

Timo Kinnunen  
Särkiniementie 16 A 41  
70700 Kuopio

## TIETOKONE

### 0.1.

#### MERKKI, INFORMAATIO JA TIETO TIETOJENKÄSITTELYLLISELTÄ KANNALTA

**Tietokoneilla** käsitellään informaatiota, mikä käsitteenä liittyy siihen, mitä tietojenkäsittelyn yhteydessä ymmärretään tiedolla. Informaatio muodostuu suuresta määrästä samankaltaisessa esitysmuodossa olevista tietueista, jota tavallisesti sanotaan tietojenkäsittelyssä tiedoksi. Informaatio voidaan esittää fyysisesti eri tavoin esimerkiksi ääniaaltoina ilmassa, mustina merkkeinä valkealla pinnalla tai sarjana sähkösykäyksiä. Informaatio esitetään tavallisesti jonkin fyysisen ilmaisimen kautta, joka tuottaa em. -laisia presentaatioita., jotka tulkittaessa muuntuvat aivojen sähkökemiallisiksi sykäyksiksi, joiden muodostumisen ja muodostamisen kautta muodostetaan inhimillisiä merkitysrakenteita, tai presentaatiot tulevat tulkituiksi jo olemassaolevien aivojen sähkökemiallisten tallenteiden referensseistä. Termi tieto liittyy toisaalta sellaisiin informaation presentaatioihin, joita voidaan pitää merkkeinä, kuten kirjaimiin ja numeroihin.

### 0.2.

#### MERKKI , INFORMAATIO JA TIETO FILOSOFISELTA KANNALTA

Filosofiselta kannalta edellämainittu käsitteiden sisällönmäärittäminen tuntuu perin rajoittuneelta, kuten esimerkiksi käsitys siitä, mitä **TIETO** on, sillä filosofinen tieto -ja tietäminen eivät ankkuroitu samalla tavoin ulkoisiin presentaatioihin, tai fyysisiin ja psyykkisiin konteksteihin, vaan se kytkeytyy erilaisiin filosofisiin järjestelmiin, joissa **MERKIN** sisällöt eivät rajoitu esimerkiksi suomen- tai englanninkielen sanoihin, latalaiseen aakkosistoon, tai arabialaisiin numeroihin niin kuin ne ilmenevät fyysisinä presentaatioina sinällään, tai jollaisina atomaarisina yksikköinä niitä voidaan käsitellä ilman, että käsiteltäisiin aidosti filosofisia käsitteitä tai merkkejä. Se, mikä filosofian kannalta on informaationkäsittelyä [esimerkiksi olioiden sentapaista aiheutettua liikettä toisten olioiden suhteen, joka tuottaa tietoa niistä] on filosofian piirissä jatkuvan väittelyn aiheena; saadut vastaukset riippuvat siitä, käsitetäänkö todellisuuden muodostuvan esimerksi konfiguroituvista

atomeista, inhimillisen agentin aktiivisesti muodostamista merkityskytkeytymistä, olioista niihin liittyvine merkitysmääriytymineen, psyykkisistä elementeistä, monadeista, hengestä jne., eli millainen on se aines, josta tuotetaan filosofisia järjestelmiä, ja voidaanko tätä ainesta käsitellä niin, että merkitykset säilyttäisivät sisältönsä - niinkuin automaattisen tietojenkäsittelyn informaation prosessoinissa oletetaan tapahtuvan. Filosofinen tietäminen ei ole informaationkäsittelyä automaattisen tietojenkäsittelyn tapaan, vaan operointia MERKEILLÄ. Kaikki, mikä on ajatusta, on oltava ilmaistavissa merkkeinä.

### 0.3

#### TIETOKONEIDEN KYTKEYTYMÄT FILOSOFIAAN JA ERITYISTIETEISIIN

**Tietokoneisiin** syvällisemmin perehtyvä törmää usein sellaisiin seikkoihin, jotka on löydetty jo ennen **tietokoneiden** keksimistä, kuten matemaattisiin notaatioihin [+,-,>><<,\*,...], logiikoihin [esimerkiksi predikaattilogiikka], matematiikan ja logiikan välimaastoon asettuvaan joukko-oppiin, lukujärjestelmiin [esimerkiksi heksadesimaalinen ja binaarinen lukujärjestelmä], systeemi- ja operaatioiden analyysiin, ja näitä kaikkia varten valmistettuihin fyysisiin laitteistoihin, jotka valtaosaltaan on kehitetty vasta äskettäin - kuten myös ohjelmiin, jotka perustuvat erilaisiin ohjelmointikieliin ja sovittautuvat erilaisiin laite- ja käyttöjärjestelmien muodostamiin ympäristöihin. Näiden seikkojen tultua opituiksi voidaan esittää kysymyksiä, kuten: miksi <BI>tietokoneet on kehitetty ja mitä tarkoitusperää ne palvelevat? (On selvää, ettei vastaukseksi riitä se, että **tietokoneilla** käsitellään ja muokataan informaatiota) Tällaiset kysymykset kytkevät **tietokoneiden** tuottaman maailman inhimilliseen todellisuuteen, josta **tietokoneet** ovat alunperin lähteneetkin, ja sitä kautta kysymyksenasettelut voidaan kytkeä niin filosofiaan kuin erityistieteisiin, ja raadolliseen arkipäiväiseen kokemusmaailmaan. Toinen hyvä kysymyksenasettelu edelliseen liittyen on: kuka hyötyy tietokoneista? Tavallinen **tietokoneen** käyttäjä ajattelee tässä yhteydessä useimmiten omaa hyötyään, mutta **tietokoneilla** on runsaasti myös laajempia eettisiä, esteettisiä, moraalisia ja yhteiskunnallisia ulottuvuuksia. Hyviä kysymyksiä olisivat esitettäväksi sentapaiset kysymykset, kuten mitä on tietäminen ja millaisena se tietokoneissa esiintyy ja mitä on ylimalkaan todellisuus?

Pyrin tässä tarkastelussani pääsääntöisesti kuvaamaan **tietokonetta** sellaisena, kuin miltä se näyttää välittömästi individuaalin käyttäjän kannalta funktionaalisina yksikköinä, jolla viitataan siihen, että **tietokoneen** keskeiset osat on valmistettu tiettyä tehtävää varten ja ovat ymmärrettävissä tämän tehtävän pohjalta konstruoituina yksikköinä. Tavallinen **tietokoneen** käyttäjä saa tietokoneen hankinnan myötä joukon käsikirjoja ja valmisohjelmia, ja on

tavallisesti kykenemätön itse konstruoimaan yksinkertaisiakaan ohjelmia; useimmiten käyttäjä haluaa yksinkertaistettua tietoa siitä, kuinka hänen laitteensa toimii. Täydellisesti ymmärrettynä **tietokone** on hyvin mutkikas kokonaisuus, ja siinä voidaan hahmottaa useammanlaisia funktionaalisia, filosofisia, moraaliseettisiä, yhteiskunnallisia ja tieteellisessä maailmassa hahmottuvia kokonaisuuksia, joita ovat näiden koneiden yhteiskunnalliset, uskonnolliset, psykologiset ja sosiologiset ulottuvuudet. **Tietokoneessa** ilmenevät myös erilaiset kielelliset kontekstit ja matemaattis-loogiset notaatiot, sekä erilaiset fyysiset ympäristöt. Kuitenkin tavallisen käyttäjän on helpompaa mieltää **tietokone** tiettyjen konkreettisten funktionaalisten kuvausten, kuin esimerkiksi muiden mainitsemiäni ulottuvuuksien kautta. Pyrin yleisesti käyttämään erilaisista, tavalliselle käyttäjälle tutummista ja toisaalta ainoastaan specialistien käyttämistä **tietokoneista** seuraavankaltaisia lähinnä suorituskykyyn, kapasiteettiin, nopeuteen, ja fyysiseen ulottuvuuteen liittyvillä nimityksillä, jotka kuvaavat toimintaa.

#### 0.4. OHJELMOINTIKIELISTÄ

Varsinaisesta ohjelmoinnista kiinnostuneet voivat perehtyä yleisesti sentapaisiin temaattisiin kokonaisuuksiin, kuten lausekieleen, kääntäjän perusrakenteeseen, ohjelmointikieliin, perustietorakenteisiin ja vaihtoehtoisii ohjelmointiparadigmoihin. Alunperin ohjelmointi suoritettiin kirjoittamalla tarvittava määrä komentorivejä käsin, joissa tehdyt kirjoitusvirheet merkitsivät tavallisesti sitä, ettei ohjelma toiminut halutulla tavalla, tai ei lainkaan. Myöhemmin kehitettiin välineitä, joilla ohjelmointi voitiin suorittaa keskusteluohjelmointina, mikä tarkoittaa sitä, että **tietokone** tarjoaa pyydetessä joukon käytetyn ohjelmointikielen valideja vaihtoehtoja komentoriveiksi, jolloin kirjoitusvirheitä ei ainakaan niiden kohdalla synny, jolloin voidaan keskittyä itse ohjelman yleiseen rakenteeseen ilman pelkoa muodollisista, vaikeasti paikannettaviin yksityiskohtiin asettuvista virheistä. Huomautan tässä yhteydessä siitä triviaalilta tuntuvasta erosta ohjelmoinnin ja ohjelman päätekstön välisestä erosta, ettei ohjelmoinnissa suoritettavat valinnat ole samoja kuin ohjelman valikkojen vaihtoehtoista suoritettavat valinnat. Samasta ohjelmointikielestä voi olla olemassa useita kääntäjiä. Paras yhdistelmä on tekstikääntäjä [check-out compiler] ja optimoiva kääntäjä. Kun ohjelmaa käytetään ensimmäistä kertaa, käytetään tekstikääntäjää, jossa on hyvät diagnoosirutiinit, mutta joka ei kehitä tehokasta koodia, ja kun ohjelma on testattu kuntoon, käännetään se optimoivalla kääntäjällä, jossa ei tarvita diagnoosia, mutta joka kääntää mahdollisimman tehokkaan konekieliohjelman.

Ei-numeerisiin sovelluksiin on olemassa sentapaisia ohjelmia, kuten **IPL-V**, **LISP** ja **Snobol**, joista **LISP** on tullut tunnetuksi tekoälysovelluksissa [Artificial Intelligence]. Simulointikielistä mainittakoon **GPSS** ja **Simscrip**. Keskustelukielistä tunnetuin on **BASIC** [Beginner's All purpose Symbolic

Instruction Code], joka on kehitetty Dartmouth-Collegessa USA:ssa. Toinen suosittu keskustelukieli on **APL** [A Programming Language] jonka kehittäjä on amerikkalainen Kenneth Iverson. Eniten levinneisiin tehtävänläheisiin kieliin kuuluvat **Algol** [ALGOrithmic Language] ja **Fortran** [FORmula TRANslation]. Hallinnollisia sovelluksia varten on olemassa mm. **Cobol** [Common Business Oriented Language].

#### 0.4.0. KONEKIELI

**Tietokoneen** omalla ohjelmointikielellä kirjoitettua ohjelmaa sanotaan konekieliseksi ohjelmaksi. Konekielisen ohjelman pituus riippuu suuresti käytettävissä olevien käskyjen määrästä: mitä enemmän käskyjä, sitä lyhyempiä ohjelmia ohjelmoija voi tehdä. Yksinkertaisin koneen malli lienee kuvitteellinen Epsilon, joka on yksiosoitteinen - kussakin käskyssä on vain yksi osoite, samoin kuin siinä ohjelmoijalla on käytettävissään vain yksi rekisteri (akkurekisteri, eli laskija). Todellisissa tietokoneissa on runsaasti konekielikäskyjä, kuten myös useita rekistereitä, kuten ainakin akku,- indeksi,- ja kantarekisterit, joita kutakin voi olla useampia kappaleita, sekä yleinen rekisteri, jossa voidaan käyttää missä tahansa edellisten tehtävissä. Konekielisten käskyjen määrä erottaa eri **tietokoneita** toisistaan, mutta niitä voidaan silti sanoa yleistietokoneiksi siinä mielessä, että erikoistuminen tietentyyppeihin tehtäviin tapahtuu yleisimmin valikoidun kokoonpanon avulla, joka voi koskea niin oheislaitteita kuin itse keskusyksikön modifiointia. Siten samankin koneen eri muunnelmilla voi olla käytettävissään erilainen määrä käskyjä ja valmiuksia erilaisiin oheislaittekokoonpanoihin.

#### 0.4.1. ASSEMBLY

Assembly Language tarkoittaa symbolista konekieltä, jonka lauseet ovat tavallista englanninkieltä; siinä **tietokoneen** annetaan muuntaa alkukielinen ohjelma konekielelle. Itse ohjelman luominen tapahtuu kirjoittamalla ASCII -merkeistä koostuvia **ASSEMBLY** -ohjelmakäskyjä annettuja merkintätapoja käyttäen. Muodostetut käskyrivit voidaan koostaa itsenäisesti toimiviksi ja tiedostonimellä käynnistyviksi ohjelmiksi käsittelemällä ASCII -tiedostoja tietyillä erillisohjelmilla, joita kuvaan myöhemmin. Eräs tapa ohjata **tietokonetta**, jota voidaan käyttää erityisesti **kotitietokoneissa**, on luoda eräajoja [.BAT], joissa kirjoitetaan suoraan haluttu joukko komentorivejä, joissa käytetään tiukan määrämuotoisia ilmaisuja, jotka koneen käyttöjärjestelmä kääntää konekielelle; myös yksittäisten käskyjen anto on mahdollista, mutta molemmissa tapauksissa työskentelyn on tapahduttava käyttöjärjestelmän tasolla. Tällainen mainittu ohjaus liittyy useimmiten laitteen vallitsevien oletusarvojen muuttamiseen jotta voitaisiin esimerkiksi ajaa laitteessa jokin

hyötyohjelma. Kehittyneemmissä ohjelmointikielissä käännösohjelma tuottaa yhdestä lauseesta useita konekäskyjä.

#### 0.4.2. BASIC

Käytettäessä ohjelmoinnissa sellaisia kääntäjä-tulkkeja kuten **BASIC** ja **PASCAL** ohjelmakäskyt kirjoitetaan yleensä niiden ehtojen vallitessa, joita tulkissa on. Ohjelmakäskyistä muodostuvat tiedostot tallennetaan ja niiden tunnisteosat saavat **kotitietokoneissa** muodon .BAS tai .PAS. Tällaisina ne voidaan toteuttaa ainoastaan ajamalla kääntäjän kautta. Kun tiedostot muunnetaan varsinaisiksi ohjelmatiedostoiksi niiden tunnisteosat saavat [muunostavasta riippuen] muodon .EXE tai .COM. Ohjelma voidaan ajaa itsenäisesti.

#### 0.4.3. Fortran

Lausekeskeinen kieli Fortran [**FOR**mula **TRAN**slation] kehitettiin IBM:n piirissä 1950-luvun puolivälissä, ja sittemmin sen ovat ottaneet ohjelmistoonsa useat muutkin valmistajat. Voidaan sanoa, että Fortranilla on kirjoitettu suurin osa laskentaohjelmista. Kielestä on olemassa useita muunnelmia, ja ohjelman edistyneempien versioiden perään on lisätty roomalainen numero ilmaisemaan sen kehitystasetta, kuten esimerkiksi Fortran IV. Alkuperäisessä Fortranissa käytettiin tavallisen reikäkorttilävistimen symboleja, kuten kirjaimia ja numeroita, sekä eräitä laskentaan liittyviä erikoismerkkejä [kuten +,-,/,\*,\*\*,()]. Näiden avulla voidaan ilmaista matemaattisia lakukaavoja, kuten  $Y=(X+Z)/B$ . Näiden merkkien lisäksi on olemassa Fortran-sanoja, kuten IF, joka tarkoittaa ohjelmassa ehdollista hyppykäskyä.

#### 0.4.4. Algol

Lausekeskeinen kieli Algol [**ALGO**rithmic **L**anguage] on kehitetty, ja sitä käytetään laskenta-algoritmien ilmaisemiseen. Algol määriteltiin 1950-luvun lopulla kahden yhdistyksen (Association for Computing Machinery ja Gesellschaft für Angewandte Mathematik und Mechanik) toimesta. Perusversiosta käytettiin nimeä Algol 60, ja sitä edistyneempiä versioita nimetään suuremmilla arabialaisilla numeroilla. Algolin etuna Fortranin verrattuna on sen hierarkkinen lohkorakenne ja itse ohjelman tietty dynaamisuus sen eri kohtien määrittelyssä.

#### 0.4.5 Cobol

Kertova kieli Cobol [**CO**mmun **B**usiness **O**riented **L**anguage] kehitettiin Yhdysvaltain Puolustusministeriön kehoituksesta haluttaessa päästä eroon laitteistoriippuvuudesta myös hallinnollisen tietojenkäsittelyn puolella. Kielen lopullisen muotoilun suoritti Codasyl [**CO**nference on **DA**ta **SY**stems **L**anguage]. Cobolissa voidaan käyttää engallninkieltä lähes normaaliin tapaan annettaessa ohjelmakäskyjä, kuten DIVIDE LUKUMÄÄRÄ INTO SUMMA GIVING KESKIARVO. Cobol-ohjelma on jaettu neljään lohkokon: Identification Division, joka sisältää ohjelman nimen, päiväyksen ja muita tietoja; Enviroment Division, joka käsittää tiedot tietokoneesta, jolla ohjelma tullaan ajamaan (koneen tyyppi, keskusmuistin koko, erilaisten syöttö- ja tulostuslaitteiden ja ulkoisten muistien tyypit ja määrät); Data Division, joka sisältää yksityiskohtaiset määrittelyt käsiteltävistä tiedoista (kentän pituus, aakkosnumeerinen tai numeerinen jne.); sekä Procedure Division , joka käsittää ohjelman lauseet, joista generoituu konekieli ohjelman käskyt tehtävien suorittamiseksi.

#### 0.4.6. Olio-ohjelmointi ja sovelluskehittimet

Olio-ohjelmointiin liittyvää C++ -ohjelmointikieli kehitettiin yksinkertaisemmista C-kielistä, joita voidaan käyttää korkeammalla abstraktion tasolla, kuten muitakin korkeamman tason ohjelmointikieliä. Sovelluskehittimet, kuten IGRESS, kehitettiin siitä syystä, että ohjelmistoja ei voida käyttää erilaisissa konteksteissa samalla tavoin, vaan niitä on räätelöitävä tarkoitukseensa soveltuviksi; niiden käyttö liittyykin kiinteästi, ja näkyvästikin tietojärjestelmien asennukseen esimerkiksi organisaatioiden palvelukseen.

0.5.

TERMIN TIETOKONE FUNKTIONAALISTA MÄÄRITTELYÄ

#### 0.5. A. Tietokone

Yleistermiä **tietokone** käytän aina kun haluan puhua **tietokoneista yleensä** - riippumatta niiden käyttöjärjestelmistä, kapasiteetista, nopeudesta, soveltuvista hyötyohjelmistoista, käytetyistä ohjelmointikielistä, teknisestä toteutuksesta ja laiteympäristöistä. Keskeistä tällaisessa kuvauksessa on se, että on kyseessä laite, jolla prosessoidaan vaihtelevin kriteerein ja periaattein kaikkea sitä, mikä on kyetty kvantifioimaan ja merkkeinä ilmaisemaan ja mitä ihminen tiedoksi nimittää. Primitiivisessä **tietokoneessa** työskentelee enimmäkseen vain yksi yksikkö kerrallaan (vrt. Epsilon), jolloin laitteen eri

rakenneosilla on huono hyötysuhde. Parempi hyötysuhde saavutetaan silloin, kun annetaan usean ohjelman lähes samanaikaisesti käyttää samaa konetta niin, että esimerkiksi ohjelma A voi työskennellä samanaikaisesti kun esimerkiksi ohjelma B odottaa syöttöä. Tällaista kutsutaan moniajoksi [multiprogramming], jollaista tarvitaan silloin, kun halutaan luoda taloudellisesti kannattavaa tosiaikajärjestelmää. Nykyisissä **tietokoneissa** on tällaista varten keskeytysmahdollisuus [interrupts], joka tarkoittaa sitä, että keskeytyksen sattuessa **tietokone**, sitten kun se on suorittanut loppuun meneillään olevan käskyn, voi hypätä pois tästä ohjelmasta määrättyyn osoitteeseen samalla kun se merkitsee eräästä muistin rekisteristä sen käskyn osoitteen, joka olisi ollut seuraavana toteutusvuorossa ellei keskeytystä olisi tullut. Siitä muistipaikasta, johon tietokone hyppää keskeytyksen sattuessa, alkaa ohjelma, joka analysoi keskeytyssyyntä, ja alkaa sen mukaisiin toimenpiteisiin. Tärkeä tekninen järjestely tässä yhteydessä on muistin suojaus, jossa jokaiselle ohjelmalle osoitetaan oma alueensa keskusmuistista, josta se estetään teknisesti pääsemästä ulos. Toinen tekninen järjestely on se, että tietyt toimenpiteet ovat etuoikeusasemassa, eikä niitä voi suorittaa mikään muu kuin valvontaohjelma; esimerkiksi jonkin muun ohjelman sijaitessa valvontaohjelman alueella voi ainoastaan valvontaohjelma heittää tunkeutujan muualle. Siten valvontaohjelmalla on etuoikeus asettaa ja muuttaa suojattujen muistialueiden kokoa ja sijainteja. Valvontaohjelma on koneen valmistajan rakentama, ja sitä sanotaan myös systeemiohjelmaksi (erotuksena sovellusohjelmista). Valvontaohjelma voi olla osa laajemmasta käyttöjärjestelmästä [Operating System], joka on kokoelma ohjelmia käytösaantoihin. Koneenvalmistajien tarjoamat käyttöjärjestelmät saattavat sisältää useita valinnaisia optioita, joiden lisäys merkitsee sitä, että käyttöjärjestelmä varaa itselleen suuremman osan keskusmuistista (monissa **kotitietokoneiden** käyttöjärjestelmistä on mahdollista asettaa erilaisia optioita, ja määrittellä käyttöjärjestelmän muistista varaaman tilan suuruus suhteessa sovellusohjelmiin asennuksen yhteydessä, ja sen jälkeenkin).

