ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO
DEPARTAMENTO DE ENGENHARIA DE ELETRICIDADE
LABORATÓRIO DE SISTEMAS DIGITAIS

SISTEMA CROSS-MONTADOR

Maria Alice G.V. Ferreira

1973

SINOPSE

O presente trabalho descreve detalhadamente o sistema Cross-Montador escrito para o computador Pato Feio, desenvolvido para o computador IBM-1130.

A finalidade de um sistema Corss-Montador é principalmente auxiliar o desenvolvimento de "software" durante a fase de desenvolvimento do "hardware" de um computador.

Quando se dispõe de um tal sistema, pode-se desenvolver em paralelo estas duas fases do projeto de um computador. Nestas condições pode-se lançar mão de um recurso adicional que é o emprego de um programa Simulador, processando-se neste, os programas já escritos que não podem ser processados na máquina final, por esta não estar ainda implementada.

Uma segunda finalidade de um tal sistema é auxiliar o "debugging" de programas, numa fase seguinte, com a máquina já em estado operacional. As vantagens de uso de um tal sistema nesta fase, é principalemente na utilização de máquinas de maior porte que a máquina objetivo, com maiores recursos daquela de que se dispõe. Por exemplo, a utilização de uma máquina com recursos de disco para preparar programas para uma máquina que dispõe apenas de um sistema de fita perfurada.

Neste trabalho estão descritos os principais módulos e arquivos utilizados pelo sistema, bem como a interrelação entre os diversos módulos de programa.

ÍNDICE

Programa Auxiliar para Listagem de Saída
de Disco do Cross-Montador..... 001
Definição dos Arquivos de Disco..... 003
Árvore de Rotinas Chamadas pelo Programa..... 004
Especificação das Variáveis em Commom - 1ªFase..... 006
Especificação das Variáveis em Commom - 2ªFase..... 007
Rotina de Tratamento da Fase de Locação..... 008

WZESS..... 009

Programas de Tratamento da Segunda Fase..... 022

WZGER..... 023
WZBTB..... 034

Rotinas de Carater Geral..... 039

WZAND/WZIOR..... 040
WZCDM..... 044
WZPK..... 048
WZITR..... 052
WZCMP..... 057

Rotinas de Tratamento Geral do Montador..... 062
(Rotinas de Busca na Tabela de Símbolos)

WZPRC..... 063
WZPRO..... 063
WZTRZ/WZTRA..... 070
WZCNV/WZDEC..... 077
wZMN..... 084

Rotinas de Impressão..... 090

WZPAG..... 091
wZPRN/WZIER/WZERR..... 096

WZWER..... 102
WZLIS..... 106
WZSSI..... 111
WZSAI..... 115
WZTES..... 120
WZRSL..... 124

Rotinas de Iniciação..... 129

WZ2..... 130
WZ1..... 135
WZHDR..... 140
WZINI..... 146

Rotinas Tratadoras de Pseudos..... 152

WZENT..... 153
WZCNT..... 158
WZEQU..... 163
WZCOD..... 169
WZISS..... 174

Rotinas de Tratamento de Buffers..... 179

WZWPT..... 180
WZINS..... 184
WZFIM..... 188
WZTRT..... 192
WZDTB..... 197
WZEND..... 201

Rotinas de Tratamento das Instruções..... 206

WZPNL..... 207
WZADR..... 216
WZIME..... 211
WZDES..... 222
WZIIO..... 226
WZOPE..... 231
WZMIC..... 236

100

```
// JOB    0001 0010                                                    *     E102,MAL

LOG DRIVE   CART SPEC   CART AVAIL  PHY DRIVE
  0000        0001        0001        0000
  0001        0010        0010        0001
                          100B        0002

V2 M09   ACTUAL 16K  CONFIG 16K

* LABORATORIO DE SISTEMAS DIGITAIS * EPUSP *  06/ 12/ 74  * 18 H  28 MIN

// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
*IOCS(1403PRINTER,DISK)
      DIMENSION IVAC(80)
      DIMENSION INIC(48)  ,NIT(10,4)
      DATA INIC / 21*0,'A','B','C','D','E','F','0','1','2','3','4',
     1        '5','6','7','8','9',  11*0 /
      DEFINE FILE 8(4800,1,U,NR)
      JM=0
      JJ=1
      NR=1
      WRITE(5,1)
    1 FORMAT(1H0,10X,'LISTAGEM DA SAIDA DE DISCO DO CROSS-MONTADOR')
      READ(8'NR) IVAC
      NFIM=IVAC(8)*20
      GO TO 6
    5 READ(8'NR) IVAC
    6 DO 15 JT=1,80,10
      DO 300 I=1,10
      JM=JM+1
      JI=I+JT-1
      CALL WZITR(IVAC(JI),J,K,L,M)
      NIT(I,1)=INIC(J)
      NIT(I,2)=INIC(K)
      NIT(I,3)=INIC(L)
      NIT(I,4)=INIC(M)
  300 CONTINUE
      IF(JM-NFIM)310,310,311
  310 WRITE(5,7600) JJ,((NIT(I,L),L=1,4),I=1,10)
 7600 FORMAT(1H ,I6,10(5X,4A1))
   15 JJ=JJ+10
      GO TO 5
  311 KLL=10-(JM-NFIM)
      WRITE(5,7600)JJ,((NIT(I,L),L=1,4),I=1,KLL)
   20 CALL EXIT
      END

UNREFERENCED STATEMENTS
 20

FEATURES SUPPORTED
 ONE WORD INTEGERS
 IOCS

CORE REQUIREMENTS FOR
 COMMON       0  VARIABLES     196  PROGRAM     334
```

END OF COMPILATION 002

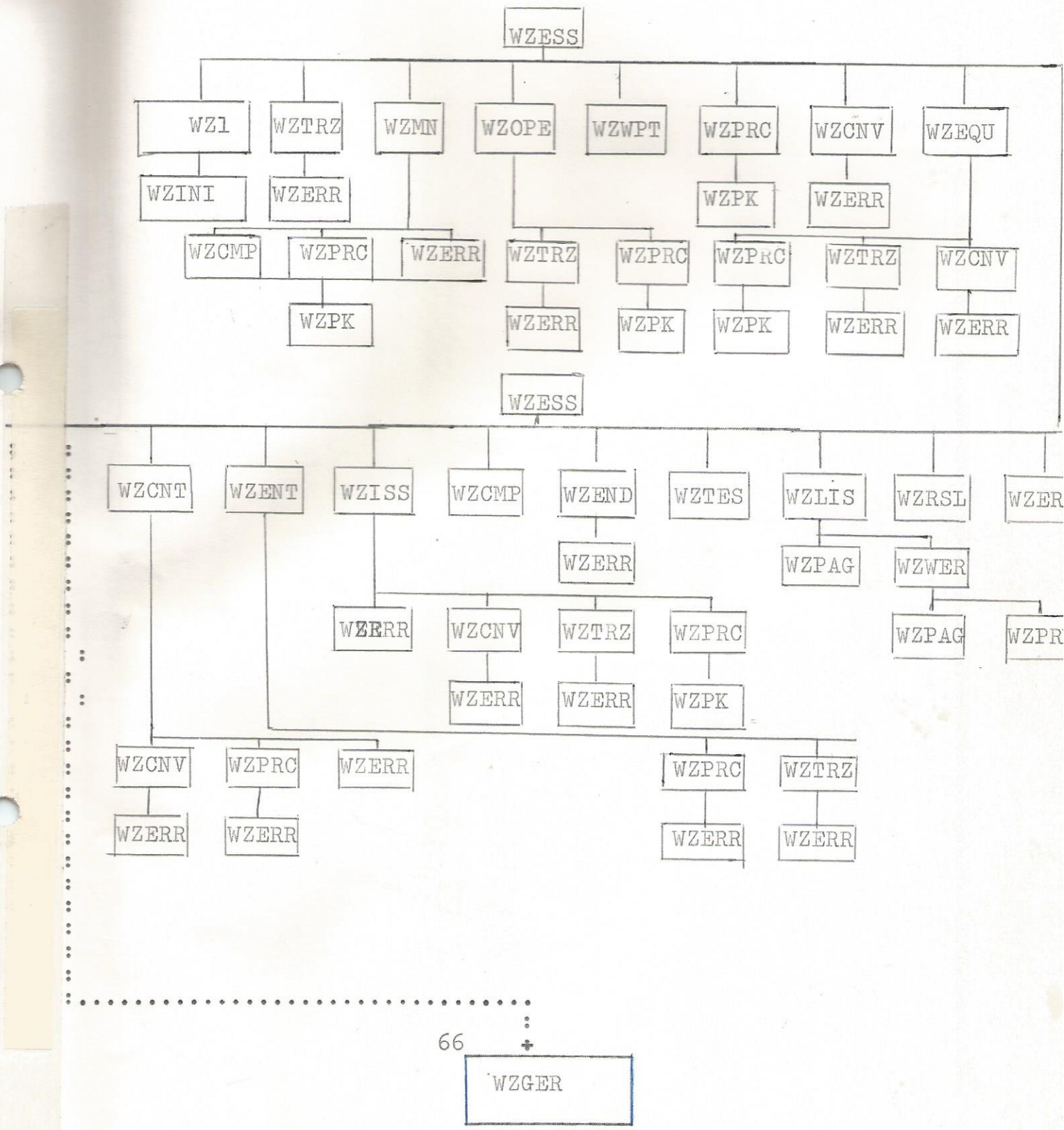
```
// DUP

*DELETE              LSTDR      0010
CART ID 0010   DB ADDR  1BE5   DB CNT   001C

*STORE       WS  UA  LSTDR            0010
CART ID 0010   DB ADDR  1D2B   DB CNT   001C

// XEQ WZESS
```

```
PAGE   1

// JOB    0001 0010                                        003    *     E

LOG DRIVE   CART SPEC   CART AVAIL  PHY DRIVE
  0000        0001        0001        0000
  0001        0010        0010        0002
                          302F        0001

V2 M09   ACTUAL 16K  CONFIG 16K

* LABORATORIO DE SISTEMAS DIGITAIS * EPUSP *  31/ 10/ 74  * 11 H

// DUP

*DEFINE FIXED AREA        0005      0010
CART ID 0010  CYLS FXA  0029  DBS AVAIL 0570  FLET SECTOR ADDR  10

*DELETE              WZN07     0010
CART ID 0010   DB ADDR  0100   DB CNT   0080

*DELETE              WZN06     0010
CART ID 0010   DB ADDR  0880   DB CNT   0190

*DELETE              WZN08     0010
CART ID 0010   DB ADDR  0180   DB CNT   00F0

*DELETE              WZN05     0010
D 26 NAME NOT FOUND IN LET/FLET


*STOREDATA  WS  FX  WZN06 0025      0010
CART ID 0010   DB ADDR  0370   DB CNT   0190

*STOREDATA  WS  FX  WZN07 0008      0010
CART ID 0010   DB ADDR  0500   DB CNT   0080

*STOREDATA  WS  FX  WZN08 0015      0010
CART ID 0010   DB ADDR  0580   DB CNT   00F0

*STOREDATA  WS  FX  WZN05 0100      0010
CART ID 0010   DB ADDR  0670   DB CNT   0640

// *
```

