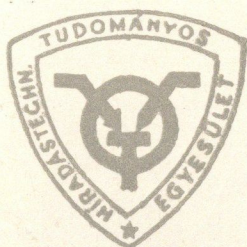


E 870

HÍRADÁS- TECHNIKA



A HÍRADÁS-
TECHNIKAI
TUDOMÁNYOS
EGYESÜLET
LAPJA

10

XXII. ÉVFOLYAM, 10. SZÁM, 289—320 OLDAL, BUDAPEST, 1971. OKTÓBER HÓ

TARTALOM

Számítógépek alkalmazása	289	✓
✓ TREMBECZKI SÁNDOR: A System 4—50 típusú számítógép hardware felépítése	290	✓
✓ KISS DÉNES—GAJÁRI GYULA: A System 4-es operációs rendszer	296	
Könyvismertetés	301	
Pályázati felhívás	301	
✓ SÁGODY ISTVÁN: A System 4 típusú számítógépek adatkezelési rendszere	302	✓
✓ DR. GRACZA LAJOSNÉ: Az ICL System 4 szállítási feladat programcsomagjának ismertetése	311	✓
✓ MAJOROS KÁSZON: A System 4 PERT programcsomagja	314	✓
Szemle	310, 313, 316	
Tartalmi összefoglalások	319	
Обобщения	319	
Zusammenfassungen	320	
Summaires	320	
Résumés	B/III	

Szerkesztőség: BOGLÁR GYULA főszerkesztő, SZÖLLŐSI GYÖRGYNE szerkesztőségi titkár, BALOGH PÁL, DR. SÁRKÖZI GÉZA kandidátus és MAY PÉTER tudományos szerkesztők, DR. FLESCHE ISTVÁN, DR. RUPPENTHAL PÉTER szerkesztőségi munkatársak. — A szerkesztőség címe: Budapest II., Mártírok útja 85. I. em. 140. Telefon: 183-772 — A Híradástechnikai Tudományos Egyesület címe: Budapest V., Szabadság tér 17. Telefon 113-027

Szerkeszti a szerkesztőbizottság

INDEX: 25.375

HÍRADÁSTECHNIKA

Kiadja a Lapkiadó Vállalat Budapest VII., Lenin körút 9—11. Telefon 221-285. Felelős kiadó: SALA SÁNDOR igazgató. Terjeszti a Magyar Posta. Előfizethető bármely postahivatalnál, a kézbesítőknél, a Posta hírlapüzleteiben és a Posta Központi Hírlapirodánál (KHI, Budapest V., József nádor tér 1.) vagy közvetlenül postautalványon, valamint átutalással a KHI 215—96162 pénzforgalmi jelzőszámra. Előfizetési díj: félévre 36 Ft, egész évre 72 Ft. Egyes szám ára: 6 Ft. Megjelenik havonta. A folyóirat külföldre előfizethető: „KULTURA” P. O. B. 149 Budapest, 62.

71.6499 Egyetemi Nyomda, Budapest. Felelős vezető: JANKA GYULA igazgató

Számítógépek alkalmazása

Örvendetes, hogy iparágunkban az utóbbi időben a számítógépek száma gyorsan növekszik. Lapunk évek óta fontos feladatának tekinti, hogy a számítógépek híradástechnikai ipari alkalmazását előmozdítsa, ennek érdekében számos erre vonatkozó cikket közöltünk és rendszeresen beszámolunk a kidolgozásra került számítógépprogramokról.

A közelmúltban több nagyvállalat és kutatóintézet létesített számítóközpontot és ezek egyre jobban kiépülnek. Műszaki célú alkalmazásokra a gépek egy része még nincs optimálisan kihasználva. Régebben végzett szakembereink még kevéssé ismerik a számítógépes eljárásokat és ezért nem is élnek a számítógép adta lehetőségekkel. Ezen a téren a Híradástechnikai Tudományos Egyesület igen sikeres tanfolyamok szervezésével igyekszik segíteni, mi pedig lapunk ilyen tárgyú celszámaival kívánunk olvasóink rendelkezésére állani.

A Magyar Híradástechnikai Egyesülésbe tömörült vállalatok közös beruházással számítóközpontot létesítettek, mely alapkiépítésben jelenleg már üzemben is van. Az új számítóközpont fiatal szakemberei nagy lelkesedéssel és jó felkészültséggel fogtak hozzá, hogy megfelelő műszaki programkönyvtárat is létrehozzanak. A közeljövőben számos vállalat veszi majd igénybe mindennapi rutin munkáknál a központ szolgáltatásait és ezáltal igen sok olyan mérnök is gépközelbe kerül, akiknek eddig még nem volt módjuk, hogy feladataikat számítógéppel oldják meg. De nem csak a kutatás és műszaki tervezés, hanem a technológia, a termelés és a termelésirányítás, azaz a műszaki-gazdasági vezetés is sokat profitálhat a számítógép széleskörű igénybevételétől.

Jelen celszámunkat a Magyar Híradástechnikai Egyesülés Számítóközpontjának munkatársai írták azzal a céllal, hogy a központ most működő gépi komplexumát, az adattárolási módszereket és az operációs rendszert ismertessék főbb vonásaiban. A számítógép alkalmazásának a gép ismerete nem feltétele, de úgy véljük, hogy bizonyos fokú ismerete az általános műszaki intelligencia szerves része.

A jövőben rendszeresen szeretnénk beszámolni a Magyar Híradástechnikai Egyesülés Számítóközpontjának és más számítóközpontok híradástechnikai tárgyú munkájáról és eredményeiről. Celszámunkhoz készült Jámber István cikke, mely a jelen lapba nem fért be, egyik legközelebbi számunkban jelenik meg.

A cikkek számos szakkifejezése idegen, sokszor a magyar szöveg „angolnak” tűnik. Szakembereink ma még így beszélnek és így írnak, mert a számítógéptudomány annyira sok új fogalmat hozott, hogy a magyar terminológia még nem alakult ki, de ennek mielőbbi létrejöttét feltétlenül szükségesnek tartjuk.

Szerkesztőség

A System 4–50 típusú számítógép hardware felépítése

ETO: 681.321.11 System 4

A System 4 gépcsalád berendezései korszerű elektronikus adatfeldolgozó rendszerek kiépítését teszik lehetővé. A rendszeren belül a készülékek és a programok nagyfokú kompatibilitását biztosítják.

A 4–50 típusú központi egység a sorozaton belül közepes teljesítményűnek számít. Logikai áramköreiben igen gyors (6 ns késleltetésű) ECL típusú integrált kapukat használnak fel. A mikroprogramozott vezérlő rendszer 144 gépi utasítás végrehajtására képes, ezek maguk is már viszonylag bonyolult feladatokat végeznek el.

A központi egységhez a perifériák szabályos csatlakozóáramkörön (standard interfacen) keresztül csatlakoznak. Az input-output műveletek minimális időre vonják el a központi egységet a program feldolgozásának munkájától.

A hardware-felépítés elősegíti hatékony operatív rendszer kialakítását. Ez utóbbi a gép munkáját úgy irányítja, hogy lehetőleg mind a központi egység, mind a perifériák ideje ki legyen használva minimális gépkezelői beavatkozással (multiprogramozás). A központi egység megszakítási (interrupt) rendszere, programállapotai, tárcímzési és input-output megoldásai mind ezt a célt szolgálják.

1. Információ ábrázolása a gépben

A System 4–50 központi egység fixpontos bináris, lebegőpontos bináris, decimális és logikai jellegű adatokon tud műveleteket végezni. Az információ ábrázolása egységes formai rendszerben történik.

1.1 Az információ formai szerkezete

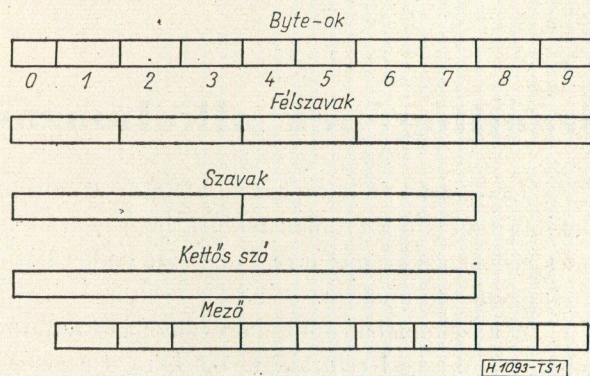
A gépben az információábrázolás alapegysége a byte. Ez 8 bitből áll, néha páratlan paritású ellenőrző bit kíséri. A byte tartalmát gyakran két hexadecimális karakter megadásával írják le.

A byte az a legkisebb egység, amit az operatív memóriából ki lehet olvasni vagy oda be lehet írni. A tárban minden byte-nak egyedi címe van. A cím a rendelkezésre álló tárméretben belül tetszőleges lehet.

Több byte felhasználásával épülnek fel a nagyobb információs egységek: a félszó, a szó, a kettős szó és a mező. A félszó a memóriában egymás mellett elhelyezkedő két olyan byte, amelyek közül az első címe (együtt a félszó címe) páros. A szó négyel oszttható címen kezdődik és négy byte-ból áll, a kettős szó nyolccal osztható címen kezdődik és nyolc byte hosszú.

Egymás mellett elhelyezkedő byte-okból úgynevezett mező is kialakítható. A mezőt a bal szélső byte címe és a benne foglalt byte-ok száma írja le.

A mező címe tetszőleges érték lehet. Aritmetikai mező hossza legfeljebb 16, logikai jellegű mezőé legfeljebb 256 byte. Az információ ábrázolásának formai kereteit összefoglalva az 1. ábra mutatja.



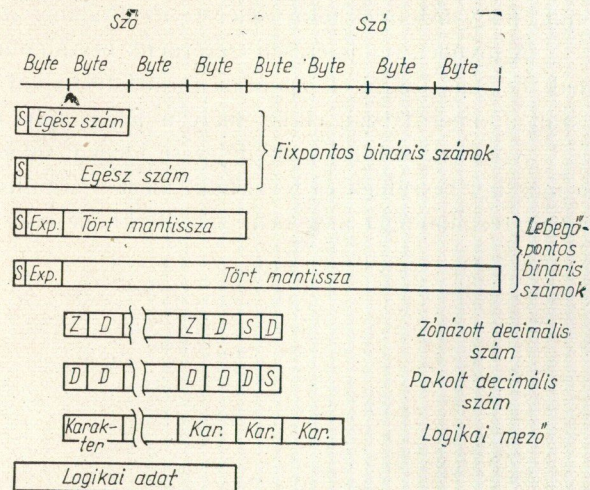
1. ábra. Az információ ábrázolásának formai egységei

1.2 Információábrázolás módja

A gépben ábrázolható adattípusokat összefoglalva a 2. ábra mutatja.

Fixpontos bináris számok félszó vagy szó hosszúak lehetnek. A legértékesebb helyen álló bit adja meg az előjelet, 1 érték jelzi a negatív számot. Ezután 15, illetve 31 biten a szám ábrázolása következik pozitív számnál egyenes, negatív számnál kettős kiegészítő kódban. A fixpontos bináris aritmetika csak egész számokkal dolgozik.

A félszavas ábrázolás pontossága korlátozott (szám-tartománya kb. ± 32 ezer), de a műveleti idő rövid és a tárat igen jól kihasználja. Korlátozott pontosságú



S = előjel; Z = zóna; D = decimális számjegy

H 1093-TS2

2. ábra

vagy szabályozástechnikai feladatokra igen alkalmas. A szavas ábrázolás (tartománya kb. ± 2 milliárd) pontossága minden gyakorlati igényt kielégít.

Lebegőpontos ábrázolásban a számot az alábbi ki-fejezés írja le:

$$x = \pm \text{mantissza} \times 16^{(\text{karakterisztika}-64)}$$

A szám gépi ábrázolását a 2. ábra mutatja. Az előjel a mantissza előjele. A mantissza hexadecimális tört-szám. A kitevőt a gép 64 többlétes kódban ábrázolja. Az így kapott karakterisztika 7 bites bináris kód, amelyet mindig pozitív számnak tekintünk. Így a kitevő -64 és $+63$ közötti érték lehet, ami kb. 10^{-78} és 10^{76} közötti számtartománynak felel meg.

A mantissza rövid lebegőpontos számnál 24 bit-ből, hosszú számnál 56 bitből áll. Ez a pontosság tudományos számításokhoz is kielégítő.

A logikai adatok szóhosszúságúak vagy mezők lehetnek. A logikai műveletek a byte-ot bitelrendezésnek tekintik, nincs előjel, kitevő stb. Ilyen például az alfanumerikus információ ábrázolása.

Az input-output műveletek során minden egyes karaktert a tárban 8 bites EBCDIC kód ábrázol, amely egy-egy byte-ot foglal el. Az EBCDIC kód a számokat binárisan kódolt decimális formában írja le. A szám kódja a byte jobb oldali felén helyezkedik el, a másik felet egy „zóna” foglalja el. Információt csak az utolsó számjegy zónarésze hordoz: az előjelet.

Ez a „zónázott” forma a tárat rosszul használja ki. Decimális aritmetikai műveletek végrehajtása előtt a számot „pakolt” formára hozzák, vagyis a felesleges zónák helyére nagyobb helyértékű számjegyeket írnak. Így a szám körülbelül fele akkora memóriatér-be tömöríthető. A decimális műveletek operandusai és eredményei mindig pakolt számok. Egy adatot kiírása előtt zónázott formára kell hozni.

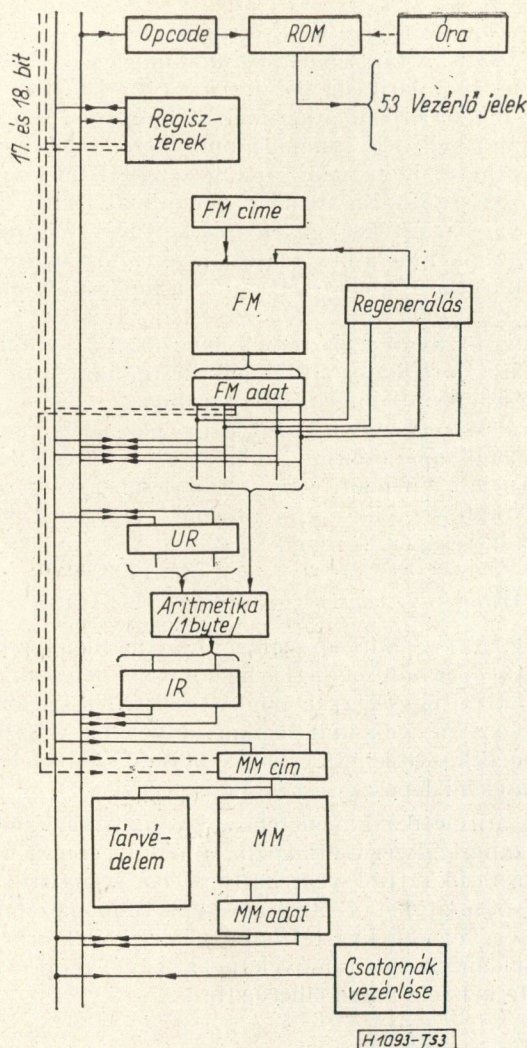
2. A gép tömbvázlata

A 3. ábra a 4-50 központi egység tömbvázlatát mutatja. Nézzük sorba a feltüntetett főbb egységeket.

2.1 Fő tároló

A 4-50 gépbe maximálisan 256 K byte méretű fő tároló építhető. A hagyományos felépítésű fermitár ciklusideje 1,44 μ s. Egyidejűleg mindig 2 byte kiolvasása történik meg. Ha nem kell mindkét byte-ba új információt írni, úgy az egyik byte regenerálódik. A tárban minden byte-ot páratlan paritású ellenőrző bit kísér.

A tárat 2 K byte méretű blokkonként védeni lehet. Minden blokkhoz egy-egy 4 bites kulcs tartozik, összesen tehát 16 kulcs létezik. Ebből kettő speciális, a fennmaradó 14 különböző programokhoz rendelhető. A program csak olyan tárrészhez fér hozzá, amelynek védelmi kulcsa saját kulcsával azonos. Ez a rendszer a multiprogramozás alapvető kelléke, mert biztosítja a programok és adataik kölcsönös hozzáférhetetlenségét.



3. ábra. A központi egység tömbvázlata

2.2 Belső adatmozgatás

A gép belső adatátvitelének rendszere az adatsín (data bus) elnevezésű gyűjtőszínrendszer. Ez az egész gépet behálózó vonalrendszer, amelyhez megfelelő vonalerősítő és vevő áramkörök segítségével minden regiszter bemenete és kimenete is csatlakozik. A 4-50 központi egységben a sínrendszer 16 információs bitet tud továbbítani párhuzamosan. 2 speciális vonal szolgál a főtárbeli cím két legértékesebb (17. és 18.) bitjének átvitelére.

A gépen belüli adatátvitel úgy történik, hogy a vezérléstől jövő jelek nyitják egy forrás-regiszter kimeneti és egy fogadó regiszter bemeneti kapurendszerét. Így az adóregiszter tartalma átmásolódik a fogadó regiszterbe. A sínrendszer egyszerre csak egy regiszter-pár számára hozzáférhető. A táruk információ-forgalma minden esetben adatregisztereiken keresztül történik. Ez a kiépítés alapvetően meghatározza a gép szervezését. A gépcsald nagyobb egységei gyorsabb működésüket jelentős mértékben a műveletek párhuzamosítását lehetővé tevő sínrendszerüknek köszönhetik.

2.3 Gyors memória

Van a gépben egy 128 szavas gyors memória is (Fast Memory, FM). Ez speciális felépítésű ferrit-tár

— egy bit tárolására két ferritmag szolgál — ciklus-ideje 480 ns. A tár szöszervezésű, minden szó 32 információs- és 2 paritásbitből áll.

Az FM ferritmemória, amelyet kiolvasás után regenerálni kell. Egy külön kapurendszer segítségével biztosítható, hogy nem csak az eredeti információ írható vissza a tárba, hanem a tartalmat 1 bittel léptetni vagy az értéket eggyel vagy kettővel módosítani lehet. Ez a kiolvasott adatot nem változtatja meg, de a következő kiolvasáskor már a módosított információt kapjuk.

Az FM szavait regiszternek nevezik. A programozó számára 16 regiszter áll rendelkezésre, hogy azokban tetszése szerinti információt tároljon.

Az FM regiszterei közül négy pár lebegőpontos műveletek operandusait tartalmazza. A regiszterek közel fele a központi egység vezérlésében vesz részt (megszakítási rendszer, csatornák vezérlése) vagy munkakereszként szolgál.

2.4 Aritmetikai egység

A 4—50 központi egységben az aritmetika egy byte méretű operandusokon párhuzamosan dolgozik. Az aritmetikai egység csak bináris és BCD összeadásra és kivonásra, valamint logikai műveletek végzésére alkalmas közvetlenül. A magasabbrendű műveleteket a gép elemi lépések sorozataként végzi.

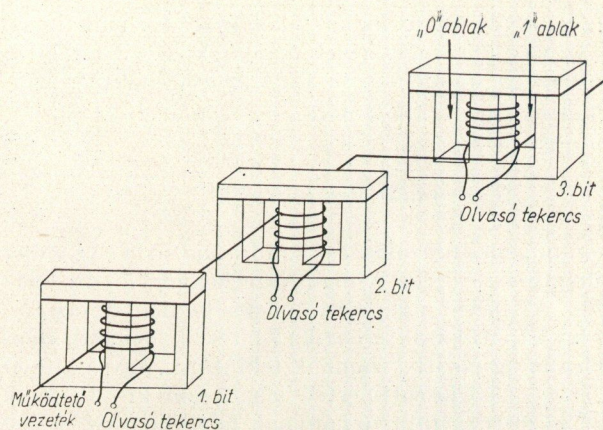
Az aritmetika kimenetéhez egy matrixnak nevezett kapurendszer csatlakozik, amely az eredményen módosításokat tud végrehajtani. Ez a matrix a 8 bites byte-ot két 4 bites részként tudja kezelni és segítségével egy kvartettbe speciális kódok írhatók be (például decimális műveletnél az előjel kódja) vagy a byte két fele felcserélhető stb.

2.5 Vezérlés

A gépcsaládnak 144 gépi utasítása van, ezek maguk is viszonylag bonyolult feladatokat végeznek el. Ilyen körülmények között a hagyományos, logikai kapukat alkalmazó vezérlési lánc túlságosan drága megoldás lenne és a tervezők mikroprogramozás mellett döntöttek.

Mikroprogramot alkalmazó vezérlő rendszerben minden gépi utasításnak (és néhány egyéb műveletnek is) egy-egy mikroprogram felel meg, amely a kívánt sorrendben tartalmazza az elvégzendő elemi gépi lépéseket. A mikroprogramokat rendszerint egy fixtár (Read Only Memory, ROM) tartalmazza. A fixtárban az információt a felépítés struktúrája hordozza, az olvasás biztonsága nagyobb, mint az operatív táráké.

A 4—50 központi egységben mágneses elven működő fixtár van (4. ábra). A memória szöszervezésű, egy szó 53 információs és 1 ellenőrző bitet tartalmaz. Minden bíthez egy-egy E—I alakú ferritelemekből összeállított zárt köpenytranszformátor tartozik. A középső oszlopon többmenetes olvasótekerces helyezkedik el. A bit információtartalmát a tekercesben indukált jel polaritása hordozza. A magot az egyik ablakon átfűzött működtető vezetékben folyó áram gerjeszti. Nyilvánvaló, hogy a meghajtó vezetéknek a másik ablakon történő átvezetése a kiolvasott jel polaritásának megváltozásához vezet. Megállapodás-



4. ábra

szerűen az egyik ablakot „0”, a másikat „1” értékűnek tekintik. Egy szó működtető vezetéke áthalad minden transzformátor egyik ablakán, a szó kívánt tartalmának megfelelően.

A kiolvasott mikroprogramot a vezérlő egység dekódolói a gép különböző részeihez továbbított vezérlő jelekké alakítják.

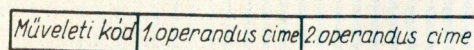
A mikroprogram minden szava egy 480 ns időtartamú elemi lépés (Elementary Operation, EO) tartamára meghatározza a gép állapotát. 15 bit (F—V—C mezők) meghatározza az elemi aritmetikai műveletet, 14 bit (M—S—D mezők) vezérli a gépen belüli adatátvitelt és az FM címzést, 18 bit (T—N—A mezők) megadja a mikroprogram következő szavának címét és 6 bit (E—I mezők) speciális vezérlő jeleket kelt.

A mikroprogram lényegében egy lefutó vezérlés. A címgeneráló logika csak a mikroprogram kezdő címét állítja elő, ezután egyik lépés hívja be a következőt. Minden lépésben elvégezhető a gép egy állapotjelzőjének vizsgálata és ennek alapján a program kétféle ágazhat. Bármely ágon ér is véget a mikroprogram, az utolsó lépésben keltett speciális vezérlő kóddal felhívja magára a címgeneráló logika figyelmét, amely meghatározza a következő mikroprogram kezdőcímét. Logikai kapuáramkörök tehát csak a mikroprogramok megindítását, összekapcsolását végzik.

3. Egy gépi utasítás végrehajtásának menete

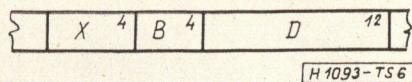
A System 4 gépcsalád kétcímű gépi utasításokkal dolgozik. Az utasítások általános formáját az 5. ábra mutatja. Az eredmény mindig az 1. operandus helyén keletkezik, a 2. operandus változatlan marad. Az operandusok lehetnek a főtárban, regiszterben vagy magában a programban; ennek megfelelően több utasítástípus létezik.

A főtárban levő operandust a program nem címzi meg közvetlenül. Az úgynevezett indexelhető cím

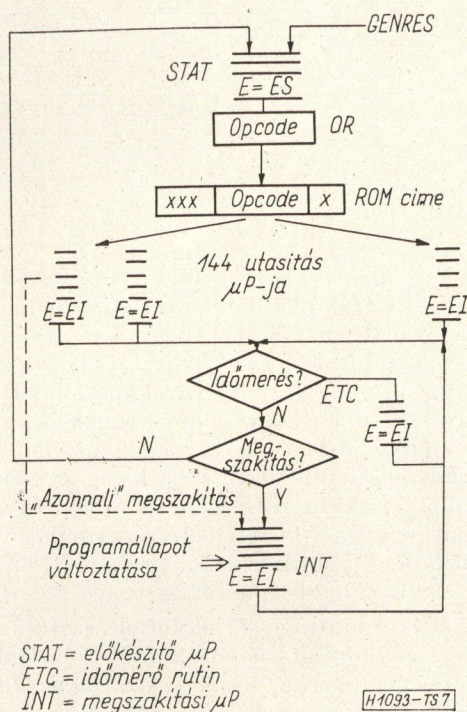


H 1093-TS 5

5. ábra



6. ábra



7. ábra. Egy utasítás végrehajtásának menete

megadásának módját a 6. ábra mutatja. X és B egy-egy regiszter címét megadó 4 bites kód, D egy 12 bites bináris szám, az úgynevezett relatív cím.

