

Softari

NOKIA DATAN OHJELMISTOLEHTI • 1/1990



Järjestelmien rajapinnat hämärtyvät

Nokia Datan ohjelmistolehti

Päätoimittaja:
Terhi Ogbeide,
puh. (90) 567 2994
email: Ogbeide—Terhi—NOK

Toimitus:
Jyrki J. J. Kasvi,
puh. (90) 567 4064
email: Kasvi—Jyrki—FRL

SoftariTiimi:
Mikko Helander, Kari Kaipainen,
Outi Kuusela, Rami Raulas,
Tapio Sulkava, Anne Typpö,
Esko Uotila

Ulkoasu:
Jyrki J. J. Kasvi

Osoitteenmuutokset:
Takakannen palvelukortilla

Toimituksen osoite:
Nokia Data/Softari
PL 780
00101 HELSINKI

Painosmäärä:
12 000

Kirjapaino:
Helsingin Liikekirjapaino Oy

Julkaisija:
Nokia Data

ISSN 0782-8209

Sisäily:	
Pura ja kokoa	2
Uusi tapa luoda verkkosovelluksia	3
S2500, Tiimi ja Mikot avoimesti yhteen	6
DPS 6000 hajautuviin tietojärjestelmiin	8
Enemmän irti Windows-sovelluksista	10
Värikkäiset nestekidenäytöt tulevat jo tänä vuonna	16
Ergonomiia ruudun molemmin puolin	20
Learning Ada — Part 7	24
Moni Windows pyörii vajaateholla — turhaan	28
Verkolla yksilölliseen luokkaopetukseen	33
Uusi työasema uusiin tarpeisiin	36
Tuoteuutiset	38
Mikkonen	40

Pura ja kokoa

Siinä missä 80-luvun myyntimies kehui kau-pittelemansa laitteen soveltuvan asiakkaan kaikkiin kuviteltavissa oleviin tarpeisiin, 90-luvulla myydään erikoisratkaisuja ja mitatilaustyötä. Joskus kyseessä ovat jopa samat laitteet — ja myyntimiehet. Avointen tietojärjestelmien nopea yleistyminen onkin tehnyt mahdolliseksi hajauttaa tietojenkäsittely juuri sinne, missä kukin tehtävä on luotevinta suorittaa. Tietokannat voivat sijaita edelleen vanhan keskuskoneen massamuistissa, mutta niiden pohjalta tehtävät raportit laaditaan omassa työasemassa. Valmis dokumentti puolestaan lähtee maailmalle työryhmäverkon tietoliikennevälityksellä.

Hajautettu avoimuus tarjoaa joustavuutta; järjestelmä on helppo sopeuttaa organisaation muutoksiin minimikustannuksin. Vanha-kantainen keskitetty järjestelmä joudutaan sen sijaan hyvin usein purkamaan ja koaamaan alusta loppuun. Lisäksi koko organisaatio on siirtymävaiheen ajan hyvin haavoittuvainen.

Vapaus vaatii myös vastuuta — etenkin tietotekniikassa. Avoimet tietojärjestelmät ja tietoverkot vaativat paljon paitsi suunnittelijoiltaan myös käyttäjiltään. Vasta virusten räjähdysmäinen leviäminen on havahduttanut ihmiset pohtimaan tietoturvan välttämättömyyttä, mutta virukset ja tietovarkaavat eivät ole ainoa uuden tekniikan mukanaan tuoma ongelma. Esimerkiksi kansainvälisiin tietokoneverkkoihin on jo nyt ilmestynyt tahallaan väärinnettyjä uutisia, joiden tarkoituksena ei ole ollut suinkaan hauskuttaa vaan vahingoittaa. Tällaiset viestit saattavat muutamassa tunnissa saavuttaa satoja tuhansia lukijoita eri puolilta maailmaa.

Myös tietojärjestelmien suunnittelijoiden tulisi tuntea vastuunsa — ja ajatella hien. Jos esimerkiksi hajautettua järjestelmää käsitellään keskitetyn systeemin tapaan, sille käy hyvin pian kuin eräälle amerikkalaiselle puhelinyhtiölle, joka päivitti kertaheittolla kaikkien kustantensa ohjelmat kokeilematta niitä ensin niistä yhdestäkään...

Jyrki J. J. Kasvi



Tietotekniikkakauppiaskin joutuu seuraamaan muutovirtauksia ja markkinoiden oikkuja.



Uusi tapa luoda verkkosovelluksia

Verkkopohjaisten niin kutsuttujen neljännen aallon sovellusten rakenne perustuu työasema - palvelin - arkkitehtuuriin, jossa tieto säilytetään relaatio tietokannoissa ja kommunikointikielenä työaseman ja palvelimen välillä käytetään SQL:ää. Työaseman käyttöliittymä, graafinen tai merkkipohjainen, valitaan tarpeen mukaan, mutta graafisten liittymien suosio kasvaa koko ajan.

Tämä vuosi tulee olemaan käännekohta käyttöliittymäkehityksessä, sillä Windows 3.0 ja OS/2 Presentation Manager tarjoavat hyvän perustan sovelluskehitykselle ilman tuskastuttavia muisti- ym. rajoituksia. Lisäksi ne mahdollistavat sovellusten siirtämisen käyttöjärjestelmästä toiseen; samat sovellukset tulevat itse asiassa lopulta toimimaan molemmissa ympäristöissä sellaisenaan.

Käyttöliittymän rakentaminen muodostaa yhä merkittävämmän osan sovelluskehityksestä, sillä käyttäjät ovat oppineet vaatimaan yhä helppokäyttöisempiä ja joustavampia sovelluksia.

Tähän tilanteeseen Nokia Data tuo Tiimi-sovelluskehitysympäristönsä. Se perustuu avoimiin standardeihin, sisältää valikoidun työvälineitä ja määrittelee ohjelmointirajapinnat sekä näyttöjen rakenteen. Kaiken nivoo yhteen kattava sarja oppaita. Tätä kokonaisuutta soveltamalla on helppo luoda yhdenmukaisia Tiimi-sovelluksia, jotka eivät vaadi käyttäjää opettelemaan uutta liittymää aina sovelluksesta toiseen siirtyessään.

Ympäristöä ei tarjota yhtenä suurena pakettina, vaan se on jaettu pienempiin osiin, työkalupakkeihin (toolset, toolpack). Näitä on tarjolla niin C-ohjelmioijalle kuin tehokäyttäjällekkin.



Yritysten tietojärjestelmien hajauttaminen (downsizing) yleistyy jatkuvasti, mutta mitä se itse asiassa on, mitä se merkitsee sovelluksille? Sitä etä suur- ja minitietokoneille rakennettujen sovellusten vanhennuttua uusia sovelluksia ei enää tehdäkään keskitetyiksi vaan ne kehitetään mikropohjaisiin työryhmäverkkoihin.

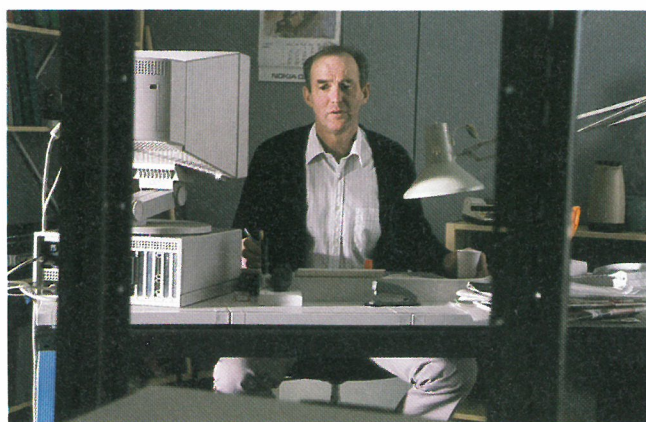
Työryhmäverkoista muodostuu kaikkien uusien sovellusten kehittämisen pääasiallinen perusta oli sitten kyse perinteisistä mikrosovelluksista kuten kortistoinnista ja tiedon loppujalostamisesta tai operatiivisten ja yritykselle kriittisten sovellusten toteuttamisesta.

Tiimi-sovellukset on tarkoitettu työryhmäverkkoon, mutta niitä voidaan kehittää myös yksittäisissä mikroissa, mikä vähentää merkittävästi kehitysvaiheen kustannuksia.

Tiedonhallintaan kaksi SQL-järjestelmää

Tiimissä on kaksi tietokantavaihtoehtoa: SQLBase, joka on osa suurempaa SQL System -kokonaisuutta, ja SQL Server, joka on osoittanut tehokkuutensa muissa ympäristöissä. Molemmat ovat OS/2-käyttöjärjestelmässä toimivia SQL-pohjaisia relaatiotietokantoja, mutta kuitenkin erilaisia. Näin kehittäjä voi valita kahdesta vaihtoehdosta parhaiten juuri käsillä olevaan projektiin sopivan ratkaisun.

SQLBase on mikroverkkoihin varta vasten kehitetty IBM DB2 (/SAA) -maailmaan nojaava ratkaisu, jonka perustana on avoi-



muus ja erilaiset yhdyskäytävät muihin tietokantoihin. SQL Serverin tavoitteena on sen sijaan ollut alusta alkaen mahdollisimman suuri on-line-suorituskyky minikonepohjaisessa ympäristössä, mikä on nyt siirretty myös OS/2-maailmaan.

SQLBase ja muut Guptan ohjelmistot

SQLBase muodostaa Gupta Technologies Inc. kehittämän SQL System -tuoteperheen perustan, jonka muut osat ovat SQL Windows-sovelluskehitin ja joukko yhdyskäytäviä, esimerkiksi SQLNet. SQLBase oli ensimmäinen mikroverkossa toimiva SQL-pohjainen monen käyttäjän relaatiotietokanta.

Nokia Data on ollut alusta alkaen mukana SQL System -tuotteiden kehityksessä käyttäen niitä useiden järjestelmiensä perustana. Tällaisia ovat olleet esimerkiksi vähittäiskaupan järjestelmät, pankkijärjestelmät sekä Tiimi-sovellukset.

SQLWindows on edelleen ainoa merkittävä MS-WINDOWS-pohjainen sovelluskehitin. Siitä on muodostumassa yleinen työväline (front-end) myös muita tietokantoja (SQL Server, IBM DB2, Oracle, jne.) hyödyntäviä sovelluksia kehitettäessä.

Kolmas painopiste, yhdyskäytävät soveltuvat Tiimi-sovellukset yrityksen muihin tietojärjestel-

miin. Niiden avulla sovellus voi käyttää muissa ympäristöissä olevia tietokantoja edes tietämättä missä ne sijaitsevat. SQLNet-konaisuus (IBM DB2) on ollut saatavilla jo jonkin aikaa, ja Oracle- sekä IBM OS/2 EE DBM-gatewayt tulevat markkinoille vielä tämän vuoden kuluessa.

Käytäviä tulee olemaan kahden tyyppisiä, toinen vaatii erillisen gateway-koneen (DOS) tai -sovelluksen (OS/2), toinen taas toimii samassa työasemassa itse sovelluksen kanssa (gateway router). Nokia Data on myös itse kehittänyt erilaisia vaihtoehtoja muiden ympäristöjen liittämiseksi esimerkiksi SQLWin-sovellukseen, kuten Tandem NonStopSQL:n.

SQLBasea on myyty nyt noin 4 vuotta, ja tänä aikana Gupta on saanut asiakkaitaan monia merkittäviä yrityksiä, kuten Bank of American, Texaco, Arthur Anderson & Co:n, Manufacturer's Hanoverin ja Traveler's Insurancen. OEM-sopimuksia Gupta Technologies on tehnyt Nokia Datan lisäksi Hewlett-Packardin, Siemensin, Nixdorfin ja Mitsubishiin kanssa.

Microsoftin SQL Server hajauttaa sovelluksen

SQL Serverin taustana on Sybase-yhtiön Unix-pohjainen tietokanta, jonka Microsoft siirsi OS/2-käyttöjärjestelmään. SQL Server integroi kaksi eri ympäristöä tiedonhallintakokonaisuudeksi, jossa työasemasovellus pystyy käyttämään esimerkiksi tehokasta Mikko300/Unix-konetta samalla tavalla kuin mikropohjaisista OS/2-palvelimista.

SQL Server tarjoaa tietokantapalvelimen, jossa on runsaasti tehokkuutta parantavia piirteitä kuten tallennetut SQL-lauseet ja triggerit. Niiden avulla voidaan hajauttaa esimerkiksi sovellus, jolloin sen osia voidaan ajaa samanaikaisesti eri palvelimissa ja lisätä siten järjestelmän kokonaissuorituskykyä. Tätä hajauttamista tukee myös SQL Serveriin

sisällytetty two phase commit-toiminto. Näitä ominaisuuksia hallitaan Transact-SQL-kielellä, joka sisältää tavanomaisen SQL:n lisäksi tavallisten ohjelmointikielten piirteitä, kuten IF-THEN-ELSE rakenteet, muuttujien käsittelyn sekä ajastimen hallinnan. Transact-SQL onkin siten itse asiassa täysimittainen ohjelmointikieli.

Tulossa on myös muihin tietokantoihin suuntautuvien yhdyskäytävien rakentamisen mahdollistava Open Server -paketti. Ensimmäinen valmis yhdyskäytävä tulee IBM DB2:een.

SQL Server on saatavilla yli kymmeneen eri kone- ja käyttö-

Windows 3.0 ja OS/2 Presentation Manager tarjoavat hyvän perustan sovelluskehitykselle ilman tuskastuttavia muisti- ym. rajoituksia.

järjestelmäympäristöön, ja se on yksi tehokkaimmista saatavilla olevista OLTP-tiedonhallintajärjestelmistä. Työryhmäverkkojen SQL Serverille on myös tulossa joukko erilaisia edustavilaineita, sillä johtavat ohjelmistotalot kuten Borland, Ashton-Tate, Information Builders, Mdb's jne. ovat ilmoittaneet kehittävänsä SQL Serveriä tukevia ohjelmistoja.

3GL-työkalupakki ohjelmoijan avuksi

Perinteisistä ohjelmointikielistä Tiimi tukee C:tä ja COBOLia. Niiden käyttöä helpottamaan on tarjolla myös joukko erilaisia apuvälineitä ja ohjeistusta. Näytönhallintaan käytetään Ergo Formsia, jonka avulla saadaan piirrettyä merkkipohjaisia näyttöjä, jotka toimivat C- tai COBOL-ohjelmista kutsuttuina. Muita työkalupakista löytyviä vä-

lineitä ovat erilaiset testaus- ja mittausohjelmistot.

Tietokantaa käytetään joko API:n kautta (C) tai niin kutsutun embedded SQL:n avulla (ESQL/COBOL). API:n käyttö on joustavampaa ja tehokkaampaa, mutta ESQL on helpompaa, joten sen myös oppii nopeammin.

Näillä välineillä on mahdollista rakentaa Tiimi-sovelluksia, joiden pääpaino on yksinkertaisuudessa ja syöttö-päivitys-toiminnoissa. Työasemille ei tässä ympäristössä aseteta juuri mitään vaatimuksia, joten kustannukset jäävät haluttaessa alhaisiksi myös itse tuotantoverkossa.

Nokia Data on kehittänyt SATin, SAL Application Translatorin, joka kääntää SQLWindows-SAL-sovellukset C-koodiksi.

4GL/GUI-työkalupakki käyttöliittymäkehitykseen

Graafisen käyttöliittymän rakentamisen perustana on SQLWindows-kehitin. Sillä voidaan rakentaa sovelluksia, jotka toimivat MS-Windows-ympäristössä ja myöhemmin sellaisenaan myös OS/2 Presentation Managerissa. Ensimmäiset tietokannat SQLWindowsia käytettäessä ovat SQLBase ja SQL Server, mutta myös muita voidaan hyödyntää.

Nokia Datan on jo runsaasti kokemusta sovellusten kehittämisestä MS-Windows-ympäristöön. Tämä kokemus on kerätty SQLWindows Developer's Guideen ja Nokia User Interface Style Guideen, joita soveltamalla voidaan luoda yhtenäisiä Tiimi-sovelluksia.

Sovellusten suorituskyvyn parantamiseksi Nokia Data on kehittänyt SATin, SAL Application Translatorin, joka kääntää SQLWindows-SAL-sovellukset C-koo-

diksi, josta saadaan tavanomaisesti kääntämällä ja linkkaamalla sellaisenaan ajettavissa oleva ohjelma. Tarjolla on myös muita apuneuvoja, kuten help-toimintojen rakentamista helpottavia välineitä ja raporttikehittämiä.

Tietokantapaketti ylläpidon tueksi

Tietokannan ylläpito on vaikeaa ja vaatii usein paljon käsityötä. Rakenteen optimointi, tietojen poisto ja uudelleen lataus sekä tilastotietojen kerääminen ovat hyviä esimerkkejä toiminnoista, joita helpottamaan Nokia Datan tietokantapakettin apuvälineet on tarkoitettu. Relaatiotietokannan suunnittelu vaatii myös runsaasti työtä, jota nopeuttamaan on tarjolla ohjeistusta sekä koulutusta.

PowerUser-paketti yhdistää sovellukset

Tehokäyttäjä tarvitsee välineet, joilla hän voi nopeasti kehittää ad hoc -tyyppisiä sovelluksia. Pääasialliset tarpeet liittyvät tietojen keräämiseen, analysointiin ja esittämiseen. Sovellusten on toimittava myös luontevasti keskenään eli tiedon on siirryttävä eri sovellusten välillä kivuttomasti. Näihin tarpeisiin on tarjolla esimerkiksi EXCEL-SQLBase-linkki, XLSQL. Sen avulla Excel-taulukko voidaan liittää SQLBase-tietokantaan, jolloin tietokannan tiedon muuttuessa myös sitä vastaava taulukon solu muuttuu. Saman menetelmän avulla solu voidaan kiinnittää myös esimerkiksi DB2-tietokantaan.

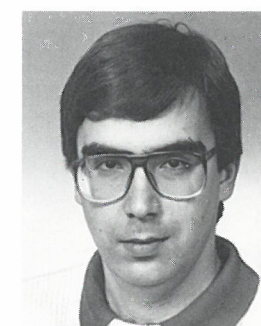
XLSQL mahdollistaa tiedon esittämisen graafisesti Excelin chart-toiminnon kautta. SQL Server -ympäristössä sama voidaan toteuttaa Q+E-ohjelmiston avulla. Tulossa on vastaavia liittymiä myös Lotus 1-2-3:aa, Word for Windowsia, ja monia muita so-

vellusohjelmia varten. Näin kaikki tietokannassa oleva tieto on myös tavallisten PC-sovellusten käytettävissä.

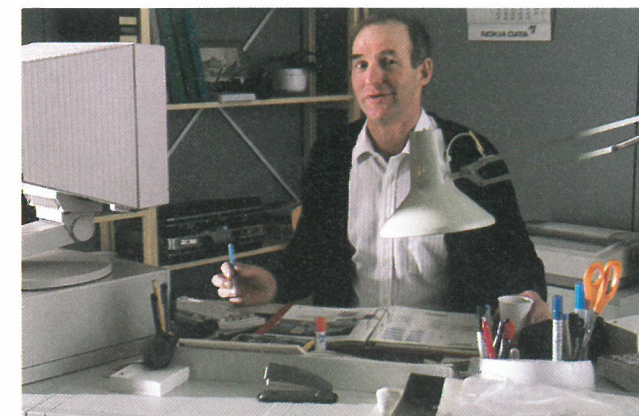
SQL SYSTEM -tuoteperheeseen kuuluvan ExpressWindowsin avulla kuka tahansa voi luoda yksinkertaisen syöttö-päivityskysely-sovelluksen tuntematta lainkaan SQL:ää tai tietokannan rakennetta.

Tulevaisuus: CASE, OOP ja RPC

Kaikki työkalupakit kehittyvät koko ajan. Uusia välineitä evaluoidaan ja kehitetään, jotta sovelluskehitys sujuisi yhä vaivattomammin ja itse kehitysprosessin hallinta helpottuisi. Tekniikan ja markkinoiden kypsyessä Nokia Data tulee tarjoamaan välineitä myös CASEen ja tietämyspohjaisten järjestelmien kehittämiseen. Kaksi muuta merkittävää uutta aluetta ovat RPC (Remote Procedure Call) eli sovelluslogiikan hajauttaminen ja oliokeskeinen ohjelmointi, OOP. Piirteitä molemmista on jo käytössä, mutta niiden täysimittainen käyttöönotto on vielä edessä. ■



Esa Aho työskentelee Nokia Datan Tuotemarkkinoinnissa vastuualueenaan Tiimin ohjelmistot.



S2500-liitântä tuotteilla

S2500, Tiimi ja Mikot avoimesti

S2500-järjestelmän liitântä-tuotteet tarjoavat joukon eri vaihtoehtoja S2500:n, Tiimin ja Mikkojen yhteenliittämiseen. Sovellusten välinen kommunikointi, tiedostojen siirrot ja pääteyhteudet toimivat sekä S2500-järjestelmästä Tiimiin ja Mikkoihin että Tiimistä ja Mikrosta S2500-järjestelmään päin.

Nokian Tiimi-työryhmäverkko on avoin työasema-palvelin-arkkitehtuuriin perustuva järjestelmäratkaisu, joka käyttää henkilökohtaisina työasemina lähiverkoon liitettyjä MikroMikkoja. Tähän asti verkon tietokanta- ja sovelluspalvelimina ovat toimineet MikroMikkojen palvelinmallit sekä Mikko Unix-järjestelmät. Uusien liitântätuotteiden myötä myös S2500 voidaan nyt liittää joukon jatkoksi.

Uudet avoimet sovellusmahdollisuudet

S2500 ja Tiimi muodostavat ja monipuolisen sovellusjärjestelmän. Tiimi-työasemaverkko antaa S2500-järjestelmän käyttäjille mahdollisuuden hyödyntää Tiimin tarjoamia uusia sovellusalueita yhdessä S2500-järjestelmän valmissovellusten kanssa.

Yhdyskäytäväratkaisu yhdistää Tiimin MikroMikko-työasemat S2500-järjestelmään. Työryhmäverkon yhteiset palvelut tuovat ympäristöön joustavuutta ja Tiimin henkilökohtaiset työvälineet valmisohjelmistonsa. S2500 tarjoaa työryhmälle puolestaan mahdollisuuden hyödyntää laajoja yri-tyssovelluksiaan.

Tiimi-työryhmäverkko tuo S2500-järjestelmän käyttäjälle työvälineet tekstinkäsittelyyn, taulukkolaskentaan, henkilökohtaisiin rekistereihin, esitysgrafiikkaan ja julkaisujen tuottamiseen. Jokain en käyttäjä työskentelee omalla työasemallaan ja käyttää siinä olevia ohjelmistoja. Sen lisäksi käytettävissä ovat S2500-järjestelmän yrityssovellukset liikekirjanpitoon, myyntireskontraan, ostoreskontraan, materiaalihallintoon, tuotannonohjaukseen, auto-



S2500-järjestelmä yhdistyy osaksi avointa Tiimi-työryhmäverkkoa. Näin S2500 voi toimia yhdessä Tiimin työasemina toimivien MikroMikkojen, OS/2- ja Mikko-palvelinten sekä Nokian avointen Unix-järjestelmien kanssa.

kauppaan, korjaamotoimintaan, varaosamyyntiin, palkanlaskentaan ja pankkiyhteyksiin. Lisäksi Tiimi-työryhmäverkko tarjoaa tarvittaessa vielä palvelimissa olevien yhteisten sovellusten palvelut.

Siirtyminen verkosta S2500-järjestelmään

Käyttäjän on helppo siirtyä Tiimin ohjelmistojen käytöstä S2500-järjestelmään. Käyttäjän ei tarvitse kuin käynnistää LANCON-yhteys-ohjelmisto, jonka avulla hän pääsee käyttämään S2500-järjestelmän sovelluksia. Siirrettäessä tietoa S2500-järjestelmästä Tiimiin, käyttäjä voi valita kohteeksi joko oman työasemansa tai verkon palvelimen.

S2500-järjestelmän tietoja voivat käyttää hyväksi kaikki Tiimi-työasemaverkon käyttäjät käyttö-oikeuksiensa rajoissa. S2500-järjestelmä vaatii käyttäjältään oikeuden siirtyä Tiimistä yrityksen sovelluksiin ja tietoihin. Kun yhteys S2500-järjestelmään otetaan omalta työasemalta käyttäjän tulee antaa henkilökohtainen käyttäjätunnuksensa ja salasana. Väliakainen siirtyminen S2500:sta Tiimiin ja takaisin yhteyttä katkaisematta on kuitenkin mahdollista.

Tarpeellinen tieto tallennetaan Tiimiin

Käyttäjän omilla työvälineillään kuten tekstinkäsittely-, taulukkolaskenta- tai grafiikkaohjelmistoilla luoma tieto tallennetaan Tiimi-työryhmäverkkoon. Tallennus voidaan tehdä omalle työasemalle tai verkon palvelimelle. Tiimin tiedot voivat olla myös S2500-järjestelmästä siirrettyjä. Käyttäjä tai sovellusohjelma voi siirtää tiedostot joko omalle työasemalleen tai Tiimin palvelimeen työryhmän yhteiseen käyttöön. Käyttäjä voi lisäksi tehdä poimintoja S2500:n sovellusten tuottamista tiedoista ja siirtää vain haluamansa osat (EPLINK).

S2500-sovellusten tuottama tieto on Tiimiin siirrettyinä valmiina tekstinkäsittelyä tai taulukkolaskentaa varten. Myyntitilastojen seuranta, tilastointi, ostoehdotusten tekeminen, budjetointi, tilinpäätös, hinnoittelu ja muut laskentaa tai jatkokirjoitusta vaativat tehtävät on mahdollista tehdä omassa työasemassa olevilla ohjelmistoilla. Hinnoittelun lopputulos voidaan tulostaa esimerkiksi palvelimeen liitettyllä laser-tulostimella ja lähettää valmis hinnasto suoraan myyntiverkostolle.

S2500 ja Mikkojen sovellukset

S2500-järjestelmä liittyy MikroMikko-työasemien ja -palvelimien lisäksi yhtä voimakkaasti myös verkon Mikko-palvelimiin. Mikot ovat Nokian avoimia Unix-pohjaisia järjestelmiä, jotka toimivat verkon tietokanta- ja sovelluspalvelimina. S2500 voi kommunikoida sovelluksillaan Mikkojen sovellusten kanssa sekä siirtää tiedostoja järjestelmien välillä. Kumpaankin järjestelmään liitetty käyttäjät voivat ottaa pääteyhteyden sekä S2500:aan että Mikkoon.

yhteen

Mikot ovat avoimia Unix-järjestelmiä, joille löytyy markkinoilta runsaasti valmissovelluksia S2500-käyttäjien toimintovaihtoehtoja laajentamaan. Tiimin työasemalta käsin käyttäjä voi hyödyntää sekä Mikkojen että S2500:n sovelluksia. Sovellukset voivat myös keskustella ja kommunikoida suoraan keskenään nopean lähiverkon (EAM, Ethernet Access Module) välityksellä. S2500-järjestelmän Ethernet-liitântä mahdollistaa asiakaskoh- taisten sovellusratkaisujen kehittämisen yhdessä Unix-sovellusten kanssa. Lisäksi S2500-järjestelmä voi hyödyntää Mikkojen tehokkaita tiedonhallintajärjestelmiä ja SQL-tietokantoja kommunikoida suoraan Mikko-sovellusten kanssa. EAM on S2500:n ohjelmistotuote, joka luo älykäs- tä Ethernet-sovitinta hyödyntävän sovellusrajapinnan (API) S2500:n ja Mikkojen välille.

