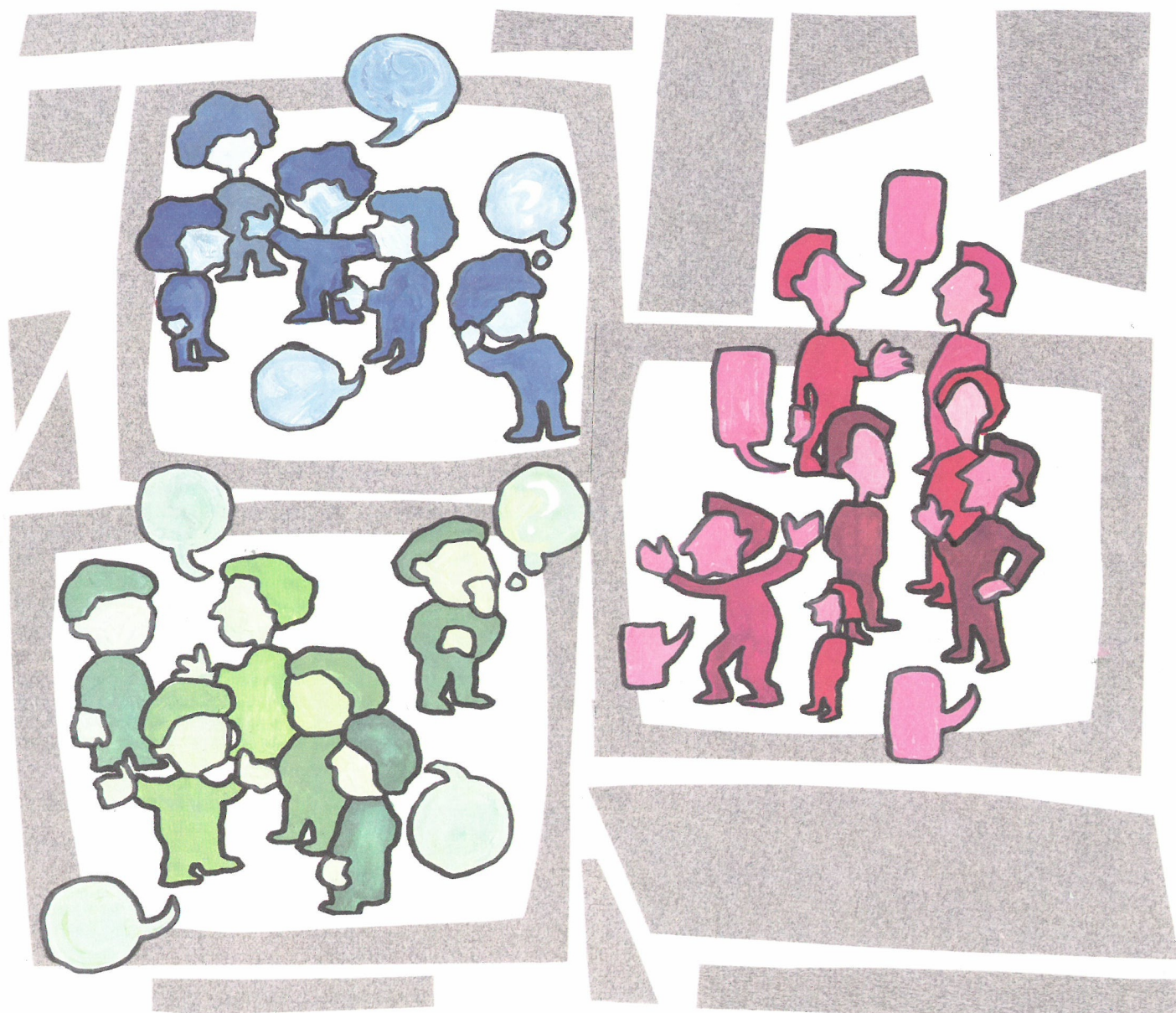


Softari

NOKIA DATAN

OHJELMISTOLEHTI 4/88



**Tuoteperheessä puhutaan
omaa kieltä**

Nokian ohjelmistolehti

Päätoimittaja:

Anne Typpö puh. (90) 567 2981

Softaritiimi:

Kari Kaipainen, Jukka Oksanen,
Hannu Markus, Tapio Sulkava,
Ossi Syrjä, Anne Typpö,
Esko Uotila, Raija Visamo

Ulkoasu ja kuvitus:

Taina Sonninen puh. (90) 567 3260

Toimituksen osoite:

Nokia Data
Softari-lehti
PL 780
00101 HELSINKI

Painosmäärä: 11 000

Kirjapaino: Vientipaino

Julkaisija:

Nokia Data

ISSN 0782-8209

Kansikuva: Taina Sonninen

Sisällysluettelo

Tiedon kulku puhuttaa	2
Tietoliikenne on avointen	
tietojärjestelmien perusta	3
Selkeyttä ohjelmisto valintoihin	4
DataPerfect taltioi tiedot	6
Joustavat yhteydet ulkomaailmaan	8
Tiimin työasemat 3270-päätteinä	10
Eicon tiedonsiirtoon	12
Modeemi korjaa tiedonsiirtovirheet	14
EMC -enemmän kuin modeemi	14
Funet yhdistää ja välittää	15
DDE siirtää tietoa Windows-ympäristössä	18
Moniajossa sen salaisuus	21
Ruori kauppiaan avuksi	24
Learning Ada Part 2	26
Omia sovelluksia Windows-ympäristöön	29
Muistin sokkeloissa	31
CTB-testaustyökalu ANSI C:lle	33
"Riisiteessa"	35
Tuoteuutiset	37
Pakina	40

Tiedon kulku puhuttaa

Tietojärjestelmien avoimuus ja yleisesti tunnustetut standardit tarjoavat organisaatioille mahdollisuuden tehokkaaseen tietojenkäsittelyyn. Avainkysymyksiä ovat tietokoneverkot ja tietoliikenne.

Tietokonelaitteiden liitettävyyks on lähiverkkojen kantavia ajatuksia. Nykyisin jo joka kolmas Nokia Datan myymä mikro liitetään verkkoon. Työryhmäverkoissa käyttäjät jakavat tehokkaasti verkon palvelut kuten esimerkiksi tulostimet, tiedostot ja tietoliikenneyhteydet. Kun verkkoja voidaan liittää toisiinsa ja näitä taas keskuskoneisiin, syntyy hyvin monipuolinen koko organisaation laajuinen järjestelmä.

Tietokoneverkkojen avoimuus on mahdollista osin teollisuusstandardien ja osin kansainvälisten standardointijärjestöjen ansiosta. IEEE-järjestöjen projektiryhmä 802 on standardoinut tiedonsiirron tärkeimmät alueet. Avoimuus takaa tietokonetuotteiden korkean laadun, koska kilpailutilanteessa voi aina valita parasta.

Työryhmäverkkojen työasemien miellyttävimpiä piirteitä on ergonomia. Monipuolisten verkkopalvelujen lisäksi älykäs työasema hoitaa käyttöliittymän ergonomisesti. Graafinen liit-
tymä ja esimerkiksi hiiri ovat omiaan lisäämään käyttömukavuutta. Perinteinen pääte ei tähän pysty. Tietokonetuotteet tarjoavat lisäksi omia ohjelmistopakettejaan käyttöympäristöratkaisuihin, kuten esimerkiksi Nokia Data, joka on koonnut Tiimi-työryhmään Windows- ja Perfect-tuoteperheet.

Verkkokaapelointia voi verrata tieverkostoon. Vasta protokollat ja ohjelmistot muodostavat liikennesäännöt. Tämä mahdollistaa verkkojen hallitun kommunikoinnin. Esimerkiksi Tiimi-työryhmäverkoista voidaan kommunikoida NAS-suurkoneiden kanssa 3270-yhteyksien avulla ja X.25-yhteyksien avulla pakettiverkkojen kautta. Bridge- ja Eicon-tuotteet monipuolistavat verkkojen liikennöintimahdollisuuksia ja liitettävyyttä.

Tietoliikenne on avointen tietojärjestelmien perusta

Ennen tietokone oli ylhäisessä yksinäisyydessään toimiva laite, jota valkotakiset alan salatieteisiin perehtyneet spesialistit käyttivät. Nykyisin tietokone on tuttu ja yhä suurempi ja suurempi osa ihmisistä käyttää työssään sen palveluja. Tässä palvelujen hajauttamisessa on avaintekijänä tietoliikenne. Sen avulla voidaan tietokoneen palvelut tuoda käyttäjien luo. Riippumatta siitä, onko tietokonesovellus keskitetty vai hajautettu, sijaitseeko tietokone laite naapurihuoneessa vai toisella paikkakunnalla, käyttäjä näkee sovelluksen samalla tavalla.

No aivan näin ruusuinen ei tilanne useinkaan vielä ole. Mutta kehitys kulkee nopeaa tahtia kohti tilannetta, jolloin erilaiset ja erimerkkiset laitteet keskustelevat sujuvasti ja vaihtavat tietoja keskenään. Juuri tätähän avoimuus tietojärjestelmien kohdalla tarkoittaa.

Avoimien tietojärjestelmien tietoliikenteelle asettama vaatimus on, että käyttäjä voi omalta työasemaltaan, oli se sitten mikro, pääte, tai tekninen työasema, olla yhteydessä kaikkiin niihin tietokoneisiin, joiden sovelluksia hän käyttää.

Koska varsinkin suurissa organisaatioissa on useimmiten useiden valmistajien tietokonelaitteita, avoimuuden vaatimus on myös melkoinen haaste. Paras ratkaisu tähän haasteeseen on standardien hyväksikäyttö. Oetaan esimerkiksi kolme valmistajaa A, B ja C, jotka kaikki ovat ratkaisseet tietoliikenteensä eri tavoin. Tällöin tarvitaan erilliset sovitimet A:n ja B:n, A:n ja C:n sekä B:n ja C:n väliseen kommunikaatioon. Jos nyt vielä hankitaan laitteita neljänneltä valmistajalta, tarvitaan kolme eri sovitusta lisää. Mutta jos kaikki valmistajat tukevat samaa standarditapaa hoitaa tietoliikenne, sovitimia ei tarvita, vaan kaikki neljän valmistajan laitteet voivat kommunikoida toistensa kanssa suoraan.

Lähiverkoissa, joilla yleisesti tarkoitetaan suppealla alueella toimivia suuren siirtonopeuden omaavia tietoliikenneverkkoja, on käytännössä muodostunut kaksi standardiratkaisua. Ethernet-verkot sopivat monitoimijaympäristöön, erityisen hyvin sitä tukevat UNIX-käyttöjärjestelmää käyttävät tietokoneet.

TokenRing-verkot soveltuvat varsinkin ympäristöön, jossa käytetään IBM-tietokoneita. Keskustelu jomman kumman ratkaisun teknisestä paremmuudesta ei useimmiten ole oleellista, vaan tärkeintä on miten verkko tukee käytettäviä sovelluksia. Nokia Data tukee molempia verkkoratkaisuja tasapuolisesti ja pystyy näin tarjoamaan parhaan mahdollisen ratkaisun asiakkaan tarpeisiin.

Lähiverkkoratkaisujen tuomia etuja ovat muun muassa mahdollisuus jakaa tiedostoja useiden käyttäjien kesken ja tallentaa mikron käyttäjän omia tiedostoja jollekin toiselle koneelle sekä kalliiden oheislaitteiden, kuten kirjoittimien ja modeemien, yhteiskäyttö. Tällaiset edut on usein helppo mitata esimerkiksi rahassa. Huomattavasti vaikeammin mitattavissa ovat erittäin merkittävät lähiverkon avulla saatavat hyödyt kuten työtehon nousu, käyttömukavuuden parantuminen ja ylläpidon helpottuminen. Jos esimerkiksi vasteajat tietoliikenneverkon kuormittumisesta johtuen kasvavat suuriksi, käyttäjien keskittyminen heikkenee, he turhautuvat ja tekevät virheitä. Lähiverkon tarjoama suuri siirtokapasiteetti nostaa näin työtehoa ja parantaa käyttömukavuutta.

Tietoliikenneverkko ei koskaan ole valmis. Siihen lisätään jatkuvasti uusia laitteita, niiden paikkaa vaihdellaan ja tehdään uusia määrittelyjä. Lähiverkon käytöllä voidaan usein esimerkiksi vähentää erilaisten kaapelointitöiden määrää muutostöiden yhteydessä. Kaapelin hinta on useimmiten epäolennainen, sillä suurin kustannus muodostuu kaapelointiin liittyvästä työmäärästä. Unohtaa ei toki sovi niitä haittoja, jotka kaapeloinnin aiheuttama melu ja erikoisjärjestelyt aiheuttavat työntekijöille. Juuri tästä syystä uusiin rakennuksiin useimmiten asennetaan erilaista kaapelointia runsaasti yli senhetkisen tarpeen.

Verkon ylläpitoon tarvittava työmäärä kasvaa helposti verkon koon kasvua nopeammin. Paitsi että ylläpito on työteliästä, ongelman muodostaa myös osaavien ihmisten puute. Alan kasvu on ollut niin nopeaa, ettei koulutettua työvoimaa ole saat-

vissa. Tarvittavan koulutuksen määrää voidaan pienentää käyttämällä standardiratkaisuja. Niukat asiantuntijaresurssit voidaan näin käyttää paremmin hyväksi.

Rakennettaessa useille paikkakunnille ulottuvia tietoliikenneverkkoja on myös käytettävissä standardiratkaisuja. Esimerkiksi pakettikytkentäiset data-verkot, joita Suomessa ovat Datapak ja Digipak, tarjoavat nopean ja luotettavan siirtoverkon, jonka välityksellä voi kotimaan lisäksi olla yhteydessä kymmeen muihin maihin. Useimmat valmistajat, kuten Nokia Data, tarjoavat X.25-liittymiä laitteisiinsa, joilla X.25- ja lähiverkot liitetään joustavasti toisiinsa.

Tietoliikenneverkkoja suunniteltaessa ja rakennettaessa on syytä muistaa, että ne palvelevat vuosikausia. Näin ollen kannattaa ottaa huomioon myös tulevaisuuden tarpeet. Kun verkko pohjautuu avoimuuteen ja standardeihin, on myös helpompi sopeutua tulevaisuuden mukanaan tuomiin muutoksiin ja yllätyksiinkin. ■

Simo Poikola työskentelee Nokia Datan Tuotemarkkinoinnissa tietoliikenteestä vastaavana tuotepäällikkönä.



Uffe Ekholm



Selkeyttä ohjelmistovalintoihin

Kannattaako toimistojen ohjelmistoja valittaessa kiinnittää huomiota testeissä menestymiseen, monipuolisiin ominaisuuksiin, valmistajan nimeen vai kenties yksittäisiin toimintoihin? Näihin kysymyksiin vastaa Tiimin tuoteperheajattelu.

Markkinoilla on todella runsaasti toimistokäyttöön tarkoitettuja ohjelmistoja, joista varmasti jokaisessa on omat hyvät puolensa. Runas tuotevalikoima saa kuitenkin ostajan aivan ymmälleen. Valitako viimeisimmissä testeissä ominaisuuksiltaan etevimmäksi osoittautunut ohjelmisto vai vähemmän ominaisuuksia sisältävä kilpailija?

Nokia Data mietti toimistoille tarjoamiensa sovellusten valikoidaan sekä asiakkaan että kehitystyön kannalta. Mietinnän tuloksena Tiimi-työryhmään määriteltiin kaksi toimistosovellusten kokonaisuutta, tuoteperhettä.

Tuoteperhe on sisäisesti eheä, toimistojen yleisimmät toiminnot sisältävä ohjelmistokokonaisuus, joka toimii ohjenuorana toimiston sovellusvalikoimaa hankittaessa ja kasvatettaessa.

Sisäisesti eheä kokonaisuus

Koska markkinoilla on jo valtava määrä valmisohjelmistoja, tuoteperheiden sovellusten valintaa ei tehty yksittäisten ominaisuuksien tai nimen perusteella. Pääpaino asetettiin synergiavaikutuksiin, jotka saadaan ohjelmistoja yhdistelemällä.

Kaikkein perustavin ajatus tuoteperheitä mietittäessä oli mahdollisimman yksinkertainen tiedonsiirto sovelluksesta toiseen. Tarkoitus on, että tuoteperheen ohjelmistot ovat yhteensopivia toimintaympäristön ja muiden saman tuoteperheen sovellusten kanssa.

Ajattelun lähtökohtana on myös nähty yhtenäinen tapa käynnistää ohjelmistot ja mahdollisimman yhtenäinen tapa käyttää ohjelmistoja. Tämä madaltaa ohjelmistojen oppimiskynnystä merkittävästi ja lyhentää myös oppimisaikaa.

Yleisimmät toiminnot hallinnassa

Yleisellä käsitteellä ei voida ratkaista kaikkia erikoistapauksia. Siksi tuoteperheä käsitettä luotaessa määriteltiin viisi yleisintä toimiston toimintoa, jotka otetaan mukaan määritelmään.

Tuoteperheen ohjelmistoilla hoidetaan tekstin kirjoittaminen, laskelmien tekeminen, erilaisten kortistojen ylläpito, kaavioiden piirtäminen ja viimeistelyjen asiakirjojen tuottaminen.

Ajattelutapana tuoteperhe on kuitenkin avoin. Yrityskohtaisia erityisohjelmistoja voidaan lisätä tuoteperheeseen, kunhan ne vain sopivat yhteen muiden ohjelmistojen kanssa.

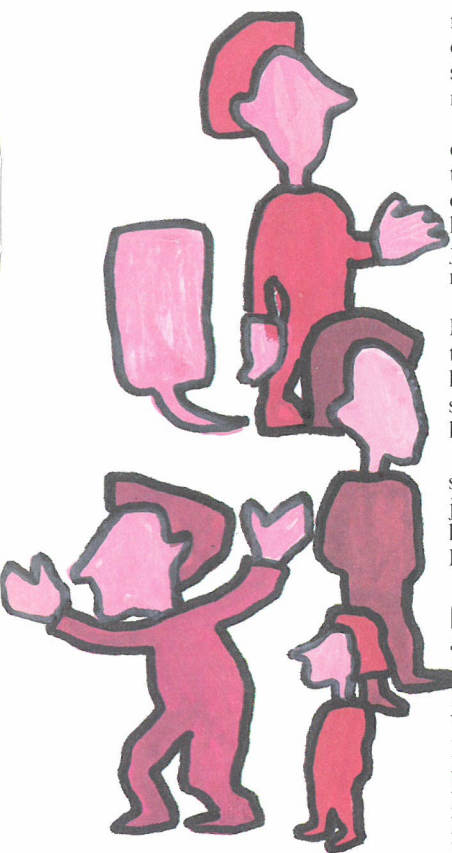
Kaksi erilaista toimintaympäristöä

Toimintaympäristö on tuoteperheen perustuote, joka yhtenäistää ohjelmistojen käynnistyksen ja käytön sekä tarjoaa sovellusten välisen tiedonsiirron mekanismit. Lisäksi se tarjoaa muutamia yleisiä apuohjelmia kuten laskin ja kalenteri.

Nokialainen toimisto toimii Windows- ja Perfect-ympäristöissä, joissa tekstit tuotetaan tekstinkäsittelyohjelmistolla, laskelmia tehdään taulukkolaskentaohjelmistolla, kortistoja ylläpidetään tiedonhallintaohjelmistolla, grafiikkaa tehdään esitysgrafiikkaohjelmistolla ja viimeistellyt dokumentit tuotetaan julkaisuohjelmistolla.

Tuoteperheissä nimenomaan toimintaympäristöt tarjoavat palveluja, jotka ovat käyttäjälle näkymättömiä, mutta silti oleellisen tärkeitä miellyttävässä henkilökohtaisessa tiedonkäsitelyssä.





mistoa Exceliä. Microsoftin Excelissä yhdistyvät helppokäyttöisyys, runsaat ominaisuudet ja näyttävä grafiikka.

Yksinkertaiset kortistot ylläpidetään käyttämällä Excelin kortistointiominaisuutta. Kortistot ovat siten yhdistettävissä taulukkolaskentaan. Myös lomakepohjainen tiedonsyöttö ja selailu on mahdollista.

Graafiset esitykset tuotetaan Micrografx:n Designer-ohjelmistolla. Designer soveltuu kaikenlaiseen piirtämiseen aina teknisistä kuvista näyttävien tekstikalvojen tuottamiseen.

Laadukkaat dokumentit, joissa yhdistellään kuvia, numeroita ja tekstejä näyttävään muotoiluun, tehdään Aldusin PageMaker-ohjelmistolla.

Perfect-tuoteperhe

Perfect-tuoteperhe rakennetaan WordPerfect Corporationin Library- tuotteen ympärille. Se tarjoaa tiedonsiirtopalveluiden lisäksi käyttäjäkohtaisten valikojen tekomahtoisuuden. Librarystä on saatavissa myös erillinen versio työryhmiä varten.

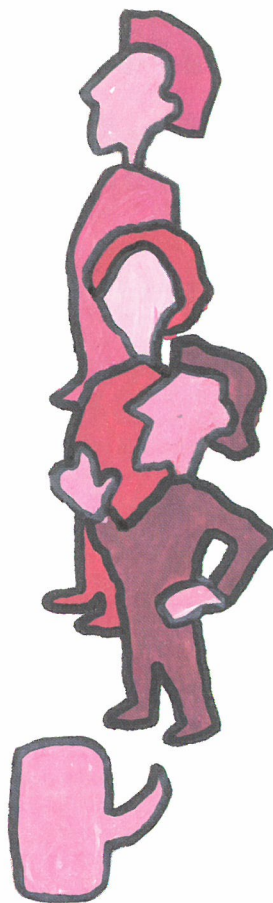
Tekstinkäsittelyyn on valittu WordPerfect, joka on markkinoiden ehdottomasti suosituin tekstinkäsittelyohjelmisto.

Laskelmien tekemisessä käytetään saman yhtiön PlanPerfect-taulukkolaskentaohjelmiä. Se on WordPerfect-käyttäjän ihanteellinen yhteistyökumppani, sillä toimintonäppäimet ovat mahdollisimman pitkälle yhtenevät ja tiedonsiirto toimii joustavasti.

Edelleen saman yhtiön DataPerfect-tuote on tarkoitettu erilaisiin tiedonhallintatehtäviin. DataPerfect on suomenkielinen kuten muutkin WordPerfect Corporationin tuotteet. DataPerfectin käyttäminen noudattaa samaa tuttua linjaa kuin muut Perfect-tuotteet.

Esitysmateriaalit tuotetaan Lotus Freelance Plussalla, joka on parhaimmillaan, kun yhdistetään tekstejä kuviin ja tuotetaan kalvoja. Freelance Plussalla voi myös tehdä yritysgrafiikkaa.

Viimeistellyt asiakirjat tehdään Ventura Publisher -ohjelmistolla, joka soveltuu hyvin myös pitkien asiakirjojen tuottamiseen. Asiakirjat saadaan muotoiltua käyttämällä erilaisia tyyli-määrittäjiä.



Windows-tuoteperhe

Windows-tuoteperhe rakentuu nimensä mukaisesti Microsoftin Windows-tuotteen ympärille. Windows on yhtenäinen käyttöliittymä, jota voidaan käyttää sekä näppäimistön että hiiren avulla.

Windows-tuoteperheen näkyvimpiä piirteitä on graafinen käyttöliittymä ja ikkunointi. Ikkunoinnin avulla voidaan käyttää useita sovelluksia yhtäaikaan ja siirtyä sovelluksesta toiseen osoittamalla haluttua sovellusta hiirellä.

Tekstien tuottamiseen on valittu Microsoftin Word-ohjelmisto, joka on saatavissa myös suomenkielisenä. Se soveltuu käytettäväksi Windows-ympäristössä huolimatta siitä, että se ei ole varsinainen Windows-ohjelmisto, sillä se pystyy muun muassa vaihtamaan tietoa toisten sovel-lusten kanssa Windowsin Clip boardin avulla.

Laskelmat tehdään käyttämällä markkinoiden ehdottomasti parasta taulukkolaskentaohjel-

Nokia Datan panos

Nokia Datan Tuotekehitys keskittää ohjelmisto-osaamisensa tuoteperheiden tuotteisiin. Tarkoituksena on erottautua ohjelmistojen välittäjistä tarjoamalla todellisia ratkaisuja. Näin Nokia Datasta löytyy asiantuntemusta aina ohjelmistovalinnasta käytön tukemiseen.

Kun Nokia Data on vakiinnuttanut tuotevalikoimansa, se tarjoaa asiakkaille perustuotteiden lisäksi vielä ylimääräisiä palveluja. Näistä ylimääräisistä palveluista kerrotaan myöhemmin tarkemmin, mutta esimerkkeinä voidaan mainita käyttöliittymän yksinkertaistaminen ja valmiit tekstinkäsittelypohjat.

Vakiinnuttaminen ei kuitenkaan tarkoita paikallaan pysymistä. Nokia Datassa tutkitaan jatkuvasti uusia ohjelmistoja, jotka voidaan sisällyttää tuoteperheisiin joko tuoteperhetuotetta täydentävänä tuotteena tai mahdollisesti jopa korvaamaan jonkin tuotteen. ■



Serjafoto

Tapio Sulkava vastaa Productivity Tools -ryhmästä, joka suorittaa ohjelmistovalinnat Nokia Datan Tuotekehityksessä.

DataPerfect taltioi tiedot

↑Taulu-F7	Lisää-F9	Muokkaa-F6	Selaa-↑
—KULUNKI.OSA-3— Osasto: Hallinto Alkusaldo: 350,00 Nykysaldo: 422,00 Tilikauden menot: 0,00 Tilikauden tulot: 72,00 Tulot ☐ Menot ☐ —KULUNKI.YHT-4— Tili: Konttoritarvike Tulot yht: 75,00 Menot yht: 268,00 Menot ☐ Tulot ☐	—KULUNKI.TUL-10— Tulotositteen numero: 0004 Päiväys: 15.03.88 Maksaja: Kalupakki Ky Määrä: 40,00 Maksun aihe: kopiopaperi Tili: ♦Konttoritarvike Osasto: ♦Hallinto —KULUNKI.MEN-6— Menotositteen numero: 0011 Päiväys: 4.09.88 Saaja: Reiman Tukku Oy Määrä: 200,00 Maksun aihe: Kirjekuoret Tili: ♦Konttoritarvike Osasto: ♦Myynti		

Utte Ekholm

**DataPerfect-tietokantaohjelmisto on parhaimmillaan
pienehköjä tietokantoja käsiteltäessä. Valtteina ovat
yhteensopivuus muun Perfect-tuoteperheen kanssa ja
suomenkielisyyys.**

DataPerfect on WordPerfect Corporationin tekemä relaatiotietokantaohjelmisto, jolla voidaan sekä luoda että käyttää tietokantoja ilman ohjelmointia. Lisäksi se on täysin suomenkielinen ja yhteensopiva WordPerfect Libraryn kanssa. Näin ollen siihen voi esimerkiksi tehdä työtä automatisoivia makroja.

DataPerfect toimii verkossa, joka tukee MS-DOS 3.0:n tiedostojen yhteiskäyttö- ja lukituskomentoja. DataPerfectin ohjelmätiedostojen on silloin oltava sellaisessa hakemistossa, jossa ne ovat kaikkien sovelluksen käyttäjien käytettävissä. Verkko-käytössä DataPerfect sallii samanaikaisesti 99 käyttäjää samassa tietokannassa, kun jokaiselle tietokannalle on luotu oma alihakemisto.

Tietokannalla työskentely jakautuu tietokannan suunnitteluun ja luontiin sekä tietokannan varsinaiseen käyttöön. Jos suunnittelija ei ole sama henkilö kuin käyttäjä, heidän on oltava tiiviissä yhteistyössä suunnitteluvaiheessa, jotta käyttäjän tarpeet tulevat huomioonotetuiksi. Tietokannan rakenteen muuttaminen sen käyttöönottamisen jälkeen voi nimittäin osoittautua hankalaksi.

DataPerfectin parhaisiin puoliin kuuluu, että sitä voidaan käyttää WordPerfect Libraryn kanssa, mikä helpottaa huomattavasti tietojen siirtämistä DataPerfectin ja esimerkiksi WordPerfect-ohjelmiston välillä. Siirrettävät tiedot kopioidaan tai leikataan WordPerfect Libraryn apupöydälle, josta ne ovat haettavissa muihin WordPerfect Libraryn ohjelmiin. DataPerfectiä voidaan kuitenkin käyttää myös erillisenä ohjelmistona ilman WordPerfect Libraryä.

