1541 Kalvolevy-asema Käyttäjän opas

commodore
COMPUTER

VIC-1541
KALVOLEVYASEMA
KÄYTTÄJÄN OPAS

COPYRIGHT
PCI PET-Commodore Inc.
Oy PCI-Data Ab

SISÄLLYSLUETTELO

Sivu

1. **Yleistä** 5
   - Esipuhe 5
   - Erittely 6

2. **Laitteiston kytkeminen** 9
   - Laatikon sisältö 9
   - Kaapelien kytkeminen 10
   - Virran kytkeminen 10
   - Levyn laittaminen asemaan 11
   - Levyaseman käyttö VIC-20:n tai CBM-64:n kanssa 11

3. **Ohjelmien käyttö** 13
   - Levyllä olevien ohjelmien lataus 13
   - LOAD 13
   - Levyn hakemisto 14
   - Nimen lyhentäminen ja kysymysmerkki 15
   - SAVE 15
   - Tallennus aiemmin tallennetun ohjelman päälle 16
   - VERIFY 16
   - DOS-tukiohjelma 16

4. **Levykomennot** 19
   - OPEN ja PRINT # 19
   - NEW 20
   - COPY 20
   - RENAME 20
   - SCRATCH 21
   - INITIALIZE 21
   - VALIDATE 21
   - DUPLICATE 22
   - Virhekanavan lukeminen 22
   - CLOSE 22

5. **Peräkkäistiedostot** 25
   - OPEN 25
   - PRINT # ja INPUT # 25
   - GET # 27
   - Hakemiston lukeminen 29

6. **Hajatiedostot** 33
   - Tietokanavan avaaminen hajatiedostojen lukua varten 33
   - BLOCK-READ 34
   - BLOCK-WRITE 34
   - BLOCK-ALLOCATE 35
   - BLOCK-FREE 35
   - Hajatiedoston käyttö 36
   - BUFFER-POINTER 37
   - USER1 ja USER2 38

7. **Suhteelliset tiedostot** ... 39
Suhteellisten tiedostojen perustaminen ... 39
Suhteellisten tiedostojen käyttö ... 40

8. **Levyohjaimen ohjelmointi** ... 45
BLOCK-EXECUTE ... 45
MEMORY-READ ... 45
MEMORY-WRITE ... 46
MEMORY-EXECUTE ... 46
USER-komennot ... 46

9. **Levyaseman laitenumeron vaihtaminen** ... 49
Ohjelmallinen menetelmä ... 49
Laitteistomenetelmä ... 49

Liitteet
A. Yhteenveto levykomennoista ... 51
B. Yhteenveto levyaseman virheilmoituksista ... 53
DOS-virheilmoitukset ... 55
C. Esittelylevyn ohjelmia ... 59
D. Levyn rakenne ... 71

1. YLEISTÄ

Esipuhe

Tervetuloa 1541 KALVOLEVYASEMAN seuraan. 1541 levyasema on Commodore-64 ja VIC-20 tietokoneiden nopein, tehokkain ja helppokäyttöisin tietojen tallennusjärjestelmä. Tässä käsikirjassa kerrotaan, miten voit hyödyntää kaikki levyaseman ominaisuudet oletpa sitten aloittelija tai ammattilainen.

Jos olet aloittelija, tutustu aluksi muutaman seuraavan kappaleen avulla levyaseman toimintaperiaatteisiin. Taitojesi kasvaessa tulee myös jäljempänä oleville kappaleille käyttöä.

Jos olet ammattilainen, tämä käsikirja kertoo sinulle, että 1541-levyasemalla voidaan suorittaa lähes kaikki tehtävät, mitä levyasemalla voidaan ajatella suoritettavan.

Oletpa minkä tasoinen käyttäjä hyvänsä, 1541-levyasema parantaa tietokonejärjestelmäsi suorituskykyä huomattavasti.

Ennen kuin siirrymme 1541:n käytön yksityiskohtiin, huomautamme muutamasta tärkeästä asiasta.

Tämä käsikirja on vain levyaseman käyttöohje. Jos etsimäsi tieto ei suoranaisesti liity levyasemaan, on se etsittävä joko Commodore-64:n tai VIC-20:n käyttäjän oppaasta tai Programmer's Reference Guide'sta.

Vaikka tässä kirjassa annetaan jokaiselle toiminnalle tarkat ohjeet, edellyttää niiden käyttäminen BASIC-kielen ja sen levykomentojen tuntemista. Jos kuitenkin haluat vain ladata ja tallentaa aiemmin kirjoitettuja ohjelmia, löydät tästä käsikirjasta pelkisteyt ohjeet sitäkin varten.

No niin... siirtykäämme varsinaiseen asiaan.

Levykomennot jakaantuvat eri vaikeusasteisiin. Kappaleessa kolme kerrotaan, millä käskyillä ohjelmat tallennetaan levylle ja ladataan levyltä. Kappaleessa neljä opetetaan käskyjen lähettäminen levylle ja esitellään levyn kunnossapitokomennot.

Kappaleessa viisi kerrotaan, kuinka peräkkäistiedostoja käsitellään. Nämä toiminnot ovat lähes samanlaisia kuin vastaavat kasettiaseman toiminnot (mutta paljon nopeampia). Kappaleessa kuusi esitellään komennot, joiden avulla käsitellään hajatiedostoja: kuinka jokainen levyllä oleva tieto löydetään sekä kuinka levy jakautuu uriin ja lohkoihin. Kappaleessa seitsemän kerrotaan suhteellisista tiedostoista. Suhteellinen tiedosto on paras tapa tallentaa tietokantoja, erityisesti käytettäessä peräkkäistiedostoja niiden apuna.

Kappaleessa kahdeksan kerrotaan levynohjauspiirien ohjelmoinnista konekielellä. Viimeisessä kappaleessa opetetaan laitenumeron vaihtaminen katkaisemalla levyaseman sisällä oleva hyppyjohdin tai ohjelmallisella menetelmällä. Muista, että kaikkia tässä kirjassa olevia asioita ei tarvitse oppia kerralla. Aluksi riittää neljän ensimmäisen kappaleen tunteminen. Kun tutustut vielä pariin seuraavaan kappaleeseen, hallitset jo useimmat toiminnot. Levyasemaan tutustuminen auttaa sinua monella tavoin — käyttö nopeutuu, luotettavuus kasvaa ja tietojenkäsittelykyky tulee paljon monipuolisemmaksi.

Erittely

1541 levyasemalla voidaan tallentaa 144 erilaista ohjelmaa ja/tai tiedostoa yksipuoliselle 5 1/4 '' kalvolevylle. Tallennettavien tavujen maksimimäärä on yli 174 000.

Levyasemassa on sekä levyohjain että täydellinen levynkäyttöjärjestelmä, yhteensä 16 kt ROM- ja 2 kt RAM-muistia. Näiden piirien ansiosta Commodore 1541 levy-yksikkö on ''älykäs'' laite. Tämä tarkoittaa, että levyasema ei tarvitse tietokoneen muistitilaa toimintojensa suorittamiseen. Levyasema voi suorittaa komentoja samaan aikaan, kun tietokone tekee muita tehtäviä. Tämä parantaa huomattavasti järjestelmän toimintoja (syöttöä ja tulostusta).

1541 levyasemalla perustettavia tiedostoja voidaan lukea ja kirjoittaa myös Commodore 4040- ja 2031- levyasemilla ja levyt ovat vaihtokelpoisia näiden järjestelmien välillä. 1541:n avulla voidaan lisäksi lukea vanhemmalla Commodore 2040 levyasemalla tallennettuja ohjelmia.

1541-levyasemassa on kaksisuuntainen sarjaliikenneväylä. Väylän signaalit muistuttavat IEEE-488 rinnakkaisliitäntää, jota käytetään Commodore PET -tietokoneissa. Erona on kuitenkin se, että tässä laitteessa käytetään vain yhtä johdinta kahdeksan sijasta. Sarjaporttiin voidaan liittää enemmän kuin yksi laite, koska levyaseman takana on kaksi liitäntäporttia. Näiden porttien avulla voidaan ketjuttaa useampia laitteita. Tietokoneessa voidaan käyttää jopa neljää levyasemaa ja yhtä kirjoitinta samanaikaisesti.

KUVA 1.1 VIC 1540/1541 levyaseman tekniset tiedot.

TALLENNUSKAPASITEETTI:

Kokonaiskapasiteetti	174848 tavua/levy
Peräkkäistiedostot	168656 tavua/levy
Suhteelliset tiedostot	167132 tavua/levy 65565 tietuetta/tiedosto
Hakemiston merkintöjä	144/levy
Sektoreita/ura	17—21
tavuja/sektori	256
Uria	35
Lohkoja	683 (664 lohkoa vapaana)

IC -PIIRIT:

6502	mikroprosessori
6522(2)	syöttö/tulostus, sisäiset ajastimet
Puskuri 2114(4)	2kt RAM

KOKO:

Mitat

Korkeus	97 mm
Leveys	200 mm
Syvyys	374 mm

SÄHKÖISET OMINAISUUDET:

Jännite	220 V AC
Taajuus	50 Hz
Teho	25 W

TIEDON TALLENNUS

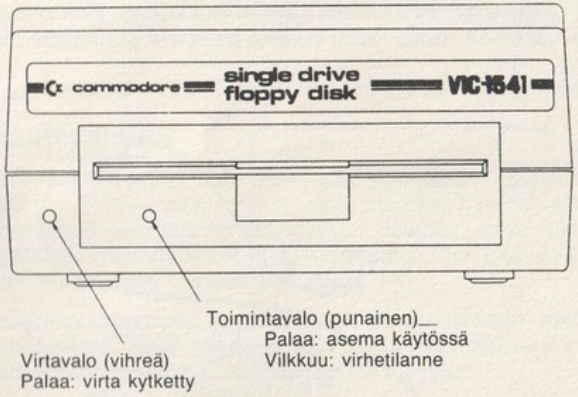
Kalvolevy	Standardi mini 5 1/4'', yksipuolinen

2. LAITTEISTON KYTKEMINEN

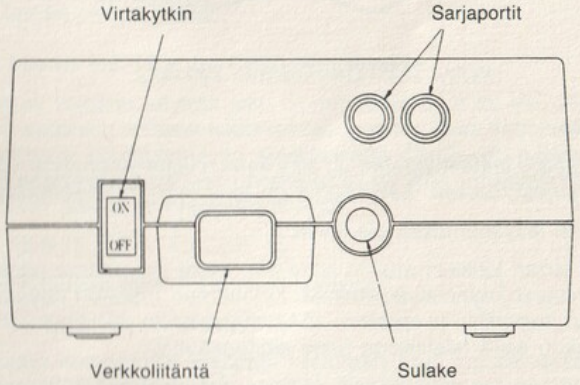
Laatikon sisältö

1541 kalvolevyaseman lisäksi laatikossa on harmaa verkkojohto musta sarjaliitäntäkaapeli, käsikirja ja esittelylevy. Verkkojohdon toisessa päässä on levyaseman takaosaan liitettävä ja toisessa päässä maadoitettuun pistorasiaan liitettävä pistoke. Sarjaliitäntäkaapelin molemmat päät ovat tarkalleen samanlaisia. Sen 6-napaisista pistokkeista toinen liitetään Commodore-64 tai VIC-20 tietokoneeseen ja toinen levyasemaan.

Älä tee mitään ennen kuin olet lukenut seuraavan kappaleen!



KUVA 2.1 Etupaneeli



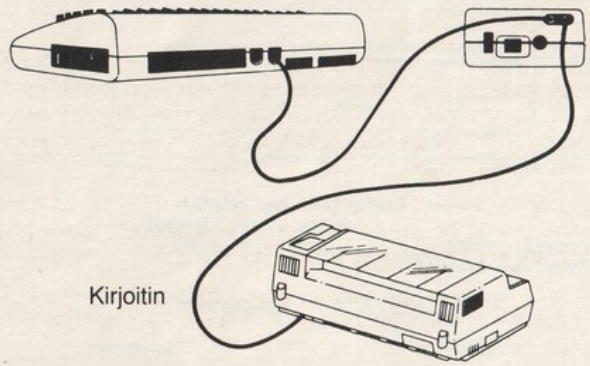
KUVA 2.2 Takapaneeli

Kaapelien kytkeminen

Ensimmäinen tehtävä on verkkokaapelin kytkeminen levyaseman taakse (kts. kuva 2.2). Kaapeli menee paikalleen vain oikein päin. Laita nyt kaapelin toinen pää pistorasiaan. *Jos levyasemasta kuuluu ääntä kytke siitä virta pois takaseinässä olevalla kytkimellä. Älä koskaan liitä levyasemaan kaapeleita, kun siinä on virta päällä.*

Seuraavaksi kytketään sarjaliitäntäkaapeli jompaan kumpaan levyaseman takaseinässä olevaan porttiin. Kytke tietokoneesta virta pois ja kytke kaapelin toinen pää tietokoneen takaosassa olevaan porttiin. Siinä kaikki!

Jos sinulla on kirjoitin tai tai toinen levyasema, kytke sarjaliitäntäkaapelin toinen pää siihen ja toinen pää levyaseman vapaaseen sarjaliitäntäporttiin (kts. kuva 2.3). Kappaleessa 9 on ohjeet useamman levyaseman käyttämistä varten. Ellet ole aiemmin käyttänyt useampaa levyasemaa, aloita ensin yhdellä asemalla ja totuttele järjestelmään.



KUVA 2.3 Levyaseman kytkentä

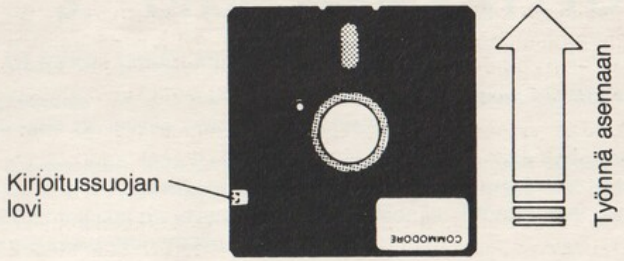
Virran kytkeminen

Kun olet kytkenyt kaikki laitteet yhteen, on aika kytkeä laitteistoon virta. *Laitteet on kytkettävä oikeassa järjestyksessä. Levyasema on aina kytkettävä viimeisenä päälle.*

Tarkista ensin, ettei levyasemassa ole levyä.

Kun olet kytkenyt virran kaikkiin muihin laitteisiin, kytke levyasema päälle. Nyt kaikki muut laitteet suorittavat käynnistysrutiininsa. Kirjoittimen moottori käynnistyy ja kirjoituspää siirtyy telan puoliväliin ja takaisin. 1541 levyaseman punainen virhevalo syttyy ja vihreä valo vilkkuu sekä televisioon tulee aloitusnäyttö.

Kun levyaseman lamppujen vilkkuminen päättyy, voit aloittaa työskentelyn.



KUVA 2.4 Levyn laittaminen asemaan.

Levyn laittaminen asemaan

Avaa levyaseman luukku painamalla sitä hieman sisäänpäin. Luukku nousee itsestään ylös. Jos asemassa on levy, pieni jousi työntää sitä ulos. Ota haluamasi levy ja työnnä se varovasti asemaan niin, että etikettipuoli on ylöspäin, levyn kuoressa oleva suuri aukko menee sisään ensimmäisenä ja kirjoitussuojan lovi (joka on esittelylevyssä peitetty teipillä) on vasemmalla puolella (kts. kuva 2.4).

Työnnä levy asemaan varovasti. Lopulta tunnet pienen liikkeen, eikä levy enää tule itsestään pois asemasta. Sulje levyaseman luukku vetämällä sitä varovasti alaspäin, kunnes se napsahtaa paikalleen. Nyt voit käyttää levyä.

Muista aina ottaa levy pois asemasta ennen kuin katkaiset virran tai kytket virran päälle. Älä koskaan poista levyä asemasta, kun punainen toimintavalo palaa! Asema voi tällöin tuhota levyllä olevat tiedot.

Levyaseman käyttö VIC-20:n tai CBM-64:n kanssa

1541 levyasemaa voidaan käyttää joko Commodore 64:n tai VIC-20:n kanssa. Näillä tietokoneilla on kuitenkin erilaiset tiedonsiirtonopeudet. Siksi tiedonsiirron nopeus määritään ohjelmallisesti. Levyasema on alkutilassaan Commodore-64:n nopeudella. Jos levyasema halutaan kytkeä VIC-20:n nopeudelle, on sille annettava alla oleva komento (kytkettäessä virta päälle tai ohjelman alussa):

```
OPEN 15,8,15,"UI—":CLOSE 15
```

Levyasema palautetaan 64:n nopeudelle käskyllä

```
OPEN 15,8,15,"UI+":CLOSE 15
```

Tämän tyyppisten komentojen käytöstä kerrotaan lisää kappaleessa 4. U-komennon (USER = käyttäjä) kaikki toiminnot esitetään kappaleessa 8.

3. OHJELMIEN KÄYTTÖ

LEVYLLÄ OLEVIEN OHJELMIEN LATAUS

Tässä kappaleessa olevat ohjeet riittävät niille, jotka haluavat käyttää vain valmiita moduulille, kasetille tai levylle tallennettuja ohjelmia.