Moniajo ja tiedonvälitys suovat mahdollisuuden monipääsyyn [Multiple Access], joka tarkoittaa sitä, että usea käyttäjä voi samanaikaisesti käyttää samaa tietokonetta päätteeltä käsin. Monipääsy tunnetaan myös nimellä osituskäyttö [time sharing]. Monissa osituskäyttösystemeissä päätteenkäyttäjä voi rakentaa ja ajaa omia ohjelmiaan vapaasti ainoana rajoitteenaan muistialueensa koko. Ns. erityissysteemeissä käyttäjä ei voi ohjelmoida, vaan hänen on syötettävä tämän erityissovelluksen tarvitsemia tietoja tarkalleen ennaltamäärätyssä järjestyksessä, tai annetulla tavalla. Moniajossa, ja erityisesti osituskäytössä käyttöjärjestelmän annetaan hoitaa sellaiset tehtävät kuten ohjelmien sisään- ja ulosheitot ulkoiselta muistilta keskusmuistiin, ja päinvastoin, koska keskusmuistiin ei kaikkea kerrallaan saata mahtua. Mikäli tällaisia tapahtumia tarvitaan paljon heikkenee systeemin suorituskyky samalla, koska sen hallinto vie aikaa [overhead time; yleisrasiteaika]. Mikäli **tietokone** on tehokas, on hyvä käyttää aika-annostusta [time slicing], esimerkiksi antamalla joka sekunnin ajaksi jokaiselle käyttäjälle 20

millisekunnin aika käyttöön, ja tämä kvantti riittää tehokkaallekin pääteikäyttäjälle.

## 0.5. B. Kotitietokone

Termiä **kotitietokone** käytän silloin, kun tarkoitan yleisesti 8086/8088, tai 80286-80486 suorittimilla [mikroprosessoreilla] varustettuja pieniä PC tai AT-tason pöytä- tai kannettavia, pääasiallisesti erillisinä yksikköinä käytettäviä laitteita. **Kotitietokoneiksi** voidaan laskea myös pöytä- tai kannettavat Macintosh, ATARI, AMIGA jne. tietokoneet, joiden käyttöjärjestelmissä on nykyisin myös DOS-levykkeiden luku-, tallennus-, ja alustusmahdollisuus ja DOS:iin sovitettujen ohjelmien ajomahdollisuus. **Kotitietokoneiksi** en kuitenkaan laskisi sellaisia laitteita, kuten NINTENDO ja COMMODOR, joita käytetään pääsääntöisesti erilaisten tietokonepelien pelaamiseen, koska ne on sitä varten suunniteltuja.

Termiä **kotitietokone** käytän myös siksi, että se kuvastaa tällaisen laitteen käyttötapaa ja sitä, että laite on ajateltavissa aktiiviseksi, integratiiviseksi pienikokoiseksi- ja tehoiseksi agentiksi, joka käyttää passiivisesti ulkoisia tehokkaampia tietolähteitä työskennellessään lähes missä tahansa ja millaisissa olosuhteissa hyvänsä, kuten esimerkiksi kotona, jossa on tavallisimmin runsaasti erilaisia elektromagneettisten laitteiden aiheuttamia häiriöitä, kuten jännitepiikkejä sähköverkossa, magneettikenttiä, ja häiritseviä videosignaaleja ja usein jopa epävakaisuutta jatkuvassa, tasalaatuisen energian saannissa. Usein käytetty termi **mikrotietokone** on hämäännyttävä monessakin mielessä - sehän ei viittaa juuri muuhun kuin käytetyn **tietokoneen** pieneen kokoon, ja termin etuosaa, mikro, käytetään muiden laitteiden ja laiteosien, tai toimintojen yhteydessä. On esimerkiksi helppoa sekoittaa termi **mikrotietokone** kotitalouksissa käytettyyn kotimikroon, joka on mikroaalloilla toimiva uuni, tai sekoittaa edellämainitut termit **tietokoneen** fyysiseen ja keskeiseen elementtiin, suorittimeen eli mikroprosessoriin, tai sekoittaa se vuorostaan sellaiseen keskeiseen käyttöjärjestelmän elementtiin, kuten komentotulkkiin eli komentoprosessoriin [COMMAND.COM], tai sekoittaa kaikki edellämainitut kiintolevyjen ja käyttömuistien ulospäiseen kommunikointiin liittyvään mikrokanavaan, tai tekstinkäsittelyyn liittyviin termeihin, kuten mikroaskellukseen ja mikrotuumaan.

Termillä **kotitietokone** viitataan siis sellaiseen fyysisiltä ulottuvuuksiltaan pieneen ja integroituihin laitteeseen, jota voidaan pitää itsenäisenä, keskittettynä tietojenkäsittely-yksikkönä, josta toisaalta voidaan olla verkko- tai tietoliikennöintisuhteessa toisiin, mahdollisesti samanlaisiin tai tehokkaampiin **tietokoneisiin**, ja hallita niitä integroitujen työtehtävien puitteissa, tai mikäli kyvyt antavat myöden, saavuttaa niihin nähden laajempikin kontrolli. Toisaalta termillä **kotitietokone** viitataan nimenomaan sellaiseen laitteistoon, jolla



voidaan työskennellä lähes missä tahansa juuri laitteiston vaatiman pienen tilan vuoksi; siten termi tarkoittaa erityisesti myös erilaisia kannettavia [portable] koneita. **Kotitietokoneella** voidaan myös suorittaa erilaisia etätyöskentelyn muotoja lähinnä sen kytkeytyvyydestä, pienestä koosta, ja siirrettävyydestä johtuen.

### **0.5.C. Lähiverkot , avoimet verkot ja joustavat tietoliikennöintiratkaisut**

Verkot ja niitä kuvaavat verkkoteoriat liittyvät teoreettiselta kannalta mm. joukko-oppiin, predikaattilogiikkaan, lauselogiikkaan ja vastaaviin, lähinnä filosofian ja matematiikan piirissä kehitettyihin oppeihin, jotka kuvaavat annetulla tavalla määriteltyjen ja attributoitujen objektien tai olioiden suhdetta toisiinsa tietyissä konteksteissa, tai universaalilla, yleispätevällä tasolla. **Lähiverkot** on ymmärrettävä, fyysisinä esinejoukkoina tarkasteltuina, useammista, ehkä itsenäisestikin käytetyistä **kotitietokoneista** tai muista ehkä yksinomaan päätteen tapaan käytetyistä yksiköistä rakennetuiksi verkoiksi, joissa on tavallisesti keskusyksikkö, joka on muita yksiköitä tehokkaampi ja jossa verkko-ohjelmisto sijaitsee, ja josta käsin muita verkkoon kuuluvia laitteita voidaan hallita. Termejä **avoimet verkot** ja **joustavat tietoliikennöintiratkaisut** voisi mielestäni käyttää silloin, kun useampia **tietokoneita** on kytketty niin, että niillä on esimerkiksi **lähiverkkoratkaisuissa** jokin verkon moduleista, tai sen ulkopuolelta kontrolloimassa verkkoa. **Joustavat tietoliikennöintiratkaisut** merkitsisivät tässä tapausessa sitä, että tiedon siirto olisi mahdollista mistä verkon solmusta mihin tahansa muuhun verkon solmuun, ja myös sen ulkopuolelle ja sisään samoin edellytyksin. Laajemmissa tietoliikennekytkeytymissä [jotka voivat olla enemmän tai vähemmän suljettuja tai avoimia] yksi tai useampia moduleita, tai niiden ryhmiä ovat joustavassa ja vastavuoroisessa suhteessa toisiinsa, tai johonkin suurempaan keskusyksikköön, ja mahdollisesti [sen kautta] muihin suuriin keskusyksikköihin, joiden päätteinä tai tehtävärajattuina isäntinä käyttäjäliittymät toimivat tietoliikenneyhteydessä ollessaan tai kytkeytyessään eriasteisin toimintavaltuuksin **tietoliikenneverkkoon** kuuluviin **tietokoneisiin**.

### **0.5.D. Keskisuuret tietokoneet**

Termiä **keskisuuret tietokoneet** voidaan mielestäni soveltaa erilaisiin tehtävärajattuihin integratiivisiin tietojenkäsittelyllisiin kokonaisuuksiin, kuten tietojenkäsittelylaitteistoihin, joita käytetään mm. tuotettaessa kerroskuvauksen [CT] informaatiota näyttöpäätteelle. Termiä voidaan epäilemättä soveltaa myös eräisiin hieman avoimempiin järjestelmiin, joita

käytetään mm. ydinvoimaloiden ja teollisuuslaitosten prosessiseurannassa. Termin kaikkien keskeisin sovellusalue lienee kuitenkin eräät avoimet järjestelmät, joihin voivat liittyä myös tavalliset **kotitietokoneen** käyttäjät. Usein - puhuttaessa ohjelmista, joita näissä **keskisuurissa tietokoneissa** käytetään - voidaan puhua tekoälysovelluksista [Artificial Intelligence] tai **asiantuntijajärjestelmistä**. Tosin näitä esiintyy myös **kotitietokoneiden** ohjelmistoissa, mutta kapasiteetiltaan ja nopeudeltaan vähäisempinä muunnoksina.

### 0.5.E. Suurtietokoneet

Termiä **suurtietokone** käytän puhuessani suurista, tietyin kvalitaatioin mitattuna integratiivisista, fyysisistä yksikkökokonaisuuksista, joiden suoritusnopeus on suuri, ja joihin voi olla fyysisesti kiinteästi kytkettyinä, tai tilapäisin modeemi yms. -kytketymin samanaikaisesti useita kymmeniä, ellei satoja liittymiä, jotka voivat suorittaa niissä erilaisia tehtäviä lähes samanaikaisesti, koska yksittäiset ajoajat ovat hyvin lyhyitä. Tällaisia laitteistoja käyttävät mm. erilaiset tutkimuslaitokset ja yliopistot.

**Suurtietokoneen** kapasiteetti ei sinällään tietenkään kohoa liittymien määrän noustessa - varsinkaan tapauksissa, joissa ne ovat **suurtietokoneeseen** passiivisessa otossuhteessa, vaan pikemminkin on ajateltavissa, että se on sellainen integratiivinen tietojenkäsittely-yksikkö, jonka ominaisuudet on suunniteltu erityisesti huomioiden mm. nopeus, tehokkuus ja taloudellisuus, mihin pienempitehoisilla yksiköillä työskenneltäessä ei kyetä.

**Suurtietokoneen** käyttö- ja massamuistitilat ovat valtavia verrattuna esimerkiksi tavallisimmin käytössä oleviin **kotitietokoneisiin**. Eräs keskeinen piirre, joka on **suurtietokoneilla** on se, että niiden suunnittelussa on huomioitu myös tavalliset tietokoneenkäyttäjät **kotitietokoneineen** tai päätetyöskentelyineen.

### 0.5.F. Supertietokoneet

Termiä **supertietokone** voidaan mielestäni käyttää silloin, kun tietokonesuunnittelussa lähdetään siitä, että laitteistolla suoritetaan hyvin mittavia laskutoimituskokonaisuuksia suurella nopeudella ja tehokkuudella, ja että laitteistoa käytetään nimenomaan suurten, integroitujen tehtävien suorittamiseen, kuten armeijan ja sääpalvelun piirissä. Tavalliset **kotitietokoneen** käyttäjät työskentelevät näillä laitteilla äärimmäisen harvoin, jos koskaan. Tavanomaisimpia käyttäjiä esimerkiksi yliopistojen piirissä lienevät erilaiset tutkijat, joiden tutkimusalueet liittyvät pääsääntöisesti tällöin luonnontieteisiin, kuten kemiaan, fysiikkaan, biologiaan ja lääketieteeseen.

Yleisesti **tietokone** voidaan karkeasti jaoittaa laitteiston ja ohjelmien suhteen seuraaviksi osioiksi:

Laitteisto [HARDWARE]

Käyttöjärjestelmä [mm. DOS; DRDOS; XENIX; PS/2]

Järjestelmä-SOFTWARE [mm. Basic, C, C++, Cobol, Fortran, Pascal]

Loppukäyttäjän työkalut [END-USER TOOLS]

### 1.1. LAITTEISTO [Hardware]

Laitteistolla tarkoitan yleisessä mielessä kaikkia sellaisia fyysisiä laitteita, joita voidaan käyttää tietojenkäsittelytehtävien yhteydessä. Tällöin käsitteen piiriin kuuluisivat mm. erilaiset nauha- ja cd tallentimet, still- ja videokamerat sovittimineen, syntetisaattorit MIDI-liitäntöineen, puheen ja tekstintunnistimet, rumpu- ja tasoskannerit, virtalähteet, kaapeloinnit ja linjakytkenät jne.

**Kotitietokoneesta** puhuessani tarkoitan suppeassa mielessä sellaista fyysistä laitteistoa ja funktionaalisten komponenttien kokonaisuutta, joka koostuu keskusyksiköstä, näppäimistöstä, monitorista ja kirjoittimesta. Tämä jaoittelu johtuu siitä, että esimerkiksi **kotitietokoneita** myydään peruskokoonpanossaan näistä elementeistä koostuvana kokonaisuutena, ja että **tietokone** mielletään tällä tavoin koostuvaksi yleisessä tietoisuudessa.

**Kotitietokoneiden** tapaisten, enemmän tai vähemmän integroitujen laitteiden keskusyksikön sisäinen fyysisyys sisältää sen virransyöttö ja lämmönsäätelyjärjestelmän, keskusmuistin ja laitteen toimintaan kytkeytyvät ROM ja RAM piirit, emokorttiin tai korttipaikkoihin [SLOT] asennetut lisämuistit, keskus- ja rinnakkaissuorittimet, levyasemat, kiintolevyt, porttiliitännät, kaiuttimen, hiiren mahdollisine kortteineen [BUS MOUSE], sisäiset kaapeloinnit ja oheislaitteet kortteineen ja kytkentöineen jne.

Laajasti käsitettynä **kotitietokone** voi sisältää edellisten lisäksi kytkennät erilaisiin **tietoliikenneverkkoihin** ja **suurtietokoneisiin** modeemien tai suorien liitäntöjen välityksellä, auditiivis-visuaaliset sensorit, anturit ja tulkintalaitteet ja tietenkin tällaisten informaatioisältöjen muunnos- ja käsittelylaitteet.

**Tietokoneiden** voidaan yleisesti katsoa muodostuvan tietyistä toiminnallisista

moduleista - aivan kuin niidenkin voidaan katsoa muodostuvan tietyistä, identifioitavista moduleistaan.

Rajoittuneessa funktionaalisessa mielessä **tietokone** voi tarkoittaa myös itsenäisesti toimivaa valvonta,- analyysi,- testaus- tai kontrollointivälinettä, joka käsittelee tietoa fyysikaalisen laitteen asettamassa toimintaympäristössä, jolloin "ohjelmat" toimivat fyysisten piirien asettamissa puitteissa. Viimeksimainitunlaisia **tietokoneita** käytetään esimerkiksi teollisuuden valmistus- ja kontrollointitehtävissä. Eräissä tapauksissa ohjelmiston [SOFTWARE] ja laitteiston [HARDWARE] suhde on elimellinen rajoitetummassa mielessä, kuten käytettäessä sellaisia ohjelmia, joihin liittyy eräitä fyysisiä osia, kuten kortteja ja moduleita, joita ilman ohjelmat eivät voi toimia, koska ohjelmat on rakennettu niin, että osa ohjelasilmukoista kulkee näiden fyysisten osioiden kautta. Monien valvonta,- analyysi,- tai testauslaitteiden osalta eron tekeminen ohjelman ja itse laiteosioiden välillä on vaikeaa siksi, että monissa tapauksissa ohjelma toimii fyysisissä, tarkoitusta varten konstruoiduissa piireissä. **Kotitietokoneen** osalta esimerkkinä olkoon vaikkapa ROM BIOS, joka suorittaa laitteiston testauksen laitetta käynnistettäessä.

Kaupallisesti ajateltuna on edullista valmistaa tiukasti tietyn fyysikaalisen laitteen rajaamassa toimintaympäristössä toimivia tietojenkäsittely-yksiköitä, koska niiden laitton kopiointi on huomattavasti vaikeampaa kuin ohjelmien, joita voidaan helposti kopioida ja ajaa erilaisissa laiteympäristössä.

**Kotitietokone** on esimerkkinä sellaisesta avoimesta, fyysisestä järjestelmästä, jossa on mahdollista ajaa kymmeniätuhansia erilaisia ohjelmia, joiden laitton monistaminen on helppoa, ja joista valmistaja ei koskaan voi saada täyttä hyötyä.

<MS>Ohjelmalevykkeiden erilaiset suojaukset eivät ole kovinkaan tehokkaita, mistä esimerkkinä olkoon vaikkapa LOTUS 1-2-3 ohjelman muokkaus sellaiseksi, että sitä voidaan rajoituksetta kopioida kiintolevyille: ohjelman 123.COM tiedosto ajetaan UNGUARD-ohjelmalla, joka muuntaa sen 123.EXE tiedostoksi.

Laitteisto [HARDWARE] on sitä, mikä useimmille tulee mieleen puhuttaessa laiteympäristöstä, ylläpidosta, huollosta ja korjauksista. Ohjelmisto [SOFTWARE] on taas jotakin sellaista, mikä liittyy käyttöympäristöön, loppukäyttöön, ohjelmien asennukseen, dataan, jne.

**Kotitietokoneet** kootaan pitkälti sarjavalmistetuista komponenteista, moduleista, ja käyttäjä voi itsekin asentaa peruskokoonpanoon lisälaitteita esimerkiksi asettamalla uusia ohjainkortteja laitteineen ja installoimalla ne laitteen käyttöjärjestelmään korttien valmistajan haluamalla tavalla ja

tietokoneenvalmistajan suositusten mukaisesti. Useimmissa tapauksissa ei yhteensopivuus ole ongelma.

Yksinkertaisimmassa kokoonpanossaan **kotitietokoneen** laitteistoon [HARDWARE] kuuluu siis fyysisinä [ja samalla myös funktionaalisina] elementteinä:

1.1.1. Keskusyksikkö

1.1.2. Näppäimistö

1.1.3. Monitori

1.1.4. Muita oheislaitteita,

kuten kirjoitin, kiintolevyt, lisälevyasemat, rinnakkaisprosessorit, lisämuistit, sekä kaapelit, modeemit, faksit, ja erilaisia niiden käytön yhteydessä keskusyksikköön liitettäviä kortteja.

<MS>Mikäli jokin <BS>kotitietokoneen<MS> emokortti, tai jokin siihen liitetyistä korteista vioittuu, ei sitä yleensä kannata korjata, mikä on eräs integroidun modurakenteen kirouksista. Monet seikat tuottavat <BS>kotitietokoneen<MS> toimintaan häiriöitä, jotka eivät ole varsinaisia vikoja. Korttien komponentit esimerkiksi kestävät huonosti jännitteenvaihteluja ja korkeita lämpötiloja, josta syystä tietokonetta yleensäkin -tai sen oheislaitteita ei tulisi asettaa lämpöpattereiden läheisyyteen, tai suoralle auringovalolle alttiiksi. Ja koska <BS>kotitietokone<MS> toimii suunnilleen television lähestystaajuudella, voivat sen videotoinnot häiriintyä esimerkiksi lähellä mahdollisesti sijaitsevien TV-lähetinten ja vastaanottimien vaikutuksesta.

<BS>Kotitietokoneet<MS> tulisi yleisesti aina kytkeä maadoitettuun pistorasiaan, mikä ei kuitenkaan aina estä mm. jääkaappien, pesukoneiden, ja muiden vastaavien laitteiden aiheuttamia häiriöitä mm. monitorissa -ja mikä näkyy monitorissa, voi olla vaikutusta muuallakin. Huomioitakoon, että ukonilmalta ei voida tietokonetta suojata katkaisemalla siitä virta ja jättämällä virtajohto seinään, koska virtaa pääsee silti muuntajan ensiökäämiin; tehokas suoja ukonilmalta on virtajohdon ottaminen seinästä irti. Tavanomaiset jännitepiikit ja jännitteenvaihtelut verkossa ja satunnaiset sähkökatkokset elimonoidaan tehokkaimmin käyttämällä esimerkiksi hakkurivirtalähteitä, jotka syöttävät <BS>kotitietokoneeseen<MS> virtaa tasaisesti akuista. Hakkurivirtalähteet eivät kuitenkaan kykene estämään tehokkaasti voimakkaiden salamaniskujen aiheuttamien jännitepiikkien aiheuttamia vaurioita <BS>kotitietokoneessa<MS>, mikäli virtalähde on päällä ukonilman

aikana, josta syystä virtalähde olisi syytä sammuttaa ja ottaa sen virtajohto pistokkeesta irti.