Yksinkertainen käyttöä

Sovellus koostuu tavallisesti useista tauluista. Esimerkiksi "asiakas"-nimisessä tietokannas-

sa voisi olla "nimi"- ja "yritys"-taulu. Kumpikin taulu sisältää useista kentistä koostuvia tietueita. Tällä tavalla asiakkaita koskevat tiedot ryhmitellään aiheittain. Liikkuminen taulujen, tietueiden ja kenttien välillä tapahtuu toiminto- ja nuolinäppäinten avulla. Käyttäjä voi halutessaan lisätä, kopioida tai poistaa tietueita.

Kun tietueita muokataan, lista muista samassa taulussa olevista tietueista saadaan näppäimen painalluksella. Listan järjestys riippuu sillä hetkellä muokatus-ta kentästä. Jos esimerkiksi otetaan lista, kun muokataan "etunimi"-kenttää, lista esiintyy etunimen mukaisessa järjestyksessä, jos suunnittelija on näin määritellyt.

Edellä mainitussa esimerkissä "nimi"-taulussa voisi olla "yritys"-tauluun yhdistetty "yritys"-kenttä. Kun syötetään uusi tietue ei tarvitse syöttää "yritys"-kenttää, sillä DataPerfect näyttää automaattisesti kaikki jo syötetyt yritykset. Näin säästyy myös levytilaa ja aikaa, sillä esimerkiksi sadan samaan yritykseen kuuluvan asiakkaan tiedot tarvitsee syöttää vain kerran.

Tietokannasta on usein hyödyllistä tulostaa raportteja. DataPerfectin raporttiluettelosta valitaan, millainen raportti halutaan ajaa. Luettelossa on ohjelman luoma perusraportti ja lisäksi voi olla tietokannan suunnittelijan määrittelemiä raportteja. Näin saa helposti esimerkiksi valmiita kirjeitä lähetettäväksi sellaisille asiakkaille, joiden maksu on erääntynyt.

Tietokannan sisällön voi myös viedä tiedostoon tekstinkäsittely- tai julkaisuohjelmistolla muokkaamista varten.

Suunnittelua tarvitaan

Kun sovelluksen piirteet on määriteltä, ne täytyy muokata DataPerfectin ymmärtämään muotoon. Täytyy siis määritellä taulut, taulujen kentät, kenttien

muoto ja mahdolliset yhteydet taulujen välille.

Taulu voidaan määritellä näyttöön tiettyyn paikkaan. Tämä on hyödyllistä varsinkin silloin, kun taulujen välille on luotu yhteyksiä. Näin estetään tauluja menemästä toistensa päälle.

Kenttämuotoon voi liittää lisämääreitä, joilla annetaan kentälle jokin erityisominaisuus. Voidaan esimerkiksi määritellä, että johonkin kenttään täytyy syöttää jokin arvo, tai ettei kentän sisältöä voi muokata (esimerkiksi yhteenlaskettavien kenttien loppusummakenttä). Suunnittelijalla on mahdollisuus rajata kenttään hyväksyttävä arvoalue, jonka perusteella ohjelma tarkistaa syötötiedon. Arvoalueen voi liittää merkkikenttään, numeeriseen kenttään, päiväys- tai kellonaikakenttään.

DataPerfect laskee yhteen kenttien arvot tai tulostaa kenttään laskentakaavan lopputuloksen. Molemmat toiminnot ovat helposti toteutettavissa. Laskentaa käytetään lisäämään tai vähentämään kenttien arvoja. Kaavan avulla saadaan sama tulos, mutta laskentatoiminto laskee kentän arvon välittömästi uudelleen, kun laskentakentän arvoa muutetaan. Kaavan avulla voidaan kenttään syöttää tietty päivämäärä tai muu arvo.

Selauskenttälueetelo on määriteltävissä taulun jokaiselle kentälle. Luetteloon valitut kentät tulostuvat näytölle selauksen aikana. Jokaiselle selauskenttälueetelolle voi myös valita hakemiston, jonka perusteella kentät lajitellaan.

Kun taulujen välille halutaan luoda yhteyksiä, voidaan käyttää joko ovia tai oviaukkoja. Ovea käytetään useimmin tilanteissa, joissa tarkastellaan tietueiden alajoukkoa. Oviaukko taas liittyy kiinteästi tavalliseen kenttään ja siihen ei voi liittää automaattista selausikkunaa. Se soveltuu parhaiten monen suhde yhteen -yhteyksiin kuten asiakaskunta - yritys. ■

Tekniset tiedot

DataPerfect voi tallentaa 510 megatavua tekstiä yhteen tietokantaan. Yksi tietokanta voi sisältää 80 tiedostoa, joista jokaiseen mahtuu 16 miljoonaa tietuetta. Yhteen tietueeseen voi määritellä 80 kenttää.

DataPerfect vaatii teollisuusstandardin mukaisen mikron, DOS version 2.1 tai sitä uudemman ja 192 kilotavua muistia.

Toimiakseen verkossa DataPerfect vaatii DOS version 3.0 tai sitä uudemman, joka tukee tiedostojen yhteiskäyttöä ja lukitusta. Tuettuihin verkkoihin kuuluu Nokia MS-Net.

DataPerfectin toimitukseen kuuluvat kuvitettu käyttöoppaat sekä suunnittelijaa että loppukäyttäjää varten. Kumpaankin käyttöoppaaseen kuuluu opiseluosa, jossa DataPerfectin toiminnot esitellään, sekä toiminto-kohtainen hakuosa. Värikoodattu näppäimistöavain ja pikaopas auttavat käyttäjää oppimaan nopeasti DataPerfectin perustoiminnot.

Joustavat yhteydet ulkomaailmaan

Tiimi-työryhmäjärjestelmien ulkoisia yhteyksiä laajentamaan on Nokia Datassa kehitetty ohjelmistotuote X.25 PAD Support Program. Sen avulla liitetään Tiimi-työryhmäjärjestelmä X.25-pakettiverkon kautta muihin tietokonejärjestelmiin.

Yleensä PAD:lla (Packet Assembler/Dissambler) tarkoitetaan erillistä laitetta, johon voidaan kytkeä useita erilaisia asynkronisia päätteitä. Se voi olla myös pakettiverkon tarjoama palvelu, johon pääte on yhteydessä puhelinlinjan kautta. PAD-laite ohjaa päätteiden toimintaa ns. PAD-parametrien avulla.

X.25 PAD Support -ohjelmiston laiteympäristö muodostuu Tiimi-työryhmäverkon työasemista ja yhdestä tai useammasta X.25-tietoliikennepalvelimesta. Ohjelmiston pääasiallinen tehtävä on luoda ja ylläpitää keskitetysti tietoliikennesyhteyksiä työryhmäverkon ja pakettiverkoon liitetyn isäntäkoneen välillä.

X.25-yhteydet luodaan erillisen linjaohjaimen, ts. X.25-palvelimen kautta. Ohjain valvoo työasemien ja etäällä sijaitsevien isäntäkoneiden välisiä yhteyksiä. Lähiverkossa voi olla useita X.25-palvelimia ja yksi palvelin voi hoitaa 32 samanai-kaista pääteyhteyttä.

Tietoliikennepalvelimena voidaan käyttää mitä tahansa AT-luokan miktoa, esimerkiksi MikroMikko 3TT:tä. Ohjain varustetaan ICC-tietoliikenneohjainkortilla ja luonnollisesti myös tarvitta-

avalla lähiverkkosovittimella. Tietoliikennepalvelimessa ei tarvita näyttöä eikä näppäimistöä.

Nokian X.25 PAD -ohjelmistoa voidaan käyttää kaikissa Tiimi-työryhmäjärjestelmissä. Kuvassa 1 on esitetty tyypillinen TokenNet-työryhmäverkon laitteistoratkaisu.

Isäntäkoneyhteydet pääte-emulaattoreilla

Lähiverkon työasemat kommunikoivat isäntäkoneen kanssa pääte-emulaattoriohjelmiston avulla. Varsinaisen pääte-emuloinnin lisäksi emulaattorit sisältävät myös kirjoittimen hallintaan ja tiedostojen siirtoon liittyviä ominaisuuksia.

Nokian pääte-emulaattorit toimivat MS-DOS-käyttöjärjestelmän kanssa. Hotkey-ominaisuuden ansiosta ne mahdollistavat myös muiden sovellusten käytön samanaikaisesti päätetoiminnon kanssa.

WS 7800 ja WS 220 -emulaattorien avulla voidaan lähiverkon työasemasta olla yhteydessä isäntäkoneeseen X.25-tietoliikennepalvelimen kautta. WS 220 ja WS 7800 -emulaattoriohjelmat tarjoavat monipuoliset X.25-liitännät.

Nokian WS 7800 -emulaattori soveltuu erinomaisesti Honeywell-järjestelmäliitännöihin. Se tarjoaa liitännät esimerkiksi DPS 6 PLUS ja DPS 8 -keskuskoneisiin. Koska DPS 6:ssa ei ole X.25-pääteliitännätukea, siihen liityttäessä tarvitaan asynkronisia linjoja ja erillinen PAD-laite (Kuva 2, vaihtoehdot 2, 4 ja 5). DPS 8:aan liitytään DATANETin kautta, jolloin tarvitaan PAD-ohjelmisto.

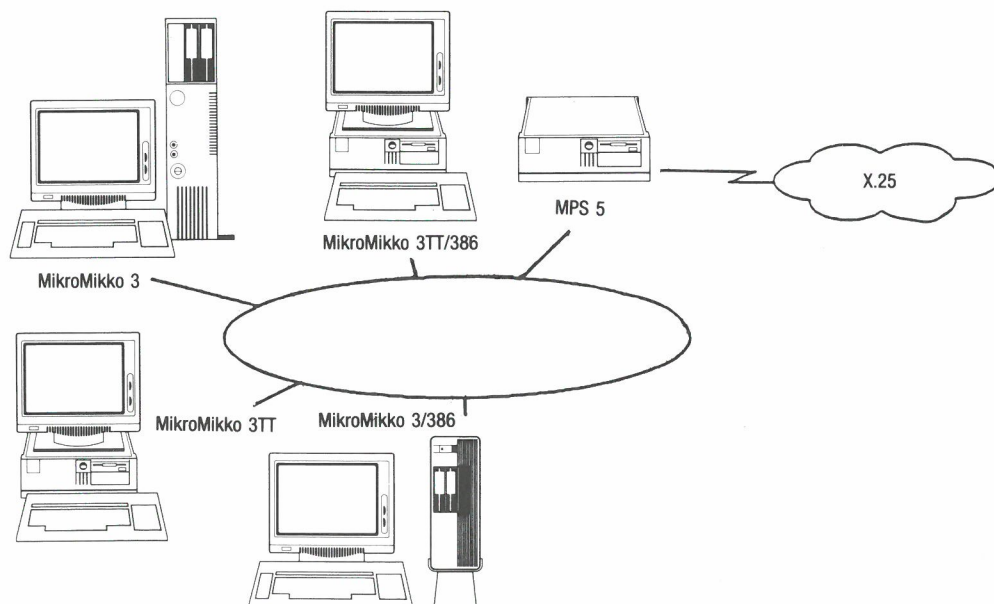
Muihin paitsi Honeywell-järjestelmiin liityttäessä käytetään WS 220 -emulaattoriohjelmistoa. Sen avulla liitytään esimerkiksi VAX-laitteisiin, joissa on VAX PSI -ohjelmisto. Tämä ohjelmisto mahdollistaa liittymisen pakettiverkkoon (Kuva 2, vaihtoehdot 1 ja 3).

WS 220 emuloi DEC:n VT 220 -päättettä (VT 200, VT 100 ja VT 52 -malleja) sekä Nokian VDU 202 -päättettä.

WS 7800 -emulaattori emuloi Honeywellin VIP 7813 -päättettä ja muistuttaa toiminnoltaan hyvin paljon WS 220 -emulaattoria.

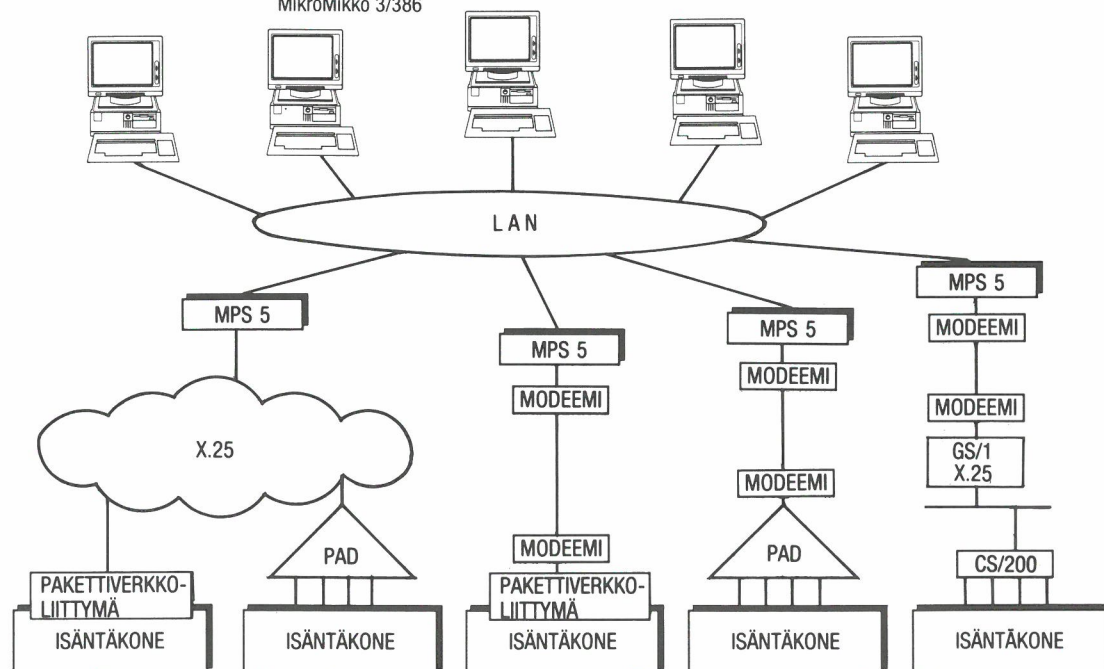
WS 220 ja WS 7800 -ohjelmistot sisältävät pääte-emuloinnin lisäksi ominaisuuksia, joiden avulla voi siirtää tiedostoja isäntäkoneen ja työaseman välillä sekä tulostaa tietoa lähiverkon

Kuvassa on esitetty Tiimin TokenNet-ympäristön X.25 PAD Support Program. Verkon työasemana voi olla AT-tasoinen MikroMikko, joka varustetaan TokenNet I tai TokenNet II -sovittimella. Nokian X.25 TokenNet -tietoliikennepalvelin on MikroMikko 3TT ilman näyttöä ja näppäimistöä. Palvelimessa on TokenNet II -sovitin ja ICC-tietoliikenneohjain. Verkko-ohjelmistona tarvitaan kussakin laitteessa LAN Support Program -ohjelmisto, joka tarjoaa järjestelmän vaatiman NetBios-rajapinnan. Työasemissa käytetään WS 220 tai WS 7800 -emulointiohjelmistoja isäntäkone-liitännän mukaisesti.



A.O. taulukoissa on esitetty X.25 PAD Support Programin laitteistovaatimukset erilaisissa verkkoympäristöissä sekä palvelimien että työasemien osalta. Tähdellä merkityt kokoonpanot suositellaan käytettäväksi X.25 -palvelimina.

TokenNet-kaapelointi	X.25-palvelin	Näppäimistö	Näytönohjain	Työmuisti	Levykeko	Sovittimet
	X.25 TokenNet-palvelin	Ei	Ei	1 MB	1,44 MB	TokenNet II ja ICC*
	MikroMikko 3TT MikroMikko 3TT/386	Ei	Ei	1 MB	1,44 MB	TokenNet II ja ICC
	MikroMikko 3 MikroMikko 3/386	Ei	Ei	1 MB	1,2 MB	TokenNet II ja ICC
	*) X.25 TokenNet-palvelin on MikroMikko 3TT:n käsittely-yksikkö, joka sisältää TokenNet II- ja ICC-sovittimet					
EtherNet-kaapelointi	X.25-palvelin	Näppäimistö	Näytönohjain	Työmuisti	Levykeko	Sovittimet
	X.25 Ethernet-palvelin	Ei	Ei	1MB	1,44 MB	EtherLink Plus ja ICC*
	MikroMikko 3TT MikroMikko 3TT/386	Ei	Ei	1 MB	1,44 MB	EtherLink Plus ja ICC
	MikroMikko3 MikroMikko 3/386	Ei	Ei	1 MB	1,2 MB	EtherLink Plus ja ICC
	*X.25 Ethernet-palvelin on MM 3TT:n käsittely-yksikkö, joka sisältää EtherLink Plus ja ICC-sovittimet					
NetNet-kaapelointi	X.25-palvelin	Näppäimistö	Näytönohjain	Työmuisti	Levykeko	Sovittimet
	MPS4	Ei	Ei	768 kB	720 kB	IOE
TokenNet-kaapelointi	Työasema	Näppäimistö	Näytönohjain	Työmuisti	Sovitin	
	MikroMikko 3TT MikroMikko 3	Nokia 100, 102	VDC, CGA, EGA	1 MB	TokenNet I tai TokenNet II	
	MikroMikko 3TT/386 MikroMikko 3/386	Nokia 102	NDC, VGA, EGA	1 MB	TokenNet I tai TokenNet II	
EtherNet-kaapelointi	Työasema	Näppäimistö	Näytönohjain	Työmuisti	Sovitin	
	MikroMikko 3 TT MikroMikko 3	Nokia 100, 102	NDC, VGA, EGA	1 MB	EtherLink Plus	
	MikroMikko 3 TT/386 MikroMikko 3/386	Nokia 102	NDC, VGA, EGA	1 MB	EtherLink Plus	
NetNet-kaapelointi	Työasema	Näppäimistö	Näytönohjain	Työmuisti	Sovitin	
	MikroMikko 3TT MikroMikko 3	Nokia 100, 102	NDC, VGA, EGA	1 MB	ICC	
	MikroMikko 3TT/386 MikroMikko 3/386	Nokia 102	NDC, VGA, EGA	1 MB	ICC	



X.25 PAD -ohjelmisto mahdollistaa yhteydet esimerkiksi Honeywell- ja DEC-isäntäkoneisiin. Yhteydet voidaan hoitaa joko yleisen pakettiverkon kautta tai paikallisina X.25-yhteyksinä. Keskuskoneisiin, joissa ei ole X.25-pääte-liitäntätukea, tarvitaan erillinen PAD-laite.

X.25 PAD Support Program/Järjestelmän ominaisuudet

- Virtuaaliyhteyksien lukumäärä/yhdyskäytävä	32 max.
- Työasemien lukumäärä/yhdyskäytävä	LAN-kohtainen
- Yhdyskäytävien lukumäärä/LAN	100 max
- Automaattinen liitäntämääritys/työasema	10 max.
- Yhteyksien lukumäärä/työasema	1
- Tiedonsiirtonopeus	19 200 bit/s
- Käyttöjärjestelmä	MS-DOS 3.X
- Työaseman ja yhdyskäytävän välinen yhteyskäytäntö	NetBios-käytäntö (TokenNet, EtherNet) Nokian oma (NetNet)
- Tulevien kutsujen tuki	Kyllä
- Automaattinen kytkentä	Kyllä
- Emulaattorit	WS 220, WS 7800
- Autologon	Kyllä (emulaattorin ominaisuus)
- Hotkey	Kyllä
- Tiedostonsiirto (teksti- ja binäärimuotoiset)	Kyllä



Sarjafoto

Taina Hurme työskentelee Nokia Datan Tuotekehityksessä järjestelmätukitehtävissä.

kirjoittimille. Tiedostonsiirto on kaksisuuntaista ja sekä tekstitiedostoja että binäärimuotoisia tiedostoja voidaan siirtää. Siirrossa käytetään joko Kermit- tai Nokian PCXFER-protokollaa.

Ennen käytön aloittamista X.25-palvelin konfiguroidaan X25CONF-ohjelmalla. Sen avulla voidaan määritellä tiettyjä X.25-yhteyden ominaisuuksia. Yleensä konfigurointi tehdään vain ohjelmiston asennusvaiheessa. Samaa konfigurointiohjelmaa käytetään myös tiettyjen työasemakohtaisten parametrien asettamiseen kuten työasemanumero/nimi.

Työasemassa ajettava X.25 PAD -ohjelmisto konfiguroidaan toimituslevykkeellä olevan PADCONF-ohjelman avulla. Määriteltäviin parametreihin kuuluvat muun muassa X.3-parametrit, emulaattorin valinta, erilaiset yhteysparametrit ja yhteydenottotapa, joka voi olla automaattinen tai manuaalinen.

Työasemassa käytettävä emulaattori konfiguroidaan ympäristön mukaan. Tärkeintä on määritellä pääteyhteys NET-tyyppiä.

Sekä X.25 PAD Support Program että WS-emulaattorit toimivat kaikissa Nokian valmistamissa AT-tason MikroMikoissa. Työasemissa, joissa on positiivinen, NDC-ohjaimella varustettu näyttö, voidaan haluttaessa käyttää valintanäppäimiä.

Jos käytetään MS-Net-verkko-ohjelmistoa, tiedostoja voidaan siirtää myös lähiverkon tiedostopalvelimen ja isäntäkoneen välillä. MS-Net-verkkokäytössä myös kirjoitintulostukset voidaan ohjata lähiverkon palvelimen kirjoittimille.

Suosituksien mukaiset PAD-toiminnot

X.25-yhdyskäytävä on hyväksytty kaikissa Skandinavian maissa. Nokian X.25 PAD Support -ohjelmisto sisältää CCITT:n suositusten X.3, X.28 ja X.29 (1984) mukaiset PAD-toiminnot. Ohjelmistoa voidaan käyttää ympäristöissä, joissa fyysisenä tiedonsiirtoverkkona on yleinen puhelin- tai dataverkko ja jotka tarjoavat pakettikytkentäisen tai piirikytkentäisen palvelun.

CCITT:n suositus X.3 määrittelee PAD-laitteen tehtävät ja X.28 määrittelee tavan, jolla pääteeltä ohjataan PAD:n toimintaa. Se määrittelee PAD-komennot, joilla käyttäjä voi kysellä ja asettaa PAD-parametrien arvoja sekä palvelusignaalit, joilla PAD antaa käyttäjälle tietoja muun muassa kutsujen onnistumisesta ja virhetilanteista. X.28 määrittelee myös yhteydenottokomennon haluttuun keskuskoneeseen.

CCITT:n suositus X.29 määrittelee tavan, jolla pakettiverkkoon liittynyt keskuskone tai muu etälaite ohjaa PAD:n toimintaa. Se määrittelee myös miten käyttäjädataa välitetään, ja miten keskuskoneen sovellusohjelmisto voi asettaa ja saada selville PAD-parametrit käyttäjälle näkymättömästi. Lisäksi se määrittelee virheellisiin asetuksiin liittyviä virhekoodeja.

X.25 PAD -ohjelmisto mahdollistaa tiedonsiirron 19 200 bitin sekuntinopeudella. Yhteydet voidaan luoda joko automaattisesti tai manuaalisesti. Samanaikaisia virtuaaliyhteyksiä voi X.25-palvelimen kautta olla 32. Verkossa voi olla useita tietoliikennepalvelimia.

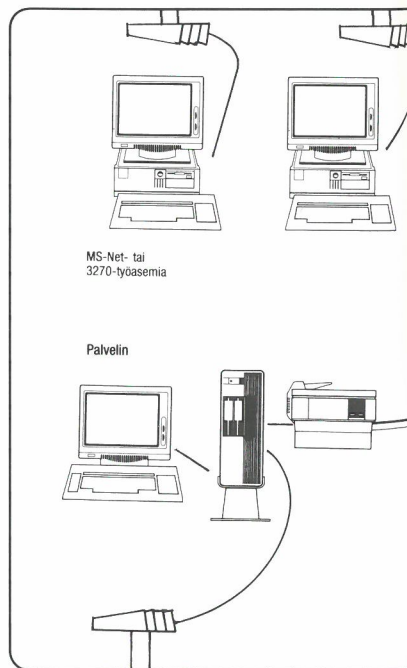
Työasemissa edellytetään olevan vähintään MS-DOS:n versio 3.10. Ohjelmiston toimimisen kannalta on myös oleellista, että työasemiin on asennettu lähiverkko-ohjelmisto, joka tarjoaa standardin mukaisen NetBios-rajapinnan. Lisäksi tarvitaan sopiva pääte-emulointiohjelmiston kuten WS 220 tai WS 7800.

X.25 PAD Support Program -ohjelmisto sisältää sekä X.25-palvelimeen että työasemaan tarvittavat ohjelmistot. Työasemaohjelmisto sisältää OSI-WS- ja PAD-ohjelmiston. Yhdyskäytäväohjelmistoon kuuluu osana X25CONF-ohjelma, jonka avulla voidaan konfiguroida yhdyskäytävän ominaisuuksia kuten virtuaaliyhteyksien lukumäärä ja siirrettävien pakettien koko. Varsinaista lähiverkon hallintaohjelmistoa ei vaadita, vaikkakin ohjelmistoa on mahdollista käyttää myös samanaikaisesti esimerkiksi MS-Netin kanssa. ■

MikroMikot ovat tehokkaita Tiimi-verkon työasemia, jotka myös voivat toimia 3270 -pääteinä Net-3270-ohjelmiston avulla.

Kesällä julkistettu Net-3270 versio 3.0 on Nokian kehittämä ohjelmistotuote, joka on tarkoitettu Tiimin TokenNet-työryhmäverkkoon.

Net-3270:n avulla Tiimi-verkko kytketään keskuskoneeseen. Sen avulla isäntäkoneessa olevia sovelluksia käytetään lähiverkon työasemissa ja siirretään tietoa isäntäkoneen ja työaseman välillä.



Net-3270 laiteympäristö

Tiimin työasemat 3270-päätteinä

Ominaisuudet

Ohjelmiston yhteyskäytännönä on SNA/SDLC ja se sisältää myös X.21-tuen.

Net-3270-työasemat emuloivat 3278 Model 2 -näytön toimintoja ja 3287-kirjoitintoimintoja. Net-3270-yhdyskäytävä toimii 3274-51C-ohjaimen mukaisesti. Työasemassa voidaan suorittaa samanaikaisesti jopa neljä Net-3270-istuntoa MS-DOS-istunnon lisäksi. Net-3270 on pysyvästi muistiin ladattava ohjelmisto (memory resident), ja Net-3270-istunnot voivat toimia MS-DOS-istunnon taustalla.

Samasta työasemasta voidaan määritellä jopa neljä samanaikaista yhteyttä eri isäntäkoneisiin neljän yhdyskäytävän kautta. Kukin yhdyskäytävä voi ylläpitää yhden isäntäkoneyhteyden kerrallaan. Net-3270-istunnon aikana tulostuvat tiedot järjestelmän tilasta näyttoon Net-3270-ikkunan tilariville.

WS 3270 -työasemaohjelma

on toteutettu modulaarisesti siten, että eri toiminnoille on omat käsittelyrutiinit. Tämä mahdollistaa sen, että työasemaohjelma voidaan konfiguroida sisältämään vain niiden toimintojen käsittelyrutiinit, jotka kukin työasema tarvitsee. Tämän seurauksena minimoituu muistiin ladattavan työasemaohjelman koko ja muistitilaa vapautuu muihin tarkoituksiin. Työasemaohjelman varaama muistitila voidaan vapauttaa, kun yhteyttä ei enää tarvita. Muistitilan lisäksi vapautuvat silloin myös työaseman käyttämät Net-3270-verkkoresurssit.

Net-3270-ohjelmisto tukee fyysisiä V.24-, X.21- ja X.21 SHM/MPS (Short Hold Mode and Multiple Port Sharing) -yhteyksiä. Yhteys isäntäkoneeseen voidaan muodostaa käyttämällä kiinteitä linjoja tai valintalinjoja. Yhdyskäytävä tukee linjanopeutta 19 200 bittiä sekunnissa V.24-yhteyksillä ja linjanopeutta 9 600 bittiä sekunnissa X.21-yhteyksillä.

X.21 Datex-yhteys isäntäkoneeseen voidaan luoda automaattisesti tai manuaalisesti. Kaikki samaan yhdyskäytävään liitetyt työasemat voivat käyttää samaa X.21-yhteyttä, kun yhteys isäntäkoneeseen on luotu.

Yhteys luodaan automaattisesti siten, että yhdyskäytävä ottaa yhteyttä isäntäkoneeseen välittömästi järjestelmän käynnistyttyä. Yhteydenoton epäonnistuessa voi yhdyskäytävä yrittää yhteyttä uudelleen määrätyn ajan kuluessa määrättyin aikavälein.

Manuaalisesti yhteys voidaan luoda ottamalla suoraan yhteyttä määrättyyn isäntäkoneen numeroon tai näppäimistöltä annettavaan numeroon.

