Magiszter

Számítástechnikai szerkesztőség

szoftver IBM PC,XT,AT kompatibilis mikroszámítógépekre

MIREP

Listagenerálási eszköztár C nyelvekhez

Programvédelem nélkül!

GONDOLA–KISS
KRAFFT–SCHILL–SZEGEDI

MIREP

Listagenerálási eszköztár

Szoftver-dokumentáció
Sorozatszerkesztő: Pintér Tibor
A kiadványt szakmailag ellenőrizte: Benedikti István

© Akadémiai Kiadó és Nyomda Vállalat
GONDOLA–KISS–KRAFFT–SCHILL–SZEGEDI, 1990
Minden jog fenntartva

A szoftver-dokumentáció a DOG programcsomaggal készült
© Akadémiai Kiadó és Nyomda Vállalat · Schill – Szegedi

56347

GONDOLA ZOLTÁN–KISS GYÖRGY
KRAFFT WALTER–SCHILL RÓBERT–SZEGEDI JÁNOS

MIREP

Listagenerálási eszköztár

Akadémiai Kiadó, Budapest 1990

MTAK
0 00003 68998 1

677241

MAGYAR TUDOMÁNYOS AKADÉMIA KÖNYVTÁRA

AKADÉMIAI KIADÓ ÉS NYOMDA VÁLLALAT
Felelős kiadó: Hazai György főigazgató
Felelős szerkesztő: Nagy Zoltán
A borítót Szabadi Zoltán tervezte
ISBN 963 05 5748 7
ISSN 0238–3896
A kézirat lezárva: 1990. május 30.

TUD. AKADÉMIA KÖNYVTÁRA
Könyvtári szám: 5000 /19.90

T a r t a l o m

1. Bevezetés 1

2. Áttekintés 3
2.1. A MIxxx adatfeldolgozási eszköztár 3
2.2. A listagenerátor alapelvei 4

3. Alapfogalmak és definíciók 7
3.1. Lista 7
3.2. Listadefiníció 7
3.3. Listaleíró file 7
3.4. Listagenerátor 8
3.5. Elemi adatterület 8
3.6. Elemi változó 10
3.7. Formátum 10
3.8. Formátumelem 10
3.9. Elemisor 10
3.10. Fejlécsor 11
3.11. Összegsor 11
3.12. Szövegkonstans 12
3.13. Adathivatkozás 12
3.14. Rendszerváltozó 12
3.15. Hashmark parancs 13
3.16. Lokális nyomtató 13
3.17. Hálózati nyomtató 13
3.18. Szűrőfile 14
3.19. Nyomtató definíció 14
3.20. Illeszkedési pont 14

4. A listaleíró nyelv 15
4.1. Opciók állítása 15
4.1.1. REPORT= 15
4.1.2. SORTED= 15
4.1.3. DEVICE= 16
4.1.4. LSTFDIR= 16
4.1.5. LSTFNAME= 16
4.1.6. SERVER= 17
4.1.7. CLASS= 17
4.1.8. COPYNUM= 17
4.1.9. PRNTYPE= 17
4.1.10. CHARSIZE= 18
4.1.11. LINESPACING= 19
4.1.12. PAGELENGTH= 19
4.1.13. DATEMODE= 20
4.1.14. OPENMODE= 20
4.1.15. DETPRINT= 20
4.1.16. FORMFEED= 21
4.1.17. FRSTFF= 21
4.1.18. LASTFF= 21
4.2. Az elemi adatterület leírása 22

4.2.1. DETAILFIELDS=..... 22
4.2.2. FIELD=..... 22
4.3. Formátumelemek..... 23
4.3.1. Rendszerváltozó..... 24
4.3.2. Elemi változó..... 24
4.3.3. Elemi változó előző értéke..... 25
4.3.4. Összegző..... 25
4.3.5. Számláló..... 26
4.3.6. Átlag..... 26
4.3.7. Szövegkonstans..... 27
4.3.8. Hashmark parancs..... 27
4.3.8.1. New_Page..... 28
4.3.8.2. Total_Page..... 28
4.3.8.3. Total_Header..... 28
4.3.8.4. Detail_Header..... 29
4.3.8.5. Detail_Select..... 29
4.3.8.6. Detail_Print..... 29
4.3.8.7. Header_Select..... 29
4.3.8.8. Header_Print..... 30
4.3.8.9. Cr_Lf..... 30
4.3.8.10. Char_Size..... 30
4.3.8.11. Line_Spacing..... 31
4.3.8.12. Page_Length..... 32
4.3.8.13. Page_No..... 32
4.3.8.14. Printer_Control..... 32
4.4. Az elemisor formátuma..... 33
4.4.1. DETAIL=..... 33
4.4.2. DFORMAT=..... 33
4.5. A fejlécsor formátuma..... 34
4.5.1. HEADER=..... 34
4.5.2. HFORMAT=..... 35
4.6. Az összegsor formátuma..... 36
4.6.1. TOTAL=..... 36
4.6.2. CONTROL_ON=..... 36
4.6.3. TOTAL_ON=..... 36
4.6.4. FORMAT=..... 37

5. A REPEDIT report editor..... 39
5.1. A REPEDIT hívása és opciói..... 39
5.2. Az elemi adatterület leírása..... 41
5.2.1. Elemi változó létrehozása..... 42
5.2.2. Elemi változó módosítása..... 43
5.2.3. Elemi változó törlése..... 43
5.3. Formátumok megadása..... 44
5.3.1. Formátum kiválasztása..... 45
5.3.2. Pozícionálás a formátumon belül..... 46
5.3.3. Formátumelem létrehozása..... 46
5.3.3.1. Rendszerváltozó..... 47
5.3.3.2. Elemi változó..... 48
5.3.3.3. Elemi változó előző értéke..... 48
5.3.3.4. Összegző..... 48
5.3.3.5. Számláló..... 48

5.3.3.6. Átlag..... 48
5.3.3.7. Szövegkonstans..... 49
5.3.3.8. Hashmark parancs..... 49
5.3.3.8.1. New_Page..... 50
5.3.3.8.2. Total_Page..... 50
5.3.3.8.3. Total_Header..... 50
5.3.3.8.4. Detail_Header..... 51
5.3.3.8.5. Detail_Select..... 51
5.3.3.8.6. Detail_Print..... 51
5.3.3.8.7. Header_Select..... 51
5.3.3.8.8. Header_Print..... 51
5.3.3.8.9. Cr_Lf..... 52
5.3.3.8.10. Char_Size..... 52
5.3.3.8.11. Line_Spacing..... 52
5.3.3.8.12. Page_Length..... 53
5.3.3.8.13. Page_No..... 53
5.3.3.8.14. Printer_Control..... 53
5.3.4. Formátumelem módosítása..... 54
5.3.5. Formátumelem törlése..... 55
5.3.6. Formátumelem mozgatása..... 55
5.3.7. Formátumelem másolása..... 56
5.3.8. Formátum másolása..... 56
5.3.9. Formátum törlése..... 56
5.3.10. A formátum neve és leírása..... 56
5.3.11. A felhasználói C függvények..... 57
5.4. Opciók állítása..... 57
5.4.1. A listadefiníció leírása..... 58
5.4.2. A rendezettség leírása..... 59
5.4.3. Az output eszköz..... 59
5.4.4. Az output listafile könyvtára..... 59
5.4.5. Az output listafile neve..... 59
5.4.6. A hálózati server neve..... 59
5.4.7. A hálózati nyomtatás osztálya..... 60
5.4.8. Nyomtatási példányszám..... 60
5.4.9. A nyomtató típusa..... 60
5.4.10. A karakterméret..... 60
5.4.11. A nyomtatási sortávolság..... 61
5.4.12. A nyomtatási lap hossza..... 61
5.4.13. A dátum kiírási formátuma..... 62
5.4.14. Az output lista megnyitási módja..... 62
5.4.15. Az elemisor kiírásának letiltása..... 62
5.4.16. Teljes lapkitöltés kérése..... 63
5.4.17. Első lapemelés kérése..... 63
5.4.18. Utolsó lapemelés letiltása..... 63
5.5. Az editált lista lemezre írása..... 63
5.6. Headerfile generálása..... 64
5.7. C függvénytábla generálása..... 64
5.8. Egyéb parancsok..... 64
5.8.1. Leírások ellenőrzése..... 65
5.8.2. Nem használt elemi változók..... 65
5.8.3. Kiírási szélességek ellenőrzése..... 65
5.8.4. Az összegzési szintek beállítása..... 65

5.8.5. Kiszállás az operációs rendszerbe..... 66
5.9. Makrózás..... 66
5.10. Kiszállás a rendszerböl..... 67

6. Segédprogramok..... 69
6.1. A REPDOC listadokumenter..... 69
6.2. A REPTEST listatesztelö..... 70
6.3. A QTXTORTX, mint MISAM konvertáló..... 70
6.4. A QTXTORTX, mint MIQUEL konvertáló..... 71
6.5. Az FTXTORTX, MIFORM konvertáló..... 71

7. Listák készítése..... 73
7.1. Listázó keretprogram..... 75
7.2. Lekérdezésen alapuló listázás..... 75
7.3. Egy általános listázóprogram..... 75

8. A MIREP futtató rendszere..... 77
8.1. Elökészítö rutinok..... 77
8.1.1. r_read()..... 78
8.1.2. r_select()..... 78
8.2. Listajellemzök lekérdezése..... 79
8.2.1. r_get_charsize()..... 79
8.2.2. r_get_class()..... 79
8.2.3. r_get_copynum()..... 79
8.2.4. r_get_datemode()..... 80
8.2.5. r_get_device()..... 80
8.2.6. r_get_dscr()..... 80
8.2.7. r_get_formf()..... 81
8.2.8. r_get_frstff()..... 81
8.2.9. r_get_lastff()..... 81
8.2.10. r_get_linespacing()..... 82
8.2.11. r_get_lstfdir()..... 82
8.2.12. r_get_lstfname()..... 82
8.2.13. r_get_name()..... 83
8.2.14. r_get_openmode()..... 83
8.2.15. r_get_pagelength()..... 83
8.2.16. r_get_prntype()..... 84
8.2.17. r_get_server()..... 84
8.2.18. r_get_sorted()..... 84
8.3. Listajellemzök beállítása..... 85
8.3.1. r_redirect()..... 85
8.3.2. r_set_charsize()..... 85
8.3.3. r_set_class()..... 86
8.3.4. r_set_copynum()..... 86
8.3.5. r_set_datemode()..... 86
8.3.6. r_set_device()..... 87
8.3.7. r_set_dscr()..... 87
8.3.8. r_set_formf()..... 88
8.3.9. r_set_frstff()..... 88
8.3.10. r_set_lastff()..... 88
8.3.11. r_set_linespacing()..... 89
8.3.12. r_set_lstfdir()..... 89

8.3.13. r_set_lstfname() 89
8.3.14. r_set_name() 90
8.3.15. r_set_openmode() 90
8.3.16. r_set_pagelength() 90
8.3.17. r_set_prntype() 91
8.3.18. r_set_server() 91
8.3.19. r_set_sorted() 91
8.4. Végrehajtó rutinok 92
8.4.1. r_charsize() 92
8.4.2. r_close() 92
8.4.3. r_detail_header() 93
8.4.4. r_detail_print() 93
8.4.5. r_exec() 94
8.4.6. r_fill_page() 94
8.4.7. r_header_print() 95
8.4.8. r_init() 95
8.4.9. r_linespacing() 96
8.4.10. r_new_page() 96
8.4.11. r_printer_ctr() 96
8.4.12. r_total_header() 97
8.4.13. r_total_init() 97
8.4.14. r_total_page() 98
8.4.15. r_total_print() 98
8.5. Állapotkezelő rutinok 99
8.5.1. r_act_detail() 99
8.5.2. r_act_header() 99
8.5.3. r_act_total() 100
8.5.4. r_get_first() 100
8.5.5. r_get_jump() 101
8.5.6. r_get_printfp() 101
8.5.7. r_globprn_status() 101
8.5.8. r_set_first() 102
8.5.9. r_set_jump() 102
8.5.10. r_set_printfp() 102
8.5.11. r_status() 103
8.6. Adatlekérdező és módosító rutinok 103
8.6.1. r_atodat() 103
8.6.2. r_convdat() 104
8.6.3. r_d_all_valued() 104
8.6.4. r_d_buffer() 104
8.6.5. r_buflen() 105
8.6.6. r_d_get_value() 105
8.6.7. r_d_is_valued() 105
8.6.8. r_d_set_value() 106
8.6.9. r_d_valued() 106
8.6.10. r_get_count() 107
8.6.11. r_get_total() 107
8.6.12. r_o_get_value() 107
8.6.13. r_set_count() 108
8.6.14. r_set_total() 108
8.6.15. r_sysdate() 109
8.6.16. r_s_get_value() 109

8.6.17. r_s_set_value() ... 109
8.7. Az elemi változók kezelörutinjai ... 110
8.7.1. r_d_dec() ... 110
8.7.2. r_d_is_total() ... 110
8.7.3. r_d_length() ... 111
8.7.4. r_d_name() ... 111
8.7.5. r_d_number() ... 111
8.7.6. r_d_offset() ... 112
8.7.7. r_d_sernum() ... 112
8.7.8. r_d_size() ... 112
8.7.9. r_d_total_onoff() ... 113
8.7.10. r_d_type() ... 113
8.7.11. r_uniq_total() ... 113
8.8. Formátumkezelö rutinok ... 114
8.8.1. r_detail_name() ... 114
8.8.2. r_detail_number() ... 115
8.8.3. r_detail_onoff() ... 115
8.8.4. r_detail_select() ... 115
8.8.5. r_detail_sernum() ... 116
8.8.6. r_get_detprint() ... 116
8.8.7. r_header_name() ... 116
8.8.8. r_header_number() ... 117
8.8.9. r_header_onoff() ... 117
8.8.10. r_header_select() ... 117
8.8.11. r_header_sernum() ... 118
8.8.12. r_set_detprint() ... 118
8.8.13. r_searchcontrol() ... 118
8.8.14. r_total_name() ... 119
8.8.15. r_total_number() ... 119
8.8.16. r_total_only_more() ... 119
8.8.17. r_total_onoff() ... 120
8.8.18. r_total_sernum() ... 120
8.8.19. r_total_to_detail() ... 121
8.9. Egyéb rutinok ... 121
8.9.1. r_check() ... 121
8.9.2. r_cmd_need_arg() ... 121
8.9.3. r_control_check() ... 122
8.9.4. r_create_report() ... 122
8.9.5. r_cummulate() ... 122
8.9.6. r_endspool() ... 123
8.9.7. r_exec_func() ... 123
8.9.8. r_spool() ... 123
8.9.9. r_write() ... 124
8.10. Csatlakozási pontok rutinjai ... 124
8.10.1. r_set_funcs() ... 124
8.10.2. r_set_before() ... 125
8.10.3. r_set_after() ... 125
8.11. A felhasználói C függvények ... 126

9. A listázás környezete ... 127
9.1. Nyomtatók illesztése ... 127
9.1.1. Szüröfile-ok ... 127

9.1.2. PTCED szürőfile editor..... 128
9.1.2.1. A PTCED indítása..... 128
9.1.2.2. A standard nyomtatóvezérlések..... 131
9.1.2.3. USER vezérlő szekvenciák..... 133
9.1.2.4. Ékezetes betűk..... 133
9.1.2.5. Kvázigrafikus karakterek..... 134
9.1.2.6. Karakterkonverzió..... 135
9.1.2.7. A szürőfile tesztelése..... 136
9.2. Lokális nyomtató megadása..... 137
9.3. Hálózati nyomtató megadása..... 137
9.4. Lemezre küldött listák..... 138
9.5. Képernyőre küldött listák..... 138
9.6. Standard listakezelési funkciók..... 138

10. A disztributív anyag..... 141

F. Függelékek..... 143
F.A. File-leírások..... 144
F.A.1. ANYAG..... 144
F.A.2. TERMEK..... 145
F.A.3. TERM_ANY..... 145
F.B. Listaformátumok..... 147
F.B.1. ANYAG..... 147
F.B.1.1. Általános jellemzők..... 147
F.B.1.2. Fejléc, elemisor és összegsor formátumok.. 148
F.B.2. TERMEK..... 148
F.B.2.1. Általános jellemzők..... 148
F.B.2.2. Fejléc, elemisor és összegsor formátumok.. 149
F.B.3. TERM_ANY..... 149
F.B.3.1. Általános jellemzők..... 149
F.B.3.2. Fejléc, elemisor és összegsor formátumok.. 150
F.B.4. TESTTECH..... 151
F.B.4.1. Általános jellemzők..... 151
F.B.4.2. Fejléc, elemisor és összegsor formátumok.. 152
F.C. Lekérdezési file-kapcsolatok..... 154
F.C.1. TERM_ANY..... 154
F.C.1.1. MISAM leírók..... 154
F.C.1.2. Link struktúra..... 154
F.D. A TERM_ANY.RTX listaleíró file..... 156
F.E. A REPEDIT által generált TERM_ANY.H..... 158
F.F. A REPTEST által generált listakép..... 159
F.G. Forrásprogramok..... 161
F.G.1. TERM_ANY.C..... 161
F.G.2. TERM_QRY.C..... 168
F.G.3. LIST.C..... 172

1. Bevezetés

A MIREP egy olyan C nyelvhez kapcsolódó eszköztár, amely adatállományok listáinak gyors és rugalmas elöállítását teszi lehetövé.

A MIREP a MIxxx adatfeldolgozási eszköztár része, de illeszkedési felülete olyan, hogy könnyen használható más lekérdezö- és filerendszerekkel, illetve önállóan is.

A C nyelvhez való kapcsolódást tárgykód könyvtárak (Microsoft és Turbo C) biztosítják.

Jelen dokumentáció a felhasználói információkat tartalmazza, beleértve a programozáshoz szükséges ismereteket is.

A dokumentáció második fejezetében áttekintést adunk a listagenerálás helyéröl az adatfeldolgozási folyamatban, ezzel a MIREP helyét is kijelölve a MIxxx eszköztárban. Ugyanebben a fejezetben leírjuk a MIREP rendszer mögött álló koncepciót.

A listagenerálás témája számos sajátos fogalmat, kifejezést használ. Ezek jelentésére általános megállapodás nincs, az értelmezés nagyon függ a konkrét programrendszertöl. A harmadik fejezetben ezeket az alapfogalmakat vesszük sorra úgy, ahogyan azokat a MIREP értelmezi. Tárgyalásunkban az alapfogalmak együttese a MIREP alapelveiröl, müködéséröl és belsö szerkezetéröl is kellöen részletes információt ad.

A negyedik fejezet a listaleíró file szerkezetét ismerteti, vagyis azt a kulcsszavas szerkezetü "nyelvet" a írja le, amellyel a lista jellemzöit adjuk meg. A file szerkezetének megismerése azért is fontos, mert részletesen tájékoztat a listagenerátor lehetöségeiröl.

Más közelítésü tárgyalást ad az ötödik fejezet, amely a listaleíró file-ok kezelésére szolgáló segédprogramok közül a legfontosabbat, a REPEDIT nevü report editor használatát ismerteti. Az elözö fejezet nyelvleírásszerüen, ez pedig egy interaktív eszközön (értelmezön) keresztül közelíti ugyanazt a témát.

A hatodik fejezetben több, a listaleírással összefüggö segédprogram következik. Generatív, elemzö vagy dokumentatív eszközökröl van szó.

A hetedik fejezet a report végrehajtás lehetöségeit ismerteti. Ezen belül szó lesz az egyszerü listázási keretprogramokról, lekérdezésen alapuló listázóprogramokról és egy általános listázóprogramról.

A nyolcadik fejezetben részletesen tárgyaljuk a MIREP futtató rendszer moduljait és a felhasználó által megírandó, ún. felhasználói C függvényeket.

A kilencedik fejezet a listázás konkrét környezetével foglalkozik, többek között felöleli a nyomtatóillesztések és a hálózati nyomtatás témakörét.

A disztributív anyag tartalmát és az installálás menetét tartalmazza a tizedik fejezet.

A dokumentációban a jobb megértés kedvéért egy viszonylag egyszerű példa alapján szemléltetjük a MIREP környezet elemeit. A függelékek e példához tartozó file-ok, lekérdezések, listák, a MIREP editor által generált C modulok és a példaprogramok forrásainak listáit tartalmazzák.

Jelen dokumentáció kettős feladatot lát el. Egyrészt ismertetést és felhasználói útmutatást nyújt a kezdő felhasználók részére, másrészt egy teljeskörű referencia-kézikönyv szerepét is betölti. E második funkcióból fakad a viszonylag sok átfedés, így pl. az egyes listaelemek magyarázata szerepelhet a listaleíró nyelv ismertetésénél, a REPEDIT report editor megfelelő fejezeténél és a funkciót megvalósító futtatórendszerbeli rutin leírásánál is.

2. Áttekintés

Ennek a fejezetnek a célja a MIREP listagenerálás jelentőségének és alapelveinek a bemutatása.

2.1. A MIxxx adatfeldolgozási eszköztár

A hálózati adatfeldolgozási programrendszerek fejlesztésének egy sok szempontból előnyös módszere a hagyományos programnyelvekből indul ki. A leginkább használt nyelvek a COBOL és a C. Az IBM XT/AT gépeknél, a NOVELL hálózatoknál és a UNIX rendszereknél a C tűnik előnyösebbnek. Az általános célú programnyelvek előnye a "csomagolt" rendszerekkel (pl. a dBase, Dataflex) szemben a fejlesztés során a kötetlenség, a fejlesztett rendszereknél a rugalmasság, a gyorsaság, a jobb átvihetőség.

Ezek az előnyök más szempontból komoly hátrányokat jelentenek. Mivel a hagyományos programnyelvek az adatfeldolgozást megközelítően sem támogatják olyan mértékben, mint a csomagolt rendszerek, a kötetlenségért és a gyorsaságért komoly fejlesztési munkatöbblettel kell fizetni.

A hátrányokat csak úgy lehet kiküszöbölni, hogy az ismétlődő adatfeldolgozási funkciókra különböző prototípusokat dolgozunk ki, s ezeket építjük be programjainkba.

A prototípusok három válfaja:

- Önálló funkcióval rendelkező, különálló programként működő prototípusok, az ún. építőkövek ("bricks"-ek).

- Programnyelvből hívható tárgykönyvtárak.

- Forrásnyelvi prototípusok, többszörösen felhasználható modulok.

A MI Software által kifejlesztett MIxxx adatfeldolgozási eszközök az építőköves és a tárgykönyvtáras prototipizálással segítik a fejlesztést. Használatukkal a fejlesztési ráfordítás a csomagolt rendszerekkel azonos, megmarad ugyanakkor a programnyelvi közelítés összes előnye.

A MIFORM rendszer az adatbevitel, a MIREP a listakészítés, a MIQUEL a lekérdezés, a MISAM a file kezelés témakörét fedi le. A MIFORM rendszer a Magister Software terjesztésében MFORM néven kapható.

A fejlesztés keretrendszereként a C_ASE (C oriented Applied Soft-

ware Engineering; magyarul: C nyelven alapuló alkalmazott szoftvertechnológia) használható.

2.2. A listagenerátor alapelvei

A listagenerátor célja:

- A programozói munka egyszerűsítése.
- Annak biztosítása, hogy a felhasználó a listák formáját és adattartalmát saját ízlése szerint változtathassa.
- A listák tervezésének megkönnyítése.
- A listák dokumentálásának támogatása.

Ennek érdekében a MIREP a listákat egy leíró file alapján kezeli. A leíró file viszonylag könnyen, generátorokkal vagy a REPEDIT nevű report editorral hozható létre. A programok a MIREP futtatási rendszer moduljain keresztül a leíró file-t használják. A táblázatos közelítésmód miatt a leíró file változtatása általában nem igényel programváltoztatást.

A leíró file kulcsszavas, karakteres formájú, közönséges szövegszerkesztővel is kezelhető.

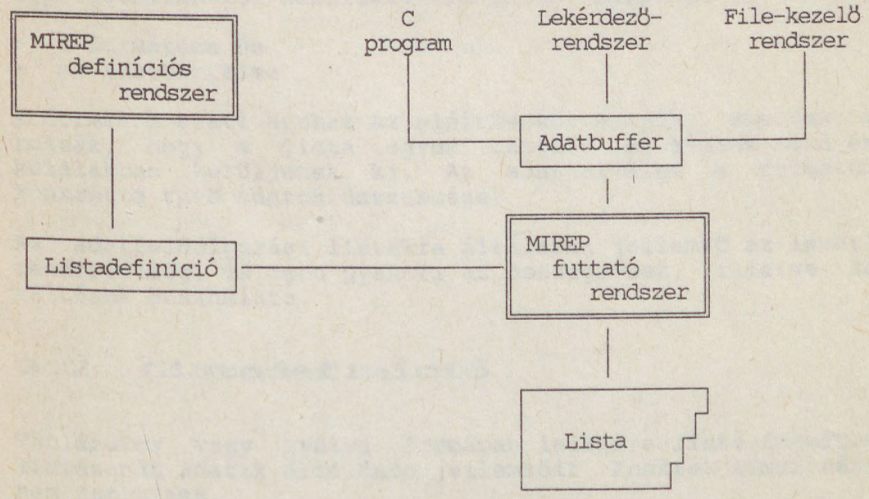
A MIREP további jellemzői:

- A futtatórendszernek ciklikusan átadott elemi adatbuffer alapján automatikusan elkészíti a kívánt listát.
- Automatikusan kezeli
  - o a lapváltást és fejlécezést,
  - o a kontrollváltásokat,
  - o az összegzéseket,
  - o a számlálásokat,
  - o az átlagképzéseket.
- Dinamikusan váltható fejléceket enged meg.
- Tetszőleges számú, dinamikusan váltható részletező sor, ill. sorok kezelését teszi lehetővé.
- A kontrollfokozat nélküli végösszegzésen kívül tetszőleges számú kontrollváltás, illetve részösszegzés automatikus végrehajtását támogatja.
- Kiküszöböli a listázás formai részének programozását.

- Lehetőséget ad a rugalmas nyomtatóillesztésekre, munkaállomáshoz tartozó vagy hálózati nyomtatókon való listázásokra.

- Nemcsak a MIxxx eszközökkel együtt, hanem más file és lekérdező rendszerekkel együtt is használható. Ezt az teszi lehetővé, hogy a listakezelés adatkapcsolata a környezettel szigorúan az "elemi adatbuffer"-nek nevezett adatterületre szorítkozik, s mindegy, hogy az elemi adatbuffert milyen egyéb szoftver állítja össze.

A MIREP koncepcióját az 1. ábra szemlélteti:



1. ábra

3. Alapfogalmak és definíciók

A dokumentációban gyakran használunk olyan fogalmakat, amelyeknek a jelentése nem feltétlenül egyértelmű. A következő fejezetben közölt kis szótár ezen fogalmakat fejti ki bővebben.

3.1. Lista

Egy adathalmazról készített kimutatás. Jellemző rá:

- a formátuma és
- az adattartalma.

A formátum alatt azokat az előírásokat értjük, amelyek meghatározzák, hogy a lista egyes lapjain az adatok hová és milyen külalakban kerüljenek ki. Az adattartalom a formátumkeretet konkréttá tevő adatok összessége.

Az adatfeldolgozási listákra általában jellemző az ismétlődés, a rendezettség, és igen gyakori az összegzések, illetve kontrollváltások használata.

3.2. Listadefiníció

Táblázatos vagy nyelvi formában leírja a lista formátumát és a listázandó adatok általános jellemzőit. Konkrét kimutatási adatot nem tartalmaz.

A definíció áll:

- listázási opciókból,
- adatleírásból,
- fejlécsor leírásokból,
- elemisor leírásokból és
- összegsor leírásokból.

3.3. Listaleíró file

A listadefiníciót egy kötelezően .RTX kiterjesztésű, ún. listaleíró file-ban tároljuk. A listaleíró file a REPEDIT report editorral, segédprogramokkal és a futtató rendszer beolvasó függvényével kezelhető.

3.4. Listagenerátor

Olyan szoftver, amely a listák előállítását automatizálja, illetve segíti. A MIREP típusú listagenerátorok programfüggetlen listadefiníciókkal dolgoznak, és lényegében két részből állnak: a lista definíciót létrehozó és az azt értelmező részből. Az értelmező rész lehet önálló program, vagy programnyelvbe ágyazott modulgyűjtemény.

A MIREP-ben létezik ugyan egy önálló végrehajtó program, de ez csak egyszerű listázásokra képes, és erősen file-, illetve lekérdezőrendszer függő. A listák végrehajtatását általában olyan C nyelvű keretprogramok végzik, amelyek a MIREP tárgykönyvtár moduljait használják.

Egy listadefiníció értelmezése általában a 2. ábrán látható feldolgozási ciklus szerint történik.

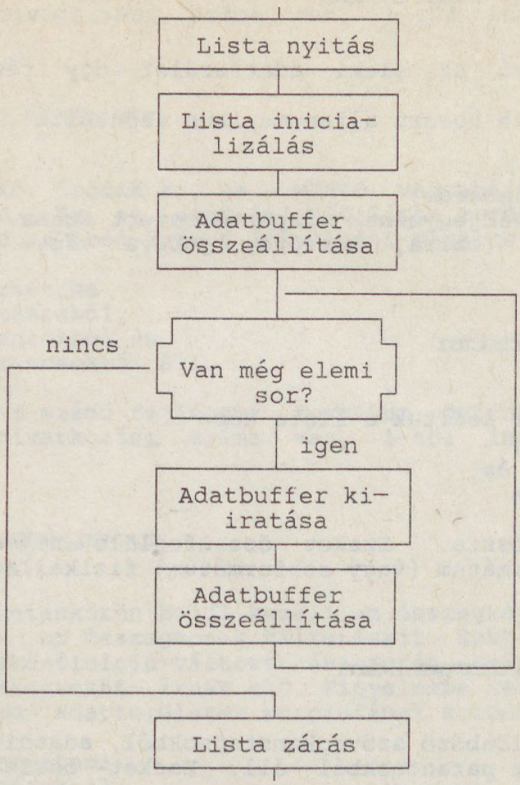
A listadefiníció értelmezését a listázó programkerettel, illetve a listaértelmezési folyamat megadott helyein csatlakozó felhasználói C függvények révén befolyásolni lehet.

3.5. Elemi adatterület

A 2. ábrán bemutatott diagramból látható, hogy a listázás külvilággal való kapcsolódási része lényegében egy adatbuffer kiíratásának sorozatos kérése. Az összes többi listázási részlet, a fejlécezéstől kezdve az összegképzésig és kiírásokig a MIREP rutinok dolga. Ezt az adatbuffert elemi adatterületnek nevezzük.

Az elemi adatterület szerkezete a listadefinícióban van leírva. Itt kell megadni a benne szereplő változók nevét, típusát, hosszát, kiíratási formáját stb.

A listázási forma megadásakor csak az elemi adatterület változóira, illetve a belőlük képzett változókra (pl. total, azaz összeg) lehet hivatkozni.



2. ábra

Az elemi adatterületen szereplő adatok eredetére a MIREP semmi megkötést nem tesz. Ez a tény teszi file- és lekérdezőrendszertől függetlenné. Az adatterület feltöltéséről a keretprogramnak kell gondoskodnia. Az adatok általában két csoportba sorolhatók. Az egyikbe azok tartoznak, amelyek eredete közvetlenül az adatállomány – ezek az elsődleges elemi adatok – a másik csoportot az elsődleges adatokból számított adatok alkotják. A MIREP jelenlegi változatában a számítást a keretprogramnak kell végeznie, a listaleíró nyelv kalkulatív parancsokat nem tartalmaz.

3.6. Elemi változó

Az elemi változó az elemi adatterület egy tétele, amelyre jellemző:
- a neve (max. 15 hosszú alfanumerikus azonosító),
- a leírása,
- a típusa,
- a kiírási szélessége,
- a decimális számjegyeinek alapértelmezett száma,
- az igazítása (balra, középre, jobbra vagy jobbra vezető '0'-kal).

3.7. Formátum

Mint korábban már leírtuk a lista képe
- az elemisorok,
- a fejlécsorok és
- az összegsorok

képéből áll össze. Ezeket összefoglaló néven formátumnak nevezzük. Egy formátum (vagy sorformátum) fizikailag nem csak egy sor lehet.

3.8. Formátumelem

Egy formátum különböző szövegkonstansokból, adathivatkozásokból, illetve speciális parancsokból áll. Ezeket összefoglaló néven formátumelemeknek nevezzük.

3.9. Elemisor

Az elemi adatterület kiíratott formája az elemisor. Ez fizikailag nem korlátozódik egy listasorra, bár leggyakrabban valóban egyetlen sort ad. Az elemisor formátumát, valamint azt, hogy az elemi adatterületről mely változók és milyen sorrendben kerüljenek be a kiíratásba, teljes egészében a lista definíciója határozza meg, s így tetszés szerint változtatható.

Az elemisor formátuma
- adathivatkozásokból,
- szövegkonstansokból és
- hashmark (#) parancsokból áll.

Tetszés szerinti számú elemisor formátum definiálható. Minden elemisornak hivatkozási száma van, 1-től induló folyamatos számozással.

3.10. Fejlécsor

A fejlécsor akkor íródik ki, ha a MIREP végrehajtó mechanizmusa új lapot kezd. Ez az eset áll elő lista kezdetén, lap végére éréskor, vagy direktben kapott lapváltási parancs esetén.

A fejlécsor formátuma
adathivatkozásokból,
szövegkonstansokból és
hashmark parancsokból áll.

Tetszés szerinti számú fejlécsor formátum definiálható. Minden fejlécsornak hivatkozási száma van, 1-től induló folyamatos számozással.

3.11. Összegsor

A MIREP saját hatáskörön belül kezeli az összegképzést, a kontrollváltást és az összegsorok kiíratásait. Ebből eredően a felhasználó a listadefiníció változtatása révén tetszésének megfelelő összegfokozatokat írhat elő. Figyelembe kell vennie természetesen az elemi adatterületek sorozatának a rendezettségét!

Az összegsor formátuma
adathivatkozásokból,
szövegkonstansokból és
hashmark parancsokból áll.

Ezen kívül minden összegsorhoz meg kell adni, hogy melyik elemi változó értékváltozása lesz az összegsor kiíratásának a kezdeményezője (kontrollváltozó), illetve meg kell jelölni azon változókat, amelyekre összegzést kérünk.