## 1.2. KÄYTTÖJÄRJESTELMÄ

[DOS, DRDOS, XENIX, PS/2]

### 1.2.1. MSDOS -KÄYTTÖJÄRJESTELMÄ

DOS -käyttöjärjestelmistä tarkastelemme aluksi **IBM:n** ja **Microsoftin** kehittämää käyttöjärjestelmää, joiden ero oli aiemmin lähinnä siinä, että **IBM:n kotitietokoneissa** voitiin käyttää sellaista BASIC- tulkkiä, joka hyödyntää **IBM:n** laitteissa olevia ROM -muistipiirejä, jotka puuttuvat **IBM** klooneista, joita varten kehitettiin GWBASIC-tulkki, joka ei toiminnassaan tarvitse mainittuja ROM-piirejä. Tämän lisäksi DOS [Disc operating system] oli alunperin käyttöjärjestelmä, joka soveltui **kotitietokoneisiin**, joissa oli mikroprosessori, jonka tyyppi on 8086/8088. Nämä pulssikoodilla toimivat tietokonelaitteet [PC=Pulse Code] aktivoitiin niin, että käynnistyksen yhteydessä niiden keskusmuistiin syötettiin levyiltä [tietystä kalvolevyasemasta] luettu käyttöjärjestelmä. Näiden laitteiden keskusmuistin määrä oli kiinteä, ja niissä käytetyt käyttöjärjestelmät eivät kyenneet käsittelemään suuria, lohkottuja massamuisteja, kuten teollisuusstandardin mukaisia yli 512 MB:n partitioita. Yhteistä näiden menneen ajan ja modernien **kotitietokoneiden** kanssa on se, että juuri käyttöjärjestelmän välityksellä käyttäjä "kommunikoi" **kotitietokoneensa** kanssa. Käyttöjärjestelmä on linkki laitteen ymmärtämän konekielen ja ohjelmien ohjelmointikielen välillä; se kykenee myös muuntamaan internaalisista käskyjonoista rakennettujen eräajojen ASCII-merkeistä koostuvan kielen konekiellelle.

Nyttemmin (AT-tasoisten **kotitietokoneiden** myötä) suorittimet [prosessorit] ovat parantuneet ja nopeutuneet niin, että jo 80286 -tasolla voidaan puhua 12 MHz:n kellotaajuudesta ja ROM [Read Only Memory] -pohjaisten ohjelmien (BIOS, Video BIOS) suorittamisesta nopeasta 16 bitin RAM [Random Access] -muistista (SHADOW RAM). Kyse on toisaalta suorittimien yleisestä nopeuden lisääntymisestä ja muistinkäytön moninaisen hallinnan mahdollisuuksista. Varhaisemmilla **kotitietokoneilla** DOS:n käytettävissä olevan keskusmuistin laajuus oli kiinteä ja vailla varsinaisia muokkausmahdollisuuksia - esimerkiksi kiintolevyjen parametrien valinnan ja määrittelyjen suhteen. Käytännössä RAM voi tuolloisilla DOS-käyttöjärjestelmillä teoreettisesti olla 640 Kb - mikä ei riittänyt kuitenkaan muuttuneissa olosuhteissa. Itse asiassa ohjelmien ajamiseen käytettävissä oleva tila ei ole tuo maksumaallinen 640 Kb:n arvo, koska osan siitä varaa itselleen käyttöjärjestelmä, ja varsin pian useat

hyötyohjelmista on kehitetty sellaisiksi, että ne vaativat yli 500 Kb:n keskusmuistitilaa ja silloinkin tietyn määrän EMS-muistia sen lisäksi, kuten esimerkiksi VENTURA-julkaisuohjelmisto. Nykyään, kun massamuisteissa olevia tiedostoja kyetään kompressoimaan, on tavallista, että myös käyttömuistista osia voidaan tiivistää, jolloin siinä voidaan ajaa suurempia ohjelmia, jolloin tarvitaan luonnollisesti kääntäjää, joka muuntaa kompressoitun tiedon tarvittaessa tavanomaiseksi, ja päinvastoin.

Seuraavassa eräs hahmotelma 286-suorittimella varustetun **kotitietokoneen** muistista, jossa keskusmuisti [Conventional Memory] muodostaa 640 Kb, ja siitä seuraava UPPER MEMORY-osio sisältää muistit aina 1 Mb:iin asti, mikä muodostuu ROM- blokeista ja EMS muistista.

Conventional memory ->>640K  
0h 10000h 20000h 30000h 40000h 50000h 60000h 70000h

**0K 64K 128K 192K 256K 320K 384K 448K**  
 @@@@  
 @@@@ @ xxxx \*\*\*\*\* xxxxxxxxxxxxxx  
**512K 576K 640K 704K 768K 832K 896K 960K 1MB**

<MO>80000h 90000h A0000h B0000h C0000h D0000h E0000h F0000h  
 Upper memory <<-640K ->>1MB

Key: @@@ =RAM xxx =ROM \*\*\* =EMS

Seuraavassa ylläolevassa kaaviossa skemaattisesti kuvattujen erilaisten muistien osoitteet ja koot.

Address	Owner	Size	Type
0:0000	-----	A0000h, 655 360	----- RAM-
C000:0000	-----	8000h, 32 768	-----ROM-
CC00:0000	EMS	10000h, 65 536	----- EMS memory
E000:0000	-----	20000h, 131 072	-----ROM-

Varsinainen 640 Kb:n käyttömuisti [RAM] voi muodostua sisällöllisesti hyvinkin monenlaisista muistiaineksista, joilla kullakin on oma enemmän tai vähemmän pysyvä osoitteensa. Käyttöjärjestelmä sisältää komentotulkin (COMMAND.COM)

lisäksi myös järjestelmätiedostot, joista I/O tarkoittaa INPUT/OUTPUT -toimintojen ohjausta, kuten mm. **kotitietokoneen** monitorin ja oheislaitteiden kytkeytymistä "keskusyksikön" toimintaan. Käyttöjärjestelmä ladataan koneen käynnistyksen yhteydessä koneen keskusmuistiin. MSDOS:n tapauksessa järjestys muistissa on seuraavanlainen:

0000:0000

Keskeytyksvektoritaulu [Interrupt Table]

XXXX.0000

IO.SYS - MSDOS vuorovaikutus HARDWAREEN.

XXXX:0000

MSDOS.SYS - MSDOS 21H-Funktiot, Muistivaraukset (levymuistit l. Disk Buffers, määrältään yleensä 10-20) [BUFFERS] ja asennetut oheislaitteet [DEVICE DRIVERS]. Sekä muistinvaraukset -että oheislaitteet asetetaan CONFIG.SYS -tiedostoon komentoriveiksi, esim.:

BUFFERS=20

DEVICE=ANSI.SYS

XXXX:0000

Komentotulkin residenttiosa pysyy muistissa [sijaitsee heti MSDOS-järjestelmän ja sen data-alueen jälkeen. Tämä osio sisältää rutiinit, joilla prosessoidaan 23H keskeytyksiä [Control-C EXIT -osoitteet] ja fataalisen virheen kohdatessa tapahtuvaa 24H keskeytystä, kuten myös rutiinit, joilla ladataan komentotulkin transienttiosa, mikäli tarpeen. Kaikki tavanomainen MSDOS virheenkäsittely tapahtuu tässä osassa COMMAND.COM tiedostoa. Tämä osa sisältää myöskin tiedotukset virheen laadusta ja ABORT, RETRY or IGNORE -viestien prosessoinnin.

XXXX:0000

Alustusosa [Initialization Part], joka ottaa käynnistyksen yhteydessä kontrollin; se sisältää AUTOEXEC.BAT tiedoston pohjalta tapahtuvan alustuksen prosessoinnin. Alustusosa määrittää myöskin segmenttiosoitteet, joista ohjelmat voidaan ladata. Tämä osa komentotulkia jää syrjään tehtävänsä suoritettuaan.

XXXX:0000



## USER STACK FOR .COM FILES [256 bytes]

XXXX:0000

Transienttiosa komentotulkkia ladataan (käyttö) muistin yläalueelle; se sisältää kaikki internaaliset komennot ja BATCH-tiedostojen prosessoinnin.

Transienttiosa tuottaa myös järjestelmän tasolla kohdistimen muodon [PROMPT], lukee komennot näppäimistöltä [tai BATCH-tiedostosta], ja aiheuttaa sen, että ne toteutetaan.

<MS>Vaikka käyttäjä luulisikin "kommunikoivansa" suoraan käyttöjärjestelmän ja sen konekielisen maailman kanssa, esimerkiksi kirjoittaessaan eräajoja .<BS>BAT<MS> [BATCH-prosessointi], ei hän näin tee, vaan käyttöjärjestelmä kääntää käyttäjän sanallisin määrittein antamat käskyt <BS>konekielelle<MS>, ja vain siten käyttäjä kommunikoi näitä tiedostoja muodostaessaan käyttöjärjestelmän kanssa. Käyttäjän suhde koneeseen on epäsuora.

Eksternaalisille komentoille tämä samainen osa muodostaa komentorivin ja sisältää EXEC järjestelmäkutsun 4BH, jolla ohjelmille [.COM tai .EXE tiedostoille] saadaan lataus ja siirtokontrolli; kun jokin eksternaalinen komento kirjoitetaan tai jokin ohjelma ajetaan EXEC järjestelmäkutsun avulla määrittää DOS alimman mahdollisen vapaan muistiosoitteen ohjelman aloituskohdaksi - tätä kohtaa sanotaan ohjelmasegmentiksi. Ensimmäiset 256 bittiä ohjelmasegmentistä on asennettu jo asennettaessa EXEC järjestelmää, ja ohjelma ladataan niin, että loppuosaa siitä seuraa tätä ennaltamääritettyä blokkia. Sellainen .EXE ohjelmätiedosto, jonka MINIALLOC ja MAXALLOC on asetettu nolnaan ladataan niin korkealle muistialueella kuin mahdollista.

**Kotitietokoneen** muistirakennetta voidaan skematisoida monin eri tavoin, ja eri muistilajeilla on erilaiset yhteydet käyttöjärjestelmän toimintaan ja ohjelmiston tapaan käyttää muistia. Uusilla käyttöjärjestelmillä muistinkäyttöä voidaan tehokkaasti ohjata. Hyvin yleisesti muistin eri lajeja voidaan nimetä seuraavasti [huomattakoon, että kyse on fyysisistä yksiköistä, jotka voidaan ottaa käyttöön konfiguroimalla **kotitietokonetta** ohjelmallisesti]:

DOS-muistiksi,

Lisätyksi [EXTENDED] muistiksi

Laajennetuksi [EXPANDED] muistiksi,

EMS -muistiksi,

SHADOW-RAM -muistiksi, ja

CACHE-muistiksi

AT -tason **kotitietokoneissa** on laitteen sisällä ohjelmoitava tai valikkopohjainen CMOS -muistiyksikkö, joka voidaan "avata" laitetta käynnistettäessä tietyllä näppäinyhdistelmällä. On myös olemassa erillisiä ohjelmia [kuten Disk Manager], joilla voidaan tehdä tiettyjä asennuksia. Muistiyksikössä on tiedot mm.:

päivämäärästä  
[päivämääräasetys, kellonaika]

kalvolevyasemista  
[levyasemien määrä ja niiden koko]

kiintolevyistä  
[kiintolevyn/ levyjen parametrit]

laajennetusta muistista  
[laajennetun muistin laji ja määrä]

tunnusnasta  
[vaaditaan laitetta käynnistettäessä]

jne.

**Kotitietokoneen** muistia on voitu siis laajentaa luomalla uusia muistikategorioita, jotka eivät ole riippuvaisia DOS:n rajoituksista [Tämä on käytännöllistä esimerkiksi käytettäessä OS/2 -käyttöjärjestelmää]. Osa muistista ei ole kuitenkaan "oikeaa" siinä mielessä, että se muodostuu esimerkiksi sähköisistä levyasemista [RAMDISK], joiden kapasiteetti on laaja, ja joihin siirrettyinä ohjelmat toimivat nopeasti, koska signaalit siirtyvät sähkönsä kulkunopeudella. Nämä levyasemat vastaavat toiminnaltaan fyysisiä levyasemia. Ohjaamalla esimerkiksi julkaisuohjelmien virtuaalitiedostot tämännäntapaisiin muisteihin saadaan niille lisää nopeutta pelkästään siksi, että ohjelman käyttäessä kiintolevyn fyysistä muistikapasiteettia sen toiminnot ovat huomattavasti hitaampia toistuvine "fyysisine" lukemisineen ja talletuksineen. Julkaisuohjelmien järjestelmätiedostot kynenevät käyttämään EMS-muistia, mikäli sitä on saatavissa.

Käytettäessä 80386 prosessoria saavutetaan 32 bitin RAM-muisti ja voidaan

käyttää sille sovitettuja uusia ohjelmia, jotka eivät toimi 80286-prosessorilla varustetuissa laitteissa. Pienimmillään laitteen kellotaajuus on n. 16 MHz ja keskusmuisti voidaan jakaa halutulla tavalla. 80386 jaettava RAM-muisti voidaan yhdistää keskusyksikköön [Central Processing Unit] jopa 32 -bittisellä väylällä jopa ilman odostustiloja kellotaajuuden ollessa 20 MHz.

Mikäli laitteessa käytetään laajoja julkaisujärjestelmiä [DESKTOP], tai CAD/CAM -ohjelmistoja [CAD = Computer Aided Designing; CAM = Computer Aided Manufacturing], tai se on ajateltu liitettäväksi suurten lähiverkkojen tiedostopalvelijaksi, voidaan siihen ajatella mainittua 32 bitin RAM-muistia, 33 MHz:n kellotaajuutta, CACHE välimuistia, mikrokanavaa, ja siihen voidaan ajatella asennettavaksi numeerinen 80387 rinnakkaisprosessori, ja 24 megatavuun laajennettua keskusmuistia, joka lohkotaan halutulla tavalla.

Käyttöjärjestelmä on aina konfiguroitava, mikäli laitteiston peruskokoonpanoon lisätään uusia välineitä, tai merkistöä laajennetaan [ANSI], tai kytkentöjä muutetaan, tai näppäimistöä kansallistetaan, tai grafiikkakorttia vaihdetaan jne., mikä tarkoittaa mm. sitä, että uudet oheislaitteet, ja jotkut ohjelmista vaativat käyttöjärjestelmän mukautumista niiden läsnäoloon, jotta ne ylipäättään toimisivat: onhan käyttöjärjestelmä eräänlainen tulkki ohjelmien ja laitteiston [HARDWARE] välillä. DOS on siten sekä fysikaalinen, että myös käyttäjään suuntautunut. Tiedostot IO.SYS, MSDOS.SYS, ja COMMAND.COM juurihakemistoon sijoitettuina riittävät yksin aktivoimaan **kotitietokoneen**, vaikka levykkeellä, tai kiintolevyllä ei olisikaan muita käyttöjärjestelmätiedostoja, mutta tällöin kotitietokone käynnistyy oletusarvoin, ja ei aktivoi oheislaitteita tai sovita käyttöjärjestelmää niiden mahdollisille ohjauskorteille; siten useimmat käyttöjärjestelmän eksternaaliset ohjelmat ja sovellusohjelmistot voidaan sijoittaa sivuun juurihakemistosta omaan hakemistoonsa.

Käyttöjärjestelmän taso on varsin ratkaiseva, sillä niitäkin on monenlaisia - sallien käyttäjille erilaisia vapausasteita. Useat hyötyohjelmista vaativat tietyn tasoisen käyttöjärjestelmän toimiakseen. Käytetyn DOS -version olisi oltava lisäksi homogeeninen kokonaisuus, mikä tarkoittaa sitä, että eri tason järjestelmäin eksternaalisia käskyjä ei tulisi pitää yhdessä, koska kukin kokonaisuus on suunniteltu toimimaan parhaiten juuri sen "version" puitteissa, joihin ne on aluperin asetettu. Jokaisen laitteen mukana on yleensä jokin käyttöjärjestelmä, jota tulisi käyttää.

<MS>Joissakin tapauksissa uusi käyttöjärjestelmä voidaan kopioida sellaisenaan levykkeiltä kiintolevylle entiseen alihakemistoon, kun sieltä on tuhottu ensin entiset eksternaaliset ohjelmat ja varmistettu, että järjestelmätiedostot ovat siirrettävissä kiintolevyn juurihakemistoon niin, että <BS>kotitietokone<MS> käynnistetään kalvovelyltä levyasemasta A:, jonka jälkeen järjestelmätiedostot siirretään juurihakemistoon komennolla SYS C:.

Siirto onnistuu, mikäli järjestelmätiedostojen koko on sama sekä vanhassa -että uudessa käyttöjärjestelmässä, koska useimmissa DOS-käyttöjärjestelmissä järjestelmätiedostot sijaitsevat aina kiintolevyllä kaikkein lähinnä alkulatauslohkoa. Huomattakoon, että tämä proseduuri ei tuhoa kiintolevyn partitiointitietoja, mutta saattaa aiheuttaa sen, että näppäimistö- ja muut määritykset eivät toimi mahdollisesti aiemmin määritellyissä AUTOEXEC.BAT ja CONFIG.SYS konfigurointikommentoreilta annettuna, koska uuden käyttöjärjestelmän internaalisten ja eksternaalisten komentojen kytkimet saattavat poiketa aiemman käyttöjärjestelmän vastaavista, ja komentojen muoto esimerkiksi näppäimistöasennuksessa saattaa olla erilainen. Ilman käsikirjaa on hyvin vaikeaa selvittää, mikä on konfigurointitiedostojen komentojen oikea muoto. Mikäli järjestelmätiedostojen siirrossa ilmenee ongelmia, olisi käytettävä seuraavankaltaista menetelmää. Käynnistetään <BS>kotitietokone<MS> kalvovleyasemasta A:, uudella, siirrettäväksi aiotulla käyttöjärjestelmällä, ja kun tämä on tehty käynnistetään uuden käyttöjärjestelmän eksternaalinen ohjelma FDISK, jolla tarkastetaan, kykeneekö uusi käyttöjärjestelmä lukemaan kiintolevyllä mahdollisesti vallitsevan partitioinnin. Mikäli kykenee, käynnistetään erikseen saatavissa oleva ohjelma Norton Disk Doctor, joka käynnistetään ja valitaan siitä opitio, jolla kiintolevy voidaan tehdä käynnistyväksi uudella käyttöjärjestelmällä - ohjelma muodostaa järjestelmätiedostoille tarvittavan tilan, ja suorittaa siirron.

Käyttöjärjestelmän tasolla opetoitaessa pätee, että jos yrittää ajaa jonkun käyttöjärjestelmän eksternaalisia käskyjä sitä alempain järjestelmän käyttöjärjestelmässä, saattaa laitteen monitorissa näkyä ilmoitus:

Incorrect DOS-Version

Tällainen näyttö on mahdollinen, jos koetetaan ajaa esimerkiksi MSDOS 3.30 -eksternaalisia käskyjä MSDOS 2.11 -käyttöjärjestelmässä. Käytännössä - mikäli tätä rajoitusta ei huomioida - ei aiheudu niin suuria ongelmia, kuten luulisi. Käyttöjärjestelmän vaihtaminen ei vaatisi kaikkien hallussa olevain kalvovleykkeiden alustuksen [formatoinnin] uusimista, ja tiedostojen kopioimista alustetuille levykkeille, jos kiintolevyllä olisi hankittu MSDOS 4.01. Ainut, mikä olisi huomioitava - kun tarpeellinen asennus uudelle käyttöjärjestelmälle olisi tehty - olisi se, että pyrittäisiin toimimaan kunkin istunnon aikana tämän uuden järjestelmän ympäristössä, minkä voi tehdä käyttämällä SET tai SHELL -komentoja, joilla komentotulkin paikka määritellään, ja josta se on tarvittaessa löydettävissä. Vallitsevan käyttöjärjestelmän tason voi tarkistaa yksinkertaisesti kirjoittamalla internaalisen käskyn VER, jolloin komentotulkki tutkii keskusmuistissa olevan aktiivisen järjestelmän version, ja tulostaa sen esimerkiksi:

Ver. 4.01

MSDOS-käyttöjärjestelmä voidaan esittää myös supistetussa muodossa niin, että huomioidaan pelkästään varsinaiset järjestelmätiedostot:

IO.SYS tai IBMBIO.SYS tms. (INPUT/OUTPUT)

Liittää **kotitietokoneen** ohjelmiston [software] sen laiteympäristöön [Hardware], ja on välittävänä linkkinä molempiin. IO huolehtii **kotitietokoneesi** Input/output järjestelyistä.

MSDOS.SYS tai IBMDOS.SYS tai IBMDOS.COM tms.

Tämä on sitä, mitä tarkoittaa MS-DOS -käyttöjärjestelmä; sillä on roolinsa mm. osoitteiden löytymisessä, ja **kotitietokoneen** keskusmuistiin syötetyn ohjelman toimivuudessa.

COMMAND.COM

Komentotulkki, joka prosessoi annettujen käskyjen toteutumista; siihen ovat kirjoitetut kaikki käytettävissä olevat internaaliset käskyt. Tällaisia käytetään mm. luotaessa BAT -tiedostoja. Varsinaisia "osoitekäskyjä" BAT -tiedostojen käskyt eivät siis ole, tai konekieltä. Komentotulkkiin on kirjoitettu, että sellaiset tiedostot, joiden nimen tunnusosana on **.COM** -tai **.EXE** -ovat ohjelmina ajettavia EXEC järjestelmäkutsujen kautta [COM = COMMUNICABLE; EXE =EXECUTABLE].