Abszolút tárcím = B mezőben megcímzett (bázis) regiszter tartalma + X mezőben megcímzett (index) regiszter tartalma + D mezőben levő szám.

A D mezőben megadott számmal közvetlenül 4096 byte címezhető. A programot ilyen méretű modulokból lehet felépíteni, a modulokat a bázisregiszter tartalmának megfelelő beállításával össze lehet kapcsolni. Ez a felépítés előnyt jelent programok írásánál, fordításánál és szerkesztésénél egyaránt.

Az indexregiszter segítségével igen egyszerűen lehet egy modulon belül ciklust szervezni anélkül, hogy az utasítások alakja megváltozna. A ciklus végén tehát nem kell helyreállítani a programot egy következő felhasználás számára.

A címzési rendszer hátránya, hogy még a gépi kódú utasítás nem hajtható végre közvetlenül, mert nem tartalmaz abszolút tárcímeket. Az utasítás végrehajtását megelőzi egy előkészítő gépi művelet (staticising), amelynek feladata, hogy kiszámítsa és tárolja az abszolút tárcímeket. A 7. ábra egy gépi utasítás végrehajtásának menetét mutatja.

Induláskor (General Reset után) vagy egy megelőző utasítás befejezésekor a gép rátér az előkészítő művelet mikroprogramjának végrehajtására. Ennek során kiolvassa a főtárból a programszámláló regiszter tartalma alapján a soronkövetkező utasítás műveleti kódját és tárolja azt egy hardware regiszter-

ben (OR). Ezután a mikroprogram elvégzi a műveleti kód által jelzett előkészítő műveleteket és tárolja a kiszámított címet vagy címeket. Az előkészítő művelet módosítja a programszámláló regiszter tartalmát, amely a soronkövetkező utasításra mutat.

Az előkészítés végén a mikroprogram $E=ES$ (End of staticising) vezérlő jelet kelt, amelynek hatására a vezérlés behívja az előkészített utasítás mikroprogramját. Minden gépi utasítás mikroprogramjának kezdőcíme a műveleti kód egy bittel balra léptetett értékével egyezik meg. Így egyszerűen az OR regisztert a ROM címregiszterébe másolva hívható a kívánt mikroprogram.

Az utasítás végrehajtásának utolsó lépésében $E=EI$ (End of instruction) vezérlő jel keletkezik. Ekkor a logika többféle soronkövetkező feladat között dönt. Van a gépben egy időmérő rutin (ETC), amely a hálózati feszültség minden periódusában egy mező tartamából 6-ot levon. Ha a mező tartalma negatívba fordul, megszakítás következik be. A rutin 20 ms-onként automatikusan beszűrődik az utasítások közé.

Ha nem kell módosítani az időmérő számot, a logika megvizsgálja, hogy nem váratkozik-e feldolgozandó megszakítási igény. Ha igen, rátér a megszakítások közös feldolgozó mikroprogramjára (lásd a 4. pontot).

Ha nem váratkozik megszakítási igény, a logika újból behívja az előkészítő mikroprogramot és ezután végrehajtja a soronkövetkező utasítást.

4. Megszakítási (interrupt) rendszer

A gép teljesítőképességének jó kihasználása érdekében egy úgynevezett vezérprogram (operating system) végzi a rendszer legmagasabb szintű vezérlését. Ez lehetővé teszi többek között a multiprogramozást is. Ha egy programnak váratkoznia kell — például adatokat olvastat be vagy írat ki, a gépkezelőnek mágnesszalagot kell cserélnie stb. — az így felszabaduló időt a központi egység egy másik program futtatására fordítja. A vezérprogram tehát igyekszik a központi egységet állandóan munkával ellátni.

Javítható hibák előfordulásakor a vezérprogram automatikusan behívja a korrigáló rutint. Ha száz-ezer műveletbe kerül egy hiba elhárítása, ez — 10 μs átlagos műveleti idő mellett — annyi, mintha a kezelő 1 másodperc alatt beavatkozott volna!

A vezérprogram működésének alapvető kelléke a megszakítási rendszer.

4.1 A gép programállapotai

A központi egység négy programállapotban működhet ($P1$, $P2$, $P3$, $P4$). A gyorsmemóriában minden állapothoz külön regiszterkészlet tartozik. Amikor a gép egy állapothoz tartozó programon dolgozik, a többi három állapot programja váratkozik.

Minden programállapot csak a saját regiszterkészletéhez fér hozzá. A $P3$ állapot annyiban játszik kitüntetett szerepet, hogy számos regisztere kettős jelentésű. Ezek egyrészt a $P3$ állapotbeli program operandusai, másrészt a $P1$ és $P2$ állapothoz tartozó állapotmeghatározó (például programszámláló, megszakítási maszk stb.) regiszterek. Így a $P3$ állapot-

beli program mintegy uralja a többi állapotot, míg azok nem tudják megváltoztatni saját állapotjelzőiket.

4.2 Programmegszakítás

Program feldolgozása során a gép *P1* állapotban van. Ha valamilyen ok miatt szükség van a vezérprogram segítségére, a gép elhagyja a *P1* állapotban végzett munkát és *P3* állapotba, a vezérprogram irányítása alá kerül. Ezt a folyamatot nevezik programmegszakításnak.

A megszakításnak sokféle oka lehet. Fellép, ha egy aritmetikai művelet végrehajtásakor hiba keletkezett, ha a programban címzési hiba van, ha a program meg akar indítani egy input-output műveletet, ha egy perifériális berendezés befejezett egy input-output műveletet, ha az operátor parancsot akar adni a rendszernek stb.

A megszakítások alapvető jellegük szerint két csoportra oszthatók: „end” és „immediate” típusúakra. A legtöbb megszakítás „end” (vége) típusú, ezek az éppen folyó gépi utasítás befejezésekor érvényesülnek (lásd a 7. ábrát). Ilyenkor a programot a megszakítás bekövetkezése csak annyiban érinti, hogy végrehajtását rövid időre felfüggesztették. Ez a helyzet például egy input-output művelet befejezésekor.

Katasztrofális hatású gépi vagy programozási hiba azonnali (immediate) megszakítást okoz. Ilyenkor a gép félbehagyva a folyamatban levő munkáját, azonnal rátér a megszakítás feldolgozására. Ilyen például a programban észlelt címhiba.

Minden programállapothoz tartozik egy-egy megszakítási maszk-regiszter. Ezek 32 bites regiszterek, minden bitjük egy-egy megszakítás maszkolására szolgál. Ha egy megszakítás maszkja 0, a megszakítás nem érvényesül, függőben marad.

Általában a *P1* és *P2* állapotban minden megszakítás megengedett, míg a *P3* állapotban csak a gépi hibák megszakításai érvényesülnek.

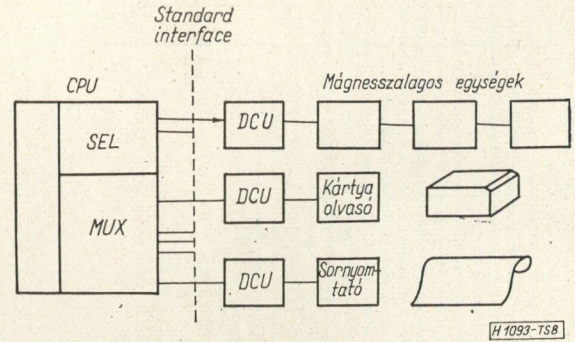
Megszakításkor a gép automatikusan áttér *P3* állapotba. *P4* állapotba csak paritáshiba vagy tápellátási hiba viszi a gépet. Ez kívül esik az üzemszerű működés határain. A vezérprogram *P3* állapotbeli kódolása elemzi a megszakítás okát és behívja a megfelelő feldolgozó rutinokat. Ezek közül egyesek *P2* állapotban futnak, amely állapot a vezérprogram „munkái” számára van fenntartva.

A megszakítás feldolgozása után a legmagasabb periódusú folytatható programon dolgozik tovább a gép *P1* állapotban a következő megszakítás bekövetkezéséig.

5. Input-output rendszer

A System 4–50 központi egység és a perifériális berendezések közötti kapcsolat vázlatát a 8. ábra mutatja. A központi egységben a csatornának nevezett elektronika vezérli az adatforgalmat.

A perifériák és a központ közé egy-egy illesztő egység (Device Control Unit, DCU) kapcsolódik, amely vagy a vezérelt periféria szekrényében, vagy önállóan helyezkedik el és típusától függően egy vagy több perifériát irányít.



8. ábra

A DCU biztosítja, hogy a tőle a központhoz futó kábel (trunk) a standard interface fizikai megvalósulása legyen. Ez azt jelenti, hogy a csatorna szabványos jelsorozatok formájában „tárgyal” bármelyik DCU-val, amely értelmezi a kapott információt és az egyes készülékek jellegének megfelelően reagál.

A standard interface nagy rugalmasságot biztosít a rendszer kiépítésében, mert az egész gépcsalád alapvető perifériái közősek, könnyű a bővítés és akár a központi egység is kicserélhető a perifériák változtatása nélkül. A rendszer hátránya viszont, hogy sok illesztő elektronika beépítését igényli.

A 4–50 központi egységbe kétféle csatorna építhető. A szelektorcsatornához 2 DCU csatlakozhat, egyikük megcímezése a másikat kizárja az adatforgalomból. A gépbe legfeljebb 3 szelektorcsatorna építhető. Ezek működhetnek párhuzamosan, ha az együttesen átvitt adatmennyiség nem haladja meg a 465 kbyte/s értéket. A szelektorcsatornához a háttérmemóriaként szolgáló mágneslemez és mágnesszalagos berendezéseket kapcsolják.

A multiplexor csatornához 9 DCU kapcsolható és ezek egyidejűleg megcímezhetők. A csatorna időosztásos elv szerint valamennyi perifériát ki tudja szolgáltatni adott prioritási sorrend szerint.

A multiplexor csatorna és a 3 szelektorcsatorna működhet párhuzamosan feltéve, hogy az együttesen átvitt adatmennyiség nem haladja meg a 465 kbyte/s értéket. Ezen előírás megsértése egyes perifériáknak az adatforgalomból való kizáródását vonja maga után. Teljes kiépítés esetén tehát egyidejűleg 12 periféria bonyolíthat le adatforgalmat a 4–50 központi egységgel: 3 a szelektorcsatornákon és 9 a multiplexor csatornán keresztül.

5.1 Egy input-output művelet végrehajtása

A rendszert úgy tervezték, hogy a perifériákkal bonyolódó adatátvitel csak rövid időre vonja el a központi egységet a program végrehajtásának munkájától.

Minden input-output műveletet a főtárban a programban elhelyezhető kettős szó, az úgynevezett csatorna-parancsszó (Channel Command Word, CCW) specifikál. Ez többek között tartalmazza a perifériának küldendő parancsot, az érintett főtárbeli mező kezdőcímét és hosszát. Egy CCW többször is felhasználható. A CCW összeállítása nem jelentkezik közvetlenül a programozói gyakorlatban, mert azt a programozó által definiált file-kezelési mód (például

soros feldolgozásnál DTFSR) és más makróutasítások információiból a fordítóprogram készíti el.

A felhasználásra kerülő CCW címét egy másik szó, a csatornacímzó (Channel Address Word CAW) adja meg. Egy input-output művelet megkezdése előtt ennek tartalmát be kell másolni a tár egy fix helyére, ahonnan a gép felhasználja.

Ezen előkészületek után a programnak ki kell adnia egy eszköz megindítási (Start Device, SDV) utasítást. Ez tartalmazza a megindítandó eszköz egyedi címét. A rendszerben 256 eszköz címzésére van lehetőség.

A Start Device utasítás végrehajtása az input-output művelet első fázisa, ez még a központi egység teljes figyelmét igényli. Az utasítás végrehajtása során a központi egység megvizsgálja a címzett eszköz állapotát és ha azt kielégítőnek találja, közli vele a CCW-ben levő parancsot. A CCW-ben és a CAW-ben levő információt pedig az utasítás elhelyezi az illető csatornához tartozó regiszterekben.

Ezután következik az input-output művelet automatikus része. A Start Device utasítás sikeres végrehajtása után a központi egység folytatja programfeldolgozó tevékenységét. Ha az input-output művelet megindító programnak várakoznia kell a művelet befejezésére, úgy a rendszer egy másik programon dolgozik tovább. Ez a multiprogramozás alapelve. Ha nincs a gépben folytatható program, a központi egység úgynevezett üres (Idle, tétlenségi) utasítást hajt végre.

A Start Device utasítás hatására a megcímzett perifériális berendezés megkezdí működését és szerkezetétől függő idő múlva lesz alkalmas egy byte átvitelére. (A „standard interface” mentén az adatforgalom byte-onként történik.) Például a kártyaolvasó behúzott egy kártyát és az első oszlop az olvasófej alá ért, a mágnesszalag elérte névleges sebességét stb.

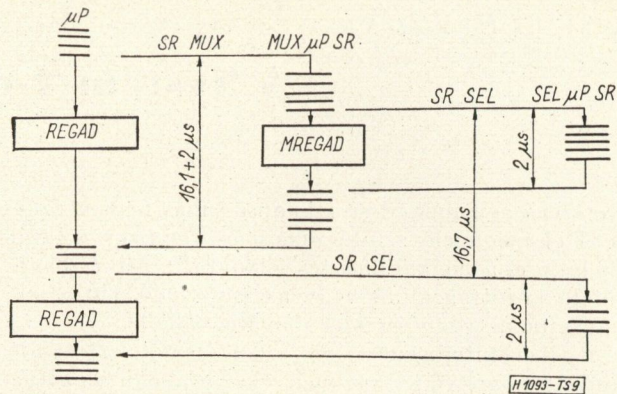
Amikor ez bekövetkezett, a perifériához tartozó DCU kiszolgálási igényt (service request, SR) küld a központi egységnek, jelet ad az interface kábel erre szolgáló vezetékére. Hangsúlyozni kell, hogy ez egyedül a periféria működésétől függ és a központi egység jeleihez képest teljesen aszinkron időzítésű esemény.

Vizsgáljuk meg a helyzetet egy konkrét példán. Tételezzük fel, hogy két berendezést indítottunk el a rendszerben, egy mágnesszalagos egységet a 2. szelektor csatornán és a kártyaolvasót a multiplexor csatornán. Tegyük fel továbbá, hogy a kártyaolvasó kért hamarabb kiszolgálást. (A mágnesszalagtól 6,4 μ s-onként, a kártyaolvasótól 525 μ s-onként jön kiszolgálási igény.)

Az input-output művelet során a főtárból olvas ki vagy oda ír be információt a gép. A 4—50 központi egység csak egyetlen úton tud a főtárhoz férni és így a periféria kiszolgálásának idejére a programutastáshoz tartozó mikroprogramot fel kell függeszteni.

Amikor megérkezik az SR jel, az esetleg éppen folyó főtárbeli ciklus végén a gép beírja a soronkövetkező mikroprogramlépés címét egy hardware regiszterbe (REGAD) és rátér a periféria kiszolgálására. Erre a célra is egy mikroprogram szolgál (MUX μ P SR a 9. ábrán).

Mivel a multiplexor csatornán a 9 trunk bármelyikéről érkezik SR jel, a mikroprogram először megkér-



9. ábra. A mikroprogramok együttműködése input-output műveletekben

dezi a kiszolgálandó eszköztől a címét, majd ennek alapján előveszi a főtár egy különleges részéből az eszközhöz tartozó csatornainformációkat és betölti a multiplexor csatorna regiszterkészletébe. Ezután sor kerül a byte átvitelére a standard interface mentén, majd a csatorna-információ visszakerül a főtár speciális részébe. Ez a kiszolgálási művelet viszonylag hosszú időt vesz igénybe. Példánkban feltételeztük, hogy a mágnesszalagos egység is működik. Kiszolgálását nem lehet elhalasztani a kártyaolvasó kiszolgálásának befejezéséig, mert a szalag elhalad a fej előtt és elvész az információ.

Ennek megakadályozása érdekében a szelektorcsatornán jövő kiszolgálási igény nemcsak egy utasítás mikroprogramját, hanem még a multiplexorra kapcsolt eszköz kiszolgálását is felfüggeszti. A szelektor kiszolgálása gyorsan végbemegy, mert a szükséges információk állandóan a csatorna regiszterkészletében vannak, lényegében csak egy főtárbeli ciklust kell lopni („cycle stealing”) a várakozásra kényszerített rutintól.

Ha mind a multiplexor-, mind a szelektorcsatornán érkezett igényt kiszolgáltatta a gép, akkor visszatér az eredeti mikroprogram megszakítási helyéhez és folytatja a munkát.

Az input-output művelet adatátviteli fázisa tehát megosztja a központi egység idejét az éppen futó program és az adatforgalom kiszolgálása között. Ennek vezérlését a mikroprogramok szintjén logikai kapuk végzik, mintegy tudat alatt történik.

Ha átvitelre került a megfelelő számú byte, a központi egység megvizsgálja, hogy a CCW kíván-e láncolást. Több parancsból úgynevezett lánc alakítható ki, és a művelet sor egyetlen Start Device hatására fut le. A láncolás alapvető jelentőségű mágnesszalagos tárolóknál.

A kívánt művelet vagy műveletek befejezése után a DCU program-megszakítást okozó jelet küld a központi egységnek. Hatására a vezérprogram megvizsgálja a hibajelző információkat, ellenőrzi az elvégzett művelet pontosságát. Ha mindent rendben talál, átadja a vezérlést a legmagasabb prioritású folytatható programnak.

E rövid ismertetés nem törekedhet teljességre még a fogalmak felsorolása terén sem, de ha sikerült néhány olvasó érdeklődését felkelteni egy szép és modern szervezésű rendszer iránt, elérte célját.

A System 4-es operációs rendszer

ETO: 681.32 System 4

Korszerű gép működése manapság már nem képzelhető el megfelelő szintű operációs rendszer nélkül. Ez látja el a logikai illesztés feladatát a gép és használója között azáltal, hogy az ismétlődő, rutin jellegű munkák döntő részét magára vállalja.

A gépek intelligenciájával szemben támasztott állandóan növekvő követelményeket a „behuzalozott” alaputasításokkal nem lehet kellőképpen követni, szükséges tehát egy operációs rendszer, amely az utasítások kombinációival (vagyis gépi kódban megírt programok segítségével) a szolgáltatások széles skáláját nyújtja a felhasználó felé.

Az operációs rendszer nemcsak kényelmi szempontokat szolgál. A gép nagy sebességét nem lehetne kihasználni az operációs rendszer vezérlő tevékenysége nélkül, hiszen ha figyelembe vesszük, hogy egy másodperces „gyors” operátori beavatkozás alatt a gép százezer műveletet is végezhet, akkor világossá válik, hogy minél több feladatot kell a gépre és minél kevesebbet az operátorra bízni.

Formailag az operációs rendszer nem sokban különbözik más, gépi kódban megírt (v. lefordított) felhasználói programrendszerétől. Lényeges különbség azonban, hogy a privilegiált utasításokon keresztül olyan processzor állapotokhoz és tárrészekhez is hozzáférhet, amelyek egyéb programok számára tiltottak, másrészt feladata éppen ezen felhasználói programok zavartalan futásának biztosítása.

A mai gépek operációs rendszere rendkívül széleskörű és szerteágazó feladatot lát el, s ezért terjedelmük sokszorosan meghaladhatja az operatív memória tároló kapacitását. A fenti ok miatt a legtöbb rendszert moduláris szerkezettel készítik. Ezek szervezésére jellemző, hogy egy állandóan jelenlevő rezidens rutin a szükségletnek megfelelően egy külső memóriából hívja be a megfelelő rendszerprogramot.

Ilyen esetben a rendszerprogramok (-rutinok) mágneselemezre vagy dobon helyezkednek el, és a gép működése során a rezidens rész vezérlő hatására az egyes szegmensek állandó jellegű vándorlása zajlik le a külső memória és az operációs rendszer számára fenntartott ún. overlay területek között.

1. A J szintű operációs rendszer

1.1 A rendszer jellemzői

A System 4-es gépcsald J operációs rendszere (JOP) az ún. „batch” típusú géptermi munkák irányítására készült. Helyileg két disc-en, a rendszer működéséhez és a kész programok futásához állandóan üzemben levő SYSTEM DISC-en, valamint a programok készítéséhez (fordítás, komponálás, tesztelés) szükséges TRIALS DISC-en helyezkedik el. Az operatív memória (ferrit) és a perifériák kapcsolatát az 1. ábra mutatja.

A teljesen üres gép beindítása a SYSTEM DISC felhelyezése után operátori beavatkozással hardware úton ún. öntöltő (self-loading) rutinok segítségével végezhető el. Ennek hatására egy SUPERVISOR-nak nevezett program generálódik a SYSTEM DISC-ről az alacsony című ferritmemória részekbe és ezután minden funkcionális feladatot (beleértve az input-output vezérlést is) ez a program lát el.

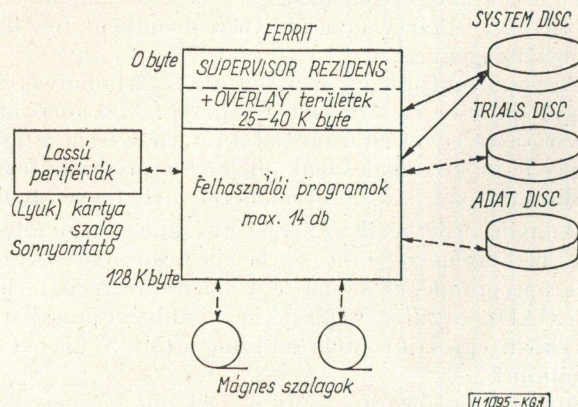
A gépet a jó kihasználás érdekében multiprogramozottra tervezték. Ennek lényege, hogy egyszerre több program helyezkedhet el a ferritmemóriában, s a központi egység a programokat prioritásuk sorrendjében futtatja. Az állapotváltás mindig megszakítással (interrupt) következhet be. Ha egy program perifériához nyúl, vagy egyéb okból futását nem folytathatja, arról a SUPERVISOR az Interrupt Status Register-en keresztül azonnal tudomást szerez, megvizsgálja a megszakítás okát, kezdeményezi a csatorna átvitelt (I/O esetén) és átadja a vezérlést egy alacsonyabb prioritású programnak. A perifériális átvitel befejezése szintén interrupt-ot okoz; ennek hatására a SUPERVISOR megvizsgálja az átvitel helyességét és visszaadja a vezérlést a legmagasabb prioritású programnak.

Elméletileg 14 program helyezkedhet el a gépben, ez a szám azonban a gyakorlatban — a kiépítéstől függően — 1–3 programra redukálódik.

A megszakított programok folytatásához szükséges összes információt a SUPERVISOR a SCRATCHPAD gyorsítótároló regisztereiben, a SYSTEM DISC-en, valamint a saját területén generált táblázatokban (Supervisor Tables) helyezi el. Az utóbbiak bizonyos részeihez a felhasználó is hozzáférhet.

A „batch” rendszerű üzemeltetés másik jellemző vonása, hogy a betöltésre váró programok a SYSTEM DISC egy megfelelő partíciójában sorbaállnak, és a rendszer a rangjuknak és perifériális, valamint memóriaszükségletüknek megfelelően választja ki a megüresedett helyre betöltendő programot.

A sorba állított programok 0-tól 9-ig terjedő rangot kaphatnak. A 9-es rangú program minden meg-



H 4095-K67

1. ábra. A memória és a perifériák kapcsolata

Beérkezett: 1971. IV. 15.

üresedett helyet és perifériát lefoglal egészen addig, amíg igényei kielégíthetővé nem válnak, s a program betöltésre nem kerül. A következő kategóriába tartoznak a 2—8-ig terjedő rangsorral rendelkezők. Üresedés esetén a rendszer csökkenő rangsorban végigvizsgálja a programokat és betölti azt a legmagasabb rangút, amelynek periféria és memóriaigénye kielégíthető. A vizsgálat során az átugrott programok rangja 1-gyel emelkedik. Az 1-es rangot kapott programok rangja nem változik, a 0-ás rangú pedig csak sorba áll, de betöltésére csak rangjának megváltoztatása után kerülhet sor.