Järjestelmä voi katkaista fyy-

sisen yhteyden automaattisesti ellei yhteyttä ole käytetty määrätyn ajan kuluessa (Short Hold Mode). Täten kertyy yhteyskustannuksia vain todellisesta yhteysajasta. Ohjelmisto avaa yhteyden uudelleen, jos työasema tai isäntäkone lähettää tietoja. Yhteyden katkeaminen ja uudelleen avautuminen ovat automaattisia, käyttäjälle näkymättömiä toimintoja.

Järjestelmä sisältää Nokian kehittämän sovellusohjelmointiliittymän, joka mahdollistaa työasemien ja isäntäkoneen välisessä yhteydessä käytettävien vuorovaikutteisten MS-DOS-ohjelmien kehittämisen.

Laiteympäristö

Kuvassa 1 on laiteympäristö, jossa lähiverkon työasemat voivat olla yhteydessä isäntäkoneeseen toimimalla 3270-päätteinä.

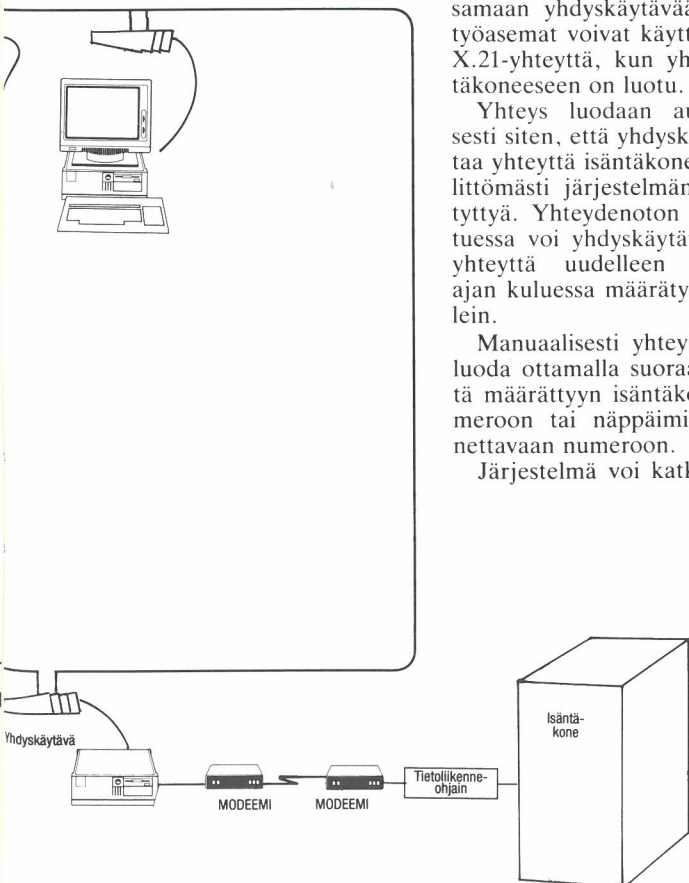
TokenNet-lähiverkko koostuu älykkäistä palvelimista ja työasemista, jotka liitetään rengasmaiseen runkokaapeliin TokenNet-liitäntäyksikön ja TokenNet-liitäntäkortin avulla. Liitäntäkortin ja liitäntäyksikön väliin kytketään liitäntäkaapeli. Verkon ulkoiset yhteydet hoitaa erillinen tietoliikennepalvelin.

Kaikki AT-tason MikroMikot voivat olla lähiverkon 3270-työasemana. Työasemassa on näytön ja näppäimistön lisäksi oltava TokenNet I -liitäntäkortti verkkoliitäntää varten sekä levykeasema. Net-3270-ohjelmisto tukee Nokian 100- ja 102-näppäimistöä sekä VGA-, NDC- ja EGA-näyttöä.

Yhdyskäytävä yhdistää Tiimin isäntäkoneeseen. Yhdyskäytävän tehtävänä on luoda ja ylläpitää tietoliikenneyhteyttä sekä katkaista yhteys. Verkossa voi olla useita yhdyskäytäviä, joihin toimii tyypillisesti levykeasemalla varustettu MikroMikko 3TT. Näyttö ja näppäimistö ovat valinnaiset.

Yhdyskäytävänä toimivassa MikroMikossa on oltava TokenNet II -liitäntäkortti verkkoliitäntää varten sekä ICC-tietoliikennekortti, josta saadaan modeemiliitäntä verkon ulkopuolisiin yhteyksiä varten. Verkossa voi olla useita yhdyskäytäviä.

Järjestelmää konfiguroitaessa voidaan määritellä jokaiselle työasemalle kirjoitin, jolle 3270-työaseman tulostus ohjataan.





Tarvittaessa käyttäjä voi ohjata tulostuksen jollekin toiselle verkkoon liitettylle kirjoittimelle. Työasemalle määritellyn kirjoittimen tunnus näkyy kuvaruudussa Net-3270-ikkunan tilarivillä.

Isäntäkoneessa sijaitsevat Net-3270-istunnon aikana käytettävät sovellukset, joita työasemat käyttävät yhdyskäytävän kautta. Isäntäkone tarvitsee tietoliikenneohjaimen ja modeemin pääteyhteyksiä varten.

Ikkunat	
Laitetyyppi	LU2
3270-ikkunoiden minimimäärä	0
3270-ikkunoiden maksimimäärä	4
3270-ikkunoiden oletusmäärä	1
Värejä	4
Värien konfigurointi	Tallennetaan tiedostoon, Net-3270-ohjelmiston mukana toimitetaan 2 tiedostoa Ei MS-DOS-taustatoimintaa On 3270-taustatoimintaa Oletusarvot Nokia 100 ja 102 näppäimistöille, Konfiguroitava hotkey Käytettävissä Nokia 100 näppäimistöllä ja MS-DOS versiossa 3.20
3270-ikkunoissa MS-DOS -ikkunassa Hotkey	
Valintanäppäimet	
MSDI	
Laitetyyppi	LU2
Toiminto	Toimii MS-DOS- ja 3270-sovellusten välisenä yhteytenä
Kirjoittimet	
Laitetyyppi	LU1, LU3 tai molemmat
Maksimimäärä	1 lähiverkon työasemalle; vaikka lähiverkon työasemaan olisi liitetty 2 kirjoitinta, vain 1 kirjoitin voidaan konfiguroida Net-3270-käyttöön. LPT1, LPT2, LPT3, COM1 tai COM2
Kirjoitinportit	Oheistulostus työasemalta, Systeemitulostus isäntäkoneelta
Tulostuslajit	
Yhdyskäytävä	
Laitetyyppi	PU2
Fyysinen yhteys	V.24 tai X.21
V.24-yhteyslajit	Half Duplex, Full Duplex Multipoint, Full Duplex Point to Point Flag Idle, Full Duplex Point to Point No Flag Idle
X.21-yhteyslajit	Non-switched, Manually switched, Short Hold Mode (SHM), Short Hold Mode/Multiple Port Sharing (SHM/MPs)
Maksimi linjanopeus	19 200 bps, jos V.24 9 600 bps, jos X.21
LU-yhteydet	Korkeintaan 64 ikkunaa (LU2), Korkeintaan 32 MSDI (LU2), Korkeintaan 32 kirjoitinta (LU1/LU3), Korkeintaan 96 yhdyskäytäväkohtaista LU-yhteyttä

Ohjelmisto

Työasemien ja yhdyskäytävän konfigurointia ja ohjelmistojen asennusta helpottavat ja nopeuttavat niitä varten kehitetyt asennus- ja konfigurointiohjelmat.

Lähiverkossa käytettävän käyttöjärjestelmän on oltava MS-DOS 3.20 IO.SYS 2.01 tai uudempi versio. Net-3270-ohjelmistossa työasemat ja yhdyskäytävä kommunikoivat keskenään NetBios-kutsuja käyttämällä.

Net-3270-ohjelmisto koostuu kolmesta tuotteesta. Net-3270 -työasemaohjelmisto. WS 3270 on asennettava jokaiseen Net-3270-työasemaan toimivaan MikroMikkoon. Net-3270 SNA -yhdyskäytäväohjelmisto käynnistyy automaattisesti ja se on konfiguroitava ympäristön mukaiseksi ennen käyttöönottoa. Net-3270 XFER -tiedostonsiirto-ohjelmisto mahdollistaa tiedostonsiirron työasemien ja isäntäkoneen välillä, mikäli isäntäkoneessa on tiedostonsiirtoa tukeva ohjelmisto (IND \$ FILE).

XFER-tiedostonsiirto-ohjelmisto on erikseen tilattavissa oleva työasemakohtainen ohjelmisto. Sen käyttö edellyttää Net-3270-työasemaohjelmiston asennusta. XFER-ohjelmiston avulla voidaan siirtää tiedostoja työasemien ja isäntäkoneen TSO-, CICS- tai VM/CMS-ympäristön välillä.

Lähiverkon verkko-ohjelmiston on oltava Nokian MS-Net 2.00 tai uudempi versio. Net-3270-ohjelmiston käyttö ei vaadi MS-Net-ohjelmiston asennusta.

Mikäli lähiverkossa ei käytetä MS-Net-ohjelmistoa, jokaiseen työasemaan on erikseen asennettava Nokian Lan Support Program 1.52 tai uudempi versio. ■

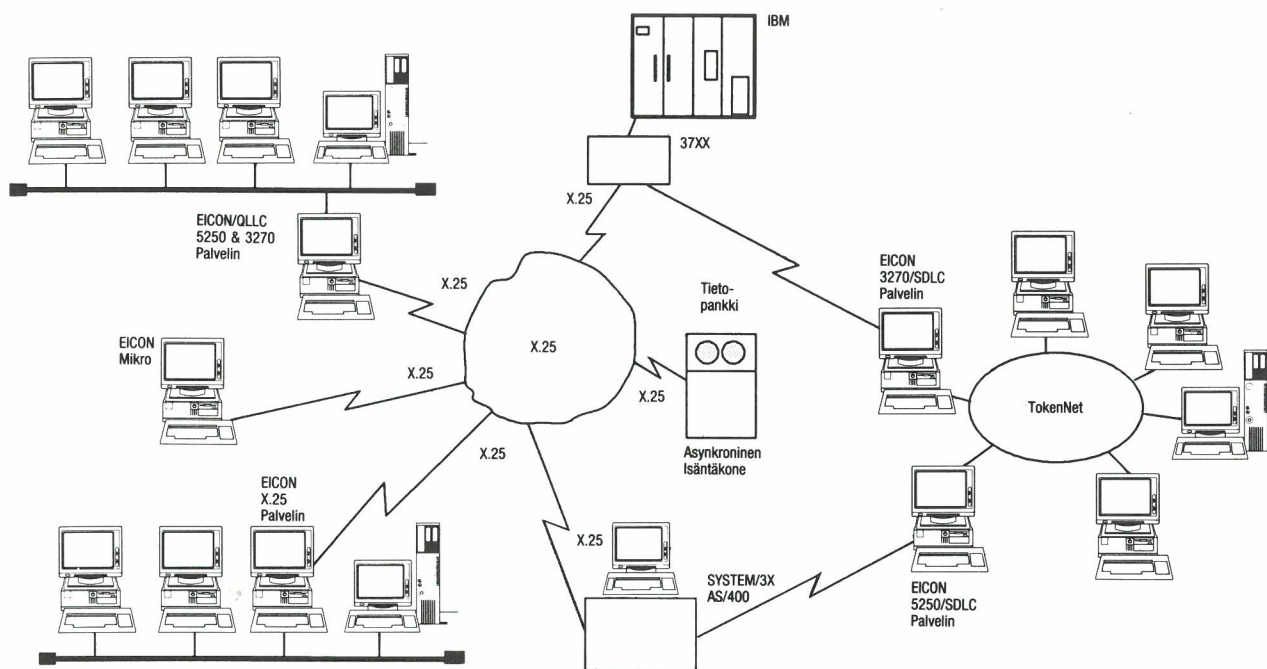
Pentti Laari

Eicon-tuotteet koostuvat mikrosovittimikortteista ja tietoliikenneohjelmistoista. Ne täydentävät hyvin Nokian tuotevalikoimaa.

Eicon-tuoteperhe koostuu tietoliikennekortteista, Access/X.25-, Access/QLLC-, Access/SDLC- ja Bridge-ohjelmistoista, joilla eri paikkakunnilla sijaitsevat työasemat, lähiverkot ja isäntäkoneet liitetään toisiinsa. Työasemissa voidaan käyttää myös esimerkiksi WS 220, WS 7800, Reflection- tai Softronics-emulaattoreita. Siirtotieksi sopii synkroninen soitto- ja kiinteä linja tai pakettikytkentäinen verkko.

Network Adapter, Dial Network Adapter (DNA) ja Single-Port Communication Coprocessor (SPCC) -tietoliikennekortit perustuvat 68008-suorittimelle. Se vastaa tietoliikenteen vaativista prosessoinneista ja vapauttaa näin mikron muiden sovellusten käyttöön. Network Adapter ja DNA-kortti on tarkoitettu teollisuusstandardin ja SPCC-kortti mikrokanava-arkkitehtuurin mukaisille mikroille.

Eicon tiedonsiirtoon



Access/X.25-ohjelmiston ominaisuuksia:

Access/X.25-ohjelmistolla liitetään joko yksittäinen työasema tai NetBios-yhteensopiva lähiverkko pakettikytkentäiseen verkkoon.

- Perustuu CCITT:n suosituksiin X.3, X.25, X.28, X.29, X.32 ja X.121
- 32 tai 254 virtuaaliyhteyttä konfiguraatiosta riippuen
- Valikkopohjainen käyttöliittymä
- Työasemassa samanaikaisesti 9 istuntoa
- "Hot key" DOS:n ja istuntojen välillä
- Kutsuhakemisto sadalle symboliselle kohdenimelle
- Automaattinen aloitus- ja "log on" -toiminto
- Paketti- ja kehitystason statistiikka
- "Trace"-toiminto
- TTY- ja VT 100 -emuloinnit

Ohjelmisto-optiot:

- Access/API (Application Programming Interface)
- 3rd Party Emulation Support (ITI)
- NetBios Bridge/X.25 ja NetBios Remote PC/X.25
- Attach/X.25
- Bridge/X.25 ja Remote X.25 Shell (Novell Netware)

Access/QLLC 3270 & 5250 Support -ohjelmistot

Access/QLLC-ohjelmisto tarjoaa Access/X.25-ohjelmiston ominaisuuksien lisäksi 3270- ja 5250-emuloinnin pakettikytkentäisen verkon kautta.

Access/QLLC 3270 Support Program emuloi:

- 3274 (Model 51C)
- 3278 (Model 2, 3, 4 ja 5)
- 3279 (Model S2A, S3G optiona)
- 3287
- 3270

Access/QLLC 5250 Support Program emuloi:

- 5294 (Model 1)
- 5251 (Model 11 ja 12)
- 5291 (Model 1 ja 2)
- 5292 (Model 1)
- 5256
- 5219

Ohjelmisto-optiot:

- 3270 Graphics (3279 Model S3G)
- Access/API
- IBM Personal Services/PC
- IBM PROFS PC
- IBM GDDM-PCLK
- 3rd Party Applications

- 3rd Party Emulation Support (ITI)
- NetBios Bridge/X.25 ja NetBios Remote PC/X.25
- Bridge/X.25 ja Remote X.25 Shell (Novell Netware)
- Eicon APPC (LU6.2/X.25 Support)
- Attach/X.25

Access/QLLC 5250 & 3270 Support -ohjelmisto tarjoaa molempien mainittujen ohjelmistojen toiminteet samanaikaisesti. Siten esimerkiksi yksittäisestä mikrosta voidaan kytkeytyä pakettiverkon kautta samanaikaisesti 3270-yhteyksillä sekä suurkankeisiin että System/3X-sarjan tietokoneisiin.

Access/SDLC

Access/SDLC-ohjelmistot tarjoavat soitto- tai kiinteän linjan kautta 3270- tai 5250-emuloinnin yksittäisille mikroille tai verkkoon kytketyille työasemille. Molemmat ohjelmistot pystyvät emuloimaan enintään jopa 32 CU:ta ja 254 LU:ta, jolloin isäntäkone näkee yhteyden multi-point-linjana.

Access/SDLC-ohjelmistot emuloivat samoja laitetyppejä kuin vastaavat QLLC-ohjelmistot. Myös ohjelmisto-optiot ovat samat lukuunottamatta Attach- ja Bridge/X.25-optioita. ■

Access/X.25-ohjelmiston tarjoamat tietoliikennemahdollisuudet. X.25-yhdyskätäväksi määriteltä mikro voi samanaikaisesti toimia myös työasemana.

Modeemi korjaa tiedonsiirtovirheet

Virheenkorjaavien modeemien tarve kasvaa sitä mukaan kuin valintaisten linjojen tiedonsiirtonopeudet kasvavat.

Tahdistamaton tiedonsiirto on tullut pitkään toimeen 300 tai 1200 bitin sekuntinopeuksilla, jolloin siirtovirheiden määrä valintaisilla linjoilla pysyy vielä kohtuullisena. Nykyiset sovellukset ja kustannustekijät ovat kuitenkin synnyttäneet tarpeen siirtää tietoa nopeammin. Tällöin myös virheiden vaikutukset kasvavat tuntuvasti varsinkin valintaisilla yhteyksillä.

Puhelinverkossa häiriöt esiintyvät useimmiten ryöppyinä. Esimerkiksi 4800 ja 9600 bitin siirtonopeuksia voi hyödyntää parhaiten, jos käytössä on ryöpymäisille häiriöille sopiva virheenkorjausmenetelmä. ARQ eli kehysrakenteinen virheenkorjaustapa on osoittautunut sopivaksi. Sen siirtokapasiteetti ei laske jyrkästi, ennenkuin virheitä on todella paljon. Itse asiassa siirtokapasiteetti saattaa nousta 10–15 prosenttia, jos kehysrakenteinen siirtotapa riisuu siir-

rettävästä datasta pariteetti- sekä alku- ja loppubitit.

Ohjelmallinen virheenkorjaus

Tahdistamatonta tiedonsiirtoa helpottamaan on kehitetty ARQ-tyyppisiä virheenkorjaavia tiedonsiirto-ohjelmia kuten Kermit, Xmodem, Ymodem ja Qmodem. Ne muodostavat tiedosta paketteja ja lisäävät tarkistussummaa. Vastaanotossa ohjelma pyytää paketin uudelleenlähetyksestä virheen ilmaantuessa.

Ohjelmallinen virheenkorjaus onnistuu, kun lähettävän ja vastaanottavan tietokoneen ohjelmat ovat yhteensopivia. Pulmia voi kuitenkin aiheutua siitä, että tieto ei aina siirry sellaisesta tietokoneesta toiseen, jossa siirto-ohjelma voi toimia. Ohjelmia on lisäksi monia ja standardointi puutteellista.

Modeemit virheenkorjaajina

Kun virheenkorjaus sijoitetaan modeemiin, tietokoneen tai päätelaitteen resurssit vapautetaan muuhun käyttöön. Modeemeille virheenkorjaus sopii, koska ne on usein, automaattivalintatoinnin yleistyessä varustettu ohjausproessoreilla. Niillä on myös virheenkorjaustoiminnon ohjaamiseen tarvittavaa tietoa yhteyden laadusta.

Merkittävät modeemivalmistajat ovat ottaneet käyttöön HDLC-tyyppisen virheenkorjaavan yhteyskäytännön ja esimerkiksi Nokia käyttää CCITT:n suosittelemaa LAPB:tä. Valmistajien toteutukset kuitenkin eroavat toisistaan ja käytännössä on tarvittu saman valmistajan modeemit yhteyden molempiin päihin. USA:ssa MNP-yhteyksikäytäntö on kaikkein yleisin.

CCITT on tehnyt keväällä 1988 päätöksen valmistella virheenkorjauksesta suosituksen V.42. Tulevan suosituksen määrittelemä yhteyskäytäntö LAPM sisältää MNP:n pääosan. Näin laitekannan enemmistö saadaan toimimaan yhdessä uuden suosituksen mukaisten modeemien kanssa. Valmisteluun osallistuvat kaikki merkittävät valtiolliset telehallinnot sekä laitevalmistajat kuten Nokia. ■

Lippo Rantanen

ECM - enemmän kuin modeemi

ECM 4896 -modeemit tulivat markkinoille kaksi vuotta sitten, ensimmäisinä laatuaan maailmassa.

CITT:n V.32-suosituksen mukainen ECM 4896 on enemmän kuin tavallinen modeemi. Se on signaaliprosessori, joka hoitaa muiden tehtävien ohessa perusmodeemin toiminnot – merkkien muuttamisen lähteviksi äänitaajuussignaaleiksi ja saapuvien signaalien muuttamisen merkeiksi.

ECM 4896 korjaa puhelinverkon vääristymät tehokkaalla automaattisella siirtokorjaimella. Signaalin käsittelyyn käytetty kaiunpoistotekniikka vaatii modeemilta suorituskykyisen kaiunpoistajan. Virheenkorjain pake-

toi lähtevät merkit HDLC-kehyksiin ja purkaa vastaanotetut kehykset.

ECM 4896 käsittelee signaaleja 10 miljoonan käskyn sekuntivauhdilla ja tammikuussa tuotantoon tuleva ECM 9632 kaksi kertaa nopeammin. ECM-modeemissa on kenties enemmän tehoa kuin siihen kytketyssä mikrossa!

Vaativassa tietokoneen ja päätteen välisessä asynkronisessa työskentelyssä, jossa tiedonsiirrolta vaaditaan ehdotonta virheettömyyttä, tarvitaan virheenkorjaavat 1200–9600 bitin sekuntinopeudella toimivat modeemit.

Kaksisuuntaiset eli CCITT:n suositusten V.22, V.22bis ja V.32 1200, 2400, 4800 ja 9600 bitin sekuntinopeuksiset modeemit toimivat puhelinverkossa tahdistetusti. Tulevat ja menevät bitit liikkuvat modeemien välillä samassa tahdissa ja modeemit välittävät tietovirran bitti bitiltä.

Kun halutaan siirtää tahdistam-

mattomia merkkejä, tavallinen kaksisuuntaisen V.22, V.22bis ja V.32 modeemi vain muuntaa merkit tahdistettuun siirtoon sopiviksi. Se ei muokkaa tietoa mitenkään, vaan siirtovirheet välittyvät päätelaitteille.

Virheenkorjaava modeemi, kuten ECM 4896, pakkaa siirrettävät tahdistamattomat merkit HDLC-kehyksiin ja varustaa kehykset tarkistussummalla. Vastaanottava ECM 4896 tietää tarkistussumman perusteella, onko puhelinyhteys aiheuttanut siirtovirheitä. Jos siirtovirheitä havaitaan, modeemi pyytää HDLC-kehyksen uudelleenlähetyksestä.

Muutamat modeemivalmistajat, muun muassa Nokia Data-siirto ja Hayes, ovat ottaneet käyttöön CCITT:n kansainvälisen standardin, LAPB:n, joka on yleisesti käytössä X.25-pakettiverkoissa. Nyt jo tuotannossa oleva ECM 4896M tukee sekä LAPB- että MNP 2–5 -yhteyksikäytäntöjä. ■

FUNET YHDISTÄÄ JA VÄLITTÄÄ

FUNET on tietoliikenneverkko, joka yhdistää Suomen kaikki seitsemäntoista yliopistoa ja korkeakoulua toisiinsa. Lisäksi verkkoon on liittynyt kymmenkunta valtion tai yksityisten yritysten tutkimusta harjoittavaa yksikköä. FUNET-projekti sai alkunsa vuonna 1984. Hankkeen rahoittaja on opetusministeriö.

Idea maamme ylintä opetusta antavien laitosten yhdistämisestä toisiinsa ei voi lukea minkään yksittäisen tahon ansioksi. Opetusministeriössä kuitenkin aikoinaan havaittiin korkeakoulujen tarve verkkovälitteiseen tiedonsiirtoon. Sen pohjalta lähdettiin luomaan FUNETia.

Suomen korkeakouluverkosta on juuri avautunut yhteys muiden Pohjoismaiden vastaaviin verkkoihin. Yhdessä ne muodostavat NORDUNet-nimisen kokonaisuuden. NORDUNet on puolestaan saanut oikeuden liittyä Yhdysvaltojen tiedeverkkoon, jota kutsutaan NSFNetiksi. Tämä merkitsee, että mistä tahansa suomalaisesta yliopistosta pääsee päätteen avulla suoraan yhteyteen minkä tahansa noin 200–300:n USA:laisen yliopiston kanssa.

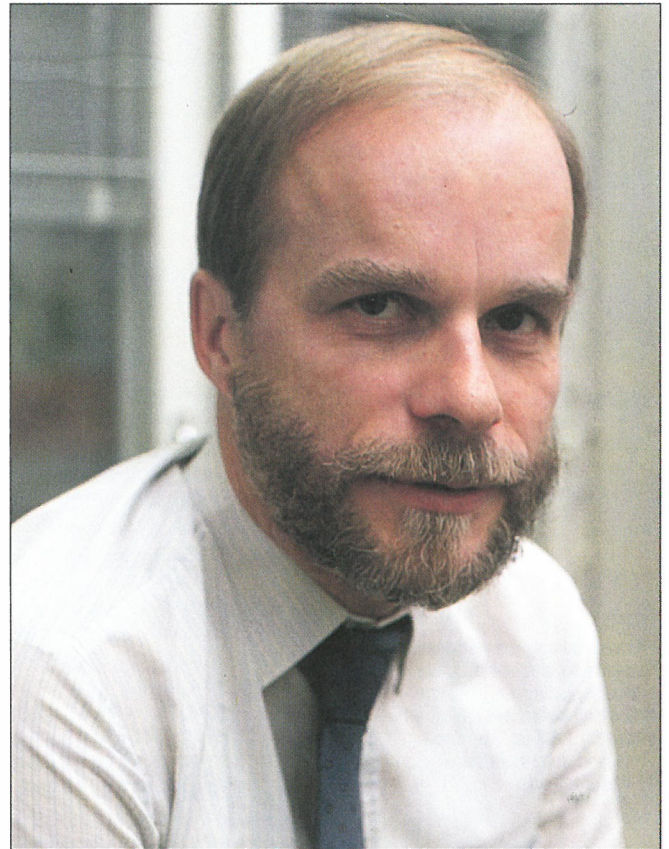
NSFNet-yhteyttä voidaan pitää hyvin merkittävänä saavutuksena. Kaikilla pohjoismaisen NORDUNet-verkon jäsenillä on oikeus NSFNetin käyttöön; muualla Euroopassa käyttöoikeus on aniharvoilla. Lisääntyvä kontaktimäärä lisää myös verkon hyödyllisyyttä.

FUNETin henkilökunnan palkat, laitteistojen hankinnat ja ulkomaiset käyttökulut muodostavat noin kahden, kolmen miljoonan markan kustannukset vuodessa. Summa katetaan opetusministeriön budjetista. Verkon kotimaiset käyttökulut ovat noin puolitoista miljoonaa markkaa vuodessa. Se jaetaan verkon käyttäjien kesken.

FUNET Euroopan kärkeä

Korkeakouluverkon johtoryhmässä rahoittajaa eli opetusministeriötä edustaa erikoistutkija, filosofian tohtori Risto Raitio. Hänen mukaansa verkkoa lähdettiin luomaan tarvepohjalta. Toistaiseksi saatujen kokemusten mukaan FUNET on palvelutarkoitustaan erinomaisesti.

"FUNET edustaa Euroopan kärkeä tietoliikenneverkoissa. Kokemukset ovat olleet hyviä ja



verkko kehittyy koko ajan. Tällä hetkellä FUNET yhdistää nopeilla linjoilla eri korkeakoulujen paikallisverkot, mikä mahdollistaa melkoisen liikennöinnin. Aluksi FUNET toimi vain pakettiverkkona", sanoo Raitio.

"Suomessa FUNET yhdistää toisiinsa viidestäsadasta tuhatteen suurta tietokonetta. Verkon tiedonsiirtonopeus on nyt 64 kilobittiä sekunnissa. Vuoden 1989 alussa Helsingin ja Tampereen välille asennetaan linja, joka pystyy kahden megabitin sekuntinopeuteen", hän jatkaa.

Raition mukaan tutkijat pitävät FUNETia tärkeimpänä Suomen korkeakoulujen välisenä atk-yhteistyön muotona. Tutkijoiden välinen ajatusten ja tiedon vaihto on tehostunut merkittävästi. Samaten erilaiset palvelut, muun muassa maamme ensimmäisen supertietokoneen käyttö, ovat tasapuolisesti kaikkien ulottuvilla.

"Monet tutkijat pitävät FUNETia tärkeimpänä yhteistyön muotona maamme korkeakoulujen välillä", sanoo opetusministeriön erikoistutkija Risto Raitio.