BAT -tiedostoista juurihakemistoon [ROOT] sijoitettu AUTOEXEC.BAT on sellainen, jonka samaan hakemistoon sijoitettu komentotulkki COMMAND.COM järjestelmätiedostoiheen ajaa aina laitetta käynnistettäessä, mikäli sellainen on olemassa. Kyseisiä AUTOEXEC.BAT eräajoja ei kerrallaan käynnistävällä levyllä- tai levykkeellä voi olla enempää kuin yksi. Komentotulkki toteuttaa varsinaiset ohjelmat yleisesti hierarkisesti siten, että samannimisistä ohjelmätiedostoista ajetaan ensisijaisimmin **.COM** -tunnuksinen ohjelma, ja jollei sellaista ole, niin sitten se ajaa **.EXE** -tunnusosalla varustetun samannimisen tiedoston - mikä on syytä muistaa muunnettaessa **.COM** tiedostoja **.EXE** tiedostoiksi purettaessa esimerkiksi kopiointisuojuuksia, tai päivitettäessä ohjelmistoja vanhojen ohjelmaversioiden ollessa muotoa **.COM** ja uusien muotoa **.EXE**. Hierarkian alimmaisena ovat muut BATCH-eräajotiedostot .BAT, joita ajetaan laitteen käynnistyksen jälkeen esimerkiksi käynnistettäessä varsinaisia ohjelmia. Mikäli AUTOEXEC.BAT eräajoon on kirjoitettu esimerkiksi APPEND ja PATH -määritykset, hakee komentotulkki

niiden osoittamista hakemistoista ohjelmanimiä, jos ne annetaan käyttöjärjestelmän tasolla. Myös muilla tämäntapaisilla määritteillä, kuten ASSIGN ja JOIN voidaan ohjata komentotulkin toimintaa ohjelmahaun suhteen.

### 1.2.1.1. AUTOEXEC-ERÄAJO

AUTOEXEC.BAT

Alkulatauksessa [Boot] aktivoituva eräajo, jolla voidaan ladata keskusmuistiin mm. polkumäärittäminen [path], SET -asetukset, komentovihje [PROMPT], näppäimistöohjain [KEY], hiiri , jne.

AUTOEXEC eräajo voi saada esimerkiksi seuraavanlaisia sisältöjä:

```
ECHO OFF
PROMPT $e[0m$p$g
PATH C:\;C:\SYST\DOS;C:\BAT;C:\SYST;C:\DRIVERS;
APPEND C:\;C:\SYST\DOS;C:\BAT;C:\SYST;C:\DRIVERS;
FRECOVER /SAVE
KEYB SU
MOUSE
PCTOOLS/ R64
CD C:\SYST
_SHELL_
```

-Ylin komentorivi [ECHO OFF] aiheuttaa sen, etteivät siitä seuraavat komennot eräajossa tulostu näytölle ellei anneta sen kumoavaa käskyä [ECHO ON] tai anneta käskyä, jonka seurauksena näytölle tulee jokin tietty tiedotus [ECHO \*\*\*KAIKKI HYVIN\*\*\*],

-siitä alempi komentorivi [PROMPT] aiheuttaa sen, että kohdistin saa käyttöjärjestelmän tasolla tietyn muodon, koon tai värin, tai tausta saa jonkin oletusarvosta poikkeavan värin; kun näppäimistöllä kirjoitetaan jokin merkki, tulee se näytölle kohdistimen osoittamaan paikkaan,

-siitä alempi komentorivi [PATH] luo käyttöjärjestelmän oletusarvosta [kuten A:\ ja C:\] poikkeavan uuden hakemistopolun, jota käyttöjärjestelmää hyödyntää annettaessa käyttöjärjestelmän tasolla ohjelmakäskyjä muotoa **.COM**, **.EXE**, ja **.BAT**.,

-siitä alempi komentorivi [APPEND] luo saantipolon, jota käyttöjärjestelmä

hyödyntää hakiessaan ohjelmien tarvitsemia aputiedostoja; ohjelmatiedostot aputiedostoineen saavat tällöin sijaita eri hakemistoissa; ohjelmat eivät toimi, mikäli niiden tavoitettavissa ei ole joko kaikkia tai erikseen määrättyjä tiedostoja,

-siitä alempi komentorivi [FRECOVER] on erikseen saatavissa oleva ohjelma, joka luo (kiintolevyille) tiedoston, joka mahdollistaa tiedostojen tarkan palauttamisen levyille, mikäli laitteeseen tulee jokin häiriö; joissakin käyttöjärjestelmissä on vastaavat ominaisuudet omaavia eksternaalisia komentoja, jotka tallentavat ao. tiedot,

-siitä alempi komentorivi [KEYB] aktivoi laitteeseen suomalaisen näppäimistön tapauksissa, joissa käytetään vain yhtä **kotitietokonetta** erillisenä, eikä esimerkiksi kytkettynä verkkoon,

-siitä alempi komentorivi [MOUSE] aiheuttaa sen, että hakemistopolulla sijaitseva erikseen saatavilla oleva Microsoft-hiiri asennetaan muistiresidentiksi, jonka seurauksena on se, että hiirtä voidaan käyttää niissä ohjelmissa, joissa tälle oheislaitteelle on tuki; Logitechin hiirellä vastaava asennus voisi olla muotoa MOUSE SER 1, ja tämän hiiren etuna on se, että se voidaan poistaa muistista niin haluttaessa,

Huom!

Hiiri [Mouse] aktivoidaan yleensä joko kirjoittamalla konfigurointitiedostoon [CONFIG.SYS] hiiren laiteohjainkomentorivi, esimerkiksi:

```
DEVICE=MOUSE.SYS
```

tai kirjoittamalla käynnistävään eräajoon (AUTOEXEC.BAT) mainittu komentorivi:

```
MOUSE
```

tai jokin muu hiiren käynnistyskomento, mikäli hiiri on muu kuin Microsoft Mouse.

-siitä alempi komentorivi [PCTOOLS] aiheuttaa sen, että erikseen saatavissa oleva ohjelma asennetaan muistiresidentiksi niin, että se voidaan käynnistää milloin tahansa - esimerkiksi käytettäessä tekstinkäsittelyohjelmaa,

-siitä alempi komentorivi [CD] aiheuttaa sen, että aktiivi hakemisto vaihtuu juurihakemistosta tietyksi, käskyllä määritellyksi alihakemistoksi,

-kaikkein alin rivi [ \_SHELL\_ ] kytkee erikseen saatavilla olevan käynnistysvalikon, joka toimii käyttöjärjestelmän alaisuudessa, mutta jonka kautta käyttäjä työskentelee.

### **1.2.1.2.CONFIGUROINTITIEDOSTO**

.SYS -tiedostoista CONFIG.SYS on sellainen, jonka komentotulkki ajaa aina laitetta käynnistettäessä, ja se sijoitetaan juurihakemistoon:

CONFIG.SYS

Alkulatauksessa aktivoituva laiteohjain-konfigurointitiedosto, jolla asennetaan oheislaitteiden kytkentöjä.

Tiedosto voi näyttää seuraavanlaiselta, kun sillä asennetaan ANSI ajuri ja oheislaitteena laajennettu RAM levyasema:

```
BUFFERS=25  
FILES=25  
DEVICE=ANSI.SYS  
DEVICE=EMS.BIN /P1 P2 P3 S3/  
DEVICE=RAMDISK.BIN /6000/
```

-tähän nimenomaiseen tiedostoon voidaan tallentaa, kirjoittaa, tai luoda myös muita määrittämiä, kuten tiedot muista oheislaitteista, ja erityisesti mainittakoon tässä yhteydessä se, että DISK MANAGER ohjelman tapaisilla välineillä siihen voidaan tallentaa tiedot kiintolevyistä ja tavasta, joilla ne on lohkottu. Tämä DM ohjelma on keino, jolla käyttäjä voi suoraan muuttaa BIOSIN sisältämiä tietoja ja tallentaa ne CONFIG.SYS tiedostoon. Jos sattuisi niin, että "yhteys" kiintolevyyn menetettäisiin syystä tai toisesta, niin se voidaan "rakentaa uudelleen" tiedostossa CONFIG.SYS olevien tietojen perusteella.

### **1.2.1.3.MSDOS KÄYTTÖJÄRJESTELMÄN INTERNAALISET KOMENNOT**



Seuraavassa esittelen lähinnä Microsoftin DOS-käyttöjärjestelmän komentoja. On todennäköistä, että komennot, ja niiden esiteltyt kytkimet ovat käyttökelpoisia DOS-käyttöjärjestelmissä yleensäkin, vaikka joissakin kytkimiä olisi enemmänkin kuin tässä on esitelty.

### **BREAK .....**

Asettaa CONTROL-C katkaisuvälilyyden, joka toimii eräissä ohjelmissa ja erityisesti eräajojen yhteydessä käytettynä. Tavallisesti DOS tarkistaa joka kerta kun näppäimistöllä kirjoitetaan, onko CONTROL-C -näppäinyhdistelmää painettu; mikäli on valittu BREAK ON -vaihtoehto, ulottuu DOS:n suorittama tarkistus myös sentapaisiin prosesseihin, kuten lukemiseen levykkeiltä, tai niille kirjoittamiseen, sekä ohjelmien toiminnan pysäyttämiseen tiettyjen edellytysten vallitessa. Mikäli on valittu vaihtoehto BREAK OFF, ulottuu tarkistus ainoastaan näppäimistöille, näytölle ja kirjoittimen toiminnan pysäyttämiseen.

BREAK [ON] tai [OFF]

### **CHDR (CD)..**

Komento vaihtaa oletushakemiston. Esimerkiksi komentovihjeen C:\ vallitessa komento tuo näyttöön samalla uuden oletushakemiston nimen toimittaessa käyttöjärjestelmän tasolla. Komentoa voidaan kuitenkin hyödyntää myös muulla tavoin, kuten esimerkiksi haluttaessa käynnistää WordPerfect tekstinkäsittelyohjelma [joka sijaitsee esimerkiksi hakemistossa C:\PROGE\WP50 niin, että haluttu hakemisto, jossa tekstit ovat, kuten C:\DATA\TEKSTIT, asetetaan oletushakemistoksi, jolloin se on automaattisesti käytössä käynnistettäessä tekstinkäsittelyohjelmaa. Tämän toteuttamiseksi luodaan eräajo .BAT, joka luo tilanteen, jossa tekstinkäsittelyohjelma käynnistetään hakemistosta, jossa sillä kirjoitetut tekstit sijaitsevat. Eräajoon kirjoitetaan tällöin komentorivit:

```
CD C:\DATA\TEKSTIT
C:\PROGE\WP50\WP
```

CD

esimerkkejä:

```
CD C:\HAKEMIST.001
CD E:\HAKEMIST.001\HAKEMIST.002
CD A:\HAKEMIST.005
```

CD..  
CD\

## **CLS .....**

Komento tyhjentää näytön käyttöjärjestelmän tasolla (komento ESC[2J), jolloin kohdistin siirtyy näytössä samalla vasempaan yläreunaan. Komentoa voidaan käyttää myös eräajoissa .BAT sen komentoriveinä.

CLS

## **COPY .....**

Komento monistaa komennon yhteydessä spesifioidun tiedoston sisällön, tai useampia tiedostoja. Käyttöjärjestelmä estää tavallisesti kopioinnin, mikäli kopiointia yritetään samassa hakemistossa samalle tiedostonimelle. Eräissä uudemmissa käyttöjärjestelmissä voidaan tiettyä kytkintä käyttäen COPY -komentoa monistaa haluttu tiedosto myös samannimisen, jo olemassaolevan tiedoston päälle. Mikäli kopioidaan levyasemasta toiseen, kopioituu tiedosto (tai tiedostot) tavallisesti siellä olevien tiedostojen päälle riippumatta siitä, onko siellä samannimisiä tiedostoja kuin mitä kopioitava tiedosto tai tiedostot ovat. Eräissä uudemmissa käyttöjärjestelmissä COPY -komentoantoon voidaan liittää kytkin, joka saa aikaan samannimisyyden tarkistuksen. Suojatut tiedostot tai järjestelmätiedostot eivät kopioidu tavallisesti käytettäessä COPY -komentoa. Uudemmissa käyttöjärjestelmissä myös niiden monistus on mahdollista käyttäen tiettyä kytkintä. Kopiointisuojatut levykkeet eivät ole mahdollisia yleensä kopioida tällä komennolla, mutta on olemassa runsaasti erillisiä ohjelmia, joiden avulla kopiointisuojatut levykkeet on helppo monistaa, ja myös ohjelmia, joiden avulla ohjelmia voidaan käyttää kiintolevyllä niissä olevasta suojauksesta huolimatta.

COPY [<<d:>>] <<filename>>[/V]/[A]/[B]

esimerkkejä:

COPY A: TIEDOSTO.001 TIEDOSTO.002  
COPY \*.001 A:  
COPY \*.001 B:  
COPY \*.\* A:  
COPY \*.\* B:  
COPY TIED????.\* A:  
COPY TIED????.001 B:  
COPY B\HAKEMIST.BBB\\*. \* A:\HAKEMIST.AAA

**TIME .....**

Komento esittää vallitsevan kellonajan käyttöjärjestelmän tasolla. Komennolla voidaan myös muuttaa mahdollista virheellistä kellonaikaa.

TIME [<<tunnit>>: <<minuutit>>]

**CTTY .....**

Komento muuttaa konsolia [TTY] jolla annetaan komentoja DOS -käyttöjärjestelmälle. Tämän komennon käytön yhteydessä on syytä sivuta "kiellettyjen tiedostonimien tarkentimien" käyttöä. Kiellettyjä tarkentimia ovat .PRN, .AUX, ja .CON. ja .NUL, koska DOS käyttää niitä oman toimintansa säätelyssä. Esimerkiksi komennolla CTTY AUX muunnetaan vallitsevan laitteen I/O -ohjaus AUX-porttiin, johon on kiinitettynä toinen terminaali.

CTTY <<|<L>aite>>

esimerkkejä:

CTTY AUX  
CTTY CON  
CTTY VT52

**DATE .....**

Komento esittää vallitsevan päivämäärän käyttöjärjestelmän tasolla. Komennolla voidaan myös muuttaa virheellistä päivämäärää. Mikäli käyttöjärjestelmää on konfiguroitu COUNTRY -asetuksella, on vuoden, kuukauden ja päivämäärän järjestys erilainen kuin oletustilassa.

DATE [<<kk>>; kuukausi=1-12; <<pp>>; päivä=1-31; <<vv>>; vuosi=80-99

**DEL .....**

Komento tuhoaa nimetyn tiedoston käyttöjärjestelmän tasolla, tai useampia nimettyjä tiedostoja samanaikaisesti (ERASE). Komennon yhteydessä voidaan käyttää korvausmerkkejä.

DEL [<<d:>>] <<polkunimi>>

esimerkkejä:

DEL A:TIEDOSTO.001  
DEL \*.\*

```
DEL *.001
DEL\HAKEMIST.001\TIEDOSTO.001
DEL A:*. *
DEL A:\HAKEMIST.AAA\*. *
```

### **DIR .....**

Komento esittää oletus- tai halutussa hakemistossa olevat tiedostonimet. Komennon yhteydessä voidaan käyttää korvausmerkkejä.

```
DIR [<<d:>>] [<<polkunimi>>] [/P]/[W]
```

esimerkkejä:

```
DIR
DIR A:
DIR *. *
DIR TIEDOSTO.001
DIR *.001
DIR A:\HAKEMIST.001\TIEDOSTO.001
DIR/P
DIR/W
```

### **EXIT .....**

Komento antaa sinun poistua komentotulkin [COMMAND.COM] alaisuudesta, ja palata aiemmalle tasolle, mikäli sellaista on. Tämä vastaa tilannetta, jossa otetaan tilapäisesti käyttöön toinen komentotulkki, ja luovutaan sen käytöstä. Komentoa käytetään myös ajettaessa jotakin hyötyohjelmaa ja haluttaessa palata käyttöjärjestelmän tasolle antamalla komentotulkin EXEC [järjestelmäkutsu 4BH]. Kun sitten kirjoitetaan EXIT palataan takaisin hyötyohjelmaan

```
EXIT
```

### **MKDIR (MD)..**

Komento muodostaa uuden hakemistonimen; yhden kerrallaan ja niin, että edetään hakemistopuun rakentamisessa rungosta oksiin päin. Eräs tämän komennon käyttötarkoitus on eräiden ohjelmistoasennusten esitoimenpide. Jos esimerkiksi suoritetaan tietyn ohjelman asennus käyttäen esimerkiksi asennuslevykkeellä olevaa INSTALL ohjelmaa, luodaan etukäteen esimerkiksi kiintolevylle hakemisto MOS, jota oletushakemistona käyttäen asennus sitten suoritetaan:

MD MOS  
CD MOS  
A:INSTALL

MD [<<d:>>] <<polkunimi>>

esimerkkejä:

MD HAKEMIST.001  
MD\HAKEMIST.001\HAKEMIST.002  
MD A:\HAKEMISTO.AAA

### **RMDIR (RD)...**

Komento poistaa hakemistonimen; yhden kerrallaan; hakemiston tulee olla tyhjennetty (tiedostojen tuhottuja) ennen poistoa. Mikäli tiedostojen tuhoamisessa ei käytetä tehokkaita työkaluja, saattaa hakemistoihin jäädä esimerkiksi piilotettuja [hidden] tai kirjoitussuojattuja tiedostoja, joita ei voida tavallisesti hävittää käyttäen DEL -komentoa. Eräissä uudemmissa käyttöjärjestelmissä on mukana eksternaalisia ohjelmia, joiden avulla voidaan hävittää kaikki poistettavissa hakemistoissa olevat tiedostot.

RD [<<d:>>] <<polkunimi>>

esimerkkejä:

RD HAKEMIST.001  
RD\HAKEMIST.001\HAKEMIST.002  
RD A:\HAKEMIST.AAA

### **PATH .....**

Komento asettaa halutun hakemistopolun komennoille; annettaessa jokin ohjelmanimi käyttöjärjestelmä tarkistaa annetun hakemistopolun etsiessään ohjelmaa. Mikäli käyttöjärjestelmän tasolla annetaan pelkkä PATH -komento, tulee näyttöön vallitseva polkumäärittäminen. Mikäli annettaisiin komento PATH; poistuisi vallitseva polkumäärittäminen kokonaan.

PATH [[<<d:>>] [<<polkunimi>>]; [<<d:>>] [<<polkunimi>>] ...]

esimerkkejä:

PATH

PATH;  
PATH C:\HAKEMIST.001;C:\HAKEMIST.002

**PROMPT** .....

Komento muotoilee kohdistimen esitystapaa käyttöjärjestelmätasolla.

PROMPT <<teksti>>

esimerkkejä:

PROMPT

PROMPT p\$g\$

PROMPT t\$d

PROMPT \$v\$n

PROMPT \$e[7m\$n:\$e[m

**REN** .....

Komennolla annetaan jollekin tiedostolle toinen nimi.

REN [<<d:>>] <<polkunimi>> <<polkunimi>>

esimerkkejä:

REN TIEDOSTO.001 TIEDOSTO.002

**SET** .....

Komentoa käytetään muuttaessa ympäristöä, joka muodostuu sarjasta ASCII ketjuja (vähemmän kuin 32K), joilla on aina muoto NIMI=parametri; komentoprosessorin rakentama ympäristö koostuu ainakin levyaseman ja komentotulkin aseman määrittelevästä ketjusta, esimerkiksi COMSPEC=C:\COMMAND.COM; komentoa SET käytetään esimerkiksi asetettaessa **kotitietokoneeseen** uusi haluttu ensisijainen looginen levyasema, jolta käyttöjärjestelmä on löydettävissä; tämän lisäksi voidaan niin haluttaessa antaa tietokoneistunnon kestäessä jatkuvastikin määritteitä SET, PATH ja PROMPT, joilla siten muotoillaan ympäristöä; asennettu ympäristö on staattinen, koska se on kopio käynnistyksen yhteydessä asetetusta ympäristöstä, josta syystä käytön yhteydessä voidaan antaa toistuvia asennuskomentoja (muuttaa esimerkiksi ympäristön komentotulkin asemaa määrittelevä levyasema C: levyasemaksi A: tai muuttaa TTY arvo VT52:een; mikä tapahtuu kaikissa niissä eräajoissa, joissa on komentorivinä %TTY% jonka tilalle asettuu VT52; mikäli eräajot on yleisesti suunniteltu niin, että niissä on

korvattavia %TTY% ketjuja; mikäli jokin ohjelma muuttaa jatkuvasti koneen käynnistyksen yhteydessä annettuja määrittämiä, ei se vaikuta erityisen tuhoisalla tavalla staattisen ympäristön arvoihin)

SET [<<ketju=ketju>>]

esimerkkejä:

SET TTY=VT52  
SET

### **VOL** .....

Komento esittää käyttöjärjestelmän tasolla esitettynä levyn tai levykkeen VOLUME LABEL nimikkeen muutamata sitä.

VOL [<<d:>>]

### **TYPE** .....

Komento esittää käyttöjärjestelmän tasolla esitettynä spesifioidun tiedoston sisällön näytölle.

TYPE [<<d:>>] <<tiedostonimi>>

esimerkkejä:

TYPE TIEDOSTO.001  
TYPE A:TIEDOSTO.001

### **VER** .....

Komento esittää käyttöjärjestelmän tasolla näyttöön käytössä olevan MSDOS version.

VER

### **VERIFY (/V)..**

Komento VERIFY ON aiheuttaa sen, että levyille tai levykkeelle kirjoitettujen tiedostojen virheettömyys tarkistetaan operaation jälkeen.