A System 4 operációs rendszerében a programokat különböző nyelveken (ALGOL, FORTRAN, COBOL, USERCODE stb.) megírt függetlenül lefordítható modulok halmazából állíthatjuk össze. A modulokat szegmensekké, a szegmenseket — a helytakarékoság céljából — faagszerű struktúrává kialakított „overlay” területű programmá komponálhatjuk össze. Ez esetben az illetékes szegmensek behívásáról a program automatikusan (az operációs rendszer szervezéséhez hasonló módon) gondoskodik.

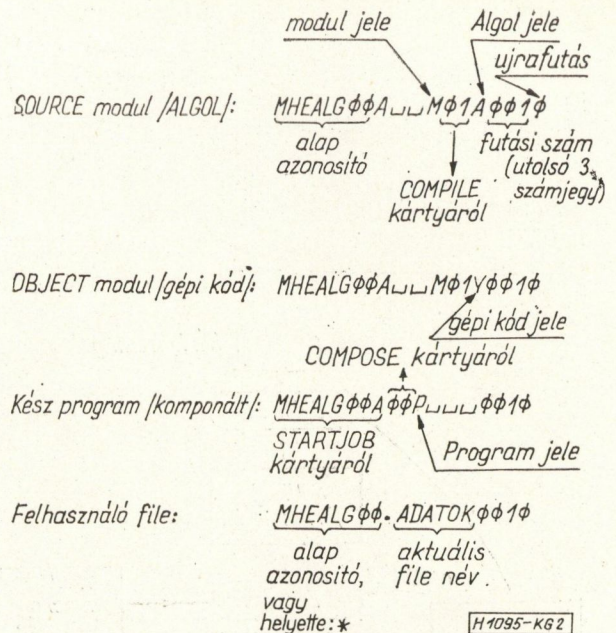
Nagyon fontos fejezete az operációs rendszer tevékenységének a file-okká szervezett adatok kezelése — a file-szervezés. Erről — a jelentőségének megfelelően — egy külön cikken tájékoztattuk az olvasót, itt csupán egy-két jellemző vonást szeretnénk kiemelni.

A file-szervezési módok felhasználásuk szerint két csoportba oszthatók: a rendszer által használt file kezelés (DTFPA) és a felhasználó által használt eljárások. Míg az előbbi fix hosszúságú (382 byte-os) blokkokat használ mind a rendszerprogramok, mind a felhasználói forrás (SOURCE) és tárgy (OBJECT) modulok, valamint a komponált programok kezelésére, addig a másik rész a variációk sokaságát nyújtja a programozó számára a legoptimálisabb szervezési eljárás kiválasztása érdekében. A leggyakrabban használt file azonosítókat a 2. ábrán mutatjuk be.

1.2 A rendszer felépítése

Az operációs rendszer funkcionális szempontból négy fő részre tagolható (l. 3. ábrát).

Az EXECUTIVE tartalmazza a már említett SUPERVISOR mellett a JOB CONTROL-t s így együttesen a programok betöltésétől kezdve a futtatásig minden tevékenységet ellátnak.

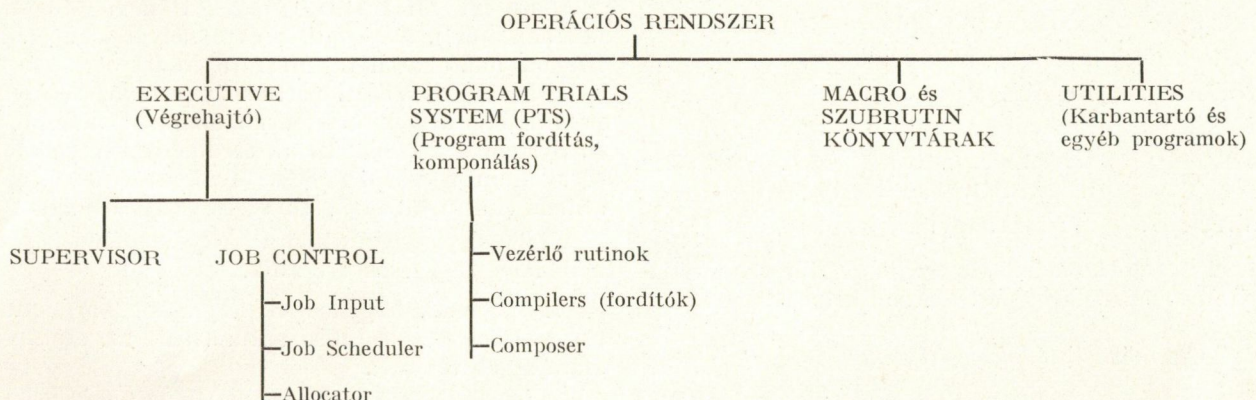


2. ábra. File azonosítók

A PROGRAM TRIALS SYSTEM (PTS) egy olyan programrendszer, amely minden más programhoz hasonlóan a SUPERVISOR felügyelete alatt fut. Magában foglalja a fordítóprogramokat (ALGOL, FORTRAN, COBOL, USERCODE), ezek vezérlő rutinjait, a forrásmodul betöltéséhez és javításához szükséges AMENDER-t, valamint a COMPOSER programot.

A PTS a programkészítés során használja a rendszer MACRO és SZUBRUTIN könyvtárát is. A MACRO utasítások pár utasításból álló előre elkészített programsomagot jelentenek, melyekre a programozó egy névvel hivatkozik. A fordítás során a gép ezeket a programrészeket beilleszti a helyére. A SZUBRUTINOK már nagyobb programrészeket tartalmaznak és a MACRO utasításokkal ellentétben — a behívások számától függetlenül — a gép csak egy példányban kapcsolja őket a programhoz. A programkészítés vázlatos menetét a 4. ábra mutatja.

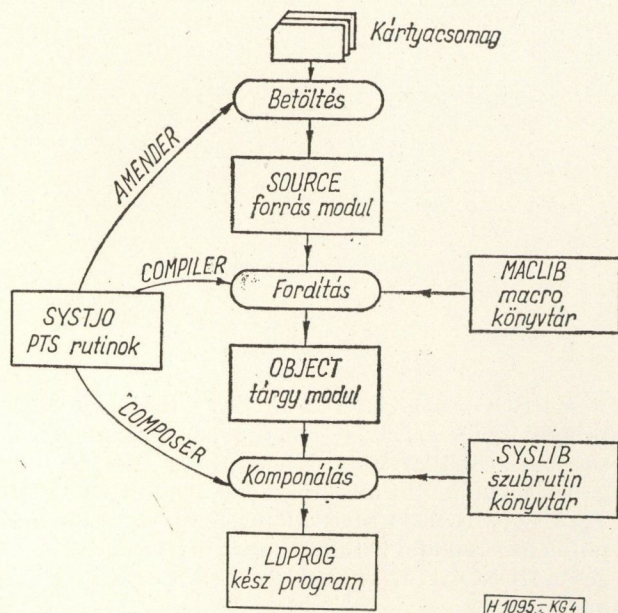
A UTILITIES címszó alatt elsősorban rendszert karbantartó programokat értünk. Különböző vizsgálóprogramok, a lemezek és mágnesszalagok használatához szükséges címkekepző, másoló rutinok,



3. ábra. Az operációs rendszer felosztása

valamint speciális programok tartoznak ebbe a kategóriába.

Szervezés szempontjából az operációs rendszer 13 alappartícióban foglal helyet a SYSTEM és TRIALS DISC-en. A partíciók felépítése olyan, hogy újabb track-ek dinamikusan kapcsolódhatnak hozzájuk azáltal, hogy minden track tartalmazza az öt megelőző és követő track relatív címét (cím láncolás). Ezzel elérhető, hogy a partíciók addig nem telnek be, amíg egy szabad track van a disc-en.



4. ábra. A rendszerpartíciók szerepe a programkészítésben

Az alappartíciók a következők:

- LDPROG: A kész felhasználói programokat és az Executive rendszert tartalmazza
- SYSJOB: Katalogizált programleírások (Job descriptions)
- SYSXEC: Betöltésre váró programok (Job queue)
- INWELL: Adatok számára fenntartott hely (Input well)
- OUTWEL: Üzenetek a futás alkalmával (Job journal)
- POLWEL: Pseudo-off-line sornyomtató pufferja
- TEMPRG: Próba alatt álló programok tára
- TSTPRG: Karbantartó programkönyvtár
- SOURCE: Forrásnyelvi modulok
- OBJECT: Lefordított modulok
- SYSLIB: Standard függvények
- SYSTJO: Fordítóprogramok (de a szervező rutin a LDPROG-ban van!)
- MACLIB: Macro könyvtár

Ezekben belül a modulok és programok file-ok formájában ún. másodlagos partíciókban helyezkednek el.

1.3 Vezérkártyák (Job description)

Az operációs rendszer által nyújtandó szolgáltatásokat a programozó a vezérkártyák vagy munka le-

író paraméterek segítségével veheti igénybe. Különböző karakterek különböztetik meg más (program v. adat) kártyáktól. Hatáskörük szerint két csoportra oszthatjuk őket. Az első csoportba a kész programok futtatásához szükséges vezérkártyák tartoznak, amelyek megjelölik a betöltendő program helyét a disc-en, a konfiguráció és memóriaigényt, valamint a programban használt file-ok szimbolikus neve és az aktuális file-nevek közötti kapcsolatot. (Ezáltal elérhető, hogy egy program átalakítás nélkül különböző névvel ellátott file-okat is feldolgozhat.)

A futtatási vezérkártyákat kényelmi okokból a disc-en is tárolhatjuk és futtatáskor csak a katalógusszámmra kell hivatkoznunk.

További lehetőségként itt adhatók meg a program bemeneti adatai is. A kártyák beolvasása után az adatok a disc-re kerülnek (INWELL partíció) és a program sorra kerülésekor innen jutnak be a memóriába.

A másik csoportba tartoznak a programösszeállítás során használt PTS vezérkártyák. Ezek hívják be a javítórutint (AMENDER), a fordító és komponáló rutinokat (COMPILER, COMPOSER), valamint közlik a programkészítéshez szükséges információkat.

Tekintettel arra, hogy a programkészítő rendszer (PTS) egy felhasználói programhoz hasonló felépítésű, futásakor azzal azonos elbírálás alá tartozik, és ezért futását egy szokásos munkaleíró kártyacsomaggal kell kezdeni. Azután következnek az előbb említett PTS kártyák, amelyeket már maga a PTS fog felhasználni.

Tovább bonyolítja a helyzetet, hogy egy PTS futtatása alkalmával az általa összeállítandó program futtatása is kezdeményezhető, s így egy TRIALS szekción belül (a PTS programkészítő rendszer futtatása) a kész programok futtatását kezdeményező és a programkészítő kártyacsomagok váltogathatják egymást.

A következőkben két példát mutatunk be, amelyek egy ALGOL program futtatását, ill. készítésével kombinált futtatását mutatják be. A hivatkozások megkönnyítése érdekében minden sor elejére egy számot írtunk, ami a valóságban természetesen nem létezik.

1. Példa

Adott egy MHEALGØØAØØP nevű program (12 karakter!), amely 10 Kbyte helyet igényel. A bemenő adatokat a futtatási vezérkártyákkal tesszük be az INWELL partícióba [erre az ALGOL programban a 60-as referenciaszámmal hivatkozunk (l. System 4 ALGOL-t), az eredményeket pedig a sornyomtatón várjuk (30-as referenciaszám)]. Készítsük el a futtatáshoz szükséges vezérkártyákat!

A teljesség igénye nélkül a kártyák szerepéhez az alábbi megjegyzéseket fűzzük.

1. Megjelöli, hogy melyik programot akarjuk futtatni és mi lesz a futási szám (csak az utolsó 3 számjegyet tekint).

2. Közli, hogy a program a LDPROG partícióban található.

1. `11 JOB MHEALG  $\phi\phi A\phi\phi P, \phi\phi\phi\phi 1$`
futásszám
2. `11 PRGLIB LDPROG`
3. `11 STORE  $\phi\phi 2\phi$`
4. `11 CONFG LP=1`
5. `11 RSP 1, A,  $\phi 8$`
6. `11 SDFILE LPI, *.SORNYO (A3 $\phi$ )`
7. `11 DATA A6 $\phi$`
8. `9.5, 3, 43.2, 15, 1, 5,`
9. `1*`
10. `11 EXEC`

H1095-KG 5

5. ábra. ALGOL program futtatása

1. `11 JOB TRIALS,  $\phi\phi\phi\phi 1$`
2. `11 PPOGLIB LDPROG`
3. `11 STORE  $\phi 12\phi$`
4. `11 CONFG D=1, LP=1`
5. `11 RSP 1, A,  $\phi 8$`
6. `11 RAFILE VOL  $\phi\phi 1 (C1\phi), *.PHRCWK$`
7. `11 SDFILE LP1, *.PHPCPR`
8. `11 EXEC`
9. `11 ALLOC SOURCE  $\phi\phi 1$`
10. `11 ALLOC OBJECT  $\phi\phi 1$`
11. `11 STARTJOB MHEALG  $\phi\phi A$`
12. `11 AMEND  $\phi 1$ , INSERT, U, A`

 ALGOL
PROGRAM
CSOMAG

13. `1*`
14. `11 COMPILE  $\phi 1$ , S, MODULE1`
15. `11 COMOSE  $\phi\phi$ , NOCTL, SAVEX`
`** INCLUDE  $\phi$ , * ( $\phi 1$ )`
16. `11 JOB MHEALG  $\phi\phi A\phi\phi P, \phi\phi\phi\phi 1$`

ld. 1. példát

- `11 EXEC`
17. `11 ENDJOB`
18. `11 ENDTRIALS`
19. `11 ENDIPT`

H1095-KG 6

6. ábra. ALGOL program készítése és futtatása

3. Megadja, hogy a program helyfoglalása 10 Kbyte ($1\frac{1}{2}$ K-nként számlál).

4. A program 1 sornyomatot fog használni. (Az INWELL partíció a SYSTEM DISC-en van, ezért ezt nem kell feltüntetni.)

5. 1-es rang, A stream (jelen esetben nincs szerepe) és 8-as prioritás.

6. Összerendeli az MHEALG $\phi\phi$.SORNYO aktuális sornyomatot file nevet a programban használt szimbolikus file névvel (30-as referenciaszám). Figyeljük meg, hogy a * (csillag) a programazonosító első 8 karaktere helyett áll.

7. Ez a kártya jelzi, hogy a disc INWELL partíciójába rakjuk be a soronkövetkező adatokat.

8. A kártya az adatokat tartalmazza.

9. Az INWELL partícióban elhelyezett adatfile végét jelzi.

10. A programleíró vezérkártyák végét jelzi.

2. Példa

A most következő példánkban feltételezzük, hogy az előző, egyetlen modulból álló program még csak forrás nyelven, kártyákra lyukasztott állapotban szerepel, és a feladat a program fordítása, komponálása és futtatása.

A teljes vezérkártya készletet a 6. ábra mutatja.

Az egyes kártyák szerepe a következő:

1. Megjelöli, hogy a futtatandó program a PTS lesz (a TRIALS szó a program azonosítója helyett áll).

2–8. A PTS program futtatási vezérkártyái.

9–10. Közli a géppel, hogy szükség lesz a $\phi\phi 1$ -es disc-en található két partícióra.

11. Programösszeállítás következik, melynek első 9 karaktere ott látható.

12. Egy $\phi 1$ -es ALGOL modult helyezünk el a SOURCE partícióba.

13. A programnak vége.

14. Fordítsd le a $\phi 1$ -es modult.

15. Komponáld programmá a lefordított modult. A kész program a LDPROG partícióba kerül. A komponált program verzió száma $\phi\phi$ (10–11. programazonosító karakter, a 12. P betű automatikusan hozzáíródik).

16. Futtasd a megadott (jelen esetben éppen az előbb komponált) programot.

17. A programösszeállítás befejeződött.

18. A PTS vége.

19. Az operátor által OI F-el behívott JOB INPUT program számára jelzi a bevitel végét.

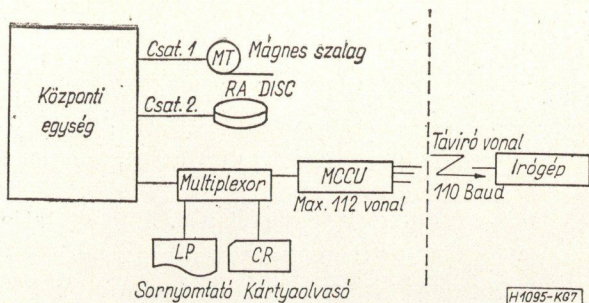
2. A MULTIJOB (MJ) szintű operációs rendszer

A MULTIJOB operációs rendszer a J szintű továbbfejlesztett változata. Kihelyezett pultok telepítését teszi lehetővé és ezeken keresztül biztosítani tudja mindazon szolgáltatásokat a távoli végállomásról, amelyeket a gépteremben lehetne igénybe venni.

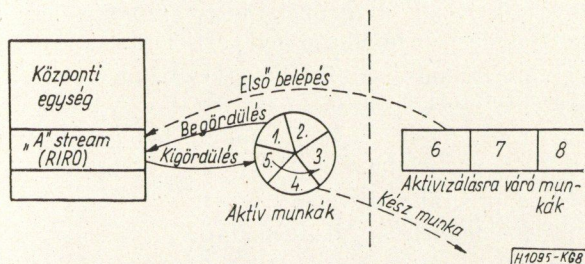
Az ICL cég a System 4-es gépcsáladhoz a kihelyezett pultok és végállomások egész rendszerét fejlesztette ki. Ezek a kiépítéstől függően a 110 Baud-os távíróvonalról kezdve a 4800 Baud-os nagy sebességű adatátviteli vonalig különböző típusú csatornákat igényelnek. A legbonyolultabb kiépítésben a végállomáson sornyomatot, kártyaolvasó, rajzgép vagy grafikus display működtethető, míg a legegyszerűbb esetben csupán egy távíró, vagy közvetlen vonalra kötött írógép jelenti a végállomást.

Kihelyezett pultokat tulajdonképpen a J szintű operációs rendszerhez is kapcsolhatunk. Lényeges különbség azonban, hogy ilyenkor a végállomásról a központba, illetve onnan vissza csak adatokat lehet továbbítani. Minden egyéb munkát (programbehívást, javítást stb.) a gépteremből kell kezdeményezni.

A kihelyezett pultok telepítéséhez egy különálló hardware egységre, a Multi-channel Communications Control Unit (MCCU)-ra van szükség a központban. Lényegében ez egy 3 Kbyte memóriával rendelkező kis célszámológép, amely a hozzá csatlakozó pufferekkel együtt a csatlakoztatott pultok



7. ábra. Kihelyezett pultos kiépítés



8. ábra. Időosztásos munkák lebonyolítása

adminisztrálását végzi. Az MCCU a multiplexor csatornára csatlakozik.

A legegyszerűbb kiépítésű konfigurációt a 7. ábra mutatja.

2.1 A RIRO (Roll-in Roll-out) stream koncepció

Az egyik legjelentősebb változás a batch rendszerű feldolgozáshoz képest az új RIRO stream koncepció.

Stream-ek a J rendszerben is léteznek és azt jelentik, hogy a gép memóriáját különálló, független egységekre oszthatjuk (A–F-ig jelölve), amelyekhez külön-külön állíthatunk sorba betöltésre váró programokat. (A multiprogramozás a stream-ektől függetlenül a gépben levő programok között változatlanul fennáll.) Ezáltal a nagy gép szervezés szempontjából esetleg előnyösebben kihasználható kisebb egységekre bontható.

Az MJ szinten ezeket a batch stream-eket egy újabb típussal egészítik ki: ez a RIRO (Roll-in Roll-out) stream. A gépbe kerülő programok most már nem a befejezésükig maradnak a memóriában, hanem egy megadott idő után „kigördülnek” a disc-re és átadják helyüket a soronkövetkezőknek. Ez a helycsere akkor is létrejön, ha a futó program perifériához nyúl, vagy valamilyen oknál fogva a futását nem folytathatja. Ily módon elérhető, hogy a gép több pultot „mintavételiesen” tud kiszolgálni és a távoli végállomás kezelője alig veszi észre, hogy nem ő az egyetlen felhasználó.

A RIRO stream működését a 8. ábra szemlélteti.

A stream-ek jellemzőit (RIRO vagy batch rendszer, memória és periféria elosztás, a RIRO streamben az egy programra jutó idő szelet stb.) az igényeknek megfelelően mindig a gépteremből a rendszer generálásakor kell kialakítani. A jó RIRO stream kialakításánál a programcsere 80–90%-a az input/output utasítás hatására jön létre, azaz mikor a program amúgy is várakozásra kényszerülne.

2.2 A File-ok kezelése és archivációs rendszere

Kihelyezett pultról minden feladatot file formájában lehet megadni. Egy file tartalmazhat összefüggő adatrendszert, de egy programrész vagy modul, valamint egy vezérkártyából összeállított csomag is kezelhető file azonosító alatt.

A rendszerben kétféle file-típust különböztetünk meg: a dedikált és a nem-dedikált file-okat. Az előbbiek teljesen hasonlóak a J szinten levőkhöz és csak program által vagy gépteremből hozhatók létre. A második típust a felhasználó pultról generálhatja, ezért rövidebb életű és terjedelmű disc file-okat értünk alattuk. Ezek a nem-dedikált file-ok közös disc-en helyezkednek el teljesen rendszertelenül. A nyilvántartásokról és a hely kiosztásról külön rendszerprogramok intézkednek. Ha ez a disc betelik, az újabb file-ok helyigényéről egy automatikus archivációs rendszer gondoskodik oly módon, hogy a legrégiben használt file-okat nyilvántartásba veszi és mágnesszalagra másolja, ugyanakkor gondoskodik arról, hogy igény esetén azok újra hozzáférhetőkké váljanak.

Az operációs rendszer gondoskodik a file-ok védelméről is. A felhasználó, miután létrehozta a saját file-ját, megfelelő utasítás segítségével megtilthatja, ill. megengedheti más felhasználók számára a hozzáférést. Helytelen utasítások kiadására a rendszer figyelmeztető üzenettel reagál.

2.3 A kihelyezett pultok kezelése

A pult használatára a felajánlós rendszer jellemző. Az utasítások és adatok begépelését mindig a gépnek kell kezdeményezni egy ún. „prompt” jellel. Utasítások várása esetén ez egy ?// (kérdőjel–vonás–vonás–szóköz)-ből álló jelsorozat, amelyet a felhasználó az utasítás leírásával folytathat. Az utasításokhoz tartozó adatok átvitelét a gép egy ? (kérdőjel) megadásával kezdeményezi. Minden sor végére a kezelőnek egy ETX (End of TeXt) jelet kell gépelnie, amely a CNTL és C billentyű egyidejű leütéséből áll. Kényelmes lehetőség kínálkozik a leírt szöveg javítására és törlésére, továbbá tetszőleges sorok közbeszúrására.

A munka elkezdésekor a LINE és START kapcsolót kell lenyomni, s a motor üzemi fordulatanak elérése után bármely karakter leütésére a gép a következő szöveget írja: ?// LOGIN. A felhasználónak ezután saját 6 karakteres nevét kell begépelnie. A gép ellenőrzi, hogy van-e ilyen regisztrált név és ha van, akkor a jelszókra kérdez a PASSWORD?□ kifejezéssel. A jelszó az illetéktelen névhasználat ellen biztosít és beírásakor biztonság kedvéért nem jelenik meg a papíron. A begépelte jelek nem közvetlenül, hanem „visszhang” formájában a gépből visszaküldve kerülnek kinyomtatásra, ezért a gép bizonyos jelek kinyomtatását le tudja tiltani.

Helyes jelszó után a gép kinyomtatja az időt, a felhasználható tárhelykapacitást és elkezdődhet a munka.

Minden munka befejezését a ?//□ LOGOUT utasítás zár le. Ezt követően a papíron megjelenik a dátum, valamint az elhasznált és a maradék gépidő.

A pultról végezhető tevékenységeket durván három részre oszthatjuk:

- a) file-ok létrehozása és javítása,
- b) előkészített munkák futtatása,
- c) programkészítés.

A felhasználó a rendelkezésre álló számos variációs lehetőség közül válogathat. Nem sürgős munkákat „batch” rendszerben futtathat le, másokat RIRO stream-ekbe küldhet. A futtatás lehet interaktív és közbeavatkozás nélküli. A programfordítás történhet egyszerre vagy lépésenként. Végül — különösen műszaki-tudományos programoknál — nagyon hasznos lehetőség a GO utasítás, amely egyetlen modulból álló programok automatikus fordítását, komponálását és futtatását végzi el minden külső beavatkozás nélkül.

3. Összefoglaló vélemény

A System 4-es gépcsalád egy hatásos és rendkívül rugalmas operációs rendszerrel rendelkezik, amely főbb vonásaiban az IBM 360-as rendszerével azonos.