Jos käytät levyasemaa, laita levy varovasti asemaan niin, että etiketti on levyn yläpuolella ja kauimpana levyasemasta. Tarkista, että levyn kuoressa oleva pieni lovi (se saattaa olla peitetty teipillä) on levyn vasemmalla laidalla. Työnnä levy varovasti asemaan ja sulje suojaluukku painamalla sitä alaspäin.

Kirjoita: LOAD "OHJELMAN NIMI",8 ja paina RETURN-näppäintä. Levyasemasta kuuluu pieni surina ja näyttöön tulee viesti:

SEARCHING FOR OHJELMAN NIMI
LOADING

READY.
■

Kun näyttöön tulee READY. ja ■ vilkkuu, kirjoita RUN. Ohjelman suoritus alkaa, kun painat RETURN-näppäintä.

LOAD

Levyaseman käskyt ovat samoja kuin Commodore kasettiaseman käskyt. Levyasemaa varten on kuitenkin muutama erityiskomento. Tärkein ero on, että kaikissa levyaseman käskyissä on annettava ohjelman nimi. Kasettiasemallahan voidaan ladata ensimmäinen kasetilla oleva ohjelma kirjoittamalla vain LOAD. Levyllä on monta ohjelmaa, jotka ladattaessa ovat saman arvoisia. Siksi levyasemalle on kerrottava, mikä ohjelma halutaan. Tämä tehdään ohjelman nimen avulla. Lisäksi on määritettävä levyaseman laitenumero. Ellei laitenumeroa anneta, tietokone olettaa, että ohjelma luetaan nauhalta.

LOAD-KOMENNON MUOTO:

LOAD nimi$,laiteNo,komentoNo

Ohjelman nimi on merkkijono. Merkkijono on nimi kirjoitettuna lainausmerkkien sisään tai jonkin merkkijonomuuttujan sisältö. Käyttökelpoisia nimiä ovat: "HEI", "OHJELMA 1", A$, NIMI$.

Levyaseman laitenumero on 8. Jos käytät useampaa kuin yhtä levyasemaa, voit muuttaa laitenumeroa kappaleessa 9 kerrotulla tavalla. Tässä kirjassa oletetaan levyaseman laitenumeroksi 8.

KomentoNo ei ole pakollinen. Jos sitä ei anneta tai se on nolla, ohjelma ladataan normaalisti eli ensimmäisestä BASIC-ohjelmaa varten varatusta muistipaikasta lähtien. Jos numero on 1, ohjelma ladataan *tarkalleen muistin siihen paikkaan, missä se on ollut ennen tallennusta.* Jos tietokoneiden muistin rakenne vaihtelee, kuten VICillä käytettäessä 5kt, 8kt tai suurempaa lisämuistia, alkaa muistin BASIC-alue eri paikoista. KomentoNo:n 0 avulla voidaan BASIC-ohjelmat ladata normaalisti. KomentoNo:a 1 käytetään yleensä konekielillä ja merkkiryhmien tai muiden muistista riippuvien toimintojen yhteydessä.

ESIMERKKEJÄ

```
LOAD"TESTI",8          OHJELMAN NIMI
LOAD"OHJELMA 1",8      LAITE No.
LOAD A$,J,K            KOMENTONo.
LOAD"KONEKIELI",8,1
```

HUOM! Laitenumerot, komentonumerot ja merkkijonot voidaan korvata muuttujilla, mikäli ne on ohjelmassa aiemmin määritelty.

Levyn hakemisto

Kasettiasema on peräkkäistiedostoja tallentava laite. Sillä voidaan lukea tiedot vain järjestyksessä nauhan alusta alkaen ja edeten loppua kohti. Kasettiasemaa käytettäessä ei voida hypätä tiedostojen yli eikä palata taaksepäin, jos tiedosto halutaan tallentaa vanhemman version päälle.

Levyasema on hajahakulaite. Luku- ja kirjoituspää voi mennä mihin hyvänsä levyn pisteeseen ja poimia sieltä minkä hyvänsä levyn pisteeseen ja poimia sieltä minkä hyvänsä lohkon. Kussakin lohkossa voi olla 256 tavua tietoa. Levyllä on yhteensä 683 lohkoa.

Onneksi sinun ei itse tarvitse huolehtia yksittäisten lohkojen etsimisestä. Levyasemassa on ohjelma, jonka nimi on levynkäyttöjärjestelmä (Disk Operating System = DOS). Tämä ohjelma jäljittää lohkot puolestasi ja järjestää lohkot lohkokarttaan (Block Availability Map = BAM) ja hakemistoon.

Lohkokartta on luettelo kaikista levyllä olevista 683 lohkosta. Se tallennetaan levyn keskelle, keskinavan ja ulkoreunan puoliväliin. Aina kun ohjelma tallennetaan tai tiedosto suljetaan, BAM päivittyy käytössä olevien lohkojen mukaiseksi.

Hakemisto on levyllä olevien ohjelmien ja tiedostojen nimilista. Levyllä se on heti lohkokartan jälkeen. Hakemistoon sopii 144 tietoa, joista kukin sisältää tiedoston nimen ja tyypin, listan käytetyistä lohkoista ja alkamislohkon. Hakemisto päivittyy automaattisesti aina, kun ohjelma tallennetaan tai tiedosto avataan kirjoittamista varten. Varoitus: Lohkokartta päivittyy vasta, kun tiedosto suljetaan, vaikka hakemisto muuttuu välittömästi toiminnan jälkeen. Ellei tiedostoa suljeta kunnolla, voivat kaikki tiedostossa olevat tiedot tuhoutua.

Hakemisto voidaan ladata tietokoneen muistiin aivan kuten tavallinen BASIC-ohjelma.

Laita levy asemaan ja kirjoita:

```
LOAD"$",8
```

Tietokone vastaa:

```
SEARCHING FOR $
FOUND $
LOADING
READY.
```

Nyt hakemisto on tietokoneen muistissa. Kirjoita LIST, niin saat sen näyttöön. Jos haluat tulostaa hakemiston kirjoittimella, kirjoita alla oleva komento (tässä esimerkissä kirjoittimen laitenumero on 4).


```
OPEN 4,4: CMD 4: LIST
```

HUOM! Käytettäessä CMD-komentoa tiedosto on suljettava komennnolla PRINT#4: CLOSE 4. Asia selitetään tarkemmin VIC 1525/1515 kirjoittimen käsikirjassa.

Hakemisto voidaan lukea lataamatta sitä muistiin DOS-tukiohjelmalla. Hakemistoa voidaan tutkia myös BASIC-ohjelman avulla. Tämä on selitetty kappaleessa 5, joka käsittelee GET# -komentoa.

Nimen lyhentäminen ja kysymysmerkki

Kasettiasemaa käytettäessä voidaan halutulla kirjaimella alkava ohjelma ladata jättämällä jäljellä olevat kirjaimet pois. Siis LOAD "T" lataa ensimmäisen kasetilla olevan ohjelman, jonka nimi alkaa T-kirjaimella. LOAD "HEI" lataa ensimäisen ohjelman, jonka nimi alkaa kirjaimilla HEI, esimerkiksi "HEIPPARALLAA".

Levyaseman käytettäessä nimen lyhentäminen suoritetaan erityismerkin avulla. Kun kirjoitat tähden (*) ohjelman nimen jälkeen, tietokone tietää, että haluat ladata minkä tahansa ohjelman, jonka nimi alkaa kirjoittamillasi kirjaimilla.

LYHENNETYN RAKENTEEN MUOTO:

```
LOAD nimi$+"*",8
```

Voi olla merkkijonomuuttuja tai lainausmerkeissä oleva nimi

Lyhyesti sanottuna: Jos haluat ladata ensimmäisen T:llä alkavan ohjelman, kirjoita: LOAD"T*",8.

Jos nimeksi annetaan "*", ladataan muistiin viimeksi käytössä ollut ohjelma. Ellei mitään ohjelmaa ole aiemmin ladattu, ladataan hakemiston ensimmäinen ohjelma.

Ohjelman nimessä oleva merkki voidaan korvata kysymysmerkillä (?). Levyllä olevan ohjelman nimeä verrataan LOAD -käskyssä annettuun ohjelman nimeen, mutta niitä merkkejä ei tarkisteta, jotka ovat kysymysmerkkien kohdalla.

Jos annetaan esimerkiksi komento LOAD "T?KU",8, täyttävät hakuehdon nimet TIKU, TAKU, jne.

Levyn hakemistoa ladattaessa voidaan käyttää lyhennettyä rakennetta ja kysymysmerkkejä, jos halutaan listata jokin erityinen ohjelmaryhmä. Jos annat komennon LOAD "$0:TESTI",8, tulee hakemistoon vain ohjelma nimeltään TESTI (jos se on levyllä). Komennolla LOAD "S0:T*",8 poimitaan kaikki ohjelmien nimet, jotka alkavat kirjaimella T. Komento LOAD "$0:T?STI",8 poimii kaikki viisikirjaimiset ohjelmien nimet, joiden ensimmäinen kirjain on T, kolmas on S, neljäs on T ja viides I. Komento LOAD "$0:T?STI*",8 etsii kaikki sellaiset nimet, joiden ensimmäinen, kolmas, neljäs ja viides kirjain ovat kriteerin mukaisia.

SAVE

Ohjelma tallennetaan levylle SAVE-komennolla, jonka perään annetaan ohjelman nimi ja laitenumero. Aivan kuten kasettiaseman SAVE-komennossa, voidaan komennon jälkeen antaa komentonumero, joka estää ohjelman automaattisen uudelleensijoittumisen ladattaessa ohjelma takaisin muistiin (kts. aiempi LOAD-komento).

SAVE -KOMENNON MUOTO:

SAVE nimi$,laiteNo,komentoNo

LaiteNo ja komentoNo selitetään LOAD -komennon yhteydessä sivulla 12.

Kun annat levyasemalle SAVE -komennon, levynkäyttöjärjestelmän (DOS:n) on suoritettava useita tarkistuksia. Ensin se tutkii hakemistosta, onko annetun niminen ohjelma jo levyllä. Seuraavaksi se tutkii, onko hakemistossa tilaa uudelle tiedostolle. Kolmanneksi se tutkii lohkokartan ja tarkistaa, onko levyllä tilaa muistissa olevalle ohjelmalle. Jos kaikkiin kysymyksiin tulee myönteinen vastaus, ohjelma tallentuu levylle. Jos jotain on vialla, alkaa virhevalo vilkkua.

Tallennus aiemmin tallennetun ohjelman päälle

Levyllä olevaa ohjelmaa on usein korjailtava ja se on tallennettava uudelleen. Entisen ohjelman poistaminen levyltä ja uuden tallentaminen erikseen olisi epämukavaa.

Jos ohjelman nimen eteen ensimmäiseksi merkiksi kirjoitetaan '' '' ja sen jälkeen 0 sekä kaksoispiste (:), ohjelma tallentuu samalla nimellä tallennetun ohjelman päälle. Levyasema etsii hakemistosta vanhan ohjelman, poistaa siihen kirjoitetut tiedot, perustaa uuden samannimisen hakemistonkohdan ja tallentaa uuden ohjelman normaalisti.

SAVE -KOMENNON MUOTO TALLENNETTAESSA VANHAN OHJELMAN PÄÄLLE:

SAVE ''@ 0:'' + nimi$,laiteNo,komentoNo

Jos tiedoston nimi on esimerkiksi TESTI, se tallennetaan uudelleen samalla nimellä antamalla komento: SAVE''@ 0:TESTI'',8.

Numerolla 0: määritetään se levyaseman yksikkö, jolla komento suoritetaan, jos tietokoneeseen on kytketty kaksoislevyasema. 1541:n yhteydessä ei numerolla ole merkitystä.

VERIFY

VERIFY-komennolla verrataan tietokoneen muistissa ja levyllä olevia ohjelmia keskenään. VERIFY-komentoon on liitettävä myös laiteNo. Tietokone vertaa ohjelman jokaista tavua erikseen rivien linkitykset mukaanlukien. Linkitykset voivat vaihdella muistin koostumuksen vaihdellessa. Jos ohjelma on tallennettu esimerkiksi VIC-20:llä, jossa on 5 kt:n lisämuisti, ja ladataan VIC-20:llä, jossa on 8 kt:n lisämuisti, ei vertailu onnistu, koska linkitykset osoittavat eri muistipaikkoihin.

VERIFY-KOMENNON MUOTO:

VERIFY nimi$, laiteNo

DOS -tukiohjelma

Esittelylevyllä on DOS-tukiohjelma. Levyn hakemistosta se voidaan ladata nimellä WEDGE (VIC-20:lle ja Commodore-64:lle on omat ohjelmansa). Lataa ohjelma ja aja se. Ohjelma poistuu automaattisesti muistista, kun se ajetaan. Muistin tila pienenee tämän ohjelman vuoksi muutamalla sadalla tavulla, mutta se tarjoaa mukavan tavan lähettää levykomentoja.

Kun DOS -tukiohjelma on ajettu, voidaan LOAD -komento korvata merkillä ''/''. Paina vain vino-viiva ja sen jälkeen ohjelman nimi, niin ohjelma ladataan levyltä. Nyt ei enää tarvita LOAD -komentoa, pilkkua eikä numeroa 8.

Levykomennot lähetetään näppäinten ''@ '' tai '' > '' avulla. Jos kirjoitat: @$ (tai >$), hakemisto tulee näyttöön, vaikka sitä ei ladata muistiin. Näillä näppäimillä voidaan myös korvata PRINT# -komento, kun levyasemalle lähetetään seuraavassa kappaleessa kerrottuja komentoja.

Levyaseman virhekanava voidaan tarkistaa (kun punainen valo vilkkuu) näppäilemällä joko @ tai > ja painamalla RETURN. Näyttöön tulee täydellinen virheilmoitus. Se sisältää virhenumeron, tekstin ja uran sekä lohkon numeron.

4. LEVYKOMENNOT

OPEN ja PRINT #

Tähän saakka olemme tutkineet levyaseman yksinkertaisimpia käyttömahdollisuuksia. Jos levyaseman kanssa halutaan työskennellä enemmän, on käytettävä BASIC -kielen OPEN- ja PRINT # -komentoja (näiden komentojen käyttö selitetään tarkemmin VIC-20:n tai Commodore-64:n Käyttäjän Oppaassa). Olet ehkä tottunut niiden käyttöön kasettiaseman kanssa, kun olet avannut tiedostoja OPEN-käskyllä ja kirjoittanut niihin tietoja PRINT # -käskyllä. Näitä komentoja käytetään aivan samalla tavoin levyaseman kanssa. Sen lisäksi näillä komennoilla voidaan käyttää komentokanavaa. Komentokanavan avulla voidaan vaihdella tietoja levyaseman ja tietokoneen välillä.

OPEN-KOMENNON MUOTO:

OPEN tiedostoNo, laiteNo, kanavaNo,tekstiS

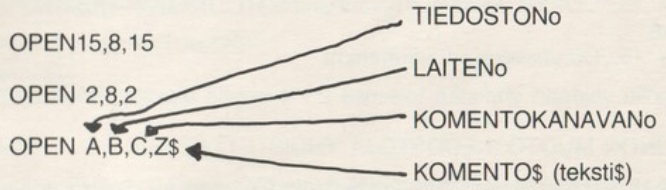
TiedostoNo voi olla mikä tahansa luku väliltä 1—255. Ohjelmassa määritellään tämän numeron avulla, mitä tiedostoa käsitellään. Numeroja, jotka ovat suurempia kuin 127 kannattaa kuitenkin välttää, koska PRINT # -komento aiheuttaa näitä numeroja käytettäessä ylimääräisen rivinsiirron aloitettaessa uusi rivi. Näitä numeroita käytetäänkin varsinaisesti sellaisten kirjoittimien kanssa, joissa ei ole automaattista rivinsiirtoa.

Levyaseman laiteNo on tavallisesti 8.

KanavaNo voi olla mikä hyvänsä luku välillä 2—15. Tällä numerolla määrätään tietokoneen ja levyaseman välinen kommunikointikanava. Kanavat 0 ja 1 on varattu käyttöjärjestelmän käyttöön. Niitä käytetään tietoja tallennettaessa ja ladattaessa. Kanavilla 2—14 sirrellään tietoja tiedostoihin ja kanava 15 on *komentokanava.*

TekstiS on merkkijono, joka kirjoitetaan tiedostoon kuten PRINT # -komennolla. Tällä tavalla on helppo lähettää kanavalle yksi komento.

ESIMERKKEJÄ OPEN-KOMENNON KÄYTÖSTÄ:



PRINT # -komento toimii samoin kuin PRINT-komento. Erona on vain se, että tieto ei mene näytölle vaan toiseen laitteeseen, tässä tapauksessa levyasemaan. Käytettäessä tietokanavaa PRINT # lähettää tiedot levyaseman puskuriin, josta tieto menee edelleen levylle. Käytettäessä PRINT # -komentoa komentokanavalla komento menee suoraan levyasemalle.

LEVYKOMENTOJEN LÄHETTÄMISMUOTO:

OPEN 15,8,15,komento$

tai

PRINT # 15,komento$

NEW

Tätä komentoa tarvitaan, kun levyä käytetään ensimmäistä kertaa. NEW-komento poistaa levyltä kaikki aiemmin tallennetut tiedot, asettaa ajastus- ja lohkomerkit levylle sekä perustaa hakemiston ja lohkokartan. NEW-komennolla voidaan myös poistaa aiemmin formatoidun levyn hakemisto. Tämä on uudelleenformatointia nopeampi tapa.