Tiedonsiirtoverkon periaate

Verkkovälitteisen tiedonsiirron periaate on verrattain yksinkertainen. Tietokoneet yhdistetään toisiinsa kaapelilla, jonka välityksellä ne voivat vaihtaa tietoa keskenään. Lisäksi tarvitaan tietoliikenneohjelmistot eli protokollat. Useat toisiinsa liitetyt tietokoneet muodostavat verkon, jossa ne voivat keskinäisen tiedonsiirron lisäksi käyttää hyväkseen toistensa ominaisuuksia sekä verkon muita palvelimia. Kapasiteetiltaan pienen mikron näppäilijä voi siis suorittaa hyvinkin vaativia laskentatehtäviä jollain samaan verkkoon kyteillä suurtietokoneella.

FUNET ei kuitenkaan muodostu yksittäisistä toisiinsa liitetystä tietokoneista, vaan se on verkko, joka kytkee toisia verkkoja yhteen. Jokaisella yliopistolla tai korkeakoululla on tätä nykyä oma paikallisverkkonsa. Ne ovat nimensä mukaisesti alueellisia verkkoja, joilla tietyn oppilaitoksen eri rakennuksissa sijaitsevat tietokoneet ja oheislaitteet on yhdistetty. FUNET linkittää nämä paikalliset verkot toisiinsa.

Verkon käyttömahdollisuudet

Verkolla on monia käyttömuotoja. Yksi henkilökohtainen viestintämahdollisuus on sähköinen posti. Mikroilta tai päätteilä voi lähettää viestejä, elektronisia kirjoja, muiden verkkoon kuuluvien tietokoneiden postilaatikoihin. Viestit kulkevat muuta-

massa sekunnissa vaikka maapallon puolelta toiselle.

Toinen verkon tarjoama viestintämuoto on uutisryhmät. Ne ovat ikäänkuin sähköisiä ilmoitustauluja. Tauluja on FUNETissa kaikkiaan noin 300, ja kullakin niistä käsitellään määrätyn aihepiirin asioita. Aiheet vaihtelevat mekatroniikasta ja ikkunointijärjestelmistä vitsinikkarointiin. Ilmoitustauluilla voi myös kysellä tai kertoa meneillään olevista tutkimuksista.

Kolmas hyvin tärkeä verkon palvelumuoto on mahdollisuus siirtää tiedostoja paikasta toiseen. Tällöin esimerkiksi yhdysvaltalainen ja suomalainen tutkija voivat yhteistyönä julkaista teoksen, jonka eri luvut ja osiot ovat viiveettä kumpaisenkin tarkistettavissa ja korjattavissa. Ilman elektronista tiedonsiirtoyhteyttä samat toimenpiteet vaatisivat lukuisia pitkiä puhelinsoittoja ja melkoista kirjeenvaihtoa.

Tiedemiehen kokemukset

Tiedemaailmaa korkeakouluverkon johtoryhmässä edustaa apulaisprofessori Juha Heinänen Tampereen teknillisestä korkeakoulusta. Verkko on hänen mielestään erittäin tärkeä tutkimuksen apuväline.

Heinäsen mukaan FUNET palvelee kaikkia tieteenaloja, mutta eniten käyttäjiä löytyy niiltä aloilta, joilla on perinteisesti totuttu työskentelemään tietokoneilla. Tekniikan ja luonnontieteiden edustajien lisäksi verkon käyttö on nyttemmin yleistymässä käyttäytymistieteil-

jöiden ja humanistien parissa.

"Verkon tärkeimpänä käyttömuotona pidän sähköpostia. Se on peruspalvelu, jota ilman ei osaisi olla siihen kerran tutustutuaan. Myös ilmoitustaulut ovat hyvin tärkeitä. Sitä kautta saan nykyään suurimman osan teknisiä uutuuksia koskevasta tiedosta. Myös mahdollisuus siirtää tiedostoja ja ohjelmistoja verkon kautta parantaa tutkijoiden yhteistyötä huomattavasti. Paikallisverkon käyttö helpottaa myös päivittäistä työviestintää. Annan esimerkiksi kaikki harjoitustehtävät opiskelijoille sähköisessä muodossa", Heinänen kertoo.

Tietoliikenteen määrä on FUNETissa vaikuttava. Esimerkiksi pelkästään yhden tamperelaisen tietokoneen päivän viestisaldo liikkuu tuhansissa. Heinänen mukaan verkon viestintäliikenteen kokonaismäärää on vaikea hahmottaa. Ilmoitustaululiikenne on joka tapauksessa massiivisinta, noin kymmenen megatavua viikossa.

Vain avoimuus on järkevää

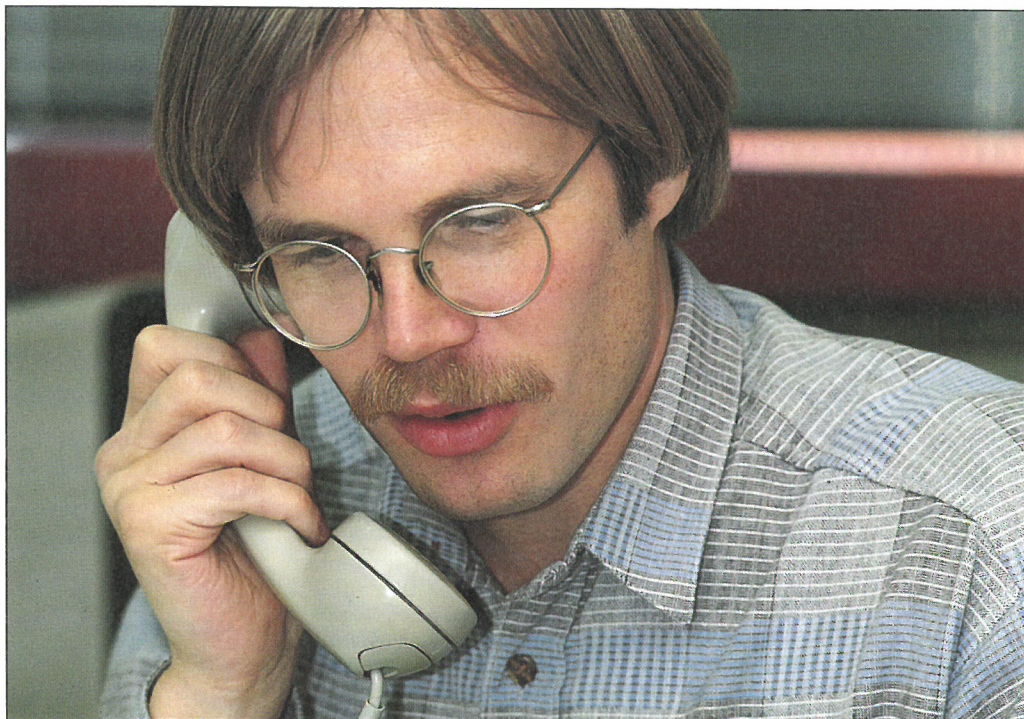
Korkeakouluverkko toimii avoimuuden pohjalta. Se tarkoittaa, että verkossa toimii eri protokollia. Toisin sanoen, eri valmistajien tietokoneet voidaan yhdistää viestimään keskenään. Pääpaino on kuitenkin laiteoimittajista riippumattomissa protokollissa.

"Avoin verkko on ainoa järkevä vaihtoehto", Risto Raitio painottaa. "FUNETissa on pyritty vuodesta 1986 lähtien laitevalmistajista riippumattomaan liikkumiseen. Valtionhallinnon omassa verkkoprojektissa olisi syytä ottaa oppia korkeakouluverkosta saaduista hyvistä kokemuksista."

Raitio viittaa hankkeeseen, jonka tarkoituksena on luoda tietoliikenneyhteys kaikkien ministeriöiden välille. Valtionhallinnossa ei Raition mukaan ole kovinkaan hanakasti tutustuttu FUNETin ratkaisuihin. Hän kuitenkin jatkaa, että FUNET-tietous leviää ja että avoimuuden periaatteen eduista ollaan ottamassa vaarin.

"Korkeakoulujen avoimista verkoista otetaan vauhtia sähköisen kommunikoinnin yleistymiseen, jatkossa nykyisen puhelinverkon käytön tasolle. Verkko tuo viestimiseen vapauden aikaan ja paikkaan nähden. Fyysiset välimatkatkaan eivät merkitse mitään. Tietoliikenne ja sitä hoitava tekniikka vastaavat nykyajan ja jopa huomispäivän haasteisiin", sanoo erikoistutkija Raitio.

"Korkeakouluverkon toimivuuden taustalla on lähtökohta, jonka mukaan yksittäisten tietokoneiden sijaan paikalliset verkot on yhdistetty toisiinsa", apulaisprofessori Juha Heinänen TTKK:sta toteaa.



FUNETin ongelmat

Käytännössä FUNETin ongelmat liittyvät Raition mielestä standardointityön hitauteen ja mutkallisuuteen. Kaikki tietokoneet eivät vielä voi kommunikoida kansainvälisesti keskenään. Standardeihin sopeutumisen aste vaihtelee hyvin paljon maittain.

Apulaisprofessori Heinänen näkee FUNETissa muun muassa seuraavat hankaluudet. "Elisan ja FUNETin välinen viestintä ontuu. Korkeakouluverkossa voi hyvin lähettää vaikkapa sata-, kaksisataasivuisia raportteja. Elisassa se ei onnistu. Elisa on selvä ad hoc -ratkaisu. Näkisinkin kernaasti nykyistä useamman yrityksen tutkimusyksikön liittyvän suoraan FUNETiin. Tämä olisi tarpeellista monien yhteistyöprojektien takia."

"Toinen pieni ongelma on FUNETin organisaatio. Toistaiseksi ei ole täysin selvää, mitä FUNETilla milloinkin tarkoitetaan. Virallinen organisaatio se ei ole, vaikka sanaa siten välillä käytetäänkin. Tulevaisuudessa ehkä puhutaan FUNET Inc:stä tai ry:stä", Heinänen sanoo.

Tietoturvallisuus verkkoviestinnässä

Avoimuuden lisäksi korkeakouluverkon periaate on, että sen käyttö on tarkoitettu tutkimus- ja tuotekehitystiedon välittämiseen. Verkkoon liittyvät yritykset eivät saa käyttää verkkoa operatiivisen eli hallinnollisen tai kaupallisen tiedon siirtoon. Tämä vähentää osaltaan tietoturvallisuusriskejä.

"FUNETin tapainen verkko on aina otollinen hakkeroinnin kohde. Vaikka verkossa ei ole tarkoitus välittää strategisesti elintärkeää tietoa, täytyy sen mielestäni silti olla suojattua", Risto Raitio katsoo. "Hakkerointi on rikollisena toimintana aina tuomittavaa. Minä en ainkaan halua, että postilaatikkoni tai tiedostojeni sisältöä pääsee kuka tahansa lukemaan. Niinpä verkossa siirrettävä tieto kannattaa aina Cryptata eli salakirjoittaa, mikäli on olemassa pienikin riski, että väärin käsiin joutuessaan tieto voi aiheuttaa vahinkoa tai menetyksiä."

Juha Heinänen ei puolestaan näe tietoturvallisuutta korostuneen merkittävänä seikkana korkeakouluverkossa. "Tietosuoja ei FUNETissa voi aina taata. Eikä se ole tarpeenkaan. FUNET palvelee avointa tutkimustyötä ja julkisuus on tieteellisen tutkimuksen yksi peruskriteeri", hän sanoo.



"Yritysten ei pidä siirtää verkossa sellaista tuotekehitystietoa, jota ei ole tarkoitettu julkiseksi. Samaten verkkoon kuuluvien hallinnollisten yksiköitten täytyy itse huolehtia tietosuojastaan tai luoda itselleen oma verkko. Operatiivisen tietonsa takia esimerkiksi kouluhallituksen liittymistä verkkoon ei ole pidetty suotavana. FUNET on tieteellistä tietoa varten luotu", Heinänen toistaa.

OTANET esimerkki alueverkosta

Espoon Otaniemessä toimiva alueverkko, OTANET, on Suomen suurin paikallisverkko. Verkon runkona on kahdeksanhaarainen tähti. Toistaiseksi vastaan neljä sakaraa on käytössä. OTANET yhdistää sekä toisiinsa että FUNETiin Teknillisen korkeakoulun, Valtion teknillisen tutkimuskeskuksen, Valtion tietokonekeskuksen ja Otaniemen Teknologia kylän. TKK:n kautta verkkoon on kytketty myös Teekkarikylä.

OTANET on toteutettu Nokian tuotteilla. Tähti rakentuu optiselle kaapeloinnille. Verkkoon liittyneet yksiköt ovat tähdessä kiinni silloilla, joiksi osa on valinnut Nokian Bridge-tuotteen.

"OTANETin syntyvaihe ajoittuu kesään 1987, jolloin verkko kytkettiin toimintaan", OTANETin projektipäällikkö Mikko Salminen kertoo. "Lokakuussa 1987 juhlittiin verkon virallisia avajaisia. Yhteistyösopimukset puolestaan allekirjoitettiin vuot-

ta myöhemmin. Tämä on hyvä osoitus Otaniemessä vallitsevasta yhteishengestä: ideasta siirryttiin toteutukseen vuodessa eikä byrokratian annettu jarruttaa verkon käyttöönottoa."

Verkko hyödyttää kaikkia

"Verkko luo alueen tutkijoille, työntekijöille ja opiskelijoille läpinäkyvän tietojenkäsittelyympäristön. Verkosta hyötyvät sekä ne, joilla ei ole varaa omaan suurkankeeseen että ne, joilla suurkanke on. Hyvin tehokkaiden tietokoneiden kapasiteetti on nyt kaikkien käytettävissä. Toisaalta suurkankeiden omistajat saavat laitteensa käyttöasteen optimoitua. Koneen ostoa on se, mikä maksaa, ei käyttö", Salminen sanoo.

OTANETiin on tällä hetkellä kytketty noin 2000 henkilökoh- taista tietokonetta. Verkko laajenee lähinnä yksiköiden sisäisten liittymien määrän lisäysten myötä.

"Elektroniset viestintäpalvelut ovat aivan selvästi kasvavan trendin alaisia. Tulevaisuudessa sähköpostin käyttö on ainakin nuorelle polvelle yhtä luonnollista kuin puhelimella soittaminen tänään. Nopeus, luotettavuus, riippumattomuus ajasta ja paikasta sekä mahdollisuus määrällisesti isojen tietopakettien siirtoon ovat ominaisuuksia, jotka määrittelevät huomisen viestintää", sanoo Mikko Salminen. ■

"Tietoliikenneverkot luovat uutta viestinnällistä infrastruktuuria tiede- ja yritysmaailman välille", sanoo OTANET-hankkeen projektipäällikkö Mikko Salminen.

DDE siirtää tietoa Windows-ympäristössä

Windows-sovellukset voivat vaihtaa tietoa keskenään apupöydän välityksellä. Windows-ympäristöön on myös määritelty Dynamic Data Exchange -protokolla, jonka avulla tietoa voi siirtää Windows-sovellusten välillä paljon monipuolisemmin.

Kun kahden tietokoneen välille on saatu toimiva tiedonsiirtoyhteys, on täytetty tietokoneiden väliselle kommunikoinnille asetettu välttämätön edellytys. Lisäksi on sovittava säännöt, joiden mukaan tieto kulkee. Puhutaan tietoliikenneprotokollista. Tyypillisesti protokollassa määrätään siirrettävän datan muoto sekä kuka lähettää mitään siirtotapahtuman eri vaiheissa.

Windows-ympäristön protokolla

Windows-ympäristöön tehtyjä sovelluksia kutsutaan Windows-sovelluksiksi. Ne poikkeavat sisäiseltä logikaltaan melkoisesti perinteisistä sovelluksista. Tavallisesti Windows-sovellukset vastaanottavat ja käsittelevät ikkunoissa käyttäjän syötteistä aiheutuneita viestejä.

Windows-sovellusten välillä on "tiedonsiirtoyhteys", sillä ne voivat lähettää viestejä myös toisilleen. Dynamic Data Exchange eli lyhennettynä DDE on tätä tarkoitusta varten määritelty protokolla.

DDE:tä tukevat sovellukset voivat vaihtaa tietoa ilman käyttäjän apua. Esimerkiksi tietokantaohjelmisto voi automaattisesti lähettää muuttuneet tiedot grafiikkaohjelmistolle, joka piirtää sitten uusien tietojen mukaisen kuvaajan näytölle.

DDE-protokollassa varataan ja määritellään joukko Windows-viestejä DDE-liikennöintiä varten. Windows ei kuitenkaan tarjoa sovellukselle valmiita DDE-ominaisuuksia, vaan DDE:n mukainen lähetys- ja vastaanottologikka on erikseen ohjelmoitava kuhunkin sovellukseen. DDE-protokollan mukaan siirrettävän tiedon muoto määritellään samalla tavalla kuin siirrettäessä tietoa sovellusten välillä apupöydän välityksellä.

Tieto siirretään kirjoittamalla ja lukemalla globaaleja muisti-

objekteja, joita kutsutaan Windowsissa atomeiksi. Viestissä välitetään ainoastaan atomin osoite. Windows-versiossa 1.x atomien käsittely hoidetaan erillisellä kirjastolla DDE.EXE. Versiosta 2.0 lähtien atomien käsittely on Windowsin vakiovaruste.

DDE-protokollan mukaan kommunikoivien sovellusten on oltava samanaikaisesti aktiivisina. Tämä voi suurikokoisissa ohjelmissa aiheuttaa muistitilaongelmia.

Tieto määritellään kolmitasoiseksi

DDE käyttää kolmitasoisia hierarkkista tiedon määrittelyä. Ylimmällä tasolla on sovellus (Application). Sovelluksella voi olla monta teemaa (Topic). Kukin teema voi sisältää useita alkioita (Item).

Teema on looginen tietokokonaisuus, joihin sovellus ryhmittelee tietonsa. Usein käytetty jako on yksi teema sovelluksen jokaista datatiedostoa kohti.

Alkio on tietopalanen, joka voidaan siirtää DDE:tä käyttämällä. Sovellus ryhmittelee teeman tiedot alkioiksi. Niiden ei tarvitse olla samanmuotoisia edes saman teeman alla. Jos esimerkiksi teemana on taulukko lukuja, toinen alkio voi sisältää yhden luvun ja toinen alkio vaikkapa kokonaisen taulukon rivin.

Alkion siirtomuoto on jokin rekisteröidyistä apupöydän esitysmuodoista. Windows-ympäristössä on valmiiksi rekisteröity joukko apupöydän esitysmuotoja ja sovellukset voivat itse määritellä niitä lisää. Tiedon siirron varmistamiseksi tulisi sovelluksen tukea lisäksi ainakin yhtä valmiiksi rekisteröityä esitysmuotoa.

Hyvän "DDE-tyylin" mukaisesti jokaisella sovelluksella tulisi olla erityinen System-teema. Tämän teeman alkioiden sopivaa sisältöä ovat muun muassa teeman omien alkioiden nimet,

sovelluksen teemojen nimet ja tilatiedot, sen tukemat apupöydän esitysmuodot ja muut siihen liittyvät yleiset tiedot.

DDE:n mukaiset tapahtumat

Kun halutaan siirtää tietoa DDE:n mukaisesti, perustetaan ensin yhteys kahden Windows-sovelluksen välille. Yhteyden alustaa toinen sovelluksista, jota kutsutaan sen jälkeen isännäksi (client). Toista osapuolta kutsutaan rengiksi (server). Sekä isännän että rengin on tuettava DDE:tä. Windows-sovelluksella voi olla samanaikaisesti auki useita DDE-yhteyksiä, joista osassa se toimii isäntänä ja osassa renkinä.

Windows-ympäristössä viestit lähetetään aina ikkunoihin, joten sekä isännän että rengin on perustettava jokaista DDE-yhteyttä varten oma yhteysikkunansa, jota käyttäjä ei yleensä näe.

DDE-liikenne koostuu monista tapahtumista (transaction). Tapahtuman aikana isäntä ja renki lähettävät toisilleen useita toisiinsa kiinteästi liittyviä viestejä. Seuraavassa kuvataan yksittellen DDE:hen liittyvät tapahtumat.

Alustustapahtumassa luodaan isännän ja rengin välinen yhteys. Erikoistapauksissa saatetaan luoda myös useita yhteyksiä.

Isäntäsovellus alustaa DDE-yhteyden lähettämällä kaikille aktiivisille sovelluksille viestin WM_DDE_INITIATE. Viesti sisältää sovelluksen ja teeman nimet sekä isännän yhteysikkunan osoitteen. Jos isäntä jättää sovelluksen ilmoittamatta, mikä tahansa sovellus kelpaa. Jos isäntä jättää teeman ilmoittamatta, se haluaa alustaa yhteydet rengin jokaiseen teemaan. Kun renki vastaanottaa sitä koskevan alustuspyynnön, se lähettää isännälle kuitausviestin WM_DDE_ACK, joka sisältää

rengin yhteysikkunan osoitteen ja sovelluksen sekä tuetun teeman nimet. Jos isännän pyyntö koski kaikkia tuettuja teemoja, renki lähettää jokaisesta erillisen kuittausviestin.

Tiedon hakutapahtuman aikana isäntä hakee rengiltä alkion sisällön lähettämällä viestin WM_DDE_REQUEST. Viesti sisältää alkion nimen ja esitysmuodon tunnuksen. Jos renki tukee pyydettyä alkiota ja esitysmuotoa, se muodostaa alkiosta atomin ja lähettää isännälle viestin WM_DDE_DATA, joka sisältää atomin osoitteen. Renki voi sisällyttää viestiinsä isännälle käsittelyn kuittauspyynnön.

Jos renki ei tue pyydettyä alkiota tai esitysmuotoa, se lähettää

voi olla samanaikaisesti aktiivisena useita päivityspyynnöitä.

Isäntä voi valita kahdesta päivitystavasta: renki lähettää isännälle muuttuneen tiedon tai vain tiedon päivitystarpeesta. Tällöin isäntä sopivaksi katsomanaan ajankohtana päivittää tietonsa normaalilla hakutapahtumalla.

Isäntä lähettää rengille viestin WM_DDE_ADVISE, joka sisältää alkion nimen, esitysmuodon ja päivitystavan. Jos renki tukee päivitystä kyseisessä esitysmuodossa, se lähettää isännälle myönteisen kuittausviestin. Sen jälkeen renki on velvollinen havaitsemaan kyseisessä alkiossa tapahtuvat muutokset ja lähettämään niistä tiedon isännälle. Muussa tapauksessa

Komentojonon suoritustapahtuman aikana isäntä suorittuttaa rengillä komentosarjan. DDE-protokolla määrittelee ainoastaan komentojonon syntaksit. Itse komennot ja niiden parametrit ovat sovelluskohtaisia. Renki voi tätä kautta esimerkiksi tarjota muille ohjelmille liittymän omaan makrokieleensä.

Isäntä lähettää komentojonon rengille viestissä WM_DDE_EXECUTE. Renki suorittaa komennot ja lähettää isännälle myönteisen tai kielteisen kuittausviestin riippuen siitä, onnistuiko komentojen suoritus. Isäntä voi vapaasti valita, jääkö se odottamaan kuittausviestiä.

Sekä isännän että rengin on aina oltava valmiita **yhteyden**

SQLWindows-sovelluskehitin ei itse tue DDE:tä, mutta siinä on välineet, joilla DDE-tuki voidaan rakentaa kehitettävään sovellukseen.



Uffe Ekholm

isännälle kielteisen kuittausviestin WM_DDE_ACK. Tällöin isäntä voi pyytää samaa tietoa uudestaan toisessa esitysmuodossa. Tyypillisesti isäntä pyytää tietoa ensin "rikkaimmassa" esitysmuodossa ja tarvittaessa yksinkertaistaa esitysmuotoa asteittain.

Tiedon vientitapahtuman aikana isäntä vie rengin alkioon uuden sisällön. Isäntä muodostaa vietävästä tiedosta ensin atomin. Sen jälkeen se lähettää rengille viestin WM_DDE_POKE, joka sisältää alkion nimen, atomin osoitteen ja esitysmuodon tunnuksen.

Jos renki tukee kyseistä alkiota ja esitysmuotoa, se sijoittaa alkiolle uuden arvon ja lähettää isännälle myönteisen kuittausviestin. Muussa tapauksessa renki lähettää isännälle kielteisen kuittausviestin.

Päivityspyynnön alustustapahtuman aikana isäntä ilmoittaa rengille, minkä alkion muutoksista isäntä haluaa automaattisesti tiedon. Yhdessä yhteydessä

renki lähettää isännälle kielteisen kuittausviestin

Automaattisen päivitystapahtuman aikana renki ilmoittaa isännälle tiedon muuttumisesta isännän haluamalla tavalla.

Kun renki havaitsee muutoksen päivityspyynnön omaavassa alkiossa, se lähettää isännälle viestin WM_DDE_DATA. Viesti sisältää alkion nimen ja päivitystavasta riippuen mahdollisesti myös alkion uuden sisällön. Renki voi myös sisällyttää viestiin vaatimuksen, että käsiteltävään tiedon isäntä lähettää vastaukseksi kuittausviestin.

Kun isäntä haluaa **lopettaa automaattisen päivityksen**, se lähettää rengille viestin WM_DDE_UNADVISE. Viesti sisältää alkion nimen. Jos rengillä on merkittynä päivityspyyntö kyseiseen alkioon, se poistaa merkinnän ja lähettää isännälle myönteisen kuittausviestin. Jos kyseiseen alkioon ei ole merkitty päivityspyynnöä, renki lähettää isännälle kielteisen kuittausviestin.

purkamiseen. Sovellus on lopetettavaan suorituksen velvollinen purkamaan kaikki DDE-yhteytensä.

Sovellus ilmoittaa halukkuutensa yhteyden lopettamiseen lähettämällä toiselle sovellukselle viestin WM_DDE_TERMINATE. Viesti sisältää samalla lupauksen olla lähettämättä enempää viestejä ja luvan hävittää yhteyttä koskevat tiedot. Vastanottaja lähettää kuittauksena myös viestin WM_DDE_TERMINATE.

DDE käytännön sovelluksissa

DDE:tä hyödyntäviä sovelluksia on toistaiseksi vähän ja DDE-tuen taso vaihtelee suuresti. On kuitenkin selvää, että tulevaisuudessa kunnollinen DDE-tuki on yhä merkittävämpi sovelluksen mainosvaltti, kun kamppailaan markkinaosuuksista.

Liikevaihtoveron sisältävät tarjoushintamme ovat voimassa maaliskuun 1989 loppuun saakka.

Moniajossa sen salaisuus

MS-DOS:in verrattuna OS/2:n ehkä oleellisin uusi piirre on moniajo. Sen keskeisiä käsitteitä ovat istunto, prosessi ja tehtävä.

OS/2-käyttöjärjestelmässä voi ajaa useaa sovellusta samanaikaisesti. Suorituksen samanaikaisuus on kuitenkin tarkkaan ottaen näennäistä, sillä suoritin kykenee käsittelemään vain yhden tehtävän koodia kerrallaan. Samanaikaisuuden vaikutelma saadaan syntymään vaihtamalla suoritettavaa ohjelmaa riittävän usein ja nopeasti. Suorituksen siirtämisestä ohjelmasta toiseen huolehtii käyttöjärjestelmä.

OS/2 on keskeyttävä (pre-emptive) käyttöjärjestelmä, jossa ohjelma ei voi itse vaikuttaa suoritustuoroonsa. Se ei voi estää tai ennakoita keskeytystilaa, joka on ohjelmalle näkymätön tapahtuma. Tehtävän vaihtamiseen kuluu noin 850 mikrosekuntia, mutta keskeytysaika voi pahimmillaan venyä aina kahteen millisekuntiin asti.

Loogisella tasolla OS/2 tarjoaa käyttäjälle ja ohjelman tekijälle moniajoa kolmella loogisella tasolla. Uusia avainkäsitteitä ovat istunto, prosessi ja tehtävä.

Istunto

Istunto (Session) on virtuaalinen laitteisto, johon kuuluu virtuaalinen näyttö, näppäimistö ja hiiri. Käyttöjärjestelmä pitää samanaikaisesti yllä enintään 16 istuntoa.