Mikkojen sovelluksia on mahdollista käyttää myös Tiimin ulkopuolisilta S2500-pääteiltä. 240- ja 290-työasemilla voidaan ottaa yhteys Mikkoon ja kirjoit- tautua käyttäjäksi aivan kuin työ- asema olisi suoraan liitetty Unix-järjestelmään (ELO, Ethernet Login). Käyttäjä voi siten hyödyntää Mikko-järjestelmiä suoraan omalta pääteeltään. Pääteyhteys on avoin (telnet-protokolla) ja mahdollistaa liitännän verkon kaikkiin Mikko-palvelimiin (multi host). Käyttäjä valitsee haluamansa järjestelmän käynnistäes- sään lähiverkon kautta kulkevaa pääteyhteyttä. Työasema voi olla liitettyä S2500:een myös etäisoh- jaimen (C-XC), kiinteän linjan, X.21- tai X.25-verkon kautta. ELO on S2500:n ohjelmistotuote, joka tarjoaa 240/290-työasemille mahdollisuuden kirjoittautua Mikko Unix-järjestelmien käyttä- jäksi.

S2500-järjestelmän käyttäjä tai sovellusohjelma voi myös siirtää tiedostoja S2500:sta Mikkoon ja Mikosta S2500:een. Tiedostot siirtyvät ethernet-verkkoa pitkin (EFT, Ethernet File Transfer). Kaikki verkon Mikot ovat käytet- tävissä sillä siirron yhteydessä

S2500-käyttäjä valitsee kohteena olevan Mikko-palvelimen. EFT sisältää myös toiminnot Mikon hakemistojen luontia, vaihtoa ja listausta varten. Sovellusohjel- mien suorittamat tiedostojen siir- rot voidaan hoitaa tausta-ajoina yöaikaan (BMS). Itse siirto ta- pahtuu siirtoprotokolla-standar- dia (FTP) noudattaen. EFT on S2500:n ohjelmistotuote, joka mahdollistaa binaarimuotoisten sarjatiedostojen siirron S2500:n ja Mikkojen välillä.

Mikkojen sovelluksia on mahdollista käyttää myös verkon ulkopuolisilta S2500-pääteiltä.

S2500-sovelluksiin myös Tiimin pääteiltä

Mikot voivat käyttää myös tavalli- sia päätteitä. Nämä päätteet voi- vat ottaa myös yhteyden S2500-järjestelmään (UNICON, UNIX Connection), kirjoittautua käyt- täjiksi ja hyödyntää S2500-so- velluksia suoraan omalta asci- i-pääteeltään (VT100). Käyttäjä voi vapaasti siirtyä S2500:n ja Mi- kon sovellusten välillä (shell es- cape).

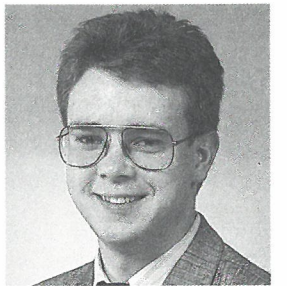
Pääte voidaan liittää Mikko- järjestelmään joko suoraan tai päätepalvelimen avulla. Näin nykyiset S2500-sovellukset ovat käy- tettävissä myös Tiimi-verkon kai- kille käyttäjille. UNICON on Mikkojen ohjelmistotuote, joka tarjoaa ascii-työasemille pääsyn S2500:n käyttöjäksi. UNICON si- sältää osoitejonon, jonka kautta useat käyttäjät jakavat käytettä- vissä olevia "S2500-linjoja".

S2500-järjestelmän liitântä Ti- imin työasemiin, työryhmän pal- velimiin ja Mikko-palvelimiin on monipuolinen ja tiivis. S2500-jär- jestelmän käyttäjän käynnistämä sovellus voi kommunikoida ver- kossa, käyttäjä voi itse siirtyä

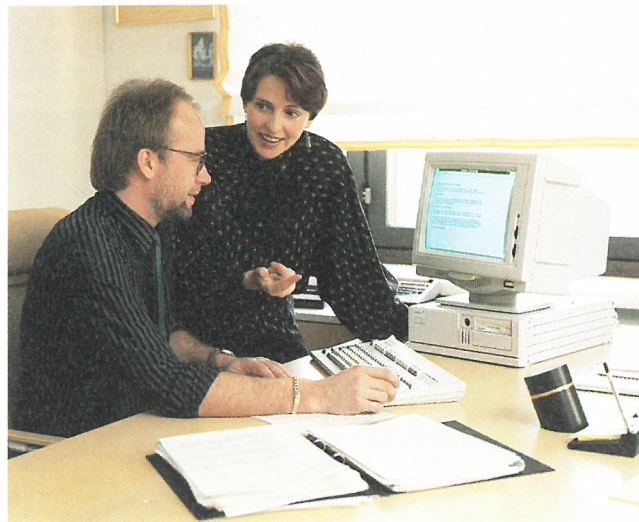
Unix-järjestelmän käyttäjäksi ja käynnistää siellä olevia sovelluk- sia tai tiedonsiirtoyhteyden. Mikko-järjestelmän käyttäjä voi vastaavasti siirtyä välillä käyttä- mään S2500-järjestelmän yritys- sovelluksia. Tiimin henkilökoh- taiset työasemat voivat käyttää sekä S2500-järjestelmää että Mikroja sovelluspalveliminaan, siirtää kummastakin tietoa työ- asemaan tai työryhmän palveli- meen tai antaa koko kommuni- koinnin sovellusohjelmien tehtä- väksi.

S2500, Tiimi ja Mikot mahdollisuus avoimuuteen

Tiimi ja Mikot antavat S2500-järjestelmän käyttäjälle mahdol- lisuuden joustavasti hyödyntää uusia ja avoimia järjestelmärat- kaisuja. S2500 yhdessä Tiimin henkilökohtaisten työasemien, OS/2-palvelimien ja Mikko Unix-järjestelmien kanssa on kauas- kantoista avoimuuden strategiaa. Tiimin tarjoamat uudet sovellus- alueet ratkaisevat monia uusia yritysten organisaation tarpeita, toteuttaen samalla kaikki nyky- järjestelmän tarjoamat palvelut.



Glen-Koskela työskentelee Nokia Datan Tuotemarkki- noinnissa S2500-tuotteiden tuotepäällikkönä.





DPS 6000 hajautuviin tietojärjestelmiin

Nokian DPS 6000 -minutietokoneen tarjoamat useat kymmenet laajat tuotanto-sovellukset antavat lyhyet vasteajat ja suuren tapahtumankäsittelytehon. Näitä sovelluksia täydennetään sekä työasemajärjestelmäliitäntöillä että työryhmäjärjestelmässä toimivilla sovelluksilla. Jatkossa Tiimi-työasemajärjestelmän ja DPS 6000:n sovellukset toimivat entistä enemmän rinnakkain täydentäen toisiaan. Sovellusten käyttäjät saavat täyden hyödyn henkilökohtaisen työaseman sovellusten, Tiimi-työryhmäsovellusten ja DPS 6000 -sovellusten yhdistelmästä.

Yksittäisistä liittymistä DPS 6000 Tiimiin

Jo vuosia sitten otettiin ensimmäiset askeleet mikrojen liittämiseksi DPS 6 -koneisiin. Pääte-emulointia ja Kermit-pohjaista tietojen siirtoa käytetään hyväksi silloin, kun DPS 6 -sovelluksia halutaan ajaa mikroilta käsin. DPS 6:n tietokannoista tiedot imaistaan mikroihin, joissa tiedot jatkojalostetaan henkilökohtaisilla työvälineillä.

Toisessa vaiheessa DPS 6 PLUSsalle julkistettiin ONEserver-ohjelmisto, jonka avulla DPS 6 PLUSsaa voidaan käyttää mikrojen tiedosto- ja tulostuspalvelimena. Tietoja ei tarvitse siirtää DPS 6 PLUSsan ja mikron välillä, vaan DPS 6 PLUSsan levytiloja voidaan käyttää suoraan mikron levyinä. Tietokantapöimintojen tiedot saadaan suoraan oman työaseman käytettäväksi LACS-lähiverkko-liitäntään läpi.

Nykyinen vaihe, DPS 6000 Tiimi, yhdistää Nokian Tiimi-työasemajärjestelmän ja DPS 6000:n

parhaat ominaisuudet. DPS 6000 Tiimi tarjoaa DPS 6000:n palvelut nokialaisen Tiimi-järjestelmän käyttäjille.

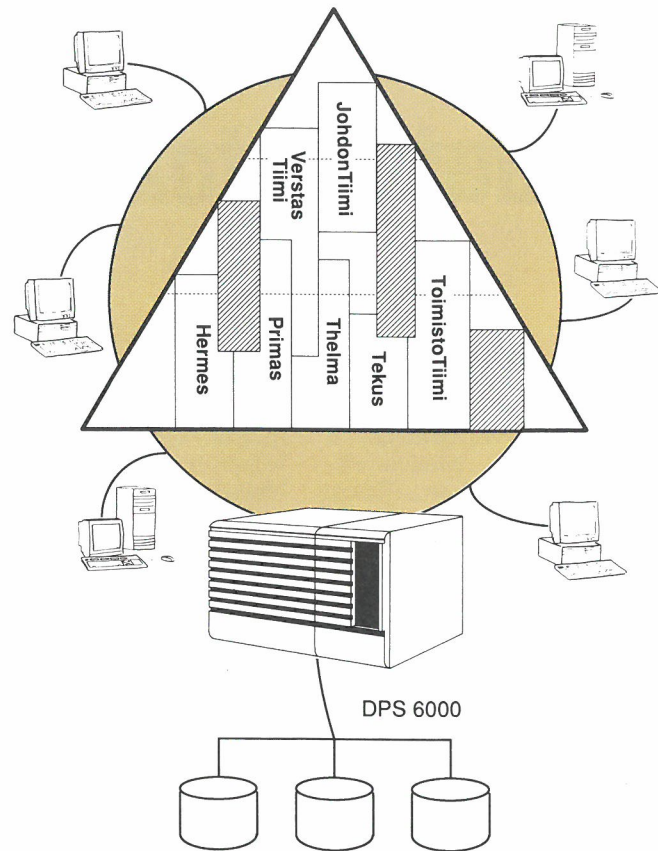
DPS 6000:n ja Tiimin sovellukset

DPS 6000:n tärkeimpiä ominaisuuksia ovat useat kymmenet valmiit, suomalaisen käyttöön ja suomalaisiin tarpeisiin tehdyt sovellukset. Lisäksi Nokia Datan ja useiden ohjelmistotalojen pitkäaikainen kokemus näistä sovelluksista takaa järjestelmien toimivuuden ja kehittymisen myös jatkossa. Käytettävissä on myös Toimi-toimistojärjestelmä sekä joukko erilaisia tietoliikenneohjelmistoja.

Tyypillistä DPS 6000-sovelluksille on niiden laajuus ja monipuolisuus, jotka vaativat nopeita vasteaikoja ja suurta tapahtumankäsittelytehoa useiden kymmenien käyttäjien tuotantoympäristössä. DPS 6000:n sovelluksissa käytetään operatiivisella tasolla usein myös yrityksen tietokantaa, jossa on erittäin paljon yksityiskohtaista tietoa.

Tämän päivän Tiimi-sovelluksille on ominaista se, että ne mahdollistavat taktisen tason suunnittelun ja ohjauksen. Tiimi-palvelimiin poimitaan usein ns. lattiason tietoja yhteenvetotietoa jostakin työasemajärjestelmän ulkopuolisesta järjestelmästä, kuten DPS 6000:sta. Esimerkiksi VerstaTiimi toimii teollisuuden toiminnanohjausjärjestelmän, Primaksen kanssa yhteistyössä.

Tiimi-sovellukset räätälöidään vastaamaan käyttäjien tarpeita. Tämän lisäksi niihin tehdään liittymät käyttäjän perusjärjestelmiin. Usein liittymiksi riittävät



Nykyinen laitteisto-arkkitehtuurin kehitys tukee tietojenkäsittelyn hajauttamista. Työ tehdään siellä, missä se on mielekkäintä ja tehokkainta. Tärkeää osaa hajautuksessa näyttävät työasemat, työryhmäjärjestelmät ja DPS 6000 Tiimi.

säännöllisin väliajoin tapahtuvat tietojen siirrot esimerkiksi GLINKin avulla, mutta joskus tarvitaan myös suoria sovellusliittymiä työasemalta DPS 6000:een. Tiimi-työasemaverkon avulla samaan työasemaan saadaan yksittäismikron, verkkotyöaseman ja pääteemulointi. Ympäristö on aina tuttu. Käyttäjälle voidaan toteuttaa sovellusympäristö, jossa hänen ei tarvitse tietää missä kulloinkin sovellus ja tieto sijaitsevat. Yhdenmukaisen käyttöliittymän kautta näyttää siltä kuin sovellus ja tietoa olisivat osa käyttäjän henkilökohtaista tietojärjestelmää.

DPS 6000 Tiimin tarjoamat palvelut

Nokia DPS 6000 Tiimi tarjoaa Tiimi-työasemajärjestelmän käyttäjälle DPS 6000:n sovelluspalveluiden lisäksi sovellusliittymän mikroverkossa toimivan sovelluksen ja DPS 6000:n sovelluksen välillä, DPS 6000:n monipuoliset tietoliikenneominaisuudet, toimistojärjestelmäpalvelut sekä mikroille tarkoitettut tiedosto- ja tulostuspalvelut.

Ensinnäkin DPS 6000 Tiimi tarjoaa työasemajärjestelmän käyttäjille DPS 6000:n nopeat sovelluspalvelut. Tiimin työasemal-

ta voidaan ajaa kaikkia DPS 6000-sovelluksia, kuten Primasta, Hermestä, Tekusta ja Thelmaa. Palvelut hoidetaan pääte-emuloinnilla.

Toiseksi DPS 6000 Tiimi tarjoaa sovellusliittymän työasemajärjestelmässä toimivan sovelluksen ja DPS 6000:n välille. Tällöin työaseman sovelluksen käytettävissä on sekä Tiimi-sovelluksen että DPS 6000:n tietokantojen tiedot.

DPS 6000:n ohjelmaliittymän avulla Tiimi-sovellus voi poimia tietoja TPS 6 -tietokannoista toisaikaisesti lähettämällä työasemasovelluksesta SQL-poimintakäskyn DPS 6000:lle. DPS 6000 tulkitsee käskyn, poimii tietokannasta halutut tiedot ja lähettää ne takaisin työaseman sovellukselle.

Tietoliikennettä tarvitaan jo nykyään runsaasti, mutta verkkojen lisääntyminen ja kansainvälistyminen lisää tarvetta tulevaisuudessa huomattavasti. DPS 6000 Tiimi antaa Tiimin käyttäjälle mahdollisuuden käyttää omalta työasemaltaan DPS 6000:n tarjoamia tietoliikennepalveluja. DPS 6000 pystyy hoitamaan mm. kansainväliset tietopankkiyhteydet esimerkiksi Quik-Comm MARK III:een, telex-yhteydet, X.400-sanomanvälityksen sekä tiedostojen siirrot eri koneiden välillä niin OSI:n FTAMia kuin SNA:n RJE:tä käyttäen. Tieto liikkuu tietysti myös eri DPS-koneiden välillä DSA-verkon palvelujen kautta.

DPS 6000 Tiimi yhdistää myös Toimin ja ToimistoTiimin toimituspalvelut. Työasemajärjestelmässä käytetään usein omia ohjelmistoja tekstinkäsittelyyn, esitysgrafiikkaan ja taulukkolaskentaan (ToimistoTiimin Perfect- ja Windows-tuoteperheet). DPS 6000:n levytiloja voidaan käyttää työasemajärjestelmien asiakirjojen arkistona (ONELibrary). Tarvittaessa DPS 6000:ssa voidaan hoitaa myös asiakirjojen siirtäminen eri järjestelmästä toiseen. ONExchange-ohjelmiston avulla on mahdollista konvertoida tietoja noin 30 eri ohjelmiston, kuten ToimistoTiimin WordPerfectin ja MS-Wordin sekä Toimin välillä.

Lopuksi DPS 6000 pystyy toimimaan yksittäisen mikron tiedosto- ja tulostuspalvelimena (ONEserver). Työaseman tietoja voidaan tallentaa DPS 6000:n levyille yhtä helposti kuin käyttäjän oman työaseman massamuistiin. Tietojen välittäminen työasemalta DPS 6000:lle ja päin vastoin hoidetaan tiedostopalveluja käyttäen. Sekä työaseman että DPS 6000:n on oltava suoraan liitettyinä Ethernet-verkkoon.

Tiimin liittäminen DPS 6000:een

Tiimi-työasemajärjestelmä liitetään DPS 6000:een omalla yhdyskäytävällä. Yhdyskäytävänä toimii Tiimi-työasemajärjestelmään liitetty mikro (MikroMikko 3 tai 4), joka pystyy huolehtimaan 16:sta samanaikaisesta istunnosta työaseman ja DPS 6000:n välillä. Istunto voi olla tavallinen pääteistunto tai ohjelmaliittymä. Tiimi-työasemaverkosta voi olla auki useita yhdyskäytäviä joko samaan tai useaan eri DPS 6000:een.

Yhdyskäytävässä huolehditaan tietojen siirtämisestä työasemaverkon NetBIOS-tasolta DPS 6000:n OSI-protokollaan. Yhdyskäytävään ladataan verkko-ohjelmiston lisäksi tietojen välityksen hoitava ohjelmisto. Yhdyskäytävän välitysohjelmisto on rakennettu NetBIOSin päälle ja on tarkoitettu käytettäväksi ensisijaisesti Nokia MS Lan Manager-pohjaisissa Ethernet-verkoissa. NetBIOS-liitäntä mahdollistaa DPS 6000:n liittämisen myös useisiin erilaisiin verkkoihin.

Yhdyskäytävänä toimii Tiimi-työasemajärjestelmään liitetty mikro, joka pystyy huolehtimaan 16:sta samanaikaisesta istunnosta työaseman ja DPS 6000:n välillä.

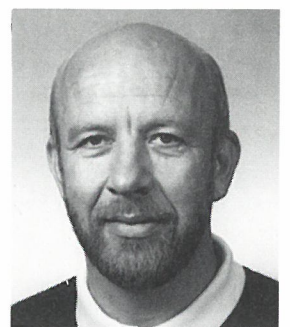
Yhdyskäytävä käyttää MS-DOS-käyttöjärjestelmää. Lisäksi tarvitaan kaksi verkkokorttia, joista toinen liitetään Ethernetin avulla DPS 6000:n LACS-lähiverkko-ohjaimeen ja toinen mikroverkkoon. DPS 6000 käyttää tällä hetkellä Etherlink (3C501)-korttia, mutta myöhemmin on voidaan hyödyntää myös Etherlink II (3C503)-kortteja.

Verkon DOS-työasemissa käytetään pääte-emulointiin ja tietojen siirtoon GLINK-ohjelmistoa, joka keskustelee suoraan NetBIOS-tasolla. GLINKiä voidaan ajaa myös Windowsin alla.

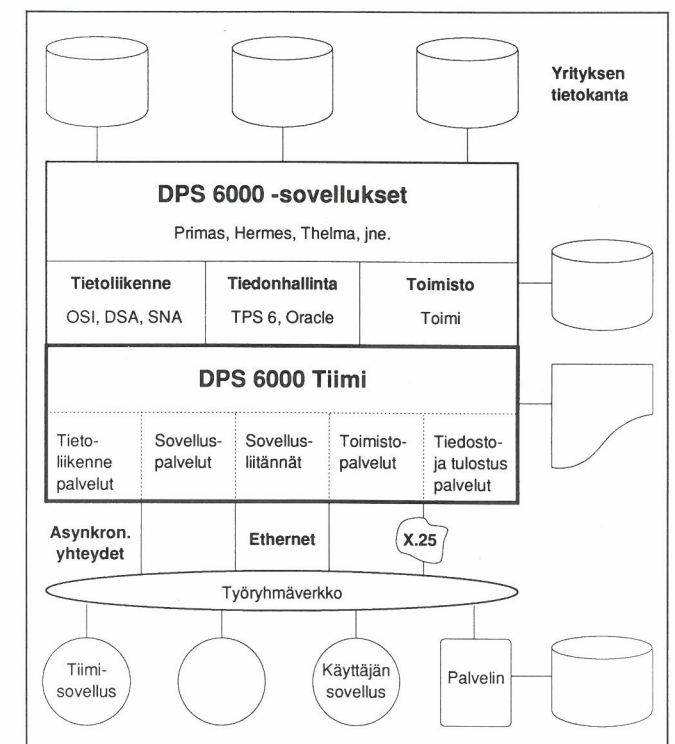
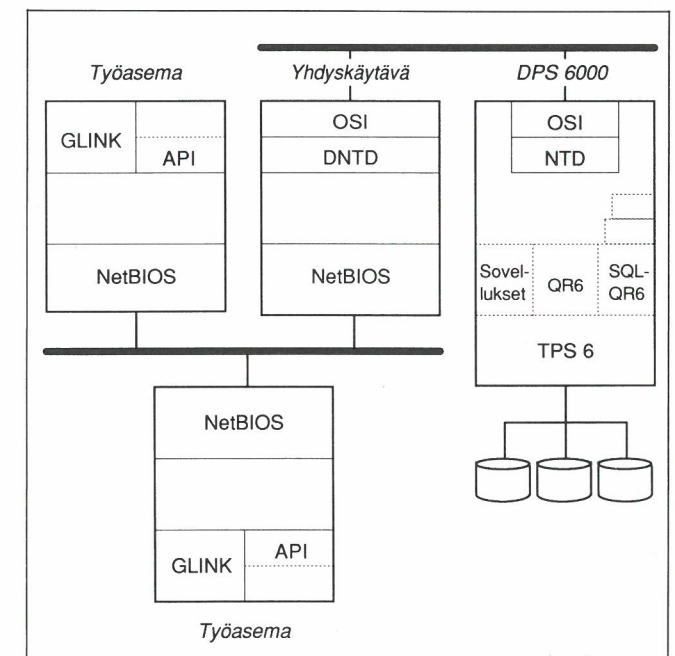
Tiimi-työasemaverkon voi liittää DPS 6000:een myös tahdistamattomilla yhteyksillä käyttämällä Nokian AACS-ohjelmistoa (Softari 1/89).

Hajauttaminen etenee edelleen

Tätä nykyä järjestelmät hajauteaan toimimaan useissa keskenään kommunikoivissa laitteissa. Nokia DPS 6000 Tiimin avulla voidaan DPS 6000-sovelluksia kehittää edelleen siten, että sovellusten suoritus tapahtuu siellä, missä se on mielekkäintä ja tehokkainta; osa sovelluksista toimii omassa työasemassaan, osa Tiimi-työryhmäverkon palvelimissa ja osa uudistuneessa Nokia DPS 6000:ssa.



Risto Taskula työskentelee Nokia Datan järjestelmätuotteet-osastolla.



Enemmän irti Windows-sovelluksista

**Softarin numeros-
sa 4/88 Tero Koski-
virta mainitsi, että
tulevaisuudessa
kunnollinen DDE-
tuki on yhä merkit-
tävämpi sovelluk-
sen mainosvaltti,
kun kamppaillaan
markkinaosuuk-
sista.**

Samassa mikrossa toimiviin Windows-sovelluksiin voi rakentaa Dynamic Data Exchange -protokollan avulla tiedonsiirtoyhteyksiä, joilla hyvinkin erilaiset sovellukset saadaan toimimaan yhteistyössä. Jo perinteiseksi käynyt esimerkki DDE:n käyttöalueesta on tietoliikennesovellus, joka "kuuntelee" linjalta tulevia pörssikurssien muutoksia ja välittää tiedon edelleen DDE:n avulla toiselle Windows-sovellukselle, joka näyttää tuoreet tiedot graafisen esityksen muotoksina.

Softarin numnerossa 4/88 Tero Koskivirta esitteli DDE-tekniikkaa yleisesti. Tässä artikkelissa selvitetään DDE-tekniikan toteuttamista käytännössä nimenomaan SQL-Windows-sovellukseen. Tätä tietoa voi käyttää hyödykseen myös, jos suunnittelee DDE:n käyttöä Windows SDK, OS/2 Presentation Manager tai esim. Excel-ympäristössä. Artikkelit on tarkoitettu lähinnä ohjelmoijille, jotka jo tuntevat Windows- tai SQL-Windows-ohjelmoinnin perusteet.

Tiedon siirtämisen yleiset periaatteet

Windows-sovellusten välillä voidaan siirtää tietoa leiketaulun avulla. Eräisiin Windows-sovelluksiin on myös rakennettu dynaaminen, tapahtumapohjainen tiedonsiirtotapa (DDE), joka voi käynnistyä ja toimia myös ilman käyttäjän toimenpiteitä, jos niin halutaan.

DDE-tekniikassa sovellusten välillä tieto siirtyy ennalta määritettyjen DDE-sanomien avulla. Sanomien parametrina voi olla esimerkiksi globaalin muistiolion osoite (kahva), jonka avulla varsinaisesta siirrettävästä tiedosta saadaan ote. Microsoft on määritellyt DDE-sanomat, niiden parametrirakenteet ja sen, millä tavoin sovellusten tulee vastata saamiinsa DDE-sanomiin. Näitä määritelmiä kutsutaan protokollaksi nimeltä Dynamic Data Exchange (DDE).

Sen jälkeen, kun kaksi sovellusta on muodostanut DDE-yhteyden välilleen lähettämällä tietyt DDE-protokollan mukaiset sanomat, yhteyden käynnistää kut-sutaan isännäksi (client) ja vas-

Atomi tarkoittaa käytännössä sitä, että Windows varaa jollekin merkkijonolle tilaa atomi-taulukosta, mutta merkkijonon asemesta käytetäänkin sen sijaintipaikan ilmaisevaa kokonaislukuarvoa, atomia.

taajaa rengiksi (server). Sovellusten tulee siis olla käynnissä yhtä aikaa ja tietoisia toisistaan. Tosin saattaa olla, etteivät ne ole kiinnostuneita siitä, mikä sovellus vastaa, vaan vain siitä, mitä tietoa saadaan.

DDE-tekniikan avulla voidaan toteuttaa kolmentyyppistä tiedonsiirtoa: kertaluonteinen tiedonsiirto, jatkuva tiedonsiirtoyhteys ja komentojen suorituttaminen rengillä. Kuvissa 1, 2 ja 3 on esitetty kaavioina näihin kolmeen eri tapaan tarvittavat sanomavälitykset.

Kuvista huomaamme, että kaikki DDE-sanomatunnukset alkavat merkinnällä WM_DDE_. Lisäksi kaikkien näiden kolmen tavan aloitus- ja lopetusvaiheet ovat samanlaiset. Aloitusvaiheessa isäntä lähettää tietyn nimiselle sovellukselle tai kaikille WM_DDE_Initiate-sanoman kysyäkseen: "Olisiko jollakulla tätä tietoa?". Tähän renki mahdollisesti vastaa WM_DDE_Ack-sanomalla eli "Minulla on pyytämäsi tieto!". Myös yhteyden lopetus on kaikissa samanlainen. Siinä joko isäntä tai renki ilmaisee halunsa lopettaa keskustelun lähettämällä WM_DDE_Terminate-sanoman, johon toinen vastaa samalla sanomalla. Tämä onkin ainoa sanoma, jonka voi ensimmäiseksi lähettää joko isäntä tai renki.

Seuraavaksi tarkastellaan kuvan 1 mukaisen kertaluonteisen tiedonsiirron toteuttamista SQL-Windows-sovelluksessa vaikkapa tilanteessa, jossa juuri käynnistetty isäntäsovellus haluaa rengiltä käyttäjän tunnistetiedon.

Toisen sovelluksen käynnistäminen

Kun isäntäsovellus haluaa renki-sovellukselta palveluksia, se pyrkii aloittamaan keskustelun lä-

hettämällä kuvan 1 mukaisesti WM_DDE_Initiate-sanoman. Tämän jälkeen rengin tulisi vastata isännälle WM_DDE_Ack-sanomalla. Jos näin ei kuitenkaan tapahdu, kyseinen renki ei ilmeisesti ole hengissä ja se pitää elvyttää.