Tetszés szerinti számú összegsor definiálható, 1-től induló folyamatos számozással. Az összegsorok sorszámai lényegesek: az i-edik összegzés kiírása előtt előbb listázásra kerül az (i+1)-edik összegsor, ha a definícióban szerepel. A speciális 0-dik összegsor a kontrollfokozat nélküli "mindösszesen" (Grand Total) összegsor.

Az összegsor sorszámát szintnek is nevezzük, mivel a nem 0 szintű összegsorok egyben kontrollváltást is jelentenek és a sorszám nem más, mint a kontrollváltás szintje. A fentiek alapján természetesen a legnagyobb sorszámú (szintű) összegzés jelenti a legbelső kontrollváltást.

3.12. Szövegkonstans

A formátumok lényeges elemei a fix szövegkonstansok. A fejléc-sorok általában sok szövegkonstanst, az elemisorok kevesebbet használnak.

A szövegkonstansok változatlan formában jelennek meg a listán. Ékezetes betűk használata a CWI standard szerint lehetséges. A nem ASCII karakterek használata megengedett, de nyomtatásuk az aktuális nyomtató típusától függ.

3.13. Adathivatkozás

A formátumok leírásakor hivatkozható adatok:
- elemi változó értéke,
- az elemi változó előző értéke,
- az elemi változó megadott összegzési szintű összege,
- az elemi változó megadott összegzési szintű átlaga,
- az adott összegző fokozatban eddig kiírt elemisorok száma,
- a MIREP rendszerváltozók.

Listázáskor az adathivatkozások helyére a konkrét adatok kerülnek, az elemi adatterület leírásakor, vagy az összegzés definiálásakor megadott formában.

3.14. Rendszerváltozó

A MIREP a következő változókat kezeli automatikusan:

- lapszám,
- rendszerdátum,
- lapon belüli sorszám,
- lista elejétől számított sorszám.

Ezeket rendszerváltozóknak nevezzük, mivel értékük meghatározása a MIREP rendszer hatáskörébe tartozik.

3.15. Hashmark parancs

A listasorok leírásai a passzív, kiíratási formát meghatározó információk mellett aktív, a lista végrehajtását befolyásoló parancsokat is tartalmazhatnak. Pl. válthatják az aktuális fejléc számot, lapot emelhetnek. Tekintettel arra, hogy ezeket a parancsokat mind a REPEDIT report editor, mind pedig a REPDOC listadokumenter a hashmark (#) karakterrel jelöli, gyüjtö nevük: hashmark parancs.

A hashmark parancsok a következök:

New_Page	lapváltás,
Total_Page	feltételes lapváltás összegsor után,
Total_Header	feltételes fejlécsor kiíratás,
Detail_Header	feltételes fejlécsor kiíratás,
Detail_Select	elemisor kiválasztása,
Detail_Print	elemisor explicit kiíratása,
Header_Select	fejlécsor kiválasztása,
Header_Print	fejlécsor explicit kiíratása,
Cr_Lf	üres sor beillesztése a formátumba,
Char_Size	karakterméret állítása,
Line_Spacing	sortávolság állítása,
Page_Length	lapméret állítása,
Page_No	lapszám állítása,
Printer_Control	nyomtatófüggö vezérlés.

3.16. Lokális nyomtató

A MIxxx adatfeldolgozási rendszereknél feltételezzük, hogy minden munkaállomáshoz tartozhat saját nyomtató. Jellemzö rájuk a típus. Elképzelhetö, hogy nem egységes típusúak.

A lokális nyomtató fogalma azért fontos, mert például az alkalmazói rendszerekben a naplózásokat mindig kötelezöen a lokális nyomatókon kell nyomtatni. Ha nincs a munkaállomáson lokális nyomtató, akkor a naplók lemezes file-okban örzödnek meg.

3.17. Hálózati nyomtató

A nagy listákat célszerü hálózati nyomtatón kiíratni. Egy alkalmazáshoz egy központi nyomtatót rendelhetünk hozzá. A központi nyomtatót jellemzi:

- típusa,
- a server azonosító neve, amihez a nyomtató rendelve van,

- a spool osztály (class, queue szám),
- a példányszám.

3.18. Szűrőfile

Az egyes nyomtatókon az ékezetes karakterek és az írásmódot vezérlő karaktersorozatok eltérőek. Tekintettel arra, hogy hálózat esetében a programozó nem tudja előre, milyen típusú nyomtatóval rendelkező munkaállomáson nyomtatják majd listáit vagy naplóit, kénytelen programjaival egy általános listaformát létrehozni. A kiíratáskor az általános formát át kell vezetni egy szűrőfile-on, ami tulajdonképpen egy karaktertranszformációs tábla. A szűrőfile lényegében definiálja a nyomtatót a kiíratás céljára.

A szűrőfile-ok a PTCED nevű szűrőfile editorral módosíthatók, illetve hozhatók létre.

A MIREP disztributív anyaga tartalmazza a leggyakoribb nyomtatókhoz definiált szűrőfile-okat.

3.19. Nyomtató definíció

Kötelező szabály, hogy a nyomtató neve és a szűrőfile alapneve megegyezik. Egy nyomtatót akkor tekintünk definiáltnak, ha vele egyező nevű szűrőfile létezik. A szűrőfile létrehozásához a PTCED editor használható.

3.20. Illeszkedési pont

A lista kiíratási folyamatának bizonyos pontjain lehetőség van felhasználói C függvények hívására. Ezeket a pontokat illeszkedési pontoknak nevezzük. A lista végrehajtásának minden lényeges mozzanatához tartozhat illeszkedési pont, mégpedig az egyes formátumok kiírása (végrehajtása) előtt és után.

4. A listaleíró nyelv

A lista leírása kulcsszavas szerkezetü. Általános formája:

<kulcsszó>=<argumentum>

Pl.
DETAILFIELDS=20

A leírási parancsokat csoportokra oszthatjuk:

- Opciókat kezelö parancsok.
- Az elemi adatterületet leíró parancsok.
- Elemisor formátumát leíró parancsok.
- Fejlécsor formátumát leíró parancsok.
- Összegsor formátumát leíró parancsok.

A * jellel kezdett sorok magyarázat soroknak minösülnek.

A D. függelék a példaként használt TERM_ANY listaleíró file-t tartalmazza.

4.1. Opciók állítása

A lista általános opcióit a következö parancsokkal határozhatjuk meg.

4.1.1. REPORT=

Funkciója:
A listadefiníció leírásának megadása.

Formája:
REPORT=<report-név>,<leírás>

A listadefiníció leírása csak dokumentációs célt szolgál.

4.1.2. SORTED=

Funkciója:
Az elemi adatok kívánt rendezettségének leírása.

Formája:
SORTED=<leírás>

A rendezettségi információ csak dokumentációs célt szolgál.

4.1.3. DEVICE=

Funkciója:
A nyomtatandó lista eszközének meghatározása.

Formája:
DEVICE=<eszköz>

Az eszköz lehet:
C (Consol) képernyő,
D (Diskfile) lemezen tárolt file,
L (Localprinter) lokális nyomtató,
G (Globalprinter) hálózati nyomtató.

Alapértélmezés szerint: 'L' (Localprinter).

4.1.4. LSTFDIR=

Funkciója:
Ha a lista lemezen tárolt file-ba készül, akkor a file könyvtárának megadása.

Formája:
LSTFDIR=<könyvtárnév>

Alapértelmezés szerint: az aktuális könyvtár.

4.1.5. LSTFNAME=

Funkciója:
Ha a lista lemezen tárolt file-ba készül, akkor a file nevének megadása.

Formája:
LSTFNAME=<filenév>

Alapértelmezés szerint: <report-név>.LST

ahol a <report-név> a listaleíró file neve, könyvtár és kiterjesztés nélkül.

4.1.6. SERVER=

Funkciója:
A globális nyomtató server-ének kijelölése.

Formája:
SERVERNAME=<server_neve>

Ha a kiíratás nem globális nyomtatóra megy, az utasítás hatástalan.

4.1.7. CLASS=

Funkciója:
A hálózati nyomtatás osztályának ("queue" számának) kijelölése.

Formája:
CLASS=<queue_szám>

Ha a kiíratás nem globális nyomtatóra megy, az utasítás hatástalan.

4.1.8. COPYNUM=

Funkciója:
A globális nyomtatón kért példányszám megadása.

Formája:
COPYNUM=<példányszám>

Alapértelmezés szerint: 1 példány.
Ha a kiíratás nem globális nyomtatóra megy, az utasítás hatástalan.

4.1.9. PRNTYPE=

Funkciója:
A nyomtató típusának meghatározása.

Formája:
PRNTYPE=<nyomtató_típusa>

A nyomtató típusa egy legfeljebb 8 karakterből álló azonosító. A nyomtató akkor definiált, ha a nyomtató típusának megfelelő szűrőfile rendelkezésre áll az aktuális, illetve a

SET PTC=<ptc-könyvtár>

DOS paranccsal beállított könyvtárban.

A MIREP disztributív anyaga tartalmazza a leggyakoribb nyomtatókhoz definiált szűrőfile-okat.

4.1.10. CHARSIZE=

Funkciója:
A karakterméret beállítása a listázás megkezdése előtt.

Formája:
CHARSIZE=<karakterméret>

A karakterméret a következők valamelyike lehet:
Normal,
Elite,
Condensed.

A következő ábra az egyes karakterméreteknél a hüvelykenkénti (inch-enkénti) karakterszámot, valamint a keskeny és széles nyomtatópapírnál az egy sorba írható karakterek számát szemlélteti:

	Karakter/ inch	Papírméret,kar/sor	
		8 inch	13,2 inch
Normal	10	80	132
Elite	12	96	158
Condensed	17	137	225

A karakterméret nagymértékben nyomtatófüggő, ezért ennek változtatása feltételezi a nyomtató típusának meghatározását, és az ehhez kapcsolódó szűrőfile "standard nyomtatóvezérlések" megfelelő elemeinek helyes beállítását.

Az alapértelmezés szerint: Normal.

4.1.11. LINESPACING=

Funkciója:
A sortáv beállítása a listázás megkezdése előtt.

Formája:
LINESPACING=<sortáv>

A sortáv azonosító a következők valamelyike lehet:
N Normal,
H Half spacing,
W Whitout spacing.

A következő ábra az egyes sortávokat, valamint a 12 hüvelyk hosszú nyomtatópapírnál az egy lapra írható sorok számát szemlélteti:

		Sortáv	Sor/lap
Normal	(normális sorköz)	1/6	72
Half spacing	(fél sorköz)	1/8	96
Whitout spacing	(sorköz nélkül)	7/72	123

A sortáv vezérlése nagymértékben nyomtatófüggő, ezért ennek változtatása feltételezi a nyomtató típusának meghatározását, és az ehhez kapcsolódó szűrőfile "standard nyomtatóvezérlések" megfelelő elemeinek helyes beállítását.

Az alapértelmezés szerint: Normal.

4.1.12. PAGELENGTH=

Funkciója:
A lista egy lapjára kerülő fizikai sorok számának meghatározása.

Formája:
PAGELENGTH=<a sorok száma>

Az alapértelmezés szerint: 65.

4.1.13. DATEMODE=

Funkciója:
A dátum típusú mezők kiíratási módjának meghatározása.

Formája:
DATEMODE=<formátum>

A formátum lehetséges értékei:
E English,
H Hungarian,
G German.

A megfelelő dátumformátumok rendre (E: év, H: hó, N: nap):
HH/NN/EE (angol)
EE/HH/NN (magyar)
NN/HH/EE (német)

Alapértelmezés szerint: Hungarian.

4.1.14. OPENMODE=

Funkciója:
A listafile nyitási módjának meghatározása.

Formája:
OPENMODE=<nyitási mód>

A nyitási mód lehet
W Write vagy
A Append.

"Write" módban az output listafile felülíródik, "Append" módban az új lista a már meglévő végére kerül. Az opció hatástalan, ha a lista nem file-ba kerül. Append módot kell használni lemezre írt naplók esetén.

Alapértelmezés szerint: "Write".

4.1.15. DETPRINT=

Funkciója:
Az elemisor kiíratásának ki-, ill. bekapcsolása.

Formája:
DETPRINT=OFF

Alapértalmezés szerint: ON

4.1.16. FORMFEED=

Funkciója:
A lapok elejére kerülő FormFeed (lapemelés) karakter helyett a lapok teljes kitöltésének kérése.

Formája:
FORMFEED=OFF

Alapértelmezés szerint: ON, ami azt jelenti, hogy új lap kezdete előtt a rendszer nem tölti ki az aktuális lap maradék sorait üres sorokkal, hanem a FormFeed karakter segítségével a következő lap tetejére ugrik. Az opciót akkor kell OFF értékre állítani, ha ez a megoldás valamilyen oknál fogva nem megfelelő, és a lapokon lévő sorok számának azonosnak kell lennie.

4.1.17. FRSTFF=

Funkciója:
A legelső lapra a FormFeed (lapemelés) karakter kiíratásának kérése.
Formája:
FRSTFF=ON

Alapértelmezés szerint: OFF

Ha az opciót ON-ra állítjuk, akkor a lista legelső fejléce új lapra kerül. Az opció beállítására akkor lehet szükség, ha nem biztos, hogy a nyomtató a listázás kezdetekor lap tetején áll.

4.1.18. LASTFF=

Funkciója:
Az utolsó lap után a FormFeed (lapemelés) karakter kiíratásának letiltása.

Formája:
LASTFF=OFF

Alapértelmezés szerint: ON

Ha az opciót OFF-ra állítjuk, akkor a lista generálása után a nyomtató nem áll a következő lap elejére.

4.2. Az elemi adatterület leírása

A következőkben az elemi adatterület megadási formáját ismertetjük.

4.2.1. DETAILFIELDS=

Funkciója:
Megadja, hogy az elemi adatterület hány változóból áll.

Formája:
DETAILFIELDS=<az elemi adatterület mezőinek a száma>

4.2.2. FIELD=

Funkciója:
Egy elemi változó leírása.

Formája:
FIELD=n,d,t,w,d,j

ahol
n (Name) a mező neve, amely maximum 15 karakteres alfanumerikus azonosító,
d (Description) a mező leírása,
t (Type) a mező típusa, amely lehet:
B: Byte (1 byte-os előjeles egész),
S: unSigned byte (1 byte-os előjel nélküli egész),
I: Int (2 byte-os előjeles egész),
U: Unsigned int (2 byte-os előjel nélküli egész),
L: Long (4 byte-os előjeles egész),
M: dátuM (dátum belső formátumú, 4 byte-on),
F: Float (4 byte-os lebegőpontos érték),
D: Double (8 byte-os lebegőpontos érték),
T: Text (szöveg),
w (Width) a mező kiírási szélessége,

d (Dec) a mező decimális számjegyeinek száma (csak lebegőpontos értéknél vesszük figyelembe),
j (Justify) kiírási szélességen belüli igazítás, amely lehet:
L (Left) balra,
M (Middle) középre,
R (Right) jobbra, vagy
Z (ZeroRight) jobbra vezető '0' karakterekkel.

Az elemi változó utolsó három jellemzője a mező alapértelmezés szerinti kiírási formáját adja meg. A formátumokban való hivatkozáskor e jellemzők felülbírálhatók.

4.3. Formátumelemek

A leíró file-ban szereplő három fajta formátum leírása (elemisor formátuma, fejlécsor formátuma és összegsor formátuma) azonos felépítésű:

- először egy bevezető sor azonosítja a formátumot és meghatározza annak általános jellemzőit,
- majd annyi formátumelemet leíró sor következik, ahány elemből áll össze a formátum.

Formálisan:
<bevezető_sor_kulcsszó>=s,n,b,a,d
<formátum_elem_kulcsszó>=<formátum_elem>
<formátum_elem_kulcsszó>=<formátum_elem>
<formátum_elem_kulcsszó>=<formátum_elem>
.
.
.

ahol
s (Serial number) a formátum sorszáma,
n (Name) a formátumot azonosító név (max. 8 hosszú, alfanumerikus azonosító),
b (Before C function) a formátum kiírása előtt meghívandó felhasználói C függvény neve (max. 15 hosszú alfanumerikus név),
a (After C function) a formátum kiírása után meghívandó felhasználói C függvény neve (max. 15 hosszú alfanumerikus név),
d (Description) a formátum leírása.

A <formátum_elem> a következők valamelyike:
Rendszerváltozó (System),
Elemi változó (Detail),
Elemi változó előző értéke (Old detail),
Összegző (totaL),
Számláló (Count),
Átlag (Average),

Formátum szöveg (Text),
Hashmark parancs (Hashmark command).

4.3.1. Rendszerváltozó

A négy rendszerváltozó a következő formában hivatkozható:

<formátum_elem_kulcsszó>=S,n,l,r,c,w,d,j

ahol
S (System) jelöli, hogy a formátumelem egy rendszerváltozó,
n (Name) név, a rendszerváltozó neve, ami lehet:
date (rendszerdátum),
pageno (aktuális lapszám),
lineno (aktuális sorszám),
linetotal (a lista kezdetétől számított sorszámláló érték),
l (Level) szint, jelen változatban kötelezően 0,
r (Row) sor, a formátumelem sora 0-tól számítva,
c (Column) oszlop, a formátumelem oszlopa 0-tól számítva,
w (Width) szélesség, a formátumelem kiírási szélessége,
d (Dec) decimális számjegyek száma, rendszerváltozóknál 0,
j (Justify) kiírási szélességen belüli igazítás, amely lehet:
L (Left) balra,
M (Middle) középre,
R (Right) jobbra, vagy
Z (ZeroRight) jobbra vezető '0' karakterekkel.

4.3.2. Elemi változó

Az elemi változóra való hivatkozás formája:

<formátum_elem_kulcsszó>=D,n,l,r,c,w,d,j

ahol
D (Detail) jelöli, hogy a formátumelem az elemi adatbuffer egyik változója,
n (Name) név, a változó neve,
l (Level) szint, jelen változatban kötelezően 0,
r (Row) sor, a formátumelem sora 0-tól számítva,
c (Column) oszlop, a formátumelem oszlopa 0-tól számítva,
w (Width) szélesség, a formátumelem kiírási szélessége,
d (Dec) decimális számjegyek száma lebegőpontos adatnál,

j (Justify) kiírási szélességen belüli igazítás, amely lehet:
  L (Left) balra,
  M (Middle) középre,
  R (Right) jobbra, vagy
  Z (ZeroRight) jobbra vezető '0' karakterekkel.

4.3.3. Elemi változó előző értéke

Egy elemi változó előző értékére való hivatkozás formája:

<formátum_elem_kulcsszó>=O,n,l,r,c,w,d,j

ahol
O (Old detail) jelöli, hogy a formátumelem az előző elemi adatbuffer egyik változója,
n (Name) név, a változó neve,
l (Level) szint, jelen változatban kötelezően 0,
r (Row) sor, a formátumelem sora 0-tól számítva,
c (Column) oszlop, a formátumelem oszlopa 0-tól számítva,
w (Width) szélesség, a formátumelem kiírási szélessége,
d (Dec) decimális számjegyek száma lebegőpontos adatnál,
j (Justify) kiírási szélességen belüli igazítás, amely lehet:
  L (Left) balra,
  M (Middle) középre,
  R (Right) jobbra, vagy
  Z (ZeroRight) jobbra vezető '0' karakterekkel.

4.3.4. Összegző

Az összegzett adatok hivatkozása:

<formátum_elem_kulcsszó>=L,n,l,r,c,w,d,j

ahol
L (totaL) jelöli, hogy a formátumelem egy összegző,
n (Name) név, annak az elemi változónak a neve, amelyiket összegezni kell,
l (Level) szint, az összegző fokozat szintje,
r (Row) sor, a formátumelem sora 0-tól számítva,
c (Column) oszlop, a formátumelem oszlopa 0-tól számítva,
w (Width) szélesség, a formátumelem kiírási szélessége,
d (Dec) decimális számjegyek száma,

j (Justify) kiírási szélességen belüli igazítás, amely lehet:
L (Left) balra,
M (Middle) középre,
R (Right) jobbra, vagy
Z (ZeroRight) jobbra vezető '0' karakterekkel.

4.3.5. Számláló

Az elemi adatsor számláló hivatkozása:

<formátum_elem_kulcsszó>=C,,l,r,c,w,d,j

ahol
C (Count) jelöli, hogy a formátumelem egy számláló,
l (Level) szint, az összegző fokozat szintje,
r (Row) sor, a formátumelem sora 0-tól számítva,
c (Column) oszlop, a formátumelem oszlopa 0-tól számítva,
w (Width) szélesség, a formátumelem kiírási szélessége,
d (Dec) decimális számjegyek száma,
j (Justify) kiírási szélességen belüli igazítás, amely lehet:
L (Left) balra,
M (Middle) középre,
R (Right) jobbra, vagy
Z (ZeroRight) jobbra vezető '0' karakterekkel.

Az 'l' (Level, szint) értékének lehetséges értékei:

-1 a legbelső összegző fokozaton belül listázott elemisorok száma,
0 a listázott elemisorok száma összesen,
1 az 'l'-edik szintű kontrollváltások száma.

A számláló használatát jól mutatja az F. függelékben található teszt lista, amely a B. függelék 4. pontjában dokumentált TESTTECH listadefiníció alapján készült, a REPTEST segédprogram segítségével.

4.3.6. Átlag

Az átlagolt adatok hivatkozása:

<formátum_elem_kulcsszó>=A,n,l,r,c,w,d,j

ahol
A (Average) jelöli, hogy a formátumelem egy átlag,
n (Name) név, annak az elemi bufferbeli változónak a neve, amelyiket átlagolni kell,

l (Level) szint, az összegző fokozat szintje,
r (Row) sor, a formátumelem sora 0-tól számítva,
c (Column) oszlop, a formátumelem oszlopa 0-tól számítva,
w (Width) szélesség, a formátumelem kiírási szélessége,
d (Dec) decimális számjegyek száma,
j (Justify) kiírási szélességen belüli igazítás, amely lehet:
    L (Left) balra,
    M (Middle) középre,
    R (Right) jobbra, vagy
    Z (ZeroRight) jobbra vezető '0' karakterekkel.

4.3.7. Szövegkonstans

A formátumban lévő szöveg megadása:

<formátum_elem_kulcsszó>=T,r,c,txt

ahol
T (Text) jelöli, hogy a formátumelem egy szövegkonstans,
r (Row) sor, a formátumelem sora 0-tól számítva,
c (Column) oszlop, a formátumelem oszlopa 0-tól számítva,
txt a kiírandó szöveg.

Ékezetes betűk használata a CWI standard szerint lehetséges. A nem ASCII karakterek használata megengedett, de nyomtatásuk az aktuális nyomtató típusától függ.

4.3.8. Hashmark parancs

A hashmark parancsok az adott formátum valamilyen aktív funkcióját írják le. A formátum értelmezése során a
    Cr_Lf,
    Char_Size,
    Line_Spacing,
    Page_Length,
    Page_No,
    Printer_Control
parancsok azonnal végrehajtásra kerülnek, a többi hashmark parancs viszont csak a formátum értelmezésének befejezése után.

Az hashmark parancsok megadásának általános formája:

<formátum_elem_kulcsszó>=H,n,a,r,c

ahol
H (Hashmark) jelöli, hogy a formátumelem egy hashmark parancs,
n (Name) név, a hashmark parancs neve,

a (Argument) paraméter, hashmark parancs paramétere,
r (Row) sor, a formátumelem sora 0-tól számítva,
c (Column) oszlop, a formátumelem oszlopa 0-tól számítva,

Az 'a' paraméter nem minden hashmark parancsnál kötelező, ahol nem kell, ott 0-t kell megadni.

4.3.8.1. New_Page

Funkciója:
Lapdobás kiváltása.

Formája:
<formátum_elem_kulcsszó>=H,New_Page,0,r,c

Paramétere: nincs.

4.3.8.2. Total_Page

Funkciója:
Lapdobás kiváltása, de összegsoroknál csak akkor, ha nem követi alacsonyabb szintű összegsor kiírása.

Formája:
<formátum_elem_kulcsszó>=H,Total_Page,0,r,c

Paramétere: nincs.

4.3.8.3. Total_Header

Funkciója:
Kiíratja a megadott sorszámú fejlécsort, de összegsoroknál csak akkor, ha nem követi alacsonyabb szintű összegsor kiírása. Használata akkor indokolt, ha az összegsor után lapváltás nélkül folytatódik a lista, és a következő elemisorok előtt fejlécet akarunk íratni.

Formája:
<formátum_elem_kulcsszó>=H,Total_Header,a,r,c

Paramétere: a kívánt fejlécsor sorszáma.

4.3.8.4. Detail_Header

Funkciója:
Kiíratja a megadott sorszámú fejlécsort, de csak akkor, ha az előző pontban leírt "Total_Header" parancs végrehajtására nem volt szükség.

Formája:
<formátum_elem_kulcsszó>=H,Detail_Header,a,r,c

Paramétere: a kívánt fejlécsor sorszáma.

4.3.8.5. Detail_Select

Funkciója:
Az aktuális elemisort átváltja a megadottra.

Formája:
<formátum_elem_kulcsszó>=H,Detail_Select,a,r,c

Paramétere: a kívánt elemisor formátum sorszáma.

4.3.8.6. Detail_Print

Funkciója:
Kiíratja a megadott elemisort.

Formája:
<formátum_elem_kulcsszó>=H,Detail_Print,a,r,c

Paramétere: a kívánt elemisor formátumának sorszáma.

4.3.8.7. Header_Select

Funkciója:
Az aktuális fejlécsort átváltja a megadottra.

Formája:
<formátum_elem_kulcsszó>=H,Header_Select,a,r,c

Paramétere: a kívánt fejlécsor formátumának sorszáma.

4.3.8.8. Header_Print

Funkciója:
Kiíratja a megadott fejlécsort.

Formája:
<formátum_elem_kulcsszó>=H,Header_Print,a,r,c

Paramétere: a kívánt fejlécsor formátumának sorszáma.

4.3.8.9. Cr_Lf

Funkciója:
A formátum végén explicit üressor kiíratása.

Formája:
<formátum_elem_kulcsszó>=H,Cr_Lf,0,r,c

Paramétere: nincs.

4.3.8.10. Char_Size

Funkciója:
A karakterméret átállítása listázás közben.

Formája:
<formátum_elem_kulcsszó>=H,Char_Size,a,r,c

Paramétere: a kívánt karakterméret azonosítója.

A karakterméret a következők valamelyike lehet:
Normal,
Elite,
Condensed.

A következő ábra az egyes karakterméreteknél a hüvelykenkénti (inch-enkénti) karakterszámot, valamint a keskeny és széles nyomtatópapírnál az egy sorba írható karakterek számát szemlélteti:

	Karakter/ inch	Papírméret,kar/sor	
		8 inch	13,2 inch
Normal	10	80	132
Elite	12	96	158
Condensed	17	137	225

A karakterméret nagymértékben nyomtatófüggő, ezért ennek változtatása feltételezi a nyomtató típusának meghatározását, és az ehhez kapcsolódó szűrőfile "standard nyomtatóvezérlések" megfelelő elemeinek helyes beállítását.

4.3.8.11. Line_Spacing

Funkciója:
A sortáv átállítása listázás közben.

Formája:
<formátum_elem_kulcsszó>=H,Line_Spacing,a,r,c

Paramétere: a kívánt sortáv azonosítója.

A sortáv azonosító a következők valamelyike lehet:
N Normal,
H Half spacing,
W Whitout spacing.

A következő ábra az egyes sortávokat, valamint a 12 hüvelyk hosszú nyomtatópapírnál az egy lapra írható sorok számát szemlélteti:

		Sortáv	Sor/lap
Normal	(normális sorköz)	1/6	72
Half spacing	(fél sorköz)	1/8	96
Whitout spacing	(sorköz nélkül)	7/72	123

A sortáv vezérlése nagymértékben nyomtatófüggő, ezért ennek változtatása feltételezi a nyomtató típusának meghatározását, és az ehhez kapcsolódó szűrőfile "standard nyomtatóvezérlések" megfelelő elemeinek helyes beállítását.

4.3.8.12. Page_Length

Funkciója:
A laphossz átállítása a megadott értékre listázás közben.

Formája:
<formátum_elem_kulcsszó>=H,Page_Length,a,r,c

Paramétere: a kívánt laphossz.

4.3.8.13. Page_No

Funkciója:
Az aktuális lapszám beállítása a megadott értékre.

Formája:
<formátum_elem_kulcsszó>=H,Page_No,a,r,c

Paramétere: a kívánt lapszám.

4.3.8.14. Printer_Control

Funkciója:
Nyomtatófüggő vezérlés.

Formája:
<formátum_elem_kulcsszó>=H,Printer_Ctr,a,r,c

Paramétere: a kívánt nyomtatóvezérlés azonosítója.

A nyomtatóvezérlés azonosítója a következők valamelyike lehet:

Underline on	aláhúzás bekapcsolása,
Underline off	aláhúzás kikapcsolása,
Double strike on	dupla írás bekapcsolása,
Double strike off	dupla írás kikapcsolása,
Italic/alternate on	dőlt betűs írás bekapcsolása,
Italic/alternate off	dőlt betűs írás kikapcsolása,
User byte 0	1. speciális vezérlés,
User byte 1	2. speciális vezérlés,
User byte 2	3. speciális vezérlés,
User byte 3	4. speciális vezérlés,

User byte 4	5. speciális vezérlés,
User byte 5	6. speciális vezérlés,
User byte 6	7. speciális vezérlés,
User byte 7	8. speciális vezérlés,
User byte 8	9. speciális vezérlés,
User byte 9	10. speciális vezérlés.

A nyomtató vezérlése nagymértékben nyomtatófüggő, ezért ennek változtatása feltételezi a nyomtató típusának meghatározását, és az ehhez kapcsolódó szűrőfile "standard nyomtatóvezérlések" megfelelő elemeinek helyes beállítását.

4.4. Az elemisor formátuma

A következő két parancs az elemisor formátumát határozza meg.

4.4.1. DETAIL=

Funkciója:
Egy elemisor formátumának azonosítása.

Formája:
DETAIL=s,n,b,a,d

ahol
s (Serial number) az elemisor formátumának sorszáma,
n (Name) az elemisor formátumát azonosító név (max. 8 hosszú, alfanumerikus azonosító),
b (Before C function) a formátum kiírása előtt meghívandó felhasználói C függvény neve (max. 15 hosszú alfanumerikus név),
a (After C function) a formátum kiírása után meghívandó felhasználói C függvény neve (max. 15 hosszú alfanumerikus név),
d (Description) az elemisor formátumának leírása.

A végrehajtás kezdetén mindig az 1-es hivatkozási számú elemisor leírás értelmeződik. Más hivatkozási számokra átváltani a "Detail_Select" hashmark paranccsal, vagy C modulhívással lehet. A DETAIL utasítást az elemisor formátumát leíró formátumelemek követik.

4.4.2. DFORMAT=

Funkciója:
Az elemisor formátum egy elemét határozza meg.

Formája:
DFORMAT=k,n,l,r,c,w,d,j

ahol
k (Kind) a formátumelem fajtája, amely lehet:
S System,
D Detail,
O Old detail,
L totaL,
C Count,
A Average,
T Text,
H H-command,
n (Name) a formátumelem neve, illetve "text" fajtájú formátumelemnél a kiírandó szöveg,
l (Level) az összegzési szint "total", "count" és "average" fajtájú formátumelemeknél, illetve "H-command" fajtájú formátumelemeknél a Hashmark parancs paramétere,
r (Row) a formátumelem sora (0-tól kezdve),
c (Column) a formátumelem oszlopa (0-tól kezdve),
w (Width) a formátumelem kiírási szélessége,
d (Dec) a formátumelem decimális számjegyeinek száma, lebegőpontos értéknél,
j (Justify) kiírási szélességen belüli igazítás, amely lehet:
L (Left) balra,
M (Middle) középre,
R (Right) jobbra, vagy
Z (ZeroRight) jobbra vezető '0' karakterekkel.

4.5. A fejlécsor formátuma

A következő két parancs a fejlécsor formátumát határozza meg.

4.5.1. HEADER=

Funkciója:
Egy fejlécsor formátumának azonosítása.

Formája:
HEADER=s,n,b,a,d

ahol
s (Serial number) a fejlécsor formátumának sorszáma,
n (Name) a fejlécsor formátumát azonosító név (max. 8 hosszú, alfanumerikus azonosító),
b (Before C function) a formátum kiírása előtt meghívandó felhasználói C függvény neve (max. 15 hosszú alfanumerikus név),

a (After C function) a formátum kiírása után meghívandó felhasználói C függvény neve (max. 15 hosszú alfanumerikus név),
d (Description) a fejlécsor formátumának leírása.

A végrehajtáskor - ha programból nem bíráltuk felül - mindig az 1-es hivatkozási számú fejlécsor leírás értelmeződik. Más hivatkozási számokra átváltani a "Header_Select" hashmark paranccsal, vagy C modulhívással lehet. A HEADER utasítást a fejlécsor formátumát leíró formátumelemek követik.

4.5.2. HFORMAT=

Funkciója:
A fejlécsor formátum egy elemét határozza meg.

Formája:
HFORMAT=k,n,l,r,c,w,d,j

ahol
k (Kind) a formátumelem fajtája, amely lehet:
S System,
D Detail,
O Old detail,
L totaL,
C Count,
A Average,
T Text,
H H-command,
n (Name) a formátumelem neve, illetve "text" fajtájú formátumelemnél a kiírandó szöveg,
l (Level) az összegzési szint "total", "count" és "average" fajtájú formátumelemeknél, illetve "H-command" fajtájú formátumelemeknél a Hashmark parancs paramétere,
r (Row) a formátumelem sora (0-tól kezdve),
c (Column) a formátumelem oszlopa (0-tól kezdve),
w (Width) a formátumelem kiírási szélessége,
d (Dec) a formátumelem decimális számjegyeinek száma, lebegőpontos értéknél,
j (Justify) kiírási szélességen belüli igazítás, amely lehet:
L (Left) balra,
M (Middle) középre,
R (Right) jobbra, vagy
Z (ZeroRight) jobbra vezető '0' karakterekkel.

4.6. Az összegsor formátuma

A következő négy parancs az összegsor formátumát határozza meg.

4.6.1. TOTAL=

Funkciója:
Bevezeti a megjelölt szintű összegsor leírását.