6 - WZN06
9 - WZN05
8 - WZN08
7 - WZN07

ÁRVORE DE ROTINAS CHAMADAS PELO PROGRAMA

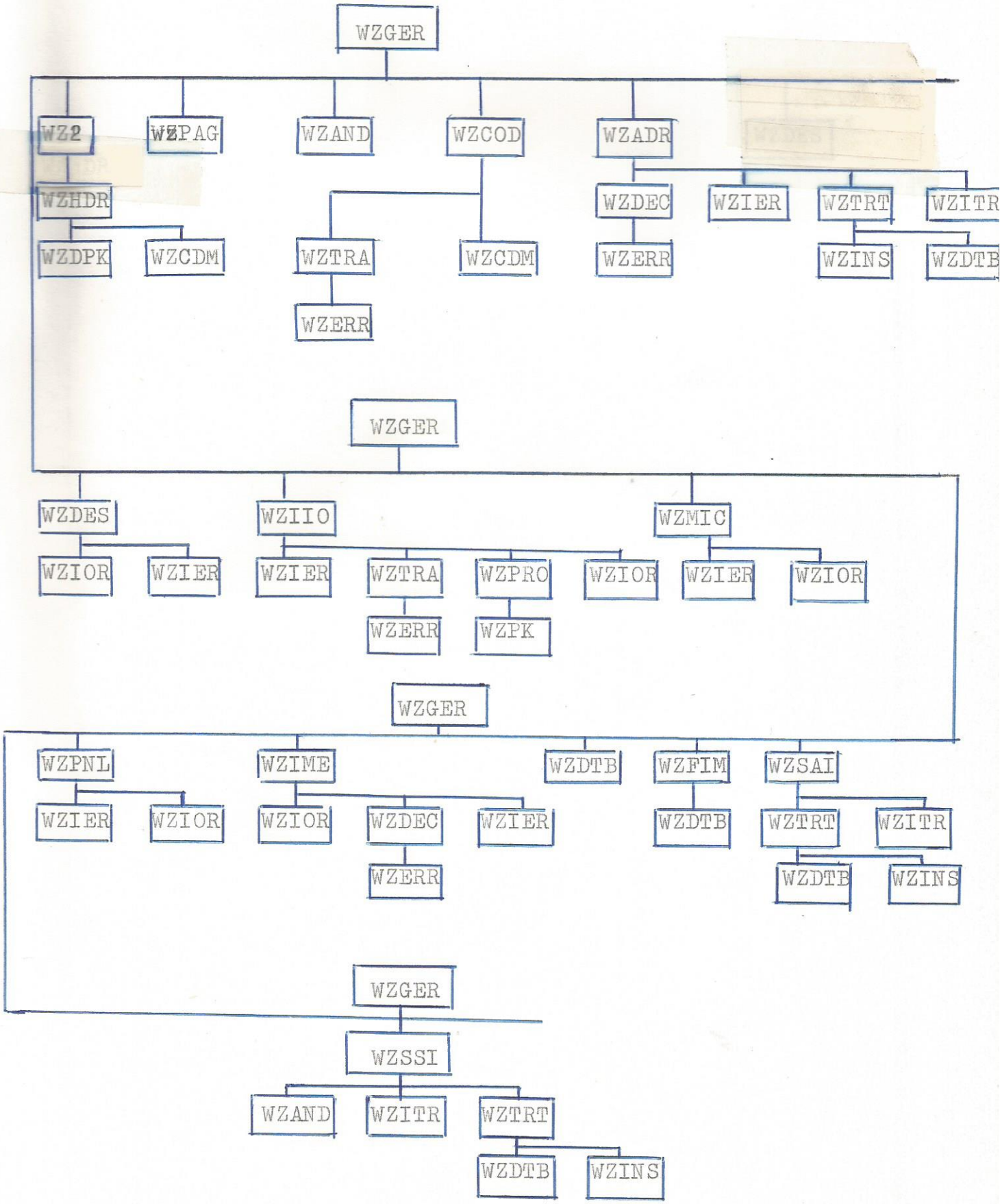
A diferença de nível não indica hierarquia, mas simplesmente facilidade de esquematização.



..... chamada por CALL LINK

ÁRVORE DE ROTINAS CHAMADAS PELO PROGRAMA

A diferença de nível não indica hierarquia, mas simplesmente facilidade de esquematização.



ESPECIFICAÇÃO DAS VARIÁVEIS EM COMMON

PRIMEIRA FASE

VARIÁVEL	SIGNIFICADO
IPAG	Número da página
KONT	Contador de linhas/página
IPRIN	Váriavel lógica, que indica impressora
ICAB(37)	Matriz de cabeçalho
ID5	Váriável lógica que indica arquivo de cartões
ID6	Idem para arquivo de informação suplementar
ID7	Idem para erros
ID8	Arquivo de saída de disco
N5	Variável do arquivo ID5
N6	Variável do arquivo ID6
N7	Variável do arquivo ID7
IAPSI	Ponteiro da tabela de símbolos
IAPEQ	Ponteiro da tabela de equivalência
NOEQU(100)	Tabela de equivalência
NOSIM(400,5)	Tabela de Símbolos
IENSI(400)	Tabela de endereços da Tabela de símbolos
IORIG	Endereço de carregamento do programa
ISTRT	Temporário para reter este valor
ICNT	Controle de imp. cartões de cont./última pseu
LST	Controle de listagem "ent" definida
ISWER	-------
NUM(16)	Matriz de números de 0 a 15
INIC(62)	Matriz dos caracteres permitidos
ICOL(320)	Buffer de leitura de cartões
ISTEC(90)	Buffer de informação suplementar
IAPTC	Ponteiro do buffer de informação suplementar
IS	-----
IERR	Contador de erros
IAPER	Ponteiro para o buffer de erros
IERRA(80)	Buffer de erros
IND	----
IAPIN	----
IRET	----
IAC	----
ICOMP	----
IABS	indica se o programa é absoluto ou relocável
IPENT	indica ponto de execução do programa

ESPECIFICAÇÃO DAS VARIÁVEIS EM COMMON

SEGUNDA FASE

VARIÁVEL	SIGNIFICADO
IPAG	Número de página
KONT	Contador de linhas/página
IPRIN	Variável lógica que indica impressora
ICAB(37)	Matriz de cabeçalho
ID5	Variável lógica que indica arquivo de cart
ID6	Idem para arquivo de informação suplementa
ID7	Idem para erros
ID8	Arquivo de saída de disco
N5	Variável associada ao arquivo ID5
N6	Variável associada ao arquivo ID6
N7	Variável associada ao arquivo ID7
IAPSI	Ponteiro para a tabela de símbolos
IAPEQ	Ponteiro para a tabela de equivalência
NOEQU(100)	Tabela de equivalência / erros
NOSIM(400,5)	Tabela de símbolos
IENSI(400)	Endereços da Tabela de símbolos
IORIG	Ponteiro pra a tabela de erros desta fase
ISTRT	Endereço de carregamento de um bloco
ICNT	Indicador de listagem de comandos de cart.
LST	Indicador de listagem cont.
ISWER	indica à tabela de erros qual o passo.
NUM(16)	Tabela de números de 0 a 15
INIC(62)	Tabela de valores permitidos
ISAI(320)	Buffer da saída de disco
ISTEC(90)	Buffer de informação suplementar
IAPTC	Ponteiro para o buffer de informação suple
IS	Contadora de palavras no bloco de dados
IERR	Indicador de erro no comando
IAPER	Sinalizador de erros
IERRA(80)	Buffere de erros
IND	Indicador de relocação
IAPIN	Ponteiro para o buffer de saída de disco
IRET	Palavra que contém um conjunto de ind. rel.
IAC	Contadora de palavras em um bloco de dados
ICOMP	Palavras necessárias para encher buffer de disco
IABS	Indica se o programa é absoluto
IPENT	Indica ponto de execução do programa.