Isäntäsovellus voi pyytää käyttäjää käynnistämään tietyn sovelluksen, mutta isäntäsovellus voi käynnistää rengin itsekin. SQL-Windows-sovelluksessa tämä käy SalLoadApp-funktion avulla esimerkiksi seuraavaan tapaan:

```
Call SalLoadApp
('SQLRUN.EXE','SERVER.RUN')
```

Tällä kutsulla käynnistetään SQLWindowsin ajonaikainen suoritinohjelma SQLRUN.EXE ja annetaan sille parametrina suoritettava ohjelma SERVER.RUN.

Näin molemmat Windows-sovellukset on saatu käyntiin, ja keskustelu voi alkaa. Isäntäsovelluksen pääikkuna on nähtävissä kuvassa 4 ja renki-sovelluksen ikkuna kuvassa 5. Seuraavaksi käydään läpi joitakin peruskäsitteitä kuvan 6 ohjelmakoodin avulla. Siitä nähdään, millaista SQLWindowsin SAL-kielistä koodia tarvitaan keskustelun aloittamiseksi.

Tunnistetietoja ja parametritavuja

DDE-sanomissa ja -funktio-kutsuissa esiintyy kolme tärkeää tunnistetietoa: sovellus (application), aihepiiri (topic) ja alkio (item). Näitä voitaisiin verrata vaikka palaverikutsuun, joka kohdistetaan tietyille tai kaikille (application) ja jossa näkyy aihepiiri (topic), jota palaveri koskee. Jokainen palaverista kiinnostunut voisi sitten kohderyhmän ja aihepiirin perusteella päätellä, haluaako liittyä keskusteluun vai ei.

DDE-keskustelun toteuttaminen SQLWindows-sovellukseen

Nämä tiedot esiintyvät miltei kaikissa DDE-sanomissa. Sen lisäksi varsinainen tieto (esim. palaverissa asiasta jaettu monistennippu) välitetään globaalin muistialueen kautta.

Esimerkkisovelluksissa sovellus-tietona on merkkijono 'Softserv', aihepiirinä 'Käyttäjä' ja alkiona 'fldstrUser', joka on sen kentän nimi isäntäsovelluksessa, johon rengiltä haetaan uusi arvo. Kun yhteys otetaan, etsitään sovellusta 'Softserv' ja aihepiiriä 'Käyttäjä'. Yhteyden luomisen jälkeen vastapuoli tunnistetaan ikkunan kahvan, eikä enää sovelluksen nimen, perusteella. Tämän jälkeen kustakin tiedosta lähetetään oma WM_DDE_DATA-sanomansa, jossa ilmaistaan, mistä alkioista (item) on kysymys. Itse tieto välitetään WM_DDE_DATA-sanoman mukana globaalissa muistialueessa.

Hyvän Windows-tavan mukaisesti sovelluksessamme tulisi olla myös System-aihepiiri, mutta yksinkertaisuuden vuoksi se on jätetty pois esimerkiksi.

Yllämainitut tunnistetiedot välitetään vastaanottajalle sanoman mukana. Koska näille sanoman mukana lähetettävälle parametrille on varattu vain 2+4 tavua tilaa (wParam ja lParam), näitä tietoja ei välitetäkään merkkimuotoisina, vaan ns. atomeina. Atomia tarkoittaa käytännössä sitä, että Windows varaa jollekin merkkijonolle tilaa atomi-taulukosta, mutta merkkijonon asemesta käytetäänkin sen sijaintipaikan ilmaisevaa kokonaislukuarvoa, atomia. Windows tarjoaa funktioita, joiden avulla atomitauluun voidaan lisätä ja poistaa merkkijonoja. Samalla se tarkistaa, ettei päällekkäisyyttä ilmene. Jos merkkijono on jo atomitaulukossa, Windows palauttaa saman atomiarvon, eikä turhaan varaa tilaa kahdelle samantyyppiselle merkkijonolle.

Isäntä ottaa yhteyden renkiin

Esimerkkisovelluksessamme yhteydenotto tapahtuu isäntäsovelluksesta, kun käyttäjä valitsee päävalikosta vaihtoehdon Ota yh-

teys!. Yhteydenotto voi tietenkin tapahtua myös automaattisesti.

Yhteydenotossa isäntäsovellus lähettää rengille WM_DDE_Initiate-sanoman funktion SalDDESendAll avulla kuvan 6 tapaan. Ennen kyseisen funktion kutsua sanoman mukana lähtevät sovellus- ja teematiedot on viety atomitauluun, jotta niitä vastaavat atomienumerot voitaisiin välittää sanoman parametritiedoissa. Alkiotietokin (item) on muunneltu valmiiksi atomiksi. Sen avulla isäntä pystyy myöhemmin tarkistamaan, koskeeko rengiltä tuleva tieto isäntää kiinnostavaa alkioita.

Kuvassa 7 on esitetty rengin vastaanottaman WM_DDE_Initiate-sanoman sisältämät tiedot. SQLWindows ja Windows rakentavat sen isäntäsovelluksessa suoritettuna funktiokutsun SalDDESendAll perusteella.

On tärkeää huomata, että sanomien lähettämiseen käytetään kahta erilaista funktiotyyppiä: post- ja send-funktioita. Post-funktioilla lähetetyt sanomat menevät vastaanottajaikkunan sanomajonoon ja tulevat käsiteltyiksi järjestyksessä, kunhan niiden aika tulee. Send-funktioilla lähetetyt sanomat taas eivät menekään tuohon jonoon, vaan Windows kutsuu vastaanottajaikkunan ikkunaproseduuria ikään kuin aliohjelmana, jolloin send-sanoma käsitellään välittömästi ennen jonossa olevia sanomia.

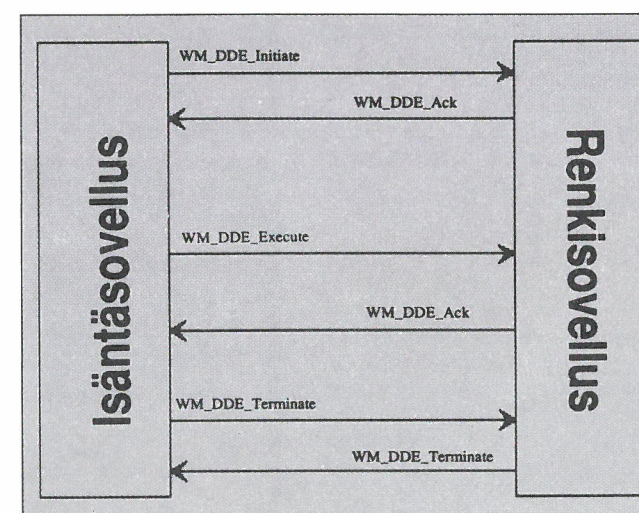
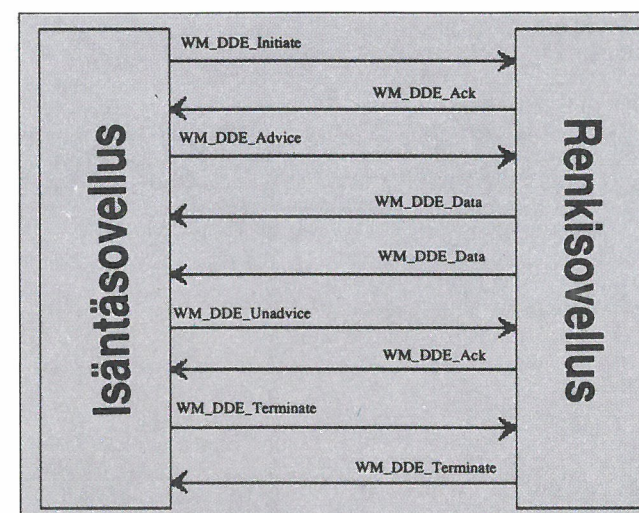
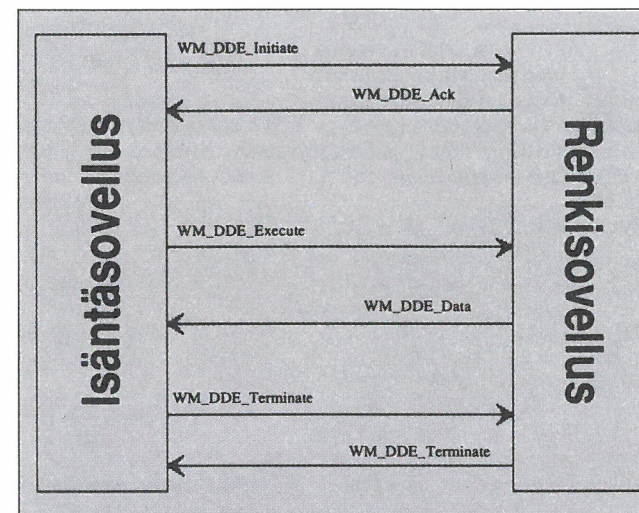
Yhteydenottoon liittyvät sanomat lähetetään siis aina send-funktioilla, kun taas kaikki muut DDE-sanomat lähetetään post-funktioilla. Näin yhteydenottovaiheessa ei sovellus juutu odottamaan toisen vastausta turhaan.

Kuvassa 6 näkyvässä isäntäsovelluksen yhteydenottokoodissa voidaan siis heti funktion SalDDESendAll jälkeen testata muuttujan bInitialized arvoa. Kontrolli on funktiokutsun vaihtokutsusta välillä käynyt Windowsilla sekä mahdollisella renki-sovelluksella, joka on lähettänyt WM_DDE_Ack-sanoman isäntäsovellukselle. Isäntäsovelluksen Form Windown Message Actions -kohdassa WM_DDE_Ack on käsitelty ja parhaassa tapauksessa muuttuja bInitialized saanut arvon TRUE. Sitä voidaan siis testata funktiosta SalDDESendAll palattaessa. Alunperin

Kuva 1: Kertaluonteisessa tiedonsiirrossa tarvittavat sanomat.

Kuva 2: Jatkuva tiedonsiirtoyhteydessä tarvittavat sanomat.

Kuva 3: Komennon suorittamisessa tarvittavat sanomat.



FK Tiina Mikonmäki toimii Tietospesialistit Oy:ssä vastaavana asiantuntijana. Hänen erikoisosamistansa on SQLWindows-, Windows- ja OS/2 Presentation Manager -ohjelmointi sekä atk-ammattilaisten koulutus. Hänet tavoittaa puhelinnumerosta 915-49 388 tai mailitse Mikonmäki Tiina frl.

isännän lähettämään WM_DDE_Initiate-sanomaan tarvittu sovellus- ja teema-atomitkin voidaan poistaa, sillä niitä vastavat tiedot on jo talletettu globaaleihin muuttujiin WM_DDE_Ack-sanoman käsittelyn yhteydessä.

Kun isäntäsovellus on lähettänyt WM_DDE_Initiate-sanoman SalDDESendAll-funktiolla, se jää odottamaan serverin WM_DDE_Ack-sanomaa, kuten kuvasta 1 näkyy.

Renkisovellus on valmistautunut vastaanottamaan WM_DDE_Initiate-sanoman, sillä sen Form Window Message Actions -kohtaan on koodattu vaihtoehto On WM_DDE_Initiate. Sitä voidaan tarkastella kuvassa 6 näkyvän koodin avulla.

Renki käyttää funktiota SalDDEExtract, joka purkaa sanoman mukana tulleet parametrit muuttujien hWndClient, aReceivedApplication ja aReceivedTopic arvoiksi. Tämän jälkeen renkin tulisi tarkistaa, että pyydetty sovellus- ja teematiedot ovat renkille sopivia. Jos näin on, renki muodostaa WM_DDE_Ack-sanoman vaatimat atomit ja lähettää kyseisen sanoman isännälle ilmoittaen, että on valmis yhteistyöhön.

Isäntä puolestaan vastaanottaa WM_DDE_Ack-sanoman, purkaa sen parametrit omiin muuttujiinsa ja muuttaa vielä sovellus- ja teema-atomit merkkijonoiksi. Sen jälkeen isäntä tutkii, ovatko nämä kiinnostavia, ja jos ovat, on valmis yhteistyöhön. Esimerkkiohjelmassa tulisi käyttää sisäkkäisiä if-lauseita, jotta oltaisiin valmistautuneita myös sellaisiin tilanteisiin, joissa kaikki ei menekään hyvin.

Nämä vaiheet käydään siis aina läpi, kun DDE-yhteyttä muodostetaan. Tämän jälkeen voidaan sitten valita kuvien 1-3 esittämistä kolmesta eri yhteydenpitotavasta. Käymme seuraavassa läpi niistä ensimmäisen, kertaluonteisen tiedonsiirron.

Renkisovellus vastaa isännältä saamaansa WM_DDE_Request-sanomaan lähettämällä isännälle pyydetyn tiedon WM_DDE_Data-sanoman parametrina.

Kertaluonteisen tiedonsiirron toteutus

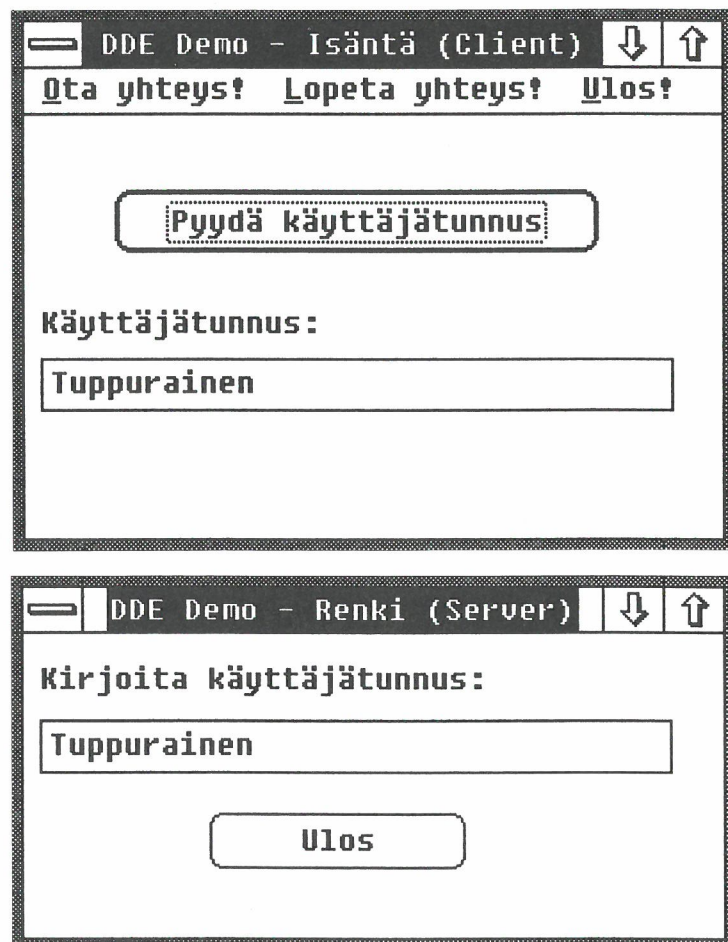
Tässä vaiheessa käyttäjä on isäntäsovelluksen Ota yhteys! -valikon avulla saanut luotua yhteyden isäntä- ja renkisovelluksen välille. Nyt hänen tulisi kirjoittaa kuvassa 5 näkyvän renkisovelluksen käyttäjätunnus-kenttään oma nimensä ja sen jälkeen painaa kuvassa 4 näkyvän isäntäsovelluksen Pyydä käyttäjätunnus -painonappia.

Kyseisen painonapin koodi näkyy kuvassa 8. Siinä isäntäsovellus lähettää renkille WM_DDE_Request-sanoman, jolla isäntä pyytää renkiltä CF_Text-tyyppistä tietoa, joka koskee nimenomaan altem-atomin ilmoittamaa alkia (esimerkissä 'fldstrUser').

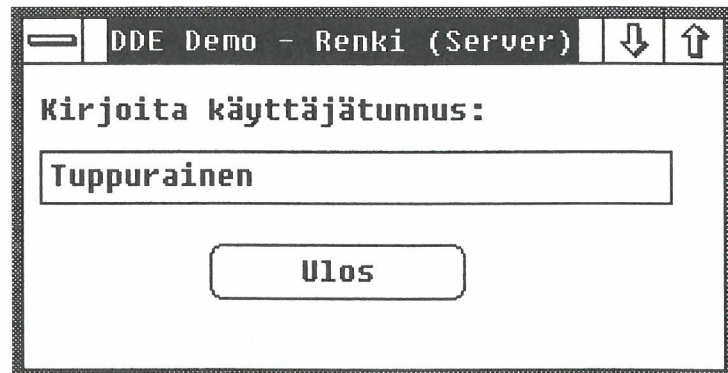
Renkisovellus vastaa isännältä saamaansa WM_DDE_REQUEST-sanomaan lähettämällä isännälle pyydetyn tiedon WM_DDE_Data-sanoman parametrina.

WM_DDE_Data-sanoman rakenne näkyy kuvassa 9. Siinä lParam-parametrin HIWORD-osassa on alkion atomi, jotta isäntä tietäisi, mitä tieto koskee. lParamin LOWORD-osassa on globaalin muistialueen kahva. Tämä globaali muistialue varataan funktiolla SalDDEAlloc, joka palauttaa arvonaan varatun muistialueen kahvan.

Tähän globaaliin muistialueeseen talletetaan paitsi varsinainen välitettävä tieto, myös muutamia lisätietoja. Nämä lisätiedot varaa muistialueen alusta 2 sanaa käyttöönsä. Ensimmäisessä sanassa ovat bitit DDE_fRelease



Kuva 4: Isäntäsovelluksen ikkuna.



Kuva 5: Renkisovelluksen ikkuna.

(13, tuleeko isännän vapauttaa globaali muistialue) ja DDE_fAck (15, jääkö renki odottamaan isännän kuittauksia). Toisessa lisätietosanassa ilmoitetaan välitettävän tiedon esitysmuoto (CF_Text).

Isäntäsovellus kannattaa ra-

kentaa siten, että se pyytää ensin renkiltä tietoa itselleen sopivimmassa muodossa, ja jos renki vastaa negatiivisella WM_DDE_Ack-sanomalla, isäntä laskee vaatimuksiaan ja pyytää seuraavaksi sopivammassa muodossa. Tiedon esitysmuoto voi myös olla so-

velluksen itse määrittelemä, mutta tällöin esitysmuoto tulee ensin rekisteröidä käyttäen Windows SDK:n funktiota RegisterClipboardFormat.

Kun isäntäsovellus vastaanottaa renkiltä WM_DDE_Data-sanoman, se tutkii ensin, onko al-

Kuva 6: DDE-keskustelun aloitukseen tarvittava koodi.

Isäntäsovellus:

Application Description: SOFTCLIE: Client application.

Design-time Settings

Global Declarations

External Functions

Constants

System

User

String: DDE_Application = 'Softserv'

String: DDE_Topic = 'Käyttäjä'

String: CLIENTNAME = 'SOFTCLIE.APP'

String: SERVERNAME = 'SOFTSERV.RUN'

Variables

Form Window: Client Form

Title: DDE Demo - Isäntä (Client)

Display Settings

Menu

Menu Item: &Ota yhteys!

Keyboard Accelerator: (none)

Menu Settings

Actions

If NOT SalFileOpen(fhServer, SERVERNAME, OF_Read)
Call SalMessageBox('Ei lydy: ' || SERVERNAME, 'VIRHE!', MB_Ok)

Else

Call SalFileClose(fhServer)

! Etsi server / topic / kentätieto

Set aApplication = SalDDEAddAtom(DDE_Application)

Set aTopic = SalDDEAddAtom(DDE_Topic)

Set altem = SalDDEAddAtom('fldstrUser')

Set blninitialized = FALSE

Call SalDDESendAll(WM_DDE_Initiate, hWndForm, aApplication, aTopic)

! Edellisessä funktiokutsussa kontrolli siirtyy Windowsille, joka kutsuu kaikkia Windows-sovelluksia. Niistä joku/jotkut saattavat lähettää WM_DDE-ACK -sanoman tälle sovellukselle. Sen käsittelyssä Form Window:n Message Actions -kohdassa blninitialized-muuttujan arvoa saatetaan muuttaa.

!

If not blninitialized

If SalMessageBox('Onko SOFTSERV.RUN käynnistetty?', 'DDE DEMO: MB_YesNo | MB_DefButton2) = IDNO
Call SalLoadApp('SQLRUN.EXE', SERVERNAME)

Else

Call SalMessageBox('Käynnistä SOFTSERV Run-valikosta User moodiin!...', 'DDE DEMO, MB_Ok)

Else

Call SalEnableWindow(hWndPBRequest)

! Appl / topic -atomit voi poistaa!

Call SalDDEDeleteAtom(aApplication)

Call SalDDEDeleteAtom(aTopic)

Contents

Variables

Message Actions

On WM_DDE_Ack

If blninitialized = TRUE

! Tässä vastaanotettaisiin REQUESTille negatiivista ACK-sanomaa ja toimittaisiin sen mukaisesti...

Else

! Tulossa ACK-vastaus lähettämäämme INITIATE-sanomaan

! Tarkistetaan, ovatko sovellus ja teema pyydyt

Call SalDDEExtract(wParam, lParam, hWndServer, aReceivedApplication, aReceivedTopic)

Call SalDDEGetAtomName(aReceivedApplication, strApplication, 40)

If strApplication != DDE_Application

! Ei ollut meille tarkoitettu DDE-sanoma

Call SalDDEGetAtomName(aReceivedTopic, strTopic, 40)

If strTopic != DDE_Topic

! Ei ollut meitä kiinnostava teema

Call SalDDEDeleteAtom(aReceivedApplication)

Call SalDDEDeleteAtom(aReceivedTopic)

Set blninitialized = TRUE

Renkisovellus:

Application Description: SOFTSERV: DDE-demon server-ohjelma.

Design-time Settings

Global Declarations

Constants

System

User

String: DDEServer_Application = 'Softserv'

String: DDEServer_Topic = 'Käyttäjä'

Variables

Form Window: ServerForm

Title: DDE Demo - Renki (Server)

Display Settings

Menu

Contents

Variables

Message Actions

On WM_DDE_Initiate

Call SalDDEExtract(wParam, lParam, hWndClient, aReceivedApplication, aReceivedTopic)

! Tässä tarkistettaisiin INITIATE:n mukana tulleet sovellus- ja teematiedot.

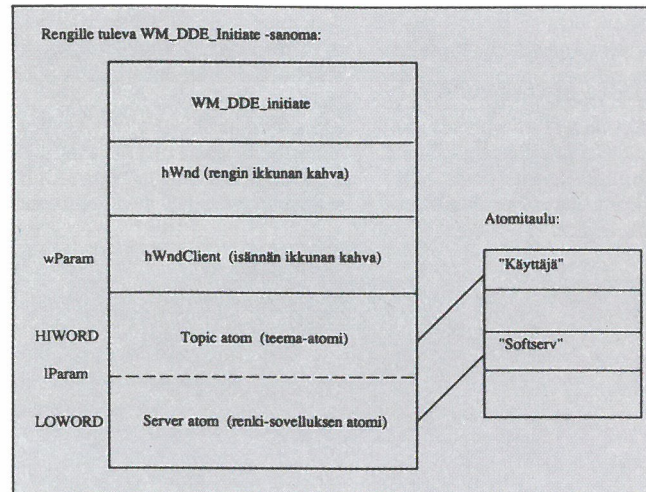
! Lisätään atomit taas ennen ACK-sanomaa.

Set aApplication = SalDDEAddAtom(DDEServer_Application)

Set aTopic = SalDDEAddAtom(DDEServer_Topic)

Call SalDDESend(hWndClient, WM_DDE_Ack, hWndForm, aApplication, aTopic)

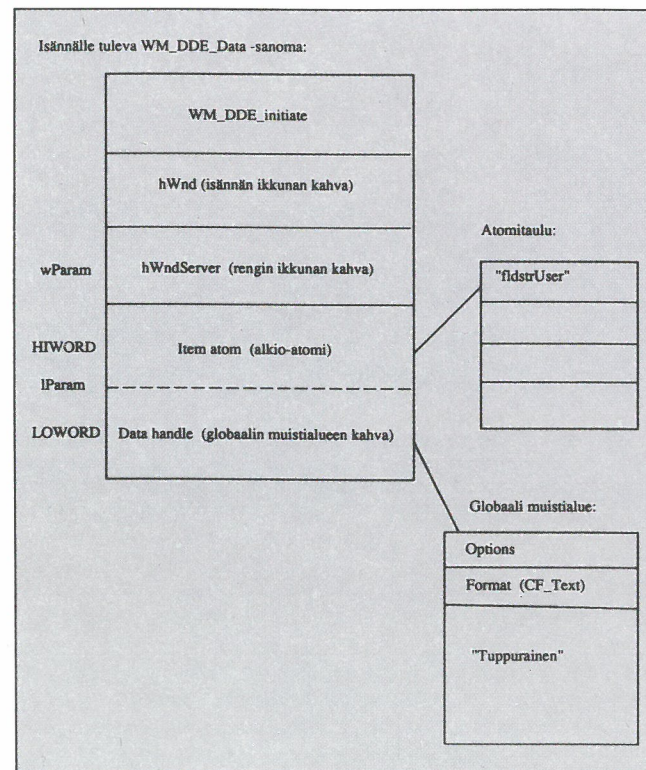
DDE



Kuva 7: WM_DDE_Initiate -sanoman rakenne.

Kuva 8: Tiedon pyyntö rengiltä ja tiedon lähettäminen isännälle.

Kuva 9: WM_DDE_Data -sanoman rakenne.



Isäntäsovellus:

```
Form Window: ClientForm
Title: DDE Demo - Isäntä (Client)
Display Settings
Menu
Contents
  Data Field: fidstrUser
  Data
  Display Settings
  Message Actions
Background Text: Käyttäjätunnus:
Pushbutton: pbRequest
Title: Pyydä käyttäjätunnus
Window Location and Size
Keyboard Accelerator: (none)
Message Actions
  On SAM_Click
    ! Pyydä käyttäjätunnus serveriltä
    Call SalDDEPost( hWndServer, WM_DDE_Request,
      hWndForm, CF_Text, altem )
  On SAM_Create
    Set hWndPBRequest = hWndItem
    Call SalDisableWindow( hWndPBRequest )
Variables
Message Actions
  On WM_DDE_Data
    Call SalDDEExtract( wParam, lParam, hWndServer, nData,
      aReceivedItem )
    ! Tässä pitäisi tarkistaa, että tullut aReceivedItem on sama kuin
    pyydettiin (ks. ACK-sanoman käsittely)
    If not SalDDEExtractDataText( nData, nStatus, strData, 500 )
      Call SalMessageBox( 'Ei pysty erottamaan dataa sanomasta',
        DDE Client', MB_Ok )
    If not nStatus & DDE_fResponse
      ! Tämä ei ollutkaan vastaus lähettämääme REQUESTiin!
    If nStatus & DDE_fAck
      ! Tässä lähetettäisiin WM_DDE_Ack-sanoma serverille, joka on
      ! pyytänyt kuittausta asettamalla DDE_fAck-lipun statukseen.
      ! Meidän esimerkissämme server ei ole pyytänyt kuittausta.
    Else
      ! Server ei odota kuittausta, poista aReceivedItem-atomia
      Call SalDDEDeleteAtom( aReceivedItem )
    If nStatus & DDE_fRelease
      Call SalDDEFree( nData )
      Set fidstrUser = strData
```

Renkisovellus:

```
Form Window: ServerForm
Title: DDE Demo - Renki (Server)
Display Settings
Menu
Contents
Variables
Message Actions
  On WM_DDE_Request
    Call SalDDEExtract( wParam, lParam, hWndClient, nFormat,
      altem )
    ! Tarkista, että altem vastaa sovitua teemaa (ks. isäntä-sovellus:
    ! Form Window:n Message Actions -kohdassa oleva ACK-sano-
    maan reagointi
    !
    ! Esitysmuodon tulee olla CF_Text
    If nFormat != CF_Text
      Call SalMessageBox( 'En pysty välittämään muuta kuin
        CF_Text-tyyppistä tietoa!', 'DDE-renki', MB_Ok )
    ! Tehdään tilanvaraus varsinaiselle välitettävälle tiedolle
    Set nData = SalDDEAlloc()
    ! Kerro isännälle, että tässä tule vastaus REQUEST-sanomaan ja
    ! käske isäntää poistamaan tilanvaraus, kun on vastaanottanut sen
    Call SalDDESetDataText( nData, DDE_fResponse | DDE_fRelease,
      fidstrUser )
    Call SalDDEPost( hWndClient, WM_DDE_Data, hWndForm,
      nData, altem )
```

Miksi ei siis käytettäisi valmiita Windows-sovelluksia DDE:n avulla mahdollisimman paljon hyväksi?