Formája:
TOTAL=s,n,b,a,d

ahol
s (Serial number) az összegsor formátumának sorszáma,
n (Name) az összegsor formátumát azonosító név (max. 8 hosszú, alfanumerikus azonosító),
b (Before C function) a formátum kiírása előtt meghívandó felhasználói C függvény neve (max. 15 hosszú alfanumerikus név),
a (After C function) a formátum kiírása után meghívandó felhasználói C függvény neve (max. 15 hosszú alfanumerikus név),
d (Description) az összegsor formátumának leírása.

A szintek számozása 0-tól kezdődik, folyamatosan felfelé, amelyek közül a kontrollfokozat nélküli mindösszesen (Grand Total) összegsor a 0-ás.

4.6.2. CONTROL_ON=

Funkciója:
Megadja, hogy melyik elemi változó értékének változásákor kell aktivizálódnia az összegsor kiírásának. Más szavakkal: kijelöli a szinthez tartozó kontrollváltozót.

Formája:
CONTROL_ON=<elemi_változó_neve>

4.6.3. TOTAL_ON=

Funkciója:
Megadja, hogy az adott szinten mely elemi változó értékeit kell összegezni, vagy átlagolni. Egy szinthez több változót is ki lehet jelölni összegzésre, külön-külön TOTAL_ON utasításokkal.

Formája:
TOTAL_ON=<elemi_változó_neve>

4.6.4. FORMAT=

Funkciója:
Az összegsor formátum egy elemét határozza meg.

Formája:
FORMAT=k,n,l,r,c,w,d,j

ahol
k (Kind) a formátumelem fajtája, amely lehet:
S System,
D Detail,
O Old detail,
L totaL,
C Count,
A Average,
T Text,
H H-command,
n (Name) a formátumelem neve, illetve "text" fajtájú formátumelemnél a kiírandó szöveg,
l (Level) az összegzési szint "total", "count" és "average" fajtájú formátumelemeknél, illetve "H-command" fajtájú formátumelemeknél a Hashmark parancs paramétere,
r (Row) a formátumelem sora (0-tól kezdve),
c (Column) a formátumelem oszlopa (0-tól kezdve),
w (Width) a formátumelem kiírási szélessége,
d (Dec) a formátumelem decimális számjegyeinek száma, lebegőpontos értéknél,
j (Justify) kiírási szélességen belüli igazítás, amely lehet:
L (Left) balra,
M (Middle) középre,
R (Right) jobbra, vagy
Z (ZeroRight) jobbra vezető '0' karakterekkel.

5. A REPEDIT report editor

A REPEDIT a listadefiníciók párbeszédes karbantartását lehetővé tevő segédprogram. Egyaránt alkalmas meglévő listák módosítására és új listák létrehozására. Rávezető párbeszédeivel és listatervezésre is módot adó képernyős megoldásaival nagymértékben segíti a listadefiníciók kezelését. Külön előnye, hogy a képernyőn el nem férő, tehát 80 karakternél szélesebb listák is jól kezelhetők speciális, szinkronizált ablakkezelése segítségével.

5.1. A REPEDIT hívása és opciói

A report editor program indítása a következő paranccsal lehetséges:

>REPEDIT [-o <katalógus>] [-b] [-i] <reportfile>

ahol	<katalógus>	a generált headerfile és C függvénytábla modulja ebbe a katalógusba kerül,
	-b	fekete-fehér üzemmód,
	-i	a kezdeti képernyő letiltása,
	<reportfile>	a listaleíró file neve.

Az indítási parancs formája a -h opcióval lekérdezhető.

Ha az editor indításakor nem adjuk meg a "-i" opciót, akkor indítás után a képernyőn a következő kép áll össze:

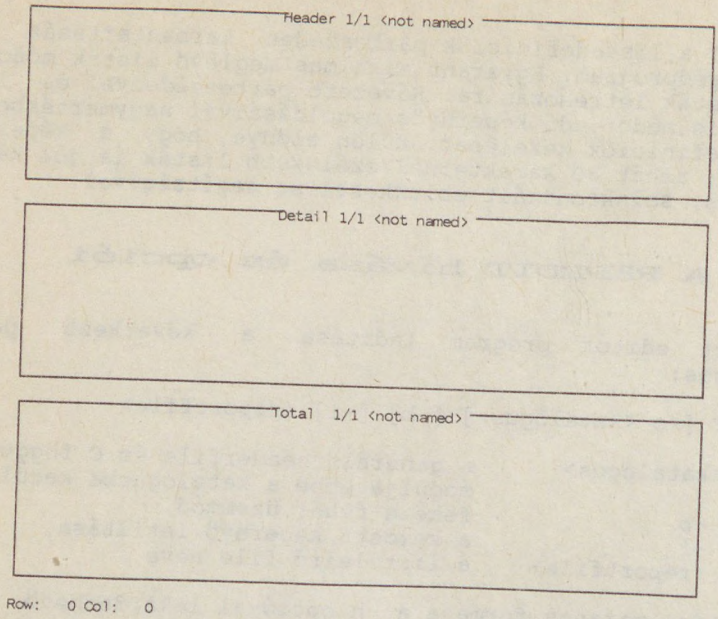
```
RepEdit  V1.00
Produced by MI Software

Press any key to start
        and
use F1 for help.
```

Tetszés szerinti billentyű leütésére (vagy 5 másodperc eltelte után) a bejelentkező képernyő eltűnik, s a formátumdefiníciós képernyő jelenik meg.

Ha létező listaleíró file-t adtunk meg, akkor betölti annak információit, ha nincs ilyen file, akkor üres táblázattal indul.

A formátumdefiníciós képernyő három, egymás alatti ablakot tartalmaz. A legfelső ablakban a fejlécsor formátumokat, a középsőben az elemisor formátumokat, az alsóban pedig az összegsor formátumokat határozhatjuk meg.

Egy új lista formátumdefiníciós képernyője a 3. ábrán látható.



3. ábra

A REPEDIT ismertetésékor egy lehetséges listaleírási logikát követünk. Először definiáljuk az elemi adatterületet, majd a fejlécsor(oka)t, elemisor(oka)t, összegsor(oka)t; a végén pedig beállítjuk az opciókat.

Az editor használatának elsajátításában jelentős segítséget nyújt az F1-gyel lekérhető információs ablak, amely a 4. ábrán látható. Az információs ablak az editor által értelmezett parancsok, illetve a parancsoknak megfelelő billentyűk rövid angol nyelvű leírását tartalmazza.

```
Position: cursor keys                  Position to elements: ^cursor keys
INS  : Insert/modify element           Alt-A: set 'After' C function for format
DEL  : Delete actual element           Alt-B: set 'Before' C function for format
ESC  : Exit                            Alt-C: Copy marked item to actual position
Tab  : jump to next window             Alt-D: Description of format
C    : Control-on for total            Alt-E: Erase all format elements
D    : Detail variables                Alt-F: generate C Function table
G    : Global infos                    Alt-G: Get format from other format
M    : Mark/unmark item                Alt-H: generate Header
N    : Next level of format            Alt-K: checK descriptions
P    : Previous level of format        Alt-L: set Levels to actual in every totals
S    : System variables                Alt-M: Move marked item to actual position
F1   : this help                       Alt-N: Name of format
F9   : OS Shell                        Alt-P: Pull marked item from other format
^A   : Advance to end of elem          Alt-S: Save report to file
^C   : Centering info                  Alt-U: search not Used detail variable
Alt= : Record key macro                Alt-X: eXchange kind of form element
Alt- : End of record key macro
                                                          <ESC,^,V,>,<>
```

4. ábra

5.2. Az elemi adatterület leírása

A D betű leütésére megjelenik az elemi adatterület 5. ábrán látható definíciós táblája.

Detail variables

Use cursor keys to position, ENTER to select/modify, ESC to quit
INS to insert, ^A to append, DEL to delete.

Name	Descr	Size	Type	Width	Dec.	Justify	Control	Total
tkod	*	10	Text	10	0	Left	1	N
tnev	*	40	Text	40	0	Left		N
tmenny	*	8	Double	14	2	Right		N
anyagkod	*	5	Text	5	0	Left		N
anyagnev	*	30	Text	30	0	Left		N
mennyiseg	*	8	Double	9	2	Right		N
egysegar	*	8	Double	9	2	Right		N
ertek	*	8	Double	9	2	Right		Y
tertek	*	8	Double	14	2	Right		Y

5. ábra

A táblázat első oszlopa az elemi változó nevét tartalmazza.

A második oszlopban egy '*' látható akkor, ha a változónak van leírása, egyébként a mező üres.

A harmadik oszlopban a változó belső ábrázolású méretét láthatjuk byte-ban, információs céllal.

A negyedik oszlop a változó típusát mutatja.

Az ötödik oszlop a változó alapértelmezett kiírási szélessége, a hatodik oszlop pedig lebegőpontos változó esetén a decimális jegyek száma.

A hetedik oszlop a balra (Left), középre (Middle), jobbra (Right) vagy a jobbra vezető '0'-kkal (ZeroRight) igazítást jelöli.

A nyolcadik oszlop mutatja annak az összegző fokozatnak a szintjét, amelynek kontrollváltozója az aktuális változó.

Az utolsó oszlop azt mutatja, hogy a változóra történik-e összegzés vagy átlagszámítás. Lehetséges értékei: Y (Yes, igen) vagy N (No, nem).

5.2.1. Elemi változó létrehozása

Az elemi változók táblázatába két módon szúrhatunk be új elemet:

- az aktuális változó elé az INS billentyű leütésével és
- az utolsó változó mögé a CTRL-A billentyű leütésével.

Az új változó megadásakor az editor először a mező nevét kéri, az "Enter new name:" kérdéssel. Az editor által feltett kérdésre az ESC billentyűvel utasithatjuk vissza a választ. Ez a lehetőség általában igaz: az univerzális megszakító és funkcióelhagyó billentyű az ESC.

A változó neve egy maximum 15 karakter hosszú, alfanumerikus azonosító lehet. Nem megengedett több azonos nevű elemi változó használata.

A név megadása után a "Select type" kérdés a mező típusára vonatkozik. A képernyő bal felső sarkában megjelenik a lehetséges választék:

```
Select type
 Byte
 unS. byte
 Int
 Uns. int
 Long
 date (M)
 Float
 Double
 Char
 Text
```

A kiemelt betű leütésével, vagy ráállás után az ENTER billentyű leütésével választhatjuk ki a megfelelő típust.

A harmadik kérdés ("Width") a mező kiírási hosszára vonatkozik.

A negyedik kérdés csak lebegőpontos változónál jelenik meg. Itt a decimális jegyek számát adhatjuk meg.

A kérdések megválaszolása után az új adatmező bekerül a táblázatba és felhasználható a formátumok meghatározásában.

5.2.2. Elemi változó módosítása

A képernyőn látható információk változtatását a kurzorbillentyűkkel való ráállás után az ENTER billentyű leütésével kezdeményezhetjük.

Bizonyos mezőkhöz a rendszer nem enged hozzáférni, ezek más módon kapnak értéket, és itt csak információs szerepük van. (Pl. annak jelzése, hogy a változó egy összegfokozat kontrollváltozója, az összegsornak a megadásakor kerül be a táblázatba. Módosítani is csak ott lehet.)

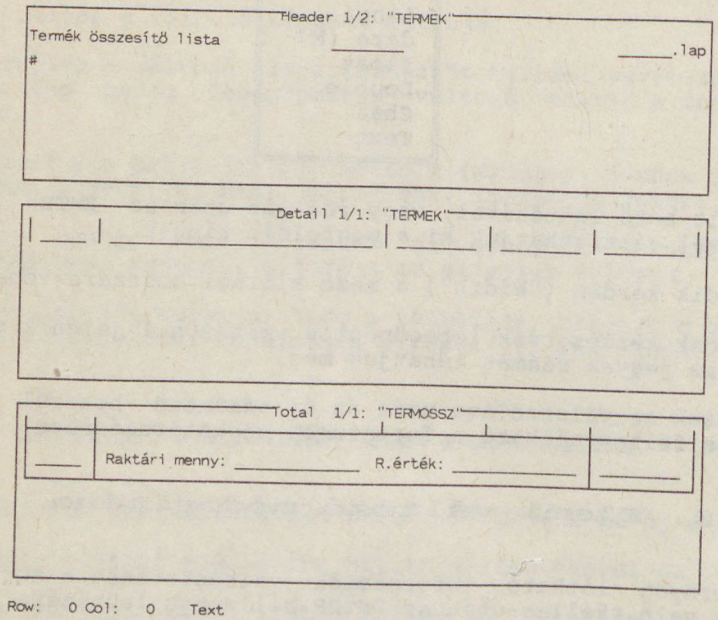
5.2.3. Elemi változó törlése

Az aktuális elemet a DEL billentyűvel törölhetjük, akkor ha az elemi változóra egyetlen formátumban sincs hivatkozás.

5.3. Formátumok megadása

A fejlécsor, az elemisor, az összegsor lényegében azonos módon adható meg, a formátumdefiníciós képernyő megfelelő ablakában.

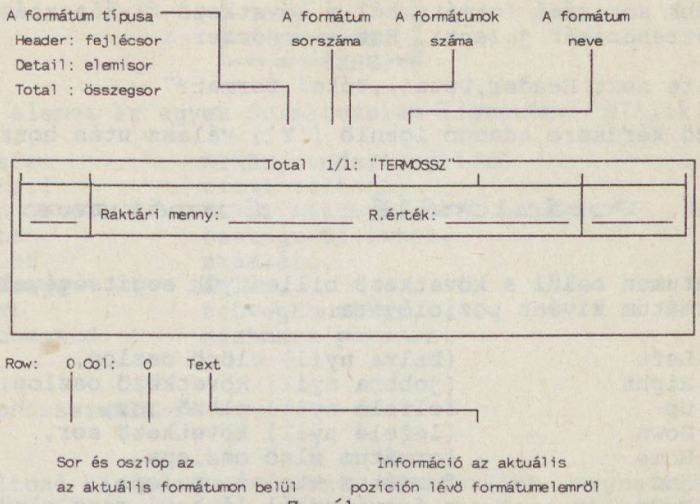
A példánkra vonatkozó formátumdefiníciós képernyőt szemlélteti a 6. ábra:



6. ábra

A képernyőn megjelenő információk jelentése a 7. ábrán látható.

A képernyő alján helyzetjelző kiírást láthatunk. Megjelenik az ablakon belüli sor- és oszloppozíció (mindkettő 0-tól számozva), és ha az aktuális pozíció valamilyen formátumelemhez tartozik, akkor összefoglaló információt láthatunk a formátumelemről.



7. ábra

Egy formátum maximális mérete:
fejlécsor : 35 x 225,
elemisor : 70 x 225,
összegsor : 70 x 225 karakter lehet.

Egy formátum aktuális méretét a formátumelemek határozzák meg. Ezek alapján a formátum mérete a legkisebb olyan méret, amelyen kívül nincs formátumelem. A formátum aktuális mérete két esetben játszik szerepet:

- egy formátumelem középre helyezésekor és
- dokumentáláskor.

5.3.1. Formátum kiválasztása

A formátumdefiníciós képernyőn az aktuális ablakot a TAB billentyű segítségével választhatjuk ki. Az aktuális ablakot a kurzor villogása jelzi.

Mint korábban már leírtuk, több fejléc, elemisor és összegsor

formátumunk lehet. Az aktuális formátum ablakában a keret felső sora információt tartalmaz arról, hogy mely sorszámú formátum az aktuális és hány ilyen típusú formátum van összesen. Az adott formátum ablakában az előző vagy a következő sorszámú formátumra a P (Previous, előző) vagy a N (Next, következő) betű leütésével ugorhatunk. Egy üres formátumról az előző sorszámú formátumot választva az üres formátum automatikusan eltörlődik. A létező legnagyobb sorszámú formátumból a következő kiválasztása új formátum létrehozását jelenti. Ezt a rendszer a

"Create next Header,Detail,Total format?"

ellenőrző kérdésre adandó igenlő ('Y') válasz után hozza létre.

5.3.2. Pozícionálás a formátumon belül

A formátumon belül a következő billentyűk segítségével állhatunk rá a formátum kívánt pozíciójára:

Left	(balra nyil) előző oszlop,
Right	(jobbra nyil) következő oszlop,
Up	felfelé nyil) előző sor,
Down	(lefelé nyil) következő sor,
Home	formátum első oszlopa,
End	formátum utolsó oszlopa,
PgUp	a formátumból látható rész első sora,
PgDn	a formátumból látható rész utolsó sora,
Ctr-Left	előző formátumelem,
Ctr-Right	következő formátumelem,
Ctr-Home	első formátumelem,
Ctr-End	utolsó formátumelem,
Ctr-A	Az aktuális formátumelem utáni pozíció.

5.3.3. Formátumelem létrehozása

A formátum megfelelő (üres) pozíciójára való ráállás után az INS billentyűvel kérhetjük új formátumelem létrehozását. Erre a bal felső sarokban megjelenik a választék:

```
┌Select kind┐
│ System    │
│ Detail    │
│ Old detail│
│ totaL     │
│ Count     │
│ Average   │
│ Text      │
│ H-command │
└──────┤ESC├┘
```

A táblázat elemei az egyes formátumelem-típusokat jelölik:

System	MIREP rendszerváltozó,
Detail	elemi változó,
Old detail	elemi változó előző értéke,
totaL	összegzett érték,
Count	számláló,
Average	átlag,
Text	szövegkonstans,
H-command	hashmark parancs.

5.3.3.1. Rendszerváltozó

Rendszerváltozó kérésére az elemi adatterület meghatározásánál ismertetett táblázathoz hasonló, a négy rendszerváltozót tartalmazó táblázat jelenik meg (8. ábra):

System variables

Use cursor keys to position, ENTER to select/modify, ESC to quit
INS to insert, ^A to append, DEL to delete.

Name	Descr	Size	Type	Width	Dec.	Justify	Control	Total
pageno		4	Long	6	0	Right		N
date		4	date (M)	8	0	Left		N
lineno		2	Int	5	0	Right		N
lineglobal		4	Long	5	0	Right		N

8. ábra

A táblázat elemei közül kiválaszthatjuk a megfelelőt. A rendszerváltozók jellemzői nem módosíthatók.

5.3.3.2. Elemi változó

Elemi változó kérése esetén az elemi adatterület meghatározásánál ismertetett táblázat jelenik meg. A táblázatban ekkor csak kiválasztani lehet, a változó jellemzőit itt nem változtathatjuk meg.

5.3.3.3. Elemi változó előző értéke

Elemi változó előző értékének kérésekor az elemi adatterület meghatározásánál ismertetett táblázat jelenik meg. A táblázatban ekkor csak kiválasztani lehet, a változó jellemzőit itt nem változtathatjuk meg.

5.3.3.4. Összegző

Összegző kérésekor az elemi adatterület meghatározásánál ismertetett táblázat jelenik meg. A táblázatban ekkor csak kiválasztani lehet, a változó jellemzőit itt nem változtathatjuk meg.

Összegzéshez csak numerikus típusú változó választható.

5.3.3.5. Számláló

Számláló kérése esetén a képernyő közepén megjelenő ablakban meg kell adni, hogy melyik szintű összegzés, illetve kontrollváltás számlálóját kérjük.

A szint értékének lehetséges értékei:

-1 a legbelső összegző fokozaton belül listázott elemisorok száma,
0 a listázott elemisorok száma összesen,
1 az 'l'-edik szintű kontrollváltások száma.

5.3.3.6. Átlag

Átlag kérésekor az elemi adatterület meghatározásánál ismertetett táblázat jelenik meg. A táblázatban ekkor csak kiválasztani lehet, a változó jellemzőit itt nem változtathatjuk meg.

Átlagoláshoz csak numerikus típusú változó választható.

5.3.3.7. Szövegkonstans

Szövegkonstansokat egyszerűen a szöveg beütésével lehet megadni. A szövegkonstans maximális hossza a formátum maximális szélességéig vagy az azonos sorban lévő következő formátumelemig terjedhet.

A szöveg megadásakor ékezetes betűket is definiálhatunk úgy, hogy a betű előtt megadjuk a három lehetséges ékezettípus (' : ") közül a megfelelőt (pl.: a "körülmény" szó megadása a "k:or:ulm'eny" karaktersorozattal lehetséges). Mivel a teljes magyar betűkészlet a legtöbb számítógépen nincs meg, ezért a rövid és a hosszú magánhangzók egy része egyformán jelenik meg.

A karakterkészlet kvázigrafikus jelei jól alkalmazhatók táblázatos kiíratási formátum keretezéséhez. Ehhez a REPEDIT a numerikus billentyűk SHIFT státuszban való leütését a következő táblazatban mutatott karaktereknek értelmezi (a billentyűket jelképező négyszögek bal felső sarkában a billentyű eredeti jelentése):

7 ┌	8 ┬	9 ┐	
4 ├	5 ┼	6 ┤	- ─
1 └	2 ┴	3 ┘	+ │

Ha a Scroll-Lock billentyű bekapcsolt állapotban van, akkor a billentyűk az ábrázolt karakterek kétvonalas változatát jelentik.

5.3.3.8. Hashmark parancs

Hashmark parancs kérésére megjelenik a lehetséges parancsok táblázata, ahol értelemszerűen kiválasztható a megfelelő:

New_Page	lapváltás,
Total_Page	feltételes lapváltás,
Total_Header	feltételes fejlécsor kiíratás,
Detail_Header	feltételes fejlécsor kiíratás,
Detail_Select	elemisor kiválasztása,
Detail_Print	elemisor explicit kiíratása,
Header_Select	fejlécsor kiválasztása,
Header_Print	fejlécsor explicit kiíratása,
Cr_Lf	üres sor beillesztése a formátumba,
Char_Size	karakterméret állítása,
Line_Spacing	sortávolság állítása,
Page_Length	lapméret állítása,
Page_No	lapszám állítása,
Printer_Control	nyomtatófüggő vezérlés.

Azoknál a hashmark parancsoknál, amelyeknek kötelező paraméterük van, a rendszer kéri a parancs paraméterét. A Char_Size, a Line_Spacing és a Printer_Control parancsoknál a megfelelő érték egy táblázatból választható ki, a többi paraméteres parancsnál pedig a képernyő közepén megjelenő ablakban adható meg.

5.3.3.8.1. New_Page

Funkciója:
Lapdobás kiváltása.

Paramétere: nincs.

5.3.3.8.2. Total_Page

Funkciója:
Lapdobás kiváltása, de összegsoroknál csak akkor, ha nem követi alacsonyabb szintű összegsor kiírása.

Paramétere: nincs.

5.3.3.8.3. Total_Header

Funkciója:
Kiíratja a megadott sorszámú fejlécsort, de összegsoroknál csak akkor, ha nem követi alacsonyabb szintű összegsor kiírása. Használata akkor indokolt, ha az összegsor után lapváltás nélkül folytatódik a lista, és a következő elemisorok előtt fejlécet akarunk íratni.

Paramétere: a kívánt fejlécsor sorszáma.

5.3.3.8.4. Detail_Header

Funkciója:
Kiíratja a megadott sorszámú fejlécsort, de csak akkor, ha az előző pontban leírt "Total_Header" parancs végrehajtására nem volt szükség.

Paramétere: a kívánt fejlécsor sorszáma.

5.3.3.8.5. Detail_Select

Funkciója:
Az aktuális elemisort átváltja a megadottra.

Paramétere: a kívánt elemisor formátum sorszáma.

5.3.3.8.6. Detail_Print

Funkciója:
Kiíratja a megadott elemisort.

Paramétere: a kívánt elemisor formátum sorszáma.

5.3.3.8.7. Header_Select

Funkciója:
Az aktuális fejlécsort átváltja a megadottra.

Paramétere: a kívánt fejlécsor formátum sorszáma.

5.3.3.8.8. Header_Print

Funkciója:
Kiíratja a megadott fejlécsort.

Paramétere: a kívánt fejlécsor formátum sorszáma.

5.3.3.8.9. Cr_Lf

Funkciója:
A formátum végén explicit üressor kiíratása.

Paramétere: nincs.

5.3.3.8.10. Char_Size

Funkciója:
A karakterméretet átállítása listázás közben.

Paramétere: a kívánt karakterméret azonosítója.

A karakterméret a következők valamelyike lehet:
Normal,
Elite,
Condensed.

A következő ábra az egyes karakterméreteknél a hüvelykenkénti (inch-enkénti) karakterszámot, valamint a keskeny és széles nyomtatópapírnál az egy sorba írható karakterek számát szemlélteti:

	Karakter/ inch	Papírméret,kar/sor	
		8 inch	13,2 inch
Normal	10	80	132
Elite	12	96	158
Condensed	17	137	225

A karakterméret nagymértékben nyomtatófüggő, ezért ennek változtatása feltételezi a nyomtató típusának meghatározását, és az ehhez kapcsolódó szűrőfile "standard nyomtatóvezérlések" megfelelő elemeinek helyes beállítását.

5.3.3.8.11. Line_Spacing

Funkciója:
A sortáv átállítása listázás közben.

Paramétere: a kívánt sortáv azonosítója.

A sortáv azonosító a következők valamelyike lehet:

N Normal,
H Half spacing,
W Whitout spacing.

A következő ábra az egyes sortávokat, valamint a 12 hüvelyk hosszú nyomtatópapírnál az egy lapra írható sorok számát szemlélteti:

		Sortáv	Sor/lap
Normal	(normális sorköz)	1/6	72
Half spacing	(fél sorköz)	1/8	96
Whitout spacing	(sorköz nélkül)	7/72	123

A sortáv vezérlése nagymértékben nyomtatófüggő, ezért ennek változtatása feltételezi a nyomtató típusának meghatározását, és az ehhez kapcsolódó szűrőfile "standard nyomtatóvezérlések" megfelelő elemeinek helyes beállítását.

5.3.3.8.12. Page_Length

Funkciója:
A laphossz átállítása a megadott értékre.

Paramétere: a kívánt laphossz.

5.3.3.8.13. Page_No

Funkciója:
Az aktuális lapszám beállítása a megadott értékre.

Paramétere: a kívánt lapszám.

5.3.3.8.14. Printer_Control

Funkciója:
Nyomtatófüggő vezérlés.

Paramétere: a kívánt nyomtatóvezérlés azonosítója.

A nyomtatóvezérlés azonosítója a következők valamelyike lehet:

Underline on	aláhúzás bekapcsolása,
Underline off	aláhúzás kikapcsolása,
Double strike on	dupla írás bekapcsolása,
Double strike off	dupla írás kikapcsolása,
Italic/alternate on	dőlt betűs írás bekapcsolása,
Italic/alternate off	dőlt betűs írás kikapcsolása,
User byte 0	1. speciális vezérlés,
User byte 1	2. speciális vezérlés,
User byte 2	3. speciális vezérlés,
User byte 3	4. speciális vezérlés,
User byte 4	5. speciális vezérlés,
User byte 5	6. speciális vezérlés,
User byte 6	7. speciális vezérlés,
User byte 7	8. speciális vezérlés,
User byte 8	9. speciális vezérlés,
User byte 9	10. speciális vezérlés.

A nyomtató vezérlése nagymértékben nyomtatófüggő, ezért ennek változtatása feltételezi a nyomtató típusának meghatározását, és az ehhez kapcsolódó szűrőfile "standard nyomtatóvezérlések" megfelelő elemeinek helyes beállítását.

5.3.4. Formátumelem módosítása

Egy már korábban meghatározott formátumelem részletes lekérdezését és estleges módosítását az elem első pozíciójára való ráállás után az INS billentyű leütésével kezdeményezhetjük.

Rendszerváltozó, elemi változó, elemi változó előző értéke, összeg, számláló és átlag esetén a következő abalak jelenik meg, értelemszerűen kitöltve.

```
═══════════════════Format element:═══════════════════
Kind        : totaL
Name        : ertek
Size        : 8
Type        : Double
Width       : 9
Dec         : 2
Justify     : Right
Position    : 1,67
Level       : 1
═══════════════════════════════════════════════ESC═══
```

Az összegző fokozatot jelző szint (Level) sor csak összeg, számláló és átlag esetén jelenik meg. Számláló esetén a név, a méret és a decimális jegyek száma nincs kitöltve. A formátumelem jellemzői közül csak a szélesség (Width), decimális jegyek száma (Dec, csak lebegőpontos változónál), az igazítás (Justify) és az

összegző fokozat szintje (Level, összegnél, számlálónál és átlagnál) változtatható meg.

A hashmark parancsnál az INS hatására a következő táblázat jelenik meg:

```
═══════════════════Format element:═══════════════════
Kind              : H-command
Name              : Detail_Header
Argument          : 2
═══════════════════════════════════════════════ESC═══
```

Itt ráállás után a hasmark parancsot és annak paraméterét változtathatjuk meg.

Szövegkonstansoknál a szöveg tetszés szerint változtatható, az előző fejezetben leírtak szerint.

Az elemi változóra hivatkozó (Detail, Old detail, TotaL, Average) formátumelem fajtája az Alt-X billentyűvel a következő fajtára módosítható. (Pl. a Detail formátumelemből Old detail, az Average formátumelemből Detail formátumelem lesz).

5.3.5. Formátumelem törlése

Egy formátumelem annak első pozíciójára való ráállás után a DEL billentyűvel törölhető a formátumból.

5.3.6. Formátumelem mozgatása

Ha egy formátumelem helyét meg akarjuk változtatni az aktuális formátumon belül, akkor az M betű leütésével (Mark) meg kell jelölni, majd a kívánt új pozíción az Alt-M (Move) billentyű leütésére a formátumelem a kívánt pozícióra kerül.

Gyakran kell az egyes formátumelemek középre helyezni. Ehhez a formátumelemre való ráállás után a Ctr-C (Center) billentyű leütése adhat segítő információt. A billentyű leütésére a képernyőn megjelenik egy táblázat, amely tartalmazza:

- a formátum szélességét,
- a formátumelem szálességét,
- a formátumelem előtt lévő oszlopok számát és
- a formátumelem mögött lévő oszlopok számát.

Ennek alapjan könnyen eldönthető, hogy mely irányba és hány oszloppal kell a formátumelemet elmozgatni.

5.3.7. Formátumelem másolása

A formátumelemeket másolhatjuk az aktuális formátumból, vagy egy másik formátumablakból. Mindkét esetben először az M betű leütésével meg kell jelölni a másolni kívánt elemet, majd

- az aktuális formátumon belüli másoláskor a kívánt pozíción le kell ütni az Alt-C (Copy) billentyűt,
- másik formátumablakból való másoláskor a TAB, N, P billentyűkkel rá kell állni a kívánt formátumra és a kívánt pozíción le kell ütni az Alt-P (Pull) billentyűt, majd ki kell választani, hogy mely típusú formátumból akarjuk a megjelölt formátumelemet bemásolni.

5.3.8. Formátum másolása

Egy teljes formátumot egy üres(!) formátumba az Alt-G (Get) billentyű segítségével másolhatunk át. A billentyű leütése után ki kell választani, hogy melyik típusú formátumot akarjuk másolni, majd azt, hogy a kiválasztott típusú formátumokból mely sorszámú a másolandó.

5.3.9. Formátum törlése

Egy formátum összes elemét az Alt-E (Erase) billentyűvel törölhetjük.

Ha egy formátum összes elemét töröltük és a formátum típuson belül ez a legnagyobb sorszámú formátum, akkor a P (Previous) billentyűvel az előző sorszámú formátumot választva az üres formátum automatikusan törlődik.

5.3.10. A formátum neve és leírása

A formátum nevét az Alt-N (Name) billentyű leütése után állíthatjuk be, illetve módosíthatjuk. A formátum neve egy maximum 8 karakteres alfanumerikus azonosító. A formátum neve dokumentációs célon kívül azonosítóként szolgál a futtatórendszer néhány függvénye számára.

A formátum rövid leírását az Alt-D (Description) billentyű leütése után adhatjuk meg. Ékezetes karakterek a szövegkonstansok megadásánál leírt módon üthetők be. A formátum leírása csak dokumentációs célokat szolgál.

5.3.11. A felhasználói C függvények

A lista végrehajtásába annak lényeges pontjain be tudunk avatkozni, felhasználói C függvények illesztésével. E lényeges pontok az egyes formátumok kiírása, illetve végrehajtása előtti és utáni időpont. A formátumhoz rendelhető két felhasználói C függvény ezért Before (előtte) és After (utána) C függvényként is jelöljük.

A formátum végrehajtása előtt meghívandó C függvényt az Alt-B (Before), a végrehajtás után meghívandó függvényt pedig az Alt-A (After) billentyű leütése után lehet megadni, vagy módosítani. A felhasználói C függvény neve egy maximálisan 15 karakter hosszú, a C nyelv szabályainak megfelelő azonosító lehet.

5.4. Opciók állítása

A listadefiníció globális opcióit a G billentyű leütése után módosíthatjuk.

A billentyű leütése után a megjelenik a 9. ábrán látható ablak:

```
Global infos:
A. REPORT=        : Termék kimutatás
B. SORTED=        : A termék kódjára
C. DEVICE=        : Diskfile
D. LSTFDIR=       : \WORK
E. LSTFNAME=      : WORK.LST
F. SERVER=        : FS1
G. CLASS=         : 2
H. COPYNUM=       : 1
I. PRNTYPE=       : EPSON
J. CHARSIZE=      : Elite
K. LINESPACING=   : Normal
L. PAGELENGTH=    : 65
M. DATEMODE=      : Hungarian
N. OPENMODE=      : Write
O. DETPRINT=      : Yes
P. FORMFEED=      : Yes
Q. FRSTFF=        : No
R. LASTFF=        : Yes
                                    ESC
```

9. ábra

Az egyes opciókat az első oszlopban található betű leütése után, vagy ráállás és az ENTER leütése után módosíthatjuk.

A szöveges és a numerikus információkat egyszerűen meg kell adni, ahol pedig a választási lehetőség kötött, ott a képernyő jobb felső sarkában megjelenő táblázatban állíthatjuk be a kívánt értéket.

Az egyes opciók jelentését a 4. A listaleíró nyelv fejezet első (Opció állítás) pontja tárgyalja részletesen, a fenti táblázat bal oldalán lévő szöveggel megegyező néven.

5.4.1. A listadefiníció leírása

Funkciója:

A listadefiníció leírásának megadása (csak dokumentációs céllal).