Könyvismertetés

K. Géher: *Theory of Network Tolerances* (A hálózati toleranciák elmélete), Akadémiai Kiadó, Budapest 1971. 184 oldal, angol nyelven.

Dr. Géher Károly sok éves kutatómunkájának eredményeit foglalja össze ebben a könyvben. Számos nemzetközi konferencián tartott előadást e témakörbe tartozó részeredményekről, valamint több cikket is publikált, de most első ízben kerül egész toleranciakutatásra vonatkozó anyaga összefoglalva az olvasó kezébe. Ez nem csak egyszerű összefoglalása eddigi publikációinak, mert azonkívül, hogy a teljes anyag az ismert részeket is új megvilágításba helyezi, számos eddig nem publikált kutatási eredményt és gondolatot is tartalmaz. A könyv maga nemében páratlanul gazdag anyagot ölel fel. Ezt legjobban a 8 fejezetre tagolt anyag tartalomjegyzék-szerű ismertetésével tudjuk érzékeltetni. Az első fejezet a toleranciaszámítások érvényességi területével, az áramkör-tervezés és az érzékenység kapcsolatával, valamint a tudományág fejlődéstörténetével foglalkozik. A 2. fejezetben a hálózati paraméterek, a bilineáris és bikvadrátikus tételek, az első és másodfokú gyökhelyek kérdései kerülnek tárgyalásra. A 3. fejezet az érzékenységeket tárgyalja. Az érzékenységi invariánsok tárgyalását az érzékenység közvetett meghatározásának módszerei követi. A tárgyalási módszerek: az érzékenység meghatározása a bilineáris tételből, a transzfer függvényből, az állapotváltozós analízis segítségével, a szórás mátrixból és a jelút gráfjából. Ezután az érzékenységi függvény realizálhatósági feltételeit tárgyalja. A 4. fejezet a tolerancia analízissel és az optimalizálással foglalkozik. Tárgyalja a legrosszabb esetre történő tervezést, a statisztikai tervezést, a vegyes

A rugalmasságra jellemző, hogy az üzemeltető a rendszer generálásakor döntheti el milyen rezidens és overlay területeket nyit meg a ferritmémóriában a SUPERVISOR számára, továbbá bizonyos szabadsága van az alappartíciók elhelyezését illetően is.

A rendelkezésre álló szolgáltatások nem kielégítő volta esetén bármely partícióba újabb modulok építhetők be. Ezt a nagyfokú rugalmasságot viszont bizonyos többlet vezérkártya megadással kell „megfizetni”.

A J szintű rendszer főleg géptermi munkák, az MJ szintű kihelyezett pultok üzemeltetésére alkalmas. Kihelyezett pultok J szintű csatlakozása esetén a rendszerbe egy külön CCP-nek (Communication Control Package) nevezett programcsomagot kell beilleszteni, amely a P2 állapotban működik és a pultról jövő és menő üzenetek ügyeit intézi. Ilyenkor a pultok funkciója csak adat be- és kiviteltre korlátozódik. MJ szinthez kapcsolva a pultokat egy time-sharing rendszerű szolgáltatás építhető ki.

tervezést és a Monte Carlo analízist. Az optimalizálási eljárásokkal foglalkozó alfejezetek: minimál érzékenységi hálózatok, folytonosan ekvivalens hálózatok, minimális költségű hálózatok. Az 5. fejezet a lineáris passzív hálózatok címet viseli. Foglalkozik az érzékenységek és a toleranciák meghatározásával a frekvenciatartományban, a komplex frekvenciatartományban és az időtartományban. Részletesen foglalkozik a szerző a villamos szűrők tolerancia érzékenységeivel, a létrahálózatok érzékenységeivel, a szűrőtoleranciák számításával és az optimális áramköri elem tűrésekkel. A 6. fejezet a lineáris aktív hálózatok analízise és szintézise kapcsán felmerülő toleranciakérdéseket, így az aktív RC szintézist és a gyök érzékenység optimalizálását vizsgálja. A 7. fejezet a logikai áramkörök statikus és dinamikus toleranciáival foglalkozik. A 8. fejezetben tömör összefoglaló anyag található az iteratív szintézis kérdéseiről, így a lineáris programozás optimalizálási feladatairól, tervezési problémákról és a tervezési eljárások fejlődéséről.

A könyvet 123 címet felsoroló irodalomjegyzék zárja, mely az eddigi hazai irodalom teljes összefoglalását is tartalmazza.

Dr. Géher Károly munkája már a megjelenéstől eltelt rövid idő alatt is méltán keltett külföldön is nagy érdeklődést és hozott számos elismerést a szerzőnek. Úgy véljük a hazai szakemberek is tankönyvként és kézikönyvként egyaránt hasznosan forgathatják e művet.

Az Akadémiai Kiadó világszínvonalon álló, példás kiállításban adta ki e munkát és ez egyike a legszebb szakkönyveinknek.

B. Gy.

PÁLYÁZATI FELHÍVÁS

A KANDÓ KÁLMÁN VILLAMOSIPARI MŰSZAKI FŐISKOLA (Bp. VIII. Tavaszmező u. 17.) pályázatot hirdet a VEZETÉKNÉLKÜLI HÍRADÁSTECHNIKAI TANSZÉKEN betöltendő főiskolai tanári állásra.

A kinevezendő főiskolai tanár feladata: a mikrohullámú technika témakörébe tartozó tárgyak oktatása, az ezen témakörbe tartozó jegyzetek írása és a laboratóriumi mérések továbbfejlesztése.

Pályázati feltételek: műszaki vagy tudományegyetemi végzettség, legalább 10 éves felsőoktatási és ipari gyakorlat, legalább egy világnyelv ismerete.

Tudományos fokozattal rendelkezők előnyben részesülnek.

A pályázatot a közzétételtől számított 4 héten belül kell benyújtani a Főiskola főigazgatójához.

(Budapest VIII., Tavaszmező u. 17.)

A System 4 típusú számítógépek adatkezelési rendszere

ETO: 681.327 System 4

A számítógépekkel kapcsolatban mindig a műveletvégzés hallatlanul nagy sebességét szokták hangsúlyozni. Ez azonban csak az egyik oldal. Amit legalább ilyen joggal hangsúlyozni kell az az, hogy egy számítógépes rendszer a műveleteket nagytömegű adattal képes elvégezni, és hogy ez a két oldal szerves egészet alkot.

1. Bevezetés

Amikor nagytömegű adatról beszélünk, akkor nem tehetünk különbséget gazdasági, műszaki vagy tudományos alkalmazás között pusztán aszerint, hogy a feladat adatfeldolgozás jellegű-e vagy sem. A jelenségek egyre sokoldalúbb vizsgálata minden területen együtt jár az információigény növekedésével. Az információtárolás és -visszakeresés problémája pedig gyors ütemben vezetett új tárolóeszközök, illetőleg adat-szervezési és programozási módszerek kifejlesztéséhez.

1.1 Adatmozgás, adatátvitel

A számítógépek műveleti utasításai köztudomásúan túllépik a szokásos aritmetikai műveletek körét. Az operandusok is többfélék lehetnek. Numerikus és logikai operandusok mellett alfanumerikus operandusokkal is dolgozunk. Ez utóbbiak jelsorozatok (bit-sorozat, karakter-sorozat), amelyekkel kapcsolatban olyan műveletek fordulhatnak elő, mint például egy karaktersorozatnak (vagy egy részének) átvitele az operatív memória egyik területéről egy másikra, vagy egy karaktersorozat egyes részeinek kiemelése, két sorozat egymás mellé helyezése, összehasonlítása, rendezése stb.

Ezek az operandusok még csak elemi adatok. Nagytömegű adat kezelésénél azonban ritkán maradunk az elemi adatok szintjén. A feldolgozás szempontjából valamilyen módon összetartozó (általában egy egyed leírására szolgáló) numerikus és alfanumerikus adatok logikai adategységet alkotnak. Ezt az adategységet nevezzük **rekordnak**, a teljes adathalmazt alkotó rekordok összességét pedig **file**-nak.

Rekordot alkotnak egy gyártmány komponenseinek leírásai, egy vállalat nyilvántartásában a dolgozók személyi adatai, egy készletnyilvántartásban az egyes készletelemek adatai: megnevezés, méret, minőség, egyéb paraméteradatok, mennyiség, tárolóhely stb. Ezek után pedig beszélhetünk készletrekordokról, készletfile-okról stb.

A file-ok általában nagytömegű, akár több millió adatból álló halmazt jelentenek. Részben a kívüllég és a számológép közötti közlekedést szolgálják, részben pedig az állandó jellegű adatok tárolására szol-

gálnak. Az állandó jellegű adatokat tartalmazó file-okat szokták törzsfile-nak (master file) nevezni. A tárolóközeg típusa szerint beszélünk kártya-file-ról, papírszalag-file-ról, sornyomtató-file-ról, valamint mágnesszalag- és mágneslemez-file-ról. Ennek a megfogalmazásnak az alapján azután minden művelet, amit hagyományosan beolvasásnak és kiírásnak szoktunk mondani, file-kezelésnek tekinthető.

Az olvasóberendezéseken keresztülmenő adatok input-file-t, a lyukasztó- és kiíróberendezéseken keresztülmenő adatok pedig output-file-t alkotnak. Mint-hogy mágnesszalagra és mágneslemezre lehet írni, de olvasni is lehet róluk, az adott művelettől függően beszélhetünk input vagy output file-ról. Igaz, a mágneslemeznél ez nem mindig egyértelmű, mert annál egyazon menetben lehet beolvasni és kiírni is.

Az adatátviteli műveleteknek ez az általánosítása lehetővé teszi, hogy a perifériától függetlenül mindig ugyanazok a rutinok végezzék az adatkezelést. Az adatkezelés szempontjából a rekordok oszthatatlan egységeket jelentenek. Programunkban nem elemi adatok átvitelére adunk utasításokat, hanem a file-okkal, illetőleg rekordjaikkal dolgozunk. A file rekordjai között csak terjedelmük alapján teszünk különbséget: lehetnek fix és változó hosszúságúak.

A fizikai adatátvitel egysége a blokk. A rekordokhoz hasonlóan minden adatblokk lehet fix- vagy változó hosszúságú a benne tárolt rekordoktól függően. Előfordulhat, hogy egy blokkban csak egy rekord van, vagyis a blokk és a rekord egybeesik. Ekkor beszélünk blokkolatlan rekordokról. A perifériális berendezések egy részében a tárolóközeg természetéből adódóan csak blokkolatlan rekordokkal találkozunk. Egy lyukkártya vagy a sornyomtató egy sora egyszerre egy blokkot, illetőleg egy rekordot is alkot.

Mágnesszalagos és mágneslemez berendezésekben az átvitel és a tárolás gazdaságossága céljából egy blokk általában több rekordot tartalmaz. A blokk és rekord viszonya, illetőleg a rekordok hossza szerint négy esetet különböztethetünk meg:

- A rekordok blokkolatlanok és fix hosszúságúak; a blokk és a rekord egybeesik.
- A rekordok blokkolatlanok és változó hosszúságúak; egy blokk = egy rekord, a blokk is változó hosszúságú.
- A rekordok blokkoltak és fix hosszúságúak; minden blokkban meghatározott számú rekord fér el, a blokk is fix hosszúságú.
- A rekordok blokkoltak és változó hosszúságúak; minden blokkban több rekord is lehet és csak a maximális blokkhosszt írjuk elő.

A különböző esetek egy file-on belül nem keverhetők.

1.2 Soros file-ok

A fenti felsorolással lényegében kimerítettük a leg-egyszerűbb szervezésű file-ok formátumát. Kiegészítésül még el kell mondani, hogy minden file elején és végén vannak bizonyos szabványos és bizonyos szabadon megadható információk, amelyeket címkéknek nevezünk. A címkék a file azonosítására, kezdetének és végének jelölésére szolgálnak, feladatuk továbbá a file védelmének biztosítása.

A címkék között helyezkednek el a meghatározott típusú blokkok. A blokk típusa a lassú perifériális berendezéseknél meghatározott: fix hosszúságú és egy blokk csak egy rekordot tartalmaz. Mágnesszalagnál és mágneslemeznél a blokk típusát szabadon választhatjuk. A file-on belül a blokkokat (rekordokat) semmi sem különbözteti meg egymástól, legfeljebb a file-ban elfoglalt sorrendjük. A mágneslemezes file-ok kivételével, ahol más rekordstruktúrájú file-ok is tárolhatók, valamennyi file blokkjaihoz csak fizikai tárolásuk sorrendjében férhetünk hozzá. Ezt nevezzük soros file-szervezésnek, az ilyen szervezésű file-okat pedig egyszerűen soros file-oknak.

2. Input-output műveletek

Amint az előzőekben ismertetett felfogásból kiderül, elemi adatokra csak az operatív memórián belül hivatkozhatunk. Az operatív memória és a külvilág, illetőleg a háttérmemóriák és az operatív memória közötti átvitelnél csak a file és rekordjai (blokkjai) játszanak szerepet. Soros file-oknál még a file rekordjaira sem hivatkozhatunk egyedileg (például névvel).

Mínthogy ez a felfogás némileg eltér a kisebb gépeknél megszokottól, ahol például a **READ A** vagy **PRINT A** utasítás egy elemi adat átvitelét eredményezte, röviden vázoljuk egy elemi adat elérésének útját.

Az eljárást input file-ra mondjuk el, de ugyanez érvényes output file-ra is. Egy elemi adat eléréséhez a következő lépések szükségesek:

a) Hozzáférés a file-hoz

Első lépés a file megnyitása (OPEN). Ennél a műveletnél a file-kezelő software lefoglalja és a programhoz hozzáférhetővé teszi a kérdéses perifériális berendezést. A file elején levő címkék alapján ellenőrzi, hogy valóban a kívánt file-t tartalmazza-e a berendezés (a megfelelő kártyacsomag vagy szalagtekercs), és a program jogosult-e a file tartalmához hozzáférni stb.

b) Hozzáférés a file blokkjaihoz

Második lépés a tényleges adatátvitel elindítása. A beolvasott blokkok fogadására ki kell jelölnünk az operatív memóriában egy puffert. Ehhez kell ismernünk a file blokkjainak méretét, illetőleg változó hosszúságú, blokkolt rekordok esetén a megengedett maximális blokkméretet.

Az adatátvitelt az operatív memória kijelölt puffertérülete és a perifériális berendezés között a központi vezérlőegység kezdeményezi, de a tényleges átvitt már az egyes berendezések fölé rendelt perifériális vezérlőegységek végzik.

Az adatátvitelt meggyorsíthatjuk, ha két puffertérületet jelölünk ki. Ekkor felváltva kapjuk a file rekordjait hol az egyik, hol a másik puffertben. Ezzel elérhetjük, hogy amíg az egyik blokk feldolgozása folyik, addig a soron következő blokkal feltöltődik a másik puffertérület. Ezt kettős puffertelésnek is szoktuk nevezni.

c) Hozzáférés a rekordokhoz

Blokkolatlan rekordok esetén a blokkal természetesen már a rekordot is elértük. Blokkolt rekordoknál azonban további feladatot jelent a rekordoknak a puffertérületen történő elérése. Itt különbséget kell tennünk logikai és fizikai szintű file-kezelés között. Logikai szinten a programozónak nem kell törődnie az átvitel fizikai egységével, a blokkal. Soros file-szervezésnél az a követelmény, hogy minden újabb makróutasítás a file soron következő rekordját tegye hozzáférhetővé. Ez kétféle módon lehetséges:

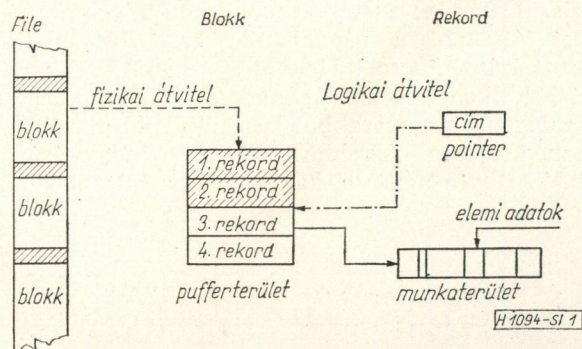
- Egy általános regiszter megadásával kijelölünk egy pointermezőt. A file-kezelő software azután ebbe helyezi el a soron következő rekord címét függetlenül attól, hogy a rekord melyik puffertérületen helyezkedik el.
- Az operatív memóriában kijelölünk egy újabb puffert — nevezzük ezt munkatérületnek —, amely éppen egy rekord befogadására alkalmas. Minden újabb logikai átviteli utasítás hatására a file kezelő software a file puffertérületéről ebbe mozgatja át a soron következő rekordot.

Amint a puffer kimerült, a következő logikai átviteli utasítás hatására a file-kezelő software automatikusan behozza a file következő blokkját, és ennek első rekordját teszi hozzáférhetővé. Az 1. ábrán vázlatosan összefoglaljuk a lehetőségeket.

d) Hozzáférés az elemi adatokhoz

Ezután már egyszerű a dolgunk. Ha a logikai átvittel az egyes rekordok munkatérületre kerülnek, akkor már csak a munkatérületet kell megfelelő részekre osztanunk (névvel ellátnunk). Az egyes részekre azután a megfelelő nevekkkel hivatkozhatunk.

Ha a rekordokat a file puffertérületén és nem munkatérületen kapjuk, akkor definiálhatunk egy fiktív munkatérületet, amelynek báziscíme a pointerben levő érték lesz.



1. ábra

Output file-ok esetén a munkaterületen állítjuk össze a rekordokat és a blokk és file puffterületéről kerül ki a file-ba.

A file végén az utolsó blokk után, a megnyitáshoz hasonlóan kötelező a file lezárása (CLOSE). Megfelelő ellenőrzések és a végcímkék feldolgozása után a file lezárásával felszabadul a programhoz rendelt perifériális berendezés.

3. Adattárolás és adatvisszakeresés

A törzsadatok (Master File) tárolására szolgáló háttérmemóriák a mágnesszalagos és mágneslemez berendezések. Általában nagy mennyiségű adat hosszabb ideig tartó tárolására használjuk fel őket. A hosszabb idejű tárolás természetesen nem jelenti azt, hogy közben a file, illetőleg az egyes rekordok tartalma változatlan marad. A file-ba új rekordok kerülnek, ezt nevezzük beszúrásnak, régiek kihálnak, ez a törlés. Az egyes rekordok tartalma is változik, minthogy a törzsadatok mellett rendszerint változó adatokat is találunk. Például egy készletrekordban a készletelemet leíró adatok változatlanok, de a készlet mennyisége, a rendelés, a beérkezés stb. változnak. Ez a rekordok naprakésszé tétele.

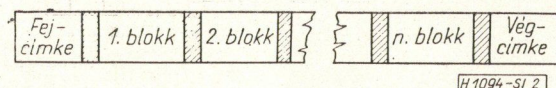
A változás adatait szoktuk tranzakciós adatoknak nevezni, ezek összességét pedig tranzakciós file-nak. Ha a tranzakciós file-t tekintjük elsődlegesnek, akkor a feldolgozásnál a tranzakciós rekordokhoz kell visszakeresni a törzsfile megfelelő rekordjait és végül előállítani a törzsfile módosított változatát.

Egy másfajta alkalmazásban a file-ból csak bizonyos feltételnek eleget tevő rekordokat akarunk ki-keresni és ezekről jegyzéket készíteni. Mindkét esetben az jelenti a problémát, hogyan tudjuk a file-t oly módon megszervezni és tárolni, hogy később bármely rekordja könnyen előkereshető legyen. Ebből a szempontból lényeges különbség van a mágnesszalagon és a mágneslemezen tárolt file-ok között.

3.1 Mágnesszalagos file-ok

A mágnesszalagos berendezés adottságainak megfelelően a mágnesszalagon csak soros file-szervezés valósítható meg. A mágnesszalagos file-ok formátumát vázlatosan a 2. ábrán láthatjuk.

A vonalkázott részek az állandó hosszúságú blokk-közöket jelzik, amelyekre azért van szükség, hogy a szalag fel tudjon gyorsulni az íráshoz vagy olvasáshoz szükséges sebességre. A blokk-közök jelentős szerepet játszanak a mágnesszalag kapacitásának kihasználásában. Kisebb blokkhosszúságnál kisebb puffterület kell a memóriában, de több blokk-köz lesz a szalagon, így rosszabbul használjuk ki a szalagot és megfordítva. Kisebb blokkméret választásával (kb. 700 byte alatt) a hasznos kapacitás 60% alá is eshet.



2. ábra

Ezen belül a file formátumát illetően a blokk és a rekord viszonya, valamint a rekordok hossza szerint a már ismertetett négy esetet különböztetjük meg. Minthogy a file rekordjai (blokkjai) között a file-kezelés szempontjából nincs semmi különbség, csak sorrendjük különböztetheti meg őket, a mágnesszalagon nincs címzés. A berendezés csak egy blokk kezdetét és végét tudja érzékelni. Ha a file-nak egy meghatározott rekordját keressük, akkor a file kezdetétől rendre be kell olvasnunk a rekordokat, amíg a keresetthez nem érünk. Ez a soros feldolgozás. A mágnesszalagos file rekordjait tehát csak úgy tudjuk módosítani, ha a teljes file-t beolvassuk, és a módosítást az operatív memóriában elvégezzük, majd a módosított rekordokat egy másik szalagra írjuk ki.

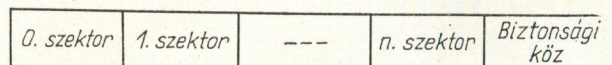
Ez az eljárás azzal a hátránnyal jár, hogy mindig a teljes file-t át kell másolnunk még akkor is, ha annak csak egy kisebb hányadát érintik a módosítások. Nagy előnye viszont, hogy automatikusan adódnak a biztonsági adatháttérrel jelentő file-generációk. A módosított file a módosítás előtti file leszármazottja, viszonyuk az apa-fiú viszony. Ha a módosított file-t is módosítjuk, és az előző file-okat megőrizzük, akkor már nagyapa—apa—fiú generációk állnak rendelkezésünkre. Ha azután utólag kiderül, hogy a legutóbbi vagy akár a megelőző módosítás hibás volt, akkor a feldolgozást szükség szerint a nagyapától vagy az apától megismételhetjük. Bármilyen volt is a hiba oka, hibás tranzakciós adatok, helytelen kezelés vagy a rendszer hibája, törzsfile-unkat helyre tudjuk állítani.

3.2 Mágneslemez file-ok

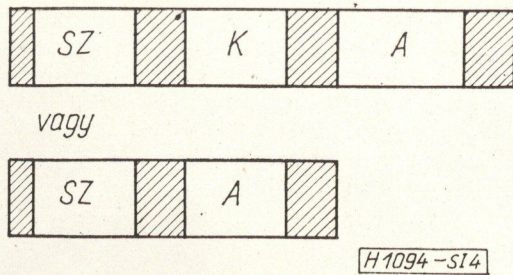
A mágneslemez, felépítésénél és működési elvénél fogva, sokkal több lehetőséget nyújt bonyolult adatstruktúrák tárolására és visszakeresésére. Mint ismeretes, a tárolás az egyes lemezfelületeken koncentrikus körök mentén, *sávokon* történik. A tengelytől azonos távolságra levő, a párhuzamos felületeken egymás fölött elhelyezkedő sávok összességét *cilindernek* nevezzük. A cylinder-fogalom bevezetésének fontos gyakorlati jelentősége az, hogy az egy cilinderen elhelyezkedő sávok egyetlen író/olvasófej-beállítással elérhetők.

Az egyes sávok *szektorokra* oszlanak. Minden szektor egy adatblokk tárolására alkalmas. A szektorokat sávonként 0-tól kezdve számozzuk. A 0. szektornak kitüntetett szerepe van; ezt a szektort a software használja fel a sáv azonosítására és a sáv tartalmának leírására. Az egyes szektorok az adatblokkok felírásának sorrendjében kerülnek a sávra.

Különös lehetőségként minden adatblokkhoz hozzárendelhetünk egy azonosításra szolgáló adatmezőt; ez a hardware-kulcs. Elnevezése onnan ered, hogy a System 4 gépeknek vannak olyan hardware utasításai, amelyek a kulcsmező értéke alapján képesek egy adatblokk megkeresésére. A lemez sávjainak formátuma a 3. ábrán látható.



3. ábra



4. ábra. Sz = számláló-mező, K = hardware kulcs, A = adatblokk

A sáv végén a biztonsági köz a forgási sebesség stabilizálásához járul hozzá. Az egyes szektorok formátumait a 4. ábra szemlélteti. A vonalkázott részek a mezők közötti elválasztó közőket jelzik. Az adatblokkot követő blokk-köz hossza az adatblokk hosszával arányosan változik. Ezzel biztosítani lehet, hogy egy kiolvasott adatblokkot a mágnesszalagtól eltérően ugyanoda vissza is írassunk. Ez a file-ok módosításánál egészen új lehetőséget jelent. Mágneslemezen nem kell a módosítandó file-t átmásolni, elegendő csak a módosítandó blokkokat beolvasni, és a módosítás elvégzése után azokat saját helyükre visszairni. Ennek az eljárásnak a neve *módosítás helyben* (update in situ).

