

ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

DEPARTAMENTO DE ENGENHARIA DE ELETRICIDADE

LABORATORIO DE SISTEMAS DIGITAIS

SISTEMA CROSS-CARREGADOR E BIBLIOTECA

Maria Alice G. V. Ferreira

1973

SINOPSE

O presente trabalho contém uma descrição detalhada do sistema Cross-Carregador e da Biblioteca desenvolvidos para o computador Pato Feio. Os presentes programas foram escritos para um computador IBM-1130, com 16 K palavras e disco.

Sua finalidade é integrar com o sistema Cross-Monyador um sistema de compilação e montagem de programas para aquele computador de controle. Os programas após a fase de depuração e composição podem ser armazenados na Biblioteca na forma de programas objetos. A função da Biblioteca é também a de armazenar módulos de programas na forma relocável, os quais são compostos pelo Cross-Carregador.

Um programa uma vez 'debugado' e já composto na forma de programa objeto absoluto pode ser retirado a fim de ser incorporado a Biblioteca final do computador de controle, através de cartões perfurados.

Como o computador de controle não dispõe de entrada e através de cartões perfurados, estes devem ser transformados em fitas perfuradas através de um programa reconfigurador, escrito para o computador HP-2116-B.

Nesta publicação estão descritos cada um dos programas integrantes do sistema Cross-Carregador/Biblioteca e tem a finalidade de constituir o manual de sistema deste conjunto de programas.

INDICE

Sistema de Biblioteca.....	001
WZW.....	002
WZBIG.....	003
WZBIB.....	008
WZINC.....	010
WZEXC.....	017
WZPAC.....	022
WZIND.....	025
WZCAR.....	028
WZRT.....	032
WZLSI.....	038
WZPG/WZBLS.....	042
 Sistema Cross-Carregador.....	046
WZV.....	047
WZLOA.....	052
WZVER.....	062
WZENC.....	066
WZDLE.....	069
WZAN.....	069
WZZ.....	071
WZSME.....	073
WZCOL.....	076
WZLOC.....	078
WZBC.....	081
WZWRT.....	083
WZDS.....	085
WZMAP.....	087
WZGRA.....	090
WZTB.....	093
WZPOE.....	095
WZCAL.....	097
WZSDS.....	101
WZTER.....	103
 Rotinas gerais.....	108
WZREL.....	109
WZBUS.....	112

WZBB.....	115
WZC1.....-	118
WZBH.....	120
WZDCM.....	123

001

SISTEMA DE BIBLIOTECA

WZW
WZBIG
WZBIB
WZINC
WZEXC
WZPAC
WZIND
WZCAR
WZRT
WZLEVWZLSI
WZLSI
WZBLS
WZPG

002

NOME DO PROGRAMA

Programa de biblioteca

CÓDIGO DO PROGRAMA

WZW

LINGUAGEM USADA

Fortran

TIPO DE PROGRAMA

Programa principal

SISTEMA SUPORTE E HARDWARE

IBM-1130

CHAVES DE PAINEL E SAÍDAS ALTERNATIVAS

Nenhuma

DESCRIÇÃO

Este programa trata todos os comandos permitidos pelo sistema de biblioteca, usando para isso a rotina WZBIG. (ver diagramas de blocos e cartões de controle anexos)

SEQUÊNCIA DE CHAMADA

// XEQ WZW 1

*FILES(1,N08),(11,WZN11),(12,BBB12),(13,BBB13)

facultativo

SUB-ROTINAS USADAS

WZBIG

OBSERVAÇÕES E MENSAGENS DE ERRO

Nenhuma

VARIÁVEIS PRINCIPAIS

Nenhuma

NOME DO PROGRAMA

Rotina de gerência da biblioteca

CÓDIGO DO PROGRAMA

WZBIG

003

LINGUAGEM USADA

Fortran

TIPO DE PROGRAMA

SUBrotina

SISTEMA SUPORTE E HARDWARE

IBM-1130

CHAVES DO PAINEL E SAÍDAS ALTERNATIVAS

Nenhuma

DESCRIÇÃO

Este programa faz o gerenciamento do sistema de biblioteca.

São funções desta rotina:

- ler um cartão de comando e identificá-lo.
- executar as funções pedidas, desviando à rotina tratadora de cada caso.
- determinar a validade de cada comando, imprimindo o comando, e uma mensagem de erro, no caso de não ser válido.
- preparar a chamada das rotinas tratadoras de cada caso específico.
- ler novo comando ao fim de cada execução, até encontrar o cartão de ~~FIM~~ FIM.

SEQUENCIA DE CHAMADA

CALL WZBIG

SUB-ROTINAS USADAS

WZPG

WZBL

WZCL

WZBIB

WZINC

WZEXC

WZBB

WZPAC

WZIND

WZCAR

WZRT

WZLSI

OBSERVAÇÕES E MENSAGENS DE ERRO

Nenhuma

VARIAVEIS PRINCIPAIS

• 004

Nome	Tipo	Função
IND	I,C,()	contém a diretoria da biblioteca.
IBUF	I,C,()	buffer de leitura de disco de um setor.
COM	R,()	contém os comandos válidos para a biblioteca.
IREST	I,()	buffer de cartão (complemento).
IS	I	caracter "X".
FC	R	comando a tratar.
FB	I,(),C	nome da biblioteca.
IP	I,C,()	nome do programa.
I	I	ponteiro para o buffer IREST.
M	I	indica se diferente de zero, a ocorrência de erro.
N	I	número convertido a binário.
IA	I	caracter "A" - indica programa absoluto.

PAGE 4

005

```
// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
SUBROUTINE WZBIG
INTEGER FB(3)
DIMENSION IND(20),IBUF(320),COM(10),IREST(60),IP(5)
COMMON ITJDO(1430)
EQUIVALENCE (ITUDO(21),IBUF(1)),(ITUDO(1),IND(1))
EQUIVALENCE (IND(1),IP(1)),(IND(6),FB(1))
DATA IA/'A'/
DATA IPRIN/5/,IAST/**/
DATA COM/'BIBL','INCL','EXCL','COMP','EXTS','ASM','LOAD','FIT',
1  'FIM ','LIST'/
IPAG=1
CALL WZPG(IPAG)
10 READ(2,1) IS,FC,FB,IP,IREST
1  FORMAT(A1,A4,2X,3A2,2X,5A1,60A1)
WRITE(IPRIN,2) IS,FC,FB,IP,IREST
2  FORMAT(1H0,5X,A1,A4,2X,3A2,2X,65A1)

C
C      IF(IS-IAST) 20,30,20
20 WRITE(IPRIN,3)
3  FORMAT(1H0,5X,'COMANDO INVALIDO')
CALL EXIT

C      DESCOBRE O COMANDO
C
30 DO 40 I=1,10
IF(FC-COM(I)) 40,50,40
40 CONTINUE
GO TO 20

C
C      50 GO TO(100,200,300,400,500,600,700,800,900,1000),I
C      COMANDO BIBL - GERA BIBLIOTECA
C
100 I1=1
I1=1
105 CALL WZB1(IREST,I,M)
IF(M) 10,108,10
108 CALL WZC1(IREST,I,N,M)
IF(M) 10,110,10
110 GO TO(111,112,113,114),I1
111 IND(1)=N
I1=2
GO TO 105
112 IND( 9)=N
I1=3
GO TO 105
113 IND(10)=N
114 CALL WZBIB
GO TO 10

C      INCLUSAO
```

200 I=1
CALL WZB1(IREST,I,M)
IF(M) 10,210,10
210 IF(IREST(I)-IA) 230,220,230
220 IND(9)=0
225 CALL WZINC(M)
GO TO 10
230 IND(9)=1
GO TO 225

C
C
C EXCLUSAO

300 CALL WZEXC
GO TO 10

C
C
C COMPACTACAO

400 CALL WZBB(M, IDISK)
IF(M) 900, 401,900
401 CALL WZPAC(M, IDISK,0)
GO TO 10

C
C
C EXTENCAO

500 CALL WZBB(M, IDISK)
IF(M) 900,505,900
505 CALL WZB1(IREST,I,M)
IF(M) 540,510,540
510 CALL WZC1(IREST,I,N,M)
IF(M) 900,520,900
520 DO 530 J=1,N
530 CALL WZIND(M, IDISK)
GO TO 10
540 N=1
GO TO 520

C
C
C ASSEMBLER

600 CALL EXIT

C
C
C LOADER

700 CALL WZCAR
RETURN

C
C
C FITA

800 CALL WZRT
RETURN

C
C
C FIM

900 RETURN

PAGE 5

1000 CALL WZBB(M, IDISK)
IF(M) 900,1001,900
1001 CALL WZLSI(IDISK, IPAG)
GO TO 10
END

007

FEATURES SUPPORTED
ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZBIG
COMMON 1430 VARIABLES 94 PROGRAM 342

RELATIVE ENTRY POINT ADDRESS IS 0087 (HEX)

END OF COMPIRATION

// DJP

*DELETE WZBIG 0010
CART ID 0010 DB ADDR 018A DB CNT 0018

*STORE WS U4 WZBIG 0010
CART ID 0010 DB ADDR 1973 DB CNT 0019

// EJECT

NOME DO PROGRAMA

Rotina de geração da biblioteca

008

CÓDIGO DO PROGRAMA

WZBIB

LINGUAGEM USADA

Fortran

TIPO DE PROGRAMA

Subrotina

SISTEMA SUPORTE E HARDWARE

IBM-1130

CHAVES DE PAINEL E SAÍDAS ALTERNATIVAS

Nenhuma

DESCRIÇÃO

Este programa é encarregado de gerar a gerência do programa de biblioteca, Recebe já locados :

NBLOC - número de blocos.

NOME - nome da biblioteca.

NSET - número de setores da biblioteca.

NDISK - número do arquivo a ser usado (número lógico).

A partir destes dados, calcula:

NBLOU = NSET * 16 + 1

NBLOR - número de blocos restantes = NBLOC - NBLOU

NBLOL - primeiro bloco livre = NBLOC - NBLOR + 1

NPROG = Ø - número de programas na biblioteca.

Calculadas estas variáveis, a gerência é gravada no disco, no arquivo referenciado por NDISK.

SEQUENCIA DE CHAMADA

CALL WZBIB

SUB-ROTIMAS USADAS

Nenhuma

OBSERVAÇÕES E MENSAGENS DE ERRO

Nenhuma

VARIÁVEIS PRINCIPAIS

Nome	Tipo	Função
NBLOC	C,I	número total de blocos na biblioteca - IND(1).
NBLOU	C,I	número de blocos usados na biblioteca - IND(2).
NBLOR	C,I	número de blocos restantes na biblioteca - IND(3).
NBLOL	C,I	próximo bloco livre - IND(4).
NPRO	C,I	número de programas na biblioteca - IND(5).
NSET	C,I	número de setores na biblioteca para índice-IND(9).
NDISK	C,I	arquivo de disco onde deve ser feita a gravação.

PAGE 31

// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
SUBROUTINE WZBIB

* GERACAO DA BIBLIOTECA *
* NBLLOC = NUMERO DE BLOCOS TOTAL *
* NBLLOU = NUMERO DE BLOCOS OCUPADOS *
* NBLOL = NUMERO DE BLOCOS LIVRES *
* NBLOR = NUMERO DE BLOCOS RESTANTES *
* NPRO = NUMERO DE PROGRAMAS *
*DEVE RECEBER EM
* IND(9) - NUMERO DE SETOTES/INDICE *
* IND(10)- NUMERO DO DISCO *
* NINDL - PRIMEIRA POSICAO DO INDICE *

COMMON IND(10)
EQUIVALENCE (IND(1),NBLLOC),(IND(2),NBLLOU),(IND(3),NBLOR),(IND(4),
1 NBLOL),(IND(5),NPRO),(IND(10),NINDL),(NSET,IND(9))
NBLLOU=NSET*16+1
NBLOR=NBLLOC-NBLLOU
NBLOL=NBLLOC-NBLOR+1
NPRO=0
NDISK=IND(10)
NINDL=21
WRITE(NDISK*1) IND
CALL WZBLS(NDISK)
RETURN
END

FEATURES SUPPORTED
ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZBIB
COMMON 10 VARIABLES

2 PROGRAM 54

RELATIVE ENTRY POINT ADDRESS IS 0006 (HEX)

END OF COMPIILATION

// DJP

*DELETE WZBIB 0010
D 25 NAME NOT FOUND IN LET/FLET

*STORE WS JA WZBIB 0010
CART ID 0010 DB ADDR 0278 DB CNT 0005

NOME DO PROGRAMA

Rotina de inclusão de programa na biblioteca.

CÓDIGO DO PROGRAMA

WZINC

LINGUAGEM USADA

Fortran

TIPO DE PROGRAMA

Subrotina

SISTEMA SUPORTE E HARDWARE

IBM-1130

CHAVES DE PAINEL E SAÍDAS ALTERNATIVAS

Nenhuma

DESCRIÇÃO

Esta rotina recebe do programa chamante os seguintes parâmetros:

nome da biblioteca (IND(6-7-8)).

nome do programa (IND(1-2-3-4-5)).

tipo do programa (IND(9)).

O nome do programa, juntamente com este e todas as suas características é incluído na biblioteca especificada. No índice da biblioteca são incluídos: nome, tipo, comprimento em "disk blocks", endereço de disco, endereço de carregamento e de execução, bem como próximo endereço livre da memória. Se se tratar de uma subrotina estas características são incluídas para cada um dos seus pontos de entrada. O programa é incluído a partir da primeira posição livre.

SEQUÊNCIA DE CHAMADA

CALL WZINC(M)

M - indica se diferente de zero a ocorrência de erro

SUB-ROTIÑAS USADAS

WZCDM

WZABS

WZREL

WZBB

WZBUS

WZPAC

OBSERVAÇÕES E MENSAGENS DE ERRO

PAGE 1

012

// JOB 0001 0010

* E102,MAL

LOG DRIVE	CART SPEC	CART AVAIL	PHY DRIVE
0000	0001	0001	0000
0001	0010	0010	0001

V2 M09 ACTUAL 16K CONFIG 16K

* LABORATORIO DE SISTEMAS DIGITAIS * EPUSP * 24/ 06/ 74 * 20 H 27 MIN

// FOR

*LIST ALL

*ONE WORD INTEGERS

SUBROUTINE WZINC(M)

C *INCLUI PROGRAMA NA BIBLIOTECA ESPECIFICADA *
C * O NOME DA BIBLIOTECA E RECEBIDO EM IND(6-8) *
C * O NOME DO PROGRAMA E RECEBIDO EM IND(1-5) *
C * O FORMATO DO PROGRAMA E RECEBIDO EM IND(9) *
C *OS ARQUIVOS DE BIBLIOTECA TEM NUMEROS 12 E 13 *
C *O ARQUIVO DE PROGRAMA ABSOLUTO TEM NUMERO 11 *
C * O ARQUIVO DE PROGRAMA RELOCAVEL TEM NUMERO 1 *
C *O PRIMEIRO BLOCO DA BIBLIOTECA TEM A FORMA *
C * NBLOC - NUMERO TOTAL DE BLOCOS - POS.1 *
C * NBLOU - NUMERO DE BLOCOS USADOS - POS.2 *
C * NBLOR - NUMERO DE BLOCOS RESTANTES . POS.3 *
C * NBLOL - PRIMEIRO LIVRE - POS.4 *
C * NPROG - NUMERO DE PROGRAMAS - POS.5 *
C * LIBNO - NOME DA BIBLIOTECA - POS.6,7 E 8 *
C * NSET - NUMERO DE SETORES DO INDICE - POS.9 *
C * NINDL - PRIMEIRA POS.LIVRE PARA INDICE - POS.10 *
C *O FORMATO DO INDICE E O SEGUINTE *
C * NOME (1-2) *
C * FORMATO E TIPO DO PROGRAMA (3) *
C * COMPRIMENTO (DISK BLOCKS) *
C * ENDERECO (NUMERO DO DISK BLOCK) *

DIMENSION IND(20),IBUF(320),IPOG(320),IHEAD(60),NOMES(5)

COMMON ITUDO(1430)

EQUIVALENCE(IND(1),ITUDO(1)),(IBUF(1),ITUDO(21)),(IPOG(1),ITUDO
1 (341)),(IHEAD(1),ITUDO(661)),(NBLOR , IND(3)),(NBLOU,
1 IND(2)) ,(NBLOL,IND (4)),(IND(5),NPRO)

EQUIVALENCE (IND(1),NBLOC),(IND(9),NSET),(NINDL,IND(10))

DATA IPRIN/5/

C TESTA VALIDADE DO ROTULO - FORM=0 - ABS.
C FORM=1 - REL.

DO 121 I=1,5

121 NOMES(I)=IND(I)

CALL WZCDM(NOMES,N1,N2).

L=IND(9)+1

GO TO (1,2),L

1 ID=11

GO TO 3

2 ID=1

3 IN=1

READ(ID*IN) IHEAD

PAGE 2

013

```
GO TO (4,5),L
4 CALL WZABS(ID,IN,M)
IK=3
GO TO 6
5 CALL WZREL(M,IK)
6 IF(M) 7,8,7
7 WRITE(IPRIN,100)
100 FORMAT(1H0,5X,'PROGRAMA PEDIDO NAO SE ENCONTRA NO ARQUIVO')
RETURN

C
C      BUSCA BIBLIOTECA
C
8 CALL WZBB(M,DISK)
IF(M) 49,20,49

C
C      VERIFICA SE NOME DO PROGRAMA ESTA NO INDICE
C
20 CALL WZBUS(M,IPONT,NDB,IPNT1,DISK,N1,N2)
IF(M)18,9,18
9 WRITE(IPRIN,106)
105 FORMAT(1H0,5X,'NOME DO PROGRAMA JA CONSTA NA BIBLIOTECA')
M=1
RETURN

C
C      TESTA INCLUSAO DO PROGRAMA
C
18 READ(DISK*1) IND
IF(IHEAD(8)-NBLOR) 60,60,35
35 NBLOK=NBLDC-NBLDU
IF(IHEAD(8)-NBLOK) 50,50,40
102 FORMAT(1H0,5X,'BIBLIOTECA ESGOTADA')
40 WRITE(IPRIN,102)
49 RETURN
50 CALL WZPAC(M,DISK,1)
IF(M) 49,60,49

C
C      TESTA INCLUSAO DO INDICE
C
60 NX=NSET*320+20
NY=NINDL+(IK/3)*8
IF(NY-NX) 120,120,70
70 IF(NBLOR-1) 75,90,90
75 NBLOK=NBLDC-NBLDU
IF(NBLOK) 80,80,90
80 WRITE(IPRIN,102)
RETURN
90 CALL WZPAC(M,DISK,2)
IF(M) 49,120,49

C
C      INCLUI PROGRAMA
C
120 NIN=(NINDL+320)/320
READ(DISK*NIN) IBUF
J=NINDL-(NIN-1)*320
IF(IHEAD(1)) 700,122,700
122 MN=IHEAD(5)+10
IBUF(J+6)=IHEAD(MN)
```

```

      WRITE(5,9090) NINDL,J
9090 FORMAT (1H ,2I6)
     IF(IHEAD(3)-1024)130, 160,130
C
C     INDICE DE PROG.PRINCIPAL
C
130 IBUF(J)=N1
IBUF(J+1)=N2
IB=IHEAD(3)/256
IF(IB-6)131,132,131
132 IBUF(J+2)=IHEAD(14)*256+IB
GO TO 133
131 IBUF(J+2)=IB
133 IBUF(J+3)=IHEAD(8)
IBUF(J+4)=NBLOL
IBUF(J+5)=IHEAD(12)
IBUF(J+7)=IHEAD(4)
GO TO(135,134),L
134 IBUF(J+2)=-IBUF(J+2)
135 WRITE(IDISK'NIN) IBUF
GO TO 600
C
C     TESTA NOME DE SUBROTTINA
C
160 K1=IK+9
DO 190 K=10,K1,3
IF(IHEAD(K)-N1) 190,170,190
170 IF(IHEAD(K+1)-N2)190,500,190
190 CONTINUE
WRITE(IPRIN,105)
105 FORMAT(1H0,5X,'NOME ESPECIFICADO NAO E ENTRY POINT ACEITO')
RETURN
C
C     INCLUI SUBROTTINA
C
200 I=IHEAD(8)
K=NBLOL
K1=IK+9
KK1=IHEAD(3)/256
GO TO (505,504),L
504 KK1=-KK1
505 DO 550 JJ=10,K1,3
N1=IHEAD(JJ)
N2=IHEAD(JJ+1)
CALL WZBUS(M,IPDNT,NDB,IPNT1,DISK,N1,N2)
IF(M)506,9,506
506 CONTINUE
IBUF(J )=N1
IBUF(J+1)=N2
IBUF(J+2)=KK1
IBUF(J+3)=I
IBUF(J+4)=K
IBUF(J+5)=IHEAD(JJ+2)
IBUF(J+6)=IHEAD(MN)
IBUF(J+7)=IHEAD(4)
J=J+8
IF(J-320) 550,560,560

```

PAGE 4

015

550 CONTINUE
GO TO 135
560 WRITE(IDISK*NIN)IBUF
NIN=NIN+16
J=1
GO TO 550

C
C INCLUI PROGRAMA
C

600 I=IHEAD(8)
IN=1
KB=0
READ(ID*IN) IPOG
IDK=NBL0L
J=(I*20+320)/320
DO 650 K1=1,J
IN=IN+16
IF(I=16) 620,620,610
610 K2=320
I=I-16
GO TO 625
620 K2=I*20
625 WRITE(IDISK*NBL0L)(IPOG(KK),KK=1,K2)
IDK=IDK+16
650 CONTINUE

C
C TERMINO
C

WRITE(IPRIN,107)NOMES,IND(6),IND(7),IND(8),NBL0L,IHEAD(8)
I=IHEAD(8)
NBL0L=NBL0L+I
NBL0R=NBL0R-I
NINDL=NINDL+(IK*8)/3
NBLOU=NBLOU+I
NPRO=NPRO+IK/3
WRITE(IDISK*1) IND
WRITE(5,9090) NINDL,J
M=0
RETURN
700 IBUF(J+2)=1
IBUF(J)=N1
IBUF(J+1)=N2
IBUF(J+3)=IHEAD(8)
IBUF(J+4)=NBL0L
GO TO 135
107 FORMAT(1H0,5X,'PROGRAMA ',5A1,' INCLUIDO NA BIBLIOTECA ',3A2, 6X,
1*ENDERECO ',I5,' EXTENSAO EM DB ',I5)
END

VARIABLE ALLOCATIONS

ITUOD(IC)=7FFF-7A6A	IND(IC)=7FFF-7FEC	IBUF(IC)=7FEB-7EAC	IPOG(IC)=7E
NBLOJ(IC)=7FFE	NBL0L(IC)=7FFC	NPRO(IC)=7FFF	NBLOC(IC)=7F
NOMES(I)=0004-0000	I(I)=0005	N1(I)=0006	N2(I)=00
IN(I)=000A	IK(I)=000B	IPRIN(I)=000C	IDISK(I)=00
IPNT1(I)=0010	NBLOK(I)=0011	NX(I)=0012	NY(I)=00
MN(I)=0016	IB(I)=0017	K1(I)=0018	K(I)=00
<B(I)=001C	IDK(I)=001D	K2(I)=001E	KK(I)=00

NOME DO PROGRAMA

15. 017

Rotina de exclusão de um programa da biblioteca.