Ékezetes karakterek a szövegkonstansok megadásánál leírt módon adhatók meg.

5.4.2. A rendezettség leírása

Funkciója:
Az elemi adatok kívánt rendezettségének leírása (csak dokumentációs céllal).

Ékezetes karakterek a szövegkonstansok megadásánál leírt módon adhatók meg.

5.4.3. Az output eszköz

Funkciója:
A nyomtatandó lista eszközének meghatározása.

Az eszköz lehet:
C (Consol) képernyő,
D (Diskfile) lemezen tárolt file,
L (Localprinter) lokális nyomtató,
G (Globalprinter) hálózati nyomtató.

5.4.4. Az output listafile könyvtára

Funkciója:
Ha a lista lemezen tárolt file-ba készül, akkor a file könyvtárának megadása.

5.4.5. Az output listafile neve

Funkciója:
Ha a lista lemezen tárolt file-ba készül, akkor a file nevének megadása.

5.4.6. A hálózati server neve

Funkciója:
A globális nyomtató server-ének kijelölése. Ha a kiíratás nem globális nyomtatóra megy, megadása hatástalan.

5.4.7. A hálózati nyomtatás osztálya

Funkciója:
A hálózati nyomtatás osztályának ("queue" számának) kijelölése.

Ha a kiíratás nem globális nyomtatóra megy, megadása hatástalan.

5.4.8. Nyomtatási példányszám

Funkciója:
A globális nyomtatón kért példányszám megadása.

Ha a kiíratás nem globális nyomtatóra megy, megadása hatástalan.

5.4.9. A nyomtató típusa

Funkciója:
A nyomtató típusának meghatározása.

A nyomtató típusa egy legfeljebb 8 karakterből álló azonosító. A nyomtató akkor definiált, ha a nyomtató típusának megfelelő szűrőfile rendelkezésre áll az aktuális, illetve a

SET PTC=<ptc-könyvtár>

DOS paranccsal beállított könyvtárban.

A MIREP disztributív anyaga tartalmazza a leggyakoribb nyomtatókhoz definiált szűrőfile-okat.

5.4.10. A karakterméret

Funkciója:
A karakterméret beállítása a listázás megkezdése előtt.

A karakterméret a következők valamelyike lehet:
Normal,
Elite,
Condensed.

A következő ábra az egyes karakterméreteknél a hüvelykenkénti (inch-enkénti) karakterszámot, valamint a keskeny és széles nyomtatópapírnál az egy sorba írható karakterek számát szemlélteti:

	Karakter/ inch	Papírméret,kar/sor	
		8 inch	13,2 inch
Normal	10	80	132
Elite	12	96	158
Condensed	17	137	225

A karakterméret nagymértékben nyomtatófüggő, ezért ennek változtatása feltételezi a nyomtató típusának meghatározását, és az ehhez kapcsolódó szűrőfile "standard nyomtatóvezérlések" megfelelő elemeinek helyes beállítását.

5.4.11. A nyomtatási sortávolság

Funkciója:
A sortáv beállítása a listázás megkezdése előtt.

A sortáv azonosító a következők valamelyike lehet:
N Normal,
H Half spacing,
W Whitout spacing.

A következő ábra az egyes sortávokat, valamint a 12 hüvelyk hosszú nyomtatópapírnál az egy lapra írható sorok számát szemlélteti:

		Sortáv	Sor/lap
Normal	(normális sorköz)	1/6	72
Half spacing	(fél sorköz)	1/8	96
Whitout spacing	(sorköz nélkül)	7/72	123

A sortáv vezérlése nagymértékben nyomtatófüggő, ezért ennek változtatása feltételezi a nyomtató típusának meghatározását, és az ehhez kapcsolódó szűrőfile "standard nyomtatóvezérlések" megfelelő elemeinek helyes beállítását.

5.4.12. A nyomtatási lap hossza

Funkciója:
A lista egy lapjára kerülő fizikai sorok számának meghatározása.

Az alapértelmezés szerint: 65.

5.4.13. A dátum kiírási formátuma

Funkciója:
A dátum típusú mezők kiíratási módjának meghatározása.

A formátum lehetséges értékei:
E English,
H Hungarian,
G German.

A megfelelő dátumformátumok rendre (E: év, H: hó, N: nap):
HH/NN/EE (angol)
EE/HH/NN (magyar)
NN/HH/EE (német)

Alapértelmezés szerint: Hungarian.

5.4.14. Az output lista megnyitási módja

Funkciója:
A listafile nyitási módjának meghatározása.

A nyitási mód lehet
W Write vagy
A Append.

"Write" módban az output listafile felülíródik, "Append" módban az új lista a már meglévő végére kerül. Az opció hatástalan, ha a lista nem file-ba kerül. Append módot kell használni lemezre írt naplók esetén.

Alapértelmezés szerint: "Write".

5.4.15. Az elemisor kiírásának letiltása

Funkciója:
Az elemisor kiíratásának ki-, ill. bekapcsolása.

Alapértalmezés szerint: ON

5.4.16. Teljes lapkitöltés kérése

Funkciója:
A lapok elejére kerülő FormFeed (lapemelés) karakter helyett a lapok teljes kitöltésének kérése.

Alapértelmezés szerint: ON, ami azt jelenti, hogy új lap kezdete előtt a rendszer nem tölti ki az aktuális lap maradék sorait üres sorokkal, hanem a FormFeed karakter segítségével a következő lap tetejére ugrik. Az opciót akkor kell OFF értékre állítani, ha ez a megoldás valamilyen oknál fogva nem megfelelő és a lapokon lévő sorok számának azonosnak kell lennie.

5.4.17. Első lapemelés kérése

Funkciója:
A legelső lapra a FormFeed (lapemelés) karakter kiíratásának kérése.

Alapértelmezés szerint: OFF

Ha az opciót ON-ra állítjuk, akkor a lista legelső fejléce új lapra kerül. Az opció beállítására akkor lehet szükség, ha nem biztos, hogy a nyomtató a listázás kezdetekor lap tetején áll.

5.4.18. Utolsó lapemelés letiltása

Funkciója:
Az utolsó lap után a FormFeed (lapemelés) karakter kiíratásának letiltása.

Alapértelmezés szerint: ON

Ha az opciót OFF-ra állítjuk, akkor a lista generálása után a nyomtató nem áll a következő lap elejére.

5.5. Az editált lista lemezre írása

A listadefiníció aktuális állapotának listaleíró file-ba való mentését az ALT-S (Save) billentyűvel kérhethük. A billentyű leütése után egy ablak jelenik meg, ahol meg kell adni a kívánt file-nevet. A rendszer automatikusan az input leírófile nevét ajánlja fel. A név megadását az ENTER billentyűvel kell befejezni. Az ESC billentyűt és az üres file-nevet a rendszer a

mentési kérelem visszavonásaként értelmezi.

Ha a listadefiníciót megváltoztattuk, akkor a MIREP a rendszerből való kilépéskor figyelmeztet erre és automatikusan felkínálja a listadefiníció file-ba való kimentését.

5.6. Headerfile generálása

A MIREP futtató rendszerének függvényeit természetesen egy, a felhasználó által megírandó C program fogja meghívni. Az egyes elemi változók sorszámait megadó konstansdefiníciók és az elemi adatbuffer C nyelvből való értelmezését elősegítő struktúra definíciója hasznos segítője lehet a programozásnak. A rendszer az ezeket tartalmazó headerfile-t az Alt-H (Header generating) billentyű hatására, a listaleíró file nevével megegyező, de .RH kiterjesztésű file-ba generálja. Ezt a file-t a C program az #include metaparanccsal használhatja fel.

A példánkban szereplő listadefiníció alapján generált headerfile az E. függelékben található.

Ha már korábban generáltunk headerfile-t, és az elemi adatbuffert oly módon változtattuk meg azóta, hogy az új headerfile generálását teszi szükségessé, akkor kilépéskor a MIREP rendszer automatikusan felajánlja a headerfile új változatának generálását.

5.7. C függvénytábla generálása

Ha a listadefinícióban felhasználói C függvényeket illesztettünk egyes formátumokhoz, akkor biztosítani kell a futtató rendszer számára ezen C függvények ismeretét. Ezt a REPEDIT által generált C függvénytábla és az "r_set_funcs()" függvény hívásával lehet biztosítani.

A függvénytáblát a rendszer az Alt-F (Function table) billentyű hatására készíti el, a listaleíró file nevével megegyező, de .RCT kiterjesztésű file-ban.

5.8. Egyéb parancsok

A következő parancsok olyan funkciókat valósítanak meg, amelyek a listadefiníció korrektségének ellenőrzését, és az egész listadefiníciós eljárás megkönnyítését szolgálják.

5.8.1. Leírások ellenőrzése

A listadefinícióban dokumentációs céllal leírhatjuk:

- a lista rendeltetését,
- az adatok kívánt rendezettségét,
- az egyes elemi változók jelentését,
- az egyes formátumok nevét és
- az egyes formátumok funkcióját.

Gyakran előfordul, hogy a definiálás során elfelejtkezünk e leírásokról, és a REPDOC segédprogrammal elkészített dokumentációnk így hiányos. Ezt előzhetjük meg az Alt-K (checK descriptions) billentyűhöz rendelt funkcióval, amely figyelmeztet, hogy a listadefiníció mely részénél hiányzik a leírás.

5.8.2. Nem használt elemi változók

A listadefiníció karbantartása során, formátumelemek eltörlése után lehetséges, hogy a korábban deklarált elemi változók közül némelyikre nincs többé szükség. Ezt ellenőrizhetjük könnyedén az Alt-U (Unused detail variable) billentyűvel. A billentyű hatására a rendszer megkeresi azt az elemi változót, amelyre egyetlen formátumban sincs hivatkozás, és felajánlja a változó törlését az elemi adatbufferből.

5.8.3. Kiírási szélességek ellenőrzése

Az egyes elemi változókra való hivatkozásoknál felülbírálhatjuk az adott változó konkrét kiírási szélességét. Az Alt-W (Width) billentyű segítségével megkereshetjük azokat a hivatkozásokat, amelyek nem az alapértelmezés szerinti kiírási szélességgel definiáltunk.

5.8.4. Az összegzési szintek beállítása

Az összegsorok gyakran hasonlítanak egymásra, csak más-más kontrollváltás hatására aktivizálódnak. Igy egy összegsort gyakran más formátumból, másolás után alakítjuk ki. A másolgatások során előfordulhat, hogy pl. a 2-es szintű összegsor formátum 1-es, vagy 0-ás szintű összegre, átlagra vagy számlálóra hivatkozik. Az Alt-L (Levels) billentyűvel minden összegsor formátum minden ilyen típusú hivatkozását a megfelelő szintre lehet állítani.

5.8.5. Kiszállás az operációs rendszerbe

Az F9 billentyű hatására a MIREP rendszer felfüggeszti működését és átadja a vezérlést az operációs rendszernek. A DOS parancsmódban végrehajtott műveletek után az

```
>exit
```

paranccsal térhetünk vissza a lista definiálásához, és a munkát ott folytathatjuk, ahol az F9 billentyű leütése előtt tartottunk. Természetesen visszatéréskor az aktuális könyvtárnak és lemezmeghajtónak ugyanannak kell lennie, mint az F9 leütésekor volt.

5.9. Makrózás

A REPEDIT rendszer lehetőséget ad az egymás után többször végrehajtandó parancssorozatok megjegyzésére és lejátszására. Ezt makrózásnak nevezzük.

A leütött billentyűk megjegyzését az Alt= (Alt egyenlő) billentyűkombináció leütésével kérhetjük. A billentyű leütése után megjelenő kérdésre le kell ütni azt a billentyűt, amelyhez a megjegyzett parancsokat hozzá akarjuk rendelni. Ez az F2, F3, F4, F5, F6, F7, F8 vagy F10 funkcióbillentyűk valamelyike lehet. Ezután a rendszer jegyzi a leütött billentyűket mindaddig, amíg az Alt- (Alt mínusz) billentyűt le nem ütjük. A parancsrögzítés tényét a jobb felső sarokban villogó "Recording key macro" felirat jelzi.

A parancssor rögzítése után a hozzárendelt billentyű hatására a rendszer visszajátsza (végrehajtja) a rögzített parancsokat, és ezt a jobb felső sarokban villogó "Replaying key macro" feliratal jelzi.

Jelen változatban több makró definiálása, a makró file-ba való kiírása és file-ból való beolvasása nem lehetséges.

Tipikus példa a makrózás használatára az, amikor egy formátum összes elemét egy sorral lejjeb akarjuk csúsztatni. Az eljárás menete a következő:

1. álljunk a formátum utolsó elemére a Ctr-End billentyűvel,
2. üssük le az Alt= makrókezdő billentyűt,
3. üssük le az F10-es billentyűt, mint makróbillentyűt,
4. az 'M' billentyűvel jelöljük meg a formátumelemet,
5. léptessük a kurzort egy sorral lejjeb,
6. az Alt-M billentyűvel mozgassuk az új helyére a megjelölt elemet,

7. a Ctr-Left billentyűvel ugorjunk az előző formátumelemre,
8. fejezzük be a makró definiálását az Alt- billentyűvel,
9. ismételten üssük le az F10 makróbillentyűt addig, amíg a formátum első eleme is a megfelelő helyre kerül.

5.10. Kiszállás a rendszerből

A MIREP rendszerből az ESC billentyű segítségével szállhatunk ki. A biztonság kedvéért a rendszer felteszi a következő kérdést:

"Are You sure to quit REPEDIT?"

(Biztos, hogy ki akar szállni a REPEDIT rendszerből?). A kérdésre adott ESC vagy 'N' (No) válasz esetén folytathatjuk a listadefiníció módosítását. Igenlő 'Y' (Yes) válaszra, ha volt változtatás a listadefinícióban, akkor a rendszer felajánlja a file-ba való mentést:

"The modification will be lost! Do You want to save this report?"

(A változtatások elvesznek! Akarja menteni a listadefiníciót?) Nemleges válasz (ESC, 'N') esetén a rendszer kiszáll. Igenlő válasz esetén a listadefiníció új változata kerül az inputként megadott listaleíró file-ba.

Ha korábban már generáltattunk headerfile-t és az elemi adatbuffer megváltozott, akkor a rendszer automatikusan felajánlja az új headerfile generálását a

"Generate new headerfile?"

(Új headerfile generálása?) kérdéssel. A lehetséges válaszok ugyanazok, mint az előzőekben.

Ha korábban már generáltattunk C függvénytáblát és a felhasználói C függvények változtak, akkor a rendszer automatikusan felajánlja az C függvénytábla generálását a

"The C function table is new! Do You want to generate?

(A C függvénytábla új! Akarja generálni?) kérdéssel. A lehetséges válaszok ugyanazok, mint az előzőekben.

6. Segédprogramok

A listaleíró file-ok legfontosabb segédeszköze kétségtelenül a REPEDIT report editor. Bizonyos esetekben igen hasznosak lehetnek a következő generatív és egyéb segédprogramok:

- nyomtatható dokumentáció készítése egy listadefinícióból,
- a listadefiníció tesztelése tesztadatok generálásával,
- listadefiníció létrehozása MISAM file leírásból,
- listadefiníció létrehozása MIQUEL lekérdezés definícióból,
- listadefiníció létrehozása MIFORM adatbeviteli képernyő definícióból.

A generatív eszközök erősen kötődnek a MIxxx adatfeldolgozási eszköztárhoz.

6.1. A REPDOC listadokumenter

Funkciója:
Dokumentációt készít a listadefinícióról.

A program hívása:

```
>REPDOC rtxfile [-o output] [-d] [-g] [-i] [-a] [-c c] [-l l]
```

ahol	rtxfile	a listaleírófile neve,
	-o output	a dokumentációs file neve (def.: .RDC),
	-a	a meglévő output file-hoz hozzáírandó (alapértelmezés: új dokumentáció),
	-g	az elemi változokról ne készüljön dokumentáció,
	-d	PTFORM szövegformázó program számára felkészített dokumentáció forrás (alapértelmezés.: CWI szabvány formátum),
	-i	'-d' esetén beillesztett (Include-olt) dokumentáció,
	-c c	PTFORM fejezet szint, 1-5 (alapértelmezés: 1: $CPT.),
	-l l	nyelv (E: angol, H: magyar, G: német alapértelmezés: E).

A példánknak használt listadefiníció dokumentációja a C. függelékben található.

6.2. A REPTEST listatesztelő

Egy lista megtervezésekor még a REPDOC programmal generált dokumentációval is nehéz a lista végleges képét áttekinteni. Ha még nincs meg a listaleírás alapján listázó program, vagy ha még nem állnak rendelkezésre megfelelő adatok, akkor szintén nehéz a végleges listakép ellenőrzése. A REPTEST segédprogram automatikus adatgenerálással segít feloldani ezt a problémát.

A program hívása:

```
>REPTEST rtxfile [-d] [-c ptcfile] [-o output] [-r n] [-l l n]
```

ahol	rtxfile	a listaleíófile neve,
	-c ptcfile	a szűrőfile neve,
	-o output	az output listafile vagy eszköz,
	-r n	az összegfokozaton belül generálandó elemisorok száma (alapért.: 3),
	-l l n	az 'l'-edik összegfokozatból generálandó kontrollváltások száma (alapért.: 3),
	-d	az elemisorok kiíratásának letiltása.

A példánkban használt listaleírás alapján a

```
>REPTEST TESTTECH
```

paranccsal generált lista az F. függelékben található.

6.3. A QTXTORTX, mint MISAM konvertáló

Egyszerűbb listák definiálásakor jó kiindulópont lehet a lista alapjául szolgáló MISAM állomány leírása. MISAM file-leírásból a QTXTORTX segédprogrammal hozhatunk létre listaleírófile-t, amelyet aztán a REPEDIT editorral már könnyen módosíthatunk.

A program hívása:

```
>QTXTORTX -f fdffile [-r rtxfile]
```

ahol	-f fdffile	az input MISAM leírófile neve,
	-r rtxfile	az output listaleírófile neve.

Ha az output listaleírófile nevét nem adjuk meg, akkor az az input file nevéből képződik.

6.4. A QTXTORTX, mint MIQUEL konvertáló

Lekérdezési feladatok esetén - ha a lekérdezés a MIQUEL lekérdező rendszer segítségével történik - hasznos, ha a lekérdezésbe bevont adatokból indulunk ki a listadefiníció kialakításához. MIQUEL lekérdezés leírásból a QTXTORTX segédprogrammal hozhatunk létre listaleírófile-t, amelyet aztán a REPEDIT editorral már könnyen módosíthatunk.

A program hívása:

```
>QTXTORTX -q qtxfile [-r rtxfile]
```

ahol	-q qtxfile	az input MIQUEL leíró file neve,
	-r rtxfile	az output listaleíró file neve.

Ha az output listaleíró file nevét nem adjuk meg, akkor az az input file nevéből képződik.

6.5. Az FTXTORTX, MIFORM konvertáló

A MIFORM adatbeviteli képernyő leírásokból e segédprogrammal hozhatunk létre listadefiníciót.

Funkciója:
Adatbeviteli képernyőhöz csatlakozó listadefinícióhoz induló leírófile készítése.

A program hívása:

```
>FTXTORTX formfile [-r rtxfile]
```

ahol	formfile	az input MIFORM leírófile neve,
	-r rtxfile	az output listaleírófile neve.

Ha az output listaleírófile nevét nem adjuk meg, akkor az az input file nevéből képződik.

7. Listák készítése

A dokumentáció eddigi része csaknem teljes egészében a listadefinícióval foglalkozott. A listadefinícióból azonban csak akkor lesz adatokat tartalmazó kész lista, ha létezik egy olyan mechanizmus, amely képes valahonnan adatokat gyűjteni, és a listadefiníciónak megfelelően elkészíteni a listát.

Ez a mechanizmus lehet egy kész, általános célú listázó program, vagy egy külön erre a listadefinícióra készített C nyelvű program.

A C nyelvű program is különböző bonyolultságú lehet, függően attól, hogy milyen egyéb segédeszközöket használunk, illetve, hogy mennyire kívánjuk módosítani a standard listázási folyamatot.

A listázó C keretprogramok logikája és modulszerkezete egységes. A logikai menet pszeudo nyelven megfogalmazva:

```
PROGRAM lista
#
GLOBAL elemi_buffer, file azonosítók és bufferek
#
a listadefiníció megnyitása
az adatállományok megnyitása
az elemi_buffer kezdeti feltöltése
IF (a feltöltés sikertelen) THEN
        MESSAGE Üres kiválasztás
ELSE
        a lista inicializálása
        WHILE  (van elemi_buffer) DO
                lista végrehajtása (elemi_buffer-rel)
                az elemi_buffer feltöltése
        ENDWHILE
        a lista lezárása
ENDIF
az adatállományok lezárása
END
```

A logikai szerkezetnek megfelelő sematikus C program forrása:

```
/*
GLOBAL elemi_buffer, file azonosítók és bufferek
*/

main()
{
int     rc;
```

```
        r_read(..);
        open_files(..);
        rc = getfirstdetail(..);
        if (rc != SUCCESS)
                message("Üres kiválasztás");
        else
                {
                r_init(..);
                WHILE  (rc == SUCCESS)
                        {
                        r_exec(..);
                        rc = getnextdetail(..);
                        }
                r_total_print(0);
                r_close(..);
                }
        close_files(..);
        }
```

Az r_ karakterekkel kezdődő függvények a MIREP futtatórendszerét alkotó tárgykódkönyvtár moduljai:

r_read()	listadefiníció beolvasása,
r_init()	lista inicializálása,
r_exec()	listázás,
r_total_print(0)	mindösszesen listázása,
r_close()	lista lezárása.

Részletes ismertetésük a következő fejezetben található. A többi modult a programozónak kell megírnia.

A következőkben mintákat adunk listakészítő C programokra. Az első a Függelékek elején ismertetett példának lekérdező rendszertől független listázását valósítja meg.

A második példa ugyanennek a feladatnak a megoldása MISAM file- és MIQUEL lekérdező rendszerrel.

A harmadik példa egy MISAM file- és MIQUEL lekérdező rendszeren alapuló általános listázóprogram.

A megadott példák a teljes állományt listázzák. Az intervallum megadása és annak figyelése, hogy az elemi adatterület bizonyos feltételeknek eleget tesz-e, új modulokat és kiegészítéseket követel.

A példaprogramok forrásai a G. függelékben találhatók.

7.1. Listázó keretprogram

A G. Függelék 1. pontjában találálható TERM_ANY.C forrásprogram a Függelékek elején leírt feladat olyan megoldását tartalmazza, amely az adatbázis ASCII változatának listázását végzi el.

7.2. Lekérdezésen alapuló listázás

Az előző pontban ismertetett program legnagyobb része az elemi adatterület előállításával foglalkozik. Ezeket teljesen vagy majdnem teljesen automatizálhatjuk, ha a lekérdező rendszer (MIQUEL) moduljait használjuk.

A G. Függelék 2. pontjában találálható TERM_QRY.C forrásprogram ezt a változatot tartalmazza.

7.3. Egy általános listázóprogram

Funkciója:

Programozás nélkül lehetővé tesz listázásokat és lekérdezéseket. Megkötései a következők:

- MISAM file-okból vagy MIQUEL lekérdező rendszerrel megadott definíciók alapján gyűjti össze az adatokat,
- a standard listázási logikával dolgozik,
- számított mezők nincsenek.

A tapasztalatok szerint az adatfeldolgozási rendszerek listáinak 50 - 60 százaléka a standard listázóval kezelhető.

Hívása:

```
LIST -q <a lekérdezést leíró file neve>
     -f <a MISAM file leíró file neve>
     -r <a listadefiníciós file neve>
     -c <a szűrőfile>
     -o <output listafile vagy eszköz>
```

A -q és -f opciók egymást kölcsönösen kizárják.

A -f opciós listázás esetén, ha az adatállományhoz több indexfile is tartozik, akkor a program az első index szerint listáz.

A szűrőfile és az output megadása opcionális.

A listázót könnyen használhatjuk lekérdezésre is, ha a "-o CON" és "-c CONSOL" opciókkal a képernyőre kérjük a listát.

A standard listázó forrása (LIST.C) a G. függelék 3. pontjában található.

8. A MIREP futtató rendszere

A MIREP futtatórendszer C tárgykód könyvtárak formájában áll a felhasználó rendelkezésére. A tárgykód könyvtárak több C nyelvi implementációhoz és több memóriamodellhez léteznek.

A könyvtárakhoz tartozik a MIREP.H headerfile, amely tartalmazza a futtatórendszer rutinjai számára paraméterként átadandó, vagy azoktól visszatérési értékként kapott konstansok definícióját és az egyes könyvtári rutinok prototípusait.

A futtatórendszer rutinjait többféle sorrendben tárgyalhatjuk. Fontosság szerint előre lehet venni az r_read(), r_init(), r_d_setvalue(), r_exec(), r_total_print() és r_close() rutinokat, tekintettel arra, hogy a legtöbb esetben (az alapesetben) ezeket, és csak ezeket használjuk. A többi rutin vagy az opcióállítási trükkök, vagy pedig a csatlakozó C programok szempontjából fontos.

Célszerűbbnek tartottuk a modulok funkció szerinti csoportosítását. Itt a következő kategóriákat lehet felállítani:

- előkészítő rutinok,
- listajellemzőket lekérdező rutinok,
- listajellemzőket módosító rutinok,
- végrehajtó vagy a végrehajtást befolyásoló rutinok,
- a lista generálási állapotokat kezelő rutinok,
- adatlekérdező és módosító modulok,
- az elemi változók jellemzőit kezelő rutinok,
- csatlakozási pontokat kezelő modulok,
- az egyes formátumokat kezelő rutinok és
- egyéb, elsősorban a futtatórendszer belső használatára szolgáló rutinok.

Az egyes könyvtári függvények leírása tartalmazza:

- a rutin funkciójának összefoglalását,
- a használat szintaxisát,
- a rutin működésének leírását,
- a függvény visszatérési értékeinek magyarázatát és
- utalást az esetleges kapcsolódó függvényekre.

8.1. Előkészítő rutinok

A listák tényleges elkészítése előtt kerül sor a

- a listaleíró file beolvasására,
- a listázási opciók át-, illetve beállítására és

- a listázás inicializálására.

A következökben az e funkciókat megvalósító modulokat ismertetjük.

8.1.1. r_read()

Funkció — Egy listaleíró file beolvasása.

Használat
```
int r_read(fname)
    char   *fname;        A listaleíró file neve
```

Leírás — A függvény beolvassa a megadott nevü file-ból a listadefiníciót. Nemlétezö vagy beolvashatatlan file esetén a -1 visszatérési értékkel jelzi a hibát. Ha a leíró file hibás, akkor angol nyelvü hibaüzenet kiírása után abortáltatja a programot. Több lista egyidejü készítése esetén a függvény visszatérési értéke használható az "r_select()" függvény paramétereként a kívánt listadefiníció aktivizálásához.

Visszatérés
- -1 nemlétezö vagy beolvashatatlan leíró file esetén,
- n a beolvasott listadefiníció sorszáma 0-tól számozva.

Lásd még — r_select()

8.1.2. r_select()

Funkció — Több lista egyidejü készítésénél a megfelelö listadefiníció aktivizálása.

Használat
```
int r_select(sernum)
    int    sernum;        A kívánt lista sorszáma
```

Leírás — Több beolvasott listadefiníció közül a megadott sorszámút aktivizálja. A függvény sikeres végrehajtása után a megadott sorszámú lista lesz az aktuális. A sorszám a korábban meghívott "r_read()" függvény visszatérési értéke lehet.

Visszatérés
- -1 ha a megadott sorszámú listadefiníció nem került beolvasásra,
- sn az átadott sernum paraméter egyébként.

Lásd még r_read()

8.2. Listajellemzők lekérdezése

Ebben a fejezetben a listázás opcióit lekérdező és beállító rutinokat ismertetjük.

8.2.1. r_get_charsize()

Funkció A karakterméret lekérdezése.

Használat int r_get_charsize()

Leírás Visszaadja a beállított karakterméretet, amely lehet
- CS_NORMAL (10 karakter/hüvelyk),
- CS_ELITE (12 karakter/hüvelyk) vagy
- CS_CONDENS (15 karakter/hüvelyk)

(definíciójuk a MIREP.H headerfile-ban).

Visszatérés A karakterméret.

Lásd még r_set_charsize(),

8.2.2. r_get_class()

Funkció A hálózati nyomtató sorszámának lekérdezése.

Használat int r_get_class()

Leírás Visszaadja a hálózati nyomtató sorszámát.

Visszatérés A hálózati nyomtató sorszáma.

Lásd még r_set_class()

8.2.3. r_get_copynum()

Funkció A hálózati nyomtatás példányszámának lekérdezése.

Használat int r_get_copynum()

Leírás — Visszaadja a hálózati nyomtatás példányszámát.

Visszatérés — A hálózati nyomtatás példányszáma.

Lásd még — r_set_copynum()

8.2.4. r_get_datemode()

Funkció — A dátum formátum lekérdezése.

Használat — int r_get_datemode()

Leírás — Visszaadja a dátum formátumát, amely lehet:
- DM_ENGLISH (mm/dd/yy),
- DM_HUNGARY (yy/mm/dd) vagy
- DM_GERMAN (dd/mm/yy)

(definíciójuk a MIREP.H headerfile-ban).

Visszatérés — A dátum formátuma.

Lásd még — r_set_datemode(), r_convdat(), r_atodat()

8.2.5. r_get_device()

Funkció — Az output eszköz lekérdezése.

Használat — int r_get_device()

Leírás — Visszaadja a beállított output eszközt, amely lehet
GLOBALPRINTER,
LOCALPRINTER,
CONSOL,
DISKFILE
(definíciójuk a MIREP.H headerfile-ban).

Visszatérés — Az output eszköz.

Lásd még — r_set_device(), r_redirect()

8.2.6. r_get_dscr()

Funkció — Visszaadja a listadefiníció leírását.

Használat — char *r_get_dscr()

Leírás Visszaadja a listadefiníció leírását.

Visszatérés A listadefiníció leírása.

Lásd még r_set_dscr()

8.2.7. r_get_formf()

Funkció A FormFeed (lapemelés) karakter használatának lekérdezése.

Használat int r_get_formf()

Leírás Visszaadja, hogy új lap kezdetekor a listában lesz-e FormFeed (lapemelés) karakter.

Visszatérés
1 ha új lap kezdetekor a listában lesz FormFeed karakter,
0 egyébként (a oldalak üres sorokkal töltődnek fel).

Lásd még r_set_formf(), r_get_frstff(), r_get_lastff()

8.2.8. r_get_frstff()

Funkció A legelső FormFeed (lapemelés) használatának lekérdezése.

Használat int r_get_frstff()

Leírás Visszaadja, hogy a legelső lap nyomtatása előtt lesz-e a listában FormFeed (lapemelés) karakter.

Visszatérés
1 ha a legelső lap nyomtatása előtt lesz a listában FormFeed (lapemelés) karakter,
0 egyébként.

Lásd még r_set_frstff(), r_get_formf()

8.2.9. r_get_lastff()

Funkció A legutolsó FormFeed (lapemelés) használatának lekérdezése.

Használat int r_get_lastff()

Leírás — Visszaadja, hogy a legutolsó lap nyomtatása után lesz-e lapemelés.

Visszatérés
1 ha a legutolsó lap nyomtatása után lesz a listában FormFeed karakter,
0 egyébként.

Lásd még — r_get_lastff(), r_set_frstff()

8.2.10. r_get_linespacing()

Funkció — A sortáv lekérdezése.

Használat — int r_get_linespacing()

Leírás — Visszaadja az aktuális sortávot, amely lehet:
- LS_NORMAL (1/6 hüvelyk),
- LS_HALF (1/8 hüvelyk) vagy
- LS_WITHOUT (7/72 hüvelyk)

(definíciójuk a MIREP.H headerfile-ban).

Visszatérés — Az aktuális sortáv.

Lásd még — r_set_linespacing(), r_get_pagelength()

8.2.11. r_get_lstfdir()

Funkció — Az output listafile könyvtárának lekérdezése.

Használat — char *r_get_lstfdir()

Leírás — Visszaadja az output listafile könyvtárának a nevét. Ha ez az alapértelmezés szerinti aktuális könyvtár, akkor a visszatérési érték NULL.

Visszatérés — Az output listafile neve.

Lásd még — r_set_lstfname(), r_set_lstfdir()

8.2.12. r_get_lstfname()

Funkció — Az output listafile nevének lekérdezése.

Használat — char *r_get_lstfname()

Leírás Visszaadja az output listafile nevét.

Visszatérés Az output listafile neve.

Lásd még r_set_lstfname(), r_redirect()

8.2.13. r_get_name()

Funkció Visszaadja a listadefiníció nevét.

Használat char *r_get_name()

Leírás Visszaadja a listadefiníció nevét.

Visszatérés A listadefiníció neve.

Lásd még r_set_name()

8.2.14. r_get_openmode()

Funkció Az output listafile megnyitási módjának lekérdezése.

Használat int r_get_openmode()

Leírás Visszaadja az output listafile megnyitási módját, amely lehet:
- OM_WRITE (felülírás) vagy
- OM_APPEND (hozzáírás)

(definíciójuk a MIREP.H headerfile-ban).

Visszatérés Az output listafile megnyitási módja.

Lásd még r_set_openmode()

8.2.15. r_get_pagelength()

Funkció Az aktuális laphossz lekérdezése.

Használat int r_get_pagelength()

Leírás Visszaadja az aktuális laphosszt.

Visszatérés Az aktuális laphossz.

Lásd még r_set_pagelength(), r_get_linespacing()

8.2.16. r_get_prntype()

Funkció A nyomtató típusának, illetve a szűrőfile nevének lekérdezése.

Használat char *r_get_prntype()

Leírás Visszaadja a beállított nyomtatótípust, azaz a szűrőfile nevét.

Visszatérés A nyomtató típusa.

Lásd még r_set_prntype()

8.2.17. r_get_server()

Funkció A hálózati nyomtató file-server nevének lekérdezése.

Használat char *r_get_server()

Leírás Visszaadja a hálózati nyomtató file-server nevét.

Visszatérés A hálózati nyomtató file-server neve.

Lásd még r_set_server()

8.2.18. r_get_sorted()

Funkció A rendezettségi információ lekérdezése.

Használat char *r_get_sorted()

Leírás Visszaadja a rendezettségi információt.

