

Számítástechnikai sorozat 8.

ÖSSZEÁLLÍTOTTA: BÖLCSFÖLDI JÓZSEF

Az OS Job Control nyelve

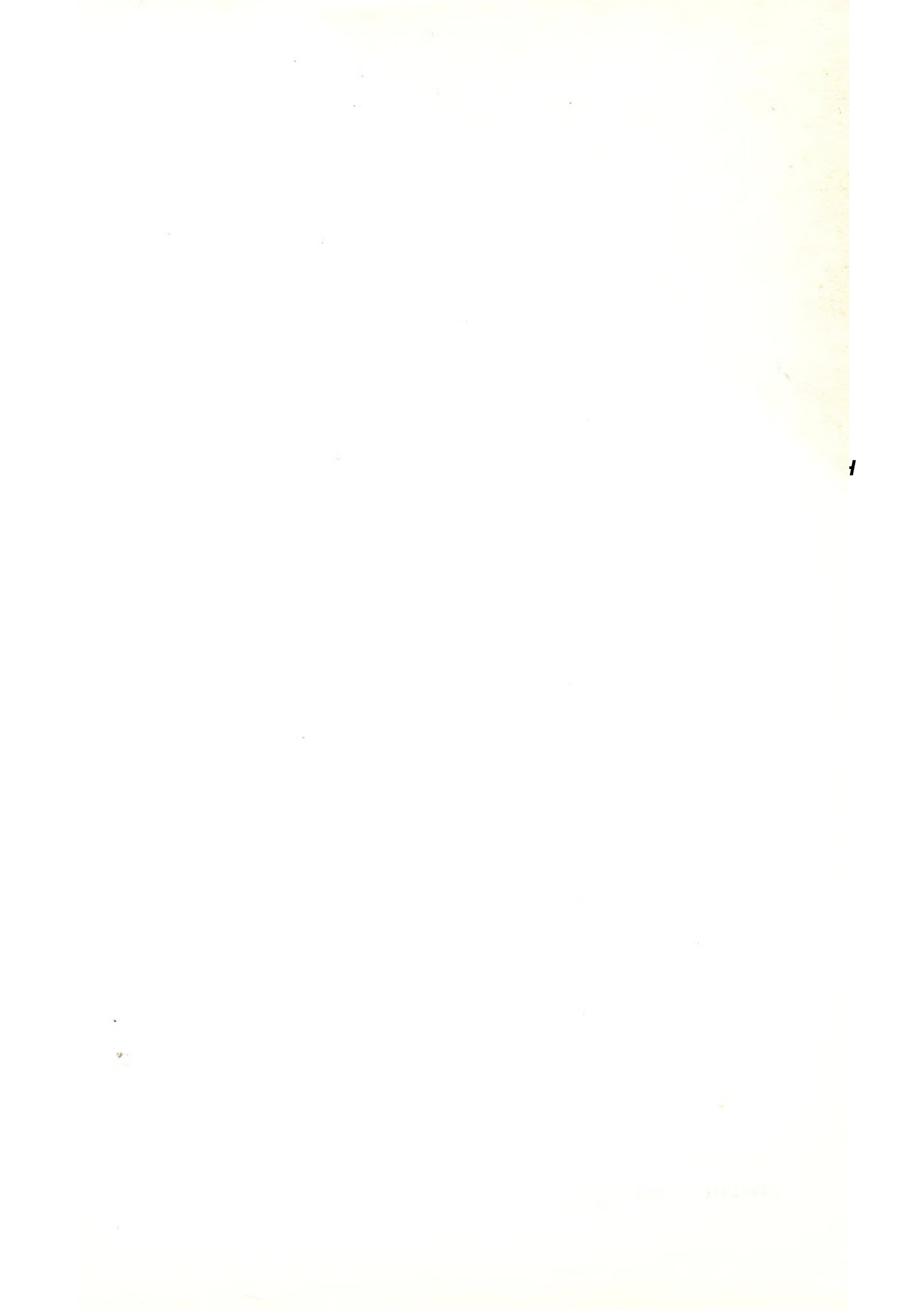


STATISZTIKAI KIADÓ VÁLLALAT





**Az OS Job Control
nyelve**



Számítástechnikai sorozat 8.

Az OS Job Control nyelve

STATISZTIKAI KIADÓ VÁLLALAT, BUDAPEST 1977

Összeállította:

Bölcsföldi József

Lektorálta:

Kovács János

ISBN: 963 340 158 5

© Statisztikai Kiadó Vállalat, 1977



Tartalomjegyzék

	Kulcsszó	Oldal
ELŐSZÓ	-	11
1. BEVEZETÉS	-	12
1.1 ESZR operációs rendszerek	-	12
1.2 Az OS felépítése és működése (vázlatosan)	-	12
1.2.1 A job-kezelés	-	12
1.2.2 Task-kezelés	-	13
1.2.3 Információkezelés	-	13
2. AZ OS JCL NYELV	-	15
2.1 A JCL nyelv rendeltetése, előnye	-	15
2.2 Utasításkészlet	-	15
2.3 Általános formai szabályok	-	16
2.3.1 A JCL utasításokról általában	-	17
2.3.1.1 A JCL utasítás általános formája	-	17
2.3.1.2 Az egyes utasításmezők funkciója és az általános formális szabályok	-	18
2.3.2 JCL utasítás folytatása új sorban	-	19
2.4 A job utasítás	JOB	19
2.4.1 A job-utasítás pozicionális paraméterei	-	20
2.4.1.1 Elszámolási információ	-	20
2.4.1.2 A programozó neve	-	22
2.4.2 A job-utasítás kulcsszavas paraméterei	-	23
2.4.2.1 A job futásának megszakítása	COND	23
2.4.2.2 A végrehajtási lista szabályozása	MSGLEVEL	24
2.4.2.3 Késleltetett újraindítás megadott kópia azonosítótól abnormális befejeződés esetén	RESTART	26

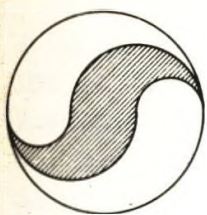
	Kulcsszó	Oldal
2.4.2.4	Automatikus újraindítás engedélyezése, illetve eltiltása	RD 27
2.4.2.5	A job osztályba sorolása	CLASS 28
2.4.2.6	A rendszer üzeneteinek osztályba sorolása	MSGCLASS 29
2.4.2.7	A job prioritásának megadása	PRTY 30
2.4.2.8	A job futási idejének korlátozása	TIME 31
2.4.2.9	Job visszatartása a job-input várakozási sorban (input-queue-ban)	TYPRUN 31
2.4.2.10	A job helyigényének megadása	REGION 31
2.4.2.11	A job által felhasznált terület átengedése más job-oknak és megfordítva	ROLL 32
2.4.3	Példák a job-utasítás paraméterezésére	- 33
2.5	Az EXEC utasítás	EXEC 33
2.5.1	Az EXEC utasítás pozicionális paramétere	PGM, PROC 34
2.5.2	Az EXEC utasítás kulcsszavas paramétere	- 36
2.5.2.1	Elszámolási információ megadása	ACCT 36
2.5.2.2	A lépés végrehajtásának feltételhez kötése	COND 37
2.5.2.3	A végrehajtandó program paraméterezése JCL szinten	PARM 38
2.5.2.4	A lépés automatikus újraindításának engedélyezése, illetve tiltása	RD 40
2.5.2.5	A lépés futási idejének korlátozása	TIME 40
2.5.2.6	A lépés prioritásának megadása	DPRTY 41
2.5.2.7	A lépés helyigényének megadása	REGION 42
2.5.2.8	A lépés által felhasznált terület átengedése más job-oknak és megfordítva	ROLL 43

		Kulcsszó	Oldal
2.6	A DD utasítás	DD	43
2.6.1	A DD utasítás neve	-	46
2.6.2	A DD utasítás pozicionális paramétere	-	47
2.6.3	A DD utasítás kulcsszavas paramétere	-	49
2.6.3.1	Az állomány nevének megadása	DSNAME	49
2.6.3.2	A diszpozíció megadása	DISP	51
2.6.3.3	A perifériális egységtípus vagy egység kijelölése	UNIT	53
2.6.3.4	A kötet (volume) kijelölése	VOLUME	55
2.6.3.5	Az állománycimke-információk megadása	LABEL	56
2.6.3.6	Az adatvezérlő blokk (DCB) információk megadása	DCB	58
	Az állomány szervezettsége	DSORG	
	A blokk mérete	BLKSIZE	
	A blokk keresési határok megadása BDAM esetén	LIMCT	
	A rekord formájának és jellemzőinek megadása	RECFM	
	A rekordhossz megadása	LRECL	
	A rekordkulcs hosszának megadása	KEYLEN	
	A rekordkulcs pozíciójának megadása	RKP	
	A pufferezés típusa	BFTEK	
	A pufferek szóhatárra illesztése	BFALN	
	A pufferek számának megadása	BUFNO	
	A pufferek hosszának megadása	BUFL	
	A kártyaolvasási, illetve -lyukasztási mód előírása	MODE	
	A kártyalerakó kiválasztása	STACK	
	A kód megadása lyukszalag esetén	CODE	
	A mágnesszalag írássűrűségének megadása	DEN	

		Kulcsszó	Oldal
	A cilinderen belüli, elsődleges tulcsordulás-terület kijelölése	CYLOFL	
	A cylinder index terület megadása	NTM	
	A sortávolság rögzítése sornyomtatón	PRTSP	
	Rendelkezés az input/output hibát "előidéző" blokkról	EROPT	
	A vezérlőprogram opcionális szolgáltatásainak specifikálása	OPTCD	
	A READ, a WRITE és a CHECK makroutasítások használata	NCP	
	Példák a DCB paraméter alkalmazására	-	
2.6.3.7	Terület kijelölése közvetlen hozzáférésű táron	SPACE	76
2.6.3.8	Terület kijelölése közvetlen hozzáférésű táron, osztott cilinderekkel	SPLIT	79
2.6.3.9	Terület kijelölése közvetlen hozzáférésű táron egy korábban allokált területből	SUBALLOC	81
2.6.3.10	Az állomány elkülönítése meglevő állományoktól	SEP	83
2.6.3.11	Hivatkozás korábbi SEP paraméterre	AFF	84
2.6.3.12	Hivatkozás későbbi DD utasításra	DDNAME	84
2.6.3.13	Az output osztály megadása	SYSOUT	85
2.6.3.14	A logikai rekordok számának korlátozása output állomány esetén	OUTLIM	87
2.7	A JCL parancs	-	87
2.8	A JCL megjegyzés	-	88

	Kulcsszó	Oldal
2.9 A határoló utasítás	-	88
2.10 A // utasítás	-	89
2.11 A JCL eljárás	-	89
2.11.1 Bevezetés	-	89
2.11.2 Az input-árambeli JCL eljárás- deklaráció lezárása	PEND	90
2.11.3 Alapértelmezés szerinti értékek hozzárendelése a szimbolikus paraméterekhez	PROC	90
2.11.4 Példa JCL eljárás alkalmazására	-	91
2.11.5 DD utasítás megadása JCL eljárás valamely lépéséhez	-	92
1. FÜGGELÉK		
A programok fordítására, szerkesztésére és futtatására vonatkozó katalogizált JCL eljárások	-	93
2. FÜGGELÉK		
A különböző eszköztípusok listája	-	105
3. FÜGGELÉK		
Példák teljes job-ra	-	109
4. FÜGGELÉK		
A felhasználói hibakódok és jelentésük	-	113
IRODALOM		123
SZÓGYÜJTEMÉNY		124





Előszó

E programozói kézikönyv azokhoz a felhasználókhöz szól, akik nagy-számítógépeken OS rendszerben dolgoznak.

Rendszeres felépítésben tartalmazza azokat az OS JCL (Operating System Job Control Language) lehetőségeket, amelyeket a felhasználó munkája során kihasználhat. Bevezetésképpen röviden ismerteti az OS rendszer felépítését és működését, majd az egyes JCL utasítások paraméterezési lehetőségeivel, szerepével, hatásával foglalkozik, mindvégig a felhasználó szempontjából tekintve azokat.

A kötet fő fejezetei az OS job főbb utasításainak megfelelően kapcsolódnak egymáshoz. Ha az ismétlések elkerülése végett visszahivatkozásokra van szükség, akkor a mélyebb szintről a magasabb szintre történik a hivatkozás, mint pl. az EXEC utasítás egyes paramétereinél a JOB utasítás megfelelő paramétereire.

Az egyes fejezeteken belül a tárgyalt JCL utasítás paramétereit - ha a formai korlátok ezt megengedik - fontosságuk és gyakoriságuk sorrendjében tárgyalja. Ennek különösen a viszonylag sok paraméterrel rendelkező DD utasítás esetében van jelentősége.

Az egyes paraméterek tárgyalásakor a következő felépítést követi:

- a paraméter megadásának formája,
- értelmezése, jelentése,
- alapértelmezése,
- alkalmazása,
- példák.

E kézikönyv az OS JCL valamennyi lehetőségét tárgyalja. (Kivétel a JCL parancs.) Konkrét géptípus, konfiguráció, generáláskor szükitett operációs rendszer esetén azonban előfordulhat, hogy bizonyos paraméterek használata értelemszerűen nem jöhet szóba.

Végül köszönetet mondok Kovács Jánosnak, a SZÁMKI osztályvezetőjének, Braun Péternek és Horniák Gábornak, a Villamosipari Kutató Intézet főosztályvezető helyettesének, illetve osztályvezetőjének, valamint Varsányi Andrásnak, a Számítógépalkalmazási Kutató Intézet munkatársának a kézikönyvvel kapcsolatos hasznos megjegyzéseiért.

Budapest, 1977. január hó

Bölcsföldi József



1. Bevezetés

1.1 Az ESZR operációs rendszerek

Az ESZR nagyszámítógépeken jelenleg általában a következő rendszerek használatosak:

DISC OPERATING SYSTEM (DOS),
OPERATING SYSTEM (OS).

E leírásban az OS JCL nyelvet ismertetjük. A tárgyalás kiterjed az OS rendszer MFT és MVT változataira is.

1.2 Az OS felépítése és működése (vázlatosan)

A felhasználó szempontjából az operációs rendszer az alábbi feladatokat látja el.

1.2.1 A JOB-KEZELÉS

A *beolvasó/értelmező* - READER/INTERPRETER - rendszerprogram lyukkártyáról (vagy mágnesszalagról, mágneslemezzel) beolvassa a felhasználói job-ot, ebbe beleépíti azokat a JCL eljárásokat, amelyekre ez a felhasználói job esetleg hivatkozik és az így nyert job-ot formailag ellenőrzi. Ha a job formailag hibátlan, akkor az ebben levő JCL utasítások alapján a job-ra vonatkozó definíciókat, információkat az *input várakozási sorban* (input queue-ben) mágneslemezen helyezi el, a felhasználói job-ban esetleg meglevő egyéb információkat (programokat, a programok működéséhez szükséges adatokat) szintén mágneslemezzel, az erre a célra fenntartott input területre írja ki. A job-bal kapcsolatos üzeneteket az *output várakozási sorban* (output queue-ban) mágneslemezen helyezi el.

Az *ütemező* - INITIATOR/TERMINATOR - rendszerprogram ütemezi be elvégzésre a gép számára azokat a job-okat, amelyek az input várakozási sorban futásra várnak.

A *kiíró* - WRITER - program a job lefutása után kiírja sornyomtatóra vagy mágnesszalagra (vagy kilyukasztja lyukkártyára) az output várakozási sorban levő információkat.

A MASTER SCHEDULER rendszerprogram teremti meg a kapcsolatot az operátori konzol és a fenti job-kezelő programok, illetve a futó felhasználói program és az operátori konzol között. Az operátor a MASTER SCHEDULER közvetítésével operátori parancs révén avatkozhat bele a beolvasó, az ütemező, ill. a kiíró program munkájába, a futó felhasználói program pedig a MASTER SCHEDULER közvetítésével írhat a konzolra.

1.2.2 TASK-KEZELÉS

A *vezérlőprogram* - SUPERVISOR - ütemezi a *központi egység* (Central Processing Unit - CPU) és a *perifériák* használatát az alábbiak szerint. Mivel egy gépben általában csak egy CPU van, és ezt egy adott pillanatban csak egy program használhatja, multiprogramozásu rendszerről lévén szó, az egy adott pillanatban párhuzamosan "futó" felhasználói programok mindegyike a következő állapotok valamelyikében van:

- használja a CPU-t,
- CPU-ra vár,
- input/output művelet befejezésére vár.

A központi egységet a reá várakozó felhasználói programok prioritásuk alapján a SUPERVISOR által meghatározott ütemezés szerint használhatják.

A futó felhasználói program input/output műveletet közvetlenül nem indíthat, a perifériákat csak a vezérlőprogramon keresztül használhatja. A futó felhasználói programból jövő input/output igény alapján a perifériák és a központi tár közötti tényleges adatátvitelt a SUPERVISOR bonyolítja le. Ez a privilégizált vagy supervisor állapot. Az adatátvitel (pl. egy blokk beolvasása) után a privilégizált állapot megszűnik.

1.2.3 INFORMÁCIÓKEZELÉS

Az *adatok* kezelésére vonatkozóan e könyv 2.6 pontjára hivatkozunk.

A *programok* kezelésével kapcsolatban itt az alábbiakat emeljük ki.

Az IEBUPDTE nevű rendszerprogram segítségével a forrásnyelvű programot elhelyezhetjük mágneslemezen levő *forrásnyelvű programkönyvtárban*. A már könyvtárban levő forrásnyelvű program javítása ugyancsak az IEBUPDTE nevű rendszerprogrammal történhet.

A forrásnyelvű programot a programkönyvtárból (esetleg lyukkártyáról) a megfelelő fordítóprogrammal lefordítva, a kapott gépi kódu programot (tárgymodul) vagy azonnal átadjuk az IEWL nevű szerkesztő-rendszerprogramnak (Linkage Editor), ill. a LOADER nevű szerkesztő-betöltő rendszerprogramnak, vagy elhelyezzük *tárgymodul-programkönyvtárban*.

A tárgymodul-könyvtárakban levő gépi kódu, de logikailag általában nem zárt, tehát nem futtatható modulokat az IEWL szerkesztő rendszerprogram segítségével egyetlen, logikailag zárt, tehát futtatható programmá (load modul) szerkesztjük és elhelyezzük a *load modul programkönyvtárban*, ahonnan később a központi tárba töltve indíthatjuk. (Másik lehetőség a tárgymodulok összeszerkesztése a LOADER programmal és a keletkező futtatható program azonnali indítása.)



2. Az OS JCL nyelv

2.1 A JCL nyelv rendeltetése, előnye

A JCL nyelv a rendszerprogramok, illetve a felhasználói programok azonosítására, paraméterezésére, e programok működéséhez szükséges állományok (2.6 pont) leírására szolgál. Lehetővé teszi továbbá bizonyos operátori tevékenységek automatizálását kódolás segítségével.

Ha a perifériákra vonatkozó információkat a programból JCL szintre "kiemeljük", ún. perifériafüggetlen programot kaphatunk.

2.2 Utasításkészlet

Az OS JCL utasítások a következők:

*Az utasítás kulcsszó
vagy egyéb azonosító*

*Az utasítás jelentése
röviden*

JOB

A job kezdetének jelzése, nevének megadása, paraméterezése

EXEC

A job-on belüli lépés (step) kezdetének jelzése, a végrehajtandó program vagy JCL eljárás nevének megadása, paraméterezése.

DD

Az állomány leírása.

Parancs

Az operációs rendszernek szóló parancs.

Határoló

Az input árambeli adatok behatárolása.

//

A job végének jelzése.

***Az utasítás kulcsszó
vagy egyéb azonosító***

***Az utasítás jelentése
röviden***

PROC	Egy JCL eljárásdeklaráció kezdetének jelzése, paraméterezése.
PEND	Egy JCL eljárásdeklaráció végének jelzése.
Megjegyzés	Magyarázó szöveg.

2.3 Általános formai szabályok

E leírásban a következő metanyelvi jelölésrendszert fogjuk használni:

<i>Jel</i>	<i>Jelentés</i>
nagybetűs szó	JCL nyelvi kulcsszó
kisbetűs szó	A felhasználó által - az idevágó megszorítások figyelembevételével - választható név vagy egyéb információ.
[]	Az ilyen zárójelben álló JCL elem nem kötelező.
{ }	Az ilyen zárójelben álló elemek közül kötelező egyet és csak egyet megadni.
- - -	Más kifejezések következhetnek.
. . .	Az előző kifejezés többször ismétlődhet.
Megjegyzés	Ha egy paraméternek alparaméterei vannak, akkor ezeket egy kerek zárójelben kell felsorolni. Ha az alparaméterek közül csak az első pozicionális alparamétert vagy csak egy kulcsszavas alparamétert írunk ki, akkor ez a kerek zárójel elmaradhat. Ezt a kivételt metanyelvi szinten nem jelöljük.

Karakterkészlet

- a) az angol ABC 26 betűje (nagybetűk)
- b) a decimális számjegyek
- c) # @ \$
- c) speciális karakterek, mint , . / ' () & + - = ▽

Megjegyzés: ▽ a szóközt jelöli.

2.3.1 A JCL UTASÍTÁSOKRÓL ÁLTALÁBAN

2.3.1.1 A JCL utasítás általános formája

//	név	utasításkód	operandus	megjegyzés
----	-----	-------------	-----------	------------

Az elhatárolt részeket *utasításmezőknek* nevezzük.

Az utasításmezők között legalább egy szóköznek kell állnia, kivéve a // és "név" mezőket, melyek között nem lehet ▽. (Az OS-ben használatos nyelvek utasításait kódlapra írjuk. A kódlap egy sora egy 80 pozíciós lyukkártyának felel meg.) Az egyes JCL utasítások a következő mezőket tartalmazzák:

Az utasítás neve

Az utasításmezők

JOB	// név JOB [operandus] [megjegyzés]
EXEC	// [név] EXEC operandus [megjegyzés]
DD	// [név] DD operandus [megjegyzés]
PROC (katalogizált)	// [név] PROC operandus [megjegyzés]
PROC (input)	// név PROC [operandus] [megjegyzés]
PEND	// [név] PEND [megjegyzés]
JCL parancs	// parancs operandus [megjegyzés]
Határoló	/* [megjegyzés]

Az utasítás neve

//

Megjegyzés

Az utasításmezők

//

// * megjegyzés

Jegyzet: "megjegyzés" csak akkor írható, ha irtunk operandust.

2.3.1.2 Az egyes utasításmezők funkciója és az általános formális szabályok

Az egyes JCL utasítások speciális szabályait 2.4-től kezdve tárgyaljuk.

Az 1. és 2. pozíción // jeleknek kell állni. Ez alól kivétel a / * utasítás.

Név

A név legalább 1 és legfeljebb 8 karakter lehet. Csak alfanumerikus jeleket és a # @ \$ jeleket tartalmazhatja. A névnek a 3. pozíción kell kezdődnie és első karaktere nem lehet számjegy.

A név azonosítja a JCL utasítást és lehetővé teszi, hogy egy másik JCL utasításból vagy egy programból erre a JCL utasításra hivatkozzunk.

Utasításkód

Az utasításkód a JCL utasítás típusát vagy - JCL parancs esetén - a parancsot határozza meg.

Operandus

Az operandusmező egymástól vesszővel elválasztott paramétereket tartalmaz.

Megjegyzés A megjegyzésmező a job-ot kezelő személy tájékoztatását megkönnyítő információkat tartalmaz.

A JCL utasítást kötelező az 1. pozíción kezdeni és a kódolást a sor 71. pozícióján be kell fejezni.

2.3.2 JCL UTASÍTÁS FOLYTATÁSA ÚJ SORBAN

A JCL parancsnak, JCL megjegyzésnek, a /* és // utasításoknak nem lehet folytatás sora. Csak az operandust vagy a megjegyzést lehet új sorban folytatni.

Az operandusmező folytatásának szabályai

- I. Az operandus kódolását egy paramétert vagy alparamétert követő vessző után lehet megszakítani.
- II. A megszakítás után - legalább egy szóközt követően - még ugyanebben a sorban állhat megjegyzés.
- III. A megszakított sor 72. pozíciójára lehet szóköztől különböző megjelölő karaktert írni.
- IV. A folytatás sor 1. és 2. pozícióján // jeleknek kell állni.
- V. Az operandust a folytatás sorban legkorábban a 4., legkésőbb a 16. pozíción lehet folytatni.

A megjegyzésmező folytatásának szabályai

- I. A megjegyzés szöveg bárhol megszakítható.
- II. A 72. pozícióra egy szóköztől különböző karaktert kell írni.
- III. A folytatás sor 1. és 2. pozícióján // jeleknek kell állni.
- IV. A megjegyzés szöveget a folytatás sorban legkorábban a 4. pozíción lehet folytatni.

2.4 A job-utasítás

Forma:

//név JOB [operandus] [megjegyzés]

A job-utasítás *alapvető hatása* kettős:

- a) a job kezdetét jelöli,
- b) azonosítja a job-ot a JOB utasítás nevének segítségével.

A job-utasítás további hatásait paramétereinek tárgyalásánál ismertetjük.

A következő szavak és karakterek - mivel a DISPLAY parancs kulcsszavai - job-névként nem használhatók:

CONSOLES	A	U
DSNAME	N	
JOBNAMEs	Q	
SPACE	R	
STATUS	T	

Példák job-névre

//JOB1	JOB
// 25	JOB

Az operációs rendszer az első job-ot a JOB1 névvel, a másodikat a 25 névvel azonosítja.

Megjegyzés Az egyes számítóközpontokon belül a job nevével kapcsolatban különböző megszorításokat szokás alkalmazni az egyértelműség, a könnyebb kezelhetőség stb. érdekében.

2.4.1 A JOB-UTASÍTÁS POZICIONÁLIS PARAMÉTEREI

A pozicionális paraméterek a job-nak programmal történő automatikus elszámolását, a téma és a programozó nevének megadását teszi lehetővé.

Használatuk nem kötelező. Sorrendjük kötött.

A job-utasításnak két pozicionális paramétere van. Ezek a következők:

2.4.1.1 Elszámolási információ (MFT, MVT változatban használható)

A job-nak programmal történő automatikus elszámolását teszi lehetővé.

Forma:

[munkaszám] [, járulékos információ, ...]

Megszorítások

- a) Az alparamétereket egymástól vesszővel kell elválasztani.
- b) Ha az elszámolási információnak legalább két alparamétere van, akkor az egész elszámolási információt zárójelek vagy aposztrófok közé kell helyezni.