CÓDIGO DO PROGRAMA

WZEXC

LINGUAGEM USADA

Fortran

TIPO DE PROGRAMA

Subrotina

SISTEMA SUPORTE E HARDWARE

IBM-1130

CHAVES DO PAINEL E SAÍDAS ALTERNATIVAS

Nenhuma

DESCRIÇÃO

Esta rotina exclui um programa da biblioteca, excluindo o seu nome do índice. Este, é compactado a cada exclusão. Os programas não sofrem compactação.

A diretoria da biblioteca é atualizada.

SEQUÊNCIA DE CHAMADA

CALL WZEXC

SUB-ROTINAS USADAS

WZBB

WZCDM

WZBUS

OBSERVAÇÕES E MENSAGENS DE ERRO

Podem ser impressas as seguintes mensagens:

"NOME DO PROGRAMA NAO FIGURA NA BIBLIOTECA"

- o nome perfurado no cartão de entrada não consta do índice da biblioteca.

"PROGRAMA nnnnn FOI EXCLUIDO DA BIBLIOTECA"

- o programa foi encontrado, e excluído o seu nome do índice.

VARIÁVEIS PRINCIPAIS

Nome	Tipo	Função
IND	I,C,()	diretoria da biblioteca
N1,N2	I,	nome do programa em "name code".
MAT	I,()	matriz de trabalho.
M	I	indica se diferente de zero a ocorrência de erro.
IDISK	I	número do arquivo de disco que contém a biblioteca.
NDB	I	extensão em "disk blocks" do programa.

PAGE 5

016

STATEMENT ALLOCATIONS

100	=003B	106	=0055	102	=006E	9090	=007D	105	=0082	107	=009C
4	=011F	5	=012A	6	=012E	7	=0132	8	=0138	20	=0140
49	=0179	50	=017B	60	=0184	70	=019F	75	=01A5	80	=01AF
132	=0229	131	=023D	133	=0246	134	=027C	135	=0286	160	=028F
505	=02E3	506	=0306	550	=036C	560	=0378	600	=0388	610	=03BF

FEATURES SUPPORTED

ONE WORD INTEGERS

CALLED SUBPROGRAMS

WZCDM	WZABS	WZREL	WZBB	WZBUS	WZPAC	SWRT	SCOMP	SIDAI
SDCOM	SDAI	SDIX						

INTEGER CONSTANTS

1=002C	5=002D	11=002E	3=002F	320=0030	20=0031
256=0036	6=0037	9=0038	16=0039	0=003A	

CORE REQUIREMENTS FOR WZINC

COMMON 1430 VARIABLES 44 PROGRAM 1112

RELATIVE ENTRY POINT ADDRESS IS 00C9 (HEX)

END OF COMPIRATION

// DJP

*DELETE WZINC 0010
CART ID 0010 DB ADDR 19C8 DB CNT 0045

*STORE WS UA WZINC 0010
CART ID 0010 DB ADDR 1AF5 DB CNT 0045

// EJECT

Nome	Tipo	Função
IPONT	I	ponteiro inferior no caso de subrotina.
IPNTL	I	ponteiro superior no caso de subrotina.
IBUF	I,C	búffer de leitura e atualização do índice.
NINDL	I	primeira posição livre do índice.
NPRO	I	número de programas gravados.
NBLOU	I	número de blocos usados.

PAGE 9

019

```
// FOR
*ONE WORD INTEGERS
*LIST ALL
    SUBROUTINE WZEXC
C ****
C *EXCLUI PROGRAMA DA BIBLIOTECA ESPECIFICADA *
C   * O NOME DO PROGRAMA E RECEBIDO EM IND(1-5) *
C   * O NOME DA BIBLIOTECA E RECEBIDO EM IND(6-8) *
C   *OS ARQUIVOS DA BIBLIOTECA TEM NUMERO 12 E 13 *
C   *O PRIMEIRO BLOCO DA BIBLIOTECA TEM A FORMA *
C     * NBLOC - NUMERO TOTAL DE BLOCOS - POS.1. *
C     * NBLOU - NUMERO DE BLOCOS USADOS - POS.2 *
C     * NBLDR - NUMERO DE BLOCOS RESTANTES - POS.3 *
C     * NBLOL - PRIMEIRO BLOCO LIVRE - POS.4 *
C     * NPROG - NUMERO TOTAL DE PROGRAMAS - POS.5 *
C     * LIBNO - NOME DA BIBLIOTECA - POS.6,7 E 8 *
C     * NSET - NUMERO DE SETORES DE INDICE *
C     * NINDL - PRIMEIRA POS. LIVRE PARA INDICE *
C   *O FORMATO DO INDICE E O SEGUINTE *
C     * NOME(1-2) *
C     * COMPLEMENTO (4) *
C ****
DIMENSION IND(10),IBUF(360)
DIMENSION MAT(8)
COMMON ITUDO(1430)
EQUIVALENCE (IND(1),ITUDO(1)),(IBUF(1),ITUDO(21)),(NBLOU,IND(2))
1   (NPRO,IND(5)),(IND(10),NINDL)
DATA IPRIN/5/
C
C   BUSCA BIBLIOTECA
C
DO 5 I=1,8
5  MAT(I)=IND(I)
CALL WZCDM(MAT,N1,N2)
CALL WZBB(M,DISK)
IF(M) 10,20,10
10 RETURN
C
C   VERIFICA SE O PROG. ESTA NA BIBLIOTECA
C
20 CALL WZBUS(M,IPONT,NDB,IPNT1 ,DISK,N1,N2)
IF(M) 30,40,30
30 WRITE(IPRIN,100)
100 FORMAT(1H0,5X,'NOME DO PROGRAMA NAO FIGURA NA BIBLIOTECA')
RETURN
40 READ(DISK'1) IND
NBLOU=NBLOU-NDB
C
C   COMPACTA INDICE
C
NIN=(IPONT+320)/320
READ(DISK'NIN) IBUF
N=IPNT1 -IPONT+8
IP=IPONT-(NIN-1)*320
I=IP+N
60 IBUF(IP)=IBUF(I)
IBUF(IP+1)=IBUF(I+1)
```

PAGE 10

020

```
IBUF(IP+2)=IBUF(I+2)
IBUF(IP+3)=IBUF(I+3)
IBUF(IP+4)=IBUF(I+4)
IBUF(IP+5)=IBUF(I+5)
IBUF(IP+6)=IBUF(I+6)
IBUF(IP+7)=IBUF(I+7)
IP=IP+8
IF(IP>320) 75,70,70
70 WRITE(IDISK*NIN)(IBUF(I),I=1,320)
NIN=NIN+16
READ(IDISK*NIN)(IBUF(I),I=1,320)
NIN=NIN+16
IP=1
75 I=I+N
IPONT=IPONT+8
IF(IPONT-NINDL+N) 60,200,200
C
C   FINALIZACAO
C
200 WRITE(IDISK*NIN)(IBUF(K),K=1,IP)
NINDL=NINDL-N
NPRO=NPRO-N/8
WRITE(IDISK*1) IND
WRITE(IPRIN,101) MAT
101 FORMAT(1H0,5X,'PROGRAMA ',5A1,' FOI EXCLUIDO DA BIBLIOTECA ',3A
M=0
RETURN
END
```

VARIABLE ALLOCATIONS

ITUDO(IC)=7FFF-7A6A	IND(IC)=7FFF-7FF6	IBUF(IC)=7FEB-7E84	NBLDOU(IC)
MAT(I)=0007-0000	I(I)=0008	N1(I)=0009	N2(I)
IPONT(I)=000D	NDB(I)=000E	IPNT1(I)=000F	IPRIN(I)
IP(I)=0013	K(I)=0014		

STATEMENT ALLOCATIONS

100 =0029	101 =0043	5 =0065	10 =0084	20 =0086	30 =00
200 =01AC					

FEATURES SUPPORTED
ONE WORD INTEGERS.

CALLED SUBPROGRAMS

WZCDM	WZBB	WZBUS	SWRT	SCOMP	SIOAI	SUBSC	SDRED	SDWR
-------	------	-------	------	-------	-------	-------	-------	------

INTEGER CONSTANTS

1=0024	8=0025	320=0026	16=0027	0=0028
--------	--------	----------	---------	--------

CORE REQUIREMENTS FOR WZEXC

COMMON 1430 VARIABLES	36 PROGRAM	452
-----------------------	------------	-----

RELATIVE ENTRY POINT ADDRESS IS 0060 (HEX)

END OF COMPIILATION

// DUP

*DELETE WZEXC 0010

=7FFE NPRO(IC)=7FFB NINDL(IC)=7FF6
=000A M(I)=000B IDISK(I)=000C
=0010 NIN(I)=0011 N(I)=0012

3 40 =0099 60 =00CF 70 =015B 75 =0198

SDCOM SDAI SDIX

PAGE 11

CART ID 0010 DB ADDR 019A DB CNT 0021

021

*STORE WS UA WZEXC 0010
CART ID 0010 DB ADDR 1B22 DB CNT 001D

// EJECT

NOME DO PROGRAMA

Rotina de compactação da biblioteca

CÓDIGO DO PROGRAMA

WZPAC

LINGUAGEM USADA

Fortran

022

SISTEMA SUPORTE E HARDWARE

IBM-1130

CHAVES DE PAINEL E SAÍDAS ALTERNATIVAS

Nenhuma

DESCRIÇÃO

Este programa compacta a biblioteca, iliminando espaços vazios, originados pela exclusão de programas. Para tal reestruturação, diretoria e índice são alterados.

SEQUÊNCIA DE CHAMADA

CALL WZPAC(M, IDISK, ISW)

M - parâmetro de erro. Ocorreu erro se M é diferente de zero.

IDISK - arquivo de disco que deve ser compactado.

ISW - a compactação deve se dar com extensão do índice de um setor (320 palavras).

SUB-ROTINAS USADAS

WZIND

OBSERVAÇÕES E MENSAGENS DE ERRO

Pode ocorrer a seguinte mensagem de erro:

'ERRO GRAVE NA BIBLIOTECA'

O significado desta mensagem é o seguinte: um erro ocorreu, sem que desse ter ocorrido, eventualmente por um problema de má estruturação de programas, ou então em consequência de uma rotina ter-se perdido, ou sido incorretamente usada. Deve-se regravar todo o programa.

VARIÁVEIS PRINCIPAIS

Nome	Tipo	Função
IBUF	I,C,()	buffer de leitura e tratamento do índice.
IND	I,C,()	diretoria da biblioteca.
NBLOC	I	número total de blocos.
NBLOR	I	número de blocos restantes.
NBLOU	I	número de blocos usados.
IDISK	I	arquivo que contém a biblioteca.
N	I	extensão de um programa em "disk blocks".
IENDL	I	primeiro bloco livre para programa.
NSET	I	número de setores de índice.

```

/* FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
  SJROUTINE WZPAC(M, IDISK, ISW)
  ****
  *SUBROTAINA DE COMPACTACAO DA BIBLIOTECA *
  *PODE OJ NAO EXTENDER O INDICE             *
  *      ISW=1 - EXTENDE O INDICE            *
  *O PRIMEIRO BLOCO DA BIBLIOTECA TEM A FORMA   *
  *      NBLOU - NUMERO DE BLOCOS JSADD - POS,2    *
  *      NBLOR - NUMERO DE BLOCOS RESTANTES - POS.3  *
  *      NINDL - PRIMEIRA POSICAO LIVRE PARA INDICE - POS10 *
  *FORMATO DO INDICE                         *
  *      NOME(1-2)                            *
  *      COMPRIMENTO (4)                      *
  *      ENDERECO (5)                         *
  ****
DIMENSION IND(10),IBUF(320),IBUF1(320)
COMMON ITUDO(1430)
EQUIVALENCE (IND(1),ITUDO(1)),(IBUF(1),ITUDO(21)),(IND(2),NBLOU),
1      (IND(3),NBLOR),(IND(10),NINDL)
EQUIVALENCE (IBUF1(1),ITUDO(1000))
EQUIVALENCE (NBLOC,IND(1)),(NSET,IND(9))
DATA IPRIN/5/
N1=1
READ(IDISK*N1) IBUF
READ(IDISK*1) IND
IF(IND(5))17,17,18
13 NBLOK=NBLDC-NBLOU
IF(NBLOK-NBLOR) 50,10,50
      EXTENDE INDICE
10 IF(ISW-1) 15,16,15
15 RETURN
15 CALL WZIND(M, IDISK)
17 RETURN
      COMPACTA
50 J=21
N=IBUF(24)
IADD=IBUF(25)
IEND=IADD
IENDL=NSET*16+2
<=J
60 IF(IENDL-IADD)79,155,70
70 WRITE(IPRIN,101)
101 FORMAT(1H0,5X,'ERRO GRAVE NA BIBLIOTECA')
M=1
RETURN
      MOVE PROGRAMA
79 IBUF(J+4)=IENDL
IF(N-16)78,77,77
77 NX=320

```

PAGE 18

GO TO 80
73 NX=N*20
80 READ(IDISK*IADD)(IBUF1(II),II=1,NX)
WRITE(IDISK*IENDL)(IBJF1(II),II=1,NX)
IF(N-16) 150,150,90
90 IENDL=IENDL+16
IADD=IADD+16
N=N-16
GO TO 80
150 IFICA=IENDL
IENDL=IENDL+N

155 J=J+8
<=K+8
IF(J-320)180,160,160
160 WRITE(IDISK*N1)IBUF
N1=N1+16
READ(IDISK*N1)IBUF
J=1
180 IF(K-N1DL) 190,181,181
181 WRITE(IDISK*N1) IBUF
NBLDR=NBLDK
IND(4)=IENDL
WRITE(IDISK*1)IND
GO TO 10
190 IADD=IBJF(J+4)
N=IBJF(J+3)
IF(IADD-IEND) 200,205,200
200 IEND=IADD
N=IBUF(J+3)
GO TO 60
205 IBUF(J+4)=IFICA
GO TO 155
END

024

FEATURES SUPPORTED
ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZPAC
COMMON 1430 VARIABLES 14 PROGRAM 374

RELATIVE ENTRY POINT ADDRESS IS 0026 (HEX)

END OF COMPIILATION

// DJP

*DELETE WZPAC 0010
D 26 NAME NOT FOUND IN LET/FLET

*STORE WS JA WZPAC 0010
CART ID 0010 DB ADDR 01FE DB CNT 0018

// EJECT

NOME DO PROGRAMA

Rotina para extender o índice da biblioteca

CÓDIGO DO PROGRAMA

025

WZIND

LINGUAGEM USADA

Fortran

TIPO DE PROGRAMA

Subrotina

SISTEMA SUPORTE E HARDWARE

IBM-1130

CHAVES DE PAINEL E SAÍDAS ALTERNATIVAS

Nenhuma

DESCRIÇÃO

Este programa extende o índice da biblioteca, identificada pelo parâmetro IDISK. A extensão é sempre de um setor. Isto implica em se relocar todos os programas de 16 "disk blocks" para o fim da biblioteca. O índice da biblioteca é naturalmente atualizado.

SEQUÊNCIA DE CHAMADA

CALL WZIND(M, IDISK)

M - indica se diferente de zero que ocorreu erro.

IDISK - identifica a biblioteca, a extender índice.

SUBROTINAS USADAS

Nenhuma

OBSERVAÇÕES E MENSAGEM DE ERRO

Pode ocorrer a seguinte mensagem de erro:

'IMPOSSIVEL ESTENDER INDICE - OVERFLOW'

VARIÁVEIS PRINCIPAIS

NOME	Tipo	Função
NBLOC	I	número total de blocos.
NBLOU	I	número de blocos usados.
NBLOR	I	número de blocos restantes.
NBLOL	I	próximo bloco livre.
NPROG	I	número de programas.
NSET	I	próximo endereço livre de índice.
NINDL	I	número de setores de índice.
IBUF	I	buffer de leitura
IND	I	diretoria da biblioteca.

PAGE 7

026

```
// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
  SUBROUTINE NZIND(M, IDISK)
  ****
  *ESTA SUBROTINA RELOCA A BIBLIOTECA, EXTENDENDO O *
  *INDICE *
  *O PRIMEIRO BLOCO DA BIBLIOTECA TEM A FORMA *
  *   NBLLOC - NUMERO DE BLOCOS TOTAL - POS.1 *
  *   NBLOU - NUMERO DE BLOCOS USADOS -POS.2 *
  *   NBLOR - NUMERO DE BLOCOS RESTANTES - POS.3 *
  *   NBLDL - NUMERO DE BLOCOS LIVRES - POS.4 *
  *   NBLOG - NUMERO TOTAL DE PROGRAMAS - POS.5 *
  *   NSET  - NUMERO DE SETORES DE INDICE - POS.9 *
  *   NINDL - PRIM.POS.LIVRE PARA INDICE - POS.10 *
  *O FORMATO DO INDICE E O SEGUINTE -
  *   NOME(1-2) *
  *   ENDERECO (5) *
  ****
DIMENSION IND(10),IBUF(320)
COMMON ITUDO(1430)
EQUIVALENCE (IND(1),ITUDO(1),IBLOC),(IBUF(1),ITUDO(21)),(IND(2),
1           NBLOU),(IND(3),NBLOR),(IND(4),NBLDL),(NPROG,IND(5)),
2           (NSET,IND(9)),(NINDL,IND(10))
DATA IPRIN/5/
READ (IDISK'1)IND
IF(NBLOR-16) 10,20,20
10 WRITE(IPRIN,100)
100 FORMAT(1H0,5X,'IMPOSSIVEL ESTENDER INDICE - OVERFLOW')
M=1
RETURN
C
C      CALC. ULTIMO SETOR(I), PRIM.SETOR(J)
C
20 I=NBLDL-16
J=(NINDL+320)/320
<=I+16
25 READ(IDISK'<)IBUF
WRITE(IDISK'K) IBUF
<=I
I=I-16
IF(I-J) 30,25,25
C
C      ACERTA NOVOS ENDEREÇOS DE PROGRAMAS
C
30 I=21
J=1
<=1
READ(IDISK'<)IBUF
35 IF(J-NPROG) 40,40,80
40 IBUF(I+4)=IBUF(I+4)+16
I=I+8
J=J+1
IF(I-320) 35,50,50
50 WRITE(IDISK'K) IBUF
<=K+16
READ(IDISK'<)IBUF
```

PAGE 8

I=1
GO TO 35

027

FINALIZACAO

80 NBLQJ=NBLQJ+16
WRITE(IDISK^K) IBUF
NBLQR=NBLQR-16
NSET=NSET+1
NBLQL=NBLQL+16
WRITE(IDISK^I) IND
M=0
RETURN
END

FEATURES SUPPORTED
ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZIND
COMMON 1430 VARIABLES 6 PROGRAM 240

RELATIVE ENTRY POINT ADDRESS IS 0024 (HEX)

END OF COMPILATION

// DJP

*DELETE WZIND 0010
D 26 NAME NOT FOUND IN LET/FLET

*STORE WS JA WZIND 0010
CART ID 0010 DB ADDR 01A2 DB CNT 0010

// EJECT

NOME DO PROGRAMA

Rotina de chamada do carregador

CODIGO DO PROGRAMA

WZCAR

028

LINGUAGEM USADA

Fortran

TIPO DE PROGRAMA

Subrotina

SISTEMA SUPORTE E HARDWARE

IBM-1130

CHAVES DE PAINEL E SAIDAS ALTERNATIVAS

Nenhuma

DESCRIÇÃO

Ver a do programa WZV

SEQUENCIA DE CHAMADA

CALL WZCAR

SUB-ROTIMAS USADAS

ver WZV

OBSERVAÇÕES E MENSAGENS DE ERRO

ver WZV

VARIAVEIS PRINCIPAIS

ver WZV

67 3046076FC
69 253486304
71 284166F00
73 3043076DE
75 263686700
77 310297935
79 1660340DB
81 164394037
83 -2843790EB
85 194804C18
87 307887844
89 -3253680E8
91 276486C00
93 3042076D4
95 286737001
97 16431402F
99 -3868F0E4

029

// FOR

*ONE WORD INTEGERS

*LIST SOURCE PROGRAM

SUBROUTINE WZCAR

* PREPARA ENTRADA PARA O LOADER, TESTANDO *

* CARTAO DE CHAMADA *

* ARQUIVOS - 12 E 13 - BIBLIOTECAS *

* 11 - SAIDA DO MONTADOR *

* 1 - SAIDA DO CARREGADOR *

* 14 - ERROS *

* INDICADORES - *

* IND(17)-NUMERO DE BIBLIOTECAS *

* (0=1 E 1=2) *

* IND(18)-MAPA DA SAIDA *

* IND(11)-SUBROTIÑA DE INTERRUP-*

* CAO (0=SIM E 1=NAO) *

* IND(12-13-14-15-16)-NOME DA *

* SUBROTIÑA DE INT. *

* IND(6-7-8)-NOME DA BIBL.DO PRO*

* GRAMA *

* IND(1-2-3-4-5)-NOME DO PROG. *

* IND(19)-SAI EM CARTOES *

INTEGER FB(3)

DIMENSION MAT(4)

DIMENSION IP(5),IND(20),IREST(60),IBUF(320)

DIMENSION IE(400)

COMMON IVI(3000)

EQUIVALENCE (ISEC,IND(17))

EQUIVALENCE (IVI(1),IND(1)),(IVI(21),IBUF(1))

1 , (IND(1),IP(1)),(IND(6),FB(1))

DATA IBK,IBK2/' ', ' '

DATA MMM/'M'/,ICC/'C'/

IND(18)=0

IND(19)=0

I=1

C C BUSCA PEDIDO DE MAPA

C CALL WZB1(IREST,I, M)

PAGE 3

030

```
IF(M)10,4,10
C
C      BUSCA PEDIDO SAIDA EM CARTOES
C
4 IF(IREST(I)-MMM)6,5,6
5 IND(18)=1
I=I+1
CALL WZB1(IREST,I,M)
IF(M) 10,6,10
6 IF(IREST(I)-ICC) 30,7,30
7 IND(19)=1
I=I+1
C
C      BUSCA NUMERO DE BIBLIOTECAS
C
CALL WZB1(IREST,I,M)
IF(M) 10,30,10
10 WRITE(5,1001)
1001 FORMAT(1H ,5X,'COMANDO INVALIDO')
15 CALL EXIT
30 CALL WZC1(IREST,I,N,M)
IF(M) 10,40,10
40 IF(N-2)50,60,10
50 ISEC=0
GO TO 65
60 ISEC=1
C
C      BUSCA SUBROTINA DE INTERRUPCAD
C
65 CALL WZB1(IREST,I,M)
IF(M) 70,80,70
70 IND(11)=1
GO TO 125
80 DO 85 J=12,16
85 IND(J)=IBK
IND(11)=0
N=I+4
IF(N-60) 110,110,86
86 N=60
110 K=12
DO 120 J=I,N
IF(IREST(J)-IBK) 90,125,90
90 IND(K)=IREST(J)
120 K=K+1
C
C
125 IF(FB(1)-IBK2) 140,130,140
130 IDISK=1
ID=1
GO TO 160
C
C      BUSCA PROGRAMA NA BIBLIOTECA
C
140 CALL WZBB(M,DISK)
IF(M)10,142,10
142 CALL WZCDM(IP,N1,N2)
CALL WZBUS(M,IPONT,NDB,IPNT1,DISK,N1,N2)
```

PAGE 4

031

```
IF(M)10, 150,10
150 NIT=(IPONT+320)/320
      N=IPONT-(NIT-1)*320
      READ(IDISK*NIT) IBUF
      ID=IBUF(N+4)
      IE(351)=IDISK
      IE(301)=ID
      WRITE(5,9091) IND
160 CALL WZLOA(IDISK, ID)
      DO 333 I=1,200,2
      CALL WZZ(N,I)
      CALL WZBH(N,MAT)
      333 WRITE(5,3333) I,MAT
      3333 FORMAT(1H ,I5,4A1)
      9091 FORMAT(1H ,20I6)
      RETURN
      END
```

UNREFERENCED STATEMENTS

15

FEATURES SUPPORTED
ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZCAR
COMMON 3000 VARIABLES 484 PROGRAM 406

RELATIVE ENTRY POINT ADDRESS IS 0206 (HEX)

END OF COMPILATION

// DUP

*DELETE WZCAR 0010
D 26 NAME NOT FOUND IN LET/FLET

*STORE WS UA WZCAR 0010
CART ID 0010 DB ADDR 1929 DB CNT 001A

// EJECT

NOME DO PROGRAMA

Rotina que retira um determinado programa da biblioteca e o coloca na memória a fim de realizar o "dump" em cartões.

