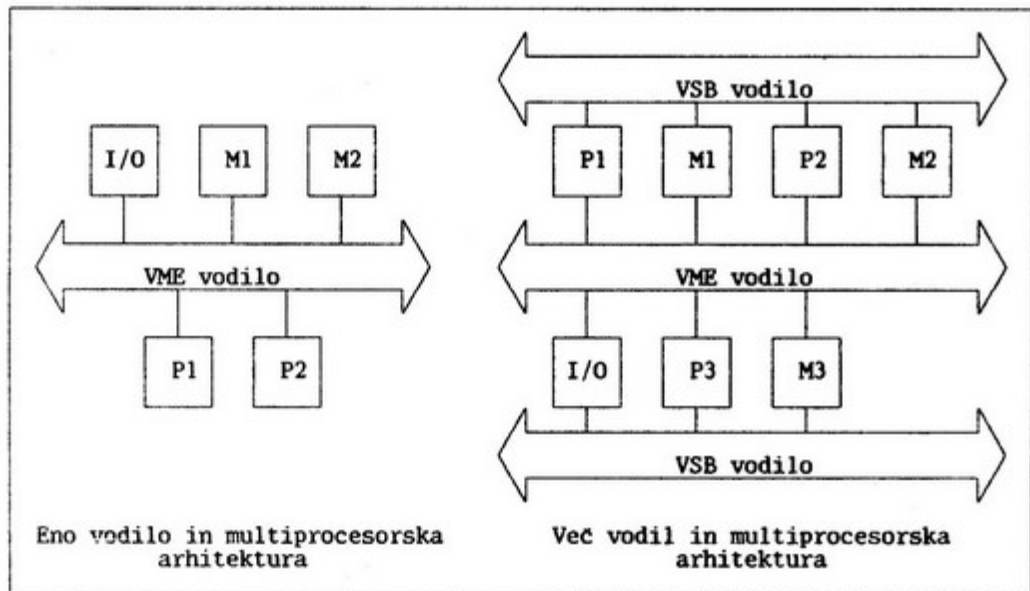
- 2.3. Za zahtevnejše računalniške sisteme je potrebno izdelati distribuirano procesiranje:

- * paralelno procesiranje
- * inteligentni krmilniki



2.3.1. Paralelno procesiranje izvedemo z dodajanjem večih procesorskih modulov.

Pri samo enem vodilu (Time Shared ali Common Bus) z dodajanjem procesnih modulov povečamo promet na vodilu, kar vodi v zmanjšanje prepustnosti in naraščanje arbitražne logike. Problem rešujemo z uvedbo dodatnega VSB (VME Subsystem Bus) vodila. VSB vodilo je v standardiziranju in ga proizvajalci modulov že nudijo.



2.3.2. Cache pomnilniki so hitri pomnilniki, ki so med procesorjem in centralnim pomnilnikom, ki vsebujejo tiste dele programa, ki se izvaja. Z uvajanjem velikih cache pomnilnikov na procesne module nastajajo problemi koherentnosti med centralnimi pomnilniki. Slika podatkovnih semaforjev v cache pomnilniku mora biti enaka kot v centralnem pomnilniku v katerega vstopa več procesorjev. VSB vodilo že vsebuje signal za izločitev naslovov, katerih vsebina se ne sme shranjevati v cache pomnilniku.

2.4. Krmilni moduli (disk, trak, grafika, komunikacije, itd.) morajo zadovoljevati naslednje kriterije:

- * hitri prenos podatkov na vodilu
- * vgrajena inteligenca
- * unifikacija aparature in programske opreme

2.4.1. Hitrost VME vodila je 40Mbytes/s, zato je potrebno, da se s krmilniki približamo takim hitrostim. Primer: za krmilnike diskov SMD-E je hitrost 3 Mbytes/s, SCSI hitrost 1,2 Mbytes/s ali 4Mbytes/s ESDI hitrost 1,2 Mbytes/s, IPI Level2 hitrost 10 Mbytes/s). Vodilo razbremenimo dela s krmilniki in ne zmanjšamo performanc sistema tako, da krmilnik prenaša podatke na VME vodilu hitreje kot jih dobi s periferije.

Krmilnike konstruiramo tako, da rabijo samo tretino prepustnosti vodila za delo.

2.4.2. Inteligentni krmilniki razbremenjujejo poseganje operacijskega sistema na krmilnik. Deli nalog, ki jih izvaja operacijski sistem se prenesejo na krmilnik. Primer je grafični krmilnik, kjer se implementirajo razen splošnih grafičnih komand tudi kompletni grafični operacijski sistem GKS, emulacije raznih alfanumeričnih grafičnih terminalov, komunikacije s tastaturo, misko, ploterji, itd. Pri diskovnem krmilniku se v najenostavnejšem primeru prenesejo logični kanalski programi na krmilnik.

2.4.3. Pri razvoju in nadgrajevanju sistema unificiramo pristop z vodila na krmilnik. Tako probleme prekinitev, pristopanja na vodilo, itd. rešujemo le enkrat za vse krmilnike. Veliko število integriranih vezij, ki so relativno enostavna rešimo s posebnim vezjem, ki ga proizvajalec lahko izdelava po naših zahtevah.

Unifikacijo programske opreme na krmilniku delimo na programsko jedro in programske komande.

* Za programsko jedro na krmilniku izberemo real-time jedro s potrebno razvojno opremo. Jedro, ki seveda zadovoljuje naše zahteve je enako na vseh krmilnikih.

* Programske komande s katerimi operiramo s strani operacijskega sistema naj bodo na čim višjem nivoju, kar zagotavlja že inteligentni krmilnik. Komande so standardne (podobne že kakšnemu standardnemu krmilniku) tako, da jih programer enostavno in hitro vključi v operacijski sistem.

* Računalniški sistemi so omejeni s prepustnostjo vodila. VME vodilo omogoča enostavno dodajanje pomnilnika, procesorjev, krmilnikov, diskov, komunikacij, itd. Proti zasičenosti vodila se borimo z veliko prepustnostjo krmilnikov in inteligenco krmilnikov, ki tudi razbremenijo vodilo.



2.5. Malo je razlike med operacijskim sistemom za več procesorski sistem in velikimi računalniki. Kompleksnost operacijskega sistema, ko več procesorjev deluje istocasno se kaže v zahtevi, da podpre več asinhronih nalog, ki se izvajajo hkratno. Naloga izdelave oz. predelave standardnega operacijskega sistema je zelo zahtevna. Z zahtev operacijskega sistema morajo izhajati tudi aparaturne zahteve (mehanizem interprocesorskih komunikacij, mehanizem sinhronizacije procesov, itd.)

Priprava programske opreme za krmilnike je lažja, ker ne sega globoko v operacijski sistem tako, da prilagodimo le kanalske programe.



OPERACIJSKI SISTEM UNIX

Avtorji referata: Aleksander Hadzi, dipl. ing.
Milan Palian, dipl.ing.
Zvonimir Stipetić, dipl.ing.
vsi ISKRA DELTA



Ljubljana, maj 1987



Članek podaja oris operacijskega sistema Unix. Poleg posplošenega pregleda stanja Unixa na področju razvoja računalnikov in tržnih smernic v svetu je predstavljena tudi naloga IDC pri uveljavljanju in uporabi Unixa.

1. UNIX

Unix je ime za skupino operacijskih sistemov, ki se razvijajo pri firmi AT&T. Zgodovina Unixa se začne leta 1968 v Bell laboratorijih s Kenom Thompsonom, ki se je odločil, da za interno uporabo v svojem laboratoriju naredi nov operacijski sistem za računalnik PDP-7. Prva uporabna verzija je bila zaključena leta 1971. Ta operacijski sistem je zaradi svojega koncepta in uporabnosti začel postajati v okviru laboratorija zanimiv.

Kmalu so se odločili, da razvijejo tudi višji jezik v katerem bo napisan operacijski sistem. Razvoj tega jezika je vodil Dennis Richie. Prvo ime za ta jezik je bilo B, kasneje C. S tem, da je bil operacijski sistem napisan v višjem jeziku se je pojavila tudi možnost enostavnega prenašanja na druge računalnike. S PDPja so prenesli operacijski sistem na računalnik Interdata 8/32. Unix se začne razširjati po maju 1975, ko je bila zaključena verzija 6, široko popularnost pa je dobila verzija 7, leta 1977.

Takrat AT&T se ni imel licence za komercialno razpečevanje računalniških produktov. Unix je brezplačno prišel v razne raziskovalne institute in na univerze. Pojavile so se "Unix hiše", ki so Unix dodelovale po svoje. Kljub temu pa je večina inaic Unixa skušala slediti standardom. Ameriški poslovneži so hitro spoznali vrednost standardnega operacijskega sistema. Takoimenovani "UNIX wizard" (čarovnik), ki pozna UNIX se bo v trenutku znašel na katerem koli UNIX stroju in ni potrebna nobena uvajalna doba ali tečaji.

1.1 VERZIJE

Unix čedalje bolj pridobiva na pomembnosti in uporabnosti. Po Unixu verzija 7, je leta 1981 izdelana pri AT&Tju komercialna izdaja Unix sistem 3 in leta 1983 Unix system 5. Velikost verzije 7 je bila 10 K v C jeziku in 1K v zbirniku, sedanji Unix 5 release 3 pa obsega 2e 75K v C jeziku in 2.5K v zbirniku.



* Berkeley Unix

Na Kalifornijski Univerzi v Berkeleyu so se z veliko vnamo lotili dopolnjevanja Unixa verzije 7. Njihov Unix se je kmalu zelo oddaljil od AT&Tjevega. Vendar je zaradi domiselnih rešitev in izboljšav postal zelo popularen. Kasneje so se takoimenovane "Berklijske izboljšave" začele uporabljati tudi na drugih portacijah Unixa, vključno z AT&Tjevim. Komercialno ime za Berkeley portacijo je BSD (Berkeley Software Distribution). Berkeley Unix je najbolj razširjen na računalnikih tipa VAX. Trenutno aktualna je verzija 4.3 BSD. Inačico Berkeley Unixa prodaja DEC pod imenom ULTRIX.

* XENIX

Xenix je Microsoftova portacija Unixa z dodatki. Trenutno je to najbolj razširjena portacija Unixa na mikroročunalnike. Najbolj razširjene so portacije na Intelove mikroprocesorje tipa iAPX 80x86 in Motoroline 680x0. Poleg vseh funkcij, ki jih nudi AT&Tjev Unix V nudi se precej več. Zelo uporabna so orodja, ki omogočajo povezavo in prenosljivost na operacijski sistem MS DOS. To je razumljivo, saj iz MicroSofta izhaja tudi MS DOS.

* UNIPLUS

UNIPLUS je Unisoftova portacija Unixa z lastnimi dodatki in izboljšavami. Najbolj znana je na procesorjih MOTOROLA 680x0. Kot MicroSoft se niti UniSoft ne trudi, da bi dobil odobritev svoje portacije od AT&Tja in uporabljal ime Unix, saj jim je dovolj lastno ime.

* IZPELJANKE UNIXA

Obstaja precej operacijskih sistemov, ki imajo določene ideje pobrane iz Unixa, vendar ne izhajajo iz originalnih AT&T izvirnih programov. Do večje ali manjše mere pa podpirajo Unix kompatibilnost ali zunanji izgled. Za določene potrebe imajo prednosti pred Unixom. Tipični primeri so:

- OS-9 je primeren za aplikacije v realnem času in je zelo kompakten.
- REGULUS je primeren za aplikacije v realnem času in se skuša čimbolj približati Unix standardu.



2. STANDARDI

Možnost nabave izvirnih programov za Unix sistem in dejstvo, da so le-ti zelo prenosljivi, je privedlo do pojava različnih verzij operacijskega sistema. Vendar so mnoge firme in organizacije v svetu zainteresirane za standardizacijo Unixa, ker bi to vplivalo na zmanjšanje stroškov pisanja programske opreme ter razširilo ponudbo.

* SVID

SVID (System V Interface Definition) je specifikacija "Unix System V", ki jo je objavila firma AT&T, da bi omogočila določiti nivo kompatibilnosti drugih Unixov z njihovim produktom.

* X/OPEN

X/OPEN je organizacija, v katero so vključeni vsi največji evropski proizvajalci računalniške opreme (Olivetti, Bull, Siemens, ICL, Nixdorf itd.), pa tudi nekateri ameriški (DEC, HP, Sperry). Standard te organizacije je zasnovan na "Unix System V", s čimer si članice obetajo osvajanje večjega tržnega segmenta zaradi širše ponudbe.

* usr/group

usr/group je združenje uporabnikov Unixa. To je zelo močna skupina podprta z konkretnim tržiščem. Celo AT&T mora upoštevati standarde, ki jih je podpira usr/group.

* POSIX

POSIX je standard, ki ga predlaga organizacija IEEE v sprejem kot bodoči ANSI standard. Objavljeni dokument standardizira sistemske servise, ki jih operacijski sistem nudi (IEEE 1003.1). Delo poteka tudi na izdelavi standarda (IEEE 1003.2) za komandni interpreter (shell) in pomožne (utility) programe. Prav tako se predlaga postopek za verifikacijo ujemanja s standardom (IEEE 1003.3).

POSIX je najvažnejši poskus standardizacije, ker bo eventualno privedel do operacijskega sistema po ANSI standardu. Edini kandidat za standardni operacijski sistem je Unix, čigar lastnik (AT&T) je že objavil, da bodo nove verzije njihovega produkta usklajene s tem standardom.



3. LASTNOSTI OPERACIJSKEGA SISTEMA UNIX

* Več programski sistem

Unix omogoča, da se istočasno izvaja več programov. Podpira sinhronizacijo procesov in medprocesno komunikacijo.

* Več uporabniški sistem

Unix podpira več uporabnikov. Omogoča povezavo in zaščito med uporabniki. Ima močna orodja za administracijo uporabnikov.

* Dodeljevanje časovne rezine

Dodeljevanje časa posameznim procesom deluje na principu dodeljevanja časovne rezine s preempcijo.

* Kompatibilnost datotek, periferije in medprocesne komunikacije

Unix obravnava periferne naprave in medprocesno komunikacijo enako kot datoteke.

* Asinhroni procesi

Unix omogoča procesom, da čakajo na dogodek. S tem ne trošijo računalniškega časa.

* Hirarhični sistem datotek

Sistem datotek je urejen na disku v hirarhičnem (drevesnem) smislu. Omogoča transparentno montiranje drugih sistemov datotek v osnovno hirarhijo.

* Visoka stopnja prenosljivosti

Unix je kot operacijski sistem sam enostavno prenosljiv na različne računalnike. Po drugi strani pa so programi, ki so razviti na enem Unix računalniku enostavno prenosljivi na drugega.

3.1 GRADNJA PROGRAMSKIH REŠITEV POD UNIXOM

Filozofija Unixa je, da nudi primerno okolje in primitive za gradnjo kompleksnih programskih rešitev iz preprostih modulov. Uporabljamo lahko programe iz sestave Unixa in lastne programe.

Ena od osnovnih primitiv je možnost preusmeritve vhodno izhodnih kanalov. Primer:

```
nroff -mm < doc1 > doc1.out 2>errors
```

Oblikovalec teksta `nroff` čita vhodno datoteko `doc1`, preusmeri izhod na datoteko `doc1.out` in preusmeri izpisovanje napak na datoteko `errors`.



Druga zelo uporabna primitiva je pipe. Pipe omogoča prenos toka podatkov iz pišočega procesa v čitajoči. Primer:

```
ls : wc -l
```

Izpis direktorija napeljemo v program za štetje vrstic. Rezultat je število vrstic v direktoriju. Uporaba pipea zmanjša potrebo za uporabo začasnih datotek.

Navedene možnosti omogočajo elegantno integracijo programov.

3.2 SISTEMSKÉ USLUGE

Unix se deli na uporabniški del, to je del, ki je neposredno dostopen uporabniškim programom in jedro.

Jedro operacijskega sistema opravlja za uporabnika različne sistemske funkcije. Tipične funkcije so:

- * Kontrola izvajanja procesov s kreacijo, terminacijo, suspenzijo in komunikacijo.
- * Razvrščanje procesov. Jedro mora na pravičen način dodeljevati časovne intervale posameznim aktivnim procesom in omogočiti njihovo izvajanje. Po preteku določenega časa, odstavi izvajanje procesa in dodeli procesor drugemu procesu. Kasnje zopet omogoči izvajanje odstavljenega procesa
- * Dodeljevanje spomina za izvajanje procesa. Jedro dovoljuje procesom, da uporabljajo dele spomina, medtem ko ščiti druge dele. Če zmanjka spomina jedro sprosti posamezne dele spomina s tem, da jih začasno prepíše na zunanji pomnilnik (swap). Če jedro prepíše celoten proces na zunanji pomnilnik gre Swapping verzijo Unix jedra, če pa prepíše le posamezne strani procesa, gre za Paging verzijo Unixa.
- * Dodeljevanje zunanjega pomnilnika za učinkovito pisanje in citanje uporabniških podatkov. Ta usluga omogoča upravljanje s sistemom datotek.
- * Omogoča doseganje perifernih enot kot so terminali, diski, tračne enote in mreže.

Jedro omogoča uporabo svojih uslug enotno in transparentno ne glede na različne fizične karakteristike sistema in perifernih enot.

Jedro nudi procesom, ki se izvajajo na uporabniški ravni vse potrebne usluge, ki jih nudi lahko samo jedro. Usluge, ki se pa lahko opravijo na uporabniškem nivoju, pa opravljajo programi, ki se izvajajo na uporabniškem nivoju. Ti programi so lahko standardni Unix programi ali pa programi, ki jih napiše uporabnik sam.



Primer programa, ki nudi usluge na uporabniškem nivoju je komandni interpreter (shell). Komandni interpreter uporablja osnovne usluge jedra: pisanje in čitanje terminala, nalaganje in izvajanje programov, preusmerjanje kanalov, pipe itd. Uporabniški programi pa uporabljajo usluge komandnega interpreterja. Uporabnik lahko napiše tudi svoj komandni interpreter.

4. KAJ UNIX PONUJA SVOJIM UPORABNIKOM

4.1 PROGRAMERJEM

Večina operacijskih sistemov je bila projektirana med drugim tudi z namenom, da učinkovito izrabi zmogljivosti specifične računalniške arhitekture in ponudi tržno zanimiv produkt.