Visszatérés A rendezettségi információ.

Lásd még r_set_sorted()

8.3. Listajellemzők beállítása

Ebben a fejezetben azokat a függvényeket ismertetjük, amelyek segítségével a lista egy-egy lapjának a formáját befolyásoló opciók kerdezhetők le, illetve állíthatók át.

8.3.1. r_redirect()

Funkció: A lista intelligens átirányítása.

Használat:
```
int r_redirect(output)
    char   *output;  Az output file/eszköz neve
```

Leírás: Az átadott paraméternek megfelelően átirányítja az output-ot. A paraméter lehetséges értékei:
- GLOBALPRINTER
  A listát a hálózati nyomtatóra kérjük,
- LOCALPRINTER
- PRN
  A listát a lokális nyomtatóra kérjük,
- CONSOLE
- CON
  A listát a képernyőre kérjük, (interaktív mód, 24 soros oldalak)
- egyéb
  A listát a megadott nevű könyvtárba és file-ba kérjük.

Visszatérés: Nincs.

Lásd még: r_set_device(), r_set_lstfdir(), r_set_lstfname()

8.3.2. r_set_charsize()

Funkció: A karakterméret beállítása a listázás megkezdése előtt.

Használat:
```
int r_set_charsize(cs)
    int   cs;            A karakterméret
```

Leírás: Beállítja a karakterméretet a megadottra, amely lehet
- CS_NORMAL (10 karakter/hüvelyk),
- CS_ELITE (12 karakter/hüvelyk) vagy
- CS_CONDENS (12 karakter/hüvelyk)

(definíciójuk a MIREP.H headerfile-ban).

Visszatérés -1 hibás paraméter esetén,
0 egyébként.

Lásd még r_get_charsize()

8.3.3. r_set_class()

Funkció A hálózati nyomtató sorszámának beállítása.

Használat
```
int r_set_class(class)
    int   class;          A nyomtató sorszáma
```

Leírás Beállítja a hálózati nyomtató sorszámát a megadottra.

Visszatérés Nincs.

Lásd még r_get_class()

8.3.4. r_set_copynum()

Funkció A hálózati nyomtatás példányszámának beállítása.

Használat
```
int r_set_copynum(copynum)
    int copynum;          A másolatok száma
```

Leírás Beállítja a hálózati nyomtatás példányszámát a megadottra.

Visszatérés Nincs.

Lásd még r_get_copynum()

8.3.5. r_set_datemode()

Funkció A dátum formátum beállítása.

Használat
```
int r_set_datemode(dmode)
    int   dmode;          A dátum formátuma
```

Leírás — Beállítja a dátum formátumát a megadottra, amely lehet:
- DM_ENGLISH (mm/dd/yy),
- DM_HUNGARY (yy/mm/dd) vagy
- DM_GERMAN (dd/mm/yy)

(definíciójuk a MIREP.H headerfile-ban).

Visszatérés — Nincs.

Lásd még — r_get_datemode(), r_convdat(), r_atodat()

8.3.6. r_set_device()

Funkció — Az output eszköz beállítása.

Használat
```
int r_set_device(device)
    int   device;        A kívánt output eszköz
```

Leírás — Beállítja az output eszközt. Az eszköz a következők valamelyike lehet:
```
GLOBALPRINTER,
LOCALPRINTER,
CONSOL,
DISKFILE
```
(definíciójuk a MIREP.H headerfile-ban).

Visszatérés
```
-1  hibás paraméter esetén,
 0  egyébként.
```

Lásd még — r_get_device(), r_redirect()

8.3.7. r_set_dscr()

Funkció — A listadefiníció leírásának megváltoztatása.

Használat
```
int r_set_dscr(dscr)
    char *dscr;          Az új leírás
```

Leírás — Beállítja a listadefiníció leírását a megadottra.

Visszatérés — Nincs.

Lásd még — r_get_dscr()

8.3.8. r_set_formf()

Funkció: A FormFeed (lapemelés) karakter használatának beállítása.

Használat:
```
int r_set_formff(onoff)
    int   onoff;          Be- vagy kikapcsolás
```

Leírás: Be- vagy kikapcsolja új lap kezdetekor a FormFeed karakter használatát.

Visszatérés: Nincs.

Lásd még: r_get_formf(), r_set_frstff(), r_set_lastff()

8.3.9. r_set_frstff()

Funkció: A legelső FormFeed (lapemelés) karakter használatának beállítása.

Használat:
```
int r_set_frstff(onoff)
    int   onoff;          Be- vagy kikapcsolás
```

Leírás: Beállítja, hogy a legelső lap nyomtatása előtt legyen-e a listában FormFeed karakter (onoff=1) vagy sem (onoff=0).

Visszatérés: Nincs.

Lásd még: r_get_frstff(), r_set_lastff()

8.3.10. r_set_lastff()

Funkció: A legutolsó FormFeed (lapemelés) használatának beállítása.

Használat:
```
int r_set_lastff(onoff)
    int   onoff;          Be- vagy kikapcsolás
```

Leírás: Beállítja, hogy a legutolsó lap nyomtatása után lesz-e a listában FormFeed karakter (onoff=1) vagy sem (onoff=0).

Visszatérés: Nincs.

Lásd még: r_set_lastff(), r_get_frstff()

8.3.11. r_set_linespacing()

Funkció — A sortáv beállítása a listázás megkezdése előtt.

Használat
```
int r_set_linesapcing(ls)
    int    ls;          A kért sortáv
```

Leírás — Beállítja a sortávot a megadottra, amely lehet:
- LS_NORMAL (1/6 hüvelyk),
- LS_HALF (1/8 hüvelyk) vagy
- LS_WITHOUT (7/72 hüvelyk)

(definíciójuk a MIREP.H headerfile-ban).

Visszatérés
-1 hibás paraméter esetén,
0 egyébként.

Lásd még — r_get_linespacing(), r_set_pagelength()

8.3.12. r_set_lstfdir()

Funkció — Beállítja az output listafile könyvtárának nevét.

Használat
```
int r_set_lstfdir(dir)
    char *dir;          A könyvtár neve
```

Leírás — Beállítja az output listafile könyvtárának a nevét a megadottra.

Visszatérés — Nincs.

Lásd még — r_get_lstfdir(), r_set_lstfname()

8.3.13. r_set_lstfname()

Funkció — Az output listafile nevének beállítása.

Használat
```
int r_set_lstfname(fname)
    char *fname;        A listafile neve
```

Leírás — Beállítja az output listafile nevét a megadottra.

Visszatérés — Nincs.

Lásd még — r_get_lstfname()

8.3.14. r_set_name()

Funkció — A listadefiníció nevének megváltoztatása.

Használat
```
int r_set_name(name)
    char *name;          Az új név
```

Leírás — Beállítja a listadefiníció nevét a megadottra.

Visszatérés — Nincs.

Lásd még — r_get_name()

8.3.15. r_set_openmode()

Funkció — Az output listafile megnyitási módjának beállítása.

Használat
```
int r_set_openmode(omode)
    int    omode;        A megnyitás módja
```

Leírás — Beállítja az output listafile megnyitási módját, amely lehet:
- OM_WRITE (felülírás) vagy
- OM_APPEND (hozzáírás)

(definíciójuk a MIREP.H headerfile-ban).

Visszatérés — Nincs.

Lásd még — r_get_openmode()

8.3.16. r_set_pagelength()

Funkció — A laphossz beállítása.

Használat
```
int r_set_pagelength(pl)
    int pl;              A kívánt laphossz
```

Leírás — Beállítja a laphosszt a megadottra.

Visszatérés — Nincs.

Lásd még — r_get_pagelength(), r_set_linespacing()

8.3.17. r_set_prntype()

Funkció — A nyomtató típusának, illetve a szürőfile nevének beállítása.

Használat
```
int r_set_prntype(type)
    char *type;          A nyomtató típusa
```

Leírás — Beállítja a nyomtató típusát, illetve a szürőfile nevét a megadottra.

Visszatérés — Nincs.

Lásd még — r_get_prntype()

8.3.18. r_set_server()

Funkció — A hálózati nyomtató file-server nevének beállítása.

Használat
```
int r_set_server(server)
    char *server;        A file-server neve
```

Leírás — Beállítja a hálózati nyomtató file-server nevét a megadottra.

Visszatérés — Nincs.

Lásd még — r_get_server()

8.3.19. r_set_sorted()

Funkció — A rendezettségi információ megváltoztatása.

Használat
```
int r_set_sorted(sorted)
    char *sorted;                Az új információ
```

Leírás — Beállítja a rendezettségi információt a megadottra.

Visszatérés — Nincs.

Lásd még — r_get_sorted()

8.4. Végrehajtó rutinok

A listadefiníció beolvasása, a lista inicializálása és a lista lezárása között az érdemi listázási részt általában egyetlen függvény, az "r_exec()" végzi. Különleges igények esetén a listagenerálás folyamatába számos módon beavatkozhatunk a következőkben ismertetett függvények segítségével.

8.4.1. r_charsize()

Funkció — Az aktuális karakterméret megváltoztatása.

Használat
```
int r_charsize(charsize)
   int charsize;          Az új karakterméret
```

Leírás — Megváltoztatja a karakterszélességet a megadottra a listagenerálás közben.

A karakterméret a következők valamelyike lehet:
CS_NORMAL,
CS_ELITE,
CS_CONDENS
(definíciójuk a MIREP.H headerfile-ban).

Visszatérés
-1 hibás paraméter esetén,
0 egyébként.

Lásd még — r_set_charsize(), r_get_charsize()

8.4.2. r_close()

Funkció — Az aktuális lista lezárása.

Használat — int r_close()

Leírás — Elvégzi az aktuális lista lezárásával kapcsolatos összes funkciót (pl. utolsó lap emelése, output lezárása, lefoglalt memóriaterület felszabadítása stb).

Visszatérés — Nincs.

Lásd még — r_read(), r_init()

8.4.3. r_detail_header()

Funkció — Elemisorhoz tartozó fejléc kiíratása.

Használat
```
int r_detail_header(sernum)
    int  sernum;        A fejlécsor sorszáma
```

Leírás — Kiíratja a megadott sorszámú fejlécsort lapemelés nélkül akkor, ha nem összegsor következik.

Visszatérés
- 0 hibátlan végrehajtás esetén,
- n jelzőkód egyébként, amely
  - -1 ha a listát a képernyőre átirányítottuk, és az operátor a lapváltáskor az ESC billentyű leütésével jelezte, hogy a listázás folytatása szükségtelen,
  - n a felhasználói C függvény visszatérési értéke, ha az nem 0.

Lásd még — r_total_header(), r_header_print()

8.4.4. r_detail_print()

Funkció — Az elemisor explicit kiíratása.

Használat
```
int r_detail_print(sernum)
    int  sernum;        Az elemisor sorszáma
```

Leírás — Kiíratja a megadott sorszámú elemisort. Az aktuális elemisor kiírása az "r_exec()" függvény hívásakor automatikus, kivéve ha
- az "r_set_detprint()" függvénnyel letiltottuk vagy
- az elemisor "Before" C függvénye 0-tól eltérő értékkel tért vissza.

Az r_detail_print() függvény végrehajtásakor az egyébként automatikus kontrollváltás, összegzés és számlálás elmarad.

Visszatérés
- 0 hibátlan végrehajtás esetén,
- n jelzőkód egyébként, amely
  - -1 ha a listát a képernyőre átirányítottuk, és az operátor a lapváltáskor az ESC billentyű leütésével jelezte, hogy a listázás folytatása szükségtelen,
  - n a felhasználói C függvény visszatérési értéke, ha az nem 0.

Lásd még r_set_detprint()

8.4.5. r_exec()

Funkció A lista generálása.

Használat int r_exec()

Leírás A feltöltött elemi adatbuffer alapján végzi a listázást. Ezen belül automatikusan kezeli
- az elemisor kiíratását,
- a lapváltást és a fejlécezést,
- a kontrollváltásokat,
- az összegzéseket,
- a számlálásokat,
- az átlagképzéseket.

Visszatérés
0 hibátlan végrehajtás esetén,
n jelzőkód egyébként, amely
-1 ha a listát a képernyőre átirányítottuk, és az operátor a lapváltáskor az ESC billentyű leütésével jelezte, hogy a listázás folytatása szükségtelen,
n a felhasználói C függvény visszatérési értéke, ha az nem 0.

Lásd még r_d_set_value(), r_set_first(), r_get_first(), r_get_jump(), r_set_jump()

8.4.6. r_fill_page()

Funkció Az aktuális lap kitöltése üres sorokkal.

Használat int r_fill_page()

Leírás Kitölti az aktuális lapot üres sorokkal, tehát a következő kiírás új lapon kezdődik majd.

Visszatérés Nincs.

Lásd még r_new_page()

8.4.7. r_header_print()

Funkció: Fejlécsor explicit kiíratása.

Használat:
```
int r_header_print(sernum)
    int   sernum;        A fejlécsor sorszáma
```

Leírás: Kiíratja a megadott sorszámú fejlécsort lapemelés nélkül.

Visszatérés:
-1 ha a megadott sorszámú fejlécsor nem létezik,
0 egyébként.

Lásd még: r_detail_header(), r_total_header()

8.4.8. r_init()

Funkció: A lista inicializálása.

Használat:
```
int r_init(ptcdir)
    char   *ptcdir;      A szürőfile-ok könyvtára
```

Leírás: Elvégzi a listázás megkezdéséhez szükséges összes inicializálást (pl. output megnyitása, buffer allokáció, szürőfile beolvasása stb).
A 'ptcdir' paraméter a kívánt szürőfile könyvtárának a neve, amelyre akkor van szükség, ha az nem az aktuális, illetve a SET PTC= környezettel megadott könyvtárban található. Ha a kívánt szürőfile az aktuális vagy a PTC= környezettel megadott könyvtárban található, akkor a paraméter értéke NULL.

Visszatérés:
0 hibátlan inicializálás esetén,
n a hiba kódja egyébként, amely lehet:
SPOOL_ERR
Spool megnyitási hiba hálózati nyomtató esetén,
OUTPUT_OPEN
Az output listafile vagy eszköz nem nyitható meg írásra,
GETCONV_ERR
A szürőfile nem olvasható be.
(definíciójuk a MIREP.H headerfile-ban).

Lásd még: r_redirect()

8.4.9. r_linespacing()

Funkció — Az aktuális sortáv megváltoztatása.

Használat
```
int r_linespacing(linespacing)
   int linespacing;      Az új sortáv
```

Leírás — A sortávot a megadottra állítja a listagenerálás közben. A sortáv a következők valamelyike lehet:
LS_NORMAL,
LS_HALF,
LS_WITHOUT
(definíciójuk a MIREP.H headerfile-ban).

Visszatérés
-1 hibás paraméter esetén,
0 egyébként.

Lásd még — r_set_linespacing(), r_get_linespacing()

8.4.10. r_new_page()

Funkció — Lapemelés kérése.

Használat — int r_new_page()

Leírás — A függvény végrehajtásának hatására a következő kiírás új lapon kezdődik.

Visszatérés — Nincs.

Lásd még — r_fill_page()

8.4.11. r_printer_ctr()

Funkció — Nyomtatófüggő vezérlés.

Használat
```
int r_printer_ctr(ctr)
   int   ctr;          A nyomtatóvezérlő parancs
```

Leírás — Végrahajtja a megadott nyomtatóvezérlést. A 'ctr' paraméter a következők valamelyike lehet:

PC_UNDERON : aláhúzás bekacsolása,
PC_UNDEROFF : aláhúzás kikacsolása,
PC_DOUBLEON : duplán írás bekapcsolása,
PC_DOUBLEOFF: duplán írás kikapcsolása,

PC_ITALICON : dőlt betűs írás bekapcsolása,
PC_ITALICOFF: dőlt betűs írás kikapcsolása,
PC_USER0 : 1. speciális vezérlés,
PC_USER1 : 2. speciális vezérlés,
PC_USER2 : 3. speciális vezérlés,
PC_USER3 : 4. speciális vezérlés,
PC_USER4 : 5. speciális vezérlés,
PC_USER5 : 6. speciális vezérlés,
PC_USER6 : 7. speciális vezérlés,
PC_USER7 : 8. speciális vezérlés,
PC_USER8 : 9. speciális vezérlés,
PC_USER9 : 10. speciális vezérlés.

Az egyes parancsoknak megfelelő konstansok definíciója a <MIREP.H> headerfile-ban található.

Visszatérés
-1 hibás paraméter esetén,
0 egyébként.

8.4.12. r_total_header()

Funkció
Összegsorhoz tartozó fejlécsor feltételes kiíratása.

Használat
```
int r_total_header(sernum)
    int    sernum;      A fejlécsor sorszáma
```

Leírás
Lapemelés nélkül kiíratja a megadott sorszámú fejlécsort, ha az még kifér az aktuális lapra és nem összegsor következik.

Visszatérés
0 hibátlan végrehajtás esetén,
n jelzőkód egyébként, amely
-1 ha a listát a képernyőre átirányítottuk, és az operátor a lapváltáskor az ESC billentyű leütésével jelezte, hogy a listázás folytatása szükségtelen,
n a felhasználói C függvény visszatérési értéke, ha az nem 0.

Lásd még
r_detail_header(), r_header_print()

8.4.13. r_total_init()

Funkció
Az összegzők nullázása.

Használat

```
int r_total_init(level)
    int    level;        Az összegfokozat szintje
```

Leírás

A rutin segítségével kikényszeríthető a megadott és az annál nagyobb összegfokozatok gyüjtőinek nullázása. A megfelelő összegfokozatok nullázása kontrollváltás esetén automatikus!

Visszatérés

Nincs.

Lásd még

total_print()

8.4.14. r_total_page()

Funkció

Összegsor kiírása után feltételes lapdobás.

Használat

int r_total_page()

Leírás

A rutin hatására az összegsor formátum kiírása után egy új lap kezdődik, de csak akkor, ha nem követni azonnal egy alacsonyabb szintü összegfokozat kiírása. Ezzel a módszerrel elkerülhető, hogy az egyidejüleg több szintü kontrollváltáskor az alacsonyabb szintü összegsorok feleslegesen új lapon kezdődjenek.

Visszatérés

Nincs.

Lásd még

r_total_print()

8.4.15. r_total_print()

Funkció

Összegsor(ok) kiíratása.

Használat

```
int r_total_print(level)
    int    level;        Az összegfokozat szintje
```

Leírás

A rutin segítségével kiírathatjuk a megadott összegző fokozatig az összes összegsort. Kontrollváltáskor a megfelelő szintig ez automatikus. A listázás végén az utolsó elemisor kiíratása után, ha a mindösszesen összegsor (Grand Total) létezik, és erre a listában is szükségünk van, akkor a lista lezárása előtt meg kell hívni a rutint a 0 paraméterrel!

Visszatérés
0 hibátlan végrehajtás esetén,
n jelzőkód egyébként, amely
-1, ha a listát a képernyőre átirányítottuk, és az operátor a lapváltáskor az ESC billentyű leütésével jelezte, hogy a listázás folytatása szükségtelen,
n a felhasználói C függvény visszatérési értéke, ha az nem 0.

Lásd még r_total_init(), r_total_page(), r_total_header(), r_total_onoff()

8.5. Állapototkezelő rutinok

A listadefinícióban a lista helyére vonatkozó információkat szituációtól függően viszonylag gyakran kell változtatni (pl. ugyanazt a listadefiníciót akarjuk használni képernyős lekérdezéshez és lokális nyomtatón való nyomtatáshoz). A lista helyének módosítására szolgálnak a következőkben ismertetett függvények.

8.5.1. r_act_detail()

Funkció Az aktuális elemisor formátum sorszámának visszaadása.

Használat int r_act_detail()

Leírás Visszaadja, hogy a definiált elemisor formátumok közül melyik a kiválasztott.

Visszatérés
-1 ha nincs kiválasztott elemisor formátum,
n az aktív elemisor formátum sorszáma egyébként.

Lásd még r_detail_select()

8.5.2. r_act_header()

Funkció Az aktuális fejlécsor formátum sorszámának visszaadása.

Használat int r_act_header()

Leírás — Visszaadja, hogy a definiált fejlécsor formátumok közül melyik a kiválasztott.

Visszatérés
-1 ha nincs kiválasztott fejlécsor formátum,
n az aktív fejlécsor formátum sorszáma egyébként.

Lásd még — r_header_select()

8.5.3. r_act_total()

Funkció — Az aktuális összegsor formátum sorszámának visszaadása.

Használat — int r_act_total()

Leírás — Visszaadja, hogy az összegsorok közül melyik az aktív.

Visszatérés
-1 ha nincs kontrollváltás
n az aktív összegsor formátum sorszáma egyébként.

Lásd még — r_total_onoff()

8.5.4. r_get_first()

Funkció — A listázás fázisának lekérdezése.

Használat — int r_get_first()

Leírás — A függvény hívásának a felhasználói C függvényekben van értelme. A függvény visszaadja, hogy az "r_exec()" függvény legelső végrehajtása van-e folyamatban.

Visszatérés
1 ha az "r_exec()" függvény legelső végrehajtása van folyamatban,
0 egyébként.

Lásd még — r_exec(), r_set_first()

MAGYAR TUDOMÁNYOS AKADÉMIA KÖNYVTÁRA

8.5.5. r_get_jump()

Funkció — Formátum kiírás letiltásának lekérdezése.

Használat — int r_get_jump()

Leírás — Visszaadja, hogy a következő kiírandó formátum kiírása tiltva van-e vagy sem. A letiltás csak az "r_set_jump()" rutin felhasználói C függvényből való hívásával végezhető el.

Visszatérés — 1 ha a kiírás az "r_set_jump()" függvény hívásával tiltva van,
0 egyébként.

Lásd még — r_set_jump()

8.5.6. r_get_printfp()

Funkció — Az output listafile-mutató lekérdezése.

Használat — void *r_get_printfp()

Leírás — Visszaadja az output listafile-mutatóját.

Visszatérés — Az output listafile-mutatója.

Lásd még — r_set_printfp()

8.5.7. r_globprn_status()

Funkció — Visszaadja a hálózati nyomtató státuszát.

Használat — int r_globprn_status(int class)
int class; A hálózati nyomtató "queue" száma

Leírás — Visszaadja a megadott osztályú hálózati nyomtató státuszát.

Visszatérés — -1 ha az adott hálózati nyomtatóról nincs információ,
-2 ha az adott hálózati nyomtató nem fogadóképes,
0 egyébként.

Lásd még r_set_device(), r_set_server(), r_set_class(), r_redirect()

8.5.8. r_set_first()

Funkció Az "r_exec()" függvény elsöként való végrehajtásának kikényszerítése vagy letiltása.

Használat
```
int r_set_first(onoff)
    int  onoff;          Be- vagy kikapcsolás
```

Leírás A futtatórendszer számára meghatározza, hogy a következö "r_exec()" végrehajtás elsöként értelmezödjön-e vagy sem. Az elsöként való végrehajtás esetén a kontrollváltás vizsgálata elmarad.

Visszatérés Nincs.

Lásd még r_get_first()

8.5.9. r_set_jump()

Funkció A következö formátum kiírásának letiltása vagy engedélyezése.

Használat
```
int r_set_jump(onoff)
    int  onoff;   Letiltás vagy engedélyezés
```

Leírás Felhasználói C függvényben hívható rutin, amely az aktuális formátum kiírását tiltja le (vagy engedélyezi).

Visszatérés Nincs.

Lásd még r_get_jump()

8.5.10. r_set_printfp()

Funkció Az output listafile-mutató átállítása.

Használat
```
int r_set_printfp(fp)
    void *fp;   Az output listafile-mutatója
```

Leírás Beállítja az output listafile-mutatóját a megadottra.

Visszatérés Nincs.

Lásd még r_get_printfp()

8.5.11. r_status()

Funkció A listagenerálási állapot lekérdezése.

Használat
```
int r_status(act)
    int *act;    Az aktuális sorszám
```

Leírás Visszaadja, hogy a listagenerálás során milyen típusú és milyen sorszámú formátum kiíratása van folyamatban. A típus a következők valamelyike:

F_HEADER: fejlécsor formátum,
F_DETAIL: elemisor formátum,
F_TOTAL : összegsor formátum.

Ha a sorszámra nincs szükség, akkor az 'act' paraméter lehet NULL. A függvény hívásának akkor van értelme, ha az őt hívó felhasználói C függvény több funkció ellátására képes a listagenerálási állapottól függően, és az esetleges mélyebb modulbeágyazás miatt a futtatórendszertől kapott paraméterek nem hozzáférhetők.

Visszatérés Az éppen kiírás alatt lévő formátum típusa.

Lásd még r_act_header(), r_act_detail(), r_act_total()

8.6. Adatlekérdező és módosító rutinok

A következő rutinokkal a lista adattartalmát kédezhetjük le, illetve módosíthatjuk.

8.6.1. r_atodat()

Funkció Karakteres dátum átalakítása belső ábrázolású formába.

Használat
```
long r_atodat(dstr)
    char *dstr;          A dátum stringje
```

Leírás Az átadott karakteres dátum konverziója tömörített, belső ábrázolású formába. A karakteres

dátum értelmezése a beállított dátum formától függ.

Visszatérés — A dátum tömörített formájú ertéke.

Lásd még — r_set_datemode(), r_convdat()

8.6.2. r_convdat()

Funkció — Belső ábrázolású dátum átalakítása karakteres formába.

Használat
```
char *r_convdat(dl)
    long  ld;            A dátum tömörített értéke
```

Leírás — Az átadott tömörített, belső ábrázolású dátumot karakteres formába konvertálja, és visszaadja egy statikus karaktertömbben. A karakteres dátum formája a beállított dátum formától függ.

Visszatérés — A karakteres dátumot tartalmazó statikus karaktertömb címe.

Lásd még — r_set_datemode(), r_atodat()

8.6.3. r_d_all_valued()

Funkció — Az összes elemi változó feltöltöttségének jelzése.

Használat
```
int r_d_all_valued(onoff)
    int onoff;           Be-, kikapcsolás
```

Leírás — Jelzi a futtatórendszer számára, hogy az összes elemi változó kapott (onoff = 1), vagy nem kapott értéket (onoff = 0).

Visszatérés — Nincs.

Lásd még — r_d_valued(), r_d_is_valued(), r_d_set_value()

8.6.4. r_d_buffer()

Funkció — Az elemi adatterület bufferének lekérdezése.

Használat char *r_d_buffer()

Leírás Visszaadja az elemi adatterület számára automatikusan allokált memóriaterület kezdőcímét.

Visszatérés Az elemi adatterület bufferének címe.

Lásd még r_d_buflen()

8.6.5. r_buflen()

Funkció Az elemi adatterület méretének lekérdezése.

Használat int r_d_buflen()

Leírás Visszaadja az elemi adatterület méretét byte-ban.

Visszatérés Az elemi adatterület mérete.

Lásd még r_d_buffer()

8.6.6. r_d_get_value()

Funkció Elemi változó aktuális értékenek lekérdezése.

Használat
```
int r_d_get_value(sernum,outbuf)
    int   sernum;        Az elemi változó sorszáma
    char *outbuf;        Az output buffer
```

Leírás A megadott sorszámú elemi változó értékét betölti a második paraméterként megadott bufferbe. Az output buffer méretének elegendőnek kell lennie az érték tárolásához.

Visszatérés
-1 ha a megadott sorszámú elemi változó nem létezik, vagy korábban nem kapott értéket,
0 egyébként.

Lásd még r_d_set_value()

8.6.7. r_d_is_valued()

Funkció Az elemi változó feltöltöttségének lekérdezése.

Használat

```
int r_d_is_valued(sernum)
    int sernum;          Az elemi változó sorszáma
```

Leírás

Visszaadja, hogy a megadott elemi változó kapott-e valamilyen módon értéket.

Visszatérés

0 ha az elemi változó még nem kapott értéket,
1 egyébként.

Lásd még

r_d_set_value(), r_d_valued(), r_d_all_valued()

8.6.8. r_d_set_value()

Funkció

Elemi változó értékének beállítása.

Használat

```
int r_d_set_value(sernum,inpbuf)
    int   sernum;        Az elemi változó sorszáma
    char *inpbuf;        Az input buffer
```

Leírás

A megadott sorszámú elemi változó értékét betölti a második paraméterként megadott bufferből.

Visszatérés

-1 ha a megadott sorszámú elemi változó nem létezik,
0 egyébként.

Lásd még

r_d_get_value()

8.6.9. r_d_valued()

Funkció

Elemi változó feltöltöttségének jelölése.

Használat

```
int r_d_valued(sernum,onoff)
    int   sernum;        Az elemi változó sorszáma
    int   onoff;         Be- vagy kikapcsolás
```

Leírás

Jelzi a futtató rendszer számára, hogy az elemi változó kapott-e értéket. Ha egy elemi változónak az "r_d_set_value()" függvénnyel értéket adunk, akkor az mindaddig "értékkel rendelkező", amíg az "r_d_valued()" függvénnyel ezt meg nem változtatjuk.

Visszatérés

-1 ha a megadott sorszámú elemi változó nem létezik,
0 egyébként.

Lásd még r_d_set_value()

8.6.10. r_get_count()

Funkció Számláló lekérdezése.

Használat
```
int r_get_count(level,pcount)
    int   level;        Az összegző szintje
    long  *pcount;      Az output érték változója
```

Leírás A megadott szintű számláló értékét betölti az output változóba.

Visszatérés
-1 ha a megadott szintű összegsor nem létezik,
0 egyébként.

Lásd még r_set_count()

8.6.11. r_get_total()

Funkció Összegző lekérdezése.

Használat
```
int r_get_total(level,sernum,ptotal)
    int   level;        Az összegző szintje
    int   sernum;       Az ö. elemi v. sorszáma
    double *ptotal;     Az output érték változója
```

Leírás A megadott szintű és sorszámú összegző értékét betölti az output változóba.

Visszatérés
-1 ha a megadott szintű összegsor nem létezik, vagy ha a megadott sorszámú elemi változó nem létezik, vagy ha a megadott sorszámú elemi változó nem összegzett változó,
0 egyébként.

Lásd még r_set_total()

8.6.12. r_o_get_value()

Funkció Elemi változó előző értékének lekérdezése.

Használat
```
int r_o_get_value(sernum,outbuf)
    int   sernum;       Az elemi változó sorszáma
    char *outbuf;       Az output buffer
```

Leírás

A megadott sorszámú elemi változó előző értékét betölti a második paraméterként megadott bufferbe. Az output buffer méretének elegendőnek kell lennie az érték tárolásához.

Visszatérés

-1 ha a megadott sorszámú elemi változó nem létezik,
0 egyébként.

Lásd még

r_d_get_value()

8.6.13. r_set_count()

Funkció

Számláló értékének beállítása.

Használat

```
int r_set_count(level,pcount)
    int   level;        Az összegző szintje
    long  *pcount;      Az input érték változója
```

Leírás

A megadott szintű számláló értékét betölti az input változóból.

Visszatérés

-1 ha a megadott szintű összegsor nem létezik,
0 egyébként.

Lásd még

r_get_count()

8.6.14. r_set_total()

Funkció

Összegző értékének beállítása.

Használat

```
int r_set_total(level,sernum,ptotal)
    int   level;        Az összegző szintje
    int   sernum;       Az ö. elemi v. sorszáma
    double *ptotal;     Az input érték változója
```

Leírás

A megadott szintű és megadott sorszámú összegző értékét betölti az input változóból.

Visszatérés

-1 ha a megadott szintű összegsor nem létezik, vagy ha a megadott sorszámú elemi változó nem létezik, vagy ha a megadott sorszámú elemi változó nem összegzett változó,
0 egyébként.

Lásd még

r_get_total()

8.6.15. r_sysdate()

Funkció — A számítógépben beállított dátum lekérdezése.

Használat — long r_sysdate()

Leírás — Visszaadja a számítógépben beállított dátumot tömörített, belső formában.

Visszatérés — A gépi dátum tömörített formában.

Lásd még — r_convdat(), r_atodat(), r_s_set_value()

8.6.16. r_s_get_value()

Funkció — Rendszerváltozó értékének lekérdezése.

Használat

```
int r_s_get_value(sernum,outbuf)
    int   sernum;   A rendszerváltozó sorszáma
    char *outbuf;   Az output buffer
```

Leírás — A megadott sorszámú rendszerváltozó értékét betölti a második paraméterként megadott bufferbe. Az output buffer méretének elegendőnek kell lennie az érték tárolásához. A sorszám a következők valamelyike lehet:

```
S_PAGENO (az aktuális lapszám, típusa: T_LONG),
S_DATE   (az aktuális dátum, típusa: T_DATE),
S_LINENO (az aktuális lapon kiírt fizikai sorok
          száma, típusa: T_INT),
S_LINEGL (ez eddig kiírt összes fizikai sorok
          száma, típusa: T_LONG)
(definíciójuk a MIREP.H headerfile-ban).
```

Visszatérés

```
-1  ha a megadott sorszámú rendszerváltozó nem
    létezik,
 0  egyébként.
```

Lásd még — r_s_set_value()

8.6.17. r_s_set_value()

Funkció — Rendszerváltozó értékének beállítása.

Használat

```
int r_s_set_value(sernum,inpbuf)
    int   sernum;    A rendszerváltozó sorszáma
    char *inpbuf;    Az input buffer
```

Leírás

A megadott sorszámú rendszerváltozó értékét betölti a második paraméterként megadott bufferből. A sorszám az előző pontban ismertetettek valamelyike lehet.

Visszatérés

-1 ha a megadott sorszámú rendszerváltozó nem létezik,
0 egyébként.

Lásd még

r_d_get_value()

8.7. Az elemi változók kezelőrutinjai

Összetettebb listaműveletek esetén szükség lehet az elemi változók különböző jellemzőinek dinamikus lekérdezésére. Ezt a célt szolgálják az ebben a pontban ismertetett függvények.

8.7.1. r_d_dec()

Funkció

Az elemi változó decimális számjegyeinek lekérdezése.

Használat

```
int r_d_dec(sernum)
    int   sernum;        Az elemi változó sorszáma
```

Leírás

Visszaadja az elemi változó decimális jegyeinek a számát.

Visszatérés

Az elemi változó decimális jegyeinek a száma.

Lásd még

r_d_length()

8.7.2. r_d_is_total()

Funkció

Visszaadja, hogy az elemi változó összegzett változó-e.