CODIGO DO PROGRAMA

WZRT

LINGUAGEM USADA

Fortran

TIPO DE PROGRAMA

Subrotina

SISTEMA SUPORTE E HARDWARE

IBM-1130

CHAVES DE PAINEL E SAIDAS ALTERNATIVAS

Nenhuma

Descrição

Este programa, busca e copia da biblioteca um determinado programa na memória principal. O programa deve estar em formato absoluto, e é copiado integralmente, mesmo que não se inicie na posição Ø de memória.

SEQUENCIA DE CHAMADA

CALL WZRT

SUB-ROTINAS USADAS

WZBB

WZCDM

WZBUS

WZLEV

OBSERVAÇÕES E MENSAGENS DE ERRO

O programa produz uma das seguintes mensagens:

" PROGRAMA NAO ESTA NA BIBLIOTECA"

o programa pedido não se encontra na biblioteca.

" PROGRAMA NÃO ESTA NO FORMATO PEDIDO"

o programa encontrado, não está em formato absoluto.

" PROGRAMA JA SE ENCONTRA NA MEMORIA"

o programa foi carregado, e o controle é passado ao programa que perfura os cartões.

VARIAVEIS PRINCIPAIS

Nome	Tipo	Função
IDISK	I.	contém o número do arquivo que contém a biblioteca especificada através de IND(6-7-8).
IP	I	matriz que contém o nome do programa,

Nome	Tipo	Função	033
N1,N2	I	nome do programa.	
M	I	indica erro se diferente de zero.	
IPONT	I	ponteiro inferior se for rotina.	
IPNT1	I	ponteiro superior idem.	
NDB	I	extensão do programa.	
IBUF	I	buffer de leitura do programa.	
IEND	I	endereço do programa no disco.	
IND	I	cabeçalho do programa absoluto.	
ICAR	I	endereço de carregamento.	

PAGE 1

// JDB 0001 0010

034 E102,

LOG DRIVE	CART SPEC	CART AVAIL	PHY DRIVE
0000	0001	0001	0000
0001	0010	0010	0002

V2 M09 ACTUAL 16K CONFIG 16K

* LABORATORIO DE SISTEMAS DIGITAIS * EPUSP * 20/ 05/ 74 * 17 H 57 M

// FOR

*ONE WORD INTEGERS

*LIST ALL

```
SUBROUTINE WZRT
DIMENSION IND(20),IBUF(320),IP(5),FB(3),IN(20)
COMMON ITJDO(3000)
EQUIVALENCE (ITUDO(1),IND(1)),(ITUDO(21),IBUF(1)),
1          (IND(1),IP(1)),(IND(6),FB(1))
CALL WZBB(M,DISK)
IF(M) 10,20,10
10 RETURN
C
20 CALL WZCDM(IP,N1,N2)
CALL WZBUS(M,IPONT,NDB,IPNT1,DISK,N1,N2)
IF(M) 18,30,18
18 WRITE(5,1000)
1000 FORMAT(1H0,10X,'PROGRAMA NAO ESTA NA BIBLIOTECA')
RETURN
C
30 IK=(IPONT+20)/20
READ(DISK'IK)(IBUF(I),I=1,40)
II=IPONT-(IK-1)*20
IF(IBUF(II+2)) 35,35,40
35 WRITE(5,1001)
1001 FORMAT(1H0,10X,'PROGRAMA NAO ESTA NO FORMATO PEDIDO')
RETURN
C
40 NDB=IBUF(II+3)-1
IEND=IBUF(II+4)
READ(DISK'IEND) IND
IEND=IEND+1
ICAR=0
ITUDO(2999)=IND(4)
ITUDO(3000)=IND(3)
DO 100 JK=1,NDB
READ(DISK'IEND) IN
DO 90 IK=1,20
CALL WZLEV(IN (IK),ICAR)
90 ICAR=ICAR+1
100 CONTINUE
WRITE(5,1002)
1002 FORMAT(1H0,10X,'PROGRAMA JA SE ENCONTRA NA MEMORIA')
CALL LINK(WZSDS)
END
VARIABLE ALLOCATIONS
ITJDO(IC)=7FF-7448   IND(IC)=7FF-7FEC   IBUF(IC)=7FEB-7EAC   IP(IC)=
M(I )=0014           IDISK(I )=0015           N1(I )=0016           N2(I )=
```

PAGE 2

135

IPNT1(I)=001A IK(I)=001B I(I)=001C II(I
JK(I)=0020

STATEMENT ALLOCATIONS

1000 =002B 1001 =0040 1002 =0057 10 =0076 20 =0078 18 =00
100 =012A

FEATURES SUPPORTED
ONE WORD INTEGERS

CALLED SUBPROGRAMS

WZB3 WZCDM WZBUS WZLEV SWRT SCOMP SUBSC SDRED SDAI

INTEGER CONSTANTS

5=0026 20=0027 1=0028 40=0029 0=002A

CORE REQUIREMENTS FOR WZRT

COMMON 3000 VARIABLES 38 PROGRAM 278

RELATIVE ENTRY POINT ADDRESS IS 006D (HEX)

END OF COMPIRATION

// DJP

*DELETE WZRT 0010
CART ID 0010 DB ADDR 1959 DB CNT 0011

*STORE WS UA WZRT 0010
CART ID 0010 DB ADDR 199E DB CNT 0012

// EJECT

NOME DO PROGRAMA

Rotina para colocar uma palavra de informação em determinado endereço

CODIGO DO PROGRAMA

WZLEV

LINGUAGEM USADA

Fortran

TIPO DE PROGRAMA

Subrotina

SISTEMA SUPORTE E HARDWARE

IBM-1130

CHAVES DE PAINEL E SAIDAS ALTERNETIVAS

Nenhuma

DESCRIÇÃO

Esta rotina coloca uma palavra de informação em um endereço que é calculado por :

/7800 + endereço fornecido.

SEQUENCIA DE CHAMADA

CALL WZLEV(IDADO,IEND)

IDADO : dado a ser guardado

IEND : endereço fornecido

SUB-ROTIMAS USADAS

Nenhuma

OBSERVAÇÕES E MENSAGENS DE ERRO

Nenhuma

VARIAVEIS PRINCIPAIS

Nenhuma

PAGE 1

037

// JCB 0001 0010

* E102,MAL

LOG DRIVE	CART SPEC	CART AVAIL	PHY DRIVE
0000	0001	0001	0000
0001	0010	0010	0001

V2 M09 ACTUAL 16K CONFIG 16K

* LABORATORIO DE SISTEMAS DIGITAIS * EPUSP * 17/ 05/ 74 * 16 H 28 MIN

// ASM

*LIST

0000	26A53165	00001	ENT	WZLEV
0000 0	0000	00002	WZLEV DC	0
		00003	*****	
		00004	* ROTINA PARA COLOCAR NA MEMORIA *	
		00005	* UMA PALAVRA *	
		00006	* PARAMETROS- *	
		00007	* PALAVRA A GUARDAR *	
		00008	* ENDERECO *	
		00009	*****	
0001 01	55800000	00010	LDX I1	WZLEV
0003 0	C101	00011	LD 1 1	CARREGA ENDERECO
0004 0	8008	00012	A X7800	ACERTA ENDERECO
0005 0	D002	00013	STD F1+1	
0006 0	C100	00014	LD 1 0	CARREGA PALAVRA
0007 00	D4000000	00015	F1 STD L 0	
0009 0	7102	00016	MDX 1 2	
000A 0	69F5	00017	STX 1 WZLEV	ACERTA RETORNO
000B 01	4C800000	00018	B I WZLEV	
000D 0	7800	00019	X7800 DC /7800	
000E		00020	END	

000 OVERFLOW SECTORS SPECIFIED

000 OVERFLOW SECTDRS REQJIRED

003 SYMBOLS DEFINED

NO ERROR(S) AND NO WARNING(S) FLAGGED IN ABOVE ASSEMBLY

// DJP

*DELETE WZLEV 0010
D 26 NAME NOT FOUND IN LET/FLET

*STORE WS JA WZLEV 0010
CART ID 0010 DB ADDR 1981 DB CNT 0002

// EJECT

NOME DO PROGRAMA

Programa para listar índice de biblioteca.

CÓDIGO DO PROGRAMA

WZLSI

LINGUAGEM USADA

Fortran

TIPO DE PROGRAMA

Subrotina

SISTEMA DE SUPORTE E HARDWARE

IBM-1130

CHAVES DE PAINEL E SAÍDAS ALTERNATIVAS

Nenhuma

DESCRIÇÃO

O programa lista inicialmente o cabeçalho da biblioteca, isto é:

- nome.
- total de blocos.
- blocos usados.
- blocos restantes.
- bloco livre.
- número de programas.
- número/setores.

A seguir lista os programas contidos na biblioteca: 40/página.

São listados para cada programa:

- nome.
- comprimento.
- endereço 1º bloco.

SEQUÊNCIA DE CHAMADA

CALL WZLSI (IDISK).

IDISK - variável que indica qual o arquivo que contém a biblioteca.

SUB-ROTINAS USADAS

WZPG

WZBLS

WZDCM

OBSERVAÇÕES

Nenhuma

VARIÁVEIS PRINCIPAIS

Nome	Tipo	Função
IPAG	I	contador de páginas.

Nome	tipo	Função
ID	I	registro de disco.
IPR	I	contador de programas que vão sendo impressos.
IBUF	I,(),C	buffer de leitura.
INOME	I,(),C	nomes dos programas.
IEND	I,(),C	endereço dos programas.
IBL	I,(),C	número de blocos dos programas.

PAGE 15

040

```
// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
SJBROUTINE WZLSI(IDISK,IPAG)
***** LISTA 3 INDICE DA BIBLIOTECA , 64 PROGRAMAS POR PAGINA *
***** DIMENSION IND(20),IBUF(320),IEND(64),IBL(64),INOME(64,3)
COMMON ITJDO(1430)
EQUIVALENCE (ITUDO(1),IND(1)),(IBUF(1),ITUDO(21)),(IEND(1),ITUDO
1           (341)),(IBL(1),ITJDO(405)),(ITUDO(469),INOME(1,1))
DATA IPRIN/5/
CALL WZPG(IPAG)
CALL WZBLS(IDISK)
IPR=IND(5)
IF(IPR) 2,80,2
2 ID=2
5 WRITE(IPRIN,1000)
J=1
READ(IDISK'ID) IBUF
IF(IPR-40)10,20,20
10 K=IPR
GO TO 30
20 K=64

TRECHO DE IMPRESSAO

30 DO 40 I=1,K
N1=IBUF(J)
N2=IBUF(J+1)
IBL(I)=IBUF(J+3)
IEND(I)=IBUF(J+4)
CALL WZDCM(N1,N2,N3)
J=J+8
INOME(I,1)=N1
INOME(I,2)=N2
40 INOME(I,3)=N3
WRITE(IPRIN,1001)((INOME(J,N),N=1,3),IBL(J),IEND(J),J=1,K)
IF(IPR -40) 80,80,50
50 IPR=IPR-40
ID=ID+16
IPAG=IPAG+1
CALL WZPG(IPAG)
WRITE(IPRIN,1002)
1002 FORMAT(1H ,///)
GO TO 5

FIM

1001 FORMAT( 4(5X,3A2,I5,I5,5X))
1000 FORMAT(////////, 4(5X,'NOME',2X,'COMP',,'END',5X),/,,
1     4(5X,'PROG',2X,'DB ',11X),//)
80 RETURN
END
```

FEATURES SUPPORTED
ONE WORD INTEGERS

NOME DO PROGRAMA

Rotina que escreve cabeçalho / lista gerência da bibliotec

CÓDIGO DO PROGRAMA

WZPG / WZBLS

LINGUAGEM USADA

Fortran

TIPO DO PROGRAMA

Subrotina

SISTEMA SUPORTE E HARDWARE

IBM-1130

CHAVES DE PAINEL E SAÍDAS ALTERNATIVAS

Nenhuma

DESCRIÇÃO

Estas rotinas, respectivamente, geram o cabeçalho de página, e escrevem o título da gerência da biblioteca, bem como lista as principais variáveis nesta.

SEQUÊNCIA DE CHAMADA

CALL WZPG(IPAG)

IPAG - contador de páginas

CALL WZBLS(IDISK)

IDISK - arquivo cuja gerência deve ser listada

SUB-ROTIÑAS USADAS

Nenhuma

OBSERVAÇÕES E MENSAGENS DE ERROS

Nenhuma

VARIÁVEIS PRINCIPAIS

Nenhuma

```

// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
  SUBROUTINE WZBLS(IDISK)
  *****
  *LISTA CABECALHO DA BIBLIOTECA *
  *****
COMMON IND(20)
EQUIVALENCE (I1,IND(1)),(I2,IND(2)),(I3,IND(3)),(I4,IND(4)),
1           (I5,IND(5)),(I9,IND(9))
DATA IPRIN/5/
READ(IDISK*1)IND
WRITE(IPRIN,10)(IND(I),I=6,8),I1,I2,I3,I4,I5,I9
10 FORMAT(1H0,5X,'BIBLIOTECA ',3A2,/,6X,'TOTAL',T16,'BLOCOS',T26,
J      'BLOCOS',T36,'BLDCO',T46,'NUM.DE',T56,'SETORES',/,6X,
J      'BLOCOS',T16,'JSADDS',T26,'REST.',T36,'LIVRE',T46,'PROG.',
J      T56,'INDICE',/,6X,6(I5,5X))
RETURN
END

```

FEATURES SUPPORTED
ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZBLS
COMMON 20 VARIABLES 4 PROGRAM 134

RELATIVE ENTRY POINT ADDRESS IS 0059 (HEX)

END OF COMPILATION

// DJP

*DELETE WZBLS 0010
D 25 NAME NOT FOUND IN LET/FLET

*STORE WS UA WZBLS 0010
CART ID 0010 DB ADDR 02~~55~~ DB CNT 000A

NOME DO PROGRAMA

Rotina de paginação

CÓDIGO DO PROGRAMA

944

WZPG

LINGUAGEM USADA

Fortran

TIPO DO PROGRAMA

Subrotina

SISTEMA SUPORTE E HARDWARE

IBM-1130

CHAVES DO PAINEL OU SAÍDAS ALTERNATIVAS

Nenhuma

DESCRIÇÃO

Esta rotina inicializa página, imprimindo cabeçalho, e incrementando o contador de páginas.

SEQUÊNCIA DE CHAMADA

CALL WZPG(IPAG)

IPAG : contador de páginas

SUB-ROTINAS USADAS

Nenhuma

OBSERVAÇÕES E MENSAGENS DE ERRO

Nenhuma

VARIÁVEIS PRINCIPAIS

Nenhuma

PAGE 16

045

```
// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
  SUBROUTINE WZPG(IPAG)
  DATA IPRIN/5/
  WRITE(IPRIN,100) IPAG
100 FORMAT(1H1,10X,'L A B O R A T O R I O D E S I S T E M A S D I G
  1 I T A I S - L.S.D. - PAG.',I4,/)
  IPAG=IPAG+1
  RETURN
  END
```

FEATURES SUPPORTED
ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZPG
COMMON 0 VARIABLES 2 PROGRAM 74

RELATIVE ENTRY POINT ADDRESS IS 0036 (HEX)

END OF COMPIILATION

// DUP

*DELETE WZPG 0010
CART ID 0010 DB ADDR 0234 DB CNT 0005

*STORE WS UA WZPG 0010
CART ID 0010 DB ADDR 1871 DB CNT 0006

// EJECT

946

SISTEMA CROSS-CARREGADOR

WZV

WZLOA

WZVER

WZENC

WZDLE

WZAN

WZZ

WZSME

WZCOL

WZLOC

WZBC

WZWRT

WZDS

WZMAP

WZGRA

WZTB

WZPOE

WZCAL

NOME DO PROGRAMA

Programa Carregador

CÓDIGO DO PROGRAMA

WZV

947

LINGUAGEM USADA

Fortran

TIPO DE PROGRAMA

Programa principal

SISTEMA SUPORTE E HARDWARE

IBM-1130

CHAVES DE PAINEL E SAÍDAS ALTERNATIVAS

Nenhuma

DESCRIÇÃO

Este programa lê um cartão indicando as opções permitidas para o carregamento. O cartão pode ter a seguinte forma:

*LOAD BBBBPPPPP M C n NNNNN

onde *LOAD - é dispensável

BBBBBB - nome da biblioteca onde deve ser buscado o programa.

- se este campo é deixado em branco, o programa se encontra no arquivo de disco do Assembler.

PPPPP - nome do programa na biblioteca BBBB.

M - pode ser dispensado, e se perfurado indica pedido de mapa de carregamento

C - indica pedido de perfuração de cartões. Se não aparecer no cartão, o programa é armazenado no arquivo de saída do carregador.

n - se igual a 2, indica duas bibliotecas à disposição do carregador

NNNN - rotina de interrupção a ser utilizada.
pode ser dispensada.

SEQUENCIA DE CHAMADA

É chamado por :

// XEQ 1

*FILE(1,N08),(11,WZN11),(12,BBB12),(13,BBB13),(14,WZNOQ)

N08 - arquivo de saída do montador

WZN11 - arquivo de saída do carregador

BBB12 e BBB13 - arquivos de biblioteca

WZNOQ - arquivo que contém as mensagens de erro do carregador.