Yhteydenpidon hallittu päättäminen

Kaikissa yhteydenpitotavoissa yhteyden päättäminen tapahtuu samoin: joko renki tai isäntä lähettää WM_DDE_Terminate-sanoman, jää odottamaan toisen vastausta, ja vasta sen jälkeen mahdollisesti lopettaa toimintansa. Esimerkkisovelluksien yhteyden päättämiseen liittyvä koodi on nähtävissä kuvassa 10.

Muita tapoja pitää yhteyttä

Esimerkkiohjelman avulla on saatu käytyä DDE-tekniikan käytännön toteutuksen hieman hankalatkin yksityiskohdat läpi. Tä-

män jälkeen ei pitäisi olla vaikeuksia toteuttaa myös kahta muuta yhteydenpitotapaa: jatkuva tiedonsiirtoyhteyttä ja komennon suorituttamista rengillä.

Näistä jatkuva tiedonsiirtoyhteys on varmaankin eniten käytetty tapa ja sopii mm. alussa mainittuun pörssikurssisovellukseen. Komennon suorituttaminen taas on suhteellisen suoraviivaista. Siinä esimerkiksi Excelille lähetetään suoraan merkkimuotoisena Excelin ymmärtämiä käskyjä.

Entäs sitten Presentation Manager

Tässä esitetyt periaatteet sopivat suurimmalta osin DDE-tekniikan toteuttamiseksi myös OS/2 Presentation Manager (PM) -sovel-

luksiin. Eräitä pieniä muutoksia on kuitenkin tehty.

PM:ssä yhteydenotto suoritetaan juuri sitä varten toteutetuilla funktioilla, eikä send-funktioita käyttäen. Muitakin DDE-toimenpiteitä varten on PM:ssä olemassa joukko omia funktioita.

PM:ssä DDE-sanomien parametrit talletetaan struktuureihin (DDEINIT & DDESTRUCT), joten niihin on saatu mahtumaan enemmän tietoa. Vakiomuotoisen struktuuri viedään yhteiseen (GIVEABLE) muistisegmenttiin ja sen perään varataan välitettävän tiedon määrästä riippuen vaihtelevan kokoinen alue, jossa varsinainen tieto välitetään.

DDE-tekniikan pääperiaatteet ovat Presentation Managerissa kuitenkin samat kuin Window-

DDE-järjestelmät tulevaisuudessa

Tulevaisuudessa Windows- ja Presentation Manager -maailmaan tulee yhä lisää uusia sovelluksia, joista löytyy DDE-tuki. Sovellukset alkavat olla niin monipuolisia, että ohjelmointityötä tekeillä on täysi työ keskittyessään omimalle alueelleen. Miksi ei siis käytettäisi valmiita Windows-sovelluksia DDE:n avulla mahdollisimman paljon hyväksi?

Tällä hetkellä DDE on mahdollista samassa laitteessa pyörivien Windows-sovellusten välillä. Microsoft on kuitenkin arvioinut myös eräitä sellaisia DDE-tuotteita, joiden avulla DDE-tekniikka tulisi lähitulevaisuudessa mahdolliseksi myös verkon eri koneissa toimivien sovellusten välillä. ■

Kuva 10: DDE-keskustelun päättämiseen tarvittava koodi.

DDE

Värilliset nestekidenäytöt tulevat jo tänä vuonna

Kuvaputki on edelleen pisimmälle kehitetty, eniten käytetty ja halvin näyttötekniikka. Kuvaputken suurin haittapuoli on näytön koko. Monessa käyttökohteessa on paljon etua, jos näyttö on litteä. Lisäksi litteä näyttö mahdollistaa monia sovelluksia, joita kuvaputkella ei voida toteuttaa. Yksi esimerkki on kannettava mikrotietokone.

Mikään litteä näyttö ei vielä vakavasti uhkaa kuvaputken johtosamaa. Kaikkein lupaavimmalta haastajalta vaikuttaa tällä hetkellä nestekidenäyttö.

Markkinanäkymiä ja sovellusvaihtoehtoja

Japani hallitsee selvästi nestekidenäyttöjen kehittämistä ja markkinoita. Nestekidenäyttöjä on jo nyt tarjolla laaja valikoima, mutta suuren luokan liiketoiminta on vasta alkamassa. Japanin merkittävimmät nestekidenäyttöjen kehittäjät ovat viime vuoden aikana ilmoittaneet useiden miljardien markkojen uusista tutkimuksista ja tuotantohankkeista.

Tällä hetkellä markkinoilta löytyy 10–12 tuuman VGA-tason mustavalkonäyttöjä tietokonesovelluksiin sekä 3–6 tuuman, 220–276:n vaakarivin värillisiä taskutelevisioita. Useimmat japanilaisten suurinvestoinneista tähtäävät noin 10 tuuman värinäyttöön, jonka tärkeimpiä käyttökohteita olisivat mikrotietokoneet.

LCD-tekniikalla on lisäksi paljon mahdollisuuksia projektiönäyttöjen kehittämisessä. Projektionäytössä pienten LCD-elementtien kuva heijastetaan suurikokoiselle (40–100 tuumaa) varjostimelle. Sovellutusalueita ovat esimerkiksi HDTV, mutta myös tietotekniikan tulevaisuuden työasemat.

Korkeatasoisten nestekidenäyttöjen ongelmana on kallis hinta. Tuotantomäärien on kasvettava nykyisestä ja valmistuksen saannon parannuttava, jotta kustannukset pienenisivät.

Nestekidenäytöt perustuvat jo pitkään tiedossa olleisiin fysikaalisiin ilmiöihin. Näyttöjen kehittä-

Nestekide- eli LCD-näytöt ovat parantaneet asemiaan litteiden näyttöjen välisessä kilpailussa. LCD-tekniikan tutkimukseen, kehittämiseen ja tuotteistamiseen on panostettu viime vuosina runsaasti, ja nyt tulokset alkavat myös näkyä.

Mikrotietokoneet ovat kehittyneiden nestekidenäyttöjen tärkein sovellutusalue. Niille tarkoitettuja mustavalkonäyttöjä on ollut markkinoilla jo jonkin aikaa, ja vuonna 1990 alkaa myös värinäyttöjen esiinmarssi.

mistä ovat viime vuosikymmeninä vauhdittaneet mikroelektronikan ja materiaalitutkimuksen edistysaskeleet.

Nestekide muokkaa sen läpi kulkevaa valoa

Eräillä orgaanisilla yhdisteillä on kiinteän ja nestemäisen olomuodon välissä ns. nestekiteinen olomuoto. Nestekidemolekyylit ovat säännönmukaisesti järjestäytyneitä kuten kiinteässä, kiteisessä aineessa. Samalla molekyylit voivat helposti liukua toistensa lomitse tai muuttaa asentoaan toisten molekyylin suhteen aivan kuten nesteessä.

Nestekide-ilmiö havaittiin jo vuonna 1888. Ensimmäisiä elektronikan nestekidesovelluksia alettiin kehittää 1960-luvulla. Aluksi nestekiteillä toteutettiin yksinkertaisia, vähän tehoa kuluttavia numeronäyttöjä esimerkiksi laskimiin ja kelloihin. Kun nestekidenäyttöjen tekniikka kehittyi, laajeni myös näyttöjen käyttöalue.

Nestekide ei synnytä valoa itse, vaan ainoastaan muokkaa sen läpi kulkevaa sähkömagneettista säteilyä. Yksinkertaisimmissa nestekidenäytöissä riittää ympäristön valo, joka heijastuu näytön takapinnalta. Useimmiten tarvitaan kuitenkin erillinen valonlähde näytön taakse, esimerkiksi elektroluminenssilevy tai fluoresenssiputket.

Nestekiteen molekyylit ovat optisesti anisotrooppisia, eli niiden optiset ominaisuudet riippuvat kulmasta, jossa valo molekyylin kohtaa. Sähkökentän avulla molekyylin asentoa voidaan helposti muuttaa. Samalla muuttuvat nestekiteen optiset ominaisuudet.

Parin viime vuosikymmenen aikana on esitetty suuri joukko tekniikoita, joilla nestekiteen molekyylejä voidaan ohjata kahden tilan välillä. Toisessa tilassa valo läpäisee nestekiteen ja toisessa tilassa se pysähtyy.

Eniten sovelluksia on ollut tekniikoille, joilla siirtymä on-tilasta

off-tilaan on mahdollisimman nopea ja joilla valon kontrasti tilojen välillä on mahdollisimman suuri.

Kiertonemaattinen perusrakenne

Nestekidenäyttöjen perusrakenne on kiertonemaattinen (twisted nematic, TN) LCD. Kuvassa 2 sen rakennepiirite on esitetty yksinkertaistetusti.

Kahden lasilevyn välissä on noin kymmenen mikrometrin paksuinen nestekidekerros. Lasilevyjen ja nestekiteen välissä on puolestaan ohut läpinäkyvien johtimien kerros, joiden avulla voidaan ohjata nestekiteen läpi vaikuttavaa jännitettä. Lasilevyjen ulkopinnoilla on polarisaattorit, joiden läpäisy-suunnat ovat toisiaan vastaan kohtisuorassa.

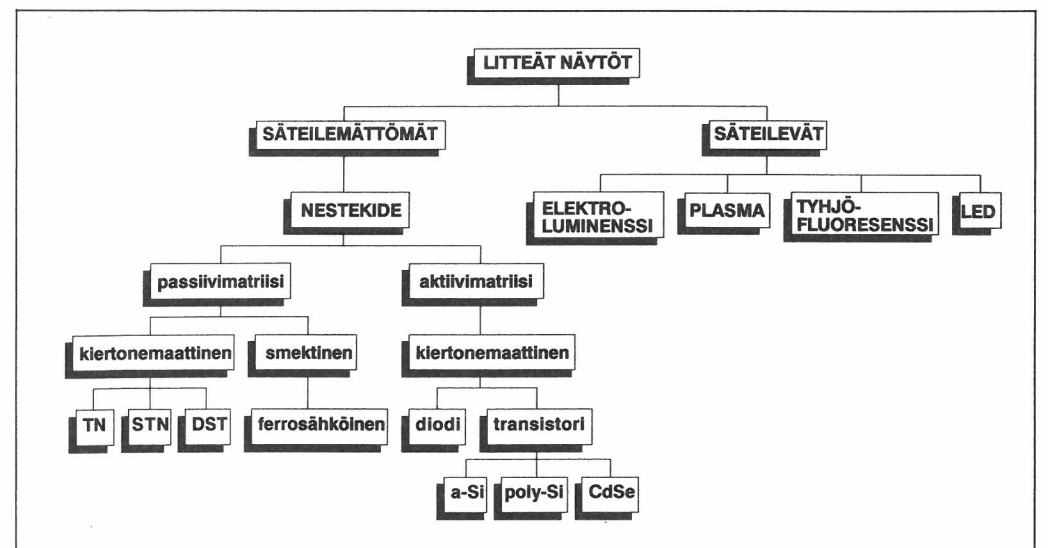
Kiertonemaattisessa rakenteessa nestekidemolekyylit on ylä- ja alapinnalla kiinnitetty polarisaattorien suuntiin. Pintojen välissä molekyylin suunta kiertyy 90 astetta.

Polarisaattorin ja lasin läpi nestekiteeseen tuleva valo on lineaarisesti polarisoitunutta. Jos nestekiteen yli ei ole kytketty jännitettä, muuttuu valon polarisaatio-suunta molekyylin kierron mukaisesti 90 astetta. Tähän on syynä molekyylin optinen anisotropia. Koska valo läpäisee myös toisen polarisaattorin, on kyseessä nestekidenäytön ”kirkas tila”.

Jos nestekiteen yli kytketään jännite (5–20 V), useimmat molekyylit asettuvat sähkökentän suuntaan. Nestekiteeseen tulevan valon polarisaatio-tila ei nyt juurikaan muutu. Valo ei enää läpäise 90 asteen suunnassa olevaa alapuolista polarisaattoria. Tämä on näytön ”pimeä tila”.

90 asteen kiertonemaattinen nestekidenäyttö vie vähän tehoa, ja se on halpa valmistaa. Kytkentäajat ovat muutamia kymmeniä millisekunteja. Haittana on huono kontrasti, noin 2–3:1. Lisäksi näytön kuva näkyy hyvin vain kaapeassa katselukulmassa.

Aktiivimatriisi-LCD:n prototyypit ovat suurimmillaan 14-tuumaisia TV-näyttöjä. Transistoreita yhdessä näytössä voi olla jopa 1,5 miljoonaa.



Kuva 1: Tärkeimmät litteiden näyttöjen teknikat.

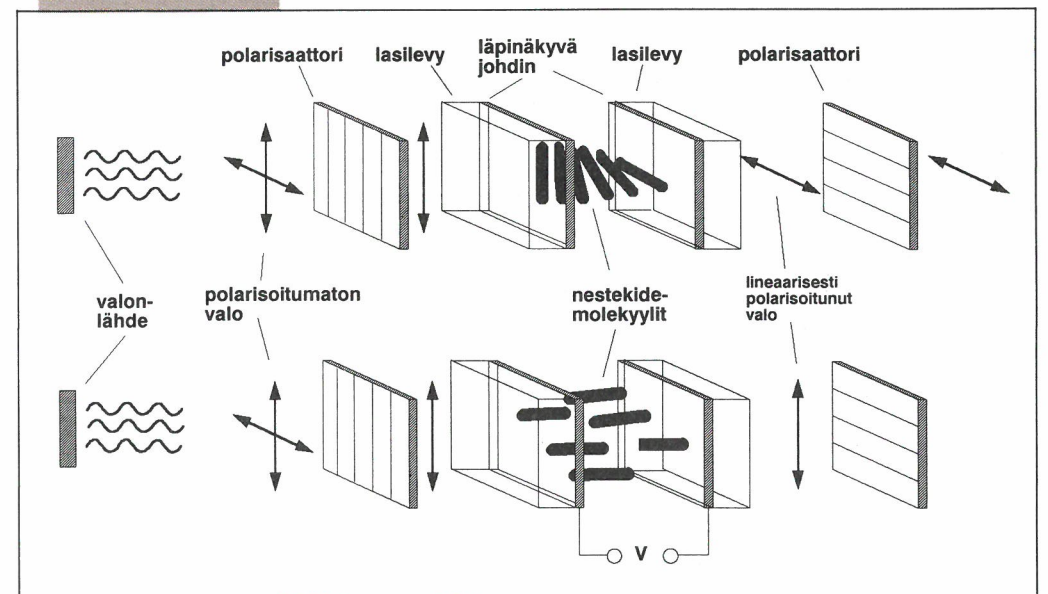
Parempi kontrasti STN-rakenteella

Superkiertonemaattisissa nestekiteissä (supertwisted nematic, STN) molekyylin kiertymäkulma on yli 180 astetta. STN-rakenteessa tarvitaan TN-näyttöä selvästi pienempi jännite-ero ohjaamaan näyttöä on- ja off-tilojen välillä. Kontrastiksi saadaan näin yli 5:1.

STN-nestekidenäytön haittana on harmaatasojen vähyys. Näyttö

on väriltään kelta-musta tai valkoinen, STN:llä ei saada mustavalkoista kuvaa. STN-nestekiteen noin 100 ms:n kytkentäajat asettavat myös rajoituksia sovelluksille.

Kannettavissa mikrotietokoneissa on käytetty STN-tyyppisiä nestekidenäyttöjä. STN:n puutteet ovat kuitenkin niin merkittäviä, että rakenne alkaa vähitellen väistyä kehittyneempien tekniikoiden tieltä.



LCD-näyttöjen tekniikka

STN-tyypisen nestekidenäytön kontrasti paranee, kun siihen yhdistetään optinen kompensattori. Samalla saadaan mustavalkokuva. Kompensatio voidaan toteuttaa toisella nestekidekerroksella, joka vastaa optisilta ominaisuuksiltaan alkuperäistä STN-nestekidettä. Ainoastaan molekyylien kiertosuunnan on oltava päinvastainen.

Tavallisessa STN-rakenteessa nestekiteeseen tuleva valkoinen valo jakautuu eri värien elliptisesti polarisoituihin komponentteihin, joiden ellipsiakselit ovat eri suunnissa. Kun rakenteeseen lisätään kompensattori, kaikki nestekiteen läpäisseet värikomponentit saadaan taas samaan suuntaan lineaarisesti polarisoituneiksi eli valo on valkoista. Kontrasti on yli 10:1.

Kun optinen kompensatiokerros on nestekidettä, puhutaan kaksinkertaisesta STN-rakenteesta (double supertwisted, DST). Toinen nestekidekerros vaatii kuitenkin

kin yhden lasilevyn lisää, mikä kasvattaa näytön painoa ja pak-suutta.

Yksinkertaisemmin kompensatio voidaan toteuttaa polymeerikalvolla, jonka optiset ominaisuudet vastaavat nestekidettä. Kal-

Suurimmat STN-pohjaisten näyttöjen prototyypit ovat jo noin 20 tuuman kokoisia.

vokompensaattorien valmistuksessa on ollut hankaluuksia, mutta nyt ongelmat alkavat olla ohi. Nestekidenäyttöjen kehityssuunta onkin tällä hetkellä sellainen, että DST-nestekidenäytöt alkavat tehdä tietä kalvorakenteille.

Kompensoituja STN-nestekidenäyttöjä käytetään pääasiassa sylimikroissa, mutta myös eräissä pöytämalleissa. Kompensoitujen STN-rakenteen haittana on hitaus, mikä tulee vastaan esimerkiksi hiirtä käytettäessä.

Suurimmat STN-pohjaisten näyttöjen prototyypit ovat jo noin 20 tuuman kokoisia. Suotimien avulla mustavalkokuvasta saadaan myös värillinen. Kuvan kontrasti ja kirkkaus ovat kuitenkin sen verran vaatimattomia, ettei tälle näyttötyypille ennusteta menestystä.

STN-pohjaisten näyttöjen lisäksi laboratorioissa on kehitteillä lukuisia joukko muun tyyppisiä nestekidenäyttöjä. Lupaavimpia ovat ferrosähköiset smektiset ja homeotrooppiset nestekiteet, joilla päästään hyvään kontrastiin ja lyhyisiin kytkentäaikoihin. Molemmilla näyttötyypeillä on kuitenkin vaikeita valmistusteknisiä ongelmia, jotka viivästyttävät tuotteistamista.

Myös Nokialla on omaa nestekidetutkimusta. Nokia Display Technicsin Esslingenin laboratoriossa Länsi-Saksassa kehitetään omaa EVA-tekniikkaa (extended viewing angle), jossa mustavalkokuva saadaan aikaan erikoisella yksikerrosrakenteella. EVA-näyttöillä ei ole vielä kaupallisia sovelluksia.

Aktiivimatriisilla päästään videokuvaan

Yksinkertaisimmat kiertonemaattiset nestekidenäytöt löytyvät kellojen ja laskinten segmenttinäyttöistä. Jokaista näyttösegmenttiä ohjataan suoraan. Tilanne muuttuu monimutkaisemmaksi, kun segmenttien sijaan on ohjattava usean vaaka- ja pystyrivin matriisinäyttöä. STN-nestekide sopii kontrastinsa vuoksi paremmin matriisirakenteeseen kuin tavallisen kiertonemaattinen nestekide. Mutta STN:lläkään ei päästä videokuvaan.

Matriisinäytön jännitettä ohjataan vaakariivi kerrallaan. Esimerkiksi PAL-järjestelmän 625 juovan näytössä jännitepulssi siirtyy riviltä toiselle 64 mikrosekunnin vä-

lein. Jokainen kuvaelementti saa jännitepulssein 40 millisekunnin välein.

Nestekide ei reagoi suoraan jännitepulsseihin, vaan jännitteen neliölliseen keskiarvoon. Matriisirakenteessa kuvaelementin kokema jännite-ero on- ja off-tilan välillä on paljon pienempi, kuin jos elementin jännite olisi koko ajan päällä tai pois päältä. Lisähaittana matriisirakenteessa on ylikuuluminen eli ohjauspulssi aiheuttaa pienen jännitteen myös lähimpiin viereisten rivien kuvaelementteihin.

Ongelmat ratkeavat, kun jokaiseen kuvaelementtiin yhdistetään erillinen kytkin. Ohjauspulssin tullessa kuvaelementtiin kytkin siirtyy joko on- tai off-tilaan. Kuvaelementin jännite pysyy samana seuraavaan ohjauspulssiin asti.

Kytkeäkin rakennetta kutsutaan aktiivimatriisiksi. Kuvan kontrasti on hyvin korkea, jopa 40:1. Aktiivimatriisissa voidaan käyttää tavallista kiertonemaattista TN-nestekidettä, jossa on paljon harmaasävyjä ja videokuvaan riittävä 30-40 millisekunnin nopeus.

Aktiivimatriisiin kytkin voidaan tehdä diodeilla, mutta yleisemmin käytetään ohutkalvotransistoreita (thin film transistor, TFT). Ensimmäiset transistorit valmistettiin kadmiumselenidistä (CdSe). Pian materiaaliksi kuitenkin tuli amorfinen pii (a-Si), jonka valmistusprosessi on CdSe:tä helpompi.

Tällä hetkellä useimmat aktiivimatriisit tehdään amorfisesta piistä, joskin myös korkeamman prosessointilämpötilan vaatimaa monikiteistä piitä (poly-Si) käytetään.

Aktiivimatriisi-LCD:n prototyypit ovat suurimmillaan 14-tuumaisia TV-näyttöjä. Transistoreita yhdessä näytössä voi olla jopa 1,5 miljoonaa. Aktiivimatriisi-LCD:tä voi pitää kooltaan maailman suurimpana integroituna piirinä. Valmistusongelmat ovat sen mukaisia.

Muut tekniikat ovat jäämässä varjoon

Tärkeimmät muut litteän näytön tekniikat ovat plasma-, elektroluminenssi-, tyhjöfluoresenssi- ja LED-näytöt.

Puna-oranssin väriset plasmanäytöt ovat menneet parina viime vuonna hyvin kaupaksi. Pääasiallisimpana käyttökohteena ovat olleet kannettavat mikrotietokoneet.

Plasmanäyttöjen haittana on korkea jännite (150-200 V) ja tehonkulutus (suurimmillaan 20-40 W). Plasmanäyttö ei sovellu akkukäyttöiseen sylimikroon. Myös plasmanäyttöjen yksivärisuus on melkoinen ongelma. Mustavalkoisia ja monivärisiä plasmanäyttöjä on toki esitelty, mutta kilpailukykyisiin kaupallisiin tuotteisiin menee vielä vuosikausia.

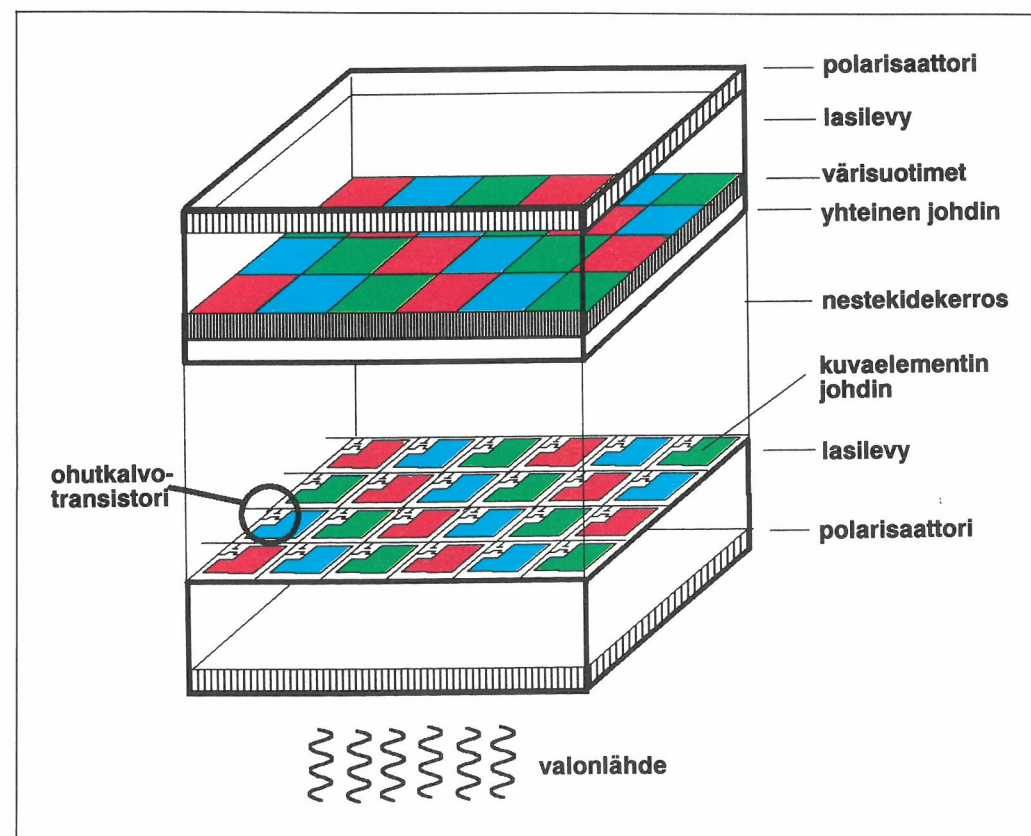
Korkea käyttöjännite, tehonkulutus ja värit ovat ongelmana myös elektroluminenssinäytöissä. EL-tekniikalla saadaan erittäin hyvä kelta-musta kuva, mutta täysvärikuvaa saadaan odottaa pitkälle 90-luvulle.

Plasma- ja EL-näyttöjen mahdollisuudet kilpailla nestekidenäyttöjen kanssa ovat huonontuneet viime vuosina. Nestekidenäytöt ovat teknisessä kehityksessä ja myynnissä selvästi edellä, ja etumatka näyttää koko ajan kasvavan. Nestekidenäyttöjen tutkimukseen ja kehitykseen sijoitetut rahasummat ovat niin suuria, että monen mielestä LCD:n menestys 90-luvulla on taattu. Plasma- ja EL-näyttöille jäävät valikoidut markkinaraot, joita niitäkin kyllä löytyy.

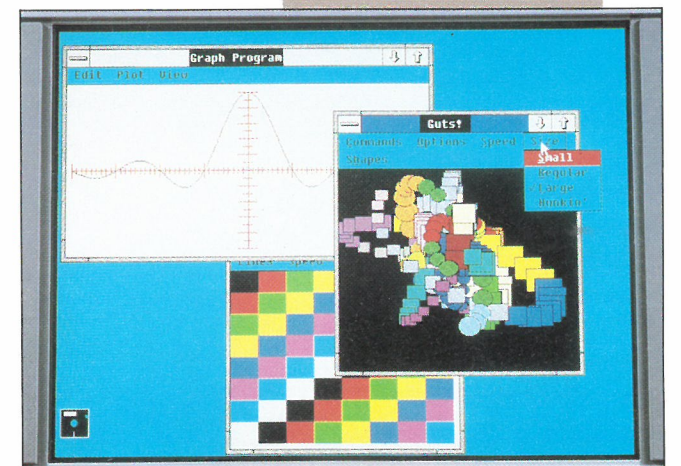
Tyhjöfluoresenssitekniikka sopii parhaiten pieniin muutaman merkin näyttöihin, esimerkiksi viihde-elektronikassa ja autoissa. Suuria matriisinäyttöjä ei teknikalla kannata toteuttaa.