Még több lehetőséget rejt magában az a tény, hogy a mágneslemezes file blokkjai egyedileg címezhetők. A hardware-cím két részből áll, ezek:

a sávcím és a blokkcím.

A sávcím alapján az író vagy olvasó-fejek beállnak a megfelelő cilinderre, majd bekapcsolódik a kívánt sávhoz tartozó fej. A blokkcím kétféle lehet: vagy az adatblokkhoz tartozó hardware-kulcs értéke, vagy az adatblokkot tartalmazó szektor sávon belüli sorszáma.

4. File-szervezés és file-feldolgozás

Az előző részekben beszéltünk a soros file-okról és azok feldolgozásáról. Ez a legegyszerűbb és legáltalánosabb file-szervezési mód. A mágneslemez kivételével az összes többi periférián csak ez a file-szervezési mód áll rendelkezésünkre. A következőkben ismertetésre kerülő file-szervezési és feldolgozási módok csak mágneslemezen tárolt file-oknál alkalmazhatók.

4.1 Direkt és szekvenciális szervezés

Teljes általánosságban mind a file-szervezésben, mind a feldolgozásban két esetet különböztetünk meg: a direkt és a szekvenciális módszert.

Mint hogy a feldolgozási mód a file struktúrájától többé-kevésbé függetlenül választható, ezért gyakorlatilag 4-féle feldolgozási esetre juthatunk:

- direkt file-ok direkt feldolgozása,
- direkt file-ok szekvenciális feldolgozása,
- szekvenciális file-ok direkt feldolgozása,
- szekvenciális file-ok szekvenciális feldolgozása.

A direkt file-szervezésre, illetőleg feldolgozásra az jellemző, hogy az adatkezelő software szempont-

jából a file bármely rekordja azonos elhelyezkedésűnek számít. Bármely rekord elhelyezése vagy bármely rekorhoz való hozzáférés elvben azonos feladatot jelent. A file rekordjait nem a fileban elfoglalt sorrendjük, hanem pontosabban meghatározott tárolóhelyük különbözteti meg. Más szóval ez azt jelenti, hogy a direkt file rekordjaira címükkel hivatkozunk. Hangsúlyozni kell azonban, hogy a tárolóhelyek között nincs kitüntetett. A file felállításakor az egymást követő rekordoknak nem kell szomszédos tárolóhelyre kerülniük. Hasonlóképpen a feldolgozás során minden egyes rekordot direkt hivatkozással veszünk elő, külön megadva a címét.

Ha a file rekordjainak fizikai tárolása nem feltétlenül követi a rekordok logikai sorrendjét, de valamilyen módon ez a sorrend előállítható, akkor szekvenciális file-szervezésről beszélünk. Az említett logikai sorrend rendszerint a rekordok valamilyen kitüntetett mezőjén (Control Field) alapul. Szokás ezt rekord-kulcsnak is mondani. A kulcsmező jellegétől függően azután a file rekordjait numerikusan vagy alfanumerikusan sorba rendezhetjük. (Ez a rekord-kulcs nem tévesztendő össze az adatblokkhoz rendelt hardware-kulccsal. A rekord-kulcsnak a rekordon belül nem feltétlenül szükséges összefüggő adatmezőt alkotnia.) Ha a felállított file-ban a rekordok fizikai sorrendje egybeesik logikai sorrendjükkel, akkor soros file-ról beszélünk. A soros file-ok tehát a szekvenciális file-ok speciális esetét alkotják. Ezeknél a rekordkulcs el is maradhat, minthogy a rekordok sorrendje meghatározó jellegű.

A szekvenciális file-szervezés azt is jelenti, hogy a file rekordjait tárolásuk sorrendjétől függetlenül, logikai sorrendjükben bármikor elővehetjük. Direkt módon szervezett file rekordjai között is fennállhat valamilyen logikai sorrend. Ha a file rekordjait ebben a sorrendben dolgozzuk fel, akkor beszélünk direkt file szekvenciális feldolgozásáról.

Mint láttuk a lemezes tárolóknál — ezeket szokták direkt elérésű tárolóknak is nevezni — előkerül a címzés lehetősége, illetőleg a file rekordjainak azonosításánál a címzés problémája. A címzés nálunk a file kezdetéhez viszonyított relatív címek megadását jelenti. Ehhez két adat kell: a rekordot tartalmazó sáv címe (sorszáma), és a kérdéses sávon belül a rekordot tartalmazó adatblokk (szektor) sorszáma.

4.2 Címzési módszerek

A fontosabb címzési módszerek lényegében három nagy csoportba oszthatók:

- címgenerálás,
- indexelés,
- a címek láncolása.

A címgenerálást a direkt file-szervezésben, illetőleg feldolgozásban alkalmazzuk. Ehhez feltételezzük, hogy minden rekord azonosítható egy kulcsmezővel. A probléma általában onnan ered, hogy ha file-ok rekordjait kulcsaik szerint sorba rendeznénk, akkor az így kapott kulcstartomány sokszorosan nagyobb terjedelmű lenne, mint a ténylegesen tárolandó rekordok összessége. Ezért ezek a rekordkulcsok átalakítás nélkül nem alkalmasak közvetlen címzésre. A címgenerálás elvi feladata olyan algoritmus előállí-

tása, amely a kulcstartománynak egy tárolásra alkalmas címtartományra leképezését állítja elő. Rendszerint azt is meg kell kívánnunk ettől a leképezéstől, hogy a kulcstartomány egyes szakaszain összetörölődő rekordokat a címtartományon egyenletes eloszlásúvá alakítsa át. Nem kell kikötnünk viszont a leképezés egyértelműségét, megengedhetjük, hogy több rekord kulcshoz tartozzék ugyanaz a cím; ebben az esetben a leképezés egy több rekord tárolóhelyéül szolgáló blokk címét adja. Általában csak a rekordok tényleges eloszlásának ismeretében, empirikus úton lehet kiválasztani a többféle lehetséges kulcstranzformáció közül a legmegfelelőbbet. A címtranzformáció lehetőséget ad arra is, hogy a file felállításakor a címtartományt ne töltsük be teljesen. Így a később beszűrő rekordok részére tudunk helyet biztosítani.

Indexelés

Az információ visszakeresésénél a leggyakrabban alkalmazott módszer az indexelés. Az index lényegében egy táblázat, amelynek rovatai a rekordkulcsok és a tárolóhelyek közötti összefüggést adják meg. Ily módon egy adott rekordhoz való hozzáférés két lépésből áll. Először az indexben megkeressük a rekord kulcsához a tárolóhely címét, azután az adott címről előkeressük a rekordot. Három fő indexelési módot ismerünk:

a) A bázisindex

Ennél a módszernél az indexben a file minden rekordja számára a rekordkulcsok sorrendjében fenntartunk egy rovatot. Ebben a rovatba minden rekordkulcs mellé beírjuk a tárolóhely címét is. Keresésnél így egy adott kulccsal rendelkező rekordot könnyen megtalálunk, még ha fizikai elhelyezése a kulcssorrendtől eltérő is. A probléma új rekordok beszűrőkor vagy törlésekor jelentkezik. Akkor vagy minden esetben át kell rendeznünk az indexet, hogy az új rekordok rovatai helyükre kerüljenek, vagy valamilyen más módszerrel (például címláncolással) kell az index rovatainak szekvenciális sorrendjéről gondoskodnunk.

b) Az implicit index

Implicit indexről akkor beszélünk, ha az indexben minden lehetséges rekordkulcs számára fenntartunk egy rovatot. Ennek a módszernek az az előnye, hogy új rekordok beérkezése nem okoz nehézséget, és nem kell a rekordkulcsokat is tárolnunk. Nyilvánvalóan nem használható fel olyan esetekben, amikor a kulcstartomány jelentősen meghaladja a ténylegesen létező rekordok számát.

c) Küszöbindex

Ennél a módszernél fel kell tételeznünk a file rekordjainak valamilyen rendezettségét. A rekordokat kulcsaik növekvő sorrendjében töltjük be, és meghatározott számú rekordból álló csoportokat — mondhatnánk logikai blokkokat — állítunk össze. Az index ebben az esetben kulcsként az egyes csoportokban előforduló legnagyobb értékű kulcsot, címként pedig

az egyes csoportok első rekordjának címét fogja tartalmazni. Ez nyilvánvalóan tárolóhely-megtakarítást jelent, mert az indexnek csak annyi rovata lesz, ahány rekord-csoportot a file-ban kialakítottunk. Új rekordok beszűrő, illetve törlése itt is nehézséget jelent. Ennek részleteit azonban az indexes-szekvenciális file-szervezésnél fogjuk ismertetni.

Címláncolás

A file rekordjainak szekvenciális sorrendjét biztosíthatjuk oly módon is, hogy minden egyes rekord tárolásánál külön információként megadjuk a szekvenciális sorrendben az utána következő rekord címét is. Ezek a címek képletesen egy láncot alkotnak; e lánc segítségével a file rekordjai bármelyik ponttól kezdve logikai sorrendben elővehetők. Ezeket a címzési információkat szokták a rekordok pointermezőinek nevezni. Az elnevezés arra utal, hogy egy-egy ilyen pointer mintegy rámutat a sorrendben következő rekord kezdetére. Természetesen semmiféle korlátozás nincs arra nézve, hogy egy adott rekordnak hány pointermezője lehet, vagyis egy rekordon hány információs lánc haladhat keresztül. Két információs lánc esetén például kézenfekvő, hogy az egyikkel a file elejétől végéig, a másikkal pedig ellenkező irányban haladva tudjuk elővenni a rekordokat. Ilyen címláncok alkalmazása azonban sokkal bonyolultabb adatstruktúrák kialakítását és kezelését is megengedi.

5. A System 4 file-kezelő software-je

5.1 File Control és Data Management

Az eddigiekben olyan általánosított műveletekről beszéltünk, amelyeknek operandusai file-ok, illetőleg a file rekordjai voltak. Az operandusok szerint a feladatkört ketté lehet választani:

- a file-kezelés (File Control), amely a file-t tartalmára való tekintet nélkül összetartozó egységnek tekinti és így végez vele műveleteket;
- az adatkezelés (Data Management), amely a rekordokkal megint tartalmukra való tekintet nélkül foglalkozik; feladata a rekordok egymás közötti kapcsolatainak, lényegében a file-on belüli adatstruktúrának a kialakítása és a rekordok mozgatása.

A gyakorlatban elmosódik a különbség a File Control és a Data Management között. Szoktak általában file-kezelésről beszélni, amibe beleértik az adatkezelést is, vagy adatkezelésről, hozzá tartozónak érezve a File Control-t.

A System 4 standard adatkezelő software-jének elnevezése is File Control Package — a továbbiakban FCP —, míg a file-kezelés egy részét segédprogramok látják el.

A következőkben röviden összefoglaljuk a file-okkal kapcsolatos műveleteket.

a) A file felállítása:

- a file formátumának kialakítása; standard címkék megadása,
- a tárolóközégek közötti adatmozgatás,
- a rekordok betöltése.

b) A file karbantartása:

- a file új rekordokkal történő bővítése,
- a rekordok állandó mezőinek módosítása,
- rekordoknak a file-ból való törlése,
- a file átszervezése; a file struktúrájának, egyszerűbb esetben a rekordok sorrendjének megváltoztatása.

c) A file naprakészé tétele:

Ez rendszerint az a rutinfeldolgozás, amely a file rekordjaiban levő változó mezők aktualizálását végzi.

d) A file törlése.

e) Egyéb:

- a file tartalmának átmásolása más tárolókra; biztonsági háttért alkotó file-generációk kialakítása.

5.2 System 4 File Control Package

A System 4 operációs rendszerébe tartozó FCP-nek két szintje van:

- Fizikai szinten történik a tényleges adatmozgatás az operatív memória és a perifériális berendezések között; a fizikai FCP rutinjai a Supervisor részét alkotják, és az egyes programok futtatásakor állandóan a memóriában vannak.

szimbolikus file-név	DTF...	kulesszó ₁ =aktuális paraméter, kulesszó ₂ =aktuális paraméter, kulesszó _n =aktuális paraméter
----------------------	--------	---

Vannak kötelező és vannak igényektől függően megadható kulcsszavak. Ilyen kötelező paraméter például a blokkméret byte-okban, a programunkban definiált puffterület címe szimbolikus nevével megadva; soros file-oknál a tárolóközeg (illetőleg ami ezzel egyenértékű, a perifériális berendezés fajtája), a file típusa (input vagy output) stb.

pl.	<i>PELDA</i>	DTSR	DEVICE = <i>TAPE</i> , TYPEFL = <i>INPUT</i> , IOAREA = <i>PUFFER</i> , BLKSIZE = 450
-----	--------------	-------------	--

A pozíció makró általános alakja:

$$\text{szimbolikus file-név} \parallel \mathbf{DTF} \dots \parallel p_1, p_2, \dots, p_n$$

ahol $p_1, \dots, p_n$ aktuális paraméterek. Ezek között is vannak kötelezők és elhagyhatók. Elmaradt paraméter helyét két egymás után következő elválasztójel (vessző) jelzi.

Az imperatív makrók egyetlen operandusa szekvenciális feldolgozásnál a szimbolikus file-név, mint-hogy ezek a makrók mindig a sorrendben a következő rekordot teszik hozzáférhetővé.

pl.	PUT	<i>FILEA</i>	vagy
	GETIS	<i>FILEB</i>	

Direkt feldolgozásnál a file-definíciótól függetlenül az imperatív makró paramétereként is megadhatjuk

- Logikai szinten a felhasználó közlekedhet az FCP-vel; a logikai FCP rutinjai az aktuális igényeknek megfelelően beépülnek a felhasználó programjaiba.

A logikai FCP rutinjaira makróutasításokkal hivatkozhatunk. Ezek a makrók funkciójuk szerint két csoportra oszthatók:

- definíció makrók és
- imperatív makrók.

A program elején definiáljuk a felhasználni kívánt file-okat, erre szolgálnak a definíciós makrók.

Az ezekkel a file-okkal végzendő speciális műveleteket az imperatív makrókkal írhatjuk elő. Minden definíció többé-kevésbé meghatározza, hogy az adott file-lal milyen műveleteket végezhetünk. Ez egyben a file védelmét is szolgálja, mert például nem fordulhat elő, hogy „véletlenül” olyan file-ba írjunk, amelynek definíciójában ezt nem engedték meg.

5.3 Definíciós makrók és imperatív makrók

A definíciós makróknak kétféle alakjával találkozhatunk:

- kulcsszavas makrók és
- pozíciós makrók.

Az első csoportban a file-ra vonatkozó információkat a kódapon mintegy kérdőív kitöltéseként adjuk meg. Általános alakban:

$\text{kulcsszó}_1 = \text{aktuális paraméter},$
 $\text{kulcsszó}_2 = \text{aktuális paraméter},$
 $\text{kulcsszó}_n = \text{aktuális paraméter}$

a puffterterület címét és hosszúságát. Lényeges paraméter a pointer címe. Minden író- vagy olvasó-utasítás előtt a pointer-mezőbe kell helyoznunk a kívánt blokk relatív címét. Ha hardware-kulcsot is felhasználunk, akkor a paraméterek között szerepelnie kell a hardware-kulcs befogadására kijelölt kulcsmező címének is.

```
pl.  || READ || FILEC, (ID), PUFFER, 600,
      ||      || KULCS, POINTER
```

Ha a feldolgozás alatt végig ugyanazokat a paramétereket használjuk, akkor az imperatív makróban ezek a paraméterek elmaradhatnak, mert mindegyik makrónál a file-definícióban megadott paraméterek lesznek érvényesek.

pl. || **READ** || *FILEC*, (ID)

6. Standard file-kezelés

A következőkben a System 4 standard file-kezelő software-jének, az FCP-n belül rendelkezésre álló file-kezelő eljárásait ismertetjük.

6.1 Soros file-kezelés

A file-definíció makró a **DTFSR**. Ez a lassú perifériális berendezéseknél és mágnesszalagnál alkalmazható egyetlen módszer, amellyel lehetővé válik

a rekordok logikai szintű kezelése. Természetesen felhasználhatjuk lemezes file-oknál is. Az adatok logikai szintű kezelését a mágnesszalagnál már ismertettük.

6.2 Direkt file-kezelés

A file-definíciós makró a **DTFDT**. Ezt a módszert használhatjuk fel lemezes file-ok fizikai blokkjainak (szektorainak) közvetlen elérésére. Logikai szinten egy adott szektor eléréséhez a file-kezeléshez viszonyított relatív sávcímét és a sávon belüli relatív szektorcímét kell a pointer-mezőbe helyezni. Hardware-kulcs alkalmazása esetén a pointerbe csak a relatív sávcím kerül, a szektorcím helyett a hardware-kulcs aktuális értékét kell a kulcsmezőbe helyezni.

Logikai szinten a file által elfoglalt sávokat 0-tól kezdve sorszámozzuk. A file kezdeténél az első cylinder sávjai a 0, 1, 2, ..., 9. sorszámot kapják, a következő cylinder első sávja a 10, és így tovább. A cylindereket a lemez kerületétől a tengely felé, a sávokat pedig az egyes cylindereken felülről lefelé számozzuk. A relatív szektorcím a sávon belüli sorszámot jelenti. Minden sávon az első szektor a 0 sorszámu, de ezt a software foglalja le. Az első, adatblokkot tartalmazó szektor az 1-es sorszámu.

A direkt file-kezelés imperatív makrói a pointer által előírt szektorok átvitelét végzik. A szektorban levő adatblokk felbontása rekordokra, illetőleg összeállítás a rekordokból a programozó feladata.

Külön makróutasítások szolgálnak a file formátumának kialakítására. A file felállítása ugyanis rendszerint két lépésben történik:

- a formátum kialakítása,
- az adatblokkok betöltése.

A két művelet szétválasztásának oka kézenfekvő: egy sávon belül a szektoroknak csak relatív helyzetük van. A hardware nem tudhatja, hogy hol van egy sáv harmadik szektorának a kezdete, ha a másodikat még nem vittük fel.

A betöltést és a feldolgozást is végezhetjük direkt módon és szekvenciálisan. A két módszert keverni is lehet. A direkt feldolgozáshoz szükséges címeket vagy cimgeneráló algoritmussal számítjuk vagy indexből vesszük. Az index felállítását és karbantartását is programmal kell elvégeznünk, minthogy az indexes direkt file-kezelés standard rutinjai (a **DTFID** makró) egyelőre nem hozzáférhetők.

6.3 Indexes-szekvenciális file-kezelés

A legtöbb lehetőséget nyújtó változat az indexes-szekvenciális file-szervezés. Az FCP itt a többszintes indexelés és a címláncolás előnyeit kapcsolja össze. A file-definíciós makró a **DTFIS**.

Az alapgondolat egyszerű. A file-t két részre bontjuk: indexekre és tárolóterületre. A tárolóterület megint tovább bomlik elsődleges adatterületre és túlsordulási területekre.

Minden rekordban egyértelműen ki kell jelölnünk egy numerikus hivatkozási mezőt, ez lesz a rekord-kulcs. A rekordokat kulcsaik növekvő sorrendjében töltjük be és a feldolgozás során ezzel a kulccsal hivatkozunk a kívánt rekordra.

Ez a file-szervezési mód hallatlanul megkönnyíti a felhasználó számára a legkellemetlenebb feladatot: rekordokat szekvenciális sorrendbe beszúrni vagy onnan törölni.

A file betöltésekor a rekordok blokkoltan az elsődleges adatterületre kerülnek. A file felállításakor előre gondolhatunk a későbbi bővítésre úgy, hogy az elsődleges adatblokkokban helyet hagyunk a majdan érkező rekordoknak. Ha az idők folyamán annyi új rekord érkezik, hogy az elsődleges adatblokkból már kicsordul, akkor a túlsordult rekordokat az FCP automatikusan a túlsordulási területen helyezi el, de pointermezőkkel az elsődleges adatblokkhoz láncolja őket.

Ez a file-szervezési mód maximálisan alkalmazkodik a lemeztároló fizikai adottságaihoz.

Az indexes-szekvenciális file-oknál éppen ezért cylinderenként találunk elsődleges adatterületet és túlsordulási területet, tehát az elsődleges adatblokkból túlsorduló rekordokat a fej mozgatása nélkül elővehetjük a cylinder túlsordulási területéről.

Az egész file-hoz tartozik azonban még egy túlsordulási terület. A teljes biztonság ugyanis megkívánja, hogy ha olyan szerencsétlenül oszlanának el az újonnan érkező rekordok, hogy egyes cylindereken már a túlsordulási terület sem képes befogadni őket, még mindig legyen tárolóhely számukra. Ezt a területet a független túlsordulási területnek nevezzük. Ide csak végszükség esetén kerülnek a rekordok, de azok címét az FCP megint csak hozzáláncolja az elsődleges adatblokkhoz.

Ez a három tárolóterület — az elsődleges adatterület, a cylinderek túlsordulási területe és a független túlsordulási terület — együttesen képes a file logikai blokkjainak a tárolására úgy, hogy emellett még a lemeztároló leggazdaságosabb kihasználását biztosítsa.

A file logikai blokkján egy meghatározott kulcstartományba eső kulccsal jelölt rekordok összességét értjük. A logikai blokkokat úgy kell kialakítani, hogy kulcstartományaik között ne legyen átfedés. A logikai blokkok kialakítását a file felállításakor végezzük oly módon, hogy megadjuk a kulcstartomány felső határát, vagyis a kérdéses blokkokhoz tartozó legnagyobb logikai kulcsot. A file felállításakor meglevő rekordok általában nem töltik ki a kulcstartományt. A felhasználó problémája éppen az, hogy a később érkező rekordok hogyan fognak eloszlani az egész file-ra kiterjedő kulcstartomány egyes részei, a logikai blokkok között. A későbbi eloszlásra vonatkozó előzetes ismeretek alapján kell kialakítani az elsődleges adatblokkok betöltési arányát, valamint a cylinderek elsődleges adatterülete és túlsordulási területe közötti arányt. Az FCP rendkívül rugalmasan minderre lehetőséget biztosít.

Ezután már bárhogyan bővül is a file, az FCP címláncolással mindenkor biztosítja, hogy a file rekordjait szekvenciálisan, logikai sorrendjüknek megfelelően dolgozhassuk fel. Mindig a leggazdaságosabb elhelyezésre törekszük, ezért ha törléssel hely szabadul fel egy elsődleges adatblokkban, automatikusan úgy rendezi át a rekordokat, hogy a felszabadult helynek megfelelően visszahozhasson egy túlsordult rekordot

a cylinder és ennek eredményeképpen esetleg a független túlsordulási területről is.

6.4 Fizikai file-kezelés

A file-definíciós makró a **DTFPH**. Ez a módszer minden file-nál alkalmazható. A file-kezelést fizikai szinten végzi, a logikai FCP rutinjait nem használja fel. A **DTFPH** makróutasításai lényegében a Supervisor input/output rutinjainak közvetlen hívását végzik. A file formátumának kialakítása, a blokkok elhelyezése és a rekordok blokkolása is a programozót terheli. Ezt a módszert csak olyan esetekben érdemes használni, amikor a standard file-kezelő eljárások nem nyújtanak kielégítő megoldást.

7. Adatbázis

7.1 Output-orientációjú rendszerek

A szokásos adatfeldolgozási rendszerekben az adatokat egy meghatározott terület specifikus igényeinek kielégítésére gyűjtik össze, tárolják és tartják naprakész állapotban. Természetesen fellép az a kényszer, hogy a speciális felhasználás igényei szerint alakítsák ki az elemi adatok sorrendjét és optimális tárolását. Ebből adódik az a helyzet, hogy egy adott szervezeten belül (vállalat, intézet, intézmény) azokat az adatokat, amelyeket különböző célokra használnak fel, különböző helyeken is tárolják, méghozzá az egyes céloknak megfelelően esetleg más alakban és sorrendben. Ez vezet a decentralizált adatkarbantartáshoz és a merev, output-orientációjú rendszerek kialakításához.

Ilyen esetekben különösen nehéz az egyes adatok konzisztenciájának biztosítása. Ezek a rendszerek meglehetősen ellenállnak az integrációs törekvéseknek. Egy speciális információs igény kielégítéséhez például több részterület feldolgozási eredményeit kell összesíteni, illetőleg kivonatot készíteni belőlük (feltéve természetesen, hogy adataik konzisztensek, és egyformán naprakész állapotban vannak).