SUB-ROTINAS USADAS

WZBL

WZCL

WZBB

WZCDM

WZBUS

WZLOA

048

OBSERVAÇÕES E MENSAGENS DE ERRO

Podem ser impressas as seguintes mensagens de erro:

"COMANDO INVALIDO"

- o cartão fornecido está errado;
- dois pedidos de M ou C
- não foi indicado o numero de bibliotecas

VARIÁVEIS PRINCIPAIS

Nome	Tipo	Função
IND	I	prox buffer de leitura do cartão
IREST	I	Buffer de leitura da segunda parte do cartão
FB	I,()	contém nome da biblioteca
IP	I,()	contém nome do programa
I	I	ponteiro para o buffer IREST
M	I	indica se diferente de zero a ocorrência de erro
IND(18)	I	se igual a 1, indica pedido de mapa
IND(19)	I	se igual a 1, indica pedido de saída de cartões
ISEC	I	se igual a 1, idica duas bibliotecas a disposição do programa
IND(11)	I	se igual a zero, indica rotina de interrupção indicada
IND(12 a 16)	I	nome da subrotina de interrupção dse se existente

PAGE 1

// JOB 0001 0010

049

E102,M/

LOG DRIVE	CART SPEC	CART AVAIL	PHY DRIVE
0000	0001	0001	0000
0001	0010	0010	0001

V2 M09 ACTUAL 16K CONFIG 16K

* LABORATORIO DE SISTEMAS DIGITAIS * EPUSP * 10/ 05/ 74 * 09 H 12 M

// FOR

*ONE WORD INTEGERS

*LIST SOURCE PROGRAM

*IDCS(CARD,1403PRINTER,DISK)

C * PREPARA ENTRADA PARA O LOADER, TESTANDO *
C * CARTAO DE CHAMADA *
C * ARQUIVOS - 12 E 13 - BIBLIOTECAS *
C * 11 - SAIDA DO MONTADOR *
C * 1 - SAIDA DO CARREGADOR *
C * 14 - ERROS *
C * INDICADORES - *
C * IND(17)-NUMERO DE BIBLIOTECAS *
C * (0=1 E 1=2) *
C * IND(18)-MAPA DA SAIDA *
C * IND(11)-SUBROTINA DE INTERRUP-*
C * CAD (0=SIM E 1=NAO) *
C * IND(12-13-14-15-16)-NOME DA *
C * SUBROTINA DE INT. *
C * IND(6-7-8)-NOME DA BIBL.DO PRO*
C * GRAMA *
C * IND(1-2-3-4-5)-NOME DO PROG. *
C * IND(19)-SAI EM CARTOES *

INTEGER FB(3)

DIMENSION MAT(4)

DIMENSION IP(5),IND(20),IREST(60),IBUF(320)

DIMENSION IE(400)

COMMON IVI(3000)

DEFINE FILE 1(400,80,J,N1)

DEFINE FILE 11(101,20,U,N11)

DEFINE FILE 12(48,20,U,N12)

DEFINE FILE 13(48,20,U,N13)

DEFINE FILE 14(24,40,U,N14)

EQUIVALENCE (ISEC,IND(17))

EQUIVALENCE (IVI(1),IND(1)),(IVI(21),IBUF(1))

1 ,,(IND(1),IP(1)),(IND(6),FB(1))

DATA IBK,IBK2/' ',' '/

DATA MMM/'M'/,ICC//C//

READ(2,1000)IS,FC,FB,IP,IREST

1000 FORMAT(A1,A4,2X,3A2,2X,5A1,60A1)

IND(18)=0

IND(19)=0

I=1

C BJSCA PEDIDO DE MAPA

PAGE 2

050

C
C CALL WZB1(IREST,I,M)
IF(M)10,4,10
C
C BUSCA PEDIDO SAIDA EM CARTOES

C
4 IF(IREST(I)-MMM)6,5,6
5 IND(18)=1
I=I+1
CALL WZB1(IREST,I,M)
IF(M) 10,5,10
5 IF(IREST(I)-ICC) 30,7,30
7 IND(19)=1
I=I+1

C
C BUSCA NUMERO DE BIBLIOTECAS

C
C CALL WZB1(IREST,I,M)
IF(M) 10,30,10
10 WRITE(5,1001)
1001 FORMAT(1H ,5X,'COMANDO INVALIDO')
15 CALL EXIT
30 CALL WZC1(IREST,I,N,M)
IF(M) 10,40,10
40 IF(N-2)50,60,10
50 ISEC=0
GO TO 65
60 ISEC=1

C
C BUSCA SUBROTINA DE INTERRUPCAO

C
65 CALL WZB1(IREST,I,M)
IF(M) 70,80,70
70 IND(11)=1
GO TO 125
80 DO 85 J=12,16
85 IND(J)=IBK
IND(11)=0
N=I+4
IF(N-60) 110,110,86
85 N=60
110 K=12
DO 120 J=I,N
IF(IREST(J)-IBK) 90,125,90
90 IND(K)=IREST(J)
120 K=K+1

C
125 IF(FB(1)-IBK2) 140,130,140
130 IDISK=1
ID=1
GO TO 160

C
C BUSCA PROGRAMA NA BIBLIOTECA

C
140 CALL WZBB(M,DISK)
IF(M)10,142,10

PAGE 3

142 CALL WZCDM(IP,N1,N2)
CALL WZBUS(M,IPONT,NDB,IPNT1,DISK,N1,N2)
IF(M)10, 150,10 951
150 NIT=(IPONT+320)/320
N=IPONT-(NIT-1)*320
READ(DISK,NIT) IBUF
ID=IBUF(N+4)
IE(351)=DISK
IE(301)=ID
WRITE(5,9091) IND
160 CALL WZLDA(DISK,ID)
DO 333 I=1,200,2
CALL WZZ(N,I)
CALL WZBH(N,MAT)
333 WRITE(5,3333) I,MAT
3333 FORMAT(1H ,I5,4A1)
9091 FORMAT(1H ,20I6)
CALL EXIT
END

UNREFERENCED STATEMENTS

15

FEATURES SUPPORTED
ONE WORD INTEGERS
IOCS

CORE REQUIREMENTS FOR
COMMON 3000 VARIABLES 528 PROGRAM 456

END OF COMPIILATION

// DJP

*DELETE WZV 0010
CART ID 0010 DB ADDR 1870 DB CNT 0020

*STORE WS UA WZV 0010
CART ID 0010 DB ADDR 1924 DB CNT 001F

// EJECT

NOME DO PROGRAMA

Rotina de carregamento de um programa

CÓDIGO DO PROGRAMA

WZLOA

952

LÍNGUAG EM DE ESCRITA

Fortran

TIPO DE PROGRAMA

Subrotina

SISTEMA SUPORTE E HARDWARE

IBM-1130

CHAVES DE PAINEL E SAÍDAS ALTERNATIVAS

Nenhuma

DESCRIÇÃO

Esta rotina, faz o carregamento de um programa, que pode se encontrar em uma biblioteca, em formato relocável, ou então, no arquivo de saída do ~~xxkxkxkxkxk~~ Cross-montador.

O programa principal, qualquer que seja o seu nome, é chamado pelo carregador "PRINC", e este nome aparece no mapa de carregamento quando este é requisitado. A saída deste programa pode ser feita através de cartões perfurados, ou então através de um arquivo de disco residente, dependendo de opção de cartão (ver programa principal).

As funções deste programa são:

- verificar se programa se encontra em formato relocável.
- determinar a rotina de interrupção.
- verificar tipo do programa principal e da subrotina de interrupção, realizando uma série de checagens sobre ambos.
- determinar as chamadas de subrotina do programa principal e das subrotinas de interrupção, e das outras subrotinas chamadas, até que todas se encontrem presentes na tabela de carregamento.
- gerar a tabela de interrupção,
- compor o programa e as rotinas de interrupção e demais rotinas chamadas.
- imprimir o mapa de carregamento, se pedido.
- chamar o programa de ~~xxkxkxkxkxk~~ perfuração de cartões, se pedido. Em caso contrário, gerar a saída de disco para o programa montado.
- imprimir mensagens de erro, no caso destes ocorrerem.

SEQUENCIA DE CHAMADA

CALL WZLOA(IDISK,IEND)

953

IDISK - arquivo onde se encontra o programa .

IEND - endereço de disco onde se encontra o programa .

SUB-ROTINAS CHAMADAS

WZREL

WZCDM

WZCAL

WZVER

WZENC

WZIOR

WZDS

WZAN

WZSME

WZDLE

WZLOC

WZPOE

WZMAP

WZGRA

WZWRT

OBSERVAÇÕES E MENSAGENS DE ERRO

Nenhuma

VARIAVEIS PRINCIPAIS

Nome	Tipo	Função
ITAB	I	TAbela do carregador
ITABL	I	Tabela de interrupção
IPRIN	I	Número da impressora
ITAT	I	primeira palavra em "name code" do prog. princ.
ITA	I	segunda palavra em "name code" do prog. princ.
ITOC	I	se igual a 1 saída em cartões .
ICAT	I	se igual a 1 saída de mapa de carregamento .
ISEC	I	se igual a zero há subrotina de int. especif.
IHEAD	I	cabeçalho de programa é relocável
MEMAX	I	tamanho máximo de memória
MOCUP	I	memória livre
ICOMO	I	extensão do "common" do programa principal
IE4	I	pontos de execução dos programas
IE5	I	tabela dos pontos de carregamento dos programas
IE6	I	tabela dos próximos endereços livres
IE7	I	tabela dos tipos dos programas

Nome	Tipo	Função
IE8	I	diz se um programa deve ser carregado (1) ou então é um outro ponto de entrada de uma subrotina (2).
IE9	I	tamanho em "disk blocks".
IE10	I	endereço do programa na biblioteca.
IE11	I	arquivo de biblioteca onde está o programa.
IPNT	I	ponteiro para a tabela de carregamento.
M	I	indica se diferente de zero ocorrência de erro.
IPROX	I	indica próximo endereço livre na memória.
M6	I	número de canal de uma subrotina.
ICARG	I	endereço de carregamento de um programa.
MM	I	indicador de relocação.
ICONT, ND	I	contadores.
IPONT, IL	I	ponteiros.
IDCAR	I	endereço de carregamento de uma palavra na memória.

PAGE 1

055

// JOB 0001 0010

* E102, MALICE

LOG DRIVE	CART SPEC	CART AVAIL	PHY DRIVE
0000	0001	0001	0000
0001	0010	0010	0001

V2 M09 ACTUAL 16K CONFIG 16K

* LABORATORIO DE SISTEMAS DIGITAIS * EPUSP * 18/ 06/ 74 * 11 H 55 MIN *

// FOR

*ONE WORD INTEGERS

*LIST SOURCE PROGRAM

```
SUBROUTINE WZLOAD(IDISK,IEND)
INTEGER BUF(320)
DIMENSION IPA(2)
DIMENSION IND(20),IBUF(320),MAT(5),IHEAD(60), IE(400), IE4(50),
, IE5(50),IE7(50),IE8(50),IE9(50),IE10(50), ITABL(16)
DIMENSION ISUB(2),IE6(50), ITAB(100)
DIMENSION IE11(50)
COMMON IVI(3000)
EQUIVALENCE (IPA(1),NN1),(IPA(2),NN2)
EQUIVALENCE (IND(1),IVI(1)),(IBUF(1),IVI(21)), (IVI(661),
1 IHEAD(1))
EQUIVALENCE (IE(1),IE4(1)),(IE(51),IE5(1)),(IE(101),IE6(1)),
1 (IE(151),IE7(1)),(IE(201),IE8(1)),(IE(251),IE9(1)),
2 (IE(301),IE10(1)),(IE(351),IE11(1))
DATA ITABL/16*0/,IPUG/ZF000/,ITABL/16*0/
DATA ITAT,ITA/Z1764,Z9543/
DATA IPRIN,IV,MEMAX/5,13,4095/
ITAB(1)=ITAT
ITAB(51)=ITA
DO 3333I=1,50
3333 IE8(I)=0
```

C
C TRATA DO CARREGAMENTO DO PROGRAMA
C
C

```
ITOC=IND(18)
ICAT=IND(19)
ISEC=IND(17)
DO 500 I=1,5
500 MAT(I)=IND(I)
READ(IDISK*IEND) IHEAD
```

C
C VERIFICA CABECALHO P/ RELOCAVEL
CALL WZREL(M,IK)
IF(M) 501,502,501
501 WRITE(IPRIN,109)
109 FORMAT(1H0,5X,'PROGRAMA PEDIDO NAO SE ENCONTRA NO ARQUIVO')
RETURN

C
C FORMAT RELOCAVEL
C

```
502 I3=IHEAD(3)/256
MEMAX=MEMAX-IHEAD(5)-128
ICOMO=IHEAD(5)
```

PAGE 2

056

CALL WZCDM(MAT,NN1,NN2)
IE7(1)=I3
IF(I3-2) 520,540,550

C
C ABSOLUTO
C

520 IE4(1)=IHEAD(12)
IE5(1)=IHEAD(13)
GO TO 660

C
C
C RELOCAVEL
C

540 IE4(1)=IHEAD(12)
IE5(1)=0
GO TO 660
550 IF(I3 -4) 540, 15,570
570 IF(I3-6) 501, 15,501
660 IE6(1)=IHEAD(4)
IE9(1)=IHEAD(8)
IE8(1)=1
IE10(1)=IEND
IE11(1)=IDISK

C
C INTERRUPCAO
C

IF(IND(11)) 670,680,670
670 INTD=0
GO TO 690
680 INTD=1
DO 681 I=1,5
681 MAT(I)=IND(I+11)
CALL WZCDM(MAT,N1,N2)
ITAB(2)=N1
ITAB(52)=N2
IPNT=2
IM=1
GO TO 695

C
C NAO HA ESPECIFICACAO DE INTERRUPCAO
C

690 IPNT=1
IM=1

C
C VAI BUSCAR CALLS DO PROG.PRINCIPAL E SUBROTINAS
C

695 IDK=IDISK
CALL WZCAL(IDK ,IHEAD(8),IEND,ITAB,IPNT,M)
IF(M) 691,700,691
700 IF(IPNT-IM) 710,740,710
710 IM1=IM+1
IM=IPNT
CALL WZVER(IPNT,M,ITAB,IBUF,ISEC,DISK,IV,IE)
IF(M)691,720,691
691 RETURN

C
C

PAGE 3

057

C
720 DO 730 I=IM1,IM
IF(IE8(I)-1) 730,722,730
722 CALL WZCAL(IE11(I),IE9(I),IE10(I),ITAB,IPNT,M)
IF(M)691,730,691
730 CONTINUE
GO TO 700
740 CONTINUE

C
C
IF(IE7(1)-1) 10,30,10
10 IF(IE7(1)-2)15,20,15
15 M=6
GO TO 3000

C
C PROGRAMA PRINCIPAL RELOCABEL
C

20 IE4(1)=IE4(1)+10
IE5(1)=IE5(1)+10
IE6(1)=IE6(1)+10
GO TO 60

C
C PROGRAMA PRINCIPAL ABSOLUTO
C

30 IF(IE5(1)-(IE5(1)/2)*2)35,40,35
35 M=3
GO TO 3000
40 IF(IE5(1)-4) 45,50,60
45 M=4
GO TO 3000
50 IF(INTD) 55,60,55
55 M=5
GO TO 3000
60 IPROX=IE6(1)
IF(INTD) 73,90,73

C
C TRATA SUBROTINA DE INTERRUPCAO
C

73 IF(IE7(2)+4) 75,80,75
75 M=7
GO TO 3000
80 IE4(2)=IE4(2)+IPROX
IE5(2)=IE5(2)+IPROX
IE6(2)=IE6(2)+IPROX
IE8(2)=1
6011 FORMAT(1X,20I6)
IPROX=IE6(2)
IPN=3
GO TO 100
90 IPN=2

C
C *SUBROTINAS*

C
100 IF(IPN-IPNT) 99,99,300
99 DO 120 I=IPN,IPNT
IF(IE7(I))112,114,114
112 IF(IE7(I)+4) 130,118,130

PAGE 4

114 M=16
GO TO 3000
113 IF(IE8(I)-2) 115,120,115
115 IE4(I)=IE4(I)+IPROX
IE5(I)=IE5(I)+IPROX
IE6(I)=IE6(I)+IPROX
IPROX= IE6(I)
120 CONTINUE
CALL WZENC(IE,IPNT,M)
IF(IPROX-MEMAX) 125,125,121
121 M=11
GO TO 3000
125 MCUP=MEMAX-IPROX
GO TO 300

058

C
C E/S
C

130 MC=-IE7(I)
MC1=MC/256
MR=MC-MC1*256
MC=MC1
IF(MR-6) 133,141,133
133 M=7
GO TO 3000
141 IF(MC)150,160,160
150 M=8
GO TO 3000
160 IF(MC-16) 170,150,150
170 J=MC+1
IF(ITABL(J)) 180,190,180
180 M=9
GO TO 3000

C
C

190 MN=IE4(I)+4+IPROX
CALL WZIDR(MN,IPUG)
ITABL(J)=MN
IF(INTD) 118,200,118
200 M=10
GO TO 3000

C
C TABELA PRONTA - DEVE-SE LOCAR OS PROGRAMAS
C

300 CONTINUE
DO 1100 J=1,IPNT
IF(IE8(J)-1) 1100,1010,1100
1010 IDK=IE11(J)
IN=IE10(J)
ND=IE9(J)
IF(ND-16) 1013,1015,1015
1013 K=ND*20
GO TO 1017
1015 K=320
1017 READ(IDK*IN)(BUF(I),I=1,K)
JJ=BUF(6)+9
IF(BUF(5)-ICOMO) 5001,5001,5000
5000 M=24

PAGE 5

059

GO TO 3000
5001 ICARG=IE5(J)
ISUB(1)=0
IDCAR=IE5(J)
IPONT=BUF(6)+10
IRELC=0
ICONT=BUF(IPONT+1)
IPONT=IPONT+2
1019 IF(ICONT) 1020,1100,1020
1020 ICONT=ICONT-2
IF(ICONT) 2600,2600,1021
1021 ICONT=ICONT-1
2047 IF(ICONT-8) 1024,1022,1022
1024 IX=ICONT
ICONT=0
GO TO 1025
2600 IL=IPONT
IL=IL-1
GO TO 2100
1022 IX=8
ICONT=ICONT-8
1025 IK0=BUF(IPONT)
IL=IPONT+1
2000 IF(IRELC) 1032,1030,1032
1032 IRELC=0
IDCAR=IDCAR+1
CALL WZDS(MM,IK0)
GO TO 1070
1030 IF(ISUB(1)) 1050,1040,1050
1040 CALL WZDS(MM,IK0)
IF(MM) 1051,1042,1051
C
C DADO ABSOLUTO
C
1042 JC=BUF(IL)
CALL WZAN(JC,-256)
CALL WZSME(JC, IDCAR)
IDCAR=IDCAR+1
JC=BUF(IL)
CALL WZDLE(JC,8)
IF(JC) 1036,1070,1036
1035 IRELC=JC
CALL WZSME(JC, IDCAR)
GO TO 1070
C
C RELOCABEL
C
1051 IF(MM-1) 1060,1055,1060
1055 JC=BUF(IL)+ICARG
IRELC=JC
CALL WZAN(JC,-256)
CALL WZSME(JC, IDCAR)
IDCAR=IDCAR+1
CALL WZDLE(IRELC,8)
CALL WZSME(IRELC, IDCAR)
GO TO 1070
C

PAGE 6

060

C CHAMA SUBROTINA
C
1060 IF(MM=3) 1065,1068,1065
1065 M=13
GO TO 3000
1068 JC=BUF(IL)
IDCAR=IDCAR+1
ISUB(1)=JC
1070 IX=IX-1
IF(IX) 1085,1090,1085
1085 IL=IL+1
GO TO 2000
1090 IF(ICONT) 2100,2100,2150
2100 IF(IL=319) 2200,2200,2300
2150 IL=IL+1
ICONT=ICONT-1
IPONT=IL
IF(ICONT) 2100,2100,2047
2200 IL=IL+1
IDCAR=ICARG+BUF(IL)
ICONT=BUF(IL+1)
IL=IL+2
IPONT=IL
GO TO 1019
C
C VAI LER MAIS UM BLOCO
C
2300 ND=ND-16
IF(ND=16) 2310,2320,2320
2310 K=ND*20
GO TO 2340
2320 K=320
2340 READ(IDK*IN)(BUF(I),I=1,K)
IL=1
IPONT=1
GO TO 2200
C
C
1050 ISUB(2)=BUF(IL)
CALL WZLOC(IDCAR,ITAB,IPNT,IE4,ISUB,M)
ISUB(1)=0
IF(M) 3000,2400,3000
2400 CALL WZDS(MM,IKD)
GO TO 1070
C
C
1100 CONTINUE
C
C VERIFICA TABELA DE INTERRUPCAO
C
IF(INTD) 4000,4020,4000
4000 I=16
4005 IF(ITABL(I)) 4020,4010,4020
4010 I=I-1
IF(I) 4020,4020,4005
4020 N=IE4(2)-IE5(2)
IF(N-(N/2)*2) 4030,4040,4030

PAGE 7

961

```
4030 M=14
      GO TO 3000
4040 N=IE5(2)/2 -1
      IF(N) 4030,4030,4050
4050 N=IE4(2)+2
      DO 4051 J=1,I
      IRELC=ITABL(J)
      CALL WZSME(IRELC,N)
      M=N+1
      IRELC=ITABL(J)
      CALL WZDLE(IRELC,8)
      CALL WZSME(IRELC,M)
      GO TO 4051
4051 N=N+2
      CALL WZPDE(IE4(1),IE5(1),IE4(2),IPROX)
      IVI(2999)=IPROX
      IVI(3000)=IE5(1)
C      IMPRESSAO DO MAPA DE MEMORIA
C
      IF(ITOC-1) 4401,4300,4401
4300 CALL WZMAP(MOCUP,INTD,IE4, BUF,IPNT,ITAB,ITABL)
4401 IF(ICAT-1) 4403,4406,4403
4405 CALL LINK(WZSDS)
4403 CALL WZGRA(IPROX,IPA,ICOMO,IE4(1),INTD,IE4(2),BUF,IE5(1))
      RETURN
C
C
C
3000 CALL WZWRT(M)
      RETURN
      END
```

UNREFERENCED STATEMENTS

6011

FEATURES SUPPORTED
ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZLOA
COMMON 3000 VARIABLES 896 PROGRAM 1716

RELATIVE ENTRY POINT ADDRESS IS 0384 (HEX)

END OF COMPIILATION

// DUP

*DELETE WZLOA 0010
CART ID 0010 DB ADDR 1A4C DB CNT 006B

*STORE WS UA WZLOA 0010
CART ID 0010 DB ADDR 1AD2 DB CNT 0068

// EJECT

NOME DO PROGRAMA

Rotina de preenchimento da tabela do carregador.

CÓDIGO DO PROGRAMA

WZVER

062

LINGUAGEM USADA

Fortran

TIPO DE PROGRAMA

Subrotina

SISTEMA SUPORTE E HARDWARE

IBM-1130

CHAVES DE PAINEL E SAÍDAS ALTERNATIVAS

Nenhuma

DESCRIÇÃO

Esta rotina preenche a tabela do carregador. Este tem o seguinte formato:

ITAB	IE4	IE5	IE6	IE7	IE8	IE9	IE10	IE11

→ biblioteca de disco
→ endereço em "disk block"
nesta biblioteca
→ extensão em "disk blocks"
→ endereço de carregamento do disco
→ tipo do programa
→ próximo endereço do programa
→ endereço de carregamento
→ ponto de entrada
→ nome do programa

A rotina descobre em que arquivo se encontra o programa, a certando a tabela de carga. Do índice da biblioteca retira todas as informações necessárias ao carregamento.