Unix je nastal kot izraz nezadovoljstva programerjev z delovnim okoljem, ki so ga nudili tedanji operacijski sistemi. Posledica tega je, da je Unix visoko prenosljivo, enostavno, a vseeno zelo zmogljivo orodje za razvoj programov. Prenosljivost je pogojena s previdno (včasih tudi konzervativno) uporabo eksotičnih zmogljivosti specifične računalniške arhitekture, medtem ko moč in enostavnost uporabe izhajata iz usmeritve načrtovalcev, da rešijo konkretne probleme razvoja programske opreme in ne nekakšnih vsesplošnih potreb uporabnikov na trgu.

Unix se sestoji iz velikega števila majhnih, enostavnih in dobro premišljenih komponent, ki jih je mogoče izredno enostavno in učinkovito kombinirati. Na ta način lahko izkušen programer hitro in enostavno iz obstoječih (preverjenih) gradnikov sestavi programsko rešitev, ki jo tudi zelo lahko vzdržujemo.

Prenosljivost programov je posledica uporabe visjih programskih jezikov in neodvisnosti programov od vhodno-izhodnih perifernih naprav. Razen tega sta nabor in način uporabe sistemskih servisov dobro definirana in stabilna.

Poleg navedenih splošnih ugodnosti nudi Unix sistem tudi veliko število modulov in programov, ki so razvoju programov posebej namenjeni. V nadaljevanju navajamo kratek opis nekaterih najbolj priljubljenih orodij:



* programski jezik C

Ker je sam operacijski sistem pisan v tem jeziku, pomeni, da je to tudi najboljše podprt jezik na Unixu. Razen prevajalnika in pripadajočih knjižnic obstaja še cela vrsta pomožnih programov, ki olajšujejo razvoj C programov:

- lint delno semantično preverja programe in opozarja na neprenosljive programske konstrukte
- cb (C beautifier) ureja C programski tekst po splošno veljavnem formatu
- sdb, adb pomožni program za odkrivanje napak v C programih
- lex, yacc orodje za podporo pri pisanju jezikovnih procesorjev (prevajalnikov, interpreterjev ipd.)
- lex generira podporo za leksično analizo izvirnega programa
- yacc pa za analizo LALR(1) gramatike

* sccs

Orodje za vzdrževanje izvirnih modulov kompleksnih programskih sistemov. Omogoča spremljanje, dobavljanje in arhiviranje vseh verzij programskih modulov.

Obenem omogoča sinhronizacijo dela večjega števila programerjev na kompleksnem projektu.

* make

Program avtomatizira postopek prevajanja in povezovanja kompleksnih programov, sestavljenih iz večjega števila medsebojno odvisnih modulov. Prevajajo se samo tisti moduli, ki so bili spremenjeni od zadnjega prevajanja (in seveda tisti, ki so od njih odvisni!).

* curses

Paket za pisanje programov za delo z zaslonskimi terminali najrazličnejših tipov. Sestoji iz podprogramov, ki znajo deliti zaslon na več neodvisnih "oken", skozi katera se piše ali bere. Tako napisani programi niso vezani za katerikoli tip terminala. Sekvence za krmiljenje prikazovanja na zaslonu se nahajajo v podatkovni bazi paketa, v katero zlahka dodamo opis zmogljivosti in krmilne sekvence popolnoma novega tipa terminala.

* množica drugih programskih jezikov

Na voljo je cela vrsta prevajalnikov in interpreterjev raznih dobaviteljev programske opreme. Razen standardnih jezikov kot so FORTRAN, COBOL, Pascal in Lisp obstaja vrsta prevajalnikov za specialne jezike, ki so bili razviti na Unixu (fp, ratfor, efl).



4.2 UPORABNIKOM

* komandni interpreter (shell)

Uporabnik krmili sistem s pomočjo programa, ki terminalski dialog transformira v invokacije sistemskih servisov, ki proizvedejo zeljene učinke. Karakteristično za Unix je, da komandni interpreter (shell) ni zabetoniran v jedro operacijskega sistema, temveč je to povsem navaden "uporabniški" program, kar pomeni, da ga lahko napiše vsak programer. Mnogi so to tudi storili in je zato uporabnikom na voljo široka paleta komandnih interpreterjev. Povsem naravno je, da so najpopolnejši in najbolj razširjeni komandni interpreterji tisti, ki so jih napisali tvorci Unixa za lastno uporabo.

Popularni "shell programi" so interpreterji izredno močnih programskih jezikov za upravljanje z računalniškim sistemom. Z uporabo teh jezikov lahko izkušen uporabnik hitro in enostavno spremeni obliko komand ter na ta način prekroji dialog s sistemom po svojem okusu. Tako definirane "komande" so v bistvu mini aplikacije, ki pogosto inicirajo istočasno izvajanje večjega števila programov, pri čemer se izhodni podatki enega programa usmerjajo na vhod naslednjega v sekvenci (pipeline).

Komandni interpreterji, ki navdušujejo programerje in izkušene uporabnike so istočasno tudi predmet žolčnih razprav in kritik. Povsem naravno je, da idealno orodje za izkušene mojstre ne more biti hkrati tudi idealno orodje za začetnika ali občasnega uporabnika računalniškega sistema. Najbolj pogosta je obtožba, da je Unixov shell kriptičen in nerazumljiv. Na prvi pogled se zdi, da je to tudi res, saj je shell pisan za hiter vnos kompleksnih komand, ki bi ob večji verboznosti postale predolge in nepregledne.

Če so uporabnikove potrebe drugačne, je možno problem rešiti s prilagajanjem obsoječega shell programa ali z izbiro primernejšega.

* datoteke

Uporabnikom je Unix nepogrešljiv kot enostaven in zelo fleksibilen medij za vzdrževanje vseh vrst podatkov. Le-ti so shranjeni v datotekah brez posebne notranje strukture. Posamezni programi (kot n.pr. editor) seveda interpretirajo vsebino datoteke na svoj način (kot vrstice s tekstom). V imenih datotek se lahko pojavljajo praktično vsi ASCII znaki, kar omogoča uporabo preglednejših imen, kot so na primer "25-Maj-1987" ali "LetnaSola?". Datoteke so hierarhično katalogizirane, kar pomeni, da so sezname datotek (directories) tudi sami datoteke in so s tem lahko vpisani v drugi seznam (directory).



* komuniciranje

Unix služi uporabnikom tudi kot sredstvo za medsebojno komuniciranje. Razen skupnih podatkov v datotekah, do katerih kontrolira dostop operacijski sistem, je uporabnikom na voljo sistem elektronske pošte in povezovanja v mrežo z drugimi sistemi. Elektronska pošta omogoča poleg pošiljanja in sprejemanja sporočil tudi njihovo klasifikacijo in shranjevanje v fascikle.

Povezovanje v mreže se lahko izvede tudi preko navadnih terminalskih kanalov in ne zahteva dodatne opreme, pri tem so lahko zveze komutirane ali najete. Če obstaja naprava za avtomatsko pozivanje (Auto Calling Unit), je možno, da se računalniška mreža avtomatično vzdržuje skozi javno komutirano telefonsko omrežje. Sporočilo za uporabnika na oddaljenem sistemu je mogoče oddati kadarkoli, poslano pa bo takrat, ko bo zveza s tem sistemom vzpostavljena (bodisi ročno ali avtomatično).

* obdelava tekstov

Računalniki namenjeni kot stroji za računanje se čedalje več uporabljajo kot stroji za obdelavo teksta. Na Unixu se skoraj vsi podatki hranijo v tekstualni obliki. Razlog za to je intuitivna razumljivost teksta kakor tudi dobra opremljenost z orodji za njihovo obdelavo. Tekstovne informacije se navadno nahajajo v datotekah. Obdelava sestoji iz "prepuščanja" teksta skozi program (imenovan tudi "filter"), ki vrši razne transformacije. Rezultat se lahko shrani v drugo datoteko, prikaže na ekran, ali pa se usmeri na vhod v drugi program, ki bo izvedel naslednjo fazo transformacije.

Standardni "filtrski" programi izvajajo operacije sortiranja, komparacije, transliteracije in razne druge transformacije tekstov. Nad primerno formatiranimi tekstovnimi datotekami (tabele) so podprte tudi relacijske operacije: select, project in join.

Za ilustracijo naj povemo, da so tudi datoteke, ki jih uporablja operacijski sistem, shranjene v tekstovni obliki. Spisek uporabnikov, spisek sistemov v mreži, opis karakteristik terminalov itd. so primeri takih datotek, ki jih sistemski administrator vzdržuje s standardnimi programi za obdelavo tekstov.

Primer:

Sortiran spisek uporabnikov na sistemu naredimo tako, da z uporabo programa cut (ki implementira relacijsko operacijo "project") izvlečemo prvo kolono iz datoteke s podatki o uporabnikih, rezultat pa usmerimo v program za sortiranje (v tem primeru po ASCII sekvenci).

```
cut -f1 -d: /etc/passwd ! sort
```


* analiza numeričnih podatkov

S pomočjo paketa stat je možna statistična obdelava numeričnih podatkov, ki se nahajajo v tekstovnih datotekah (lahko tudi med narativnim tekstom). Komponente paketa lahko s pomočjo "cevi" (pipes) kombiniramo v mrežo, skozi katero zatem spustimo podatke na obdelavo.

Nad podatki lahko uporabimo aritmetične funkcije, napravimo običajne izračune kot so sumiranje, korelacija in linearna regresija. Dobljene rezultate lahko prikažemo na zaslonskem terminalu, če pa imamo grafično opremo, paket podpira tudi prikazovanje v grafični obliki (histogrami, "kolač" diagrami, x-y graf itd.)

* priprava dokumentov

Dokumenti se vnašajo s pomočjo zaslonskega urejevalnika besedil v datoteko. Za formatiranje dokumenta v željeno obliko obstaja cela vrsta programov. Njihov rezultat so pisma, referati ali knjige, ki vsebujejo vneseno besedilo, prikazano z različnimi oblikami črk, oblikovanimi tabelami, matematičnimi formulami in risbami.

Kvaliteta izhodne oblike besedila je odvisna od razpoložljive strojne opreme. Obstajata dva takšna programa, s pomočjo katerih je možno z malo truda isti tekst obdelati na oba načina:

nroff formatira besedilo za tiskanje v obliki "pisalni stroj"

troff formatira besedilo za laserske tiskalnike, fotostavne stroje in podobno

Dodatni programi služijo za izdelavo bibliografije, stvarnih in abecednih kazal. Za angleški jezik obstajajo tudi programi za preverjanje pravopisa in "bralnosti" besedila: spell, diction, style.

4.3 SISTEMSKEMU PROGRAMERJU

Za systemskega programerja sta najbolj pomembni odprtost in fleksibilnost sistema. Systemskih servisov je malo in so relativno stabilni.

Tudi koncept datoteke na Unixu nudi velike prednosti. Datoteka je samo niz znakov, vhodno-izhodne operacije na perifernih enotah so identične operacijam na datotekah. Pravzaprav je vsaka periferna naprava na sistemu reprezentirana z datoteko ustreznega imena. Operacijski sistem pa vhodno-izhodne operacije na to datoteko preusmeri na ustrezno periferno enoto. To pomeni, da se lahko isti program uporablja za pisanje na disk, terminal ali tračno enoto.



Zelo pomembno je tudi to, da je interna struktura operacijskega sistema znana in objavljena v mnogih publikacijah, kar znatno olajšuje pisanje sistemskih programov.

4.4 ORGANIZACIJI

Baziranje na Unixu nudi organizaciji, ki ga uporablja, vse prednosti istega operacijskega sistema na računalnikih od mikroročunalnika do main-frame sistema. Sistemi različnih dobaviteljev so med seboj kompatibilni.

Ena od posledic visoke stopnje prenosljivosti Unixa je, da večina proizvajalcev opreme za nove arhitekture in tehnologije izbirajo ravno ta operacijski sistem. To pomeni, da Unix omogoča izrabo prednosti, ki jih nudi hiter razvoj na področju strojne opreme brez mučnega prehoda na nov operacijski sistem.

Na ta način organizacija ščiti svojo investicijo v programsko opremo, ker ima garancijo, da bodo programi delovali tudi na novi strojni opremi. Portabilnost programske opreme je verjetno največja prednost Unixa, čeprav je pri tem potrebno upoštevati dejstvo, da je tudi pod Unixom možno napisati program, ki ne bo prenosljiv na drugi sistem. Prednost Unixa z ozirom na druge operacijske sisteme je v tem, da je tu lažje napisati portabilen program, ne pa da je vsak program avtomatično prenosljiv.

5. PODROČJA UPORABE

5.1 ZNANSTVENO-TEHNIČNO OKOLJE

Če pod tem naslovom razumemo izobraževalne ustanove, univerze, inštitute in laboratorije, dobimo skupino uporabnikov, med katerimi je Unix pravzaprav nastal. Poleg razloga portabilnosti je na njegovo povečano popularnost prav gotovo vplivalo dejstvo, da se je že nekaj generacij studentov v razvitih deželah izsolalo na Unixu. Še danes je v rabi kot osnova za pedagoško in raziskovalno dejavnost na področju operacijskih sistemov.

V tem okolju najbolj pride do izraza fleksibilnost in odprtost sistema, kot tudi možnost zadovoljevanja potreb heterogenih skupin uporabnikov na raznorodni strojni opremi. Najpomembnejša je uporaba paketov za statistiko, pripravo dokumentov in širok izbor prevajalnikov za programske jezike.



5.2 CAD/CAM IN CAE

Na tem področju je že težko najti sistem, večji od osebnega računalnika, ki ni baziran na Unixu. Razlog za to tiči v prenosljivosti in adaptibilnosti operacijskega sistema na posebno strojno opremo, ki se v CAD/CAM in CAE sistemih uporablja. To velja tako za nove, zmogljivejše procesorje, kot tudi za specializirane koprocesorje in inteligentne periferne krmilnike.

5.3 POSLOVNI INFORMACIJSKI SISTEMI

V času začetkov Unixa je bil le-ta po marsičem nesprejemljiv kot osnova za poslovne aplikacije. Od takrat se je veliko tega spremenilo v Unixu, v poslovnem svetu in tudi v aplikacijski programski opremi.

Prvotne verzije Unixa niso bile primerne za rabo v poslovne namene iz naslednjih razlogov:

- * Neučinkovito delo z velikimi datotekami. Unix je optimiziran za delo z velikim številom malih datotek in za optimalno izrabo prostora na diskih.
- * Napake v delovanju strojne opreme so hitro privedle do izpada celega sistema ali do izgube podatkov.
- * Komandni jezik (shell) vrača zelo skope informacije o napakah.
- * Poslovne aplikacije so bile "batch" obdelave na "velikih" računalnikih, medtem ko je bil Unix izrazito interaktiven sistem za majhne računalnike.

Danes se Unix vse pogosteje uporablja na računalnikih za poslovne aplikacije. Nekatere spremembe, ki so do tega privedle, so:

- * Unix je dodelan, zelo dobro reagira na napake v delovanju strojne opreme.
- * Široka uporaba sistemov za upravljanje z bazami podatkov (DBMS) namesto datotek omogoča tako varnost kot tudi hitro dobavljanje podatkov iz kompleksnih in velikih podatkovnih baz.
- * Sodobni trend k distribuirani obdelavi podatkov in interaktivnemu delu je načel mnoga področja uporabe, za katera je Unix idealno kvalificiran. Supermikro in mini računalniki z Unixom se se posebej uporabljajo za gradnjo informacijskih sistemov za podporo poslovnih odločitev in za avtomatizacijo pisarniskega poslovanja (office automation).
- * Unix je dobavljiv tudi na "največjih" sistemih, poleg tega je dobavljiv tudi prevajalnik za Cobol, ki je tradicionalni jezik poslovnih aplikacij. Implementirani so tudi drugi potrebni mehanizmi kot so "ISAM", "file locking" in "record locking".



- * Moderne rešitve, na primer relacijske baze podatkov in nove generacije programskih jezikov, so tudi že dobavljive na Unix sistemih. Pri mnogih uporabnikih se namesto "shell" komandnega interpreterja uporabljajo t.im. jeziki četrte generacije.
- * Kot (edini) kandidat za standard operacijskega sistema in njegova priljubljenost v razvojno-raziskovalnem okolju garantira, da se bodo tudi nove generacije programske opreme pojavile ravno na Unixu.

5.4 INDUSTRIJSKE APLIKACIJE

Največja ovira za močnejši prodor Unixa na področje industrijskih aplikacij je dejstvo, da ni zastavljen kot real-time operacijski sistem. Osnovni problem ni v tem, da Unix počasi reagira na zunanje dogodke. Če bi bilo tako, bi bilo tudi interaktivno delo nemogoče. Odzivni čas na Unixu je odvisen od obremenitve sistema in ni možno favorizirati procesov, ki bi opravljali časovno kritične operacije.

Ta problem je rešljiv in to na več načinov. Eden je ta, da se Unix poenostavi in s tem pridobi odlike real-time sistema. Takšen program se poveže s standardnim Unixom na istem ali drugem procesorju.

Druga možnost je uporaba operacijskega sistema kot je OS-9, ki predstavlja adaptacijo Unix konceptov na real-time zahteve. Ker je pri tem ohranjen Unixov programski interface, se na ta način doseže določena stopnja kompatibilnosti.

6. PRIHODNOST

Proces komercializacije Unixa se bo z novimi poskusi standardizacije gotovo nadaljeval. Pričakovati je, da bo Unix System V postal standard. Najmočnejša konkurenta, Xenix in Berkeley, sta odpadla, ker je Xenix že kompatibilen z Unixom, Berkeley pa ne želi biti omejen s standardi.

Vstop IBM-a na Unix tržišče bo vsekakor vzbudil povečano zanimanje za uporabo v poslovnih sistemih. Pri tem pa ostane odprto vprašanje, kakšne posledice utegne imeti poskus IBM-a, da vzpostavi svoj Unix standard.

Dandanašnji je veliko pozornosti namenjeno distribuiranim sistemom in grafiki. Na področju Unix operacijskih sistemov so močne težnje, da se ta področja prav tako standardizirajo. Dva produkta se smatrata kot izrazita kandidata za standard. To sta NFS (Network File System) firme Sun in X z univerze MIT. NFS omogoča transparenten dostop do distribuiranih datotek, medtem ko X nudi obdelavo grafičnih prikazov in teksta na istem zaslonu.