Például:

' 330921, TEMAXQ' vagy
(330921, TEMAXQ) de
330921

- c) Ha az elszámolási információt új sorban kívánjuk folytatni, akkor az elszámolási információnak zárójelek között kell állnia. A zárójelek helyett aposztrófok ebben az esetben nem használhatók.
- d) Az elszámolási információ - az alparamétereket elválasztó vesszőket is beleértve - 142 karakternél hosszabb nem lehet.
- e) Ha az elszámolási információ valamely alparamétere speciális karaktert tartalmaz (kivéve, ha ez a jel -), akkor vagy az egész elszámolási információt aposztrófok közé, vagy a szóbanforgó alparamétert aposztrófok közé és az elszámolási információt zárójelek közé kell tenni.

Például:

' 540620, 74/05/10' vagy
(540620, ' 74/05/10')

Ha az aposztrófot vagy az & jelet kívánjuk speciális karakterként használni, akkor az adott helyen két aposztrófot, ill. & jelet kell kódolni.

Például:

(110810, 'SOPRON & & NAGYCENK')

- f) Az elszámolási információ alparaméterei pozicionálisak.

Alkalmazása

Elszámolási információ kódolása akkor szükséges, ha az operációs rendszer részeként működik egy olyan program, amelyik a különböző job-ok automatikus költségelszámolását végzi. Ha ilyen program nincs, akkor az "elszámolási információ" nem szükséges.

Példák elszámolási információra

- a) //J1 JOB 338910
- b) //J2 JOB (338910, '74/05/10', TEMA1)
- c) //J3 JOB 338-910
- d) //J4 JOB (, '74/05/10', TEMA2)

A J4 nevű job esetében munkaszámot nem adtunk meg, csak járulékos információkat.

2.4.1.2 A programozó neve (MFT, MVT)

Forma:

[, a programozó neve]

Megszorítások

- a) Ha a programozó nevét kódoljuk, akkor a névnek az elszámolási információ vessző-terminátora után kell állni.
- b) A név 20 karakternél hosszabb nem lehet.
- c) Ha a név speciális karaktereket tartalmaz a pont kivételével, akkor aposztrófok közé kell tenni. Ha a név speciális karakterként aposztróft tartalmaz, akkor az adott helyre két aposztróft kell írunk.
- d) Ha a programozó nevét nem használjuk, akkor vessző-terminátorát sem kell megadni.

Példák

- a) // # I JOB , J. J. SZOMBATI
- b) // MT25 JOB (322600, BUDAPEST), SZEREDAI

2.4.2 A JOB-UTASÍTÁS KULCSSZAVAS PARAMÉTEREI

A kulcsszavas paraméterek

a job	osztályba sorolását, futtatásának megszakítását, prioritásának megadását, automatikus újraindítását, helyigényének megadását, futásidejének korlátozását,
a rendszer üzeneteinek	osztályba sorolását, korlátozását teszik lehetővé.

Használatuk nem kötelező. Tetszőleges sorrendben írhatók.

Alapértelmezés

Ha nem adunk értéket a job-utasítás kulcsszavas paramétereinek, akkor általában a rendszer beolvasóprogramjának megfelelő paramétere van érvényben.

A lehetőségek a következők:

2.4.2.1 A job futásának megszakítása (MFT, MVT)

Forma:

[COND=((kód, operátor) , ...)]

Értelmezése

Egy job-ban egymás után általában több programot is futtatunk, más szóval *egy job általában több lépésből áll*. A rendszer a lépés végrehajtása után egy visszatérési kódot - return code - szolgáltat, mely a [0;4095] zárt intervallumba eső egész szám.

A "kód" értékeként a fenti intervallumba eső egész értéket írhatunk elő.

Az "operátor" értékei a következők lehetnek:

GT	nagyobb, mint	(greater than)
GE	nagyobb vagy egyenlő	(greater or equal to)

EQ	egyenlő	(equal)
LT	kisebb, mint	(less than)
LE	kisebb vagy egyenlő	(less or equal to)
NE	nem egyenlő	(not equal)

A rendszer a COND paraméterben megadott "kód"-dal és az aktuális lépés végrehajtásából eredő visszatérési kóddal mint operandusokkal végzi el az "operátor"-ral megadott logikai műveletet. Ha ez a reláció igaz, akkor a job futása befejeződik; ellenkező esetben a rendszer megkezdi a job következő lépésének végrehajtását. COND paramétert EXEC utasításban is meg lehet adni. Ha két helyen megadjuk, akkor a job utasításbeli érték van érvényben. Tehát, ha a job utasításban megadott COND feltétel alapján a job futását meg kell szakítani, akkor a rendszer az EXEC utasításban megadott COND feltételt már nem vizsgálja.

Alapértelmezése

Kritikus értéket meghaladó visszatérési kód esetén a job futása megszakad. Ez a kritikus érték az operációs rendszer különböző programjainál általában különböző.

Példák a COND paraméter alkalmazására:

a) //DOMBAI JOB COND=(3, LT)

Ha 3 kisebb, mint az aktuális lépés visszatérési kódja, akkor a job futása befejeződik, azaz a további lépéseket a rendszer nem hajtja végre.

b) //SQ1 JOB COND=((30, LT), (20, GT))

Ha 30 kisebb, mint az aktuális lépés visszatérési kódja vagy 20 nagyobb ennél a visszatérési kódnál, akkor a job futása megszakad.

2.4.2.2 A végrehajtási lista szabályozása (MFT, MVT)

Forma:

[MSGLEVEL = (utasítások, üzenetek)]

Az "utasítások" és az "üzenetek" pozicionális alparaméterek. Ha az ezeket elválasztó vesszőt kódoljuk, akkor a zárójel is kell, különben elhagyható.

Az "utasítások" értékeként a következő számjegyek valamelyike írható elő: 0, 1, 2

Jelentésük:

- 0 a végrehajtási listán (output) kizárólag a JOB utasítás jelenik meg a JCL utasítások közül.
- 1 az összes input JCL utasítás, a job-ban használt katalogizált JCL eljárások utasításait (kirejtését) és a PROC utasítás paramétereit - a szimbolikus paraméterekbe történő behelyettesítés után - kilistázza. (A JCL eljárások leírása a 2.11 pontban található.)
- 2 a rendszer csak az elsődleges input JCL utasításokat listázza.

Az "üzenetek" alparaméter értéke a következő számjegyek valamelyike lehet: 0, 1

Jelentésük:

- 0 a rendszer csak a job abnormális befejezése esetén írja ki az allokációs és diszpozíciós közleményeket, egyébként csak egy szűkszavu közleményt fog kilistázni.
- 1 a rendszer a job befejezése esetén az összes allokációs és diszpozíciós közleményt kilistázza.

Alapértelmezése

az RDR eljárásban lévő érték.

Példák az MSGLEVEL paraméter alkalmazására

- a) //M1 JOB MSGLEVEL=(2,1)
- b) //N1 JOB MSGLEVEL=(,0)
- c) //K1 JOB MSGLEVEL=2

Az M1 nevű job output listájára csak az inputból jövő JCL utasítások kerülnek, viszont a job abnormális befejeződése esetén részletes hibajelentést kapunk.

Az N1 nevű job "utasítások" alparaméterére vonatkozóan az operációs rendszer generálásakor megadott érték van érvényben. Ugyanez áll a K1 nevű job "üzenetek" alparaméterére.

2.4.2.3 Késleltetett újraindítás megadott kópia azonosítótól abnormális befejeződés esetén (MFT, MVT)

Forma:

$$\left[\text{RESTART} = \left(\begin{array}{l} * \\ \text{stepnév} \\ \text{stepnév. eljárásstepnév} \end{array} \right) \left[\begin{array}{l} \text{,kópia} \\ \text{azonosító} \end{array} \right] \right]$$

Az alparaméterek jelentése

- * a végrehajtás az első lépés (step) elejétől, illetőleg az első lépés itt megadott kópia azonosítójától ismétlődik.

stepnév

a végrehajtás a "stepnév" nevű lépés elejétől, illetőleg ezen lépés itt megadott kópia azonosítójától újra kezdődik.

stepnév. eljárásstepnév

a végrehajtás a "stepnév" nevű lépésből hívott JCL eljárás "eljárásstepnév" nevű lépésének elejétől, illetőleg ezen lépés itt megadott kópia azonosítójától ismétlődik.

kópia azonosító

a programnak az a pontja, ahonnan a feldolgozást ismételni kell. Ha a kópia azonosító kódolva van, akkor a végrehajtás a nevezett lépés megadott kópia azonosítójától ismétlődik. Ha a kópia azonosítót nem adunk meg, akkor a feldolgozás a megnevezett lépés elejétől ismétlődik.

Ha kópia azonosítót megadunk, akkor a job utasítás után egy SYSCHK nevű DD utasítást is definiálni kell.

Az operációs rendszer minden programbeli kópia azonosítónál kimentti a programot a SYSCHK nevű DD utasításban leírt állományba. Ha később újraindítás válik szükségessé, akkor a kópia azonosítónak megfelelő utolsó programállapotot a SYSCHK állomány alapján visszaállítja és a program végrehajtása innen ismétlődik.

Alapértelmezés:

nincs újraindítás.

Példák késleltetett újraindításra

a) //Z1 JOB RESTART=(*, INP)

Ha a futás - INP után - megszakad, akkor az első lépés INP nevű kópia azonosítójától fog ismételni.

b) //Z2 JOB RESTART=(MEIS.SN, ER)

Abnormális befejeződés esetén a végrehajtás a MEIS nevű lépésben meghívott JCL eljárás SN nevű lépésében levő ER nevű kópia azonosítójától ismétlődik.

2.4.2.4 Automatikus újraindítás engedélyezése, illetve eltiltása (MFT, MVT)

Forma:

$$\left[RD = \begin{Bmatrix} R \\ RNC \\ NC \\ NR \end{Bmatrix} \right]$$

Jelentésük

- R - az automatikus lépés-újraindítás megengedett
- RNC - az automatikus lépés-újraindítás megengedett
- kópia azonosítótól történő újraindítás nincs megengedve (letiltja a CHKPT makró végrehajtását)
- NC - sem automatikus lépés-újraindítás, sem kópia azonosítótól történő automatikus újraindítás nincs megengedve (letiltja a CHKPT makró végrehajtását)
- NR - sem automatikus lépés-újraindítás, sem kópia azonosítótól történő automatikus újraindítás nincs megengedve, de a CHKPT makró kópia azonosítót beiktathat.

Ha MFT vagy MVT rendszerben RD=R vagy RD=RNC értéket specifikálunk, akkor a MSGLEVEL első alparaméterét célszerű 1-nek választani, azaz az összes JCL utasítást kilistáztatni a visszahívások nyomkövetése céljából. Ugyanezekben az esetekben minden job-lépést egyedi névvel kell ellátni.

Példák

- a) //TOVA JOB RD=R, MSGLEVEL=(1, 0)

Abnormális befejeződés esetén az automatikus újraindítást engedélyezi.

- b) //VETO JOB RD=NR, MSGLEVEL=(1, 1)

Automatikus újraindítást nem engedélyezünk, de a CHKPT makró kópia azonosítót beiktathat.

Alkalmazás

Az automatikus újraindítás lehetőségével csak bizonyos esetekben lehet és alapos megfontolás után célszerű élni.

RESTART előírása gépidőigényes programok esetében célszerű, így bizonyos lépések felesleges és időpazarló újrafuttatását elkerülhetjük.

Az újraindítást az operációs rendszer csak bizonyos feltételeknek eleget tevő visszatérési kód esetén és az operátor "YES" üzenete után hajlandó elvégezni.

Alapértelmezés:

nincs újraindítás.

2.4.2.5 A job osztályba sorolása (MFT, MVT)

Forma:

[CLASS= job-osztály]

A "job-osztály" egyetlen karakter: az A, B, C, D, E, F, G, H, I, J, K, L, M, N, O betűk valamelyike.

A CLASS paraméter megadásával a job-ot a fenti 15 osztály valamelyikébe soroljuk.

Értelmezése és alkalmazása

Például MFT rendszerben a központi egység minden partíciójához egy vagy több osztály van rendelve. A központi egység részekre osztását és az osztályok hozzárendelését a rendszer generálásakor végezzük el. A felosztás és az osztályok hozzárendelése később konzolról módosítható.

Legyen pl. ez a hozzárendelés a következő:

<u>A partíció neve</u>	<u>A partíció osztálya</u>	<u>A partíció mérete</u>
P0	AB	100 K
P1	A	80 K
P2	A	20 K

Minden job csak vele azonos osztályu partícióban futhat. Ha pl. egy 92K helyigényű job-ot akarunk ebben a rendszerben futtatni, akkor ezt a job-ot B osztályunak kell definiálnunk. A megfelelő job-utasítás a következő lehet:

```
!/Q1 JOB 330800, CSENGERI, CLASS=B
```

Alapértelmezés: A.

Ha nem létező osztályt adunk meg, akkor a job nem fut, "eltűnik".

2.4.2.6 A rendszer üzeneteinek osztályba sorolása (MFT, MVT)

Forma:

```
[ MSGCLASS=output-osztály ]
```

Az "output-osztály" egyetlen karakter: az angol ABC valamelyik betűje vagy egy számjegy.

Értelmezése és alkalmazása

A rendszer az alapértelmezés szerinti outputot (SYSOUT, 2.6.3.13 pont) egy erre a célra a rendszer által fenntartott mágneslemez-területen - output várakozási sor, *output queue* - gyűjti.

A különböző osztályu outputokat különválasztva iratjuk ki. Így az MSGCLASS paraméter alkalmazásával elérhetjük az üzenetek elkülönülését az egyéb outputoktól.

Alapértelmezése: A.

Példa az MSGCLASS-paraméter alkalmazására

```
//T25A1F12 JOB MSGCLASS=E
```

Mivel a végrehajtandó program SYSOUT outputjára nézve is az A osztály a szokásos alapértelmezés, a rendszer üzeneteinek E osztályba

való sorolásával elkülönítjük ezeket az üzeneteket a futó program SYSOUT outputjától.

2.4.2.7 A job prioritásának megadása (MFT, MVT)

Forma:

[PRTY=prioritás]

ahol a prioritás egész értékre:

$0 \leq \text{prioritás} \leq 13$

A PRTY paraméter megadásával az azonos osztályu job-ok közötti prioritást határozzuk meg.

Ha a központi egységre több, azonos osztályu job vár a rendszer által erre a célra fenntartott mágneslemez-területen - input várakozási sor, *input queue* -, akkor elsőnek az a job fut, amelynek nagyobb a prioritása.

Példa a PRTY paraméterre:

//A JOB PRTY=10, CLASS=F

2.4.2.8 A job futási idejének korlátozása (MFT, MVT)

Forma:

$$\left[\text{TIME} = \left\{ \begin{array}{c} (x, y) \\ 1440 \end{array} \right\} \right]$$

ahol az x és y egész értékekre:

$0 \leq x < 1440$

$0 \leq y < 60$

Jelentésük

x a percek száma

y a másodpercek száma

A job legfeljebb $60x+y$ másodperc tiszta belső időt (CPU) használhat el. Ennek eltelte után a job automatikusan törlődik.

TIME=1440 azt jelenti, hogy a futási idő nincs korlátozva.
x és y pozicionális alparaméterek.

A TIME paramétert a végrehajtást kezdeményező (EXEC) utasításban is előírhatjuk és ezáltal az adott lépés futási idejét korlátozhatjuk.

Példák

- a) //F1 JOB TIME=(28, 30)
Ez a job legfeljebb 28 perc 30 másodperc CPU időt használhat fel.
- b) //ROVID JOB TIME=(, 30)
A rendszer a job-ot 30 másodperc CPU idő eltelte után törli.
- c) //F3 JOB TIME=1440
Az F3 job futási ideje elvileg korlátlan.

2.4.2.9 A job visszatartása a job-input várakozási sorban (input-queue-ban) (MFT, MVT)

Forma:

[TYPRUN=HOLD]

Jelentése: a rendszer az adott job-ot mindaddig az input várakozási sorban (lásd 2.4.2.7-ben) tartja, amíg az operátor ezt meg nem változtatja.

Alkalmazása

TYPRUN=HOLD előírása akkor szükséges, ha a szóban forgó job csak bizonyos előfeltételek teljesülése esetén indulhat.

Például

//T1 JOB TYPRUN=HOLD

A T1 nevű job csak akkor léphet tovább az input várakozási sorból, ha az operátor kiadta a RELEASE T1 parancsot.

2.4.2.10 A job helyigényének megadása (MVT)

Forma:

[REGION=értékK]

ahol az érték egész számra:
érték ≤ 16382

Értelmezése

A job minden lépésének "érték" * 1024 byte összefüggő területet jelöl ki a központi tárban.

REGION paramétert a végrehajtást kezdeményező (EXEC) utasításban is megadhatunk.

A JOB utasításban megadott REGION paraméter felülírja az EXEC utasításban előírt értéket.

Alapértelmezése

az RDR nevű JCL eljárásban levő érték.

Alkalmazása

MVT rendszerben, mivel nincsenek partíciók, az operációs rendszer a REGION érték alapján jelöl ki elsődleges területet a lépés számára.

Például

```
//B1 JOB REGION=130K
```

A rendszer a job minden lépésének 130K összefüggő területet jelöl ki a központi tárból.

2.4.2.11 A job által felhasznált terület átengedése más job-oknak és megfordítva (MVT)

Forma:

[ROLL= (x, y)]

ahol x és y értékeként a YES és NO értékek valamelyike írható elő.

Jelentésük

x a job-lépés által felhasznált terület átengedhető-e más job számára?

y a job lépései igényt tartanak-e más job által felhasznált területre?

A ROLL paraméter mindkét alparaméterét ki kell írni.

A ROLL paraméter EXEC utasításban is megadható.

Ha mindkét helyen előírjuk, akkor az EXEC utasításbeli érték figyelmen kívül marad.

Alapértelmezése

az RDR nevű JCL eljárásban lévő érték.

Például

```
//DINAM JOB REGION=120K,ROLL=(YES,YES)
```

A 120K terület más job-nak átengedhető, illetve a más job által használt terület átvehető.

2.4.3 PÉLDÁK A JOB-UTASÍTÁS PARAMÉTEREZÉSÉRE

a) //E25 JOB 335600,TORDAI,CLASS=B,COND=(7,LT),
// MSGLEVEL=(2,0),MSGCLASS=C,TIME=(5,20)

Az operációs rendszer ezt a job-ot az E25 névvel azonosítja. A programmal történő automatikus költségelszámolás a 335600 munkaszám alapján lehetséges. A programozó neve Tordai. Ez a job csak B osztályu partícióban futhat (MFT). Ha 7 kisebb, mint az aktuális lépés visszatérési kódja, akkor a job futása befejeződik, azaz a további lépéseket nem hajtja végre. Csak az input JCL utasítások listázását kérjük, abnormális befejeződés esetén szűkszavu hibajelentéssel. A rendszer üzeneteit C osztályba sorolva szeparáljuk ezeket a program SYSOUT outputjától. A job 5 perc és 20 másodperc CPU idő eltelte után törlődik.

b) //E26 JOB

Valamennyi paraméter alapértelmezés szerinti értékét használjuk. (Kivéve az elszámolási információt és a programozó nevét, melyeknek nincs alapértelmezése.)

2.5 Az EXEC utasítás

Forma:

```
// [ név ] EXEC operandus [ megjegyzés ]
```


Értelmezése

Az EXEC utasítás egy lépés kezdetét jelöli.

A "név" azonosítja a lépést a job-on belül. A "név" használata általában nem kötelező. Ha erre a lépésre később hivatkozni akarunk, akkor értelemszerűen meg kell adni a nevet.

Az EXEC utasítás

a végrehajtandó program vagy JCL eljárás

megnevezését, paraméterezését,

a lépés (step)

végrehajtásának feltételhez kötését,

prioritásának megadását,

ujraindításának engedélyezését, ill. eltiltását,

helyigényének megadását,

futási idejének korlátozását,

a központi tárbeli hely

átengedését,

az elszámolási információ

megadását

teszi lehetővé.

2.5.1 AZ EXEC UTASÍTÁS POZICIONÁLIS PARAMÉTERE (MFT, MVT)

Forma:

$$\left\{ \begin{array}{l} \text{PGM} = \left\{ \begin{array}{l} \text{programnév} \\ \text{*.stepnév.ddnév} \\ \text{*.stepnév.eljárásstepnév.ddnév} \end{array} \right\} \\ \text{PROC} = \left\{ \begin{array}{l} \text{eljárásnév} \\ \text{eljárásnév} \end{array} \right\} \end{array} \right\}$$

Jelentésük

programnév

A végrehajtandó program neve.

A végrehajtandó programnak egy programkönyvtár egyik elemének kell lennie.

stepnév.ddnév

A végrehajtandó program a "stepnév" nevű lépésben levő "ddnév" nevű DD utasítással leírt programkönyvtár-elem.

stepnév, eljárásstepnév, ddnév

A végrehajtandó program a "stepnév" nevű lépésben hívott JCL eljárás "eljárásstepnév" nevű lépésében levő "ddnév" nevű DD utasítással leírt programkönyvtár-elem.

eljárásnév

A hívott JCL eljárás neve.

Az EXEC utasítás pozicionális paraméterének meg kell előznie a kulcsszavas paramétereket.

Alapértelmezése nincs.

Példák:

a) //STEP1 EXEC PGM=PROG1

Ezzel az utasítással a PROG1 nevű könyvtár-elemet mint programot kívánjuk végrehajtani.

b) //ST1 EXEC PGM=IEWL
//SYSLMOD DD DSN=LIBR1(PR1), DISP=OLD
//ST2 EXEC PGM=*.ST1.SYSLMOD

Az ST1 nevű EXEC utasítással az IEWL nevű programot (Linkage Editor), az ST2 nevű utasítással pedig az ST1 nevű lépés SYSLMOD nevű DD utasításban leírt programkönyvtár-elemet (a LIBR1 nevű könyvtár PR1 nevű elemét) mint programot indítjuk.

c) //ST EXEC PGM=IEFBR14

Ezzel az utasítással az IEFBR14 nevű speciális rendszerprogramot indítjuk.

d) // EXEC PROC=ASMFCLG

Ezzel az EXEC utasítással az ASMFCLG nevű JOB CONTROL eljárást hívjuk meg.

e) // EXEC ASMFCLG

Az előbbivel azonos hatása.

Az ASMFCLG nevű katalogizált JCL eljárás sorrendben a következő programokat aktivizálja:

1. az ASSEMBLER fordítóprogramot
2. a Linkage Editort (szerkesztőprogramot)
3. a szerkesztett programot.

(Az ASMFCLG tartalma az 1. számú függelékben található.)

2.5.2 AZ EXEC UTASÍTÁS KULCSSZAVAS PARAMÉTEREI

2.5.2.1 Az elszámolási információ megadása (MFT, MVT)

Forma:

[ACCT [.eljárásstepnév] = (elszámolási információ,....)]

Az elszámolási információ lehetővé teszi, hogy a lépés automatikus költségelszámolását programmal végezzük el (lásd még 2.4.1.1 pont).

Megszorítások:

- a) Az alparamétereket egymástól vesszővel kell elválasztani.
- b) Ha csak egy alparamétert írunk, akkor a zárójel elhagyható.
- c) Az elszámolási információ maximális hossza - az alparamétereket elválasztó vesszőket is beleértve - 142.
- d) Ha valamely alparaméter speciális karaktert tartalmaz, akkor aposztrófok közé kell tenni. Ez alól kivétel a kötőjel ('-'). Az aposztrófok nem részei az ACCT információnak. Ha egy alparaméter aposztróft tartalmaz, akkor az adott helyen két aposztróft kell írni. Ugyanez vonatkozik a & jelre is.

Alapértelmezése nincs.

Példák

- a) //S11K EXEC PGM=BY, ACCT=('74/05/*')
- b) //S2 EXEC DKY3, ACCT.S1=363363,
// ACCT.S2=985360

A DKY3 nevű JCL eljárást hívjuk meg, ennek S1 nevű lépéséhez a 363363, az S2 nevű lépéséhez a 985360 elszámolási információt rendeljük.

2.5.2.2 A lépés végrehajtásának feltételhez kötése (MFT, MVT)

Példák

a) //ST5 EXEC PGM=P1, COND=(8, GE, ST2)

Ha 8 nagyobb vagy egyenlő, mint az ST2 nevű lépés végrehajtásából adódó visszatérési kód, akkor az ST5 nevű lépés nem hajtódik végre.

b) // # 58 EXEC PGM=F4, COND=((3, LT, S4), (7, GT, S3. S31))

Ha 3 kisebb, mint az S4 nevű lépés visszatérési kód értéke vagy 7 nagyobb, mint az S3 nevű lépésből hívott JCL eljárás S31 nevű lépésének visszatérési kódja, akkor az # 58 nevű lépést nem hajtja végre.

c) //KP EXEC ASMFC, COND=EVEN

A KP nevű lépést feltétlenül végrehajtja. Azaz az ASMFC nevű JCL eljárást az előző lépések normális vagy abnormális befejeződésétől függetlenül meghívja.

d) //A1 EXEC PGM=HIBA, COND=ONLY

Ha a job előző lépései közül legalább egy abnormálisan fejeződött be, akkor az A1 nevű lépést végrehajtja, különben az A1 utáni lépés ütemezésébe kezd.

2.5.2.3 A végrehajtandó program paraméterezése JCL szinten (MFT, MVT)

Forma:

[PARM [. eljárásstepnév] =érték]

Jelentésük

eljárásstepnév

Az "eljárásstepnév" a PARM paramétert tartalmazó EXEC utasítással hívott JCL eljárás egyik lépésének a neve.

érték

Az "érték" a végrehajtandó program paraméterének értéke.

A rendszer ezt az "érték"-et átadja a végrehajtandó programnak.

Assembler program esetén az átadás módja: a rendszer az értéket az adott particióban valahol tárolja és e tárolómező címének a címét az 1 regiszterbe helyezi. PL/1 program esetén a főprogram paramétere kapja meg az értéket.

Megjegyzés: az "érték" a felhasználói program szempontjából több paraméter is lehet.

Megszorítások

- a) Az "érték" legfeljebb 100 karakter lehet.
- b) Ha az érték egynél több kifejezést tartalmaz egymástól vesszőkkel elválasztva, akkor az "érték"-nek aposztrófok vagy zárójelek között kell állnia.
- c) Ha az "érték" speciális karaktert tartalmaz, akkor vagy az egész "érték"-et aposztrófok közé, vagy a speciális karaktert tartalmazó kifejezést aposztrófok közé, az "érték"-et pedig zárójelek közé kell tenni.