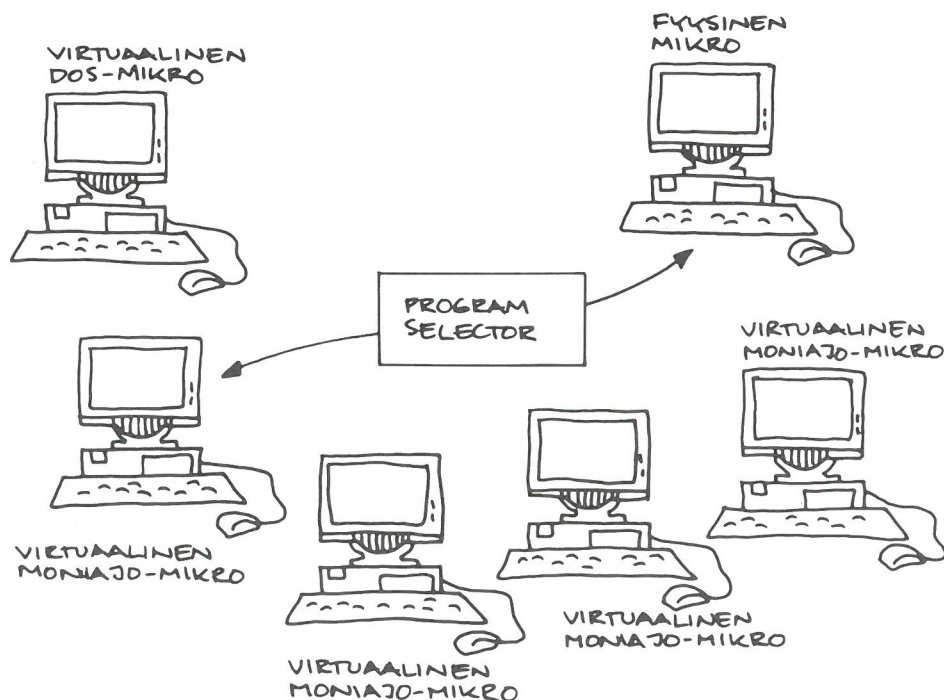
Jokaisella sovellusohjelmalla pitää olla istunto, ts. virtuaalinen laitteisto, johon ohjelma kuuluu. Ohjelmat lukevat virtuaalista näppäimistöä, kirjoittavat virtuaaliseen näyttöön ja seuraavat virtuaalisen hiiren liikkeitä. Todellisia fyysisiä laitteita on vain yksi kappale kutakin.

Käyttäjä kykenee ohjelmavalitsimen (program selector) avulla liittämään fyysisen laitteiston haluamaansa istuntoon. Ohjelmavalitsin toimii eräänlaisena "kanavanvalitsimena". Käyttäjä voi sen avulla käynnistää uusia istuntoja ja siirtyä istunnosta toiseen.

OS/2 versiossa 1.0 ohjelmavalitsimena toimii merkkipohjainen Session Manager ja versiossa 1.1 eteenpäin grafiikkapohjainen Windows Presentation Manager.

OS/2:n korkeintaan 16 istunnosta järjestelmä varaa kolme omaan käyttöönsä. Käyttäjä voi siten hyödyntää 13 istuntoa. OS/2 on kuitenkin yleensä konfiguroitu siten, että DOS-tilalle varataan yksi istunto, joka käynnistyy automaattisesti järjestelmän käynnistyksen yhteydessä. Varsinaisia OS/2:n moniajo-ohjelmille jää loppujen lopuksi maksimissaan 12 istuntoa.

Fyysiseen laitteistoon liitetyn istunnon sanotaan olevan edusnäytöllä. Muut istunnot ovat taustanäytöllä ja ohjelmat etenevät lukuunottamatta DOS-ohjelmaa. Kun DOS-ohjelma joutuu taustanäytölle, käyttöjärjestelmä pysäyttää sen suorituksen. Edetessään tausta-ajossa DOS-ohjelma saattaisi nimittäin sotkea edusnäytöllä olevan ohjelman tulostuksen.



DOS-tilan ohjelmaa ajettaessa suoritin viedään ns. real-tilaan, jossa ohjelma kykenee suoraan viittaamaan fyysiseen osoitteeseen (DOS-ohjelmille on tyypillistä muun muassa suoraan näyttömuistiin kirjoittaminen).

Prosessi

Jokainen istunto voi sisältää yhden tai useamman prosessin. Prosessi on OS/2-käyttöjärjestelmässä ajettavan ohjelman .exe-tiedoston ilmentymä. Se muodostuu kokoelmasta järjestel-

män resursseja ja vähintään yhdestä tehtävästä (ohjelman etenevästä osasta). Yksinkertaisimmillaan prosessin käsitteen voi rinnastaa MS-DOS-ohjelmaan.

Käyttöjärjestelmä suojaa prosessit tarkasti toisiltaan. Prosessi ei kykene vahingossa tai tahallaan häiritsemään toista prosessia. Poikkeuksena ovat ns. lapsiprosessit.

Ohjelma kykenee ohjelmallisesti käynnistämään toisen prosessin, jota nimitetään lapsiprosessiksi. Käynnistänyttä prosessia sanotaan äiti- tai isäprosessiksi. Äitiprosessi saa käynnistyksessä tietoonsa lapsiprosessin identifioivan tunnuksen ns. PID-luvun, jota käyttämällä äitiprosessi kykenee tarvittaessa esimerkiksi tuhoamaan lapsiprosessin.

Lapsiprosessi voi käynnistyä joko synkronisesti tai asynkronisesti. Synkronisessa käynnistyksessä äitiprosessi odottaa lapsiprosessin päättymistä ja jatkaa vasta sen jälkeen omaa etenemistään. Asynkronisessa käynnistyksessä lapsiprosessi "liipaitaan" käyntiin ja äitiprosessi saattaa jatkaa etenemistään täysin riippumatta lapsiprosessin tilasta. Asynkronisesti etenevät prosessit voivat säädellä toistensa etenemistä käyttäen hyväkseen erilaisia prosessien välisiä kommunikointimenetelmiä.

Tehtävät

Tehtävä (säie, thread) on yksikkö, jolla jaetaan suorittimen aikaa. Käyttöjärjestelmä ei siis jaa suoritinaikaa prosesseille, vaan tehtäville. Jokaisessa prosessissa on vähintään yksi tehtävä. Usean tehtävän prosessissa on useita itsenäisesti rinnakkain eteneviä ohjelman osia.

Kun ohjelma on tehty korkean tason ohjelmointikielillä kuten C-kieli, Pascal tai Ada, tehtävä vastaa asynkronisesti etenevää aliohjelmaa. Käynnistettäessä ohjelmassa on vain yksi tehtävä (= pääohjelma, main program), joka kykenee ohjelmallisesti käynnistämään muita tehtäviä.

Tehtävät eivät muodosta prosessin kaltaista jälkeläispuuta. Äiti-lapsi-suhteen puuttuessa tehtävät ovat lähinnä sisaruksia keskenään. Ainoan poikkeuksen muodostaa ykköstehtävä eli ensiksi käynnistetty tehtävä, jonka päättyessä kuolevat muutkin prosessin tehtävät.

Tehtävät jakavat keskenään prosessin resurssit. Jos esimerkiksi jokin tehtävä avaa tiedoston, saman prosessin toinen tehtävä voi käyttää hyväkseen saatua tiedostoa. Tehtävän "yksityisiä" resursseja ovat tehtävään liittyvät suorittimen rekisteriarvot, pino ja prioriteetti. Tehtäväkohtaisten rekisteriarvojen ja pinnon avulla käyttöjärjestelmä kykenee palauttamaan keskeytettynä olleen tehtävän ympäristön.

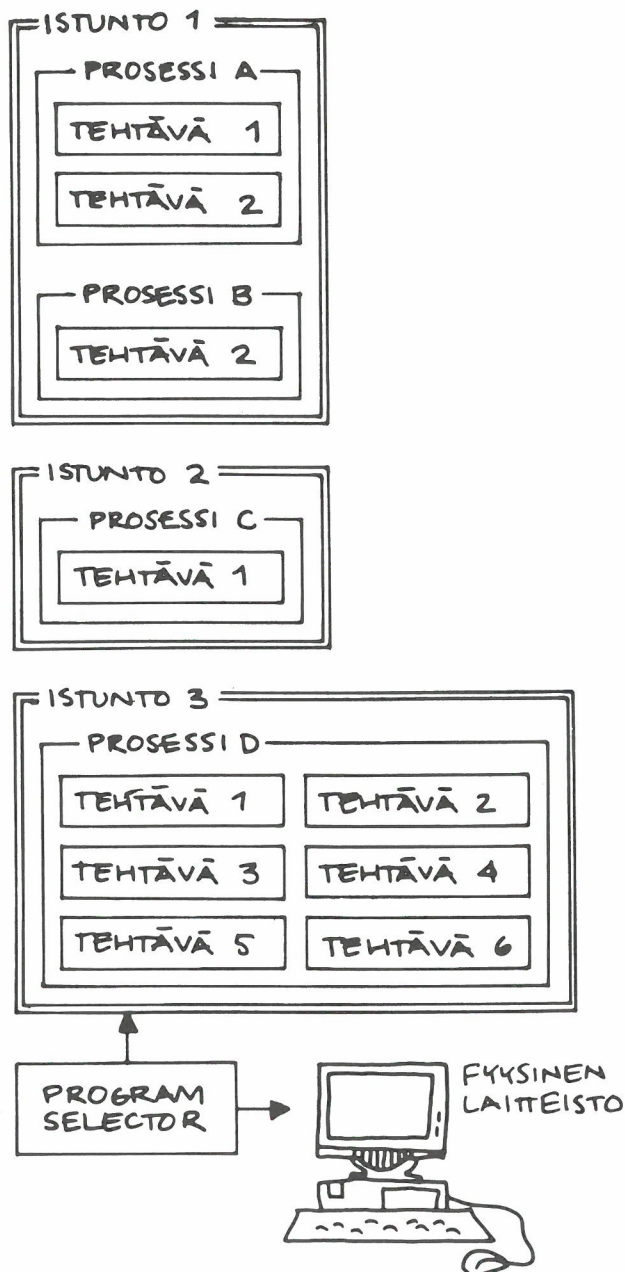
OS/2-käyttöjärjestelmässä voi samanaikaisesti olla korkeintaan 255 tehtävää, joista käyttöjärjestelmä varaa vähintään yhden itselleen. Oletusarvoisesti järjestelmä sallii enintään 64 tehtävää. Tehtävien maksimilukumäärä voidaan kuitenkin konfiguroinnilla nostaa. Lukumäärä asetetaan antamalla konfigurointitiedostoon määre `THREADS=<lukumäärä>`. Lukumäärä voi vaihdella välillä 16-255. Prosessia kohden suositeltava tehtävien maksimilukumäärä on 12. Prosessikohtainen ehdoton yläraja on 56 tehtävää. Kun jokainen prosessi vaatii vähintään yhden tehtävän, myös prosessien järjestelmäkohtainen yläraja on 255.

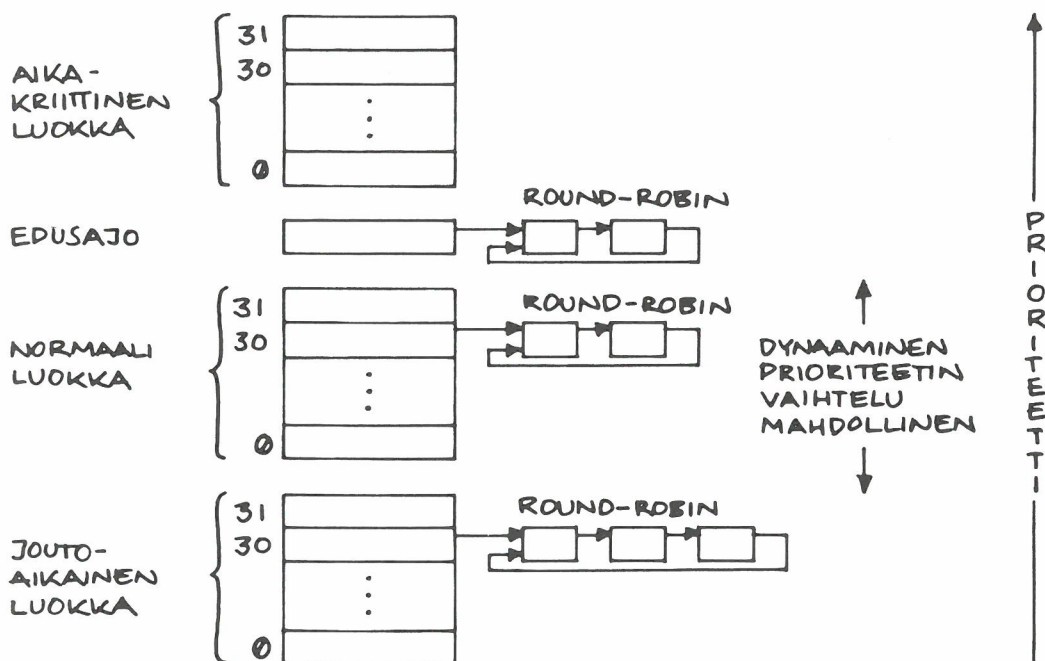
Priorisointi

Tavallinen osituskäyttöympäristö pohjautuu suur- ja minikone-ratkaisuille. Osituskäyttöympäristössä järjestelmän kallein resurssi on laitteisto, jonka käyttöä pyritään optimoimaan. Sitä varten kannattaa jopa palkata henkilökuntaa, jonka tehtävänä on pelkästään järjestelmän tehokkuuden ylläpito. Henkilöstön palkkaus ja hallintokulut saattavat silti muodostua pieniksi verrattuna tehotoman käytön aiheuttamiin kuluihin.

OS/2 rakentuu toisenlaiseen ajatukseen käyttöjärjestelmän ja laitteiston tehtävästä. OS/2 on suunniteltu nykyaikaiseen toimistoympäristöön henkilökohtaisten työasemien käyttöjärjestelmäksi. Laitteistokustannusten osuus ei pitkällä aikavälillä muodostakaan suurinta kustannuserää, vaan työryhmäarkkitehtuurissa kallein resurssi on järjestelmää hyödyntävä ihminen. Tärkeintä ei ole laitteiston, vaan "käyttäjän optimointi".

OS/2 pyrkii ennen kaikkea minimoimaan vastausviiveen. Käytön helpottuminen näkyy selkeästi OS/2:n tavassa priorisoida prosesseja ja tehtäviä. Prosessit





on jaettu kolmeen pääluokkaan: aikakriittisiin, normaaleihin ja joutoaikaisiin prosesseihin. Kussakin pääluokassa on 32 alitasoa. OS/2:n vuoronantaja jakaa suoritinaikaa kulloinkin korkeimmalla prioriteetillä oleville tehtäville.

Korkeimmalla tasolla olevat tehtävät vuorottelevat Round-Robin-periaatteen mukaisesti. Round-Robin-kierrossa mukana olevat tehtävät ovat ympyrämuotoisessa jonossa ja kukin saa vuorollaan yhtäpitkän suoritinvuoron. Suoritinvuoron pituus on jokin 31,25 millisekunnin monikerta, jonka OS/2 oletusarvoisesti asettaa. Käyttäjä voi asettaa rajat suoritinvuorolle konfiguroinnin yhteydessä. TIMESLICE-asetuksella määrätään tehtävän saama minimi- ja maksimiaika.

Aikakriittiset prosessit ovat nimensä mukaisesti prosesseja, joiden suorituksen pitää edetä tietty minimimäärä aikayksiköissä, tai joiden pitää nopeasti reagoida ulkoisiin tapahtumiin. Tällaisia prosesseja ovat esimerkiksi nopeaa tietoliikenneyhteyttä ylläpitävät ohjelmat. Aikakriittisten prosessien prioriteetti on aina normaaleja ja joutoaikaisia prosesseja suurempi.

Ohjelmat kuuluvat oletusarvoisesti normaaliin prioriteettiluokkaan. Erityisasemaan nousevat edusajossa olevat prosessit, joiden prioriteetti on tavallisia prosesseja suurempi. Ainoas-

taan aikakriittisten prosessien prioriteetti on edusajossa olevien normaaliin prosessien yläpuolella. Käytännössä ne ohjelmat, joiden kanssa käyttäjä on interaktiivisessa yhteydessä, saavat kaiken tarvitsemansa suoritinajan lähes heti. Tämä parantaa huomattavasti järjestelmän vastausnopeutta.

Tausta-ajossa olevat normaalit prosessit etenevät käytännössä kohtuullista vauhtia. Interaktiivisessa yhteydessä oleva ohjelma joutuu useimmiten odottamaan suurimman osan aikaa käyttäjän vastausta. Odotusajoina edusajossa oleva prosessi on pois Round-Robin-kierrosta. Tällöin alemmilla tasoilla olevat prosessit voivat edetä. Muita poissulkevia tilanteita ovat esimerkiksi levyhaut.

Joutoajokseen luokkaan kuuluvat prosessit saavat muilta luokilta jäävän suoritinajan. Tähän luokkaan kuuluvat prosessit saattavat esimerkiksi testata laitteistoa, laskea piin likiarvon tai suorittaa muuta vähemmän kiireellistä tehtävää.

Pelkästään korkeimmalla prioriteetillä olevien prosessien ajaminen saattaa johtaa tausta-ajossa olevat prosessit toivottoomaan tilanteeseen. Vahvasti suoritinta kuormittavat edusajossa olevat prosessit vievät alemmilla prioriteeteilta lähes kaiken suoritinajan.

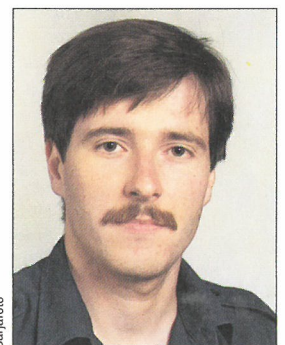
Jotta alhaisella prioriteetillä olevat prosessit eivät kokonaan

pysähtyisi, käyttöjärjestelmä asettaa ns. nälkiintymisrajan. Oletusarvoisesti nälkiintymisraja on kolme sekuntia, mutta se voidaan konfiguroinnissa määrittellä MAXWAIT-asetuksella muuksikin.

Kun aktiivinen prosessi on odottanut nälkiintymisrajan verran suoritinaikaa, käyttöjärjestelmä nostaa dynaamisesti sen prioriteettia. Nälkiintyneen prosessin prioriteetti kasvaa korkeimmalle tasolle ja saa suorittimen käyttöönsä yhden viipaleen (31,25 ms) ajaksi, minkä jälkeen prosessin prioriteetti putoaa takaisin alkuperäiseen tasoonsa.

Käyttöjärjestelmä vaihtelee normaaliin luokkaan kuuluvien prosessien prioriteettia myös tilastollisin perustein. I/O-painotteinen prosessi rasittaa suorittimen kuormitusta vähemmän kuin vastaavalla prioriteetillä oleva suoritinkeskeinen prosessi. Käyttöjärjestelmä saattaa järjestelmän tehokkuutta parantaakseen nostaa I/O-keskeisen prosessin prioriteettia muuttamalla yksiköllä ja vastaavasti laskea suoritinpainotteisen prosessin prioriteettia.

OS/2:n käyttämä dynaaminen prioriteettien vaihtelu on esitetävissä konfiguroinnissa. PRIORITY=absolute-asetuksella esitetään prioriteettien muuttumisen. Tavallisia sovelluksia käytettäessä käyttäjä ei kuitenkaan hyödy absoluuttisten prioriteettien käytöstä. ■



Sarjafoto

Koulutussuunnittelija Markku Mäkinen vastaa Nokia Datan tietotekniikan koulutuskeskuksessa NITECissa OS/2-koulutuksesta. Hän on opiskellut OS/2:ä muun muassa Microsoft University Training Centerissä Washingtonissa.

Ruori kauppiaan avuksi

Mikrossa toimivan Ruori-sovelluksen avulla kauppias seuraa tarkasti liikkeen raha-virtoja. Raportteja myynnin kehityksestä voidaan tulostaa vaikka päivittäin.

Tietokoneiden ohjaamat kassajärjestelmät ovat tulleet vähittäiskauppaan. Näin on mahdollista saada entistä enemmän ja monipuolisempaa tietoa kaupan toiminnasta kuten päivittäisestä myynnistä, kampanjoista ja kassalinjojen toiminnasta. MikroPOS-kassajärjestelmä jo sinänsä tarjoaa runsaasti mahdollisuuksia päivittäiseen, viikottaiseen ja kuukausittaiseen seurantaan ja tuottavuuden tarkkailuun.

Jotta kassajärjestelmän tuottamaa tietoa voitaisiin kuitenkin hyödyntää vielä tehokkaammin, tiedon jatkojalostusta varten on kehitetty MikroPOS-kassajärjestelmän yhteydessä toimiva Ruori-sovellus. Se on vähittäiskaupan päätöksenteon tukijärjestelmä, jolla kauppias voi suunnitella toimintaansa ja seurata tuloksen kehittymistä.



Studio Laine & Raunio

Grafiikkaa ja dynaamista raportointia

Ruori-sovellus toimii MikroMikko 3 tai 3TT-mikrotietokoneessa, joka on verkon välityksellä yhteydessä kassajärjestelmään. Ruori-mikron voi myös irrottaa verkosta ja kauppias voi käyttää sitä esimerkiksi tekstinkäsittelyyn. Lisäksi kauppias tarvitsee raporttikirjoittimen sekä taulukkomuotoisten raporttien että tekstien ja kuvien tulostamiseen.

Ruorin käyttöä opastavat suomenkieliset valikot. Virhetilanteistakin selvittää näytön tulostuvien ohjeiden avulla. Ruorin ominaispiirteitä ovat dynaaminen raportointi ja havainnollinen grafiikka.

Taulukkomuotoisesta aineistosta tuotetaan pylväsdiaگرامmeja valikoiden avulla. Valikois-

ta määritellään muutamalla näppäinpainalluksella diagrammi, joka tulostuu joko näyttöön tai kirjoittimelle. Esimerkiksi budjetoitaessa luvut voidaan siirtää taulukoista diagrammeiksi ja näin lukea selkeät kehitystrendit kuvasta.

Myös aineiston lajittelu käy kätevästi. Esimerkiksi päivän kampanjat voi järjestää tuoton ja tuottoprosentin mukaan ja verrata tuottoprosenttiltaan parhaita tuotteita markkamääräisesti eniten tuottaneisiin tuotteisiin. Yhtä tarkka seuranta olisi pelkän laskimen avulla melkoisen mahdollista.

Kaiken näytössä näkyvän saa tulostettua paperille. Raportteja voi tulostaa eri tavoin lajiteltuna. Näin samasta aineistosta voi tulostaa monenlaisia raportteja. Taulukkomuotoisia raportteja täydentävät samasta aineistosta tuotetut kuvat.

Tiedot siirretään Ruori-sovellukseen

Kassajärjestelmästä kerätään päivittäisiä myynti- ja katetietoja tavararyhmittäin, päivän kampanjoista tuotteittain ja koko liikkeen osalta muun muassa päivän asiakasmäärä. Tiedot siirretään kassajärjestelmästä Ruorille muokattavaksi taulukkolaskentaohjelmiston ymmärtämään muotoon. Ruori-sovellus siis kerää tiedot kassajärjestelmästä, siirtää tiedot Ruori-mikroon ja muokkaa tiedot taulukkolaskentaohjelmistolla.

Tietojen keräily on toteutettu Ada-ohjelmointikielellä, siirtoa ja muokkausta kontrolloivat MS-DOS-komentojonot ja käyttöliittymässä on käytetty sekä MPSForms-näytönkäsitteilytyökaluja että Lotus 1-2-3 -taulukkolaskentaohjelmistoa. Käyttäjälle näkyvä osa eli varsinainen tietojenkäsittely perustuu Lotus 1-2-3:een. Sen käyttöä ei kuitenkaan tarvitse opetella, vaan käytössä opastavat valmiit suomenkieliset valikot.

Kun Ruori-mikro asetetaan tiedonsiirron valmiustilaan, se kerää ja siirtää tiedot kassajärjestelmästä Ruori-sovellukseen yön aikana. Odottamattomat häiriöt, kuten sähkökatkokset eivät hävittä tiedoja, vaan ne voidaan siirtää seuraavana päivänä.

Tiedot tallennetaan laskenta-alustoille

Kun tiedot on siirretty Ruori-mikroon, ne tallennetaan Lotus 1-2-3:n ymmärtämään muotoon ns. laskenta-alustoille. Alustoja on neljä perustyyppiä: osastobudjetti, tavararyhmät, kampanjat ja kokonaismyynnin keräyslista.

Osastobudjettialusta sisältää osastotason myynti- ja katetiedot edelliseltä ja kuluvalta vuodelta sekä budjetit kuluvalle ja tulevalle vuodelle. Osastoalustojen tiedot kumuloidaan viikko- tai kuukausitaso tiedoiksi.

Tavararyhmäalustan tiedot ovat nimensä mukaisesti tavararyhmätason myynti- ja katetietoja, jotka on jaoteltu normaali- ja kampanjamyyntiin sekä kokonaismyyntiin. Tiedot kerätään viikottain tai kuukausittain. Li-

säksi asennusvaiheessa voidaan määritellä haluttu ajanjakso, jolta tiedot kerätään.

Kampanja-alusta sisältää yhden päivän kampanjat tuotteittain. Siitä näkee osto- ja myyntihinnan, tuoton, tuotto-prosentin, myynnin kappaleittain ja markkoittain sekä panostuksen.

Kokonaismyynnin keräyslista on juokseva lista, joka sisältää koko liikkeen päivittäiset myyntitiedot puolen vuoden ajalta. Listaan lisätään päivittäin yksi rivi ja vanhin rivi jää pois. Normaali- ja kampanjamyyntiin sekä kokonaismyynnin lisäksi keräyslista sisältää tunnuslukuja kuten keskiostos/asiakas, kappaletta/asiakas, myynti/työtunnit ja myynti/neliometri. Listalta näkee myös päivittäisen asiakasmäärän. Lisäksi alustalla on kenttä kommenttia varten. Näin

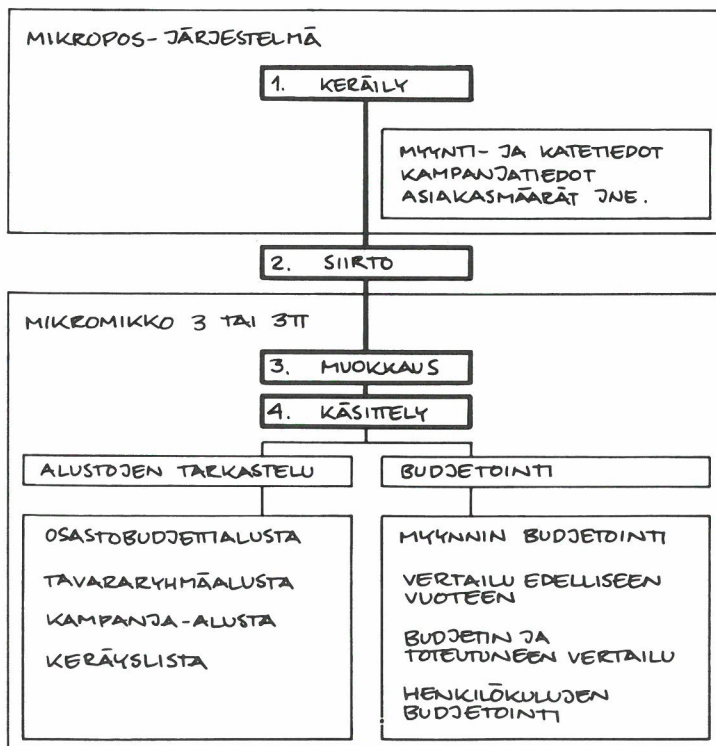
jos esimerkiksi tulos on jostain erityisestä syystä ollut poikkeuksellisen hyvä, syyn voi kirjata raporttiin.

Ruorin budjetointiosassa voi tehdä viikko- ja kuukausibudjetteja vanhojen tavararyhmäalustojen pohjalta. Vertailutiedoksi voi valita haluamansa ajanjakson tiedot, jolloin voi esimerkiksi budjetoida pääsiäisviikon myynnin edellisen vuoden pääsiäisviikon perusteella huolimatta sen ajoittumisesta kalenterilla. Myynnin lisäksi voi budjetoida henkilökuluja osastoittain.

Tavararyhmätason tietoja voi vertailla mihin tahansa aiempaan ajanjaksoon, esimerkiksi tammikuun 1988 myyntiä tammikuun 1987 myyntiin. Myös budjetoitua ja toteutunutta myyntiä voi vertailla. ■

MikroPOS-kassajärjestelmä on esitelty Softarissa 3/1988.

Ruorin päätoiminnot ovat tiedon keräily, siirto, muokkaus ja käsittely.



Katri Rekola työskentelee Nokia Datan vähittäiskaupan järjestelmät -osastolla ATK-suunnittelijana.

Veikko Kehä &
Lassi Juutinen

Learning Ada

Part 2

In Part 2 we step aside from the language itself, and discuss the suitability of Ada for the Object Oriented Design (OOD) method and construction of software components.