Használat

```
int r_d_is_total(sernum)
    int   sernum;        Az elemi változó sorszáma
```

Leírás Visszaadja, hogy a megadott sorszámú elemi változóra van-e összegzés.

Visszatérés 1 ha az elemi változó összegzett változó,
0 egyébként.

Lásd még r_d_total_onoff()

8.7.3. r_d_length()

Funkció Az elemi változó kiírási szélességének lekérdezése.

Használat
```
int r_d_length()
    int   sernum;        Az elemi változó sorszáma
```

Leírás Visszaadja az elemi változó kiírási szélességét.

Visszatérés Az elemi változó kiírási szélessége.

Lásd még r_d_dec(), r_d_size()

8.7.4. r_d_name()

Funkció Az elemi változó nevének lekérdezése.

Használat
```
char *r_d_name(sernum)
    int   sernum;        Az elemi változó sorszáma
```

Leírás Visszaadja az elemi változó nevét.

Visszatérés Az elemi változó neve.

Lásd még r_d_sernum()

8.7.5. r_d_number()

Funkció Az elemi változók számának lekérdezése.

Használat int r_d_number()

Leírás Visszaadja az elemi változók számát.

Visszatérés Az elemi változók száma.

Lásd még r_d_sernum()

8.7.6. r_d_offset()

Funkció Az elemi változó elemi adatterületen belüli pozíciójának lekérdezése.

Használat
```
int r_d_offset(sernum)
    int   sernum;        Az elemi változó sorszáma
```

Leírás Visszaadja az elemi változónak az elemi adatterületen belüli kezdőpozícióját.

Visszatérés Az elemi változó elemi adatterületen belüli kezdőpozíciója.

Lásd még r_d_buffer(), r_d_size()

8.7.7. r_d_sernum()

Funkció Az elemi változó sorszámának lekérdezése.

Használat
```
int r_d_sernum(name)
    char *name;          Az elemi változó neve
```

Leírás Visszaadja a megadott nevű elemi változó sorszámát.

Visszatérés Az elemi változó sorszáma.

Lásd még r_d_name()

8.7.8. r_d_size()

Funkció Az elemi változó belső ábrázolású mérete.

Használat
```
int r_d_size(sernum)
    int   sernum;        Az elemi változó sorszáma
```

Leírás Visszaadja az elemi változó belső ábrázolású méretét.

Visszatérés Az elemi változó mérete.

Lásd még r_d_type()

8.7.9. r_d_total_onoff()

Funkció — Az elemi változó összegzésének ki-, ill. bekapcsolása.

Használat
```
int r_d_total_onoff(sernum,onoff)
    int   sernum;        Az elemi változó sorszáma
    int   onoff;         Ki- vagy bekapcsolás
```

Leírás — Ki (onoff = 0) vagy bekapcsolja (onoff = 1) a megadott sorszámú elemi változóra az összegzést.

Visszatérés
```
-1  hibás paraméter esetén, vagy ha az elemi vál-
    tozó nem numerikus,
 0  egyébként.
```

Lásd még — r_d_is_total()

8.7.10. r_d_type()

Funkció — Az elemi változó típusának lekérdezése.

Használat
```
int r_d_type(sernum)
    int   sernum;        Az elemi változó sorszáma
```

Leírás — Visszaadja az elemi változó típusát, amely a következők valamelyike lehet:
```
T_BYTE   (1 byte-os előjeles egész),
T_UBYTE  (1 byte-os előjel nélküli egész),
T_INT    (2 byte-os előjeles egész),
T_UINT   (2 byte-os előjel nélküli egész),
T_LONG   (4 byte-os előjeles egész),
T_DATE   (dátum belső formátumú, 4 byte-on),
T_FLOAT  (4 byte-os lebegőpontos érték),
T_DOUBLE (8 byte-os lebegőpontos érték),
T_TEXT   (szöveg)
```
(definíciójuk a MIREP.H headerfile-ban).

Visszatérés — Az elemi változó típusa.

Lásd még — r_d_size()

8.7.11. r_uniq_total()

Funkció — Az elemi adatterületen ismétlődően érkező adat összegzésének korlátozása.

Használat

```
int r_uniq_total(sernum)
    int   sernum;        Az elemi változó sorszáma
```

Leírás

Több adatállományból összeállított listáknál előfordul, hogy egy elemi változó a legmélyebb összegfokozaton csak egy értéke van, de az összegfokozaton belül több rekord is van. Ha e változót összegezzük, akkor ennek értéke annyiszor kummulálódik, ahány rekord tartozik az összegfokozathoz. Ezt elkerülhetjük az "r_uniq_total()" függvény hívásával, amely hatására a megadott sorszámú elemi változóra a legmélyebb összegfokozatban csak az első érték számítódik. A függvény használatára tipikus példa egy numerikus kontrollváltozó összegezése.

Visszatérés

-1 ha a megadott sorszámú elemi változó nem létezik, vagy nem összegzett változó,
0 egyébként.

Lásd még

r_d_sernum()

8.8. Formátumkezelő rutinok

Mint a korábbiakban már leírtuk, az egyes listadefiníciókban több elemi-, fejléc- és összegsor formátum definiálható. Az ezek kezelését (kiválasztását, letiltását stb.) végző rutinhalmazt ismertetjük a következőkben.

8.8.1. r_detail_name()

Funkció

A megadott sorszámú elemisor formátum nevének visszaadása.

Használat

```
char *r_detail_name(sernum)
    int   sernum;     Az elemisor formátum
                      sorszáma
```

Leírás

Visszaadja a megadott elemisor formátum nevét.

Visszatérés

0 ha nincs ilyen sorszámú elemisor formátum, illetve a formátumnak nincs neve,
n a formátum neve egyébként.

Lásd még

r_total_sernum()

8.8.2. r_detail_number()

Funkció — A definiált elemisor formátumok számának visszaadása.

Használat — int r_detail_number()

Leírás — Visszaadja a definiált elemisorok számát.

Visszatérés — A definiált elemisorok száma.

Lásd még — r_detail_sernum(), r_detail_name()

8.8.3. r_detail_onoff()

Funkció — Elemisor kiírásának letiltása.

Használat —
```
int r_detail_onoff(sernum,onoff)
    int  sernum;        Az elemisor sorszáma
    int  onoff;         Ki- vagy bekapcsolás
```

Leírás — A megadott sorszámú elemisor kiírását letiltja (onoff = 0), illetve engedélyezi (onoff = 1).

Visszatérés —
-1 ha a megadott sorszámú elemisor nem létezik,
0 egyébként.

Lásd még — r_detail_sernum(), r_detail_name()

8.8.4. r_detail_select()

Funkció — Az aktuális elemisor beállítása.

Használat —
```
int r_detail_select(sernum)
    int  sernum;        Az elemisor sorszáma
```

Leírás — Átállítja az aktuális elemisort a megadott sorszámúra. A lista inicializálása után automatikusan az 1-es sorszámú elemisor az aktuális.

Visszatérés —
-1 ha a megadott sorszámú elemisor nem létezik,
0 egyébként.

Lásd még — r_detail_print()

8.8.5. r_detail_sernum()

Funkció — A megadott nevű elemisor formátum sorszámának lekérdezése.

Használat
```
int r_detail_sernum(name)
    char *name;    Az elemisor formátum neve
```

Leírás — Visszaadja a megadott nevű elemisor sorszámát.

Visszatérés
-1 ha a megadott nevű elemisor nem létezik,
n az elemisor sorszáma egyébként.

Lásd még — r_detail_name()

8.8.6. r_get_detprint()

Funkció — Az elemisor formátum kiírás letiltásának lekérdezése.

Használat
```
int r_get_detprint()
```

Leírás — Visszaadja, hogy az elemisor formátum kiírása le van-e tiltva.

Visszatérés
0 ha az elemisor formátum kiírása le van tiltva,
1 egyébként.

Lásd még — r_set_detprint()

8.8.7. r_header_name()

Funkció — Fejlécsor formátum nevének lekérdezése.

Használat
```
char *r_header_name(sernum)
    int   sernum;    A fejlécsor formátum
                     sorszáma
```

Leírás — Visszaadja a megadott fejlécsor formátum nevét.

Visszatérés
0 ha nincs ilyen sorszámú fejlécsor formátum, illetve a formátumnak nincs neve,
n a formátum neve egyébként.

Lásd még — r_header_sernum()

8.8.8. r_header_number()

Funkció — A definiált fejlécsor formátumok számának lekérdezése.

Használat — int r_header_number()

Leírás — Visszaadja a definiált fejlécsorok számát.

Visszatérés — A definiált fejlécsorok száma.

Lásd még — r_header_sernum(), r_header_name()

8.8.9. r_header_onoff()

Funkció — Fejlécsor kiírásának letiltása.

Használat —
```
int r_header_onoff(sernum,onoff)
    int  sernum;        A fejlécsor sorszáma
    int  onoff;         Ki- vagy bekapcsolás
```

Leírás — A megadott sorszámú fejlécsor kiírását letiltja (onoff = 0), illetve engedélyezi (onoff = 1).

Visszatérés —
-1 ha a megadott sorszámú fejlécsor nem létezik,
0 egyébként.

Lásd még — r_header_sernum(), r_header_name()

8.8.10. r_header_select()

Funkció — Az aktuális fejlécsor beállítása.

Használat —
```
int r_header_select(sernum)
    int  sernum;        A fejlécsor sorszáma
```

Leírás — Átállítja az aktuális fejlécsort a megadott sorszámúra. A lista inicializálása után automatikusan az 1-es sorszámú fejlécsor az aktuális.

Visszatérés —
-1 ha a megadott sorszámú fejlécsor nem létezik,
0 egyébként.

Lásd még — r_header_print()

8.8.11. r_header_sernum()

Funkció

A megadott nevű fejlécsor formátum sorszámának lekérdezése.

Használat

```
int r_header_sernum(name)
    char *name;    A fejlécsor formátum neve
```

Leírás

Visszaadja a megadott nevű fejlécsor sorszámát.

Visszatérés

-1 ha a megadott nevű fejlécsor nem létezik, a fejlécsor sorszáma egyébként.

Lásd még

r_header_name()

8.8.12. r_set_detprint()

Funkció

Az elemisor formátum kiírásának letiltása vagy engedélyezése.

Használat

```
int r_set_detprint(onoff)
    int   onoff;    Letiltás vagy engedélyezés
```

Leírás

A különböző listázások során gyakran előfordul, hogy az elemisorokra a listában nincs szükségünk, csak a kontrollváltásokra és az ezzel összefüggő összegzésekre, számlálásokra és átlagokra. Ilyenkor az elemisor kiíratását tilthatjuk le vagy engedélyezhetjük e rutin segítségével.

Visszatérés

Nincs.

Lásd még

r_detail_select()

8.8.13. r_searchcontrol()

Funkció

Az összegsor kontrollváltozójának lekérdezése.

Használat

```
int r_searchcontrol(sernum)
    int   sernum;    Az összegző fokozat szintje
```

Leírás

Visszaadja a kért összegsor kontrollváltozójának sorszámát.

Visszatérés -1 ha a kért összegsor nem létezik,
n a kontrollváltozónak kijelölt elemi változó sorszáma egyébként.

8.8.14. r_total_name()

Funkció A megadott sorszámú összegsor formátum nevének lekérdezése.

Használat
```
char *r_total_name(sernum)
    int   sernum;   Az összegsor sorszáma
```

Leírás Visszaadja a megadott sorszámú összegsor formátum nevét.

Visszatérés 0 ha nincs ilyen sorszámú összegsor formátum, illetve a formátumnak nincs neve,
n a formátum neve egyébként.

Lásd még r_total_sernum()

8.8.15. r_total_number()

Funkció A definiált összegsor formátumok számának lekérdezése.

Használat int r_total_number()

Leírás Visszaadja a definiált összegsorok számát.

Visszatérés A definiált összegsorok száma.

Lásd még r_total_sernum(), r_total_name()

8.8.16. r_total_only_more()

Funkció Egy tételes összegző fokozat kiírásának letiltása.

Használat
```
int r_total_only_more(sernum)
    int sernum;          Az összegsor szintje
```

Leírás

A függvény hatására a futtatórendszer a megadott szintű összegsor formátumot csak akkor íratja ki, ha az összegsor csoportjában egynél több rekord volt.

Visszatérés

-1 hibás paraméter esetén,
0 egyébként.

Lásd még

r_total_sernum(), r_total_name()

8.8.17. r_total_onoff()

Funkció

A megadott összegsor ki-, illetve bekapcsolása.

Használat

```
int r_total_onoff(sernum,onoff)
    int   sernum;   A kívánt összegsor szintje
    int   onoff;    Ki- vagy bekapcsolás
```

Leírás

A megadott szintű összegsort kapcsolja ki, illetve be. A sernum paraméter dinamikusan az "r_total_sernum()" függvénnyel kérdezhető le.

Visszatérés

-1 ha az megadott szintű összegsor nem létezik,
0 egyébként.

Lásd még

r_total_sernum()

8.8.18. r_total_sernum()

Funkció

A megadott nevű összegsor formátum szintjének lekérdezése.

Használat

```
int r_total_sernum(tname)
    char *tname;   Az összegsor formátum neve
```

Leírás

Visszaadja a megadott nevű összegsor szintjét. A sorszámra az "r_total_onoff()" függvény számára, mint paraméterre lehet szükség.

Visszatérés

-1 ha a megadott nevű összegsor nem létezik,
n az összegsor szintje egyébként.

Lásd még

r_total_onoff()

8.8.19. r_total_to_detail()

Funkció — Összegsor formátumból elemisor formátum képzése.

Használat — int r_total_to_detail()

Leírás — Ha az elemisorok kiírását az "r_set_detprint()" rutinnal letiltjuk, akkor többnyire a legbelső szintü kontrollváltás összegsora veszi át az elemisor szerepét. Ekkor az összegsorból e rutin segítségével törölhetjük a TOTAL_PAGE és a TOTAL_HEADER hashmark parancsokat.

Visszatérés — Nincs.

Lásd még — r_setdetprint()

8.9. Egyéb rutinok

A következő rutinok a futtatórendszer belső függvényei, használatuk a felhasználó C programjából tilos. A dokumentációban való szerepeltetésük csak a leírás teljességét szolgálja.

8.9.1. r_check()

Funkció — Listadefiníció beolvasásának ellenörzése.

Használat — void r_check()

Leírás — Ellenörzi, hogy van-e beolvasott listadefiníció.

Visszatérés — Nincs.

8.9.2. r_cmd_need_arg()

Funkció — Hashmark parancs kötelező paraméterének lekérdezése.

Használat — int r_cmd_need_arg(hc)
int hc; A hashmark parancs sorszáma

Leírás — Visszaadja, hogy a hashmark parancsnak van-e kötelező paramétere.

Visszatérés
0 ha a hashmark parancsnak nincs kötelező paramétere,
1 egyébként.

8.9.3. r_control_check()

Funkció
Kontrollváltas lekérdezése.

Használat
int r_control_check()

Leírás
A függvény visszaadja, hogy a következő "r_exec()" hívásnál melyik összegző szinten lesz kontrollváltás.

Visszatérés
-1 ha nincs kontrollváltás,
n a kontrollváltás összegző szintje egyébként.

Lásd még
r_exec()

8.9.4. r_create_report()

Funkció
Üres listadefiníció létrehozása.

Használat
```
int r_create(name)
    char *name;          A listadefiníció neve
```

Leírás
A megadott névvel létrehoz egy üres listadefiníciót.

Visszatérés
Nincs.

8.9.5. r_cummulate()

Funkció
A megadott elemi változó értékének hozzáadása a megfelelő összegzőkhöz.

Használat
```
int r_cummulate(sernun)
    int   sernum;        Az elemi változó sorszáma
```

Leírás
A megadott sorszámú elemi változó aktuális értékét hozzáadja a megfelelő összegzőkhöz.

Visszatérés
-1 ha a megadott sorszámú változó nem létezik, illetve nem összegzett változó,

0 ha az elemi változó egyedi összegzett változó, és a legbelső kontrollváltáson belül már összegzésre került,
1 egyébként.

Lásd még r_uniq_total()

8.9.6. r_endspool()

Funkció Hálózati nyomtatás lezárása.

Használat int r_endspool()

Leírás Lezárja a hálózati nyomtatást.

Visszatérés -1 hiba esetén,
0 egyébként.

8.9.7. r_exec_func()

Funkció Felhasználói C függvény név szerinti végrehajtatása.

Használat
```
r_exec_func(name,table,level,isafter)
    char *name;        A C függvény neve
    int  table;        A formátum típusa
    int  level;        A formátum sorszáma
    int  isafter;      Végrehajtás utáni hívás
```

Leírás A rutint a futtatórendszer hívja, programból való hívása tilos.

8.9.8. r_spool()

Funkció Inicializálja a hálózati nyomtatást.

Használat
```
int r_spool(sname,class,copies)
    char *sname;       A server neve
    int  class;        A spool osztály
    int  copies;       A kért példányszám
```

Leírás Inicializálja a hálózati nyomtatást.

Visszatérés -1 hiba esetén,
0 egyébként.

8.9.9. r_write()

Funkció: Listdadefiníció kiírása leíró file-ba.

Használat:
```
int r_write(fname)
    char *fname;          A leíró file neve
```

Leírás: Kiírja az aktuális listadefiníciót a megadott nevü leíró file-ba.

Visszatérés:
-1 hiba esetén,
0 egyébként.

8.10. Csatlakozási pontok rutinjai

A következő rutinok a felhasználói C függvények és a futtató rendszer kapcsolatát teremtik meg.

8.10.1. r_set_funcs()

Funkció: A felhasználói C függvények táblájának átadása a futtatórendszer számára.

Használat:
```
int r_set_funcs(cft)
    struct CFUNC *cft;  A REPEDIT által generált
                        C függvénytábla
```

Leírás: Ennek a függvénynek a segítségével lehet a listadefinícióban megadott felhasználói C függvények tábláját a futtató rendszer számára ismertté tenni. E függvény hívása után a futtatórendszer az egyes formátumokhoz rendelt "Before" és "After" C függvényeket el tudja érni, a hivatkozásokat az "r_exec()" végrehajtása során fel tudja oldani. A függvény paramétere a REPEDIT editor által generált felhasználói C függvénytábla kell.

Visszatérés: A listadefinícióban definiált felhasználói C függvények száma.

Lásd még: r_set_before(), r_set_after()

8.10.2. r_set_before()

Funkció

Sorformátumhoz kiírás előtt végrehajtandó C függvény hozzárendelése.

Használat

```
int r_set_before(table,sernum,fn,fp)
    int   table;        A formátum típusa
    int   sernum;       A táblabeli sorszám
    char *fn;           A felhasználói C függvény
                        neve
    int  (*fp)();       A felhasználói C függvény
```

Leírás

A rutin segítségével a sorformátum kiírása előtt végrehajtandó felhasználói C függvényt rendelhetünk a sorformátumhoz.
A table paraméter azt adja meg, hogy melyik típusú sorformátumhoz akarunk C függvényt rendelni. Lehetséges értékei:

F_HEADER (fejlécsor),
F_DETAIL (elemisor) vagy
F_TOTAL (összegsor)

(definíciójuk a MIREP.H headerfile-ban).

A sernum paraméter a sorformátum sorszáma, illetve összegsornál a szintje,
az fn paraméter a C függvény neve,
az fp paraméter pedig a C függvény.

Visszatérés

A definiált felhasználói C függvények száma.

Lásd még

r_set_after(), r_set_funcs()

8.10.3. r_set_after()

Funkció

Sorformátumhoz kiírás után végrehajtandó C függvény hozzárendelése.

Használat

```
int r_set_after(table,sernum,fn,fp)
    int   table;        A formátum típusa
    int   sernum;       A táblabeli sorszám
    char *fn;           A felhasználói C függvény
                        neve
    int  (*fp)();       A felhasználói C függvény
```

Leírás

A rutin segítségével a sorformátum kiírása után végrehajtandó felhasználói C függvényt rendelhetünk a sorformátumhoz.
A table paraméter azt adja meg, hogy melyik

típusú sorformátumhoz akarunk C függvényt rendelni. Lehetséges értékei:
F_HEADER (fejlécsor),
F_DETAIL (elemisor) vagy
F_TOTAL (összegsor)
(definíciójuk a MIREP.H headerfile-ban).

A sernum paraméter a sorformátum sorszáma, illetve összegsornál a szintje,
az fn paraméter a C függvény neve,
az fp paraméter pedig a C függvény.

Visszatérés — A definiált felhasználói C függvények száma.

Lásd még — r_set_before(), r_set_funcs()

8.11. A felhasználói C függvények

A helhasználói C függvények a jól körülhatárolt illeszkedési pontokon kapják meg a vezérlést a MIREP futtató rendszerétől.

Funkció — Sorformátumhoz kiírása előtt vagy után különleges funkciók ellátása.

Használat

```
int USERFUNCTION(table,level,isafter)
    int   table;        A formátum típusa
    int   level;        Formátum sorszáma
    int   isafter;      Végrehajtás utáni hívás?
```

A függvény neve természetesen a listadefinícióban megadott kell, hogy legyen. A paraméterek csak abban az esetben fontosak, ha ugyanazt a felhasználói C függvényt több formátumhoz, ill. végrehajtás előtt hívandó és után hívandó függvényként is a listadefinícióba illesztettük. Ekkor ugyanis a paraméterek alapján lehet eldönteni, hogy a függvény milyen végrehajtási fázisban kapja meg a vezérlést.

A függvény tetszőleges utasításokat tartalmazhat, beleértve az egyes futtatórendszerhez tartozó rutinok hívását is. Ügyelni kell azonban arra, hogy ne végezzünk el olyan műveletet, amely a listagenerálás logikájával ellentétes.

A függvény visszatérési értéke nagyon fontos:

0 esetén a listagenerálás folytatódik,
n (egyébként) az "r_exec()" függvény a hatáskörébe tartozó feladatokat nem végzi el, és a megadott visszatérési értékkel visszatér a felhasználói programba.

9. A listázás környezete

A hálózati listázásról már az eddigiekben is gyakran esett szó. Most összefoglaljuk a tudnivalókat.

A hálózati listázás sajátosságai:

- Nem lehet előre tudni, milyen típusú nyomtatón fog kiíródni az állomány.
- Célszerű megengedni az átirányításokat, vagyis a listát lehessen kérni lokális nyomtatóra, globális nyomtatóra, konzolra, file-ba.
- A globális nyomtató akármelyik server-en lehet.
- A file-ba kért listáknak felhasználónként el kell különülniük.

A MIxxx rendszerekben és könyvtárakban a fenti problémákra létezik egy javasolt megoldás.

9.1. Nyomtatók illesztése

A hálózathoz tudnunk kell tetszés szerinti nyomtatót illeszteni. Nem érdemes megpróbálni lefedni az összes elérhető nyomtatót; egészen biztosan a készenlét napján jelentkezik valaki egy újjal.

Helyette eszközt kell adni a felhasználónak a nyomtató illesztésére. Az eszközzel egy szűrőfile-t lehet létrehozni, amelyet a lista végrehajtása közben a kiíró programrész automatikusan felhasznál.

9.1.1. Szűrőfile-ok

Az egyes szűrőfile-ok azt határozzák meg, hogy a MIREP által generált belső formában létrehozott lista milyen módon kerüljön ki az aktuális nyomtatóra. A szűrőfile-ok nevének kiterjesztése kötelezően .PTC (Pseudo Text Conversion).

A szűrőfile-ok alapvetően a következőket tartamazzák:

- Standard nyomtatóvezérlő szekvenciák (pl. dőlt betű, karakterméret stb.).
- Ékezetes karaktereknek megfelelő szekvenciák.
- Kvázigrafikus karakterek szekvenciái.

- Speciális, ún. USER szekvenciák (pl. széles lapos betük).
- Karakter elrendelések.

A MIREP disztributív anyagában lévö szüröfile-ok:

BROTHER	.PTC	Brother M-1709,
CONSOL	.PTC	CWI szabvány karakterkészlet vezérlés nélkül,
EPSON	.PTC	EPSON, tetszöleges típus,
IBM	.PTC	IBM nyomtató,
IBM-EPS	.PTC	IBM karakterkészlet EPSON vezérléssel,
NONE	.PTC	ékezet és vezérlés nélkülire konvertálás.

A szüröfile-okat a PTCED editor programmal (Pseudo Text Conversion EDitor). lehet létrehozni és módosítani.

9.1.2. PTCED szüröfile editor

A PTCED editor lehetöséget ad arra, hogy egy nyomtatóhoz szürö-file formájában definiáljuk az ékezetes betük, nyomtatóvezérlö parancsok stb. konkrét nyomtatási byte-szekvenciáit.

Az editorral létrehozhatunk új szüröfile-t, vagy módosíthatunk egy már meglévöt.

A következö ábrákon az EPSON.PTC szüröfile tartalma látható.

9.1.2.1. A PTCED indítása

A program hívása:

>PTCED [-b] ptcfile

ahol ptcfile a szüröfile neve
-b a fekete-fehér üzemmódot jelöli.

A program használata során a kiemelt betü, vagy a kurzorbillentyükkel való ráállás után az ENTER billentyü leütésével hajthatjuk végre a kívánt funkciót.

Az ESC billentyü az aktuális funkció vagy menü elhagyását jelenti.

A program indítása után megjelenő fő menü:

EPSON.PTC
Standard control bytes User control bytes Accentuated letters Graphic characters Other input-output strings Character table Description of PTC file Read in DOG's CNV file Test actual setting Write tables to file

A menü egyes tételei rendre a következők:

- Standard nyomtatóvezérlések táblázata.
- Felhasználó (USER) által definiált speciális vezérlések táblázata.
- Ékezetes betűk táblázata.
- Kvázigrafikus karakterek táblázata.
- Input-output string-táblázat.
- Karakterkonverziós tábla.
- Szűrőfile leírás megadása.
- DOG konvertáló táblázat beolvasása.
- Szűrőfile tesztelése.
- Kimentés file-ba.

Az Input-output stringek táblázatában tetszőleges karaktersorozatra határozhatjuk meg, hogy milyen karaktersorozat kerüljön ki helyette.

A szűrőfile leírás megadásával tetszőleges magyarázó szöveget rendelhetünk a szűrőfile-hoz.

A DOG konvertáló táblázat beolvasása arra ad lehetőséget, hogy azok a felhasználók, akik rendelkeznek a DOG szövegformázó környezettel (Magister Software, (c) 1987) és az ott definiált konverziós file-okat fel akarják használni, e funkcióval beolvashatják azokat.

A kimentés file-ba funkciót csak akkor kell használni, ha a PTCED indításakor megadott szűrőfile helyett egy másik file-ba akarjuk az aktuális állapotot kimenteni.

A többi funkciót a következő fejezetekben ismertetjük.

Minden egyes inputra megadható üres output, ami azt jelenti, hogy

az adott szűrőfile használatakor ennek az inputnak a hatására nem íródik ki semmi.

A következő funkciók input-output táblázat formájában kezelhetők.

Az input értékek nem módosíthatók.

Az output byte-szekvencia megadásának lehetséges módjai:

- Az egyes byte-ok numerikus értékenek megadása, egymástól vesszővel elválasztva,

- A látható karakterek karakteres formában való megadása idézőjel közé zárva (egy idézőjelpár között több ASCII karakter is szerepelhet).

A numerikus értékek megadási módját az F3 billentyű leütése után választhatjuk ki:

Decimal: decimális kód,
Octal : oktális kód,
Hexa : hexadecimális kód.

Az F4 billentyűvel a 127-es karakternél magasabb értékű karakterek kiírási módját változtathatjuk:

Numeric : az F3-mal beállított numerikus forma,
Character: karakteres forma.

Az F2 billentyű az aktuális táblázat és a karakterkonverziós táblázat közötti váltást teszi lehetővé.

9.1.2.2. A standard nyomtatóvezérlések

A funkció kérésekor a 10. ábrán látható táblázat jelenik meg:

Standard control bytes	
Description of Input	Output
Init printer	.....No output.....
Reset printer	.....No output.....
Help page	.....No output.....
Underline on	ESC,'-',1
Underline off	ESC,'-',0
Double strike on	ESC,'G'
Double strike off	ESC,'H'
Italic/alternate on	ESC,'4'
Italic/alternate off	ESC,'5'
Normal charcter size	18,ESC,'P'
Elite charcter size	18,ESC,'M'
Condensed charcter size	ESC,'P',15
Normal line spacing (1/6 inch)	ESC,'2'
Half line spacing (1/8 inch)	ESC,'0'
Without line spacing (7/72 inch)	ESC,'1'

10. ábra

Az egyes nyomtatóvezérlések magyarázata:

- Init printer
  Nyomtató inicializálás, csak akkor szükséges, ha valamilyen, az egész listára vonatkozó nyomtatóvezérlést akarunk beállítani. Pl. bizonyos nyomtatók megengedik a felhasználó által definiált karakterek használatát, és így pl. a hosszú nagy 'ü' szépen definiálható (az "Ékezetes betűk" táblázatban az itt definiált saját karakterre kell hivatkozni).

- Reset printer
  Nyomtató visszaállítás, csak az előző funkcióban beállított vezérlések megszüntetésére szolgál (pl. a felhasználó által definiált karakterértelmezés megszüntetése).

- Help page
  Segítő információban új lap kezdete, a MIREP környezetben nincs értelmezve.

- Underline on
  Aláhúzás bekapcsolása.

- Underline off
  Aláhúzás kikapcsolása.

- Double strike on
  Kétszeres, dupla írás bekapcsolása.

- Double strike off
  Kétszeres, dupla írás kikapcsolása.

- Italic/alternate on
  Dölt betüs írás vagy alternatív karakterkészlet bekapcsolása.

- Italic/alternate off
  Dölt betüs írás vagy alternatív karakterkészlet kikapcsolása.

- Normal charcter size
  Normális karakterméret (lásd még a CHARSIZE= opciót).

- Elite charcter size
  Félkeskeny karakterméret (lásd még a CHARSIZE= opciót).

- Condensed charcter size
  Keskeny karakterméret (lásd még a CHARSIZE= opciót).

- Normal line spacing (1/6 inch)
  Normális sortáv (lásd még LINESPACING= opciót).

- Half line spacing (1/8 inch)
  Fél sortáv (lásd még LINESPACING= opciót).

- Without line spacing (7/72 inch)
  Sortáv nélküli írás (lásd még LINESPACING= opciót).

A táblázat bal oldala nem módosítható. A táblázat jobb oldalát az ENTER billentyü leütése után változtathatjuk, az elözö pontban leírt módon.

9.1.2.3. USER vezérlő szekvenciák

A funkció kérésekor a 11. ábrán látható táblázat jelenik meg:

User control bytes		
Description of	Input	Output
	User control 0	.....No output.....
	User control 1	.....No output.....
	User control 2	.....No output.....
	User control 3	.....No output.....
	User control 4	.....No output.....
	User control 5	.....No output.....
	User control 6	.....No output.....
	User control 7	.....No output.....
Enlarged+Cond ON	User control 8	14,15
Enlarged+Cond OF»	User control 9	18,20

11. ábra

A felhasználó által szabadon meghatározható szekvenciák különleges funkcióra használatosak, pl. bizonyos nyomtatóknál a betűtípus átváltása.

A MIREP rendszer Printer_Control hashmark parancsa lehetőséget ad ezek használatára, de figyelembe kell venni, hogy a USER szekvenciák csak némely nyomtatóhoz definiálhatók azonos funkcióval, más nyomtatók esetén ezeknek nem lesz output karakter, ill. vezérlő sorozata.

A példánkban használt EPSON szűrőfile-ban a 8-as és 9-es USER vezérlőket definiáltuk az Enlarged+Condensed nyomtatási mód be- és kikapcsolására.

9.1.2.4. Ékezetes betűk

Az ékezetes betűk kezelése a CWI szabvány szerinti karakterkészlettel működő nyomtatóknál a legegyszerűbb, hiszen a MIREP alapvetően ezt a karakterkészletet használja. Ebben az esetben az ékezetes betű output megfelelője önmaga. A nyomtatók többsége viszont a magyar karakterekre nincs fekészítve, így az ezeknek megfelelő outputot meg kell határozni.

Ilyenkor (pl. az EPSON nyomtatónál) az ékezetes betűk két módon adhatók meg:

1. meglévő karakter kinyomtatása 3 lépésben:
   a. annak a nemzeti karakterkészletnek a kiválasztása, amelyben szerepel a kívánt ékezetes karakter,
   b. a megfelelő karakter nyomtatása,
   c. az alap nemzeti karakterkészlet visszaállítása.

2. a megfelelő ékezet ráütése az ékezet nélküli megfelelő betűre:
   a. az ékezet nélküli karakter nyomtatása,
   b. visszalépés egy mezővel a BackSpace karakter segítségével,
   c. az ékezetnek megfelelő karakter nyomtatása.

A fenti két megoldást jól szemlélteti a funkció kérésekor megjelenő táblázat (12. ábra):

Accentuated letters					
Input					Output
	160	a0H	\240	'á'	ESC,'R',6,'{',ESC,'R',0
	132	84H	\204	'ä'	ESC,'R',2,'{',ESC,'R',0
	130	82H	\202	'é'	ESC,'R',10,'`',ESC,'R',0
	161	a1H	\241	'í'	'i',8,''''
	162	a2H	\242	'ó'	ESC,'R',6,'¦',ESC,'R',0
	148	94H	\224	'ö'	ESC,'R',5,'¦',ESC,'R',0
	147	93H	\223	'ő'	'o',8,'"'
	163	a3H	\243	'ú'	'u',8,''''
	129	81H	\201	'ü'	ESC,'R',2,'}',ESC,'R',0
	150	96H	\226	'ű'	'u',8,'"'
	143	8fH	\217	'Á'	'A',8,''''
	142	8eH	\216	'Ä'	ESC,'R',2,'[',ESC,'R',0
	144	90H	\220	'É'	ESC,'R',5,'@',ESC,'R',0
	73	49H	\111	'I'	'I' See the character table
	149	95H	\225	'Ó'	'O',8,''''
	153	99H	\231	'Ö'	ESC,'R',5,'\',ESC,'R',0
	167	a7H	\247	'Ő'	'O',8,'"'
	151	97H	\227	'Ú'	'U',8,''''
V	154	9aH	\232	'Ü'	ESC,'R',2,']',ESC,'R',0

12. ábra

9.1.2.5. Kvázigrafikus karakterek

A MIREP a szövegkonstansok definiálásakor támogatja a kvázigrafikus karakterek megadását. Ezek természetesen szintén nyomtatófüggőek, az IBM karakterkészlettől eltérő nyomatatóknál gondoskodni kell ezek kiírási módjáról. A kvázigrafikus karak-

terek kezelését a 13. ábrán látható táblázatban (a korábbiakban már ismertetett módon) valósíthatjuk meg.