ROTINA DE TRATAMENTO DA FASE DE LOCAÇÃO OU PRIMEIRO PASSO

NOME DO PROGRAMA	CÓDIGO DE ATIVIDADE NUM. DO DOCUMENTO	CÓDIGO DO PROGRAMA
PROGRAMA DE TRATAMENTO DAS INSTRUÇÕES NA PRI:- MEIRA FASE		WZESS
	PROGRAMADOR	/ / DATA

LINGUAGEM USADA

FORTRAN [X] ASSEMBLER [ ] BASIC [ ] ALGOL [ ] .... [ ]

TIPO DE PROGRAMA PRINCIPAL X

DRIVER [ ] SUBROUTINE [ ] FUNCTION [ ] OUTROS [ ]

IDENTIFICAÇÃO DOS DISCOS (FITAS)	SISTEMA SUPORTE (SIST. OPERACIONAL E HARDWARE)
0010	

CHAVES DO PAINEL OU SAIDAS ALTERNATIVAS

CHAVE 1

DESCRIÇÃO DO PROGRAMA

Este programa faz o tratamento das instruções na primeira fase do processamento. As suas principais funções são:

- leitura dos comandos e sua gravação em disco.
- determinação do tipo do comando, o que é feito através de rotina de pesquisa em tabela de símbolos.
- determinação da existência de rótulos das instruções.
- verificação de existência de operando nas instruções, e tratamento adequado em cada caso.
- verificação de indicação de indireto nos comandos e tratamento correspondente.
- tratamento dos comentários.
- atualização do contador de instruções.
- gerenciamento do buffer de informação suplementar, bem como parte de sua geração.
- tratamento de algumas pseudos específicas, como: BLOC, COM, ORG, DEFE, DN, EBC, ASC, END, EXT, ABS.
- determinação do número de erros da primeira fase, e no caso de ocorrência destes, cancelamento do programa ao fim desta fase, com listagem de comandos e erros.
- vistoria da tabela de símbolos, com possível cancelamento do programa.
- chamada do programa tratador da segunda fase.

 (use o verso, se necessário)

CÓDIGO DO PROGRAMA
CALL WZESS

SEQUÊNCIA DE CHAMADA E DESCRIÇÃO DAS VARIÁVEIS - ARGUMENTO

// XEQ WZESS 1
*FILES(9,N05),(6,N06),(7,N07),(8,N08)
se em formato DSF

// XEQ WZESS
se em formato DCI

SUB-ROTINAS USADAS

WZ1, WZTRZ, WZPRC,
WZMN, WZOPE, WZWPT,
WZCNV, WZEQU, WZCNT,
WZENT, WZISS, WZCMP,
WZERR, WZEND, WZTES,
WZLIS

OBSERVAÇÕES

OU MENSAGENS DE ERROS

Podem ocorrer os seguintes erros:
ERRO 18 - nome definido por external é rótulo do programa.
ERRO 3 - rótulo não é seguido de brancos.
ERRO 4 - falta de operando em uma instrução onde é obrigatório
ERRO 5 - campo de indireto contém caracter inválido.
ERRO 6 - instrução ABS não é a primeira do programa
ERRO 7 - instrução contém operando inválido
ERRO 12 - sintaxe errada em comando EBC ou ASC
ERRO 14 - pseudo contém rotulo não permitido.
ERRO 19 - pseudo de controle contém rótulo.
ERRO 23 - pseudo ORG contém origem maior que 4095
ERRO 26 - pseudo COM não contém rótulo, ou então o mesmo com EQU
ERRO 35 - pseudo contém operando inválido, ou então não contém operando obrigatório.
~~ERRO 38 - falta operando ORG~~.
ERRO 17 - endereçamento indireto em instrução não permitida.
ERRO 27 - pseudo DEFE contém endereçamento indireto.
ERRO 20 - falta o ponto de entrada de um programa principal.

 (use o verso, se necessário)

CÓDIGO DE PROGRAMA

WZESS

VARIÁVEIS PRINCIPAIS - APENAS AS QUE NÃO CONSTAM DO COMMON

TIPO
(Dimension, Common, Real, Integer, Data, etc)

Nome	Tipo	Função da Variável
ISTEP	I	contador de instruções
IENTR	I	número de pontos de entrada de uma subrotina
IFP	I	ponteiro para a tabela de external
IABS	I	indica se 1, programa absoluto
IKCRD	I	conta cartões de comandos
L1,L2	I,II	respectivamente posição inicial e final do rótulo
M	I	indica quando 1, ocorrencia de erro
IIND	I	indica quando 1, ocorrencia de indireto
INMNE	I	indicador de tipo de comando
IOPTG	I	posição a analisar no operando
LW	I	tamanho do item no buffer de inf. supl.
ICK	I	indica ocorrencia de pseudo ABS, ou ORG
NS	I	indica ocorrencia de pseudo ISS
KR	I	ultimo entry point de subrotina na tabela de simbolos
IFPX	I	tabela de external
IPAUT	I	caracter indicador de "não erro"

ANEXOS: DIAGRAMA DE BLOCOS [ ]

LISTAGEM DO PROGRAMA [ ]

LISTAGEM DO EXEMPLO [ ]

CABARITO DOS CARTÕES DE ENTRADA OU DA IMPRESSORA [ ]

FOLHA DE ALTERAÇÃO VERSÃO 1 [ ]

VERSÃO = [ ]

```
// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
*IOCS(CARD,1403PRINTER,DISK)
C     PRIMEIRA PARTE
C
C     ****************************************************************
C    *                MONTA A TABELA DE SIMBOLOS.
C    *                DESCOBRE O MNEMONICO
C    *                MONTA TABELAS ASSOCIADAS
C    ****************************************************************
C     INMNE - INDICADOR DE MNEMONICO
C           = 1 - MNEMONICO
C           = 2 - PSEUDO
C           = 3 - MACRO
C           = 4 - FIM
C           = 5 - COMENTARIO
C           = 6 - ERRO
C     ICK - INDICADOR DE ABSOLUTO OU RELOCAVEL
      DEFINE FILE   9(400,80,U,N5)
      DEFINE FILE 6(100,80,U,N6)
      DEFINE FILE 7(30,80,U,N7)
      DEFINE FILE 8(4800,1,U,NR)
      REAL IFPX(50)
      DIMENSION ISTRP(80),MAT(4)
      COMMON IPAG,KONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEQ,NOEQU(100),NOSIM(400,5),IENSI(400)
      COMMON IORIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ICOL(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
      COMMON IABS,IPENT,KERRR
      EQUIVALENCE (IIS,INIC(4))
      EQUIVALENCE (ICONT,IERR),(IBCO,INIC(38)),(IESTR,INIC(45))
      EQUIVALENCE (ISTRP(1),ISTEC(1)),(IPONT,INIC(1)),(IPAUT,INIC(44))
      EQUIVALENCE (ICNT,KR),(IENTR,LST),(IAC,NSTAT,NS)
      DATA IBAR /'/'/
      DATA ICARD,MK,ISTEP,ICK /2,1,2*0/
      DATA IKCRD/0/
C
C     INICIALIZACAO
C
      KR=0
      IENTR=0
      IFP=0
      IABS=0
      CALL WZ1
C
C     COMECA O PROCESSAMENTO
C
      NF=1
  200 READ(ICARD,10)(ICOL(I),I=1,80)
   10 FORMAT(80A1)
      IKCRD=IKCRD+1
C
C
      WRITE(ID5'N5)( ICOL(I),I=1,80)
C
```

```
C
  209 IF(ICOL(21)-IESTR) 210,4300,210
  210 IF(ICOL(21)-IBCO)220,310,220
C
C     A INSTRUCAO CONTEM LABEL
C
  220 L1=21
      CALL WZTRZ(L1,L2,1,ISW1,M)
      IF(M)230,280,230
C
C     OCORREU ERRO
C
  230 L1 = 0
  280 I=L2+2
      GO TO 320
C
C     INSTRUCAO SEM LABEL
C
  310 I=22
      L1 = 0
C
C     VERIFICA SE O LABEL E SEGUIDO DE BRANCOS
C
  320 IF(I-27) 330,410,410
  330 DO 400 K=I,26
      IF(ICOL(K)-IBCO)405,400,405
  400 CONTINUE
C
C     VERIFICA BRANCOS SEGUINDO  O CODIGO
C
  410 JIND=0
      IF(ICOL(31)-IBCO) 510,507,510
  507 IF(ICOL(33)-IBCO) 510,508,510
  508 IF(ICOL(34)-IBCO) 510,509,510
  509 IF(ICOL(32)-IIS ) 512,511,512
  512 IF(ICOL(32)-IBCO) 510,520,510
  511 JIND=1
      ISTEP=ISTEP+1
C
C     BUSCA DO MNEMONICO
C     ESCOLHE A ROTINA ESPECIFICA
C
  520 CALL WZMN(IK,L1,L2,ISTEP,IFP,IFPX)
      INMNE=ISTEC(IAPTC+1)
      IOPTG=ISTEC(IAPTC+3)
      GO TO (2000,750,9000,2000,2000,841),INMNE
C
C     ERRO
C
  840 ISTEC(IAPTC+1)=6
  841 LW=2
      GO TO 6900
C
C     COMENTARIO
C
 4300 ISTEC(IAPTC+1)=5
      IKCRD=IKCRD-1
```