Ha ez a speciális karakter aposztróf vagy & jel, akkor az adott helyre két aposztróft, illetve két & jelet kell írni.

- d) Ha az "érték" új sorban folytatódik, akkor zárójelek között kell állnia.

Alapértelmezése nincs.

Példák

- a) `//A EXEC PGM=DK1, PARM=50`
Az 50 értéket a DK1 nevű programban felhasználhatjuk.
- b) `//Y1 EXEC ASMFCLG, PARM, ASM= LIST, PARM. GO=20`

Az ASMFCLG nevű JCL eljárás ASM nevű lépésében hívott programnak a LIST paraméterértéket, a GO nevű lépésében hívott programnak a 20 paraméterértéket adja át.

2.5.2.4 A lépés automatikus újraindításának engedélyezése, illetve tiltása (MFT, MVT)

Forma:

$$\left[\text{RD } [\text{. eljárásstepnév}] = \left\{ \begin{array}{c} \text{R} \\ \text{RNC} \\ \text{NC} \\ \text{NR} \end{array} \right\} \right]$$

Jelentésük

eljárásstepnév

Az RD paramétert tartalmazó EXEC utasításból hívott JCL eljárás egyik lépésének a neve.

R, RNC, NC, NR

Jelentésük - egy lépésre vonatkoztatva - azonos a 2.4.2.4 alattiakkal.

Példák

a) //P1 EXEC PGM=MYAF, RD=R

Ha P1 nevű lépés abnormálisan fejeződik be, akkor a végrehajtását újra elkezd.

b) //P2 EXEC TURUL, RD.T1=NR, RD.T2=R

A TURUL nevű JCL eljárás T1 nevű lépésére nézve csak azt engedélyezzük, hogy a CHKPT makró kópia azonosítót beiktassa, T2 lépésére nézve viszont lehetővé tesszük az automatikus újraindítást.

2.5.2.5 A lépés idejének korlátozása (MFT, MVT)

Forma:

$$\left[\text{TIME } [\text{. eljárásstepnév}] = \left\{ \begin{array}{c} (x, y) \\ 1440 \end{array} \right\} \right]$$

Jelentésük

eljárásstepnév

A TIME paramétert tartalmazó EXEC utasításból hívott JCL eljárás egyik lépésének a neve.

x,y,1440

Jelentésük és a megfelelő korlátok
- egy lépésre vonatkoztatva - azo-
nosak a 2.4.2.8 alattiakkal.

Alapértelmezése

az RDR nevű JCL eljárásban lévő érték.

Példák

a) //Z EXEC PGM=M2, TIME=(8, 20)

Az M2 nevű program 8 perc 20 másodperc CPU idő felhasználása után automatikusan törlődik.

b) //Y EXEC ZP1, TIME.Z1=(1, 50), TIME.Z2=6

A ZP1 nevű JCL eljárás Z1 nevű lépésének 1 perc 50 másodperc CPU időt, Z2 nevű lépésének 6 perc CPU időt engedélyezünk.

2.5.2.6 A lépés prioritásának megadása (MVT)

Forma:

$$\left[\text{DPRTY} [\text{. eljárásstepnév}] = (\text{érték}_1, \text{érték}_2) \right]$$

Jelentés

A DPRTY prioritás azt határozza meg, hogy milyen sorrendben kapnak a futó programok központi tárterületet és CPU támogatást.

A prioritást a rendszer az $\text{érték}_1 * 16 + \text{érték}_2$ összefüggés alapján számítja ki.

Alapértelmezése:

jobpr * 16 + 11, ahol "jobpr" a JOB utasítás PRTY paraméterében megadott érték vagy ennek alapértelmezése.

Az alparaméterek jelentése

eljárásstepnév

Az "eljárásstepnév" a DPRTY paramétert tartalmazó EXEC utasításból hívott JCL eljárás egyik lépésének neve.

Megszorítások

érték₁ és érték₂ egészekre nézve:

$$0 \leq \text{érték}_1 \leq 15$$

$$0 \leq \text{érték}_2 \leq 15$$

A DPRTY alparaméterei pozicionálisak.

Az "érték₂" alapértelmezése 11.

Példák

a) //SK1 EXEC PGM=PPP, DPRTY=(12, 9)
A PPP nevű program végrehajtási prioritása 201.

b) //SK2 EXEC QUI, DPRTY.STEP4=7

A hívott QUI nevű JCL eljárás STEP4 nevű lépésének végrehajtási prioritása 123.

2.5.2.7 A lépés helyigényének megadása (MVT)

Forma:

[REGION [. eljárásstepnév] =értékK]

Jelentés

eljárásstepnév

Az "eljárásstepnév" a REGION paramétert tartalmazó EXEC utasításból hívott JCL eljárás egyik lépésének neve.

A rendszer a lépés számára "érték" * 1024 byte összefüggő területet jelöl ki a központi tárban.

Megszorítások

az "érték" egészre nézve:

$$0 \leq \text{érték} \leq 16383$$

Példák

a) //V1 EXEC PGM=ALI, REGION=125K

Az ALI nevű program számára 125 Kbyte összefüggő területet jelöl ki a központi tárban.

b) //V2 EXEC INP, REGION.INP1=20K

Az INP nevű JCL eljárás INP1 nevű lépésének 20K összefüggő területet kérünk.

2.5.2.8 A lépés által felhasznált terület átengedése más job-oknak és megfordítva (MVT)

Forma:

[ROLL [. eljárásstepnév] = (x, y)]

Jelentése és a megfelelő korlátok - egy lépésre vonatkoztatva - azonosak a 2.4.2.11 alattiakkal.

Példák

a) //SB EXEC PGM=SA, REGION=80K, ROLL=(YES, YES)

A kért 80K területet az SA nevű program más job-oknak átengedi, illetve a szükséges részt más job-októl átveszi.

b) //SC EXEC MODUL, ROLL.S6=(NO, NO),
// REGION.S6=100K

A hívott MODUL nevű JCL eljárás S6 nevű lépésének 100K összefüggő területet jelöl ki és a felhasznált területet más job-oknak nem engedi át, illetve területet más job-októl nem kap.

2.6 A DD utasítás

Forma:

//[név] DD operandus [megjegyzés]

A DD utasítással a job által használt állományok írhatók le.

Az OS terminológia élesen megkülönbözteti a file és az állomány fogalmát. File-on szimbolikus perifériális információhalmazt, állományon valóságos perifériális információhalmazt értünk. Mivel a programbeli megnyitás hatására a kettő "eggyé olvad", jelen leírásban a továbbiakban az *állomány* megjelölést használjuk.

A lehetséges állományok a következők:

1) Szekvenciális állomány

A szekvenciális állomány blokkok sorozata. Szervezettsége ebben lényegileg ki is merül. A blokkokat GAP-ek (rések, üres helyek) választják el egymástól. Ha a mágnesszalagos szekvenciális állomány valamely blokkját felülírjuk, akkor a szóbanforgó blokk utáni információ elveszettnek tekintendő.

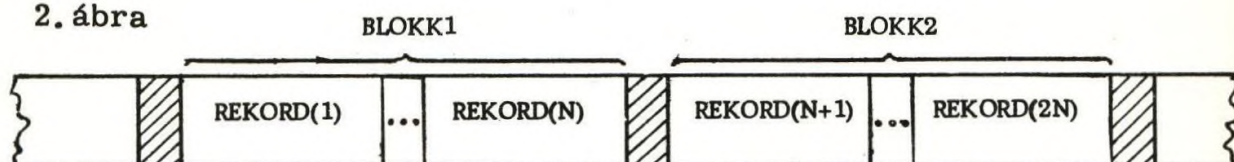
A blokkok csak a fizikai elhelyezkedésük sorrendjében olvashatók (mágnesszalag esetén mindkét irányban).

A blokk állhat egy vagy több fix hosszúságú rekordból (1. és 2. ábra), egy vagy több változó hosszúságú rekordból (3. ábra), egyetlen definiálatlan hosszúságú rekordból (4. ábra), egy vagy több rekordszegmensből (5. ábra). A szegmens a rekordnak egy része.

1. ábra



2. ábra



$N \geq 1$. Ha $N=1$, akkor F (fixed length) rekordokról, különben FB (blocked fixed length) rekordokról beszélünk.

3. ábra



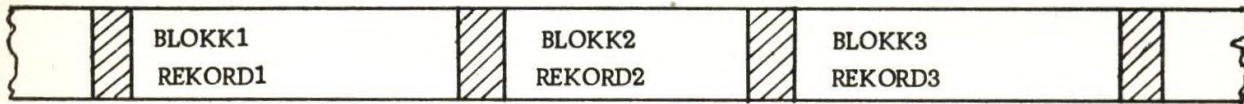
ahol

BH1 a teljes blokk hossza GAP-tól GAP-ig,

RH1 a blokk első rekordjának hossza + RH1 hossza.

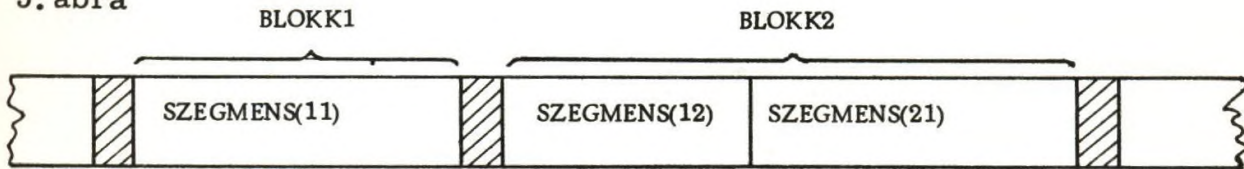
Ha változó hosszúságú rekordok esetén egy blokkban csak egy rekordot tárolunk, akkor V (variable length) rekordokról, különben VB (blocked variable length) rekordokról beszélünk.

4. ábra



Az ilyen rekordokat U (undefined length) rekordoknak nevezzük.

5. ábra

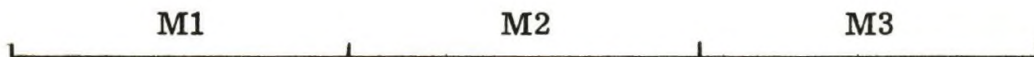


2) Particionált állomány

A particionált állomány olyan szervezettséget jelent, amely lehetővé teszi, hogy az egész információhalmazra egy névvel (állománynév), ezen kívül a halmaz egyes részhalmazaira (tagok, member-ek) két névvel (állománynév és a tag neve) hivatkozzunk.

Például

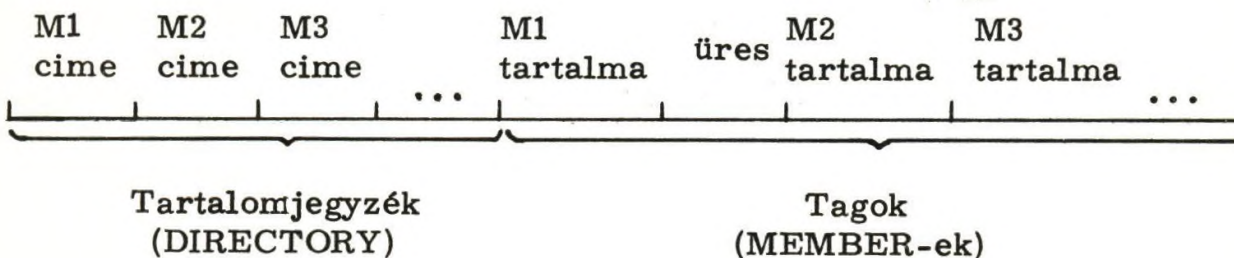
L1



A teljes állományra az L1 névvel, az M2 részre az L1(M2) névvel hivatkozhatunk.

Particionált állomány csak közvetlen hozzáférést megengedő perifériákon (pl. mágneslemez) létesíthető.

Felépítése vázlatosan a következő: (pl.)



A rendszer a tagok nevét és az állományon belüli relatív kezdőcímüket a szóban forgó kötetben levő tartalomjegyzékben (DIRECTORY) tárolja.

A szekvenciális és a particionált állomány a rendszer működése szempontjából alapvető fontosságu. A rendszerkönyvtárak leggyakrabban particionált állományok.

A teljesség kedvéért röviden vázoljuk azokat az állományokat is, amelyeket elsősorban az alkalmazási programokban használunk.

3) *Indexszekvenciális állomány*

Az indexszekvenciális állomány olyan szervezettségű információhalmazt jelent, amely lehetővé teszi, hogy a különböző rekordokhoz az állomány nevének és a rekord egy meghatározott azonosítójának (a rekordkulcsnak) a megadásával tetszőleges sorrendben hozzáférhessünk.

4) *Direkt állomány*

A direkt állomány olyan szervezettségű perifériális információhalmazt jelent, amely lehetővé teszi, hogy a különböző rekordokhoz az állomány nevének és az állományon belüli relatív rekordcímnek vagy a rekordkulcsnak a megadásával tetszőleges sorrendben hozzáférhessünk.

2.6.1 A DD UTASÍTÁS NEVE

Érvényesek a 2.3.2 pont alatti, a névvel kapcsolatos megszorítások. Továbbá, ha egy lépésben több DD utasításnak ugyanazt a nevet adjuk, akkor a rendszer valamennyi DD alapján allokal állományt, de a szóban forgó névvel történő hivatkozás mindig az első ilyen nevű DD utasításra való hivatkozás lesz.

DD utasításhoz két esetben nem írható név:

- a) ha a DD utasításban leírt állományt láncolni (concatenation) akarjuk egy előző DD utasításban definiált állománnyal,
- b) ha az indexszekvenciális állomány leírásához szükséges 3 DD utasítás közül a második és a harmadik DD utasításról van szó.

Speciális DD nevek:

JOBLIB

az ilyen nevű DD utasítással egy privát könyvtárat definiálunk, amely a job bármely lépéséből elérhető.

STEPLIB

az ilyen nevű DD utasítással egy privát könyvtárat definiálunk, amely

STEPLIB

csak abból a lépésből aktivizálható, amely tartalmazza ezt a STEPLIB nevű DD utasítást.

SYSABEND

a rendszer a job ~~abnormális~~ befejezése esetén az ilyen nevű DD utasítással definiált állományba kiírja annak a partíciónak a teljes tartalmát hexadecimális és karakteres formában, amelyben a szóban forgó job futott.

SYSUDUMP

a rendszer a job abnormális befejezése esetén az ilyen nevű DD utasítással definiált állományba kiírja a program által használt központi tárterületek tartalmát hexadecimális és karakteres formában (DUMP).

SYSCHK

az ilyen nevű DD utasítással egy "ujraindítási" állományt definiálunk, melyet a rendszer csak akkor használ, ha a job utasítás alapján a RESTART megengedett. Ez esetben a job futása közben a megadott RESTART pontoknál (2.4.2.3) a rendszer a teljes programot kimentti a SYSCHK nevű DD utasítással definiált állományba, s később innen betöltve újra indíthatja.

2.6.2 A DD UTASÍTÁS POZICIONÁLIS PARAMÉTERE

Forma:

$$\left[\left\{ \begin{array}{l} * \\ \text{DATA} \\ \text{DUMMY} \end{array} \right\} \right]$$

Jelentésük

- * a DD utasítást követő adatokhoz az ezeket felhasználó program az input-áramban jut hozzá. Az input-áram JCL utasítások és adatok sorozata. Ezt az operációs rendszer olvassa be egy input egységről és a JCL utasításokat az input várakozási

sorban (input queue) általában mágneslemezen, az adatokat a rendszer input területen (elérhető DD * utasítással) általában szintén mágneslemezen tárolja.

Megszorítások

MFT, MVT

Egy lépéshez több DD * utasítás írható. A DD * utasítást követő adathalmazt nem kötelező határoló utasítással lezárni. Katalogizált JCL eljárás nem tartalmazhat DD * utasítást.

DD * utasítás és a hozzá tartozó határoló utasítás között vezérlőkártyák nem olvashatók be.

DATA

Tulajdonságai azonosak a * paraméter tulajdonságaival. Kivéve: a DD * utasítást követő állományban JCL utasítások nem fordulhatnak elő, ezzel szemben a DD DATA utasítást követő állományban JCL utasítások is lehetnek.

DUMMY

Az így definiált állomány figyelmen kívül marad. A program úgy fut le, hogy a DUMMY állománnyal kapcsolatos input-output utasítások *fizikailag* hatástalanok.

A DD utasítás pozicionális paramétere természetesen nem kötelező. Ha használjuk, akkor meg kell előznie a DD utasítás kulcsszavas paramétereit.

Példák

- a) //D1 DD *
- b) //D2 DD DATA

A D1 és D2 nevű DD utasítások az adatok helyéül az input-áramot jelölik meg. D2 esetben az adatok között JCL utasítások is lehetnek. D1 esetben nem.

A *pozicionális paraméter* mellett az input-árambeli adatokat határoló karakterpár is előírható.

Forma:

[DLM=' XY']

ahol XY tetszőleges karakterpár.

Szokásos *alapértelmezése*: /*.

Például:

//ddnév DD *, DLM=' QQ'

esetén az input-árambeli állományt a QQ karakterpár zárja le.

2.6.3 A DD UTASÍTÁS KULCSSZAVAS PARAMÉTEREI

2.6.3.1 Az állomány nevének megadása

Forma:

$$\left\{ \begin{array}{l} \text{DSNAME} \\ \text{DSN} \end{array} \right\} = \left\{ \begin{array}{l} \text{név} \\ \text{név (member név)} \\ \text{név (generációs szám)} \\ \text{név (area név)} \\ \& \& \text{név} \\ \& \& \text{név (member név)} \\ \& \& \text{név (area név)} \\ * . \text{ddnév} \\ * . \text{stepnév. ddnév} \\ * . \text{stepnév. eljárásstepnév. ddnév} \end{array} \right\}$$

Jelentés:

név	Az állomány neve. Ez vagy a 2.3.2 alattiaknak megfelelő név vagy ilyen nevekből álló minősített név. Pl.: ABAALKU.Z1.KOMAROM
név (member név)	Az állomány a "név" nevű particionált állomány "member név" nevű tagja.

név (generációs szám)	Az állomány a "név" nevű generációs adatcsoport "generációs szám" azonosítóju állománya. (A "generációs szám"-nak egész értéknek kell lennie.)
név (area név)	Az állomány a "név" nevű indexszekvenciális állomány "area név" nevű területe lesz. (Az INDEX, a PRIME vagy az OVERFLOW terület.)
& & név	Ezzel a névvel egy ideiglenes állományt definiálunk, mely legkésőbb a job lefutása után törlődik.
& & név (member név)	Az állomány a " & & név" nevű ideiglenes particionált állomány "member név" nevű tagja (member-je) lesz.
& & név (area név)	Az állomány a " & & név" nevű ideiglenes indexszekvenciális állomány "area név" nevű területe lesz. (Az INDEX, a PRIME vagy az OVERFLOW terület.)
*. ddnév	Az állomány neve ugyanaz a név lesz, mint ezen job egy korábbi, "ddnév" nevű DD utasításában megadott állomány neve volt.
*. stepnév. ddnév	Az állomány neve ugyanaz a név lesz, mint ezen job "stepnév" nevű korábbi lépésében levő, "ddnév" nevű DD utasításában megadott állománynév volt.
*. stepnév. eljárásstepnév. ddnév	Az állomány neve ugyanaz a név lesz, mint ezen job "stepnév" nevű korábbi lépéséből hívott JCL eljárás "eljárásstepnév" nevű lépésének "ddnév" nevű DD utasításában megadott állománynév volt.

Alapértelmezése nincs.

Példák

- a) //DK DD DSN=MET26
- b) //S1 DD DSN=LIBR1(MODUL1)

Az állomány a "LIBR1" nevű particionált állomány "MODUL1" nevű member-je lesz.

c) //S2 DD DSN= & & MUNKA ___

Az állomány egy " & & MUNKA" nevű ideiglenes állomány lesz.

d) //T1 EXEC PGM=PR5
 //DT1 DD DSN=ALD1 ___
 //T2 EXEC PGM=PRG ___
 //DT2 DD DSN=*.T1.DT1 _---

A DT2 nevű DD utasításban definiált állomány neve ugyanaz lesz, mint a T1 nevű korábbi lépés DT1 nevű DD utasításban definiált állomány neve volt. (Vagyis mindkét állománynév ALD1 lesz.)

2.6.3.2 A diszpozíció megadása (MFT és MVT)

Forma:

$$\left\{ \text{DISP} = \left(\begin{bmatrix} \text{NEW} \\ \text{OLD} \\ \text{SHR} \\ \text{MOD} \end{bmatrix} \begin{bmatrix} , \text{DELETE} \\ , \text{KEEP} \\ , \text{PASS} \\ , \text{CATLG} \\ , \text{UNCATLG} \\ , \end{bmatrix} \begin{bmatrix} , \text{DELETE} \\ , \text{KEEP} \\ , \text{CATLG} \\ , \text{UNCATLG} \end{bmatrix} \right) \right\}$$

Értelmezése

Az *első alparaméter* az állománynak a job-lépés futása előtti állapotát határozza meg.

Az első alparaméter értékei:

NEW	Az állomány a szóban forgó lépés végrehajtása alatt keletkezik.
OLD	Az állomány létezett a szóban forgó lépés futása előtt.
SHR	Az állomány létezett a szóban forgó lépés futása előtt és - <i>csak inputra</i> - egyidőben több job hozzáférhet.
MOD	Ha az állomány létezett a lépés futása előtt, akkor az író-olvasó fejet a feldolgozás kezdetén az állomány

MOD

utolsó rekordja után állítja, ha nem létezett, akkor az állomány a szóban forgó lépés futása alatt keletkezik.

Az első alparaméter alapértelmezése: NEW.

A *második alparaméter* azt határozza meg, hogy a rendszer hogyan kezelje az állományt a *job-lépés normális befejeződése* esetén.

A második alparaméter értékei:

DELETE

Az állományra a továbbiakban nincs szükség. A rendszer a job-lépés lefutása után az állományt a job-tól elengedi, egyben törli, ha az direkt hozzáférést megengedő priférián volt.

KEEP

Az állomány a job lefutása után is megőrződik.

PASS

Az állományt a job-lépés lefutása után átadja a job következő lépésének.

CATLG

Az állomány a job lefutása után is megőrződik és leírása a rendszerkatalógusba kerül.

UNCATLG

Az állomány a job lefutása után is megőrződik, de leírását törli a rendszerkatalógusból.

A második alparaméter alapértelmezése

Ha az állomány a job futása előtt létezett, akkor megőrződik, különben törlődik.

A *harmadik alparaméter* azt határozza meg, hogy a rendszer hogyan kezelje az állományt a *job lépés abnormális befejeződése* esetén.

A harmadik alparaméter értékeinek jelentése azonos a második alparaméternél megadott jelentésekkel.

A DISP paraméternek *nincs alapértelmezése* (csak egyes alparamétereinek van).

Példák

a) //Y DD DSN=AX5, DISP=(NEW, CATLG, DELETE) ____

Az AX5 állomány e job végrehajtása alatt keletkezik. Ha a job-lépés hiba nélkül lefut, akkor az AX5 állomány megőrződik és leírása bekerül a rendszerkatalógusba, különben törlődik.

b) //Z DD DSN=BILD.A.A1, DISP=(NEW, , KEEP) ____

A BILD.A.A1 állomány e job végrehajtása alatt keletkezik. Ha a job-lépés hibátlanul lefut, akkor törlődik, különben megőrződik.

c) //X DD DSN=SYS1.MACLIB, DISP=SHR ____

A SYS1.MACLIB nevű állományt a szóban forgó joblépéssel egyidőben más jobok is használhatják - inputként.

2.6.3.3 A perifériális egységtípus vagy egység kijelölése

Forma:

$$\left[\begin{array}{l} \text{UNIT}=(\text{egységim} \\ \text{egységtípus} \\ \text{csoportnév} \\ \text{UNIT}=\text{AFF}=\text{ddnév} \end{array} \left[\begin{array}{l} , \text{egységek száma} \\ , P \\ , \\ \end{array} \right] \left[\begin{array}{l} [, \text{DEFER}] \\ [, \text{SEP}=(\text{ddnév}, \dots)] \end{array} \right) \right]$$

- Jelentésük:

egységim

Egy konkrét perifériális egységet azonosít annak címével. A cím 3 hexadecimális jelből áll, ezek közül az első jel a csatorna kódját, a második jel a kontrolegység kódját, a harmadik jel pedig a perifériális egység kódját jelenti.

Pl.: UNIT=283 esetén a 2-es csatornán lévő, 8-as kontrolgéphez csatlakozó, 3-as kódú perifériális gépről van szó.

egységtípus

Egy egységtípust azonosít annak kódjával; pl.: UNIT=2311 esetén a 2311 típusu mágneslemezekről van szó.

(A különböző egységtípusok listája a 2. sz. függelékben található.)

csoportnév	Az egységeknek egy meghatározott csoportját azonosítja. Ez a csoport lehet egyelemű is. A csoport nevét a rendszer generálásakor kell megadni. Pl.: UNIT=SYSSQ jelentheti az összes mágnesszalagos és az összes mágneslemez-es egységet együttvéve.
egységek száma	Azoknak az egységeknek a darabszámát adja meg, amelyeket az állományhoz használni kívánunk.
	<i>Alapértelmezés: 1.</i>
P	Az állomány összes kötetéhez külön egységet rendelünk.
DEFER	A rendszernek ki kell jelölnie egy egységet az állományhoz, de az állományt tartalmazó kötetet az állomány megnyitásáig nem kell az egységre feltenni.
SEP=(ddnév1,...)	Azt jelenti, hogy az itt felsorolt nevű DD utasításokkal definiált egységekre nem kerülhet a szóban forgó állomány. A szeparálásnak ez a formája különbözik a 2.6.3.10 pontban leírt csatornaszeparálástól.
AFF=ddnév	Az állományra nézve a "ddnév" nevű DD utasítással definiált UNIT van érvényben. A "ddnév" ugyanezen job-lépés korábbi DD utasításának a neve.

Alapértelmezése

A rendszerkatalógusban az adott állománynévhez tartozó UNIT érték. Ha az állomány nem katalogizált, akkor a UNIT paraméter megadása kötelező.