SEQUENCIA DE CHAMADA

CALL WZVER(IPNT,M,ITAB,IBUF,ISEC,DISK,IV,IE)

IPNT - ponteiro para a tabela de carregamento

M - indica se diferente de zero a ocorrência de erro

ITAB - tabela de carregamento

IBUF - buffer de leitura

ISEC - indica se igual a zero que somente uma biblioteca se encontra no sistema

IDISK - primeira biblioteca de disco

IV - segunda biblioteca de disco

IE - tabela do carregador (complemento)

SUB-ROTINAS USADAS

WZBUS

WZCDM

WZTB

OBSERVACÕES E MENSAGENS DE ERRO

Pode surgir a seguinte mensagem:

"PROGRAMA PPPPP NAO FIGURA NA BIBLIOTECA"

uma das subrotinas exigidas para carregamento não está na biblioteca

VARIÁVEIS PRINCIPAIS

Nome	Tipo	Função
IPNT	I	ponteiro para a tabela de carregamento
ITAB	I	tabela de carregamento

PAGE 13

```
// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
    SUBROUTINE WZVER(IPNT,M,ITAB,IBUF,ISEC,DISK,IV,IE)
    DIMENSION IE(1)
    DIMENSION ITAB(1),IBUF(1),IC(2)
    DATA IPRIN/5/
    I=1
    99 IF(IE(I+200)-1)120,100,100
    100 I=I+1
        IF(I-IPNT) 99,99,101
    101 M=0
        RETURN
C
C
    120 N1=ITAB(I)
        N2=ITAB(I+50)
        CALL WZBUS(M,IPONT,NDB,IPNT1,DISK,N1,N2)
        IT=DISK
        IF(M) 130,150,130
    130 IF(ISEC) 140,135,140
    140 CALL WZBUS(M,IPONT,NDB,IPNT1,IV,N1,N2)
        IT=IV
        IF(M) 135,150,135
    135 M=21
        CALL WZDCM(N1,N2,N3)
        WRITE(IPRIN,1000) N1,N2,N3
    1000 FORMAT(1H0,5X,'PROGRAMA ',3A2, 'NAO FIGURA NA BIBLIOTECA')
    149 RETURN
C
C
    150 K=(IPONT+20)/20
        READ(IT'K)(IBUF(J),J=1,120)
        K=IPONT-(K-1)*20
    151 IC(1)=IBUF(K)
        IC(2)=IBUF(K+1)
        CALL WZTB(IC,IPNT,M,ITAB,KL)
        IF(M) 200,300,200
    200 M=21
        RETURN
    300 IE(KL) =IBUF(K+5)
        IE(KL+50)=IBUF(K+6)
        IE(KL+100)=IBUF(K+7)
        IE(KL+150)=IBUF(K+2)
        IE(KL+200)=2
        IE(KL+250)=NDB
        IE(KL+300)=IBUF(K+4)
        IE(KL+350)=IT
        IF(IPONT-IPNT1)301,220,220
    301 K=K+8
        IPONT=IPONT+8
        GO TO 151
    220 IE(KL+200)=1
        GO TO 100
        END
```

PAGE 14

FEATURES SUPPORTED
ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZVER
COMMON 0 VARIABLES 28 PROGRAM 412

RELATIVE ENTRY POINT ADDRESS IS 003C (HEX)

END OF COMPIRATION

// DUP

*DELETE WZVER 0010
CART ID 0010 DB ADDR 1ACB DB CNT 001D

*STORE WS UA WZVER 0010
CART ID 0010 DB ADDR 1B23 DB CNT 001A

// EJECT

NOME DO PROGRAMA

Rotina de tratamento de pontos de entrada multiplos

CODIGO DO PROGRAMA

WZENC

LINGUAGEM USADA

Fortran

966

TIPO DE PROGRAMA

Subrotina

SISTEMA SUPORTE E HARDWARE

IBM-1130

CHAVES DE PAINEL E SAIDAS ALTERNATIVAS

Nenhuma

DESCRICAÇÃO

No caso em que se tem de carregar uma rotina que possui múltiplos pontos de carregamento, todos os pontos de entrada são incluídos na tabela de carregador. Entretanto somente uma vez esta rotina deve ser incluída na tabela. Um dos seus pontos de entrada indica então que ela deve ser carregada e os demais, servem somente para indicarem opção de entrada, e indicam "não carregamento". Esta rotina identifica estes pontos de entrada múltiplos, acertanto a tabela de acordo com as características daquele que será carregado.

SEQUENCIA DE CHAMADA

CALL WZENC(IE,IPNT,M)

IE - Tabela de complementação da tabela do carregador

IPNT - ponteiro para a tabela do carregador

M - indica se diferente, de zero, a ocorrência de erro

SUB-ROTINAS USADAS

Nenhuma

OBSERVAÇÕES E MENSAGENS DE ERRO

Nenhuma

VARIAVEIS PRINCIPAIS

Nenhuma

PAGE 1

// JOB 0001 0010

Q67

E102,M

LOG DRIVE	CART SPEC	CART AVAIL	PHY DRIVE
0000	0001	0001	0000
0001	0010	0010	0001
		400F	0002

V2 M09 ACTUAL 16K CONFIG 16K

* LABORATORIO DE SISTEMAS DIGITAIS * EPUSP * 22/ 04/ 74 * 12 H 10 M

// FOR

*ONE WORD INTEGERS

*LIST ALL

SUBROUTINE WZENC(IE,IPNT,M)

DIMENSION IE(1)

DO 10 J=1,IPNT

IF(IE(J+200)-2) 10,20,10

10 CONTINUE

RETURN

20 L=IE(J+350)

N=IE(J+300)

DO 40 JJ=1,IPNT

IF(J-JJ) 30,40,30

40 CONTINUE

M=22

RETURN

30 IF(L-IE(JJ+350)) 40,31,40

31 IF(N-IE(JJ+300)) 40,32,40

32 IE(J+100)=IE(JJ+100)

LL=IE(JJ+50)-IE(J+50)

IE(J+50)=IE(JJ+50)

IE(J)=IE(J)+LL

RETURN

END

VARIABLE ALLOCATIONS

J(I)=0000

L(I)=0001

N(I)=0002

JJ(I)

STATEMENT ALLOCATIONS

10 =0032 20 =003D 40 =0059 30 =0068 31 =0073 32 =00

FEATURES SUPPORTED

ONE WORD INTEGERS

CALLED SUBPROGRAMS

SJBSC SJBIN

INTEGER CONSTANTS

1=000A

2=000B

22=000C

CORE REQUIREMENTS FOR WZENC

COMMON 0 VARIABLES 10 PROGRAM 174

RELATIVE ENTRY POINT ADDRESS IS 000D (HEX)

PAGE 2

END OF COMPILATION

// DJP

068

*DELETE WZENC 0010
CART ID 0010 DB ADDR 1830 DB CNT 0008

*STORE WS JA WZENC 0010
CART ID 0010 DB ADDR 1830 DB CNT 0008

// EJECT

NOME DO PROGRAMA

Programa de deslocamento e mascaramento

969

CÓDIGO DO PROGRAMA

WZDLE

WZAN

LINGUAGEM USADA

Assembler

TIPO DE PROGRAMA

Subrotina

SISTEMA SUPORTE E HARDWARE

IBM-1130

CHAVES DE PAINEL E SAÍDAS ALTERNATIVAS

Nenhuma

DESCRIÇÃO

O primeiro programa, desloca um determinado dado para a esquerda, de um número de posições especificado.

O segundo, realiza a operação lógica AND entre dois dados deixando o resultado no primeiro.

SEQUENCIA DE CHAMADA

CALL WZDLE(IDAD,IDES)

IDAD - dado a ser deslocado

IDES - numero de deslocamentos

CALL WZAN (IDAD,IMASC)

IDAD - dado a ser mascarado

IMASC - máscara

SUB-ROTINAS USADAS

Nenhuma

OBSERVAÇÕES E MENSAGENS DE ERROS

Nenhuma

VARIAVEIS PRINCIPAIS

Nenhuma

PAGE 35

070

// ASM
*LIST
0000 26A444C5 00001 ENT WZDLE
0011 26A41540 00002 ENT WZAN
0000 0 0000 00003 WZDLE DC **-*
0001 01 65800000 00004 LDX I1 WZDLE
0003 00 C5800001 00005 LD I1 1
0005 00 D4000002 00006 STD L 2
0007 00 C5800000 00007 LD I1 0
0009 0 1200 00008 SLA 2
000A 00 D5800000 00009 FIM STD I1 0
000C 0 7102 00010 MDX I 2
000D 0 6901 00011 STX I RET+1
000E 00 4C000000 00012 RET BSC L 0
0010 0 0000 00013 TEMP DC 0
0011 0 0000 00014 WZAN DC **-*
0012 01 65800011 00015 LDX I1 WZAN
0014 00 C5800000 00016 LD I1 0
0016 00 E5800001 00017 AND I1 1
0018 0 70F1 00018 MDX FIM
001A 00019 END

000 OVERFLOW SECTORS SPECIFIED

000 OVERFLOW SECTORS REQUIRED

005 SYMBOLS DEFINED

NO ERROR(S) AND NO WARNING(S) FLAGGED IN ABOVE ASSEMBLY

// DJP

*DELETE WZDLE 0010
D 26 NAME NOT FOUND IN LET/FLET

*STORE WS JA WZDLE 0010
CART ID 0010 DB ADDR 0284 DB CNT 0003

// EJECT

NOME DO PROGRAMA

Programa para determinar o conteúdo de uma determinada posição de memória

CÓDIGO DO PROGRAMA

WZZ

LINGUAGEM USADA

Assembler

TIPO DE PROGRAMA

Subrotina

SISTEMA SUPORTE E HARDWARE

IBM-1130

CHAVES DE PAINEL E SAÍDAS ALTERNATIVAS

Nenhuma

DESCRIÇÃO

Este programa coloca em uma dada palavra o conteúdo de uma posição de memória calculada a partir do seguinte:

endereço = /7800. + endereço fornecido.

SEQUÊNCIA DE CHAMADA

CALL WZZ(IDADO, IESN)

IDADO - palavra que conterá o resultado

IESN - endereço fornecido

SUB-ROTINAS USADAS

Nenhuma

OBSERVAÇÕES E MENSAGENS DE ERRO

Nenhuma

VARIÁVEIS PRINCIPAIS

Nenhuma

PAGE 36

072

// ASM
*LIST

0000	26A69000	00001		ENT	WZZ
0000	0	00002	WZZ	DC	0
0001	01	55800000	00003	LDX	I1 WZZ
0003	00	C5800001	00004	LD	I1 1
0005	0	1801	00005	SRA	1
0006	0	8009	00006	A	H7800
0007	0	D001	00007	STD	FIT+1
0008	00	C4000000	00008	FIT	LD L 0
000A	00	D5800000	00009	STD	I1 0
000C	01	74020000	00010	MDM	L WZZ,2
000E	01	4C800000	00011	B	I WZZ
0010	0	7800	00012	H7800	DC /7800
0012			00013	END	

000 OVERFLOW SECTORS SPECIFIED

000 OVERFLOW SECTORS REQUIRED

003 SYMBOLS DEFINED

NO ERROR(S) AND NO WARNING(S) FLAGGED IN ABOVE ASSEMBLY

// DJP

*DELETE WZZ 0010
D 26 NAME NOT FOUND IN LET/FLET

*STORE WS UA WZZ 0010
CART ID 0010 DB ADDR 0287 DB CNT 0002

// EJECT

NOME DO PROGRAMA

Rotina para armazenar um byte de informação em um determinado endereço.

CÓDIGO DO PROGRAMA

WZSME

LINGUAGEM USADA

Assembler

TIPO DE PROGRAMA

Subrotina

SISTEMA SUPORTE E HARDWARE

IBM-1130

CHAVES DE PAINEL E SAÍDAS ALTERNATIVAS

Nenhuma

DESCRIÇÃO

O programa armazena um byte de informação em um endereço correspondente, sendo este endereço calculado por um deslocamento em relação ao endereço /7800. Se o endereço é ímpar, o dado é armazenado no byte da direita, e se for par, no byte da esquerda. O byte vem sempre no byte da esquerda.

SEQUENCIA DE CHAMADA

CALL WZSME(IDADO, IDCAR)

IDADO - dado recebido

IDCAR - endereço

SUB-ROTINAS USADAS

Nenhuma

OBSERVAÇÕES E MENSAGEM DE ERRO

Nenhuma

VARIAVEIS PRINCIPAIS

Nenhuma

```

// ASM
*LIST
0000 26A62505 00001      ENT      WZSME
0000 0 0000 00002      WZSME  DC      *-* 
0001 0 D836 00003      STD      TEMP
0002 0 692A 00004      STX      1 SV1+1
0003 0 6A2B 00005      STX      2 SV2+1
0004 0 6B2C 00006      STX      3 SV3+1
0005 01 65800000 00007      LDX      I1 WZSME
0007 00 C5800000 00008      LD       I1 0
0009 0 D02D 00009      STD      GUARD
0004 00 C5800001 00010      LD       I1 1      *CARREGA ADCAR
000C 01 4C040019 00011      BOD      IMPAR
000E 0 1801 00012      SRA      1
000F 0 8025 00013      A       H7800
0010 0 D025 00014      STD      ADDRS *ARMAZENA EM ADDRS
0011 01 C4800036 00015      LD       I  ADDRS
0013 0 1008 00016      SLA      8
0014 0 1808 00017      SRA      8
0015 0 E821 00018      DR      GUARD
0016 01 D4800036 00019      STD      I  ADDRS
0018 0 700F 00020      MDX      FIM
0019 0 1801 00021      IMPAR  SRA      1
001A 0 801A 00022      A       H7800
001B 0 D01A 00023      STD      ADDRS
001C 0 C01A 00024      LD      GUARD
001D 0 1808 00025      SRA      8
001E 0 D018 00026      STD      GUARD
001F 01 C4800036 00027      LD       I  ADDRS
0021 0 1808 00028      SRA      8
0022 0 1008 00029      SLA      8
0023 0 E813 00030      DR      GUARD
0024 01 D4800036 00031      STD      I  ADDRS
0026 0 C010 00032      LD      GUARD
0027 0 1008 00033      SLA      8
0028 00 D5800000 00034      FIM      STD  I1 0
002A 01 74020000 00035      MDM      L  WZSME,2
002C 00 65000000 00036      SV1      LDX  L1 0
002E 00 66000000 00037      SV2      LDX  L2 0
0030 00 67000000 00038      SV3      LDX  L3 0
0032 0 C805 00039      LDD      TEMP
0033 01 4C800000 00040      BSC      I  WZSME
0035 0 7800 00041      H7800  DC      /7800
0036 0 0000 00042      ADDRS DC      0
0037 0 0000 00043      GUARD DC      0
0038 0 0000 00044      TEMP   DC      0
003A          00045      END

```

000 OVERFLOW SECTORS SPECIFIED

000 OVERFLOW SECTORS REQUIRED

010 SYMBOLS DEFINED

NO ERROR(S) AND NO WARNING(S) FLAGGED IN ABOVE ASSEMBLY

// DJP

*DELETE

WZSME

0010

PAGE 34

D 26 NAME NOT FOUND IN LET/FLET

075

*STORE WS UA WZSME 0010
CART ID 0010 DB ADDR 027F DB CNT 0005

// EJECT

NOME DO PROGRAMA

076

Programa para montar uma instrução de PUG

CÓDIGO DO PROGRAMA

WZCOL

LINGUAGEM USADA

Assembler

TIPO DE PROGRAMA

Subrotina

SISTEMA SUPORTE E HARDWARE

IBM-1130

CHAVES DE PAINEL E SAÍDAS ALTERNATIVAS

Nenhuma

DESCRIÇÃO

Esta rotina coloca no buffer de memória uma instrução de "PUG end".

SEQUENCIA DE CHAMADA

CALL WZCOL(N, IDCAR)

N - palavra a ser colocada no buffer

IDCAR - endereço a ser preenchido + 1

SUB-ROTINAS USADAS

Nenhuma

OBSERVAÇÕES E MENSAGENS DE ERRO

Nenhuma

VARIAVEIS PRINCIPAIS

Nenhuma

PAGE 3

// ASM
*LIST 977

0000	26A43593	00001	ENT	WZCOL
0000 0	0000	00002	WZCOL	DC 0
0001 01	65800000	00003	LDX	I1 WZCOL
0003 03	C5000000	00004	LD	L1 0
0005 0	D00D	00005	STD	SME1
0006 0	D01A	00006	STD	SME2
0007 00	C5000001	00007	LD	L1 1
0009 0	D00A	00008	STD	SME1+1
000A 0	D017	00009	STD	SME2+1
000B 00	C5800000	00010	LD	I1 0 *CARREGA DADO
000D 0	18D8	00011	RTE	24
000E 00	D5800000	00012	STD	I1 0
0010 0	D01D	00013	STD	GUARD
0011 30	26A62505	00014	CALL	WZSME
0013 0	0000	00015	SME1	DC 0
0014 0	0000	00016		DC 0
0015 01	C4800014	00017		LD I SME1+1
0017 0	9015	00018		S UM
0018 01	D4800014	00019		STD I SME1+1
001A 0	C013	00020		LD GUARD
001B 0	1008	00021		SLA 8
001C 0	18C8	00022		RTE 8
001D 00	D5800000	00023		STD I1 0
001F 30	26A62505	00024		CALL WZSME
0021 0	0000	00025	SME2	DC 0
0022 0	0000	00026		DC 0
0023 01	C4800014	00027		LD I SME1+1
0025 0	8006	00028		A DDIS
0026 01	D4800014	00029		STD I SME1+1
0028 01	74020000	00030	FIM	MDM L WZCOL,2
002A 01	4C800000	00031		BSC I WZCOL
002C 0	0002	00032	DDIS	DC 2
002D 0	0001	00033	UM	DC 1
002E 0	0000	00034	GJARD	DC 0
0030		00035		END

000 OVERFLOW SECTORS SPECIFIED

000 OVERFLOW SECTORS REQUIRED

007 SYMBOLS DEFINED

NO ERROR(S) AND NO WARNING(S) FLAGGED IN ABOVE ASSEMBLY

// DJP

*DELETE WZCOL 0010
D 26 NAME NOT FOUND IN LET/FLET

*STORE WS JA WZCOL 0010
CART ID 0010 DB ADDR 0186 DB CNT 0004

// EJECT

NOME DO PROGRAMA

Rotina para geração da instrução de PUG

CÓDIGO DO PROGRAMA

WZLOC

978

LINGUAGEM USADA

Fortran

TIPO DE PROGRAMA

Subrotina

SISTEMA SUPORTE E HARDWARE

IBM-1130

CHAVES DE PAINEL E SAIDA ALTERNATIVA

Nenhuma

DESCRIÇÃO

Esta rotina trata uma pseudo CALL, gerando uma instrução PUG. A pseudo CALL tem a forma:

CALL nome

Esta pseudo é identificada através de um código de relocação conveniente : 3, seguido de "nome", codificado em "name code". É gerado em seu lugar:

PUG nome

com "nome" convenientemente traduzido por seu endereço.

Para isso "nome" é pesquisado na tabela do carregador e seu endereço determinado.

SEQUENCIA DE CHAMADA

CALL WZLOC(IDCAR,TAB,IPNT,IE4,R,M)

TAB - tabela de carregamento

IDCAR - controle da saída- endereço onde colocar a inst

IPNT - ponteiro para a tabela de carregamento

IE4 - endereço das rotinas em TAB

R - nome buscado

M - indica se diferente de zero, que ocorreu erro.

SUB-ROTINAS USADAS

WZIOR

WZCOL

OBSERVAÇÕES E MENSAGENS DE ERRO

Nenhuma

VARIAVEIS PRINCIPAIS

Nenhuma

PAGE 1

// JCB 0001 0010

079 E102, MALL

LOG DRIVE	CART SPEC	CART AVAIL	PHY DRIVE
0000	0001	0001	0000
0001	0010	0010	0001

V2 M09 ACTUAL 15K CONFIG 16K

* LABORATORIO DE SISTEMAS DIGITAIS * EPUSP * 15/ 04/ 74 * 10 H 29 MIN

// FOR

*LIST ALL

*ONE WORD INTEGERS

SUBROUTINE WZLOC(IDCAR,ITAB,IPNT,IE4,ID,M)

DIMENSION ITAB(1),IE4(1),ID(2)

DATA IPJG/ ZF000/

DO 50 J=1,IPNT

IF(ITAB(J)-ID(1))11,10,11

11 IF(ITAB(J+50)-ID(2)) 50,10,50

10 N=IE4(J)

CALL WZIDR(N,IPUG)

CALL WZCOL(N, IDCAR)

RETURN

50 CONTINUE

M=12

RETURN

END

VARIABLE ALLOCATIONS

J(I)=0000 N(I)=0001 IPUG(I)=0002

STATEMENT ALLOCATIONS

11 =002A 10 =0037 50 =004A

FEATURES SUPPORTED

ONE WORD INTEGERS

CALLED SUBPROGRAMS

WZIDR WZCOL SJBSC SUBIN

INTEGER CONSTANTS

1=0006 12=0007

CORE REQUIREMENTS FOR WZLOC

COMMON 0 VARIABLES 6 PROGRAM 84

RELATIVE ENTRY POINT ADDRESS IS 0008 (HEX)

END OF COMPIILATION

// DJP

*DELETE WZLOC 0010

D 26 NAME NOT FOUND IN LET/FLET

*STORE WS JA WZLOC 0010

PAGE 2

CART ID 0010 DB ADDR 0180 DB CNT 0006

980

// EJECT

NOME DO PROGRAMA

Rotina de determinação de comandos "CALL"

081

CÓDIGO DO PROGRAMA

WZBC

LINGUAGEM USADA

Fortran

TIPO DE PROGRAMA

Subrotina

SISTEMA SUPORTE E HARDWARE

IBM-1130

CHAVES DE PAINEL E SAÍDAS ALTERNATIVAS

Nenhuma

DESCRIÇÃO

Este programa busca códigos de relocação 3, de tal maneira, que ao detetar um, extraí o valor codificado em seguida (nome do programa em "name code"), e o inclui na tabela de carregamento.

SEQUÊNCIA DE CHAMADA

CALL WZBC(IPNT,M,IB,N,TAB)

IPNT - ponteiro para a última posição da tabela de carregamento.

M - parâmetro de erros, indicando erro se diferente de zero.

IB - matriz de 9 posições, contendo as palavras de um bloco de dados.

N - posição de próximo bloco de dados (primeira após o indicador de relocação).

TAB - tabela de relocação (carregamento).

SUB-ROTINAS USADAS

WZTB

WZDS

OBSERVAÇÕES

Nenhuma.

VARIÁVEIS PRINCIPAIS

Nome	Tipo	Função
------	------	--------

M	I	se igual a 0 não ocorreu erro.
---	---	--------------------------------

L	I	indicador de relocação
---	---	------------------------

IC	I,()	matriz que conterá nome da rotina a ser passada a rotina de busca em tabela.
----	------	--

PAGE 37

// FOR
*ONE WORD INTEGERS
*LIST ALL
SUBROUTINE WZBC(IPNT,M,IB,N,ITAB) 082
DIMENSION IB(9),IC(2),ITAB(1)
COMMON ITJDD(1430)
DO 10 J=2,9
CALL WZDS(L,IB(1))
IF(L-3)10,20,10
10 CONTINUE
15 RETURN
20 IC(1)=IB(J)
IF(J-9) 25,30,30
25 IC(2)=IB(J+1)
J=J+1
17 CALL WZTB(IC,IPNT,M,ITAB,KL)
CALL WZDS(L,IB(1))
IF(M) 15,10,15
30 IC(2)=N
J=10
GO TO 17
END

VARIABLE ALLOCATIONS

ITJDD(IC)=7FFF-7A6A IC(I)=0001-0000 J(I)=0002 L(I)=000

STATEMENT ALLOCATIONS

10 =0035 15 =003E 20 =0040 25 =0051 17 =0062 30 =007A

FEATURES SUPPORTED
ONE WORD INTEGERS

CALLED SUBPROGRAMS

WZDS WZTB SUBSC SUBIN

INTEGER CONSTANTS

2=0008 9=0009 3=000A 1=000B 10=000C

CORE REQUIREMENTS FOR WZBC

COMMON 1430 VARIABLES 8 PROGRAM 126

RELATIVE ENTRY POINT ADDRESS IS 0000 (HEX)

END OF COMPILATION

// DJP

*DELETE WZBC 0010
D 26 NAME NOT FOUND IN LET/FLET

*STORE WS JA WZBC 0010
CART ID 0010 DB ADDR 0289 DB CNT 0009

// EJECT

NOME DO PROGRAMA

Programa de impressão de erros.