vê se é coment ou tem rot

013

014

```
      LW=2
      GO TO 6900
C
C
C     INSTRUCOES COM OPERANDOS SIMBOLICOS
C
 1200 CALL WZOPE(ISTEP,JIND)
      JIND=0
      ISTEP=ISTEP+2
      GO TO 4200
C     TESTE DE EXISTENCIA DE OPERANDO PARA MNEMONICO
C
 2000 IF(ICOL(IOPTG)-IBCO) 2010,2020,2010
 2010 GO TO (2050,2050,2050,2060,2050,2050,2050,2050,2060,2050),IK
 2020 GO TO (2070,2070,2050,2050,2070,2070,2070,2070,2050,2070),IK
 2050 GO TO (1200,845,845,2100,1200,2100,2100,845,845,2100),IK
C
C     INCREMENTA CI DE 1 - INSTRUCAO CURTA
C
 2100 ISTEP = ISTEP + 1
      ISTEC(IAPTC+4)=1
      GO TO 846
C     INCREMENTA CI DE 2 - INSTRUCAO LONGA
C
  845 ISTEP=ISTEP+2
      ISTEC(IAPTC+4)=2
  846 ISTEC(IAPTC)=IAPTC+5
      IF(JIND) 2105,855,2105
  855 IAPTC=IAPTC+5
 4200 CONTINUE
C
C     VERIFICA SE VETOR DE INFORMACAO SUPLEMENTAR ESTA COMPLETO
C
      IF(IAPTC-80) 4211,4211,4220
 4211 NF=IAPTC
      GO TO 4210
 4220 ISTEC(NF)=IAPTC-80
      WRITE(ID6'N6) ISTRP
      DO 4221 I=81,IAPTC
 4221 ISTEC(I-80)=ISTEC(I)
      IAPTC=IAPTC-80
      NF=IAPTC
 4210 CALL WZWPT(IPAUT)
      GO TO (200,8000),MK
C
C     PSEUDO FOI ENCONTRADA
C
  750 IENMN=ISTEC(IAPTC+2)
      GO TO(760,760,760,760,770,770,770,770,770,770,760,760,751,760,76
     1      770,760,760,760,760,760,760,751,751) ,  IENMN
C
C     DESVIO A ROTINA DE TRATAMENTO DA PSEUDO
C
  751 GO TO (3200,3000,845,5100,6199,6199,6199,6199,6199,6199,3300,310
     1      3800,3400,3600,6500,5000,5500,6199,5500,5620,7100),IK
C
C     TESTES DE OPERANDOS PARA PSEUDOS
```

015

```
C
  760 IF(ICOL(IOPTG)-IBCO)751,761,751
  776 IF(L1)777,845,777
  777  CALL WZPRC(1,K,ISTEP,L1,L2)
      NOSIM(K,4)=1
      GO TO 845
  770 IF(ICOL(IOPTG)-IBCO)761,751,761
C
C     TRATA DA PSEUDO 'BLOCO'
C
 3000 CALL WZCNV(IOPTG-1,N,K1,K2,1,M,38)
      LW=5
      IF(M)6800,3010,6800
 3010 IF(L1)3030,3050,3030
 3030 CALL WZPRC (1,LK,ISTEP,L1,L2)
      NOSIM(LK,5)=N
      IF(NOSIM(LK,4)) 3050,3040,3050
 3040 NOSIM(LK,4)=4
 3050 ISTEP=ISTEP+N
      ISTEC(IAPTC+4)=N
      GO TO 6800
C
C     TRATA DA PSEUDO EQUIVALENCE
C
 3100 LW=4
      IF(L1) 3110,3305,3110
 3110 CALL WZEQU(L1,L2,ISTEP)
      GO TO 6800
C
C     TRATA DA PSEUDO DE DEFINICAO
C
 3200 CALL WZCNT(L1,L2,ISTEP)
      LW=6
      ISTEP=ISTEP+1
      GO TO 6800
C
C     TRATA DA PSEUDO COMMON
C     NO CASO DA PSEUDO COMMON A ROTINA DETERMINA O NUMERO DE POSICOES
C     A SEREM USADAS. SE A INSTRUCAO NAO CONTIVER TAL NUMERO, O NOME
C     NAO E INSERIDO NA TABELA.
C
 3300 LW=4
      IF(L1)3310,3305,3310
 3310 CALL WZPRC(1,K,ISTEP,L1,L2)
      CALL WZCNV(IOPTG-1,N,K1,K2,1,M,38)
      NOSIM(K,5)=N
      IF(M) 4200,3330,4200
 3330 IF(NOSIM(K,4)) 6800,3331,6800
 3331 NOSIM(K,4)=3
      GO TO 6800
C
C     TRATA DA PSEUDO 'ORG'
C
 3400 ICK=1
      IABS=1
      IF(ICOL(IOPTG)-IBAR) 3401,3433,3401
 3401 CALL WZCNV(IOPTG-1,N,K1,K2,1,M,38)
```

```
 3402 LW=6
      IF(M) 6200,3440,6200
 3433 CALL WZCNV(IOPTG,N,K1,K2,0,M,38)
      GO TO 3402
 3440 IF(N-4096) 3450,3470,3470
 3450 ISTEP=N
      ISTEC(IAPTC+4)=N
 3460 GO TO 6201
C
C     TRATA DA PSEUDO ENTRY
C
 3600 CALL WZENT(KR,IENTR,NSTAT)
      LW=4
      GO TO 6200
C
C     TRATA DA PSEUDO ISS
C
 5000 CALL WZISS(IKCRD)
      LW=4
      GO TO 6200
C
C     TRATA DA PSEUDO DEFEN
C     EQUIVALE A 'PLA'
C
 5100 IF(JIND) 5102,5101,5102
 5101 ISTEC(IAPTC+1)=1
      ISTEC(IAPTC+2)=20
      IF(L1)5105,5107,5105
 5105 CALL WZPRC(1,K,ISTEP,L1,L2)
      IF(NOSIM(K,4)) 5107,5106,5107
 5106 NOSIM(K,4)=1
      GO TO 5107
 5107 IK=1
      GO TO 2000
C
C     DEFINE NOME
C
 5620 CALL WZTRZ(IOPTG,IFIM,ISW,ISW1,M)
      LW=4
      IF(M) 6200,5630,6200
 5630 CALL WZPRC(1,K,ISTEP,IOPTG,IFIM)
      IF(NOSIM(K,4))6200,5631,6200
 5631 NOSIM(K,4)=7
      GO TO 6200
C
C     PSEUDO EBC
C
 5500 IF(ICOL(IOPTG)-IPONT ) 5510,5520,5510
 5520 IOPTG=IOPTG+1
      K=IOPTG
      DO 5530 J=1,36
      JJ=73-J
      IF(ICOL(JJ  )-INIC(1))5530,5540,5530
 5530 CONTINUE
      GO TO 5510
 5540 J=37-J
      ISTEC(IAPTC+4)=1
```

```
      ISTEC(IAPTC+5)=J
      LW=6
      ISTEP=ISTEP+J
      IF(L1)5441,6800,5441
 5441 CALL WZPRC(1,LK,ISTEP,L1,L2)
      IF(NOSIM(LK,4)) 6800,5542,6800
 5542 NOSIM(LK,4)=1
      GO TO 6800
C
C     TRATA DA PSEUDO 'END'
C
 3800 ISTEC(IAPTC+1)=4
      LW=2
      IF(NS) 3801,3802,3801
 3802 IF(KR) 3801,3880,3801
 3880 IF(ICOL(35)-IBCO)  3881,208,3881
 3881 L3=35
      CALL WZTRZ(L3,L2,1,ISW1,M)
      IF(M) 3870,3882,3870
 3882 CALL WZPRC(2,K,ISTEP,L3,L2)
      KK=NOSIM(K,4)+1
      GO TO(208,3884,3884,208,208,208,208,208,3884,3884,208),KK
 3884 IPENT=IENSI(K)
      GO TO 3801
 3801 IF (L1)  3860,3870,3860
 3870 MK=2
      GO TO 6800
C
C     EXTERNAL
C
 7100 IF(ICOL(35)-IBCO)7101,7135,7101
 7101 L3=35
      CALL WZTRZ(L3,L2,1,ISW1,M)
      IF(M)7119,7103,7119
 7103 IF(L2-L3-3)7106,7106,7135
 7106 CALL WZPRC(2,K,ISTEP,L3,L2)
      IF(NOSIM(K,4))7135,7109,7135
 7109 IF(K-IAPSI)7135,7112,7135
 7112 IAPSI=IAPSI-1
      DO 7114 II=1,4
 7114 MAT(II)=IBCO
      IT=1
      DO 7117 II=L3,L2
      MAT(IT)=ICOL(II)
 7117 IT=IT+1
      CALL WZCMP(F,MAT(1),MAT(2),MAT(3),MAT(4))
      IFP=IFP+1
      IFPX(IFP)=F
 7119 GO TO 6200
 7135 CALL WZERR(18)
      GO TO 6200
C
C     TESTE DE LABEL
C
 6199 IKCRD=IKCRD-1
 6200 LW=4
 6201 IF(L1) 6205,6800,6205
```

```
C
C     PSEUDO ABS
C
 6500 IF(IKCRD-1) 3405,6505,3405
 6505 ICK=1
      IABS=1
      LW=4
      GO TO 6201
C
C
C
 6800 IF(JIND) 6805,6900,6805
 6900 ISTEC(IAPTC)=IAPTC+LW
      IAPTC=IAPTC+LW
      GO TO 4200
C
 8000 CALL WZWPT(IPAUT)
      WRITE (ID6'N6)   ISTRP
      IORIG=ICK
      INREL=0
C
C     FINALIZACAO DA PRIMEIRA FASE
C
      IF(IERR )100,201,100
  100 M=0
      GO TO 250
C
C     NAO OCORRERAM ERROS NA PRIMEIRA FASE ATE A FINALIZACAO
C     TERMINA PREPARO DA TABELA DE SIMBOLOS
C
  201 CONTINUE
      IAPER=IAPER-1
      CALL WZEND(IERR)
      CALL WZWPT(IPAUT)
C
C     VERIFICA SE OCORRERAM ERROS NO PREPARO FINAL DA TABELA DE SIMBOLO
C
      IF(IERR ) 100,231,100
  231 CALL WZTES(N)
      ISTRT=ISTEP
      M=1
C
C     VERIFICA DEFINICAO DOS  SIMBOLOS DA TABELA DE SIMBOLOS
C
      IF(N)250,900,250
  250 IERRA(1)=IAPER-1
      WRITE(ID7'N7) IERRA
      CALL WZLIS(INREL,ISTEP)
      IF(M-1) 899,257,899
  257 CALL WZRSL
  899 WRITE(IPRIN,110) IERR
  110 FORMAT(1H0,10X,'TOTAL DE ERROS NO PROGRAMA = ',I4,/,11X,
     1       'PROGRAMA DESCONTINUADO')
C
C
C
      CALL EXIT
```

```
  900 CALL LINK(WZGER)
C
C
C
 9000 ISTEC(IAPTC+1)=2
      ISTEC(IAPTC+2)=3
      IK=3
      ISTEC(IAPTC+3)=27
      GO TO 845
C
C     TRATAMENTO DOS ERROS
C
  208 CALL WZERR(20)
      GO TO 3801
  405 CALL WZERR(3)
      GO TO 840
  510 CALL WZERR(5)
      GO TO 840
  761 CALL WZERR(35)
      IF(IENMN-13) 840,3800,840
 2060 CALL WZERR(4)
      GO TO 840
 2070 CALL WZERR(7)
      GO TO 840
 2105 CALL WZERR(17)
      GO TO 855
 3470 CALL WZERR(23)
      GO TO 6200
 3305 CALL WZERR(26)
      GO TO 6800
 5510 CALL WZERR(12)
      GO TO 840
 3860 CALL WZERR(19)
      GO TO 3870
 6205 CALL WZERR(14)
      GO TO 6800
 3405 CALL WZERR(6)
      GO TO 6505
 6805 CALL WZERR(17)
      GO TO 6900
 5102 CALL WZERR(27)
      GO TO 840
      END

UNREFERENCED STATEMENTS
 209      776      3460

FEATURES SUPPORTED
 ONE WORD INTEGERS
 IOCS

CORE REQUIREMENTS FOR
 COMMON    3134  VARIABLES     174  PROGRAM    1720

END OF COMPILATION

// DUP
```

```
*DELETE              WZESS     0010
CART ID 0010   DB ADDR  1F00   DB CNT   01F0
```