Pienet merkinäytöt ovat myös LEDien sovellutusalueita. Tehonkulutus ja matriisirakenteen vaikeudet ovat olleet suurien LED-näyttöjen kehityksen esteenä. ■

Kuva 3: Ohutkalvotransistoreilla toteutetun aktiivimatriisi-LCD:n rakenne.



Nokia Datan Hannoverin CEBIT-messuilla esittelemä värillinen nestekidenäytön prototyyppi käy hyvin esimerkiksi paitsi litteiden näyttöjen joustavuudesta myös siitä, ettei niiden kuvan laatu enää jää jälkeen kuvaputkesta.



Ergonomiaa ruudun molemmin puolin

CTRL-Z

CONT.

EXIT

RESUME

POISTU

QUIT

LOGOUT

BYE

STOP

CTRL-C

LOPETA

Softarin aiemmissa numeroissa on käsitelty näyttöpäätetyön fyysistä ergonomiaa. Nyt tarkastellaan lähemmin ihmisen ja tietokoneen välistä vuorovaikutusta, käyttäjän varsinainen työvälineiden — ohjelmistojen ergonomiaa.

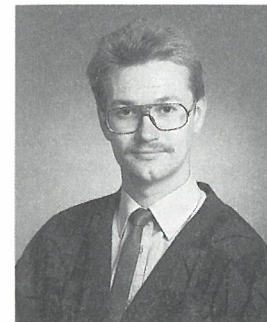
Ohjelmistoergonomia-käsitteen määrittelyä voidaan johtaa Jorma Saaren esittämästä ergonomian määrittelmästä: "Ergonomia on soveltava tietoa-alue, joka sopeuttaa teknistä ympäristöä ihmisen ominaisuuksiin. Se pyrkii ottamaan huomioon ihmisen elimistön ja käyttäytymisen toimintatilat teknisissä ratkaisuissa." Tietokonetta käyttävä ihminen työskentelee tietokoneen käyttöliittymän muodostamassa ympäristössä. Käyttöliittymän T. Malone määrittelee niiksi osiksi tietokoneohjelmaa, jotka yhdistävät inhimillisen käyttäjän tietokoneen tarjoamiin mahdollisuuksiin.



Ohjelmistoergonomia voitaisiin määritellä ergonomian osaksi, joka tutkii tietokoneen ohjelmiston ja sen käyttöliittymän sopivuutta ihmiselle, ja joka pyrkii ottamaan huomioon hänen tietojenkäsittelykykynsä.

Koska käyttöliittymää koskevat asiat riippuvat monista tekijöistä, on vaikea antaa yleispäteviä suunnittelusääntöjä hyvän käyttöliittymän tekemiseksi. Yleissääntönä voidaan kuitenkin pitää sitä, että ihmisen tulisi hallita tietokonetta eikä päinvastoin. Itse asiassa tietokoneohjelman ja sen käyttöliittymän tulisi mahdollisimman paljon muistuttaa sitä sisäistä mallia, joka työntekijällä on työstään. Kyse on työkalun tekemisestä tiettyyn tehtävään. Työkalusuunnittelussa on selviö, että uusi työkalu helpottaa ja auttaa työn tekemistä. Joskus tuntuu kuitenkin siltä, etteivät tietotekniikan ammattilaiset ota huomioon käyttäjien toivomuksia, vaan tekevät ohjelmistoista teknisesti hyvinkin elegantteja käyttäjän kustannuksella.

Ohjelmistoergonomia



Timo Kurki on TLK:n vastaava työympäristösuunnittelija.

Tehtäväanalyysi

Tärkein käyttöliittymän suunnittelun lähtökohta on työtehtävän tarkka analyysi. Tietokoneohjelmiston tulee olla työkalu, jonka avulla työ voidaan tehdä tehokkaasti ja helposti. Tämä ei onnistu, jos ei tunneta riittävästi niitä tehtäviä, joista työprosessi todellisuudessa koostuu. Tällöin on erittäin tärkeää, että työprosessin tuntevat ohjelmiston tulevat käyttäjät osallistuisivat suunnitte-

nitteluprosessiin ja kuinka paljon heidän toivomuksiaan oli otettu huomioon yrityksissä. Kuvassa 2 on esitetty, kuinka työntekijät kokivat tietokoneen vaikuttavan työhönsä. Yrityksessä numero kolme työntekijät olivat tyytyväisempiä tietokonejärjestelmään kuin kahdessa muussa yrityksessä. Tässä yrityksessä työntekijät olivat päässeet osallistumaan järjestelmän suunnitteluun, joten se auttoi heitä paremmin työtehtävissään järjestelmän vastatessa heidän toiveitaan.

Kysymys/vastaus

Kysymys/vastaus-tyylissä tietokone tavallaan ohjaa ihmistä pienin askelin eteenpäin eikä virheitä pääse helposti syntymään. Tämä keskustelutyyli käy kuitenkin turhauttavaksi, kun käyttäjän kokemus lisääntyy. Hän alkaa ennakoida tulevia kysymyksiä ja turhautuu vastatessaan niihin, varsinkin jos koneen vasteajat ovat pitkät.

Valikko

Valikolla tarkoitetaan luetteloa annettavissa olevista käskyistä. Valinta voidaan tehdä siirtämällä näytössä oleva kohdistin kohdistimensiirtonäppäimillä tai hiirellä halutun vaihtoehdon päälle ja lähettämällä sanoma sen hyväksymisestä (esim. painamalla return/enter), kirjoittamalla vaihtoehdon numero tai kirjainkoodi. Valintatapojen paremmuutta on tutkittu parhaan valintastrategian löytämiseksi.

Israelilaisessa tutkimuksessa verrattiin edellä mainittuja kolmea vaihtoehtoa ja päädyttiin siihen, että lyhyin reagointiaika halutun vaihtoehdon valitsemisessa oli kirjainkoodi, jolla oli merkityksellinen sisältö valittavan vaihtoehdon kanssa (esimerkiksi "H" / HELP). Tällainen assosiaatio auttaa, mutta järkevän kirjainyhteyden keksiminen voi olla vaikeaa. Joskus numerokin voi olla havainnollisempi, jos vaihtoehdon luonne on sen mukainen.

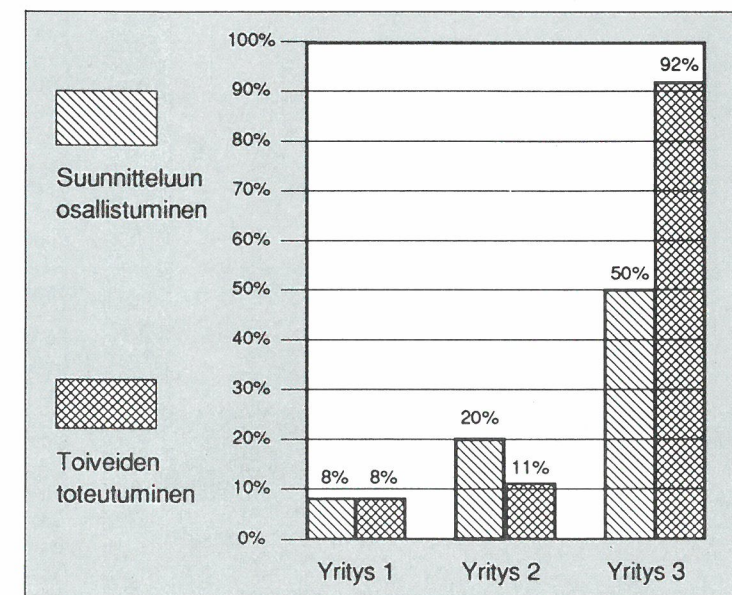
Tutkimusten perusteella on asetettu kriteereitä hyvälle valikolle. Valikkovaihtoehtojen lukumäärä riippuu tarvittavien toimintojen määrästä. Jos tarvittavia toimintoja on runsaasti ja valikko kohti on vain muutama vaihtoehto, se johtaa hyvin useisiin hierarkiatasoihin. Hyvässä valikossa voi olla jopa useita kymmeniä vaihtoehtoja, jos ne on pystytty ryhmittämään järkeviksi, löytämistä helpottaviksi kokonaisuuksiksi. Jos ryhmittämistä ei voida tehdä tai kyseistä valikkoa käytetään harvoin, suositeltava lukumäärä on 4-8 valittavaa vaihtoehtoa näyttöä kohti.

Eräs tulevaisuuden visioista on tietokonejärjestelmä, joka tunnistaa käyttäjänsä taitotason ja valitsee käyttöliittymän sen mukaisesti.

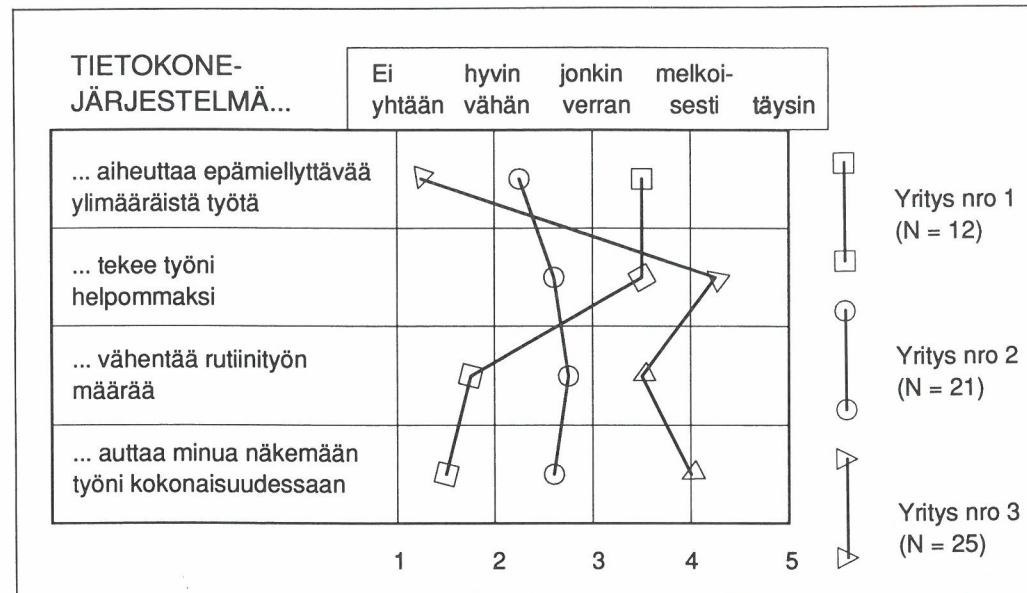
luun, sillä he hallitsevat työn paremmin kuin tietokonejärjestelmän suunnittelijat. Käyttäjien osallistumisella on muitakin etuja: työntekijät saavat aikaisemmassa vaiheessa tietoa järjestelmän tulemisesta, ja he tutustuvat uuteen järjestelmään paremmin.

Sveitsiläisessä tutkimuksessa selvitettiin kolmessa vakuutusalan yrityksessä niiden käyttämien käyttöliittymien ominaisuuksia. Kuvassa 1 on esitetty, kuinka paljon työntekijät osallistuivat suun-

Keskustelutyyliä tarkoitetaan tässä sitä tapaa, jolla käyttäjä ja tietokone kommunikoivat keskenään. Kyse voi olla esimerkiksi näppäimistöltä annettavista käskyistä, vastaamisesta kysymyksiin tai varsinkin tulevaisuudessa yleistävästä suullisesti annettavista komennoista. Keskustelutyyliä valittaessa olisi tärkeintä ottaa huomioon erilaisten käyttäjien tarpeet. Kokenut käyttäjä tarvitsee erilaisen keskustelutyylin kuin aloittelija.



Kuva 1: Työntekijöiden osallistuminen tietokonejärjestelmän suunnitteluun ja heidän toivonsa toteutuminen.



Kuva 2: Käyttäjien arviot tietokonejärjestelmien eduista (mediaanit).

Sveitsiläisessä tutkimuksessa verrattiin hierarkioiden ja valikossa esiintyvien vaihtoehtojen lukumäärää. Kuvasta 3 voidaan todeta, että yrityksessä kolme oli eniten vaihtoehtoja valikkoa kohti ja vähiten hierarkioiden rakennepuu", josta käy selvästi esille kolmannen yrityksen järjestelmän leveä mutta matala rakenne. Vali-

kosta toiseen siirtymisen tulisi olla mahdollisimman sujuvaa ja monipuolista.

Lomake

Lomakkeenomaista syöttötapaa voidaan käyttää silloin, kun työtehtävä on samankaltaisten tietoartikkelien syöttämistä. Ihmiset

ovat tottuneet täyttämään lomakkeita tietoja antaessaan, joten tällainen kommunikointi noudattaa arkielämästä tuttua mallia. Jokaisen kentän (täytettävän kohdan) jälkeen ohjelman tulisi tarkistaa, että annettu tietoalkio on oletettua tyyppiä, ja antaa tarvittaessa välittömästi korjausmahdollisuus ohjeineen. Täyttäjäräjästyksen olisi kuitenkin oltava vapaasti valittavissa.

Toimintonäppäimet

Funktio- eli toimintonäppäimille on määritelty tiettyjä toimintoja, jotka aiheuttavat käskujen suorittamisen tai tuovat näyttöön valikon. Yleensä nämä näppäimet on sijoitettu näppäimistöä omalle alueelleen ja niistä on usein kartta, joka kertoo kunkin näppäimen merkityksen. Tämä on usein toteutettu myös kuvaruudulla, jolloin näppäinten merkitys on esitetty esimerkiksi kuvaruudun alalaidassa. Tyypillisiä toimintonäppäimiä käytäviä ohjelmia ovat esimerkiksi tekstinkäsittelyohjelmat. Tällaisen ohjelman sujuva käyttö vaatii jatkuvaa ohjelmalla työskentelyä, koska muutoin näppäinten merkitys ei pysy mielessä, vaan joudutaan käyttämään karttaa apuna, mikä puolestaan hidastaa työskentelyä.

Suorat komennot

Suorilla tietokoneelle annettavilla komendoilla työskentely on mahdollisimman nopeaa ja sujuvaa. Tällöin ei edetä valikkoa pitkin haluttuun toimintoon, vaan käsky annetaan suoraan tietokoneelle, joka suorittaa toiminnon välittömästi. Tällaisen järjestelmän hallitsemisen vaatiman taidon hankkiminen kestää kuitenkin kauan ja vaatii hyvin sisäistetyn mallin ohjelman rakenteesta ja toiminnasta.

Keskustelutyylin valinnasta voidaan antaa taulukossa 1 esitetty suositus. Ei saa kuitenkaan unohtaa, että samaa järjestelmää käyttää taidoiltaan erilaisia käyttäjiä, jolloin ohjelman joustavuus joutuu koetukselle. Eräs tulevaisuuden visioista on tietokonejärjestelmä, joka tunnistaa käyttäjänsä taitotason ja valitsee käyttöliittymän sen mukaisesti.

Taulukko 1. Keskustelutyylin valinta käyttöliittymässä käyttäjäkokeneisuuden mukaan.

Kokeneisuus	Keskustelutyylit
Vasta-alkaja	Kysymys/vastaus
Vähäinen kokemus	Valikko tai lomake
Keskinkertainen kokemus	Funktio/toimintonäppäimet
Runsas kokemus	Suorat komennot

Ohjaus

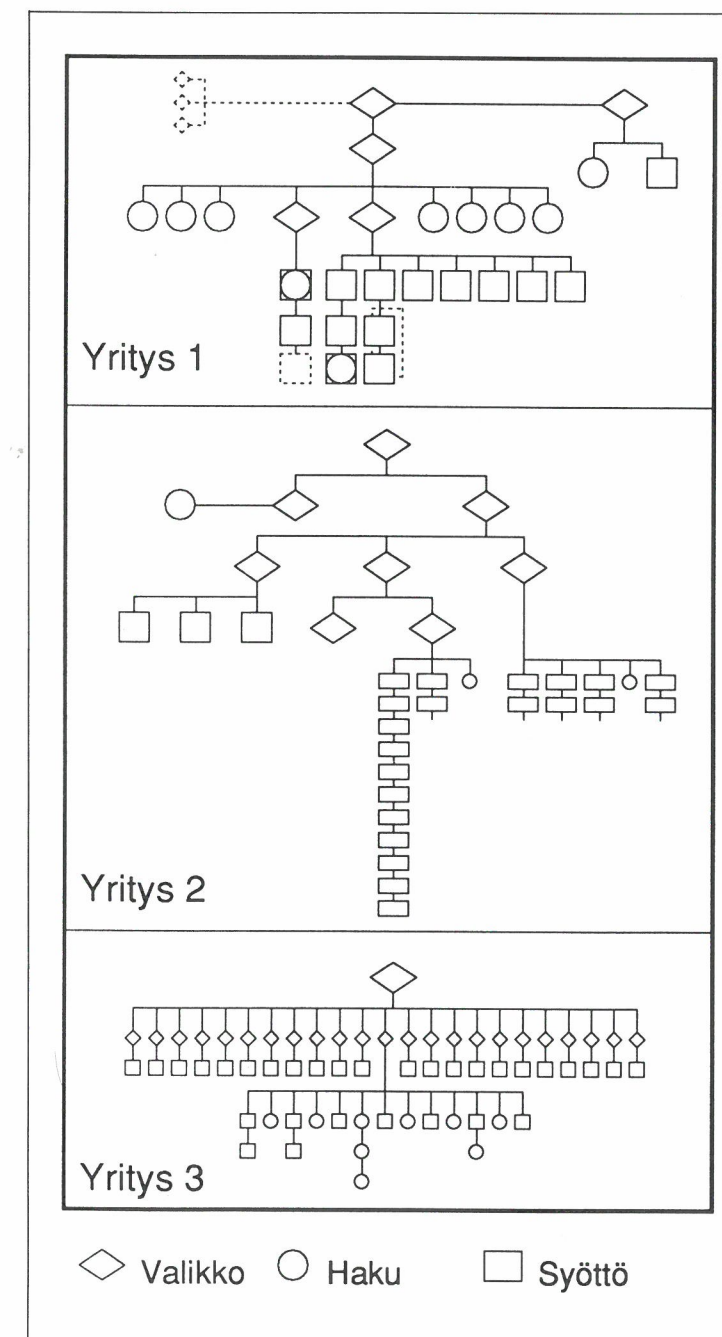
Koska käyttäjän on ohjattava järjestelmää, tietokoneen on annettava tietoa siitä, mitkä toiminnot ovat mahdollisia. Ohjelma tulisi voida keskeyttää halutussa kohdassa ja palata siihen takaisin myöhemmin. Palaamisen edelli-

seen kohtaan ohjelmaa tulisi olla mahdollista, esimerkiksi syötetyn tiedon muuttamiseksi. Viimeisen käskyn perumisen tulisi olla helpoa (undo-käsky). Turhat alku- ja loppurutiinit tulisi karsia pois. Jos ollaan tekemässä peruuttamattomia toimintoja, niistä tulee aina olla ilmoitus sekä varmistus, että kyseistä toimintoa todella halutaan.

Kirjallisuutta:

- M. Paananen, R. Raunio (1988) Tuoteverstaasorganisaation hajautettu visuaalinen ohjausjärjestelmä. Diplomityö, Teknillinen korkeakoulu, tietojenkäsittelyopin laboratorio.
- P. Spinas (1985) VDU-work and user-friendly man-computer-interaction: Analysis of dialogue structures. Teoksessa L. Norros, M. Vartiainen (toim.) Psychological aspects of the technological and organizational change in work. Proceedings of the symposium organized by the Finnish Psychological Society and the Psychological Society of GDR, Espoo 22.-24. October 1985. Yliopistopaino, Helsinki 1985.
- J.-C. Sperandio (1987) Software ergonomics of interface design. Behaviour and Information Technology 6, 271-278 (3).
- D. Shinar, H. I. Stern (1987) Alternative option selection methods in menu-driven computer programs. Human Factors 29, 453-459 (4).
- K. Paap, R. Roske-Hofstrand (1986) The optimal number of menu options per panel. Human Factors 28, 377-385 (4).
- H. von Benda (1987) On the design of human-computer interaction for administrative offices. Behaviour and Information Technology 6, 343-346 (3).
- J. Saalo, J. Lehtelä ja Pentti Seppälä (1988) Keskustele- van tietokonekäytön tehokkuus ja psyykinen kuormittavuus, esitutkimus. Työterveyslaitos, Työsuojelurahasto.

Kuva 4: Yritysten tietojärjestelmien rakenteiden vertailu.



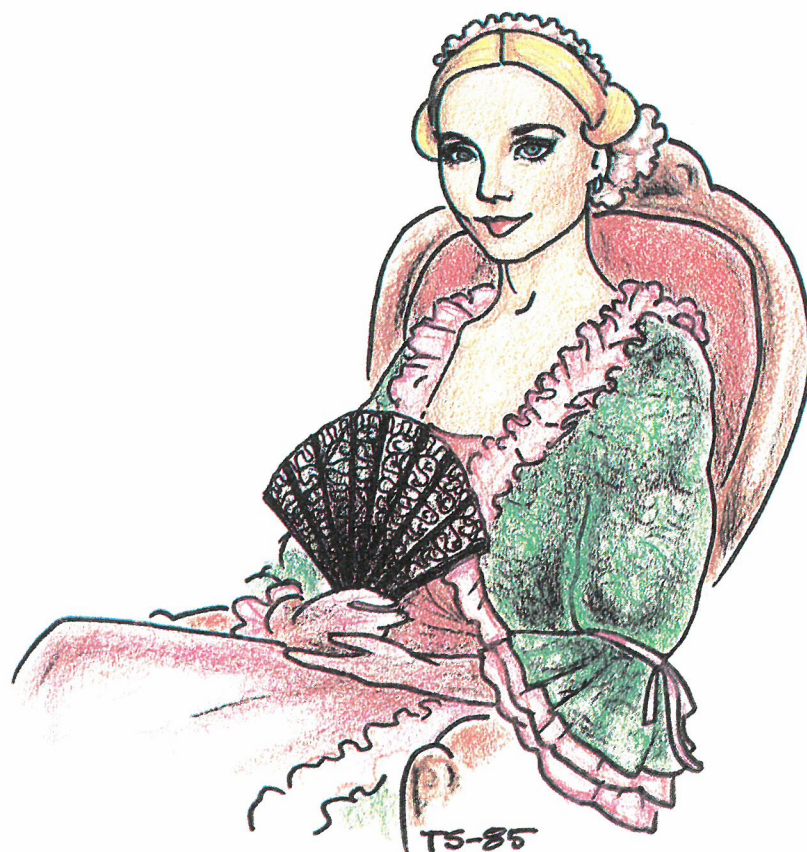
Virhetilanteet

Virheilmoituksen tulisi seurata välittömästi virheellistä toimintaa. Virheilmoituksen yhteyteen on hyvä liittää äänimerkki helpottamaan virheen havaitsemista. Ilmeisistä virheistä voidaan varoittaa etukäteen tai estää kokonaan virheellisen toiminnon suorittaminen. Virheilmoituksista tulisi ilmetä virheen laatu ja ohjeet siitä, miten virhe voidaan korjata. Jos korjausta ei osata tehdä tulisi käyttäjälle esittää tapa lopettaa ohjelman käyttö sillä kertaa ja tallentaa tehty työ. Ohjelma ei koskaan saisi mennä "nuriin", eli sellaiseen tilaan, jossa jatkaminen ei ole mahdollista. Jos näin kuitenkin käy, tulisi kaikki ennen virhetilannetta tehty työ saada talteen.

Tekniikan kehittyminen ja tiedon lisääntyminen ihmiselle sopivasta käyttöliittymästä tulee oleellisesti helpottamaan tietokoneen kanssa työskentelyä. Osa tietokoneiden tuotantokustannusten alenemisesta käytetään käyttäjäystävällisyyden parantamiseen. Asiaan voivat vaikuttaa myös käyttäjät, jotka suosivat käyttäjäystävällistä laitetta ja järjestelmää. Ei pidä kuitenkaan luulla, että se ratkaisisi kaikki ongelmat, vaan töiden sisältöön tulisi kiinnittää entistä enemmän huomiota, etteivät ositetut ja yksitoikkoiset työt lisääntyä aina- kaan tietotekniikan käyttöönoton seurauksena.

Learning Ada

Part 7 — Parallelism



controls the execution. In fact, tasking always causes a certain run-time overhead in a micro environment: the executable program file contains a special run time system that often requires 10 kilobytes as a minimum.

Switching of processes also takes time: transmitting a value from one task to another can be 60 times slower than calling a procedure with the same type of parameter.

Despite these drawbacks tasking, when applicable, shortens and simplifies a program text and makes it easier to maintain. After all, the overhead caused by task switching generally has no meaning for the user.

So when is it worth using tasks?

1. You can't say when things will happen. A good example is a controller process that interacts in real time with physical processes in the real world.
2. You don't care when things will happen.
3. It is easier to think of a problem as a set of independent processes (concurrent objects) instead of a main program controlling the tens or hundreds of different states the system can be in.

A good example of a multi-tasking problem is an lift controller. It is natural to write the program using tasks, so that each lift object and each passenger object is represented by task, and one object acts as the main controller. You can't know when someone will press a button, and you don't care in what order the buttons are pressed.

Of course the lift problem could be solved by using subprograms, that sequentially check all the buttons inside and outside the lifts, check the positions of each lift, and move them accordingly. The number of possible states is large: lifts times floors times up/down times empty/non-empty times what else ... We really don't envy the poor programmer!

On the other hand, purely sequential processes do not need tasks. If your objects are easy to implement without tasks, or if you have a ready-made sequential algorithm, don't use tasks!

If tasking makes programming easier, you are using it correctly. If it makes programming harder, you are misusing it.

Task Specifications and Task Bodies

Let's assume that you have decided to implement a lift package using tasks. Here we show one way of constructing task to object oriented design.

Task are similar to packages: they have a specification and a body:

```
with BUILDING_SPEC; -- This package contains the
-- number of lifts and
-- floors in the building.
package LIFT is
  task type LIFT_TYPE is
```

```
    entry GOTO_FLOOR
      (X: BUILDING_SPEC.FLOOR_TYPE);
    entry OPEN_DOOR;
  end
end
package body LIFT is
  task body LIFT_TYPE is
  begin
    ...
  end;
end;
```

Here the whole task definition is embedded in a package which, in its visible part, declares a **task type** whose definition contains two **entry declarations**: (GOTO_FLOOR and OPEN_DOOR).

A task cannot be compiled separately, so it must **always** be written within a package or a subprogram. So it always has a **master** to controls its execution. Tasks declared in a master begin when the master is started up, and the master cannot end its execution till its tasks have been completed. It is also possible to activate tasks dynamically using the **new** allocator (see the Reference Manual for details).

Purely sequential processes do not need tasks. If your objects are easy to implement without tasks, or if you have a ready-made sequential algorithm, don't use tasks!

A task specification can only contain entry declarations. The entries are used for task synchronization. They can have parameters (like GOTO_FLOOR-declaration above) that transmit information from one task to another.

The task type is used outside the package to declare the task objects:

```
with BUILDING_SPEC;
procedure MAIN is
  LIFTS : array
    (BUILDING_SPEC.NBR_LIFTS_TYPE)
    of LIFT_TYPE;
  FREIGHT_LIFT : LIFT_TYPE;
begin
  ...
  FREIGHT_LIFT.GOTO (3);
  FREIGHT_LIFT.OPEN_DOOR;
  ...
end;
```

In Figure 2 an ordinary package body is compared to a task body.