Példák

a) //D10 DD UNIT=2400___

Ezzel az utasítással a 9 csatornás mágnesszalagegység típust jelöljük ki.

b) //D11 DD UNIT=191____

Az egység az 1-es csatornán levő 9-es kontrolegységhez csatlakozó, 1-es kódszámu konkrét egység.

c) //D12 DD UNIT=284____

//D13 DD UNIT=AFF=D12____

A D13 nevű DD utasításra nézve a D12 nevű DD utasításban definiált UNIT van érvényben.

2.6.3.4 A kötet (volume) kijelölése

- *Forma:*

$$\left[\begin{array}{l} \text{VOLUME} \\ \text{VOL} \end{array} \right] = ([\text{PRIVATE}] [, \text{RETAIN}] [, \text{sorszám}] [, \text{dbszám}] [,] \left[\begin{array}{l} \text{SER}=(\text{kötetnév}, \dots) \\ \text{REF}=\text{állománynév} \\ \text{REF}=\ast, \text{ddnév} \\ \text{REF}=\ast, \text{stepnév}, \text{ddnév} \\ \text{REF}=\ast, \text{stepnév}, \text{eljárás-} \\ \text{stepnév}, \text{ddnév} \end{array} \right]) \right]$$

— Az alparaméterek értelmezése

PRIVATE

A kötetet output állomány céljára a rendszer nem használhatja, kivéve, ha explicit igény alapján allokal a kötetre.

A kötetet a job-lépésben való utolsó használat után elengedi (kivéve, ha RETAIN van előírva).

RETAIN

Azt eredményezi, hogy a kötetet a rendszer a job-lépésben való utolsó használat után sem engedi el.

sorszám

Azt specifikálja, hogy egy létező többkötetes állomány esetén melyik kötetet akarjuk a feldolgozást kezdeni.

dbszám

Egy output állományhoz használt kötetek maximális darabszáma.

SER=(kötetnév, ...)

Azoknak a köteteknek a neve, amelyek az állomány elhelyezkedik. Ez lehet egyetlen kötet is.

REF = { állománynév
*.ddnév
*.stepnév.ddnév
*.stepnév.eljárásstepnév.ddnév }

Az itt definiálandó kötet legyen ugyanaz, mint a job egy *korábbi* DD utasításában definiált kötet volt, vagy legyen a katalógusban lévő állománynévhez tartozó kötet.

A VOLUME paraméter *alapértelmezése*:

A rendszerkatalógusban az adott állománynévhez tartozó VOLUME érték. Ha az állomány nincs katalógizálva és a UNIT értéke nem csoportnév, akkor a VOLUME megadása kötelező.

Például

//D1 DD UNIT=2311, VOL=(PRIVATE, , , SER=222222)___

A 222222 nevű kötetet kérjük, melyet output céljára a rendszer csak specifikus kéréssel használhat.

2.6.3.5 Az állománycímke-információk megadása

Forma:

[LABEL=([áll. sorsz.] [, SL [, PASSWORD [, IN [,] [EXPDT=yyddd]]] [, SUL [, NOPWREAD [, OUT [RETPD=nnnn]]] [, NSL [, NL [, BLP [,]]]]]]

Az alparaméterek jelentése

áll. sorsz.

Az állomány sorszámát jelenti az adott köteten belül. Olyan mágnesszalagnál használják, amelyen több állomány helyezkedik el.

Alapértelmezés: 1.

SL	Az állomány standard címkés.
SUL	Az állományt standard címkével és külön felhasználói címkével kérjük.
NSL	A mágnesszalagos állomány nem standard címkés.
NL	Címke nélküli mágnesszalagos állományt definiál.
BLP	Azt specifikálja, hogy a rendszer nem végez címkeellenőrzést a mágnesszalagon.
PASSWORD	Az állomány mindaddig nem használható, amíg az operátor nem közli a rendszerrel a jelszót.
NOPWREAD	Azt specifikálja, hogy az állományból olvashatunk a jelszó közlése nélkül, de rekord cseréje, törlése vagy új rekord beírása előtt az operátornak közölnie kell a rendszerrel a jelszót.
IN	Az állomány csak inputra használható.
OUT	Az állomány csak outputra használható.
EXPDT=yyddd	Az állomány az yy év (az évszám utolsó két jegye) <i>ddd</i> napjáig megőrződik, azaz az adott dátumig nem törölhető és nem írható felül.
RETPD=nnnn	Az állomány <i>nnnn</i> napig megőrződik, azaz <i>nnnn</i> napig nem törölhető és nem írható felül.

A LABEL paraméter alapértelmezése: SL.

(A kötet címkéjében levő információ, ha az állomány a lépés futása előtt létezett.)

.Példák

a) //C1 DD DSN=TAG, DISP=(NEW, KEEP), UNIT=2400,
 // VOL=SER=130120, LABEL=(4, NL, EXPDT=99365)

A TAG nevű állomány a job végrehajtása alatt keletkezik és az 1999. év 365. napjáig megőrzendő. A 130120 nevű 9 csatornás mágnesszalag 4. állománya lesz, címke nélkül.

b) //C2 DD DSN=A. B. Y3, DISP=OLD, LABEL=(, NSL),
 // UNIT=2400, VOL=SER=110280

Az A. B. Y3 nevű állomány nem standard címkes és a job futása előtt a 110280 nevű 9 csatornás mágnesszalagon már létezett.

2.6.3.6 Az adatvezérlő blokk (DCB) információk megadása

Az operációs rendszer az állománnyal többek között az adatvezérlő blokk (DCB) segítségével tartja fenn a kapcsolatot. Az adatvezérlő blokk egy kb. 90-100 byte terület a központi tárban, amely az illető perifériára, illetve állományra vonatkozó alapvető információkat tartalmazza, pl. a DD utasítás paramétereiként megadott értékeket is.

Az operációs rendszer az állomány azonosításához szükséges információkat az alábbi helyekről gyűjti össze:

- a programbeli adatvezérlő blokkból (DCB-ből).
- a JCL szinten megadott DD utasításból,
- a kötetben lévő címkéből (ha az állomány már korábban létezett).

Az állomány programbeli megnyitásakor (OPEN) ellenőrzi, hogy a különböző helyeken megadott paraméterek között nincs-e ellentmondás, illetve van-e hiányzó vagy hibásan közölt érték. Ha ilyen nincs, akkor a megnyitás sikeres lesz, ellenkező esetben a program futása az OPEN-nél megszakad.

Az állomány azonosításához szükséges paraméterek logikailag a következőképpen osztályozhatók:

- csak a programbeli DCB-ben adható meg,
- csak JCL szinten, DD-ben adható meg,
- DCB-ben is, DD-ben is megadható.

Ha egy paraméterhez a programbeli DCB-ben is és a JCL-beli DD-ben is rendelünk értéket, akkor a programbeli érték van érvényben. Un. perifériafüggetlen programot tehát úgy nyerhetünk, hogy a programon belül csak az itt kötelező paramétereket deklaráljuk, az összes többi

paramétert JCL szintre emeljük ki. Így a program megváltoztatása nélkül, kizárólag a hozzá tartozó JCL utasítások cseréjével dolgozhatunk fel különböző paraméterű állományokat.

Forma:

$$\left[\left\{ \begin{array}{l} \text{DCB}=(\text{alparaméter lista}) \\ \text{DCB}=(\left\{ \begin{array}{l} \text{állománynév} \\ \text{*. ddnév} \\ \text{*. stepnév. ddnév} \\ \text{*. stepnév. eljárásstepnév. ddnév} \end{array} \right\} [, \text{alp. lista}]) \end{array} \right\} \right]$$

Az alparaméterek jelentése

állománynév

A DCB információt a rendszer az "állománynév" nevű katalogizált állomány DCB információjának másolásával nyeri.

A katalogizált állománynak direkt hozzáférést megengedő köteten kell lennie és e kötetnek a szóban forgó job futása alatt az egységen kell lennie.

*. ddnév

A rendszer a DCB információt a jobnak egy korábbi, "ddnév" nevű DD utasításában megadott DCB információ másolásával nyeri.

*. stepnév. ddnév

A rendszer a DCB információt egy "stepnév" nevű, korábbi job-lépés "ddnév" nevű DD utasításban megadott DCB információ másolásával nyeri.

*. stepnév. eljárásstepnév. ddnév

A rendszer a DCB információt egy "stepnév" nevű, korábbi job-lépésből hívott JCL eljárás "eljárásstepnév" nevű lépésében lévő "ddnév" nevű DD utasításban definiált DCB információ másolásával nyeri.

A DCB alparaméter-lista a következő alparamétereket tartalmazhatja:

1. Az állomány szervezettsége

$$\left[\begin{array}{c} \text{DSORG} = \end{array} \right] \left[\begin{array}{c} \text{DA} \\ \text{DAU} \\ \text{GS} \\ \text{IS} \\ \text{ISU} \\ \text{PO} \\ \text{POU} \\ \text{PS} \\ \text{PSU} \end{array} \right]$$

Jelentésük:

DA	Direkt állomány (2.6 pont)
DAU	Direkt állomány, át nem helyezhető.
GS	Grafikus adatvezérlő blokk.
IS	Indexszekvenciális állomány.
ISU	Indexszekvenciális állomány, át nem helyezhető.
PO	Particionált állomány
POU	Particionált állomány, át nem helyezhető.
PS	Szekvenciális állomány.
PSU	Szekvenciális állomány, át nem helyezhető.

Alapértelmezése: PS.

A különböző feldolgozási módokkal kapcsolatban a következő szervezettségű állományok jöhetnek szóba:

Feldolgozási mód

Kapcsolatos állományfajták

BASIC DIRECT
ACCESS METHOD (BDAM)

DA, DAU, PS, PSU
(Megjegyzés: az utóbbi kettő csak az állomány létrehozásakor használható és ekkor a DD utasításban DA, az assembler programbeli DCB-ben PS irandó.)

BASIC INDEXED SEQUENTIAL ACCESS METHOD (BISAM)	IS
BASIC PARTITIONED ACCESS METHOD (BPAM)	PO, PO U
BASIC SEQUENTIAL ACCESS METHOD (BSAM)	PS, PSU
EXECUTE CHANNEL PROGRAM (EXCP)	PS, PO, DA, IS
GRAPHIC ACCESS METHOD (GAM)	GS
QUEUED INDEXED SEQUENTIAL ACCESS METHOD (QISAM)	IS, ISU
QUEUED SEQUENTIAL ACCESS METHOD (QSAM)	PS, PSU

Példa a DSORG alparaméter alkalmazására

```
//DOO DD DCB=(DSORG=PO____)____
```

A DOO nevű DD utasítással egy particionált állományt definiálunk.

2. A blokk mérete

```
[(BLKSIZE= blokkméret)]
```

Előírható BDAM, BPAM, BSAM, QISAM (csak outputra), QSAM esetén.

BDAM, BPAM, BSAM, QISAM, QSAM esetén

A blokk maximális hosszát adja byte-ban.

A specifikálható maximális érték 32760.

RECFM=F esetén a blokkméret nem lehet kisebb, mint a logikai rekordhossz. (A RECFM-re nézve lásd "A rekord formájának és jellemzőinek megadása" című részt a 2. 6. 3. 6 pontban.)

RECFM=FB esetén a blokkméret csak a logikai rekordhossz egész számu többszöröse lehet.

RECFM=V: a blokkméretbe a rekordhossz információt tartalmazó első szót is bele kell érteni.

RECFM=VB esetén a blokkméretbe a rekordhossz és blokkhossz információkat tartalmazó szavakat is bele kell érteni.

Alapértelmezése

A programbeli DCB-ben vagy a kötet címkéjében lévő érték.

Például:

```
//D3 DD DCB=BLKSIZE=3000____
```

A maximális blokkméret 3000 byte.

3. A blokk keresési határok megadása BDAM esetén

[LIMCT=szám]

BDAM és OPTCD=E esetén

Relativ blokk címzés esetén azoknak a blokkoknak a számát, relativ sávcímzés esetén azoknak a sávoknak a számát specifikálja, amelyekre a keresést előírjuk.

Alapértelmezés

A keresés az egész állományra kiterjed.

Megjegyzés Ha az OPTCD=E kiterjesztett keresés nincs megadva, akkor a LIMCT figyelmen kívül marad.

Például

```
//D21 DD DCB=(DSORG=DA, OPTCD=E,  
// LIMCT=108____)____
```

Az állományra kiterjesztett keresést írunk elő 108 blokknyi, illetve sávnyi intervallumra.

4. A rekord formájának és jellemzőinek megadása

[RECFM=tipus]

BDAM, BPAM, BSAM, EXCP,
QISAM output, QSAM esetén

Az állomány rekordjainak formáját és egyéb tulajdonságait specifikálja.

Tipusként a következő betűk bizonyos kombinációi írhatók elő:

- A A rekord első byte-ja standard sornyomtató vezérlő karaktert tartalmaz.
- B A rekordok blokkoltak.
- F Fix hosszúságú rekordok.
- M A rekord első byte-ja "gépi kód" vezérlő karaktert tartalmaz.

S *Fix hosszúságú rekordok esetén:*

a rekordok standard blokkonként íródnak fel; az utolsó blokk vagy sáv kivételével az állományban nincs csonka blokk vagy feltöltetlen sáv.

Változó hosszúságú rekordok esetén:

a rekord átnyulhat a következő sávra (spanned record). Ez esetben csere-puffer (BFTEK=E) nem használható.

- T A rekord a tulcsordulás sávra írható, ha szükséges. Ez esetben csere-puffer (BFTEK=E) nem használható.
- U Definiálatlan hosszúságú (undefined) rekordok.
- V Változó hosszúságú rekordok.

A RECFM alparaméter *alapértelmezése*: U.

Kivéve: QISAM esetén V az alapértelmezés.

A különböző feldolgozási módok esetén az alábbi kombinációk értelmesek:

BDAM esetén

$$\left\{ \begin{array}{l} U \\ V \\ F \quad [T] \end{array} \right\}$$

BPAM esetén

$$\left\{ \begin{array}{ll} U \quad [T] & \begin{bmatrix} A \\ M \end{bmatrix} \\ V & \begin{bmatrix} B \\ T \\ BT \end{bmatrix} \begin{bmatrix} A \\ M \end{bmatrix} \\ F & \begin{bmatrix} B \\ T \\ BT \end{bmatrix} \begin{bmatrix} A \\ M \end{bmatrix} \end{array} \right\}$$

BSAM és QSAM esetén

$$\left\{ \begin{array}{l} U \quad [T] \quad \begin{bmatrix} A \\ M \end{bmatrix} \\ V \quad \begin{bmatrix} B \\ S \\ T \\ BS \\ BT \\ ST \\ BST \end{bmatrix} \quad \begin{bmatrix} A \\ M \end{bmatrix} \\ F \quad \begin{bmatrix} B \\ S \\ T \\ BS \\ BT \\ ST \\ BST \end{bmatrix} \quad \begin{bmatrix} A \\ M \end{bmatrix} \end{array} \right\}$$

QISAM esetén

$$\left\{ \begin{array}{l} V \quad [B] \\ F \quad [B] \end{array} \right\}$$

Alapértelmezése

A DCB-ben vagy a kötet címkéjében lévő érték.

Például

//D29 DD DCB=(DSORG=PS, RECFM=VBA____)____

A D29 nevű DD utasítással leírt állomány változó hosszúságú blokkolt rekordokat tartalmaz és a sornytó vezérlésére a standard vezérlő karaktert használjuk.

5. A rekordhossz megadása

[LRECL=hossz]

BPAM, BSAM, QSAM és QISAM
output esetén

A logikai rekord aktuális vagy maximális hosszát jelenti. Fix és változó hosszúságú rekordok esetén a logikai rekordhossz szükséges; változó hosszúságú rekord esetén a maximális rekordhosszt jelenti.

LRECL ≤ BLKSIZE, kivéve a változó hosszúságú átnyúló (spanned) rekordok esetét.

Alapértelmezése

A DCB-ben vagy a kötet címkéjében lévő érték.

Például:

```
//D22 DD DCB=(BLKSIZE=300, LRECL=26,  
// DSORG=PS ___ ) ___
```

A logikai rekord nem nagyobb, mint 26 byte (változó hosszúságú vagy definiálatlan hosszúságú rekordok).

6. A rekordkulcs hosszának megadása

[KEYLEN=szám]

BDAM, BPAM, BSAM, EXCP és QISAM output esetén használt rekordkulcs hosszát jelenti byte egységben.

A maximális kulcshossz 255.

A kulcshossz alsó határa függ az állomány szervezettségétől.

Alapértelmezése

A DCB-ben vagy a kötet címkéjében levő érték.

Például:

```
//D20 DD DCB=(DSORG=IS, KEYLEN=12 ___ ) ___
```

Ezen indexszekvenciális állomány rekordjait 12 byte hosszúságú kulcsok azonosítják.

7. A rekordkulcs pozíciójának megadása

[RKP=szám]

QISAM output esetén

A rekordkulcs első byte-jának rekordon belüli pozícióját definiálja. A rekord első karakterére nézve RKP=0.

Ha RKP=0 van érvényben fix hosszúságú blokkolt rekordok esetén, akkor a kulcs a rekord első karakterével kezdődik és OPTCD=L nem specifikálható.

Változó hosszúságú rekordok esetén

Ha OPTCD=L van érvényben, akkor $RKP \geq 5$,
különben $RKP \geq 4$.

Alapértelmezés: $RKP=0$

Például:

//D32 DD DCB=(DSORG=IS, RKP=12, KEYLEN= 6 ____) ____

A D32 nevű DD utasítással definiált indexszekvenciális állomány rekordjait a rendszer a rekord 12. pozíciójától kezdődő 6 byte hosszúságú információ alapján (rekordkulcs) azonosítja.

8. A pufferezés típusa

$$\left[\text{BFTEK} = \begin{Bmatrix} \text{S} \\ \text{E} \\ \text{A} \\ \text{R} \end{Bmatrix} \right]$$

Jelentésük:

EXCP esetén:

S Egyszerű puffer.
E Cserepuffer.

Egyszerű puffer Általában blokknyi méretű központi tárbeli terület. Egy adott pufferben vagy csak input rekordok, vagy csak output rekordok (esetleg csak "update" rekordok) lehetnek. Tehát a rekordok mozgatása a központi tárban általában elkerülhetetlen.

Cserepuffer A szóbjövő pufferek rekordnyi és azonos hosszúságú részeit címeik cseréje által cseréli ki. Így kerül egy rekord pl. az input pufferből az output pufferbe. Tehát a rekordok mozgatása helyett a rekordláncokat módosítja.

QSAM esetén

S Egyszerű puffer.
E Cserepuffer. A RECFM alparaméterben tulcsordulás sáv nem specifikálható. Változó hosszúságú blokkolt rekordok vagy átnyúló (spanned) rekordok esetén nem használható.
A Rekord terület puffer.

Alapértelmezés: S.

BDAM vagy BSAM esetén:

- R Rekord puffer.
BDAM módon történő létrehozás esetén megengedi, hogy egy logikai rekord "átnyuljon" a következő sávra.

Például

//D2 DD DCB=(BFTEK=E ____) ____

A D2 nevű DD utasítással definiált állományt cserepuffer technikával kívánjuk feldolgozni.

9. A pufferek szóhatárra illesztése

$$\left[\text{BFALN} = \begin{Bmatrix} \text{F} \\ \text{D} \end{Bmatrix} \right]$$

F

Csak BDAM, BISAM, BPAM, BSAM, EXCP, QISAM, QSAM esetén,

valamennyi puffer teljesszó határon kezdődik,

D

valamennyi puffer duplaszó határon kezdődik.

Alapértelmezés: D.

10. A pufferek számának megadása

[BUFNO=a pufferek száma] BDAM, BISAM, BPAM, BSAM,
EXCP, QISAM és QSAM
esetén.

A szóban forgó adatvezérlő blokkhoz és egyben adatállományhoz tartozó pufferek számát jelenti.

Az előírható maximális érték 255, illetve a rendszergenerálásakor megadott korlát.

A *BUILD* nevű, pufferkészletet kijelölő assembler makró használata esetén kötelező specifikálni.

A *GETPOOL* nevű, pufferkijelölő assembler makró használatakor a rendszer a *GETPOOL* makróban megadott számú puffert használ (BDAM, BISAM, BPAM, BSAM, QISAM, QSAM).

Automatikus pufferezés (BPAM, BS AM) esetén kötelező specifikálni.

Automatikus pufferezés (QISAM, QSAM) esetén opcionális. Ha nem írjuk elő, akkor a rendszer 2 pufferral dolgozik.

Dinamikus pufferezés (BDAM, BISAM) esetén opcionális. Ha nem írjuk elő, akkor a rendszer 2 puffert használ.

Például

```
//D7 DD DCB=(BUFNO=5, BUFL=90    )
```

Itt 5 db, egyenként 90 byte hosszúságú puffert írunk elő.

11. A pufferek hosszának megadása

[BUFL=pufferhossz]

BDAM, BISAM, BPAM, EXCP,
QISAM, QSAM esetén használható.

A BUFL a puffer hosszát specifikálja.
Maximális hossz 32760.

BDAM esetén

A BUFL alparamétert csak akkor kell megadni, ha a kapcsolatos DCB makró dinamikus pufferezést ír elő.

BPAM, BSAM, ÉS QSAM esetén opcionális. Ha hiányzik, akkor a vezérlőprogram a blokkméret és a kulcshossz alapján határozza meg a puffer hosszát.

Kártyaolvasó és bináris olvasás esetén BUFL=160 kötelező.

BISAM és QISAM esetén

Nem szükséges, ha automatikus vagy dinamikus pufferezést irtunk elő.

Például

```
//D18 DD DCB=(BUFL=800_____)
```

Az itt megadott adatvezérlő blokkhoz 800 byte hosszú pufferek tartoznak.

12. A kártyaolvasási, illetve -lyukasztási mód előírása

$$\left[\text{MODE} = \begin{Bmatrix} \text{C} \\ \text{E} \end{Bmatrix} \right]$$

Az alparaméterek jelentése:

C bináris mód
E EBCDIC mód
C esetén BLKSIZE=160 kötelező.

Alapértelmezés: E.

Például

//D23 DD DCB=(BLKSIZE=160, MODE=C____)____

Az adatkártyákat binárisan kívánjuk olvasni.

13. A kártyalerakó kiválasztása

$$\left[\text{STACK} = \begin{Bmatrix} 1 \\ 2 \end{Bmatrix} \right] \quad \text{BSAM, EXCP, QSAM esetén}$$

STACK=1 esetben a kártyákat az 1 azonosítóju, STACK=2 esetben a 2 azonosítóju kártyalerakóba helyezi.

Alapértelmezés: 1.

Ha STACK paramétert megadtunk, akkor CODE, KEYLEN, PRTSP paraméter nem specifikálható.

Például

//D34 DD DCB=(DSORG=PS, STACK=2____)____

Valamennyi kártya a 2 azonosítóju lerakóba kerül.

14. A kód megadása lyukszalag esetén

$$\left[\text{CODE} = \begin{Bmatrix} \text{A} \\ \text{B} \\ \text{C} \\ \text{F} \\ \text{I} \\ \text{N} \\ \text{T} \end{Bmatrix} \right]$$

Az alparaméterek jelentése

A	ASCII kódú 8 csatornás lyukszalag
B	Burroughs kódú 7 csatornás lyukszalag
C	NCR kódú 8 csatornás lyukszalag
F	Friden kódú 8 csatornás lyukszalag
I	PTTC kódú 8 csatornás lyukszalag
N	Nem kérünk konvertálást
T	Teletype kódú 5 csatornás lyukszalag

Például

//D12 DD DCB=(DSORG =PS, CODE=C ____) ____

Inputnál a rendszer feltételezi, hogy a lyukszalag NCR kódban van lyukasztva és olvasás közben elvégzi az NCR → EBCDIC kódkonverziót.

Outputnál kiírás előtt konvertálja az adatokat EBCDIC-ből NCR kódra.

15. A mágnesszalag írássűrűségének megadása

$$\left[\text{DEN} = \left\{ \begin{array}{c} 0 \\ 2 \\ 3 \end{array} \right\} \right]$$

BSAM, EXCP, QSAM esetén

A DEN paraméter a mágnesszalag írássűrűségét adja meg az alábbiak szerint:

DEN=	<i>Írássűrűség</i> (jel/mm)
0	8
2	32
3	64

Szokásos *alapértelmezése*: 2

Például

//D15 DD UNIT=2400, DCB=(DEN=3 ____) ____

A D15 nevű DD utasítással leírt mágnesszalag írássűrűsége 64 jel/mm.

16. A cylinderen belüli, elsődleges tulcsordulás-terület kijelölése QISAM output esetén

[CYLOFL = szám]

Azt specifikálja, hogy az egyes cylindereken belül hány sáv legyen a tulcsordulás-terület.

Megszorítás a "szám" egész értékre:

$0 \leq \text{szám} \leq 99$.

Például

//D14 DD DCB=(DSORG=IS, CYLOFL=2 ___) ___

Minden cylinderen belül a két utolsó sáv a tulcsorduló rekordok számára van fenntartva.

17. A cylinder index terület megadása

[NTM=szám]

QISAM esetén

A cylinder index táblázat felépítéséhez szükséges sávok számát jelenti. Ha az előírt sávok megtelnek, a rendszer egy fő (master) index táblázatot épít fel.

NTM csak akkor szükséges, ha OPTCD=M van érvényben.

Ha NTM értéket nem specifikáltunk, de az OPTCD=M értéket igen, akkor az utóbbi figyelmen kívül marad.

Például

//D25 DD DCB=(DSORG=IS, NTM=3 ___) ___

Egy indexszekvenciális állomány felépítéséhez cylinder index táblázat céljára 3 sáv területet írtunk elő.

18. A sortávolság rögzítése sornyomtatón

$\left[\text{PRTSP} = \left\{ \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \end{array} \right\} \right]$

BSAM, EXPC, QSAM
esetén

Két sor írás közötti soremelések számát specifikálja.

Alapértelmezés: 1.

- Például

//D28 DD DCB=(DSORG=PS, PRTSP=3____)____

Két sor írás között 3 soremelés van a sornyomtatón.

19. Rendelkezés az input/output hibát „előidéző” blokkról

$$\left[\text{EROPT} = \left\{ \begin{array}{c} \text{ACC} \\ \text{SKP} \\ \text{ABE} \end{array} \right\} \right] \quad \text{QSAM esetén}$$

- Az alparaméterek jelentése

ACC	fogadja el a hibás blokkot
SKP	ugorja át a hibás blokkot
ABE	a rendszer törölje a szóban forgó programot (ABNORMAL END).