Razvojna prizadevanja IDC so usmerjena k doseganju kompatibilnosti z nastajajočimi standardi. Aktivnosti v tej smeri se odvijajo v okviru projekta "Deltix", katerega cilj je izdelava standardnega Unixa za družino Delta računalniških sistemov z dodatki, ki bodo omogočali povezavo strojne opreme, kompatibilnost programske opreme in podatkov. Deltix operacijski sistem bo kompatibilen s standardom, ki bo najbolj razširjen in tržno zanimiv. V tem trenutku je to System V firme AT&T, po morebitnem sprejetju standarda pa bo to lahko tudi POSIX.

CTIVO:

- * Understanding the UNIX system - A Conceptual Guide, AT&T Documentacion 307-132, 1986
- * The Design of the Unix Operating System, M.J.Bach, Prentice-Hall, Inc, 1986
- * X/OPEN Portability Guide, ISBN 0444 87839 4



**RACUNALNIK V INDUSTRIJSKEM
OKOLJU**

Avtorja referata: Mitja Zakelj, dipl.ing.
Slavko Rožič, dipl. ing.
oba ISKRA DELTA



Ljubljana, maj 1987



UVOD

O nujnosti uvajanja računalnikov v proizvodnjo ne kaže izgubljati besed, čeprav to v prvi fazi pomeni odpuščanje delavcev (primer FIAT). Dejstvo je, da podjetje ali država, ki želi preživeti v današnjem konkurenčnem svetu, ne sme slepo vztrajati na ohranjanju zaposlenosti, ampak na prestrukturiranju. Znana je maksima: če želiš vztrajati na zaposlenosti, ne vztrajaj na zaposlenosti.

Računalnik v industriji upravlja tok informacij med različnimi elementi, ki sestavljajo proizvodno enoto: avtomatiziranimi stroji in procesi, oddelki in ljudmi v njih.

Ceprav nekateri učbeniki navajajo, da se je prvi regulator pojavil že 1789, prvi digitalni pa 1897 - zaradi letnice se bojim, da je bil ta prvi regulator (z malo pesniške širine) morda Robespierre - pa prve računalniške aplikacije v vodenju industrijskih procesov zaradi neprilagodljivosti, cene in majhnih procesnih moči teh računalnikov srečamo šele konec petdesetih let.

Ti prvi poskusi so dobili več posnemovalcev v sredini šestdesetih let, ko so se pojavili prvi mini računalniki s podobno močjo, a precej nižjo ceno.

Pravi prodor računalnika v industrijsko okolje pa je bil pravzaprav šele v sedemdesetih letih, ko so prišli na trg mikroprocesorji z veliko boljšim razmerjem moč/cena in, vsaj za to področje uporabe, številnimi drugimi dobrimi lastnostmi. Njihova moč je danes primerljiva z močjo manjših miniračunalnikov, le cena je približno 2/3 nižja.

Že samo cena mikroračunalnika, hiter razvoj komunikacij in novi pristopi v razvoju programske opreme, ki so računalnik približali uporabniku, njegova večja neobčutljivost - iz klimatiziranih centrov ga lahko preselimo v agresivno industrijsko okolje - in pa, zaradi te selitve, drastično zmanjšana cena kabliranja med posameznimi procesnimi enotami, so spremenili tudi pristop k procesnemu vodenju. Uveljavila se je distribuirana obdelava podatkov, ki v večini primerov poveča zanesljivost in celotno produktivnost proizvodnih linij in tvori okostje bodočih CIM (Computer Integrated Manufacturing) sistemov.

CIM ni strogo določen in enoumno uporabljan termin. Nekateri ga uporabljajo že za vsak računalniško voden proces, drugi zahtevajo, da vključuje CAD, avtomatsko regulacijo, robote, da iz proizvodnih hal popolnoma izrine ljudi, tretji pa zahtevajo celo povezavo CIM kot popolno avtomatizacijo z MIS (Management Information Systems). CIM torej predstavlja kompleksen poslovno-tehnološko-produkcijski sistem.



VODENJE PROCESOV

Industrijske procese v grobem lahko s stališča računalniškega vodenja delimo v dve, na videz povsem različni, področji:

- * področje kontinuiranih procesov (pridobivanje in prenos energije razni kemični procesi ipd.) in
- * področje diskretnih procesov (industrija kovinskih in plastičnih elementov, avtomobilska industrija, tekstilna industrija ipd.)

Tudi metode za nadzor ali avtomatizacijo teh dveh področij so bile in so deloma še popolnoma različne. Te razlike pa se kažejo v glavnem le pri reguliranju osnovnih produkcijskih enot. Ko ta najnižji nivo obdelamo, pa imata oba tipa procesov vedno več skupnega. To lepo dokazujeta tudi oba tipična predstavnika orodij za avtomatizacijo za posamezni področji - PLC (Programmable Logic Controller), razvit za vodenje diskretnih in digitalni regulator, ki navadno izvaja različne PID funkcije, razvit za vodenje kontinuiranih procesov. Ko se mikroprocesorska moč veča, lahko enemu ali drugemu dodajamo različne funkcije, tako da se meja med njima briše. Čim više v omenjeni hierarhični strukturi procesnega vodenja gremo, tem več podobnosti zasledimo.

Kljub različnosti industrijskih procesov, ki jih nadziramo in vodimo, mora vsak sistem vodenja zagotoviti:

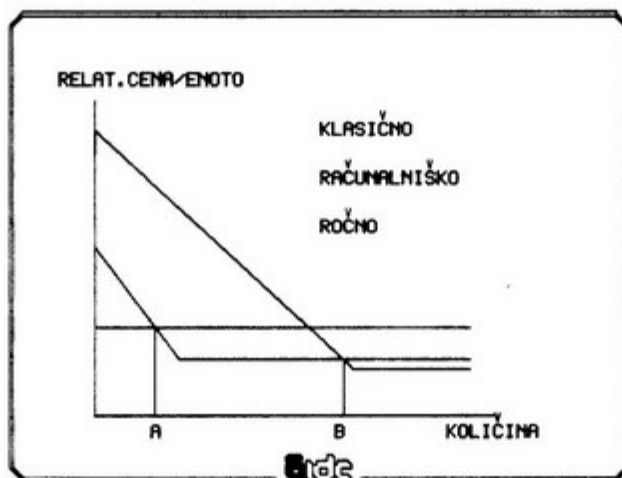
- * zajemanje in spremljanje velikega števila procesnih veličin. Te so lahko po naravi zelo različne in raztresene po velikih področjih, prevedemo pa jih vedno v signalizacije, meritve ali števe;
- * obdelavo teh veličin. Ta je lahko preprosta - le prikaz spremembe, lahko pa je le droben faktor v kompleksni regulacijski enačbi;
- * prikaz trenutnega stanja procesa, ki je za različne hierarhične nivoje vodenja spet lahko različen;
- * prenos avtomatskih ali ročno vnešenih ukazov nazaj v proces.

Kompleksnejši sistemi pa lahko vsebujejo še:

- * izbor posameznih pomembnih podatkov iz množice trenutnih procesnih veličin in vnos le-teh v historično bazo podatkov za kasnejše analize in obdelave;
- * optimizacijo vodenih procesov po različnih kriterijih;
- * prilagoditev produkcije trenutnim potrebam ob upoštevanju varnosti in optimalnega izkoriščanja, npr. izdelava avtomobila po naročilu.



Ce sploh ne upoštevamo očitnih prednosti, ki jih nudi mikroprocesorska avtomatizacija pred klasično mehansko ali električno zaradi množice informacij, ki so vsak hip na voljo in prilagodljivosti trenutnim potrebam, ampak gledamo le relativno ceno na enoto proizvoda glede na proizvedeno količino (glej sliko 1), vidimo, da so za majhne količine najnižja vlaganja v ročno proizvodnjo, za srednje serije je najugodnejša mikroračunalniška avtomatizacija, za zelo velike pa fiksna, klasična. Tudi če bi bila razlika v ceni med načinoma avtomatizacije večja, pa so prednosti, ki nam jih nudi programirna avtomatizacija tako očitne, da o izbiri ne bi smelo biti dvoma.



Slika 1.

Vodenje procesov torej obsega:

- * Regulacijo vsake posamezne enote v procesu. S tem zagotovimo optimalno delovanje posameznih enot, povečamo zanesljivost, zmanjšamo porabo energije, materiala, zaznamo in kompenziramo neregularna stanja.
- * Nadzor posameznih enot in celotnega sistema, primerjavo vrednosti spremenljivk s postavljenimi mejnimi vrednostmi, alarmiranje ...
- * Optimizacijo glede na fiksne ali spremenljive kriterije. Optimiramo lahko na nivoju proizvodne enote, grupe ali postopka.
- * Koordinacijo medsebojno odvisnih enot v sistemu vodenja
- * Upravljanje, to je prilagajanje planom in trenutnim zahtevam

To so seveda le osnovne naloge računalnika, ki vodi določen proces. Poleg teh je potrebnih še precej funkcij, ki niso nič manj zahtevne, le da za neposrednega uporabnika niso interesantne, kot je to npr. komunikacija med posameznimi distribuiranimi mikroročunalniki, diagnostika in avtodiagnostika, ki izboljša zanesljivost celotnega sistema ipd.

Vsakemu nivoju funkcij lahko priredimo svoj računalnik, navadno celo več, odvisno pač od zahtevnosti procesa, vse funkcije pa so načeloma lahko združene tudi v enem močnejšem računalniku.

Funkcije distribuiranega procesnega vodenja lahko neposredno primerjamo z obstoječo strukturo upravljanja v podjetju.

Poleg podobnosti distribuirane obdelave podatkov s hierarhično strukturo klasičnega ročnega vodenja pa ima večračunalniška struktura pred enoračunalniškim sistemom se naslednje prednosti:

- * vsak mikroročunalnik je samostojna enota. Odgovornost je zato distribuirana, izpade je lažje nadomestiti, zanesljivost delovanja celotnega sistema je večja. Se tako dober sistem je povsem nesprejemljiv, če ni zanesljiv
- * viri podatkov so navadno raztreseni po objektu. Pri večračunalniških sistemih odpade veliko kabliranja, odzivi na spremembe so lahko hitrejši.
- * v kompleksnejših sistemih vodenja je navadno potreben večračunalniški centralni nadzorni sistem. Hierarhično distribuirano obdelovane procese lažje preslikamo v centralne sisteme.
- * zmanjša se obremenitev posameznih procesorjev, zato je lažji tudi razvoj kontrolnih in servisnih programov ter večja možnost diagnostike. Poveča se sposobnost zaznave nenadnega velikega števila dogodkov.

Taka razdelitev pa prinese seveda tudi nekaj problemov:

- * pojavi se problem medprocesorskih komunikacij, zlasti zaradi se ne popolnoma uveljavljenih standardov za to področje, zahtevanih hitrosti in zanesljivosti delovanja v navadno zelo motenem industrijskem okolju. V nekaj letih bo to verjetno izpiljena verzija MAP/TOP, ali za skromnejše potrebe, njun podsklop. Zgodnje aplikacije so uporabljale povezavo v zvezdo, vse enote so bile s centralno povezane preko svojih lastnih linij. Te povezave so zlasti v večjih sistemih predrage. Tudi standardne mreže, ki se uporabljajo pri običajni obdelavi podatkov, navadno v real-time svetu ne pridejo v poštev. Danes se najbolj uporabljata dve varianti: token passing (zeton na vodilu) - izmenično dodeljevanje pravice komuniciranja posameznim enotam in pa multidrop polling sistem, kjer je le en procesor stalno glavni in on poziva ostale in zbira podatke od njih. Master je lahko eden od procesnih računalnikov ali pa poseben procesor. Mi za ta namen, če le zahteve niso prehude, uporabljamo inteligentni asinhroni vmesnik na VME vodilu.



* v zelo distribuiranem vodenju se na višjih nivojih podaljša pristopni čas do nekaterih podatkov, ker večina osnovnih informacij ni neposredno dostopna.

Res je tudi, da se kapacitete računalnikov neprestano večajo in da poprečen mikroračunalnik lahko kontrolira več različnih procesov in opravlja še dodatne naloge z višjih nivojev, in da so na voljo nova programska orodja, ki razvoj in implementacijo procesne aplikacije zelo pohitrijo.

Stopnja distribucije je torej odvisna od specifikacije procesa, zahtevane hitrosti odziva - navadno je važnejša hitra reakcija na spremembo v procesu kot pristopnost podatkov na višjem nivoju - zahtevnosti obdelav in prikazov ter prostorske porazdelitve nadziranega procesa. Večja distribuiranost pa omogoča tudi postopno izgrajevanje sistema nadzora in vodenja industrijskega procesa in povezavo na MIS sistem, seveda le, če se držimo že uveljavljenih mednarodnih standardov ali vsaj standardov večjih proizvajalcev.

DELTINA POT

Cepprav IDC z različnimi paketi podpira večino funkcij, ki se pojavljajo pri celostnem nadzoru in vodenju procesov, od zajema podatkov in regulacij, CAD/CAM sistemov, raznih optimizacij pa do vodenja skladišč in poslovnih obdelav, pa se uporabniki redko odločajo za kompleksne sisteme vodenja. Vzroki so različni: nepoznavanje možnosti, finance ... , prav tu pa si uporabniki sami velikokrat zapirajo vrata. Odločajo se za ceneje sisteme vodenja majhnih proizvajalcev, ki jim morda celo trenutno zadoščajo, takoj ko pa jih želijo dopolniti ali povezati v kompleksnejši sistem, pa se pojavijo problemi. Na nekaterih aplikacijah se je pokazalo, da je ceneje zamenjati obstoječo opremo kot jo povezati v sistem. Ker pa je napačno odločitev težko priznati, se teže pa dobiti denar za popravljanje napak, raste relativno malo sistemov. Večina ostane le pri regulaciji in nadzoru.

Postaja DIPS-M spada v kategorijo daljinskih postaj z majhnim številom vhodno-izhodnih signalov in serijsko povezavo na centralni računalnik.

Kot taka predstavlja sorazmerno poceni rešitev povsod tam, kjer je število vhodno/izhodnih signalov majhno ali pa so ti na široko raztreseni po objektu.

Večje število takih postaj je možno preko serijske linije povezati na poljuben centralni računalnik (TRIGLAV, PARTNER, DELTA 800 ipd.), ki ima vmesnik za serijsko komunikacijo in tako postaviti učinkovit distribuiran sistem za daljinski nadzor in vodenje. Možna je tudi priključitev na daljinske postaje družine DIPS 85 v vlogi majhne satelitske postaje. Ker je postaja DIPS-M mikroročunalniška in ima fleksibilno programsko opremo, jo je možno uporabljati v razne namene, kot npr.:

- * daljinska postaja
- * koncentrador podatkov
- * lokalni avtomat
- * regulator

Postaja je zgrajena tako, da kljubuje tudi težkim pogojem industrijskega okolja, kot so: velike temperaturne spremembe, prah, elektromagnetne motnje ipd.

Vsi vhodno/izhodni priključki so galvansko ločeni od zunanosti ter zaščiteni proti motnjam, prenapetostim in kratkim stikom.

Vhodni del digitalnih in analognih vhodov ter izhodni del posameznih digitalnih in analognih izhodov je narejen s posebnimi zamenljivimi miniaturnimi moduli. V isti konektor lahko vstavimo katerikoli modul in tako določimo vrsto priključka. Bogat izbor modulov (različne vhodne in izhodne napetosti ali tokovi) nam omogoča, da lahko prilagodimo postajo najrazličnejšim zahtevam.

Zaradi izredno majhne porabe lahko deluje v neprodušnem ohišju brez ventilacije tudi v širokem temperaturnem območju.

Napaja se lahko s celim spektrom standardnih izmeničnih ali enosmernih napetosti in je primerna tudi za baterijsko napajanje. Zaradi majhnega števila elementov in velike stopnje integracije ima veliko obratovalno zanesljivost.

Za uporabnika je še posebno primerna zaradi enostavnega vzdrževanja in servisiranja. Vsi digitalno vhodno/izhodni signali imajo za indikacijo LED diodo. Ob morebitni okvari je možno celotno postajo hitro zamenjati z rezervno.

Priključevanje zunanjih kablov je enostavno in zanesljivo ter izvedeno direktno na tiskano vezje preko posebnih vrstnih sponk.

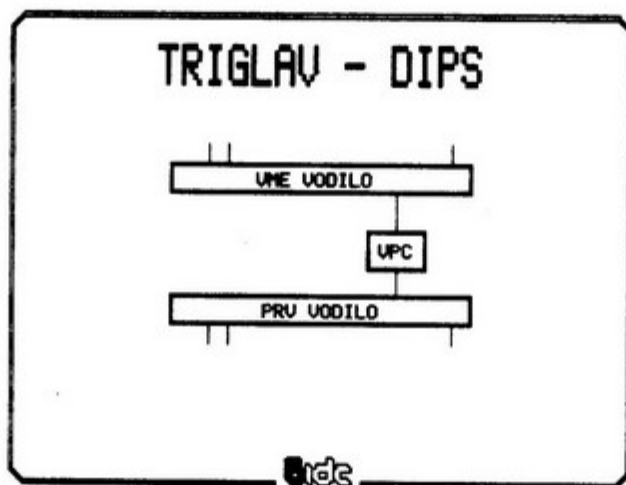
Zaradi majhnega kompaktnega ohišja je primerna za stensko montažo tudi v primerih, ko je na razpolago malo prostora.