NEW-KOMENNON MUOTO UUTTA LEVYÄ FORMATOITAESSA:

PRINT # 15, "NEW0:nimi,id" LEVY-YKSIKKÖ

tai lyhennettynä:

PRINT # 15, "N0:nimi,id"

OPEN 1,8,15
PRINT#1, "N0:LEVYNNIMI, TN"
CLOSE 1

NEW-KOMENNON MUOTO HAKEMISTOA POISTETTAESSA:

PRINT # 15,"N0:nimi"

Nimi on levyn nimi. Se tulee näyttöön vain listattaessa hakemisto. ID-koodi on mikä hyvänsä kahden merkin pituinen tunniste, ja se tallennetaan hakemiston lisäksi kaikkiin levyn lohkoihin. Koska levyasema tarkastaa aina ID-koodin se huomaa, jos poistat epähuomiossa levyn asemasta tallennuksen aikana.

COPY

Tällä komennolla voidaan kopioida mikä hyvänsä levyllä oleva ohjelma tai tiedosto. Kopiointia ei voida suorittaa levyasemasta toiseen (ellei käytetä 4040:n tyyppistä kaksoislevyasemaa), vaan ohjelma on tallennettava samalle levylle toisella nimellä.

COPY-KOMENNON MUOTO:

PRINT # 15,"COPY0:uusinimi = 0:vanhanimi" LEVY-YKSIKKÖ

tai lyhennettynä:

PRINT # 15,"C0:uusinimi = 0:vanhanimi"

COPY-komennolla voidaan yhdistää toisiinsa 2 - 4 levyllä olevaa tiedostoa.

COPY-KOMENNON MUOTO TIEDOSTOJA YHDISTETTÄESSÄ:

PRINT # 15"C0:uustied = 0:vanhatied1,0:vtied2,0:vtied3,0:vtied4"

ESIMERKKEJÄ COPY-KOMENNOSTA:

PRINT # 15,"C0:KOPIO = 0:ALKUPERÄINEN"
PRINT # 15,"C0:PÄÄTIEDOSTO = 0:NIMI,0:OSOITE,0:PUHELIN"

RENAME

Tällä komennolla voidaan vaihtaa hakemistossa olevan ohjelman nimi. Toimitus on nopea, koska nimen vaihto on ainoa toimenpide.

RENAME-KOMENNON MUOTO:

PRINT # 15,"RENAME0:uusinimi = vanhanimi"

tai lyhyemmin:

PRINT # 15,"R0:uusinimi = vanhanimi"

ESIMERKKI RENAME-KOMENNOSTA:

PRINT # 15,"R0:MYYRÄ = MÄYRÄ"

RENAME-komento toimii vain avattujen tiedostojen yhteydessä.

SCRATCH

Tämän komennon avulla voidaan poistaa levyltä tarpeettomat tiedostot ja ohjelmat. Näin saadaan levylle tilaa uusien tietojen syöttämistä varten. Ohjelmia voidaan poistaa yksitellen tai ryhmissä nimiä lyhentämällä ja/tai kysymysmerkkien avulla.

SCRATCH-KOMENNON MUOTO:

PRINT # 15,"SCRATCH0:nimi" LEVY-YKSIKKÖ

tai lyhyemmin:

PRINT # 15,"S0:nimi"

Jos tarkistat virhekanavan SCRATCH-komennon jälkeen, niin uranumerolle varattu numero kertoo montako tiedostoa olet pyyhkinyt. Jos hakemistossa on esimerkiksi tiedostot KEPPI ja TAPPI, voit pyyhkiä molemmat tiedostot pois komennolla: PRINT # 15,"S0:??PPI". Jos hakemistossa on tiedostot nimeltään: KONE, KAIRA, KUUME ja KRUUNA, voit pyyhkiä ne kaikki pois komennolla PRINT # 15,"S0:K*".

INITIALIZE

Ellet voi suorittaa haluamaasi toimintaa, koska levyasema on virhetilassa, INITIALIZE-komento palauttaa levyaseman siihen tilaan, missä se on kytkettäessä virta päälle. Levyasema on sovitettava tämän jälkeen tietokoneen nopeudelle (kts. kpl.2).

NITIALIZE-KOMENNON MUOTO:

PRINT # 15,"INITIALIZE"

tai lyhyemmin:

PRINT # 15,"I"

VALIDATE

Kun levyä on käytetty jonkin aikaa, voi hakemisto mennä epäjärjestykseen. Kun tiedostoja tallennetaan ja poistetaan useita kertoja, voi levylle jäädä pieniä, muutaman lohkon pituisia tyhjiä välejä. Ne ovat käyttökelvottomia pienuutensa vuoksi. VALIDATE -komennolla järjestetään levyllä olevat tiedostot uudelleen niin, että kaikki lohkot voidaan käyttää hyväksi.

Levyllä voi olla myös avattuja tiedostoja, joita ei ole suljettu kunnolla. VALIDATE-komento poistaa myös tällaiset tiedostot, koska ne ovat käyttökelvottomia.

Tämän komennon käytössä on pieni riski. Hajatiedostoja (kts. kappale 6) käytettäessä poistuvat tiedostoa varten varatut lohkot. Siksi VALIDATE -komentoa ei pitäisi koskaan antaa levylle, jossa on hajatiedostoja.

VALIDATE-KOMENNON MUOTO

```
PRINT# 15,"VALIDATE"
```

tai lyhyemmin

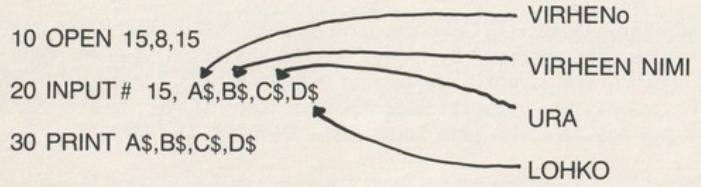
```
PRINT# 15,"V"
```

DUPLICATE

Tämä on kaksoislevyaseman (esim. 4040) käyttöjärjestelmän komento, jolla kopioidaan toisessa levy-yksikössä olevan levyn sisältö toisessa yksikössä olevalle levylle. Yhden levyn asemassa tällä komennolla ei ole käyttöä.

Virhekanavan lukeminen

Virhekanavaa ei voida lukea ilman DOS-tukiohjelmaa, ellei sitä varten kirjoiteta omaa ohjelmaa. Virhekanavan lukemisessa tarvittava INPUT# -komento ei toimi ohjelman ulkopuolella. Virhekanava voidaan lukea tämän lyhyen BASIC-rutiinin avulla.



INPUT#-komennolla luetaan komentokanavalta neljä virhetilannetta kuvaavaa muuttujaa. Ensimmäinen, kolmas ja neljäs muuttuja ovat numeroita. Ne voidaan lukea myös numeerisiin muuttujiin. Ensimmäinen muuttuja on virhenumero; nolla tarkoittaa, että virhetilannetta ei ole. Toinen muuttuja on virhetilanteen kuvaus. Kolmas muuttuja on virhepaikan uranumero ja neljäs on lohkonumero. (Lohkoa voidaan kutsua myös sektoriksi.)

Uralla 18 olevat virhetilanteet liittyvät lohkokarttaan ja hakemistoon. Esimerkiksi READ ERROR (lukuvirhe) uran 18 lohkossa 0 merkitsee, että levyä ei ole formatoitu.

CLOSE

Kun tiedostoa on käytetty, se on suljettava kunnolla CLOSE-komennon avulla. Tiedostoa suljettaessa DOS merkitsee lohkot lohkokarttaan ja lopettaa hakemiston täyttämisen. ELLET SULJE TIEDOSTOA, TIEDOT KATOAVAT!

CLOSE-KOMENNON MUOTO:

```
CLOSE tiedostoNo
```

Virhekanavaa (kanavaNo. 15) ei saa sulkea ennen kuin tietokanavat suljetaan. VIRHEKANAVA ON AVATTAVA ENSIMMÄISENÄ JA SULJETTAVA VIIMEISENÄ. Näin levyasema toimii virheettömästi.

Jos suljet virhekanavan, kun muut kanavat ovat auki, levyasema sulkee muutkin kanavat. BASIC-ohjelma luulee kuitenkin, että kanavat ovat auki (ellei niitä suljeta kunnolla) ja voi yrittää käyttää niitä.

HUOM! Jos BASIC-ohjelma ajautuu virhetilanteeseen, kaikki BASIC-tiedostot suljetaan, mutta levyaseman tiedostot jäävät auki. Tämä on erittäin vaarallinen tilanne. Tällaisessa tilanteessa on välittömästi kirjoitettava OPEN 15,8,15,"I". Näin levyasema alustetaan uudelleen ja tiedostot säilyvät käyttökelpoisina.

5. PERÄKKÄISTIEDOSTOT

OPEN

Levyaseman peräkkäistiedostot toimivat tarkalleen samoin kuin kasettiaseman peräkkäistiedostot, mutta paljon nopeammin. Peräkkäistiedostojen käyttöä rajoittaa se, että niitä on luettava järjestyksessä alusta loppuun. Tiedot siirretään peräkkäisinä tavuina puskurin kautta magneettiselle tallennusvälineelle. Levyaseman kaikki tiedostot luodaan samalla tavalla. Sekä peräkkäistiedostot, ohjelmatiedostot että käyttäjän tiedostot tallennetaan levylle samanlaisina. Ainoa ero on, että ohjelmatiedosto voidaan ladata LOAD-komennolla. Hakemistokin on ohjelma-tiedosto, vaikka siitä voidaan vain lukea tiedot. Ainoa erilainen tiedosto on suhteellinen tiedosto.

PERÄKKÄISTIEDOSTON AVAUKSEN MUOTO:

OPEN tiedostoNo, laiteNo, kanavaNo, "0:nimi, tyyppi, suunta"

Tiedoston numero on sama kuin kaikissa muissa OPEN-komennon sovellutuksissa ja sen avulla viitataan koko ohjelman ajan tiettyyn tiedostoon. LaiteNo on tavallisesti 8. KanavaNo on tietokanavan numero välillä 2—14. On mukavinta käyttää samaa numeroa sekä kanava- että tiedostonumerona, näin vältytään parhaiten sekaannuksilta. Nimi on tiedoston nimi (jos perustetaan kirjoitustiedostoa, ei kysymysmerkkiä eikä lyhennettyjä nimiä voida käyttää). Tyyppi voi olla mikä tahansa alla oleva tyyppi. Tyypin nimestä on kirjoitettava ainakin ensimmäinen kirjain. Suunnaksi on kirjoitettava joko READ (luku) tai WRITE (kirjoitus) tai ainakin jomman kumman sanan ensimmäinen kirjain.

TIEDOSTON TYYPPI	MERKITYS
PRG	Ohjelma (Program)
SEQ	Peräkkäistiedosto (Sequential)
USR	Käyttäjä (User)
REL	Suhteellinen (Relative)

ESIMERKKEJÄ PERÄKKÄISTIEDOSTON AVAAMISESTA:

OPEN 2,8,2,"0:DATA,S,W"	TIEDOSTON NIMI
OPEN 8,8,8,"0:OHJELMA,P,R"	TYYPPI
OPEN A,B,C,"0:" + A$ + "U,W"	READ/WRITE

Jos tiedosto on jo levyllä, se voidaan korvata uudella samoin kuin SAVE-komennossa korvattiin vanha tiedosto (kts. kappale 3). Nimen eteen lisätään vain a 0:.

ESIMERKKI PERÄKKÄISTIEDOSTON KIRJOITTAMISESTA TOISEN PÄÄLLE:

OPEN 2,8,2"@ 0:DATA,S,W"

PRINT # ja INPUT

PRINT # -komento toimii tarkalleen samoin kuin PRINT-komento. Erona on se, että PRINT # -komento suunnataan suoraan levyasemalle. Tämän komennon erityispainotuksen syynä on se, että kaikki PRINT-komennon muotoiluominaisuudet toimivat myös

tällä komennolla. Kannattaa siis olla varovainen kirjoitettaessa tietoja tiedostoon PRINT#-komennolla.

PRINT#-KOMENNON MUOTO TIEDOSTOON KIRJOITETTAESSA:

PRINT# tiedostoNo, tietolista

Tiedostonumero on sama, jota käytettiin OPEN-komennossa tiedostoa luotaessa.

Tietolista on sama, kuin normaalissa PRINT-komennossa eli muuttujalista ja/tai lainausmerkeissä oleva teksti. Tietoja kirjoitettaessa on kuitenkin oltava erityisen varovainen, jotta ne voidaan lukea myöhemmin.

Jos PRINT# -komennossa erotetaan tiedot pilkuilla toisistaan, niiden väliin jää tyhjiä välejä aivan kuin näytölläkin. Puolipistettä käytettäessä ei ylimääräistä tilaa varata.

Seuraavassa on kuvio, joka esittää asian havainnollisemmin. Se kuvaa peräkkäistiedostoa, joka on perustettu komennolla: OPEN 5,8,5,"0:TESTI,S,W".

	eof															
char	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...

Merkki **eof** tarkoittaa tiedoston loppumerkkiä (End Of File). **Char** = merkki. Tiedostoon kirjoitettava merkkijono tallentuu tiedostoon merkeittäin, välilyönnit mukaanluettuna. Asettakaamme muutamia muuttujia. A$ = "HELLO"; B$ = "ALL"; C$ = "BYE". Tiedosto on alla olevan kuvan mukainen, kun annetaan komento: PRINT# 5,A$;B$;C$.

	H	E	L	L	O	A	L	L	B	Y	E	CR	eof
char	1	2	3	4	5	6	7	8	9	10	11	12	13

CR on CHR$(13) -koodi eli kirjoituspään palautus rivin alkuun (Carriage Return). Se merkitään jokaisen PRINT- ja PRINT# komennon loppuun ellei rivi pääty puolipisteeseen tai pilkkuun.

HUOM! Älä jätä välilyöntiä PRINT-sanan ja #-merkin väliin äläkä yritä lyhentää komentoa näin: ?#. Käyttäjän Oppaan liitteestä löydät oikean lyhenteen.

INPUT#-KOMENNON MUOTO:

INPUT# tiedostoNo, muuttujalista

Kun tiedot luetaan muistiin INPUT#-komennolla, ei tietokoneelle voida mitenkään kertoa, että tieto ei ole yksi pitkä merkkijono. Tarvitaan jonkilainen merkki, joka erottaa tiedon osat toisistaan. Erottamiseen käytettäviä merkkejä ovat esimerkiksi CR, pilkku tai puolipiste. CR on helppo lisätä: kun kirjoitetaan jokaiseen PRINT# -komentoon vain yksi jono, niin erottaja tulee automaattisesti. Komento PRINT# 5, A$: PRINT# 5, B$: PRINT# 5, C$ lisää CR:n jokaisen muuttujan jälkeen. Näin merkkijonot on erotettu riittävän hyvin komentoa INPUT# 5,A$,B$,C$ varten. Toinen mahdollinen tapa on seuraava: Z$ = ",":PRINT# 5, AZBZC$. Näin sama toimitus vie vähemmän tilaa. Tällaisen komennon jälkeen tiedosto on seuraavanlainen:

	H	E	L	L	O	,	A	L	L	,	B	Y	E	CR	eof
char	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Jos muuttujien väliin kirjoitetaan pilkut, kuluu levyltä paljon turhaa tilaa. Komento PRINT# 5,A$,B$ näkyy tiedostossa seuraavasti:

	H	E	L	L	O							A	L	L		CR	eof
char	1	2	3	4	5	6	7	8	9	10	11	12	13	14	...	23	24

Tästä näkyy, että tilaa menee paljon hukkaan.

Tämän kappaleen opetus on: Kun käytät PRINT# -komentoa, ole huolellinen, jotta pystyisit lukemaan tallentamasi tiedot.

Tiedostoon kirjoitettu numeerinen tieto muuttuu merkkijonoksi aivan kuin olisi tehty STR$-operaatio ennen merkkijonon kirjoittamista tiedostoon. Ensimmäinen merkki on tyhjä, jos luku on positiivinen. Ensimmäiseksi merkiksi tulee miinus (—), jos luku on negatiivinen. Seuraavaksi tulee luku ja jonon viimeinen merkki on 'kursori oikealle' -merkki. Jos luvut ovat positiivisia, tulee levylle jonkin verran hukkatilaa, koska käyttämättä jää jokaista numeroa kohden kaksi merkkiä.

Komennolla PRINT#5, 1;3;5;7 on tiedoston rakenne seuraava:

		1	→		3	→		5	→		7	→	CR	eof	
char	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

PRINT#-komentoa ei kuitenkaan kannata käyttää näin, koska lukuja ei voida erottaa luettaessa toisistaan. Katso erottamismenetelmät edelliseltä sivulta.

GET#

GET# noutaa merkit levyltä yksitellen.