Alapértelmezés: ABE.

Például

//D16 DD DCB=(DSORG=PS, EROPT=SKP____)____

Valahányszor a gép hibás blokkot talál, mindannyiszor áttér a következő blokk írására, illetve olvasására.

20. A vezérlő program opcionális szolgáltatásainak specifikálása

$$\left[\text{OPTCD} = \begin{array}{c} \text{A} \\ \text{B} \\ \text{C} \\ \text{E} \\ \text{F} \\ \text{I} \\ \text{L} \\ \text{M} \\ \text{R} \\ \text{T} \\ \text{W} \\ \text{Y} \\ \text{Z} \end{array} \right] \quad \begin{array}{l} \text{BDAM, BPAM, BSAM, EXCP,} \\ \text{QISAM output, QSAM esetén} \end{array}$$

Megjegyzés Az OPTCD használata esetén a felsorolt betűk közül egymással ellentmondásban nem levőket egymás mellé kell írunk. Pl.: OPTCD=WILY.

Az alparaméterek jelentése

A	a READ és WRITE makróutasításokban ("block adress" operandus) az aktuális kötet cimeit használjuk.
B	mágnesszalag esetén a file vége felismerését nem kérjük.
C	BPAM, BSAM, QSAM esetén láncolás használatát kérjük.
E	kiterjesztett keresést (EXTENDED SEARCH) irunk elő a blokkokra, illetve sávokra. Ez esetben a LIMCT paramétert is specifikálni kell.
F	READ és WRITE makróutasítások végrehajtásakor a kötetbeli cím visszajelzését kérjük.
I	a vezérlőprogram használja az önálló tulcsordulás-területet a tulcsorduló rekordok elhelyezésére!
L	a vezérlőprogram használja fel az üres rekordok által elfoglalt területet, ha az állományba új rekordot ír be. (Üres rekord= az első byte minden bitje 1.)
M	indexszekvenciális állomány esetén fő (master) index felépítését kérjük. A főindex táblázat méretét az NTM paraméter adja. Ha NTM paramétert nem adtunk meg, akkor OPTCD=M figyelmen kívül marad.
R	BDAM esetén a READ és WRITE makróutasításokban ("block adress" operandus) az aktuális kötet cimeit használjuk.

Megjegyzés Az OPTCD használata esetén a felsorolt betűk közül egymással ellentmondásban nem levőket egymás mellé kell írunk. Pl.: OPTCD=WILY.

Az alparaméterek jelentése

A	a READ és WRITE makróutasításokban ("block adress" operandus) az aktuális kötet címeit használjuk.
B	mágnesszalag esetén a file vége felismerését nem kérjük.
C	BPAM, BSAM, QSAM esetén láncolás használatát kérjük.
E	kiterjesztett keresést (EXTENDED SEARCH) irunk elő a blokkokra, illetve sávokra. Ez esetben a LIMCT paramétert is specifikálni kell.
F	READ és WRITE makróutasítások végrehajtásakor a kötetbeli cím visszajelzését kérjük.
I	a vezérlőprogram használja az önálló tulcsordulás-területet a tulcsorduló rekordok elhelyezésére!
L	a vezérlőprogram használja fel az üres rekordok által elfoglalt területet, ha az állományba új rekordot ír be. (Üres rekord= az első byte minden bitje 1.)
M	indexszekvenciális állomány esetén fő (master) index felépítését kérjük. A főindex táblázat méretét az NTM paraméter adja. Ha NTM paramétert nem adtunk meg, akkor OPTCD=M figyelmen kívül marad.
R	BDAM esetén a READ és WRITE makróutasításokban ("block adress" operandus) az aktuális kötet címeit használjuk.

T	BSAM, illetve QSAM esetén ellenőrző összeg képzését kérjük.
W	BDAM, BPAM, BSAM, QSAM esetén közvetlen elérésű tárra íráskor ellenőrizze az írás helyességét!
Y	a vezérlőprogram használja a cylinder tulcsordulás-területeket a tulcsorduló rekordok elhelyezésére!
Z	mágnesszalagos input esetén a vezérlőprogram rövidítse a hiba javítási tevékenységét!

A különböző feldolgozási módok esetén az alábbi opciók értelmesek:

BDAM esetén:	A vagy R E F W
BPAM esetén:	C W WC
BSAM, QSAM esetén:	B C T W Z C WC ZC
EXCP esetén:	Z
QISAM esetén:	I L M R W Y

Például

```
//D26 DD DCB=(DSORG=IS, OPTCD=ILMWY,  
// NTM=2 ---)---
```

A D26 nevű DD utasításban leírt indexszekvenciális állományra vonatkozóan a vezérlőprogramtól az alábbi szolgáltatásokat kérjük:

- | | |
|----|--|
| a) | másodlagos tulcsordulás-terület használata |
| b) | az üres rekord területek felhasználása |
| c) | főindex felépítése |
| d) | az írás helyességének ellenőrzése |
| e) | az elsődleges tulcsordulás-terület használata. |

21. A READ, a WRITE és a CHECK makróutasítások használata

[NCP=szám]

BISAM, BPAM és BSAM esetén

Az egy CHECK makróutasítás előtt végrehajtandó, READ és WRITE utasítások maximális számát specifikálja.

A "szám" egész értékre:

$1 \leq \text{szám} \leq 99$,

illetve a rendszer generálásakor
megadott korlátot nem lépheti túl.

Alapértelmezés: 1.

OPTCD=C

esetén $NCP \geq 2$ kötelező.

Például

```
//D24 DD DCB=(DSORG=PS, NCP=3____)____
```

Egy CHECK makró előtt maximum 3 input/output makró használható.

Példák a DCB paraméter alkalmazására:

a) //D1 DD DSNAME=ALP, DISP=(NEW, KEEP), VOL=SER=112233,
 // UNIT=2400, DCB=(RECFM=FB, LRECL=240, BLKSIZE=960)

Az ALP nevű állomány a job-lépés végrehajtása során keletkezik, a job futása után megőrződik.

Az állományt a 112233 azonosítóju 9 csatornás mágnesszalagra írjuk a következők szerint:

4-4 db, egyenként 240 byte fix hosszúságú rekordot kívánunk egy-egy 960 byte hosszúságú blokkban elhelyezni.

b) //D2 DD DSNAME=BEL, DISP=OLD, DCB=(RECFM=VB,
 // LRECL=2048, BLKSIZE=3600), UNIT=2311,
 // VOL=SER=666666
 //D3 DD DSNAME=BECS, DISP=(NEW, CATLG, DELETE),
 // UNIT=2311, VOL=SER=555555, DCB=*.D2

A D3 nevű DD utasítással új állományt hozunk létre és a szükséges DCB információt a rendszer az ugyanezen job lépésen belüli D2 nevű DD utasításban megadott DCB információ másolásával nyeri.

c) //D4 DD DSN=AZ, DISP=(NEW, KEEP), UNIT=2311,
 // VOL=SER=444444, DCB=(A. B. C, KEYLEN=8)

A D4 nevű DD utasítással új állományt hozunk létre és a szükséges DCB információt a rendszer részben az A. B. C nevű katalogizált állomány címkéjéből nyeri, részben a D4 nevű DD utasításból (KEYLEN).

d) //D5 DD DSN=AM, DISP=(NEW, KEEP), UNIT=2400,
// VOL=SER=112234, DCB=(*.ST1.PROC5.D8, BUFNO=5).

A D5 nevű DD utasítással új állományt hozunk létre és a szükséges DCB információt a rendszer az ugyanezen job ST1 nevű korábbi lépéséből hívott PROC5 nevű JCL eljárás D8 nevű DD utasításában megadott DCB információ másolásával nyeri, illetve a pufferek számát a D5 nevű DD utasításban direkt megadjuk.

2.6.3.7 Terület kijelölése közvetlen hozzáférésű táron

Forma:

$$\left[\left[\begin{array}{l} \text{SPACE} = (\left[\begin{array}{l} \text{TRK} \\ \text{CYL} \\ \text{blokkhossz} \end{array} \right] , (\text{elsődleges} \left[\begin{array}{l} , \text{másodlagos} \left[\begin{array}{l} , \text{directory} \\ , \text{index} \end{array} \right] \left[\begin{array}{l} , \text{RLSE} \left[\begin{array}{l} , \text{CONTIG} \left[\begin{array}{l} , \text{ROUND} \end{array} \right] \end{array} \right] \left[\begin{array}{l} , \text{MXIG} \\ , \text{ALX} \end{array} \right] \end{array} \right] \end{array} \right] \end{array} \right) \end{array} \right] \right]$$

$$\left[\begin{array}{l} \text{SPACE} = (\text{ABSTR}, (\text{elsődleges}, \text{cim} \left[\begin{array}{l} , \text{directory} \\ , \text{index} \end{array} \right])) \end{array} \right]$$

Értelmezése

A SPACE paraméterrel az a terület írható le, amelyet a rendszer a direkt perifériákon (mágneslemez, mágneskártya, dob stb.) allokal.

Az alparaméterek jelentése

TRK	A területet sáv (track) egységben adjuk meg.
CYL	A területet cylinder egységben adjuk meg.
blokkhossz	Pozitív egész szám, mely az állomány átlagos blokkhosszát specifikálja. A rendszer ennek és az elsődleges mennyiségnek az alapján kiszámítja, hogy hány sáv területre van szükség.
elsődleges	Egész szám, amely azt definiálja, hogy hány sáv, cylinder, ill. blokk területre van szükség. Ha az adott kötetben még van ekkora terület, akkor a rendszer a SPACE paraméter első alparamétere és "elsődleges" által definiált területet még a program futása előtt allokalja.

másodlagos

Egész szám, amely azt jelenti, hogy *a program futása alatt* az eredetileg allokált perifériális területet a rendszer hány sáv, cylinder, illetve blokk területtel növelje meg, *ha szükséges*, azaz ha az eredeti terület betelt. A rendszer elvileg legfeljebb 15 ilyen területnövelést hajt végre. A gyakorlatban ez a maximum 15-nél kevesebb is lehet, amennyiben a szóban forgó kötetben már nincs több szabad hely.

directory

Egész szám, mely azoknak a 256 byte hosszú rekordoknak a darabszámát jelenti, amelyeket egy partitionált állomány tartalomjegyzéke (directory) részére definiálunk. E területet a felhasználónak kell definiálni, de a rendszer tölti ki.

index

Egész szám, mely azoknak a cylindereknek darabszámát jelenti, amelyeket egy indexszekvenciális állomány cylinder index táblázata részére jelölünk ki. Az indexterületet a felhasználónak kell definiálni (vagy a SPACE paraméterben, vagy egy "önálló" DD utasítással), de e területet a rendszer tölti ki.

RLSE

Azt jelenti, hogy a rendszer az allokált területből fel nem használt részt szabadítsa fel más állományok számára (CLOSE esetén).

CONTIG

Azt specifikálja, hogy elsődlegesként a rendszer összefüggő területet allokáljon.

MXIG

Azt jelenti, hogy az állomány részére a rendszer a kötetben levő legnagyobb összefüggő területet allokálja, amely legalább akkora, mint az elsődleges terület. Indexszekvenciális állomány esetén nem használható.

ALX

Azt specifikálja, hogy az állomány részére a rendszer legfeljebb 5 különálló területet allokáljon és min-

den ilyen terület legalább akkora legyen, mint amennyit az "elsődleges" mennyiséggel definiáltunk. Indexszekvenciális állomány esetén nem használható.

ROUND

Azt jelenti, hogy átlagos blokkhossz egységben definiáltuk a területet és a rendszer úgy allokaljon, hogy cylinderhatárra egészítse ki az egész területet.

ABSTR

Az állományt a kötet meghatározott helyén kívánjuk létrehozni.
Az egység ez esetben *sáv*.

cim

Az állomány első sávjának a kötetben belüli relatív sávcímét jelenti.

Alkalmazása

SPACE paraméter előírása akkor szükséges, ha közvetlen hozzáférést megengedő periférián új állományt hozunk létre vagy meglevő állomány (pl. SYSOUT, 2.6.3.13 pont) méretét változtatjuk. Egyéb esetekben hatástalan.

Példák a SPACE paraméter alkalmazására

a) //D1 DD DSN=ADAT3, UNIT=SYSDA, DISP=(NEW, PASS),
 // SPACE=(CYL, (30, 1))

Az ADAT3 állomány részére valamelyik direkt periférián 30 cylinder területet allokalítatunk. Ha szükséges, a rendszer a program futása közben ezt a periféria területet 1 cylinderenként növeli, legfeljebb 15 lépésben.

b) //D2 DD DSN=Z6, DISP=(NEW, KEEP), UNIT=2311,
 // VOL=SER=112233, SPACE=(360, (50, 2)),
 // DCB=(RECFM=FB, LRECL=36, BLKSIZE=360)

A D2 nevű DD utasítással leírt állomány számára egyenként 360 byte hosszú, 50 darab blokkot kérünk. Ha szükséges, a rendszer 2 blokkonként növeli a területet.

c) //D3 DD DSN=LIBR, DISP=(NEW, CATLG), UNIT=2311,
 // VOL=SER=333333, SPACE=(TRK, (60, , 3)),
 // DCB=DSORG=PO

A LIBR nevű könyvtári állomány a 333333 nevű lemezcsomagra kerül. Az állomány 60 sáv területet, a hozzá tartozó tartalomjegyzék 3*256 byte területet foglal el. A lépés végrehajtása után az állomány leírása bekerül a rendszerkatalógusba.

d) //D4 DD DSN=FIX, DISP=(NEW, CATLG), UNIT=2311,
// VOL=SER=ROZ001, SPACE=(ABSTR, (40, 20))

A FIX nevű állomány az ROZ001 nevű lemezcsomagon a 20. sávtól kezdődő 40 sáv területet foglalja el.

2.6.3.8 A terület kijelölése közvetlen hozzáférésű táron osztott cilinderekkel

Forma:

$$\left[\begin{array}{l} \text{SPLIT} = \left\{ \begin{array}{l} (n1, \text{CYL}, (\text{elsődleges} \left[\begin{array}{l} \text{, másodlagos} \end{array} \right])) \\ \text{mennyiség1} \left[\begin{array}{l} \text{mennyiség1} \end{array} \right] \\ n2 \\ (\text{százalék1}, \text{blokkhossz}, (\text{elsődleges} \left[\begin{array}{l} \text{, másodlagos} \end{array} \right])) \\ \text{mennyiség2} \left[\begin{array}{l} \text{mennyiség2} \end{array} \right] \\ \text{százalék2} \end{array} \right\} \end{array} \right]$$

Értelmezése

A SPLIT paraméter arra szolgál, hogy egy adott cylinder különböző sávjait különböző állományokhoz rendelhessük hozzá. Ennek előnye az, hogy ha több állományt párhuzamosan dolgozunk fel és a különböző állományok párhuzamosan feldolgozandó rekordjai egy cylinderen vannak, akkor a feldolgozás gyors lesz.

Az alparaméterek jelentése

n1	Egész szám, amely azon sávok számát jelenti cylinderenként, amelyeket az első állományhoz rendelünk.
CYL	A területet cylinder egységben adjuk meg.
elsődleges mennyiség1	Egész szám, mely azoknak a cylindereknek a számát specifikálja, amelyeket az állományokhoz használni kívánunk. A rendszer ezt a területet még a program futása előtt kijelöli.

másodlagos mennyiség1	Egész szám, amely azt jelenti, hogy a program futása alatt az eredetileg kijelölt perifériális területet a rendszer hány cylinderrel növelje meg, ha az elsődleges terület betelt.
n2	Egész szám, amely azon sávok számát jelenti cylinderenként, amelyeket egy további állományhoz rendelünk.
százalék1	$1 \leq \text{százalék} \leq 99$ és egész. jelenti azon sávok százalékarányát cylinderenként, amelyeket az első állományhoz rendelünk.
blokkhossz	Egész szám, amely az állomány átlagos blokkhosszát specifikálja. A területet átlagos blokkhossz egységben kérjük. A rendszer az átlagos blokkhossz és az elsődleges mennyiség alapján kiszámítja, hogy hány cylinder allokálása szükséges.
elsődleges mennyiség2	Egész szám, amely a kapcsolatos állományok totális blokkszámát specifikálja.
másodlagos mennyiség2	Egész szám, amely azt specifikálja, hogy a rendszer a program futása alatt az elsődleges területet hány blokknyi területtel növelje, ha szükséges.
százalék2	$1 \leq \text{százalék} \leq 99$ és egész. Azoknak a sávoknak a számát jelenti cylinderenként százalékban, amelyeket egy további állományhoz jellünk ki.

Alkalmazása

SPLIT paraméter előírása csak akkor szükséges, ha a cylindereket meg akarjuk osztani két vagy több állomány között.

Alapértelmezése: nincs cylinderosztás.

-Példák a SPLIT paraméter alkalmazására

a) //SD1 DD DSN=A, DISP=(, KEEP), UNIT=2311,
 // VOL=SER=666666, SPLIT=(5, CYL, (30, 2))
 //DS2 DD DSN=B, DISP=(, KEEP), SPLIT=3
 //DS3 DD DSN=C, DISP=(, KEEP), SPLIT=2

A 666666 nevű köteten kijelölt 30 cilindernyi terület minden egyes cylinderéből

5 sávot az A nevű állományhoz,
 3 sávot a B nevű állományhoz,
 2 sávot a C nevű állományhoz rendelünk.

b) //DS4 DD DSN=X, DISP=(, KEEP), UNIT=2311,
 // VOL=SER=112233, SPLIT=(20, 3600, (120, 2))
 //DS5 DD DSN=Y, DISP=(, KEEP), SPLIT=50
 //DS6 DD DSN=Z, DISP=(, KEEP), SPLIT=10
 //DS7 DD DSN=Q, DISP=(, KEEP), SPLIT=20

A 112233 azonosítóju kötetten allokalált 3600*120 byte terület minden egyes cylinderéből

az X nevű állományhoz 20 % sávot,
 az Y nevű állományhoz 50 % sávot,
 a Z nevű állományhoz 10 % sávot,
 a Q nevű állományhoz 20 % sávot

rendelünk.

(2311 típusu lemez esetében a 20, 50, 10, 20 % rendre 2, 5, 1, 2 sávot jelent, mivel itt 1 cylinder 10 sávból áll.)

2.6.3.9 Terület kijelölése közvetlen hozzáférésű táron egy korábban allokalált területből

Forma:

$$\left[\begin{array}{c} \text{SUBALLOC} = \left(\begin{array}{c} \text{TRK} \\ \text{CYL} \\ \text{blokkhossz} \end{array} \right) , \left(\begin{array}{c} \text{elsődleges} \\ \text{mennyiség} \end{array} \left[\begin{array}{c} \text{, másodlagos} \\ \text{mennyiség} \end{array} \right] , \text{directory} \right) \left(\begin{array}{c} \text{, ddnév} \\ \text{, stepnév, ddnév} \\ \text{, stepnév, eljárásstepnév,} \\ \text{ddnév} \end{array} \right) \right]$$

Értelmezése

A SUBALLOC paraméter alkalmazásával a szóban forgó job egy korábbi lépésében levő DD utasítással definiált területből részterületet allokalhatunk.

Az alparaméterek jelentése

Az első 4 alparaméter jelentése azonos a SPACE paraméter (2.6.3.7) megfelelő alparamétereinek jelentésével.

ddnév	A job szóban forgó lépésében levő DD utasítás neve.
stepnév	A job egy korábbi lépésének a neve.
eljárásstepnév	A job egy korábbi lépéséből hívott JCL eljárás egyik lépésének a neve.

Alkalmazása

SUBALLOC paraméter használata olyan esetekben célszerű, amikor megengedhetetlen vagy kényelmetlen, hogy a job-nak egy későbbi lépése azért nem futhat, mert akkorra már nincs elegendő hely a direkt perifériákon. Ha pl. a job első lépésében elég nagy területet allokálunk és ebből a későbbi lépéseknek SUBALLOC segítségével megfelelő területeket kihasítunk, akkor már a job első lépésénél kiderülhet, hogy az összes szükséges hely nem áll rendelkezésre.

Alapértelmezése

Önálló állományt allokál.

-Példák a SUBALLOC paraméter alkalmazására

```
a) //S1 EXEC PGM=A
    //DD1 DD DSN=MEZO, DISP=(, KEEP), UNIT=2311,
    // VOL=SER=444444, SPACE=(CYL, 50, , CONTIG)
    //S2 EXEC PGM=B
    //DD2 DD DSN=TER1, DISP=(, KEEP), SUBALLOC=(CYL,
    // (20, 1), S1.DD1)
    //DD3 DD DSN=TER2, DISP=(, KEEP), SUBALLOC=(TRK, (44, 7),
    // S1.DD1)
    //DD4 DD DSN=TER3, DISP=(, KEEP), SUBALLOC=(CYL,
    // 15, S1.DD1)
```

A 444444 azonosítóju lemezcsomagon levő MEZO nevű 50 cylinder összefüggő területből

a TER1 nevű állomány 20 cylinder,
a TER2 nevű állomány 44 sáv,
a TER3 nevű állomány 15 cylinder
területet foglal el.


```

b) //X EXEC PGM=X1
    //D1 DD DSN=ADAT1, DISP=(, KEEP), UNIT=2311,
    // VOL=SER=112233, SPACE=(CYL,100,, CONTIG)
    //D2 DD DSN=ADAT2, DISP=(, KEEP), SUBALLOC=(1024,
    // (800, 60), D1)
    //Y EXEC PGM=X2
    //D3 DD DSN=ADAT3, DISP=(, KEEP), SUBALLOC=(CYL,
    // (75,, 8), X.D1)

```

Az 112233 azonosítóju lemezcsomagon levő ADAT1 nevű 100 cylinder összefüggő területből 1024*800 byte területet az ADAT2 nevű állomány, 75 cylinder területet (+8*256 byte területet a tartalomjegyzék számára) az ADAT3 állomány foglal el.

2.6.3.10 Az állomány elkülönítése meglevő állományoktól

Forma:

```
[ SEP=(ddnév, ...) ]
```

Az alparaméter jelentése

ddnév

Ugyanazon job-ban lévő korábbi DD utasítás neve. Legfeljebb 8 ddnév írható elő.

A SEP paraméter alkalmazásával elérhető, hogy az állomány ne kerüljön ugyanarra a csatornára, mint az adott nevű DD utasítás(ok)ban leírt állomány(ok).

Alapértelmezése:

a rendszer nem szeparálja az állományt.

Példa a SEP paraméter alkalmazására

```

//S EXEC PGM=PROG1
//D1 DD DSN=N1, DISP=(OLD, PASS)
//D2 DD DSN=N2, DISP=(OLD, KEEP)
//D3 DD DSN=N3, DISP=(NEW, KEEP), VOL=SER=112233,
// UNIT=2311, SEP=(D1, D2)

```

A D3 nevű DD utasítással definiált N3 nevű állomány nem lehet sem az N1 nevű állományhoz, sem az N2 nevű állományhoz tartozó csatornán.

2.6.3.11 Hivatkozás korábbi SEP paraméterre

Forma:

[AFF=ddnév]

Az alparaméter jelentése

ddnév

a szóban forgó job-lépés korábbi DD utasításának neve.

Az AFF paraméter alkalmazásával a "ddnév" nevű DD utasításban levő SEP paraméterértéket írjuk elő. A rendszer csak akkor ügyel a szeparálásra, ha vagy az AFF, vagy a SEP paramétert kiírtuk.

Példa az AFF paraméter alkalmazására

```
//STEP1 EXEC PGM=C1  
//F1 DD DSN=A, DISP=OLD  
//F2 DD DSN=B, DISP=OLD, UNIT=2400, VOL=SER=110256  
//F3 DD DSN=C, DISP=(NEW, PASS), UNIT=2400, SEP=(F1, F2),  
// VOL=SER=110257  
//F4 DD DSN=D, UNIT=2400, DISP=(NEW, KEEP), AFF=F3,  
// VOL=SER=110258
```

Az F4 nevű DD utasítással definiált állomány (az F3 nevű DD utasítással definiált állományhoz hasonlóan) ne kerüljön se az F1, se az F2 nevű DD utasítással definiált csatornára.

2.6.3.12 Hivatkozás későbbi DD utasításra

Forma:

[DDNAME=ddnév]

ahol "ddnév" a szóban forgó job-lépés egy *későbbi* DD utasításának a neve.

Értelmezése

A DDNAME paraméter azt jelenti, hogy az állomány leírását a job-lépés egy későbbi, "ddnév" nevű DD utasításában adjuk meg.

Megszorítások

A DDNAME paraméter *mellett* csak a DCB paraméter BLKSIZE és BUFNO alparamétere írható elő.

JOBLIB nevű DD utasításban DDNAME nem írható.

DDNAME paraméter segítségével nem hivatkozhatunk olyan DD utasításra, amely maga is tartalmaz DDNAME hivatkozást.

Példa a DDNAME paraméter alkalmazására

```
//D1 EXEC PGM=*.LKED.SYSLMOD  
//FT06F001 DD SYSOUT=A  
//FT05F001 DD DDNAME=SYSIN  
//FT08F001 DD DSN=ALFA, DISP=OLD  
//SYSIN DD *  
    adatkártyák  
/*
```

Az FT05F001 nevű DD utasítás szerint az állomány leírását a SYSIN nevű későbbi DD utasításban adjuk meg. Az FT05F001 DD-néven belül a "05" egy FORTRAN periféria nevet jelent. A DDNAME paraméter segítségével az "05" FORTRAN periféria nevet "átjátszottuk" a SYSIN nevű DD utasításban leírt állományra.

2.6.3.13 Az output osztály megadása

Forma:

$$\left[\text{SYSOUT}=(\text{osztály} \left[, \text{program név} \right] \left[, \text{forma szám} \right]) \right]$$

Az alparaméterek értelmezése

osztály

Az angol ABC egy betűje vagy egy decimális számjegy. A program outputját ez esetben az operációs rendszer a rendszer-output állományba (SYSOUT, 2.6.3.13 pont), általában mágneslemezre írja, ahonnan a job befejezése után a kiíró program (writer) ugyanezt az output anyagot az osztálynak megfelelő output egységre továbbítja.