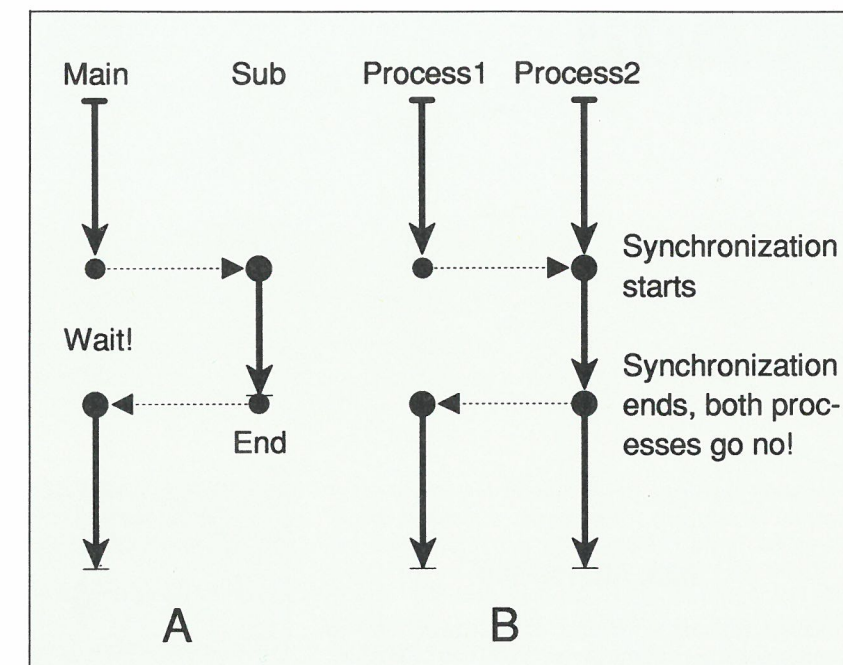


Figure 1: Sequential versus parallel processes.

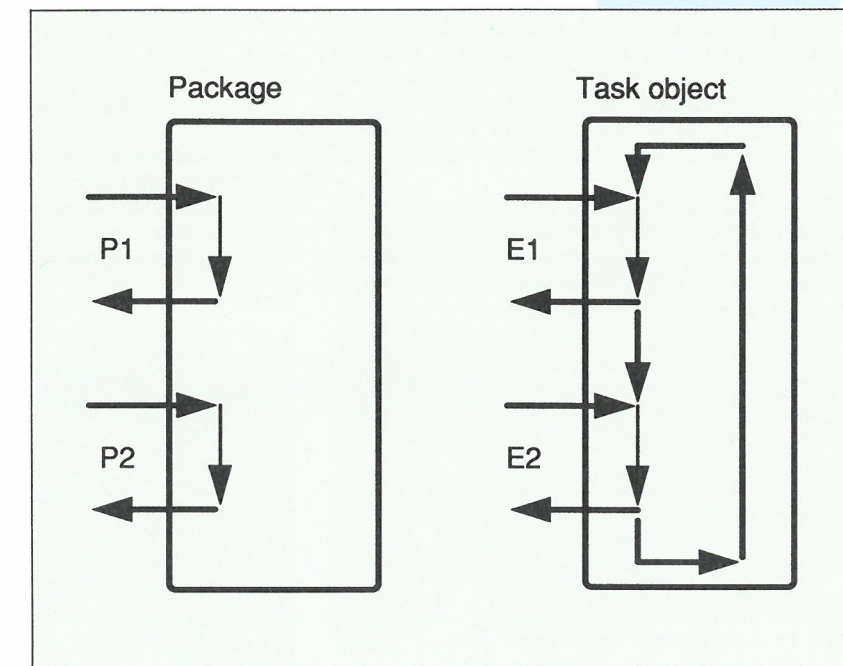
A package is a collection of distinct subprograms (the Ps), while a task body usually contains of a loop serving calls of entries (the Es).

As it is nearly always impossible to know how many times a task loop is executed, it is often written as an endless loop. Owing to the dependencies between tasks and their masters, the programmer must make sure that all the tasks in a program really complete their execution if decent ending to the main program is desired. A task either ends when it reaches the end of its body (e.g. via an exit statement within the main loop), or it can be terminated from outside with an **abort** statement. For example,

abort LIFTS (1..3);

terminates the three tasks, without worrying much about what they're doing.

Figure 2: Package body versus task body.



The last part of this course deals with parallelism in Ada. The facilities covered so far have been purely sequential. Here we show how parallel processes, called **tasks** in Ada, widen Ada's range of applications.

When a subprogram is called in a program, the caller has to wait for the subprogram to be executed (Fig. 1a). So there is only one 'thread of control', and the whole execution of the program can be modeled with a stack structure into which subprogram information is pushed and out of which it pops.

Parallelism means parallel processes in which different parts of a program run simultaneously, either independently of one another or actively cooperating through synchronization and communication (Fig. 1b).

Generally it is impossible, just by looking at a program text, to say exactly what the processes are doing at a given moment, because their speed of execution varies. This causes a certain amount of indeterminism, which makes it much harder to program parallel applications than write sequential programs. The programmer has to consider a new component, time, when dealing with the interactions between different parts of a program.

True parallelism can be achieved only with a multiprocessor computer. In a microcomputer parallelism means interleaved execution of logically concurrent processes. Each **task**, as the processes are called in Ada, gets a time slice from a scheduler that

Task Interaction

Processes running in parallel without being able to communicate are not much use. Tasks can be enabled to communicate by means of global data structures, but this is not recommended, because extreme care must be taken to ensure that two tasks are not updating the same data structure at the same time. Besides which, when object-oriented design is used, there are **no** global variables.

To make task communication safe and comprehensible, there must be exact synchronization points in the tasks. Entries declared in the task specification are used to synchronize Ada tasks.

When a task needs to synchronize with another, it issues an **entry call** to a specified entry in the other task (Fig. 3, point a). An entry call looks just like a procedure call: it contains the actual parameters corresponding to the defined in the entry declaration. If the entry call is not answered immediately, the task waits (Fig. 3, point b) until the called task reacts an **accept statement** for the called entry (point c).

When this happens, a **rendezvous** takes place: the parameters in the call are transmitted to the called task (point c), the statements within the accept statement are executed (point d), and the **out** parameters are evaluated and transmitted to the caller (point e). The caller waits for the rendezvous to be completed, after which both tasks continue in parallel (point f).

Similarly, if an accept statement is encountered without a pending entry call, the task waits for a call and a rendezvous.

The calling and called tasks are not symmetric. The calling task always names the called task, but the called task does not know which task has called it. If it is absolutely necessary for the caller to be known, its name must be transmitted as a parameter.

Each entry has a queue so several tasks can call the same entry faster than calls are served. Execution of an accept statement always removes one entry from the queue. Note that, owing to the waiting rules there cannot be more than one call from a given task in a single entry queue.

Entry calls and accept statements form the basis of Ada task interaction. Consecutive accept statements, however, serve the callers in a fixed order, and there are often situations where a task could serve any entry in a given set. It must also be possible for either the caller or the callee to stop waiting if the other does not react in a reasonable time. These features are implemented with the **select statements**.

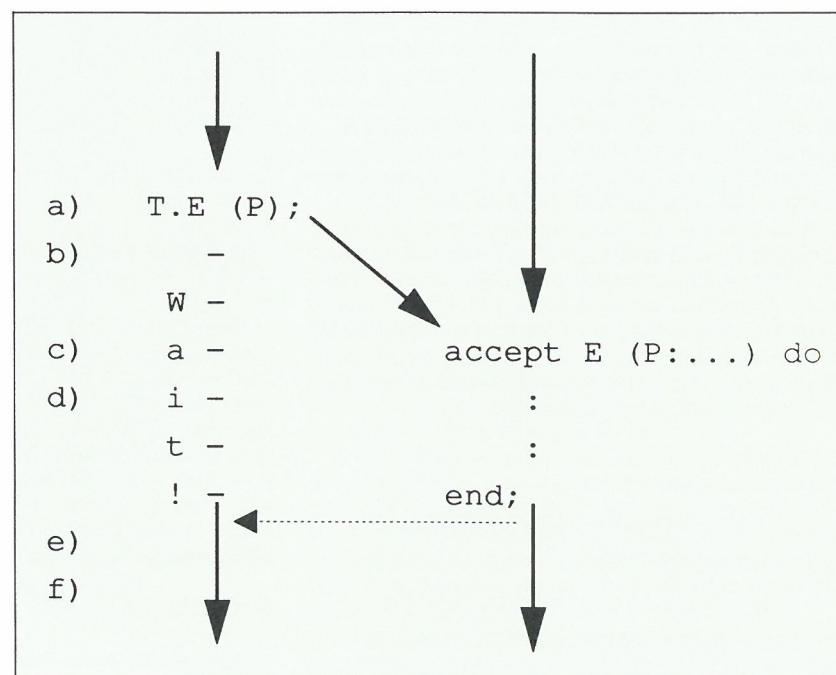


Figure 3: Task synchronization.

Select Statements

Select statements give us the following additional facilities for task interaction:

1. Cancelling an entry call if a rendezvous is not possible immediately (**conditional entry call**), or within a given time (**timed entry call**). Each such entry

```
select
  accept GOTO_FLOOR
    (X: BUILDING.SPECFLOOR__TYPE) do
    ...
  end;
or
  accept ALARM do
    ...
  end;
or
  terminate;
end select;
```

Figure 4: Waiting for several entry calls in parallel.

The last part of the series



Veikko Kehä is a Software Engineering Manager in Nokia Data Product Development.

call can optionally contain a sequence of statements that is executed if the entry call is cancelled.

2. Waiting for calls of several entries in parallel. See Figure 4.

3. Controlling the set of callable entries with Boolean expressions.

4. Executing statements other than accept statements if none of the entries are called immediately, or within a specified time.

5. Terminating the task if all the related tasks are ready for termination. This is the third way of terminating a task (cf. **abort statement** above).

The **terminate** alternative in Figure 4 cannot be chosen until all the other tasks (including the main program) have either completed their execution or are ready to choose the **terminate** alternative.

The time delays in the caller and in the callee can be used, say, to prevent a deadlock in which two tasks are both waiting for each other.

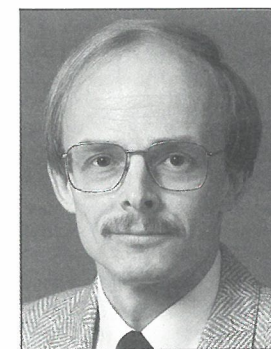
A notable aspect of Ada tasking is that the operating system is completely hidden. All necessary features are included in the language.

General Remarks

A notable aspect of Ada tasking is that the operating system is completely hidden. All necessary features are included in the language. For this reason it makes no difference whether we write a program for a mainframe computer with a complex set of machine instructions for parallelism, or for a microcomputer with no such instructions. All this puts a heavy load on the compiler, which has to generate an efficient machine language program for the target computer.

One problem in Ada tasking is the combining of tasks and exceptions. There are several special rules to be mastered for task writing.

An obvious case is the situation where a task calls another task that has already been terminated: a **TASKING_ERROR** exception is then raised in the caller. An interesting situation is that where an exception is raised during a rendezvous. If it is not handled



Lassi Juutinen is a Senior Software Engineering Specialist in the Application Environments Department of Nokia Data Product Development.

in the accept statement, it is propagated to **both** tasks. A real problem is caused by the rule that an unhandled exception is not propagated from a task to the parent unit. The task is abandoned and the exception is lost. The task just quietly 'fades away'. But Ada enables the programmer to write error handling exactly where it is needed.

The tasking facilities make it possible to build constructs that solve many classical problems of parallelism. These include the producer-consumer problem, in which data items are transmitted from one task to another using a common buffer. Conventional mutual exclusion and synchronizing facilities like semaphores, events, signals, and monitors can be modelled with the tasks. But it should be noted here that the rendezvous concept itself is more powerful and less error-prone than these lower level tools. ■

References

- /1/ Reference Manual for the Ada Programming Language, January 1983.
- /2/ Booch, G.: Software Engineering with Ada, Benjamin/Cummings, 1987.
- /3/ Barnes, J.G.P.: Programming in Ada, Addison-Wesley, 1989.
- /4/ Gehani, N.: Ada: Concurrent Programming, Prentice-Hall, 1984.

The last part of the series

This is the last part of our seven-part Ada course that started almost two years ago in Softari 3/88. We hope that the way how this series has spread over three years and two decades hasn't caused you too much inconvenience.

We are looking forward to collect these articles into a booklet that will come out sometime next summer or autumn. If you'd like to have a copy of your own, please let us know by sending a postcard to the editors of Softari. There could even be some copies of the Finnish Pascal booklet still left... Both booklets are of course free of charge.

Moni Windows pyörää vajaateholla — turhaan

Edellisessä numerossa esitetyt muistikartat ovat mukana myös tällä kertaa, sillä niistä on hyötyä Windowsin käyttöönotossa. Ennen tähän artikkeliin tutustumista myös edellinen osa kannattaa lukea läpi, sillä tekstissä oletetaan siinä esiteltyjen asioiden olevan hallinnassa.

Suurin osa tässä ja edellisessä osassa esitetyistä numeerisista tiedoista ja suosituksista perustuu Jürgen Zieglerin viime vuonna tekemään tutkimukseen.

80286 ei osaa isoja EMS-ikkunoita

Valitettavasti 80286-suorittimen ympärille rakennetut tietokoneet eivät voi käyttää kuin pieniä EMS-ikkunoita, koska ne eivät kykene korvaamaan perusmuistiin edes osittain LIM 4.0 -määrittelyn mukaisilla EMS-ikkunoilla. Tämä on yksi niistä syistä, joiden vuoksi Nokia Data suosittelee lisämuistikortteja käyttäville 80286-koneilleen LIM 3.2 -määrittelyssä pitäytymään Intelin Above Board expanded-muistia.

Kuten Softari 4/89:ssä kerrottiin, suurien EMS-ikkunoiden puute tarkoittaa käytännössä sitä, että 80286-koneet eivät pysty ajamaan useampaa kuin yhtä suurta sovellusta samanaikaisesti. Niinpä Windows onkin tarkoitettu pääasiassa 80386-koneille.

Lisämuistia 80286-koneeseen kannattaa hankkia vähintään megatavun verran. Se tulee jakaa extended- ja expanded-muistiksi — mieluiten siten, että extended-muistia jää mahdollisimman vähän. Sitä tarvitaan kuitenkin vähintään 64 kilotavua, jotta Microsoftin ja AST Researchin ylämuistialuemäärittelyt pitäisivät paikkansa.

Eräissä Nokia Datan kansainvälisissä 80286-malleissa (DT215 ja DT216) lisämuistin on oltava kooltaan vähintään kolme mega-



Softarin edellisessä numerossa käsiteltiin eri muistialueiden ja ohjelmien merkitystä Windows 2.11 -ympäristössä. Windowsin käyttöön ei päästy vielä käytännön tasolla käsi. Tässä artikkelissa keskitytään kuitenkin itse asiaan eli asennetaan Windows 2.11 MikroMikkoon. Liian moni Windows toimii nimittäin vajaateholla, koska sen asennuksessa ei ole otettu laitteiston ja Windowsin teknisiä yksityiskohtia riittävän huolella huomioon.

tavua, sillä ne pystyvät jakamaan muistiaan vain megatavun kokoisina paloina. Näin ollen extended-muistin minimikoko on kokonainen megatavu. Näissä koneissa expanded-muistin hallintaan käytetään EMM412.SYS-ohjainta, ja CONFIG.SYS:iin on kyseiselle riville lisättävä optio —D, muuten esimerkiksi Excelin kanssa voi ilmetä ongelmia.

Kaikkien expanded-muistia hyödyntävien ohjainten on löydettävä myös CONFIG.SYS-tiedostosta. Ne on kirjoitettava itse expanded-muistin ohjaimen määrittävän rivin jälkeen, jotta ne osaavat käyttää kyseistä muistialuetta.

Käytännön esimerkki 80286-ympäristössä

MikroMikko 3 sisältää muiden Nokia Datan 80286-koneiden tapaan jo itsessään 386 kilotavua extended-muistia. Niinpä kun siihen asennetaan esimerkiksi Intel Above Board, kortin sisältämä lisämuisti voidaan kokonaisuudessaan määrittellä expanded-muistiksi. Jos muistialue D0000h—DFFFFh on vapaa, CONFIG.SYS:iin voidaan kirjoittaa seuraava rivi:

```
DEVICE = <polku> EMM.SYS  
AT D000 208
```

Esimerkin viimeinen parametri (208) kertoo muistikortille I/O-toiminnoille varatun osoitteen. Lisätietoja löytyy mm. Intel Above Boardin oheismateriaalista.

80386-pohjaiset koneet antavat vaihtoehtoja

Tässä esitetyt ohjeet perustuvat oletukseen, että järjestelmään kuuluu SCSI-sovitin, joka löytyy useimmista 80386-pohjaisista tietokoneista. Adaptecin SCSI-sovitin on Windowsin kannalta suosi-

Windows 2.11 ja muistin mysteerit — Osa 2

teltavampi kuin Western Digitalin, koska se ei varaa 639 ja 640 kilotavun välistä muistialuetta omaan käyttöönsä. Koska Windows varaa EMS-ikkunalleen tilaa 16 kilotavun kokoisina paloina, Western Digitalin sovitin hukkaa kokonaisen ison EMS-ikkunan sivun (16 kt) yhden kilotavun vuoksi.

80386-koneissa voidaan käyttää sekä Windows/286:a että Windows/386:a. Windows/386:n tärkein etu piilee sen kyyssä ajaa useita DOS-sovelluksia rinnakkain. Windows/386 muuntaa automaattisesti lähes kaiken käytettävissä olevan extended-muistin expanded-muistiksi ja hyödyntää 386-suorittimien kykyä luoda virtuaalisia 8086-koneita. Jokainen virtuaalikone pystyy pyörittämään erillistä DOS-sovellusta omassa expanded-muistista viipaloidussa muistissaan. Kuitenkin mitä tulee Windows-sovelluksiin ja Windows/386:een itseensä, niitä varten luodaan vain yksi ja ainoa virtuaalikone.

Windows/286:lla on kuitenkin se etu puolellaan, että se ei sisällä EMS-emulaattoria. Windows/386:a käytettäessä ei kyetä nimittäin käyttämään muita kuin sen omaa EMS-emulaattoria. Koska EMS ei näin ollen ole järjestelmää käynnistettäessä läsnä, sen alueelle ei voida käynnistyksen yhteydessä sijoittaa ohjainohjelmia. Tästä on haittaa erityisesti Madge I-verkkokorttia käytettäessä. Tällöin kaikki ohjaimet on ladattava joko perus- tai järjestelmämuistiin, jolloin Windowsin suuren EMS-muisti-ikkunan käytettävissä oleva muistialue pienenee — käytännössä niin paljon, että suurta EMS-ikkunaa ei voida enää lainkaan käyttää. Jos tällaisessa tilanteessa aiotaan käyttää Windows-sovelluksia eikä haluta ajaa DOS-sovelluksia, on paras turvautua Windows/286:een ja EMS-emulaattoriin (386MAX tai 386LIMIT).

Toinen Windows/386:n haitta piilee siinä, että sitä käytettäessä

Windowsin itsensä viemä muistitila on noin 32 kilotavua suurempi kuin Windows/286:ssa (vrt. Softari 4/89). Suurille EMS-ikkunoille jää näin kenties ratkaisevasti vähemmän muistitilaa.

Pienin 80386-järjestelmiin suositeltava lisämuisti on kooltaan kolme megatavua. Koska EMS-emulaattori muuntaa osan tai koko extended-muistin expanded-muistiksi, kaikki yhden megatavun yläpuolelle sijoittuva lisämuisti tulisi määrittellä alustuksissa extended-muistiksi. Sekä 386MAX/386LIMIT (versio 4.04 tai myöhempi) että Windows/386 vaaravat extended-muistia automaattisesti ylämuistialueen ohjaintaan varten.

*Windows/386
muuntaa lähes kaiken
käytettävissä olevan
extended-muistin
expanded-muistiksi.*

SCSI-kiintolevyn yhteydessä tulee käyttää WDRV.SYS-ohjainta, jonka uusin ja suositeltavin versio on päivätty 9.1.1990. WDRV.SYS:in määrittely on sijoitettava CONFIG.SYS:in ensimmäiselle riville.

Mikäli EMS-emulaattorina käytetään 386MAXia tai 386LIMITiä, kaikkien expanded-muistia käyttävien ohjainten on sijaittava CONFIG.SYS-tiedostossa emulaattorin määrittelyn jäljessä.

80386, Windows/286 ja 386MAX/386LIMIT

Mikäli 80386-koneessa käytetään Windows/286:a, EMS-muistin hallinta jää 386MAX/386LIMITille. CONFIG.SYS:iin tulee näin ollen lisätä oheinen rivi heti WDRV.SYS:in sisältävän rivin pe-

rään (mikäli järjestelmässä on SCSI-sovitin):

```
DEVICE= <polku> 386MAX.SYS  
PRO= <polku> 386MAX.PRO
```

Huom: Lauseet kirjoitetaan samalle riville välilyönnillä erotettuna.

386MAX.PRO on ascii-tiedosto, joka sisältää 386MAX/386LIMITin määrittelylauseet. Sen tulisi näyttää seuraavalta:

```
RAM=C600-C800  
INCLUDE=CC00-F000
```

Huom. 1: Alueet loppuvat todellakin tasalukuihin — siis esimerkiksi C800 eikä C7FF.

Huom. 2: Lauseiden tulee olla eri riveillä.

386MAX/386LIMIT tarkistaa ensimmäiseksi järjestelmämuistin nähdäkseen, mitkä muistialueet ovat jo käytössä. Sen jälkeen se sijoittaa 64 kilotavun kokoisen yhtenäisen pienen EMS-muisti-ikkunan järjestelmämuistiin mahdollisimman korkealle. Tyypillisesti se sijoittuu E-segmenttiin (E0000h—EFFFFh).

Tämän jälkeen 386MAX (ei 386LIMIT) varaa jäljelle jääneen järjestelmämuistin käyttöjärjestelmän ylämuistiksi (high DOS memory), paitsi jos INCLUDEn avulla on toisin määrätty. Tähän muistialueeseen voidaan sijoittaa ohjain- tai muistissa pysyvästi sijaitsevia TSR-ohjelmia (Terminate and Stay Resident). Myös 386MAXin versio 4.04:ää uudemmat versiot pyrkivät sijoittamaan itsestään osan tälle alueelle.

386MAX.PRO-tiedostoon sijoitettavaa RAM-lauseetta voidaan käyttää kertomaan 386MAX/386LIMITille, että jokin muistialue tulee jättää vapaaksi. Esimerkiksi lause

```
RAM=C600-C800
```

on useissa tapauksissa välttämätön, sillä monissa VGA ROMeissa

Windows 2.11

sa on välillä C6000h—C67FFh kahden kilotavun kokoinen reikä, joka saa 386MAX/386LIMITin kuvittelemaan, että koko muistialue C6000h—C8000h on vapaa. RAM-lauseessa käytettävät osoitteet on ilmaistava neljän kilotavun tarkkuudella, ja RAM-lauseita voi olla myös useita.

INCLUDE-lauseella annetaan 386MAX/386LIMITille lupa laajentaa suurille EMS-ikkunoille varattua muistialuetta (joka sijaitsee oletusarvoisesti 640 kilotavun alapuolisella muistialueella) järjestelmämuistin puolelle. Raja voidaan määrittää 16 kilotavun tarkkuudella. Vaikka 386MAX/386LIMIT hyväksyvätkin useiden INCLUDE-lauseiden samanaikaisen käytön, niitä ei suositella käytettävän kuin yhtä kerrallaan.

Mikäli järjestelmään kuuluu RAM-puskureita käytävä verkkokortti (esim. Madge I), 386MAX/386LIMIT ei itse huomaa kyseistä puskuria. Sille on varattava tilaa RAM-lauseella erikseen, minkä lisäksi EMS-ikkunan käytettävissä olevaa tilaa on pienennettävä vastaavasti INCLUDE-lauseella:

```
RAM=C600-C800
RAM=CD00-CE00
INCLUDE=D000-F000
```

Huom. 1: Koska RAM-lauseen avulla voidaan rajata alueita vain neljän kilotavun tarkkuudella, toisen RAM-lauseen puskurille varaama alue on aloitettava osoitteesta CD000h, vaikka itse puskuuri alkaakin vasta osoitteesta CDC00h.

Huom. 2: Myös INCLUDE-lauseessa hukataan muistia, sillä vapaa muistitila alkaa jo osoitteesta CE000h. INCLUDE-lauseelle ei voida kuitenkaan antaa muistiosoitteita kuin 16 kilotavun tarkkuudella.

Kuten jo aiemmin todettiin, 386MAX kykenee lataamaan ohjaimia ja muistiin pysyvästi jääviä ohjelmia järjestelmämuistin alueelle. Tällöin on kuitenkin muistettava, että nämä ohjelmat vievät latausvaiheessa enemmän muistia kuin varsinaisessa käytössä. 386MAX ei osaa kuitenkaan enää hyödyntää näin vapautuvaa muistia. Niinpä ohjaimia kannattaa

ladata ylämuistialueelle vain ja ainoastaan silloin, kun jokin lisäkortti (esim. Madge I) vie osan järjestelmämuistista ja muodostaa samalla sinne reikiä, joita ei muuten saada käyttöön. Hyvin harva ohjain tai muistiin jäävä ohjelma kuitenkaan sopii näihin reikiin.

EMS-ikkunoiden käytettävissä olevaa aluetta ei tule koskaan missään tapauksessa pienentää INCLUDE-lauseella vain jotta jokin ohjain saadaan sullottua järjestelmämuistin puolelle! Tuloksena on pelkkää muistin tuhlausta. Ohjaimet ja muistiin pysyvästi jäävät ohjelmat kannattaa tällaisessa tapauksessa sijoittaa suosiolla perusmuistin puolelle.

Muistin riittävyyden tarkistaminen

Ennen Windows/286 2.11:n käynnistämistä kannattaa tarkistaa, paljonko perusmuistia on jäljellä ja paljonko järjestelmämuistin puolella on tilaa suurille EMS-ikkunoille. Windowsista on nimittäin mahdotonta tarkistaa, ovatko suuret EMS-ikkunat käytössä vai eivät. Jotta suuria EMS-ikkunoita voidaan käyttää, vapaan perusmuistin ja järjestelmämuistista ikkunointiin käytettävän muistitilan on oltava yhteensä suurempi tai yhtä suuri kuin 528 kilotavua:

perusmuisti ≥
528 kt — järjestelmämuisti

Laskuissaan on hyvä muistaa, että kilotavu on todellisuudessa 1024 eikä 1000 tavua.

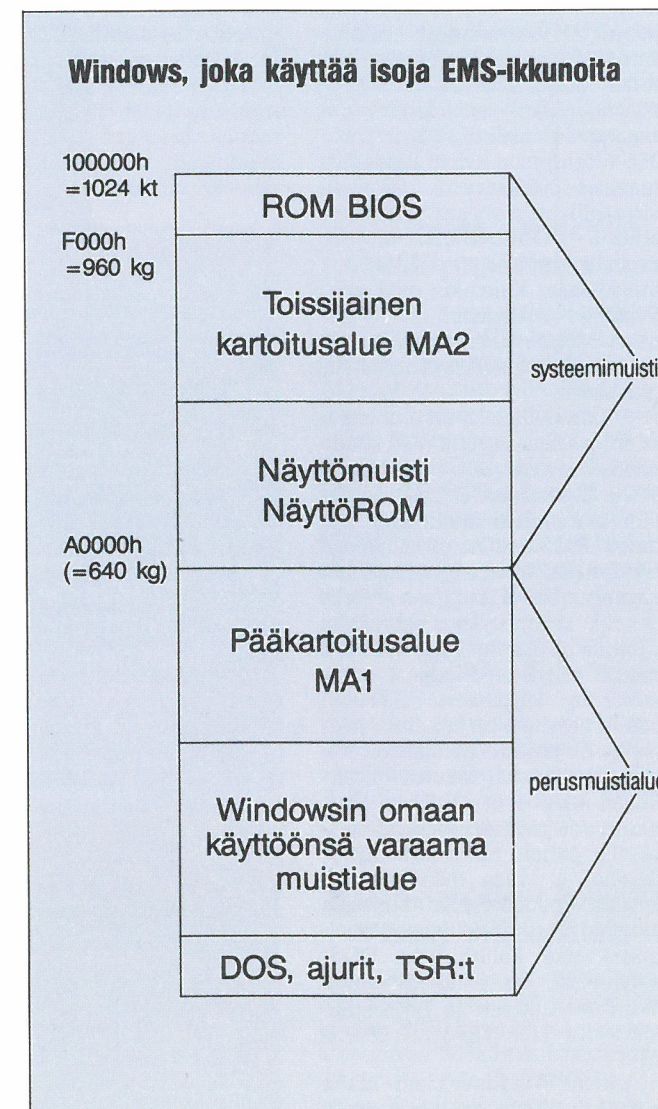
Mikäli tila ei riitä, kannattaa harkita joidenkin muistiin pysyvästi ladattujen ohjelmien ja ohjainten poistamista. Usein suurin osa niistä on käytössä aivan turhaan.