GET#-KOMENNON MUOTO:

GET# tiedostoNo,muuttujaluettelo

Tieto noudetaan yksi tavu kerrallaan, vaikka tavussa olisi CR, pilkku tai jokin muu erottava merkki. GET#-komennon kanssa on turvallisinta käyttää merkkijonoja, koska BASIC-ohjelma antaa virheilmoituksen, jos tiedostosta löytyy merkkijono, kun merkin oletetaan olevan numero.

ESIMERKKEJÄ GET# -KOMENNOSTA:

GET# 5,A$
GET# A,B$,C$,D$
GET# 5,A

Useampia merkkejä voidaan noutaa kerralla

GET# -komento on erittäin hyödyllinen tutkittaessa tiedostoa, jonka sisältöä ei tunneta, esimerkiksi tiedostoa, joka on voinut tuhoutua ohjelmaa kokeiltaessa. Tämä komento on turvallisempi kuin INPUT#, koska erottavien merkkien välillä olevien merkkien määrää ei ole rajoitettu. GET# -komennolla voidaan ottaa jokainen merkki vastaan ja sen avulla voidaan tutkia myös erottavat merkit.

Tällä esimerkkiohjelmalla voidaan tutkia jokaisen levyllä olevan tiedoston sisältö.

```
10   INPUT"TIEDOSTON NIMI";F$
20   INPUT"TIEDOSTON TYYPPI";T$
30   T$=LEFT$(T$,1)
40   IF T$<>"S" THEN IF T$<>"P" THEN IF T$<>"U" THEN 20
45   OPEN 15,8,15
50   OPEN 5,8,5,"0:"+F$+","+T$+",R"
60   GOSUB 200
70   GET# 5,A$
80   IF ST<>0 THEN PRINT ST: STOP
90   PRINT ASC(A$+CHR$(0));
100  GOTO 70
200  INPUT# 15,A$,B$,C$,D$
210  IF VAL (A$)>0 THEN PRINT A$,B$,C$;D$:STOP
220  RETURN
```

Lisätään siltä varalta, että luettu merkki on nollajono. Muuten ASC-funktio on virheellinen

virhe No
virhean nimi
URA
Lohko

virheen ilmoitus

Hakemiston lukeminen

Levyn hakemisto voidaan lukea aivan kuin muutkin peräkkäistiedostot. Tiedoston nimeksi annetaan "$". OPEN -komento on: OPEN 5,8,5,"$". Nyt GET# tutkii hakemistoa. Hakemiston muoto on samanlainen kuin ohjelmatiedoston muoto. Tiedostojen koot vastaavat rivinumeroja ja nimet on tallennettu lainausmerkeissä oleviksi merkkijonoiksi.

Levyn hakemisto voidaan tutkia tällä ohjelmalla.

```
10   OPEN 15,8,15: OPEN 1,8,1,"$"                                  LUE BAM
20   FOR L=1 TO 142:GET# 1,A$:NEXT L
30   T$(0)="DEL":T$(1)="SEQ":T$(2)="PRG":T$(3)="USR":T$(4)="REL"
40   J=17: GOSUB 500
50   N$=B$                                                         LEVYN NIMI
60   J=2                                                           ID
70   GOSUB 500
80   I$=B$
90   J=2                                                           DOS
100  GOSUB 500
110  O$=B$
120  FOR L=1 TO 88                                                 TUTKI LOHKON LOPPUOSA
130  GET#1,A$
140  NEXT L
150  PRINT CHR$(147)"LEVYN NIMI:";N$,"ID:";I$,"DOS:";O$
160  PRINT"PITUUS,"TYYPPI","NIMI"
165  FOR P=1 TO 8
170  GET# 1,T$,A$,A$
175  IF ST THEN CLOSE1:CLOSE15:END
180  IF T$=""" THEN T$=CHR$(128)                                   TIEDOSTON NIMI
190  J=15
200  GOSUB 500                                                     TIEDOSTON PITUUS
210  N$=B$
220  GET# 1,A$,A$,A$,A$,A$,A$,A$,A$,A$,A$,L$,H$
225  L=ASC(L$+CHR$(0))+256*ASC(H$+CHR$(0)):IF L=0 THEN 250
230  PRINT L,T$(ASC(T$)-128),N$
250  IF P< 8 THEN GET# 1,A$,A$
260  NEXT P: GOTO165
500  B$=""                                                         MERKKI-
510  FOR L=0 TO J                                                  JONON
520  GET# 1,A$                                                     MUO-
530  IF A$<>CHR$(96)THEN IF A$<>CHR$(160)THEN B$=B$+A$             DOSTUS
540  NEXT
550  RETURN
```

TAULUKKO 5.1: 1540/1541:n lohkokartan muoto

Ura 18, sektori 0.		
TAVU	SISÄLTÖ	MÄÄRITELMÄ
0,1	18,01	Ensimmäisen hakemistolohkon ura ja sektori
2	65	ASCII -merkki, joka ilmaisee 4040-rakenteen
3	0	Myöhemmän DOS-käytön nollalippu.
4-143		*Bittikartta. Ilmaisee, onko lohkoja vapaana.
*1 = lohko vapaa 0 = lohko varattu (kukin bitti edustaa yhtä lohkoa)		

TAULUKKO 5.2: 1540/1541:n hakemiston otsikko

Ura 18, sektori 0.		
TAVU	SISÄLTÖ	MÄÄRITELMÄ
144-161		Levyn nimi täytettynä SHIFTin kanssa annetuilla välilyönneillä
162-163		Levyn tunniste
164	160	SHIFT + välilyönti
165,166	50,65	2A:n eli DOS version ja muototyypin ASCII-muoto
166-167	160	SHIFT + välilyönti
171-255	0	Nollia, ei käytössä
HUOM! Joidenkin levyjen paikkoihin 180-191 voi tulla ASCII-merkkejä.		

TAULUKKO 5.3: Hakemiston rakenne

Ura 18, Sektori 1	
TAVU	MÄÄRITELMÄ
0,1	Seuraavan hakemiston lohkon ura ja sektori
2-31	*Tiedoston määrittely 1
34-63	*Tiedoston määrittely 2
66-95	*Tiedoston määrittely 3
98-127	*Tiedoston määrittely 4
130-159	*Tiedoston määrittely 5
162-191	*Tiedoston määrittely 6
194-223	*Tiedoston määrittely 7
226-255	*Tiedoston määrittely merkintä 8

*YHDEN HAKEMISTOPAIKAN RAKENNE

TAVU	SISÄLTÖ	MÄÄRITELMÄ
0	128 + tyyppi	Tiedoston tyypin ja $80:n TAI-funktio TYYPIT: 0 = poistettu (DEL) 1 = peräkkäistiedosto (SEQ) 2 = ohjelma (PRG) 3 = käyttäjä (USR) 4 = suhteellinen tiedosto (REL)
1,2		Ensimmäisen tietolohkon ura ja sektori
3-18		SHIFTin kanssa annetuilla välilyönneillä täytetty tiedoston nimi.
19,20		Vain suhteellisille tiedostoille: sivusektorilohkon ura ja sektori.
21		Suhteellisen tiedoston tietuekoko
22-25		Käyttämätön
26,27		Tiedoston korvaamismerkin ura ja sektori, kun käytössä on OPEN @
28,29		Tiedostossa olevien lohkojen lukumäärä: vähemmän ja enemmän merkitsevä tavu.

TAULUKKO 5.4: Peräkkäistiedoston rakenne

TAVU	MÄÄRITELMÄ
0,1	Seuraavan peräkkäistiedoston lohkon ura ja sektori
2-256	254 tavua tietoja, joihin sisältyvät CR:t ja keskeytysmerkit

TAULUKKO 5.5: Ohjelmatiedoston rakenne

TAVU	MÄÄRITELMÄ
0,1	Seuraavan ohjelmatiedoston lohkon ura ja sektori
2-256	254 tavua tietoja ohjelmasta tallennettuna CBM:n muistin muodossa (käskyt koodattuina). Tiedoston päättyminen merkitään kolmena nollatavuna.

6. HAJATIEDOSTOT

Peräkkäistiedostot ovat hyviä, kun käsitellään jatkuvia tietojonoja, mutta kaikkiin sovellutuksiin ne eivät riitä. Esimerkiksi postituslistojen kanssa työskenneltäessä ei ole mukavaa kahloa koko tiedostoa läpi etsittäessä yhtä henkilöä. Tällaiset tiedot on voitava tallentaa jonkinlaisiin **hajatiedostoihin,** joista voidaan valita mikä tahansa yksittäinen tieto tarvitsematta selata koko tiedostoa.

Commodore levyasemmassa on kahta erityyppistä hajatiedostoa. Seuraavassa kappaleessa käsiteltävät suhteelliset tiedostot ovat mukavia tietojenkäsittelytiedostoja. Tässä kappaleessa käsiteltäviä hajatiedostoja käytetään toisenlaisissa sovellutuksissa, erityisesti konekielellä työskenneltäessä.

Commodore levyaseman hajatiedostoista voidaan valita mikä hyvänsä yksittäinen levylle tallennettu tieto. Kuten ensimmäisessä kappaleessa mainittiin, tyhjällä levyllä on 683 lohkoa, joista 644 on vapaasti käytettävissä. Kukin lohko tarkoittaa yhtä jollakin uralla olevaa sektoria.

Levy jaetaan uriin, jotka ovat levyn kanssa samankeskeisiä ympyröitä. Levyllä on 35 uraa. Ensimmäinen ura on levyn ulkoreunalla ja ura 35 on sisimpänä. Ura 18 on varattu hakemistoa varten. Levynkäyttöjärjestelmä käyttää tarvittavan määrän sisimpiä uria.

Jokainen ura jaetaan edelleen sektoreihin. Koska uloimmilla urilla on enemmän tilaa, niissä on enemmän sektoreita kuin sisemmillä urilla. Uloimmilla urilla on 21 sektoria ja keskimmäisillä urilla on 17. Seuraavasta taulukosta näkyy kullakin uralla olevien sektoreiden määrä.

TAULUKKO 6.1: Urien lohkorakenne

URAN NUMERO	SEKTORIEN NUMEROT	SEKTORIEN MÄÄRÄ
1—17	0—20	21 357
18—24	0—18	19 133
25—30	0—17	18 108
31—35	0—16	17 85

Levynkäyttöjärjestelmässä olevien käskyjen avulla voidaan mille tahansa levyn uralle kirjoittaa tietoa tai lukea jokaiselta uralta ja sektorilta siellä olevaa tietoa. Lisäksi voidaan tarkistaa, mitkä lohkot (urat ja sektorit) ovat vapaina sekä merkitä käytetyt urat.

Nämä komennot lähetetään komentokanavalla (kanavaNo 15). Ne kertovat levyasemalle, mitä tiedoille on tehtävä. Tiedot luetaan avatun kanavan kautta.

Tietokanavan avaaminen hajatietojen lukua varten

Kun työskennellään hajatiedostojen kanssa, on levyasemalle avattava kaksi kanavaa: toinen komentoja, toinen tietoja varten. Komentokanava avataan kanavalle 15. Hajahakua varten avataan tietokanava antamalla tiedoston nimeksi ristikkomerkki (#).

OPEN-KOMENNON MUOTO AVATTAESSA KANAVA HAJAHAKUA VARTEN:

```
OPEN tiedostoNo, laiteNo, kanavaNo, "#"
```

tai:

```
OPEN tiedostoNo, laiteNo, kanavaNo, "# puskuriNo"
```

ESIMERKKEJÄ TIETOKANAVAN AVAAMISESTA HAJAHAKUA VARTEN:

```
OPEN 5,8,5,"#"        PUSKURIN NUMEROLLA EI VÄLIÄ
OPEN A,B,C,"#2"       VALITSE PUSKURI No 2
```

BLOCK-READ

BLOCK-READ -KOMENNON MUOTO:

```
PRINT# tiedostoNo,"BLOCK-READ:"kanava;levy-yksikkö;ura;lohko
```

tai lyhennettynä:

```
PRINT# tiedostoNo,"B-R:" kanava;levy-yksikkö;ura;lohko
```

Tällä komennolla siirretään tietolohko levyltä valitulle kanavalle. Tieto voidaan myöhemmin lukea INPUT#- ja GET#-komennoilla.

ESIMERKKIOHJELMA, JOKA LUKEE URAN 18 LOHKOSSA 2 OLEVAT TIEDOT (ja tallentaa sisällön B$:iin):

```
10  OPEN 15,8,15                          KANAVANo
20  OPEN 5,8,5,"#"                        LEVY-YKSIKKÖ
30  PRINT# 15,"B-R:" 5;0;18;2             SEKTORI
40  B$=""                                 } =LOHKO
50  FOR L=0 TO 255                        URA
60  GET# 5,A$
70  IF ST=0 THEN B$=B$+A$: NEXT L
80  PRINT "VALMIS"                        KOKOAA TAVUISTA
90  CLOSE 5:CLOSE 15                      LOHKON
```

BLOCK-WRITE

BLOCK-WRITE -komento on BLOCK-READ -komennon vastakohta. Ensin puskuriin on kirjoitettava tieto, sen jälkeen puskurin sisältö kirjoitetaan levyn oikeaan paikkaan.

BLOCK-WRITE -KOMENNON MUOTO:

```
PRINT# tiedostoNo, "BLOCK-WRITE:" kanava;yksikkö;ura;lohko
```

tai lyhennettynä:

```
PRINT# tiedostoNo, "B-W:" kanava;levy-yksikkö;ura;lohko
```

Kun tieto tallennetaan puskuriin, DOS:n osoitin lukee, montako merkkiä siellä on. BLOCK-WRITE -komennolla tämä osoitin tallentuu levylle. Siksi yllä olevan ohjelman rivillä 70 tarkistetaan ST (tilamuuttuja). ST saa nollasta poikkeavan arvon, jos yrität lukea EOF-merkin jälkeen olevaa lohkoa.

ESIMERKKIOHJELMA, JOKA KIRJOITTAA TIETOA URAN 1 SEKTORIIN 1:

```
10  OPEN 15,8,15
20  OPEN 5,8,5, "#"
30  FOR I=1 TO 40
40  PRINT# 5,"TESTI"
50  NEXT I
60  PRINT# 15,"B-W:"5;0;1;1
70  CLOSE5:CLOSE15
```

BLOCK-ALLOCATE

Jos halutaan varmistaa, että hajatiedosto toimii suhteellisena tiedostona, on vapaita lohkoja etsittäessä tarkistettava lohkokartta ja muutettava lohkokartta todellisuutta vastaavaksi, kun tiedot ovat levyllä. Kun lohkokartta on päivitetty, hajatiedosto toimii kunnolla — ellei levylle anneta VALIDATE-komentoa (kts. kappale 3).

BLOCK-ALLOCATE -KOMENNON MUOTO:

```
PRINT# tiedostoNo,"BLOCK-ALLOCATE:" levy-yksikkö;ura;lohko
```

tai lyhyemmin:

```
PRINT# tiedostoNo,"B-A:" levy-yksikkö;ura;lohko
```

Kuinka saadaan selville, mitkä lohkot ovat vapaana? Jos yrität tallentaa käytössä olevaan lohkoon, DOS antaa virheilmoituksen 65, NO BLOCK ja vaihtaa ura- ja lohkonumeroksi seuraavan vapaan lohkon ura- ja sektorinumeron. Tämän vuoksi kannattaa aina yrittää varata lohko ennen tiedon tallentamista. Jos lohko on varattu, varaa seuraava vapaa lohko. Voit lukea sen suoraan virhekanavalta.

ESIMERKKI LOHKON VARAUKSESTA:

```
10  OPEN 15,8,15
20  OPEN 5,8,5,"#"                        LEVY-YKSIKKÖ
30  PRINT# 5,"TIEDOT"
40  U=1: S=1                              URA
50  PRINT# 15,"B-A:"0,U,S
60  INPUT# 15,A,B$,C,D                    SEKTORI
70  IF A=65 THEN U=C: S=D: GOTO 50
80  PRINT# 15,"B-W:"5;0;U;S
90  CLOSE 5: CLOSE 15
```

BLOCK-FREE

BLOCK-FREE -komento on BLOCK-ALLOCATE -komennon vastakohta. Tällä käskyllä vapautetaan varattu, mutta tarpeettomaksi käynyt lohko muuhun käyttöön. Tämä komento ei toimi aivan samalla tavalla kuin SCRATCH -komento tiedostojen yhteydessä, koska levyltä ei poisteta tietoa tällä komennolla. BLOCK-FREE poistaa vain yhden merkinnän — tässä tapauksessa lohkokartasta.

BLOCK-FREE -KOMENNON MUOTO:

PRINT # tiedostoNo,"BLOCK-FREE:" levy-yksikkö;ura;lohko
tai lyhyemmin:
PRINT # tiedostoNo,"B-F:" levy-yksikkö;ura;lohko

Hajatiedostojen käyttö

Tähän saakka ainoa hajatiedostojen yhteydessä esillä ollut ongelma on se, että levyn käytettyjä lohkoja ei voida kirjata ylös. Lohkokartalle ei kuitenkaan voida merkitä lohkoa käytetyksi jonkin toisen lohkon avulla. Lohkokartalle ei myöskään voida merkitä, onko lohkossa hajatiedosto, ohjelman osa vai peräkkäis- tai suhteellinen tiedosto.