Osnovne tehnične karakteristike postaje DIPS-M:

- * CPE: HD6309E (CMOS verzija MC6809)
64 KB RAM/EPROM
- * Komunikacije: serijski priključek 50 Bd do 19200 Bd z RS-232, RS-422 ali RS-485
možnost vgradnje modema 1200 Bd (opcija)
dodatni serijski priključek 50 do 19200 Bd z RS 232
DCP protokol
- * Stevilo signalov: skupaj največ 32, od tega največ 24 digitalnih (do 24 vhodov, do 16 izhodov)
največ 8 analognih (do 8 vhodov, do 4 izhodi) z določenimi omejitvami lahko stevilo vhodov povečamo s pomočjo multiplekserja na največ 144 in izhodov na 64 (ne hkrati)

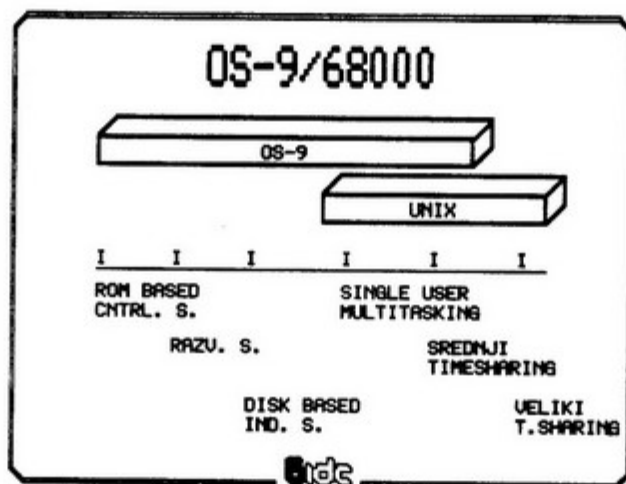
DIPS-85, večje in starejše sestre te nove teleinformacijske postaje ne bom posebej predstavljal, ker jo večina verjetno pozna, omenim naj le, da smo tudi za področje procesnega vodenja izkoristili novosti v Deltinem programu - VME družino. Napravili smo vmesnik, s pomočjo katerega lahko na VME vodilo priključimo vso procesno periferijo iz družine DIPS (slika 2) - torej A/D in D/A pretvornike, module za digitalne vhode in izhode ipd. katerikoli računalnik iz družine Triglav v poljubni konfiguraciji - mislim na različne CPU in operacijske sisteme z različnimi zunanji spominskimi enotami, komunikacijskimi vmesniki, grafiko itd. -lahko zajema podatke iz procesa in ga vodi. Aplikacije na taki konfiguraciji so že v uporabi.





Slika 2.

Se en pomemben korak smo napravili v tej smeri. Na procesor z M68010 smo portirali operacijski sistem OS-9, ki je po strukturi zelo podoben UNIXu (slika 3) - programi pisani v C so med operacijskima sistemoma prenosljivi, je pa real-time in popolnoma ROM-able, podpira visje programske jezike, ker pa je postal eden vodilnih real-time operacijskih sistemov, vsaj na Motorolnih procesorjih, tudi za orodja ni problem.



Slika 3.

Ko sem opisoval teleinformacijsko postajo DIPS-M, sem že omenil, da lahko služi tudi kot regulator. Programska oprema za to je razvita. Omogoča preprosto realizacijo poljubnega števila regulacijskih zank s poljubnim številom regulacijskih blokov. Omejitev je le čas izvajanja. Programski paket vsebuje urejevalnik oz. editor, s pomočjo katerega uporabnik sestavi regulacijsko shemo in jo shrani v posebno datoteko. Drug program to tabelo prevede v regulacijski program v C jeziku, iz tega pa C prevajalnik s pomočjo regulacijskih funkcij iz knjižnice realizira zeleno regulacijsko shemo.

Knjižnica funkcij obsega vse standardne, kot so:

- * P, PD, PI, PID, PIDD regulatorji
- * regulacijska razmerja
- * regulatorji s preklopom ročno/avtomatsko z blaženjem skokovitih sprememb signala pogreška pri preklopu
- * splošni dinamični regulatorji
- * lead-lag dinamični kompenzatorji
- * funkcijski generator
- * alarmi
- * bloki z logičnimi operacijami
- * zakasnitve
- * blok za integracijo ipd.

Regulator lahko deluje samostojno, lahko pa komunicira z nadrejenimi računalniki, kjer teče npr. paket SCADA/D, ki omogoča, kot že ime pove (Supervisory Control and Data Acquisition) zbiranje in obdelavo podatkov, prikaz stanja in nadzor procesa ter arhiviranje zbranih podatkov in komunikacijo z enakovrednimi ali nadrejenimi računalniki.

Definicija vodenja procesov pravi, da procesni računalnik upravlja tok informacij in materiala med različnimi elementi procesa. Do nedavna je bil hud problem identifikacija in spremljanje pretoka materiala. Statistika kaže, da so sistemi črtne kode pri tem 75% hitrejši in 15 000 krat bolj zanesljivi od ročnega vnosa. Takih prednosti ne gre prezreti. Poleg čitalcev in dekoderjev črtne kode, ki jih preko serijske linije lahko priključimo na poljuben računalnik, izdelujemo se master filme in tiskamo kode, kar je veliko bolj odgovoren posel, kot se nemara slisi.

To je le nekaj naših produktov s tega področja. O ostalih programskih paketih iz našega programa od CAD/CAM in CAE paketov, raznih optimizacijskih paketih, pa do vodenja skladišč in spremljanja projektov bo govora v drugih predavanjih. Končam naj, da smo sposobni in pripravljeni sodelovati.

**OBLIKOVANJE OEM TRZISCA
V JUGOSLAVIJI**

Avtorji referata: Janko Pučnik dipl.iur.
mag. Vanja Bufon, dipl.ing.
Tugo Borina dipl.ing., vsi ISKRA DELTA



Ljubljana, maj 1987



1. ALI JE JUGOSLOVANSKO TRZISCE PRIMERNO ZA UVELJAVITEV OEM PRODAJE

Tehnološka revolucija, ki predstavlja informatizacijo družbe ima več obrazov. Tako v sociološkem smislu predstavlja pogoj za distribucijo informacij - ob upoštevanju načel samoupravljanja pa tudi uveljavitev najširše participacije pri upravljanju.

Novo obdobje in s tem pogojena nova strategija pa predvsem omogoča tehnološko izobraženim ljudem, da združijo svoje inovativne sposobnosti z elementi visokih tehnologij. Prvič nudi domača računalniška industrija trgu nabavo računalniške strojne in programske opreme v obliki modulov, sklopov in podsklopov kot tudi operacijskih sistemov in programskih generatorjev v obliki reprodukcijskega materiala iz katerega si lahko vsak sam zgradi ali dogradi računalniški sistem (oziroma uporabi te elemente za vgraditev v svojo posebno rešitev - pa naj gre za testne naprave, numerično krmiljene stroje ali avtomatizirano linijo), da omenimo samo možnosti industrijske avtomatizacije.

2. KAJ TOREJ POMENI OEM ?

OEM - Original Equipment Manufacturing - se pojavlja praviloma pri produktih visokih tehnologij in predstavlja prodajo sklopov in podsklopov teh produktov ob določeni pomoči OEM proizvajalca tako pri načrtovanju uporabe teh delov (ingeniring), usposabljanju tehnično-tehnoloških kadrov, kakor tudi zagotavljanju visoko kvalitetnega servisa za te dele, itd. OEM trženje zahteva kot predpogoj določen tehnološki nivo ("tehnološko kulturo") tako s strani uporabnika kot proizvajalca. **Uporabnik** mora upoštevati pakete OEM produktov, ki so mu na voljo le kot elemente, ki jim dodaja "svojo" vrednost v obliki vgraditve ali drugačne kombinacije - smotrnejše, sodobnejše - le - teh. S tem uporabnik dobi nov produkt, ki je sodoben in hkrati zaradi tehnološke osnove potencialno kompatibilen s široko paleto sistemov računalniške strojne opreme. Cilja sta torej najmanj dva:

- * opremljanje (posodabljanje) obstoječe strojne opreme za proizvodnjo - avtomatizacijo in informatizacijo. To je proces, ki se mora intenzivno začeti v slehernem gospodarskem subjektu v Sloveniji in Jugoslaviji.
- * vgrajevanje računalniških elementov v svoje produkte (od npr. bele tehnike v pralne stroje in mikrovalovne pečice do laserskih stružnic in zag ali testnih naprav za avtomobilске motorje - v svetu so vsi ti primeri že praksa).



Tako pride uporabnik z uporabo računalniške, strojne in programske opreme po delih, do bistvene komparativne produkcije pred drugimi, ki ne uporabljajo informacijskih polproizvodov za svojo proizvodnjo ali produkte. Seveda pa predstavlja to velik izziv tudi za proizvajalca, ki mora zagotavljati:

- * standardizacijo tehnoloških funkcij posamičnih elementov
- * promptno dobavo dogovorjenih količin
- * cenovno ustrezno strukturo, da se ne ruši proizvodna cena končnega produkta
- * opozorila in usmeritve na trende tehnološkega razvoja za pravočasno prilagajanje našim domačim in svetovnim tržnim zahtevam

OEM trženje torej pomeni bistven premik naprej v preseganju miselnosti, da mora vsak posameznik ali vsaka DO razvijati in proizvajati posamezne proizvode do poslednjega dela. Takšna medsebojna povezanost grajena na tehnološkem redu in disciplini pa vodi v standardizacijo in tipizacijo rešitev. To je osnova za enostavnejše povezovanje posameznih rešitev, delov, strojev, sistemov v kompleksne sisteme. S tem bomo iz določene kvantitete prestopili prag informacijske kvalitete. Ni samo pomembno, da se prebije na mednarodni trg posamezna DO, temveč tehnološka fronta države.

3. KAKŠNA SO RAZMERJA NA OEM TRGU V EVROPI?

Celoten miniračunalniški trg v Evropi obsega v preteklem letu 5216 mio. US \$. Na tem trgu ima npr. samo DEC 2000 OEM kupcev, ki predstavljajo preko 22% celotnega prihodka. Med njimi so kot OEM kupci tudi giganti kot npr. Siemens in ITT, ki izgrajujeta telekomunikacijski sistem za evropski PTT.

Na sploh lahko v svetu ocenimo, da se je prodaja računalniških modulov zelo povečala. Pred desetletjem so morali inženirji, ki so potrebovali določene module za svoje namene le te zgraditi, kar je vzelo precej časa pa tudi poceni ni bilo. Danes so v svetu na voljo katalogi modulov in drugih sestavnih delov. Z ugrajevanjem več in več računalniške moči v te produkte so proizvajalci izgradili večmilionski (\$) trg za inženirje, ki raje pišejo aplikacijske programe in gradijo specifične module in razvijajo strojno opremo od začetka. Kakšna so razmerja med deleži na tem trgu? V vsem svetu je prisoten opisan trend rasti, ki predstavlja porast od 600 mio. v letu 1986 na 2 mlrd. v 1990. Največji porast bo pri t.i. VME vodilih, ki bodo dosegla kar 460 mio. v 1990. Pomemben delež bo na trgu obdržala družba Intel tako z Multi busom I kot z Multi busom II.



Bistvena bitka se odvija tokrat na obeh nivojih-tako cenovnem kot na kvalitetnem. Tako je bil povprečen rang cene modulov zbit od 2500 v 1982 na komaj 1500 US\$ danes. Najsodobnejši produkti so v cenovnem razredu med 4.000 in 6.000 US\$.

V teh razmerah je pomemben pojav novih kategorij t.i. "Single board" računalnikov. Ti produkti imajo bistveno večjo moč od enostavnega procesiranja in baziranja na kompatibilnost s PC programsko opremo. Opirajo pot k porabniku aplikacij "single board" računalnikov namenjenih za specialne funkcije. Omeniti velja, da se na trgu že pojavljajo prve 80386 - VMEbus kombinacije, naprimer modul SYS 80K/CPU-386, ki ima CPU z 2MB z DRAM-om in 4-imi EPROM z 512KB za uporabnikov firmware. Tudi Iskra Delta bo v kratkem k svojemu naboru OEM modulov dodala modul navedenega ranga-seveda z vsemi prednostmi, ki jih takšna prodaja predstavlja od že omenjenega plačanja z dinarji do iskoriščenja in vsposabljanja za te produkte.

Skupni imenovalac omenjenih podatkov je, da predstavlja OEM pristop - prodor (v Evropskem trgu) v tiste višje oz. specifične trge, ki bi bili sicer bistveno kasneje ali bistveno dražje obdelani s stališča informatike. Dejstvo, ki ga ne kaže zanemarjati je, da so v takih konkurenčnih pogojih na količino od nekaj 1.000 kom. nakupa atraktivni popusti, ki dosega celo do 30%.

4. KAJ TOREJ PONUJA ISKRA DELTA OEM KUPCEM?

Nabor sestavljajo moduli in sklopi iz proizvodnega programa ISKRA DELTE kot na primer različna ohišja, napajalniki, vodila (sistemsko VME, diskovno), spominski moduli (2MB), kontrolerji (za terminale, za diskovne enote, za grafiko, za komunikacije), gibki disk (1MB), winchester diskovne enote (40 MB, 80 MB), tračne enote, AD konverterji, moduli procesnega podsistema DIPS '85, monitorji (barvni, crno/beli), centralne procesne enote različnih izvedb (J11, Intel APX286, Motorola 68010) s pripadajočimi operacijskimi sistemi kot so DELTA/M, RSX-11/M, XENIX, MS-DOS, Uniplus, OS/9, različni prevajalniki C, COBOL, FORTRAN, PASCAL, informacijska orodja, IDA, programski generator FORMATIX, itd. Zanimivo je, da lahko že danes z zamenjavo enega samega hardwareskega modula (in operacijskega sistema) spremenimo en in isti računalnik TRIGLAV iz družine DELTA računalniških sistemov v TRIGLAV, ki je kompatibilen računalnikom drugih proizvajalcev, na primer DEC, IBM, AT&T, itd. Hardwareski moduli so načrtovani v skladu z mednarodnimi standardi, na primer moduli po velikosti ustrezajo velikosti dvojnega Evropa formata, grajeni so za standardno VME vodilo. Mednarodni standardi in podane specifikacije omogočajo uporabniku, da si po potrebi sam zgradi svoj lastni modul in ga vgradi v celoto ali pa obratno, da enega od navedenih modulov vgradi v nek svoj avtomat ali drugo tehnično rešitev. Na voljo so tudi prazni razvojni moduli, hardwareska orodja, merilna in testna oprema za razvoj lastnih modulov.



Nabor modulov seveda ni omejen. 32 bitna arhitektura TRIGLAVA odpira možnosti za uporabo 32 bitnih CPU modulov, zahtevnejših diskovnih kontrolerjev, komunikacij, itd., ki so že v končni fazi razvoja. Iskra Delta nudi seveda tudi že "sestavljene" računalniške sisteme družine TRIGLAV, ki služijo kot razvojni sistemi in kot splošnonamenski računalniki.

TRIGLAV je torej prvi jugoslovanski računalniški sistem, ki ga Iskra Delta trzi v obliki modulov po OEM principu (Original Equipment Manufacturer) in kot celoten sistem v določeni konfiguraciji. Razširitev OEM ponudbe tudi na druge polproizvode in proizvode je samo vprašanje interesa uporabnikov.

V OEM obliki TRIGLAV pomeni bazični repromaterial za vse proizvajalce, ki v kakršnikoli obliki vgrajujejo v svoje proizvode računalniško infrastrukturo. To so na primer proizvajalci numerično krmiljenih strojev (CNC), robotike, avtomatov in opreme za avtomatizacijo procesov, posebnih specializiranih delovnih mest, na primer CAD/CAM, merilne in laboratorijske tehnike, itd. V obliki računalniškega sistema pa pomeni TRIGLAV le enega izmed Delta računalniških sistemov med družinami PARTNER na eni strani in D 800 in D 4850 na drugi strani. Medtem ko predstavlja izvedba Partner/T delovno mesto le za enega uporabnika, pa predstavlja TRIGLAV večuporabniški sistem ali pa enouporabniški sistem z barvno grafiko. Kot je lahko PARTNER TRIGLAVU inteligentni PAKA 3000 terminal, tako je lahko TRIGLAV inteligentna podpostaja za D 800 in D 4850. Operacijski sistem DELTA/M in programska orodja IDA omogočajo popolno kompatibilnost sistema TRIGLAV s sistemi družine D 800.

V zaostrenih pogojih godpodarjenja, a ob velikem pomenu uvajanja sodobnih visokih tehnologij v jugoslovansko gospodarstvo se je Iskra Delta odločila ponuditi nosilcem najrazličnejših proizvodnih programov ekonomsko-tehnološko dostopnejšo obliko uvajanja njenih produktov v kompleksnejše proizvode jugoslovanskih proizvajalcev. Iskra Delta je v zadnjih letih namreč z razvojem in trženjem mikroročunalnikov dosegla v ekonomsko-tehnološkem smislu raven serijske proizvodnje.

Z ekonomskega vidika pomeni prodaja OEM proizvodov Iskre Delte novost za jugoslovansko tržišče, saj daje mnoge primerjalne prednosti, ki jih zainteresirani kupec ne bo prezrl :

1. Vključevanje OEM produktov v lastni tehnološki proces je v proizvodno-tehnološkem smislu zasnovano na dolgotrajnejših medsebojnih poslovnih odnosih. Ti odnosi zagotavljajo OEM kupcu ustrezne proizvodno-tržne pogoje pri planiranju, proizvodnji in trženju lastnih proizvodov.

2. Z ozirom na obseg kadrovskih in materialnih resursov, ki jih Iskra Delta vlaga v razvoj svojih produktov, so kupcu tekoče zagotovljene vse tehnološke izboljšave na določenem sklopu ali podsklopu in s tem tudi dana garancija, da kupčev proizvod tako z vidika kakovosti kot po tehnološki plati ne bo izgubil renome blaga, namenjenega najzahtevnejšim svetovnim trgom.



3. Z ozirom na značaj serijske proizvodnje, kjer ponovljivost serij zagotavlja enako kvaliteto in kompatibilnost, je servisiranje proizvajalčeve opreme omogočeno v daljšem roku. To servisiranje se vrši z zamenjavo in popravilom modulov v centrih Iskre Delte na posameznih trgih.

4. Cena OEM proizvodov se znižuje sorazmerno z obsegom naročil v določenem časovnem obdobju, kar je posebej opredeljeno s količinskim popustom v OEM pogodbi. Ne glede na to ne smemo pozabiti, da je OEM cena za velikostni razred nižja od cene oziroma investicije v lastni razvoj unikatnih modulov.

5. Z nakupom OEM proizvodov Iskre Delte kupec substituirava uvoz za njega nujnih repro-sklopov na visoki stopnji tehnološke obdelave, s čimer se izogne zapletenemu uvoznemu postopku v smislu veljavne zunanje trgovinske zakonodaje. Kontinuiteta dobave zagotavlja optimalno angažiranje finančnih sredstev zaradi krajših dobavnih rokov.