Különleges erőfeszítést kíván output-orientációjú rendszerek alapján integrált adatfeldolgozási rendszer kialakítása. A rendszer közelítő integrálása csak hosszas rendszertervezői és programozási munkával, az egyes rendszerek közötti interface-programokkal és átviteli file-okkal biztosítható. Sajnálatos módon azonban valamely terület adatainak strukturális változása kihat a feldolgozó programokra és a kapcsolt területekre is. Így ez az adatok karbantartásán kívül a programrendszer állandó karbantartását is megköveteli. További problémát jelent az adatok bizalmas jellegének megőrzése, ha az integrálás érdekében megszüntetjük a részrendszerek elszigeteltségét. Nehéz követni és ezért szabályozni is azt, hogy az interface programokon keresztül meddig juthatnak el információk, és nehéz behatárolni a hozzáférés jogosultságának körét is.

Végül még egy megoldatlan kérdés output-orientációjú rendszereknél: az adatokhoz való célszerű hozzáférés távoli végállomásokról.

7.2 Az adatbázisra épülő rendszerek

A ma már rendelkezésre álló nagykapacitású, közvetlen elérésű tárolóberendezések, adatátviteli rend-

szerek és adatszervezési módok lehetővé teszik egy adott szervezet adatainak központi kezelését. Ez egy szélesebb körű igényeket kielégítő olyan adatbázis kialakítását jelenti, amely több távoli végállomásról is elérhető és amelyből a tárolt adatok bármilyen csoportosításban visszakereshetők.

A központilag kezelt adatbázis nagy előnye, hogy csökkennek az adatok összegyűjtésére és naprakész állapotban tartására fordított erőfeszítések. Megszűnik a többszörös tárolás és ezzel a konzisztencia problémája is. Nem kizárólag előre kialakított output igényeket lehet csak kielégíteni, így a felhasználás köre nincs egy meghatározott területre korlátozva. Az adatbázis minden felhasználója igényeinek megfelelő összeállításban kérheti a naprakész információkat, mégpedig az elemi adatok szintjén. Könnyen ellenőrizhető és biztosítható, hogy bizalmas jellegű adatokhoz, csak arra jogosultak férjenek hozzá.

Miután itt az elsődleges feladat az adott szervezetre jellemző valamennyi adat összegyűjtése és központi kezelése, szokás az adatbázisra épülő rendszereket input-orientációjú rendszereknek is nevezni. Az adatbázis gyakorlatilag azt jelenti, hogy pl. egy vállalatban belül nem az egyes osztályok, főosztályok felelősek a kérdéses terület file-jainak karbantartásáért, hanem a vállalat számológépes szervezeti egysége. Egy ilyen rendszer megvalósítása természetesen hosszabb és gondosabb előkészítést igényel, mint a szokásos rendszerek területenkénti kiépítése.

7.3 Az adatbázist kezelő software

A hardware adottságai mellett két lényeges programozási technika kifejlesztésének is köszönhető az adatbázisra épülő rendszerek kialakításának lehetősége. Kidolgoztak ugyanis az információvisszakereséséhez szükséges olyan új file-szervezési módszereket, amelyek az eddigieknél sokkal hajlékonyabb elérést biztosítanak, és olyan rendszerprogramokat, amelyek az aktuális adatstruktúráról független feldolgozást tesznek lehetővé. A felhasználónak természetesen nincs sok gondja az új módszerekkel. Az adatbázist felhasználó alkalmazási programokban egyszerűen előírja, hogy milyen információs egységek kigyűjtését, visszakeresését kívánja.

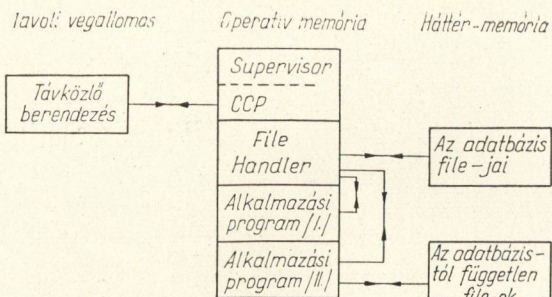
Az adatbázist *fizikailag* az adott szervezethez összegyűjtött elemi adatokat tároló file-ok összessége alkotja. Ezeknek a „fizikai” file-oknak a struktúrája általában megegyezik az eddigiekben ismertetett adatstruktúrákkal. A felhasználó a fizikai file-októl függetlenül az elemi adatok szintjén hivatkozhat az adatbázisra. Ezekből az elemi adatokból (amelyekre névvel lehet hivatkozni) tetszés szerinti módon állíthatja össze a feldolgozás *logikai* egységét, az *információs egységet*. A névvel ellátott információs egységek a file rekordjaihoz hasonló szerepet töltenek be. Speciális esetben egybeeshetnek valamelyik fizikai file rekordjaival, általában azonban olyan elemi adatmezőkből állnak össze, amelyek ténylegesen több fizikai file között oszlanak meg.

Az ICL cégnél kidolgozás alatt áll a System 4 adatbázist kezelő software-je, a DBMS (Data Base Management Software). A DBMS a hozzá tartozó segédprogramokkal együtt az adatbázis felállítását és karbantartását végzi, valamint az alkalmazási progra-

mok és az adatbázis közötti közlekedést biztosítja. A DBMS egyik alapvető része a File Handler. Ez automatikusan működésbe kezd, amint valamelyik alkalmazási program hozzá akar férni az adatbázishoz. A File Handler-t egyidejűleg több alkalmazási program is használhatja. Elsődleges feladata, hogy az alkalmazási programokban felmerülő, információs egyedekre vonatkozó igényeket értelmezze, és az igények kielégítéséhez szükséges adatátviteli műveleteket végrehajtsa.

A felhasználó programjában előforduló információs egyed (például **READ A** alakban, ahol **A** a már korábban definiált információs egyed neve) hosszú és bonyolult műveletsort, file-kezelési eljárást indít meg. A File Handler egy sor táblázatot tart fenn, amelyekből megtudja, hogy milyen műveleteket kell elvégezni ahhoz, hogy az adatbázisból előkeresse a kívánt információs egyed elemi adatait. Ezek a táblázatok együttesen alkotják a szótárat, amelyet a rendszer mágneslemezen tárol. Ebből mindig csak a keresett egyedhez szükséges rész (egy információs szegmens) kerül az operatív memóriába. Ez biztosítja, hogy az adatbázisból mindazok a fizikai file-ok hozzáférhetőek legyenek, amelyekre egy információs egyed előállításához szükség van. Pontosabban, a DBMS rutinjai egy új információs egyedre vonatkozó kérést logikai file-okra vonatkozó igényekre fordítanak le, és azután állítják össze a logikai file és az adatbázist alkotó fizikai file-ok közötti kapcsolathoz szükséges műveleteket.

Többször ismétlődő feldolgozáshoz az adatbázisból kivonat is készíthető információs file alakjában. Ez ténylegesen fizikai file, amelynek rekordjai az információs egyedek. Az információs file hosszabb ideig tartó használata azonban felveti az adatok érvényességének a kérdését; hogyan maradnak naprakészen az adatbázisból kiszakítva. Ehhez vagy az információs file karbantartását kell biztosítani, vagy a kivonat készítését kell időnként megismételni.



5. ábra

H-1094-SI 5

7.4 Az adatbázis karbantartása

Az adatbázis karbantartásánál meg kell különböztetnünk „külső” és „belső” módosítást. Az első esetben a számológépes rendszeren kívül keletkező változások átvezetését értjük: ilyen pl. új adatok bevitele az adatbázisba. Belső módosításnál az alkalmazási programok futtatásakor automatikusan adódó update-ra (naprakésszé tételre) kell gondolni; pl. új halmozott összegek egy elszámolási rendszerben. Biztonsági okokra is tekintettel, a DBMS készülő változata nem engedi meg az adatbázis közvetlen módosítását. Külön file-okat tart fenn a változtatások adatai számára (Changes files).

Az információs egyedek alakjában közölt módosításokat a rendszer átalakítja logikai file-okra vonatkozó módosításokká és ezekből készít egy közbenső transzformációs file-t. Megfelelő ellenőrzés, rendezés és összeolvasztás után kerülhet sor a fizikai file-ok módosítására.

Az adatbázis fizikai file-jainak biztonságát a DBMS automatikus generációs rendszere biztosítja.

Adatbázisra épülő rendszer vázlatos szervezési ábrája az 5. ábrán látható.

Az ábrában szereplő CCP (Communication Control Package) a távolsági adatátvitelt kezelő software-csomag; fogadja, értelmezi, sorbaállítja és kiszolgálja a távoli végállomásról érkező igényeket.

SZEMLE

Összeállította: BALOGH PÁL

A közeljövőben jelentős mértékben fejlődik a szovjet számítástechnika, és több új működési elvű és felépítésű berendezés jelenik meg.

A Szovjetunióban nagy súlyt helyeznek a nagy memóriakapacitású integrált áramkörökkel szerelt számítógépek építésére. Ezek a berendezések a számolás sebességét mintegy meghatározzák.

Sokat várnak a Minszk 32 típusú számítógéptől, aminek műveleti sebessége 30 000/másodperc.

A közelmúltban kezdték meg az ME 1000 jelű termelésirányító-számítógépcsalád sorozatgyártását. Ennek egyik tagját a komplex információfeldolgozó rendszerekben fogják alkalmazni. Az elektronikai miniszter nyilatkozata szerint a számítógépek döntően megváltoztatják a szovjet tervezés, irányítás és termelés módszereit. (*Electrie*, 1970. 24. k. 5. sz.)

*

Tesla Pardubice a Bull-General Electric-től vásárolt licenc alapján gyártja a Tesla 200 típusú elektronikus számítógépet, ami műszaki szempontból fejlettebb, mint a második számítógép-generáció tagjai, de nem éri el a harmadik generációs

komputerek színvonalát. (*Funkamateur*, 1970. 19. k. 8. sz.)

*

Az IEEE New Yorkban a Mullard mátrixelrendezésben 5×7 egyedi katódot tartalmazó kijelzőcsövet mutatott be, amivel valamennyi számjegy és betű megjeleníthető. (*Radio Mentor*, 1970, 36. k. 2. sz.)

*

Elsősorban sztereolemezjátszók, magnetofonok, playback-rendszerek sztereoleőerősítő-fokozatai számára fejlesztette ki CA 3052 jelű lineáris integrált erősítőjét az RCA.

A 16 kivezetésű dual-line-tokozású lineáris erősítőegység főbb jellemzői:

Feszültségerősítés	53 dB
Bemenőellenállás	90 kohm
Torzításmentes kimenőfeszültség	2 V _{eff}
Tápfeszültség	16 V
Környezeti hőmérséklet	-25...+85 °C

(*Electronic Industries*, 1970. 1. k. 1. sz.)

(Folytatás a 313. oldalon)

Az ICL System 4 szállítási feladat programcsomagjának ismertetése*

ETO: 681.32.06 System 4:65.012.221.656

A cikk célja a System 4—50/70 számítógép szállítási feladatot megoldó programcsomagjának ismertetése. A programcsomagot az ICL cég készítette és a számítógép egyéb alkalmazási programcsomagjaival együtt, a gép software-jéhez tartozik.

A leírásban elsősorban a programcsomag használatával kapcsolatos lehetőségeket ismertetjük. Nem térünk ki a szállítási feladat alkalmazási lehetőségeinek tárgyalására, továbbá ismertetjük fel a matematikai programozás fogalmait és a lineáris programozás és szállítási feladat kapcsolatát.

A szállítási feladat a lineáris programozási feladat általános esetének speciális változata. A feladat a következőképpen fogalmazható meg:

Határozzuk meg a

$$\sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}$$

lineáris függvény minimumát úgy, hogy a benne szereplő változók eleget tegyenek a következő feltételeknek:

$$1. \quad x_{ij} \leq 0$$

$$2. \quad \sum_{j=1}^n x_{ij} = a_i$$

$$3. \quad \sum_{i=1}^m x_{ij} = b_j$$

$$4. \quad \sum_{i=1}^m a_i = \sum_{j=1}^n b_j$$

ahol $i=1, 2, \dots, n$ és $j=1, 2, \dots, m$.

A c_{ij} pozitív egész számokat szállítási költségeknek, az x_{ij} egész számokat szállítandó mennyiségeknek, az a_i és b_j pozitív egész számokat pedig feladóhelyeknek, illetve rendeltetési helyeknek nevezzük.

Bebizonyítható, hogy a szállítási feladat bármelyik lehetséges megoldása esetén a nullától különböző x_{ij} elemek száma legfeljebb $m+n-1$. Ezt a tényt a program készítői messzemenően kihasználták a memóriában való helyfoglalásnál.

A szállítási feladat — programban felhasznált — megoldási módszerére vonatkozó további információkat az alábbi két munka tartalmaz:

G. Hadley: „Linear Programming”, 9. fejezet. Addison — Wesley, 1962.

P. T. Mcintosh: „Initial solutions to sets of related transportation problems”, Op. Res., 14. kötet, 1. sz., 1963. márc.

A program a klasszikus szállítási feladat megoldására szolgál, azaz kiszámítja, hogyan lehet azonos minőségű terméket adott feladóhelyekről adott rendeltetési helyekre elszállítani úgy, hogy a szállítási összköltség minimális legyen.

A programcsomag önmagában zárt, tehát a tényleges számítások elvégzése mellett gondoskodik az adatok beolvasásáról, ellenőrzéséről, esetleges módosításáról és az eredmények kinyomtatásáról is. Gépidő-takarékossági szempontból igen lényeges a a programnak az a tulajdonsága, hogy egy megoldott feladat eredményét hasonló feladatok megoldásához közvetlenül fel tudja használni. A program másik előnyös sajátossága, hogy az elszállítandó mennyiségek (a_i) és a szükségletek (b_j) oszlop-, ill. sorvektorából egyazon futtatás során többet is megadhatunk.

A számítások elvégzése után a gép automatikusan kinyomtatja a szállítási költséget és a kapott szállítási tervhez tartozó x_{ij} értékeket. A felhasználó kérheti a duális változók kinyomtatását is. A duális változók értékéből lehet kiolvasni azt, hogy melyek azok az — optimális szállítási programhoz tartozó — költségek, melyeknek csökkentése az optimális összköltséget is csökkentené. Kinyomtathatók továbbá a program futása során kapott részeredmények is.

A programcsomag működtetéséhez szükséges minimális konfiguráció:

- 65 K byte operatív memória (System 4 számológép, J-szintű, lemezes operációs rendszerrel)
- 1 kártyaolvasó, vagy szalagolvasó,
- 1 cserélhető lemezegység (az operációs rendszernek),
- 3 mágnesszalagos egység, vagy 1 cserélhető lemezegység,
- 1 sornyomtató, vagy 1 további mágnesszalagos egység off-line nyomtatáshoz.

A továbbiakban a program működésének részleteiből leírásával, a szolgáltatott eredmények interpretálásával foglalkozunk és néhány, a program futására vonatkozó időadatot adunk meg.

1. A programcsomag szegmensei és leírásuk

A program a következő szegmensekből áll:

- Paraméter input
- Adat input
- Induló megoldás
- Számítás
- Output.

1.1 Paraméter és adat input szegmens

Ez a két szegmens a paraméterek és az adatok beolvasását végzi.

* Az ismertetés az „ICL System 4—50/70 Transportation Reference Manual” alapján készült.

Beérkezett: 1971. IV. 22

A paraméter input szegmens megvizsgálja a file paramétereket és különféle indikátorokat állít be az input- és output-követelményeknek megfelelően. A továbbiakban ezek a paraméterek vezérlik a program futását.

Az adat input szegmens valamely feladat első futtatása során beolvassa és tárolja a szállítási feladat adatait. A beolvasott adatokról — ha a felhasználó úgy kívánja — sornymutatón listát készít és a kinyomtatott sorokat sorszámmal látja el. További futtatások során az egyes sorokra ezekkel a sorszámmal lehet hivatkozni. Így lehetséges bizonyos sorok kicserélése, törlése, vagy újabb sorok beszúrása is.

Az adat input szegmens ellenőrzi a beolvasott adatokat, a feladat méreteit és az esetleg megadott ellenőrző összegek helyességét is. Ha az ellenőrzések során hibát talál, az ellenőrzést folytatja ugyan, de a futást a szegmens végén megállítja.

1.2 Az induló megoldás szegmense

Ez a szegmens az adat input szegmens által létrehozott adatfile-okat használja fel az induló bázis megkereséséhez. Ennek módszere a szakirodalomból „északnyugati sarok” szabály, vagy az „oszlopminimumok módszere” néven ismeretes.

Ha olyan szállítási feladatunk van, mellyel azonos méretű, de tartalmilag más feladat megoldását már ismerjük, az induló megoldást úgy kaphatjuk meg, hogy az új feladat költségégeit módosítjuk az ismert feladat duális változóival. Ha az ismert feladat duális változói u_i és v_j , akkor a módosított költség-elemek $c_{ij} - u_i - v_j$ lesznek.

1.3 A számítások szegmense

Ebben a szegmensben történik az induló megoldás sorozatos javítása.

Az optimális megoldás megkeresésének algoritmus a Dantzig „Stepping Stone” módszerén alapszik, amelyet H. I. Scoins alakított át a bázis gráf-struktúrájának megfelelő követelményeknek. Az algoritmus olyan, hogy csak a bázisba bevont mennyiségeket kell az operatív memóriába tárolni. A számítások szegmense az induló bázist az induló megoldás szegmenséből, a bináris adatfile-t pedig az adat input szegmensből veszi át.

A program általában az optimális megoldás megtalálásáig számol, de előfordulhat az is, hogy a számolást — pl. a gépidőhiány miatt — meg akarjuk szakítani. Annak érdekében, hogy a számolást megszakítás esetén ne kelljen előlről kezdeni, a program a felhasználó által specifikált időközönként (meghatározott számú interáció után) az újraindításhoz szükséges információkat háttérmemórián rögzíti. Megszakítás esetén a program kiírja a duális változók értékét. Az újraindítás úgy történik, mintha egy hasonló (azonos méretű) feladatot akarnánk megoldani a duális változók ismeretében.

Az induló bázis meghatározása után az első lépés az u_i és v_j duális változók meghatározása. A bázisba bevont mennyiségekre a $c_{ij} - u_i - v_j$ érték nulla lesz. Mivel e mennyiségek száma $m+n-1$, azért $m+n-1$ darab $c_{ij} - u_i - v_j = 0$ alakú egyenletünk lesz. A változók száma $m+n$. Valamelyik u_i vagy v_j tetszőleges

rögzítése esetén tehát a többi $m+n-1$ érték könnyen meghatározható. A program az u_i duális változót nullának veszi, a többi duális változót pedig a gráfon végighaladva, a $v_j = c_{ij} - u_i$ képlet segítségével határozza meg.

A program ezután több, a bázisban nem szereplő rendeltetési helyre is kiszámolja a $c_{ij} - u_i - v_j$ értéket és következő lépésként azt a helyet vonja be a bázisba, amely a legnagyobb negatív értékek rendelkezik.

Azt, hogy a költségmátrix hány oszlopát kell bevonni a legnagyobb negatív érték meghatározásához, a program minden egyes iteráció után maga dönti el. A vizsgálandó oszlopok számát a program úgy határozza meg, hogy a tényleges számolási idő és a mágnesszalagról való olvasás ideje minimális legyen.

A bázisból elhagyandó csomópont meghatározására a program ezek után két listát készít a csomópontokról a báziscsomópontból az új szállítandó mennyiséget definiáló két csomóponthoz. A mindkét listában szereplő csomópontokat törölve, a megmaradó csomópontokból új listát készít. Ily módon egy olyan hurok jön létre, mely tartalmazza az új mennyiséget is az előző bázis elemei mellett. A váltakozó mennyiségek közül a legkisebbet törli a bázisból, a bázis többi tagját pedig a hurok ezen mennyiségével módosítja (növeli vagy csökkenti). A program az új bázisnak megfelelően a gráfot is módosítja. A fenti eljárást addig kell folytatni, amíg a $c_{ij} - u_i - v_j$ értékek minden i -re és j -re pozitívak nem lesznek. Ha valamennyi érték pozitív, a megoldás optimális.

1.4 Output szegmens

Az output szegmens a megoldást a számítások szegmenséből, az eredeti adatfile-t pedig az adat input szegmensből veszi át. Az eredmények kinyomtatása on-line vagy off-line történhet. Ha a paraméterlistában azt írjuk elő, hogy a számítást egy új feladóhely-oszloppal és rendeltetési hely-sorral meg kell ismételni, a vezérlést az induló megoldás szegmense veszi át. Egyébként a program az output szegmens végén befejeződik.

1.4.1 Részeredmények nyomtatása

A felhasználó által megadott paraméternek megfelelően a program (a számítások szegmense) a következő részeredményeket nyomtatja ki:

- a) az interáció sorszáma,
- b) az összköltség,
- c) az összköltség csökkenése az utolsó iteráció óta.

1.4.2 A végeredmény kinyomtatása

A számítások befejezése után a program először külön lapon kinyomtatja a feladat nevét, azonosító számát és az eredményként nyert minimális költséget. Ezután részletezi az egyes szállításokat. Ez a részletezés soronként a következőket tartalmazza: a feladóhely neve, a rendeltetési hely neve, az elszállítandó mennyiség, egységár, a szállítás költsége. Az eredményt feladóhelyek és ezen belül rendeltetési helyek szerint, vagy fordítva lehet rendezni — a felhasználó kívánságának megfelelően. Kívánság sze-

rint a duális változók értékét is ki lehet írni. A nyomtatásnak ez a része a következőket tartalmazza: a feladóhely (vagy rendeltetési hely) neve, az állomásra vonatkozó szállítások összköltsége és a duális változó értéke.

Ha a megoldás olyan elemeket is tartalmaz, melyekhez nem adtunk meg költségelemet, a minimális költség értéke 0 lesz.

1.4.3 Az eredmények interpretálása

A megoldásban szereplő költségelemek és a hozzájuk tartozó két duális változó összege megegyezik. Ez az összefüggés igaz lehet a megoldásban nem szereplő utak költségelemére is, ami azt jelenti, hogy az adott megoldásokon kívül létezik más, minimális költségű megoldás is az illető út bevonásával.

Ha valamilyen oknál fogva meg akarjuk változtatni a kapott megoldást — vagyis ha valamelyik igényt más helyről akarjuk kielégíteni — akkor a szállítási költség a következőképpen változik meg:

$$\text{költségváltozás} = x(c_{ij} - u_i - v_j),$$

ahol

x — a régi úton szállított mennyiség,

c_{ij} — az új úthoz tartozó költségelem,

u_i és v_j — az új úthoz tartozó duális változók.

2. A program futására vonatkozó időadatok

A táblázatban szereplő adatok tapasztalati úton nyert adatok. A „Teljes gépidő” rovatban szereplő idők a beolvasás, számítás és nyomtatás idejét tartalmazzák.

Feladóhelyek száma	Rendeltetési helyek száma	Teljes gépidő		Az interációk száma
		min	sec	
6	23	1	08	22
20	25	1	14	50
23	265	3	35	395
45	121	2	20	292
50	100	2	27	194
57	1300	61	46	5527
88	123	3	03	447
241	244	15	40	3544
294	294	36	10	869

SZEMLE

(Folytatás a 310. oldalról)

Az ez ideig családi tulajdonban levő Grundig cég 1971 áprilisától kezdődően részvénytársasággá alakul. A Grundig cég a nyugatnémet rádió-TV magnetofon piacnak jelenleg 20%-át uralja, 1969. évi forgalma (Mehrwertsteuer nélkül) 980,5 millió DM volt.

*

Az egyik legnagyobb japán elektronikai cég, a Sony Corporation nemrégiben bejelentette, hogy kisképernyős PAL rendszerű színes TV-készülékeket szállít Angliába és Franciaországba. A bejelentés érdekessége abban áll, hogy a szóban forgó készülékekbe olyan, ún. Trinitron típusú képső kerül beépítésre, amely a színes TV-rendszerektől függetlenül bármely színes TV-készülékbe beépíthető, mivel a megfelelő vétel kizárólag a készülék kapcsolástechnikai megoldásától függ. A Sony Trinitron típusú képsőve a hagyományos árnyékmasszkos képsővekkel ellentétben csak egy elektroda-rendszer tartalmaz. A Trinitron képsőveknek ugyanakkor nagy hátránya, hogy a maximális képméret 48 cm, s mint ismeretes, a kisképsőves színes TV-készülékek Nyugat-Európában nem keresettek. A szóban forgó 31 cm-es színes készülékeket egyébként Angliában 199,5 £-os, Franciaországban pedig 3150 Fr-os áron hozzák forgalomba. (Handelsblatt, 1971. febr.)