Szokasos megfeleltetés

A osztály

B osztály

programnév

sornyomtató

kártyalyukasztó

A rendszerkönyvtárban levő egyik program neve. Ha ezt a programnevet előírjuk, akkor a rendszer-output területéről az osztály által meghatározott output egységekre ez a program fogja kiírni az anyagot. Ha ezt a programnevet nem írjuk elő, akkor az output anyagnak az osztály által meghatározott egységekre történő kiírását a rendszer kiíró programja végzi.

formaszám

Legalább 1, legfeljebb 4 karakter. Alfa numerikus vagy ω , \$, #. Azt specifikálja, hogy az output állományt speciális formában kell a megfelelő egységekre kiírni. Ha ezt a paramétert használjuk, akkor a szóban forgó állomány kiírása előtt az operációs rendszer az operátornak konzolra kiírja a kiírási formára vonatkozó kívánságokat.

A SYSOUT paraméter *mellett* használható a UNIT, SPACE, OUTLIM és DCB paraméter.

Alapértelmezés

Ha a SYSOUT paramétert nem írjuk elő, akkor a szóban forgó állomány nem kerül a rendszer-outputba.

Példák a SYSOUT paraméter alkalmazására

a) //D1 DD SYSOUT=K

Az állomány a K osztály által meghatározott egységekre kerül.

b) //D2 DD SYSOUT=A, SPACE=(CYL,(80,2)), UNIT=2314

Az állomány az A osztály által meghatározott egységekre kerül. Az állomány számára a 2314 típusú lemezen levő rendszer-output területén 80 cilindert foglalunk le kiindulásképpen, s ha szükséges és lehetséges, akkor a rendszer ezt a területet 2 cylinderenként növeli, legfeljebb 15 lépésben.

2.6.3.14 A logikai rekordok számának korlátozása output állomány esetén

Forma:

[OUTLIM=szám]

ahol $1 \leq \text{szám} \leq 16777215$ és egész.

Jelentése

Az OUTLIM paraméter segítségével az output-áramba jutó állomány logikai rekordjainak számát maximalizálhatjuk.

Az OUTLIM paraméter csak a SYSOUT paraméterrel együtt hatásos; ha SYSOUT nincs, de OUTLIM van, akkor az OUTLIM figyelmen kívül marad.

Alapértelmezése

Ha OUTLIM paramétert nem írunk elő, akkor az állomány valamennyi rekordját kiírja.

Például

```
//OUT1 DD SYSOUT=A, OUTLIM=100
```

Az OUT1 nevű DD utasítással leírt állománynak csak az első 100 rekordja kerül az output-áramba és innen az A osztály által meghatározott egységre.

2.7 A JCL parancs

A parancsok az operációs rendszer munkájának menetét szabályozzák. Ez a szabályozás a gépkezelő kezében összpontosul.

A lehetőségek röviden: a beolvasó program, az ütemező program stb. indítása, program vagy job törlése, a rendszerállapot kiírása konzolra, kötet használatba vétele, illetve elengedése, a job osztályának vagy prioritásának megváltoztatása, a dátum és az idő megadása, online, ill. offline állapot beállítása stb.

2.8 A JCL megjegyzés

Forma:

```
//*szöveg
```

A megjegyzésnek a `//*` karakterhármassal kell kezdődnie. A magyarázó szöveg a 80. pozícióig írható. Ha a szöveg nem fér el egy sorban, akkor a folytatásnak egy újabb JCL megjegyzésben kell lennie.

Például

```
//* 41. SZAMU EPITOIPARI VALLALAT  
//* HATARIDO ELTOLODAS
```

2.9 A határoló utasítás

Forma:

XY szöveg

ahol XY a következő módon megadható delimiter:

```
//ddnév DD *, DLM='XY'
```

a "szöveg" pedig magyarázó szöveget jelent.

Jelentése

A határoló utasítás az input-árambeli adatok (2.6.2 pont) végét jelzi. (Az input-árambeli adatok kezdetét a `DD *` vagy a `DD DATA` utasítás jelzi.)

MFT,
MVT

A határoló utasítás `DD *` esetén nem kötelező, különben kötelező.

A határoló utasítás magyarázó szöveget tartalmazhat. A magyarázó szöveget nem lehet új sorban folytatni.

Alapértelmezése: `/*`

Például

```
//S1 EXEC PGM=A  
//D1 DD *  
    adatok  
/*
```


A //D1 DD * utasítás az input-árambeli adatok kezdetét jelöli. A D1 név az adatok azonosítására való.

A /* utasítás definiálja az adathalmaz végét.

2.10 A// utasítás

Forma:

//

Jelentése

A // utasítás egy job végét jelöli.

Minden jobot vagy egy // utasítással, vagy egy JOB utasítással (a következő job első utasításával) kell határolni.

Például:

```
//J1 JOB
//S1 EXEC PGM=B
//D1 DD DSN=A1, DISP=OLD
//D2 DD *
    adatok
/*
//D3 DD DSN=A2, DISP=OLD
//
```

Példánkban a job kezdetét a

```
//J1 JOB
utasítás, végét a
//
utasítás jelzi.
```

2.11 A JCL eljárás

2.11.1 BEVEZETÉS

A JCL eljárásdeklaráció JCL utasításokból álló utasítássorozat, amely vagy az input-áramban, vagy egy könyvtárban helyezkedik el.

A *szimbolikus paraméter* olyan, maximum 8 karakterből álló név, melynek első karaktere &, második karaktere nem számjegy és speciális jeleket nem tartalmaz.

Ha a szimbolikus paramétert szövegkonstans követi, akkor ezeket ponttal (.) kell elválasztani.

A szimbolikus paraméterhez tartozó alapértelmezés szerinti aktuális értéket magában az eljárásdeklarációban, ennek PROC utasításában lehet megadni (2.11.3 pont).

Az eljárás hívásakor a szimbolikus paraméterhez rendelt aktuális érték felülírja az eljárásdeklaráción belül megadott aktuális értéket.

Az operációs rendszer a JCL eljárás hívását a következőképpen dolgozza fel: az eljárás szimbolikus paramétereit helyettesíti az eljárás hívásakor megadott aktuális értékkel és az eljárásdeklaráció teljes szövegét bemásolja a job-nak arra a pontjára, ahonnan az eljárás hívódott.

2.11.2 AZ INPUT-ÁRAMBELI JCL ELJÁRÁSDEKLARÁCIÓ LEZÁRÁSA

Forma:

// [név] PEND [megjegyzés]

A PEND utasítás nem folytatható.

"név" csak akkor szükséges, ha az eljárásdeklarációban hivatkozunk erre a PEND utasításra.

2.11.3 ALAPÉRTTELMEZÉS SZERINTI ÉRTÉKEK HOZZÁRENDELÉSE A SZIMBOLIKUS PARAMÉTEREKHEZ

Forma:

// [név] PROC [operandusok] [megjegyzés]

- *Megszorítások*

Ha a JCL eljárásdeklaráció tartalmaz PROC utasítást, akkor ennek kell az első utasításnak lennie.

Ha az eljárásdeklaráció input-áramban van, akkor a "név" kötelező, s ez egyben az eljárás neve.

Az operandus megadása:

szimbolikus paraméternév = érték

A PROC utasításban a szimbolikus paraméterre az & jel kiírása nélkül kell hivatkozni.

Példa JCL eljárásdeklarációra

```
//LISTA PROC HALMAZ=A, MEMB=A1
//S1 EXEC PGM=IEBPTPCH
//SYSUT2 DD SYSOUT=A
//SYSUT1 DD DSN= & HALMAZ( & MEMB), DISP=OLD
//SYSIN DD DUMMY
// PEND
```

Ezzel a deklarációval a SYSUT1 nevű DD utasítás DSN= paraméterének a következő alapértelmezés szerinti értéket adtuk: DSN=A(A1).

2.11.4 PÉLDA JCL ELJÁRÁS ALKALMAZÁSÁRA

```
//J1 JOB
//RUTIN PROC PROG=1, RESZ=A1, VOL=111
//S1 EXEC PGM=ESZK & PROG
//OUT DD SYSOUT=A
//INP DD DSN= & RESZ, DISP=OLD, UNIT=2311,
// VOL=SER= & VOL.000
// PEND
//ST1 EXEC RUTIN
//ST2 EXEC RUTIN, PROG=2, RESZ=A2, VOL=222
//ST3 EXEC RUTIN, RESZ=A3, VOL=333
//
```

Ez a J1 nevű job egy RUTIN nevű JCL eljárás deklarációját és ezen eljárás 3 különböző hívását tartalmazza.

Ebből a job-ból, miután a rendszer feldolgozta az eljáráshívásokat, a következő job lesz:

```
//J1 JOB
//S1 EXEC PGM=ESZK1
//OUT DD SYSOUT=A
//INP DD DSN=A1, DISP=OLD, UNIT=2311,
// VOL=SER=111000
//S1 EXEC PGM=ESZK2
//OUT DD SYSOUT=A
//INP DD DSN=A2, DISP=OLD, UNIT=2311,
// VOL=SER=222000
//S1 EXEC PGM=ESZK1
//OUT DD SYSOUT=A
//INP DD DSN=A3, DISP=OLD, UNIT=2311,
// VOL=SER=333000
//
```

Az operációs rendszer ezt a job-ot fogja végrehajtani.

2.11.5 DD UTASÍTÁS MEGADÁSA JCL ELJÁRÁS VALAMELY LÉPÉSÉHEZ

Forma:

//eljárásstepnév. ddnév DD ___

Értelmezése

eljárásstepnév

a JCL eljárás azon lépésének a neve, amelyhez a DD utasítást megadjuk.

ddnév

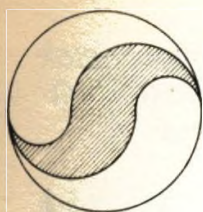
a DD utasítás neve.

Valamely eljárásstephez úgy adunk meg DD utasítást, hogy a DD utasítás neve előtt kiírjuk az "eljárásstep" nevét.

Például

//PL1L.SYSIN DD ___

jelenti: a PL1L nevű "eljárásstep"-hez tartozó SYSIN nevű DD.



1. Függelék

A programok fordítására, szerkesztésére és futtatására vonatkozó katalogizált JCL eljárások

Elöl az eljárás neve áll, utána az eljárás tartalma.

Az eljárás neve és az eljárás által aktivizált programok közötti kapcsolat a következő:

<i>Az eljárásnév végződése</i>	<i>Az eljárás által aktivizált programok</i>
C	A megfelelő fordítóprogram.
CL	A megfelelő fordítóprogram, majd a szerkesztőprogram (Linkage Editor).
CLG	A megfelelő fordítóprogram, majd a szerkesztőprogram, majd a szerkesztett program.
CG	A megfelelő fordítóprogram, majd a szerkesztőbetöltő program (Loader).
LG	A szerkesztőprogram, majd a szerkesztett program.
G	A szerkesztő-betöltő program (Loader).

Az eljárásbeli EXEC utasítás a megfelelő programot azonosítja, az utána következő DD utasítások az adott program működéséhez szükséges állományokat írják le.

A SYSUT1, SYSUT2, SYSUT3 stb. nevű DD utasítások a fordítóprogram munkájához szükséges ideiglenes állományokat definiálják. Az &LOADSET nevű állományba (ha van ilyen) kerül a lefordított program (tárgymodul), az &GOSET nevű könyvtári állomány (ha van) GO vagy MAIN, vagy RUN nevű member-jébe pedig a szerkesztett program (load modul).

A SYSPRINT nevű DD utasítás a sornyomtatót (SYSOUT=A), a SYSPCH nevű pedig a kártyalyukasztót (SYSOUT=B) azonosítja.

A különböző katalogizált JCL eljárások a megfelelő fordítóprogram egy adott változatát aktivizálják. Ebben a függelékben a következő változatok szerepelnek:

<i>Megnevezés</i>	<i>A fordítóprogram neve</i>	<i>A fordítóprogramnak melyik változatáról van szó</i>
Algol	ALGOL	F
Assembler	IEUASM	F
COBOL	IEQCBLOO	F
FORTRAN	IEYFORT	G
PL/1	IEMAA	F

ALGOFC

```
//ALGOL EXEC PGM=ALGOL, REGION=48K
//SYSPRINT DD SYSOUT=A
//SYSPUNCH DD SYSOUT=B
//SYSLIN DD DSN=&LOADSET, UNIT=SYSSQ, SEP=SYSPUNCH,
// SPACE=(3600, (10, 4)), DISP=(MOD, PASS)
//SYSUT1 DD DSN=&SYSUT1, UNIT=SYSSQ, SEP=SYSPRINT,
// SPACE=(1024, (50, 10))
//SYSUT2 DD DSN=&SYSUT2, UNIT=SYSSQ, SEP=SYSUT1,
// SPACE=(1024, (50, 10))
//SYSUT3 DD DSN=&SYSUT3, UNIT=SYSDA,
// SPACE=(1024, (40, 10))
```

ALGOFCL

```
//ALGOL EXEC PGM=ALGOL, REGION=48K
//SYSPRINT DD SYSOUT=A
//SYSPUNCH DD SYSOUT=B
//SYSLIN DD DSN=&LOADSET, UNIT=SYSSQ, SEP=SYSPUNCH,
// SPACE=(3600, (10, 4)), DISP=(MOD, PASS)
//SYSUT1 DD DSN=&SYSUT1, UNIT=SYSSQ, SEP=SYSPRINT,
// SPACE=(1024, (50, 10))
//SYSUT2 DD DSN=&SYSUT2, UNIT=SYSSQ, SEP=SYSUT1,
// SPACE=(1024, (50, 10))
//SYSUT3 DD DSN=&SYSUT3, UNIT=SYSDA,
// SPACE=(1024, (40, 10))
//LKED EXEC PGM=IEWL, PARM='XREF, LIST, LET',
// COND=(5, LT, ALGOL), REGION=96K
//SYSPRINT DD SYSOUT=A
//SYSLIN DD DSN=&LOADSET, DISP=(OLD, DELETE)
// DD DDNAME=SYSIN
//SYSLIB DD DSN=SYS1.ALGLIB, DISP=SHR
//SYSLMOD DD DSN=&GOSET(GO), UNIT=SYSDA,
// DISP=(MOD, PASS), SPACE=(1024, (50, 20, 1))
```



```
//SYSUT1 DD DSN=& SYSUT1, UNIT=SYSDA,
// SEP=(SYSLIB, SYSLMOD), SPACE=(1024, (50, 20))
```

ALGOFCLG

```
//ALGOFCLG PROC MEMB=GO
//ALGOL EXEC PGM=ALGOL, REGION=48K
//SYSPRINT DD SYSOUT=A
//SYSPUNCH DD SYSOUT=B
//SYSLIN DD DSN=& LOADSET, UNIT=SYSSQ,
// SEP=SYSPUNCH, SPACE=(3600, (10, 4)),
// DISP=(MOD, PASS)
//SYSUT1 DD DSN=& SYSUT1, UNIT=SYSSQ, SEP=SYSPRINT,
// SPACE=(1024, (50, 10))
//SYSUT2 DD DSN=& SYSUT2, UNIT=SYSSQ, SEP=SYSUT1,
// SPACE=(1024, (50, 10))
//SYSUT3 DD DSN=& SYSUT3, UNIT=SYSDA,
// SPACE=(1024, (40, 10))
//LKED EXEC PGM=IEWL, PARM='XREF, LIST, LET',
// COND=(5, LT, ALGOL), REGION=96K
//SYSPRINT DD SYSOUT=A
//SYSLIN DD DSN=& LOADSET, DISP=(OLD, DELETE)
// DD DDNAME=SYSIN
//SYSLIB DD DSN=SYS1.ALGLIB, DISP=SHR
//SYSLMOD DD DSN=& GOSET(& MEMB), UNIT=SYSDA,
// DISP=(MOD, PASS), SPACE=(1024, (50, 20, 1))
//SYSUT1 DD DSN=& SYSUT1, UNIT=SYSDA,
// SEP=(SYSLIB, SYSLMOD),
// SPACE=(1024, (50, 20))
//GO EXEC PGM=* . LKED, SYSLMOD,
// COND=((5, LT, ALGOL), (5, LT, LKED))
//ALGLDD01 DD SYSOUT=A
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSN=& SYSUT1, UNIT=SYSSQ,
// SPACE=(1024, (20, 10))
```

ALGOFCG

```
//ALGOL EXEC PGM=ALGOL, REGION=48K
//SYSPRINT DD SYSOUT=A
//SYSPUNCH DD SYSOUT=B
//SYSLIN DD DSN=& LOADSET, UNIT=SYSSQ,
// SEP=SYSPUNCH, SPACE=(3600, (10, 4)),
// DISP=(MOD, PASS)
//SYSUT1 DD DSN=& SYSUT1, UNIT=SYSSQ, SEP=SYSPRINT,
// SPACE=(1024, (50, 10))
```



```

//SYSUT2      DD DSN=&SYSUT2,UNIT=SYSSQ,SEP=SYSUT1,
//            SPACE=(1024,(50,10))
//SYSUT3      DD DSN=&SYSUT3,UNIT=SYSDA,
//            SPACE=(1024,(40,10))
//GO          EXEC PGM=LOADER,PARM=(MAP,LET,PRINT),
//            COND=(5,LT,ALGOL)
//SYSLIN      DD DSN=&LOADSET,DISP=(OLD,DELETE)
//SYSLIB      DD DSN=SYS1.ALGLIB,DISP=SHR
//LOADPRNT    DD SYSOUT=A
//SYSPRINT    DD SYSOUT=A
//ALGLDD01    DD SYSOUT=A
//SYSUT1      DD DSN=&SYSUT1,UNIT=SYSSQ,
//            SPACE=(1024,(20,10))

```

ASMFC

```

//ASM        EXEC PGM=IEUASM,REGION=50K
//SYSLIB      DD DSN=SYS1.MACLIB,DISP=SHR
//SYSUT1      DD DSN=&SYSUT1,UNIT=SYSSQ,
//            SPACE=(1700,(400,50)),SEP=SYSLIB
//SYSUT2      DD DSN=&SYSUT2,UNIT=SYSSQ,
//            SPACE=(1700,(400,50))
//SYSUT3      DD DSN=&SYSUT3,UNIT=(SYSSQ,
//            SEP=(SYSUT2,SYSUT1,SYSLIB)),
//            SPACE=(1700,(400,50))
//SYSPRINT    DD SYSOUT=A
//SYSPUNCH    DD SYSOUT=B

```

ASMFL

```

//ASM        EXEC PGM=IEUASM,PARM=LOAD,REGION=50K
//SYSLIB      DD DSN=SYS1.MACLIB,DISP=SHR
//SYSUT1      DD DSN=&SYSUT1,UNIT=SYSSQ,
//            SPACE=(1700,(400,50)),SEP=SYSLIB
//SYSUT2      DD DSN=&SYSUT2,UNIT=SYSSQ,
//            SPACE=(1700,(400,50))
//SYSUT3      DD DSN=&SYSUT3,UNIT=(SYSSQ,
//            SEP=(SYSUT2,SYSUT1,SYSLIB)),
//            SPACE=(1700,(400,50))
//SYSPRINT    DD SYSOUT=A
//SYSPUNCH    DD SYSOUT=B
//SYSGO      DD DSN=&LOADSET,UNIT=SYSSQ,SPACE=
//            (80,(100,50)),DISP=(MOD,PASS)
//LKED        EXEC PGM=IEWL,PARM=(XREF,LIST,NCAL),
//            REGION=96K,COND=(8,LT,ASM)

```



```
//SYSLIN      DD DSNNAME=&LOADSET, DISP=(OLD,DELETE)
//           DD DDNAME=SY SIN
//SYSLMOD     DD DSNNAME=&GOSET(GO), UNIT=SYSDA,
//           SPACE=(1024,(50,20,1)),DISP=(MOD,PASS)
//SYSUT1      DD DSNNAME=&SYSUT1, UNIT=(SYSDA,
//           SEP=(SYSLIN,SYSLMOD)),SPACE=(1024,(50,20))
//SYSPRINT    DD SYSOUT=A
```

ASMFCLG

```
//ASMF        PROC A=GO
//ASM         EXEC PGM=IEUASM, PARM=LOAD, REGION=50K
//SYSLIB      DD DSNNAME=SYS1.MACLIB, DISP=SHR
//SYSUT1      DD DSNNAME=&SYSUT1, UNIT=SYSSQ,
//           SPACE=(1700,(400,50)), SEP=SYSLIB
//SYSUT2      DD DSNNAME=&SYSUT2, UNIT=SYSSQ,
//           SPACE=(1700,(400,50))
//SYSUT3      DD DSNNAME=&SYSUT3, UNIT=(SYSSQ,
//           SEP=(SYSUT2, SYSUT1, SYSLIB)),SPACE=(1700,
//           (400,50))
//SYSPRINT    DD SYSOUT=A
//SYSPUNCH    DD SYSOUT=B
//SYSGO       DD DSNNAME=&LOADSET, UNIT=SYSSQ,
//           SPACE=(80,(100,50)),DISP=(MOD,PASS)
//LKED        EXEC PGM=IEWL, PARM=(XREF,LET,LIST,NCAL),
//           REGION=96K, COND=(8,LT,ASM)
//SYSLIN      DD DSNNAME=&LOADSET, DISP=(OLD,DELETE)
//           DD DDNAME=SY SIN
//SYSLMOD     DD DSNNAME=&GOSET(&A), UNIT=SYSDA,
//           SPACE=(1024,(50,20,1)),DISP=(MOD,PASS)
//SYSUT1      DD DSNNAME=&SYSUT1, UNIT=(SYSDA,
//           SEP=(SYSLIN,SYSLMOD)),SPACE=(1024,
//           (50,20))
//SYSPRINT    DD SYSOUT=A
//GO          EXEC PGM=*.LKED.SYSLMOD,
//           COND=((8,LT,ASM),
//           (4,LT,LKED))
```

ASMFCG

```
//ASM         EXEC PGM=IEUASM, PARM='LOAD', REGION=50K
//SYSLIB      DD DSNNAME=SYS1.MACLIB, DISP=SHR
//SYSUT1      DD DSNNAME=&SYSUT1, UNIT=SYSSQ,
//           SPACE=(1700,(400,50)), SEP=(SYSLIB)
//SYSUT2      DD DSNNAME=&SYSUT2, UNIT=SYSSQ,
//           SPACE=(1700,(400,50))
//SYSUT3      DD DSNNAME=&SYSUT3, SPACE=(1700,(400,50)),
//           UNIT=(SYSSQ,SEP=(SYSUT2,SYUT1,SYSLIB))
```



```
//SYSPRINT DD SYSOUT=A
//SYSPUNCH DD SYSOUT=B
//SYSGO DD DSNAME= & LOADSET, UNIT=SYSSQ,
// SPACE=(80, (200, 50)), DISP=(MOD, PASS)
//GO EXEC PGM=LOADER, PARM=(MAP, PRINT,
// NOCALL, LET)
//SYSLIN DD DSNAME= & LOADSET, DISP=(OLD, DELETE)
//SYSLOUT DD SYSOUT=A
```

COBFC

```
//COB EXEC PGM=IEQCBLOO, PARM=' DECK, NOLOAD' ,
// REGION=86K
//SYSPRINT DD SYSOUT=A
//SYSPUNCH DD SYSOUT=B
//SYSUT1 DD UNIT=SYSDA, SPACE=(460,(700,100))
//SYSUT2 DD UNIT=SYSDA, SPACE=(460, (700,100))
//SYSUT3 DD UNIT=SYSDA, SPACE=(460, (700,100))
//SYSUT4 DD UNIT=SYSDA, SPACE=(460, (700,100))
```

COBFLG

```
//COBFLG PROC AB=RUN
//LKED EXEC PGM=IEWL, PARM=' LIST, XREF, LET' ,
// REGION=96K
//SYSLIN DD DDNAME=SYSIN
//SYSLMOD DD DSNAME= & GODATA(& AB), DISP=(NEW,
// PASS), UNIT=SYSDA, SPACE=(1024, (50, 20, 1))
//SYSLIB DD DSNAME=SYS1.COBLIB, DISP=SHR
//SYSUT1 DD UNIT=(SYSDA, SEP=(SYSLIN, SYSLMOD),
// SPACE= (1024, (50, 20))
//SYSPRINT DD SYSOUT= A
//GO EXEC PGM=*. LKED. SYSLMOD, COND=(5, LT, LKED)
//SYSABEND DD SYSOUT=A
```

COBFCLG

```
//COBFCLG PROC A=RUN
//COB EXEC PGM=IEQCBLOO, REGION=86K
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD UNIT=SYSDA, SPACE=(460, (700,100))
//SYSUT2 DD UNIT=SYSDA, SPACE=(460, (700,100))
//SYSUT3 DD UNIT=SYSDA, SPACE=(460, (700,100))
//SYSUT4 DD UNIT=SYSDA, SPACE=(460, (700,100))
```



```

//SYSLIN      DD  DSNAME= & LOADSET, DISP=(MOD, PASS),
//              UNIT=SYSDA, SPACE=(80, (500, 100))
//LKED        EXEC PGM=IEWL, PARM=' LIST, XREF, LET' ,
//              COND=(5, LT, COB), REGION=96K
//SYSLIN      DD  DSNAME= & LOADSET, DISP=(OLD, DELETE)
//              DD  DDNAME=SYSIN
//SYSLMOD     DD  DSNAME= & GODATA(& A), DISP=(NEW, PASS),
//              UNIT=SYSDA, SPACE=(1024, (50, 20, 1))
//SYSLIB      DD  DSNAME=SYS1.COBLIB, DISP=SHR
//SYSUT1      DD  UNIT=(SYSDA, SEP=(SYSLIN, SYSLMOD)),
//              SPACE=(1024, (50, 20))
//SYSPRINT    DD  SYSOUT=A
//GO          EXEC PGM=*.LKED.SYSLMOD, COND=((5, LT, COB),
//              (5, LT, LKED))
//SYSABEND    DD  SYSOUT=A

```

FORTGC

```

//FORT        EXEC PGM=IEYFORT, REGION=100K
//SYSPRINT    DD  SYSOUT=A
//SYSPUNCH    DD  SYSOUT=B
//SYSLIN      DD  DSNAME= & LOADSET, DISP=(MOD, PASS),
//              UNIT=SYSSQ, SPACE=(80, (200, 100), RLSE),
//              DCB=BLKSIZE=80