CÓDIGO DO PROGRAMA

WZWRT

983

LINGUAGEM USADA

Fortran

TIPO DE PROGRAMA

Subroutine

SISTEMA SUPORTE E HARDWARE

IBM-1130

CHAVES DE PEINEL E SAÍDAS ALTERNATIVAS

nenhuma.

DESCRIÇÃO

Este programa imprime o erro de ordem "m", que lê em um arquivo de disco, o qual contém todos os erros do carregador.

SEQUENCIA DE CHAMADA

CALL WZWRT(M)

M - número do erro a ser impresso.

SUB-ROTINAS USADAS

Nenhuma

OBSERVAÇÕES E MENSAGENS DE ERROS

Imprime as mensagens gravadas no arquivo de disco.

VARIÁVEIS PRINCIPAIS

Nome	Tipo	Função.
------	------	---------

IBUF	I,()	Buffer de leitura de disco e impressão.
------	------	---

PAGE 13

```
// FOR
*ONE WORD INTEGERS
*LIST ALL
    SUBROUTINE WZWRT(M)
    DIMENSION IBUF(40)
    IPRIN=5
    IDISK=14
    READ(IDISK*M) IBUF
    WRITE(IPRIN,10)IBUF
10  FORMAT(1H0,10X,40A1)
20  RETURN
END
VARIABLE ALLOCATIONS
IBUF(I )=0027-0000 IPRIN(I )=0028           IDISK(I )=0029
```

UNREFERENCED STATEMENTS
20

STATEMENT ALLOCATIONS
10 =002C 20 =004C

FEATURES SUPPORTED
ONE WORD INTEGERS

CALLED SUBPROGRAMS
SWRT SCOMP SIOAI SUBIN SDRED SDAI

INTEGER CONSTANTS
5=002A 14=002B

CORE REQUIREMENTS FOR WZWRT
COMMON 0 VARIABLES 42 PROGRAM 36

RELATIVE ENTRY POINT ADDRESS IS 0032 (HEX)

END OF COMPILEATION

// DUP

```
*DELETE          WZWRT      0010
CART ID 0010    DB ADDR  15A1      DB CNT      0004
```

```
*STORE          WS  UA  WZWRT      0010
CART ID 0010    DB ADDR  1B38      DB CNT      0004
```

// EJECT

NOME DO PROGRAMA

Determina o indicador de relocação.

085

CÓDIGO DO PROGRAMA

WZDS

LINGUAGEM USADA

Fortran

TIPO DE PROGRAMA

Subrotina

SISTEMA SUPORTE E HARDWARE

IBM-1130

CHAVES DE PAINEL E SAÍDAS ALTERNATIVAS

Nenhuma

DESCRIÇÃO

Esta rotina testa se os dois "bits" mais significativos de uma palavra indica "11", isto é, código de relocação de um CALL.

O valor de retorno de M, parâmetro pode ser :

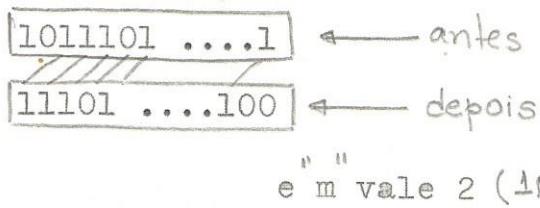
0 - absoluto

1 - relocável

2 - inválido

3 - subrotina

A palavra em teste volta deslocada de 2 posições a esquerda :



SEQUENCIA DE CHAMADA

CALL WZDS(M,IPAL)

M - código de relocação

IPAL - palavra deslocada a esquerda

SUB-ROTINAS USADAS

Nenhuma

OBSERVAÇÕES E MENSAGENS DE ERRO

Nenhuma

VARIÁVEIS PRINCIPAIS

Nenhuma

// ASM
*LIST

0000	26A44880	00001		ENT	WZDS	
0000	0	0000	00002	WZDS	DC	*-*
0001	01	65800000	00003		LDX	I1 WZDS
0003	0	1810	00004		SRA	16 * ACC=0
0004	00	C5800001	00005		LD	I1 1 * CARREGA IPAL
0006	0	E00B	00006		AND	MASK1 * TESTA
0007	0	180E	00007		SRA	14
0008	00	D5800000	00008		STD	I1 0 * M=1
			00009	*		
000A	00	C5800001	00010	FIM	LD	I1 1 * CARREGA IPAL
000C	0	1002	00011		SLA	2 * ATUALIZA IPAL
000D	00	D5800001	00012		STD	I1 1
000F	0	7102	00013		MDX	1 2 * PREPARA
0010	00	4D000000	00014		RET	BSC L1 0 *
			00015	*		
0012	0	C000	00016		MASK1 DC	/C000
0014			00017		END	

000 OVERFLOW SECTORS SPECIFIED
000 OVERFLOW SECTORS REQUIRED
004 SYMBOLS DEFINED
NO ERROR(S) AND NO WARNING(S) FLAGGED IN ABOVE ASSEMBLY

// DJP

*DELETE WZDS 0010
0 26 NAME NOT FOUND IN LET/FLET

*STORE WS UA WZDS 0010
CART ID 0010 DB ADDR 0270 DB CNT 0002

// EJECT

NOME DO PROGRAMA

Rotina para imprimir o mapa de carregamento

087

CODIGO DO PROGRAMA

WZMAP

LINGUAGEM USADA

Fortran

TIPO DE PROGRAMA

Subrotina

SISTEMA SUPORTE E HARDWARE

IBM-1130

CHAVES DE PAINEL E SAIDAS ALTERNATIVAS

Nenhuma

DESCRIÇÃO

O programa imprime:

número de posições livres de memória

endereço de execução

endereço de interrupção

tabela de interrupção

configuração de carregamento

SEQUENCIA DE CHAMADA

CALL WZMAP(MOCUP,INTD,IE4,IBUF,IPNT,ITAB,ITABL)

MOCUP - total de memória ocupada

INTD - indica se igual a zero que há rotina de interrupção

IE4 - matriz que guarda os pontos de execução das rotinas

IBUF - matriz buffer

IPNT - ponteiro para a tabela de rotinas

ITAB - tabela de rotinas do carregador

ITABL - ~~rotina~~ tabela de interrupção

SUB-ROTINAS USADAS

WZBH

WZCDM

WZPG

OBSERVAÇÕES E MENSAGENS DE ERRO

Nenhuma

VARIAVEIS PRINCIPAIS

Nome	Tipo	Função
------	------	--------

MAT	I,()	matriz de caracteres hexadecimais
-----	------	-----------------------------------

IC	I,()	matriz auxiliar para os "name code".
----	------	--------------------------------------

```

// FOR
*ONE WORD INTEGERS
*LIST ALL
      SUBROUTINE WZMAP(MOCUP,INTD,IE4,IBUF,IPNT,ITAB,ITABL)
      DIMENSION MAT(4),IE4(2),IBUF(1),ITAB(1),IC(2),ITABL(1)      088

      DATA IPRIN/5/
      IP=1
      CALL WZPG(IP)
      CALL WZBH(MOCUP,MAT)
      WRITE(IPRIN,10) MAT
10   FORMAT(1H0,10X,'MEMORIA NAO OCUPADA      ',4A1)
      CALL WZBH(IE4(1),MAT)
      WRITE(IPRIN,11) MAT
11   FORMAT(1H ,10X,'ENDERECO DE EXECUCAO      ',4A1)
12   FORMAT(1H ,10X,'ENDERECO DE INTERRUPCAO ',4A1,/,29X,'TABELA DE IN
      *TERRUPTAO',/,36X,'CANAL',/,1X,'00 01 02 03 04 05 06
      * 07 08 09 10 11 12 13 14 15')
      IF(INTD) 35,20,35
20   CALL WZBH(IE4(2),MAT)
      WRITE(IPRIN,12) MAT

      IMPRESSAO DA TABELA

      DD 30 I=1,16
      CALL WZBH(ITABL(I),MAT)
      IX=4*(I-1)
      IBUF(IX+1)=MAT(1)
      IBUF(IX+2)=MAT(2)
      IBUF(IX+3)=MAT(3)
30   IBUF(IX+4)=MAT(4)
      WRITE(IPRIN,31)(IBUF(I),I=1,64)
31   FORMAT(1X,16(4A1,1X))

      IMPRESSAO DOS PROGRAMAS E SUBROTIVAS

      35 WRITE(IPRIN,32)
      32 FORMAT(1H0,10X,'CONFIGURACAO',/,11X,'NOME      ENDERECO')
      DD 50 I=1,IPNT
      IC(1)=ITAB(I)
      IC(2)=ITAB(I+50)
      CALL WZDCM(IC(1),IC(2),IX)
      CALL WZBH(IE4(I),MAT)
      WRITE(IPRIN,40) IC,IX,MAT
40   FORMAT(11X,3A2,5X,4A1)
50   CONTINUE
      RETURN
      END

      VARIABLE ALLOCATIONS
      MAT(I )=0003-0000      IC(I )=0005-0004      IP(I )=0006      IPRIN(I )=0007

      STATEMENT ALLOCATIONS
      10    =0016   11    =0029   12    =0030   31    =008E   32    =0095   40    =00AC   20


```

CALLED SUBPROGRAMS
WZPG WZBH WZDCM SWRT SCOMP SIOAI SIDIX SIDT SUBSC SUB

INTEGER CONSTANTS
1=0012 15=0013 4=0014 64=0015 089

CORE REQUIREMENTS FOR WZMAP
COMMON 0 VARIABLES 18 PROGRAM 444

RELATIVE ENTRY POINT ADDRESS IS 00B3 (HEX)

END OF COMPIILATION

// DJP

*DELETE WZMAP 0010
D 25 NAME NOT FOUND IN LET/FLET

*STORE WS JA WZMAP 0010
CART ID 0010 DB ADDR 0213 DB CNT 0010

// EJECT

NOME DO PROGRAMA

Rotina de gravação da saída do carregador no disco.

CÓDIGO DO PROGRAMA

090

WZGRA

LINGUAGEM USADA

Fortran

TIPO DE PROGRAMA

Subrotina

SISTEMA OPERACIONAL E HARDWARE

IBM-1130

CHAVES DO PAINEL E SAÍDAS ALTERNATIVAS

Nenhuma

DESCRIÇÃO

Grava no disco, o programa carregado, precedido do seguinte cabeçalho:

- endereço de execução do programa
- comprimento do common
- endereço de carregamento do programa
- nome
- endereço da subrotina de interrupção
- comprimento do programa em "disk blocks"

O programa é guardado no arquivo de numero 11.

SEQUENCIA DE CHAMADA

CALL WZGRA(IPROX,NOME,ICOMO,IPRO,INTD,INT,IBUF,ICAR)

IPROX - próximo endereço livre da memória

NOME - nome do programa

ICOMO - tamanho do common

IPRO - endereço de execução do programa

INTD - se igual a zero não há subr. de interrupção

INT - ~~EXT~~ endereço de execução da subrotina de interrupção

IBUF - buffer de leitura

ICAR - endereço de carregamento do programa

SUB-ROTINAS CHAMADAS

WZZ

OBSERVAÇÕES E MENSAGENS DE ERRO

Nenhuma

VARIÁVEIS PRINCIPAIS

Nenhuma

PAGE 5

// *

// FOR

*ONE WORD INTEGERS

*LIST ALL

SUBROUTINE WZGRA(IPROX,NOME,ICOMO,IPRO,INTD,INT,IBUF,ICAR)

* TRATAMENTO DO PROGRAMA ABSOLUTO *

* IND(1)= ENDERECO DE EXECUCAO DO PROGRAMA*

* IND(2)= COMPRIMENTO DO COMMON *

* IND(3)= ENDERECO DE CARREGAMENTO DO PRO-*

GRAMA *

* IND(5-6) = NOME DO PROGRAMA *

* IND(7) = ENDERECO DE EXECUCAO DA SUBRO/ *

* TINA DE INTERRUPCAO *

* IND(8) = COMPRIMENTO (EM DB) *

DIMENSION NOME(2),IBUF(1)

COMMON IVI(3000)

I=1

IBUF(I)=IPRO

IBUF(I+1)=ICOMO

IBUF(I+2)=ICAR

IBUF(I+3)=IPROX

IBUF(I+4)=NOME(1)

IBUF(I+5)=NOME(2)

IBUF(I+8)=1

IF(INTD)10,11,10

11 IBUF(I+6)= INT

IBUF(I+8)= 0

ACAO DE GRAVACAO

C CALCULO DO NUMERO DE DISK BLOCKS

10 N=(IPROX/2+20)/20+1

IBUF(I+7)=N

WRITE(11'1)(IBUF(I),I=1,20)

N1=2

JJ=1

DO 1 J=1,IPROX,2

CALL WZZ(IBUF(JJ),J)

JJ=JJ+1

IF(JJ-320) 1,1,2

1 CONTINUE

C GRAVACAO DO PROGRAMA

WRITE(11'N1)(IBUF(L),L=1,320)

RETURN

2 WRITE(11'N1)(IBUF(L),L=1,320)

N1=N1+16

JJ=1

GO TO 1

END

VARIABLE ALLOCATIONS

IVI(IC)=7FFF-7448

I(I)=0000

N(I)=0001

N1(I)=000

L(I)=0005

091

PAGE 6

STATEMENT ALLOCATIONS

11 =008A 10 =009C 1 =00F0 2 =0113

092

FEATURES SUPPORTED
ONE WORD INTEGERS

CALLED SUBPROGRAMS

WZZ SUBSC SUBIN SDWRT SDCOM SDIX

INTEGER CONSTANTS

1=0010 0=0011 2=0012 20=0013 11=0014 320=0015

CORE REQUIREMENTS FOR WZGRA

COMMON 3000 VARIABLES 16 PROGRAM 294

RELATIVE ENTRY POINT ADDRESS IS 0017 (HEX)

END OF COMPIILATION

// DJP

*DELETE WZGRA 0010
CART ID 0010 DB ADDR 17C3 DB CNT 0013

*STORE WS UA WZGRA 0010
CART ID 0010 DB ADDR 1931 DB CNT 0013

// EJECT

NOME DO PROGRAMA

Rotina de pesquisa e inclusão de "entry point" na tabela do carregador.

CÓDIGO DO PROGRAMA

WZTB

LINGUAGEM USADA

FOrtran

TIPO DE PROGRAMA

Subrotina

SISTEMA SUPORTE E HARDWARE

IBM-1130

CHAVES DE PAINEL E SAÍDAS ALTERNATIVAS

Nenhuma

DESCRIÇÃO

Este programa determina se um determinado "entry point" já se encontra na tabela do carregador e o inclui se não estiver.

SEQUENCIA DE CHAMADA

CALL WZTB(R,IPNT,M,TAB)

R - nome a ser procurado

IPNT - ponteiro para a última posição da tabela

TAB - tabela do carregador

M - indica se diferente de zero a ocorrência de erro

SUB-ROTINAS USADAS

Nenhuma

OBSERVAÇÕES E MENSAGENS DE ERRO

Nenhuma

VARIÁVEIS PRINCIPAIS

Nenhuma

PAGE 48

094

// FOR
*ONE WORD INTEGERS
*LIST ALL
SUBROUTINE WZTB(IC,IPNT,M,ITAB,K)
DIMENSION ITAB(1),IC(2)
M=0
DO 100 J=1,IPNT
IF(IC(1)-ITAB(J)) 100,31,100
31 IF(IC(2)-ITAB(J+50)) 100,61,100
100 CONTINUE
30 IPNT=IPNT+1
IF(IPNT-51) 40,50,50
40 ITAB(IPNT)=IC(1)
ITAB(IPNT+50)=IC(2)
<=IPNT
RETURN
50 M=1
RETURN
61 <=J
RETURN
END
VARIABLE ALLOCATIONS
J(I)=0000

UNREFERENCED STATEMENTS
30

STATEMENT ALLOCATIONS
31 =003B 100 =004A 30 =0053 40 =005F 50 =007F 61 =0085

FEATURES SUPPORTED
ONE WORD INTEGERS

CALLED SUBPROGRAMS
SUBSC SUBIN

INTEGER CONSTANTS
0=0006 1=0007 51=0008

CORE REQUIREMENTS FOR WZTB
COMMON 0 VARIABLES 6 PROGRAM 134

RELATIVE ENTRY POINT ADDRESS IS 0009 (HEX)

END OF COMPIILATION

// DJP

*DELETE WZTB 0010
D 26 NAME NOT FOUND IN LET/FLET

*STORE WS UA WZTB 0010
CART ID 0010 DB ADDR 0461 DB CNT 0009

// EJECT

000

NOME DO PROGRAMA

Coloca cabeçalho na fita de saida do carregador

CODIGO DO PROGRAMA

WZPOE

LINGUAGEM USADA

Fortran

TIPO DE PROGRAMA

Subrotina

SISTEMA SUPORTE E HARDWARE

IBM-1130

CHAVES DE PAINEL E SAIDAS ALTERNATIVAS

Nenhuma

DESCRIÇÃO DO PROGRAMA

Preenche o cabeçalho de fita:

- posição 3 - ponto de entrada da rotina de interrupção
- posição 4 - ponto de entrada do programa
- posição 5 - eventualmente /FF00

SEQUENCIA DE CHAMADA

CALL WZPOE(IE4, IE5, IE42)

IE4 - ponto de entrada do programa

IE5 - ponto de carregamento do programa

IE42 - ponto de entrada da rotina de interrupção

SUB-ROTINAS USADAS

Nenhuma

OBSERVAÇÕES EM MENSAGENS DE ERRO

Nenhuma

VARIAVEIS PRINCIPAIS

Nome	Tipo	Função

Nenhuma

PAGE 3

096

// ASM
*LIST

		00001	*FINALIZA MONTAGEM		
		00002	* ITABL = TABELA DE INTERRUPCAO		
		00003	* I = ROTINA DIF. DE ZERO		
		00004	* N = /7800+IE5(2)/2		
		00005	* IE4(1) -		
		00006	* IE5(1) -		
		00007	* IE4(2) -		
0000	26A57585	00008	ENT	WZPOE	
0000	0 0000	00009	WZPOE	DC	*-*
0001	01 65800000	00010	LDX	I1	WZPOE
0003	0 C100	00011	LD	1	0
0004	0 D01B	00012	STO		F5+1
0005	0 C101	00013	LD	1	1
0006	0 D009	00014	STO		F11+1
0007	0 C102	00015	LD	1	2
0008	0 D003	00016	STO		F10+1
0009	0 7104	00017	MDX	1	4
000A	0 6919	00018	STX	1	FIM+1
000B	00 C4000000	00019	F10	LD	L 0 * INT(4)
000D	00 D4007802	00020	STO	L	/7802
000F	00 C4000000	00021	F11	LD	L 0 * IE5(1)
0011	0 9013	00022	S		D10
0012	01 4C10001B	00023	BNN		F4
0014	01 C4800010	00024	LD	I	F11+1
0016	0 1803	00025	SRA		3
0017	01 4C20001F	00026	BNZ		F5
0019	01 4C180023	00027	BZ		FIM
001B	01 C4000026	00028	F4	LD	L HFF00
001D	00 D4007804	00029	STO	L	/7804
001F	00 C4000000	00030	F5	LD	L 0 * IE4(1)
0021	00 D4007803	00031	STO	L	/7803
0023	00 4C000000	00032	FIM	BSC	L 0
0025	0 000A	00033	D10	DC	10
0026	0 FF00	00034	HFF00	DC	/FF00
0028		00035	END		

000 OVERFLOW SECTORS SPECIFIED

000 OVERFLOW SECTORS REQUIRED

008 SYMBOLS DEFINED

NO ERROR(S) AND NO WARNING(S) FLAGGED IN ABOVE ASSEMBLY

// DUP

*DELETE WZPOE 0010
CART ID 0010 DB ADDR 1759 DB CNT 0003

*STORE WS UA WZPOE 0010
CART ID 0010 DB ADDR 1852 DB CNT 0003

// EJECT

NOME DO PROGRAMA

GERÊncia da busca de comandos "CALL".

097

CÓDIGO DO PROGRAMA

WZCAL

LINGUAGEM USADA

Fortran

TIPO DE PROGRAMA

Subrotina

SISTEMA SUPORTE E HARDWARE

IBM-1130

CHAVES DE PAINEL E SAÍDAS ALTERNATIVAS

Nenhuma

DESCRIÇÃO

Esta rotina lê um programa a partir de um arquivo (biblioteca ou arquivo de saída do Montador), e determina quando uma pseudo CALL é encontrada. A medida que vai lendo (em blocos de 9 palavras) testa os indicadores de relocação (3), e inclui os nomes das rotinas na tabela do carregador.

SEQUENCIA DE CHAMADA

CALL WZCAL(IDISK,IEND,TAB,IPNT,M)

IDISK - arquivo de disco

IEND - endereço no arquivo de disco

TAB - tabela do carregador

IPNT - ponteiro para a tabela do carregador

M - indica se diferente de zero, ocorrência de erro.

SUB-ROTINAS USADAS

WZBC

OBSERVAÇÕES E MENSAGENS DE ERRO

Nenhuma

VARIÁVEIS PRINCIPAIS

Nome	Tipo	FUnção
IBUF	I,()	buffer de leitura
NN	I	contador de palavras examinadas
IPONT	I	primeira posição no programa a ser examinada
K	I	contador de palavras no buffer
MAT	I	matriz de trabalho que contém as palavras sendo examinadas.

PAGE 9

```
// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
    SUBROUTINE WZCAL(IDISK,NDB,IEND,ITAB,IPNT,M)
    DIMENSION IBUF(320),MAT(9),ITAB(1)
    COMMON ITUDO(1430)
    EQUIVALENCE (ITUDO(21),IBUF(1))
    ****
    C      * BUSCA CALLS DE UMA ROTINA
    C      ****
    M=0
    IF(NDB-16) 10,20,20
10   KK=NDB*20
     GO TO 25
20   KK=320
25   READ(IDISK*IEND)(IBUF(I),I=1,KK)
     KX=NDB
90   IPONT=IBUF(6)+10
C
C
100  NN=IBUF(IPONT+1)-2
     IPONT=IPONT+2
110  K=IPONT+9
     DO 120 I=1,9
     INDD=IPONT+I-1
120  MAT(I)=IBUF(INDD)
C
C
     NN=NN-9
     IF(NN) 200,200,130
130  IPONT=IPONT+9
     N=IBUF(IPONT+1)
     CALL WZBC(IPNT,M,MAT,N,ITAB)
     IF(M)315,110,315
C
C
200  CALL WZBC(IPNT,M,MAT,N,ITAB)
     IF(M)315,201,315
201  IF(K-320) 300,350,350
300  IPONT=IPONT+NN+9
305  NN=IBUF(IPONT+1)-2
     IPONT=IPONT+2
310  IF(NN) 315,320,110
315  RETURN
C
C
320  IF(IPONT-320)305,350,350
350  KX=KX-16
     IF(KX)360,365,365
360  M=0
     RETURN
C
C
365  IF(KX-16) 366,367,367
366  KK=KX*20
     GO TO 300
```

PAGE 10

029

```
380 IEND=IEND+16
READ(IDISK*IEND)(IBUF(J),J=1,KK)
IPONT=1
GO TO 100
END
```

UNREFERENCED STATEMENTS

90 310

FEATURES SUPPORTED
ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZCAL
COMMON 1430 VARIABLES 20 PROGRAM 316

RELATIVE ENTRY POINT ADDRESS IS 001C (HEX)

END OF COMPILATION

// DUP

```
*DELETE          WZCAL      0010
CART ID 0010    DB ADDR 0254    DB CNT  0014
```

```
*STORE          WS  UA  WZCAL      0010
CART ID 0010    DB ADDR 1B27    DB CNT  0014
```

// EJECT

100

PROGRAMAS DE PERFURAÇÃO DE CARTÕES

WZSDS

WZTER

NOME DO PROGRAMA

101

Programa para perfuração de cartões do carregador

CÓDIGO DO PROGRAMA

WZSDS

LINGUAGEM USADA

Fortran

TIPO DO PROGRAMA

Programa principal

SISTEMA SUPORTE E HARDWARE

IBM-1130

CHAVES DE PAINEL E SAÍDAS ALTERNATIVAS

Nenhuma

DESCRIÇÃO

Este programa é encarregado de receber no Common as variáveis de interesse da rotina WZTER, que perfura cartões, e chamar a rotina, tendo estes valores como parametros.