*

SG 3801 típusjelzéssel „quick-chip” integrált áramkört fejlesztett ki a Silicon General, amellyel számos különféle kapcsolás valósítható meg. A morzsa 50 monolit félvezető eszközéből és ellenállásából a felhasználó megfelelő összeköttetések létesítésével önállóan alakíthatja ki a számára szükséges kapcsolást. Így gyorsan és olcsón valósíthatóak meg a kísérleti és prototípusáramkörök. Nagyobb darabszámú igény esetén maszk készítésével könnyen és gyorsan alakíthatóak ki különleges áramkörök is. A „quick-chip” integrált áramkörök 1 GHz frekvenciáig nagy megbízhatósággal üzemelnek. (Internationale Elektronische Rundschau, 1970. 5. sz.)

Az Elfein 830 típusszámú reed-jelfogója mechanikai védelemmel rendelkezik. A műanyagból készített tekercstestet zsugorított, víztaszító műanyaghüvely rögzíti az érintkezőket tartalmazó üvegsőhöz. A jelfogót nyomtatott áramkörökhöz illeszkedő hajlított lábakkal vagy egyenes kivezetésekkel szállítják. A jelfogó ródiumozott záróérintkezővel készül.

Főbb jellemzői:

Kapcsolási teljesítmény	max 10 W
Kapcsolási feszültség	max 100 V
Kapcsolási áram	max 0,5 A
Az érintkező átmeneti ellenállása	max 250 mohm
Meghúzó feszültség	6—24 V
Meghúzó teljesítmény	max 165 mW
Az élettartam névl. telj. mellett	min 10 ⁷ kapcsolás
Környezeti hőmérséklet	—20...+100 °C

(Bauelemente der Elektrotechnik, 1970. 34. sz.)

*

Semitest III típusmegjelöléssel TTL-, DTL-, DTLZ-, ECL- és RTL-áramkörök, valamint tárolók, regiszterek, számlálók és frekvenciaosztók gyors vizsgálatára alkalmas készüléket fejlesztett ki a Rohde-Schwarz. A berendezéssel dual-in-line, flat-pack, TO-5 vagy egyéb, legfeljebb 16 kivezetésű tokozással rendelkező áramkörök vizsgálhatóak. Csatlakozó kábel segítségével nyomtatott áramkörökre már felszerelt integrált áramkörök is mérhetőek. A berendezés 12 V-os tápfeszültségét 8 db 1,5 V-os szárazelem szolgáltatja, de külső forrásból 40 V-ig terjedő feszültségek is csatlakoztathatóak. A vizsgálatokhoz szükséges óra-impulzust beépített RS-flip-flop szolgáltatja, természetesen külső vezérlőimpulzusok is rákapcsolhatóak a berendezésre. (Bauelemente der Elektrotechnik, 1970. 32. sz.)

(Folytatás a 316. oldalon)

A System 4 PERT programcsomagja

ETO 681.32.06 System 4:65.012.221

A gazdasági életben, különösen a termelésirányításban és a műszaki fejlesztés terén az egyre növekvő szervezési feladatok szükségessé teszik, illetve tették olyan módszerek kialakítását, amelyek biztosítják a vezető számára az egyre bonyolultabbá váló tevékenységek felett az áttekintést, a munkafolyamatok koordinálását.

Ismert szakemberek véleménye és a tapasztalatok szerint a hálótéchnika az utóbbi évtized leghatásosabb vívmánya a szervezés terén, fontos segédeszköz a vezetői — irányítói munkában. A hálótervezési módszerek előnyeit a következőképpen foglalhatjuk össze:

1. Bármely tervezési feladat ellátására alkalmas.
2. Kizárja egyes tevékenységek figyelmen kívül hagyását.
3. Biztosítja a folyamatok szemléletes ábrázolását.
4. Lehetővé teszi a tevékenységek ellátásához szükséges erőforrások reális felmérését.
5. Az esetleges tervkorrekciók hatása mennyiségiileg mérhető.
6. Lehetővé teszi az időszükséglet objektív becslését.
7. Biztosítja a kivitelezők munkájának koordinálását.
8. Lehetőséget nyújt a rendelkezésre álló erőforrások legjobb kihasználására és a fejlesztés legrovidebb időn belüli elvégzésére.
9. Egységes tervezési és beszámolási formák bevezetését teszi lehetővé.

A felsorolt előnyök ellenére sajnálatos tény, hogy a hálótervezési módszerek még mindig nem foglalják el azt a helyet, amelyet — különösen tervgazdálkodó országban — hasznosságuknál és könnyű alkalmazhatóságuknál fogva megérdemelnének. Ennek oka sok esetben a megfelelő számítógépes modell hiánya.

Az ICL ismertetésre kerülő PERT programcsomagja könnyen kezelhetően kapcsolja össze az időelemzést és az erőforráselemzést, lehetőséget nyújt hálóak összekapcsolására és a lehetséges output-ok bő választéka is feltétlenül kielégít minden felhasználói igényt.

A PERT programcsomag két fő részének — a TIME ANALYSIS (időelemzés) és a RESOURCE ANALYSIS (a projekt tevékenységeihez szükséges erőforrások szétosztása) — ismertetésénél feltételezzük a módszer lényegének ismeretét.

1. Time analysis (időelemzés)

A programcsomagnak ez a része az időbeli ütemezést adja. Meghatározza a kritikus utat és a kritikus zónát, az egyes tevékenységek kezdetének és befeje-

zésének legkorábbi és legkésőbbi időpontját (abszolút időben vagy dátum szerint), a tartalékidőket, valamint a tevékenységek időtartamában rejlő bizonytalanságból eredő szórásokat.

1.1 Bemenő adatok

A hálót definiáló adatok (tevékenységlista, időbecslések) mellett még a következő paramétereket kell megadni:

1. A futtatás típusát jelző paraméter:

- a) NEW: Ez a paraméter azt jelzi, hogy egy új háló adatait akarjuk beolvasni. A gép ellenőrzi az adatokat és elhelyezi a hálót a könyvtárban (mágnesszalagra vagy lemezre) majd végrehajtja az időelemzést.
- b) REVIEW: Ezt a paramétert akkor adjuk meg, ha a háló már a könyvtárban van, de módosítani akarjuk és úgy akarjuk újra lefuttatni.
- c) DELETE: Azt jelöli, hogy a könyvtárból törölni akarjuk a szóban forgó hálót.

2. A háló azonosítója.

3. Futás-szám.

Ezt a számot 0-tól kezdve a háló minden egyes újrafuttatásakor 1-gyel növelni kell.

4. Indikátor annak jelzésére, hogy az időelemzést erőforráselemzés is fogja-e követni.

5. A felhasználó és a projekt neve.

6. Az időadatok típusa.

7. Munkaszüneti napok.

8. Időbecslések száma.

9. Az eredmények formátumának és a kívánt rendezéseknek a specifikációja.

1.2 Eredmények

A háló beolvasásakor a gép szintaktikus és logikai ellenőrzéseket végez és a hibákról jegyzéket készít.

Hibátlan háló esetén az időelemzés után kinyomtatja a felhasználó által kívánt és paraméterkártyák segítségével megadott táblázatokat.

Egy-egy táblázatban tevékenységenként a következő rovatok szerepelhetnek:

- kezdő esemény,
- befejező esemény,
- a tevékenység megnevezése,
- a tevékenység időtartama,
- legkorábbi kezdés,
- legkésőbbi kezdés,
- legkorábbi befejezés,
- legkésőbbi befejezés,
- legkésőbbi szabad kezdés,

- teljes tartalékidő,
- szabad tartalékidő,
- független tartalékidő,
- függő tartalékidő,
- a tevékenység állandó eltérése,
- a tevékenység befejező eseményének állandó eltérése,
- annak valószínűsége, hogy a tevékenység a diktált időpontig befejeződik.

Eseményre orientált táblázat esetén a lehetséges rovatok a következők:

- az esemény azonosítója,
- az esemény megnevezése,
- a legkorábbi bekövetkezés,
- a legkésőbbi bekövetkezés,
- a diktált bekövetkezés,
- előző kritikus esemény,
- csúszás,
- állandó eltérés,
- annak valószínűsége, hogy az esemény a diktált időpontig bekövetkezik.

Minden egyes táblázathoz megadhatók továbbá ún. kötetlen rovatok a tevékenységek, ill. események további jellemzésére (pl. sorszám, kivitelező, felelős stb.), amelyek elsősorban rendezési célokat szolgálnak, továbbá üres rovatok is kézi bejegyzések számára.

A táblázatban szereplő rovatok és azok sorrendjét, valamint a rendezési szempontokat a felhasználó szabadon választhatja meg.

A program lehetőséget nyújt az eredmények vonaldiagram formájában történő ábrázolására is, ami általában jobb áttekintést biztosít és sok tekintetben megegyezik a hagyományos tervezési formákkal.

Korlátozások:

- egy-egy futtatás alkalmával legfeljebb 36 táblázatot lehet készíteni és ezek mindegyikét legfeljebb 3-féle képpen (de tetszőleges sok szempont szerint) lehet rendezni;
- a táblázat megtervezésénél figyelembe kell venni, hogy egy sorba legfeljebb 160 jel kerülhet; (vonaldiagrammoknál az esetleg szükséges lapváltást és a lapok sorszámozását a program önállóan végzi);
- a felhasználó választhat, hogy a táblázatban az eseményeket vagy a tevékenységeket nyomtatja-e ki, de egy táblázaton belül nem szerepelhetnek események és tevékenységek is;
- az üres rovatok száma táblázatonként legfeljebb 6, a kötetlen rovatoké pedig tevékenységre orientált táblázatoknál 6, eseményre orientált táblázatoknál 3 lehet.

1.3 Korlátozások a háló nagyságára vonatkozóan

131 K byte-os operatív memória és 1 időbecslés esetén a következő korlátozások érvényesek:

1. Az események maximális száma 11 300 lehet.

2. Minden eseményből legfeljebb 250 tevékenység indulhat ki és mindegyikben legfeljebb 250 találkozzhat.

Három időbecslés esetén az eseményekre vonatkozó korlátozás a fenti szám kétharmadára csökken.

2. Resource analysis (Erőforrások elemzése)

Az időelemzés után a programcsomagnak ez a része állapítja meg a projekt erőforrás-igényét és biztosítja a rendelkezésre álló erőforrások legjobb kihasználását.

Sok esetben a legfontosabb szempont a projekt határidőre történő befejezése, még akkor is, ha ezt a rendelkezésre álló erőforrások nem teszik lehetővé. Ebben az esetben *időkorlátos* projektről, illetve időkorlátos allokálásról beszélünk és a program elsődleges feladata a határidő betartása mellett az erőforrások egyenletes kihasználásával a többlet-igény minimalizálása.

Abban az esetben, ha a kapacitás-szintek még a határidő eltolódásának esetén sem léphetők túl *erőforráskorlátos* allokálással állunk szemben.

A leggyakoribb az időkorlátos és erőforráskorlátos allokálás közti kompromisszumos megoldás. A kompromisszum mértékét a felhasználó ún. döntési táblák segítségével adhatja meg.

2.1 Bemenő adatok

Az erőforráselemző program felhasználja az időelemzés eredményeit. Az időelemző program az eredmények kinyomtatásával egyidejűleg elkészíti az erőforráselemzés bemenő adataul szolgáló file-t is, ezzel kapcsolatban tehát a felhasználónak semmi gondja sincs.

Ezenkívül még a következő adatokra van szükség:

1. A futtatás típusát jelző paraméter:

- a) ADD: a háló első allokálásánál,
- b) CHANGE: a további allokálásoknál,
- c) DELETE: a háló és minden hozzá tartozó paraméter törlése a könyvtárból.

2. A háló azonosítója.

3. Futás-szám; az időelemzésnél ismertetett szabálynak megfelelően.

4. Az időadatok típusa.

5. Az egyes erőforrásokból rendelkezésre álló kapacitás. A kapacitást specifikáló rekord a következő részekből áll:

- a) erőforrás-azonosító;
- b) az erőforrás megnevezése;
- c) a halmazhatóság jelzése;
- d) a ciklikusság jelzése és a ciklus meghatározása;
- e) a rendelkezésre álló kapacitás mennyiségi megadása. Maximálisan 27 kapacitás-szint adható meg. Az egyes szinteket a következő adatok jellemzik:
 - az időpont, amelytől kezdve a kapacitás rendelkezésre áll;

- a normális kapacitás;
 - a küszöb-szint.
6. Az egyes tevékenységek erőforrásigénye. Az ezt specifikáló rekord a következő részekből áll:
- a) az erőforrásigény típusa: egyszerű vagy komplex;
 - b) a tevékenységet meghatározó kezdő és befejező esemény;
 - c) a megszakíthatóság jelzése;
 - d) a kötés jelzése;
 - e) a tevékenység prioritása (decimális szám 00–99-ig; ha nem adjuk meg, akkor a program 50-et tételez fel; 50-nél nagyobb szám kisebb prioritás, kisebb szám nagyobb prioritást jelez);
 - f) az igényelt erőforrások felsorolása; erőforrásonként
 - az erőforrás azonosítója,
 - az erőforrásigény típusa (össz-szükséglet vagy kvantált szükséglet),
 - az igényelt mennyiség (0–99998);
 - g) komplex erőforrás esetén meg kell továbbá adni:
 - azt az időt, amely a tevékenység kezdetétől az erőforrásigény felépítéséig eltelik,
 - az erőforrásigény időtartamát.
 - h) Létrát alkotó tevékenységek közönséges erőforrásigényét az eddig említett módon kell megadni. Specifikálni kell ezenkívül még magát a létrát is (a létra fokait alkotó tevékenységeket).

Korlátozások:

- Tevékenységenként legfeljebb 27 egyszerű és 18 komplex erőforrásigény adható meg.
- A létrák fokainak száma 2–14-ig terjedhet.

7. Az allokálás típusának megválasztása.

- a) Erőforráskorlátos allokálás.
- b) Időkorlátos allokálás.
- c) Allokálás az idő-, illetve erőforráskorlátok túllépésével. Az arányt a felhasználó szabadon választhatja meg.

8. A kívánt eredmények specifikálása.

2.2 Eredmények

A kizárólag időadatokat tartalmazó táblázatokat a gép az időelemzés eredményeihez hasonló formában szolgáltatja. A táblázatok az eddigiek mellett még a következő rovatokat tartalmazhatják:

- allokált kezelés,
- allokált befejezés,
- maradék tartalékidő,
- első megszakítás időtartama,
- második megszakítás időtartama,
- prioritási szám.

Az erőforrásra orientált eredmények két fő csoportra oszthatók:

- a) hisztogramok;
- b) táblázatok az erőforrás-felhasználásról, táblázatonként 5 erőforrásra vonatkozóan.

Tudatában vagyunk annak, hogy a cikk már terjedelme miatt sem tud teljes képet adni a „System 4” PERT programcsomagja által nyújtott összes lehetőségről. Reméljük azonban, hogy a leírás az érdekeltek körében visszhangra talál, a programcsomag részletesebb megismerésére és alkalmazására ösztönöz.

SZEMLE

(Folytatás a 313. oldalról)

Az Electronics Weekly becslése szerint az USA elektronikai iparának jelenlegi termelési értéke a nyugat-európai 10,5 milliárd dollárral szemben 33,5 milliárd dollár, s a nemzetközi élvonal sorrendje: USA, Japán, NSZK, Anglia, Franciaország.

Fenti számok egyúttal azt is jelentik, hogy az USA termelése kb. négyszer akkora, mint a sorrendben következő négy országé együttvéve. Az elektronikai iparon belül a legnagyobb mérvű növekedés világszerte kétségtől az elektronikus adatfeldolgozó és számítógépek tekintetében észlelhető.

Az elektronikai világpiacon jelenleg még kétségkívül az amerikai óriások uralkodnak, de komoly verseny várható Japán részéről, ahol a komputeripar éves növekedési rátája eléri a 35%-ot, s melynek elektronikai ipara 1980-ban már 12 milliárd dolláros értéket fog produkálni.

A jelenlegi fejlődés egyik jellemző vonása, hogy egy cég nem képes már domináns szerepet játszani a világpiacon, — bármilyen nagy cégről legyen is szó, ezért a jövőben a licenc

és kooperációs egyezmények eddiginél is nagyobb térhódítása várható.

A vezető tőkés országok elektronikai ipari termelése — 1968. évi adatok figyelembevételével — a három fő ágazat között az alábbiak szerint oszlott meg:

	Fogyasztási	Ipari	Távközlési
	elektronika		
USA	20,1	26,7	53,2
Japán	59,4	17,6	23,0
NSZK	27,2	46,6	26,2
Anglia	25,7	46,0	28,3
Franciaország	28,4	44,9	26,7
Olaszország	41,3	42,3	16,4

(Financial Times, 1970. május.)

A forróelektron-dióda Schottky-féle fém-félvezető záróréteggel rendelkező félvezető eszköz, amelynek rendszerint n -típusú alapanyagában a többségi töltéshordozók játsszák a főszerepet. A dióda (a többi többségi töltéshordozóval működő félvezető eszközökhöz hasonlóan) elsősorban nagyfrekvenciás áramkörökben, keverőkben, demodulátorokban kerül felhasználásra. Határfrekvenciája magasabb, küszöbfeszültsége alacsonyabb, mint a $p-n$ átmenettel rendelkező félvezető eszköze, ezek gyors működését ugyanis a töltéshordozó-tárolás jelensége korlátozza.

Mechanikai szempontból a forróelektron-dióda a $p-n$ átmenetű diódákkal teljesen egyenértékű, elektromos szempontból viszont előnyei szembeötlőek:

- kis zaj
- jobb hatások
- szélesebb négyzetes karakterisztikájú tartomány
- jobb reprodukálhatóság és egyenletesebb gyártás
- jobb nagyfrekvenciás tulajdonságok, amelyeket nem korlátoz a kisebbségi töltéshordozók élettartama
- magasabb letörési feszültség
- kisebb visszarám
- az epitaxiál-planár kivitel folytán lényegesen magasabb impulzus- és átlagteljesítmény
- nagyobb mechanikus stabilitás és rázásállóság
- kisebb érzékenység a hőmérsékletváltozásokkal szemben
- nagyobb megbízhatóság

Elméletileg a forróelektron-diódák és a nagyfrekvenciás tüssdiódák egyaránt Schottky-diódaként üzemelnek, az előbbieken azonban jobban megközelítik az ideális viselkedést. (Bauelemente der Elektrotechnik, 1969. 31. sz.)

Számítógép által vezérelt elektronmikroszkóp segítségével az elektronsugarat is felhasználják az integrált áramkörök maszkjainak kialakítására. A Westinghouse kutatóinak új módszerével az eddiginél nagyobb alkatrészsűrűség érhető el. Egyetlen hélyegnagyságú felületen mintegy 4 millió elem alakítható ki. Az elektronsugaras maszkolási technológiát igen gyors működésű LSI áramkörök megvalósítása érdekében fejlesztették ki. (Elektronics World, 1969. 6. sz.)

A General Electric PA246 jelű integrált teljesítmény erősítője 16 ohmon 5 W teljesítményt szolgáltat. A tápfeszültség és a kimenőellenállás széles határok között választható meg. A dual-in-line-tok közepén két széles hűtőkivezetéssel rendelkezik, amelyek közvetlenül a hűtőfelületre forrasztandók. Az erősítő főbb jellemzői:

- | | |
|---------------------------------|-----------------|
| – teljesítmény | max 10 watt |
| – kimenőfeszültség-löklet | max 30 V |
| – tápfeszültség | 37 V |
| – jellemző bemeneti érzékenység | 12 mV |
| – környezeti hőmérséklet | – 55... + 125°C |

(Bauelemente der Elektrotechnik, 1969. 30. sz.)

A Motorola 2N5549 típusjellel keverőfokozatokban való alkalmazásra 200 MHz határfrekvenciájú térvezérléses tranzisztort hozott piacra TO-18 tokozással. Az új tranzisztor főbb jellemzői:

- meredekség 6...15 mS
- zaj 100 MHz-en 3 dB
- bemenőkapacitás 8 pF
- visszahatási kapacitás 2 pF

(Bauelemente der Elektrotechnik, 1969. 30. sz.)

A Valvo FD-jelű IC-sorozata jelenleg 6 dinamikus tolóregisztert, 1 RAM- és 1 ROM-áramkört tartalmaz. Az áramkörösorozatot folyamatosan bővítik. Az FD-sorozatú integrált áramkörök közepes integráltsági fokúak és p -csatornás növekményes MOS-elemekből épülnek fel. Az alapáramkör kisfogyasztású és igen gyors működésű kétfázisú MOS-cella.

Az FD-alapáramkörök egyszerű és egységes felépítésük folytán könnyen csatlakoztathatóak egymáshoz és segítségükkel a digitális technika majd valamennyi feladata megoldható. Főbb jellemzőik:

- valamennyi tolóregiszter egy- és kétfázisú üzemre egyaránt alkalmas
- a tolóregiszterek hossza binárisan súlyozható
- az üzemi frekvencia egyfázisú üzemben 1 MHz, kétfázisú üzemben 3 MHz
- minden áramkör komplementer kivezetésekkel rendelkezik
- MOS-, DTL- és TTL-áramkörök vezérlésére alkalmasak
- beépített bemeneti túlfeszültségvédelemmel rendelkezik
- hermetikusan zárt, 14, 16 és 18 kivezetésű dual-in-line-típusú fém-kerámia tokozásban kerülnek forgalomba.

(Bauelemente der Elektrotechnik, 1969. 31. sz.)

Az Egyesült Államokban jelenleg 2 millió tudós és mérnök dolgozik, s ezek közül 700 000-et a kutatás-fejlesztésben foglalkoztatnak. Becslések szerint tíz év múlva a tudományos és műszaki termelékenység tízszeresére növekszik a munka eredményeképpen.

Egyre inkább előtérbe nyomulnak azok a kutatások, amelyek az emberi körülmények alakulásával foglalkoznak az iparban, az erre folyósított alapokat az 1967. évi 380 millióról 1980-ra 1 milliárd dollárra kívánják növelni. Az évtized végén megközelítőleg már 800 000 tudós és mérnök fog kutatási, fejlesztési munkát folytatni, s ezeknek 70%-a az iparban helyezkedik el. Rendkívül nagy szerep jut már a közeljövőben a kutatásokban a számítógépeknek. Pillanatnyilag a műszaki felszerelések emésztik fel a kutatási, fejlesztési ráfordítások 20%-át, tíz év alatt ez az arány 25–30%-ra nő majd, tehát a jelenlegi 13 milliárd dollárról 16 milliárdra nő. Számítógépeket alkalmazzanak a tudományos információ tárolására, az információ közvetítő központokat pedig összekötik az egész USA-ban létesített adattárolókkal.

1950–1965 között 55%-kal nőtt a kutatási, fejlesztési programok kivitelezési költsége, leginkább a fejlesztés területén. Sok ipari vállalatnál a kutatás, fejlesztés ma már a haszon felét is felemészti, noha a ráfordítások tetemes részét a szövetségi kormány finanszírozza.

A vezetések feltétlenül világosan kell látnia a kutatások időigényét, s itt gyakran le kell küzdenie a szervezettségtől idegenkedő tudósok, kutatók és mérnökök ellenállását. Ma egyre nagyobb jelentőségűvé válik az információ és a tervezés, ezért van szükség a menedzserek és kutatók együttműködésének alaposabb, nagyvonalúbb szervezésére. Nem elég pénzügyileg támogatni a kutatást, fejlesztést, a vezetőknek maguknak is tisztában kell lenniük a kutatás jelentőségével. Így például a menedzsereknek tisztában kell lenniük azzal, milyen óriási jelentőségük van az alapkutatásoknak a műszaki fejlődés szempontjából, hiszen ez lesz a jövő alkalmazott kutatása. A ma ipari kutatása holnap térül meg – feltéve, hogy korszerű az ipari kutatásirányítás. (Research Management, 1968. 6. sz.)

Az utóbbi években különösen megnövekedett Bulgáriában a különféle ipari automatizációt szolgáló berendezések gyártása. Ezek között elsősorban a különféle folyamatszabályozó és ellenőrző műszereket, valamint a termelést szabályozó műszereket, elektronikus és elektromos mérőműszereket és gépeket, számítógépeket kell megemlíteni. Ehhez az ipárhoz 22 modern gyár tartozik, három kutató, tervező és konstrukciós intézet, valamint három műszaki fejlesztő bázis. Az elektronikai műszerek gyártása Szófiában, a hidraulikus és pneumatikus műszerek gyártása Plovdivban koncentrálódik. Az anyagi és technikai bázis lehetővé teszi Bulgáriában a termelés jelentős növelését. Az elmúlt öt év alatt több mint négyszeresére növekedett a termelés. A termelés 50%-át 30 országba exportálták, így elsősorban a Szovjetunióba és a többi szocialista országba, valamint az NSZK-ba, Belgiumba, Franciaországba, Brazíliába. Ugyanezen idő alatt az export ötszörösére növekedett. Nagyon jól szerepeltek a nemzetközi piacon a bulgár gyártmányú digitális számítók, az elektronikus asztali számológák, az ELKA-65-21 típus, az analóg számológák, az A-1, A-2, A-3 típus. A legtöbb vegyipari üzem részére szállítanak különféle ellenőrző és szabályozó műszereket.

ket. Nagy volumenben állítják elő az úgynevezett elektronikus számológát ipari célokra, amely készülékek magas technikai nívóval rendelkeznek és sok ország felé exportálják. Itt elsősorban a PEB 41 IMI impulzusszámlálót kell említeni, amely összekapcsolható a Miniprint 25 típusú kiíróval.