VERIFY [ON] tai [OFF]

## **CTRL C .....**

katkaisee mm. eräajon. Katso internaalinen komento BREAK ja sen käyttö.

## **ERÄAJOJEN RAKENTAMISEN YHTEYDESSÄ KÄYTETTYJÄ INTERNAALISIA KOMENTOJA**

### **ECHO .....**

Komento kääntää esimerkiksi eräajoissa annetun komennon ECHO OFF tilapäisesti pois päältä, jolloin on mahdollista asettaa sen perään jokin haluttu tiedote.

ECHO [ON] tai [OFF]; ECHO <<teksti>>

esimerkkejä:

```
ECHO OFF
ECHO ***HOMMA TEHTY***
```

### **REM .....**

Komentoa käytetään esitettäessä jotakin kommenttia eräajojen yhteydessä.

REM <<teksti>>

```
REM ****HOMMA TEHTY****
```

### **FOR.....**

FOR on komennon laajennus, jota käytetään sekä eräajoissa, että interaktiivisessa tiedostojen prosessoinnissa. Huomioitavaa on, ettei käytettäisi <<c>>:n arvona lukuarvoja 0-9, jotteivät ne sekaantuisi BATCH-parametreihin %0-%9.

FOR

BATCH-prosessointi:  
for %% <<c>> in <<set>> do <<command>>



INTERAKTIIVINEN prosessointi:  
for % <<c>> fin <<set>> do <<command>>

esimerkkejä:

for %%f in (report memo address) do del %%f

jossa sidotaan variaabeli f tiedostoihin REPORT, MEMO ja ADDRESS, ja sen jälkeen tuhotaan ne.

**GOTO** .....

GOTO on komennon laajennus, jota käytetään eräajoissa esimerkiksi tapauksessa, jossa halutaan palata tiettyyn alkuasetelmaan, tai päättää eräajo tiettyjen ehtojen vallitessa.

GOTO <<label>>

BATCH-prosessointi:

```
C:  
IF NOT EXIST C:\PROGE\WP50\WP.EXE GOTO END  
CD\DATA\TEKSTIT  
C:\PROGE\WP50\WP  
:END
```

**IF** .....

IF on komennon laajennus, jota käytetään eräajoissa.

IF <<ehto>> <<komento>>

esimerkki:

if not exist tieto1 echo EN LÖYDÄ TIEDOSTOA!

**PAUSE** .....

Komento katkaisee väliaikaisesti eräajon etenemisen.

PAUSE

**SHIFT** .....

Kytkin sallii käytettävän enemmän kuin kymmenen korvattavaa parametria

BATCH-prosessoinnissa; tavallisesti komentotiedostoilla on kymmeneen rajoittuva määrä käsitellä parametreja %0 %9

SHIFT

**CALL .....**

Komento toimii kutsuttaessa haluttuja eräajoja käyttöjärjestelmän tasolla; komentoa voidaan käyttää sijoitettuna myös jonkin eräajon halutulle komentoriville, ja siten yhdellä eräajolla ajaa useita ohjelmia tai suorittaa useita eksternaalisia komentoja peräkkäin.

**1.2.1.4. MSDOS KÄYTTÖJÄRJESTELMÄN EKSTERNAALISET KOMENNOT**

Seuraavassa esitetään lyhennetty luettelo Microsoft MSDOS käyttöjärjestelmän [Version 2.11] eksternaalisista komennoista. Tällaista käyttöjärjestelmää käytettiin vv. 1983-1984 laitteissa, joissa oli kaksi 360 K kalvovolykeasemaa:

Volume in drive A has no label  
Directory of A:\

```
COMMAND.COM 15957 11-10-83 12:03 p
LIB.EXE 61696 10-19-83 7:52 p
EDLIN.COM 8080 0-19-83 7:51 p
.....
.....
.....
LINK.EXE 42330 10-19-83 7:51 p
MASM.EXE 80856 10-19-83 7:52 p
CREF.EXE 13824 10-19-83 7:52 p
```

Luettelosta voi panna merkille sen, että ohjelmien koko on huomattavasti pienempi kuin uudemmissa käyttöjärjestelmissä on laita; siihen kuuluu MASM.EXE [Macro Assembler], LINK.EXE [Linker], LIB.EXE [Library Manager], CREF.EXE [Cross-Reference Utility] ja luetteloon kuulumaton DEBUG.COM [Debugger], joiden avulla voidaan tuottaa ohjelmia assembly -kieltä hyödyntäen. Kyseinen kieli on tavallaan kaikkein varhaisin muoto ja tapa tuottaa itsenäisesti "ajettavissa olevia ohjelmia" käyttäen jotakin yksinkertaista

videoeditoria, joka on (tavallaan) tekstinkäsittelyohjelmien edeltäjä. Tässä luettelossa tekstieditorina on EDLIN.COM -ohjelma

Varsinainen erikseen saatavilla oleva MICROSOFT MACRO ASSEMBLER ohjelmalevyke sisältää seuraavat ohjelmat:

#### M86.EXE

(Microsoft Macro Assembler Utility Manual)  
Vaaditaan vähintään 96K keskusmuistitilaa;  
64K koodit & staattinen data / 32K ajotilaa

#### LINK.EXE

(Microsoft LINK Linker Utility Manual)  
Vaaditaan vähintään 50K keskusmuistitilaa;  
40K koodit / 10K ajotilaa

#### LIB.EXE

(Microsoft LIB Library Manager Manual)  
Vaaditaan vähintään 38K keskusmuistitilaa;  
28K koodit / 10K ajotilaa

#### CREF.EXE

(Microsoft CREF Cross-Reference Utility Manual)  
Vaaditaan vähintään 24K keskusmuistitilaa;  
14K koodit / 10K ajotilaa

#### DEBUG.EXE

(Microsoft DEBUG Utility Manual)  
Vaaditaan vähintään 13K keskusmuistitilaa;  
13K koodit / ajotila ohjelmasta riippuvainen

Käytettävien fyysisten levyasemien määrän suhteen käytännöllisin ratkaisu on kaksi, tai useampia, koska ohjelmat eivät salli luotavan lisätilaa sille levykkeelle, josta käsiteltävä data luetaan.

#### PÄÄPIIRTEITÄ

MICROSOFT MACRO ASSEMBLER on tarkoitettu 8086-pohjaisille **kotitietokoneille**, ja on yhteensopiva Intel ASM-86:n kanssa lukuunottamatta Intelin koodimakroja, makroja, ja joitakin \$ direktiivejä. Ohjelma tukee useimpia 8080 assembler-direktiivejä, ja makrot sekä konditionaalit ovat Intel 8080 standardin mukaisia.

MICROSOFT LINK LINKER UTILITY on virtuaalitetiedostopohjainen ohjelma, josta syystä se voi linkittää sellaisiakin ohjelmia jotka ovat suurempia kuin ohjelman käytössä oleva muisti. Ohjelma tuottaa osoittuvan ja ajettavissa olevan objektikoodin, sekä suorittaa jäsentelyn haluamallasi tavalla. Ohjelma voi myös suorittaa monikerrannaisen kirjastohaun käyttäen sanakirjasta tuttua hakumetodia. Ohjelma esittää istunnon aikana luotavat output-input modulit ja muita parametreja.

MICROSOFT LIB LIBRARY MANAGER voi lisätä, tuhota ja ekstraktoida moduleja luotavan ohjelman kirjastotiedostoista ja esittää myös input-output tiedoston ja modulinimet.

MICROSOFT CREF CROSS REFERENCE UTILITY tuottaa ristikkäisosoitteisen listan kaikista symbolisista nimistä MACRO ASSEMBLER-menetelmällä luodusta kantatiedostosta.

MICROSOFT DEBUG UTILITY tarjoaa keinon ajaa binaarisia ja ajettavissa olevia objektitiedostoja. DEBUGilla voidaan paikantaa assemblidusta ohjelmasta sen virheet, ja korjata ne. DEBUGilla voidaan ohjelmia muutoinkin muokata, kuten myös muuttaa GPU rekisterien sisältöjä, ja tarkistaa muutosten validius koeajamalla muokattu ohjelma.

## SYNTAKTISET NOTAATIOIT

[ ]

Symboli indikoi optionaaliseen kohtaan.

Symboli indikoi kysymykseen, johon annetaan tietty vastaus. Jos symbolimerkeillä rajataan pienten kirjainten tekstiä on kyseessä anto, jossa kirjoitetaan esimerkiksi tiedoston nimi lename, ja jos kyseessä on suurten kirjainten tekstiä on antona jokin näppäimistö.

{ }

Symboli indikoi siihen, että on useampia vastausvaihtoehtoja, joista tulee valita yksi -siitä huolimatta, että ne olisivat merkityt hakasulkusymbolin sisään.

...

Symboli viittaa siihen, että anto voidaan toistaa niin useasti kuin halutaan.

## CAPS

Isot kirjaimet indikoivat lauseiden osiin, tai komentoihin, jotka tulee antaa juuri niinkuin ne näytetään.

## OHJELMANKEHITTELYN KULKU

### 1. VAIHE

8086 Assemblykielinen kantatiedosto kirjoitetaan EDLIN-ohjelmalla tai jollakin videoeditorilla, ja tiedostolle annetaan erityistunnusosaksi .ASM -joskin sille voidaan antaa muitakin erityistunnusosia, jolloin Macro Assembleria käynnistettäessä ohjelmakomennon perään kirjoitetaan muokattavan kantatiedoston koko tiedostonimi.

### 2. VAIHE

Kantatiedosto ASSEMBLOIDAAN (ajetaan Macro Assembleriin), joka antaa assembloidulle objektitiedostolle erityistunnusosaksi .OBJ. Niitä voidaan linkittää tämän jälkeen ohjelmatiedostoihin (tai luoda niistä ohjelmatiedosto).

ASSEMBLERilla luotuja listaustiedostoja on kahdenlaisia:

2.1. Normaali listaustiedosto, joka näyttää assembloidut koodit suhteellisine osoitteineen, lauseineen ja symbolitauluineen.

2.2. CROSS-REFERENCE tiedosto, jonka erityiset kontrollimerkit mahdollistavat MS-CREFille listauksen, jossa näkyy jokaisen symbolimäärittelyn ja sen mahdollisten referenssien rivinumerosijainti.

### 3. VAIHE

MS-LINK ohjelmalla voidaan linkittää useita .OBJ tiedostoja yhteen ja luoda niistä ajettavissa oleva objektitiedosto, joka on .EXE-ohjelmatiedosto. MS-LIB ohjelmalla voidaan luoda kirjastotiedosto (tai tiedostoja) jo olemassaolevista kirjastotiedostoista.

### 4. VAIHE

ASSEMBLOITU ja LINKITETTY .EXE ohjelma ajetaan DOS:n alaisuudessa. Mikäli luotu ohjelma ei toimi, voidaan siihen tehdä vaadittavia pieniä korjauksia DEBUG ohjelmalla, jolla virheet paikannetaan ja ohjelmaa koeajetaan.

Tämäntapaiset ohjelmat mahdollistivat aikoinaan sen, että ohjelmia voitiin kirjoittaa tavallisella tekstieditorilla, kuten pääasiassa alkuun meneteltiin esimerkiksi EDLIN rivieditoria käyttäen, muuntaen tuotokset varsinaiseksi

ohjelmaksi edellämainituilla muunto-ohjelmilla. Luotaessa assembly -kielellä ohjelmia, voidaan niihin liittää hiiren käyttömahdollisuus esimerkiksi seuraavasti:

- 1.Ladataan rekisterit AX, BX, CX -ja DX parametriarvoin (tässä mainitut rekisterit vastaavat M1%,M2%,M3%, ja M4% -parametreja)
- 2.Määritellään software-päätearvo-piste (51(33H))

BASIC -tulkillilla [kuten GWBASIC -tulkillilla] voidaan "hiiri" initialisoida seuraavasti:

```
10 DEF SEG=0
20 MSEG=256*PEEK(51*4+3)+PEEK(51*4+2)
30 MOUSE=256*PEEK(5*4+1)+PEEK(51*4)+2
40 DEF SEG=MSEG
```

Tämän jälkeen annetaan CALL -komento hiirelle, joka on muotoa:

```
CALL MOUSE(M1%,M2%,M3%,M4%)
```

M1%, M2%, jne. arvot on määriteltävä ts. määriteltävä se alue, jossa hiirellä operoidaan. Tällaiset, annetut arvot ovat INTEGER -arvoja (ei desimaali, tai murtolukuja, vaan kokonaislukuja). Tämä on vain hyvin karkeatekoinen tapa tuottaa hiiri käyttöympäristöön, sillä myös sen käyttäytymistä on mahdollista monin tavoin määritellä, ja osoittaa sille paikat, missä mitään on tehtävä.

Jotkut käytetyistä ohjelmanteko-metodeista vaikuttavat suoraan video-muistiin, kuten GWBASIC, jolloin ennen kirjoittamista nuolikuvi (cursor) on piilotettava, ja vasta kirjoittamisen jälkeen "katsottava", missä se on. Ohjelmoinnissa käytetään 8086 software päätearvo-piste parametriarvoja (51), jotka viedään 8086 -rekistereihin, ja tuotetaan myös sieltä: Tätä metodologia voidaan käyttää mm. ohjelmissa BASIC, FORTRAN, COBOL, C-kieli, tai PASCAL.

Seuraavassa osa Microsoft MSDOS 4.01 käyttöjärjestelmä [Version 4.01] eksternaalisista komennoista:

```
Volume in drive C is DOS_401
Volume Serial Number is 3179-15D8
```

Directory of E:\BIN\MSDOS

```
.  
..  
LCD CPI 10703 10-06-88 12:00a  
SHARE EXE 13424 10-06-88 12:00a  
BACKUP COM 36880 10-06-88 12:00a  
GWBASIC EXE 80608 10-06-88 12:00a  
REPLACE EXE 19415 10-06-88 12:00a  
XCOPY EXE 17055 10-06-88 12:00a  
RESTORE COM 36946 10-06-88 12:00a  
EGA CPI 49068 10-06-88 12:00a  
FC EXE 15807 10-06-88 12:00a  
DEBUG COM 21574 10-06-88 12:00a  
LINK EXE 43988 10-06-88 12:00a  
DISKCOMP COM 9857 10-06-88 12:00a  
5202 CPI 370 10-06-88 12:00a  
SYS COM 11456 10-06-88 12:00a  
EXE2BIN EXE 7963 10-06-88 12:00a  
CHKDSK COM 17787 10-06-88 12:00a  
APPEND EXE 11154 10-06-88 12:00a  
ASSIGN COM 5753 10-06-88 12:00a  
ATTRIB EXE 18263 10-06-88 12:00a  
COMP COM 9459 10-06-88 12:00a  
MODE COM 22960 10-06-88 12:00a  
....  
....  
RECOVER COM 10588 10-06-88 12:00a  
GRAPHICS COM 16693 10-06-88 12:00a  
EDLIN COM 14069 10-06-88 12:00a  
PRINT COM 14131 10-06-88 12:00a  
COMMAND.COM 37557 12-19-88 12:00a  
ESC COM 93 05-03-88 12:07p  
FILESYS EXE 11129 10-06-88 12:00a  
GRAPHICS PRO 9397 10-06-88 12:00a  
IFSFUNC EXE 21739 10-06-88 12:00a  
MEM EXE 20005 10-06-88 12:00a  
NCFREQ EXE 1802 06-14-88 3:43p  
NCSELECT EXE 93325 04-10-89 10:39a  
SELECT PRT 68 03-06-89 9:11a  
SELECT DAT 23202 03-30-89 7:38a  
SELECT HLP 28746 03-30-89 10:17a  
56 File(s) 12822528 bytes free
```

Tämä käyttöjärjestelmä on tarkoitettu käytettäväksi ensisijassa kiintolevyillä -  
osin johtuen jo ohjelmien koostakin. Alleviivatut eksternaaliset ohjelmat

toimivat erityisesti konfiguroitaessa **kotitietokonetta** mm. erilaisille näytöille, näppäimistöille ja tulostusgrafiikoille, sekä useiden samanaikaisten saantipolkujen määrittämisessä tapauksissa, joissa ohjelmistojen tiedostot on hajautettu useisiin hakemistoihin. Lihavennetut eksternaaliset tiedostot kuuluvat asennusohjelmaan, jolla käyttöjärjestelmä voidaan ladata kiintolevylle.

### **1.2.2. XENIX**

Jo MSDOS 2.0 käyttöjärjestelmä tarjoaa hierarkiset [puurakenteiset] hakemistot - samanlaiset, jollaiset ovat myös XENIX käyttöjärjestelmässä. Jotkut järjestelmäkutsuista [System Calls] ovat MSDOS järjestelmässä yhteensopivia XENIX järjestelmään:

Funktio 39H  
Tee alihakemisto

Funktio 3AH  
Siirrä tiedosto hakemistosta

Funktio 3BH  
Siirry vallitsevasta hakemistosta toiseen

Funktio 3CH  
Luo tiedosto

Funktio 3DH  
Avaa tiedosto

Funktio 3FH  
Lue tiedostoa tai oheislaitetta

Funktio 40H  
Kirjoita tiedostoon tai oheislaitteelle

Funktio 41H  
Tuhoa tiedosto hakemistosta

Funktio 42H  
Funktio siirtää READ/WRITE POINTERIN jonkun seuraavista periaatteista mukaisesti:



#### 0POINTER

siirretään OFFSET-bitteihin tiedoston alusta.

#### 1POINTER

siirretään vallitsevaan sijaintiin plus OFFSET

#### 2POINTER

siirretään tiedoston loppuun plus OFFSET

#### Funktio 43H

Vaihda tiedostoattribuutit

#### Funktio 44H

I/O -kontrolli oheislaitteille

#### Funktio 45H

Funktio ottaa jo avatun FILE HANDLEN [BX] ja palaa uuteen FILE HANDLEEN [AX], joka viittaa samaan tiedostoon samassa osoitteessa

esim (MSDOS):

```
mov bx, fh
mov ah, 45H
int 21H
; ax has the returned handle
```

#### Funktio 46H

Funktio ottaa jo avatun tiedoston FILE HANDLEN [BX] ja palaa uuteen FILE HANDLEEN [CX], joka viittaa samaan tiedostoon samassa osoitteessa; mikäli HANDLE CX:ssä on jo avattuna jokin tiedosto, suljetaan se ennen operaatiota

esim (MSDOS):

```
mov bx, fh
mov cx, newfh
mov ah, 46H
int 21H
```

#### Funktio 4BH

Avaa ja lataa jokin ohjelma

#### Funktio 4CH

Päätä prosessi

## Funktio 4DH

Funktio palauttaa Exit koodiin, joka on spesifioitu lapsi-prosessissa. Se palauttaa Exit koodin vain kerran. Koodit alabitti lähetetään Exit-rutiiniin ja yläbitti johonkin seuraavista:

0Päätä/katkaise

1Control-C

2Hard Error

3Lopeta ja pysy voimassa

### 1.2.3. DRDOS 6.0

DRDOS -käyttöjärjestelmä on Western-Digitalin kehittämä käyttöjärjestelmä, joka on suurelta osin MSDOS-järjestelmien kanssa yhteensopiva. Kuitenkin DRDOS 6.0 eroaa esimerkiksi MSDOS 4.01:stä siinä, että sen avulla voidaan hallita ilman SHARE-ohjelman latausta aina 512 MB:n partitioita, DRDOS 6.0 voidaan kytkeä WINDOWS-ohjelmaan, DRDOS 6.0:n avulla voidaan hallita erilaisia laajennettuja [EXPANDED] ja lisättyjä [EXTENDED] muisteja jne. DRDOS 6.0:n valikko on hiiriohjattava ja siinä on erilaisten ohjelmien symboleina tarjolla erilaisia ikoneja. DRDOS 6.0 sisältää myös mahdollisuuden useiden ohjelmien samanaikaiseen ajoon. Seuraavassa kaaviossa eräs DRDOS käyttöjärjestelmän keskusmuistin käyttötapa tapauksessa, jossa käytettävän keskusmuistin laajuus on 640 Kb, ja lisätyn muistikortin laajuus 2000 Kb. Keskusmuistin aktiivit sisällöt koostuvat mm. muistiresidenteistä hiiri- ja näppäimistöohjelmista, käyttöjärjestelmästä ja PRO -ohjelmasta, jonka tila vapautuu heti kun ohjelman käyttö lopetetaan.

```
0:0000 ----- 400h, 1 024 Interrupt vectors
40:0000 ----- 100h, 256 ROM BIOS data area
50:0000 DR DOS 200h, 512 DOS data area
70:0000 DR BIOS B60h, 2 912 Device drivers
126:0000 DR DOS 11B0h, 4 528 System
241:0000 DR DOS FB00h, 64 256 System
4D1:0000 DR DOS 2800h, 10 240 20 Disk buffers
6BC:15C0DR DOS DE0h, 3 552 DR DOS BIOS code
5F6:3000 DR DOS 9440h, 37 952 DR DOS kernel code
11F1:0000COMMAND 1590h, 5 520 Program
134A:0000 COMMAND 210h, 528 Environment
136B:0000 ----- 80h, 128 FREE
1373:0000 MOUSE 80h, 128 Environment
137B:0000APPEND 830h, 2 096 Program
13FE:0000 KEYB 12E0h, 4 832 Program
152C:0000MOUSE 19E0h, 6 624 Program
```

```
16CA:0000 PRO A0h, 160 Environment
16D4:0000PRO 13AD0h 80 592 Program
2A81:0000 ----- 757F0h,481 264 FREE
```

DRDOS 6.0:n eräitä eksternaalisia komentoja ovat:

```
Volume in drive C is DRDOS_60
Directory of C:\DRDOS
```

```
.
..
MEMMAX EXE 2214 16.10.91 6.00
LOGIN EXE 14858 16.10.91 6.00
LOGIN TXT 1283 16.10.91 6.00
LOGIN ERR 402 16.10.91 6.00
SETUP EXE 79298 16.10.91 6.00
TASKMAX EXE 17581 16.10.91 6.00
DISKOPT EXE 50080 23.08.91 6.00
DELWATCH EXE 11527 23.08.91 6.00
DELPURGE EXE 19771 23.08.91 6.00
DISKMAP EXE 8233 23.08.91 6.00
LOCK EXE 15662 23.08.91 6.00
LOCK TXT 1309 23.08.91 6.00
LOCK ERR 422 23.08.91 6.00
....
....
....
104 File(s) 11503616 bytes free
```

### 1.3. JÄRJESTELMÄ-SOFTWARE

Tämä viittaa käyttäjän mahdollisesti itse konstruoimiin ohjelmiin erityisin, tarkoitusta varten suunnitelluin ohjelmointikielin, joilla ohjelmia on sovitettu käytettäväksi esimerkiksi DOS:n alaisuudessa. Otsikko viittaa myös muihin työkaluihin joilla ohjelmia voidaan käsitellä ja räätälöidä monin tavoin. DOS:n yhteyteen voidaan liittää systeemejä, jotka eivät alunperin oletusarvoina "kuulu" DOS:iin niin, että ne kuitenkin toimivat siinä, vaikka ovatkin itsenäisiä [esimerkiksi siten, etteivät tarvitse toimiakseen muuta kuin järjestelmätiedostot]. Ne ovat tavallaan toisia kieliä. Samoin voidaan eräisiin muihin järjestelmiin sisällyttää osioita, joiden avulla voidaan hyödyntää DOS-ohjelmia.