Käytetyistä lohkoista pidetään yleensä kirjaa kuhunkin hajatiedostoon liitettävän peräkkäistiedoston avulla. Tähän tiedostoon merkitään luettelo tietueista, urista ja lohkojen paikoista. Tällä menetelmällä on jokaista hajatiedostoa varten avattava kolme kanavaa levyasemalle: yksi komentokanava, yksi hajatiedostolle ja yksi peräkkäistiedostolle. Tämän vuoksi täytetään kaksi puskuria samaan aikaan.

Esimerkkiohjelma, joka kirjoittaa 10 hajatietuetta peräkkäistiedoston avulla:

```
10   OPEN 15,8,15
20   OPEN 5,8,5,"#"
30   OPEN 4,8,4,"@ 0:AVAIN,S,W"   ! @ kirjoittaa päälle
40   A$="TIETUE"
50   FOR R=1 TO 10
70   PRINT # 5,A$+",";R
90   U=1: S=1                     ! Uusea s=sektori
100  PRINT # 15,"B-A:"0;U;S       ! lohkon varaus ura 1 ; lohko 1
110  INPUT # 15,A,B$,C,D          ! lukee virheilmoituksen
120  IF A=65 THEN U=C: S=D: GOTO 100   ! 65 lohkovaratty
130  PRINT # 15,"B-W:"5;0;U;S
140  PRINT # 4,U: PRINT # 4,S
150  NEXT R
160  CLOSE 4: CLOSE 5: CLOSE 15
```

Esimerkkiohjelma, joka lukee 10 hajatietuetta peräkkäistiedoston avulla:

```
10   OPEN 15,8,15
20   OPEN 5,8,5,"#"
30   OPEN 4,8,4,"AVAIN,S,R"
40   FOR R=1 TO 10
50   INPUT # 4,U,S
60   PRINT # 15,"B-R:"5;0;U;S
70   INPUT # 5,A$,X
80   IF A$<>"TIETUE" OR X<>R THEN STOP
90   PRINT # 15,"B-F:"0;U;S
100  NEXT R
110  CLOSE 4: CLOSE 5
120  PRINT # 15,"S0:AVAIN"
130  CLOSE 15
```

Seuraava ura ja sektori

Tiedon tarkistus

BUFFER-POINTER

Puskurin osoittimella (Buffer Pointer) merkitään, mihin edellinen tieto on tallennettu. Se ilmoittaa myös, mihin seuraava tieto tallennetaan. Lohkossa olevat yksittäiset tavut voidaan hakea halutussa järjestyksessä muuttamalla puskurin osoittimen paikkaa. Tällä tavalla lohko voidaan jakaa edelleen tietueisiin.

Ottakaamme esimerkiksi postituslista: Jos tietojen (nimen, osoitteen, jne.) tarvitsema merkkimäärä on korkeintaan 64 merkkiä, voimme jakaa kunkin lohkon 4 tietueeseen. Kun tiedämme kyseisen uran ja sektorin, voimme etsiä yksittäisessä tietueessa olevan tiedon.

BUFFER-POINTER -KOMENNON MUOTO:

PRINT # tiedostoNo, "BUFFER-POINTER:" kanava,sijainti
tai lyhyemmin:
PRINT # tiedostoNo, "B-P:" kanava,sijainti

ESIMERKKI, JOSSA OSOITIN ASETETAAN PUSKURIN 64:NTEEN MERKKIIN:

PRINT # tiedostoNo,"B-P:"5,64

Seuraavat on esimerkkiohjelmat kirjoittavat ja lukevat hajatiedostoja, joissa lohkot on jaettu edelleen tietueisiin.

Esimerkkiohjelma, joka kirjoittaa 10 hajalohkoa ja kuhunkin 4 tietuetta:

```
10   OPEN 15,8,15
20   OPEN 5,8,5,"#"
30   OPEN 4,8,4,"@ :AVAIN,S,W"
40   A$="TIETUE"
50   FOR R=1 TO 10
60   FOR L=1 TO4
70   PRINT # 15,"B-P:"5;(L-1)*64+1
80   PRINT # 5,A$","L
90   NEXT L
100  U=1: S=1
110  PRINT # 15,"B-A:"0;U;S
120  INPUT # 15,A,B$,C,D
130  IF A = 65 THEN U=C: S=D: GOTO 110
140  PRINT # 15,"B-W:"5;0;U;S
150  PRINT # 4,U","S
160  NEXT R
170  CLOSE4: CLOSE5: CLOSE15
```

Sijoitus paikkaan 1, 65, 129 ja 193.

Etsitään vapaa ura ja sektori

Esimerkkiohjelma, joka lukee takaisin 10 hajalohkoa, joissa kussakin on 4 tietuetta:

```
10   OPEN 15,8,15
20   OPEN 5,8,5,"#"
30   OPEN 4,8,4,"AVAIN,S,R,"
40   FOR R=1 TO 10
50   INPUT # 4,U,S
60   PRINT # 15,"B-R:"5;0;U;S
```

```
70   FOR L=1 TO 4
80   PRINT# 15,"B-P:"5;(L-1)*64+1
90   INPUT# 5,A$,X
100  IF A$<>"TIETUE" OR X<>L THEN PRINT A$,X: STOP
110  PRINT A$,X
120  NEXT L
130  NEXT R
140  CLOSE4: CLOSE5: CLOSE15
```

USER1 ja USER2

USER-käskyt (=käyttäjän käskyt) on suunniteltu konekielistä ohjelmointia varten (katso lisää seuraavasta kappaleesta). USER1 ja USER2 ovat BLOCK-READ- ja BLOCK-WRITE -komentojen erikoismuotoja. Näissä käskyissä on yksi tärkeä ero: USER1:n ja USER2:n suhtautuminen puskurin osoittimeen.

BLOCK-READ -komento lukee lohkoa vain puskurin osoittimen ilmoittamaan lohkon päättymismerkkiin saakka. USER1 -komento toimii kuten BLOCK-READ, mutta pakottaa osoittimen paikkaan 255 ja lukee kaikki lohkoon kirjoitetut tiedot.

USER1-KOMENNON MUOTO:

```
PRINT# tiedostoNo,"U1:" kanava,levy-yksikkö,ura,lohko
```

tai:

```
PRINT# tiedostoNo,"UA:" kanava,levy-yksikkö,ura,lohko
```

Tämä komento toimii täysin samoin sekä U1- että UA-muodossa.

BLOCK-WRITE -komento kirjoittaa puskurissa olevat tiedot levyn lohkoon, joka määräytyy puskurin osoittimen mukaan. USER2 -komento kirjoittaa puskurissa olevan tiedon lohkoon muuttamatta lohkossa olevan puskurin osoittimen arvoa. Tämä käsky on hyödyllinen, jos lohko halutaan lukea BLOCK-READ -komennolla sen jälkeen, kun se on päivitetty BUFFER-POINTER- ja PRINT#-komentojen avulla. Päivitetty tieto tallennetaan takaisin levylle USER2-komennolla.

USER2-KOMENNON MUOTO:

```
PRINT# tiedostoNo,"U2:" kanava,levy-yksikkö,ura,lohko
```

tai:

```
PRINT# tiedostoNo,"UB:" kanava,levy-yksikkö,ura,lohko
```

Tämän komennon käytöstä on esimerkkiohjelma liitteessä C.

7. SUHTEELLISET TIEDOSTOT

Suhteellisen tiedoston avulla voidaan levyltä etsiä vaivattomasti halutut tiedot. Sen avulla tietojen käsittely on mukavaa, koska tiedostot voidaan jakaa tietueisiin ja tietueet edelleen kenttiin.

DOS huolehtii urista ja sektoreista ja sallii jopa tietueiden jatkumisen lohkosta toiseen. Tämä on mahdollista, koska DOS perustaa sivusektoreita, sarjan osoittimia kunkin tietueen alkamista varten. Kukin sivusektori voi osoittaa jopa 120 tietueeseen ja tiedostossa voi olla 6 sivusektoria. Tiedosto voi sisältää 720 tietuetta ja kussakin tietueessa voi olla 254 merkkiä, joten tiedosto voi olla kokonaisen levyn kokoinen.

Suhteellisen tiedoston perustaminen

Suhteellinen tiedosto perustetaan OPEN -komennolla. Tiedostoa ei voida korvata uudella samannimisellä tiedostolla (@ -merkin avulla). Tiedostoa voidaan laajentaa, lukea ja siihen voidaan kirjoittaa lisää tietoja.

SUHTEELLISEN TIEDOSTON PERUSTAMISKÄSKYN MUOTO:

```
OPENtiedostoNo,laiteNo,kanavaNo,"nimi,L,"+CHR$(tietuepituus)
```

ESIMERKKEJÄ SUHTEELLISEN TIEDOSTON PERUSTAMISESTA:

```
OPEN 2,8,2,"TIEDOSTO,L,"+ CHR$(100)
OPEN F,8,F,A$+",L,"+ CHR$(Q)
OPEN A,B,C,"TESTI,L,"+CHR$(33)
```

Tietueen Pituus

TAULUKKO 7.1 Suhteellisen tiedoston rakenne

TIETOLOHKO	
TAVU	MÄÄRITELMÄ
0,1	Seuraavan tietolohkon ura ja sektori
2-256	254 tavua tietoa. Tyhjien tietueiden ensimmäisessä tavussa on FF ja muissa tavuissa on 00 (kaikki binäärimuodossa). Osittain käytetyt tietueet täytetään nollilla (00).
SIVUSEKTORILOHKO	
TAVU	MÄÄRITELMÄ
0,1	Seuraavan sivusektorilohkon ura ja sektori.
2	Sivusektorin numero. (0-5)
3	Tietueen pituus.
4,5	Ensimmäisen sivusektorin (No. 0:n) ura ja sektori.
6,7	Toisen sivusektorin (No. 1:n) ura ja sektori.
8,9	Kolmannen sivusektorin (No. 2:n) ura ja sektori.
10,11	Neljännen sivusektorin (No. 3:n) ura ja sektori.
12,13	Viidennen sivusektorin (No. 4:n) ura ja sektori.
14,15	Kuudennen sivusektorin (No. 5:n) ura ja sektori.
16-256	120 tietolohkon osoittimien urat ja sektorit.

Kun OPEN -komento annetaan, levynkäyttöjärjestelmä tarkistaa ensin, onko tiedosto perustettu aiemmin. Mitään ei tapahdu, jos näin on. Vanhan suhteellisen tiedoston voi poistaa vain SCRATCH-komennolla (kts. luku4). Vanhaa ei voida korvata □□ -rakenteen avulla.

Suhteellisten tiedostojen käyttö

Aiemmin perustetun suhteellisen tiedoston avauskäsky on yksinkertaisempi.

AIEMMIN PERUSTETUN SUHTEELLISEN TIEDOSTON AVAAMISKÄSKY:

OPEN tiedostoNo,laiteNo,kanavaNo,"nimi"

Tämä komento kertoo DOS:lle, että kysymyksessä on suhteellinen tiedosto. Sekä tällä että edellisessä kappaleessa kerrotulla käskymuodolla voidaan tiedostoa sekä lukea että kirjoittaa.

Kun luet tai kirjoitat tietoja, on osoitin sijoitettava oikeaan tietueen paikkaan **ennen kuin teet mitään.**

SIJOITUS-KOMENNON (POSITION) MUOTO:

PRINT # tiedNo,"P"CHR$(kanavaNo)CHR$(tietueNoLO)CHR$(tietueNoHI)

tai

PRINT # tiedNo,"P"CHR$(kanavaNo)CHR$(tietueNoLO)CHR$(tietueNoHI) CHR$(sijainti)

ESIMERKKEJÄ SIJOITUS-KOMENNOSTA:

PRINT # 15,"P" CHR$(2) CHR$(1) CHR$(0)

PRINT # 15,"P" CHR$(KA) CHR$(T1) CHR$(T2)

PRINT # 15,"P" CHR$(4)CHR$(T1)CHR$(T2)CHR$(S)

KanavaNo
TietueNo
Sijainti tietueessa

Tietueen numeroon tarvitaan kaksitavuinen luku, koska yhteen tavuun voidaan kirjoittaa vain 256 lukua ja tiedostoon voidaan kirjoittaa yli 700 tietuetta. TietueNo LO on osoitteen vähemmän merkitsevä tavu ja tietueNo HI on osoitteen enemmän merkitsevä tavu. Tästä lasketaan varsinainen tietueen numero kaavalla:
TIETUENo = TIETUENo HI * 256 + TIETUENo LO.

Olettakaamme, että postituslistassa on alla olevan taulukon mukaiset 7 tietuetta.

KENTÄN NIMI	PITUUS	KENTÄN NIMI	PITUUS
1. Etunimi	10	6. Lääni	2
2. Sukunimi	15	7. Sotu-koodi	11
3. Osoiterivi1	20	8. Puhelin	10
4. Osoiterivi2	20		
5. Kaupunki	12	YHTEENSÄ	100

Tietueen pituus lasketaan näin. Kuhunkin kenttään kannattaa varata yksi ylimääräinen merkki. Muuten INPUT # -komento lukee paljon pidempiä tiedoston osia kuin tarvitaan, kuten peräkkäistiedostoistakin. Perustamme siis 108 merkin pituisen tietueen. Numero 1 edustaa tähän saakka suurinta perustettua tietuenumeroa. Alla oleva ohjelma suorittaa tähän mennessä esitetyt tehtävät.

tietuepituus

```
10   OPEN 1,8,15
20   OPEN 2,8,3,"0:POSTILISTA,L," + CHR$(108)
30   GOSUB 900
40   PRINT # 1,"P"CHR$(3)CHR$(1)CHR$(0)CHR$(1)
50   GOSUB 900
60   IF A=50 THEN PRINT # 2,1:GOTO 40
70   INPUT # 2,X
300  STOP
900  INPUT # 1,A,B$,C,D
910  IF (A=50) OR (A < 20) THEN RETURN
920  PRINT A;B$;C;D:STOP: RETURN
```

Ohjelman rivillä 60 tarkistettu virhenumero 50 on RECORD NOT PRESENT (tietuetta ei ole saatavilla) -virhe. Tämä merkitsee, että tietuetta ei ole perustettu. Tällainen virhetilanne on otettava ohjelmassa huomioon.

Tähän saakka suorittamamme toimenpiteet perustavat tiedoston ja ensimmäisen tietueen, mutteivät kirjoita sinne tietoja. Alla oleva ohjelma on edellisen ohjelman laajennettu versio. Tämän ohjelman avulla voidaan työskennellä postituslistojen kanssa. Tietueet on koodattu numeroilla.

POSTITUSLISTAN LUKU- JA KIRJOITUSOHJELMA:

```
5    A(1)=12:A(2)=15:A(3)=20:A(4)=19:A(5)=12:A(6)=3:A(7)=11:A(8)=10
10   OPEN1,8,15:OPEN2,8,3,"0:POSTILISTA,L,"+CHR$(108):GOSUB900
20   PRINT# 1,"P"CHR$(3)CHR$1)CHR$(0)CHR$(1):INPUT# 2,X
30   PRINT"LUKU, KIRJOITUS VAI PÄÄTTÄMINEN":INPUTJ$
35   IFJ$="P"THENCLOSE2:CLOSE1:END
40   IFJ$="K"THEN200
50   INPUT"TIETUE N:O";R:IFR<0ORR>XTHEN50
60   IFR<2THEN30
70   R1=R:R2=0:IFR1>256THENR2=INT(R1/256)
75   R1=R1-256*R2
80   RESTORE:DATA 1,ETUNIMI,12,SUKUNIMI,28,OSOITE1,49,OSOITE2,70
90   DATA KAUPUNKI,82,LÄÄNI,86,SOTU,98,PUH.
100  FORL=1TO8:READA,A$:PRINT#1,"P"CHR$(3)CHR$(R1)CHR$(R2)CHR$(A)
105  GOSUB900
110  ONA/50GOTO50:INPUT#2,Z$:PRINTA$,Z$:NEXT:GOTO50
200  PRINT:INPUT"TIETUE N:O";R:IFR<0ORR>5000THEN200
210  IFR<2THEN30
215  IFR>XTHENR=X+1:PRINT:PRINT"TIETUE N:O";R
220  R1=R:R2=0:IFR1>256THENR2=INT(R1/256)
225  R1=R1-256*R2
230  RESTORE:FORL=1TO8:READA,A$
235  PRINT#1,"P"CHR$(3)CHR$(R1)CHR$(R2)CHR$(A)
240  PRINTA$:INPUTZ$:IFLEN(Z$)>A(L)THENZ$=LEFT$(Z$,A(L))
245  PRINT#2,Z$:NEXT:IFX<RTHENX=R
247  PRINT#1,"P"CHR$(3)CHR$(1)CHR$(0)CHR$(1)
250  PRINT#*,X:GOTO200
900  INPUT#1,A,B$,C,D:IFA<20THENRETURN
910  IFA<>50THENPRINTA;B$;C;D:STOP:RETURN
920  IFJ$="L"THENPRINTB$:RETURN
```

Tämä ohjelma etsii tietueen sen numeron perusteella. Tämä ohjelma ei kirjoita EOF:n jälkeen. Jos näin yritetään tehdä, ohjelma pakottaa käyttäjän kirjoittamaan tiedot seuraavaan tietueeseen. Ohjelman kehittyneempi versio etsisi tietoyksiköt tietynlaisten tietueita edustavien "avainten" avulla. Jos haluat esimerkiksi etsiä tietueen nimen perusteella tai tulostaa tunnukset SOTU-koodin mukaan, tarvitaan erillinen lista, josta saadaan avainkoodit ja vastaavat tietuenumerot. Tämä voitaisiin tallentaa peräkkäistiedostoon.