Naša težnja je, da s tako obliko poslovnega sodelovanja prispevamo svoj delež k zmanjševanju razlik v tehnični razvitosti med jugoslovanskim in drugimi razvitimi gospodarstvi. Zavedati se namreč moramo, da so te razlike eden ključnih razlogov zaostajanja nerazvitih nacionalnih ekonomij pri vključevanju v mednarodno delitev dela, saj le-te pomenijo oviro za obsežnejši izvoz izdelkov z večjim deležem jugoslovanskega znanja. To mora konkretno predstavljati v že obstoječem OEM tržišču v Jugoslaviji (kupci tujih firm po OEM principu so npr. Iskra Telematika, Iskra Avtomatika, Institut Jozef Stefan, Rade Končar, RIZ, Beogradska računska industrija, OUR Računari - Ei Niš, Nikola Tesla, UNIS, Energoinvest - RO Iris - da omenimo le nekatere največje) postopni prehod na jugoslovanske dele, sklope in podslope. Skratka, nadomestitev ali dopolnitev tujih z domačimi proizvajalci. Seveda zahteva ta proces svoj čas.

Nikakor ne nazadnje pa to pomeni idealno priložnost za razvoj t.i. družbenih kooperativ, ki lahko na tehnoloških osnovah s potrebnim repromaterialom bistveno prispevajo k hitrejšemu in celovitemu razvoju tako informatike, produktov informatike kot predvsem konkretnih aplikacij.

OEM trg v Jugoslaviji je pogoj za izpeljavo informacijske družbe predvsem glede na omejene razmere in stanje. Hkrati načela OEM obetajo tehnološko zblíževanje in standardizacijo, ki nam je v razmerah, ko moramo vedno več in kvalitetnejše nastopati v tujini, nujno potrebno. Ne nazadnje predstavlja OEM pristop izziv za vrsto sposobnih posameznikov in organizacij, da si ob preizkušeni vrhunski tehnologiji vtorejo bližnjico v 21. stoletje.



**VEĆPROCESORSKI RACUNALNISKI
SISTEMI IDC**

I. VEĆPROCESORSKI SISTEM GEMINI

Avtorji referata: Milovan Jefić, dipl.ing. ISKRA DELTA
Marjan Murovec, dipl.org. ISKRA DELTA

II. TRIGLAV VME CPE J11

Avtor referata: Dušan Zalar, dipl.ing. ISKRA DELTA



Ljubljana, maj 1987



I. VEĆPROCESORSKI SISTEM GEMINI

1.0 UVOD

Najboljša pot za dobro odločitev, za ali proti večprocesorskemu računalniškemu sistemu, je da se oborožimo z dobrimi informacijami. To besedilo je prva stopnica na tej poti.

1.1 DOBRODOŠLI V SVET VEĆPROCESORJEV

Najnovejši član miniračunalniške družine Delta je večprocesorski računalniški sistem. To je prvi računalniški sistem našega razvojnega projekta večprocesorskih sistemov pod nazivom GEMINI zato ga bomo v nadaljevanju imenovali s tem imenom. Mnogi naši uporabniki načrtujejo prehod, ali pa že celo uporabljajo večprocesorske sisteme firme DEC, ki so nastali kot rezultat:

- * poglobljenega študija in strokovnih analiz svetovnih gibanj na tem področju;
- * aktualnosti implementacije večprocesorskih sistemov;
- * analize obstoječih večprocesorskih sistemov.

V predstavitvi, ki sledi, bomo poizkušali pojasniti čimveč dejstev, ki jih je, o lastnem računalniškem okolju, potrebno spoznati, da bi se odločali najbolj pravilno o nakupu večprocesorskega sistema.

1.2 RAZLOGI ZA GRADNJO VEĆPROCESORSKEGA SISTEMA

Obstaja več dobrih razlogov za izgradnjo in uporabo večprocesorskega sistema:

- * povečanje zanesljivosti in razpoložljivosti sistema
- * povečanje performans sistema
- * izkoristek vseh prednosti, ki jih ponuja računalniška tehnologija (elektronika in programske komponente)



| Sistemske lastnosti: | Večprocesor | VAX Cluster | Rač.mreže |
|----------------------|--------------------------|--------------------|-----------------------|
| Okolje | tesno povezan | ohlapno povezan | ohlapno povezan |
| CPE napake | skupne(GEMINI posamezne) | posamezne | posamezn |
| CPE kabinet | eden ali deljen | več rač. prostorov | več različnih lokacij |
| Zanesljivost | enojna(GEMINI večkratna) | enojna | večkratna |

Kot vidimo ima GEMINI mnoge prednosti, ki so značilne tudi za ostale konfiguracije, vendar brez nekaterih njihovih neugodnih lastnosti.

1.3 SMER ISKRA DELTE

Ze instalirana baza računalniških sistemov Delta omogoča, z vpeljavo večprocesorskega sistema GEMINI, zmanjšanje obremenitve (Work Load) podatkovnih centrov in ne povzroča podvajanj podatkov. S tem se povečata integriteta in razpoložljivost podatkovnih baz in izboljša vzdrževanje samih podatkovnih baz. To je prvi korak k decentralizirani uporabi računalnika s centraliziranim podatkovnim shranjevanjem in upravljanjem.

1.4 PRENOS APLIKATIVNE OPREME NA SISTEM GEMINI

Pri prenašanju aplikativne opreme na sistem GEMINI niso potrebne nikakršne spremembe aplikacij, ki so izdelane po programskih standardih Iskre Delte.

1.5 UVAJANJE SISTEMA GEMINI

Pri uvedbi sistema GEMINI se v največji možni meri uporabi in upošteva uporabnikova obstoječa strojna in programska oprema, s čimer se zagotavlja in ohranja upravičenost uporabnikove predhodne naložbe. Zagotovijo pa se naslednje prednosti:

a) Povečana razpoložljivost (Availability):

- * podatkov (dvostranski dostop do diskovnih enot, avtomatsko beleženje in obnavljanje podatkov)
- * programov (ker se nahajajo na skupnih diskovnih enotah)
- * terminalov (terminalski podsistem, mreža)
- * ostalih perifernih enot (npr. tiskalniki, mreža)



b) Povečanje moči procesorja:

- * Če potrebujemo zvečano moč procesorja, le-to enostavno in hitro povečamo z dogradnjo obstoječega sistema v sistem GEMINI, da pa za to ni potrebno obsežno dodajanje V/I enot (diski, tiskalniki,...)

c) Izmenjevanje podatkov poteka nadzorovano:

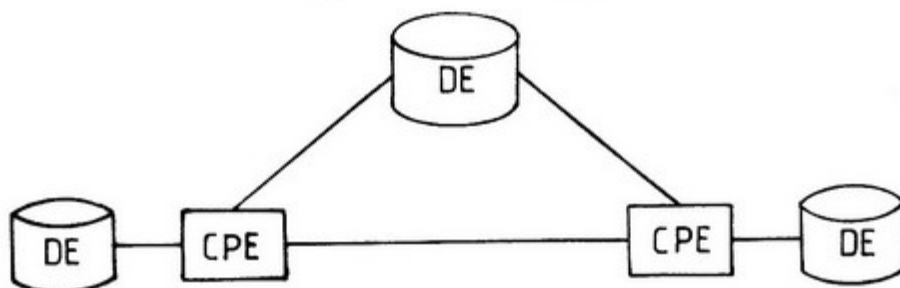
- * Podatki se znotraj sistema GEMINI sinhrono izmenjujejo, brez operaterjevega poseganja.

2.0 STROJNA OPREMA SISTEMA GEMINI

V tem trenutku pomeni prehod iz enoprocorskega okolja v večprocesorsko okolje velik izziv. Zasnova sistema GEMINI omogoča ta prehod na več načinov. Eden je, takojšnja celovita uvedba sistema GEMINI, drugi načini, odvisno od obstoječe opreme, pa praktično pomenijo dogradnjo in s tem razširitev obstoječega sistema. Tekočega dela ni potrebno v nobenem primeru prekinjati.

2.1 RAZNOVRSTNOST VSPOSTAVITVE SISTEMA GEMINI

Koncept GEMINI je moč vzpostaviti z različno zmogljivostjo, glede na uporabljene računalniške sisteme (D800, D48xx, D80xx, TRIGLAV,...). Vsem različnim postavitvam je skupna samo zahteva, da mora imeti vsak posamezen sistem eno lokalno diskovno enoto in najmanj eno diskovno enoto z dvostranskim dostopom (Dual-ported), ki si jo sistemi delijo.



Najzmogljivejši sistem GEMINI tvorita dva procesorja D80xx ter kombinacija diskovnih enot 300 ali 600 MB. Začetno nalaganje sistema se izvrši za posamezen sistem ločeno. Vsak sistem tudi ločeno, samostojno izvaja svoja dela. Sistemi si medsebojno delijo zunanje pomnilniške kapacitete.



2.2 APLIKACIJE NA VEĆPROCESORSKEM SISTEMU GEMINI

Določeni aplikativni opremi ponuja sistem GEMINI izboljšave, drugi pa ne. Izboljšave so pri:

- * transakcijskih sistemih na skupnih bazah podatkov (število uporabnikov se 2-krat poveča)
- * 100-odstotno razpoložljivih sistemih (banke, proizvodnja, ipd.)
- * interaktivnem delu

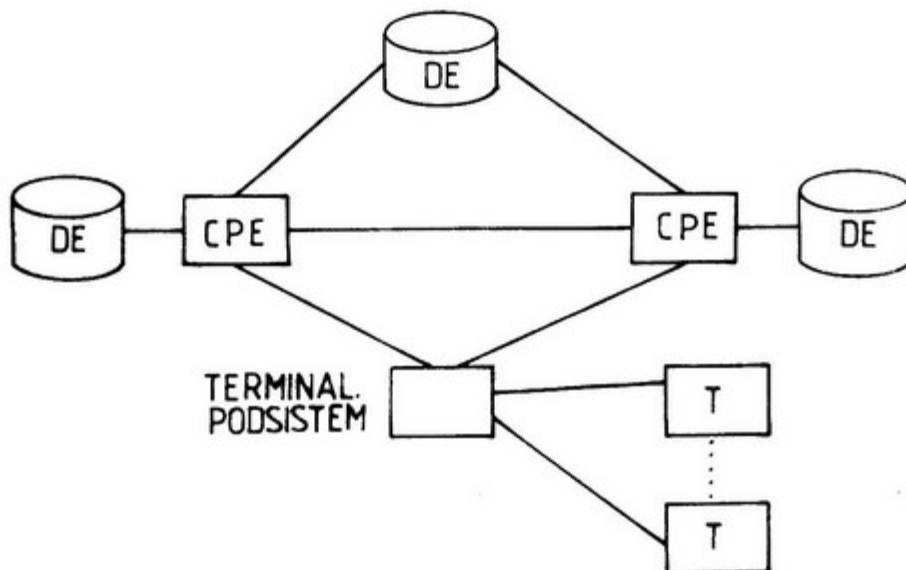
Izboljšav ni pri:

- * programu, ki zahteva velike CPE resurse

2.3 URAVNOVEŠENA OBREMENITEV SISTEMA

Uravnotešena obremenitev procesorjev in diskovnih enot je izredno pomembna, saj se s tem dosežejo optimalnejše performanse. Pri uravnoveševanju sistema je potrebno posvetiti posebno pozornost pravilni izbiri procesorja na katerem se bo izvajala vsaka posamezna aplikacija, ter načinu in številu terminalov, ki bodo priključeni na posamezen procesor.

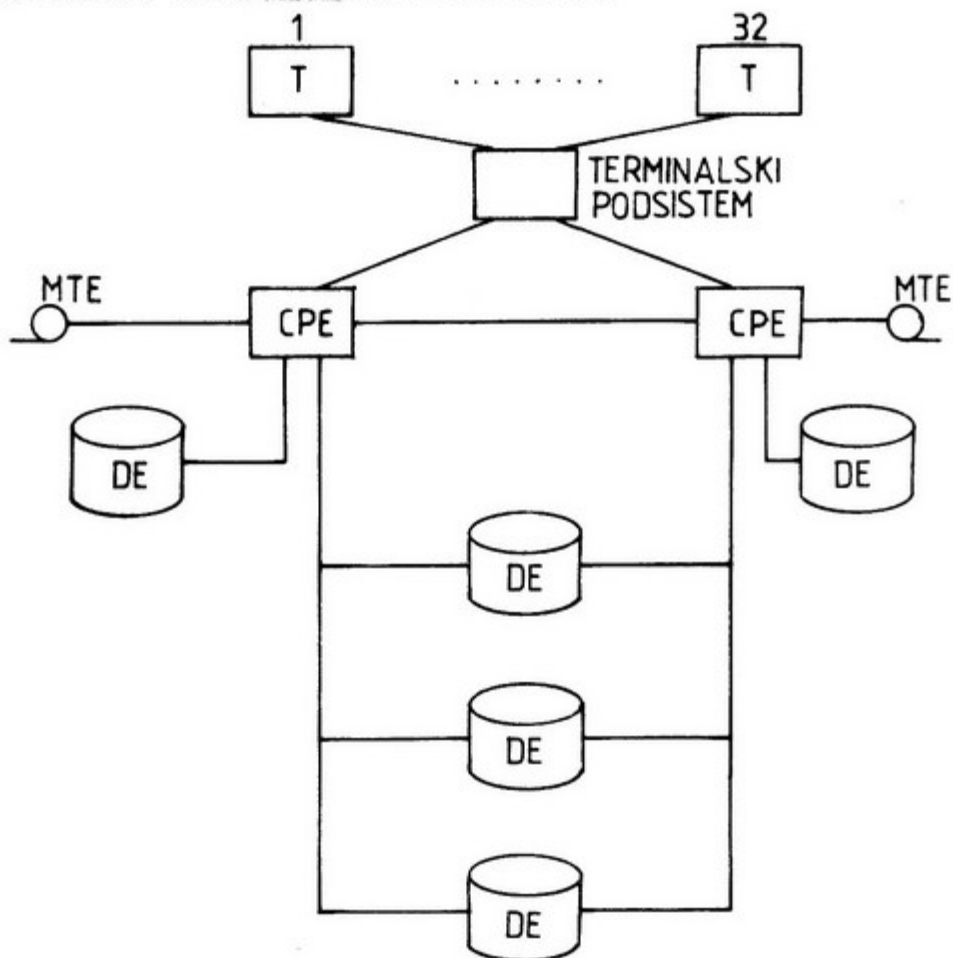
2.4 TERMINALSKI PODSISTEM



2.5 DISKOVNE ENOTE Z DVOJNIM PRISTOPOM

Sistem GEMINI podpira delo z diskovnimi enotami z dvojnimi pristopom, kar zagotavlja izjemno zanesljivost pristopanja k podatkom. Pričakovana izboljšava na tem nivoju je omogočiti dinamičen pristop na diskovne enote z dvojnimi pristopom.

2.6 ZASNOVA VECPROCESORSKEGA SISTEMA GEMINI



3.0 PROGRAMSKA OPREMA SISTEMA GEMINI

Programska oprema je sestavljena iz več delov, ki nudijo sistemu GEMINI popolno podporo za najoptimalnejše delo:

- * Operacijski sistem (DELTA/V V2.0)
- * Komunikacijska programska oprema
- * Programska oprema za podpiranje terminalskega podsistema
- * IDA informacijska orodja

3.1 OPERACIJSKI SISTEM DELTA/V V2.0

Nova verzija operacijskega sistema DELTA/V prinaša več pomembnih izboljšav in je kompatibilna z operacijskimi sistemi na drugih računalnikih serije Delta 48XX. Ta lastnost zagotavlja prenosljivost aplikacij med temi računalniškimi sistemi.

3.2 KOMUNIKACIJSKA PROGRAMSKA OPREMA

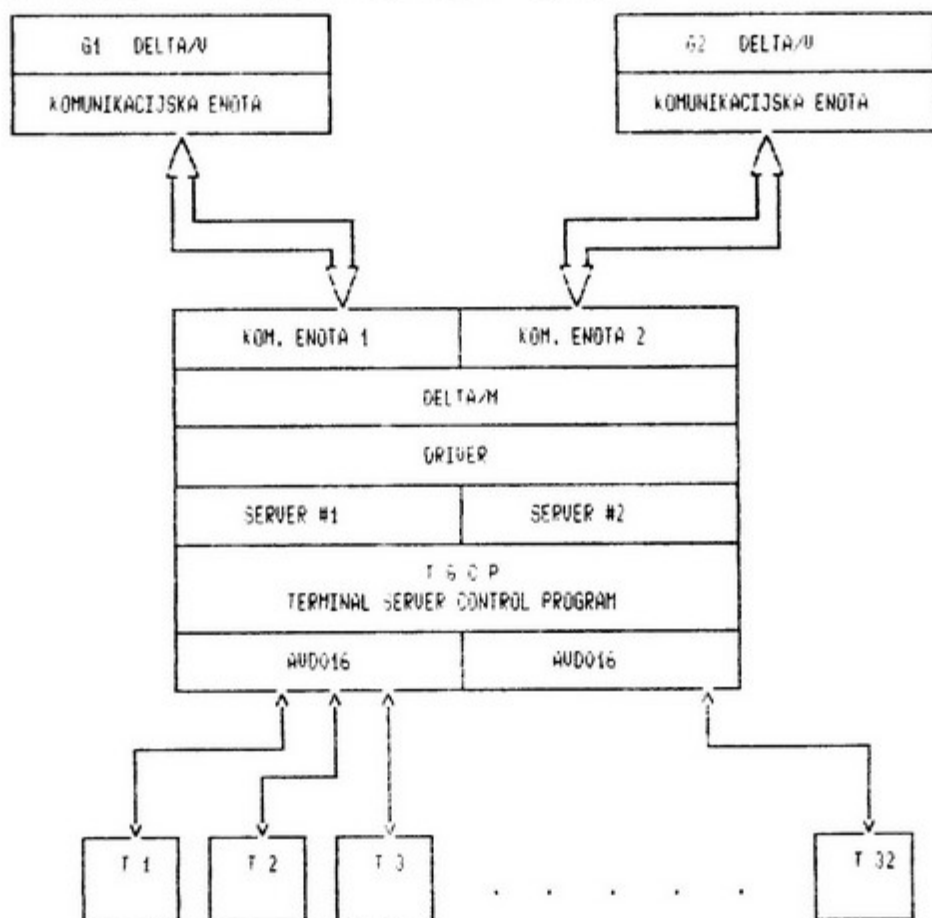
Lastnosti komunikacijske programske opreme so naslednje:

- a) Zmogljiva komunikacijska oprema (strojna in programska) povezuje sistem GEMINI v funkcionalno celoto
- b) Komunikacijske funkcije Deltinih proizvodov (IDA Baza, IDA Leksikon) so integrirane, kar daje uporabniku vtis enovitega sistema.
- c) Funkcionalnosti mreže (dostop do Remote datotek, File Transfer, virtualni terminali) omogočajo uporabniku dostop do resursov, ki so v resnici na drugem CPE, na nivoju operaterskih (DCL) ukazov in programov.