#### 1.4. LOPPUKÄYTTÄJÄN TYÖKALUT [End-user tools]

Tällaisia ovat useimmat valmisohjelmista, joiden kanssa loppukäyttäjät työssään tavallisimmin joutuvat tekemisiin, kuten tekstinkäsittelyohjelmat, taulukkolaskimet, taloushallinto- ja kirjanpito-ohjelmat, piirto-ohjelmat, CAD/CAM ohjelmistot jne., joissa käyttäjä voi suorittaa rajoitettuja muutoksia ohjelmilla työstettävälle materiaalille, datalle, joka tallennetaan työn kuluessa ja loputtua tiedostoiksi. Ne ovat sitä, mitä tavallinen käyttäjä **kotitietokoneista** ymmärtää, ja mitä hän **tietokoneiden** tavaksi toimia käsittää. Normaalisti käyttäjä sanoo "osaavansa käyttää **tietokonetta** ja ymmärtävänsä sen tapaa toimia" silloin, kun hän on opetellut valitsemaan työskentelynsä kuluessa oikein ohjelmien optioita. Ohjelmien varsinainen luominen ja valmistus on prosessin toisessa päässä, ja niihin liittyvä laitteiston kehittäminen - ja yksinkertaisimmassa päässä ketjua on loppukäyttäjä, joka käyttää välineinään valmisohjelmia l. loppukäyttäjän työkaluja [END-USER TOOLS].

Luettelon loppuun voisi lisätä itse tehdyt ohjelmien muunnokset, joita syntyy, kun tiettyjä, valmiita, ja muotoiltaviksi suunniteltuja ohjelmia kehitetään omiin käyttötarkoituksiin sopivaksi. Ohjelmantekijän tekemiä muunnoksia ovat esimerkiksi ohjelmien suomentamiset, ohjelmien parametrien muuttamiset, itse ohjelmoinnin muuntamiset jne., kun taas loppukäyttäjän tekemiä muunnoksia ovat ohjelmien asennukset tietyille näytöille ja kirjoittimille ja ohjelmien väriasennukset. Tällä tarkoitetaan siten loppukäyttäjälle ohjelmavalmistajien taholta antamia keinoja sovittaa ohjelmistoja erityyppisille laitteistoille. Itse tehdyt OHJELMAT, joiden teossa voi käyttää erilaisia ohjelmointikieliä joiden avulla ohjelmiin voi liittää hiiren, tai muuntaa niitä valokynällä [light pen] toimiviksi, vaativat suurempaa taitoa - sitä suurempaa, mitä enemmän itse suoritetaan varsinaista ohjelmointityötä. Ohjelmien vaativammissa muunnoksissa saatetaan valmistaa uusia näppäimistöohjaimia, uusia näyttöohjaimia kontrolloivia ohjelmia jne. ja siten saada ohjelmat toimimaan erilaisissa **tietokoneissa**. Loppukäyttäjälle näppäinten muokkaus merkitsee sitä, että näppäimistön voi sovittaa haluamanlaisekseen niin kuin se ohjelman kautta sitä muuttamatta on mahdollista, tai ohjelmaan voi istuttaa hiiren niin kuin se siihen ohjelmaa muuttamatta voidaan tehdä, tai valita tietyn näytön annetusta valikosta.

Ohjelmatkaan eivät siten ole täysin valmiita käytettäväksi, vaikka niitä VALMISOHJELMINA kaupassa markkinoitaisiinkin. Jotkut hyötyohjelmista [ja peleistä] ovat suojattuja, vaikka ne toimivatkin yleensä DOS:n alaisuudessa. Suojaus estää niiden mahdollisen kopiointin, tai sen pitäisi estää, sillä kyllä niitäkin voi kopioida erilaisilla kopiointi-ohjelmilla, ja kopioita voidaan myös käyttää kiintolevyllä, kun annetaan sovelias komento, joka "murtaa" niihin rakennetun suojauksen. Ohjelmien maailma on monasti sitä, että ohjelmat "saadaan kuvittelemaan", että ne toimisivat "alkuperäisinä".

## 1.5. KESKUSYKSIKKÖ

**Kotitietokone** koostuu keskusyksiköstä, näppäimistöstä, monitorista, ja usein siihen mielletään myös kuuluvaksi kirjoitin.

Keskusyksikön sisällä sijaitsevat laitteen virtalähde- ja tuuletinyksiköt, GPU [Central Processing Unit], korttiasemat [Slots], kaiutinyksikkö, AT -tason laitteissa CMOS -muisti, kiinnitysrakenteet kiintolevyille, näytönohjainkortti, ja laitetypistä riippuen eräitä muitakin komponentteja. Keskusyksiköt toimitetaan useimmiten varustettuna yhdellä kiintolevyllä, joka on sekä fyysisesti- että loogisesti alustettu, sekä näytönohjaimelle sopivalla monitorilla varustettuna.

Keskusyksikköön kuuluu ulkoisestikin havaittavina yksi tai useampia kalvovoly- ja kiintolevyasemia, sekä kirjoitin-, hiiri,- modeemi- ja monitorikaapeliliitännät, jne. Näihin liitännöihin kytkettävät sarjakytkentäiset oheislaitteet [devices] tunnetaan COM# lyhenteellä, mikä tarkoittaa sarjaporttia [Serial Port], ja rinnakkaiskytketyt lyhenteellä LPT#, mikä tarkoittaa rinnakkaisporttia [Parallel Port]. Näiden lyhenteiden merkitys on tunnettava konfiguroitaessa **kotitietokonetta** erilaisille siihen liitetyille oheislaitteille, erityisesti kirjoittimille, hiirelle ja modeemille. Käytettäessä esimerkiksi erilaisia ohjelmia ja tulostettaessa niistä numeroita, tekstejä ja kuvia esimerkiksi kirjoittimelle, tai siirrettäessä modeemilla dataa toisiin **tietokoneisiin** on niissä sitä ennen tehtävä kirjoitinmäärittely, ja asennettava modeemi, jossa eräs määriteltävä ominaisuus on porttimäärittely.

Keskusyksikön sisällä voi olla lisäksi erilaisia kortteja, joiden kautta oheislaitteet kytketään osaksi keskusyksikköä asentamalla ne korttiasemiin [Slots]. Esimerkiksi laajennettaessa laitteen muistikapasiteettia [Expanded Memory] on keskusyksikölle osoitettava, missä osoitteissa laajennettu muisti löytyy, mikä tehdään kirjoittamalla konfigurointitiedostoon [CONFIG.SYS] tarvittavat komentorivit (tai ne asentuvat automaattisesti konfigurointitiedostoon asennettaessa muistia ohjaavaa ohjelmaa). Kortin mahdolliset kytkimet [dip switches] on asennettava niihin asentoihin, joissa niiden osoitetaan konfigurointitiedoston komentoriveillä olevan. Fyysisten kytkinten asentamisen on tapahduttava ennen konfiguroitua, ja laite tulee käynnistää uudelleen asennuksen jälkeen. Näin "keskusyksikkö" ei sekaannu muistiosoitteissa ja löytää siihen liitetyt "kortit", koska konfigurointitiedoston komentorivit yleensä kirjoittavat keskusmuistiin annetut osoitetiedot. Keskusyksikössä vallitsevat muistityypit tunnetaan nimillä ROM [Read Only Memory] ja RAM [Random Access Memory]; laajennetut muistiyksiköt toimivat yleensä RAM -tilassa, jonne voidaan lukemisen ohella myös kirjoittaa ja jonka "sisältö" voi eräin osin vaihdella koneen käyttöistunnon aikana.

Jotkut korteista voivat olla keskusyksikköön kytkettyjen oheislaitteiden

ohjauskortteja, joissa asennettavia seikkoja voivat olla esimerkiksi DMA -kanava, I/O -portti, ja IRQ [Interrupt Level], jotka säädetään sekä kortin kytkimillä [dip switches] että ajamalla asennusohjelma ohjauskortin kytkentöjen mukaisesti, jolloin konfigurointitiedostoon kirjoittuvat tarvittavat komentorivit.

Keskusyksikköön voidaan asettaa haluttaessa puhelinadapterikortti, ja useita muita kortteja mikäli vain tilaa riittää, eli "sloiteja" on vapaana ja niihin kytkkeyksissä olevat ohjelmat asennetaan niille asetetuille säädöille.

### 1.5.1. MASSAMUISTIT

Kun **kotitietokoneen** käyttöjärjestelmällä ja/tai ulkoisella ohjelmistolla on luotu tietty rajattu kokonaisuus [tiedosto], joka halutaan tallentaa, on koneessa sitä varten asennettuina mm. levy -ja nauhamuistiyksiköjä, joissa yksiköiden kirjoitus- ja lukupäät ja muu siirtolaitteisto vievät ja tuovat bittivirtaa koneen keskusmuistista levyjen [rautaoksidi] magneettipinnoille ja takaisin.

Levyillä bittivirta ohjataan raidoille [tracks], joita luku- ja kirjoituspäät pyyhkivät. Raidalla luku- ja kirjoituspäät joko tunnistavat tiettyä hetkenä signaalin esiintyväksi, tai sitten raidalla ei ole kyseisellä hetkellä lainkaan signaalia, mikä tarkoittaa bittikoodiksi muutettuna sitä, että luku- ja kirjoituspäät ovat tunnistaneet tiettyä hetkenä bittiarvon "1", tai sitten arvon "0". Näistä havaituista nolista ja ykkösistä muodostuu binaarisen lukujärjestelmän koodeja, jossa kukin koodeista edustaa tiettyä reaaliarvoa; **tietokone** käsittelee pelkästään lukuja, ei siis sinällään kirjaimia tai muita symboleja.

Signaalimuoto levyillä on digitaalinen, jossa levyn raidalle tallentuu epäsäännöllisin väliajoin jotakin, tai ei mitään. Tästä syystä johtuen tallennus ja luku voidaan tehdä samanaikaisesti useammallekin raidalle, mikä nopeuttaa sekä lukua- että kirjoitusta. Tällöin käytetään samanaikaisesti useampia luku- ja kirjoituspäitä, kuten esimerkiksi kiintolevyillä on laita. Tavanomaisesti, käytettäessä tavallista levykettä, on luku- ja kirjoituspäiden määrä kaksi, jolloin ne joutuvat siirtymään dataa hakiessaan sivuttaissuunnassa raidalta raidalle enemmän kuin jos luku- ja kirjoituspäitä olisi runsaammin, mikä vie aikaa.

Tavallisimmin luku- ja kirjoituspäät käyttävät levyn- tai levykkeiden molemmilla puolilla olevia raitoja, jotka sijaitsevat samalla kohtaa toistensa suhteen. Sylinteri [cylinder] on sellainen arvo, jonka aikana luku- ja kirjoituspäät voivat lukea tai tallentaa dataa niin, ettei ole tarpeen siirtää päitä sivuttaissuunnassa. Kaksipuoleisen levyn yläpuolen ensimmäinen raita [track] on arvoltaan nolla [0], ja sivun arvo on myös nolla [0]. Saman levyn alapuolen ensimmäinen raita [track] on myös nolla [0], mutta sivun arvo on yksi [1].

Tässä tapauksessa molemmat nollaraidat [0] muodostavat yhden sylinterin [cylinder].

Tavallisella 360 KB:n levykkeellä raitoja [tracks] on neljäkymmentä, alkaen raidasta nolla [0] päätyen raitaan 39. Tavallisella 720 KB:n 3.5 tuuman levykkeellä raitoja on kahdeksankymmentä alkaen raidasta nolla [0] päätyen raitaan 79.

Dataa tallennettaessa on tarpeen "jakaa" levy tai levyke sektoreihin [sectors], joiden määrä riippuu sekä levykontrollerista [disk controller hardware] että käyttöjärjestelmästä. PC-DOS käyttää sektorijakoa, jossa jokaiseen sektoriin mahtuu 512 bittiä; käytäntö koskee sekä levykeasemia että kiintolevyjä. Tavanomaisesti kiintolevyillä on seitsemäntoista sektoria raitaa kohden. Jotkut uudemmat kiintolevyt, joiden enkoodauskeema on RLL [Run-Length-Limited] koodaus, voivat sisältää 25 tai enemmän raitaa kohden.

Tavallisesti kiintolevyllä on hyvinkin kuusisataa raitaa [tracks] tuumaa kohden, jonka vuoksi suosituiksi ovat tulleet ratkaisut, joissa kiintolevyt koostuvat useista samankeskisistä vierekkäin asetetuista levyistä. Toisaalta, koska kiintolevyjen luku- ja kirjoituspäät eivät kosketa itse levypintaa, vaan leijuvat siitä noin 10 mikrotuuman päässä, ovat kiintolevyjen valmistukseen liittyvät toleranssit hyvin tarkat ja tiukat. Tämä merkitsee myös sitä, että täysin virheettömiä yksilöitä löytyy vain harvoin.

Kiintolevyt täytyy installoida sille **tietokoneelle**, johon ne asennetaan. Esimerkkinä eräästä 40 MB:n istallaatioprosessista olkoon seuraava. Halutaan, että hankittuun, uuteen ja käyttämättömään kiintolevyyn saadaan seuraavat ominaisuudet.

---

977 CYLINDERS  
5 HEADS  
17 SECTORS  
Precomp (300)

---

AT **kotitietokoneissa** toimii käynnistyksen yhteydessä fyysinen alustus ja diagnoosiohjelma, joka mm. tutkii laitteen eri toiminnot ennen sen lopullista käynnistymistä ja "asettaa" levy- ja levykeasemat käyttövalmiuteen. Mikäli laitteessasi ei ennestään ole ollut kiintolevyä, on sinun poimittava tällainen variaatio **kotitietokoneesi** fyysiseen alustukseen osallistuvasta ROM BIOSISTA [cmos memory] siellä olevasta kiintolevyjen luettelosta. Kun valinta on tehty ja muutoksen tallennus suoritettu käynnistetään **tietokone** uudelleen alusta saakka ja alustetaan samalla kiintolevyn kontrolleri; huomattakoon, ettei siihen kytketyn kiintolevyn tarvitse olla alustettu [formatoitu] käyttöjärjestelmän ao. komennolla tätä ennen. ROM BIOSIA [cmos memory]

voidaan käsitellä aktivoimalla se tietyillä näppäimillä koneen käynnistyksen yhteydessä, tai käsittelemällä sitä DISK MANAGER tyyppisten ohjelmien kanssa [erityisesti, mikäli ROM BIOSIN [cmos memory] voidaan kirjoittaa muutoksia ja kytkeä tarvittaessa jumperit. ROM BIOSIN [cmos memory] ohjaamana kontrolleri asentaa kiintolevyä käyttökuntoon alustaen raidat, määrittäen sektorien osoitteet, valiten sektorien koot, sync bitit, gap bitit jne.

Jos kaapeloinnit on tehty oikein ja kortit oikein asennettu, sekä mahdolliset emokortin kytkennöt oikein suoritettu, niin kiintolevy on valmis lohkottavaksi [esimerkiksi FDISK tai DISK MANAGER ohjelmilla], jolloin kiintolevyyn luodaan loogiset levyasemanimet. Nämä lohkotut loogiset levyasemat [partitions] voidaan nyt alustaa [formatoida], eli niille voidaan tehdä pehmeä eli looginen formatointi. Ensimmäisen formatoitavan loogisen levyaseman nimi on C:\ ; toisen D:\ , jne., jotka voidaan määrittää kukin vaihtoehtoisilla tavoillaan niin haluttaessa. Alustuksessa muotoutuu levyille sen haluttuun loogiseen levyasemaan:

---

ALKULATAUSLOHKO	[ROOT]
TILANVARAUSTAULUKKO	[FILE ALLOCATION TABLE, FAT]
HAKEMISTOT	[DIR]
DATA-ALUE	[DATA]

---

Tällaisten rakenteiden muodostuttua voidaan kiintolevyiltä lukea tietoja, ja sinne niitä tallentaa. Aktiiviseen loogiseen levyasemaan siirretään alustuksen yhteydessä järjestelmätiedostot ja COMMAND.COM tiedosto, koska kiintolevy ei voi käynnistyä ilman niitä. Kun kaikkien loogisten levyasemien alustus on suoritettu, voidaan ohjelmat siirtää kiintolevyille, ja suorittaa siellä niillä työskennellessä tallenteita tietyillä, annetuilla käyttäjanimillä. Periaate on aina sama: annetaan tallennettaessa ja kutsuttaessa porrasmaisesti tarkentuvien määritteiden sarja, jolla tietty levyasema, ja sillä tietty osoite löytyvät.

Järjestelmätiedostojen lisäksi kiintolevyille on siirrettävä myös joukko sen eksternaalisia ohjelmia joko käyttäen valmista asennusohjelmaa, tai suoritettava oma asennus esimerkiksi kopioimalla ne haluttuun hakemistoon [jollainen niille täytyy luoda].

Tällaista **kotitietokone** pitää sisällään; se on sen olemusta -sen voi ajatella koostuvan moduuleista, joista jokaisella on oma tehtävänsä, ja jotka voidaan korvata toisilla ilman, että toiminta häiriintyy. Käyttäjät voivat lisätä koneen ominaisuuksia vain asettamalla siihen kortteja, ja korvaamalla vanhentuneita "laitemalleja" uudemmilla. Mahdollista edistystä ei voi **kotitietokoneista** silmämääräisesti tarkastellen välttämättä havaita, sillä esimerkiksi kortit ovat



hämäänyttävän samankaltaisia, nekin, vaikka yksi olisi edistyneempi, ja toinen alkeellisempi. Laitteen käyttöjärjestelmissä on edistyneempiä ja varhaisempia versioita, joiden eroa eivät käyttäjät välttämättä huomaa, koska eivät tarvitse joitakin uusista ominaisuuksista koskaan, vaikka heillä uusi järjestelmä tulisikin uuden laitteen mukana. Mahdollinen liitännöiden runsaus ei ole mikään edistyneiden merkki, kuten otaksuisi, tai lisääntynyt laitteen toimintanopeus, koska tietojenkäsittelyä suorittaa vielä useimmissa myytävissä **kotitietokoneissa** vain yksi keskusprosessori, jonka tukena saattaa olla matematiikkaprosessori lukujen desimaaliosien käsittelyä varten. Toisaalta myöskään markkinoille tullut uusi **tietokone**, joka "osaa" lukea ja muuntaa tavallisen kynällä tehdyn kirjoituksen konekirjoitustekstiksi, tai laitteet, jotka "ymmärtävät" puhetta, eivät perustu edistyksellisyydessään lisääntyneeseen nopeuteen tai liitännöihin, vaan uuteen laite- ja ohjelmointifilosofiaan.

### 1.5.2. KIINTOLEVYJEN SUORITUSARVOISTA

Kiintolevyn suorituskykyä mittaamaan on kehitetty lukuisia erilaisia keinoja ja niihin liittyviä käsitteitä:

#### HAKUAIKA

Seek time

Haku aika on se aika, joka kuluu luku- ja kirjoituspäiltä siirtyä siitä kohdasta, jossa ne sattuvat haun alkaessa olemaan siihen kohtaan, jonne "haluat" niiden siirtyvän. Haku aika vaihtelee siten sen mukaan, kuinka etäällä pisteet ovat toisistaan, on hakuajalle kehitetty useita alaluokituksia, joista tärkein on lähimmälle raidalle siirtymiseen kuluva aika [track-to-track seek time].

#### LÄHIMMÄLLE RAIDALLE SIIRTYMISEEN KULUVA AIKA

Track-to-track seek time

Tämän joltakin raidalta lähinnä sitä sijaitsevalle raidalle siirtymiseen kuluva aika on kestoaltaan AT luokan levyillä tavallisesti 8-10 millisekuntia. Hakuajat kalvolevykkeillä ovat huomattavasti pidempiä.