020

```
*STORECI    WS  UA  WZESS    1      0010
*FILES(6,WZN06),(7,WZN07),(8,WZN08),(9,WZN05)
FILES ALLOCATION
    6 0037  0019  0010 WZN06
    7 0050  0008  0010 WZN07
    8 0058  000F  0010 WZN08
    9 0067  0064  0010 WZN05
STORAGE ALLOCATION
R 41  0BBE (HEX) WDS UNUSED BY CORE LOAD
CALL TRANSFER VECTOR
 WZPRN  275E
 WZWER  261F
 WZPAG  2600
 WZIER  2591
 WZPK   2560
 WZINI  2466
 WZRSL  2219
 WZLIS  210D
 WZTES  2095
 WZEND  1F49
 WZERR  1F13
 WZCMP  1EFA
 WZISS  1E6E
 WZENT  1DD9
 WZCNT  1D0D
 WZEQU  1B34
 WZCNV  1A43
 WZPRC  191D
 WZWPT  18DB
 WZOPE  1733
 WZMN   15D1
 WZTRZ  13D3
 WZ1    1335
LIBF TRANSFER VECTOR
 SIOAI  0D28
 FARC   2732
 EBCTB  272F
 HOLTB  26E9
 GETAD  2696
 NORM   266C
 SDRED  09BE
 FSUB   24FA
 FLDX   1310
 FSUBX  24FF
 SUBIN  24DA
 HOLEZ  2426
 CARD1  2324
 FLOAT  231A
 IFIX   22EE
 PAUSE  22C3
 SCOMP  0CFB
 SIOI   0D22
 SWRT   0C1A
```

```
 FSTOX  12F4
 SDAI   099C
 SDCOM  0A3A
 SDIX   09A9
 SDWRT  09C3
 SIOIX  0CB1
 SUBSC  13A0
 SRED   0C1F
 FSTO   12F8
 FLD    1314
 PRNZ   1196
 CARDZ  10C0
 SFIO   0D37
 SDFIO  0A22
SYSTEM SUBROUTINES
 ILS04  00C4
 ILS02  00B3
 ILS00  276D
      02EE (HEX) IS THE EXECUTION ADDR
CART ID 0010   DB ADDR  2260   DB CNT   01F0

// EJECT
```

PROGRAMA DE TRATAMENTO DA SEGUNDA FASE
FASE DE GERAÇÃO OU SEGUNDO
PASSO

NOME DO PROGRAMA	CÓDIGO DE ATIVIDADE NUM. DO DOCUMENTO	CÓDIGO DO PROGRAMA
PROGRAMA DE GERAÇÃO DE CÓDIGO		WZGER
	PROGRAMADOR	/ / DATA

LINGUAGEM USADA

FORTRAN [X] ASSEMBLER [ ] BASIC [ ] ALGOL [ ] .... [ ]

TIPO DE PROGRAMA PRINCIPAL

DRIVER [ ] SUBROUTINE [X] FUNCTION [ ] OUTROS [ ]

IDENTIFICAÇÃO DOS DISCOS (FITAS)	SISTEMA SUPORTE (SIST. OPERACIONAL E HARDWARE)
0010	

CHAVES DO PAINEL OU SAIDAS ALTERNATIVAS

CHAVE 1

DESCRIÇÃO DO PROGRAMA

ESte programa tem como funções as seguintes, que correspondem a fase de geração de código do programa:

- preparação do buffer da saída de disco, correspondente ao programa em forma relocável, destinado a ser composto pelo carregador.
- gerenciamento dos buffers de leitura de cartões de disco, de informação suplementar, destinados a completar a análise iniciada na primeira fase.
- impressão da listagem de saída do programa.
- realização do controle de paginação da listagem.
- controle da listagem através de pseudos.
- montagem das instruções e chamadas de subrotinas.
- tratamento das cadeias de caracteres, definidas pelas pseudos EBC, ASC.
- controle dos erros que ocorram nesta fase. Cancelamento do programa neste caso.

 (use o verso, se necessário)

CÓDIGO DO PROGRAMA

WZGER

SEQUÊNCIA DE CHAMADA E DESCRIÇÃO DAS VARIÁVEIS - ARGUMENTO

Chamado por WZESS através de CALL LINK

SUB-ROTINAS USADAS

WZ2, WZPAG, WZADR
WZDES, WZIIO, WZPNL
WZMIC, WZIME, WZCOD
WZAND, WZSAI, WZSSI
WZITR, WZFIM, WZDTB

OBSERVAÇÕES [illegible] OU MENSAGENS DE ERROS

Não imprime mensagens de erro (não gera)

 (use o verso, se necessário)

CÓDIGO DE PROGRAMA

WZGER

VARIÁVEIS PRINCIPAIS

TIPO
(Dimension, Common, Real, Integer, Data, etc)

Nome	Tipo	Função da Variável
ISTEP	I	contador de instruções
IRET	I	PAlavra que contém os indicadores de relocaçãp
IND	I	indicador de relocação
KER	I	contador de erros
ISTRT	I	endereço de carregamento
JJ	I	ordem de entrada do cartão
IVOLT	I	indicador de fim de programa
ICOMN	I	mascara da instrução
IOCT	I	palavra onde é montada a instrução
SWITC	I	chave de impressão
KONT	I	contador de linhas
LST	I	indicador de listar programa (1)
ICNT	I	indicador de listar pseudos de controle (1)
NIPO	I	matriz de caracteres ASCII

ANEXOS: DIAGRAMA DE BLOCOS [ ]

LISTAGEM DO PROGRAMA [ ]

LISTAGEM DO EXEMPLO [ ]

GABARITO DOS CARTÕES DE ENTRADA OU DA IMPRESSORA [ ]

FOLHA DE ALTERAÇÃO VERSÃO 1 [ ]