Object-oriented methods have become very popular in recent years. Despite this popularity, no clear definition of 'object-oriented' has been given. By Object Oriented Programming (OOP) we mean programming with languages like Smalltalk, Objective-C, C++, and Eiffel. Features typical of these languages include classes, inheritance, and dynamic binding.

A class is a description of the properties of a set of objects. Inheritance means that an object belonging to a subclass inherits the properties of the superclass, perhaps with the addition of new properties and/or modification of existing ones. The complete set of properties of an object is determined dynamically during runtime. This may lead to unexpected problems in program execution because the compiler is unable to find all the connections between the objects.

So much for OOP. Below we will concentrate on the object-oriented design method and Ada. The point is that programming languages without inheritance still permit OOD.

Object Oriented Design

OOD is a design method, so the target language is not necessarily fixed. Programming languages ideal for OOD include Modula-2, Alphard, CLU, and Ada. These can be called object-based languages.

Well-structured systems developed with traditional languages (FORTRAN, COBOL, C, Pascal) tend to consist of collections of subprograms because these are the only major building blocks available. Program design often starts by defining the functional properties of a system. These languages are best

suited to techniques that concentrate upon algorithmic abstractions. Description of abstract objects is often impossible.

The basic idea behind OOD is to get as close a relationship between the real-world problem and the program as possible. Each module in the system corresponds to a logical entity (object) in the problem space. This makes the program structure much clearer.

When OOD is used, real-world data are important. In the early phases of program design computer technology is not considered at all. First we try to find the objects and the data connected with each object. The data are analysed and then we find the operations, but system building is always based on data, not on operations.

To facilitate the handling of similar objects in our programs, it is useful to define the term **abstract data type** (ADT). An ADT is a model from which several similar objects can be created by declaring variables of that type.

An object is an entity that has an internal state. This is different from subprograms, which are created every time they are called. An object does not **do**

something; it **is** something! An object is characterized by the operations that can be performed upon it, and the operations it requires of other objects. An object is a unique instance of an ADT; each object is known by its own name.

In practical data-processing terms, an object is usually a data structure plus a set of operations that can be applied to that structure. In Ada, an abstract data type is implemented with a package. The form of the data structure is defined by a private data type, and the operations by subprograms (procedures or functions).

An object in Ada is either:

1. A variable of an ADT. An object of this kind is created with a variable declaration. The type of a variable is defined in the visible part of a package. Several variables can be declared, all of which represent different objects. For example, variables of the type `GENERAL_TREE`. `TREE_TYPE` (figure 2) declared in the user program are objects of this kind.

2. A package implementing a state machine. This means that there is no data type imported from the package. It only defines the operations that can be

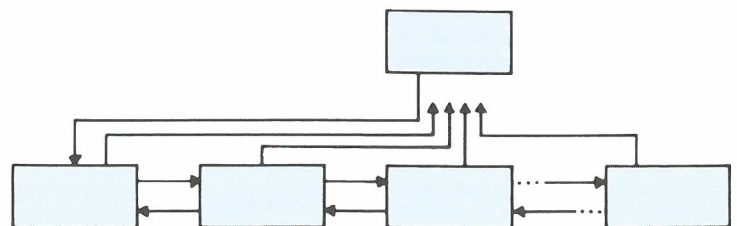


Figure 1: General tree structure.


```

package GENERAL_TREE is           — Visible part
  type TREE_TYPE is limited private;
  subtype CONTENTS_TYPE is STRING (1..10);
  procedure CREATE (
    TREE      :in out TREE_TYPE;
    CONTENTS  :in CONTENTS_TYPE);
  — Creates a new root with no subtrees.

  procedure MOVE_CHILD (
    FROM      :in out TREE_TYPE;
    TO        :in TREE_TYPE);
  — Moves a tree or subtree to be the leftmost subtree
  — of a given node.

  procedure MOVE_RIGHT (
    FROM      :in out TREE_TYPE;
    TO        :in TREE_TYPE);
  — Moves a tree or subtree to be the right sibling
  — of a given node.

  function PARENT (TREE : TREE_TYPE) return TREE_TYPE;
  — Returns a link to the father of a given tree (provided
  — that the father is not the root of the tree in which
  — case null is returned).

  function LEFTMOST_CHILD (TREE : TREE_TYPE)
    return TREE_TYPE;
  — Returns a link to the leftmost subtree.

  function LEFT_SIBLING (TREE : TREE_TYPE) return TREE_TYPE;
  — Returns a link to the subtree in the left.

  function RIGHT_SIBLING (TREE : TREE_TYPE) return TREE_TYPE;
  — Returns a link to the subtree in the right.

  function GET (TREE : TREE_TYPE) return CONTENTS_TYPE;
  — Returns contents of a given node.

  function EMPTY (TREE : TREE_TYPE) return BOOLEAN;
  — Returns value TRUE if a given node does not exist.

  TREE_ERROR : exception;
  — Raised if attempted to move a null tree.

private
  type NODE_TYPE;           — Private part
  type TREE_TYPE is access NODE_TYPE;
  type NODE_TYPE is
    record                   — This could be in
                           — the package body!
      CONTENTS : CONTENTS_TYPE;
      PARENT   : TREE_TYPE;
      CHILD    : TREE_TYPE;
      LEFT     : TREE_TYPE;
      RIGHT    : TREE_TYPE;
    end record;
end GENERAL_TREE;

```

applied to the data structure in the package. Here, one package corresponds to one object.

Program Construction

Here are the basics of an algorithm describing how the program is constructed using OOD. Everything is based on building the model of the real world from the client's point of view.

1. Describe the data in the real world. Entity-relationship (ER) diagrams can be used, or wall chart techniques can be applied to describing the real-world things. Objects can be found by grouping together data that logically belongs together. For example, a customer may be an object consisting of his number, name, and address.

Note: the main program is not an object!

2. Find out if previously made objects can be used to describe some of these objects.

3. Identify the operations offered and required by each object. For example, 'create a customer', 'change customer address'.

4. Establish the relationships between the objects. Draw a graph in which each object is a node, and an edge means that an object uses a service offered by another object. A service can mean an operation (subprogram), or an abstract data type.

5. Establish the interface of each object. Using Ada, write the specification of each object.

6. Implement each object. This means writing the program code for the package bodies.

7. Write the main programs forming the required software. This should be straightforward; it involves declaring objects and using their operations.

Figure 2: Specification of the general tree package.

Maintenance of programs designed with OOD is usually easy, because small changes in the real world correspond to small and local changes in the program. It is worth noting, too, that, when we implement objects, we produce program components which may be re-usable in future projects.

Software Components

A software component is the implementation of a general purpose object or algorithm. When programs are developed using OOD, components (or objects that can be generalized into components) are created and used more naturally than with conventional methods. They can be re-used (perhaps slightly modified) in a different environment.

Components are often parameterized, because this adds to their generality. When Ada is used, the components are mostly generic packages (or subprograms). Instead of a single component, we often have a family of components, in which each member has the same basic functionality but each component has unique time and space characteristics (static vs. dynamic, garbage collected or not). Suitable components include packages handling conventional data structures: stacks, queues, and trees. On the other hand, an object implementing a customer is not always a good component because it means completely different things to a butcher and a doctor.

To be usable, a component must denote a single abstraction, and have a narrow interface with the environment. This is almost automatic, when OOD is used. To be reusable a component must be maintainable, efficient, reliable, understandable, and, of course, correct. Errors must also be treated consistently.

Ada supports components in several ways: first, the language is standardized, and subsets and supersets are not allowed. This makes portability of units easier, and it is a necessity for the commercial component industry. Secondly, separate compilation makes it easy to add new units to the program library. Compiling a component does not differ at all from compiling a user

made compilation unit. Thirdly, the generic features of Ada provide a certain level of parameterization.

Components can be regarded as extensions of the programming language: the ADTs provided by components behave like predefined data types in the language, but the operations in the components are usually on a much higher level.

Exercise

Design a program that reads a text file and counts and displays the number of lines, words, and characters in that file. What packages are needed and what operations do they have? What are the parameters of the operations? Design the packages so that they can be reused in some other program. Send your answers to the editor of Softari. The best answer will be published in the next issue. ■

Answer to the exercise

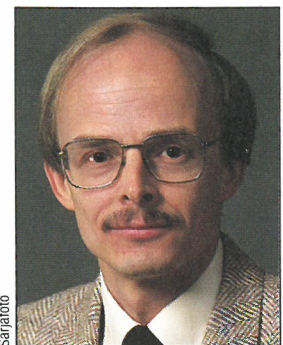
The last number contained an exercise on how to modify a binary tree package into a general tree package. A binary tree has two subtrees at the most; a general tree can have more subtrees. A first attempt might be to add more links to the subtrees. But the problem in this case is how to fix the maximum number of subtrees.

A better solution is to implement the node as a record containing a link to the left-most subtree. Every node also has links to its left and right-hand brothers and to its father. The structure is illustrated in figure 1.

This structure allows one node to have from zero to any number of subtrees. The operations, too, are slightly different. The specification of a general tree package is shown in figure 2. It has not changed much, but the changes needed in the package body are greater.



Veikko Kehä työskentelee Nokia Datan Tuotekehityksessä menetelmäpäällikkönä.



Lassi Juutinen työskentelee Nokia Datan Tuotekehityksessä menetelmäasiantuntijana.

There are some commercial Ada component manufacturers:

EV Software Engineering, Inc., Rockville, MD 20850 USA.

Grady Booch, Wizard Software, 835 S. Moore St., Lakewood, CO 80226 USA.

Ada Software Repository; Richard Conn, Echelon, Inc., 885 N. San Antonio Road, Los Altos, CA 94022 USA.

Mariadata, Box 5078, 141 05 Huddinge, Sverige.

References

- /1/ Reference Manual for the Ada Programming Language, January 1983.
- /2/ Booch, G.: Software Engineering with Ada, Benjamin/Cummings, 1986.
- /3/ Barnes, J.G.P.: Programming in Ada, Addison-Wesley, August 1983.
- /4/ Booch, G.: Software Components with Ada, Benjamin/Cummings, 1987.
- /5/ Meyer, B.: Object-oriented Software Construction, Prentice Hall, 1988.

Omia sovelluksia Windows-ympäristöön

Vaikka graafinen Windows-ympäristö on saatavuttamassa standardin käyttöliittymän aseman DOS-pohjaisissa mikrotietokoneissa, niin kauan on jouduttu odottamaan Windows-sovelluskehittämiä.

Ensimmäisten Windows-sovelluskehittäjien joukossa on SQLWindows, jossa on lisäksi SQL-liittymä. Oman erillisen tietokantajärjestelmän sijasta SQLWindows hyödyntää työryhmäverkon SQL-tietokantapalveluja.

Graafiset käyttöliittymät ovat osoittaneet käyttökelpoisuutensa eri ympäristöissä. Graafisten käyttöjärjestelmien yhtenäiset ja havainnolliset esitystavat sekä käyttömuodot ovat alentaneet samaan ympäristöön rakennettujen sovellusten oppimiskynnystä ja helpottaneet niiden hyödyntämistä. MS-DOS-ympäristön graafisen käyttöjärjestelmän standardiksi on muodostunut Microsoft Windows.

Windows-ympäristön suurimpana ongelmana on pitkään ollut Windows-ohjelmoinnin monimutkaisuus. Vaikka loppusovelluksista muodostuukin havainnollisia graafisine esityksineen, niin ohjelmien tekeminen on epähavainnoillista ohjelmakoodin kirjoittamista. Windows-pohjaisten sovelluskehittäjien tarve on mitä ilmeisin. Yhtenä ensimmäisiä on SQLWindows.

Sovelluksen ikkunat maalataan

SQLWindowsissa sovellusten ulkoasu piirretään tyhjiin sovellusikkunaan hiirellä. Sovellusikkunaan voidaan piirtää Windowsin määrittelemiä objekteja kuten tekstikenttiä, rullattavia teksti-ikkunoita, erilaisia osoitinnappuloita (button) ja taustatekstejä sekä -kuvia. Myös sovellusikkunoita käsitellään objekteina. Eri objekteja, siis myös sovellusikkunoita, voi yhdessä sovelluksessa olla useita.

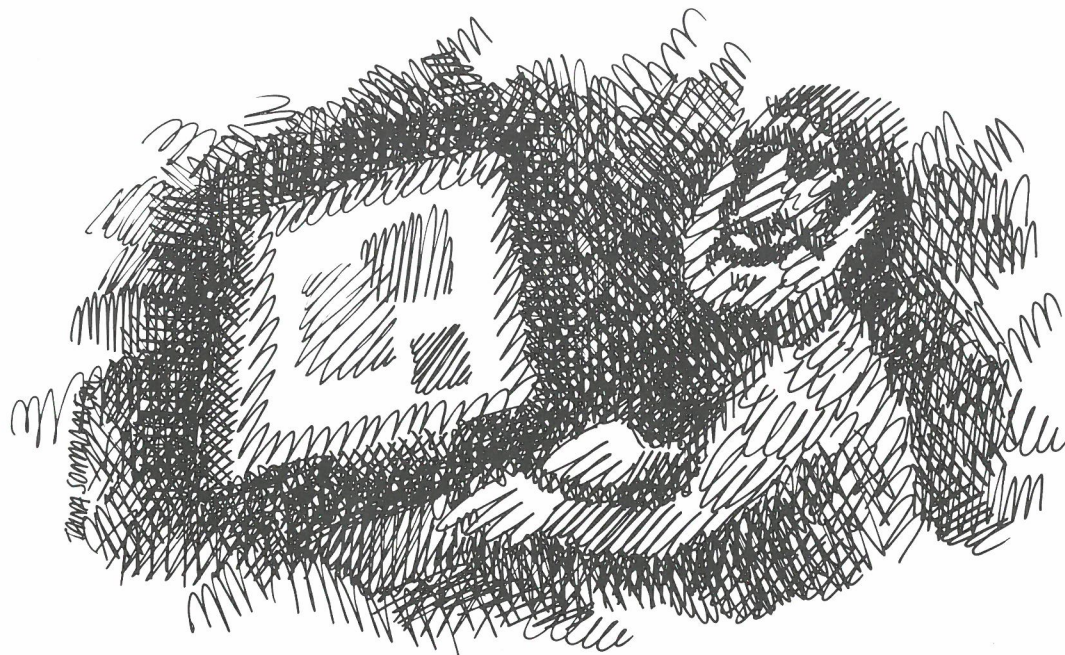
Hiiren avulla muutetaan objektien sijaintia ja kokoa sovelluskehityksen aikana. Objektien kokoa muutetaan aivan kuten Windowsin esitysgrafiikkaohjelmissa: tartutaan kohdistimella objektin reunaan ja venytetään.

Sovelluksen ulkoasua maalatessa kirjoittautuu sitä vastaava koodi kehitysaikana esillä olevaan koodi-ikkunaan. Sovellusikkuna sekä koodi-ikkuna ovat tiukasti integroitu toisiinsa siten, että toiseen tehty muutos näkyy välittömästi toisessa. Esimerkiksi jos kentän sijaintia muutetaan sovellusikkunassa, niin kentän sijaintikoordinaatit koodi-ikkunassa muuttuvat samanaikaisesti.

Piirretyistä objekteista generoituu koodi-ikkunaan kaikki tarpeellinen tieto: tyyppi, sijainti ja koko. Piirtämisen jälkeen on sovelluskehittäjän kirjoitettava objektin yksilöivä nimi koodi-ikkunaan. Piirrettyjä objektikenttiä käsitellään sovelluksessa muuttujina, joiden arvot näkyvät ajonaikaisesti vastaavien kenttien sisältönä. Näitä kenttämuuttujia voidaan käsitellä kuin mitä tahansa muuttujia SQLWindowsin sisällä.

Logiikka korkean tason kielellä

Sovelluskehittäjä voi määritellä rutiineja, jotka käynnistyvät valikossa olevista valinnoista. Lisäksi sovellukseen piirrettyihin objekteihin voidaan sisällyttää



toimintoja, jotka käynnistyvät asianomaiseen objektiin suoritetun toiminnon seurauksena. Esimerkiksi käyttäjän tekemiä kenttään kohdistuvia syötteitä varten voidaan määritellä syötteiden tarkistusohjelma varmistamaan, etteivät annetut arvot ole virheellisiä. Tarkistusohjelma voidaan määritellä käynnistyväksi aina kun kenttään kirjoitetaan tai siitä poistetaan kirjain. Vaihtoehtoisesti se voi käynnistyä vasta kun kentästä poistetaan syötön lopetuksen merkiksi.

Toimintoja, joihin saadaan myös liitettyä rutiineja, ovat kenttään saapuminen ja ikkunan avaaminen, siitä poistuminen, sen aktivoiminen tai johonkin ikkunan sisällä olevaan kenttään kirjoittaminen. Toimintorutiineihin voidaan näin esimerkiksi kirjoittaa monimutkaisiakin tarkistusali ohjelmia, joissa voi käyttää kaikkia SQLWindowsin sovelluskehityskielen komentoja ja funktioita.

SQLWindows-rutiinien logiikka kirjoitetaan SQLWindowsin omalla neljännen sukupolven ohjelmointikielellä SAL (SQLWindows Application Language). Se sisältää peruskomennot sovellusrutiinien suoritukseen (IF, ELSE, LOOP, CALL, RETURN) sekä muuttujan käsittelyyn SET-komennon. Suorituksen ohjauskomentojen lisäksi on käytettävissä toista sataa funktiota. SQLWindows sisältää SQL-funktioiden lisäksi päiväys-, teksti- ja taulukkomuuttujien käsittelyfunktioita ja sovellusikkunoiden käsittely- sekä hallintafunktioita. Syöttötietojen tarkistukseen ja ohjelmien välisen yhteyden muodostamiseen on omat funktionsa.

Ohjelmien väliset yhteydet

mahdollistavat DDE-funktiot ovat yhteisiä eri Windows-ohjelmille ja niiden avulla eri ohjelmat ja sovellukset voivat kommunikoida keskenään. Esimerkiksi SQLWindows-sovelluksen hakemat tietueet voidaan ohjata kopioitumaan samanaikaisesti toiminnassa olevaan Excel-taulukkolaskentaohjelman laskentapohjaan. DDE:n avulla voidaan myös käynnistää muita ohjelmia Windowsin sisällä. Toistaiseksi Excelin ja SQLWindowsin yhdistäminen on vain teoriaa, sillä ne eivät mahdu samanaikaisesti toimimaan Windowsin kanssa.

Prototyypistä lopulliseen sovellukseen

Sovelluksia voi SQLWindowsin avulla kehittää myös protoillen niin, että maalattua sovellusta pidetään ensimmäisenä prototyypinä. Sitten sovellukseen voidaan kirjoittaa piirrettyihin objekteihin liittyviä rutiineja ja näin osa osalta kasvattaa yksinkertaisesta, ensimmäisestä prototyypistä lopullinen sovellus. Aina rutiineja lisäämällä päästään seuraavaan prototyyppiin, kunnes viimeisessä vaiheessa sovellus on valmis ja loppukäyttäjän hyväksymä. Käyttämällä protoilua ja loppukäyttäjää protojen arvioijina voidaan varmistaa, että sovelluksesta tulee toivotunlainen.

SQL-liittymällä omat funktiot

Tietokantaliittymää varten on SQLWindowsissa käytettävissä joukko SQL-funktioita. Periaatteessa liittymä on ohjelmointikielten liittymän kaltainen.

Alussa muodostetaan yhteys joko omaan tietokoneeseen tai lähiverkon tietokantapalvelimessa olevaan tietokantaan. Tämän jälkeen ovat kaikki SQL-lauseet käytettävissä. SQL-lauseet kirjoitetaan sellaisenaan tekstimuodossa parametriksi tietokantapalvelimelle lähetettävälle SQL-Prepare-funktiolle. Sovelluksen muuttujia voidaan liittää suoraan SQL-lauseeseen. Yhteys muuttujien ja SQL-lauseen välillä on tällöin dynaaminen eli muuttujissa tapahtuva muutos näkyy välittömästi myös SQL-lauseessa.

Haku tietokannasta tapahtuu myös suoraan sovelluksen muuttujiin. SQLWindows käyttää hakukomennoissa SQLBasen tulostulokseja käsittelyä. Eli haun tuloksesta muodostetaan tietokantapalvelimeen tulostaulukko, jota työasemasta voidaan sitten selata tietueittain sekä eteen- että taaksepäin. SQLWindowsin SQLBase-liittymä hoitaa haettujen tietojen tyyppimuunnokset SQLWindowsin muuttujiksi.

SQLWindowsin SQLBase-liittymä on toteutettu kokonaisuudessaan sovelluksiin liitettävän dynaamisen funktiokirjaston avulla. Itse SQLWindows ei ole riippuvainen SQLBasesta, vaan voi toimia ilman sitä. Rakentamalla sopivia funktiokirjastoja voidaan SQLWindows mahdollisesti tulevaisuudessa liittää myös muihin tietokantajärjestelmiin.

Tarvittaessa C-kielisiä aliohjelmia

SQLWindowsissa on mahdollista määritellä globaaleja, sovelluksissa käytettäviä vakioita ja aliohjelmiä, jotka ovat kaikkien

sovelluksen rutiinien käytettävissä. Aliohjelmat voivat olla SAL-sovelluskehityskielellä tehtyjä, mutta myös ulkopuoliset C-ohjelmoitokielellä tehtyt aliohjelmat ovat käytettävissä. C-funktioiden tekemiseen ja liittämiseen SQLWindows-sovellukseen tarvitaan Windowsin kehitysympäristö Software Development Kit.

SQLWindows-sovelluksen käyttämät muuttujat, jotka eivät ole mihinkään kenttään sidottuja, määritellään myös alussa koko sovellukselle yhteisiksi. Mahdolliset muuttujien tyypit ovat: numeerinen, boolean, päiväys/aika, merkkijono ja yksiulotteinen taulukko.

Ajonaikainen versio SQLRunilla

Kun sovellus on saatu valmiiksi, se käännetään valitsemalla SQLWindowsin tallennusmuotovalikosta käännetty (Compiled) ja tallentamalla sovellus. Tämän jälkeen sovellus on ajettavissa SQLWindowsin ajonaikaisella versiolla SQLRun. Se hoitaa sovelluksen hallinnan ja sisältää kaikki SQLWindowsin funktiot. Funktiot ovat siis ajonaikaisessa kirjastossa, josta käännetty sovellukset niitä kutsuvat.

Sovelluskehittäjälle joukko apuvälineitä

SQLWindows sisältää monipuolisia sovelluskehitystä tukevia apuvälineitä. Sovelluskoodi tarkastetaan jo kirjoitusvaiheessa. Jos kirjoitettu rivi ei ole oikeellinen, SQLWindows ilmoittaa siitä välittömästi, kun sovelluskehittäjä on saanut rivin valmiiksi ja on siirtymässä seuraavalle. Joka kohdassa voi sovelluskehittäjä pyytää valintaikkunan esille, joka näyttää kaikki vaihtoehtoiset komennot koodi-ikkunassa olevan kohdistimen osoittamaan kohtaan.

SQLWindowsissa on myös apuvälineitä sovelluksen testaamiseen. Yleensä sovelluskehittämissä ei ole mitään apuvälineitä, joilla sovelluksen toimintaa voitaisiin tutkia virhetilanteiden selvittämiseksi. SQLWindowsin debuggeri kuitenkin sisältää mahdollisuuden seurata suorituksen kulkua koodi-ikkunassa sovelluksen suorituksen aikana. Myös keskeytyskohdat (break points) on mahdollista määritellä,

jotta muuttujien arvoja ja sovelluksen tilaa voitaisiin tutkia halutuissa sovelluskoodin kohdissa.

Käyttökelpoisia lisäohjelmia

Raporttien tekemiseen ei SQLWindowsin ensimmäisessä versiossa ole välineitä, mutta siitä voidaan käynnistää suoraan SQR Report Writer, joka on SQLBaseen kuuluva raportin muodostamisohjelma. SQR:ssä on oma komentokieli, joka sisältää ohjelmarakenteet, SQL-lauseet, muuttujien käsittelyn ja tulostuksen ohjauksen. SQR muodostaa komentokielisestä tiedostosta halutunlaisen raportin tulostiedostoon. Raportti voidaan ohjata tulostumaan myös näyttöön. Tulostiedostoa voidaan jatkokäsitellä tai tulostaa sellaisenaan kirjoittimelle.

SQLWindowsin toimitus sisältää lisäksi vuorovaikutteisen Windows-ympäristössä toimivan tietokannan käsittelyohjelman WinTalkin, jolla käyttäjä voi muodostaa SQL-lauseita tietokantapalvelimen suoritettavaksi. WinTalk on toiminnallisesti identtinen SQLBase-toimitukseen kuuluvan merkkipohjaisen SQLTalkin kanssa. WinTalkia voidaan käyttää tietokannan luomiseen, ad hoc -raporttien tekemiseen ja sovelluksissa käytettävien lauseiden testaamiseen. WinTalkissa ovat käytettävissä kaikki SQL-komennot sekä niiden lisäksi ympäristökomentoja tulostuksen ohjaukseen. WinTalkin avulla käyttäjä voi lähettää kerrallaan suoritettavaksi kuitenkin vain yhden SQL-lauseen.

Lisämuistin käyttö välttämätöntä

Windows-ohjelmien käytössä muodostuu helposti ongelmaksi muistitila. Kun käytetään useampia ohjelmia samanaikaisesti, lisämuistin käyttö onkin välttämätöntä.

Jotta lisämuistia pystyttäisiin hyödyntämään, ohjelmien pitää osata käyttää sitä. Monet Windows-ympäristöön kehitetyt ohjelmistot, kuten SQLWindows, osaavat hyödyntää lisämuistia ja voivat olla samanaikaisesti toiminnassa, vaikka niiden viemä muistitila on paljon suurempi kuin DOS:n käyttämä 640 kilotavua. ■

Muistin sokkeloissa

Mikroilija joutuu usein käsittämättömän sanaviidakon keskelle, kun hän yrittää tilata lisämuistia. Tässä artikkelissa selvitetään lähinnä extended-muistiin liittyviä käsitteitä.

MikroMikkojen muistia ja erillisiä lisämuistikortteja koskevat käsitteet aiheuttavat silloin tällöin epäselvyyksiä, jonka takia hankitaan vääräntyyppisiä muistikortteja.

Tekstin luettavuuden kannalta on ongelmallista, että kaikille englanninkielisille käsitteille ei ole virallisia, yksiselitteisiä suomenkielisiä vastineita. Toivon lukijoiden antavan anteeksi tämän tyylihämmästyksen.

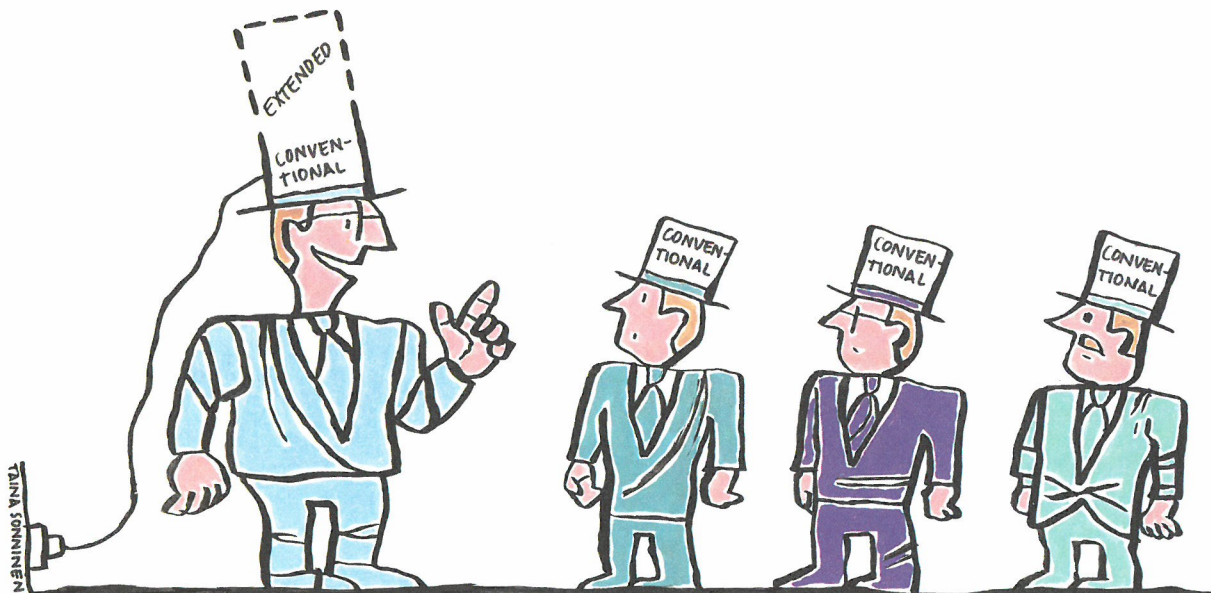
Osoiteavaruus jakautuu osiin

Suorittimen osoitettavissa olevaa muistialuetta sanotaan järjestelmän osoiteavaruudeksi (System Address Space). Realtilassa (real mode) osoiteavaruus on 1 megatavua (1024 kilotavua). Suojatussa tilassa (protected mode) osoiteavaruus on 16 megatavua 80286-suorittimeen ja 4 gigatavua 80386-suorittimeen perustuvissa mikroissa.