#### TAVOITUSAIKA

Access time

Kyseessä on se aika, mikä luku- ja kirjoituspäiltä kuluu niiden etsiessä dataa, jota haluat. Mikäli päät sijaitsevat esimerkiksi raidalla kolme, ja data raidalla

neljä, niin tavoitusaika on sama kuin lähimmälle raidalle siirtymiseen kuluva aika. Mikäli käytät säännöllisesti sentapaisia ohjelmia kuten COMPRESS ja SPEED DISK, joilla katkeilleet tiedostot yhdistetään ja järjestellään kiintolevylle haluamallasi tavalla, niin tavoitusaika voi olla hyvinkin usein lähellä lähimmälle raidalle siirtymiseen kuluva aikaa.

#### KESKIMÄÄRÄINEN TAVOITUSAIKA

Average access time

Kyseessä on aika, mikä tavoittamiseen keskimäärin kuluu, kun luku- ja kirjoituspäät siirtyvät joltakin raidalta toiselle. Voisi kuvitella, että keskimääräinen tavoitusaika [average access time] olisi sama kuin aika, joka kuluu päältä siirtymiseen nollaraidalta levyn keskikohdassa olevalle raidalle. Näin saattaisi ollakin, mikäli haku alkaisi aina nollaraidalta. Kuitenkin haku alkaa käytännössä miltä kohtaa levyä tahansa, ja siten optimaalisin tilanne näyttäisi olevan se, että haku alkaisi aina levyn keskikohdalta, koska nollaraidalta levyn toiseen päähän on paljon pidempi matka. Keskimääräinen tavoitusaika [average access time] koskee kuitenkin käytännössä kiintolevystä aluetta, joka on siitä yksi kolmannes, tai neljännes; luku- ja kirjoituspäät eivät käy jokaisessa haussa läpi kaikkea data-alueesta. Kun päät ovat tavoittaneet oikean raidan, täytyy niiden vielä etsiä siltä kohta, josta data alkaa. Tähän kuluu aikaa, jota nimitetään viiveeksi.

#### KESKIMÄÄRÄINEN VIIVE

Average latency

Kyseessä on aika, joka kiintolevyltä kuluu yhden puolipyörähdysten tekemiseen. Kiintolevy pyörii 3600 pyörähdystä minuutissa, tai yhden pyörähdysten joka 16.67 millisekunti. Keskimääräinen viive on siten 8.3 millisekuntia. Kaikki kiintolevyt pyörivät tällä samalla nopeudella, joten tämä arvo ei auta niiden keskinäisten paremmuusvertailujen tekemisessä. Sitävastoin kalvovävykkeillä, jotka pyörivät 300 kierrosta minuutissa, keskimääräinen viive [average latency] on 100 millisekuntia, tai ylikin.

#### LUKUNOPEUS

Transfer rate

Kyseessä on nopeus, jolla bittejä luetaan kiintolevyltä l. nopeudesta, jolla ne siirretään **tietokoneen** keskusmuistiin. Lukunopeus [Transfer rate] riippuu levyn kierrosnopeudesta (kiintolevyt 3600 rpm; kalvovävykkeet 300 rpm) ja siitä, kuinka tiheään bitit on raidalle pakattu. Useimmilla tämän päivän kiintolevyillä lukunopeus on 5 Megabittiä sekunnissa. Uusissa ESDI [Enhanced Small Device Interface] standardien mukaisissa kiintolevyissä ja niiden

ohjaimissa saavutetaan 10 tai jopa 15 Megabitin lukunopeuksia.

## 1.6. TIETOKONEEN KIELESTÄ

Ihmisen aivot eivät tehtyjen tutkimusten mukaan toimi **tietokoneiden** tapaan, koska niissä tietojenkäsittelyä suorittavat samanaikaisesti useat järjestelmät, ja vaikka informaationkulku ihmisaivoissa on hitaampi kuin **tietokoneissa**, niin tämä tietojenkäsittelyn rinnakkaisuus ja tietty kokoava yhdistelevyys tekevät ihmisaivoista **tietokoneita** tehokkaammat. On tehty koneita, joissa samanaikaisesti toimii useita prosessoreja, mutta se ei merkitse sitä, että olisi saavutettu samoja ominaisuuksia joita ihmisaivojen toiminnassa on.

**Tietokone** ei käsittele tai tallenna esimerkiksi numeroita tai kirjaimia siinä muodossa kuin ne näkyvät monitorissa käyttäjälle, vaan heksadesimaalikoodeina. Kuitenkin tekstinkäsittelyohjelma (tai piirto-ohjelma) näyttää voivan tuottaa hyvinkin vaihtelevia muotoja näyttöruudulle. Jokainen kirjain voidaan ajatella asetettavaksi ruuruun, joka on jaotettu sitä pienemmiksi ruuduiksi; kirjain voidaan tuottaa mustaamalla tiettyjä pikkuruutuja, ja nimeämällä näin synnytetty ruutu tietyllä koodilla. Aina kun tämä koodi annetaan, piirtyy kuvaruudulle kohdistimen osoittamaan paikkaan kirjaimen graafinen hahmo. **Tietokoneissa** koodit ovat heksadesimaalisia, ja näistä kirjaimia käsitellään useimmiten kaksimerkkisillä muodoilla. Heksadesimaalisilla merkeillä on yksi, tai useampimerkkisiä designaatioita osoittamassa tiettyjä asioita, tai omaamassa tiettyjä merkityksiä, jotka ohjelmistojensuunnittelijoiden tulee tuntea, ja joita ohjelmien käyttäjät eivät edes tule ajatelleeksi. Voidaan käyttää seuraavanlaisia heksadesimaalisia merkkejä osoittamassa:

Levyasemaa:

DRIVE:

Yksimerkkinen heksadesimaalinen arvo indikoimassa sitä, mihin loogiseen levyasemaan tiedostoa tallennetaan. Tällöin 0=A; 1=B; 2=C; 3=D, jne.

Bittiä:

BYTE:

Kaksimerkkinen heksadesimaalinen arvo asetettavaksi osoitteeseen tai rekisteriin, tai luettavaksi sieltä.

Tallennetta:

RECKORD:

1-3 merkinen heksadesimaalinen arvo käytettäväksi indikoimaan loogista tallenteen numeroa levyllä, ja niiden levyn sektoreiden lukumäärää, jotka kirjoitetaan, tai jotka ladataan muistiin. MS-DOS järjestelmäkutsut ovat esimerkiksi 0FH =Open File; 23H=Get File Size; 3DH=Open Handle.

Arvoa:  
VALUE:

Heksadesimaalinen arvo aina neljään merkkiin saakka käytettäväksi indikoimaan portin numeroa, tai niiden kertojen lukumäärää, jotka komennon tulisi toistaa toimintonsa.

Osoitetta:  
ADDRESS:

Kaksiosainen designaatio, joka koostuu joko aakkosellisesta segmenttirekisterin designaatiosta, tai neljämerkkisestä segmenttiosoiteesta, plus offset-arvosta.

Kutakin graafista [kirjain] hahmoa vastaa oma kaksimerkinen koodinsa; jos käytettäisiin pelkkiä kirjaimia a-z tai numeroita 0-9 olisi vaihtoehtoja niukalti, mutta kun niitä käytetään yhdessä kaksimerkkisten koodien muodostamiseen saadaan numeroiden ja kirjainten lisäksi muodostettua mm. graafiset piirrosmerkit ja tietokoneohjelmissa käytetyt ohjelmakoodit. Jokaiselle seikalle on oma heksadesimaalinen koodinsa, pilkulle, välilyönnille, rivinvaihdolle jne., joiden jonoista näytölle tulostuu tiettyjä graafisia hahmoja - kirjaimia.

Siten tekstinkäsittelyohjelmien avulla luodut merkit ja tekstitiedostot koostuvat tietyistä koodeista, kuten itse ohjelmatkin. Ohjelmatiedostot kuitenkin suorittavat "ajettuina" tiettyjä tehtäviä, joita tekstitiedostot eivät tee, ja niihin sisältyy lisäksi niiden toimintaan liittyvinä erityyppisiä muistiosoittemäärytyksiä, joista jotkut liittyvät laitteen tiettyihin fyysisiin osioihin.

## 1.7. TIETOKONEEN MUISTISTA

**Tietokoneissa** on käytetty aiemmin ferriittirengasmuisteja, jotka toimivat niin, että tietty sarja niitä koostuu annetusta, vastakkaisten varausten jonosta [0 tai 1], josta syystä **tietokoneen** sanotaan käsittelevän dataa binaarimuodossa, jossa informaatioisällöt muodostuvat annetusta jonosta nollija ja ykkösiä. Ferriittirengasmuistit ovat saaneet myöhemmin monia miniatyyrikokoisia johdannaisia, joiden toimintaperiaate on kuitenkin sama. Tallennettaessa jotakin binääristä sarjaa osa renkaista siis magnetoidaan ja osa taas jää magnetoimattomiksi.

Koska XT ja AT **kotitietokoneiden** käyttömuisti on sähköinen, ei mikään, mitä niillä tehdään tallentamatta kesto- tai muistiin magneettisesti aktiiviselle levytasolle. Vaikka jotkut käytetyistä ohjelmista ovat muistiresidenttejä, merkitsee tämä vain sitä, että ne varaavat muistitilaa tietyn määrän kun ne käynnistetään, ja pitävät sen varattuna aina siihen saakka kunnes laitteesta katkaistaan virta. Tällaisetkin ohjelmat voidaan laitetta käytettäessä inaktivoida erityisillä ohjelmilla.

**Suuret tietokoneet** ovat hyvin nopeita. Siten esimerkiksi osittaiskäytössä tavittava tietokoneaika jää hyvin lyhyeksi kussakin tapauksessa. Tietokoneaika on lisäksi hyvin kallista - kuten myös se aika, jota käytetään **tietokoneita** yhdistäviä verkkoja hyödynnettäessä. Silti **tietokoneet** ovat hyvin taloudellisia, sillä niillä tosiaan pystytään laskutoimituksiin, jotka menneinä aikoina olisivat vaatineet eräissä tapauksissa jopa vuosia silloisin menetelmin, ja **tietokoneet** ovat laskukoneita - siitä huolimatta, että niillä kirjoitettaisiin tekstejä, tai niillä piirrettäisiin erilaisia kuvia.

Ilman **tietokoneille** kehitettyjä ohjelmia niillä ei tekisi paljoakaan: software on niiden kohdalla keskeisempi kuin mitä ne fyysikaaliselta olemukseltaan ovat.

## 1.8. MONITORI, NÄPPÄIMISTÖ JA TULOSTIMET

### 1.8.1. MONITORI

Monitoriin tulostuvan kuvan tai tekstisivun kokoa voi käyttäjä muuttaa määrittämällä käytetyn ohjelman parametreja, mikä kuvan kohdalla tarkoittaa kuvan haluttua tulostuskokoa, ja tekstin kohdalla tekstisivun leveyttä ja pituutta. Käyttöjärjestelmän tasolla käyttäjä voi määrittää näyttökuvan sijaintia mm. DOS:n MODE komennoilla.

Käyttöjärjestelmän tasolla monitorin oletustilan leveys on yleensä maksimissaan jopa 250 merkkiä, ja minimissään 0-merkkiä, mutta on asetettu tavallisesti arvoon 80, jossa oletustilassa monitori on, kun käyttäjä ottaa

yhteyden **tietokoneeseen**. Oletusmoodi on useimmiten ASCII, mikä tarkoittaa sitä, että monitorissa merkit ovat aakkosellisia kirjaimia ja numeroita, eikä esimerkiksi HEX-koodeja. Käyttäjä voi tietyillä ohjelmilla määrittää näytön myös sellaiseksi, että hän voi editoida haluamiaan ohjelmia.

### 1.8.2. NÄPPÄIMISTÖ

**Kotitietokoneiden** näppäimistöt ovat useimmiten IBM tyyppisiä, joissa nuolinäppäiminä käytetään omaa neljää [harmaata] näppäintään, ja näppäimistössä äärimmäisenä oikealla olevat näppäimet voidaan kytkeä numeerisiksi ja koko oikeanpuolimmainen näppäinryhmä voi toimia laskimen tapaan [NUM LOCK]. Mikäli NUM LOCK on kytkettynä, toimivat [harmaiden] nuolinäppäinten yläpuolella olevat [harmaat] INSERT, HOME, Page Up, DELETE, END, Page Down näppäimet ao. toimenpidenäppäiminä ohjelmia käytettäessä, joskin myös äärimmäisenä oikealla olevia vastaavia näppäimiä voidaan käyttää kytkemällä NUM LOCK pois. Pehmonäppäimet [tavallisesti F1-F10 NÄPPÄIMET] voivat sijaita joko näppäimistön ylä laidassa tai sen vasemmalla puolella kahdessa rivissä. Pehmonäppäimiä voi näppäimistössä olla enemmänkin kuin kymmenen, mutta kymmenelläkin tulee toimeen. Tavalliset numero- ja kirjainnäppäimet toimivat esimerkiksi tekstinkäsittelyssä kirjoituskoneiden ao. näppäinten tapaan, mutta ne voivat saada muitakin funktioita käytettäessä esimerkiksi tekstinkäsittelyohjelmia. **Kotitietokoneiden** näppäimistöllä voidaan emuloida **suurten tietokoneiden** näppäimistöjä, ja tätä varten on olemassa erillisenäkin myytäviä emulointiohjelmia. Näppäimistöä voidaan siis käyttää monessakin mielessä vaihtelevilla tavoin, ja näppäimillä ei ole pysyviä merkityksiä ja käyttöjä kirjoituskoneiden tapaan.

Ohjelmista- ja ohjelmistoista on kuitenkin tullut yhä mutkikkaampia ja niissä saattaa olla satoja optioita, joiden avulla ohjelmia ohjataan. Vaikka useimpia ohjelmia- ja ohjelmistoja voidaan ohjata näppäimistöltä, niin niihin on kehitelty HIIRIOHJAITAVIA toimintoja - joko kokonaisuudessaan tai osaksi. Hiiri on pieni laite, jolla voidaan mm. siirtää kohdistinta näytössä ja kytkeä tietty toiminto asettamalla kohdistin sen avulla aktiiviselle alueelle ja painamalla hiiren näppäintä. Ohjelman valikot koostuvat siten tällaisista aktiivisista alueista, joista hiiren avulla poimitaan haluttuja vaihtoehtoja. Erityisen ryhmänsä ohjelmissa muodostavat tässä suhteessa piirto-ohjelmat, joissa on mainitunlaisten aktiivisten alueiden [optiovalikko] lisäksi laaja alue, jonka sisällä voidaan käsitellä kuvia ja tekstejä pixeligrafiikkamuodossa. Erityisen ryhmänsä muodostavat tässä erilaiset kolmiulotteiset vektorigrafiikkaohjelmat, joissa "piirtäminen" koostuu pisteiden määrittämisestä kolmiulotteisessa tila-avaruudessa, ja joissa ohjelma suorittaa tarvittavat laskutoimitukset yhdistettäessä määritellyjä pisteitä erimuotoisilla viivoilla ja pyöräyteltäessä viiva-hahmoja ympäri tietyn kolmiulotteisen kappaleen luomiseksi. Halutut arvot pisteiden määrittelyssä voidaan antaa näppäimistöltä, mutta ohjelma

määrittää nekin käyttäjän puolesta hänen "osoittaessaan" hiirtä käyttäen kohdistimella haluamansa sijainnin kolmiulotteisessa avaruudessa, jonka "suuruuden" hän on määritellyt.

### 1.8.3. TULOSTIMET

Tekstinkäsittelyohjelmalla työstetyt kirjoitukset ja kuvankäsittelyohjelmalla muokatut kuvat tulostetaan paperille käyttäen tarjolla olevia tulostuslaitteita, mutta periaatteessa tietokoneohjelmilla työstetty materiaali voidaan tulostaa millaisilla ja minkätasoisilla tahansa laitteilla. **Tietokoneella** tuotetussa materiaalissa informaatio on useimmiten sellaisessa muodossa, että siitä voidaan tuottaa lähes millaisia tahansa tulosteita, mitä kirjoituskoneella ei voi tehdä; samasta tekstistä voidaan esimerkiksi tulostaa raakavedos ja toisaalta painojälkeä oleva tulostus esimerkiksi PostScript sivunkuvauskielillä.

#### 1.8.3.1. PostScript (Apple LaserWriter, jne.)

PostScript-sivunkuvauskieli on kirjain- ja numeromerkeistä rakentuvaa tulkattavaa ohjelmakieltä, kuten esimerkiksi BASIC. Tulostimessa on tulkiohjelmalla varustettu mikroprosessori, joka komentojen mukaan piirtää tulostettavan sivun grafiikan ja kirjaimet tulostimen muistiin. Muunnos taitto-ohjelman sisäisestä kuvauskielestä PostScript-kuvauskielen on yleensä varsin helppoa, eikä se rasita päätietokoneen keskusmuistiyksikköä. Kirjoittimen ohjausohjelma (Driver) on tässä tapauksessa varsin yksinkertainen. PostScript-kielinen määrittely siirretään kirjoittimelle jatkokäsittelyä varten yleisimmin **tietokoneen** sarjaportin kautta, ja siirrettävä tietomäärä on Postscript-kielen rakenteen vuoksi erittäin suuri. Erityisesti täysgraafisten kuvien siirto on tästä syystä erittäin hidasta. Johtuen siitä, että PostScript on tulkattava ohjelmointikieli on komennot ensin tunnistettava ja käännettävä konekielille, jonka prosessori kykenee suorittamaan. Tämä on työläs ja aikaavievä operaatio, joka vaatii kirjoittimen prosessorilta suurta suorituskykyä, mikä on syynä (yhdessä PostScriptin käyttöön aiemmin liittyneiden lisenssimaksujen kanssa) kirjoitinten korkeaan hintaan.

PostScriptin tekniikka ei rajoita käytettävien kirjainleikkausten ja kokojen määrää, mutta on erittäin hidasta. Erityisesti otsikoiden, lihavoitien, isojen alkukirjainten (anfangien), bullettien (erikoismerkki kappaleen alussa), kursivoitien ja muiden tehokeinojen käyttö on tavattoman hidasta. Tämä johtuu siitä, että vain muutamaa kirjainta varten tulostin laskee bittikartat koko 250:n merkin kirjainsarjasta. PostScript on muodostunut standardiksi, koska se on yksinkertainen ja varma tapa liittää tulostin mihin hyvänsä ohjelmaan. Se on yhteensopiva myös muutamien valolatomakoneiden kanssa, mikä mahdollistaa tulostettavien töiden siirron kirjepainoon esimerkiksi levykkeellä

tai modeemiyhteydellä puhelinverkossa. Tulostaminen kirjapainossa ei edellytä, että oman lasertulostimen olisi oltava PostScript-tulostin. Ratkaiseva seikka on se, että taitto-ohjelmalla on käytössään latomakoneen kirjaimistojen kanssa yhteensopivat leveysarvot. Vedostaminen voidaan siten tehdä vaikka pistematriisikirjoittimella, kunhan käytetään oikeita leveysarvoja. Latomoa varten tulosohjaimeksi vaihdetaan PostScript-ohjain ja tulostus osoitetaan tiedostoksi levykkeelle.

#### 1.8.3.2. HP-Laser ja yhteensopivat (HP-Laser, PCL, jne.)

HP-Laserissa ja sen kanssa yhteensopivissa kirjoittimissa käytettyä ratkaisua on sanottu "perinteiseksi". Tekninen toteutus vastaa tekstinkäsittelystä tuttua kirjoittimen liittämistapaa (rinnakkaisporttiliitäntä). Taitto-ohjelman sivunkuvauskieli tulkitaan kirjoittimen ohjausohjelmassa (Driver). Sivun kuva muunnetaan kirjoittimen ohjauskielille, joka yleensä on varsin yksinkertaista, minkä vuoksi itse muunnos on työläs ja aikaavievä. Muunnos joudutaan suorittamaan päätietokoneen keskusyksiköllä, jota ei yleensä ole suunniteltu tehokasta grafiikankäsittelyä silmälläpitäen.

Kun tulostettava sivu on muunnettu kirjoittimen ohjauskielille se lähetetään kirjoittimeen rinnakkaisportin kautta, mikä muodostaa järjestelmän pahimman pullonkaulan. Erityisesti kuvanlukijalla (skannerilla) luettujen kuvien tai piirto-ohjelmilla tuotettujen kuvien siirto kirjoittimeen on aikaavieppää hitaan rinnakkaisliitännän kautta. Kirjoittimessa oleva ohjausprosessori tulkitsee kirjoittimen ohjauskieliset käskyt ja maalaa lopullisen sivun kuvaelementtitasolla. Suorituskykyyn vaikuttaa se, mihin käytetyt kirjasintyypit on varastoitu. Taitto-ohjelmien yhteydessä kirjasintyypit sijaitsevat **tietokoneen** levymuistissa, ja ne on tulostuksen yhteydessä ladattava kirjoittimeen. Rinnakkaisliitännän kautta suoritettuna tämä edelleen hidastaa toimintaa, mutta sallii samalla suuremman kirjasintyyppivalikoiman. Kirjasintyypit (Fontit) voivat sijaita myös itse kirjoittimessa ns. fonttikaseteissa, jolloin tyyppien siirtoon ei kulu aikaa, mutta valikoima on tällöin rajoitettu.