VERSÃO = [ ]

```
// *
```



```
// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
*IOCS(CARD,1403PRINTER,DISK)
C
C     PROGRAMA DE GERACAO DO CODIGO
C
      INTEGER SWITC
        DIMENSION ISTRP(80)
      DIMENSION IASC(62)
      DIMENSION NERRO(50),IDESV(50),IPSEU(24),ICOMN(50),ICO(49)
      DIMENSION NY(24),NX(49),NIPO(100),ICOL(80)
      DEFINE FILE 8(4800,1,U,NR)
      DEFINE FILE   9(400,80,U,N5)
      DEFINE FILE 6(100,80,U,N6)
      DEFINE FILE 7(30,80,U,N7)
      COMMON IPAG,KONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEQ,NOEQU(100),NOSIM(400,5),IENSI(400)
      COMMON IORIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ISAI(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
      COMMON IABS,IPENT,KERRR
      EQUIVALENCE (ICO(1),ICOMN(1)),(NX(1),IDESV(1)),(NY(1),IPSEU(1))
      EQUIVALENCE (ICOL(1),IERRA(1)),(NERRO(1),NOEQU(1))
      EQUIVALENCE (NR,N7),(IKONT,IORIG)
      EQUIVALENCE (KER,IAPER),(IBCO,INIC(38)),(IDISK,ID7)
      EQUIVALENCE (ISTRP(1),ISTEC(1))
      DATA SWITC /0/
      DATA IASC/Z2E00,Z4700,Z4800,Z4900,Z4A00,Z4B00,Z4C00,Z4D00,Z4E00,
     1 Z4F00,Z5000,Z5100,Z5200,Z5300,Z5400,Z5500,Z5600,Z5700,Z5800,Z59
     2,Z5A00,Z4100,Z4200,Z4300,Z4400,Z4500,Z4600,Z3000,Z3100,Z3200,Z33
     3,Z3400,Z3500,Z3600,Z3700,Z3800,Z3900,Z2000,Z2B00,Z2D00,Z2800,Z2C
     4,Z2900,Z2300,Z2A00,Z3D00,Z2F00,Z3C00,Z3E00,Z2400,Z4000,Z5F00,Z00
     5,Z2100,Z2600,Z3A00,Z3B00,Z5E00,Z2700,Z3F00,Z2200,Z2100 /
      DATA ICO / ZDA00,Z8200,ZD800,ZD400,ZB000,ZA000,Z8000,Z8300,Z1000
     1      Z9D00,Z3000,Z5000,Z7000,ZC040,Z9B00,Z9100,Z9500,Z8600,Z9A00
     2      Z0000,ZF000,Z2C00,Z99C0,Z4000,Z9400,Z9E00,ZD200,ZD100,Z8500
     3      Z8100,Z6000,Z8800,ZC080,ZC020,ZE000,Z9C00,ZD180,ZD110,ZD140
     4      ZD150,ZD120,ZD160,ZC010,Z9C00,Z8400,ZD100,Z9800,ZD130,ZD170
      DATA NY/6,11,4,2,6*3,2*1,7,12,3*1,10,9,10,1,5,2*8/
      DATA NX/2,7,2,2,2*1,  4,7,1,7,3*1,6,7,2*8,2*7,3*1,7,1,8,7,2,3,2*
     1          1,5,2*6,1,8,6*3,6,4*7,2*3/
      DATA        IVOLT,JJ/  1,0/
C
C     INICIALIZACAO
C
      CALL WZ2(ISTEP,IABS,IGLU)
      KERRR=0
 9191 FORMAT(1H ,20I6)
      IN3=0
      JJ=1
C
C     GERACAO DE UM COMANDO
C
      IRET=0
```

```
  392 READ(IC5'N5) ICOL
      INMNE=ISTEC(IAPTC+1)
      IENMN=ISTEC(IAPTC+2)
      IOPTG=ISTEC(IAPTC+3)
C
C     VERIFICA SE NAO E PSEUDO
C
      IF(INMNE-2)401,400,401
    1 CALL WZPAG
   10 IND=0
      N2=0
      KER=0
      GO TO 6
C
C     PSEUDOS DE TRATAMENTO ESPECIAL SE NA PRIMEIRA INSTRUCAO
C
  401 IF(INMNE-5)1,700,1
  400 IF(IENMN-19) 450,390,450
  450 CALL WZPAG
      IF(IENMN-16)500,700,500
  500 IF(IENMN-14) 10,600,10
  390 J=1
      DO 391 I=35,71
      ICAB(J)=ICOL(I)
  391 J=J+1
      JJ=JJ+1
      GO TO 392
C
C     PSEUDO ORG
C
  600 JJ=JJ+1
      ISTEP=ISTEC(IAPTC+4)
      ISTRT=ISTEP
      GO TO 2100
C
C     COMANDO ABS
C
  700 JJ=JJ+1
      WRITE(IPRIN,7510)JJ,ICOL
      KONT=KONT+1
      READ(IC5'N5) ICOL
      IAPTC=ISTEC(IAPTC)
      INMNE=ISTEC(IAPTC+1)
      IENMN=ISTEC(IAPTC+2)
      IOPTG=ISTEC(IAPTC+3)
C
C     VERIFICA SE NOVO COMANDO E PSEUDO
C
      IF(INMNE-2)1,500,1
    2 IF(IVOLT-2)5,8000,5
C
C     PROCESSAMENTO
C
    5 IND=0
      IAPTC=ISTEC(IAPTC)
      IF(IN3)17,9,17
    9 N2=0
```

```
      KER=0
      JJ=JJ+1
      IF(ISTEC(IAPTC)-IAPTC) 8,8,4
    8 READ(ID6'N6)(ISTEC(I),I=81,90)
      N6=N6-1
      IN3=1
    4 READ(ID5'N5) ICOL
C
C     PESQUISA DOS VARIOS TIPOS DE INSTRUCOES
C
      INMNE=ISTEC(IAPTC+1)
      IENMN=ISTEC(IAPTC+2)
      IOPTG=ISTEC(IAPTC+3)
    6 GO TO (1000,2000,3000,2700,2310,6000),INMNE
   17 READ(ID6'N6) ISTRP
      IN3=0
      GO TO 9
C
C     TRATAMENTO DO MNEMONICO
C
 1000 IOCT=ICOMN(IENMN)
      IK=IDESV(IENMN)
      GO TO (1100,1200,1300,1400,1500,1600,7000,1800),IK
C
C
 1100 CALL WZADR(IOCT,ISTEP,IABS)
C     INSTRUCOES ENVOLVENDO ENDERECAMENTO
      GO TO 7000
C
C     INSTRUCOES COM OPERANDOS IMEDIATOS
C
 1200 CALL WZIME(IOCT,ISTEP)
      GO TO 5000
C
C     INSTRUCOES DE DESLOCAMENTO
C
 1300 CALL WZDES(IOCT)
      GO TO 5000
C
C     INSTRUCOES DE 'LIMP'
C
 1400 CALL WZMIC(IOCT,0)
      GO TO 7000
C
C     INSTRUCOES DE PAINEL
C
 1500 CALL WZPNL(IOCT)
      GO TO 7000
C
C     INSTRUCOES DE ENTRADA E SAIDA
C
 1600 CALL WZIIO(IOCT,ISTEP)
      GO TO 5000
C
C     INSTRUCOES DE SALTOS
C
 1800 CALL WZMIC(IOCT,1)
```

```
      GO TO 7000
C
C     TRATAMENTO DAS PSEUDOS
C
 2000 IK=IPSEU(IENMN)
      GO TO (2100,6000,2300,2400,2,2600,2700,2100,2390,2500,7100,
     1       7700),IK
C
C     LISTA O COMANDO SIMPLESMENTE
C
 2100 WRITE(IPRIN,7510)JJ,ICOL
 7510 FORMAT(1H ,I5,6X,12X,80A1)
C
C     CONTROLE DE PAGINACAO
C
 2105 I=1
 2106 KONT=KONT+1
      IF(KONT-53) 2115,2110,2110
 2110 CALL WZPAG
 2115 IF(KER) 2119,2116,2119
 2116 IF(SWITC)2   ,2,2117
 2117 IOPTG=IOPTG+1
      DO 7071 I=21,80
 7071 ICOL(I)=IBCO
      GO TO 2505
C
C     IMPRESSAO DOS ERROS
C
 2119 WRITE(IPRIN,2131)NERRO(I)
 2131 FORMAT(1H ,40X,'ERRO NO. ',I2)
 2140 KER=KER-1
      KONT=KONT+1
      I=I+1
      GO TO 2106
C
C     PSEUDO CONTROLE
C
 2300 GO TO (2350,2350,2350,2350,2360,2370,2320,2330,2340,2350),
     1       IENMN
 2320 LST = 1
      GO TO 2380
 2330 LST = 0
      GO TO 2380
 2340 ICNT = 1
      GO TO 2380
 2350 ICNT = 0
      GO TO 2380
 2360 WRITE(IPRIN,2361)
 2361 FORMAT (1H )
      KONT=KONT+1
      IF(KONT-53) 2380,2370,2370
C
C     PSEUDO CABECALHO
C
 2390 J=1
      DO 2395 I=35,71
      ICAB(J)=ICOL(I)
```

```
 2395 J=J+1
 2370 CALL WZPAG
 2380 IF(ICNT) 2310,2,2310
 2310 IF(LST)2100,2,2100
C
C     COMANDO CALL
C
 2400 CALL WZCOD(N1,N2)
      IND=3
      IK=1
      IOCT=N1
      GO TO 7000
C
C     DEFINICAO DE EBCDIC
C
 2500 SWITC=ISTEC(IAPTC+5)
      IND=0
      IOPTG=IOPTG+1
      IF(IENMN-20)2550,2501,2550
 2550 DO 5011 I=IOPTG,72
      DO 5100 JK=1,62
      IF(ICOL(I)-INIC(JK)) 5100,5200,5100
 5100 CONTINUE
 5200 NIPO(I)=IASC(JK)
 5011 CONTINUE
      GO TO 2505
 2501 DO 2506 I=IOPTG,72
 2506 NIPO(I)=ICOL(I)
 2505 IOCT=NIPO(IOPTG)
      CALL WZAND(IOCT)
      N2=0
      IK=1
      GO TO 7021
C
C     DEFINICAO DE CONSTANTE
C
 2600 IOCT=ISTEC(IAPTC+5)*256
      IK=1
      GO TO 7020
C
C     PSEUDO 'END'
C
 2700 IVOLT=2
      GO TO 2100
C
C     ABERTO PARA MACRO
C
 3000 GO TO 2700
C
C     SEM CODIGO
C
 6000 CALL EXIT
 5000 CALL WZSSI(IOCT,ISTEP,JJ,N2,ISTEC(IAPTC+4))
      GO TO 2105
 7000 CONTINUE
C
C     SAIDAS
```

```
C
 7020 CONTINUE
      IK=ISTEC(IAPTC+4)
 7021 CALL WZSAI(IOCT,N2,ISTEP,IK,JJ,SWITC)
      GO TO 2105
C
C     TRATAMENTO DOS BLOCOS
C
 7500 FORMAT(1H ,I5,2X,4A1,2X,4A1,6X,80A1)
 7100 CALL WZITR(ISTEP,K1,K2,K3,K4)
      IF(IKONT  ) 7110,7120,7110
 7110 IND=IAPIN-IKONT+1
      ISAI(IND)=IRET*2**(18-2*IKONT)
 7120 CONTINUE
      ISTEP=ISTEP+ISTEC(IAPTC+4)
      CALL WZFIM(ISTEP,K)
      IF(K) 7125,7124,7125
 7124 CALL WZDTB(ISTEP)
      IS=IS-2
 7125 IKONT=0
      WRITE(IPRIN,7500)JJ,INIC(K1),INIC(K2),INIC(K3),INIC(K4),
     1            IBCO,IBCO,IBCO,IBCO,ICOL
      GO TO 2
C
C     TRATAMENTO DOS ORG
C
 7700 IF(IKONT  ) 7710,7720,7710
 7710 IND= IAPIN-IKONT+1
      ISAI(IND)=IRET*2**(18-2*IKONT)
      IKONT=0
 7720 CONTINUE
      ISTEP=ISTEC(IAPTC+4)
      CALL WZFIM(ISTEP,K)
      IF(K) 7100,7225,7100
 7225 CALL WZDTB(ISTEP)
      IS=IS-2
      GO TO 2100
C
C     DATA BREAK DE FIM DE PROGRAMA
C
 8000 CONTINUE
      CALL WZFIM(ISTEP,K)
C
C     CALCULO DO PROXIMO ENDERECO LIVRE DA MEMORIA
C
      I=0
      M=ISTEP/2
      M=ISTEP-2*M
      IF(M)8070,8080,8070
 8070 ISTEP=ISTEP+1
 8080 ISAI(IAPIN-1)=ISTEP
      ISAI(IAPIN)=I
      WRITE(IDISK'NR) ISAI
      READ(IDISK'1) ISAI
C
C     CALCULO DO NUMERO DE DISK BLOCKS OCUPADOS PELO PROGRAMA
C
```

```
      IC=IS/20
      IS=IS-IC*20
      IF(IS) 8100,8200,8100
 8100 IC=IC+1
      IF(KERRR) 9945,8200,9945
 9945 ISAI(8)=0
      ISAI(4)=0
      WRITE(IPRIN,9946) KERRR
 9946 FORMAT(1H0,10X,'TOTAL DE ERROS =',I4)
      GO TO 8201
 8200 ISAI(8)=IC
      ISAI(4)=ISTEP
 8201 WRITE(IDISK'1) ISAI
      CALL DATSW(1,N)
      IF(N)9947,9948,9947
 9947 CALL LINK(WZBTB)
 9948 CALL EXIT
C
C
      END

UNREFERENCED STATEMENTS
 9191    2140

FEATURES SUPPORTED
 ONE WORD INTEGERS
 IOCS

CORE REQUIREMENTS FOR
 COMMON   3134  VARIABLES    344  PROGRAM   1336

END OF COMPILATION

// DUP

*DELETE              WZGER     0010
CART ID 0010   DB ADDR  1F00   DB CNT   01E0

*STORECI    WS  UA  WZGER    1      0010
*FILES(6,WZNO6),(7,WZNO7),(8,WZNO8),(9,WZNO5)
FILES ALLOCATION
    6 0037  0019  0010 WZNO6
    7 0050  0008  0010 WZNO7
    8 0058  000F  0010 WZNO8
    9 0067  0064  0010 WZNO5
STORAGE ALLOCATION
R 41  0D4C (HEX) WDS UNUSED BY CORE LOAD
CALL TRANSFER VECTOR
 WZWPT  2587
 WZPK   2584
 WZINS  2552
 WZERR  252D
 WZDPK  250E
 WZCDM  2422
 WZPRO  2317
 WZTRA  21E7
```