SEQUENCIA DE CHAMADA

É chamada através de um CALL LINK(WZSDS) pelo programa carregador.

SUB-ROTINAS USADAS

WZTER

OBSERVAÇÕES E MENSAGENS DE ERRO

Nenhuma

VARIÁVEIS PRINCIPAIS

Nenhuma

PAGE 9

// FOR
*ONE WORD INTEGERS
*LIST ALL
*IDCS(CARD,1403PRINTER,DISK) 102
COMMON IVI(3000)
CALL WZTER(IVI(2999),IVI(3000))
CALL EXIT IPR04 TES
END
VARIABLE ALLOCATIONS
IVI(IC)=7FFF-7448

FEATURES SUPPORTED
ONE WORD INTEGERS
IDCS

CALLED SUBPROGRAMS
WZTER FLD FSTD CARDZ SFIO PRNZ SDFIO

CORE REQUIREMENTS FOR
COMMON 3000 VARIABLES 2 PROGRAM 44

END OF COMPIILATION

// DJP

*DELETE WZSDS 0010
CART ID 0010 DB ADDR 18C0 DB CNT 0003

*STORE WS UA WZSDS 0010
CART ID 0010 DB ADDR 1944 DB CNT 0004

// EJECT

NOME DO PROGRAMA

Rotina de perfuração dos cartões de saída.

103

CÓDIGO DE PROGRAMA

WZTER

LINGUAGEM USADA

Assembler

TIPO DE PROGRAMA

Subrotina

SISTEMA SUPORTE E HARDWARE

IBM-1130

CHAVES DE PAINEL E SAÍDAS ALTERNATIVAS

Nenhuma

DESCRIÇÃO

Este programa gera os cartões de saída para o programa reconfigurador, isto é o gerador de fita perfurada.

Os cartões são perfurados byte a byte; cada bloco de dados do cartão, é constituído de um contador de bytes e de um checksum. Os cartões apresentam um cabeçalho, o qual contém 5 bytes (5 palavras do minicomputador), respectivamente: endereço de interrupção (2), endereço de execução do programa (2) e /FF00 (1).

SEQUENCIA DE CHAMADA

CALL WZTER(IPROX,IE5)

IPROX - próximo endereço livre da memória

IE5 - ponto de execução do programa

SUB-ROTINAS USADAS

São utilizadas somente subrotinas internas:

SPOE - faz a saída de um bloco de palavras

SPEGA - toma o byte direito ou esquerdo de uma palavra cf.
o endereço

CDUMP - armazena no buffer os valores do Acumulador

CPERF - faz a perfuração

CARD1 - perfura efetivamente o cartão

OBSERVAÇÕES E MENSAGENS DE ERRO

Nenhuma

VARIÁVEIS PRINCIPAIS

Nenhuma

// ASM

*LIST

0000	26A63159	00001	ENT	WZTER	
0000 0	0000	00002	WZTER	DC	*--*
0001 01	65800000	00003		LDX	I1 WZTER
0003 00	C5800000	00004		LD	I1 0
0005 0	D031	00005		STO	PROX
0006 00	C5800001	00006		LD	I1 1
0008 0	D02F	00007		STO	PROG5
0009 0	61B0	00008	TERMI	LDX	I1 -80
000A 0	C056	00009		LD	X4
000B 0	D05A	00010		STO	SINI
000C 0	C056	00011		LD	X6
000D 0	D059	00012		STO	SCONT
000E 0	402A	00013		BSI	SPOE
000F 01	C4000037	00014		LD	L PROX * NUMERO DE POSICOES DO
0011 01	94000038	00015		S	L PROG5 * PROGRAMA
0013 0	1801	00016		SRA	1 /2
0014 0	1886	00017		SRT	6 /64
0015 0	D04A	00018		STD	SBLOC GRUPO DE 128 PALAVRAS
0016 0	1010	00019		SLA	16
0017 0	1086	00020		SLT	6 RESTO DO GRUPO
0018 0	1001	00021		SLA	1
0019 0	D04E	00022		STO	SREST
001A 01	C4000038	00023		LD	L PROG5 *END.CARGA
001C 0	D049	00024		STO	SINI
001D 0	C042	00025		LD	SBLOC *NUMERO DE BLOCOS DE
001E 01	4C180026	00026		BZ	VAI * 128 PALAVRAS
0020 0	C041	00027	SLOOP	LD	X128
0021 0	D045	00028		STD	SCONT
0022 0	4016	00029		BSI	SPOE
0023 01	74FF0060	00030		MDM	L SBLOC,-1
0025 0	70FA	00031		MDX	SLOOP
0026 0	C041	00032	VAI	LD	SREST * RESTO DO DUMP
0027 01	4C18002B	00033		BZ	STERM
0029 0	D03D	00034		STO	SCONT
002A 0	400E	00035		BSI	SPOE
002B 0	405E	00036	STERM	BSI	CFINA
002C 01	C4000037	00037		LD	L PROX
002E 0	1801	00038		SRA	1
002F 0	8035	00039		A	X7800
0030 0	D004	00040		STO	DUMPI+4
0031		00041	DUMPI	PDMP	/7800,/7FFF
0036 0	6038	00042		EXIT	
0037 0	0000	00043	PROX	DC	0
0038 0	0000	00044	PROG5	DC	0
		00045		*	
		00046		*	
		00047		*	
		00048		*****	
		00049		* SPOE -FAZ A SAIDA DE UM BLOCO DE DADOS	*
		00050		* FORMATO - NUMERO DE PALAVRAS-1 BYTE	*
		00051		* - END.CARGA - 2 BYTES	*
		00052		* - DADOS - 128 BYTES	*
		00053		* - CHECKSUM - 1BYTE	*
		00054		*****	
0039 0	0000	00055	SPOE	DC	*--*

003A	0	C02C	00056		LD	SCONT	
003B	0	9028	00057		S	X1	
003C	01	4C20003F	00058		BNZ	SPAI	
003E	0	8025	00059		A	X1	
003F	0	1008	00060	SPAI	SLA	8	
0040	0	D028	00061		STO	CHECK	
0041	0	403F	00062		BSI	CDUMP	
0042	0	C023	00063		LD	SINI	
0043	0	1888	00064		SRT	8	
0044	0	1008	00065		SLA	8	
0045	0	403B	00066		BSI	CDUMP	
0046	0	8022	00067		A	CHECK	
0047	0	D021	00068		STO	CHECK	
0048	0	1088	00069		SLT	8	
0049	0	1008	00070		SLA	8	
004A	0	4036	00071		BSI	CDUMP	
004B	0	801D	00072		A	CHECK	
004C	0	D01C	00073		STO	CHECK	
004D	0	401D	00074	SPLOO	BSI	SPEGA	
004E	0	4032	00075		BSI	CDUMP	
004F	0	8019	00076		A	CHECK	
0050	0	D018	00077		STO	CHECK	
0051	01	74010066	00078		MDM	L SINI,1	
0053	01	74FF0067	00079		MDM	L SCONT,-1	
0055	0	70F7	00080		MDX	SPLOO	
0056	0	1010	00081		SLA	16	
0057	0	9011	00082		S	CHECK	
0058	0	4028	00083		BSI	CDUMP	
0059	0	1010	00084		SLA	16	
005A	0	4026	00085		BSI	CDUMP	
005B	0	4025	00086		BSI	CDUMP	
005C	0	4024	00087		BSI	CDUMP	
005D	0	4023	00088		BSI	CDUMP	
005E	01	4C800039	00089		BSC	I SPOE	
0060	0	0000	00090	SBLOC	DC	*-*	
0061	0	0004	00091		X4	DC	4
0062	0	0080	00092		X128	DC	128
0063	0	0006	00093		X6	DC	6
0064	0	0001	00094		X1	DC	1
0065	0	7800	00095		X7800	DC	/7800
0066	0	0000	00096		SINI	DC	*-*
0067	0	0000	00097		SCONT	DC	*-*
0068	0	0000	00098		SREST	DC	*-*
0069	0	0000	00099		CHECK	DC	*-*
006A	0	0000	00100		CPROV	DC	*-*
			00101		*		
			00102		*		
			00103		*		
			00104		*****		
			00105		* SPEGA-SUBROTINA QUE DEVOLVE NO ACC O BYTE DI- *		
			00106		* REITO OU ESQUERDO DE UMA PALAVRA CUJO *		
			00107		* ENDERECHO E DADO POR (/7800+SINI/2). *		
			00108		* O BYTE DIREITO E DADO POR SINI INPAR. *		
			00109		* O BYTE ESQUERDO E DADO POR SINI PAR. *		
			00110		*****		
006B	0	0000	00111		SPEGA	DC	*-*
006C	0	C0F9	00112		LD	SINI	

006D 01 4C040078 00113 BOD SIMPA
 006F 0 1801 00114 SRA 1
 0070 0 80F4 00115 A X7800
 0071 0 D001 00116 STO *+1
 0072 00 C4000000 00117 LD L *-*
 0074 0 1808 00118 SRA 8
 0075 0 1008 00119 SLA 8
 0076 01 4C80006B 00120 BSC I SPEGA
 0078 0 1801 00121 SIMPA SRA 1
 0079 0 80EB 00122 A X7800
 007A 0 D001 00123 STO *+1
 007B 00 C4000000 00124 LD L *-*
 007D 0 1008 00125 SLA 8
 007E 01 4C80006B 00126 BSC I SPEGA
 00127
 00128
 00129
 00130
 00131 **** CDUMP-ARMAZENA NO BUFFER DE SAIDA TODO VALOR *
 00132 NO ACC. *
 00133 80 VALORES SAO GUARDADOS NO BUFFER *
 00134 QUANDO 80 VALORES SAO COMPLETADOS E CHA-*
 00135 MADA CPERF. NESSE CASO O VALOR 'CPROU' *
 00136 RETORNA NO ACC. *
 00137
 0081 00138 CDUMP EQU *+1
 0080 00 4C000000 00139 CFIM BSC L *-*
 0082 01 D5000100 00140 STO L1 SAI+81
 0084 0 7101 00141 MDX 1 1
 0085 0 70FA 00142 MDX CFIM
 0086 0 D0E3 00143 STO CPROV
 0087 0 4009 00144 BSI CPERF
 0088 0 COE1 00145 LD CPROV
 0089 0 70F6 00146 MDX CFIM
 00147
 008A 0 0000 00148 CFINA DC *-*
 008B 0 7150 00149 MDX 1 80
 008C 0 1000 00150 NOP
 008D 0 6921 00151 STX 1 SAI
 008E 0 4002 00152 BSI CPERF
 008F 01 4C80008A 00153 BSC I CFINA
 00154
 00155
 00156 **** CPERF-PERFURA BUFFER DE SAIDA, ATRAVES DE *
 00157 CARD1. TESTA SE CARTAO A PERFORAR ESTA *
 00158 EM BRANCO. SE NAO ESTA, BUSCA OUTRO CAR- *
 00159 TAO E REPETE A OPERACAO. *
 00160
 0091 0 0000 00161 CPERF DC *-*
 0092 20 03059131 00162 CLER LIBF CARD1
 0093 0 1000 00163 DC /1000
 0094 1 0100 00164 DC ENTRA
 0095 1 0151 00165 DC RERRO
 0096 20 03059131 00166 LIBF CARD1
 0097 0 0000 00167 DC 0
 0098 0 70FD 00168 MDX *-3
 0099 0 61B0 00169 LDX 1 -80

PAGE 6

009A 01 C5000151 00170 CLOOP LD L1 ENTRA+81
009C 01 4C2000AC 00171 BNZ JAPER
009E 0 7101 00172 MDX 1 1
009F 0 70FA 00173 MDX CLOOP
00A0 20 03059131 00174 LIBF CARD1
00A1 0 2000 00175 DC /2000
00A2 1 00AF 00176 DC SAI
00A3 1 0151 00177 DC RERRO
00A4 20 03059131 00178 LIBF CARD1
00A5 0 0000 00179 DC 0
00A6 0 70FD 00180 MDX *-3
00A7 0 6150 00181 LDX 1 80
00A8 0 6906 00182 STX 1 SAI
00A9 0 6180 00183 LDX 1 -80
00AA 01 4C800091 00184 BSC I CPERF
00AC 20 03059131 00185 JAPER LIBF CARD1
00AD 0 4000 00186 DC /4000
00AE 0 70E3 00187 MDX CLER
00AF 0 0050 00188 SAI DC 80
00B0 0050 00189 BSS 80
0100 0 0050 00190 ENTRA DC 80
0101 0050 00191 BSS 80
0151 0 0000 00192 RERRO DC **-*
0152 01 4C800151 00193 BSC I RERRO
00194 *
00195 *
0154 00196 END

107

000 OVERFLOW SECTORS SPECIFIED
000 OVERFLOW SECTORS REQUIRED
034 SYMBOLS DEFINED
NO ERROR(S) AND NO WARNING(S) FLAGGED IN ABOVE ASSEMBLY

// DJP

*DELETE WZTER 0010
CART ID 0010 DB ADDR 18C0 DB CNT '000C

*STORE WS UA WZTER 0010
CART ID 0010 DB ADDR 1975 DB CNT 000C

// XEQ WZV 1

*FILES(12,BBB12),(14,WZNOQ),(1,N08),(11,WZN11)

-9920-10688-15552-16064	16448-15678-15631	-3520	30931	9	0-11456-10
-31988	6148				
59	69				
1	69	69	773 12864	-4	1 35 0 0 2 4953 12
17993	-4	2	36 2	0	10 5668 12288 -2 2 38
103	101	110-15104-32767		6	-1 10 10371 8848-31679-32368
6-11264	12				
59	69				
53	53				
1	53	53	30 0	0	8 772 16384 -4 2 30
0	0	56	773 12864	-4	1 35 0 0 2 4953 12
17993	-4	2	36 2	0	10 5668 12288 -2 2 38

ROTINAS GERAIS

WZREL

WZBUS

WZBB

WZC1

WZBH

WZDCM

WZB1

NOME DO PROGRAMA

Verifica cabeçalho de relocação

109

CÓDIGO DE PROGRAMA

WZREL

LINGUAGEM USADA

Fortran

TIPO DE PROGRAMA

Subrotina

SISTEMA SUPORTE E HARDWARE

IBM-1130

CHAVES DE PAINEL E SAÍDAS ALTERNATIVAS

Nenhuma

DESCRIÇÃO

O cabeçalho tem a forma: zero, zero, tipo e subtípico de programa, comprimento do programa, comprimento do common, comprimento do header = 9, zero, número de disk-blocks do programa, zero, nome do programa (em name code), endereço do ponto de execução (caso de programa principal). No caso de subrotinas, seguem-se outros blocos do tipo: nome do ponto de entrada e respectivo endereço. No caso de sub-rotina de entrada e saída vem: número do dispositivo de ISS+50, número do dispositivo, 1, 1, zero. A função desta rotina é testar estas condições.

SEQUÊNCIA DE CHAMADA

CALL WZREL(M,IK)

M - parâmetro de erro. Indica ocorrência de erro se diferente de zero.

IK - (IK/3) - indica quantos pontos de entrada se tem.

SUB-ROTINAS USADAS

WZCDM

OBSERVAÇÕES E MENSAGENS DE ERRO

Nenhuma

VARIÁVEIS PRINCIPAIS

Nome	Tipo	Função
IND	C	diretoria da biblioteca
MAT	I	nome do programa externamente especificado
IHEAD	I,()	"header" do programa
N1	I	primeira palavra do nome em "name code"
N2	I	segunda palavra do nome em "name code".

```

// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
SUBROUTINE WZREL(M,IK)
C ****
C *TESTA FORMATO RELOCABEL
C * FORMATO - ZERO,ZERO, TIPO
C * COMP,PROGRAMA,COMMON, HEADER-9,
C * ZERO,NDB,ZERO
C * PP - NOME1,NOME2,PONTO DE ENTRADA
C * SB - IDEM
C * ISS- IDEM,ISS+50,ISS,1,1,0
C * DO NOME DO PROGRAMA VEM EM IND(1-5)
C ****
C DIMENSION IHEAD(60),MAT(5),IND(20)
C COMMON ITJDO(1430)
C EQUIVALENCE (ITJDO(1),IND(1)),(ITJDO(661),IHEAD(1)),(IHEAD(1),I1),
C 1 (IHEAD(2),I2),(IHEAD(3),I3),(IHEAD(4),I4),(IHEAD(5),I5),(IHEAD(6)
C 2 ,I6),(IHEAD(7),I7),(IHEAD(8),I8),(IHEAD(9),I9)
C
C I<=3
C IF(I1) 1,10,1
10 IF(I2) 1,20,1
20 I=I3/256
IF(I)1,1,30
30 IF(I-5)40,40,1
40 GO TO(100,100,1,200,1,300),I
1 M=1
RETURN
C
C PROGRAMA PRINCIPAL
C
100 IF(I6-3) 1,110,1
110 IF(IHEAD(12)-4095) 120,120,1
120 IF(I9)1,130,1
130 IF(I7)1,140,1
140 IF(I5-4095) 150,150,1
150 M=0
RETURN
C
C SUBROUTINE ISS
C
300 IF(I6-8)1,310,1
310 IF(IHEAD(14)) 1,320,320
320 IF(IHEAD(14)-15) 330,330,1
330 IF(IHEAD(13)) 1,340,340
340 IF(IHEAD(13)-65) 350,350,1
350 DO 380 J=15,16
IF(IHEAD(J)) 1,380,1
380 CONTINUE
IF(IHEAD(17)) 1,390,1
390 DO 400 I=1,5
400 MAT(I)=IND(I)
CALL WZCDM(MAT,N1,N2)
IF(N1-IHEAD(10)) 1,410,1
410 IF(N2-IHEAD(11)) 1,110,1

```

PAGE 23

SUBROUTINA

111

```
200 IF(I6-3) 1,210,210
210 IF(I6-36) 220,220,1
220 I=I6+9
IK=I6
DO 230 J=1,5
230 MAT(J)=IND(J)
CALL WZCDM(MAT,N1,N2)
<=0
DO 260 J=10,I,3
IF(IHEAD(J)-N1) 240,250,240
240 IF(IHEAD(J+2)-4095) 260,260,1
250 IF(IHEAD(J+1)-N2) 240,255,240
255 IF(IHEAD(J+2)-4095) 257,257,1
257 <=1
260 CONTINUE
IF(K-1)1,120,1
END
```

FEATURES SUPPORTED
ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZREL

COMMON 1430 VARIABLES 12 PROGRAM 334

RELATIVE ENTRY POINT ADDRESS IS 001A (HEX)

END OF COMPIILATION

// DJP

*DELETE WZREL 0010
D 26 NAME NOT FOUND IN LET/FLET

*STORE WS UA WZREL 0010
CART ID 0010 DB ADDR 0237 DB CNT 0015

// EJECT

NOME DO PROGRAMA

Busca nome na biblioteca.

112

CÓDIGO DE PROGRAMA

WZBUS

LINGUAGEM USADA

Fortran

TIPO DE PROGRAMA

Subrotina

SISTEMA SUPORTE E HARDWARE

IBM-1130

CHAVES DE PAINEL E SAÍDAS ALTERNATIVAS

Nenhuma

DESCRIÇÃO

Esta rotina busca o nome de uma rotina ou programa, pedidos na diretoria (índice) de uma biblioteca dada, através de um certo nome. O nome pedido, é fornecido através de IND(1-5), e depois convenientemente transformado em "name code".

O arquivo biblioteca é fornecido. A subrotina devolve ao programa chamante, um parâmetro indicando, se a rotina ou programa procurada/o foi ou não encontrado. No caso de ter sido encontrado, 2 ponteiros indicam todos os "entry points" dos programas ou rotinas pedidas. No caso de um programa, os dois ponteiros devem coincidir.

SEQUENCIA DE CHAMADA

CALL WZBUS(M,IPONT,NDB,IPNT1,DISK)

M - indica se diferente de zero a ocorrência de erro,

IPONT - ponteiro inferior

IPNT1 - ponteiro superior

NDB - tamanho em "disk blocks".

IDISK - número do arquivo em disco.