Työskenneltäessä sellaisten tiedostojen kanssa, joiden tiedetään kasvavan suuriksi, kannattaa perustaa tiedosto, jonka kapasiteetti riittää tulevaan tarpeeseen. Toisin sanoen: Jos perustat tiedoston, jonka oletat tulevan 1000 tietueen pituiseksi, anna tietuenumeroksi 1000 perustaessasi tiedostoa. Tällä tavalla DOS perustaa välitiedostoja, jotka tekevät myöhemmän käytön paljon nopeammaksi.

ESIMERKKI SUUREN TIEDOSTON PERUSTAMISESTA:

```
OPEN 15,8,15:OPEN2,8,2,"REL,L,"+CHR$(60)
PRINT# 1,"P" CHR$(2)CHR$(0)CHR$(4)CHR$(1)
PRINT# 2,"LOPPU"
CLOSE 2: CLOSE1
```

Tietue N:o
4*256+0 eli 1024

8. LEVYOHJAIMEN OHJELMOINTI

Todella hyvät ohjelmoijat voivat suunnitella rutiineja, jotka sijaitsevat ja toimivat levyohjaimessa. Levyltä tulevat DOS-rutiinit voidaan myös lisätä ohjelmaan. Rutiinit lisätään samaan tyyliin kuin DOS tukiohjelma lisätään tietokoneen muistiin.

BLOCK-EXECUTE

Tämä komento lataa levyltä konekielisen rutiinin ja suorittaa sitä kunnes saavuttaa RTS (ReTurn from Subroutine) komennon.

BLOCK-EXECUTE -KOMENNON MUOTO:

```
PRINT# tiedostoNo, "BLOCK-EXECUTE:"kanava,yksikkö,ura,lohko
```

tai lyhyemmin:

```
PRINT# tiedostoNo, "B-E:"kanava,levy-yksikkö,ura,lohko
```

MEMORY-READ

Levyasemassa on 16 kt ROM- ja 2kt RAM-muistia. Näihin tai DOS:n perustamiin puskureihin päästään suoraan MEMORY -komennoilla. MEMORY-READ-käskyllä voidaan valita **virhekanavan kautta** luettava tavu.

MEMORY READ-KOMENNON MUOTO:

```
PRINT# tiedostoNo,"M-R:"CHR$(osoitteen vähemm. merk. tavu)
CHR$(enemmän merk. tavu) (ei lyhenteitä)
```

Tavu voidaan lukea tietokoneen muistiin GET# -komennolla kanavalta 15.

Jos virhekanavalle annetaan INPUT# -komento, on vastaus järjetön. INPUT# -komento voidaan peruuttaa millä hyvänsä muulla levykomennolla (MEMORY-komentoja lukuunottamatta).

OHJELMA, JOKA LUKEE LEVYOHJAIMEN MUISTIA:

```
10  OPEN15,8,15
20  INPUT"ANNA PAIKKA";A
30  A1=INT(A/256):A2=A-A1*256
40  PRINT#15,"M-R:"CHR$(A2)CHR$(A1)
50  FOR L=1 TO 5
60  GET# 15,A$
70  PRINT ASC(A$+CHR$(0));
80  NEXT
90  INPUT"HALUATKO JATKAA";A$
100 IF LEFT(A$,1)="K" THEN 50
110 GOTO 20
```

MEMORY-WRITE

MEMORY-WRITE -komennolla voidaan levyohjaimen muistiin kirjoittaa 34 tavua kerralla. Tätä koodia voidaan 'ajaa' MEMORY-EXECUTE- ja USER -komennoilla.

MEMORY-WRITE -KOMENNON MUOTO:

```
PRINT # tiedostoNo,"M-W:"CHR$(osoitteen LO-tavu)
CHR$(osoitteen HI-tavu)merkkien lkm; datatavu
```

OHJELMA, JOKA KIRJOITTAA LEVYLLE RTS:N:

```
10  OPEN 15,8,15,"M-W:"CHR$(0)CHR$(5);1;CHR$(96)
20  PRINT # 15, "M-E:"CHR$(0)CHR$(19):REM HYPPÄÄ TAVUUN JA PALAA
    TAKAISIN
30  CLOSE 15
```

MEMORY EXECUTE

Mikä tahansa DOS-muistissa (RAM:ssa tai ROM:ssa) oleva rutiini voidaan suorittaa MEMORY-EXECUTE -komennolla.

MEMORY-EXECUTE -KOMENNON MUOTO:

```
PRINT # tied.No,"M-E:"CHR$(osoitteen LO-tavu)CHR(os:n HI-tavu)
```

Esimerkki on edellisen ohjelman rivillä 20.

USER-komennot

Kappaleessa 6 esitettyjen USER1- ja USER2-komentojen sekä kappaleessa 2 esitettyjen UI+ ja UI- komentojen lisäksi on USER-komentoja, joilla voidaan suorittaa hyppyjä levyaseman RAM-muistissa.

USER-KOMENTO	TOIMINTA
U1 tai UA	BLOCK-READ puskurin osoitinta siirtämättä.
U2 tai UB	BLOCK-WRITE puskurin osoitinta siirtämättä.
U3 tai UC	Hyppy paikkaan $0500
U4 tai UD	Hyppy paikkaan $0503
U5 tai UE	Hyppy paikkaan $0506
U6 tai UF	Hyppy paikkaan $0509
U7 tai UG	Hyppy paikkaan $050C
U8 tai UH	Hyppy paikkaan $050F
U9 tai Ui	Hyppy paikkaan $FFFA
U; tai UJ	Käynnistysvektori
UI+	Commodore-64:n nopeus
UI-	VIC-20:n nopeus

Lataamalla nämä paikat johonkin toiseen hyppykäskyyn, kuten JMP $0520, voit helposti muodostaa pidempiä levyaseman muistipaikkoja käsitteleviä rutiineja helppokäyttöisen taulukon avulla — jopa BASIC-ohjelmiin!

ESIMERKKEJÄ USER -KOMENNOISTA:

```
PRINT # 15,"U3"
PRINT # 15,"U" + CHR$(50 + Q)
PRINT # 15,"UI"
```

9. LEVYASEMAN LAITENUMERON VAIHTAMINEN

Ohjelmallinen menetelmä

Levyasema valitsee numeronsa tarkistamalla piirilevyn ja kirjoittamalla sen mukaan numeron RAM-muistin määrättyyn paikkaan. Koska operaatio on ohjelmarutiini, on vanha numero helppo korvata uudella.

LAITENUMERON VAIHTAMISKOMENTO:

```
PRINT # tiedostoNo,"M-W:" CHR$(119) CHR$(0) CHR$(2)
CHR$(osoite + 32) CHR$(osoite + 64)
```

ESIMERKKI LAITENUMERON VAIHTAMISESTA:

```
PRINT # 15,"M-W:" CHR$(119) CHR$(0) CHR$(2) CHR$(9 + 32) CHR$(9 + 64)
PRINT # Q,"M-W:" CHR$(119) CHR$(0) CHR$(2) CHR$(R + 32) CHR$(R + 64)
```

Jos käytät useampaa kuin yhtä asemaa, kannattaa aseman numero vaihtaa piirilevylle (kts. alla). Se on helppoa. Kytke asemat yksi kerrallaan ja tee tarvittava numeron muutos. Näin käytössä ei tule hankaluuksia.

Laitteistomenetelmä

Jos käytät useita levyasemia yhtaikaa, kannattaa aseman numero vaihtaa pysyvästi. Tarvitset vain tähtimeisselin ja puukon.

LAITENUMERON VAIHTAMINEN LAITTEISTOON:

1. Irroita levyasemasta kaikki kaapelit, myös virtakaapeli.
2. Käännä levyasema ylösalaisin pöydälle.
3. Poista levyaseman kotelon neljä kiinnitysruuvia.
4. Käännä asema varovasti oikein päin ja poista kotelon yläosa.
5. Etsi hyppyjohtimet.*)
6. Leikkaa joko johdin 1 tai 2 tai molemmat johtimet.
7. Laita kansi ja neljä ruuvia takaisin.
8. Kytke kaapelit takaisin.

Kun hyppyjohdin katkaistaan, sen numero lisätään laitenumeroon (8). Siis: hyppyjohdin 1 lisää luvun 1, hyppyjohdin 2 lisää luvun 2. Jos kumpikin ovat ehyitä, on laitenumero 8; jos 1 on poikki, on laitenumero 9; jos 2 on poikki, on laitenumero 10; jos molemmat ovat poikki, on laitenumero 11.

*) Hyppyjohdin on piirilevyssä oleva -kuvio. Katso levyasemaa edestäpäin. Jos hyppyjohtimet löytyvät piirilevyn vasemmalta laidalta hieman puolivälin takaa, ne on numeroitu. Jos hyppyjohtimet ovat piirilevyn vasemmasta etukulmasta mitattaessa 4 cm taakse- ja 6 cm oikeallepäin, on levyaseman etupaneelia lähempänä olevan hyppyjohtimen numero 1.

LIITE A:

Yhtenveto levykomennoista

Yleinen muoto: PRINT # tiedostoNo, komento

KOMENTO	MUOTO
NEW	"N0:nimi,id" (levyn formatoimiskäsky)
COPY	"C0:uusi tiedosto = 0:alkuperäinen tiedosto"
RENAME	"R0:uusi nimi = 0:vanha nimi"
SCRATCH	"S0:tiedoston nimi" (tiedoston poistaminen)
INITIALIZE	"I" (alustus)
VALIDATE	"V" (tiedostojen poistaminen)
DUPLICATE	ei toimi yhden levyn asemissa
BLOCK-READ	"B-R:"kanava;asema;ura;lohko
BLOCK-WRITE	"B-W:"kanava;asema;ura;lohko
BLOCK-ALLOCATE	"B-A:"asema;ura;lohko
BLOCK-FREE	"B-F:"asema;ura;lohko
BUFFER-POINTER	"B-P:"kanava;paikka
USER1 ja USER2	"Un:"kanava;asema;ura;lohko
POSITION	"P" CHR$(kanava) CHR$(tietueLO)CHR$(tietueHI) CHR$(paikka)
BLOCK-EXECUTE	"B-E:"kanava;asema;ura;lohko
MEMORY-READ	"M-R:" CHR$(os.LO) CHR$(os.HI)
MEMORY-WRITE	"M-W:"CHR$(os.LO) CHR$(os.HI) CHR$(merkkien lkm.) "tiedot"
MEMORY-EXECUTE	"M-E:"CHR$(os.LO) CHR$(os.HI)
USER -komennot	"Un:"

LIITE B

Yhteenveto CBM levyaseman virheilmoituksista

0	OK, ei virhettä.
1	Poistetut tiedostot. Ei ole virhetilanne.
2-19	Käyttämättömiä, näitä ei pitäisi esiintyä.
20	Lohkon otsikkoa ei löydy levyltä.
21	Tahdistusmerkkiä ei löydy.
22	Tietolohkoa ei löydy.
23	Tietojen tarkistussumma virheellinen.
24	Tavun dekoodausvirhe.
25	Tallennuksen tarkistus -virhe.
26	On yritetty tallentaa kirjoitussuojatulle levylle.
27	Otsikon tarkistussumma virheellinen.
28	Tieto jatkuu seuraavaan lohkoon.
29	Epäsopiva levyn tunniste.
30	Yleisrakenne virheellinen.
31	Viallinen komento.
32	Liian pitkä rivi.
33	Sopimaton tiedoston nimi.
34	Ei ole annettu tiedostoa.
39	Komentotiedostoa ei löytydy.
50	Tietuetta ei ole.
51	Tietueen tila ylitetty.
52	Tiedosto liian suuri.
60	Tiedosto avattu kirjoitusta varten.
61	Tiedostoa ei ole avattu.
62	Tiedostoa ei löydy.
63	Tiedosto on jo olemassa.
64	Tiedostotyyppi epäsopiva.
65	Lohko varattu.
66	Sopimaton ura tai sektori.
67	Sopimaton systeemiura tai -sektori.
70	Kanava varattu.
71	Hakemistovirhe.
72	Levy tai hakemisto täynnä.
73	Virran kytkemisviesti tai DOS -sekaannus yritettäessä tallentaa.
74	Levyasema ei ole valmis. (vain 8050:lle)

DOS-VIRHEILMOITUKSET

HUOM: Numeroa 20 pienemmät virheilmoitukset voidaan jättää huomiotta numeroa 1 lukuunottamatta. Ilmoitus 1 kertoo, montako tiedostoa on poistettu SCRATCH-käskyllä.

20: READ ERROR (Lohkon otsikkoa ei löydy)
Levyohjain ei löydä halutun tietolohkon otsikkoa, koska sektorinumero on virheellinen tai otsikko on tuhoutunut.

21: READ ERROR (Tahdistusmerkkiä ei löydy)
Levyohjain ei löydä tahdistusmerkkiä halutulta uralta. Syynä on luku/kirjoituspään kohdistusvirhe, levy puuttuu, on laitettu huonosti asemaan, tai on formatoimaton. Tämä virheilmoitus voi aiheutua myös laitteistoviasta.

22: READ ERROR (Tietolohkoa ei ole)
Levyohjainta on pyydetty lukemaan tai tutkimaan tietolohkoa, johon ei ole tallennettu tietoja kunnolla. Tämä virheilmoitus esiintyy BLOCK-käskyjen yhteydessä ja johtuu virheellisestä ura- ja/tai sektoripyynnöstä.

23: READ ERROR (Tietolohkon tarkistussumma virheellinen)
Tämä virheilmoitus kertoo, että yhdessä tai useammassa tietotavussa on virhe. Tieto on luettu DOS:n muistiin, mutta tiedon tarkistussumma on virheellinen. Ilmoitus voi myöskin johtua huonosta maadoituksesta.

24: READ ERROR (Tavun dekoodausvirhe)
Tieto tai otsikko on luettu DOS:n muistiin, mutta laitteistovirhe aiheutuu tietotavun virheellisestä bittiyhdistelmästä. Ilmoitus voi myöskin johtua huonosta maadoituksesta.

25: WRITE ERROR (Tallennuksen tarkistusvirhe)
Tämä ilmoitus annetaan, kun ohjain havaitsee eroja kirjoitetun tiedon ja DOS:n muistissa olevan tiedon välillä.

26: WRITE PROTECT ON (Kirjoitussuoja päällä)
Tämä ilmoitus johtuu siitä, että ohjaimella yritetään tallentaa tietolohko, kun levyaseman kirjoitussuojakytkin on alhaalla. Yleensä syynä on se, että tietoja on yritetty tallentaa kirjoitussuojatulle levylle.

27: READ ERROR (Otsikon tarkistussumma virheellinen)
Ohjain on havainnut virheen tietolohkon otsikossa. Lohkoa ei ole luettu DOS:n muistiin. Ilmoitus voi myös johtua huonosta maadoituksesta.

28: WRITE ERROR (Liian pitkä tietolohko)
Ohjain yrittää löytää tietolohkon kirjoittamisen jälkeen seuraavasta otsikosta tahdistusmerkin. Jos tahdistusmerkkiä ei löydy määrätyssä ajassa, annetaan virheilmoitus. Virheilmoitus johtuu huonosta levyn rakenteesta (tieto jatkuu seuraavaan lohkoon) tai laitteistoviasta.

29: DISK ID MISMATCH (Levyn tunnus viallinen)
Tämä ilmoitus annetaan, kun ohjainta on pyydetty suorittamaan toimintoja alustamattomalle levylle. Ilmoitus voi johtua myös levyn virheellisestä otsikosta.

30: SYNTAX ERROR (Yleisrakenne virheellinen)
DOS ei voi suorittaa komentokanavalle lähetettyä käskyä. Yleensä ilmoitus johtuu annettujen tiedostonimien väärästä lukumäärästä tai merkkien väärästä käytöstä. Esim. COPY-käskyn vasemmalla puolella voi olla kaksi tiedostonimeä.

31: SYNTAX ERROR (Viallinen käsky)
DOS ei tunne käskyä. Käsky on aloitettava ensimmäisestä merkin paikasta.

32: SYNTAX ERROR (Liian pitkä rivi)
Lähetetty käsky on pidempi kuin 58 merkkiä.

33: SYNTAX ERROR (Viallinen tiedoston nimi)
OPEN- tai SAVE-käskyjen kanssa on käytetty *-tai ?-merkkejä virheellisesti.

34: SYNTAX ERROR (Tiedosto puuttuu)
Tiedoston nimi jäi käskystä pois tai DOS ei tunnista sitä. Yleinen virhe: Kaksoispiste (:) on jäänyt pois.

39: SYNTAX ERROR (Viallinen käsky)
Tämä virhe voi johtua siitä, että DOS ei tunnista komentokanavalle (sekundääriosoite 15) lähetettyä käskyä.