033

```
 WZTRT  211F
 WZIER  20F5
 WZIOR  18BA
 WZDEC  2023
 WZHDR  1E26
 DATSW  1C18
 WZDTB  1BE8
 WZFIM  1B6B
 WZITR  1AC0
 WZSAI  19C0
 WZSSI  18F4
 WZAND  18C7
 WZCOD  1856
 WZIIO  1710
 WZPNL  16A4
 WZMIC  1650
 WZDES  15F2
 WZIME  158D
 WZADR  1365
 WZPAG  1314
 WZ2    1258
LIBF TRANSFER VECTOR
 EBCTB  250B
 HOLTB  24C5
 GETAD  2472
 NORM   2448
 SUBIN  1DE2
 HOLEZ  1DA2
 CARD1  1CA0
 FLOAT  1C96
 IFIX   1C6A
 PAUSE  1C3F
 SDCOM  0964
 SDWRT  08ED
 FIXI   1B23
 SIOIX  08DB
 SDIX   08D3
 SCOMP  0C25
 SIOAI  0C52
 SIOI   0C4C
 SWRT   0B44
 SUBSC  12A4
 SDAI   08C6
 SDRED  08E8
 FSTO   1222
 FLD    123E
 PRNZ   10C0
 CARDZ  0FEA
 SFIO   0C61
 SDFIO  094C
SYSTEM SUBROUTINES
 ILS04  00C4
 ILS02  00B3
 ILS00  25E9
      03A0 (HEX) IS THE EXECUTION ADDR
CART ID 0010   DB ADDR  2270   DB CNT  01E0

// EJECT
```

NOME DO PROGRAMA	CÓDIGO DE ATIVIDADE NUM. DO DOCUMENTO	CÓDIGO DO PROGRAMA
PROGRAMA PARA LISTAGEM DA TABELA DE SIMBOLOS		WZBTB
	PROGRAMADOR	/ / DATA

LINGUAGEM USADA

FORTRAN [X] ASSEMBLER [ ] BASIC [ ] ALGOL [ ]....[ ]

TIPO DE PROGRAMA PRINCIPAL X

DRIVER [ ] SUBROUTINE [ ] FUNCTION [ ] OUTROS [ ]

IDENTIFICAÇÃO DOS DISCOS (FITAS)	SISTEMA SUPORTE (SIST. OPERACIONAL E HARDWARE)
0010	

CHAVES DO PAINEL OU SAIDAS ALTERNATIVAS

CHAVE 1

DESCRIÇÃO DO PROGRAMA

Este programa é chamado atrves de CALL LINK pelo programa WZGER, quando a chave 1 se encontra posicionada, afim de fazer a listagem da tabela de simbolos.

 (use o verso, se necessário)

CÓDIGO DO PROGRAMA
WZBTB

SEQUÊNCIA DE CHAMADA E DESCRIÇÃO DAS VARIÁVEIS - ARGUMENTO

Chamado por CALL LINK pelo programa WZGER

SUB-ROTINAS USADAS

WZPAG
WZBH

OBSERVAÇÕES [illegible] OU MENSAGENS DE ERROS

NENHUMA

 (use o verso, se necessário)

CÓDIGO DE PROGRAMA

WZBTB

VARIÁVEIS PRINCIPAIS

TIPO
(Dimension, Common, Real, Integer, Data, etc)

Nome	Tipo	Função da Variável
		AS DO COMMON DA SEGUNDA FASE

ANEXOS: DIAGRAMA DE BLOCOS [ ]
LISTAGEM DO PROGRAMA [ ]
LISTAGEM DO EXEMPLO [ ]
GABARITO DOS CARTÕES DE ENTRADA OU DA IMPRESSORA [ ]
FOLHA DE ALTERAÇÃO VERSÃO 1 [ ]
VERSÃO = [ ]

037

```
// FOR
*ONE WORD INTEGERS
*LIST ALL
*IOCS(CARD,1403PRINTER,DISK)
      DIMENSION  MAT(4)
      COMMON IPAG,KONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEC,NOEQU(100),NOSIM(400,5),IENSI(400)
      COMMON IORIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ICOL(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
      COMMON IABS,IPENT,KERRR
C
C
      KONT=0
      CALL WZPGG
      IF(IAPSI)15,20,15
   20 WRITE(IPRIN,21)
   21 FORMAT(1HO, 40X,"TABELA VAZIA')
      CALL EXIT
C
   15 WRITE(IPRIN,10)
   10 FORMAT(1H , 40X,"TABELA DE SIMBOLOS" ,//,41X,'SIMBOLO  ENDEREGO')
      DO  30  I=1,IAPSI
      CALL WZBH(IENSI (I),MAT)
   16 FORMAT(1H , 40X, 3A2, 6X,4A1)
      KONT=KONT+1
      IF(KONT-55) 30,30,17
   17 CALL WZPGG
      WRITE(IPRIN,10)
   30 WRITE(IPRIN,16) NOSIM(I,1),NOSIM(I,2),NOSIM(I,3), MAT
      CALL EXIT
      END
VARIABLE ALLOCATIONS
  IPAG(IC)=7FFF        KONT(IC)=7FFE       IPRIN(IC)=7FFD        ICAB(IC)=7F
   ID7(IC)=7FD5          N5(IC)=7FD4          N6(IC)=7FD3          N7(IC)=7F
 NOEQU(IC)=7FCF-7F6C NOSIM(IC)=7F6B-779C IENSI(IC)=779B-760C IORIG(IC)=76
   LST(IC)=7608       ISWER(IC)=7607         NUM(IC)=7606-75F7  INIC(IC)=75
 IAPTC(IC)=741E          IS(IC)=741D        IERR(IC)=741C       IAPER(IC)=74
 IAPIN(IC)=73C9        IRET(IC)=73C8         IAC(IC)=73C7       ICOMP(IC)=73
 KERRR(IC)=73C3         MAT(I )=0003-0000      I(I )=0004

STATEMENT ALLOCATIONS
 21   =000B  10   =0016  16   =0031  20   =005D  15   =0062  17   =0084

FEATURES SUPPORTED
 ONE WORD INTEGERS
 IOCS

CALLED SUBPROGRAMS
 WZPGG   WZBH    FLD     FSTO    CARDZ   SWRT    SCOMP   SFIO    SIOAI

INTEGER CONSTANTS
     0=0008      1=0009     55=000A

CORE REQUIREMENTS FOR
 COMMON   3134  VARIABLES      8  PROGRAM    168
```