Jos esimerkiksi käytetään Madge I -korttia, jonka RAM-puskuuri sijaitsee muistialueella CDC00h—CDFFFh, tarvitaan INCLUDE-lause:

INCLUDE = D000-F000

Jos nyt annetaan CHKDSK-käsky ja huomataan, että vapaana on 470240 tavua, pystyykö Windows/286 käyttämään suuria EMS-ikkunoita?

Suurille ikkunoille on järjestelmämuistin puolella käytettävissä muistialue F0000h—D0000h = 128 kilotavua. Tämä vähennettynä 528 kilotavusta tekee 400 kilotavua. Vapaata perusmuistia on puolestaan 470240/1024 kilotavua, mikä tekee 459,2 kt, mikä on puolestaan enemmän kuin 400 kt. Näin ollen suuret EMS-ikkunat ovat käytettävissä.



Windows 2.11

Tämä lause on hyvin vaarallinen, eikä sitä tule käyttää kuin kaikkien muiden keinojen pettäessä: EMS-ikkuna alkaa annetusta osoitteesta — välittämättä lainkaan siitä, onko kyseinen muistialue vapaa vai ei!

80386-kone ja Windows/386

Madge I RAM -puskuuri ei välttämättä kuulu järjestelmään, mutta on otettu mukaan esimerkin vuoksi.

Kahden kilotavun kokoinen reikä (792—794 kt) kuuluu vain joihinkin VGA-näytönohjainkortteihin.

Windows/386 pitää itse huolta EMS-ikkunoinnista. Käynnistyessään se käy läpi järjestelmämuistin osoitteesta A0000h ylöspäin etsien käyttämätöntä muistitilaa. Se jättää kaiken löytämänsä

ROM- tai RAM-muistin väliin ja lopettaa etsinnän osoitteessa E0000h. Sen jälkeen se ottaa suurimman löytämänsä yhtäjaksoisen muistialueen suureksi EMS-ikkunaksi. Mikäli ikkuna on yli 64 kilotavun kokoinen, se hoitaa myös pienen EMS-ikkunan tehtävät. Jos se on pienempi kuin 64 kilotavua, käytettävissä on vain suuren EMS-ikkunan toiminnot. Niinpä pientä EMS-ikkunaa käyttävä Excel voi ilmoittaa, että järjestelmästä ei löydy expanded-muistia, vaikka suurta EMS-ikkunaa käyttävä Windows itse ilmoittaa expanded-muistin olevan käytettävissä.

Jotta Windows/386 kykenee käyttämään expanded-muistia, itse Windows/386:n sisältävän hakemiston on löydettävä myös AUTOEXEC.BAT-tiedoston polkumäärittämisestä.

Windows/386:n WIN.INI-tiedoston [win 386]-osaan on mahdollista sisällyttää useita erilaisia lauseita, jotka ohjaavat Windows/386:a sen etsiessä vapaata muistitilaa EMS-ikkunalleen. Niitä ei ole kuitenkaan dokumentoitu, ja oheen on koottu niistä muita tämän artikkelin esimerkkejä yksityiskohtaisempi kuvaus.

lastemmseg=etsinnän viimeinen osoite

Lauseen avulla Windows/386:lle kerrotaan, mihin osoitteeseen sen tulee lopettaa vapaan muistitilan etsintä E0000h:n asemesta. Osoite voidaan määrittellä kuudentoista kilotavun tarkkuudella.

Koska 386-MikroMikkojen muistin E-segmentti on tyhjä, Windows/386:lle voidaan antaa 64 kilotavua lisää tilaa EMS-ikkunaa varten lisäämällä WIN.INI:in lause:

lastemmseg=0F000

emmexclude=osoite-osoite

Lause käskää Windows/386:ta jättämään tietyn alueen väliin vapaata muistitilaa etsiessään. Osoitteet voidaan määrittellä neljän kilotavun tarkkuudella.

Tästä on hyötyä esimerkiksi silloin, kun Windows/386 ei kykene huomaamaan verkkokortin käyttämää puskuria. Niinpä jos järjestelmässä on esimerkiksi Madge I -kortti, joka käyttää muistialuetta CDC00h—CDFFFh puskurinaan, WIN.INI:in on lisättävä lause:

emmexclude=CD00-CE00

emmpageframe=EMS-ikkunan alkuosoite

Tämä lause on hyvin vaarallinen, eikä sitä tule käyttää kuin kaikkien muiden keinojen pettäessä. Se ohittaa Windows/386:n oman muistitilan etsintäruutiinin ja pakottaa Windows/386:n käyttämään 64 kilotavun kokoista EMS-ikkunaa. Se alkaa lauseessa annetusta osoitteesta — välittämättä lainkaan siitä, onko kyseinen muistialue vapaa vai ei! Niinpä esimerkiksi kirjoittamalla WIN.INI:in lause:

emmpageframe=0C000

saataisiin Windows/386 käyttämään VGA ROM -muistia EMS-ikkunana.

Lisäksi lause estää yli 64 kilotavun kokoisten EMS-ikkunoiden käytön, vaikka niille löytyisi muistitilaa. Lauseessa annettava alkuosoite voidaan antaa 16 kilotavun tarkkuudella.

Muistin riittävyys Windows/386:ssa

Ennen Windows/386 2.11:n käynnistämistä kannattaa tarkistaa, paljonko perusmuistia on vapaa-

Windows 2.11

Mikäli 80286-kone on liitetty työryhmä-verkkoon, sen LANMAN.INI-tiedostoa tulee muokata siten, että LAN Manager ei käytä HMA:ta.

na ja paljonko järjestelmämuistia käytettävissä suuriin EMS-ikkunoihin. Jotta Windows/386 kykenee käyttämään suurta EMS-ikkunaa, muistia on oltava käytettävissä yli yhteensä 560 kilotavua:

perusmuisti $\geq$ 560 kt — järjestelmämuisti

Jos siis esimerkiksi käytetään Madge I -korttia, jonka RAM-puskuri sijaitsee alueella CDC00h—CDFFFh, WIN.INI-tiedoston win386-osaan on lisättävä seuraavat lauseet:

```
emmxclude=CD00-CE00
lastemmseg=0F000
```

Jos nyt annetaan CHKDSK-käskey ja saadaan tulos, jonka mukaan vapaata muistitilaa on 426800 tavua, pystyykö Windows/386 käyttämään suurta EMS-ikkunaa?

Järjestelmämuistin puolella suuren ikkunan käytettävissä on siis muistialue F0000h—D0000h, mikä tekee 128 kilotavua. Tämä vähennettynä 560 kilotavusta on yhtä kuin 423 kilotavua. Vapaata perusmuistia on puolestaan 426800/1024 eli 416,8 kilotavua, mikä on vähemmän kuin 432 kilotavua. Näin ollen suurta EMS-ikkunaa ei saada käyttöön.

286- ja 386-koneille yhteisiä yksityiskohtia

Mikäli käytetään DOS 4.0:aa tai uudempaa käyttöjärjestelmäversiota, CONFIG.SYS:in BUFFERS- ja FASTOPEN-lauseissa ei saa käyttää /X-valitsinta (Aiemmissa versioissahan koko /X-valitsintahan ei edes ollut).

Muistiin pysyvästi ladattavia ohjelmia tulisi mahdollisuuksien mukaan välttää. Lisäksi monet lisäkortit (esim. HP ScanJet) saattavat viedä osansa järjestelmä-

muistista. Vapaata muistia tarvitaan vähintään 64 kilotavua (alkaen osoitteesta, joka on jokin 16:n kilotavun monikerta) yhtä pientä EMS-ikkunaa varten.

Kaikkien expanded-muistia hyödyntävien ohjainten on löydettävä myös CONFIG.SYS-tiedostosta. Ne on kirjoitettava itse expanded-muistin ohjaimen määrittelevän rivin jälkeen, jotta ne osaavat käyttää kyseistä muisti-alueita.

Windows

Windowsin alustusohjelma tarvitsee vähintään 512 kilotavua perusmuistia toimiakseen. Niinpä kun Windows/286 2.11:tä ollaan asentamassa, alustusohjelmalle on vastattava 'yes' sen kysyessä, löytyykö järjestelmästä extended-muistia. Windows/386 osaa puolestaan varata ylämuistialueohjaimensa tarvitseman extended-muistin automaattisesti. MEMSET-ohjelmaa ei kannata ajaa.

Edellisessä numerossa esiteltä SMARTDIVE on siitä hankala, että se on käytännöllisintä asentaa "käsini". Sen parametreja ei kyetä muuten asettamaan kunnon kohdalleen. Joka tapauksessa lisämuistia on oltava vähintään kaksi megatavua, sillä sille on varattava expanded-muistia vähintään 512 kilotavua, ja Windowsille tulisi jättää ainakin noin 1,5 megatavua. Enimmillään SMARTDRIVElle voi varata expanded-muistia puolitoista megatavua. SMARTDRIVE:n ohjain SMARTDRIVE.SYS vie myös perusmuistista noin 22 kilotavua.

Mikäli järjestelmässä aiotaan käyttää SMARTDRIVEa, CONFIG.SYS:iin tulee lisätä seuraava puskurimäärittäminen:

```
BUFFERS = 10
```

Jos SMARTDRIVEa ei käytet-

tä, CONFIG.SYS:iin kirjoitetaan seuraava rivi:

```
BUFFERS = 20
```

Lisäksi CONFIG.SYS:iin kirjoitetaan lause:

```
DEVICE = <polku>HIMEM.SYS
```

RAMDRIVEN käyttöä ei suositella.

Verkot

Mikäli 80286-kone on liitetty työryhmäverkkoon, sen LANMAN.INI-tiedostoa tulee muokata siten, että LAN Manager ei käytä HMA:ta. Tämä tapahtuu kirjoittamalla LANMAN.INI:n (workstation)-osaan rivi:

```
himem = no
```

Mikäli kyseessä on LAN Managerin laajennettu versio, Messenger- ja Netpopup-toimintoja ei kannata käyttää. Tämä voidaan varmistaa kirjoittamalla LANMAN.INI:iin [workstation]-osaan:

```
wrksservices =
```

Yhtäsuuruusmerkin jälkeen ei siis kirjoiteta mitään. Myös verkkokortit voivat kuluttaa järjestelmän systeemi-muistia. Tällöin tulisi varmistua siitä, että ne eivät mene päällekkäin Windowsin EMS-ikkunan kanssa. Verkkokortit voivat myös hyödyntää expanded-muistia, kunhan sitä on käytettävissä ja käytetty verkko-ohjain voidaan sijoittaa EMS-muistiin. Esimerkiksi Madge I -korttia käytettäessä tämä onnistuu lisäämällä CONFIG.SYS:iin MTOKRING.SYS-rivin loppuun valitsin /MEM/EMS. Näin saadaan säästettyä perusmuistia huomattavasti.

Mikäli kyseessä on 80386-kone, jossa käytetään Windows/386:a, käynnistysvaiheessa ei ole mitään EMS-muistia, johon esimerkiksi verkon ohjaimet voitaisiin sijoittaa. Lisäksi verkkokortin ja Windows/386:n tai 386MAX/386LIMITin välille voi muodostua ristiriitoja esimerkiksi DMA:n suhteen. Tätä tulee varoa, sillä seurauksena on koko järjestelmän kaatuminen. ■

ES-opetusverkko/VGA

Verkolla yksilölliseen luokkaopetukseen

Opetusverkon ei pitäisi aiheuttaa sydämentykytystä kenellekään tietokoneavusteista opetusta antavalle opettajalle. Jos verkko on käsitteenä outo, eikä aika riitä sen toimintojen kokeilemiseen, koko verkon voi unohtaa ja hoitaa opetuksen yksittäismikroilla. Kannattaa kuitenkin ottaa se aika ja tutustua opetusverkon toimintoihin sekä sen ohjausyksikköön, josta verkon kaikki toiminnot ovat käynnistettävissä. Näin voi säästää monet juoksemiset niin ennen opetusta kuin sen aikanakin.

Ohjelmistot kerralla koko luokan mikroihin

Tavallinen opetustilanne alkaa siirtämällä opetusohjelmistot ES-opetusverkon JAKELU-toiminnon avulla oppilaiden työasemiin. Ennen siirtämistä opettaja on voinut tehdä HALLINTA-toiminnolla kaikkiin työasemiin valmiiksi tarvittavat alihakemistot. Työ tehdään vain kerran sen asemesta, että ohjelmat asennettaisiin jokaiseen mikroon erikseen. Ajansäästö on huomattava, minkä lisäksi virhetilanteet vähenevät, koska ohjelmat asennetaan kerralla kuntoon ja toimiviksi.

Ohjelmien autoexec.bat- ja config.sys-tiedostoihin vaatimat muutokset voidaan tehdä HAL-LINTA-toiminnolla kaikkiin työasemiin yhtäaikaan. Toinen tapa muuttaa työasemien järjestelmä-tiedostoja on muokata ne opettajan työasemassa kohdalleen, ja lähettää JAKELUn avulla kaikkiin työasemiin.

Opetustilanteen ohjaus tapahtuu näytönsiirron

Jokainen ryhmäopetustilanne sisältää yhteistä ja yksilöllistä ohjausta. Mikron ruudulta näytettävät asiat on helpointa esittää verkon välityksellä. Opettaja ohjaa



esimerkin avulla ryhmän tulevaa harjoitusta. Hän laittaa ensin opetusverkon KUVANSIIRRON päälle, jolloin kaikki mitä hän tekee, näkyy verkon jokaisen mikron ruudussa. Kuvansiirto voi kohdistua myös vain osaan mikroista, missä suhteessa verkko on huomattavasti kätevämpi kuin esimerkiksi siirtoheitin.

Opettaja voi käynnistää HAL-LINTA-toiminnolla haluamansa sovelluksen valmiiksi jokaiseen työasemaan. Näin oppilaiden ei välttämättä tarvitse tutustua lainkaan itse käyttöjärjestelmään, vaan käyttää pelkästään sovellusta.

Opiskelijat voivat katsoa opettajan antamia ohjeita TIEDOKSI-toiminnon avulla. Kukin oppilas vastaa itse omasta ohjeiden lukemisestaan eli voi itse valita, milloin hän tutustuu ohjeisiin vai tarvitseeko niitä lainkaan.

Kuvansiirron aikana siirron kohteena olevan työaseman näppäimistö on lukittu, sillä oppilas ei näe, mitä hän toiminnallaan omalle työlleen aiheuttaisi. Opettaja puolestaan päättää, minkä mikron näyttöä kukin katselee. Siirto voi siis tapahtua myös oppilaan mikroilta toiselle, jolloin oppilas toimii ohjeiden tai ratkaisujen antajana.

ES-opetusverkko/VGA on väline, jonka avulla oppilaitokset ja yritykset voivat tehostaa sekä opetuksen käytännön järjestelyä että itse opetustapahtumaa. Opetusverkon päätoimintoja ovat erilaiset näytön- ja tiedostojensiirrot, työasemien hallinta sekä yhteisten oheislaitteiden käyttö.

ES-opetusverkko/VGA

Tietoliikenteen opettamisessa kannattaa hyödyntää myös verkon sisäistä tietoliikennettä.

Opettaja voi seurata omalta mikroolta oppilaiden edistymistä KUVANSIIRTON, TIEDOKSI-toiminnon tai SELAILUN avulla. SELAILU näyttää oppilastyöasemien kuvaruudut opettajalle kielistudion tapaan automaattisesti vuorotellen. Opettaja voi toki ohjata SELAILUN etenemistä myös käsin.

Asennus

- rengasmainen kaapelointi
- näytönsiirron värit erillisillä kaapeilla
- parikaapeleissa näytönsiirron tahdistus, tiedostonsiirto-kanava, komentokanava ja kolme oheislaittekanavaa
- liikennöinti opetusverkkosovittimen sarjaportti 2:n kautta
- mikron oma sarjaportti 2 muutetaan 3:ksi tai pois käytöstä
- kirjoitin liitettynä palvelimen rinnakkaisporttiin 1 (tai sarjaporttiin 1 tai 3)
- muut oheislaitteet opettajan opetusverkkoyksikön sarjaportteihin

Laitteisto

- palvelin, joka on usein opettajan työasema
- MikroMikkoja, joissa on VGA näyttö
- kaapelointi: parikaapeleita ja näytönsiirron koaksiaalikaapelit
- opetusverkon sovitinkortit kaikkiin työasemiin
- opetusverkkoyksiköt kaikkiin työasemiin
- ohjausyksikkö
- rinnakkaiskirjoitin palvelimeen (myös sarjakirjoitin kelpaa)
- kolme sarjaliitännäistä oheislaitetta: modeemi, kirjoitin ja piirturi
- PS/2-hiiret.

Konfiguroinnit

- autoexec.bat:iin polku opetusverkon ohjelmahakemistoon ESSYS
- muiden kuin palvelimen config.sys:iin DEVICE=eslpt1.sys
- palvelimeen spool-hakemisto.

Lisäharjoitukset saadaan verkon palvelimelta

Opetuksen edetessä tarvitaan kenties lisäharjoituksia. Näitä voidaan varastoida verkon palvelimen massamuistiin, josta opiskelijat voivat itse käydä niitä noutamassa. PALVELUN ollessa käynnissä he voivat siirtää valmiit työnsä opettajan tarkastettavaksi palvelimen valmiiden töiden hakemistoon.

Oppilaiden tuotokset saatava myös paperille

Opetusluokassa on järkevää pitää useita tulostimia, sillä ennen pitkää kaikki haluavat tuotoksensa paperille. Hyvä yhdistelmä on verkon palvelimeen kytketty laser-tulostin ja verkon oheislaitteina olevat kirjoitin ja piirturi tai mahdollisesti kaksi kirjoitinta.

Opiskelija tulostaa palvelimen kirjoittimelle suoraan sovellusohjelman tulostuskomennolla, jolloin kirjoitinta ei tarvitse erikseen varata. Halutessaan sen sijaan käyttää oheislaitteena olevaa tulostinta, hänen on pyydettävä opettajaa yhdistämään kyseinen oheislaitte omaan mikroonsa.

Palvelimen kirjoittimelle tulostus onnistuu, kun palvelin on ohjausyksiköstä asetettu PALVELU-tilaan. Sovellusohjelmat on asennettava käyttämään rinnakkaisporttia yksi. Mitään muita valintoja ei tarvita. Palvelin suorittaa tulosteiden jonotuksen kirjoittimelle automaattisesti. Palvelun voi keskeyttää, jolloin kesken ollut tulostus suoritetaan loppuun. Muut tulostukset säilyvät palvelimen jonotushakemistossa ja tulostuvat, kun palvelin jälleen käynnistetään.

Palvelimen näytöllä on normaalisti näkyvissä kirjoittimelle jonottavien tulosteiden nimet. Nimien alkuosa kertoo, miltä työasemalta tuloste on lähtöisin; esim. ASEMA08.079 kertoo tulosteen tulevan asemalta 8. Nimien loppuosalla on merkitystä ainoastaan PALVELU-ohjelmalle.

Oheislaitteena olevaa kirjoitinta tai piirturia käytettäessä ei tarvita palvelinta. Työaseman ja kirjoittimen yhdistäminen tapahtuu ohjausyksikön näppäimien avulla. Oheislaitteelle tulostettaessa sovelluksen on käytettävä tulostusporttinaan sarjaporttia kaksi.

Opetettavista asioista ja käytetyistä sovelluksista riippuen on syytä miettiä etukäteen, mitä kannattaa tulostaa minnekin. Tämän jälkeen voi asentaa ohjelmat sen mukaisesti. Palvelimen kautta tulostavat ohjelmat asennetaan käyttämään LPT1:stä ja muina oheislaitteina olevia tulostimia käyttävät ohjelmat asennetaan käyttämään COM2:sta. Periaatteena kannattaneen pitää sitä, että grafiikka tulostetaan pääasiassa rinnakkaisportin kautta. Kirjoittimien asennus on järkevintä tehdä ohjelmiin siinä vaiheessa, kun ne asennetaan opettajan työasemaan, siis ennen niiden siirtoa oppilaiden työasemiin. Näin asennus tarvitsee tehdä vain kerran.

Tiedon on kuljettava myös luokan ulkopuolelle

Luokkaan hankittava modeemi kannattaa asentaa opetusverkon oheislaitteeksi. Korttimodeemia ei ole järkevää käyttää, sillä se on vain yhden työaseman käytössä. Oheislaitteeksi liitetty modeemi voidaan antaa ohjausyksikön valinnalla vuorollaan minkä tahansa työaseman käyttöön. Opiskelija käyttää tietoliikenneohjelmaa aivan kuin modeemi olisi kiinni hänen omassa työasemassaan. Määrittelyissä on muistettava, että liikennöinti tapahtuu sarjaportti kakkosen kautta.

Työasemiin voi liittää omia joko sisäisiä tai ulkoisia modeemeja, mutta ne ovat vain kyseisen työaseman käytössä. Jos työaseman oma ulkoinen modeemi on liitetty kaapelilla sarjaporttiin kaksi, se on ennen yhteyden ottamista reititettävä työaseman omaan käyttöön. Opetusverkon PORTTI-ohjelma tekee tämän. Reititys tehdään myös silloin, kun käytetään muita sarjaporttiin

Kirjoittimien asennus on järkevintä tehdä ohjelmiin siinä vaiheessa, kun ne asennetaan opettajan työasemaan, siis ennen niiden siirtoa oppilaiden työasemiin. Näin asennus tarvitsee tehdä vain kerran.



Anne Konu vaikuttaa tätä nykyä TAO-Konu Oy:ssä, ja häneen saa yhteyden sekä Elisaa (Konu Anne FRL) että puhelimitse (90) 715 820.

ES-opetusverkon ohjausyksikkö.



Opetusverkon ohjelmat

- opettajan työasema / palvelin
 - ESLPT1.SYS
 - ESNET.EXE
 - PORTTI.EXE
 - JAKELU.EXE
 - PALVELU.EXE
 - PALVELU.PAR
 - OPAS.EXE
- oppilaan työasema
 - ESNET.EXE
 - PORTTI.EXE
 - ESLPT1.SYS

Opetusverkon toiminnot käynnistetään ohjausyksikön näppäimistä ja päätetään nollausnäppäimen avulla. Ohjausyksikössä on pieni nestekidenäyttö, joka kertoo verkossa käynnissä olevat toiminnot.

kaksi liitettyä paikallisia oheislaitteita.

Tietoliikenteen opettamisessa kannattaa hyödyntää myös verkon sisäistä tietoliikennettä. TIEDOSTO-toiminnolla yhdistetään kaksi työasemaa toisiinsa, minkä jälkeen ne voivat kommunikoida keskenään tietoliikenneohjelmien välityksellä.

Opetusverkko ei ole pelkkä työryhmäverkko

Edellä on kuvattu joukko opetusverkon toimintoja. Sen avulla voi hoitaa joustavasti opetuksen vaatimat käytännön järjestelyt. Lisäksi se on korvaamaton apu itse opetuksessa, esimerkkejä näytetyssä ja oppilaiden etenemistä seurattaessa. Mitkään näistä eivät

ole työryhmäverkon ominaisuuksia.

Työryhmäverkko jakaa yhteisiä resursseja, kuten tulostimia, massamuisteja ja tietoliikenneyhteyksiä. Myös opetusverkon välityksellä voidaan jakaa yhteisiä tulostimia, minkä lisäksi myös massamuistin jakamista on kokeiltu. Opetuksessa sille ei ole kuitenkaan vielä löydetty järkevää käyttöä. Käytännössä on havaittu, että työryhmäverkoissa käytetään pääasiassa ohjelmia, jotka sijaitsevat nimenomaan omassa työasemassa eikä työryhmäverkon palvelimella. Lisäksi opetuskäyttöön sopivia verkko-ohjelmia on kovin vähän tarjolla.

Työryhmäverkoilla on selvästi omat käyttöalueensa. Oppilaitoksissa näihin kuuluu esimerkiksi yhteisen oppilasrekisterin samanaikainen käyttö eri työpisteis-

tä. Mikäli organisaatiosta löytyy työryhmäverkon lisäksi myös opetusverkko, ne voidaan hyvin yhdistää myös toisiinsa. Tällöin sekä opetus- että työryhmäverkon kuuluvan työaseman kautta voidaan käyttää molempien verkkojen tarjoamia palveluita. Samalla mahdollistuu opetusohjelmistojen oppilaista keräämien tietojen välittäminen edelleen työryhmäverkon käytettäväksi.

Hämeenlinnan opettajankoulutuslaitoksessa ja normaalikoulussa on kokeiltu opetusta työryhmäverkkoa hyväksikäyttäen. Verkon hyvinä puolina on pidetty tulostinten yhteiskäyttöä ja posti-ohjelman palveluita. Pulmina on ollut verkko-ohjelmistojen vähäisyys sekä työryhmäverkon ylläpito, joka vaatii kokopäivätoimisen amanuenssin. ■

Joitakin toimintoja

JAKELU kopioi tiedostot valittujen työasemien haluttuihin hakemistoihin. Kohteena voi olla yksi kone, ryhmä tai luokan kaikki työasemat. Siirrettäviä tiedostoja valittaessa voidaan monivalintamerkkien avulla poimia vaikka koko hakemisto alihakemistoiineen edelleen välitettäväksi. JAKELU käynnistää opettajan työasemassa lähetysten ja opiskelijoiden työasemissa vastaanotto-ohjelman. Opettaja ohjaa lähetystä työasemaltaan.

HALLINTA yhdistää ohjausyksiköstä valitut työasemat opettajan työaseman näppäimistöön ja PS/2-tyyppiseen hiireen. Käyttäjän on syytä varmistaa, että kaikki hallinnassa olevat työasemat ovat samassa tilassa ja samassa hakemistossa ennenkuin ryhtyy ohjaamaan niitä toiminnoillaan. Hiirtä käytettäessä työasemat saa samaan tilaan viemällä kohdistimen ennen varsinaista toimintaa näytön kulmaan.

KUVANSIIRTO siirtää valitun työaseman näytön yhteen tai useampaan työasemaan. Siirrettävä kuva voi olla mistä työasemasta tahansa. Kuva siirtyy RGB-videosignaaleina, joten siirto on reaaliaikainen. Kaikki toiminta siirron lähteenä olevassa työasemassa näkyy samanaikaisesti kaikissa siirron kohteissa.

TIEDOKSI tekee kuvansiirron mistä tahansa työasemasta pelkäästään kuvansiirron kaapeliin. Vastaanottaen itse yhdistää napinpainalluksella siirron oman työasemansa näyttöön.

SELAILU tekee kuvansiirron opiskelijoiden työasemista vuorotellen opettajan työasemaan.

PALVELU käynnistetään palvelimeen aina, kun muilla työasemilla työskentelevät haluavat hakea sieltä tiedostoja tai tallentaa sinne tiedostoja. Tällöin opiskelija käynnistää itse omalta työasemaltaan ESNET-tiedostonsiirto-ohjelman. ESNETissä on muutama kommento, joilla voi lähettää ja vastaanottaa tiedostoja. Palvelin on kaikkien opiskelijoiden käytössä yhtäaikaaisesti. PALVELU-ohjelma toimii edellä kuvattuna tiedostopalvelimena, mutta samalla myös kirjoitinpalvelimena.