A különféle digitális mérőműszerek sorát állítják elő üzemek, laboratóriumok és tudományos kutatóintézetek részére. Ezek közül is meg kell említeni a VS-100-S típusú digitális voltmétert, melynek méréshatára 50–99–995 V-ig terjed. A VS 52 univerzális műszer szintén ebbe a kategóriába tartozik, amellyel a frekvenciamérést, impulzusszámlálást és kronométers méréseket lehet végezni. Nagyon kedvező eredményeket értek el a TS-151 típusú digitális frekvencia időmérőnél, amelynél már szilícium félvezetőt alkalmaznak. A készülék 0–30 MHz-ig terjed és végpontossága 500 mV. Egyre nagyobb jelentőségre tett szert a különféle elektromos szabályozó műszerek gyártása, így elsősorban az RTN folyamatos bemenő és kimenő szignál adóműszer, amelyet különféle ipari üzemekben széles körben alkalmaznak. (*Informationen*, 1971. 2. sz.)

1971. augusztus 27–szeptember 5. között Berlinben megrendezésre kerülő Első Nemzetközi Rádió–TV Kiállításon 10 ország 200 kiállítója képviselteti magát. A külföldi kiállítók közül a legnagyobb alapterületet a japán cégek igényelték.

A nyugatnémet Elektromosipari Szövetség elnökeinek megállapítása szerint már most egyértelműen kitűnik, hogy a kiállítás középpontjában a képlemez- és képkazettaüzlet beindítása és általában az ún. képvisszaadó technika áll. Az utóbbi időben oly sokat hangoztatott színes TV túltermeléssel kapcsolatban idézett szakszövetség elnöke rámutatott, hogy költségmegtakarítási szempontból mindenekelőtt a raktárkészletek hamarosan felszívódnak. A jelenleg nehezen áttekinthető konjunkturális helyzetre való tekintettel 1971-ben az NSZK-ban 900 000 db színes TV-készülék értékesítésével számolnak. 1971 elején a nyugatnémet háztartásokban 1 300 000 db színes TV-készülék üzemelt, ami azt jelenti, hogy a háztartások 8%-a rendelkezett színes vevőkészülékkel. (*Handelsblatt*, 1971. február.)

Az angol ICL megrendelést kapott a Szovjetuniótól komputer szállítására. A szovjet rendelés egy darab ICL 1905 és egy Ferranti Argus 500/2. típusra vonatkozik. Ezt a megrendelést a leningrádi Electrosila gyár adta fel, amely a világon a legnagyobb turbogenerátorokat gyártja. A számítógépek a kiegészítő berendezésekkel együtt ebben a gyárban állítják üzembe, ahol összegyűjtik a turbogenerátorok vizsgálata alatt kapott információkat és analizálják őket. Az 1905 típusú komputert ezenkívül a szokványos kereskedelmi célokra is felhasználják, így például a termelés és raktárellenőrzésre. A rendelés értéke 850 ezer fontot tett ki. (*Lastauto-Omnibus*, 1970. 5. sz.)

A Compagnie Internationale pour l'Informatique kilenc db Iris 50 típusú komputert adott el Romániának. Az első gépet a bukaresti adatfeldolgozó-központban (ICUPEC) már fel is szerelték. Ez az üzlet tovább erősíti a CII-nek a szocialista országok piacain kiépített pozícióját. Korábban a cég ugyanis már szállított egy 90/40 típusú komputert a román atomenergia bizottság részére, Magyarországgal pedig licenccmegállapodást kötött a 10010 típusú gépeire vonatkozóan. (*Elektronics Weekly*, 1970. április.)

Az ICL 650 000 £ értékű megrendelést kapott Magyarországtól 3 db System 4–50 típusú számítógép szállítására. A berendezések az Autokernél, a Videoton gyárban, illetve a Nehézipari Minisztériumban fognak üzemelni. Az új rendeléssel az ICL magyarországi értékesítései összesen 3,7 millió £-ra emelkedtek. Az új rendelések elnyeréséhez az ICL szövívőjeszerint az idei BNV is nagyban hozzájárult, minthogy az ICL kiállításon levő 7181 típusú video display egységet és adatvégállomást telefonvonal segítségével a Tervhivatalban üzemelő 4–70-es computerrel kötötték össze. (*Finantial Times*, 1971. július.)

A szovjet–francia színes TV együttműködés keretében a „Francecoleurs” cég és a moszkvai TV-képcsőüzem konkretizálja a „grille” típusú francia színes TV szériagyártásának megvalósítására vonatkozó elképzeléseket, miszerint 1974-től kezdődően napi 100 db ilyen típusú színes TV képcsövet fognak előállítani.

A Moszkva–Párizs közötti közvetlen színes adást elsősorban a Molnija-1 szovjet mesterséges műhold van hivatva közvetíteni, ezen túlmenően azonban földi összeköttetést is igénybe vesznek. A műhold közvetítés megvalósítását illetően már folyik a készülékek beállítása, s felszerelték a „Pleumer-Bodou”-i központot, illetve megváltoztatták a vonatkozó szovjet berendezést, melyeket az adások színesben, ill. fekete-fehérben történő közvetítése céljából Párizsban, a „Barthélémy” kísérleti laboratóriumban helyeztek el.

Az ORTF is tett javaslatot a Moszkva–Párizs közti távközlési rendszer kikísérletezésére vonatkozóan, s a későbbiekben a szovjet és francia elképzelést természetesen egyeztetik.

A szovjet nézők már láttak néhány francia színes adást, és a kulturális csereegyezmény keretében az ORTF is kapott néhány szovjet művészeti és varieté programot. (*Television*, 1970. 7–8. sz.)

A japán Hitachi Ltd. találmánya lehetővé teszi félvezetőelemek (tranzisztorok) gyártásának teljes automatizálását. A cég nyilatkozata szerint a világon elsőként fejlesztették ki eredményesen ezt az eljárást. (Az eljárás ionoknak szilíciumba nagy gyorsasággal történő belövésén alapszik.)

Ez az eljárás lehetővé teszi óránként 300 000–400 000 db dióda vagy tranzisztor-elem előállítását.

Az eljárást – tömeggyártásra való alkalmazásra – még tovább kell javítani. A módszer tökéletesítése után a Hitachi reméli, hogy a külföldi technológiától való függősége valamelyest csökkenni fog. A felfedezés lehetővé teszi, hogy a termelési folyamatokat komputer alkalmazásával automatikusan lehessen ellenőrizni. (*BTA–NFA*, 1970. március.)

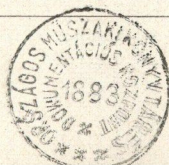
A rendelkezésre álló statisztikai publikációk szerint az NDK 1969. évi TV termelése az előző évi 400 391 db-ról 353 232 db-ra, rádiótermelése pedig 765 735 db-ra csökkent (ez utóbbi cikkben az előző évhez viszonyított csökkenés mértéke 6%).

Az alkatrészek tekintetében a csökkenés csak TV-képcsőveknél jelentkezett, amiből 1969-ben az előző évinél 20,1%-kal kevesebbet; nevezetesen 587 000 db-ot állítottak elő. A többi elektronikus alkatrész gyártása 1969 folyamán 25,4%-kal emelkedett; az 1969. évi alkatrész termelési érték 785,6 millió márka volt.

A termelés csökkenésének eredményeként az NDK-nak a KGST országok TV, ill. rádiótermelésében való részesedése 1968-ra az 1955. évi 7%-ról 5,2 %-ra (rádió), ill. 13,5%-ról 8,4%-ra (TV) mérséklődött. Ennek ellenére az NDK elektrotechnikai és elektronikai ipara dinamikusan fejlődik és az NDK ipari össztermelésének 8,9%-át szolgáltatja. (*Funkschau*, 1971. 3. sz.)

Az amerikai űrhajózási hatóságok megrendelésére az RCA olyan tárolót fog gyártani, mely laserfény segítségével 1 millió bitet írhat, tárolhat, ismételtet, olvashat vagy törölhet. Az 1,83 méter hosszú, táveső alakú berendezés 1972 tavaszán készül el, s az első olyan egység lesz, amely teljesen optikus tárolót alkalmaz. (*Blick durch die Weltwirtschaft*, 1971. július.)

A Rohde-Schwarz cég új 10 kW-os UKW adóját egy kb. 30 cm átmérőjű kerek lapról vezérlik, melyen el van helyezve a belépő áramkör teljes kapcsolása. Első alkalommal kerül sor nyomtatott áramkörök alkalmazására teljesítményadók építésénél. E fejlesztés eredményeképpen nemcsak nagyobb megbízhatóságot értek el a zavart jelentő dugaszoló kapcsolások elhagyásával, hanem a korábbi adókhoz képest kétharmadára csökkent a lap terjedelme. (*Neue Zürcher Zeitung-Handelsteile*, 1970. nov.)



Tartalmi összefoglalások

ETO 681.327.11 System 4

Trembeczki S.:

A System 4—50 típusú számítógép hardware felépítése

HÍRADÁSTECHNIKA XXII. (1971) 10. sz.

A cikk a System 4 információábrázolási formáinak összegezése után rövid leírást ad a 4—50 központi egység fő részeiről, majd a megszakítási és az input-output rendszert tárgyalja.

ETO 681.32 System 4

Kiss D.—Gajári Gy.:

System 4-es operációs rendszer

HÍRADÁSTECHNIKA XXII. (1971) 10. sz.

A cikk a System 4-es gépeslád J—MJ szintű operációs rendszerét ismerteti. Két példában bemutat egy algol program futtatásához és fordításához szükséges vezér-kártyakészletet és vázolja a kihelyezett pultos rendszer működését.

ETO 681.327 System 4

Ságody I.:

A System 4 típusú számítógépek adatközlési rendszere

HÍRADÁSTECHNIKA XXII. (1971) 10. sz.

A cikk megismerteti az adatkezelés és a file-szervezés alapfogalmait; hogyan valósulnak meg az input/output műveletek logikai szintű file-kezeléssel. Átfogó képet ad a System 4 File Control Package-ről, bemutatja főbb vonásait és lehetőségeit. Ismerteti a három fő feldolgozási módot, a soros, a direkt és az indexes-szekvenciális file-kezelést. Végül betekintést ad a készülékben levő, adatbázist kezelő software-rendszerről.

ETO 681.32.06 System 4:65.012.221:656

Dr. Gracza L.-né:

Az ICL System 4 szállítási feladat programcsomagjának ismertetése

HÍRADÁSTECHNIKA XXII. (1971) 10. sz.

A cikk a System 4 számítógépre készült és a klasszikus szállítási feladatot megoldó programot ismerteti, különös tekintettel az alkalmazott módszerre, valamint a feladat mérete és a gépidőszükséglet közti összefüggésre.

ETO 681.32.06 System 4:65.012.221

Majoros K.:

A System 4 PERT programcsomagja

HÍRADÁSTECHNIKA XXII. (1971) 10. sz.

A cikkben a System 4 számológép PERT programcsomagját ismertetjük. Külön kezeljük az időelemző és az erőforrás-allokáló részt. A hangsúlyt mindkét résznél a bemenő adatok és a várható eredmények, valamint a korlátozások ismertetésére helyezzük.

Обобщения

ДК 681.327.11 Систем 4

Трембески, Ш.:

Построение блоков ЭВМ «Система 4»

HÍRADÁSTECHNIKA (ХИРАДАШТЕХНИКА, Будапешт) XXII. (1971) № 10.

Статья, после обобщения форм изображения информации, кратко описывает важнейшие части центрального блока 4—50, потом блоки перерыва и входно-выходной системы.

ДК 681.32 Систем 4

Киш, Д.—Гаяри, Г.:

Операционная система «Система 4»

HÍRADÁSTECHNIKA (ХИРАДАШТЕХНИКА, Будапешт) XXII. (1971) № 10.

Статья излагает операционную систему уровня J—MJ семейства ЭВМ «Система 4». На двух примерах показывается комплекс управляющих карт, необходимый к программировке АЛГОЛ и переводу и описывает работу системы пульта построения в виде отдельного блока.

ДК 681.327 Систем 4

Шагоди, И.:

Система обработки данных ЭВМ типа «Система 4»

HÍRADÁSTECHNIKA (ХИРАДАШТЕХНИКА, Будапешт) XXII. (1971) № 10.

Статья излагает основные понятия обработки данных и организации запоминающего устройства, метод осуществления входно-выходных операций обработкой запоминающего устройства на логическом уровне. Дано обобщение по блоку управления запоминающего устройства ЭВМ «Система 4», показаны его важнейшие черты и возможности. Трактуются три основных вида обработки: последовательная, непосредственная и индексно-последовательная обработка запоминающего устройства. Наконец дается обзор подготовительных работ по системе программировки обработки данных.

ДК 681.32.06 Систем 4:65.012.221:656

Д-р Граца, Л.-нэ:

Изложение пакета прозранмы задачи транспорта ЭВМ «Система 4»

HÍRADÁSTECHNIKA (ХИРАДАШТЕХНИКА, Будапешт) XXII. (1971) № 10.

Статья излагает программу по решению классической задачи транспорта для ЭВМ «Система 4», принимая в особое внимание применяемый метод, а также соотношение между задачей и потребностью времени машины.

ДК 681.32.06 Систем 4: 65.012.221

Майорош, К.:

Пакет прозранмы ПЭРТ «Системы 4»

HÍRADÁSTECHNIKA (ХИРАДАШТЕХНИКА, Будапешт) XXII. (1971) № 10.

В статье описывается пакет программы ПЭРТ ЭВМ «Система 4». Отдельно трактуются блок временного анализа и блок для использования ресурсов. Подчеркивается в обеих частях выяснение входных параметров и ожидаемых результатов, а также ограничений.

Summaries

UDC 681.327.11 System 4

Trembeczki, S.:

Hardware Construction of Computer "System 4—50"

HÍRADÁSTECHNIKA (Budapest) XXII. (1971) № 10

After a summary of "System 4" data formats, the paper briefly describes the main parts of the 4—50 Central Processor Unit, then deals with the interrupt and input-output system.

UDC 681.32 System 4

Kiss, D.—Gajári, Gy.:

Operational System of "System 4" Computer

HÍRADÁSTECHNIKA (Budapest) XXII. (1971) № 10

The paper presents the J—MJ level operational system of the computer family "System 4". A control card set necessary to run and translate the Algol program is presented in two examples and the operation of system using remote terminals is illustrated.

UDC 681.327 System 4

Ságody, I.:

Data Management of Computer "System 4"

HÍRADÁSTECHNIKA (Budapest) XXII. (1971) № 10

The paper initiates into the basic concepts of Data Management and File Organisations Techniques; how to handle input/output operations via files at a logical level. It gives an outline picture of FCP, the System 4 File Control Package, describes its main features and facilities; three main accessing modes are covered: the serial, the random and the indexed sequential one. Finally, among some more advanced techniques, it mentions the System 4 Data Base Management Software which is just under development.

UDC 681.32.06 System 4:65.012.221:656

Mrs. Dr. Gracza, L.:

Survey of the Program Package for the Transport Task of the Computer ICL "System 4"

HÍRADÁSTECHNIKA (Budapest) XXII. (1971) № 10

The program solving the classical transport task made for the Computer "System 4" is described, with special respect to the applied method, and the relation between the magnitude of the task and computer time requirement.

UDC 681.32.06 System 4:65.012.221

Majoros, K.:

PERT Program Package of the Computer "System 4"

HÍRADÁSTECHNIKA (Budapest) XXII. (1971) № 10

PERT program package of the computer "System 4" is presented in the paper. The time analysis and the part for resource allocation is treated separately. In both parts the input data and the results to be expected and the explanation of the limitations are emphasized.

Zusammenfassungen

DK 681.327.11 System 4

Trembeczki, S.:

Hardware Konstruktion der Rechenanlage „System 4—50“

HÍRADÁSTECHNIKA (Budapest) XXII. (1971) Nr 10

Nach der Zusammenfassung der Informationsdatenformate des „Systems 4“ wird eine kurze Beschreibung der Hauptteile der 4—50 Zentraleinheiten gegeben. Ferner wird das Unterbrechungs- und Input-Output-System erörtert.

DK 681.32 System 4

Kiss, D.—Gajári, Gy.:

Operationssystem, des Rechners „System 4“

HÍRADÁSTECHNIKA (Budapest) XXII. (1971) Nr 10

In dem Artikel wird das Operationssystem vom Pegel J—MJ der Familie „System 4“ erörtert. Durch zwei Beispiele wird der Steuer-Kartensatz der zur Durchführung und Übersetzung des Algol-Programms notwendig ist, beschrieben. Ferner wird die Funktion des Systems mit abgetrenntem Pult illustriert.

DK 681.327 System 4

Ságody, I.:

Datenbehandlungssystem der Rechenanlage „System 4“

HÍRADÁSTECHNIKA (Budapest) XXII. (1971) Nr 10

Der Artikel erörtert die Grundbegriffe der Datenbehandlung und File-Organisationstechnik; ferner wird beschrieben, wie die Input-Output Operationen mit File-Behandlung von logischem Niveau ausgeführt werden. Es wird eine Übersicht über den System 4 File Control Package gegeben, und dessen Hauptmerkmale und Möglichkeiten illustriert. Es werden die drei Datenverarbeitungsmethoden bzw. die Serien-, Direkt und Indexfolge-Filebehandlungen beschrieben. Zuletzt wird eine Übersicht über das Software-System, welches die Datenbase behandelt, gegeben.

DK 681.32.06 System 4:65.012.221:656

Frau Dr. Gracza, L.:

Erörterung des Programmpaketes der Transportaufgaben des Rechners ICL „System 4“

HÍRADÁSTECHNIKA (Budapest) XXII. (1971) Nr 10

Der Artikel bezieht sich auf die Lösung der klassischen Transportaufgabe für die Rechenmaschine „System 4“ und erörtert das Programm, welches mit besonderem Rücksicht auf die angewandte Methode und auf den Zusammenhang der Größe der Aufgabe und des Maschinenzeitbedarfs ist.

DK 681.32.06 System 4:65.012.221

Majoros, K.:

PERT-Programmpaket des Rechners „System 4“

HÍRADÁSTECHNIKA (Budapest) XXII. (1971) Nr 10

In dem Artikel wird das PERT-Programmpaket des Rechners „System 4“ erörtert. Die beiden Hauptteile, bzw. diejenigen für die Zeitanalyse und für die Versetzung der Quellen werden von einander unabhängig behandelt. Bei allen beiden Teilen werden die Eingangsangaben und die voraussichtlichen Ergebnisse, ferner die Beschränkungen betont.



CDU 681.327.11 System 4

Trembeczki, S.:

La construction de l'ordinateur type «System 4—50»

HÍRADÁSTECHNIKA (Budapest) XXII. (1971) № 10

L'article, après une revue des formes de simulation des informations, donne une description brève des parties essentielles du groupe central et ensuite les systèmes d'interruption et d'entrée-sortie sont traités.

CDU 681.32 System 4

Kiss, D.—Gajári, Gy.:

Système d'opération de l'ordinateur «System 4»

HÍRADÁSTECHNIKA (Budapest) XXII. (1971) № 10

L'article décrit le système d'opération du niveau J—MJ de la famille d'ordinateurs «System 4». Deux jeux des cartes de contrôle pour réaliser un programme ALGOL sont présentés et l'opération du système de construction à pupitre séparé est exposée.

CDU 681.327 System 4

Ságody, I.:

Système de traitement des données dans l'ordinateur «System 4»

HÍRADÁSTECHNIKA (Budapest) XXII. (1971) № 10

L'article décrit les concepts fondamentaux du traitement des données et de l'organisation de la collection («file»); la réalisation des opéra-

tion l'entrée-sortie par un traitement logique de la collection. Une revue générale est donnée sur le paquet de contrôle de la collection du «System 4» et ses caractéristiques principales et possibilités sont présentées. Les trois méthodes de traitement de la collection: série, direct et successive sont décrites. Enfin un résumé du système «software» pour le traitement des données, étant en préparation actuellement, est donné.

CDU 681.32.06 System 4:65.012.221:656

Mme Dr. Gracza, L.:

Description du paquet de programme pour la tâche de transport de l'ordinateur «System 4»

HÍRADÁSTECHNIKA (Budapest) XXII. (1971) № 10.

L'article décrit le programme préparé pour la solution de la tâche classique de transport, par égard particulier pour la méthode appliquée, ainsi que pour la corrélation entre le volume de la tâche et le temps de l'usage de l'ordinateur.

CDU 681.32.06 System 4:65.012.221

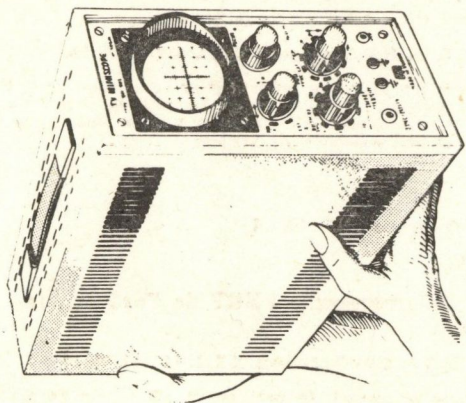
Majoros, K.:

Le paquet de programme PERT de l'ordinateur «System 4»

HÍRADÁSTECHNIKA (Budapest) XXII. (1971) № 10

L'article expose le paquet de programme PERT de l'ordinateur «System 4». Les groupes de l'analyse de temps et d'allocation des ressources sont traités séparément. Dans les deux groupes les données d'entrée, les résultats présumés et les limitations sont soulignés.

TV laboratóriumok és szervizműhelyek! Impulzustechnikai laboratóriumok! Híradástechnikai tanintézetek!



Kalibrált eltérítés, külső és belső szinkronozási lehetőség

Feszültségmérés csúctól—csúcsig

Feszültségmérés

Periódusidő-mérés

Amplitúdómoduláció mérés

Fázisszögmérés

Frekvenciamérés Lissajous-görbékkel

A készülék, mint fiók, más műszerekkel közös vázba tolható, mint önálló műszer hordozható. Eltérítőjel-generátorának áramköri megoldása speciális: az unijunction tranzisztorral felépített fűrészgenerátor lehetőséget nyújt a fűrészel finomszabályozására és így a szinkronizmus megvalósítására. Közben a fűrészel sebessége (az időalap kalibrációja) nem változik. A függőleges erősítő amplitúdóra kalibrált bemenő osztója és az eltérítőjel-generátor teszik lehetővé a vizsgált jelformák feszültségének és idejének mérését. Tápegységének feszültségei stabilizáltak.

Még megrendelhető a **TV MINISCOPE!**

Típus: TR-4351

Teljesen tranzisztorizált

MŰSZAKI ADATOK:

Függőleges erősítő

Frekvenciatartomány

DC: 0...6 MHz ($\pm$ dB)

AC: 2 Hz...6 MHz ($\pm$ dB)

Érzékenység

0,1...100 V/div

Bemenő ellenállás

névlegesen 1 Mohm 40 pF

Vízszintes erősítő

Frekvenciatartomány

10 Hz...1 MHz ($\pm$ 3 dB)

Érzékenység

jobb, mint 1 V/div

Eltérítőjel-generátor

Eltérítési sebesség

10 μ s/div...10 ms/div,

Szinkronizálás

külső és belső szinkron

Egyéb adatok

Katódsugárcső

DG 724 ($\varnothing$ 70 mm)

Hálózati feszültség

110, 127, 220, 240 V + 5%, — 10 %

Hálózati frekvencia

50 vagy 60 Hz

Teljesítményfelvétel

35 VA

Súly

kb. 4,6 kg

Méret

102 x 221 x 306 mm

Üzemidő

a készülék állandó üzemre alkalmas



Gyártja:

HÍRADÁSTECHNIKA SZÖVETKEZET

Budapest VII., Csengery u. 28

Telefon: 222-074



A műszert belföldön forgalomba hozza a

MIGÉRT

Vállalat, Budapest VI.,

Bajcsy-Zsilinszky út 37.