50: RECORD NOT PRESENT (Tietue puuttuu)
INPUT#- tai GET# -komennoilla yritetään lukea viimeisen tietueen jälkeen olevia levyn lohkoja. Tämä ilmoitus voi johtua myös siitä, että suhteellisessa tiedostossa sijoitetaan tietue tiedoston päättymismerkin jälkeen. Jos tiedostoa halutaan laajentaa lisäämällä siihen uusi tietue (PRINT# -käskyllä), virheilmoitusta ei tarvitse ottaa huomioon. INPUT#- tai GET# -käskyä ei saa käyttää tämän virheilmoituksen jälkeen ennen kuin tiedoston osoittimen arvo on muutettu.

51: OVERFLOW IN RECORD (Tietueen tila ylitetty)
PRINT#-lause ylittää tietuerajan. Tieto katkaistaan. Koska tietueen keskeytysmerkiksi lähetettävä kirjoituspään palautus luetaan tietueen pituuteen mukaan, annetaan tämä virheilmoitus, jos tietueen pituus +1 ylittää tietueelle varatun tilan.

52: FILE TOO LARGE (Tiedosto liian suuri)
Suhteellisen tiedoston tietueen paikka on niin suuri, että levyn kapasiteetti ylitetään.

60: WRITE FILE OPEN (Kirjoitustiedosto auki)
Tämä ilmoitus annetaan, jos yritetään avata kirjoitustiedosto, jota ei ole suljettu.

61: FILE NOT OPEN (Tiedosto ei ole auki)
Tämä ilmoitus syntyy, kun yritetään käyttää tiedostoa, jota ei ole avattu DOS:issa. Joskus virheilmoitusta ei anneta, vaan viesti jätetään vain huomioimatta.

62: FILE NOT FOUND (Tiedostoa ei löydy)
Etsittyä tiedostoa ei ole levyllä.

63: FILE EXISTS (Tiedosto on jo olemassa)
Luodun tiedoston nimi on jo levyllä.

64: FILE TYPE MISMATCH (Virheellinen tiedostotyyppi)
Tiedostotyyppi on eri kuin hakemistossa olevan samannimisen tiedoston tyyppi.

65: NO BLOCK (Lohko on jo varattu)
Tämä virheilmoitus esiintyy B-A -käskyn yhteydessä. Se kertoo, että lohko, jota yritettiin varata, on varattu jo aiemmin. Parametrit kertovat seuraavan vapaan lohkon ura- ja sektorinumerot. Jos parametreina on nollat (0), kaikki jäljempänä olevat lohkot ovat käytössä.

66: ILLEGAL TRACK AND SECTOR (Väärä ura ja sektori)
DOS on yrittänyt hakea uraa tai sektoria, joka ei kuulu levyn rakenteeseen. Tämä ilmoitus voi johtua myös siitä, että seuraavan lohkon lukeminen ei onnistu.

67: ILLEGAL SYSTEM T OR S (Luvaton ura tai -sektori)
Tämä erikoisviesti johtuu väärästä urasta tai sektorista.

70: NO CHANNEL (Kanavaa ei voida käyttää)
Haluttu kanava tai kaikki kanavat ovat käytössä. Kerralla DOS:iin voidaan avata korkeintaan viisi peräkkäistiedostoa. Suorasaantikanavien kautta voidaan avata kuusi tiedostoa.

71: DIRECTORY ERROR (Hakemistovirhe)
Lohkokartta (BAM) ja sisäinen laskuri eivät vastaa toisiaan. BAM:in tilanvaraus ei toimi tai DOS:n muistista on kirjoitettu jotain BAM:n päälle. Tämä virhe korjataan alustamalla levy uudestaan. Näin BAM palautetaan muistiin. Korjaustoimenpide saattaa vaikuttaa eräisiin sulkemattomiin tiedostoihin.

72: DISK FULL (Levy on täynnä)
Levyn kaikki lohkot on käytetty, tai hakemistossa on 144 tiedostoa. Tämä ilmoitus lähetetään, kun levyllä on vielä kaksi lohkoa, jotta käytössä oleva tiedosto voidaan sulkea.

73: DOS MISMATCH (DOS-sekaannus) (73, CBM DOS V2.6 1541)
DOS 1 ja DOS 2 ovat valmiita lukemaan, mutteivät kirjoittamaan. Levyjä voidaan lukea kummallakin DOS:lla, mutta toisella versiolla formatoidulle levylle ei voida kirjoittaa toisen version avulla, koska rakenne on erilainen. Tämä virheilmoitus saadaan aina kun yritetään tallentaa väärin formatoidulle levylle. (Erään apuohjelman avulla voidaan rakennetta vaihtaa.) Tämä virheilmoitus voi esiintyä myös virran kytkemisen yhteydessä.

74: DRIVE NOT READY (Asema ei ole valmis)
On yritetty lukea levyä, vaikka asema on tyhjä.

LIITE C

Esittelylevyn ohjelmia

1. HAKEMISTO (DIR)

```
4 OPEN2,8,15
5 PRINT"■":GOTO 10000
10 OPEN1,8,0,"$0"
20 GET#1,A$,B$
30 GET#1,A$,B$
40 GET#1,A$,B$
50 C=0
60 IF A$<>"" THEN C=ASC(A$)
70 IF B$<>"" THEN C=C+ASC(B$)*256
80 PRINT"■"MID$(STR$(C),2);TAB(3);"■";
90 GET#1,B$:IF ST<>0 THEN 1000
100 IF B$<>CHR$(34) THEN 90
110 GET#1,B$:IF B$<>CHR$(34)THEN PRINTB$;:GOTO110
120 GET#1,B$:IF B$=CHR$(32) THEN 120
130 PRINT TAB(18);:C$=""
140 C$=C$+B$:GET#1,B$:IF B$<>""  THEN 140
150 PRINT"■"LEFT$(C$,3)
160 GET T$:IF T$<>"" THEN GOSUB 2000
170 IF ST=0 THEN 30
1000 PRINT" LOHKOA VAPAANA"
1010 CLOSE1:GOTO 10000
2000 IF T$="Q" THEN CLOSE1:END
2010 GET T$:IF T$="" THEN 2000
2020 RETURN
4000 REM LEVYKOMENTO
4010 C$="":PRINT">";
4011 GETB$:IFB$="" THEN4011
4012 PRINTB$;:IF B$=CHR$(13) THEN 4020
4013 C$=C$+B$:GOTO 4011
4020 PRINT#2,C$
5000 PRINT"■";
5010 GET#2,A$:PRINTA$;:IF A$<>CHR$(13)GOTO5010
5020 PRINT"■"
10000 PRINT "D-HAKEMISTO"
10010 PRINT ">-LEVYKOMENTO"
10020 PRINT "Q-LOPETUS"
```

```
10030 PRINT "S-LEVYN TILA"
10100 GETA$:IFA$=""THEN10100
10200 IF A$="D" THEN 10
10300 IF A$="." OR A$=">" OR A$=">" THEN 4000
10310 IF A$="Q" THEN END
10320 IF A$="S" THEN 5000
10999 GOTO 10100
```


2. LOHKOKARTAN NÄYTTÖ (VIEW BAM)

```
100 REM VICIN JA 64:N LEVYN  LOHKOKARTTA
105 OPEN15,8,15
110 PRINT#15,"I0":NU$="N/A N/A N/A N/A N/A":Z4=1
120 OPEN2,8,2,"#"
130 Y$="▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
140 X$="▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
150 DEF FNS(Z) = 2↑(S-INT(S/8)*8) AND (SB(INT(S/8)))
160 PRINT#15,"U1:";2;0;18;0
170 PRINT#15,"B-P";2;1
180 PRINT"▒";
190 Y=22:X=1:GOSUB430
200 FORI=0TO20:PRINT:PRINT"▒▒"RIGHT$(STR$(I)+" ",3);:NEXT
210 GET#2,A$:GET#2,A$:GET#2,A$
240 TS=0
250 FORT=1TO17:GOSUB450
260 Y=22:X=T+4:GOSUB430:GOSUB540:NEXT
270 FORI=1TO2000:NEXT:PRINT"▒"
280 Y=22:X=1:GOSUB430
290 FORI=0TO20:PRINT:PRINT"▒▒"RIGHT$(STR$(I)+" ",3);:NEXT
300 FORT=18TO35
310 GOSUB450
320 Y=22:X=T-13:GOSUB430:GOSUB540:NEXT
330 FORI=1TO1000:NEXT
340 PRINT"▒▒▒▒▒▒▒"
350 PRINT#15,"B-P";2;144
360 N$="":FORI=1TO20:GET#2,A$:N$=N$+A$:NEXT
370 PRINTN$" "TS-17;"VAPAATA LOHKOA"
380 FORI=1TO4000:NEXT
390 PRINT"▒"
400 INPUT"▒▒▒▒TOINEN LEVY ?    E▒▒▒";A$
410 IFA$="K"THENRUN
420 IFA$<>"K"THENEND
430 PRINTLEFT$(Y$,Y)LEFT$(X$,X)"▒▒";
440 RETURN
450 GET#2,SC$:SC=ASC(RIGHT$(CHR$(0)+SC$,1))
460 TS=TS+SC
470 GET#2,A$:IFA$=""THENA$=CHR$(0)
480 SB(0)=ASC(A$)
490 GET#2,A$:IFA$=""THENA$=CHR$(0)
500 SB(1)=ASC(A$)
```

```
510 GET#2,A$:IFA$=""THENA$=CHR$(0)
520 SB(2)=ASC(A$)
530 RETURN
540 PRINT"██"RIGHT$(STR$(T),1);"███";
550 REM  PRINTT"   "SC"  "SB(0)" "SB(1)" "SB(2)=CHR$(0)
560 IFT>24ANDS=18THEN:PRINTMID$(NU$,Z4,1);:GOTO660
570 FORS=0TO20
580 IFT<18THEN620
590 IFT>30ANDS=17THEN:PRINTMID$(NU$,Z4,1);:GOTO660
600 IFT>24ANDS=18THEN:PRINTMID$(NU$,Z4,1);:GOTO660
610 IFT>24ANDS=19THENPRINTMID$(NU$,Z4,1);:GOTO660
620 IFT>17ANDS=20THENPRINTMID$(NU$,Z4,1);:Z4=Z4+1:GOTO660
630 PRINT"█";
640 IF FNS(S)=0 THEN PRINT"+";:GOTO660
650 PRINT"█+";:REMRIGHT$(STR$(S),1);Z4,1);:GOTO72
660 PRINT"███";
670 NEXT
```


3. URAN JA SEKTORIN NÄYTTÖ (DISPLAY T&S)

```
100 REM********************************
110 REM* LEVYN URAN JA SEKTORIN       *
120 REM* TULOSTUS NAYTOLLA TAI        *
130 REM* KIRJOITTIMELLA               *
140 REM********************************
150 PRINT"██____________________"
160 PRINT"LOHKON SISALTO"
165 PRINT"────────────────":
170 REM********************************
180 REM* VAKIOIDEN ASETUS             *
190 REM********************************
200 SP$=" ":NL$=CHR$(0):HX$="0123456789ABCDEF"
210 FS$="":FORI=64 TO 95:FS$=FS$+"█"+CHR$(I)+"█":NEXT I
220 SS$="  ":FOR I=192 TO 223:SS$=SS$+"█"+CHR$(I)+"█":NEXT I
240 DIM A$(15),NB(2)
251 D$="0"
253 PRINT"        █N█AYTTOON███████████████TAI ██P█APERILLE"
254 GETJJ$:IF JJ$="" THEN254
255 IF JJ$="N"THENPRINT"       ██NAYTTOON█"
256 IF JJ$="P"THENPRINT"       ██PAPERILLE█"
260 OPEN15,8,15,"I"+D$:GOSUB 650
265 OPEN4,4
270 OPEN 2,8,2,"#":GOSUB 650
280 REM********************************
290 REM* URA JA SEKTORI               *
300 REM* LEVYASEMAN PUSKURIIN         *
310 REM********************************
320 INPUT"███URA, SEKTORI";T,S
330 IF T=0 OR T>35 THEN PRINT#15,"I"D$:CLOSE2:CLOSE4:CLOSE15:PRINT"LOPPU":END
340 IF JJ$="N" THEN PRINT"███URA"T" SEKTORI"S"█"
341 IF JJ$="P" THEN PRINT#4:PRINT#4,"URA"T" SEKTORI"S:PRINT#4
350 PRINT#15,"U1:2,"D$;T;S:GOSUB650
360 REM********************************
370 REM* PUSKURIN TAVUN NOLLA         *
380 REM* LUKEMINEN                    *
390 REM********************************
400 PRINT#15,"B-P:2,1"
410 PRINT#15,"M-R"CHR$(0)CHR$(5)
420 GET#15,A$(0):IFA$(0)=""THENA$(0)=NL$
428 IF JJ$="N"THEN430
```

D$="0"

puskurin osoitin, 2 kanava, paikka 1

MUISTIN LUKU väh merk tavu - enemmän merk tavu

```
430 IF JJ$="P"THEN460
431 REM*******************************
432 REM* PUSKURIN LUKEMINEN NAYTTOON*
433 REM* JA MAHDOLLINEN KESKEYTYS   *
434 REM*******************************
436 K=1:NB(1)=ASC(A$(0))
438 FOR J=0 TO 63:IF J=32 THEN GOSUB 710:IF Z$="E"THEN J=80:GOTO 458
440 FOR I=K TO 3
442 GET#2,A$(I):IF A$(I)="" THEN A$(I)=NL$
444 IF K=1 AND I<2 THEN NB(2)=ASC(A$(I))
446 NEXT I:K=0
448 A$="":B$=":":N=J*4:GOSUB 790:A$=A$+":"
450 FOR I=0 TO 3:N=ASC(A$(I)):GOSUB 790
452 C$=A$(I):GOSUB 850:B$=B$+C$
454 NEXT I:IF JJ$="N" THEN PRINTA$B$
458 NEXT J:GOTO571
460 REM*******************************
462 REM* LUKU JA TULOSTUS PAPERILLA *
464 REM*******************************
466 K=1:NB(1)=ASC(A$(0))
468 FOR J=0 TO 15
470 FOR I=K TO 15
472 GET#2,A$(I):IF A$(I)="" THEN A$(I)=NL$
474 IF K=1 AND I<2 THEN NB(2)=ASC(A$(I))
476 NEXT I:K=0
478 A$="":B$=":":N=J*16:GOSUB 790:A$=A$+":"
480 FOR I=0 TO 15:N=ASC(A$(I)):GOSUB 790:IF Z$="N"THEN J=40:GOTO 571
482 C$=A$(I):GOSUB 850:B$=B$+C$
484 NEXT I
486 IF JJ$="P" THEN PRINT#4,A$B$
488 NEXT J:GOTO571
571 REM*******************************
572 REM* SEURAAVA URA JA SEKTORI    *
573 REM*******************************
575 PRINT"SEURAAVSA URA JA SEKTORI"NB(1)NB(2) "■"
580 PRINT"LUETAANKO SEURAAVA URA JA SEKTORI"
590 GET Z$:IF Z$="" THEN590
600 IF Z$="K" THEN T=NB(1):S=NB(2):GOTO330
610 IF Z$="E" THEN 320
620 GOTO 590
```

NB(2) ← #8 (1)

Lohko

JJ$ = "N" tai "P" (näytölle tai Printterille
NL$ = CHR(0)
NB(1) = työnalla olevaa ura
NB(2) = sektori
} lohkon 2 ensimmäistä tavua


```
630 REM*******************************
640 REM* ALIRUTIINIT                 *
650 REM*******************************
660 REM* VIRHERUTIINI                *
670 REM*******************************
680 INPUT#15,EN,EM$,ET,ES:IF EN=0 THEN RETURN
690 PRINT"■LEVYVIRHE ■"EN,EM$,ET,ES
700 END
710 REM*******************************
720 REM* NAYTTOTULOSTUKSEN           *
725 REM* JATKAMISVIESTI              *
730 REM*******************************
740 PRINT"■■■■JATKETAANKO? (K/E)"
750 GETZ$:IF Z$="" THEN 750
760 IF Z$="E" THEN RETURN
770 IF Z$<>"K" THEN 750
780 PRINT"■URA" T " SEKTORI"S "■":RETURN
790 REM*******************************
800 REM* TAVUN SISALTO               *
805 REM* HEKSAKOODINA                *
810 REM*******************************
820 A1=INT(N/16):A$=A$+MID$(HX$,A1+1,1)
830 A2=INT(N-16*A1):A$=A$+MID$(HX$,A2+1,1)
840 A$=A$+SP$:RETURN
850 REM*******************************
860 REM* TAVUN SISALTO               *
870 REM* ASCII-KOODINA               *
880 REM*******************************
890 IF ASC(C$)<32 THEN C$=" ":RETURN
910 IF ASC(C$)<128 OR ASC(C$)>159 THEN RETURN
920 C$=MID$(SS$,3*(ASC(C$)-127),3):RETURN
```