Conventional-muisti (Conventional Memory) on se osa järjestelmän osoiteavaruutta, joka on MS-DOS-käyttöjärjestelmän käytettävissä. Useimmissa ns. yhteensopivissa mikroissa tämä muistialue rajoittuu 640 kilotavuun. Rajoitus on 8088-suorittimen perintöä, vaikka kyseiseen muistialueeseen usein viitataan nimenomaan DOS-alueena.

Systemimuistialue (System Memory) on conventional-muistialueen (640 kilotavua) ylittävä osa järjestelmän 1 megatavun osoiteavaruudesta eli 640 ja 1024 kilotavun välinen alue. Tältä 384 kilotavun muistialueelta varaavat tilaa muun muassa

Extended-muistin hallintaa käsiteltiin Softarin numerossa 3/88 olleessa artikkelissa, jonka otsikona oli OS/2:n OHJELMALIITOS JA MUISTINHALLINTA. Siitä saa lisää tietoa extended-muistista.



BIOS, VGA, EGA, SCSI-masamuistinoihin, expanded-tyyppiset lisämuistikortit, HP ScanJet -kuvanlukija ja Token-Net I -verkkoliitännäkortti.

Aikoinaan varmasti riittäväksi katsottu systeemimuistialue on nykyisin alkanut käydä ahtaaksi, kun MS-DOS-ympäristössä käytettävät sovellukset, oheislaitteet ja lisäkortit ovat lisääntyneet ja monipuolistuneet. Tällä muistialueella ei saa esiintyä päällekkäisiä aluevarauksia, koska se johtaisi kyseistä aluetta varanneiden laitteiden ja korttien virhetilanteisiin.

Extended-tyyppinen lisämuisti

Extended-muisti (Extended memory) käsittää järjestelmän osoitevaruudesta 1 megatavua yltävän muistialueen. 80286- ja 80386-suorittimet voivat osoittaa tätä muistialuetta suojatussa tilassa (protected mode).

Extended-muistiin ei voi osoittaa real-tilassa, jossa esimerkiksi MS-DOS toimii. Mahdollisuus käyttää extended-muistialuetta RAM-levynä MS-DOS-ympäristössä perustuu siihen, että RAM-levyn liittyvä laiteohjelma hyödyntää tilapäistä toimintaa suojatussa tilassa. Näin voidaan poikkeuksellisesti siirtää tietoa conventional- ja extended-muistialueen välillä.

Esimerkiksi OS/2, Oracle, Ada ja XENIX toimivat suojatussa tilassa, jolloin on käytettävissä laajempi osoitevaruus kuin real-tilassa. OS/2-käyttöjärjestelmässä voidaan myös käyttää ns. DOS-yhteensopivaa tilaa (real box tai compatibility box), joka sijoittuu muistiavaruudessa osoitealueelle 0 - 1 megatavua, kuten MS-DOS-ympäristö real-tilassa.

Extended-muistin määrän katsominen

MikroMikkoon asennetun extended-muistin määrän voi tarkastaa koneen käynnistyksen yhteydessä. Käynnistystestin yhteydessä tulostuu näyttöön muistin määrä kilotavuina. Muistin kokonaismäärästä tulostuu 640 kilotavua nimikkeellä Base memory, joka osoittaa conventional-muistin määrän. Sama määrä tulostuu näyttöön SETUP-ohjelman käynnistyttyä. Käynnistystestin yhteydessä tulostuu näyttöön myös koneessa olevan extended-muistin määrä nimikkeellä Extra memory (MikroMikko 3/386, MikroMikko 3TT/386) tai Expansion (MikroMikko 3, MikroMikko 3TT). SETUP-ohjelman käynnistyttyä tulostuu sama määrä näyttöön nimikkeellä Expansion memory size (MikroMikko 3/386, MikroMikko 3TT/386) tai Extension memory (MikroMikko 3, MikroMikko 3TT).

Jos extended-muistia on määriteltä RAM-levyksi, RAM-levyn määrittelytiedot tulostuvat näyttöön käynnistyksen yhteydessä. Muun muassa RAM-levyn looginen kirjaintunnus ja koko kilotavuina tulostuvat näyttöön.

Extended-muistialue RAM-levynä

MikroMikko 3 ja MikroMikko 3TT -laitteistojen kokoonpanoissa ilmoitettava työmuistin määrä (RAM) ei sisällä systeemimuistialueen kokoa. Näin esimerkiksi MikroMikko 3, joka sisältää 1 megatavua työmuistia, tarjoaa 640 kilotavun conventional-muistialueen, käyttäjälle "näkyvä" systeemimuistialueen, ja 384 kilotavun extended-

muistialueen. Tässä kokoonpanossa voidaan 384 kilotavun extended-muistialuetta käyttää RAM-levynä sisällyttämällä vastaavan laiteohjelman määrittely CONFIG.SYS-tiedostoon.

Laiteohjelman nimi on RAMDRIVE.SYS Nokian toimittamassa MS-DOS versiossa 3.20 ja VDISK.SYS MS-DOS versiossa 3.30. Laiteohjelman määrittelyssä on oltava valitsin /E, jotta RAM-levy sijoittuisi extended-muistialueelle. Muussa tapauksessa RAM-levy sijoittuu conventional-muistialueelle ja vähentää MS-DOS-sovellusten käytettävissä olevaa muistialuetta. RAM-levyn koko voidaan määritellä suuremmaksi kuin 384 kilotavua, jos kokoonpano sisältää enemmän extended-muistia.

RAM-levy saa automaattisesti vastaavan loogisen kirjaintunnuksen kuin kiintolevyn ositukset ja levykeasemat. Kyseistä tunnusta käytetään RAM-levyn osoittamiseksi. RAM-levyn koko ja tunnus tulostuvat näyttöön käyttöjärjestelmän latauksen yhteydessä.

Kopioimalla ennen käyttöä ohjelmia ja tiedostoja levykkeeltä tai kiintolevyltä RAM-levylle nopeutetaan näiden tietojen käsittelyä, koska muistin käsittely on nopeampaa kuin esimerkiksi levykkeiden käsittely. On kuitenkin muistettava, että RAM-levyllä olevat tiedot katoavat, kun virta katkaistaan koneesta, joten ainakin muuttuneet tiedot on tallennettava takaisin levykkeelle tai kiintolevylle viimeistään ennen virran katkaisemista.

Useiden tiedostojen tai ohjelmien kopiointia voi helpottaa tallentamalla tarvittavat kopiointikomennot komentojonoon, josta tarvittaessa käynnistetään.

Extended-muistin kasvattaminen

Peruskokoonpanoon sisältyvän extended-muistin määrää voi kasvattaa asentamalla extended-tyyppisiä lisämuistikortteja vapaiden korttipaikkojen ja hyödynnettävissä olevan osoiteavaruuden sallimissa rajoissa. Esimerkiksi AST:n RamVantage muistikortti tarjoaa extended-muistia. Nokian tuotevalikoima sisältää 1, 2 ja 3 megatavun extended-muistikortit.

Vaihtoehtoisesti voi lisätä peruskokoonpanon extended-muistin määrää konfiguroimalla expanded-lisämuistikortilla olevan muistin osaksi tai kokonaan extended-muistikiksi. Tämä konfigurointimahdollisuus on olemassa useilla expanded-muistikorteina myytävillä muistikorteilla.

Nokian tuotevalikoimassa oleva expanded-muistikortti, Intel Above PS/286, sallii myös muistityypin valinnan kortin mukana toimitettavan asennusohjelmiston avulla. Tällainen konfiguroitavissa oleva muistikortti toimii joko extended- tai expanded-muistina tai jopa näiden yhdistelmänä ja on huomattavasti monikäyttöisempi kuin puhtaasti extended-tyyppinen muistikortti.

Useimmilla expanded-muistikorteilla suoritetaan muistin konfigurointi DIP-kytkinten ja/tai hyppyvalitsimien avulla, jolloin muistikortti on poistettava koneesta muistiominaisuuksia vaihdettaessa. Nokian toimittamalla expanded-muistikortilla on muistityypin vaihtaminen erityisen helppoa, koska kortin muisti on täysin ohjelmallisesti konfiguroitavissa.

Windows/386 osaa hyödyntää MikroMikko 3/386 ja MikroMikko 3TT/386 -laitteistojen extended-muistia expanded-muistina.

Extended-muistin määrää kasvatettaessa lisäkortein on muistettava huomioida peruskokoonpanoon jo mahdollisesti sisältyvän extended-muistin määrä lisättävän muistin alkuosoitetta määriteltäessä. Tämä merkitsee muun muassa, että on kytkettävä pois peruskokoonpanon extended-muistimäärä (384 kilotavua), jos 1 megatavun työmuistilla varustettuun MikroMikko 3 -laitteistoon asennetaan muistikortti, jonka extended-muisti määritellään alkavaksi osoitteesta 1 megatavua.

Extended-muistia varten on markkinoilla olemassa erillisiä emulointiohjelmistoja, joiden avulla extended-muistia voidaan käyttää expanded-muistina, mutta niiden käyttöä ei voi suositella heikon suorituskyvyn takia. ■

Esko Hannula

CTB-testaustyökalu ANSI C:lle

Ohjelmistotyössä testaus on oleellinen työvaihe. CTB on testaustyökalu, jolla tuotetaan monipuolisia C-kielisten ohjelmien moduulitestausympäristöjä.

Kuva 1. Kuvassa on esitetty miten suunnittelu- ja testausvaiheet vastasivat toisiaan. Testaus on monivaiheinen tapahtuma, joka on aktiivinen ohjelmointivaiheesta lähtien siihen asti kunnes asiakas on hyväksynyt tuotteen.

Ohjelmistotyön päämäärä on tuottaa tehokkaasti laadukkaita ohjelmistoja. Laadukkaalla ohjelmistolla tarkoitetaan ohjelmistoa, joka tyydyttää käyttäjän vaatimukset ja valmistuu suunnitellun aikataulun sekä budjetin mukaisesti. Ennen kaikkea ohjelmisto toimii virheettömästi ja on muokattavissa käyttäjän vaatimuksien muuttuessa.

Ennen ohjelmointia on analysoitava käyttäjän tarpeet, määriteltävä tarpeet tyydyttävä tuote ja suunniteltava ohjelmiston sekä sen moduulien toteutus. Näiden tehtävien rinnalla kulkee laadunvarmistus, jonka avulla pyritään havaitsemaan virheet mahdollisimman ajoissa.

Testaus ohjelmistotyön osana

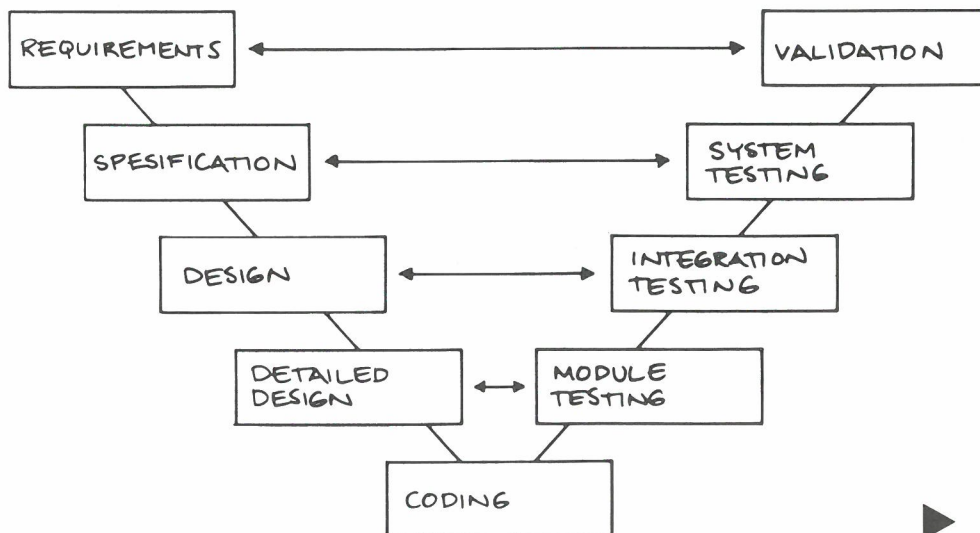
Ihminen on kuitenkin erehtyväinen: kokemus osoittaa, että viimeistään ohjelmointivaiheessa tehdään virheitä, joita laadunvarmistuksessa ei pystytty ha-

vaitsemaan. Tämän vuoksi tarvitaan testausta.

Testaus on työtä, jossa ohjelmistosta pyritään löytämään virheet, jotta voitaisiin – kun virheiden löytymistiheys pienenee riittävästi – saavuttaa riittävä luottamus ohjelmiston virheettömyyteen. Testaus on siis eri asia kuin debuggaus, jonka avulla pyritään löytämään testauksen havaitseman virheen aiheuttaja.

Alin testaus taso on moduulitestaus, jossa testataan, toteuttavatko yksittäiset moduulit määrittelynsä. Seuraava taso on integrointitestaus, jossa testataan moduulien yhteistoimintaa. Järjestelmätestauksessa testataan toteuttaako ohjelmisto toiminnallisen määrittelynsä. Ylin testaus taso on hyväksymistestaus, jossa testataan täyttääkö ohjelmisto asiakkaan sille asettamat vaatimukset.

Mitä alemmalla testaus tasolla virhe havaitaan, sitä edullisempaa sen korjaaminen on. Virheen korjauskustannukset moninkertaistuvat siirryttäessä testaus tasolta toiselle.



CTB – työkalu moduulitestaukseen

Koska kattava moduulitestaus pienentää merkittävästi ohjelmiston tuotantokustannuksia, moduulitestaukseen ja sen tehostamiseen kannattaa panostaa.

CTB (C Test Bed System) on CASE-työkalu, joka mahdollistaa C-kielisten ohjelmien moduulitestausympäristöjen automaattisen ja nopean tuottamisen. CTB:n tuottamassa ympäristössä testattavaa koodia voidaan tehokkaasti "pommittaa" testitapauksilla ja nähdä, onko koodin vaste odotettu. CTB soveltuu siis black box -tyyppiseen testaukseen, jossa testattava moduuli nähdään mustana laatikkona, jonka liittymä on tunnettu.

CTB-järjestelmässä CTB-kehitin analysoi C-kielistä lähdekoodia ja tuottaa koodista löytämilleen funktioille testiajurin (test driver). Varsinainen testaus taas tehdään CTB-testipedissä, joka muodostetaan kääntämällä ja linkittämällä generoitu testiajuri, testattava koodi ja CTB-kirjasto. CTB-kirjastossa ovat testipedin osat, jotka ovat samat testattavasta ohjelmasta riippumatta. CTB-kehittimen toimintaa esittää kuva 2.

CTB-testipedin ajaminen

Kun CTB-kehittimen tuottama koodi on käännetty, on saatu aikaan testipeti. Se on täysin itse-

näinen ohjelma, jolla varsinainen testaus voidaan suorittaa. CTB-testipedin päätaso on komentotulkki.

CTB-testipedin pää tarkoitus on testattavien toimintojen kutsuminen. Se onnistuu C:n syntaksin mukaisella komennolla kuten `BED> initialize()`.

Black box -tyyppisessä testauksessa mielenkiintoisimpia ovat kuitenkin funktiot, joilla on paluuarvo ja parametreja. Tähän tarkoitukseen käyttäjä voi testipetiä ajaessaan varata testipetimuuttujia, jotka käyttäytyvät kuten tavalliset C-ohjelman muuttujat:

```
BED> char c, *d, e[20];
BED> c = 'a'
BED> e[5] = 'X'
BED> d = strcpy(e, "C T B")
```

Yllä olevat komennot varaavat ensin kolme muuttujaa: c on character-muuttuja, d on osoitin character-tyyppiseen muuttujaan ja e on 20-komponenttinen character-taulukko. Muuttujille voidaan sijoittaa arvoja ja niihin voidaan tallettaa funktioiden paluuarvoja sekä välittää niitä parametreina testattaville aliohjelmeille. Kuten esimerkeistä näkyy, CTB:n komentokieli on hyvin C-kielen kaltaista.

Testipetimuuttujien arvoja voi tutkia: `BED> ? e "C T B"`. Samoin käyttäjä voi '?'-komennolla tutkia tyyppien määrittelyjä ja funktioiden kutsumalleja (ANSI-C terminologiassa prototyyppejä). Testipedin komentoista voi muodostaa komentotiedoston (script), jonka komen-

not suoritetaan 'do'-komennolla: `BED> do CASES.TXT`. Komento suorittaa tiedostossa CASES.TXT olevat testipetikomennot.

On myös mahdollista "äänittää" testausajo tiedostoon, jolloin testausistunnosta saadaan dokumentti. Testauskertojen tuloksia voidaan tutkia vertaamalla äänitettyjä testausajoja.

Testipedin kaikki komennot sisältävä online-help on saatavilla koko testipetiajon ajan.

Monipuolisesti hyödyllinen

Menestyksekkään testauksen takana on luova mutta järjestelmällinen testaaja. CTB:n avulla hän voi paremmin keskittyä varsinaiseen työhönsä ja antaa työkalun huolehtia tehokkaasti testausympäristön tuottamisesta.

Testityökaluilla saadaan testaukseen käytettävä aika jakautumaan järkevämmiin. Ilman testipetikehitintä testaaja joutuu käsin koostamaan testipedin, mihin kuluu huomattava osa testaukseen varatusta ajasta. CTB-kehitin tuottaa helpokäyttöisen ja monipuolisen testipedin muutamassa minuutissa, jolloin testaajalle jää merkittävästi enemmän aikaa testitapausten suunnitteluun ja analysointiin.

CTB-testipetiä voi käyttää sekä interaktiivisesti että eräajoluonteisesti. Testaaja voi tallentaa testausistunnon tulokset tiedostoon, joka mahdollistaa kahden testiajon vertailun. Myös ad hoc -luonteisen interaktiivisen testiajon aikana annetut komennot voi tallentaa tiedostoon ja suorittaa myöhemmin uudelleen.

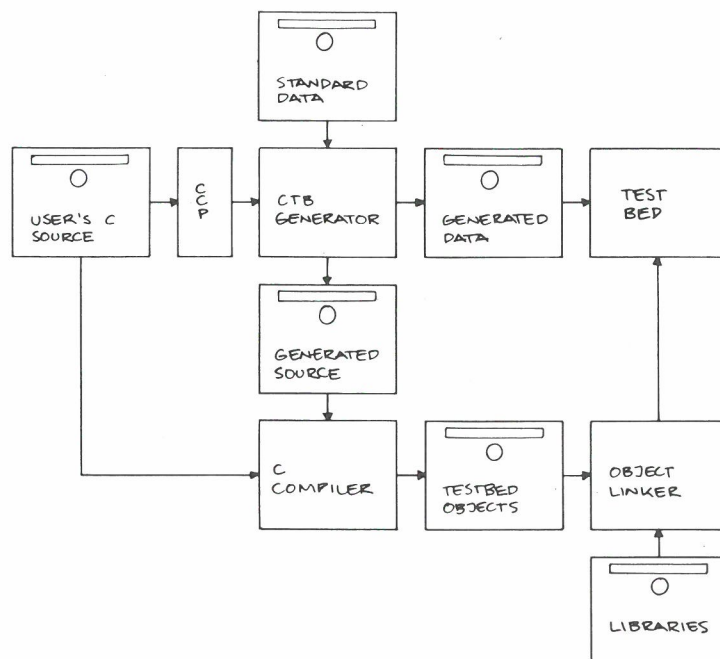
CTB:llä generoiduilla testipeideillä on yhdenmukainen käyttöliittymä, komentokieli ja tulostus. Kaikki testitapaukset voidaan kuvata CTB:n komento-kielillä. Yhdenmukaiset testiajotestaukset helpottavat testitulosten analysointia.

Onnistuneen testauksen kannalta on tärkeää, että testausympäristö on helpokäyttöinen ja testaustapahtuma miellyttävämpi. Näin testaajalla on enemmän aikaa, halua ja mahdollisuuksia useampien virheiden etsimiseen.

Nokian kehittämä tuote

CTB on tehty Nokia Datan Tampereen tuotekehitysyksikön testaustyökaluryhmässä, joka on kehittänyt myös TBGEN ja TCMON Ada-työkalut. ■

Kuva 2
CTB-kehitin analysoi testattavan C-koodin ja tuottaa testiajurin, joka käännetään ja linkitetään testattavan koodin ja CTB-kirjaston kanssa. Näin saatu testipeti mahdollistaa automaattisen, suunnitelmallisen, dokumentoidun, toistettavan ja yhdenmukaisen testauksen.



Saarisen Tapsan muistelmat osa III

”Riisisateessa”



Viimeksi jätimme Tapsan ja Leenan loikoilemaan veneen pohjalle aavalle järvenselälle. Niin siinä sitten kävi, että vannoutuneeksi poikamieheksi jo leimattu, näyttöpäätteensä kylmään rakkauteen tottunut huippuasiantuntija Tapsa Saarinen antautui. Kuin huomaamatta hän alkoi töiden jälkeen kulkea kyläravintola Buffalon ohi kohti Leenan pientä kaksiota. Päivän pituuskin alkoi Tiukka-Tasasen harmiksi lyhentyä työehtosopimuksen mukaisiin normeihin. Kun Leena eräänä aamuna sitten suorasukaisesti ehdotti pysyvämpää ratkaisua, murahti Tapsa myöntymyksensä kummemmin empimättä.

Häitä edeltävänä perjantai-aamuna Saarisen saapuessa työpaikalleen hänet otti vastaan myyntimies Mäkinen, joka käännutti hänet heti ovelta takaisin rappukäytävään.

– Mitä nyt?

– Ei mitään pulinoita, taksi ja Tasanen odottavat!

– Mistä on kysymys? Minulla on Rauta-Yhtiön driveri kesken, se pitäisi saada valmiiksi tänään.

– Nyt onkin vapaapäivä, ja Tasasen luvalla, me pidetään polttarit!

– Mitä Tiukka mukana tekee?

– Toimii maksajana. Lisäksi hän haluaa nähdä ahkerimman työntekijänsä nöyryytyksen ennen kuin tämä kokonaan siirtyy 8 tunnin työpäiviin ja lapsenhoidon.

– Minä en lähde. Konekin on päällä.

– Nyt jätkä menttiin. Mäkinen raahasi hangoittelevan Tapsan väkisin autoon. Taksi pysähtyi Viraston konekeskuksen eteen, mistä sankari juoksutettiin suoraan konesaliin, joka oli täynnä vanhoja työkalureita, suurin osa oudosti vanhentuneena ja liho-neena. Tapsa yritti tarkastella salaa omaa sivukuvaansa Ison Printterin lasioven peilistä.

Juhlapuhe pidettiin Viraston konesalin keskellä sijaitsevan virallisen museolaitteen, reikäkorttikoneen äärellä. Sen uume-niin oli Tapsakin ensimmäisen ohjelmansa työntänyt miltei parikymmentä vuotta sitten. Viri-

tys oli toiminutkin kahden kuu-kauden kivuliaan testivaiheen jälkeen. Suoraan koulunpenkiltä ohjelmointiin antautuneen nuorukaisen synnytystuskat palautettiin myös mieliin Viraston atk-johtajan, Reino Riekin, juhlapuheessa. Rieki kuvaili Saarisen kankeata juoneenpääsyä ja uran rakettimaista nousua Tapsan löydettyä konekielen omimmaksi työkalukseksi. Tietokoneen sielunelämän tuntijana hän oli kuulemma vertaansa vailla, ja puheen lopuksi luovutti Rieki Tapsalle tämän ensimmäisen ohjelman vaaleanpunaiselle juhlatkolomakkeelle tulostettuna. Listaa vilkaistessaan Saarinen nopeasti totesi koodin ala-arvoiseksi sotkuksi ja päätti visusti piilottaa sen.

Aplodien jälkeen riitti selkääntaputtajia.

– Muistatkos, kun valvottiin töissä neljä vuorokautta yhteen menoon?

– Se oli hyvä juttu, kun illalla kaikki koneet kaatuivat ja sinut haettiin umpihumalassa Buffalosta hommiin, eikä mennyt kuin vartti ja systeemit toimivat kuin Buicki!

– Muistatkos, kun sanoit Rieki-le suorat sanat?

– Se oli paras tekemäsi ohjelma, joka aina perjantaisin tasan neljältä nosti automaattisesti kaikkien viraston kirjoittimien kannet ylös. Se oli kuin kuolleitten ylösnousemus viileässä kone-salissa.

– Kai muistat, kun

Tapsa ei ehtinyt jokaiselle iloiselle kasvolle edes vastata, mutta nähdessään vuoropäällikkö Komosen Tapsa veti tämän syrjään.

– Miksi puheen piti Riekk, miksei Jokinen, joka oli sentään minun pomoni viisi vuotta?

– Jokinen vain luisui pois, vastasi Komonen vaisusti.

– Luisui pois, mitä tarkoitat?

– Jokinen vain katosi, siinä se.

Saarinen olisi ehdottomasti halunnut kuulla lisää, mutta myyntitykki Mäkinen raahasi häntä jo toisaalle. He sulkeutuivat paperiviraston takana olevaan koppiin, mihin Saarinen perinteisesti krapula-aamuna päättäneen eristäytyi kuntoaan kohentelemaan.

Mäkinen löi Tapsan käteen koskenkorvapullon.

– Tämä on juotava, ennenkuin pääset täältä pois! Ne on polttarit nyt!

Sulhaskandidaatti ryhtyi toimeen, ja vasta puolen pullon vaiheilla hän huomasi kopperon nurkassa pöydän ääressä hiljaiseen kyniä lajittelevan miehen, joka näytti jotenkin tutulta.

– Kokkonen, jumalauta, vanha bittivirtuoosi, mitä sinä täällä teet?

– Toimin varastonhoitajana, vastasi hiljainen mies, mietti hetken ja kaivoi vanhalle kollegalleen Viraston uusimman edustuskynän muistoksi. Minut on siirretty toisiin tehtäviin, ei koodi enää kulkenut.

– Sinähän olit talon paras Cobolisti.

– Niin olin. Kokkonen palasi paikalleen ja jatkoi kynien lajittelua entiseen tapaan.

– Mitä täällä oikein on tapahtunut?

Mäkinen ryhtyi hoputtamaan Tapsaa eteenpäin ja aikataulua kiristääkseen kippasi loput vaajaasta pullost. Ryyppyjen välillä hän selitti seuraavaa tehtävää.

– Tuon oven takana odottaa menetetyt onnesi kavalkadi. Nuorena miehenä olit tietämättäsi Viraston tyttöjen suuri saavuttamaton sankari, jonka jokainen olisi halunnut, mutta herra asiantuntija vain koplasi illat koneitaan. Huhun mukaan joku olisi saanut kanssasi treffit sovitua, mutta et ollut saapunut paikalle, koneet olivat olleet taas jossakin nurin. Oven takana odottaa kujanjuoksu, he ovat kaikki rivissä käytävällä. Et pääse ulos ennenkuin olet suudellut joka ainoaa!

Siellä he todella olivat. Jono

oli tutuntuntuista väkeä, mutta hieman Tapsa vierasti sitä, ettei kellään ollut enää päällä niitä kiireitä farmareita, joista operaattorit antoivat innolla arvostelupisteitä. Nämähän näyttivät aivan perheenäideiltä. Tapsa muiskautteli halusta pitkän jonoa päästä päähän. Paljoakaan ei puhuttu, muutamiin ”Muistatko minut?” tai ”Minulla on jo neljä lasta”, Tapsa ei osannut kuin hymillä idioottimaisesti. Letkan päässä odottivat Tasanen ja Mäkinen, jotka kiikkivat hieman juopuneen yljän vauhdilla taksiin. Takapenkillä järkytystään voivotteleva Saarinen sai vihdoin sanottua.

– Olenko minä todella nuo kaikki menettänyt?

– Ne kaikki, vastasi Tiukka-Tasanen päästäen sydämestään tulvivan naurun. Tapsa jatkoi seuraavaan kohteeseen.

Auto pysähtyi Sellu-tehtaan alueella vanhan pato-altaan reunalle. Tehdas ei ollut pyörinyt moneen vuoteen, mutta lieteallas oli entisessä paikassaan, mihin Sellun atk-osaston ikkunasta oli avautunut silmiä hivelevä näkymä, laguuni pelkkää moskaa.

– Mitäs me täällä tehdään, tehdashan on kuollut jo ajat sitten, ihmetteli takaikkunasta kurkki-va Tapsa.