```

FORTGCL

```

//FORT        EXEC PGM=IEYFORT, REGION=100K
//SYSPRINT    DD  SYSOUT=A
//SYSPUNCH    DD  SYSOUT=B
//SYSLIN      DD  DSNAME= & LOADSET, DISP=(MOD, PASS),
//              UNIT=SYSSQ, SPACE=(80, (200, 100), RLSE),
//              DCB=BLKSIZE=80
//LKED        EXEC PGM=IEWL, REGION=96K, PARM=(XREF,
//              LET, LIST), COND=(4, LT, FORT)
//SYSLIB      DD  DSNAME=SYS1.FORTLIB, DISP=SHR
//SYSLMOD     DD  DSNAME= & GOSET(MAIN), DISP=(NEW, PASS),
//              UNIT=SYSDA, SPACE=(1024, (20, 10, 1), RLSE),
//              DCB=BLKSIZE=1024
//SYSPRINT    DD  SYSOUT=A
//SYSUT1      DD  DSNAME= & SYSUT1, UNIT=SYSDA,
//              SPACE=(1024, (20, 10), RLSE),
//              DCB=BLKSIZE=1024
//SYSLIN      DD  DSNAME= & LOADSET, DISP=(OLD, DELETE)
//              DD  DDNAME=SYSIN

```


FORTGCLG

```
//FORTG   PROC   AB=MAIN
//FORT    EXEC   PGM=IEYFORT, REGION=100K
//SYSPRINT DD   SYSOUT=A
//SYSPUNCH DD   SYSOUT=B
//SYSLIN   DD   DSNNAME= & LOADSET, DISP=(MOD, PASS),
//              UNIT=SYSSQ, SPACE=(80, (200, 100), RLSE),
//              DCB=BLKSIZE=80
//LKED     EXEC   PGM=IEWL, REGION=96K, PARM=(XREF,
//              LET, LIST), COND=(4, LT, FORT)
//SYSLIB    DD   DSNNAME=SYS1.FORTLIB, DISP=SHR
//SYSLMOD   DD   DSNNAME= & GOSET( & AB), DISP=(NEW, PASS),
//              UNIT=SYSDA , SPACE= (1024, (20, 10, 1),
//              RLSE), DCB=BLKSIZE=1024,
//SYSPRINT   DD   SYSOUT=A
//SYSUT1    DD   DSNNAME= & SYSUT1, UNIT=SYSDA,
//              SPACE=(1024, (20, 10), RLSE), DCB=BLKSIZE=1024
//SYSLIN     DD   DSNNAME= & LOADSET, DISP=(OLD, DELETE)
//              DD   DDNAME=SYSIN
//GO         EXEC   PGM=*. LKED. SYSLMOD, COND=((4, LT,
//              FORT), (4, LT, LKED))
//FT05F001   DD   DDNAME=SYSIN
//FT06F001   DD   SYSOUT=A
//FT07F001   DD   SYSOUT=B
```

PL1DFC

```
//PL1D     EXEC   PGM=IEMAA, PARM=' DECK, NOLOAD' ,
//              REGION=52K
//SYSPRINT   DD   SYSOUT=A
//SYSPUNCH   DD   SYSOUT=B
//SYSUT3     DD   DSNNAME= & & SYSUT3, UNIT=SYSSQ,
//              SPACE=(80, (250, 250)), SEP=SYSPRINT
//SYSUT1     DD   DSNNAME= & & SYSUT1, UNIT=SYSDA,
//              SPACE=(1024, (60, 60), , CONTIG), SEP=(SY SUT3,
//              SYSPRINT, SYSPUNCH)
```


FL1LFC

```
//PL1L   EXEC  PGM=IEMAA, PARM=' LOAD, NODECK',  
//  
//SYSPRINT DD  SYSOUT=A  
//SYSLIN   DD  DSNNAME=& &LOADSET, DISP=(MOD, PASS),  
//  
//          UNIT=SYSSQ, SPACE=(80, (250, 100))  
//SYSUT3   DD  DSNNAME=& &SYSUT3, UNIT=SYSSQ,  
//          SPACE=(80, (250, 250)), SEP=SYSPRINT  
//SYSUT1   DD  DSNNAME=& &SYSUT1, UNIT=SYSDA,  
//          SPACE=(1024, (60, 60), , C ONTIG),  
//          SEP=(SYSUT3, SYSPRINT, SYSLIN)
```

PL1LFCL

```
//PL1L   EXEC  PGM=IEMAA, PARM=' LOAD, NODECK',  
//  
//          REGION=52K  
//SYSPRINT DD  SYSOUT=A  
//SYSLIN   DD  DSNNAME=&&LOADSET, DISP=(MOD, PASS),  
//          UNIT=SYSSQ, SPACE=(80, (250, 100))  
//SYSUT3   DD  DSNNAME=&&SYSUT3, UNIT=SYSSQ,  
//          SPACE=(80, (250, 250)), SEP=SYSPRINT  
//SYSUT1   DD  DSNNAME=&&SYSUT1, UNIT=SYSDA,  
//          SPACE=(1024, (60, 60), , CONTIG), SEP=(SYSUT3,  
//          SYSPRINT, SYSLIN)  
//LKED     EXEC  PGM=IEWL, PARM='XREF, LIST',  
//  
//          COND=(9, LT, PL1L), REGION=96K  
//SYSLIB    DD  DSNNAME=SYS1.PL1LIB, DISP=SHR  
//SYSLMOD   DD  DSNNAME=&&GOSET(GO), DISP=(MOD, PASS),  
//          UNIT=SYSDA, SPACE=(1024, (50, 20, 1), RLSE)  
//SYSUT1    DD  DSNNAME=&&SYSUT1, UNIT=SYSDA,  
//          SEP=(SYSLMOD, SYSLIB), SPACE=(1024, (200, 20))  
//SYSPRINT  DD  SYSOUT=A  
//SYSLIN    DD  DSNNAME=&&LOADSET, DISP=(OLD, DELETE)  
//          DD  DDNAME=SY SIN
```

PL1FLG

```
//PL1LF    PROC  A=GO  
//LKED     EXEC  PGM=IEWL, PARM='XREF, LIST',  
//  
//          REGION=96K  
//SYSLIB    DD  DSNNAME=SYS1.PL1LIB, DISP=SHR  
//SYSLMOD   DD  DSNNAME=&&GOSET(&A), DISP=(MOD, PASS),  
//          UNIT=SYSDA, SPACE=(1024, (50, 20, 1), RLSE)  
//SYSUT1    DD  DSNNAME=&&SYSUT1, UNIT=SYSDA,  
//          SEP=(SYSLMOD, SYSLIB), SPACE=(1024, (200, 20))
```



```
//SYSPRINT DD SYSOUT=A
//SYSLIN DD DDNAME=SYSIN
//GO EXEC PGM=*.LKED.SYSLMOD, COND= (9,LT,LKED)
//SYSPRINT DD SYSOUT=A
```

PL1LFCLG

```
//PL1LF PROG A=GO
//PL1L EXEC PGM=IEMAA, PARM='LOAD, NODECK',
// REGION=52K
//SYSPRINT DD SYSOUT=A
//SYSLIN DD DSNAME= & & LOADSET, DISP=(MOD, PASS),
// UNIT=SYSSQ, SPACE=(80, (250, 100)) ,
//SYUT3 DD DSNAME=&&SYUT3, UNIT=SYSSQ,
// SPACE=(80, (250, 250)), SEP=SYSPRINT
//SYUT1 DD DSNAME=&&SYUT1, UNIT=SYSDA,
// SPACE=(1024, (60, 60)), CONTIG),
// SEP=(SYUT3, SYSPRINT, SYSLIN)
//LKED EXEC PGM=IEWL, PARM='XREF, LIST',
// COND=(9, LT, PL1L), REGION=96K
//SYSLIB DD DSNAME=SYS1.PL1LIB, DISP=SHR
//SYSLMOD DD DSNAME=&&GOSET(&A),
// DISP=(MOD, PASS), UNIT=SYSDA,
// SPACE=(1024, (50, 20, 1), RLSE)
//SYUT1 DD DSNAME=&&SYUT1, UNIT=SYSDA.
// SEP=(SYSLMOD, SYSLIB), SPACE=(1024, (200, 20))
//SYSPRINT DD SYSOUT=A
//SYSLIN DD DSNAME=&&LOADSET, DISP=(OLD, DELETE)
// DD DDNAME=SYSIN
//GO EXEC PGM=*.LKED.SYSLMOD,
// COND=((9, LT, LKED),(9, LT, PL1L))
//SYSPRINT DD SYSOUT=A
//
```

PL1LFCG

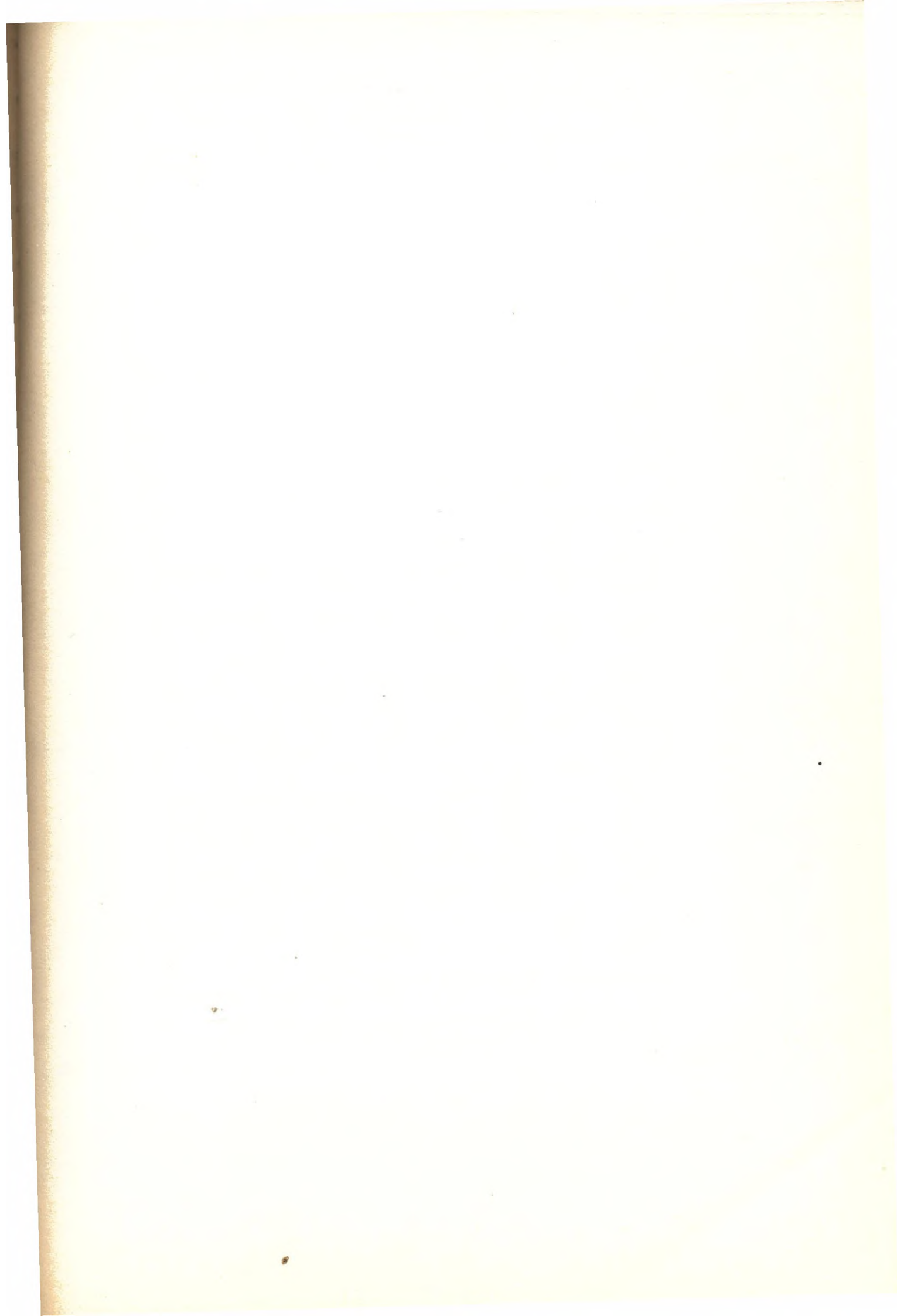
```
//PL1L EXEC PGM=IEMAA, PARM='LOAD, NODECK',
// REGION=52K
//SYSPRINT DD SYSOUT=A
//SYSLIN DD DSNAME= & & LOADSET, DISP=(MOD, PASS),
// UNIT=SYSSQ, SPACE=(80, (250, 100)) ,
//SYUT3 DD DSNAME=&&SYUT3, UNIT=SYSSQ,
// SPACE=(80, (250, 250)), SEP=SYSPRINT
//SYUT1 DD DSNAME=&&SYUT1, UNIT=SYSDA,
// SPACE=(1024, (60, 60)), CONTIG),
// SEP=(SYUT3, SYSPRINT, SYSLIN)
//GO EXEC PGM=LOADER, PARM='MAP, PRINT',
// REGION=96K, COND=(9, LT, PL1L)
```

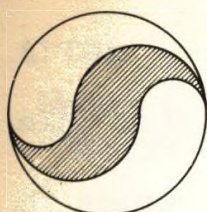


```
//SYSLIB DD DSNAME=SYS1.PL1LIB, DISP=SHR
//SYSLIN DD DSNAME= & & LOADSET, DISP=(OLD, DELETE)
//SYSLOUT DD SYSOUT=A
//SYSPRINT DD SYSOUT=A
```

PL1LFG

```
//GO EXEC PGM=LOADER, PARM='MAP, PRINT'
// REGION=96K
//SYSLIB DD DSNAME=SYS1.PL1LIB, DISP=SHR
//SYSOUT DD SYSOUT=A
//SYSPRINT DD SYSOUT=A
```





2. Függelék

A különböző eszköztípusok listája

a) ESZR

MÁGNESSZALAGOK

<i>Az eszköz típusa</i>	<i>— Az eszköz</i>	<i>A gyártó ország</i>
5012	Mágnesszalagos tár	BNK
5016	"	NDK
5017	"	SZU
5019	"	LNK
5022	"	CSSZSZK
5033	"	SZU
5035	"	LNK
5014	"	SZU
5021	"	NDK

KÖZVETLEN HOZZÁFÉRÉSŰ ESZKÖZÖK

<i>Az eszköz típusa</i>	<i>Az eszköz</i>	<i>A gyártó ország</i>
5050	Cserélhető mágneslemezes tár	SZU
5051	Fix mágneslemezes tár	SZU
5053	Mágneslemez-csomag	BNK
5052	Cserélhető mágneslemezes tár	BNK
5056	"	SZU
5058	"	CSSZSZK
5060	Fix mágneslemezes tár: (ESZ-1010-hez)	MNK
5055	Cserélhető mágneslemezes tár	NDK

EGYÉB ESZKÖZÖK

<i>Az eszköz típusa</i>	<i>Az eszköz</i>	<i>A gyártó országa</i>
6012	Lyukkártyaolvasó	SZU
6013	Lyukkártyaolvasó szabvány illesztőegységgel	SZU
6016	Lyukkártyaolvasó	CSSZSZK
6022	Lyukkártyaolvasó szabványos illesztőegységgel	MNK
6121	Lyukszalagolvasó	LNK
6122	"	LNK
6191	Lyukszalag-és lyukszalagkártya-olvasó	CSSZSZK
7010	Lyukkártyalyukasztó szabványos illesztőegységgel	SZU
7012	"	SZU
7022	"	SZU
7122	Lyukszalaglyukasztó	LNK
7192	Közepes sebességű lyukasztó	CSSZSZK
7030	Alfanumerikus nyomtató (szabv., ill. egys.)	SZU
7031	"	NDK
7033	Alfanumerikus sornyomtató	LNK
7035	"	NDK
7051	Sik-koordináta-rajzoló (szabv. ill. egys.)	SZU
7052	"	SZU
7054	Automatikus rajzgép	CSSZSZK
7902	Lyukszalag-állomás	NDK
7061	Alfanumerikus display (szabv. ill. egys.)	MNK
7063	"	MNK
7066	Display	SZU
7071	Elektromos írógép	CSSZSZK
7073	Írógép szabványos illesztő egységgel	NDK
7074	Írógép illesztő egységgel	BNK
7172	"CONSUL" elektromos írógép	CSSZSZK
7173	Elektromos írógép	NDK
7174	"	BNK
6022	Lyukszalagolvasó (szabványos, ill. egys.)	SZU
7032	Alfanumerikus sornyomtató (szabv. ill. egys.)	SZU
7034	Alfanumerikus sornyomtató	CSSZSZK
7038	"	CSSZSZK
7053	Planset-típusú grafikus regisztráló berendezés szabványos illesztéssel	SZU
7064	Alfanumerikus-grafikus display (szabv.)	SZU
7070	Konzolirógép szabványos illesztőegységgel	SZU

b) IBM

MÁGNESSZALAGOK

<i>Az eszköz típusa</i>	<i>Az eszköz</i>
2400	9 csatornás mágnesszalag. 800 bpi írássűrűség, ha a duplasűrűség nincs installálva vagy 1600 bpi írássűrűség, ha a duplasűrűség installálva van.
2400-1	Mágnesszalag, 7 csatornás kompatibilitással, adatkonverzió nélkül.
2400-2	Mágnesszalag, 7 csatornás kompatibilitással, adatkonverzióval.
2400-3	9 csatornás mágnesszalag, 1600 bpi írássűrűséggel.
2400-4	9 csatornás mágnesszalag, 800 bpi és 1600 bpi írássűrűséggel.

KÖZVETLEN HOZZÁFÉRÉSŰ ESZKÖZÖK

<i>Az eszköz típusa</i>	<i>Az eszköz</i>
2301	Mágnesdob
2302	Mágneslemez
2303	Mágnesdob
2311	Mágneslemez
2314	Mágneslemez
2321	Mágneskártya

EGYÉB ESZKÖZÖK

Az eszköz típusa

1052

1275

1285

1287

1288

1403

1419

1442

1443

2495

2501

2520

2540

2540-2

2671

3210

3215

Az eszköz

Konzolirógép

Optikai olvasó/rendező

Optikai olvasó

Optikai olvasó

Optikai olvasó

Sornyomtató

Mágneses olvasó

Kártyaolvasó/lyukasztó

Sornyomtató

Kazetta szalagolvasó

Kártyaolvasó

Kártyaolvasó/lyukasztó

Kártyaolvasó/lyukasztó

Kártyaolvasó/lyukasztó

Lyukszalagolvasó

Konzolirógép

Konzolirógép

GRAFIKUS ESZKÖZÖK

Az eszköz típusa

1053

2250-1

2250-3

2260-1

2260-2

2280

2282

Az eszköz

Sornyomtató

Display

Display

Display-állomás

Display-állomás

Filmregisztráló

Filmregisztráló/letapogató



3. Függelék

Példák teljes job-ra

a)

```
//P1 JOB 930400
//  EXEC  PGM=EP01
//STEPLIB DD DSN=ESZK.LOAD, DISP=SHR
//SYSPRINT DD SYSOUT=A
//OUT DD DSN=METO2, DISP=(MOD, KEEP), UNIT=2400,
//      VOL=SER=110026
//INP DD DSN=PARAM01, DISP=OLD
//SYSIN DD *
        adatkártyák
//
```

Megjegyzés: A PARAM01 állomány katalogizált, így csak a név és a diszpozíció megadásával elérhető.

b)

```
//ESZK6 JOB 873500, MSGLEVEL=(1,1)
//S1 EXEC PL1LFCLG
//PL1L.SYSIN DD *
        a fordítandó PL1 program kártyái
/*
//GO.INP DD *
        adatkártyák
//
```

Megjegyzés: A PL1LFCLG nevű katalogizált JCL eljárás (1. függelék) 3 eljárásstepből áll. Az eljárásstep nevei rendre: PL1L, LKED, GO. A fordítóprogram egy SYSIN nevű DD utasítással leírt állományból veszi a fordítandó programot. Ezt a SYSIN nevű DD utasítást (mivel a PL1LFCLG nevű eljárásban ilyen nincs) a felhasználónak kell megadni (2.11.5. pont).

c)

```
//J3 JOB
//  EXEC  PL1LFCLG
```



```

//PL1L.SYSIN DD *
P1: PROC OPTIONS(MAIN);
    DCL REG1 FILE RECORD KEYED ENV(REGIONAL(1)),
        INP FILE RECORD;
    DCL 1 INPREK BASED(P),
        2 ADAT CHAR(72),
        2 KULCS CHAR(8);
    DCL REKHOSSZ FIXED DEC(5);
    GET EDIT (REKHOSSZ)(F(5));
    IF REKHOSSZ < 1 |
        REKHOSSZ > 72 THEN
    DO;
        PUT SKIP EDIT (' PARAMETER HIBA REKHOSSZ=',
                        REKHOSSZ) (A, F(5));
        GOTO VEGE;
    END;
BEGIN;
    DCL MEZO CHAR(REKHOSSZ);
    ON ENDFILE(INP) GOTO STOP;
    ON KEY(REG1)
        BEGIN; I=ONCODE;
            PUT SKIP EDIT(INPREK,
                          ' ONCODE=', I) (A(72), A(8), X(3),
                          A, F(5));
            GOTO OLVAS;
        END;
    OPEN FILE(REG1) DIRECT OUTPUT,
        FILE(INP);
    OLVAS: READ FILE(INP) SET(P);
        MEZO=ADAT;
        WRITE FILE(REG1) FROM(MEZO) KEYFROM(KULCS);
        GOTO OLVAS;
    STOP: CLOSE FILE(REG1),
        FILE(INP);
    END;
    VEGE:END;
/*
//GO.REG1 DD DSN=DATAR1, DISP=(NEW, KEEP, DELETE),
            UNIT=2311, VOL=SER=333333, SPACE=(60, 1000),
            DCB=(RECFM=F, BLKSIZE=60, DSORG=DA)
//GO.SYSIN DD *
            60 paraméter kártya
/*
//GO.INP DD *
            adatkártyák
/*
//

```


Az egyes JCL utasítások szerepe

A J3 nevű JOB utasítás a job kezdetét jelöli és azonosítja a job-ot a J3 név segítségével.

Az EXEC utasítás aktivizálja a PL1LFCLG nevű JCL eljárást.

A PL1LFCLG nevű eljárás PL1L nevű lépésére vonatkozó SYSIN nevű DD utasítás a fordítandó program helyéül az input-áramot jelöli meg.

A sorrendben első/* utasítás input-árambeli adathalmaz (ez esetben a fordítandó PL1 program) végét jelzi.

A PL1LFCLG nevű JCL eljárás GO nevű lépésére vonatkozó DD utasítások:

a REG1 nevű DD a PL1 programban deklarált REG1 nevű file-hez tartozó DATAR1 nevű állomány paramétereit szolgáltatja,

a SYSIN nevű DD a PL1 programban használt SYSIN nevű file-hoz tartozó adatok helyéül az input-áramot adja meg,

az INP nevű DD a PL1 programban deklarált INP nevű file-hoz tartozó adatok helyéül szintén az input-áramot jelöli meg,

a sorrendben második, illetve harmadik /* utasítások input-árambeli adathalmaz (ez esetben a program működéséhez szükséges paraméterek, illetve adatok) végét jelzik.

A // utasítás a job végét jelenti.

Megjegyzés: OS PL1 program esetén a programbeli file neve és a JCL-beli DD utasítás neve megegyezik. Ez a név teremti meg a kapcsolatot a két szint között.

A példabeli PL1 program használja a REG1, INP, SYSIN és SYSPRINT nevű file-okat. (Utóbbi kettő a GET, illetve PUT utasítások által implicit módon van deklarálva.)

A REG1, SYSIN és INP nevű DD utasításokat a felhasználó közvetlenül, a SYSPRINT nevű DD utasítást a PL1LFCLG nevű JCL eljárásen keresztül adta meg. (A SYSPRINT nevű DD utasítás a PL1LFCLG eljárás GO nevű lépésében deklarálva van.)

Példaképpen tekintsük a REG1 nevű file kezelését.

A programon belül - a REG1-re vonatkozó filedeklarációban, illetve a reá vonatkozó OPEN-ben - azokat a paramétereket adtuk meg, amelyek csak a programban írhatók elő. A többi paraméter JCL szintre került. Ezáltal

és a fogadó MEZO dinamikus deklaráásával többféle rekordméret kezelése vált lehetővé. A példabeli J3 nevű job futtatásakor ez a rekordméret 60. A REG1 nevű DD és a SYSIN-beli paraméter értékének módosításával ez a rekordméret 1 és 72 között tetszőleges értékre beállítható.

```
d) //J4 JOB
    // EXEC FORTGCLG
    //FORT.SYSIN DD *
        DIMENSION X(1000), V(1000), FMT(20)
        .
        .
        READ(5,100)INDEX, NPERIF
100    FORMAT(I3,1X,I2)
        READ(5,101)FMT
101    FORMAT(20A4)
        .
        .
500    READ (NPERIF, FMT) (X(I), V(I), I=1, INDEX)
        .
        .
        END
    //GO.SYSIN DD *
    010 33
    2F10,5
    /*
    //GO.FT33F001 DD DSN=DATAF1, DISP=OLD
    //
```

Megjegyzés: OS FORTRAN program esetén a programbeli logikai egység-cím és a JCL-beli DD utasítás nevének 3. -4. pozícióin levő érték megegyezik. Ez az érték teremti meg a kapcsolatot a két szint között.

(A FORTGCLG JCL eljárásán belül az "5" logikai egység-cím a "SYSIN" névre van transzformálva.) A fenti FORTRAN program használja az "5" logikai egység-címet, valamint egy *tetszőleges* logikai egység-címet, melyet az "5" egység-címről olvas be és NPERIF-ben helyez el. A példabeli J4 nevű job futtatásakor NPERIF értéke 33, így az 500 címkéjű READ a "33" logikai egység-címről olvas. A SYSIN-beli paraméterek értékének és a DD utasításnak a módosításával elérhetjük, hogy a programbeli 500 címkéjű READ *futtatásonként* különböző állományok különböző formátumu adatait kezelje.