4. LEVYN TARKISTUS (CHECK DISK)

```
1 REM LEVYN TARKISTUS
2 DN=8:REM LEVYASEMAN LAITENUMERO
5 DIMT(100):DIMS(100):REM HUONON URAN JA SEKTORIN VEKTORIT
9 PRINT"■■■■————————————————————"
10 PRINT" LEVYNTARKISTUSOHJELMA
12 PRINT"————————————————————————"
20 D$="0"
30 OPEN15,DN,15
35 PRINT#15,"V"D$
45 N%=RND(TI)*255
50 A$="":FORI=1TO255:A$=A$+CHR$(255AND(I+N%)):NEXT
60 GOSUB900
70 OPEN2,DN,2,"#"
80 PRINT:PRINT#2,A$;
85 T=1:S=0
90 PRINT#15,"B-A:"D$;T;S
100 INPUT#15,EN,EM$,ET,ES
110 IFEN=0THEN130
115 IFET=0THEN200:REM END
120 PRINT#15,"B-A:"D$;ET;ES:T=ET:S=ES
130 PRINT#15,"U2:2,"D$;T;S
134 NB=NB+1:PRINT" TARKASTETTU";NB;" LOHKOA■"
135 PRINT" URA  "T;" SEKTORI "S"■ □□□"
140 INPUT#15,EN,EM$,ES,ET
150 IF EN=0THEN85
160 T(J)=T:S(J)=S:J=J+1
165 PRINT"■■HUONO LOHKO:■■",T;S""
170 GOTO85
200 PRINT#15,"I"D$
210 GOSUB900
212 CLOSE2
215 IFJ=0THENPRINT"■■■■■■EI HUONOJA LOHKOJA!":END
217 OPEN2,DN,2,"#"
218 PRINT"■■HUONOT LOHKOT","URA","SEKTORI"
220 FORI=0TOJ-1
230 PRINT#15,"B-A:";D$,T(I);S(I)
240 PRINT,,T(I),S(I)
250 NEXT
260 PRINT"■"J"HUONOT LOHKOT ON VARATTU"
270 CLOSE2:END
```



```
900 INPUT#15,EN,EM$,ET,ES
910 IF EN=0 THEN RETURN
920 PRINT"■■VIRHE  "EN,EM$;ET;ES""
930 PRINT#15,"I"D$
```

5. LEVYN KUNNON TARKISTUS (PERFORMANCE TEST)

```
1000 REM  LEVYN KUNNON TARKISTUS
1010 :
1020 REM  VIC-20 JA COMMODORE 64
1030 REM  YHDEN LEVYN LEVYASEMA
1040 :
1050 OPEN 1,8,15:OPEN15,8,15
1060 LT=35
1070 LT$=STR$(LT)
1080 NT=30
1090 PRINT"■──────────────────────────"
1100 PRINT"   LEVYN KUNNON TARKISTUS"
1110 PRINT"──────────────────────────"
1120 PRINT
1130 PRINT"    LAITA TYHJA"
1140 PRINT
1150 PRINT"    LEVY ASEMAAN."
1160 PRINT
1170 PRINT"■   PAINA ■RETURN■"
1180 PRINT
1190 PRINT"   KUN OLET VALMIS■"
1200 FOR I=0 TO 50:GET A$:NEXT
1210 GET A$:IF A$<>CHR$(13) THEN 1210
1220 :
1230 :
1240 TI$="000000"
1250 TT=18
1260 PRINT#1,"N0:TESTILEVY,00"
1270 C1$="   LEVYA FORMATOIDAAN  "+CHR$(13)
1280 C2$="■ ODOTA NOIN 80 SEKUNTIA"
1290 CC$=C1$+C2$:GOSUB 1840
1300 IF TI<NTTHEN1370
1310 PRINT"■SYSTEEMI"
1320 PRINT"■         EI VASTAA"
1330 PRINT" KOMENTOIHIN OIKEIN"
1340 GOSUB 1880
1350 :
1360 :
1370 PRINT"■LEVYASEMA KUNNOSSA"
1380 PRINT"       MEKAANINEN TESTI■"
1390 TT=21
```



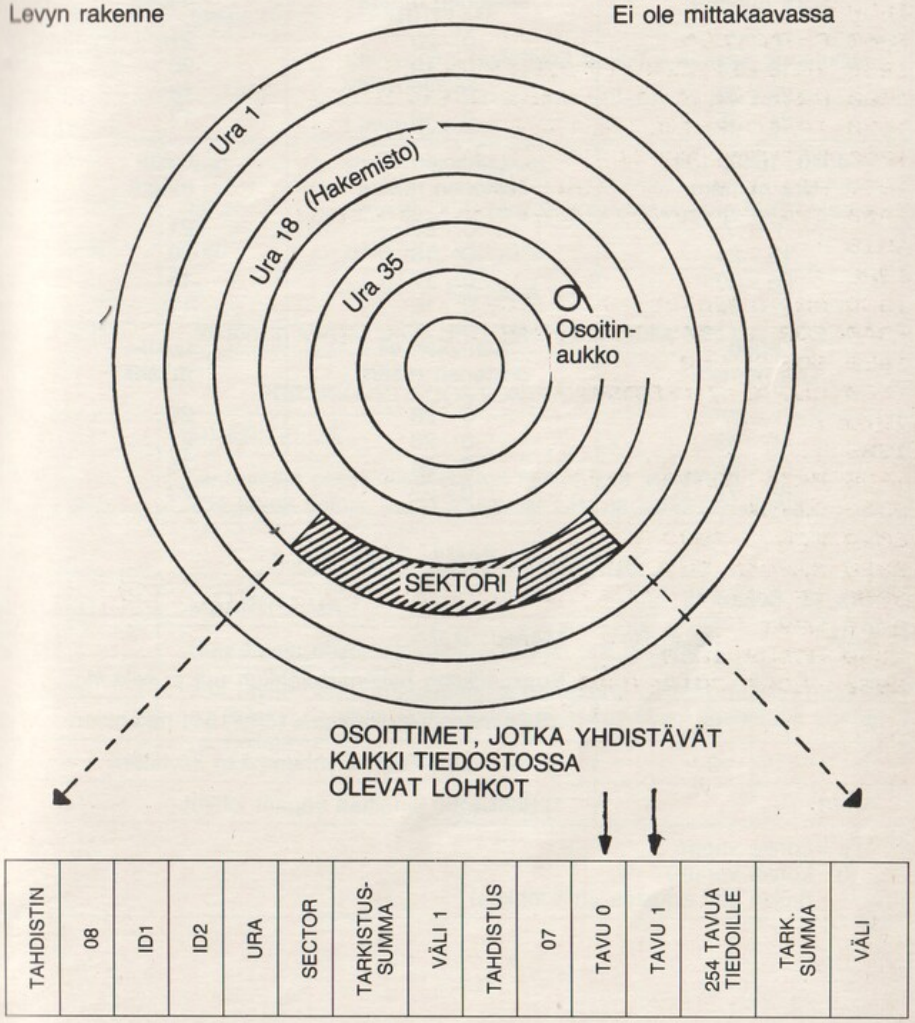
```
1400 OPEN 2,8,2,"0:TESTITIEDOSTO,S,W"
1410 CC$="KIRJOITUSTIEDOSTON AVAUS":GOSUB 1840
1420 CH=2:CC$="DATOJEN KIRJOITUS"  :GOSUB 1930
1430 CC$=CC$+" SULJETAAN"         :GOSUB 1840
1440 OPEN 2,8,2,"0:TESTITIEDOSTO,S,R"
1450 CC$="LUKUTIEDOSTON AVAUS"     :GOSUB 1840
1460 CH=2:GOSUB 1990
1470 PRINT#1,"S0:TESTITIEDOSTO"
1480 CC$="TIEDOSTON POISTAMINEN ■":TT=1 :GOSUB 1840
1490 :
1500 :
1510 TT=21
1520 OPEN 4,8,4,"#"
1530 NN%=(1+RND(TI)*254+NN%)AND255:PRINT#1,"B-P";4;NN%
1540 NN$="":FOR I=1 TO 255:NN$=NN$+CHR$(I):NEXT
1550 PRINT# 4,NN$;
1560 PRINT# 1,"U2:";4;0;LT;0
1570 CC$="KIRJOITETAAN URAAN"+LT$:GOSUB 1840
1580 PRINT#1,"U2:";4;0;1;0
1590 CC$="KIRJOITETAAN URAAN 1"         :GOSUB 1840
1600 PRINT#1,"U1:";4;0;LT;0
1610 CC$="LUETAAN URALTA"+LT$           :GOSUB 1840
1620 PRINT#1,"U1:";4;0;1;0
1630 CC$="LUETAAN URALTA 1"             :GOSUB 1840
1640 CLOSE 4
1650 :
1660 :
1670 PRINT"■ KAIKKI HYVIN!"
1680 PRINT" ──────────────"
1690 PRINT"■ POISTA LEVY ASEMASTA"
1700 PRINT"■ ENNEN VIRRAN KATKAISUA."
1710 PRINT
1720 END
1730 :
1740 :
1750 PRINT"   ■JATKETAANKO (K/E)?";
1760 FOR I=0 TO 50:GET A$:NEXT
1770 GET A$:IF A$="" THEN 1770
1780 PRINT A$"■"
1790 IF A$="E" THEN END
```

```
1800 IF A$="K" THEN RETURN
1810 GOTO 1760
1820 :
1830 :
1840 PRINT CC$
1850 INPUT# 1,EN,EM$,ET,ES
1860 PRINT"ILMOITUS:"EN;EM$;ET;ES;""
1870 IF EN<2 THEN RETURN
1880 PRINT"■ YKSIKKO"
1890 PRINT"■   ON VIALLINEN"
1900 TM$=TI$:GOSUB 1750:TI$=TM$:RETURN
1910 :
1920 :
1930 PRINT"DATAN KIRJOITUS"
1940 FOR I=1000 TO 2000:PRINT#CH,I:NEXT
1950 GOSUB1850
1960 CLOSE CH:RETURN
1970 :
1980 :
1990 PRINT"DATAN LUKU"
2000 GETA$
2010 FOR I=1000 TO 2000
2020 INPUT# CH,J
2030 IF J<>I THEN PRINT"■LUKUVIRHE:■":GOSUB 1850
2040 NEXT
2050 GOSUB 1850
2060 CLOSE CH:RETURN
```


LIITE D

Levyn rakenne

HUOM!
Ei ole mittakaavassa



1540/1541:n RAKENNE: Yksi sektori laajennettuna.

URIEN JAKO LOHKOIHIN

2040, 3040 Ura numero	Lohkojen eli sektorien määrä	Luku- määrä
1 - 17	0 - 20	21
18 - 24	0 - 19	20
25 - 30	0 - 17	18
31 - 35	0 - 16	17
4040, 1540, 1541 Ura numero	**Lohkojen eli sektorien määrä**	**Luku- määrä**
1 - 17	0 - 20	21
18 - 24	0 - 18	20
25 - 30	0 - 17	18
31 - 35	0 - 16	17
8050 Ura numero	**Lohkojen eli sektorien määrä**	**Luku- määrä**
1 - 39	0 - 28	29
40 - 53	0 - 26	27
54 - 64	0 - 24	25
65 - 77	0 - 22	23

1540/1541 BAM:N RAKENNE

Ura 18, Sektori 0.

TAVU	SISÄLTÖ	MÄÄRITELMÄ
0,1	18,01	Ensimmäisen hakemistolohkon ura ja sektori.
2	65	ASCII-merkki A, ilmoittaa 1540/1541-rakenteen.
3	0	Nollakenttä DOS:n myöhempään käyttöön.
4-143		*) Bittikartta, ilmoittaa vapaat lohkot.

*) 1 = Lohko vapaa
0 = Lohko varattu
(kukin bitti edustaa yhtä lohkoa)

1540/1541 HAKEMISTON OTSIKKO

Ura 18, sektori 0.

TAVU	SISÄLTÖ	MÄÄRITELMÄ
144-161		Levyn nimi täytettynä SHIFTin kanssa annetuilla välilyönneillä
162-163		Levyn tunniste
164	160	SHIFT + välilyönti
165,166	50,65	2A:n eli DOS version ja rakennetyypin ASCII-muoto
166-167	160	SHIFT + välilyönti
171-255	0	Nollia, ei käytössä
HUOM!	Joidenkin levyjen paikkoihin 180-191 voi tulla ASCII-merkkejä.	

PERÄKKÄISTIEDOSTON RAKENNE

TAVU	MÄÄRITELMÄ
0,1	Seuraavan peräkkäistiedoston lohkon ura ja sektori.
2-256	254 tavua tietoja, joihin sisältyvät CR:t ja keskeytysmerkit.

OHJELMATIEDOSTON RAKENNE

TAVU	MÄÄRITELMÄ
0,1	Seuraavan ohjelmatiedoston lohkon ura ja sektori.
2-256	254 tavua tietoa ohjelmasta tallennettuna CBM:n muistin muodossa (käskyt koodattuina). Tiedoston päättyminen merkitään kolmena nollatavuna.

HAKEMISTON MUOTO

Ura 18, Sektori 1	
TAVU	MÄÄRITELMÄ
0,1	Seuraavan hakemiston lohkon ura ja sektori
2-31	*Tiedoston määrittely 1
34-63	*Tiedoston määrittely 2
66-95	*Tiedoston määrittely 3
98-127	*Tiedoston määrittely 4
130-159	*Tiedoston määrittely 5
162-191	*Tiedoston määrittely 6
194-223	*Tiedoston määrittely 7
226-255	*Tiedoston määrittely 8

*YHDEN HAKEMISTOPAIKAN RAKENNE

TAVU	SISÄLTÖ	MÄÄRITELMÄ
0	128 + tyyppi	Tiedoston tyypin ja $80:n TAI-funktio TYYPIT: 0 = poistettu (DEL) 1 = peräkkäistiedosto (SEQ) 2 = ohjelma (PRG) 3 = käyttäjä (USR) 4 = suhteellinen (REL)
1,2		Ensimmäisen tietolohkon ura ja sektori
3-18		SHIFTin kanssa annetuilla välilyönneillä täytetty tiedoston nimi.
19,20		Vain suhteellisille tiedostoille: sivusektorilohkon ura ja sektori.
21 22-25		Vain suhteellisille tiedostoille: tietueen koko Ei käytössä
26,27		Tiedoston korvaamismerkin ura ja sektori, kun käytössä on OPEN @
28,29		Tiedostossa olevien lohkojen lukumäärä: vähemmän ja enemmän merkitsevä tavu.

SUHTEELLISEN TIEDOSTON RAKENNE

TIETOLOHKO	
TAVU	MÄÄRITELMÄ
0,1	Tietolohkon ura ja sektori
2-256	254 tavua tietoa. Tyhjien tietueiden ensimmäisessä tavussa on FF ja muissa tavuissa on 00 (kaikki binäärimuodossa). Osittain käytetyt tietueet täytetään nollilla (00).
SIVUSEKTORILOHKO	
TAVU	MÄÄRITELMÄ
0,1	Seuraavan sivusektorilohkon ura ja sektori.
2	Sivusektorin numero. (0-5)
3	Tietueen pituus.
4,5	Ensimmäisen sivusektorin (No. 0:n) ura ja sektori.
6,7	Toisen sivusektorin (No. 1:n) ura ja sektori.
8,9	Kolmannen sivusektorin (No. 2:n) ura ja sektori.
10,11	Neljännen sivusektorin (No. 3:n) ura ja sektori.
12,13	Viidennen sivusektorin (No. 4:n) ura ja sektori.
14,15	Kuudennen sivusektorin (No. 5:n) ura ja sektori.
16-256	120 tietolohkon osoittimien urat ja sektorit.

FORMATOINTI

OPEN 1,8,15 :PRINT # 1, "N0:LEVYN NIMI, TN":CLOSE 1

HAKEMISTO

NEW
LOAD "$,8 :LIST

OHJELMAN LATAAMINEN

NEW
LOAD "NIMI",8 *=Loput merkit ? mikä merkki kyvänä
LOAD "NIMI",8,1 ohjelma sijoitetaan muistiin

TALLETUS

NEW
SAVE "NIMI",8
SAVE "NIMI",8,1 estää autom. uudelleen sijoittamisen

TALLETU RIEMMAN OHJ PÄÄLLE

NEW
SAVE"@0:OHJ NIMI TÄYDELLISENÄ",8

TALLETUKSE TARKASTUS

NEW
VERIFY "OHJELMAN NIMI",8

LEVYNTYHJENTÄMINEN

NEW
OPEN 15,8,15 :PRINT # 15, "NEW0:NIMI, id" :CLOSE 15

HAKEMISTON POISTO

NEW
OPEN 15,8,15 :PRINT # 15, "N0:NIMI" :CLOSE 15

OHJELMAN TALLENTAMINEN SAMALLE LEVYLLE UUDESTAAN

NEW
OPEN 15,8,15 :PRINT # 15, "COPY0:UUSINIMI=0:VANHANIMI":
CLOSE 15

TIEDOSTOJEN YHDISTÄMINEN

NEW
OPEN 15,8,15 :PRINT # 15, "C0:UUSITIEDOSTO=0:VANHA1, 0:VANHA2,
0:VANHA3, 0:VANHA4" :CLOSE 15

OHJELMAN NIMEN VAIHTAMINEN

NEW
OPEN 15,8,15:PRINT # 15, "RENAME 0:UUSINIMI=VANHANIMI"
CLOSE 15

commodore
COMPUTER

Maahantuoja: PET-COMMODORE INC. Pl 148, 65101 Vaasa 10, puh. (961) 113 611.

Pet-Commodore Inc./Oy PCI-Data Ab on Commodore Electronics Ltd:n valtuuttama maahantuoja Suomessa.