Graphic characters						
	Input					Output
	179	b3H	\263	'│'		ESC,'m',4,134,ESC,'m',0
	180	b4H	\264	'┤'		ESC,'m',4,131,ESC,'m',0
	181	b5H	\265	'╡'		ESC,'m',4,131,ESC,'m',0
	182	b6H	\266	'╢'		ESC,'m',4,131,ESC,'m',0
	183	b7H	\267	'╖'		ESC,'m',4,136,ESC,'m',0
	184	b8H	\270	'╕'		ESC,'m',4,136,ESC,'m',0
	185	b9H	\271	'╣'		ESC,'m',4,131,ESC,'m',0
	186	baH	\272	'║'		ESC,'m',4,134,ESC,'m',0
	187	bbH	\273	'╗'		ESC,'m',4,136,ESC,'m',0
	188	bcH	\274	'╝'		ESC,'m',4,138,ESC,'m',0
	189	bdH	\275	'╜'		ESC,'m',4,138,ESC,'m',0
	190	beH	\276	'╛'		ESC,'m',4,138,ESC,'m',0
	191	bfH	\277	'┐'		ESC,'m',4,136,ESC,'m',0
	192	c0H	\300	'└'		ESC,'m',4,137,ESC,'m',0
	193	c1H	\301	'┴'		ESC,'m',4,129,ESC,'m',0
	194	c2H	\302	'┬'		ESC,'m',4,130,ESC,'m',0
	195	c3H	\303	'├'		ESC,'m',4,132,ESC,'m',0
	196	c4H	\304	'─'		ESC,'m',4,133,ESC,'m',0
V	197	c5H	\305	'┼'		ESC,'m',4,128,ESC,'m',0

13. ábra

9.1.2.6. Karakterkonverzió

A karakterkonverziós tábla kettős funkciójú:

- egyrészt összefoglaló táblázatként áttekinthető az összes 8 bites ASCII karakter kiírási megfelelője,

- másrészt az egyszerű input karakter - output karakter megfeleltetéseket itt adhatjuk meg azokra a karakterekre, amelyek nem szerepelnek az előbbi táblázatokban.

A táblázat "comment" oszlopa egy csillaggal kezdődik, ha a karakter helyett más kerül ki a nyomtatóra, és külön szöveg jelzi, ha a karakter egy másik táblázatban szerepel (ekkor az F2 billentyűvel ráugorhatunk a megfelelő táblázat megfelelő elemére).

A karakterkonverziós tábla képe a 14. ábrán látható.

Character table									
	Input				Output				
	Dec.	Hex.	Oct.	Ch.	Dec.	Hex.	Oct.	Ch.	Comment
^	122	7aH	172	z	122	7aH	172	z	
	123	7bH	173	{	123	7bH	173	{	
	124	7cH	174	¦	124	7cH	174	¦	
	125	7dH	175	}	125	7dH	175	}	
	126	7eH	176	~	126	7eH	176	~	
	127	7fH	177		127	7fH	177		
	128	80H	200	Ç	128	80H	200	Ç	
	129	81H	201	ü					* Accentuated letters
	130	82H	202	é					* Accentuated letters
	131	83H	203	â	131	83H	203	â	
	132	84H	204	ä					* Accentuated letters
V	133	85H	205	à	133	85H	205	à	

14. ábra

9.1.2.7. A szűrőfile tesztelése

A szűrőfile létrehozásakor lehetőségünk van on-line tesztelésre. A 15. ábrán látható menü segítségével a szűrőfile egyes elemeinek nyomtatási képét ellenőrizhetjük a lokális nyomtatón (természetesen a lokális nyomtatónak fogadóképes állapotban kell lennie).

Select a function
A. Print accentuated letters
B. Print graphics
C. Print common ASCII characters
D. Print interactively
E. Init printer
F. Reset printer
G. Underline on
H. Underline off
I. Double strike on
J. Double strike off
K. Italic/alternate on
L. Italic/alternate off
M. Normal charcter size
N. Elite charcter size
O. Condensed charcter size
P. Normal line spacing
Q. Half line spacing
R. Without line spacing
S. New line
T. Form feed

15. ábra

Az egyes funkciók jelentése értelemszerü: a szürőfile korábban ismertetett elemei közül a megfelelő kinyomtatása.

9.2. Lokális nyomtató megadása

A lokális nyomtatók definícióját egy munkaállomást leíró file-ban célszerü tárolni. Kezeléséhez három funkció szükséges:

- Munkaállomás számának lekérdezése.
- Lokális nyomtató típusának a lekérdezése képernyőről és beírása a munkaállomás definíciós file-ba.
- A típus visszaolvasása a definíciós file-ból.

A MICS alkalmazói keretrendszerben ezek a funkciók adottak, hívható C modulok, illetve menü funkció formájában.

9.3. Hálózati nyomtató megadása

Az alkalmazás rendszerparaméterei közé fel kell venni egy GLOBALPRINTER= opciót. Az opcióban meg lehet adni a nyomtató nevét, a server nevét és a kívánt példányszámot.

A hálózati nyomtató adatainak kezeléséhez a következő funkciók szükségesek:

- Editor az alkalmazás rendszerparamétereinek a felvitelére és módosítására.

- C rutin a GLOBALPRINTER= opció beolvasására.

A MICS keretrendszerben ezek a funkciók adottak.

9.4. Lemezre küldött listák

Az átirányításokat felső szintű rutinokkal kezeljük (init_lst vagy init_naplo), mert egy univerzális szituációról van szó. A lemezre küldött listák kiíratására standard segédprogramokat érdemes használni. Ennek egyetlen problémája az írásmódot megadó vezérlő karakterek elvesztése. Ezt két megoldással tudjuk pótolni:

- A segédprogramok a listadefinícióhoz is hozzáférhetnek, s onnan kiveszik a beállított írásmódot.

- Egy felhasználó által beállítható opció alapján választjuk ki az írásmódot!

A MICS rendszer a második megoldást használja.

Egyik megoldás sem engedi meg a listán belüli írásmód váltását!

9.5. Képernyőre küldött listák

A képernyőre küldött listák programozása néhány különleges megoldást igényel:

- A képernyőről le kell törölni a státuszjelző kiírást,
- Nem szabad engedni "elfutni" a listát.

9.6. Standard listakezelési funkciók

A felhasználó a MIREP rendszerrel fejlesztett listákkal komoly lehetőségekhez jut. Saját maga tudja változtatni listáinak formáját, sőt adattartalmát is.

Célszerű ezért minden alkalmazásba beépíteni egy erre a célra szolgáló funkcióhalmazt. A MICS a következő megoldást adja:

Listák kezelése
Kész lista átnézése kész lista Törlése kész lista kiíratása Lokális nyomtatóra kész lista kiíratása Hálózati nyomtatóra
listadefiníció Változtatása listadefiníció Másolása listadefiníció Elhagyása

10. A disztributív anyag

A MIREP disztributív anyaga egy felhasználói kézikönyvből és három hajlékony lemezből áll.

A MIREP környezet installálása merev lemezre a következő paranccsal lehetséges:

```
>A:MINSTALL [-d célkönyvtár]
```

ahol a "célkönyvtár" a merev lemez azonosítóját és a kívánt könyvtár nevét jelenti (pl. D:\REPORT)

Az MINSTALL program paraméter nélkül indítva az installálást a C:\MIREP könyvtárba végzi. Az installálás után (paraméter nélküli indítást feltételezve) a C:\MIREP könyvtár tartalma következő:

```
FTXTORTX.EXE: MIFORM kérdőív leírásból listaleíró file
              generálása,
PTCED   .EXE: szűrőfile editor,
QTXTORTX.EXE: listaleíró file generlálása MISAM vagy MIQUEL
              leírófile alapján,
REPDOC  .EXE: listaleírás dokumentálása,
REPEDIT .EXE: a report editor,
REPTEST .EXE: listakép készítése generált adatokkal,
```

C:\MIREP\PTC alkönyvtár tartalma (szűrőfile-ok):

```
BROTHER.PTC: Brother, M-1709,
CONSOL .PTC: CWI szabvány karakterkészlet vezérlés nélkül,
EPSON  .PTC: EPSON, tetszőleges típus,
IBM    .PTC: IBM nyomtató,
IBM-EPS.PTC: IBM karakterkészlet EPSON vezérléssel,
NONE   .PTC: ékezet és vezérlés nélkülire konvertálás.
```

A C:\MIREP\MSLIB alkönyvtár tartalma:

```
MIREP.H   : MIREP headerfile,
LMIREP.LIB: Microsoft 5.x futtatórendszer (Large memóriamodell),
SMIREP.LIB: Microsoft 5.x futtatórendszer (Small memóriamodell).
```

A C:\MIREP\TCLIB alkönyvtár tartalma:

```
MIREP.H   : MIREP headerfile,
MIREPL.LIB: TurboC 2.x futtatórendszer (Large memóriamodell),
MIREPS.LIB: TurboC 2.x futtatórendszer (Small memóriamodell).
```

A C:\MIREP\SAMPLE alkönyvtár tartalma:

```
LIST     .C  : standard listázóprogram forrása,
TERM_ANY.C   : példaprogram,
TERM_QRY.C   : MIQUEL lekérdezés alapú példaprogram,
ANYAG    .DAT: példa anyag törzsfile (ASCII formátum),
TERM_ANY.DAT: példa termékösszetétel file (ASCII formátum),
TERMÉK   .DAT: példa termék file (ASCII formátum),
ANYAG    .DBF: példa anyag törzsfile (MISAM formátum),
TERM_ANY.DBF: példa termékösszetétel file (MISAM formátum),
TERMÉK   .DBF: példa termék file (MISAM formátum),
ANYAG    .FDF: példa anyag törzsfile MISAM leíró,
TERM_ANY.FDF: példa termékösszetétel file MISAM leíró,
TERMÉK   .FDF: példa termék file MISAM leíró,
ANYAG    .NDX: példa anyag törzsfile MISAM indexállomány,
TERM_ANY.NDX: példa termékösszetétel file MISAM indexállomány,
TERMÉK   .NDX: példa termék file MISAM indexállomány,
TERM_ANY.QTX: termék lekérdezés MIQUEL leíró file,
TERM_ANY.QH  : MIQUEL QRYEDIT által generált C headerfile,
TERM_ANY.RH  : REPEDIT által generált C headerfile,
TERM_ANY.RTX: példa termékösszetétel listaleírása,
TESTTECH.RTX: technikai lehetőségeket bemutató lista-
              leírás.
```

F. Függelékek

A dokumentációban használt példa egy nagyon egyszerü, sematikus készletnyilvántartási alrendszer elemeit szemlélteti egyszerü tesztadatokkal.

A példa adatbázisa:
ANYAG: az anyagok törzsfile-ja,
TERMEK: a termékek file-ja,
TERM_ANY: a termékek anyagösszetételének file-ja.

A függelékek e példához tartozó file-ok, lekérdezések, listák, a MIREP edititor által generált C modulok és a példaprogramok forrásainak listáit tartalmazzák, illetve a szerepel még a MIREP technikai lehetöségeit bemutató TESTTECH listadefiníció dokumentációja és e listadefiníció alapján készített teszt lista.

Az A. függelék az adatbázisnak a MISAM file-kezelö rendszerrel való realizálását dokumentálja.

A B. függelék az adatbázis három listadefiníciójának és a technikai teszt listadefiníciójának a REPDOC programmal generált dokumentációit tartalmazza.

A C. függelék az adatbázis lekérdezésének a MIQUEL lekérdezési rendszerrel való megvalósítását mutatja be.

A D. függelék a TERM_ANY listaleírófile-t tartalmazza.

Az E. függelékben a TERM_ANY listadefiníció alapján a REPEDIT editorral generált C forrásmodult látható.

Az F. függelék a TESTTECH listadefiníció alapján a REPTEST programmal készült listaképet szemlélteti.

A G. függelékben találhatjuk a dokumentációban hivatkozott C programok forrásait.

F.A. File-leírások

1.: ANYAG
2.: TERMEK
3.: TERM_ANY

F.A.1. ANYAG

Adatállomány: ANYAG
Magyarázat : Az anyagok törzsfile-ja
Teljes név : ANYAG.DBF
Indexek száma : 1
Rekordmezők száma: 3
A rekord mérete : 43 byte

1. index: ANYAG
Magyarázat : Anyagkód szerinti kulcs
Teljes név : ANYAG.NDX
Kulcsdarabok száma: 1
Index mérete : 5 byte
A kulcsdarabok leírása:

Kezdet	Méret	Típus	Mező
0	5	Text	anyagkod

1. mező: anyagkod
Magyarázat: Az anyag kódja
Típus : Text
Méret : 5 byte
Ábrázolás : 5 karakter

2. mező: anyagnev
Magyarázat: Az anyag megnevezése
Típus : Text
Méret : 30 byte
Ábrázolás : 30 karakter

3. mező: egysegar
Magyarázat: Az anyag egységára
Típus : Double
Méret : 8 byte
Ábrázolás : 9 karakter (tizedes jegyek: 2)

F.A.2. TERMEK

Adatállomány: TERMEK
Magyarázat : A termékek törzsfile-ja
Teljes név : TERMEK.DBF
Indexek száma : 1
Rekordmezők száma: 3
A rekord mérete : 58 byte

1. index: TERMEK
Magyarázat : A termékkód szerinti kulcs
Teljes név : TERMEK.NDX
Kulcsdarabok száma: 1
Index mérete : 10 byte
A kulcsdarabok leírása:

Kezdet	Méret	Típus	Mező
0	10	Text	tkod

1. mező: tkod
Magyarázat: A termék kódja
Típus : Text
Méret : 10 byte
Ábrázolás : 10 karakter

2. mező: tnev
Magyarázat: A termék neve
Típus : Text
Méret : 40 byte
Ábrázolás : 40 karakter

3. mező: tmenny
Magyarázat: A termék raktári mennyisége
Típus : Double
Méret : 8 byte
Ábrázolás : 14 karakter (tizedes jegyek: 2)

F.A.3. TERM_ANY

Adatállomány: TERM_ANY
Magyarázat : Az egyes termékekhez tartozó anyagok
Teljes név : TERM_ANY.DBF
Indexek száma : 1
Rekordmezők száma: 3
A rekord mérete : 23 byte

1. index: TERM_ANY
   Magyarázat : A termékkód szerinti kulcs
   Teljes név : TERM_ANY.NDX
   Kulcsdarabok száma: 2
   Index mérete : 15 byte
   A kulcsdarabok leírása:

Kezdet	Méret	Típus	Mező
0	10	Text	tkod
10	5	Text	anyagkod

1. mező: tkod
   Magyarázat: A termék kódja
   Típus : Text
   Méret : 10 byte
   Ábrázolás : 10 karakter

2. mező: anyagkod
   Magyarázat: Az anyag kódja
   Típus : Text
   Méret : 5 byte
   Ábrázolás : 5 karakter

3. mező: mennyiseg
   Magyarázat: A termékhez szükséges mennyiség
   Típus : Double
   Méret : 8 byte
   Ábrázolás : 9 karakter (tizedes jegyek: 2)

F.B. Listaformátumok

1.: ANYAG
2.: TERMEK
3.: TERM_ANY
4.: TESTTECH

F.B.1. ANYAG

F.B.1.1. Általános jellemzők

Az anyagok listája
Rendezettség leírása : Az anyag kódjára
Kimeneti eszköz : Localprinter
Kimeneti file könyvtára :
Kimeneti file-név : ANYAG.LST
Server név (hálózatnál) :
Lista (Queue) szám -"- : 0
Másolatok száma -"- : 1

Nyomtató típusa (szürőfile): EPSON
Karakterméret : Normal
Sorköz : Normal
Sorok száma laponként : 65
Dátumformátum : Hungarian
Kimenet megnyitási módja : Write
Elemisor nyomtatása kell? : Yes
Lapemelés kell? : Yes
Első lap előtt kell emelés?: No
Utolsó lapemelés kell? : Yes

Elemi adatok száma : 3
Fejlécsor formátumok száma : 2
Elemisor formátumok száma : 2
Összegsor formátumok száma : 1

F.B.1.2. Fejléc, elemisor és összegsor formátumok

Fejléc formátum 1.: ANYAG
Anyaglista fejléce

Anyagok listája	SSSSSSSS	SSSSSS.lap
Kód Név		Egységár

Elemisor formátum 1.: ANYAG
Az anyagok

___	_____	_____
___	_____	_____
___	_____	_____

F.B.2. TERMEK

F.B.2.1. Általános jellemzők

A termékek listája
Rendezettség leírása : A termék kódjára
Kimeneti eszköz : Localprinter
Kimeneti file könyvtára :
Kimeneti file-név : TERMEK.LST
Server név (hálózatnál) :
Lista (Queue) szám -"- : 0
Másolatok száma -"- : 1

Nyomtató típusa (szürőfile): EPSON
Karakterméret : Normal
Sorköz : Normal
Sorok száma laponként : 65
Dátumformátum : Hungarian
Kimenet megnyitási módja : Write
Elemisor nyomtatása kell? : Yes
Lapemelés kell? : Yes
Első lap előtt kell emelés?: No
Utolsó lapemelés kell? : Yes

Elemi adatok száma : 3
Fejlécsor formátumok száma : 2
Elemisor formátumok száma : 2

Összegsor formátumok száma : 1

F.B.2.2. Fejléc, elemisor és összegsor formátumok

Fejléc formátum 1.: TERMEK
Termék törzslista fejléce

Termékek listája	SSSSSSSS	SSSSSS.lap
Kód Név		Mennyiség

Elemisor formátum 1.: TERMEK
A termékek

_____	_____	_____
_____	_____	_____
_____	_____	_____

F.B.3. TERM_ANY

F.B.3.1. Általános jellemzők

Termék kimutatás
Rendezettség leírása : A termék kódjára
Kimeneti eszköz : Diskfile
Kimeneti file könyvtára : \WORK
Kimeneti file-név : WORK.LST
Server név (hálózatnál) : FS1
Lista (Queue) szám -"- : 2
Másolatok száma -"- : 1

Nyomtató típusa (szürőfile): EPSON
Karakterméret : Elite
Sorköz : Normal
Sorok száma laponként : 65
Dátumformátum : Hungarian
Kimenet megnyitási módja : Write
Elemisor nyomtatása kell? : Yes
Lapemelés kell? : Yes
Első lap előtt kell emelés?: No
Utolsó lapemelés kell? : Yes

Elemi adatok száma : 9
Fejlécsor formátumok száma : 3
Elemisor formátumok száma : 2
Összegsor formátumok száma : 2

F.B.3.2. Fejléc, elemisor és összegsor formátumok

Fejléc formátum 1.: TERMEK
Termék-anyag lista fejléce

Termék összesítő lista	SSSSSSSS	SSSSSS.lap
#		

A '#' parancsok a megjelenés sorrendjében:
Detail_Header(2)

Fejléc formátum 2.: THEADER
Egy termék fejléce

Termék: _____ , Kód: _____

Anyag kód	Anyagnév	Szükséges menny.(db)	Egys.ár Ft	Egys.term. értéke

Elemisor formátum 1.: ANYAGOK
A termékhez tartozó egy anyag adatai

___	_____	_____	_____	_____
___	_____	_____	_____	_____
___	_____	_____	_____	_____

Összegsor formátum 1.: TERMOSSZ
Termékenkénti összegző
Kontrollváltás változója : tkod

```
A(z) CCCCC anyagból álló termék egy egységének anyagértéke:     TTTTTTTTT
Raktárban található mennyiség: OOOOOOOOOOOOOO db.  Értéke: TTTTTTTTTTTTT  #
```

A '#' parancsok a megjelenés sorrendjében:
Total_Header(2)

Végösszegző (Grand Total) formátum 0.:
A raktárban lévő termékek összértéke

```
A raktárban található termékek összértéke: TTTTTTTTTTTTT Ft
```

F.B.4. TESTTECH

F.B.4.1. Általános jellemzők

Technikai teszt listadefiníció
Rendezettség leírása : kod1-re és azon belül a kod2-re
Kimeneti eszköz : Diskfile
Kimeneti file könyvtára :
Kimeneti file-név : TESTTECH.TST
Server név (hálózatnál) :
Lista (Queue) szám -"- : 0
Másolatok száma -"- : 1

Nyomtató típusa (szűrőfile): CONSOL
Karakterméret : Normal
Sorköz : Normal
Sorok száma laponként : 46
Dátumformátum : Hungarian
Kimenet megnyitási módja : Write
Elemisor nyomtatása kell? : Yes
Lapemelés kell? : Yes
Első lap előtt kell emelés?: No
Utolsó lapemelés kell? : No

Elemi adatok száma : 3

Fejlécsor formátumok száma : 2
Elemisor formátumok száma : 2
Összegsor formátumok száma : 3

F.B.4.2. Fejléc, elemisor és összegsor formátumok

Fejléc formátum 1.: HEADER1
Magyarázó fejléc

Kód	Kód	Érték	TOTAL (összegző)			COUNT(számláló)				AVERAGE (átlag)			sor-szám
1.	2.		2.	1.	0.	-1.	0.	1.	2.	2.	1.	0.	

Elemisor formátum 1.: DETAIL1
A speciális hivatkozások

			TTTTT	TTTTT	TTTTT	CCC	CCC	CCC	CCC	AAAAAAA	AAAAAAA	AAAAAAA	SSSS
			TTTTT	TTTTT	TTTTT	CCC	CCC	CCC	CCC	AAAAAAA	AAAAAAA	AAAAAAA	SSSS
			TTTTT	TTTTT	TTTTT	CCC	CCC	CCC	CCC	AAAAAAA	AAAAAAA	AAAAAAA	SSSS

Összegsor formátum 2.: TOTAL2
Kontroll-váltás a 'kód 2.'-re
Kontrollváltás változója : kod2

CCC	CCC		TTTTT	TTTTT	TTTTT	CCC	CCC	CCC	CCC	AAAAAAA	AAAAAAA	AAAAAAA	SSSS

Összegsor formátum 1.: TOTAL1
Kontroll-váltás a 'kód 1.'-re
Kontrollváltás változója : kod1

CCC			TTTTT	TTTTT	TTTTT	CCC	CCC	CCC	CCC	AAAAAAA	AAAAAAA	AAAAAAA	SSSS

Végösszegző (Grand Total) formátum 0.:
Mindösszesen

				TTTTT	TTTTT	TTTTT	CCC	CCC	CCC	CCC	AAAAAAA	AAAAAAA	AAAAAAA	SSSS

F.C. Lekérdezési file-kapcsolatok

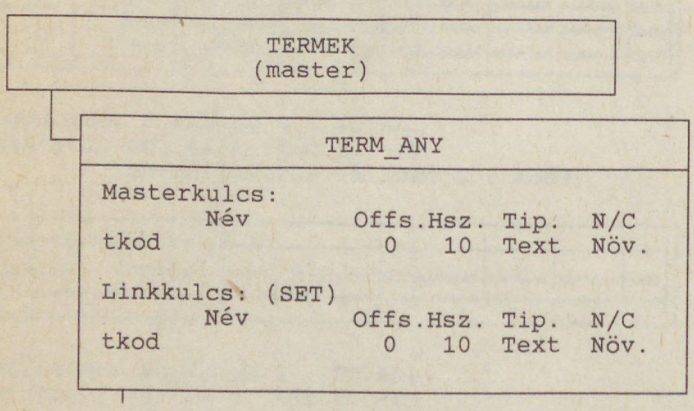
F.C.1. TERM_ANY

Query leírás: A termékek és a hozzátartozó anyagok lekérdezése

F.C.1.1. MISAM leírók

Leírófile	Alias név
TERMEK.FDF TERM_ANY.FDF ANYAG.FDF	TERMEK TERM_ANY ANYAG

F.C.1.2. Link struktúra



ANYAG

Masterkulcs:

Név	Offs.	Hsz.	Tip.	N/C
anyagkod	0	5	Text	Növ.

Linkkulcs: (UNIQUE)

Név	Offs.	Hsz.	Tip.	N/C
anyagkod	0	5	Text	Növ.

F.D. A TERM_ANY.RTX listaleíró file

```
REPORT=TERM_ANY,Termék kimutatás
SORTED=A termék kódjára
DEVICE=D
LSTFDIR=\WORK
LSTFNAME=WORK.LST
SERVER=FS1
CLASS=2
CHARSIZE=E
DETAILFIELDS=9
FIELD=tkod,A termék kódja,T,10,0,L
FIELD=tnev,A termék neve,T,40,0,L
FIELD=tmenny,A termék raktári mennyisége,D,14,2,R
FIELD=anyagkod,Az anyag kódja,T,5,0,L
FIELD=anyagnev,Az anyag megnevezése,T,30,0,L
FIELD=mennyiseg,A termékhez szükséges mennyiség,D,9,2,R
FIELD=egysegar,Az anyag egységára,D,9,2,R
FIELD=ertek,Az adott anyag értéke egységnyi termékben,D,9,2,R
FIELD=tertek,A termék raktári értéke,D,14,2,R
HEADER=1,TERMEK,,,Termék-anyag lista fejléce
HFORMAT=T,0,0,Termék összesítö lista
HFORMAT=S,date,0,0,35,8,0,L
HFORMAT=S,pageno,0,0,68,6,0,R
HFORMAT=T,0,74,.lap
HFORMAT=H,Detail_Header,2,1,0
HEADER=2,THEADER,,,Egy termék fejléce
HFORMAT=T,1,0,┌──────────────
```

```
HFORMAT=T,5,43,menny.(db)
HFORMAT=T,5,53,
HFORMAT=T,5,58,Ft
HFORMAT=T,5,65,
HFORMAT=T,5,67,értéke
HFORMAT=T,5,77,
HFORMAT=T,6,0,
DETAIL=1,ANYAGOK,,,A termékhez tartozó egy anyag adatai
DFORMAT=T,0,0,
DFORMAT=D,anyagkod,0,0,2,5,0,L
DFORMAT=T,0,8,
DFORMAT=D,anyagnev,0,0,10,30,0,L
DFORMAT=T,0,41,
DFORMAT=D,mennyiseg,0,0,43,9,2,R
DFORMAT=T,0,53,
DFORMAT=D,egysegar,0,0,55,9,2,R
DFORMAT=T,0,65,
DFORMAT=D,ertek,0,0,67,9,2,R
DFORMAT=T,0,77,
TOTAL=0,,,,A raktárban lévő termékek összértéke
TOTAL_ON=tertek
FORMAT=T,1,5,
FORMAT=T,2,5,
FORMAT=T,2,8,A raktárban található termékek összértéke:
FORMAT=L,tertek,0,2,51,14,2,R
FORMAT=T,2,66,Ft
FORMAT=T,2,70,
FORMAT=T,3,5,
TOTAL=1,TERMOSSZ,,,Termékenkénti összegző
CONTROL_ON=tkod
TOTAL_ON=ertek
TOTAL_ON=tertek
FORMAT=T,0,0,
FORMAT=T,1,0,
FORMAT=T,1,2,A(z)
FORMAT=C,,-1,1,7,5,0,M
FORMAT=T,1,13,anyagból álló termék egy egységének anyagértéke:
FORMAT=L,ertek,1,1,67,9,2,R
FORMAT=T,1,77,
FORMAT=T,2,0,
FORMAT=T,2,2,Raktárban található mennyiség:
FORMAT=O,tmenny,0,2,33,14,2,R
FORMAT=T,2,48,db.
FORMAT=T,2,54,Értéke:
FORMAT=L,tertek,1,2,62,14,2,R
FORMAT=T,2,77,
FORMAT=H,Total_Header,2,2,78
FORMAT=T,3,0,
```

F.E. A REPEDIT által generált TERM_ANY.H

```
/* "TERM_ANY.RH" generated by RepEdit (MI Software) */

#define R_tkod          0
#define R_tnev          1
#define R_tmenny        2
#define R_anyagkod      3
#define R_anyagnev      4
#define R_mennyiseg     5
#define R_egysegar      6
#define R_ertek         7
#define R_tertek        8

struct  R_TERM_ANY {
         char   tkod[10];
         char   tnev[40];
         double tmenny;
         char   anyagkod[5];
         char   anyagnev[30];
         double mennyiseg;
         double egysegar;
         double ertek;
         double tertek;
        };

/* end of TERM_ANY.RH */
```

F.F. A REPTEST által generált listakép

Kód	Kód	Érték	TOTAL (összegző)			COUNT(számláló)				AVERAGE (átlag)			sor-szám
1.	2.		2.	1.	0.	-1.	0.	1.	2.	2.	1.	0.	
1	1	1	1	1	1	1	1	0	0	1.00	1.00	1.00	6
1	1	2	3	3	3	2	2	0	0	1.50	1.50	1.50	7
1	1	3	6	6	6	3	3	0	0	2.00	2.00	2.00	8
1	1		6	6	6	3	3	0	1	2.00	2.00	2.00	10
1	2	4	4	10	10	1	4	0	1	4.00	2.50	2.50	12
1	2	5	9	15	15	2	5	0	1	4.50	3.00	3.00	13
1	2	6	15	21	21	3	6	0	1	5.00	3.50	3.50	14
1	2		15	21	21	3	6	0	2	5.00	3.50	3.50	16
1	3	7	7	28	28	1	7	0	2	7.00	4.00	4.00	18
1	3	8	15	36	36	2	8	0	2	7.50	4.50	4.50	19
1	3	9	24	45	45	3	9	0	2	8.00	5.00	5.00	20
1	3		24	45	45	3	9	0	3	8.00	5.00	5.00	22
1			24	45	45	3	9	1	3	8.00	5.00	5.00	24
2	1	10	10	10	55	1	10	1	0	10.00	1.00	5.50	26
2	1	11	21	21	66	2	11	1	0	10.50	1.91	6.00	27
2	1	12	33	33	78	3	12	1	0	11.00	2.75	6.50	28
2	1		33	33	78	3	12	1	1	11.00	2.75	6.50	30
2	2	13	13	46	91	1	13	1	1	13.00	3.54	7.00	32
2	2	14	27	60	105	2	14	1	1	13.50	4.29	7.50	33
2	2	15	42	75	120	3	15	1	1	14.00	5.00	8.00	34
2	2		42	75	120	3	15	1	2	14.00	5.00	8.00	36
2	3	16	16	91	136	1	16	1	2	16.00	5.69	8.50	38
2	3	17	33	108	153	2	17	1	2	16.50	6.35	9.00	39
2	3	18	51	126	171	3	18	1	2	17.00	7.00	9.50	40
2	3		51	126	171	3	18	1	3	17.00	7.00	9.50	42
2			51	126	171	3	18	2	3	17.00	7.00	9.50	44
3	1	19	19	19	190	1	19	2	0	19.00	1.00	10.00	46

Kód	Kód	Érték	TOTAL (összegző)			COUNT(számláló)				AVERAGE (átlag)			sor-szám
1.	2.		2.	1.	0.	-1.	0.	1.	2.	2.	1.	0.	
3	1	20	39	39	210	2	20	2	0	19.50	1.95	10.50	6
3	1	21	60	60	231	3	21	2	0	20.00	2.86	11.00	7
3	1		60	60	231	3	21	2	1	20.00	2.86	11.00	9
3	2	22	22	82	253	1	22	2	1	22.00	3.73	11.50	11
3	2	23	45	105	276	2	23	2	1	22.50	4.57	12.00	12
3	2	24	69	129	300	3	24	2	1	23.00	5.38	12.50	13
3	2		69	129	300	3	24	2	2	23.00	5.38	12.50	15
3	3	25	25	154	325	1	25	2	2	25.00	6.16	13.00	17
3	3	26	51	180	351	2	26	2	2	25.50	6.92	13.50	18
3	3	27	78	207	378	3	27	2	2	26.00	7.67	14.00	19
3	3		78	207	378	3	27	2	3	26.00	7.67	14.00	21
3			78	207	378	3	27	3	3	26.00	7.67	14.00	23
			78	207	378	3	27	3	3	26.00	7.67	14.00	25