– Katso altaan toiselle reunalle, vastasi Mäkinen. Siellä seisoi komeassa rivistössä koko Sellu-Tehtaan vanha atk-osasto, pitkät mustat popliinitakit päällään.

– Mitä he tuolla vastarannalla toljottavat, ja tuollaiset itsensä paljastajan takit päällään?

– Koska heillä on vain pelkät uima-asut siellä alla, vastasi Mäkinen. Etkö huomaa, liete ei enää haise, vesi on puhdasta! Öitä koneen äärellä valvoessasi sellun hajun tunkiessa ytimiin asti olit toivonut näkeväsi päivän, jolloin patoaltaalla pelataan vesipalloa. Se päivä on tänään!

Popliinitakit lensivät soraan ja Sellun atk uida molksi toiselle rannalle. Sitten olikin Tapsan vuoro, mutta vaatteet päällä. Vesipallo-ottelu alkoi välittömästi ja päättyi käyttöpuolen voittoon. Tapsa nautti olostaan kuin hylje aavemaisen hylätyllä tehdasalueella. Linnutkin lauloivat tavallista kuulaammin.

Hääpäivän aamuna Tapsa heräsi työhuoneensa pöydän alta kuten vanhoina hyvinä aikoina. Hänet oli sinne kannettu samuneena kyläravintola Buffalosta, jossa olivat paikalla olleet kaikki ja meno senmukaista.

Wanhassa kirkossa vapiseva sulhanen selvisi seremonioista upean morsiamensa rinnalla ylättävän tyylikkäästi. Penkit olivat täynnä ja sivulliset olivat huomanneet jopa Tiukka-Tasanen pirauttaneen muutaman kyyneleen, tosin epäiltiin hänen edelleen laskeneen Tapsan muodonmuutoksessa menettämiään työtunteja. Best man, myyntitykki Mäkinen, junaili tapahtumat hienosti, lausui alttarilta hääparille itse kirjoittamansa loppusoinnutoman muistorunon, jonka aikana herkkä tunnelma salissa oli käsin kosketeltava. Mäkiseen kasvatusten seissyt pari pystyi vain vaivoin pidettelemään naurua, koska myyntitykin kravatti oli edellisillan tapahtuminen jäljiltä sinapissa.

Kirkon ulkopuolella olivat työtoverit pystyttäneet kaapeleista ja magneettinauhosta kunniakujan, jota pitkin kulkevan nuoren parin päälle heiteltiin lutikoita ja disketin palasia. Käytävän päässä odotti Mäkisen vanha Datsun ovi auki. Takapuskuriin oli narun päähän ripustettu vanha pieniruutuinen näyttöpäätte. Tien autolle kesken juhlahumun sulki kuitenkin paikalle vauhdilla porhaltanut Rauta-Yhtiön musta auto, mistä hengästynyt atk-päällikkö Ketola juoksi sulhasen kainaloon kiinni.

– Jumalauta, Tapsa, meillä on kone jumissa ja tuotanto seisoo. Olet ainoa, joka pystyy auttamaan, nyt on kiire!

– Tapsa, onko sinun pakko, kiljaisi hätäantynyt morsian.

Mäkinen säntäili sinne tänne disketin palasia olkapäillään.

– Nyt on hätätilanne, Mäkinen huolehtii sinusta, tulen niin pian kuin pääsen, vastasi Tapsa ja juoksi mirriään löysäten Ketolan perässä Rauta-Yhtiön Volvon takapenkille.

Morsian ja Mäkinen lähtivät köröttelemään vuosien tauon jälkeen pestyllä Datsunilla kohti hääjuhlapaikkaa. Naruun kytetty päätte pirstoutui jo torin kivetukseen.

Kello kolme yöllä Saarinen hiipi hääyön viettopaikaksi valitun Grand Hotelin swiittiin. Huone oli tyhjä, Tapsa otti pöydällä olevasta puoliksi juodusta sampanjapullosta pitkät siivut ja siirtyi istumaan koskemattomalle häävuoteelle. Kun hän sai lasinsa tyhjäksi, avautui vaatekomeron ovi, ja sieltä astui esiin Leena täydessä morsiusvarustuksessa.

– Jaksatko pätsätä vielä minutkin? ■

MikroMikko 3 TopTen

MikroMikko 3 TopTen perustuu vuoden 1988 kymmenen ensimmäisen kuukauden myyntiluvuista laskettuihin suhteellisiin indekseihin.

Listalla ei ole tällä kertaa yhtään uutta tulokasta, sillä kymmenen kysytyintä ohjelmistoa ovat saavuttaneet todella vankan aseman. Järjestys on toki hiukan muuttunut niin, että tiedonsiirto-ohjelmistot ovat pudonneet listalla alaspäin ja taulukkolaskenta-ohjelmistot kivunneet ylöspäin. Samoin viime kerralla listalle noussut Designer- grafiikkaohjelmisto on saavuttanut entistä suoremman suosion.

MikroMikko 3 TopTen

1. (1) Windows	242
2. (2) WordPerfect	87
3. (4) MultiPlan	53
4. (6) Excel	41
5. (3) WS 220	
-emulaattori	38
6. (5) WS 7300	
-emulaattori	29
7. (8) Lotus 1-2-3	25
8. (7) WS 7800	
-emulaattori	23
9. (9) Symphony	20
10.(10) Designer	15

Suluissa on sijoitus edellisessä Softarissa 3/1988.

Mikroja, ohjelmistoja ja tarvikkeita voit tilata tai noutaa Nokia Datan mikromyymälästä Helsingistä, Keskuskatu 7, p. (90) 5671 tai Tampereelta, Yliopistonkatu 56, p. (931) 509 111.

Yhteistyötä Inca Systemsin kanssa

Nokia Data ja oululainen Inca Systems allekirjoittivat yhteistyösopimuksen Case Technologyn elektroniikkasuunnitteluohjelmistojen myynnistä. Sopimuksen mukaan Nokia markkinoi ohjelmistoja Sun-työasemin ja MikroMikko 3 ja 3TT-mikroihin.

Kyseinen Vanguard Stellar-ohjelmisto sisältää kaikki elektroniikkatuotteen määrittelyssä, suunnittelussa ja toteutuksessa tarvittavat työkalut.

Lisätietoja:

Mikko Helander, p. (90) 1241.

Daisy Systemsin ohjelmistoja

Nokia Data ryhtyy markkinoimaan Daisy Systemsin elektroniikkasuunnitteluohjelmistoja Sun-työasemiin.

Ohjelmistoja käytetään elektroniikan järjestelmien suunnittelun kaikissa vaiheissa piirikaaviosuunnittelusta suunnittelun varmennukseen ja piirikuvioiduun suunnitteluun. Ohjelmistoja löytyy myös integroitujen piirien suunnittelun eri osa-alueille ja piirikorttisuunnitteluun.

Lisätietoja:

Mikko Helander, p. (90) 1241.

Calma-ohjelmistoja edelleen Nokialta

Prime Computer osti GE:n Calma-liiketoiminnan. Yritykset sopivat myös CAD/CAM-ohjelmistojen kehityksestä ja hankinnasta. Nokian Calma-myynti jatkuu kuitenkin edelleen ja asiakassitoumukset pidetään.

Nokia Data on CAD/CAM-järjestelmien markkinajohtaja. Se on toimittanut Suomeen noin 150 Calma-pohjaista järjestelmää.

Lisätietoja:

Mikko Helander, p. (90) 1241.

Autaa maailmaa

Brainware Oy, Top Team ja SPR ovat yhteistyössä valmistaneet Autaa maailmaa -tietokonepelin kouluhallituksen tuella. Peliä pelaamalla eläydytään SPR:n avustustyöntekijän rooliin ja suoritetaan erilaisia avustustehtäviä kuvitteellisessa kehitysmaassa, esimerkiksi kuljetetaan ruokaa ja lääkkeitä avustuskohdeisiin, pakolaisleiriin tai kenttäsairaalaan. Myös muun muassa rahoitusjärjestelyt on suoritettava tehtävien muodossa.

Peli toimii teollisuusstandardin mukaisissa mikroissa, jossa on MS-DOS versio 2.11 tai uudempi.

Lisätietoja: SPR, Jouko Vähäsöyrinki, Sirpa Salovius tai Jukka Mäkelä, p. (90) 12931.

NASista AS/EX-mallisarja

Uusi NAS-suurkonesarja AS/EX muodostuu kaikkiaan 15 mallista ja siinä yhdistyvät entisten AS/XL-sarjan ja keskisuuren AS/VL-sarjan mallit. Näin muodostuu yhtenäisen kasvupolku pienemmästä mallista suurempaan, joka on 20-kertaa pienintä mallia tehokkaampi.

Mallin 50 ja sitä suurempien mallien suorituskykyä on lisätty monilla teknisillä parannuksilla. Näitä ovat lyhemmät kierrosajat, suurempi järjestelmänlaajuinen välimuisti, uudet logiikka-komponentit ja parempi moniprosessoinnin hyötysuhde. Esimerkiksi AS/EX 60 on keskimääräisesti 1,15-1,25 kertaa tehokkaampi kuin yksiprosessorinen edeltäjänsä. Paljon laskentaa vaativissa tehtävissä AS/EX 60 on jopa 28 prosenttia edeltäjänsä tehokkaampi.

AS/EX-sarjaan on tullut myös kaksi uutta keskikokoista mallia. Pienin malli AS/EX 10 on 0,75-kertaa suorituskykyisempi kuin aiempi AS/VL 40 (nyt AS/EX 20) ja kaksiprosessorinen AS/EX 35 on 1,7-1,9-kertaa suorituskykyisempi kuin AS/EX 25.

AS/EX-sarjan mallien toimitukset on jo aloitettu paitsi mallit 10 ja 35 ovat saatavissa vuoden 1989 ensimmäisen neljänneksen aikana.

Lisätietoja:

Matti Saavalainen, p. (90) 1241.

Interleafilla julkaisut

Nokia on solminut Interleafin kanssa sopimuksen, jonka mukaan se markkinoi Suomessa, Norjassa ja Tanskassa Interleaf-julkaisuohjelmistoja. Sopimus koskee sekä Sun- että Nokian aiemmin edustamia Apollo-työasemia.

Interleaf-ohjelmistot on suunniteltu erityisesti tekniseen dokumentointiin ja muiden vaativien julkaisujen tekoon. Ne soveltuvat erityisen hyvin nopeasti uudistuvan aineiston kuten asiakasdokumenttien, tarjouksien, luetteloiden, raporttien, koulutus- ja markkinointiaineiston tekoon.

Interleaf-ohjelmistot kehitetään Sun-työasemissa, joiden haudatun arkkitehtuurin ominaisuuksia ohjelmisto hyödyntää tehokkaasti.

Kaikkiaan lähes 50 Nokia Datan asiakasta käyttää Interleaf-ohjelmistoja. Nokia arvioi markkinaosuudekseen UNIX-työasemapohjaisessa dokumentoinnissa yli 50 prosenttia ja uskoo markkinoiden kasvavan lähivuosina 30 prosenttia vuosittain.

Lisätietoja:

Juha Salminen, p. (90) 5671.

Interleaf -käyttäjakerho perustettu

Interleaf-julkaisujärjestelmän käyttäjät ovat perustaneet Interleaf Finland Users Groupin (IFUG). Kerho on ensimmäinen USA:n ulkopuolella ja sen tarkoituksena on jakaa käyttökokemuksia, järjestää yhteisiä tilaisuuksia sekä hankkia ja jakaa tietoa tuotteista, jotka soveltuvat käytettäväksi Interleaf-järjestelmässä.

Kerhon puheenjohtaja on Markku Tuutti, Oy Tampella Ab Tamrock, p. (931) 413 411.

MultiConnect yhdistää lähiverkot

MultiConnect on modulaarinen moniporttitoistin, joka tarjoaa joustavan mahdollisuuden yhdistää erityyppisiä Ethernet-verkkoja keskitetysti toisiinsa.

Perusyksikön avulla voidaan yhdistää 15 Ethernet-segmenttiä ja laajennusyksiköiden avulla jopa 45. Segmentit voivat käyttää ohutta tai paksua Ethernet-kaapelointia, valokuitua tai kierrettyä parikaapelia. Mikä tahansa kombinaatio eri segmenteistä on mahdollinen ja järjestelmän modulaarinen rakenne tekee laajennukset ja muutokset helpoiksi.

Lisätietoja:

Simo Poikola, p. (90) 5671.

CS/210-päätepalvelija Bridgeen

CS/210-päätepalvelija on varustettu 3,5 tuuman levykeasemalla, joten se ei välttämättä tarvitse verkonhallintayksikköä. CS/210:ssa on kymmenen RS232-porttia ja sitä on saatavissa vain EtherNet-liitännällä.

XNS- ja TCP/IP-protokollien lisäksi CS/210:een saa myös OSI-protokollan mukaisen ohjelmiston, joka on kuitenkin toistaiseksi ladattava levykeasemasta. Muuten ohjelmistot voidaan ladataan myös NCS/AT:lta.

CS/210 korvaa CS/100-päätepalvelijan.

Lisätietoja:

Simo Poikola, p. (90) 5671.

MikroMikko edelleen markkinajohtaja

IDC on arvioinut vuoden 1988 kokonaismarkkinoiksi 88 000 mikroa. Nokia Datan tavoitteena on toimittaa Suomeen 20 000 mikroa tänä vuonna. Vuonna 1987 Nokia Data toimitti noin 14 000 mikroa.

Alkuvuoden menekki on ollut hyvä ja syyskuun loppuun mennessä myyntitavoitteesta oli saavutettu yli 80 prosenttia. Tämän perusteella Nokia arvioi kasvattavansa mikrojen markkinaosuuttaan viime vuoden toteutuneesta 18 prosentista 23 prosenttiin tänä vuonna.

Lisätietoja:

Anneli Martonen, p. (90) 5671.

Studiotyöskentelyä Kilossa

Mikrojen ja päätteiden kasvavan menekin johdosta Kilon tietokonetehtas organisoitiin uudelleen viime elokuussa. Tehtaalla on käytössä kaikkiaan 8 500 neliötä tuotantotilaa ja 1 500 neliötä toimistotilaa.

Tehdas jaettiin kolmeen yhtä suureen osaan ja varsinaiset tuotetehtaat sijaitsevat keskimmissä osassa. Tuotanto alkaa Materiaalit-yksiköstä, joka ostaa ja varastoi kaikki tuotannossa tarvittavat komponentit. Täältä komponentit menevät tuotetehtaille, joissa tuotteet valmistetaan.

Mikrotuotetehtas valmistaa mikrot, järjestelmätuotetehtas pankin ja kaupan järjestelmiä ja kirjoitintuotetehtas kokoaa kirjoittimia ja kassapäätteitä. Lohjan päätetehtas kuuluu samaan

ketjuun ja se valmistaa päätteitä. Tehtailta tuotteet menevät Logistiikka-yksikköön, joka vastaa asiakastoimituksista.

Samalla vaihtoyöstä on siirrytty studiotyöskentelyyn. Nyt jokainen työntekijä näkee mitä muut tekevät ja mitä koko tehdas saa aikaan. Tämä on lyhentänyt valmistusaikaa kolmesta viikosta kolmeen päivään. Samalla työnteko on vaihtelevampaa ja ergonomisempaa.

Nokia Datalla on tehtaat myös Ruotsissa Ronnebyssä ja Bräkne Hobbyssä. Ronnebyssä valmistetaan mikroja, päätteitä ja S2500-minikoneita. Bräkne Hobbyssä valmistetaan komponentteja ja näppäimistöjä.

Päätetehtaan laajennusosassa aloitettiin tuotanto

Nokian Datan Lohjalla sijaitsevan päätetehtaan laajennusosa otettiin käyttöön 8.11.1988. Tuotantotila kaksinkertaistui ja on nyt 3 000 neliometriä. Tehdas työllistää 50 henkilöä. Kasvu viime vuodesta on 29 henkilöä.

Laajennusosaan siirretään Yritysviestintä-yksikön tuotanto Oulusta. Yksikön tuotevalikoimaan kuuluvat puhelinvaihteet, sarjapuhelinjärjestelmät, maksupuhelimet ja telefaksit. Oulussa vapautuviin tiloihin laajenee

Nokia Datasiiirron modeemituotanto.

Reilun vuoden toiminnassa olleen päätetehtaan tuotantovoimien ovat kasvaneet nopeasti. Viime vuonna valmistettiin 17 000 päätettä ja tänä vuonna jo noin 32 000 päätettä. Ensimmäisenä vuonna on tarkoitus valmistaa noin 53 000 päätettä.

Lisätietoja:

Markku Leinonen, p. (912) 817 76.

MikroMikko 3TT/386:een uusi muistikortti

MikroMikko 3TT/386:t varustetaan uudella muistikortilla noin viikosta 49 lähtien. Uudella muistikortilla on kaksi lisäsarjaliitännäporttia. Nyt MikroMikko 3TT/386:ssa on valmiina kolme sarjaliitännäporttia ja yksi rinnakkaisliitännäportti. Hinnat säilyvät muuttumattomina.

Tämä kuva on jo historiallinen. Vuonna 1985 toimintansa aloittaneella Kilon tietokonetehtaalla on siirrytty studiotyöskentelyyn.



Harkitusti kohti 90-lukua

Vuonna 1984 toimintansa aloittanut Nokia Datan tietotekniikan koulutuskeskus NITEC on Suomen suurin oman alansa koulutuskeskus. Työntekijöitä on kaikkiaan 48, joista mikrokoulutus työllistää 22. Oppilaspäiviä on vuodessa 17 000, joista 8 000 liittyy mikroiin.

NITEC tutki viime kesänä toimintansa ja määritteli Nokia Datan tietotekniikkakoulutuksen 90-luvun suuntalinjat.

NITECin mikrokoulutus keskittyy tulevaisuudessa Tiimi-työryhmän Perfect- ja Windows-tuoteperheiden tuotteisiin. Koulutuksen painopiste siirtyy jatko- ja asiantuntijakoulutukseen, vaikka peruskurssit säilyvät kurssivalikoimassa. Peruskoulutus muuttuu ns. soveltavaksi koulutukseksi, jossa otetaan entistä enemmän huomioon koulutettavan oma työympäristö.

Tukihenkilö- ja asiantuntijakoulutusta laajennetaan sisällöllisesti tuntuvasti. Uutuutena markkinoidaan toimiala- tai sovellusala-kohtaisia kursseja. Ohjelmointi- ja suunnittelukoulutusta laajennetaan työmenetelmä-koulutuksen suuntaan.

Tulevaisuudessa ryhmätyönä tehtävä koulutusaineisto rakennetaan sisällöllisesti itsenäisistä

osista, moduuleista. Näin päästään yksilöllisempään koulutukseen, kun aineistoa voidaan muuttaa ja painottaa kurssikohtaisesti koulutettavien tietotason mukaan. Samoin toimialakohtaisten osien lisääminen on luontevaa.

Uutuutena NITEC tarjoaa koulutuspuitesopimuksia ja sitoutuu pitämään asiakkaan koulutustarpeista huolta järjestämällä kurssin aina tarvittaessa, esimerkiksi kun hankitaan uusia laitteita tai ohjelmistoja. Uusi panostuskohde on myös ohjelmisto- ja järjestelmäkonsultointi.

Lisätietoja:

Kari Kallio ja Hellä Helander-Hakanpää, p. (90) 5671.

NITECistä Microsoft University

Nokia Datan tietotekniikan koulutuskeskus NITEC on saanut tavoitellun Microsoft University auktorisoinnin. Se hoitaa ja kehittää Microsoft Universityn koulutusta Suomessa.

Matriisikirjoittimeksi DL 5600

24-neulaisen DL 5600 -matriisikirjoittimen tulostusnopeus vedoslaadulla on jopa 486 merkkiä sekunnissa. Raporttijälkeä tulostettaessa sen nopeus on 324 ja laatuja jälkeä tulostettaessa 162 merkkiä sekunnissa. Grafiikan tarkkuus on 144x240 pistettä neiliötuumalla.

Vakiona löytyy kolme merkistöä: Courier 10, Prestige Elite ja Boldface. Optiona on saatavissa kaksi fonttikorttia, joissa on kaksi kirjaisintyyppiä per kasetti. Vaihtoehdot ovat Orator, Letter Gothic, Scientific 12 ja Light Italic.

Kirjoittimesta löytyvät seuraavat emuloinnit: IBM Graphics Printer, IBM Proprinter XL, Diablo 630 API, Fujitsu DPL24C. Sarja- ja rinnakkaisliitäntöjen lisäksi on saatavana coax-liitäntä.

Paperinkäsittelyssä voidaan valita kitkaveto tai sisäänrakennettu kaksisuuntainen työntöraktori. Yksittäisarkit voidaan syöttää jatkolomaketta poistamatta. Kopioiden määrä voi alkuperäisen lisäksi olla seitsemän.

Toimintatilat voidaan asettaa etupaneelistä. Siinä on neljä painiketta ja LDC-näyttö.

Lisätietoja:

Pirkko-Riitta Tasanko, p. (90) 5671.

Designer kehittyä

Designeristä on tullut versio 1.3, jossa on sivuttavan muistin tuki (EMS 4.0) ja useita uusia kirjoitinhajaimia. Toiminnot ovat samat kuin versiossa 1.2.

Sivuttavan muistin tuki mahdollistaa monimutkaisemmatkin kuvat mikroissa, joihin on lisätty muistia ja 80386-suorittimeen pohjautuvissa mikroissa, joissa käytetään Windows/386-ohjelmistoa. Aikaisemmin muistin riittämättömyys on tullut esille Designerin mukana tulevien kuvien käsittelyssä. Käytettäessä sivuttavaa muistia, muistin määrä ei rajoita kuvien kokoa.

Uusi Designer hyödyntää entistä paremmin DX2100-, DX2200- ja DL2600-kirjoittimia, sillä toimituslevykkeillä on laiteohjain FUJIMTRX.DRV, joka on tehty mainittuja kirjoittimia varten. Laiteohjain tukee myös DX-kirjoittimien värioptiota.

Lisätietoja:

Jari Luukkala, p. (90) 5671.

Uusi laser

F-3000 on 18 sivua minuutissa tulostava laser, jossa A4-kokoinen paperi syötetään joko kahdesta syöttökasetista a' 250 arkia tai yksittäissyöttönä. Paperin vastaanotossa on valittavana joko paperi alaspäin tai paperi ylöspäin -vastaanottoalusta.

Tulostimessa on muistia 1,5 megatavua. Tarkkuus on 300x300 pistettä tuumalla. F-3000:ssa on 79 kiinteää kirjaisintyyppiä ja kolme käyttäjän itse ohjelmoitavaa fonttityppiä. Viivakooditulostukseen on valittavana 39 koodityppiä.

Komentokieli on PRESCRIBE ja lisäksi tulostinta voidaan ohjata 32-merkkisen LCD-näytön avulla.

Tulostimessa on kaksi luottokortinkokoista IC-korttipaikkaa, joihin voidaan tallentaa esimerkiksi logoja, makroja lomakepohjia ja tekstityyppejä. Kortit ohjelmoidaan mikroon asennettavalla IC-korttien tallennuslaitteella ja siihen liittyvällä ohjel-

mistolla.

Tulostimesta löytyvät seuraavat emuloinnit: HP Laserjet Series II, IBM Graphics Printer, Diablo 630, Qume Sprint 11, Nec Spinwriter, Line Printer ja Epson FX-80. Sarja- ja rinnakkaisliitäntöjen lisäksi saatavana coax-liitäntä.

F-3000 on tarkoitettu keskuskoneiden massatulostukseen ja verkkoyrjestelmien tulostuslaitteeksi. Sillä pystyy tulostamaan vähintään 600 000 sivua, joka on noin kaksi kertaa enemmän kuin F-1000:lla. Suositeltu käyttö on 5 000 sivua kuukaudessa.

Oikaisu

Edelliseen Softariin oli pujahtanut virheellistä tietoa Paradoxista. Paradox OS/2 ei ole ensimmäinen OS/2-sovellus, mutta ensimmäisiä kylläkin.

Nimet ovat myös menneet sekaisin eli Paradox/386 toimii MS-DOS-käyttöjärjestelmässä ja Paradox OS/2 toimii OS/2-käyttöjärjestelmän suojatussa tilassa. Paradox/386 taas ei toimi OS/2:n suojatussa tilassa eikä OS/2:n DOS-tilassa.

Pahoittelemme sekaannusta!

00003 HELSINKI

Sopimus 00380/166
VASTAUSLÄHETYS
PL 780/10
Softari
Nokia Data

Softart 4/88

☐ Haluan esitteitä seuraavista
Nokian palveluista ja tuotteista.

☐ Pyydän, että minuun otetaan
yhteyttä.

puhelin: _____

Olen kiinnostunut _____

☐ Korjaan yllämainitut osoitetiedot.☐ Poistan yllämainitut osoitetiedot☐ Lisään allamainitut osoitetiedot☐ Softarin☐ NETin rekisteristä/rekisteriin

yritys _____

osasto _____

lähiosoite _____

postinro _____

postitoimip. _____

nimi _____

Nokia
maksaa
postimaksun

Palvelukortti

JOUKKOJULKAISU



8/00010

Mitä jos..

"Uusin ohjelmamme toimii täsmälleen niin kuin ohjelma, jota käytätte juuri nyt", väitti erään itseään arvossa pitävän enklantilaisen lehden kiiltävälle sivulle pranttattu mainos. Kuvassa nökötti keraaminen, kova ja kalju kallo, joka oli lohkottu kuten paraskin paisti.

Ohjelma, joka toimii kuten oma aivosoftani? Tuntui pikaisesti ajateltuna vaikealta uskoa, että joku ohjelmoija muka saisi tutkittua oman pääkoppansa niin tarkkaan, että voisi kopioida sen muiden ihmisten käyttöön. Uskomista ei tehnyt yhtään helpommaksi sen arvovaltaisen lehdykän kiiltävä pinta, sillä arvostan itse tätä mattaa.

Tuskin ne insinöörit sentään omaan kalloonsa kajoavat. Koneita ja ohjelmia he tekevät sen takia, kun ne toimivat niin täydellisesti ihmiseen verrattuna. Insinööri-henkilö saa leikkiä jumalaa – eikä se varkella tahdo paljastaa edes itelleen omia aivoituksiaan. Sellaisiahan ne jumalat ovat.

Mutta kuitenkin – paljon tietävät kaverini kertoivat minulle, että parhaat tietokoneohjelmat ihan oikeasti kiittävät samoja polkuja kuin huippuohjelmioijan oma ajatuksenjuoksu.

Minä siihen tietysti hämmentyneenä ihmettelemään, jotta tarkoittaako se sitä, että tämä maailma tulisi inhimillisemmäksi, jos insinööriksi aikova Jokapoika alkaisi rassaamaan aivojaan. Insien känsäisissä kännyissä on meinaan tämän maapallon tulevaisuus.

Muuttaisiko ajatteleva insinööri teknologian pehmemmaksi – vai meidät ihmiset kovemmiksi? Mitä ihmettä tapahtuisi, jos huippuyksilöt lataisivat koneensa aivotuominnallaan? Tulisiko ihanteeksemme aivokone vai koneai-vo?

Voisiko maailma toisiaankin muuttua paremmaksi paikaksi elää, jos aivojaan rassaavat insinöörit tekisivät kunnon suursiivouksen yläpäässään?

Siivouksesta tulisi tietty kovasti ulosheitettävää roskaa. Sellaista tunne-elämän mössöä, joka pistää pitemmänkin miehen polvilleen. Löytäisikö se luomakunnan kruunu polvillan kontaktissaan itsensä – sen muun roskan ohella, jota on tullut heittyä tienoot täyteen?

Nousisiko sieltä mullasta uusi ihminen kyyneleet silmissään? Lauma Huippujätkeä, jotka veistelisivät tietokoneet jätepuusta ja sisukset niihin kaivelisivat kaatopaikan kansainvälisestäkin korkeasta romutarjonnasta. Nousisiko kaatoksen päälle uusi haitek-yhdyskunta, joka päihittäisi kovan maailman teknologiakylät huippupuhtailla aivoillaan?

Loppuisiko autoilla kaahaileminen ruuhkissa – kun voitaisiin körötellä pitkin tietoliikenneverkkoja? Sanottaisiinko hyvästit tapposasteille, kiireelle ja kuolonkolareille?

Tanssittaisiin auringonlaskuun linnunlaulun tahdissa ja taputettaisiin kavereita olalle. Unohdettaisiin happosasteet, kasvihuoneilmiö ja terroristit...

...ja jatkettais vaan töitä niinkuin maailmassa ei olisi mitään tapahtunutkaan. Sillä eihän siitä syntyisi kuin karmaa kaaos, jos insinöörit ryhtyisivät ajattelemaan.

Softaaja