SUB-ROTINAS USADAS

WZCDM

OBSERVAÇÕES E MENSAGENS DE ERRO

Nenhuma

VARIÁVEIS PRINCIPAIS

Nome	Tipo	Função
IP	I	ponteiro para o buffer
J	I	ponteiro no índice
NROT	I	ponteiro para o número de programas
NIN	I	variável de leitura no disco
ITEST	I	endereço a testar
NINDL	I	primeiro endereço livre no índice

```

// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
SUBROUTINE WZBUS(M,IPONT,NDB,IPNT1,DISK,N1,N2)
*****PROCURA NOME PEDIDO NO INDICE
*   O NOME DO PROGRAMA E RECEBIDO EM IND(1-5)
*   O NUMERO DA BIBLIOTECA VEM EM IDISK
*   IPONT E O PONTEIRO INFERIOR NO CASO SUBROTINA
*   IPNT1 E O PONTEIRO SUPERIOR NO CASO SUBROTINA
*****DIMENSION IND(10),IBUF(320)
COMMON ITJDO(1430)
EQUIVALENCE (IND(1),ITUDO(1)),(IBUF(1),ITUDD(341)),(NINDL,IND(10))

READ(IDISK*1) IBUF
NIN=17

CALCULA NUMERO TOTAL DE ROTULOS

NRDT=IBUF(5)
J=21
IP=21

PROCURA NOME

IF(NRDT) 19,102,19
19 DO 100 I=1, NRDT
    IF(IBUF(IP)-N1) 30,20,30
20 IF(IBUF(IP+1)-N2) 30,200,30
30 IP=IP+8
    J=J+8

FIM DE BUFFER

IF(IP-320) 100,50,50
50 READ(IDISK*NIN) IBUF
NIN=NIN+16
IP=1
100 CONTINUE
102 M=1
RETURN

NOME ENCONTRADO

200 M=0
<=IP
IT=NIN-16
<1=J
NDB=IBUF(<+3)
ITEST=IBUF(<+4)
210 IP=IP+8
J=J+8

FIM DO BUFFER

```

PAGE 13

114

```
IF(IP=320) 220,220,230
220 IF(J=NINDL) 225,240,240
225 IF(ITEST-IBJF(IP+4)) 240,210,240
230 READ(IDISK*NIN) IBUF
NIN=NIN+16
IP=1
GO TO 220

ACERTO PONTEIRO SUPERIOR E BUSCA DO INFERIOR

240 IPNT1=J-8
NIN=IT
READ(IDISK*NIN) IBUF
300 IP=K-8
J=K1-8

FIM DO BUFFER

IF(IP) 310,310,320
310 NIN=NIN-16
READ(IDISK*NIN) IBUF
IP=316
320 IF(J=20) 400,400,350
350 IF(ITEST-IBJF(IP+4)) 400,360,400
360 K=K-8
K1=K1-8
GO TO 300

ACERTA PONTEIRO INFERIOR

400 IPONT=J+8
RETURN
END
```

FEATURES SUPPORTED
ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZBUS
COMMON 1430 VARIABLES 12 PROGRAM 312

RELATIVE ENTRY POINT ADDRESS IS 0015 (HEX)

END OF COMPILATION

// DJP

*DELETE WZBUS 0010
D 26 NAME NOT FOUND IN LET/FLET

*STORE WS UA WZBUS 0010
CART ID 0010 DB ADDR 01D3 DB CNT 0014

// EJECT

NOME DO PROGRAMA

WZBB - ROTINA DE BUSCA DE BIBLIOTECA

CÓDIGO DO PROGRAMA

115

WZBB

LINGUAGEM USADA

Fortran

TIPO DO PROGRAMA

Subrotina

SISTEMA SUPORTE E HARDWARE

IBM-1130

CHAVES DE PAINEL E SAÍDAS ALTERNATIVAS

Nenhuma

DESCRIÇÃO

Esta rotina busca a biblioteca, indicada pelo nome que vem em IND(6-8), entre as duas disponíveis ao sistema.

Se não for encontrada, devolve a indicação em um parâmetro de erro. As bibliotecas são identificadas pelos números lógicos 12 e 13. A rotina devolve um destes dois números lógicos, se a biblioteca for encontrada.

SEQUÊNCIA DE CHAMADA

CALL WZBB(M, IDISK)

M - indica erro se diferente de zero.

IDISK - número da biblioteca.

SUB-ROTINAS USADAS

Nenhuma

OBSERVAÇÕES E MENSAGENS DE ERRO

Pode ser impressa a seguinte mensagem, no caso de não ter sido encontrada a biblioteca pedida:

'BIBLIOTECA PEDIDA NAO FOI ENCONTRADA'.

Neste caso M retorna com o valor 1.

VARIÁVEIS PRINCIPAIS

Nenhuma.

PAGE 30

// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
SUBROUTINE WZBB(M, IDISK) 116
DIMENSION IBUF(320), IND(20)
COMMON ITJDD(1430)
EQUIVALENCE (ITUDO(1), IND(1)), (IBUF(1), ITUDO(21))
DATA IPRIN/5/

C
C BUSCA BIBLIOTECA
C
I=1
9 GO TO (10,20,30),I
10 IDISK=12
11 READ(IDISK*1) IBUF
DO 31 J=6,8
IF(IND(J)-IBUF(J)) 15,31,15
15 I=I+1
GO TO 9
20 IDISK=13
GO TO 11
30 WRITE(IPRIN,101)
101 FORMAT(1H0,5X,"BIBLIOTECA PEDIDA NAO SE ENCONTRA NO ARQUIVO")
M=1
RETURN
31 CONTINUE
M=0
RETURN
END

FEATURES SUPPORTED
ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZBB
COMMON 1430 VARIABLES 4 PROGRAM 118

RELATIVE ENTRY POINT ADDRESS IS 0025 (HEX)

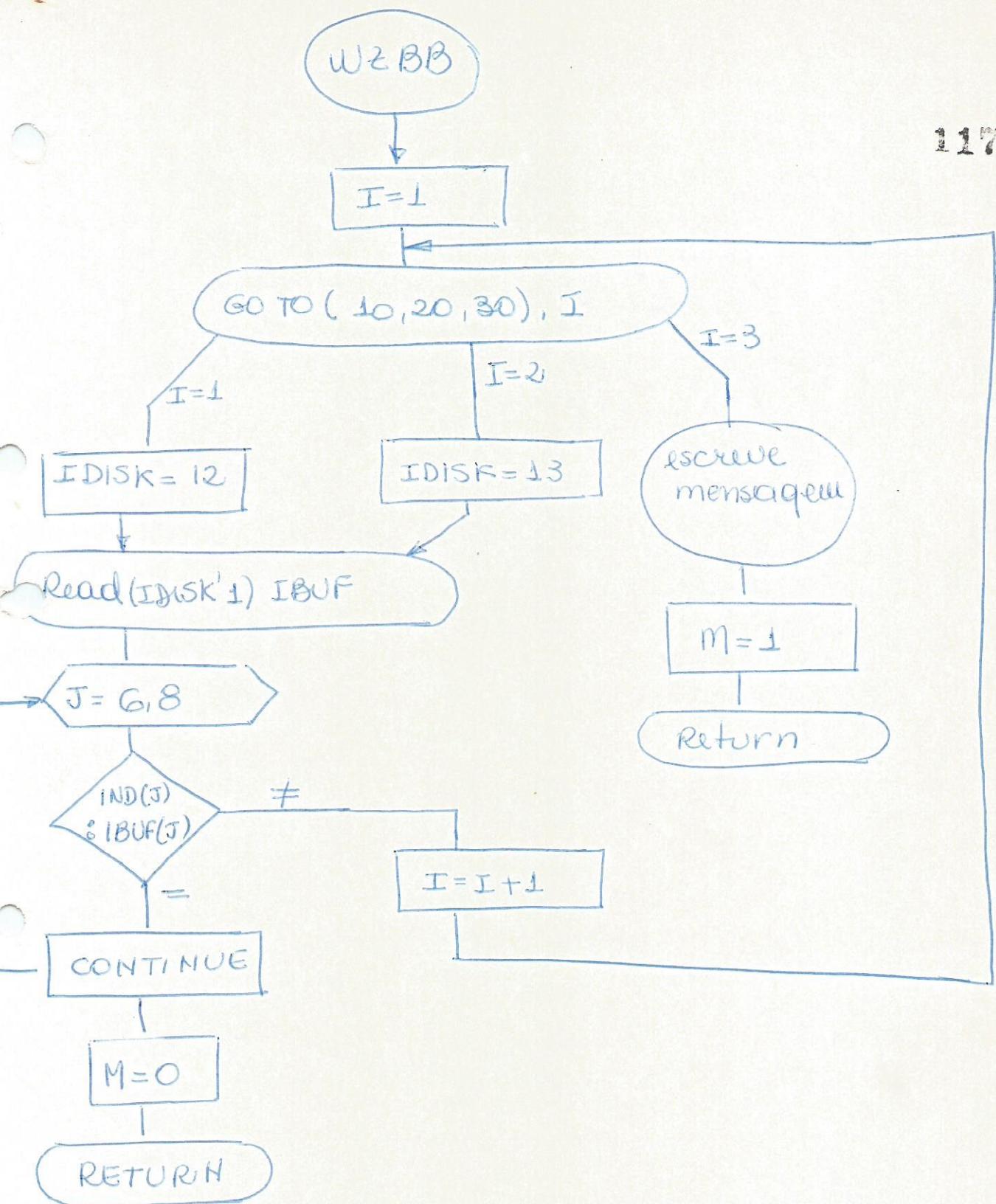
END OF COMPILATION

// DJP

*DELETE WZBB 0010
D 26 NAME NOT FOUND IN LET/FLET

*STORE WS UA WZBB 0010
CART ID 0010 DB ADDR 0270 DB CNT 0008

// EJECT



NOME DO PROGRAMA

Rotina de conversão de caracteres Al em binário

CÓDIGO DO PROGRAMA

WZC1

LINGUAGEM USADA

Fortran

TIPO DE PROGRAMA

Subrotina

SISTEMA SUPORTE E HARDWARE

IBM-1130

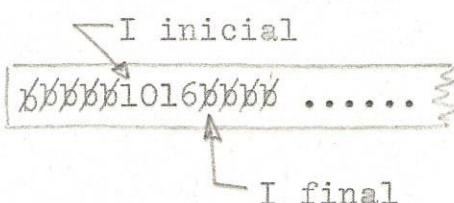
CHAVES D0 PAINEL E SAÍDAS ALTERNATIVAS

Nenhuma

DESCRIÇÃO DO PROGRAMA

Esta rotina converte uma sequência de caracteres em formato Al, localizados em um buffer de entrada, em um número binário, testando a validade dos caracteres, e encerrando a conversão ao detectar um caractere branco.

Ex:



A extensão do buffer é 60.

Se ocorrer erro um parâmetro vai indicá-lo.

SEQUENCIA DE CHAMADA

CALL WZC1(IEST,I,N,M)

IEST - buffer contendo caracteres Al.

I - ponteiro, que aponta primeiro não branco.

N - número convertido

M - indica se diferente de zero a ocorrência de erro.

SUB-ROTIMAS USADAS

Nenhuma.

OBSERVAÇÕES E MENSAGENS DE ERRO

Nenhuma.

VARIÁVEIS PRINCIPAIS

Nenhuma.

// FOR
*ONE WORD INTEGERS
*LIST ALL

SUBROUTINE WZC1(IREST,I,N,M)
DIMENSION IREST(1),ICD(10),IND(10)
DATA ICD/'0','1','2','3','4','5','6','7','8','9'/
DATA IND/0,1,2,3,4,5,6,7,8,9/,IBK/Z4040/
M=0
N=0
10 DD 40 J=1,10
IF(IREST(I)-ICD(J))40,50,40
40 CONTINUE
41 M=1
RETURN
50 N=N*10+IND(J)
I=I+1
IF(I>61)55,41,41
55 IF(IREST(I)-IBK) 10,60,10
60 RETURN
END

VARIABLE ALLOCATIONS
ICD(I)=0009-0000 IND(I)=0013-000A J(I)=0014 IBK(I)=0000

STATEMENT ALLOCATIONS
10 =0037 40 =004F 41 =0058 50 =005E 55 =0078 60 =0083

FEATURES SUPPORTED
ONE WORD INTEGERS

CALLED SUBPROGRAMS
SJBSC SUBIN

INTEGER CONSTANTS
0=0018 1=0019 10=001A 61=001B

CORE REQUIREMENTS FOR WZC1
COMMON 0 VARIABLES 24 PROGRAM 110

RELATIVE ENTRY POINT ADDRESS IS 001C (HEX)

END OF COMPIILATION

// DJP

*DELETE WZC1 0010
D 26 NAME NOT FOUND IN LET/FLET

*STORE WS UA WZC1 0010
CART ID 0010 DB ADDR 025F DB CNT 000C

// EJECT

NOME DO PROGRAMA

Rotina de conversão binário-hexadecimal

120

CÓDIGO DO PROGRAMA

WZBH

LINGUAGEM USADA

Assembler.

TIPO DE PROGRAMA

Subrotina.

SISTEMA SUPORTE E HARDWARE

IBM-1130,

CHAVES DE PAINEL E SAÍDAS ALTERNATIVAS

Nenhuma.

DESCRIÇÃO

Converte um número binário, em quatro caracteres hexadecimais.

SEQUÊNCIA DE CHAMADA

CALL WZBH(MAT, VOLTA)

MAT - número binário a converter.

VOLTA - matriz de quatro posições onde retornam os caracteres hexadecimais.

SUB-ROTINAS USADAS

Nenhuma.

OBSERVAÇÕES E MENSAGENS DE ERROS

Nenhuma.

VARIÁVEIS PRINCIPAIS

Nenhuma.

// ASM
*LIST

0000	26A42200	00001	ENT	WZBH
		00002	*	CALL WZBH(MAT,VOLTA)
		00003	*	VOLTA - MATRIZ DE QUATRO POSICOES
		00004	*	MAT - VALOR A SEPARAR EM VOLTA
		00005	*	
0000	0	0000	00006	WZBH DC *-*
0001	01	65800000	00007	LDX I1 WZBH
0003	00	C5800000	00008	LD I1 0
0005	0	D01F	00009	STO TEMP
0006	00	C5000001	00010	LD L1 1
0008	0	D01D	00011	STO VOLTA
0009	01	74FD0026	00012	MDM L VOLTA,-3
000B	0	C019	00013	LD TEMP
000C	0	100C	00014	SLA 12
000D	0	1804	00015	SRA 4
000E	0	D018	00016	STO POMP
000F	0	4018	00017	BSI RDC
0010	0	C014	00018	LD TEMP
0011	0	1804	00019	SRA 4
0012	0	100C	00020	SLA 12
0013	0	1804	00021	SRA 4
0014	0	D012	00022	STO POMP
0015	0	4012	00023	BSI RDC
0016	0	C00E	00024	LD TEMP
0017	0	1808	00025	SRA 8
0018	0	100C	00026	SLA 12
0019	0	1804	00027	SRA 4
001A	0	D00C	00028	STO POMP
001B	0	400C	00029	BSI RDC
001C	0	C008	00030	LD TEMP
001D	0	180C	00031	SRA 12
001E	0	1008	00032	SLA 8
001F	0	D007	00033	STO POMP
0020	0	4007	00034	BSI RDC
0021	01	74020000	00035	MDM L WZBH,+2
0023	01	4C800000	00036	BSC I WZBH
0025	0	0000	00037	TEMP DC *-*
0026	0	0000	00038	VOLTA DC *-*
0027	0	0000	00039	POMP DC *-*
0028	0	0000	00040	RDC DC *-*
0029	0	C0FD	00041	LD POMP
002A	0	900C	00042	S K9
002B	01	4C300035	00043	BP LETRA
002D	0	C0F9	00044	LD POMP
002E	0	E809	00045	DR MASC1
002F	01	D4800026	00046	BACK STO I VOLTA
0031	01	74010026	00047	MDM L VOLTA,+1
0033	01	4C800028	00048	BSC I RDC
0035	0	E803	00049	LETRA DR MASC2
0036	0	70F8	00050	MDX BACK
0037	0	0900	00051	K9 DC /0900
0038	0	F040	00052	MASC1 DC /F040
0039	0	C040	00053	MASC2 DC /C040
003A			00054	END

PAGE 20

122

000 OVERFLOW SECTORS SPECIFIED

000 OVERFLOW SECTORS REQUIRED

010 SYMBOLS DEFINED

NO ERROR(S) AND NO WARNING(S) FLAGGED IN ABOVE ASSEMBLY

// DJP

*DELETE WZBH 0010

D 25 NAME NOT FOUND IN LET/FLET

*STORE WS JA WZBH 0010
CART ID 0010 DB ADDR 0216 DB CNT 0005

NOME DO PROGRAMA

Rotina para converter "name code" em formato A2.

CÓDIGO DO PROGRAMA

WZDCM

LINGUAGEM USADA

Assembler .

TIPO DE PROGRAMA

subrotina .

SISTEMA SUPORTE E HARDWARE

IBM-1130 .

CHAVES DE PAINEL E SAÍDAS ALTERNATIVAS

Nenhuma .

DESCRIÇÃO

Dado um nome codificado em "name code", esta rotina o devolve dentro de três posições de uma matriz em formato A2.

Exemplo:

26A4	40D4 resulta	WZ	DC	MJ
------	------	---------------	----	----	----

SEQUÊNCIA DE CHAMADA

CALL WZDCM(n1,N2,N3)

N1 - primeira palavra em "name code" - retorna com os 2 primeiros caracteres em A2.

N2 - segunda palavra em "name code" - retorna com os 2 caracteres seguintes.

N3 - retorna com os 2 caracteres finais do nome.

SUBROTINAS USADAS

Nenhuma .

OBSERVAÇÕES E MENSAGENS DE ERRO

Nenhuma

VARIÁVEIS PRINCIPAIS

Tipo	Nome	Função
MASK1	I	máscara para o primeiro caracter _ /C000
MASK2	I	máscara de branco - /4000
MASK3	I	máscarac geral - /FF00
BK3	I	caracter branco em R1 - /0040

// ASM
*LIST

0000	26A440D4	00001
0000 0	0000	00002
		00003
		00004
		00005
0001 01	65800000	00006
0003 00	C5800000	00007
0005 0	D02B	00008
0006 00	C5800001	00009
0008 0	D029	00010
0009 0	1890	00011
000A 0	6204	00012
000B 0	4027	00013
000C 0	4026	00014
000D 0	108C	00015
000E 0	C022	00016
000F 0	188C	00017
0010 0	4022	00018
0011 0	4021	00019
0012 0	C01E	00020
0013 0	E819	00021
0014 0	E01A	00022
0015 0	D012	00023
0016 0	C012	00024
0017 0	1808	00025
0018 0	E80F	00026
0019 00	D5800000	00027
001B 0	C00F	00028
001C 0	1808	00029
001D 0	E80C	00030
001E 00	D5800001	00031
0020 0	C00B	00032
0021 0	E80E	00033
0022 00	D5800002	00034
0024 0	7103	00035
0025 0	6901	00036
0026 00	4C000000	00037
0028	0005	00038
002D 0	C000	00039
002E 0	4000	00040
002F 0	FF00	00041
0030 0	0040	00042
0031 0	0000	00043
0032 0	0000	00044
0033 0	0000	00045
0034 0	1810	00046
0035 0	18C6	00047
0036 0	1802	00048
0037 01	4C18003F	00049
0039 0	E8F3	00050
003A 01	D6000028	00051
003C 0	72FF	00052
003D 01	4C800033	00053
003F 0	E8EE	00054
0040 0	70F9	00055
0042		00056

124

ENT	WZDCM
WZDCM DC	*-*

* SUBROTINA DE TRANSFORMAR NAME CODE EM A2	

LDX	I1 WZDCM
LD	I1 0
STD	N1
LD	I1 1
STD	N1+1
SRT	16 N2 PARA EXT
LDX	2 4 NUMERO DE LETRAS
BSI	TRATA TRATA 1 CARACTER
BSI	TRATA TRATA 2 CARACTER
SLT	12
LD	N1
SRT	12
BSI	TRATA TRATA CARACTER 3
BSI	TRATA TRATA CARACTER 4
LD	N1 TRATA CARACTER
DR	MASK1
AND	MASK3 FF00
STD	TAB
LD	TAB+1
SRA	8
OR	TAB
STD	I1 0
LD	TAB+3 C
SRA	8 D
OR	TAB+2 M
STD	I1 1 P
LD	TAB+4 A
OR	BK3 C
STD	I1 2 T
MDX	1 3
STX	1 FIM+1
FIM	BSC L 0
TAB	BSS 5
MASK1 DC	/C000
MASK2 DC	/4000
MASK3 DC	/FF00
BK3 DC	/0040
N1 DC	0
DC	0
TRATA DC	0
SRA	16 LIMPA ACC
RTE	6 ULTIMO CARACTER NO ACC
SRA	2
BZ	BZ1
OR	MASK1 COLOCA '11'
STD	STD L2 TAB
MDX	2 -1
BSC	I TRATA
BZ1	OR MASK2 COLOCA '01'
MDX	STD1
END	

PAGE 29

000 OVERFLOW SECTORS SPECIFIED
000 OVERFLOW SECTORS REQUIRED
011 SYMBOLS DEFINED
NO ERROR(S) AND NO WARNING(S) FLAGGED IN ABOVE ASSEMBLY

125

// DJP

*DELETE WZDCM 0010
D 26 NAME NOT FOUND IN LET/FLET

*STORE WS JA WZDCM 0010
CART ID 0010 DB ADDR 026B DB CNT 0005

// EJECT

NOME DO PROGRAMA

Busca brancos e não brancos

CÓDIGO DO PROGRAMA

WZB1

LINGUAGEM USADA

Fortran

TIPO DE PROGRAMA

Subrotina

SISTEMA SUPORTE E HARDWARE

IBM-1130

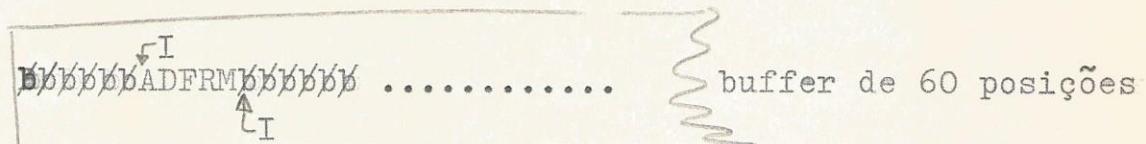
CHAVES DE PAINEL E SAÍDAS ALTERNATIVAS

Nenhuma

DESCRIÇÃO

Esta rotina busca o primeiro caracter não branco em um buffer de leitura. O ponteiro volta apontando este caracter e deve inicialmente estar apontando um branco.

Exemplo:

SEQUENCIA DE CHAMADA

CALL WZB1(IREST,I,M)

IREST - buffer onde se faz a busca.

M - indica se diferente de zero que não foi encontrado caracter diferente de branco.

I - ponteiro para o buffer.

SUB-ROTINAS USADAS

Nenhuma.

OBSERVAÇÕES E MENSAGENS DE ERRO

Nenhuma.

VARIAVEIS PRINCIPAIS

Nenhuma.

PAGE 24

```
// FOR  
*ONE WORD INTEGERS  
*LIST SOURCE PROGRAM  
SUBROUTINE WZB1(IREST,I,M) 127  
DIMENSION IREST(1)  
DATA IBK/Z4040/  
M=0  
5 IF(IREST(I)-IBK) 30,10,30  
10 I=I+1  
IF(I-61)5,20,20  
20 M=1  
30 RETURN  
END
```

FEATURES SUPPORTED
ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZB1

1 COMMON 0 VARIABLES 2 PROGRAM 50

RELATIVE ENTRY POINT ADDRESS IS 0005 (HEX)

END OF COMPIILATION

// DJP

*DELETE WZB1 0010
D 26 NAME NOT FOUND IN LET/FLET

*STORE WS UA WZB1 0010
CART ID 0010 DB ADDR 024C DB CNT 0004

// EJECT