3.3 TERMINALSKI PODSISTEM

Terminalski podsistem je realiziran na enem od 16-bitnih DELTA računalniškem sistemu, ki je sestavni del sistema GEMINI. Ta podsistem omogoča uporabnikom interaktivno delo na poljubnem procesorju v sistemu GEMINI. To je zlasti pomembno, kadar je kateri od procesorjev v izpadu in je potrebno delo nadaljevati na drugem procesorju. Poleg tega je ta podsistem načrtovan tako, da razbremeni glavne procesorske resurse. Servisni sistem upravlja s prekinitvami (Interrupti), ki ih sprožajo vhodno-izhodne operacije na terminalih.



Vse elektronske in programske komponente, razen programa TSC, so za uporabnika nevidne (transparentne). S pomočjo programa TSC upravlja uporabnik s svojim terminalom:

- * odloča se, na katerem procesorju bo delal (funkcija CONNECT)
- * lahko se izključi iz procesorja (DISCONNECT)

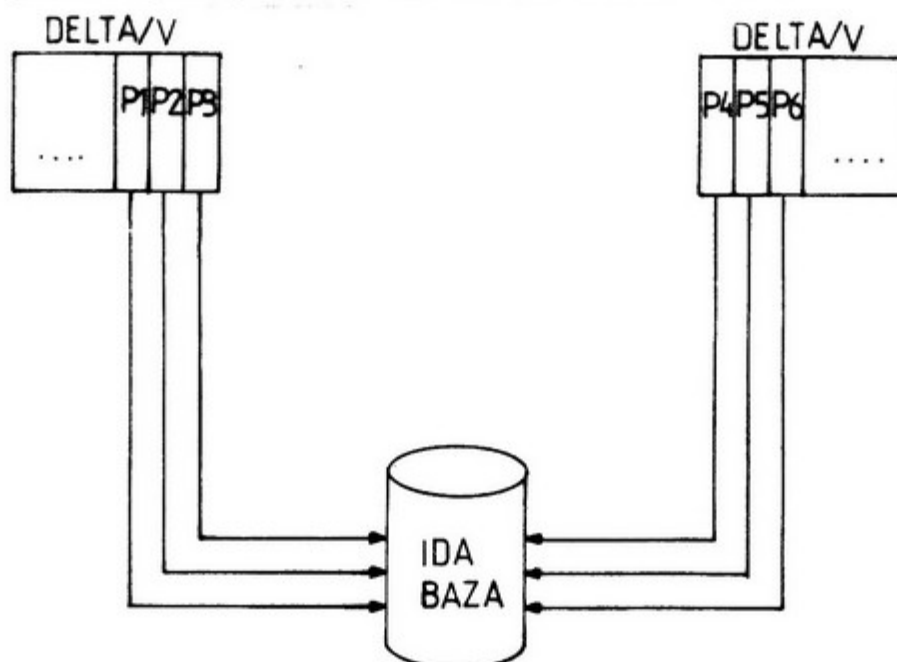


Vodja sistema pa lahko opazuje delovanje terminalskega podsistema (MONITORING). Poleg standardnega programskega paketa za servisiranje terminalov lahko uporabnik, ki to zna, izdelava svoje serverje.

Nadaljnji razvoj terminalskega podsistema pa prinaša možnost upravljanja z obremenitvijo procesorja (Load Balance) in posebne serverje (npr. IDA Ekran), ki bodo v mnogo večji meri razbremenili glavne procesorske resurse.

3.4 GLOBALNA BAZA PODATKOV

Proizvod Distribuirana IDA Baza omogoča vsem uporabnikom, ki delajo na poljubnem procesorju, da sočasno dostopajo k podatkom, ki so v globalni bazi podatkov. Seveda je realizirana učinkovita meduporabniška zaščita podatkovnih resursov ter avtomatsko beleženje sprememb (Transaction Logging) in obnova podatkov (Recovery). Ker se nahajajo podatki na diskovnih enotah z dvojnimi pristopom, se lahko programska produkcija nadaljuje tudi v primeru izpada procesorja ali diskovne kontrolne enote.



Pri tem proizvodu velja pripomniti, da je za obstoječe aplikacije popolnoma transparenten. Aplikacije se lahko brez ponovnega prevajanja in povezovanja prenesejo na sistem GEMINI. To pomeni, da je sistem GEMINI priporočljiv zlasti tam, kjer bi radi za že obstoječe aplikacije povečali število uporabnikov (terminalov).



4.0 ZAKLJUČEK

Na razvojnem sistemu GEMINI so bile opravljene meritve zmogljivosti sistema.

4.1. IDA BAZA

| okolje St.procesov | LOKALNO | REMOTE | L + R |
|-----------------------|---------|--------|------------------|
| 1.proces | 92/2 | 38/2 | X |
| 2.procesa | 100/2 | 50/2 | |
| 3.procesi | 109/2 | 51/2 | 1L, 2R 70/2 * |

Slika 6. Zmogljivosti IDA BAZE + D BAZE

*Ta rezultat je pričakovan. Nekaj (CPE) resursev na lokalni strani se porabi za NETWORK.

4.2. TESTIRANJE MAKSIMALNEGA STEVILA TERMINALOV

Opis

Program izda zahtevo za poljubno branje s terminala. Po vnosu (CR) izda funkcijo INSG in doda zapis v bazo podatkov. Program je bil aktiviran na 7 terminalih paralelno, v avtorepeat režimu, kar odgovarja številu 50 - 60 pravih terminalov. Število zahtev na disk je bilo 27 na sekundo (blizu kritične vrednosti). Ostalo stanje normalno.



4.3. LOAD TEST

LOAD test je potekal na osnovi paketa PMPS (Iskra CAOP), DBPRINT utility programa in DCL ukazov, ki so močno obremenili sistemske resurse na lokalni strani (tam, kjer je bil sistem PMPS DB). Nekatere ugotovitve:

1. S povečanjem zmogljivosti CPE se izboljša zmogljivost celega sistema.
2. Programska oprema (OS, IDA, DELTANET) je funkcioniral brez napake, čeprav je bil sistem namerno obremenjen.
3. PIOUS in sploh ves HW je deloval brez napake.
4. CPE na REMOTE strani je ostal razpoložljiv (50 - 65 %), ostali resursi neizrabljeni.
5. Distribuirana baza lahko doživi nekatere spremembe (podsheme na remote strani, izboljšava komponente DBNETSERV), ki izboljšajo performanse sistema v celoti.

4.4. OSTALE MERITVE IN TESTIRANJA

Poleg opisanega testiranja je bilo izvedenih še nekaj poiskusov, ki so bili namenjeni preverjanju uporabljenih rešitev.

4.4.1. IDA BAZA-INTENZIVNOST 50 % IN CPU INTENZIVNI PROGRAMI

Test je bil namenjen ugotavljanju ustreznosti SW komponent z vidika medsebojne vsklajenosti (OS, SIC DRIVER, BAZA, NETWORK, IDA vmesnik uporabniški programi). Test je pokazal zadovoljivo vsklajenost (funkcionalnost, parametri, kvote itd.).

4.4.2. FUNKCIONALNO TESTIRANJE

Preizkušeni so bili programi iz dveh paketov: Dinarsko poslovanje Iskre Delte (po IDC standardih) in paket PMPS Iskre Zorin, ki je bil napisan na ULTRA-TOTAL in TTW vmesniku. Pri tem smo ugotovili 100-odstotno funkcionalnost.



4.5 OCENA REZULTATOV

Testi in meritve so dokazali pravilnost pričakovanih rezultatov:

1. Sistem v obstoječem stanju zmore vsaj 1.6 večji load, kot standardni sistem 4850 na isti aplikaciji. (V končni verziji lahko pričakujem 2-kratni load)
2. Sistem zmore 2-krat večji load na različnih aplikacijah (režim dveh neodvisnih sistemov)
3. Funkcionalnost HW in SW je zanesljiva in brez napak
4. Izbrana verzija OS izboljšuje funkcionalnost sistema, neugodno pa vpliva na porabo CPE
5. SW komponente premalo uporabljajo pomnilnik, zato prihaja do debalansa:
 - * CPE preobremenjen
 - * DISKI O.K.
 - * MEMORIJA - izhodiščnost 25 % moznosti
6. Glavna prednost GEMINI-ja je njegova zanesljivost. Le malo sistemov pa vsaj tako dobro uporablja podvojene komponente.

5. LITERATURA

- * Cay Weitzman: Distributed Micro/Minicomputer Systems, Prentice-Hall 1980
- * Bell, Mudge, McNamara: Computer Engineering, Dec. 1978
- * Harvey M. Deitel: An Introduction to Operating Systems, Addison-Wesley, 1984
- * Shaiy Pilpel: A Balanced Scalable Parallel Processor, VLSI Systems Designe, March 1987
- * Siegel, Schederski: Large-Scale Parallel Processing Systems, Microprocessors and Microsystems, Vol.11 No.1 1987
- * IDC: Razvojna Dokumentacija IDC za Produkte IDA Orodja, DELTAnet, D48xx, D800, DELTA/V V 2.0, DELTA/M V 2.0



II. TRIGLAV VME CPE J11

Prispevek opisuje arhitekturo procesorskega modula VME CPE J11 in njegove možnosti za delovanje v večprocesorskem sistemu osnovane s tem modulom na skupnem VME vodilu.

1. UVOD

Triglav VME CPE J11 je lahko večprocesorski sistem. Baziran je na osnovi procesorskega modula VME CPE J11. Le ta je zasnovan s 16-bitnim mikroprocesorjem DCJ11 firme Digital Equipment Corporation. Modul ima priključek na VME vodilo, kar omogoča modularnost sistema in uporabo raznovrstnih modulov na tem vodilu. Paralelno delovanje več modulov na skupnem VME vodilu omogoča lokalni pomnilnik.

2. ARHITEKTURA MODULA VME CPE J11

Procesorski modul je sicer sestavljen iz procesorja, enote za krmiljenje pomnilnika, lokalnega pomnilnika velikosti 512kB z detekcijo paritetne napake, priključka na VME vodilo, serijskega kanala, paralelnega kanala, ure realnega časa in ROM-om z inicialnim vlagalnikom. Serijski kanal služi za povezavo na konzolni terminal, paralelni pa za povezavo na tiskalnik.

3. NASLOVNI PROSTOR MODULA

CPE J11 lahko naslovi spodnje 4MB-8kB VME vodila pri standardnem 24-bitnem naslavljanju pomnilniškega prostora in zgornjih 8kB pri skrajšanem 16-bitnem naslavljanju vhodno izhodnega prostora. Lokalni pomnilnik ima lahko naslov od 0 do 3.5MB v presledkih po 0.5MB. Dosegljiv je s strani CPE in VME vodila. Vse lokalne enote imajo fiksen naslov in so dosegljive samo s strani CPE.

4. PARALELNO DELOVANJE VEČ MODULOV

Različni naslovi lokalnega pomnilnika omogočajo koeksistenco več pomnilnikov pri neposrednem dostopu preko VME vodila do le teh. Nedosegljivost lokalnih vhodno izhodnih enot preko VME vodila pa omogoča koeksistenco le teh na enakih naslovih. CPE zaseda VME vodilo samo takrat, ko posega v globalni naslovni prostor. To omogoča paralelno delovanje procesorjev na lokalnih pomnilnikih.

Omejitev naslovnega prostora pomeni tudi omejitev števila CPE (največ osem).



5. ARHITEKTURA VEĆPROCESORSKEGA SISTEMA

Tako kot enoprocesorski lahko vsebuje tudi večprocesorski sistem pomnilnik, krmilnike masovnih pomnilkov, krmilnike za terminale, tiskalnike in mreže. Vse te enote pa so skupne (globalne) za vse procesorske module.

6. PRENASANJE PODATKOV PREKO POMNILNIKOV

Vse vhodno izhodne enote imajo možnost, da z neposrednim dostopom posežejo v katerikoli lokalni in globalni pomnilnik. Enako velja tudi za vsako CPE. Možno je tudi neposredno prenašati podatke enega dela pomnilnika v drug del pomnilnika. Vsak poseg na VME vodilo ustavi izvajanje vseh CPE.

7. PREPUSTNOST VEĆPROCESORSKEGA SISTEMA

Absolutna prepustnost je definirana kot število povprečnih opravil, ki jih sistem opravi v časovni enoti. Relativna prepustnost večprocesorskega sistema proti enoprocesorskemu je razmerje med obema absolutnima razmerjima. Pri predpostavki, da je procesor vedno aktiven in da ni istočasnih zahtev za dostop do vodila, potem je relativna prepustnost enaka:

$$p = n/(1+n(1-l))$$

kjer pomeni:

l - povprečno zasedenost lokalnega vodila

n - število CPE

p - relativno prepustnost večprocesorskega sistema

Če procesor ni vedno aktiven potem se propustnost poveča, istočasne zahteve po vodilu pa nekoliko zmanjšajo propustnost.

8. SINHRONIZACIJA V VEĆPROCESORSKEM SISTEMU

Procesi so med seboj lahko sinhronizirani s semaforji in sinhronizacijskimi ukazi. CPE J11 ima dva sinhronizacijska ukaza in sicer TEST-AND-SET, ki postavi semafor in pove prejšnje stanje le-tega, ter WRITE-INTERLOCK, ki služi za vrinjevanje novih členov v vrsto. V skrajno kritičnih primerih lahko uporabimo tudi programsko zaklepanje VME vodila. Drug način sinhronizacije je preko prekinitev. Tako so običajno sinhronizirani zunanji dogodki s procesom. V večprocesorskem sistemu pa se tako sinhronizirajo tudi procesorji med seboj. Sinhronizacija procesorjev preko prekinitev omeji število procesorjev na največ sedem, ker ma VME vodilo le 7 prekinitvenih nivojev, tako da ima vsak procesor svoj prekinitveni nivo. Na VME CPE J11 modulu ni zahtevalnika prekinitev, ta pa je nujno potreben za medsebojno sinhronizacijo procesorjev s prekinitvami. Izveden mora biti na nekem drugem VME modulu.



9. NIVO PARALELIZMA

Koncepcija lokalnega pomnilnika in sinhronizacija ali preko semaforjev ali s prekinitvami omogoča paralelizem na nivoju procesa (programa) in uporabniških paketov. Tak nivo paralelizma povečuje samo prepustnost sistema, nikakor pa ne hitrosti izvajanja posameznega procesa. Mehanizem za komunikacijo med procesi je funkcionalno enak kot pri enoprocесorskem sistemu: skupni pomnilnik, semaforji in prekinitve.



RAZVOJ INFORMACIJSKIH ORODIJ IDA

Avtorji referata: Marjan Murovec, dipl. org., ISKRA DELTA
Bojan Peček, dipl. org., ISKRA DELTA
Miran Marčič, dipl. mat., ISKRA DELTA
mag. Ranko Smokvina, dipl.oec., JADROAGENT



Ljubljana, maj 1987



1. UVOD

Programska orodja za pomoč uporabnikom pri razvoju in vzdrževanju informacijskega sistema "IDA" so grajena po ISKRA DELTA arhitekturi informacijskih orodij.

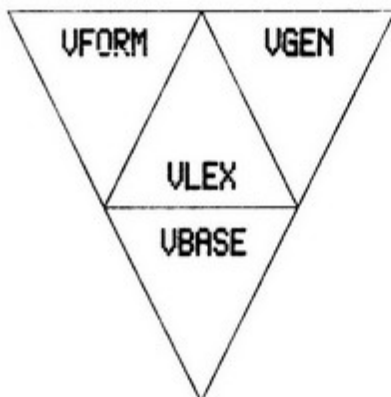
Produkt sestavljajo štiri pod sistemi: IDA-LEKSIKON, IDA-BAZA, IDA-EKRAN in IDA-COGEN. Vsi štiri pod sistemi temeljijo na skupni zasnovi in medsebojni povezanosti v celoto preko IDA-LEKSIKONA.

2. TRENUTNI RAZVOJ NA POSAMEZNIH PODSISTEMIH

2.1 ZADNJA VERZIJA IN NOVI RAZVOJ

IDA informacijska orodja so sedaj produkt, ki je instaliran na več kot 200 računalnikih. Kot vsak sistem se tudi IDA razvija. Pri tem je bilo načelo prenosljivosti s predhodno verzijo vedno temeljni moto novih verzij. Zadnja nosi oznako 1.5 in je že na voljo na računalnikih z operacijskim sistemom DELTA/V V2.0.

Spekter operacijskih sistemov, na katerih se razvijajo IDA orodja se je razširil iz poznanih DELTA/M in DELTA/V tudi na UNIX. V testiranju so prvi produkti: IDA-BAZA, IDA-EKRAN in IDA-LEKSIKON.



2.2 IDA-LEKSIKON

IDA-LEKSIKON je programski proizvod, za zbiranje in shranjevanje podatkov, ki jih potrebujejo ostala IDA orodja. Pri tem beleži tudi povezanost, oziroma medsebojne odnose med podatki.

Vendar IDA-LEKSIKON ni zasnovan samo kot centralni podatkovni slovar IDA orodij, ampak omogoča tudi končnemu uporabniku definicijo in shranjevanje lastnih tipov podatkov in njihovih povezav. Prvo so entitete in drugo relacije. Ker se obseg, predvsem pa tip entitet razlikuje med posameznimi računalniškimi centri, bodisi zaradi razlik v dejavnosti, kot tudi zaradi



različnih metodologij dela, je sirjenje seznama entitet in povezav prepuščeno uporabniku. S tem je zagotovljena uporabnost produkta v vsaki sredini.

IDA-LEKSIKON s svojo prilagodljivostjo omogoča bistveno večjo uporabnost. Od večine podatkovnih slovarjev ga loči lastnost, da omogoča dostop do shranjenih podatkov v poljubnem aplikacijskem programu. Enostavna uporaba funkcij za pregledovanje, in spreminjanje podatkov mu zagotavlja stalno ažurnost zapisov. Uporaba programa za poizvedovanje in ažuriranje pa omogoča njegovo uporabo v preprostih aplikacijah, kjer se zahteva samo zapisovanje določenih pojmov in njihovih medsebojnih povezav.