4. Függelék

A felhasználói hibakódok és jelentésük

A hibakód

A hiba

IEF127I	Uj állományhoz nem adtunk meg SPACE paramétert vagy megadtunk, de zérus területtel.
IEF128I	Uj indexszekvenciális állományhoz tartozó INDEX területre vonatkozó kérésünk hibás.
IEF129I	Uj indexszekvenciális állomány részére többköteses indexterületet kértünk.
IEF130I	Ideiglenes indexszekvenciális állomány esetén a DD utasítás DSNNAME paraméterének értéke hibás. Az állománynév után nem az INDEX, OVFLOW vagy PRIME elemek valamelyike áll.
IEF131I	Uj indexszekvenciális állomány esetén többköteses tulcsordulás-területet kértünk. A UNIT paraméterben 1-nél több egység, ezenkívül DSNNAME=név (OVFLOW) kérés szerepel.
IEF132I	Az indexszekvenciális állományt meghatározó DD utasítások között ellentmondás van. Az egyik a SPACE paraméter CYL változatát, a másik az ABSTR változatát használja.
IEF133I	Indexszekvenciális állomány esetén az egyik DD utasítás SPACE paraméterében CONTIG kérés van, a másik DD utasításban nincs.
IEF134I	Uj indexszekvenciális állományt meghatározó DD utasítás SPACE paraméterében érvénytelen CYL vagy ABSTR alparaméter van.
IEF135I	Indexszekvenciális állomány esetén az elsődleges területet meghatározó alparaméter értéke zérus.
IEF136I	Az indexszekvenciális állomány részére duplán jelöltünk ki INDEX területet.
IEF137I	A DD utasítás SUBALLOC paraméterében megnevezett állomány nem létezik.

IEF138I	A DD utasítás SUBALLOC paraméterében megnevezett állomány nem összefüggő.
IEF139I	A DD utasítás SUBALLOC paraméterében megnevezett állomány PASSWORD alatt áll.
IEF140I	Uj könyvtári állomány tartalomjegyzéke (DIRECTORY) számára a DD utasítás SPACE paraméterében kijelölt hely nagyobb, mint amennyi a köteten rendelkezésre áll.
IEF141I	Az indexszekvenciális állományt definiáló INDEX és PRIME DD utasítások sorrendje nem megfelelő.
IEF143I	Indexszekvenciális állomány esetén az utolsó láncoló (CONCATENATION) DD utasítás szűkségtelen vagy érvénytelen.
IEF144I	A DD utasítás SUBALLOC paramétere osztott cylinder (SPLIT) állományra hivatkozik.
IEF146I	Azonos nevű állományok között kívánjuk egy kötet cylindereit osztani (SPLIT).
IEF164I	Egy lépésen belül egyik DD VOL=REF vagy állománynév megadásakor hivatkozik egy másik DD-re, vagy egy lépésen belül több DD utasítás DSNAMES paraméterének ugyanaz az értéke.
IEF208I	JOB vagy EXEC utasításban a REGION paraméter nagyobb területet definiál, mint a központi tár.
IEF209I	Ujraindításkor (CHECKPOINT RESTART) nem áll rendelkezésre elegendő hely a központi tárban.
IEF210I	DD utasítás UNIT paraméterében érvénytelen egységnevet adtunk meg.
IEF211I	Katalogizált állomány esetén a rendszerkatalógus (SYSCTLG) nem áll rendelkezésre.
IEF212I ddn-	A "ddn" nevű DD utasításban leírt állomány nem létezik.
IEF213I ddn-	A "ddn" nevű DD utasítás DSNAMES paramétere nem állományt határoz meg, hanem indexblokkot.
IEF214I ddn-	A "ddn" nevű DD utasítás DSNAMES paraméterében megadott név érvénytelen vagy a rendszerkatalógusból hiányzik a megfelelő szint.
IEF215I ddn-	A "ddn" nevű DD utasítás DSNAMES paramétere formai hibát tartalmaz.

IEF216I ddn-	A "ddn" nevű DD utasításban leírt katalogizált állomány indexe érvénytelen mutatót tartalmaz vagy I/O hiba lépett fel az index olvasása közben.
IEF217I ddn-	A "ddn" nevű DD utasításban leírt állományt tartalmazó kötet nem állt rendelkezésre a futás alatt.
IEF218I ddn-	A "ddn" nevű DD utasításban megadott nevű állomány nincs a VTOC-ban.
IEF219I ddn-	A "ddn" nevű DD utasítás DSNAME paraméterében megadott csoportnév 35 karakternél hosszabb.
IEF220I ddn-.	A "ddn" nevű DD utasítás DSNAME paraméterében megadott csoportnévvel definiált állományok, valamint a további állományok együttesen túllépik az egy lépésben megengedett maximum 254 állományszámot.
IEF221I PGM=*.DD -	Az a megelőző lépés, amely létrehozta volna az itt végrehajtandó programot, nem futott, így nincs mit végrehajtani.
IEF229I	A logikai rekordhossz nagyobb, mint 32760 byte.
IEF230I	A DD utasítás UNIT paraméterében a SEP paraméteren 8-nál több DD utasítás nevét adtuk meg.
IEF231I	I/O készülék hiba miatt az allokálás lehetetlen.
IEF232I jjj-	A "jjj" nevű job számára a SYS1.SYSJOBQE nevű állományban már nincs hely.
IEF240I	A job-lépés túl sok DD utasítást tartalmaz.
IEF244I jjj sss ddn-	A "jjj" nevű job "sss" nevű lépésében lévő "ddn" nevű DD utasításban leírt állományt nem tudja allokálni. A UNIT paraméterben megadott név hibás vagy a megnevezett készülék nem osztható.
IEF245I	Egy DD utasításban megadott UNIT paraméter és VOLUME paraméter nincs összhangban.
IEF246I	Egy DD utasítás SPACE paraméterében kért hely nagyobb, mint amennyi a készüléken rendelkezésre áll.
IEF252I	A SYS1.SYSJOBQE kicsiny a job befogadására.
IEF253I	Duplanév a kötetben. Egy DD utasítás DSNAME paraméterében új állomány azonosítására olyan nevet adtunk meg, amely már előfordul a VTOC-ban.
IEF254I	Nincs már hely új állomány bejegyzésére a VTOC-ban.

IEF255I	Program betöltésekor, a használt közvetlen hozzáférésű készüléken permanens input/output hiba lépett fel.
IEF256I	Egy DD utasítás SPACE paraméterében kért konkrét sáv nem áll rendelkezésre.
IEF257I	Egy DD utasítás SPACE paraméterében kért hely nagyobb, mint amennyi a közvetlen hozzáférésű készüléken rendelkezésre áll.
IEF258I	Egy DD utasítás SPACE paraméterében megadott átlagos rekordhossz nagyobb, mint a közvetlen hozzáférésű készülék egy sávja.
IEF261I	Egy indexszekvenciális állományt definiáló DD utasításban nincs megadva PRIME terület. Egy indexszekvenciális állomány esetén a tulcsordulás-területet definiáló DD utasítás sorrendben megelőzi a PRIME területet definiáló DD utasítást.
IEF263I	Egy indexszekvenciális állományt leíró DD utasítás SPACE paramétere hibás. A SPACE paraméter ABSTR változatát használtuk és a megadott konkrét cím mégsem esik cylinder-határra vagy a megadott terület mégsem egész számú többszöröse egy cylindernek.
IEF264I	Két DD utasítás ugyanazt az indexszekvenciális állományt, ugyanazzal a területtel, ugyanazzal a DSNAMÉ értékkel definiálja.
IEF265I	Egy DD utasítás SPLIT paraméterében megadott sávok száma nagyobb, mint a készülék egy cylinderén lévő sávok száma.
IEF267I	Egy új könyvtári állományt definiáló DD utasítás SPACE vagy SUBALLOC paraméterében a tartalomjegyzék (DIRECTORY) számára igényelt terület nagyobb, mint az elsődleges terület vagy nagyobb, mint a köteten rendelkezésre álló hely.
IEF272I	A vezérlőprogram felhasználói hiba miatt nem hajtotta végre a job-lépést.
IEF273I	Egy új állományt definiáló DD utasítás DCB paraméterének DSORG alparamétere a PO vagy az IS érték, avagy a SPACE paraméter nem tartalmazza a tartalomjegyzék (DIRECTORY) alparamétert.
IEF286I	Egy DD utasítás DISP paramétere nincs összhangban az állomány állapotával.

IEF288I	Helytelen vagy hiányzó ddnév miatt az operációs rendszer nem tudja megnyitni a mágnesszalagos állományt.
IEF294I	Egy DD utasítás SYSOUT paraméterében 8-nál több output-osztály van megadva.
IEF296I	Egy job-lépésben megadott egységek száma túl nagy. (DD utasítások UNIT paraméterei)
IEF297I	A DD utasítás DCB paraméterének LRECL alparamétere 32756-nál nagyobb értéket ír elő.
IEF312I	A JCL eljáráskönyvtár definiálásakor megadott blokkméret nem osztható 80-nal.
IEF315I	A DD utasítás UNIT paraméterében definiált készülékek száma nagyobb, mint a rendelkezésre álló készülékek száma vagy nagyobb, mint a VOL paraméterben megadott kötetek száma.
IEF316I	A kiíró program (WRITER) érvénytelen vezérlő jelet talált a kiírandó állományban.
IEF318I ddn-	A "ddn" nevű DD utasítás UNIT paraméterének AFF alparaméterében érvénytelen affinitást írtunk elő.
IEF320I	Az állomány logikai rekordja számára nincs elég hely a központi tárban.
IEF321I	Input-állományban érvénytelen szegmens fordult elő.
IEF335I	A SYS1.SYSJOBQE állományban kevés hely van a job vezérlő utasításainak befogadására.
IEF336I	Az input-sorban nincs hely a job befogadására.
IEF372I ddn-	A "ddn" nevű DD utasítás VOLUME paramétere hivatkozást tartalmaz egy korábbi lépésre, de ez a korábbi lépés nem hajtott végre a COND paraméterben előírt feltételek miatt.
IEF388I jjj.sss.ddn-	A "jjj" nevű job "sss" nevű lépésében lévő "ddn" nevű DD utasításban definiált állomány allokálásához szükséges készülék nem áll rendelkezésre.
IEF389I jjj.sss.ddn-	A "jjj" nevű job, "sss" nevű lépésében lévő "ddn" nevű DD utasításban előírt szeparáció nem teljesíthető; figyelmen kívül marad.
IEF406I ddd-	A beolvasóprogramhoz (READER) tartozó JCL eljárás IEF389I nevű DD utasítása hibás, ezért a SYSIN állományt a "ddd" készüléken nem tudja megnyitni.

IEF407I ddd-	A beolvasóprogramhoz (READER) tartozó JCL eljárás IEF PDSI nevű DD utasítása hibás, ezért a SYS1. PROCLIB nevű könyvtárt nem tudja megnyitni.
IEF416I	Egy DD* vagy DD DATA utasítással definiált állomány elhelyezésére a direkt hozzáférésű készülőken nincs már hely,
IEF452I xxx	A beolvasóprogram (READER) formai hibát talált az xxx nevű job-ban. Ha a hiba a JOB utasításban van, akkor xxx helyén a JOBFAIL üzenet jelenik meg.
IEF453I jjj	Az ütemezőprogram (INITIATOR) vagy a beolvasóprogram (READER) hibát talált a job-ban.
IEF503I	Rossz írássűrűség vagy helytelen címkeblokk a kötetben.
IEF594I	Egy DD utasítás SPLIT paramétere érvénytelen. Egynél több kötetet írtak elő, ami SPLIT esetén nem megengedett vagy a SPLIT területeket definiáló DD utasítások sorrendje hibás.
IEF601I	Egy JCL eljárásban meg nem engedett utasítás fordul elő. A meg nem engedett utasítások a következők: JOB utasítás, nem // jellel kezdődő utasítás, DD * vagy DD DATA, // vagy /* utasítás.
IEF605I	Egy JCL utasításban, az utasításmezőben nincs érvényes utasításkulcsszó.
IEF606I	A JOB utasítás és az első EXEC utasítás között JOBLIB-től különböző nevű utasítás áll.
IEF607I	A job nem tartalmaz EXEC utasítást.
IEF609I	EXEC utasításban hibás paraméterkulcsszó van.
IEF610I	Egy JCL eljárás nem tartalmaz EXEC utasítást.
IEF611I	EXEC vagy DD utasításban, amellyel egy JCL eljárás valamely EXEC vagy DD utasítását kívántuk felülírni, olyan lépés nevét adtuk meg, amely nincs az eljárásban.
IEF612I	Az EXEC utasításban megadott JCL eljárást sem a SYS1. PROCLIB-ben sem az input job-ban nem találja.
IEF613I	JCL eljárás további JCL eljárást tartalmaz.
IEF615I	EXEC utasításban 8-nál hosszabb lépésnevet adtunk meg.

IEF616I	JCL utasítás alparaméter-listája újabb alparaméter-listát tartalmaz.
IEF617I	Az EXEC utasítás után következő első DD utasításnak nincs neve.
IEF618I	Egy JCL utasításban az operandus mező nincs határolva vesszővel vagy szóközzel.
IEF619I	A beolvasóprogramhoz (READER) tartozó JCL eljárásban nincs IEFDATA nevű utasítás.
IEF621I	Egy JCL utasítás valamely paramétere vesszővel van határolva vagy a 72. pozíción szóköztől különböző jel van, de a következő sor mégsem folytatás-sor.
IEF622I	Egy JCL utasítás zárójelezése hibás.
IEF623I	A JCL utasítás meg nem engedett karaktert tartalmaz.
IEF625I	Egy JCL utasításban a baloldali zárójelet szabálytalanul használtuk.
IEF626I	A + jel meg nem engedett használata.
IEF627I	A & jel meg nem engedett használata.
IEF628I	A * jel meg nem engedett használata.
IEF629I	Az aposztróf meg nem engedett használata.
IEF630I	A JCL utasítás azonosíthatatlan kulcsszót tartalmaz.
IEF631I	Egy lépésen belül 5-nél több DD utasításban használtuk a DDNAME paramétert.
IEF632I	Formai hiba a JCL utasításban.
IEF633I	A programozó neve hiányzik a JOB utasításból. (Csak akkor, ha a programozó nevének megadását a rendszer generálásakor előírtuk.)
IEF634I	A JOB utasítás nem tartalmaz elszámolási információt. (Csak akkor, ha az elszámolási információ megadását a rendszer generálásakor előírtuk.)
IEF635I	A JOB-nak nincs neve.
IEF636I	A JOBLIB nevű DD utasítás nem a JOB utasítás és az első EXEC utasítás között áll.
IEF637I	A JOB vagy EXEC utasításban megadott elszámolási információ 142 jelnél hosszabb.

IEF638I	Egy JCL utasításban megadott numerikus paraméter- vagy alparaméter-érték kívül esik a megengedett határokon.
IEF640I	Egy JCL utasítás a megengedettnél több pozicionális paramétert tartalmaz.
IEF641I	Egy JCL utasításban megadott pozicionális paramétersorozat hibás.
IEF642I	Egy JCL utasításban megadott pozicionális paraméter hosszabb a megengedettnél.
IEF643I	Egy JCL utasításban azonosíthatatlan pozicionális paraméterérték van.
IEF644I	Egy JCL utasítás numerikus paramétere nem numerikus jelet tartalmaz.
IEF645I	Egy JCL utasítás olyan "korábbi" utasításra hivatkozik, amely nem létezik.
IEF646I	Egy JCL utasításból hiányzik egy pozicionális paraméter vagy alparaméter.
IEF647I	Egy JCL utasítás neve nem alfabetikus jellel kezdődik.
IEF648I	Egy DD utasítás DISP paramétere hibás: ideiglenes állományra KEEP vagy nem katalogizálható állományra CATLG van megadva.
IEF649I	Egy lépésben 255-nél több DD utasítás fordul elő.
IEF650I	Egy JCL utasításban a / jelet meg nem engedett helyen használtuk.
IEF651I	A mínusz jel meg nem engedett használata.
IEF654I	Egy lépésben belül két DD utasításbeli DDNAME paraméterrel ugyanarra a névre hivatkoztunk.
IEF655I	Egy DD utasításon belül SYSOUT mellett DSNAME paraméter is előfordul.
IEF657I	SYSCHK nevű DD utasítást megadtunk, de a JOB utasítás RESTART paraméterében kópia-azonosító nincs megadva.
IEF660I	A JOB utasítás RESTART paraméterében kópia-azonosítót megadtunk, de a SYSCHK nevű DD utasítás hiányzik.
IEF661I	A JOB utasítás RESTART paraméterében megadott lépésnév nem létezik.

- IEF662I A PROC vagy PEND utasításnév mezejében álló
 név túl hosszú vagy meg nem engedett jelet tar-
 talmaz.
- IEF663I Az input-áramban lévő PROC utasításnak nincs
 neve.
- IEF665I A job 15-nél több eljárásdeklarációt tartalmaz.
- IEF670I Egy JCL eljárás szimbolikus paramétere sem az
 eljárásdeklarációban alapértelmezést sem az el-
 járást hívó EXEC utasításban aktuális értéket
 nem kapott; így definiálatlan.



Irodalomjegyzék

H. Á. RAMDÉN: JCL and advanced FORTRAN Programming
ELSEVIER SCIENTIFIC PUBLISHING COMPANY
Amsterdam - Oxford - New York, 1976.

Országos Számítógéptechnikai Vállalat:
Egységes Számítógép Rendszer (ESZR), Statisztikai Kiadó Vállalat, Budapest

VEB Robotron: OS/ES Jobsteuersprache

IBM System/360 Operating System:
Job Control Language Reference

Dr. Quittner Pál: Bevezetés az operációs rendszerekbe.

Szógyűjtemény

Magyar	Angol	Német	Orosz
adatállomány, állomány	data set	Datei	набор данных
adatállomány leíró blokk (DSCB)	data set control block (DSCB)	Datei-Kennsatzblock	блок управления набором данных
adat vezérlő blokk (DCB)	data control block (DCB)	Datei-Steuerblock	блок управления данными
allokálás	allocation	Zuordnung	размещение, распределение
alparaméter	subparameter	Unterparameter	подпараметр
alulcsordulás	underflow	Unterlauf	исчезновение порядка
átadott adatállomány	passed data set	Durchlassdatei	переданный набор данных
belépési pont	entry point	Eingangstelle	точка входа
beolvasó/értelmező program	READER/INTERPRETER	Leser/Interpreter	считывающая, интерпретирующая программа
betöltő program	Loader	Lader	загрузчик
bit	bit	Bit	бит, двоичный разряд
blokkméret, blokkhossz	block size	Blocklänge	размер блока, длина блока
byte	byte	Byte	байт

Magyar	Angol	Német	Orosz
címke	label	Kennsatz	метка
programszakasz (CSECT,)	control section (CSECT)	Programmabschnitt	программная секция
ddnév	ddname, dd-name	DD-name	дд - имя
DD-utasítás	DD-statement	DD-Anweisung	ДД - предложение
definiálatlan hosszú rekord	undefined length record	undefiniertes Satzlänge	запись неопределенной длины
dummy programszakasz	dummy section	Pseudoabschnitt	фиктивная секция
egység	unit	Einheit	устройство
eljárás	procedure	Prozedur, Verfahren	процедура
elkülönítés, szeparálás	separation	Trennung	отделение
exkluzív, kizáró	exclusive	exklusive	исключающее
feladat, task	task	Aufgabe, Task	задача
feltétel kód	condition code	Bedingungsschlüssel	признак результата
file	file	Datei	файл
file vége	end of file	Dateiende	конец файла
fix hosszú rekord	fixed length record	feste Satzlänge	запись постоянной длины
fordító	compiler	Übersetzer	компилятор, транслятор

Magyar	Angol	Német	Orosz
generálás	generation	Generation	генерация, генерирование
gépkezelő, operátor	operator	Maschinenbediener	оператор
határoló	delimiter	Begrenzungszeichen	ограничитель
határprioritás	limit priority	Grenz-Vorrang	предел, граница приоритета
helyettesítő sáv	alternate track	Ersatzspur	запасная дорожка
hierarchia	hierarchy	Rangordnung	иерархия
hivatkozás	reference	Bezugnahme	ссылка
ideiglenes	temporary	zwischen-	временный
ideiglenes tároló terület.	temporary storage	Zwischenspeicher	рабочая ячейка
időosztás (os)	time sharing	zeitl. verzahnte Verarbeitung	разделение времени
index szekvenciális állomány	indexed sequential data set	index sequentielle Datei	индексно-последовательный набор данных
inicializálás	initialization	Initialisierung	загрузка начальной программы
input áram	stream input	Reihenweise Eingabe	входной поток
input modul	input modul	Eingabe Modul	входной модуль
input várakozási sor	input queue	Eingabe-Warteschlange	очередь ввода
JCL, job vezérlő nyelv	job control language	Jobsteuersprache	язык управления заданиями
job, munka	job	Job	задание, работа

Magyar	Angol	Német	Orosz
karakter	character	Zeichen	символ
katalógus	catalog	Katalog	каталог
képernyő	display	Anzeige	дисплей, экран
készülék	device	Einrichtung	устройство
kiíró program	WRITER	Schreiber	программа вывода /данных/
kimentés	unload	entladen	запасение
konvertálás	conversion	Umsatzung	преобразование
könyvtár	library	Bibliothek	библиотека
kötet	volume	Datenträger	том
kötetnév	volume serial number	Datenträger-Nahme	серийный номер тома
kötet tartalomjegyzék (VTOC)	volume table of contents (VTOC)	Datenträger-Inhalt-verzeichnis	оглавление тома
központi tár (oló), főtár	main storage	Hauptspeicher	внутренняя память;
központi egység, CPU	central processing unit	Zentraleinheit	центральный процессор
közvetlen hozzáférés(ü)	direct access	direkter Zugriff	прямой доступ
kulcsszavas para-méter	keyword paraméter	Schlüsselwortparameter	параметр с ключевым словом
lépés, step	step	Step	шаг; фаза
lista	list	Liste	список
load modul	load module	Lademodul	загрузочный модуль

Magyar	Angol	Német	Orosz
makró	macro	Makro-Instruktion	макро /команда/
megjegyzés	comment	Bemerkung	комментарий, замечание
member, tag	member	Glied einer unter- gliederten Datei	член
minősített név	qualified name	gekennzeichnete Name	составное имя
módosítás, karban- tartás	update	Änderung	корректирование, обновление
munkakötet	scratch volume	Arbeitsdatenträger	ликвидированный том
mutató	pointer	Hinweisadresse	указатель
műveleti jel	operator	Operationssymbol	знак операции
műveleti kód	operation code	Befehlscode	код операции
műveleti mező	operation field	Befehlsfeld	поле операции
névmódosítás	rename	Datenneubenennung	переименование
operációs rendszer	operating system	Betriebssystem	операционная система
opció	option	Angabe, Option, Zusatz	возможность
osztott	split	aufspalten, splitten	разделенный
osztott készülék	shared device	gemeinsame Einheitbenutzung	разделенное устройство
output áram	stream output	Reihenweise Ausgabe	выходной поток
output várakozási sor	output work queue	Ausgabe-Warteschlange	очередь вывода

Magyar	Angol	Német	Orosz
összeválogatás	merge	ableichen, mischen	программа слияния, программа объединения
parancs	command	Befehlinstruktion	команда; инструкция
partíció	partition	Unterglied	разделение
particionált állomány (könyvtári állomány)	partitioned data set	untergliederte Datei	библиотечный набор данных
pozíció	position	Stellung	позиция
pozicionális para- méter	positional para- meter	Stellungsparameter	позиционный параметр
prioritás	priority	Priorität, Vorrang	приоритет
privát	private	privat	личный
programszakasz, CSECT	control section	Programmabschnitt	программная секция
rekord	record	Satz, Datensatz	запись
rekordméret, - hossz	record size	Satzlänge	размер записи, длина записи
rendezés	sort	sortieren	сортировка
rendszer	system	System	система
rezidens	residence	residenz	резидент /-ное запоминающее устройство/
régió	region	Region	область, сфера

Magyar	Angol	Német	Orosz
sáv	track	Spur	дорожка
segédprogram	utility program	Dienstprogramm	вспомогательная программа
szalagjel	tape mark	Bandmarke, Abschnitts- marke	метка, маркер ленты
szegmens	segment	Abschnitt	сегмент
szerkesztő program	Linkage Editor	Linkage Editor	редактор связей
szóköz	blank	Zwischenraum	пробел
szöveg	text	Text	текст
tartalomjegyzék	directory	Verzeichnis	справочник
tárgymodul	object module	Objektmodul	объектный модуль
tároló	storage	Speicher	память, запоминающее устройство
tár kivonat	dump	Speicherauszug	дампинг, выдача состояния памяти
terület (üres)	space	Leerspalte	пространство, область, пробел
tulcsordulás	overflow	Überlauf	переполнение
ujraindítás	restart	Wiederaanlauf	повторный запуск
üzenet	message	Nachricht	сообщение

Magyar	Angol	Német	Orosz
változó hosszú rekord	variable length record	variable Satzlänge	запись переменной длины
vezérlőprogram, szupervizor	supervisor	Überwacher, Supervisor	супервизор
végrehajtási prioritás	dispatching priority	Zuteilungs-Vorrang	управляющий приоритет
visszatérési kód	return code	Rückkehr-Code	код возврата.
visszatöltés	restore	au' Grundstellung ingen	восстановление
// utasítás	Null statement	Leer-Anweisung	предложение



950

A kiadásért felel: Kecskés József igazgató
Kiadói szerkesztő: Oláh Gyuláné
Műszaki szerkesztő: Böhm Oszkár
Borítóterv: Bogár Imre
Példányszám: 2000
Készült az MSZ 5601-59 és 5602-55 szerint
Terjedelem: 11,6 (A/5) ív; formátum: B/5
Statisztikai Kiadó Vállalat, Nyomdaüzem K-77-0619



44,— Ft

A
„SZÁMÍTÁSTECHNIKAI
SOROZAT”

kötetei közül

korlátozott példányszámban még kapható:

7. kötet

**DR. H. MERTEN:
AZ ADATBANKSZERVEZÉS KÉRDÉSEI**

A könyv segítséget nyújt az adatbankok létrehozásában és használatában érdekelt valamennyi szakember munkájának megtervezéséhez.

Ismerteti az adatszervezés és információfeldolgozás alapjait, részletesen tárgyalja a tárolási, kezelési és feldolgozási eljárásokat, külön figyelmet fordítva az adatbankkal való kommunikáció megszervezésére. Részletes javaslatot dolgoz ki az adatbankrendszerek kialakítására, a teljes rendszer megszervezésére és bevezetésére.

Ára: 55,— Ft

A kiadvány megvásárolható:

STATISZTIKAI KIADÓ VÁLLALAT
STATISZTIKAI ÉS SZÁMÍTÁS-
TECHNIKAI KÖNYVESBOLT
1024 Budapest, II., Keleti Károly u. 10.
Telefon: 158-018

Postai szállításra megrendelhető:

STATISZTIKAI KIADÓ VÁLLALAT
KÖZPONTI TERJESZTÉS
1300 Budapest 3. Pf. 99.
1033 Budapest, III.,
Kaszás u. 10—12.