F.G. Forrásprogramok

F.G.1. TERM_ANY.C

```
/* TERM_ANY.C
**
** Példaprogram a MIREP listagenerálási környezet használatához
** file- és lekérdezési rendszer nélkül.
**
** Használata:
**
** >TERM_ANY [-o output] [-c szürőfile]
**
** ahol        output     az elirányított output file vagy eszköz
**             szürőfile  a felülbírált nyomtatótipusnak megfelelő
**                        szürőfile
*/

#include <ctype.h>          /* standard C header                    */
#include <stdlib.h>         /* standard C header                    */
#include <stdio.h>          /* standard C header                    */
#include <string.h>         /* standard C header                    */

#include <mirep.h>          /* MIREP header                         */

#include "TERM_ANY.RH"      /* A REPEDIT által generált header      */

#define MAX_TERMEK      100 /* A termékek maximális száma           */
#define MAX_TERM_ANY    200 /* A termék-anyag kapcs. maximális száma*/
#define MAX_ANYAG       300 /* Az anyagok maximális száma           */

typedef struct TERMEK       /* A TERMEK file rekordja               */
        {
        char    tkod[10];   /* A termék kódja                       */
        char    tnev[40];   /* A termék neve                        */
        double  tmenny;     /* Raktári mennyiség                    */
        } STERMEK, *PTERMEK;

typedef struct TERM_ANY     /* Anyagösszetétel file rekordja        */
        {
        char    tkod[10];   /* A termék kódja                       */
        char    anyagkod[5];/* A hozzátartozó anyag kódja           */
        double  mennyiseg;  /* Az egységnyi termékhez szüks. menny. */
        } STERM_ANY, *PTERM_ANY;

typedef struct ANYAG        /* Az anyag törzsfile rekordja          */
        {
        char    anyagkod[5];/* Az anyag kódja                       */
        char    anyagnev[30];/* Az anyag neve                       */
```

```
        double  egysegar;       /* Az anyag egységára                  */
        } SANYAG, *PANYAG;

static  STERMEK     termek[MAX_TERMEK];         /* a beolvasott termékek*/
static  STERM_ANY   term_any[MAX_TERM_ANY+1];   /* -"- anyagösszetételek*/
static  SANYAG      anyag[MAX_ANYAG];           /* -"- anyagok          */

static  unsigned termek_szam = 0;       /* a beolvasott adatok száma    */
static  unsigned term_any_szam = 0;     /*              -"-             */
static  unsigned anyag_szam = 0;        /*              -"-             */

static  int     kulcs_hossz;            /* a "compare()" rendezéshez    */

static  int     read_datas(void);
static  int     compare(void *p1, void *p2);

void main(int argc, char **argv)
{
char            *s;             /* A program-opciók életelmezéséhez     */
int             t;              /* A termek rekord indexe               */
PTERMEK         pt;             /* A termek mutatója                    */
PTERM_ANY       pta;            /* Az összetétel mutatója               */
PANYAG          pa;             /* Az anyag mutatója                    */
double          ertek;          /* Anyag értéke egységnyi termékben     */
double          tertek;         /* A raktárban lévö termék értéke       */
int             rc;             /* Return Code: visszatérési érték      */

if (read_datas())
        exit(4);

if (r_read("TERM_ANY"))
        {
        fprintf(stderr,"A TERM_ANY.RTX file nem olvasható be!\n");
        exit(4);
        }

while ( (s = *++argv) != NULL)
        {
        if (*s != '-')
                {
                fprintf(stderr,"Hibás opció: %s!\n",s);
                exit(4);
                }
        ++s;                            /* A '-' átlépése               */

        switch (toupper(*s++))          /* az opció-betü átlépése       */
                {
                case 'O':               /* Output                       */
                        if (*s == '\0')
                                {
                                if ( (s = *++argv) == NULL)
```

```
                                {
                                fprintf(stderr,"Hiányzó output!\n");
                                exit(4);
                                }
                        }
                r_redirect(s);
                break;
        case 'C':               /* Szürőfile                    */
                if (*s == '\0')
                        {
                        if ( (s = *++argv) == NULL)
                                {
                                fprintf(stderr,"Hiányzó szürőfile!\n");
                                exit(4);
                                }
                        }
                r_set_prntype(s);
                break;
        default:
                fprintf(stderr,"Hibás opció: %s!\n",s);
                exit(4);
        }
    }

rc = r_init(NULL);
switch (rc)
    {
    case 0:
            break;
    case SPOOL_ERR:
            fprintf(stderr,"Spool nyitási hiba!\n");
            exit(4);
    case OUTOPEN_ERR:
            fprintf(stderr,"Az output nem nyitható meg: %s!\n",
                    r_get_lstfname());
            exit(4);
    case GETCONV_ERR:
            fprintf(stderr,"A szürőfile nem nyitható meg: %s!\n",
                    r_get_prntype());
            exit(4);
    default:
            fprintf(stderr,"Report inicializálási hiba! Kód: %d\n",rc);
            exit(4);
    }
rc = 0;


for (t = 0; t < termek_szam && rc == 0; ++t)
    {

    /* Az elemi adatbuffer feltöltése */
```

```
r_d_all_valued(0);              /* még nincs érték              */
pt = termek + t;
tertek = 0;
r_d_set_value(R_tkod,pt->tkod);
r_d_set_value(R_tnev,pt->tnev);
r_d_set_value(R_tmenny,&pt->tmenny);
r_d_set_value(R_tertek,&tertek);

kulcs_hossz = sizeof(pt->tkod);
pta = lsearch(  pt->tkod,
                term_any,
                &term_any_szam,
                sizeof(STERM_ANY),
                compare);

if (pta == NULL)                /* Nincs hozzátartozó anyag?    */
        {
        rc = r_exec();          /* Listázás                     */
        continue;               /* Következő termék             */
        }

kulcs_hossz = sizeof(pta->anyagkod);
while (memcmp(pt->tkod,pta->tkod,sizeof(pt->tkod)) == 0)
        {
        r_d_set_value(R_anyagkod,pta->anyagkod);
        r_d_set_value(R_mennyiseg,&pta->mennyiseg);
        pa = bsearch(pta->anyagkod,
                     anyag,
                     anyag_szam,
                     sizeof(SANYAG),
                     compare);
        if (pa == NULL)
                {
                r_d_valued(R_anyagnev,0);
                r_d_valued(R_egysegar,0);
                r_d_valued(R_ertek,0);
                r_d_valued(R_tertek,0);
                }
        else
                {
                r_d_set_value(R_anyagnev,pa->anyagnev);
                r_d_set_value(R_egysegar,&pa->egysegar);
                ertek = pta->mennyiseg * pa->egysegar;
                r_d_set_value(R_ertek,&ertek);
                tertek = ertek * pt->tmenny;
                r_d_set_value(R_tertek,&tertek);
                }
        rc = r_exec();          /* Listázás                     */

        if (rc != 0)            /* Nem normális végrehajtás?    */
                break;
```

```
                ++pta;                  /* Következő összetétel         */
                }
        }
if (rc)                                 /* Nem normális végrehajtás?    */
        {
        if (r_get_device() == CONSOLE)  /* Kezelői beavatkozás?         */
                {
                r_close();
                exit(0);
                }
        fprintf(stderr,"Report végrehajtás kódja: %d!\n",rc);
        r_close();
        exit(3);
        }

r_total_print(0);       /* A mindösszesen kiiratása     */
r_close();              /* Lista lezárása               */
exit(0);
}

/* Az input file-ok formátuma:
**
** TERMEK.DAT:
** t.kód      terméknév                                    t.menny
** KKKKKKKKKK NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN MMMM.MM
** 01234567890123456789012345678901234567890123456789012345678
**           1         2         3         4         5
**
** TERM_ANY.DAT:
** t.kód      a.kód menny.
** KKKKKKKKKK NNNNN MMMM.MM
** 0123456789012345678901 23
**           1         2
**
** ANYAG.DAT:
** a.kód  anyagnév                          e.ár
** KKKKK NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN AAAA.AA
** 01234567890123456789012345678901234567890123
**           1         2         3         4
*/

#define T_TKOD          0
#define T_TNEV          (sizeof(termek[0].tkod)+1)
#define T_TMENNY        (T_TNEV + sizeof(termek[0].tnev)+1)

#define TA_TKOD         0
#define TA_AKOD         (sizeof(term_any[0].tkod)+1)
#define TA_MENNY        (TA_AKOD + sizeof(term_any[0].anyagkod)+1)

#define A_AKOD          0
#define A_ANEV          (sizeof(anyag[0].anyagkod)+1)
```

```
#define A_EGYS          (A_ANEV + sizeof(anyag[0].anyagnev)+1)

#define MAXBUF  65

static int read_datas()
/* Beolvassa az "adatbázist" */
{
FILE            *fp;                    /* Az aktualis input file       */
char            buffer[MAXBUF];         /* Buffer beolvasáshoz          */
PTERMEK         pt;                     /* A termek mutatója            */
PTERM_ANY       pta;                    /* Az összetétel mutatója       */
PANYAG          pa;                     /* Az anyag mutatója            */

/* Termékek beolvasása */

fp = fopen("TERMEK.DAT","r");
if (fp == NULL)
        {
        fprintf(stderr,"A TERMEK.DAT törzs-file nem olvasható be!\n");
        return(-1);
        }
pt = termek;
while (fgets(buffer,MAXBUF,fp) != NULL)
        {
        if (termek_szam >= MAX_TERMEK)
                {
                fprintf(stderr,
                        "Túl sok rekord a TERMEK.DAT törzs-file-ban!\n");
                fclose(fp);
                return(-1);
                }
        memcpy(pt->tkod,buffer + T_TKOD,sizeof(pt->tkod));
        memcpy(pt->tnev,buffer + T_TNEV,sizeof(pt->tnev));
        pt->tmenny = atof(buffer + T_TMENNY);
        ++termek_szam;
        ++pt;
        }
fclose(fp);

/* Termékösszetételek beolvasása */

fp = fopen("TERM_ANY.DAT","r");
if (fp == NULL)
        {
        fprintf(stderr,"A TERM_ANY.DAT file nem olvasható be!\n");
        return(-1);
        }
pta = term_any;
while (fgets(buffer,MAXBUF,fp) != NULL)
        {
        if (term_any_szam >= MAX_TERM_ANY)
```

```
                {
                fprintf(stderr,
                        "Túl sok rekord a TERM_ANY.DAT file-ban!\n");
                fclose(fp);
                return(-1);
                }
        memcpy(pta->tkod    ,buffer + TA_TKOD,sizeof(pta->tkod));
        memcpy(pta->anyagkod,buffer + TA_AKOD,sizeof(pta->anyagkod));
        pta->mennyiseg = atof(buffer + TA_MENNY);
        ++term_any_szam;
        ++pta;
        }
fclose(fp);
memset(pta,0,sizeof(STERM_ANY));                /* strázsa kereséshez   */

/* Anyagok beolvasása */

fp = fopen("ANYAG.DAT","r");
if (fp == NULL)
        {
        fprintf(stderr,"Az ANYAG.DAT törzs-file nem olvasható be!\n");
        return(-1);
        }
pa = anyag;
while (fgets(buffer,MAXBUF,fp) != NULL)
        {
        if (anyag_szam >= MAX_ANYAG)
                {
                fprintf(stderr,
                        "Túl sok rekord az ANYAG.DAT törzs-file-ban!\n");
                fclose(fp);
                return(-1);
                }
        memcpy(pa->anyagkod,buffer + A_AKOD,sizeof(pa->anyagkod));
        memcpy(pa->anyagnev,buffer + A_ANEV,sizeof(pa->anyagnev));
        pa->egysegar = atof(buffer + A_EGYS);
        ++anyag_szam;
        ++pa;
        }
fclose(fp);

/* A beolvasott adatok rendezése */

kulcs_hossz = sizeof(pt->tkod);
qsort(termek,termek_szam,sizeof(STERMEK),compare);

kulcs_hossz = sizeof(pta->tkod) + sizeof(pta->anyagkod);
qsort(term_any,term_any_szam,sizeof(STERM_ANY),compare);

kulcs_hossz = sizeof(pa->anyagkod);
qsort(anyag,anyag_szam,sizeof(SANYAG),compare);
```

```
return(0);                          /* sikeres beolvasás    */
}

static int compare(p1,p2)
/* Összehasonlítja a két elemet "kulcs_hossz" hosszon */
void    *p1;
void    *p2;
{
return(memcmp(p1,p2,kulcs_hossz));
}
```

F.G.2. TERM_QRY.C

```
/* TERM_QRY.C
**
** Példaprogram a MIREP listagenerálási környezet használatához
** MISAM file- és MIQUEL lekérdezési rendszer esetén.
**
** Használata:
**
** >TERM_QRY [-o output] [-c szűrőfile]
**
** ahol        output     az elirányított output file vagy eszköz
**             szűrőfile  a felülbírált nyomtatótipusnak megfelelő
**                        szűrőfile
*/

#include <ctype.h>          /* standard C header                  */
#include <stdlib.h>         /* standard C header                  */
#include <stdio.h>          /* standard C header                  */

#include <mirep.h>          /* MIREP header                       */
#include <misam.h>          /* MISAM header                       */
#include <miquel.h>         /* MIQUEL header                      */

#include "TERM_ANY.RH"      /* A REPEDIT által generált header    */
#include "TERM_ANY.QH"      /* A MIQUEL rendszer QRYEDIT-je  -"-  */

static  void    fill_buffer(void);

void main(argc,argv)
int     argc;
char    **argv;
{
char    *s;                 /* A program-opciók éretelmezéséhez   */
int     rc;                 /* Return Code: visszatérési érték    */

if (r_read("TERM_ANY"))
        {
```

```
        fprintf(stderr,"A TERM_ANY.RTX file nem olvasható be!\n");
        exit(4);
        }

while ( (s = *++argv) != NULL)
        {
        if (*s != '-')
                {
                fprintf(stderr,"Hibás opció: %s!\n",s);
                exit(4);
                }
        ++s;                            /* A '-' átlépése               */

        switch (toupper(*s++))          /* az opció-betü átlépése       */
                {
                case 'O':               /* Output                       */
                        if (*s == '\0')
                                {
                                if ( (s = *++argv) == NULL)
                                        {
                                        fprintf(stderr,"Hiányzó output!\n");
                                        exit(4);
                                        }
                                }
                        r_redirect(s);
                        break;
                case 'C':               /* Szürőfile                    */
                        if (*s == '\0')
                                {
                                if ( (s = *++argv) == NULL)
                                        {
                                        fprintf(stderr,"Hiányzó szürőfile!\n");
                                        exit(4);
                                        }
                                }
                        r_set_prntype(s);
                        break;
                default:
                        fprintf(stderr,"Hibás opció: %s!\n",s);
                        exit(4);
                }
        }

rc = r_init(NULL);
switch (rc)
        {
        case 0:
                break;
        case SPOOL_ERR:
                fprintf(stderr,"Spool nyitási hiba!\n");
                exit(4);
```

```
        case OUTOPEN_ERR:
                fprintf(stderr,"Az output nem nyitható meg: %s!\n",
                        r_get_lstfname());
                exit(4);
        case GETCONV_ERR:
                fprintf(stderr,"A szűrőfile nem nyitható meg: %s!\n",
                        r_get_prntype());
                exit(4);
        default:
                fprintf(stderr,"Report inicializálási hiba! Kód: %d\n",rc);
                exit(4);
        }

if (q_open("TERM_ANY",(struct CFUNC *)NULL))
        {
        fprintf(stderr,"A TERM_ANY.QTX query-leíró file nem olvasható be!\n");
        exit(4);
        }

if ( (rc = q_init()) != SUCCESS && rc != NO_MATCH)
        {
        fprintf(stderr,"Query inicializálási hiba: %s!\n",q_errmsg);
        r_close();
        exit(4);
        }

while (rc == SUCCESS || rc == NO_MATCH)
        {
        fill_buffer();
        rc = r_exec();
        if (rc != 0)                        /* Hibás végrehajtás?   */
                {
                if (r_get_device() == CONSOLE)  /* Kezelői beavatkozás? */
                        {
                        q_close();
                        r_close();
                        exit(0);
                        }
                fprintf(stderr,"Report végrehajtás kódja: %d!\n",rc);
                q_close();
                r_close();
                exit(3);
                }
        rc = q_exec(NEXT);
        }

if (rc == Q_EOF || rc == Q_LKEY)
        {
        r_total_print(0);               /* A mindösszesen kiiratása     */
        q_close();                      /* File-ok lezárása             */
        r_close();                      /* Lista lezárása               */
```

```
        exit(0);
        }

/* Hibaág: nem szabályos QUERY lefutás */

fprintf(stderr,"Query hiba: %s!\n",q_errmsg);
q_close();
r_close();
exit(3);
}

static void fill_buffer()
/* Az elemi adatbuffer feltöltése */
{
char    buffer[80];             /* buffer a szöveges mezők számára      */
double  tmenny;                 /* file-ból                             */
double  mennyiseg;              /* file-ból                             */
double  egysegar;               /* file-ból                             */
double  ertek;                  /* számított érték                      */
double  tertek;                 /* számított érték                      */

/* szöveges mezők */

q_field_get (QF_tkod,buffer);
r_d_set_value(R_tkod,buffer);

q_field_get (QF_tnev,buffer);
r_d_set_value(R_tnev,buffer);

q_field_get (QF_anyagkod,buffer);
r_d_set_value(R_anyagkod,buffer);

q_field_get (QF_anyagnev,buffer);
r_d_set_value(R_anyagnev,buffer);

/* numerikus, a számításhoz szükséges mezők */

q_field_get (QF_tmenny,&tmenny);
r_d_set_value(R_tmenny,&tmenny);

q_field_get (QF_mennyiseg,&mennyiseg);
r_d_set_value(R_mennyiseg,&mennyiseg);

q_field_get (QF_egysegar,&egysegar);
r_d_set_value(R_egysegar,&egysegar);

/* A számított mezők */

ertek = mennyiseg * egysegar;
r_d_set_value(R_ertek,&ertek);
```

```
tertek = ertek * tmenny;
r_d_set_value(R_tertek,&tertek);
}
```

F.G.3. LIST.C

```
/* LIST.C
**
** A MIREP, MIQUEL és MISAM környezetre épülő általános listagenerátor
*/

#include <ctype.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#include <misam.h>
#include <miquel.h>
#include <mirep.h>

#define MAXFN   81
#define YES     1
#define NO      0

int     MI_language = 2;                /* Magyar nyelv         */

static  char    inpname[MAXFN];         /* MISAM/MIQUEL leíró   */
static  char    rtxname[MAXFN];         /* MIREP leíró          */
static  char    ptcname[MAXFN];         /* szűrőfile            */
static  char    outname[MAXFN];         /* output eszköz/file   */
static  int     is_fdf;                 /* MISAM leíró?         */

static  void    options(char **argv);
static  void    usage(char *msg);
static  void    fill_buffer(void );

void main(int argc,char **argv)
{
int     rrc;                    /* MIREP visszatérési érték     */
int     qrc;                    /* MIQUEL visszatérési érték    */
int     exitcode;               /* programkiszállási kód        */

options(argv);

if (r_read(rtxname))
        exit(4);
if (is_fdf)
        {
        if (q_one_file(inpname,NULL))
```

```
                exit(4);
        }
else
        {
        if (q_open(inpname,NULL))
                exit(4);
        }
if (ptcname[0])
        r_set_prntype(ptcname);
if (outname[0])
        r_redirect(outname);

rrc = r_init(NULL);

switch (rrc)
        {
        case 0:
                break;
        case SPOOL_ERR:
                fprintf(stderr,"Spool megnyitási hiba!\n");
                exit(4);
        case OUTOPEN_ERR:
                fprintf(stderr,"Az output nem nyitható meg: %s\n",
                                r_get_lstfname());
                exit(4);
        case GETCONV_ERR:
                fprintf(stderr,"A szüröfile nem nyitható meg:",
                                r_get_prntype());
                exit(4);
        default:
                fprintf(stderr,"MIREP inicializálási hiba %d!\n:",
                                rrc);
                exit(4);
        }

isnetwork();

qrc = q_init();

switch (qrc)
        {
        case SUCCESS:
        case NO_MATCH:
                break;
        case Q_LKEY:
        case Q_FKEY:
        case Q_EOF:
                fprintf(stderr,"Nincs listázandó rekord!\n");
                r_close();
                q_close();
                exit(1);
```

```
        default:
                fprintf(stderr,"MIQUEL init hiba: %s\n",q_errmsg);
                r_close();
                q_close();
                exit(3);
        }

while (qrc == SUCCESS || qrc == NO_MATCH)
        {
        fill_buffer();
        if ((rrc = r_exec()) != 0)
                {
                if (r_get_device() != CONSOLE)
                        fprintf(stderr,"MIREP Exec hiba: %d\n",rrc);
                break;
                }
        qrc = q_exec(NEXT);
        }

switch (qrc)
        {
        case Q_EOF:
        case Q_LKEY:
                r_total_print(0);
                exitcode = 0;
                break;
        case Q_LERR:
                fprintf(stderr,"Foglalt file: %s\n",q_errmsg);
                exitcode = 3;
                break;
        default:
                fprintf(stderr,"MIQUEL végrehajtási hiba: %s\n",q_errmsg);
                exitcode = 3;
                break;
        }
q_close();
r_close();
exit(exitcode);
}

static void fill_buffer()
/* A MIQUEL bufferből feltölti az elemi adatbuffert */
{
int     n;                      /* A MIREP változók száma               */
int     r;                      /* az aktuális MIREP változó            */
char    *rn;                    /* a MIREP változó neve                 */
int     q;                      /* a MIQUEL változó sorszáma            */
char    buffer[256];            /* buffer az áttöltéshez                */

r_d_all_valued(0);              /* a MIREP változók-nak nincs értékük   */
```

```
n = r_d_number();

for (r = 0; r < n; ++r)
        {
        rn = r_d_name(r);
        q = q_field_sernum(rn);
        if (q < 0)
                continue;       /* nincs megfelelő MIQUEL változó       */

        if (q_field_is_valued(q) == NO)
                continue;       /* a MIQUEL változónak nincs értéke     */

        q_field_get(q,buffer);
        r_d_set_value(r,buffer);
        }
}

static void options(argv)
/* Elemzi a program parancssorát */
char    **argv;
{
register char   *s;

inpname[0] = '\0';
rtxname[0] = '\0';
ptcname[0] = '\0';
outname[0] = '\0';

while ( ( s = *++argv) != NULL)
        {
        if (*s != '-')
                usage("Hibás használat");
        ++s;
        switch (toupper(*s++))  {

        case 'C':                       /* -c opció     */
                if (ptcname[0])
                        usage("Csak egy szűrőfile adható meg");
                if (*s == '\0')
                        {
                        if ( (s = *++argv) == NULL)
                                usage("Hiányzó szűrőfile");
                        }
                strcpy(ptcname,s);
                strupr(ptcname);
                break;
        case 'F':                       /* -f opció     */
                if (inpname[0])
                        usage("Csak egy MISAM/MIQUEL leíró adható meg");
                if (*s == '\0')
                        {
```

```
                if ( (s = *++argv) == NULL)
                        usage("Hiányzó MISAM leíró");
                }
        strcpy(inpname,s);
        strupr(inpname);
        is_fdf = YES;
        break;
    case 'H':                       /* -h: HELP kérése      */
        usage(NULL);
    case 'O':                       /* -o opció     */
        if (outname[0])
                usage("Csak egy output adható meg");
        if (*s == '\0')
                {
                if ( (s = *++argv) == NULL)
                        usage("Hiányzó output");
                }
        strcpy(outname,s);
        strupr(outname);
        break;
    case 'Q':                       /* -q opció             */
        if (inpname[0])
                usage("Csak egy MISAM/MIQUEL leíró adható meg");
        if (*s == '\0')
                {
                if ( (s = *++argv) == NULL)
                        usage("Hiányzó MIQUEL leíró");
                }
        strcpy(inpname,s);
        strupr(inpname);
        is_fdf = NO;
        break;
    case 'R':                       /* -r opció             */
        if (rtxname[0])
                usage("Csak egy MIREP leíró adható meg");
        if (*s == '\0')
                {
                if ( (s = *++argv) == NULL)
                        usage("Hiányzó MIREP leíró");
                }
        strcpy(rtxname,s);
        strupr(rtxname);
        break;
    default:
        usage("Hibás használat");
        }
    }
if (inpname[0] == '\0')
    usage("Hiányzó MISAM/MIQUEL leíró");
if (rtxname[0] == '\0')
    usage("Hiányzó MIREP leíró");
```

```
}

static void usage(char *msg)
/* Kiírja az opcionális hibaüzenet, a használat módját és kiszáll */
{
if (msg && *msg)
        fprintf(stderr,"\nHiba: %s!\n",msg);

fprintf(stderr,"\n\
Használat: >LIST -f fdf -q qtx -r rtx [-c ptc] [-o output]\n\
\n\
ahol    fdf     a MISAM állomány-leíró file neve\n\
        qtx     a MIQUEL lekérdezés-leíró file neve\n\
        rtx     a MIREP lista-leíró file neve\n\
        ptc     a szürö-file neve\n\
        output  az output eszköz vagy file neve\n");
exit(4);
}
```

I n d e x

Adathivatkozás 12
Adatlekérdező és módosító rutinok 103
Aláhúzás 33, 54, 131
Alapfogalmak és definíciók 7
Állapototkezelő rutinok 99
Általános jellemzők 147, 148, 149, 151
ANYAG 144, 147
Átlag 26, 48
Áttekintés 3
Bevezetés 1
C függvénytábla generálása 64
CHARSIZE= 18, 60, 132
Char_Size 30, 52, 132
CLASS= 17, 60
CONTROL_ON= 36
COPYNUM= 17, 60
Cr_Lf 30, 52
Csatlakozási pontok rutinjai 124
DATEMODE= 20, 62
A dátum kiírási formátuma 62
DETAIL= 33
DETAILFIELDS= 22
Detail_Header 29, 51
Detail_Print 29, 51
Detail_Select 29, 51
DETPRINT= 20, 62
DEVICE= 16, 59
DFORMAT= 33
A disztributív anyag 141
a DOG szövegformázó környezet 129
Dőlt betűs írás 33, 54, 132
Dupla írás 33, 54, 132
Az editált lista lemezre írása 63
Egy általános listázóprogram 75
Egyéb parancsok 64
Egyéb rutinok 121
Ékezetes betűk 12, 14, 27, 49, 57, 127, 128, 133
Elemi adatterület 8
Az elemi adatterület leírása 22, 41
Elemi változó 10, 24, 48
Elemi változó előző értéke 25, 48
Elemi változó létrehozása 42
Elemi változó módosítása 43
Elemi változó törlése 43
Az elemi változók kezelőrutinjai 110
Elemisor 10
Az elemisor formátuma 10, 33
Az elemisor kiírásának letiltása 20, 62, 93, 116, 118, 121
Előkészítő rutinok 77
Az első lapemelés kérése 21, 63, 81, 88

Fejléc, elemisor és összegsor formátumok 148, 149, 150, 152
Fejlécsor 11
A fejlécsor formátuma 11, 34
Felhasználói C függvény 93, 94, 97, 99, 102
Felhasználói C függvények 14, 57, 126
FIELD= 22
File-leírások 144
FORMAT= 37
Formátum 10
Formátum kiválasztása 45
Formátum másolása 56
A formátum neve és leírása 56
Formátum törlése 56
Formátumelem 10
Formátumelem létrehozása 46
Formátumelem másolása 56
Formátumelem módosítása 54
Formátumelem mozgatása 55
Formátumelem törlése 55
Formátumelemek 23
Formátumkezelő rutinok 114
Formátumok megadása 44
FORMFEED= 21, 63
Forrásprogramok 161
FRSTFF= 21, 63
Az FTXTORTX, MIFORM konvertáló 71
Függelékek 143
A hálózati nyomtatás osztálya 60
Hálózati nyomtató 13
Hálózati nyomtató megadása 137
A hálózati server neve 59
Hashmark parancs 13, 27, 49
HEADER= 34
Headerfile generálása 64
Header_Print 30, 51
Header_Select 29, 51
HFORMAT= 35
Illeszkedési pont 14
Karakterkonverzió 135
A karakterméret 60
Képernyőre küldött listák 138
Kiírási szélességek ellenőrzése 65
Kiszállás a rendszerből 67
Kiszállás az operációs rendszerbe 66
Kvázigrafikus karakterek 49, 134
Lapszám 12
LASTFF= 21, 63
Leírások ellenőrzése 65
Lekérdezésen alapuló listázás 75
Lekérdezési file-kapcsolatok 154
Lemezre küldött listák 138
LINESPACING= 19, 61, 132

Line_Spacing 31, 52, 132
Link struktúra 154
LIST.C 172
Lista 7
Listadefiníció 7
A listadefiníció leírása 58
Listaformátumok 147
Listagenerátor 8
A listagenerátor alapelvei 4
Listajellemzők beállítása 85
Listajellemzők lekérdezése 79
Listák készítése 73
Listaleíró file 7
A listaleíró nyelv 15
A listázás környezete 127
Listázó keretprogram 75
Lokális nyomtató 13
Lokális nyomtató megadása 137
LSTFDIR= 16, 59
LSTFNAME= 16, 59
Makrózás 66
MFORM 3
A MIREP futtató rendszere 77
MISAM leírók 154
A MIxxx adatfeldolgozási eszköztár 3
Nem használt elemi változók 65
New_Page 28, 50
A nyomtatási lap hossza 61
Nyomtatási példányszám 60
A nyomtatási sortávolság 61
Nyomtató definíció 14
A nyomtató típusa 60
Nyomtatók illesztése 127
Opciók állítása 15, 57
OPENMODE= 20, 62
Összegsor 11
Az összegsor formátuma 11, 36
Az összegzési szintek beállítása 65
Összegző 25, 48
Az output eszköz 59
Az output lista megnyitási módja 62
Az output listafile könyvtára 59
Az output listafile neve 59
PAGELENGTH= 19, 61
Page_Length 32, 53
Page_No 32, 53
Pozícionálás a formátumon belül 46
Printer_Control 32, 53
PRNTYPE= 17, 60
A PTCED indítása 128
A PTCED szűrőfile editor 14, 128
A QTXTORTX, mint MIQUEL konvertáló 71

A QTXTORTX, mint MISAM konvertáló 70
A rendezettség leírása 59
Rendszerdátum 12
Rendszerváltozó 12, 24, 47
A REPDOC listadokumenter 69
A REPEDIT által generált TERM_ANY.H 158
A REPEDIT hívása és opciói 39
A REPEDIT report editor 39
REPORT= 15, 58
A REPTEST által generált listakép 159
A REPTEST listatesztelő 70
r_act_detail() 99
r_act_header() 99
r_act_total() 100
r_atodat() 103
r_buflen() 105
r_charsize() 92
r_check() 121
r_close() 92
r_cmd_need_arg() 121
r_control_check() 122
r_convdat() 104
r_create_report() 122
r_cummulate() 122
r_detail_header() 93
r_detail_name() 114
r_detail_number() 115
r_detail_onoff() 115
r_detail_print() 93
r_detail_select() 115
r_detail_sernum() 116
r_d_all_valued() 104
r_d_buffer() 104
r_d_dec() 110
r_d_get_value() 105
r_d_is_total() 110
r_d_is_valued() 105
r_d_length() 111
r_d_name() 111
r_d_number() 111
r_d_offset() 112
r_d_sernum() 112
r_d_set_value() 106
r_d_size() 112
r_d_total_onoff() 113
r_d_type() 113
r_d_valued() 106
r_endspool() 123
r_exec() 94
r_exec_func() 123
r_fill_page() 94
r_get_charsize() 79

r_get_class() 79
r_get_copynum() 79
r_get_count() 107
r_get_datemode() 80
r_get_detprint() 116
r_get_device() 80
r_get_dscr() 80
r_get_first() 100
r_get_formf() 81
r_get_frstff() 81
r_get_jump() 101
r_get_lastff() 81
r_get_linespacing() 82
r_get_lstfdir() 82
r_get_lstfname() 82
r_get_name() 83
r_get_openmode() 83
r_get_pagelength() 83
r_get_printfp() 101
r_get_prntype() 84
r_get_server() 84
r_get_sorted() 84
r_get_total() 107
r_globprn_status() 101
r_header_name() 116
r_header_number() 117
r_header_onoff() 117
r_header_print() 95
r_header_select() 117
r_header_sernum() 118
r_init() 95
r_linespacing() 96
r_new_page() 96
r_o_get_value() 107
r_printer_ctr() 96
r_read() 78
r_redirect() 85
r_searchcontrol() 118
r_select() 78
r_set_after() 125
r_set_before() 125
r_set_charsize() 85
r_set_class() 86
r_set_copynum() 86
r_set_count() 108
r_set_datemode() 86
r_set_detprint() 118
r_set_device() 87
r_set_dscr() 87
r_set_first() 102
r_set_formf() 88
r_set_frstff() 88

r_set_funcs() 124
r_set_jump() 102
r_set_lastff() 88
r_set_linespacing() 89
r_set_lstfdir() 89
r_set_lstfname() 89
r_set_name() 90
r_set_openmode() 90
r_set_pagelength() 90
r_set_printfp() 102
r_set_prntype() 91
r_set_server() 91
r_set_sorted() 91
r_set_total() 108
r_spool() 123
r_status() 103
r_sysdate() 109
r_s_get_value() 109
r_s_set_value() 109
r_total_header() 97
r_total_init() 97
r_total_name() 119
r_total_number() 119
r_total_only_more() 119
r_total_onoff() 120
r_total_page() 98
r_total_print() 98
r_total_sernum() 120
r_total_to_detail() 121
r_uniq_total() 113
r_write() 124
Segédprogramok 69
SERVER= 17, 59
SORTED= 15, 59
Standard listakezelési funkciók 138
A standard nyomtatóvezérlések 18, 19, 31, 33, 52, 53, 54, 61, 131
Számláló 26, 48
Szövegkonstans 12, 27, 49
Szürőfile 14, 18, 19, 31, 33, 52, 53, 54, 61
A szürőfile tesztelése 136
Szürőfile-ok 127
Teljes lapkitöltés kérése 63
TERMEK 145, 148
TERM_ANY 145, 149, 154
TERM_ANY.C 161
A TERM_ANY.RTX listaleíró file 156
TERM_QRY.C 168
TESTTECH 151
TOTAL= 36
Total_Header 28, 50
TOTAL_ON= 36

Total_Page 28, 50
USER vezérlő szekvenciák 33, 54, 97, 133
Utolsó lapemelés letiltása 63
Végrehajtó rutinok 92

Bibliotheca Academiae Scientiarum
MAGYAR TUDOMÁNYOS AKADÉMIA KÖNYVTÁRA

MIREP

C nyelvű listagenerálási eszköztár

A hálózati adatfeldolgozó programrendszerek fejlesztésének sok szempontból előnyös módszere a hagyományos programnyelvekre épül. Nagy hátránya a munkaigényesség, ami adatfeldolgozást támogató tárgykönyvtárak használatával csökkenthető. Az adatfeldolgozó rendszerek listázással kapcsolatos részének elkészítését könnyíti a MIREP programcsomag.

A MIREP listagenerátor jellemzői:

- A listákat leírófájl alapján kezeli.
- A futtatórendszernek ciklikusan átadott elemi buffer alapján készíti a kívánt listát.
- Automatikusan kezeli
  - a lapváltást,
  - a kontrollváltásokat,
  - az összegzéseket,
  - a számlálásokat,
  - az átlagképzéseket.
- Dinamikusan váltható fejléceket enged meg.
- Tetszőleges számú, dinamikusan váltható részletező sor, ill. sorok kezelését teszi lehetővé.
- A kontroll-fokozat nélküli végösszegzésen kívül tetszőleges számú kontrollváltás, illetve részösszegzés automatikus végrehajtását támogatja.
- Lehetőség van a rugalmas nyomtatóillesztésekre, munkaállomáshoz tartozó vagy hálózati nyomtatókon való listázásokra.
- Segítséget nyújt a listák tervezéséhez és dokumentálásához.

A lekérdező és a file-kezelő rendszerre kiválóan alkalmasak a MIQUEL és a MISAM programcsomagok, amelyek által előállított „elemi adatbuffer" közvetlenül megfeleltethető a MIREP elemi adatbufferének. A MIREP rendszer csatlakozik az ADAPT rendszerszervezői programcsomaghoz.

1828