Kummassakin tapauksessa käytettävissä on rajattu määrä kirjasinkokoja, ja suurin käytettävissä oleva koko on tyypillisesti 28-36 pistettä (10-12mm). Koko rajoittaa myös tulostimen muistin määrä - esimerkiksi A4 -kokoisen sivun kuvaa varten tarvitaan n. 1.3 megatavua muistia. Loppu muisti tarvitaan käytössä olevien kirjainkuvasarjojen tallennukseen. Koska jokaisesta käytetystä kirjaintyylin ja pistekoon yhdistelmästä on tallennettava koko merkkisarjan kuvat tulostimen muistiin, ei tällä teknologialla totetutetun tulostimen kanssa voi käyttää useita suurikokoisia tekstityylejä samalla sivulla.

Tavallisimpia tulostettavia ovat erilaiset kirjoitukset, joita on työstetty tai tuotettu tekstinkäsittelyohjelmilla, joista haluttu tarjolla oleva tulostin valitaan ennen tulostusta. Samoin on laita tulostettaessa pixeligrafiikkaa.



Kolmiulotteisia kuvioita ei tietenkään ole mahdollista tulostaa paperille kolmiulotteisina muutoin kuin näyttämällä ne tietystä tarkastelukulmasta ja tulostamalla kuva perspektiivisenä. Kolmiulotteista grafiikkaa voidaan käsitellä kaksikulotteisilla pixeligrafiikkaohjelmilla konvertoimalla ne asianmukaisiksi tiedostoiksi. Tulostamisessa voidaan käyttää myös väritulostimia.

## 1.9. TIETOLIIKENNÖINTI

### 1.9.1. KOTITIETOKONEET JA SUURKONEET

On tavallista, että **suurissa tietokoneissa** käyttäjät eivät käsittele laitteen käyttöjärjestelmää, vaan sen tekevät erityiset koneen toiminnasta huolehtivat henkilöt, jos aihetta ilmaantuu. **Kotitietokoneissa** käyttäjän on mahdollista käsitellä ja konfiguroida koneensa käyttöjärjestelmää monin tavoin, ja hänen täytyykin osata hallita oman laitteensa käyttöjärjestelmää. Laitteensa tuntemus säästää pitkällä tähtäimellä rahaa mm. alentuneina konsultaatio- ja huoltokustannuksina. Ei ole kovinkaan todennäköistä, että kukaan suostuisi konsultoimaan ilmaiseksi **kotitietokoneen** käyttäjää pitkällä tähtäimellä, tai korjaamaan laitteita pelkästään ystävänpalveluina. Niin suuret kuin pienetkin vahingot **tietokoneiden** käytössä eivät johdu **tietokoneista** laitteina ja järjestelminä, vaan niiden väärästä käytöstä ja tehdyistä [inhimillisistä] virheistä. Tietoliikennöinnissä käyttöjärjestelmien sisäisen kielen osaaminen korostuu, koska sen avulla voidaan hallita muitakin **tietokoneita** kuin omaansa.

Yleisin tapa kontaktoida **suuren tietokoneen** kanssa on käyttää siihen fyysisesti kytkettyä päätettä. Esimerkkinä tällaisesta olkoon tapa, jolla käytin JYKDOK'ia Jyväskylän yliopiston kirjaston aarteistoa tutkittaessa vuonna 1987, joskin myös **kotitietokone** oli mukana eräänä välineenä. Esitän tämän esimerkinä siksi, että käyttämäni Jyväskylän yliopiston ensimmäinen **kotitietokone** oli tavattoman hidas, koska siinä ei ollut muuta kuin 2 kalvovevykeasemaa, käyttöjärjestelmä MSDOS 2.01, ja se oli kytketty SPERRY-yhteyden kautta VAXIIN KERMIT-ohjelman kautta, jossa voidaan käyttää erilaisia komentoja, joiden avulla mm. hallitaan toisen **tietokoneen** eräitä toimintoja niin, että "siellä" voidaan työskennellä:

```
BYE  
to remote server.
```

```
CLEAR  
key redefinitions.
```

```
CLOSE
```

log file and stop logging remote session.

COMMENT  
a command file.

CONNECT  
as terminal to remote system.

CWD  
change local working directory.

DEFINE  
macros of Kermit-MS commands.

DELETE  
local files.

DIRECTORY  
listing of local files.

DO  
a macro expansion.

EXIT  
from Kermit-MS.

FINISH  
Shut down remote server.

GET  
remote files from server.

HANGUP  
the phone.

HELP  
about Kermit-MS.

LOCAL  
prefix for local file management commands.

LOG  
remote terminal session.

LOGOUT  
remote server.

PUSH  
to MS-DOS command level.

QUIT  
from Kermit-MS

RECEIVE  
files from remote Kermit.

REMOTE  
prefix for remote file management commands.

RUN  
an MS-DOS program.

SEND  
files to remote Kermit.

SERVER  
mode of remote operation.

SET  
various parameters.

SHOW  
various parameters.

SPACE  
inquiry (about disk space).

STATUS  
inquiry (about settings).

TAKE  
commands from file.

TYPE  
display a local file.

VERSION  
display Kermit-MS program version number.

Edellä on luettelo komennoista, jotka "synkkasivat" käytössäni olleen PC-

**kotitietokoneen** MS-DOS-käyttöjärjestelmä 2.01 kanssa. [Nimitys DOS viittaa siihen, että PC-**kotitietokoneen** käyttöjärjestelmän systeemi, sekä prosessori ladataan käynnistettäessä koneen muistiin disketiltä, tai laitteeseen asennetulta kovalevyiltä. Käyttöjärjestelmää ei siis tällöin ole laitteeseen sisäänrakennettuna]. Tiettyjen protokollien jälkeen sain näytölle seuraavanlaisen tiedotteen, johon olen lisännyt tiedotteista seuraavat toimenpiteet:

kone: ENTER USERID/PASSWORD CLEARANCE LEVEL  
anna käyttäjätunnus/salasana  
kone: ENTER ACCOUNT NUMBER  
anna laskutusnumero  
kone: Oletusmoodi on ASCII  
näppäille call jykdok  
näppäille hakusana  
kone: 1 jokin lukumäärä HAKUSANA  
näppäille p  
kone: ensimmäinen hakusanan sisältämä teos  
paina RETURN  
kone: toinen hakusanan sisältämä teos  
jne.

**Suurtietokoneessa** on käytössä omat käskynsä, ja työskenneltäessä siellä on oltava käytössä tulkki, jonka avulla tietty joukko ymmärrettäviä komentoja voidaan antaa. Komennoista sain tietoa tiedustelulla:

```
CS/200T sh ?  
SHow ADDRess  
SHow DefaultParameters Äaram-name ...è  
SHow GLobalParameters  
SHow NAMES Ähost nameè  
SHow NetMAP Ä Short ö Long è  
SHow PARAMeterS Äaram-name ...è  
SHow aram-name ...  
SHow SESsions Ä P è  
SHow VERSion
```

Myös erilaisista **suurtietokoneeseen** asennetuista palveluista on oltava selvillä. Seuraavassa tuolloisia Jyväskylän yliopiston **suurtietokoneen** palveluja:

Internet Name `\*' ==

b\_2kerr mac\_303 mab\_306 mad\_tlh3  
jylk mac\_115 cs\_8 jylkf  
JYTKO JYTKOSUN SPERRY JYLFK  
mab\_221 ivecs mad\_tlh1 mad\_tlh2  
jytco jytkosun dca sperry  
ncs JYLK t\_230

**Suurtietokone** antaa istunnon alkaessa myös tietoja siitä, mitä sillä on tietyn palvelumuodon sisällä tarjolla, kuten esimerkiksi JYKDOKISSA:

871008(UP), korjauksia 230 871001(UP). Matematiikan vanhoja kirjoja lis18:54  
870800(UP). Opinnäytetyöt komennolla: BASE GRADU (Sis käytössä: BASE FENNICA)  
SUOMENKIELINEN KESKUSTELU KOMENNOLLA: LA FI ,APUA: HELP ,UUTISET: NEWS

Mikäli kontakti **suurtietokoneeseen** otetaan [minun tapaani] **kotitietokoneella**, käytetään tavallisesti KERMIT ohjelmaa, joka sisältää mahdollisuudet tarvittavien protokollien suorittamiseen ja tarvittavan kielen [käskyt] työskentelyä varten. **Kotitietokoneella** työskennellessäni saatoin "siirtää" näyttöruuduittain omaan **kotitietokoneeseeni** tietoja niitä myöhemmin rauhassa tarkastellakseni, tai paperille tulostaakseni. Seuraavassa eräistä **kotitietokoneeseeni** siirtämiäni näyttöruudullisia, joita sain JYKDOK -kokoelmasta hakusanoilla psychedelic ja mushrooms:

<MS> 6 1 PSYCHEDELIC

<MS>--- 1 ---

<MS>NO : 087754767X

<MS>MI : Furst, Peter E.

<MS> Mushrooms : psychedelic fungi / Peter E. Furst. - New York :

<MS> Chelsea House, 1986. - 127 s. : kuv. - (The encyclopedia of

<MS> psychoactive drugs). -

<MS> ISBN 0-87754-767-X

<MS> UDK 615.21:615.322

<MS> UDK 613.83:615.322

<MS> \*\*P Lääk (1)

<MS>--- 1 ---  
<MS>NO : 087754767X  
<MS>MI : Furst, Peter E.  
<MS> Mushrooms : psychedelic fungi / Peter E. Furst. - New York :  
<MS> Chelsea House, 1986. - 127 s. : kuv. - (The encyclopedia of  
<MS> psychoactive drugs). -  
<MS> ISBN 0-87754-767-X  
<MS> UDK 615.21:615.322  
<MS> UDK 613.83:615.322  
<MS> \*\*P Lääk (1)

Kommunikoitaessa eri **tietokoneiden** välillä on kyettävä ohjaamaan kohdistinta myös silloin kun näppäinten nimet ja merkitykset poikkeavat kommunikoivissa **tietokoneissa**. Seuraavassa esitetään ANSI (VT102) moodin toiminnot, jolloin kohdistinta ohjataan näyttöruudussa käyttämällä esimerkiksi seuraavia ESCAPE käskyjä:

ESC # 5 DECSWL  
Rivin yksinkertainen korkeus ja pituus

ESC # 6 DECDWL  
Yksinkertaisen rivikorkeuden kaksinkertaistettu pituus

**Heath-19 toiminnot ei-ANSI moodissa**; pari esimerkkiä:

ESC A HCUU  
Kohdistin ylös

ESC B HCUD  
Kohdistin alas

ESC C HCUF  
Kohdistin eteenpäin, voi hypätä seuraavalle riville

ESC D HCUB  
Kohdistin taaksepäin, voi hypätä edelliselle riville

**Heath toiminnot ANSI moodissa**; pari esimerkkiä:

ESC [ s PSCP

Tallenna kohdistimen asema ja attribuutit

ESC [ u PRCP

Palauta kohdistimen asema ja attribuutit

Seuraavassa muutamia termejä, joista tärkeimistä seuraavassa lyhyt luonnehdinta:

KOHDISTIN

Cursor

tarkoittaa kohdistinta, joka näkyy monitorissa viivana tai neliönä - riippuen sille annetusta määrittelystä.

VASEN YLÄNURKKAUS

Home

tarkoittaa monitorin vasenta ylänurkkausta, johon kohdistin voidaan esimerkiksi palauttaa jostakin muusta osasta sitä.

RIVIN LOPPUPÄÄ

End

tarkoittaa rivin loppupäätä.

PYYHKIMINEN

Erase

tarkoittaa poispyyhkimistä, jonka voi määritellä koskemaan esimerkiksi yhtä merkkiä, riviä- tai kaikkea, mitä monitorissa näkyy

**Tietokoneen** näppäimistö ja monitori ovat sellaisessa yhteydessä keskenään, että käyttäjä näkee useimpien näppäinten käytön monitorissa havaittavina muutoksina. Varsinkin silloin, kun käyttäjä käyttää jotakin ohjelmaa. Jos yhteydenottoon on käytetty **kotitietokonetta**, ja se on emuloitu tietyllä tavoin, ja siinä on värikortti- ja monitori, niin tällöin voi olla tarpeen määritellä myöskin käytetyt värit.

Kun istunto on lopetettu, antaa **suurtietokone** vielä teknisiä tietoja istunnosta, sekä lopuksi ilmoituksen käytetystä ajasta ja istunnon ajankohdasta, toivottaen käymään toistekin talossa.

CS/200T dc  
Disconnecting...session 1 -- disconnected from sperry(128.214.007.009 )  
CS/200T  
Esc-chr: ^] help: ^]? port:1 speed: 9600 parity:none echo:rem  
VT102 .....

- Come back soon  
820 9073005H 820 0 min 22.0 sek 23/10 1987  
18:45:16

## 1.10. JÄRJESTELMÄT

### 1.10.1. TIETOJÄRJESTELMÄN RAKENTAMINEN

Tietojärjestelmien rakentamiseen tulisi liittyä niin systeemi- kuin operaatioiden analyysi. Edellinen liittyy lähinnä niiden struktuurien luomiseen, joiden puitteissa tullaan työskentelemään, ja jälkimmäinen niihin tehtäviin, ongelma-asetelmiin ja erityisprojekteihin, joita luodun systeemin piirissä suoritetaan. Tietojärjestelmien kehittämisessä sovelletaan teoreettisena viitekehyksenä ja välineinä yleisesti joukko-oppia ja lauselogiikkaa, sekä verkko-, ja systeemiteoriaa. Joissakin sovelluksissa keskeisiä ovat myös erilaiset kieliteoriat ja niiden vaikutus tietokonejärjestelmiin, mm. denotaatio-teoria ja entiteetti-attribuutti relaatio -tietomalli. Tietojärjestelmien rakentamiseen liittyen on syntynyt myös useita koulukuntia (mm. sosiotekninen koulukunta) ja tietojärjestelmien kehittymisen kuvauksen yhteydessä on käytetty myös evoluutiofilosofian piiristä tuttuja skeemoja, kuten kilpailukyvyn paraneminen - erityisesti yritysmaailman suunnasta katsoen tietojärjestelmissä keskeisiä ovat juri kilpailukykyä lisäävät ominaisuudet, kuten myös resurssien ja kustannusten minimointi suhteessa parantuneeseen tuotokseen.

Rakennettaessa tietojärjestelmää on siis ymmärrettävä sen rooli aiotussa toimintayksikössä ja ympäristössä yleisimmin tilaajan sanelemin teoreettisin lähtökohdin, sekä myös valmiin tietojärjestelmän perusrakenne niiden suhteen. Onnistuneen tuloksen saavuttamiseksi tulisi mieluusti myös ymmärtää tietojärjestelmien laajempikin rooli yhteiskunnassa, ja ihmisten tuleva suhde järjestelmiin sen käyttäjinä ja potentiaalisina operaatioiden kohteina. Tietojärjestelmien kehittämiseen liittyvät eräinä kuvaavina käsitteinä metodologia, tekniikat, merkintätavat, rajoitteet, valinta- ja sen kriteerit, sekä systeemin ositus. Informaatioanalyysin puolelta siihen liittyvät edeltävyys,-komponentti-, ja prosessianalyysi. Tietojärjestelmän rakentamismallin ja vaihejaon tiimoilta olisi ymmärrettävä rakentamisen kokonaiskuva, jotta voitaisiin ylipäätään kartoittaa koko tehtäväkenttä oikein. Tapahtumien



suunnittelussa voidaan soveltaa myös transaktioanalyysia.

Työskentelyn kuluessa olisi kontruoitava ja hahmotettava tietojärjestelmän käyttäjän ja sen suunnittelijan välinen kommunikointi - esimerkiksi jo suunnittelun aikana ja sen jälkeen tukihenkilöiden taholta toteutettuna - sekä määriteltävä tilaajien osallistuminen suunnitteluun ja vaatimukset, joita heiltä tällöin edellytetään, sekä kartoitettava käyttäjäorganisaatiot. Esitutkimus olisi siten ymmärrettävä osana systeemisuunnittelumetodologiaa, muutos,- toiminta- ja tietoanalyysia yhtyneenä usein evoluutiofilosofisten tendenssien huomioimiseen tilaajien toiminnoissa.

Tämä kaikki liittyy spesifeinä komponentteina yleiseen tietojärjestelmän tavoitteen ja sisällön asetukseen, periaatteisiin ja vaiheistukseen, liikeprosessien analysointiin ja tieto- sekä muotoiltavaan sovellusarkkitehtuuriin.

Tietojärjestelmän edelleen kehittäminen olisi myös organisoitava ja luodattava mahdollinen epävarmuus, ja hallittava se. Sovellussuunnittelun ongelmiin olisi paneuduttava erityisesti avainkohtinaan informaatio,- komponentti,- ja tietoanalyysi, sekä prosessin- ja dialogien suunnittelu, sekä myös laadun valvonta ja tietokantojen suunnittelu.

### **1.10.2. TIETOKONEJÄRJESTELMÄT**

Olisi kartoitettava (aiottuihin tehtäviin liittyen) erilaiset tarjolla olevat tietokonejärjestelmät ja hallittava niissä käytetty konekieli, jotta laitteita voitaisiin konfiguroida ja ohjata käyttöjärjestelmien puitteissa, sekä ymmärrettävä siihen liittyen **tietokoneiden** tekninen toteutus. Tässä yhteydessä olisi kartoitettava erilaiset tarjolla olevat massamuisti- ja säilytyslaitteet, sekä oheislaitteet, sekä ymmärrettävä niiden ohjaus ja niiltä vaaditun kapasiteetin suuruus. Olisi perehdyttävä myös vaihtoehtoisesti erilaisiin käyttöjärjestelmiin ja niihin liittyviin tietoliikennöinnin, kiintolevypartitionin, oheislaittekapasiteetin jne. ongelmiin, jotta valittaisiin kokonaisuus, joka vastaa parhaiten tilaajan vaatimuksia.

### **1.10.3. TIETOKANNAN HALLINTAJÄRJESTELMÄT**

Tietokantojen rakentamiseen ja hallintaan liittyvät sentapaiset käsitteet ja toiminnot, kuten relaatiomallien rakenne, relaatioiden muodostaminen, normalisointi, predikaattilogiikka, relaatioalgebra- ja kalkyyli, SQL-kieli, samanaikaisuuden hallinta, kyselyjen optimointi, suojaukset, hajautetut tietokannat, logiikkatietomallit, tietokantakoneet ja oliokeskeiset tietokannat- ja kielet. Tietokantojen rakentaminen liittyy saumattomasti tietojärjestelmän suunnitteluun ja tietokantojen rakenne tulisikin suunnitella järjestelmän

rakentamisen yhteydessä.

## 1.11.

### ERÄITÄ SOVELLUSALUEITA OPETUKSESSA

Tein seuraavan hahmotelman vuonna 1986 itse tutustuessani ensimmäistä kertaa **kotitietokoneisiin**, ja pohtiessani **tietokoneiden** mahdollisuuksia yliopistollisen opetuksen piirissä.

#### 1.11.1. Työryhmien työskentelyn apukeino

Työryhmä -tai tiimiluonnetta omaavien tutkimushankkeiden etenemisessä **kotitietokoneet** saattavat muodostua työtä huomattavastikin helpottaviksi apukeinoiksi; kerättyyn materiaalin sähköinen siirrettävyys, ja materiaalin kiinteä kokonaisuus eivät edellytä työryhmän jäsenten fyysistä paikallaanoloa suhteessa toisiinsa. Epäilemättä **tietokoneet** lisäävät myöskin työn koordinoitavuutta.

#### 1.11.2. Useiden töiden samanaikainen tekeminen

**Kotitietokoneilla** on mahdollista varsin vaivattomalla tavalla työstää samanaikaisesti useita tekeillä olevia töitä: ei ole tarpeen etsiä materiaalia, ja hajoittaa ajatuksiaan paperien penkomisella tms., koska kalvolevyille- ja erityisesti kiintolevyille mahtuu lähes uskomaton määrä kirjoitettua tekstiä, ja hyvin suunniteltu työ muodostuu helpoksi. Tietenkin tämä kaikki vaatii tiivistä harjoittelua kirjoituspaneelin kanssa; oikotietä ei ole olemassa, koska tiettyjen AJATTELUN TAPOJEN on muututtava. On tunnettava koneen tapa toimia - olematta tietenkään itse kone; asioita on tavallaan kyettävä hallitsemaan laajasti.

#### 1.11.3. Luentojen apukeino

Sovelliaalla tavalla taltioidut tiedot voivat olla elastinen, havainnollistava materiaaalipohja, josta poimitaan tilanteen mukaan mitä erilaisimpia tietoja, ja opiskelijoille voitaneen kopioida levykkeille, tai papereille soveltuvia osia - myöhempää tarkastelua ja perehtymistä varten. Tämä edellyttää kuitenkin, että heillä on käytössään tarvittavat laitteet. Tämä edellyttää luennoitsijan kohdalla ajattelun tapain muuttumista, sillä erotuksena aiempiin metodeihin, on tämän metodin vallitsevina piirteinä mm:

a) tietty ennakoivuus

b) havainnollisuus ja materiaalin suurempi jaettavuus

c) luentojen aikana ilmenevien ideain välitön kirjattavuus

d) tiettyjen materiaaliosioiden välitön eliminoitavuus, mikäli syntyvä kokonaisuus sitä näyttää edellyttävän

e) opiskelijain mahdollisuus lisätä materiaaliin omia ideoitaan luentojen ulkopuolisellakin ajalla, mikäli heille sallitaan pääsy materiaaliin.