Zaradi svoje odprtosti in enostavnosti uporabe postaja nepogrešljivo orodje pri sistemski analizi, modeliranju podatkov, standardizaciji, dokumentaciji, programiranju, kontroli računalniškega sistema itd. Namenjen je uporabnikom z različnimi interesi in znanji. Zato vsebuje mehanizme, ki skrbijo za formalno pravilnost vsebine vnesenih podatkov. Slednje ga bistveno loči od klasičnih baz podatkov, ki skrbijo vedno samo za pravilno povezavo med zapisi, kontrola vsebine pa je prepuščena uporabniku.

IDA-LEKSIKON je večuporabniški sistem. Jedro sistema sestavlja server, s katerim lahko komunicirajo ostala IDA-ORODJA, posebno izdelani leksikonski menijski program, ali poljuben uporabniški aplikativni program. Komunikacija poteka s pomočjo leksikonskih funkcij, ki so zelo podobne funkcijam za delo z IDA-BAZO.

Nova verzija ima oznako V1.2, deluje na operacijskem sistemu DELTA/V 2.0 in je vključena v distribucijo IDA V1.5. Zagotavlja simultano uporabo do šest uporabnikov. Izdelan je program za razširitve internih podatkovnih struktur leksikonove baze. Izboljšani so izpisi na leksikonovem menijskem programu. Za naslednjo verzijo so predvidene spremembe, ki bodo omogočile interaktivno delo večjemu številu uporabnikov, ter programi za izpis pojavkov. V razvoju je tudi produkt, ki bo deloval na DELTA/M operacijskem sistemu.

2.3 IDA-BAZA

IDA-BAZA je sistem za upravljanje podatkov. Zagotavlja integriteto, varnost in neodvisnost podatkov.

Zasnova jezika za opis baze podatkov je tronivojska (podshema ali uporabniški pogled, logični in fizični pogled na podatke). V verziji IDA-BAZA V1.5 je jezik za opis baze podatkov doživel večje spremembe. Stare ".SAV" datoteke, kjer so bili opisi shranjeni po vrstnem redu vnosa, so zamenjali ".DDC" opisi. Slednji opisujejo bazo podatkov s pomočjo opisa ključnih pojmov (subschemata name, record name, data-item name, set name, itd). Zagotovljena je prenosljivost med starimi ".SAV" in novimi ".DDC" opisi.



Na operacijskem sistemu DELTA/V, deluje jezik za opis baze podatkov tudi na IDA-LEKSIKONu.

IDA-BAZA je doživela na DELTA/M sistemu znatne dopolnitve. Nova verzija omogoča beleženje transakcij enako kot na DELTA/V operacijskem sistemu. To omogoča prenosljivost aplikativnih programov kar je se posebno pomembno za družino mikroracionalnikov TRIGLAV v funkciji delovnih postaj v sklopu ISKRA-DELTA arhitekture. Sedaj omogoča sočasno delovanje do sedem baz na enem sistemu. Za kontrolo delovanja jedra baze, je izdelan monitorski program. Slednji je na majhnih sistemih zelo uporaben za optimiranje delovanja baze podatkov.

Izboljšana je bila tudi funkcionalnost IDA-BAZE v kompatibilnem načinu dela - TOTLIKE režim.

Na DELTA/V sistemu je bil izdelan program za kontrolo fizične pravilnosti baze podatkov - DBVERIFY. Hkrati omogoča tudi popravljanje najdenih napak. Enak program bo na razpolago tudi na DELTA/M arhitekturi v naslednji verziji.

Zaradi povečanja performans pri razširitvah zbirk nadrejenih zapisov, je bil razvit postopek za razširitev zbirk nadrejenih zapisov brez reformatiranja podrejenih zbirk.

Dobavljiva je tudi programska oprema za doseg globalne baze. To pomeni, da program deluje z bazo podatkov na drugem računalniku. S tem je narejen prvi korak pri razvoju distribuirane baze podatkov in pri gradnji integralnih distribuiranih informacijskih sistemov. S tem so mišljeni sistemi, ki so integrirani po funkciji in distribuirani po tehnologiji.

2.4 IDA-EKRAN

IDA-EKRAN je programsko orodje namenjeno prvenstveno programerjem, da jih razbremeni programiranja terminala, ter raznih kontrol na numeričnost, datum itd.

Paket sestavljata dva modula: ekranski editor CED za kreiranje oblike izgleda ekranske slike in programski vmesnik CRT za operacije na ekranih in ekranskih poljih. Z oddvojenostjo kode, ki skrbi za pravilno komuniciranje z ekranom, kot tudi oddvojenostjo tekstov na ekranu zagotavlja neodvisnost programa od tipa terminala (VT100 ali VT52), jezikovno neodvisnost in tudi programsko neodvisnost od terminalske slike.



Nova verzija IDA-EKRANA V1.4 na DELTA/V operacijskem sistemu ne deluje več s slovarjem podatkov v uporabnikovi bazi. Opis zaslonских slik se hrani na posebni ".CTX" datoteki, po želji pa tudi v IDA-LEKSIKONU. Novi CED je napisan v "native" jeziku za razliko od predhodnega, ki je deloval v "compatibility" načinu. Spremenjena je C-MESS funkcija, dodani pa sta GETU za branje z avtomatičnim pretvarjanje malih črk v velike, ter GETE za branje s popravljanjem vsebine (editiranje). Jezik tekstov s katerim naj dela program se definira s CRT\$LANGUAGE simbolom.

Na novi distribuciji je na voljo cobolski program, ki prikazuje uporabnost IDA-EKRANA. Dostopen je tudi izvorni program, kjer si lahko vsak ogleda način delovanja.

2.5 IDA-COGEN in IDA-PRINTGEN

IDA-COGEN in IDA-PRINTGEN sta orodji, namenjeni hitri in predvsem standardizirani uporabi ostalih IDA orodij.

IDA-PRINTGEN je namenjen predvsem končnim uporabnikom, ki lahko brez posebnega predznanja hitro sestavijo izpisne liste. Za osnovo uporablja podatke shranjene v IDA-BAZI, ali sekvenčnih listah. Omogoča selekcijo zapisov, sortiranje, postavljanje kontrolnega polja za sumarne izračune grupe, vnos formul za izračun itd.

IDA-COGEN pa je bolj namenjen programerjem, katere razbremeni dolgotrajnih in enoličnih opravil kot so opisovanje podatkovnih struktur, standardnih klicov baze, ekrana itd. S tem povečuje produktivnost, zagotavlja boljšo dokumentiranost, standardiziranost in zmanjšuje število napak.

Zadnja verzija omogoča kreiranje programa, ki lahko uporablja IDA-EKRAN tudi če ne dela z IDA-BAZO. Omogoča tudi delo s sekvenčnimi datotekami.

IDA-COGEN bo v novi verziji omogočil tudi vpisovanje v datoteko tipa "clan", delo z več (tipi) zapisov v enem programu (logični pogled), oblikovanje poročil in dokumentiranje.



3. RAZVOJNI CILJI PROGRAMSKIH ORODIJ ISKRA DELTA

Novi trendi informacijske tehnologije vplivajo tudi na ISKRA DELTA arhitekturo. Sistemi za upravljanje baze podatkov, ki temeljijo na strukturiranem opisu baze podatkov in so dosegljivi samo programski opremi, ki deluje na isti lokaciji doživljajo svoj zenit. Bodoči proizvodi, ki jih bodo nadomestili, bodo temeljili na relacijskem modelu baze podatkov in večprocesorskih sistemih. Nedvomno bodo omogočili bistveno večjo funkcionalnost.

Dosedanji razvoj programske opreme je šel v veliki meri neodvisno od razvoja strojne opreme. Pri tem je potrebno razlikovati programsko in strojno homogene sisteme, ter programsko neodvisnost od strojne opreme. Homogenost programske in strojne opreme daje visoko integrirane sisteme, medtem ko je neodvisnost programske opreme ena njenih ključnih komponent kakovosti. Zato je razvoj ISKRA-DELTE usmerjen v razvoj lastne programske in strojne opreme, čeprav se s tem odpovedujemo tržišču ISKRA-DELTA kompatibilnih računalnikov (RSX-11/M, VMS, UNIX itd).

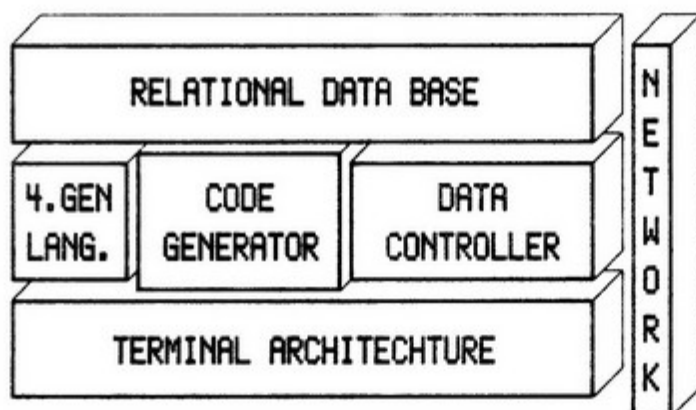


4. ZNAČILNOSTI NOVEGA PROIZVODA ISKRA-DELTA TEHNOLOGIJE

4.1 SPLOSNE ZNAČILNOSTI

Novi trendi razvoja so zahtevali tudi odgovor in dopolnitev ISKRA-DELTA Arhitekture. Nova generacija informacijskih orodij "IDA" ne bo več deljiv proizvod. Zato ga ne bo mogoče dobavljati po delih kot obstoječo verzijo, kjer so funkcije med posameznimi orodji zelo ostro razmejene. To je veljalo za posebno odliko, ker so se posamezna orodja vklapljala v že obstoječe uporabnikove produkte. Kompleksnost nove tehnologije zavrača tovrstno rešitev pri novi generaciji IDA informacijskih orodij.

Nova generacija IDA informacijskih orodij pa ne temelji samo na visoki povezanosti programskih rešitev, ampak je vezana na razvoj strojne opreme. Zato je razvoj strojne opreme povezan z razvojem programske in komunikacijske opreme s ciljem realizacije homogenega jedra sistema, ki naj bi deloval čimbolj učinkovito. To ne pomeni, da bo novi sistem vezan na strojno opremo, temveč da bo v povezavi z novo strojno opremo dosegel optimalno uporabnost.



4.2 GLAVNE ZNAČILNOSTI PODATKOVNEGA KONTROLERJA

Povezovalni del novega proizvoda je podatkovni kontroler, ki formalizira interni podatkovni model za potrebe IDA orodij, oziroma njegovih aktivnih programskih komponent. Nadgradnja modela in njegova vključitev v uporabnikov model je omogočena tudi končnemu uporabniku. Zato je podatkovni kontroler centralni sistem, kjer so zabeležene vse informacije o obliki in povezavah med uporabnikovimi in IDA elementi sistema. Medtem, ko se v IDA-LEKSIKONU informacije samo interaktivno beležijo, kar je sicer določen napredek, pa je podatkovni kontroler nepogrešljiva komponenta novega sistema. Vsebuje življensko pomembne informacije za pravilno delovanje ostalih segmentov sistema, kot je npr. interni opis za bazo podatkov.

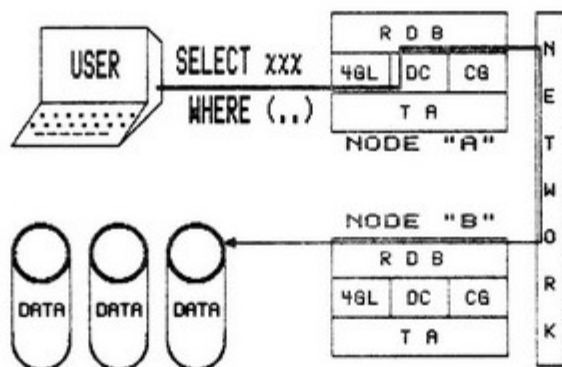


4.3 RELACIJSKA BAZA PODATKOV

Vmesnika med vhodno-izhodnimi enotami in programsko nadgradnjo sta sistem za upravljanje podatkov in sistem za delo s terminali.

Novi program za delo s podatki lahko dopolnjuje svoje strukture v realnem času, torej interni opis gradi dinamično. Slednji ni statičen, ampak je po delčkih zapisan v entitetah podatkovnega kontrolerja. Posamezni delčki zapisov se sestavijo v celoto sele ob startanju jedra, oziroma interaktivnem definiranju in odpiranju novih tabel (zapisov). Ker se lahko nove strukture dopolnjujejo dinamično, se s tem dopolnjuje tudi interni opis v podatkovnem kontrolerju.

Drugi temeljni cilj nove filozofije za delo s podatki je v neodvisnosti programskega konteksta od fizičnega konteksta baze podatkov. To pomeni, da se lahko poljubni program izvaja na kateremkoli sistemu, komunicira pa s poljubnimi bazami podatkov, ki se nahajajo na drugih vozliščih v mreži. Na vozliščih so lahko tudi računalniški sistemi drugačnega tipa, oziroma arhitekture (mešani modeli). Del tega omogoča že DDBC z omejitvijo, da deluje samo na DELTA/V sistemih. Novi sistem pa bo omogočil tovrstno komunikacijo tudi na heterogenih računalniških mrežah. Slednje je načrtovano zaradi malih računalniških sistemov kot je sedaj "TRIGLAV", katerim je s tem zagotovljeno mesto samostojnih delovnih postaj, povezanih z drugimi bazami podatkov.



Zamisli o distribuiranih samostojnih enotah, kjer se izvajajo programi na manjših sistemih z možnostjo posega v skupne baze podatkov ima poleg varnosti in večje stopnje razpoložljivosti tudi ekonomsko računico. Cena posameznega posla je na majhnih sistemih bistveno nižja, poleg tega tovrstna obdelava ne zahteva centralizacije organizacije. Bolj preprosto: baza prihaja v novem sistemu k uporabniku, oziroma je dosegljiva njegovemu osebnemu sistemu in ne več obratno, ko je moral posameznik fizično na lokacijo, kjer je bila shranjena baza podatkov, oziroma se je moral prilagajati opremi, na kateri je delovala baza. Lokacija podatkov je transparentna.

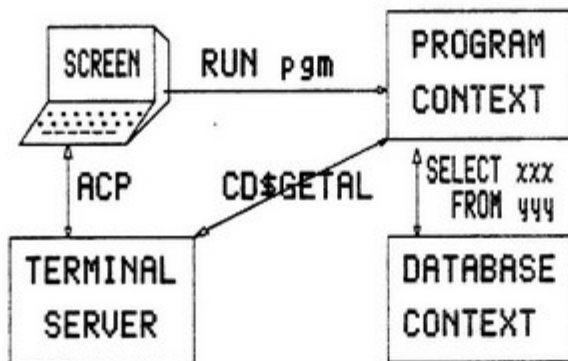


Nadgradnja osnovnemu sistemu za delo s podatki na masovnih medijih je relacijska baza podatkov. Njena izgradnja vsebuje indeksna drevesa, ki bodo omogočila urejen pregled pojavkov, posnetek trenutnega stanja (snapshot) in stike (join). Poleg ukazov za manipuliranje s podatki, bodo vgrajene tudi agregatne funkcije za razne poizvedbe. Poseben vmesnik bo omogočal delovanje programov, ki uporabljajo IDA bazo (Kompatibilnost z obstoječo bazo).

4.4 NOVE FUNKCIONALNOSTI PROGRAMA ZA DELO S TERMINALI

Novi sistem za delo z ekrani bo podpiral ekransko tehnologijo tudi za VT220 funkcije. Razvijamo distribuirano ekransko tehnologijo.

Za novo, distribuirano ekransko tehnologijo je predvidna podobna filozofija, kot pri sistemu za delo z bazo podatkov. Centralni program, ki skrbi za pravilno delovanje terminalov naj bi bil oddvojen od vozlišča, kjer se nahaja program. V sistemu GEMINI je to na primer servisni procesor. Nova ekranska tehnologija bo skrbela tudi za delo z delovnimi enotami v smislu predprocesiranja terminalskih ukazov. Preprosti ukazi bodo omogočili novo kakovost delovnih postaj.



4.5 JEZIKI ČETRTE GENERACIJE IN SNOVALCI KODE

Trendom na področju razvoja neproceduralnih jezikov bo sledil tudi novi sistem. Za končnega uporabnika bo razvit jezik za delo z bazo podatkov. Poleg funkcij, ki jih običajno nudijo tovrstni jeziki - poizvedovanje, vpis brisanje in spreminjanje podatkov v bazi bo novi sistem vseboval tudi množico vnaprej določene programske kode. Medtem, ko je stari sistem deloval predvsem kot pomoč programerju, pa bo novi sistem uporaben tudi za končnega uporabnika brez znanja programiranja COBOLA in podobnih računalniških jezikov. V ta namen je v novem sistemu zasnovano jedro, ki temelji na uporabi preprostih izvršilnih ukazov.



"Snovalci kode" bodo omogočili končnemu uporabniku hitro in enostavno delo s podatki. Komunikacija med končnim uporabnikom in snovalcem kode bo potekala po vzoru SQL standarda. Seveda se bodo vsi podatki sproti zapisovali v podatkovni kontroler, oziroma bodo od tam tudi jemali. Nabor entitet bo iz sedanjih podatkov o bazi in ekranskih slikah kot ga pozna IDA-COGEN razširjen tudi na entitete, ki bodo zagotavljale pregled nad posameznimi programi in dokumentacijo.

5. ZAKLJUČEK

Kot je že bilo zapisano, je razvoj nove generacije IDA informacijskih orodij močno vezan na razvoj programske, komunikacijske in strojne opreme. Zaradi neodvisnosti strojne opreme od programske, bo novi sistem deloval na vsem računalniških sistemih ISKRA-DELTA. Optimalna uporabnost pa bo v kombinaciji majhnih samostojnih enost v povezavi z večjimi.

Programska samostojnost zagotavlja tudi prilagoditev na nove proizvode, na nove, bolj zmogljive tehnologije. Pri tem se filozofija in uporabnost produkta ne spremeni. Tako bodo v prihodnosti posamezne komponente strojne opreme zamenjali bolj zmogljivi procesorji, vendar zasnove in organizacije uporabniških rešitev zato ne bo potrebno spreminjati.

S tem IDA omogoča prehod na nove tehnologije, za katere se predvideva veliko povečanje transakcij in njihova pocenitev. Hkrati zagotavlja stalno uporabnost uporabnikovih proizvodov ne glede na generacijske spremembe, kar je glede na stroške razvoja velikega pomena.