Uusia työasemia uusiin tarpeisiin

Työryhmäverkot ja avoimet tietojärjestelmät ovat laajentaneet mikrotietokoneen käsitettä. Vaikka kyse ei ole suinkaan paluusta keskuskoneaikaan, kaikkea ei tarvitse tehdä enää omassa koneessa, vaan osa toiminnoista voidaan siirtää verkon palvelimille. Näin ne voidaan toteuttaa paitsi edullisemmin, myös turvallisesti. Samalla päästään parantamaan myös avointen järjestelmien ongelmallista tietoturvaa ja pystytään suojautumaan entistä paremmin viruksia vastaan.

MikroMikko 4 m330 verkon työasema

MikroMikko 4 m330:n suorittimena toimii Intelin 80386SX, jonka suorituskyky riittää kaikkiin olemassa oleviin mikrosovelluksiin. Perusmuistia on vähintään yksi megatavu, ja se on laajennettavissa 16 megatavuun saakka.

MikroMikko 4 m330:n erikoisuutena on se, että se ei sisällä lainkaan levyke- eikä kiintolevyasemia. Näin sen hallinta on verkkokäytössä helppoa, sillä ohjelmistojen ja tiedostojen ylläpito voidaan keskittää. Arkoja tietoja ei voida kopioida verkosta ulos, eikä omien mahdollisesti virusten saastuttamien ohjelmistojen tuominen verkkoon ole mahdollista. Rajoitetuilla käyttöoikeuksilla voidaan estää myös tietojen tuhoaminen tai muokkaaminen.

Käyttöjärjestelmä ja sovellusohjelmat ladataan suoraan työryhmäverkon tiedostopalvelimelta. Lisäksi työasema voi käyttää palvelimen levykapasiteettia työtilana, jos sovellus sellaista tarvitsee.

Kaikki muutokset ja päivitykset hoidetaan keskitetysti verkkopalvelimelta ja varmuuskopiointi tapahtuu verkon kautta. Tällainen dynaaminen järjestelmä mahdollistaa myös sen, että käyttäjä voi hallita työasemaansa oman makunsa mukaan joko MS-DOS- tai MS OS/2 -käyttöjärjestelmän avulla.

Myös oheislaitteet ja tulostus on keskitetty palvelimille eli verkkoon ei jää ylimääräisiä aukkoja



MikroMikko-laitesarja täydentyy uusilla työasemamalleilla. MikroMikko 4 m330 on suunniteltu verkkokäyttöön eikä sisällä lainkaan massamuistia. MikroMikko 4 m356 on sen sijaan tarkoitettu selviämään myös yksin vaativistakin mikrosovelluksista.

esimerkiksi työasemiin kytkettyjen omien modeemien kautta.

Työasema on helppo asentaa, koska siinä ei ole liikkuvia mekaanisia osia, eivätkä sen oheislaitteet ja ohjelmistot vaadi erillistä asennusta. Se kuluttaa vähemmän virtaa ja vaatii myös vähemmän huoltoa kuin tavalliset henkilökohtaiset työasemat.

Lisäturvaa

MikroMikko 4 m330:n turvallisuutta lisää kaikille nelosille tyypillinen kolminkertainen salasanasuojaus, joka mahdollistaa salasanan määrittämisen paitsi itse järjestelmälle myös näppäimistölle ja käyttäjälle. Laitekotelo voidaan lisäksi lukita, eikä itse keskusyksikön koteloa voida avata lainkaan.

Koska massamuistit ja tuulettimet puuttuvat, MikroMikko 4 m330 on pieni ja hiljainen eli se

on helppo sijoittaa minne tahansa. Näppäimistö on kevyt ja herkkä.

MikroMikko 4 m330:ä on saatavana sekä Ethernet- että Token Ring -ympäristöihin. Yhteensopivuus MS LAN Manager 2.0 Remote Initial Load (RIPL) ja IBM Remote Load -ohjelmien kanssa takaa käytettävyyden myös IBM Token Ring -ympäristössä.

MikroMikko 4 m356 tehoa sovelluksiin

MikroMikko 4 m356 on MikroMikko-laitesarjan tehokkain 386-pohjainen työasemamalli. MikroMikko 4 m356 pohjautuu 33 megahertsin kellotaajuudella toimivaan Intel 80386-suorittimeen eli se on kolmanneksen nopeampi kuin useimmat muut, tavallisesti 25 megahertsin kellotaajuudella toimivat 80386-pohjaiset mikrot.

MikroMikko 4 m330 ja MikroMikko 4 m356

Massamuistittoman MikroMikko 4 m330:n kaikki perustoiminnot on integroitu peruslevylle. Se merkitsee luotettavuutta ja asennuksen ja ylläpidon helppoutta. Kuvassa koneelle pitää seurata m330:n tuotepäällikkö Priitta Tasanko.

Arkoja tietoja ei voida kopioida verkosta ulos, eikä omien, mahdollisesti virusten saastuttamien ohjelmistojen tuominen verkkoon ole mahdollista.

MikroMikko 4 m356:ssa voidaan käyttää sekä MS-DOS- että MS OS/2 -käyttöjärjestelmää. Parhaita käyttöalueita ovat suurta laskentatehoa vaativat sovellukset eli esimerkiksi julkaisuohjelmat, tietokoneavusteinen suunnittelu, vaativat Windows- ja Presentation manager -sovellukset sekä laajat taloudelliset laskelmat. Lisäksi se soveltuu myös palvelintehtäviin eli esimerkiksi työryhmäverkon gateway/bridge-asemaks.

Laitteessa on neljä sarjaliitintä, rinnakkaisliitintä ja PS/2-hiiriliitintä. Vapaana on kolme

korttipaikkaa esimerkiksi muistinlaajennuksia silmälläpitäen.

Näppäimistö on uusi standardin mukainen Nokia 102-2, joka painaa vain hiukan yli kilon.

Käsittely-yksikölle on saatavana lattialajusta tai työpöydän sivuun kiinnitettävä teline. Käsittely-yksikössä on lisäksi kotelolukko ja rengas turvaketjulle.

Nopeutta

MikroMikko 4 m356:n keskusyksikön tukena on 32 kilotavun väli-muisti. Lisäksi suorittimen rinnalle voidaan asentaa 33 megahertsin kellotaajuudella toimiva 80387-aritmetiikkasuoritin, jonka asemasta voidaan käyttää myös Weitekin 3167-aritmetiikkasuoritinta.

Kiintolevy ja muisti on mitoitettu riittämään ahkerallekin käyttäjälle. Työmuistia on vaihtoehtoisesti 4, 8 tai 16 megatavu yhdellä muistikortilla. Nopealla SCSI-ohjaimella toimivan kiintolevyn kapasiteetti on 100 tai 200 megatavu. Lisäksi MS-DOS-ympäristöön on saatavana myös sisäinen nauhavarmistin.

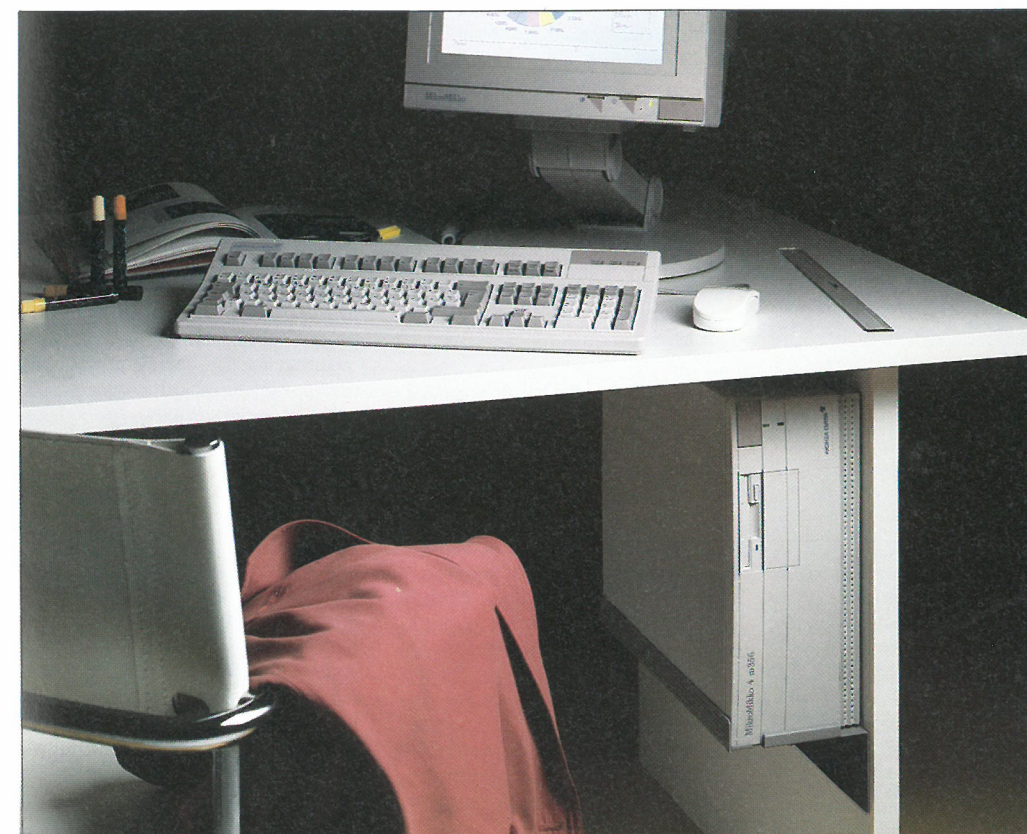
Näyttäviä näyttöjä

Kumpikin uusi työasema käyttää samoja VGA-näyttöjä kuin muutkin MikroMikko 4 -laitesarjan mallit. Tarjolla on 14 tuuman värinäyttö DU 146 sekä 15 tuuman mustavalkoinen positiivinäyttö DU 151, jotka kummatkin ovat säteilemättömiä, teräväkulmaisia ja antistaattisia.

Uusiin näyttöihin kuuluu myös matala jalusta, jonka varassa niitä voi kääntää vaakasuunnassa 40 astetta sekä pystysuunnassa alaspäin viisi ja ylöspäin 20 astetta. Jos tämä ei vielä riitä, hankitaan ergonominen niveljalusta. Sen avulla näytön korkeutta voidaan säätää pystysuunnassa 17 senttiä. Näin työasema kyetään sijoittamaan sekä istumakorkeuden että valaistuksen kannalta ihanteellisesti.

MikroMikkoihin saa arkiharmaan lisäksi myös punaisia, sinisiä ja lilanväreisiä näyttöyksiköitä. Värilliset näyttöyksiköt toimitetaan aina ergonomisen niveljalustan kanssa.

MikroMikko 4 m356 ja pöytäteline sekä ergonominen näytön niveljalusta.



Uusia lasertulostimia

LaserJet IIP ja LaseJet III ovat uusia sekä sarja- että rinnakkaisliitännällä varustettuja 300 dpi:n laserkirjoittimia, joista kummastakin löytyy vakiona 14 kirjasintyyppiä.

LaserJet IIP on tarkoitettu henkilökohtaiseksi tulostimeksi, ja se mahtuu ahtaisiin tiloihin; pöytäpintaa tarvitaan vain 350x405 mm.

LaserJet III on sen sijaan jo järkevämmän luokan tulostin. Se tulostaa tuplasti nopeammin kuin pikkuveljensä (8 sivua/min.), kiertää tekstiä sekä grafiikkaa ja piirtää tarvittaessa peilikuvia sekä kirjoittaa valkoista tekstiä mustalle pohjalle.

Kumpikin tulostin saadaan valinnaiskaseteilla sekä PostScript-että Epson FX / IBM Proprinter-yhteensopiviksi.

Lisätietoja: Leena Tanskanen, puh. (90) 5671.

Yritysviestinnältä uusi sanomanvälitys-järjestelmä

Nokia Datan Yritysviestintä alkaa myydä uutta ISMM- sanomanvälitysjärjestelmää. Yhdysvaltalaisen Comverse Technology Inc. -yhtiön kehittämällä järjestelmällä voi välittää niin kuva-, teksti- kuin puheviestejäkin. ISMM-järjestelmän (Integrated Services Message Manager) voi liittää mihin tahansa puhelinvaihteeseen tai -keskukseen.

ISMM-järjestelmän käyttäjät voivat jättää toistensa puhepostilaatikoihin viestejä, jotka vastaanottaja voi käydä kuuntelemassa mistä tahansa puhelimesta hänelle sopivana aikana salasanansa avulla. Viestejä voi myös jatkolahettää järjestelmän muille käyttäjille. Järjestelmää voi lisäksi käyttää automaattisena puhelinvastajana.

ISMM-järjestelmä ei kilpaile sähköpostin kanssa vaan täydentää sitä. Puhepostilaatikoihin voi lähettää myös telekopioviestejä. Ne vastaanottaja voi tilata katsottavakseen itseään lähinnä sijaitsevaan telekopiolaitteeseen, esimerkiksi hotellin vastaanottoon ulkoilla. Tietosuoja paranee, kun sanomat eivät loju pöydillä vastaanottajiansa odottamassa.

Tuotetta markkinoidaan ISDX- ja DIXI-vaihteiden yhteyteen. Myös yleisen puhelinverkon sovelluksista käydään keskusteluja telelaitosten kanssa.

Lisätietoja: Olof Bäckström, puh. (90) 567 2882

9- ja 24-neulaa

NokiaDataalla on joukko uusia matriisikirjoitinmalleja, joista löytyy paitsi leveä- ja kapeatelaisia myös 9- ja 24-neulaisia malleja.

DX2300- ja DX2400 ovat 9-neulaisia kirjoittimia, joiden tulostusnopeus on 270 merkkiä sekunnissa. Mikäli tulostetaan laatujälkeä (NLQ), tulostusnopeus on 543 merkkiä sekunnissa.

DL3300 ja DL3400 ovat puolestaan 24-neulaisia, joilla on kolme tulostustilaa: pikavedos (240 mki/s.), vedos (180 mki/s.) ja laatu-jälki (LQ) (60 mki/s.).

Kaikki mallit selviävät paitsi jatkolomakkeesta myös yksittäisarkeista ja sisältävät sekä sarja- että rinnakkaisliitännän. Valittavissa olevissa toimintotiloissa on sen sijaan eroja, sillä DX-mallien kanssa yhteisten Epson FX-80:n, IBM Graphics Printerin ja IBM Proprinterin lisäksi DL-mallit voivat toimia myös Diablo API ja Fujitsu DPL241 tiloissa.

Lisätietoja: Leena Tanskanen, puh. (90) 5671.

Uusi MikroMikko opetukseen

MikroMikko ES m216 on Nokian uusin erityisesti kouluopetukseen kehitetty malli. Laite edustaa ES-tuotantolinjaa (Educational Systems), joka on suunniteltu vastaamaan juuri koulumaailman tarpeita perusopetuksesta korkeakouluun.

MikroMikko ES m216:ssa on panostettu erityisesti käytön-oton sujuvuuteen. Kiintolevyllä on asennettu valmiiksi MS-DOS 4.0-käyttöjärjestelmä ja suomenkielinen Windows-käyttöliittymä, joten laite on valmis käytettäväksi jo tullessaan. Mukana seuraa ES-lisälevyke, johon sisältyy muun muassa Windowsin käyttöä helpottavia tiedostoja.

Laite on tarpeen tullen laajennettavissa neljän vapaan korttipaikan ansiosta. Myös 80286-suoritin voidaan vaihtaa tehokkaampaan.

AT-tasoinen MikroMikko ES m216 soveltuu käytettäväksi yksittäisenä tai opetusryhmän työasemana, verkon palvelimena ja opetushallinnossa. Se täyttää opetusviranomaisien koulumikroille asettamat vaatimukset.

Mikrotietokoneet ovat yleistymään päin myös peruskoulun alasteen opetuksessa. Tähän mennessä kouluissa on noin 15 000 Nokian toimittamaa mikkoa.

Lisätietoja: Ari Hakkarainen, puh. (90) 567 4028.

Mathilde tulostaa vauhdilla

Nokia Datan Suomessa edustama Bull Peripheriques on tunnettu nopeista massatulostimistaan. Malleja on tällä hetkellä kaksi: MP5050- arkkitulostin sekä jatkolomakkeelle tulostava MP6090. 2-sivutulostin.

MP5050 kykenee tulostamaan 50 sivua minuutissa ja noin 400000 sivua kuukaudessa, kun MP6090.2:n vastaavat lukemat ovat 90 sekä noin 600000. Tulostusmenetelmänä molemmat käyttävät nk. magnetofonista siirtotulostusta. Tulostustarkkuus on 240x240 pistettä neliötuumalla.

Tulostuksen muotoilu tapahtuu Windows-ympäristössä toimivan Transgen II -ohjelmiston avulla, joka koostuu edelleen Genprint- ja Genmenu-ohjelmista. Genprintin avulla Mathilde-tulostimille luodaan valmiita lomakepohjia, jotka välitetään edelleen Genmenulle, joka tuottaa varsinaisen ohjausparametrit sisältävän ohjauslevykykeen.

Varsinaiset tulostettavat tiedot voivat tulla myös suoran kanavaliitännän kautta eri tyyppisiltä mini- tai suurtietokoneilta.

Lisätietoja: Ari Hirvimäki, puh. (90) 1241.

Tandemin vikasetoinen Unix-järjestelmä

Nokia Datan tuotevalikoima laajenee Tandem Integrity S2-järjestelmällä. Se on markkinoiden ensimmäinen vikasetoinen ja jatkuvakäyntinen Unix-järjestelmä, jonka etuna on RISC-tekniikan tuoma tehokkuus.

Integrity S2 -järjestelmien potentiaalisia asiakkaita ovat ennen kaikkea teleyritykset ja valtion hallinto. Ensimmäisistä Suomeen toimitettavista Integrity S2 -järjestelmistä on tehty sopimukset Helsingin puhelinyhdistyksen ja valtioneuvoston kanslian kanssa.

RISC-tekniikka hyödyntävässä Integrity S2 -järjestelmässä on kolimodulaarinen rinnakkaisvarmennusarkkitehtuuri. Sen suorituskyky on 10–12 MIPSia. Käyttäjiä järjestelmässä voi olla 70–100.

Aloituskokoonpanon hinta on noin miljoona markkaa.

Lisätietoja: Jussi Jämiä, puh. (90) 1241.

Ohjaustyöasema töiden suunnitteluun

Ohjaustyöasema on töiden suunnitteluun ja resurssien ohjaukseen kehitetty ohjelmisto. Järjestelmällä hallitaan resurssien kohdistus työn eri vaiheille sekä kuormitus tilanne. Töiden etenemistä voi seurata vaihteittain ja tulostaa ohjausdokumentteja.

Järjestelmä perustuu älykkäseen kuormitustauluun, joka vastaa periaatteeltaan manuaalisia seinätauluja. Tuotannon kokonaistilanne sekä yksittäisten töiden ja resurssien tilanne näkyvät graafisina esityksinä ruudulla. Tietoja muokataan hiiren avulla.

Windows-pohjainen Ohjaustyöasema toimii MikroMikko 4:ssä joko itsenäisenä tai Primas-tuotannon ohjausjärjestelmän kanssa. Ohjelmiston on suunnitellut Dialogos-Team Oy.

Lisätietoja: Nokia Datan myyntineuvottelijat puh. (90) 1241

Mikko 300 -sarja markkinoille

Mikko 300 -sarjan laitteet ovat monen käyttäjän Unix-koneita, jotka ovat yhteensopivia Mikko 200 Sarjan laitteiden ja ohjelmistojen kanssa. Tällä hetkellä sarjaan kuuluu kaksi laitetta, joista Mikko 300 m67 kykenee 22 MIPSin tehoineen palvelemaan yli sataa käyttäjää yhtä aikaa. Mikko 300 m66, joka voidaan laajentaa m67:ksi, selviää enimmillään noin sadasta samanaikaisesta käyttäjästä.

Muita Mikko 300 m67:n mielenkiintoisia tunnuslukuja ovat muistiväylän nopeus (120 Mt/s), oheislaitte- eli VNE-väylän nopeus (22 Mt/s) sekä levyohjainten nopeus (6 Mt/s). Levyohjaimet kykenevät tekemään useita samanaikaisia levyhakuja jopa samalta levyltä.

Lisätietoja: markkinointipäällikkö Rami Raulas ja tuotepäällikkö Jukka Riivari, puh. (90) 5671.

MikroMikko 4 m216

MikroMikko 4 -sarja laajenee myös 80286-koneiden puolelle uuden MikroMikko 4 m216 -mallin myötä. 4 m216:n kelloaajuus on 12,5 megahertsiä, ja työmuistia on vaihtoehtoisesti joko 1,2 tai 4 magatavua.

Massamuistilaitteina on joko 20 tai 40 megatavun kiintolevyasema sekä 3,5 tuuman levyasema 1,44 megatavun kapasiteetilla. Liitäntäpuolella löytyy kaksi sarjaliitintä, rinnakkaisliitintä sekä PS/2-hiiriliitintä.

Vapaita korttipaikkoja on neljä.

Lisätietoja: tuotepäällikkö Anneli Martonen, puh (90) 5671.

MikroMikot 486-kantaan

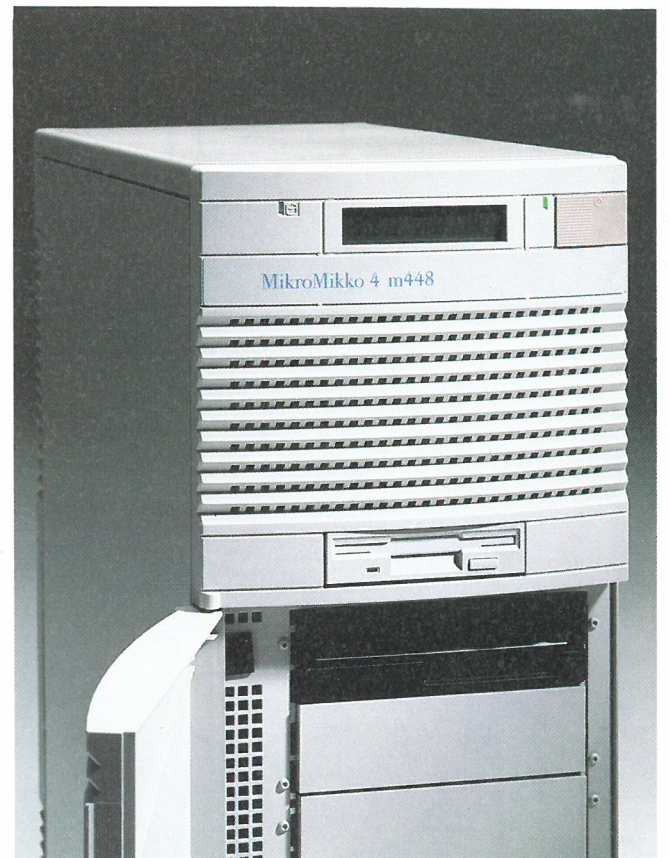
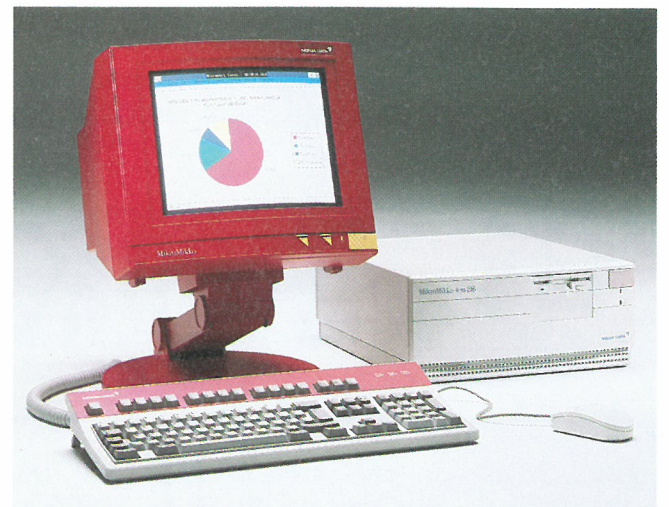
Nokia Data on tuonut ensimmäisten joukossa markkinoille oman 80486-suorittimen ympärille rakennetun mikronsa, MikroMikko 4 m488:n. Tästä laitteesta on jo ehditty huhuta pitkin syksyä, mutta virallinen julkistus tapahtui vasta Hannoverin Cebit-messuilla. Toimitukset alkavat toukokuussa.

MikroMikko 4 m488:n 80486-keskussykkinen sykkii 25 megahertsin taajuudella. Sen tukena on sisäinen liukulukusuoritin sekä 8 kilotavun kokoinen välimuisti. Työmuistin määrä vaihtelee varustuksesta riippuen 8–32 megatavun välillä.

Oheisissa löytyy kiintolevy (1 tai 2 kertaa 150, 300 tai 660 megatavua), 3,5 tuuman levykeasema (1,44 Mt), neljä sarjaliitintä, rinnakkaisliitintä sekä PS/2-hiiriliitintä. Lisäksi järjestelmään voidaan lisätä 5,25 tuuman leppuasema sekä sisäinen nauhavarustusasema.

Kahdeksan megatavun muistilla varustetusta versiosta löytyy neljä vapaata lisäkorttipaikkaa.

Lisätietoja: tuotepäällikkö Anneli Martonen, puh (90) 5671.



Tasainen kärki

Viime Softarista on jo aikaa, mutta kerrataan vielä sen verran, että lukijakyselyn tulokset antoivat tällä kertaa yllättävän tasaisen kärjen. Tekstinkäsittelyohjelmia käsittelevä 'Viitosten kaksintaistelu' sijoittui toki odotetusti keuhille, mutta myös Windowsin muistinkäytön satoja valottanut 'Mitä mihinkin ja miksi' sekä Teknillisen Korkeakoulun tietoverkkoja käsitellyt 'Työasemasta superiin' saivat oman osansa. Myös lukijanovelli eli 'High Score' oli miellyttänyt useita vastajia.

Ja tällä kertaa paidat yrittävät lähteä seuraaviin osoitteisiin: Simo Sauni, Heino Kärkäinen, Jarmo Immonen, Timo Virtanen sekä Pentti Tikala. Toivottavasti toimitus saa ne tällä kertaa liikkeelle vähän viime kertaista ripeämmmin (mitkäs viisi pakettia tuossa nurkassa...)

00003 HELSINKI

Sopimus 00380/166

VASTAUSLÄHETYS

PL 780/10

Softari
Nokia Data

Nokia
maksaa
postimaksun

Palvelukortti Softari 1/90

Softari-kysely

(Anna arvosana (—, 0 tai +) jutust:
haluatko aiheesta lisätietoja)

	Sivu	—	0
Pura ja kokoa	2	☐	☐
Uusi tapa luoda verkko-sovelluksia S2500, Tiimi ja Mikot avoimesti	3	☐	☐
ytteen	6	☐	☐
DPS 6000 hajautuviin tietojärjestel- miin	8	☐	☐
Enemmän irti Window-sovelluksista	10	☐	☐
Värikkäät nestekidenäytöt tulevat jo tänä vuonna	16	☐	☐
Ergonomia ruudun molemmin puolin	20	☐	☐
Learning Ada — Part 7	24	☐	☐
Moni Windows pyörii vajaan kahdeksan — turhaan	28	☐	☐
Verkolla yksilölliseen luokkaopetuk- seen	33	☐	☐
Uusia työasemia uusiin tarpeisiin	36	☐	☐
Tuoteuutiset	38	☐	☐
Mikkonen	40	☐	☐

Vastanneiden kesken arvotaan viisi
T-paitaa.

Jos voitaa paitaa, kokoni on: _____

- ☐ Korjaan yllämainitut osoitetiedot.
☐ Poistan yllämainitut osoitetiedot
☐ Lisään allamainitut osoitetiedot
☐ Netin ☐ Softarin rekisteristä/rekisteriin
☐ Haluan lehden kotiosoitteeseeni

yritys _____
 osasto _____
 lähiosoite _____
 postinro _____ postitoimip. _____
 nimi _____



8/0010

MIKKONEN