6. LITERATURA

- (1) Paul Ward & Stephen J. Mellor
"STRUCTURED DEVELOPMENT FOR REAL-TIME SYSTEMS" - Yourdon Press, New York 1985
- (2) James Martin
"DATA-BASE ENVIRONMENT" - Prentice Hall, Englewood Cliffs 1983
- (3) Joachim W. Schmidt & Michael L. Brodie
"RELATIONAL DATABASE SYSTEMS" - Springer Verlag, Berlin 1983
- (4) David K. Hsiao
"ADVANCED DATABASE MACHINE ARCHITECTURE" - Prentice Hall, Englewood cliffs 1983
- (5) Jeffrey D. Ullman
"PRINCIPLES OF DATABASE SYSTEMS" - Computer science press 1982
- (6) C.J. Date
"A GUIDE TO DB2" - Addison-Wesley publishing 1984
- (7) Wilbert O. Galitz
"HANDBOOK OF SCREEN FORMAT DESIGN" 1983
- (8) B.W. Lampson, M. Paul, H.J. Siegert
"DISTRIBUTED SYSTEMS ARCHITECTURE AND IMPLEMENTATION"
Springer-Verlag Berlin 1981
- (9) Dimitris Charafas
"FOURTH AND FIFTH GENERATION PROGRAMMING LANGUAGES"



**PROBLEMI RAČUNALNISKIH JEZIKOV
IN RAZVOJA APLIKACIJ**

Avtor referata: Milan Vukelić, dipl.ing. ISKRA DELTA
Milijan Đokić, dipl.ing. ISKRA DELTA
Vili Kovačić, dipl.oec. ISKRA DELTA



Ljubljana, maj 1987



1. KRATEK ZGODOVINSKI PREGLED

Problem razvoja poslovnih aplikacij na računalnikih je prisoten že od samega začetka, tj. od pojave prvih računalnikov.

Osnovni računalniški jezik (assembler) je zmožen rešiti tudi najbolj zahteven poslovni problem, toda razvoj aplikacije je v tem jeziku težaven, predvsem zaradi zapletenosti jezika samega. Zaradi tega ga praviloma ne uporabljamo pri razvoju poslovnih aplikacij. Danes v svetu uporabljajo v ta namen nesteto visjih programskih jezikov, najpogosteje COBOL, PL/I, BASIC, FORTRAN in druge. Osnovna pomanjkljivost vseh je, da zahteva solanega profesionalca, ki lahko uspešno razvije aplikacijo in potem se uspešno tudi vzdržuje obstoječo.

Ena ameriška raziskava predvideva, da bo že pred začetkom naslednjega desetletja več kot 95% ljudi, ki koristijo računalnik, neprofesionalcev (end user). Iz tega izhaja potreba, da se razvoj (in tudi vzdrževanje) aplikacij poenostavi in olajša, posebej ker ista raziskava predvideva da bo 80% aplikacijskih programov napisanih od strani neprogramerjev v enem od takoimenovanih naravnih jezikov (neproceduralni, user-friendly, end-user jeziki), ki jih skupno imenujejo software 4. generacije, in tukaj spadajo različni Query jeziki, program generatorji in aplikacijski generatorji.

2. GENERATORJI SOURCE KODE - IDA COGEN

IDA Cogen je vmesna rešitev problema razvoja poslovnih aplikacij. Produkt olajša COBOL programerjem delo pri pisanju programov, toda namenjen je profesionalcem, tako kot številni drugi generatorji source kode. Problem prevajanja, povezovanja in nasploh vzdrževanja ostane nespremenjen.

3. SOFTWARE ORODJA 4.GENERACIJE

Osnovni cilj teh orodij je omogočiti tudi neprofesionalcem uspešno delo pri razvoju aplikacij na računalnikih.

Prednosti teh orodij so enostavnost uporabe v "naravnem" jeziku, relativno hiter razvoj aplikacijskega programa ali aplikacije ter velika verjetnost, da je program (in/ali rezultat) pravilen - testiranja niso nujna.

Pomankljivosti so povezane z zmožnostjo posameznega produkta v tem smislu, da je ponavadi enostaven za uporabo ter brez napak tisti produkt, ki ima manjše zmožnosti (različni report writeri, query jeziki in td.)



V to grupo lahko uvrstimo DATATRIEVE kot dober QUERY jezik, ki ga neprofesionalec lahko uspešno uporablja že po 2 dneh treninga, toda samo za enostavnejše operacije in manj zahtevne aplikacije.

FORMATIX, ki je tipični predstavnik program generatorja je 100% zanesljiv in brez napak pri programih določenega tipa (programi za vnos in ažuriranje podatkov, menu programi ter enostavnejši izpisni programi). Ne moremo ga uvrstiti med generatorje aplikacij, ker se zahtevnejša aplikacija z njim ne da generirati (čeprav lahko razvijemo tudi do 50% aplikacije kot je na primer GLAVNA knjiga). FORMATIX je enostaven, hiter in zanesljiv.

APPLICATION GENERATING PROGRAM-AGP

V sledečem vam želimo na kratko predstaviti novi programski produkt AGP - Application Generating Program. Produkt se uvršča v software 4. generacije in služi za hitro, učinkovito in enostavno razvijanje in vzdrževanje poslovnih aplikacij. AGP direktno generira izvršno programsko kodo (brez prevajanja in linkanja). To ni programski jezik niti query jezik ampak predstavlja povsem nov pristop pri reševanju problemov, ki se pojavljajo v poslovnih aplikacijah.

Produkt služi tako za prototyping kot za izdelavo končne delovne verzije aplikacije. Namenjen je sistem analitikom, projektantom in organizator-programerjem. Ocenjuje se, da z njim lahko nadomestimo klasično programiranje s področja poslovnih aplikacij tudi čez 90 % primerov. Prav tako ocenjujemo, da se z uporabo tega produkta povečuje produktivnost programiranja za 10 in večkrat.

Paket v tej verziji vključuje delo z RMS datotekami pri čemer izkorišča vse možnosti ki ji daje RMS, in ki jih v višjih programskih jezikih (npr: Cobol-u) ni možno uporabljati (segmentirani ključi, dublicirani primarni ključi). Programi izdelani s tem produktom so povezljivi (chain funkcija) s programi izdelanimi s klasičnimi programskimi jeziki.

Aplikacije odnosno programi izdelani s tem paketom imajo optimalno kodo (zelo majhna koda). Meritve učinkov so pokazale tudi optimalnost v izvajanju. Pri interaktivnih programih je hitrost izvajanja enaka kot pri klasično generiranih cobolskih programih, v batch obdelavah pa je hitrost izvajanja samo cca 10% manjša od enakega cobolskega programa. S tega vidika AGP lahko označimo, za razliko od nekaterih drugih funkcionalno sličnih produktov, kot produkt orodje z visokimi performabsami. AGP ne rešuje vseh problemov, vendar to kar rešuje, rešuje na najboljši možni način.



Paket je izdelan za sisteme Delta 4850, GEMINI, ADRIA in VAX pod operacijskim sistemom Delta/V odnosno VMS. Predvidena je izdelava tega paketa za sisteme Triglav (Delta/M in RSX).

Razen RMS, ki je sestavni del Sistemov Delta/V in Delta/M, bo paket imel tudi povezavo (interface) na razne data base sisteme: RDB, IDA baza, DBMS.

Dialog z AGP je maksimalno orientiran na to, da uporabniku olajša delo (default odgovori, spisek vseh možnosti in on-line pomoč za delo - HELP).

Produkt je bil (verzija 0.1) prvič predstavljen na sejni Interbiro 86 in je - sodeč po anketi, ki smo jo izvedli med uporabniki - projektanti, sistem analitiki in programerji naletel na zelo dober sprejem in interes za čimprejšnjo aktivno uporabo.

Za uporabo AGP-a bo organiziran petdnevni tečaj v Iskra Delta centru v Novi Gorici. Na razpolago je tudi prva verzija priročnika (reference manual). Prva distribucijska verzija proizvoda bo na razpolago v začetku meseca junija 87.

Funkcionalno je AGP sestavljen iz dveh osnovnih celot:

* DATA DICTIONARY (DD)

V DD so vse informacije o podatkih ki se uporabljajo v aplikaciji:

- tip file system-a (RMS, RDB, IDA, ...),
- naziv datotek in tipi datotek
- veze med posameznimi datotekami (ključi, ...)
- strukture vseh zapisov,
- nazivi polj in njihove karakteristike,
- kontrolne tabele, datoteke,
- informacije o generiranih programih
- povezave med programi itd.

* PROGRAM GENERATOR (PG)

Služi za generiranje interaktivnih programov, batch programov in poročil.

Predpogoj za generiranje programov je, da uporabnik definira - nastavi Data dictionary, ki je lahko za vsako aplikacijo poseben ali pa skupen za več aplikacij.

V generatorju interaktivnih programov je vključen screen painter, ki omogoča risanje ekranskih slik po takoimenovanem windowing principu. Slike v fazi risanja (kreiranja) in v run timu so popolnoma identične. Screen painter je vključen tudi pri generiranju poročil (report generator), kar omogoča uporabniku, da takoj vidi izgled liste, ki jo želi kreirati.



Produkt je odprt za implementacijo novih funkcij pri čemer bo odločilen kriterij potrebe prakse, upoštevajoč pri tem zahtevo, da produkt ne sme biti kompliciran za uporabo.

DATA DICTIONARY - DD

DD je prva faza kreiranja vsake aplikacije. V DD se nahajajo informacije o podatkih o posamezni aplikaciji ali zbiru več aplikacij.

DD se uporablja na dva načina:

Prvi je, ko projektant ve v naprej vse definicije o podatkih. Projektant enostavno vnaša informacije o podatkih, ki se nanašajo na aplikacijo ki bo kreirana. Lahko se tudi menja neka predhodno vnešena definicija, vnese se neka nova predhodno pozabljena definicija ali se izbriše obstoječa nepotrebna definicija.

Drugi način uporabe je, da se v DD lahko implementirajo že obstoječe aplikacije (izdelane s klasičnim programiranjem) na ta način, da se u DD kopira opis datoteke neke obstoječe aplikacije izdelane v nekem višjem programskem jeziku (Cobol,..). DD omogoča tudi uporabo datotek za programiranje v "klasičnih" višjih programskih jezikih.

PROGRAM GENERATOR - PG

PG predstavlja drugi del AGP-a. Ko je DD uspešno definiran in napolnjen s potrebnimi informacijami se aktivira PG.

PG je podobno kot DD interaktivni program, ki dela v dialogu z uporabnikom, kar pomeni, da mora uporabnik odgovoriti na določeno število vprašanj. Odgovori so najpogosteje izbor ene od več možnosti. V slučaju, da uporabniku ne odogovarja nobena od prikazanih variant, se lahko s pomočjo posebne tipke aktivira "instruction mode". Uporabnik ima v tem primeru na razpolago večje število specifičnih ukazov.

1. MENU

GENERATOR

Menu-ji so programi ki povezujejo programe v aplikacijo. Treba je definirati ekran in navesti potrebne povezave na druge programe.

2. INTERACTIVE PROGRAM

GENERATOR

Interaktivni programi imajo intenzivno komunikacijo s terminalom in uporabljajo vse lastnosti video terminalov.



3. BATCH PROGRAM and REPORT

GENERATOR

imajo omejeno komunikacijo z uporabnikom, ki se odraža v enostavnem accept ali display na začetku programa. Program se naprej odvija po programsko definiranem ciklusu brez vpliva uporabnika.

Izbire Batch program and Report generator-ja so:

- * A REPORT - - - - - poročila iz datotek
- * A BATCH PROGRAM - - - - - obdalava z datotekami
- * REPORT and BATCH - - - - - kombinacija report in batch programov
- * MAILING LABELS - - - - - izpis nalepk in nalepkam podobnih izpisov (splošne nakaznice, virmani, nalogi, čeki, ...)
- * LETTER PRINTING - - - - - pisanje pisem in drugih dokumentov, ki se kombinirajo iz več datotek.
- * RECORD SELECTION & SORT - - - - selekcija in sortiranje slogov v datotekah.

PG kot rezultat daje task / image file, ki je takoj na razpolago za izvajanje. Vse informacije o generiranih programih so shranjene v DD-ju. To omogoča izvajanje popravkov v generiranih programih.

AGP producira tudi vso potrebno interno - tehnično dokumentacijo, ki je potrebna za nadaljne vzdrževanje programov.

Katerakoli definicija se lahko izpiše za interno dokumentacijo. Ta dokumentacija se lahko lista na ekranu ali pa se izpiše na tiskalniku.

Na vseh nivojih kreiranja aplikacije je možno dokumentirati posamezne segmente s pomočjo internega Tekst procesorja.

V svetu je ogromno število podobnih proizvodov. Nastejmo samo nekaj značilnih: RELLY, POWERHOUSE, SMARTSTAR, XENTIX ...) Praviloma so zelo dragi in z istimi dobrimi in slabimi lastnostmi kot AGP. Nobeden ne generira, tako kot AGP, kodo že pripravljeno za izvajanje, in za vse nastete velja, da morajo biti instalirani na računalniškem sistemu tudi v fazi uporabe aplikacije, ki smo jo prethodno razvili, kar za AGP ni nujno (možno je razvito aplikacijo prenesti na drug računalnik istega tipa, ki nima instaliranega AGP-ja in jo uspešno uporabljati).



4. TRENDI RAZVOJA

Razvoj orodij 4. generacije je v smeri, da bi čimveč povečali zmožnosti le teh (ideal je razviti 100% poljubno poslovno aplikacijo z uporabo orodja), pri tem pa ohraniti enostavnost in zanesljivost, kar je zelo težka naloga, ki v celoti najbrž ne bo rešena do konca tega stoletja!



**NOVI PRISTOPI VRS
PRI VZDRZEVANJU
RACUNALNISKE OPREME**

**Avtorja referata: Drago Babnik, ing.
Lidija Perko, oba ISKRA DELTA**



Ljubljana, maj 1987



Vse aktivnosti v VZDRŽEVANJU RAČUNALNIŠKIH SISTEMOV (VRS) so podrejene doseganju enega samega cilja: nuditi uporabniku vsestransko pomoč pri uporabi računalniške opreme, kajti nakup računalniške opreme postaja projekt, ki poskuša rešiti vse zahteve in potrebe kupca.

V projekt nakupa računalniške opreme se VRS vključuje s "presales supportom", kjer v sodelovanju s prodajnimi inženirji svetuje kupcu najustreznejšo aparaturno in programsko opremo. To konkretno pomeni, da se investiranje v računalniško opremo uresničuje v zanesljivosti, učinkovitosti in uporabnosti računalniške opreme kot orodja za doseganje poslovnih ciljev.

Za uresničevanje tako postavljenih ciljev je VRS organiziran po načelu dislociranih vzdrževalnih centrov, ki so med seboj mrežno povezani in delujejo kot samostojne celote v okviru smernic razvoja in strategije VRS - Iskre Delte. Tako postavljen organizacijski koncept omogoča neposreden stik z uporabnikom.

VRS se zaveda dinamičnega spreminjanja tržnih zakonitosti, zato razvija nove oblike vzdrževanja in njegove spremljajoče aktivnosti.

Izbira ustrezne oblike vzdrževanja je prvi in najpomembnejši korak pri oblikovanju profesionalnega odnosa med VRS-jem in uporabnikom.

Pri tem VRS upošteva :

- * tip uporabnika (osnovno dejavnost, računalniško znanje)
- * konfiguracijo aparaturne in programske opreme
- * vrsto aplikativne programske opreme
- * okolje (geografsko, računalniški prostor)

VRS je na podlagi teh meril organiziral:

- * za uporabnike računalniške opreme, ne glede na tip vzdrževalne pogodbe, možnost prijavljanja okvar in opravljanja intervencij tudi izven rednega delovnega časa
- * za uporabnike terminalske in mikro računalniške opreme nov pristop vzdrževanja po načelih:
 - . "man-in-the-van"
 - . "unit-swap"
 - . količinski popust



* vzdrževanje aplikativne programske opreme s statusom produkta, kar uporabniku omogoča:

. enovit postopek prijavljanja motenj v delovanju računalniške opreme ter

. krajšo in učinkovitejšo pot do funkcionalnosti računalniške opreme

* vzdrževanje programske opreme, v katerem VRS nudi uporabniku telefonsko pomoč in nove verzije programske opreme

Nove oblike vzdrževanja so možne le kot nadgradnja obstoječih, zato VRS uvaja nekatere spremljajoče aktivnosti vzdrževanja, ki prispevajo k popolnejši in kvalitetnejši ponudbi.

Vsak večji vzdrževalni center v okviru Inženiringa računalniških centrov (IRC) nudi uporabniku pomoč pri izbiri, izgradnji in opremljenosti računalniškega prostora. Koordinira pa tudi vse aktivnosti pred začetkom instalacije še v času asembliranja in končnega testiranja opreme ter instalacijo samo.

Specialistično usposobljen kader za vzdrževanje aparaturne in programske opreme nudi uporabniku večnivojsko kontinuirano pomoč pri uporabi računalniške opreme v začetnem uvajalnem obdobju in v iskanju boljših rešitev pri vsakodnevnem delu z računalniško opremo. Izkušnje na tem področju kažejo, da telefonska pomoč kot oblika svetovanja pri delu z računalniško opremo, v večini primerov uspešno reši uporabnikov problem in s tem prispeva k večji zanesljivosti, učinkovitosti in uporabnosti računalniške opreme. Zato to svetovalno aktivnost razširjamo tudi na vse večje vzdrževalne centre v Jugoslaviji.

V vseh večjih vzdrževalnih centrih specializirane enote za popravilo terminalske in mikro računalniške opreme skrajšujejo čas popravila in povečujejo odstotek operativnosti tovrstne opreme.

Odkup rabljene računalniške opreme izvaja VRS z namenom, da bi poenotil in posodobil instalirano bazo. Na ta način želi zagotoviti boljše performanse računalniških sistemov glede zanesljivosti v delovanju v povezavi z novimi dosežki tehnologije vzdrževanja računalniške opreme.

VRS uporabnikom nudi možnost preskrbe s potrošnim materialom v vseh vzdrževalnih centrih oz. posreduje naročilo ustrezni centralni službi v Ljubljani.



BELEZKE



Ljubljana, maj 1987

Šola
LETNA ŠOLA

Šola