037

```
N6,N7
I(400)

TC
AC,ICOMP

IMBOLO  ENDERECO')

), MAT

FD        ICAB(IC)=7FFC-7FD8   ID5(IC)=7FD7        ID6(IC)=7FD6
03          N7(IC)=7FD2      IAPSI(IC)=7FD1      IAPEQ(IC)=7FD0
9B-760C IORIG(IC)=760B       ISTRT(IC)=760A       ICNT(IC)=7609
96-75F7  INIC(IC)=75F6-75B9  ICOL(IC)=75B8-7479 ISTEC(IC)=7478-741F
C       IAPER(IC)=741B       IERRA(IC)=741A-73CB   IND(IC)=73CA
7       ICOMP(IC)=73C6        IABS(IC)=73C5      IPENT(IC)=73C4
04

=0062  17   =0084  30   =008A

   SFIO    SIOAI   SIOIX   SUBSC   PRNZ    SDFIO
```

```
END OF COMPILATION

// DUP

*DELETE              WZBTB      0010
CART ID 0010   DB ADDR  1F00   DB CNT   00C0

*STORECI     WS  UA  WZBTB            0010
R 41  23D8 (HEX) WDS UNUSED BY CORE LOAD
CALL TRANSFER VECTOR
 WZBH    0CE4
 WZPGG   0CB0
LIBF TRANSFER VECTOR
 EBCTB   0F89
 HOLTB   0F43
 GETAD   0EF0
 NORM    0EC6
 SIOI    066C
 HOLEZ   0E86
 CARD1   0D84
 FLOAT   0D7A
 IFIX    0D4E
 PAUSE   0D23
 SIOAI   0672
 SIOIX   05FB
 SUBSC   0CC6
 SCOMP   0645
 SWRT    0564
 FSTO    0C42
 FLD     0C5E
 PRNZ    0AE0
 CARDZ   0A0A
 SFIO    0681
 SDFIO   036C
SYSTEM SUBROUTINES
 ILS04   00C4
 ILS02   00B3
 ILS00   0F8D
      0238 (HEX) IS THE EXECUTION ADDR
CART ID 0010   DB ADDR  2380   DB CNT   00C0

// EJECT
```

ROTINAS DE CARACTER GERAL

As rotinas constantes desta parte destinam-se a operações de caracter geral, principalmente a manipulação de bits e caracteres.

WZAND
WZIOR
WZCDM
WZPK
WZITR
WZCMP

NOME DO PROGRAMA	CÓDIGO DE ATIVIDADE NUM. DO DOCUMENTO	CÓDIGO DO PROGRAMA
PROGRAMA DE MASCARAMENTO - OR entre dois valores. - AND entre um valor e a mascara /FF00.		WZAND/WZDOR
	PROGRAMADOR	/ / DATA

LINGUAGEM USADA

FORTRAN [ ] ASSEMBLER [x] BASIC [ ] ALGOL [ ] .... [ ]

TIPO DE PROGRAMA PRINCIPAL

DRIVER [ ] SUBROUTINE [x] FUNCTION [ ] OUTROS [ ]

IDENTIFICAÇÃO DOS DISCOS (FITAS)	SISTEMA SUPORTE (SIST. OPERACIONAL E HARDWARE)
0010	

CHAVES DO PAINEL OU SAIDAS ALTERNATIVAS

nenhuma

DESCRIÇÃO DO PROGRAMA

O primeiro programa realiza a operação OU entre dois valores, guardando o resultado da operação no primeiro.

Ex:

1º valor : 1001 1000 1100 0010
2º valor : 0001 0010 1111 1000

1º valor : 1001 1010 1111 1010 (resultado)

O segundo programa, realiza a operação AND entre o primeiro valor e a máscara / FF00, isto é o segundo byte do valor é feito zero, e o primeiro é conservado.

Ex:

1º valor : 1001 1000 1100 0010
2º valor : 1111 1111 0000 0000

1º valor : 1001 1000 0000 0000 (resultado)

CÓDIGO DO PROGRAMA

WZAND / WZIOR

SEQUÊNCIA DE CHAMADA E DESCRIÇÃO DAS VARIÁVEIS - ARGUMENTO

```
CALL  WZIOR ( VAL1, VAL2)
CALL  WZAND ( VAL1, VAL2)
      VAL1 - primeiro valor
      VAL2 - segundo valor
```

SUB-ROTINAS USADAS

NENHUMA

OBSERVAÇÕES [illegible] OU MENSAGENS DE ERROS

NENHUMA

 (use o verso, se necessário)

CÓDIGO DE PROGRAMA

WZIOR / WZAND

VARIÁVEIS PRINCIPAIS

TIPO
(Dimension, Common, Real, Integer, Data, etc)

Nome	Tipo	Função da Variável
VT	ROTULO	
ST	ROTULO	
RET	ROTULO	
N	máscara	valor de / FF00, para mascarar o valor na rotina WZAND.

ANEXOS:

DIAGRAMA DE BLOCOS [ ]

LISTAGEM DO PROGRAMA [x]

LISTAGEM DO EXEMPLO [ ]

GABARITO DOS CARTÕES DE ENTRADA OU DA IMPRESSORA [ ]

FOLHA DE ALTERAÇÃO VERSÃO 1 [ ]

VERSÃO = [ ]

```
0000    26A49599 00001          ENT     WZIOR
000D    26A41544 00002          ENT     WZAND
                 00003    *    *WZIOR
                 00004    *
                 00005    *     ESTA ROTINA REALIZA OPERACAO 'OU' ENTRE 2
                 00006    *     VALORES, GUARDANDO O RESULTADO NO PRIEI-
                 00007    *     RO DELES.
                 00008    *    *WZAND
                 00009    *     ESTA ROTINA REALIZA OPERACAO 'AND' ENTRE
                 00010    *     2 VALORES. O RESULTADO FICA NO PRIMEIRO VALOR
                 00011    *
0000 0  0000     00012    WZIOR DC       0
0001 01 65800000 00013          LDX  I1 WZIOR
0003 00 C5800000 00014          LD   I1 0       CARREGA O PRIMEIRO VALOR
0005 00 ED800001 00015          OR   I1 1       FAZ 'OR' COM O SEGUNDO
0007 00 D5800000 00016    VT    STO  I1 0       ARMAZENA NO PRIMEIRO VALOR
0009 0  7102     00017          MDX   1 2
000A 0  6901     00018    ST    STX   1 RET+1
000B 00 4C000000 00019    RET   BSC  L  0
                 00020    *
000D 0  0000     00021    WZAND DC       0
000E 01 65800000 00022          LDX  I1 WZAND
0010 00 C5800000 00023          LD   I1 0
0012 0  E004     00024          AND     N
0013 00 D5800000 00025          STO  I1 0
0015 0  7101     00026          MDX   1 1
0016 0  70F3     00027          MDX     ST
0017 0  FF00     00028    N     DC      /FF00
0018             00029          END     WZIOR

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

// DUP

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


*STORE      WS  UA  WZIOR           0010
CART ID 0010   DB ADDR  02B4   DB CNT   0003

// EJECT
```

NOME DO PROGRAMA	CÓDIGO DE ATIVIDADE NUM. DO DOCUMENTO	CÓDIGO DO PROGRAMA
Transformação em "Name Code"		WZ@DM
	PROGRAMADOR	/ / DATA

LINGUAGEM USADA

FORTRAN [ ] ASSEMBLER [X] BASIC [ ] ALGOL [ ] .... [ ]

TIPO DE PROGRAMA

DRIVER [ ] SUBROUTINE [X] PRINCIPAL FUNCTION [ ] OUTROS [ ]

IDENTIFICAÇÃO DOS DISCOS (FITAS)	SISTEMA SUPORTE (SIST. OPERACIONAL E HARDWARE)

CHAVES DO PAINEL OU SAIDAS ALTERNATIVAS

DESCRIÇÃO DO PROGRAMA

O programa transforma o nome de um símbolo codificado em 5A1 em Name Code. A transformação é a seguinte:

Seja o nome: ABCDE

$A = C1_{16} = 11\ 00\ 0001 \longrightarrow \boxed{00}\ 00\ 0001$

$B = C2_{16} = 11\ 00\ 0010 \longrightarrow 00\ 0010$

$C = C3_{16} = 11\ 00\ 0011 \longrightarrow 00\ 0011$

$D = C4_{16} = 11\ 00\ 0100 \longrightarrow 00\ 0100$

$E = C5_{16} = 11\ 00\ 0101 \longrightarrow 00\ 0101$

Resulta: 0000 0001 0000 1000 $= (0108)_{16}$

0011 0001 0000 0101 $= (3105)_{16}$

 (use o verso, se necessário)

CÓDIGO DO PROGRAMA

WZCDM

SEQUÊNCIA DE CHAMADA E DESCRIÇÃO DAS VARIÁVEIS - ARGUMENTO

CALL WZCDM (MAT, NOME1, NOME2)
MAT — contém o nome codificado em 5A1
NOME1 — contém o nome em "name code" (1ºP)
NOME2 — contém o nome em "name code" (2ºP)

SUB-ROTINAS USADAS

—— // ——

OBSERVAÇÕES [illegible] OU MENSAGENS DE ERROS

—— // ——

 (use o verso, se necessário)

CÓDIGO DE PROGRAMA

WZCDM

VARIÁVEIS PRINCIPAIS

TIPO
(Dimension, Common, Real, Integer, Data, etc)

Nome	Tipo	Função da Variável

ANEXOS: DIAGRAMA DE BLOCOS [ ]

LISTAGEM DO PROGRAMA [ ]

LISTAGEM DO EXEMPLO [ ]

GABARITO DOS CARTÕES DE ENTRADA OU DA IMPRESSORA [ ]

FOLHA DE ALTERAÇÃO VERSÃO 1 [ ]

VERSÃO = [ ]

```
0000    26A43114 00001          ENT     WZCDM
                 00002    *
                 00003    *     ESTA ROTINA TRANSFORMA 5 CARACTERES EM FORMA
                 00004    *     TO A1 EM 2 PALAVRAS EM CODIGO M.
                 00005    *
0000 0  0000     00006    WZCDM DC      0
0001 01 66800000 00007          LDX  I2 WZCDM
0003 0  C200     00008          LD    2 0          CARREGA END. NM(1)
0004 0  8019     00009          A       UM
0005 0  D002     00010          STO     LOOP3+1
0006 0  61FB     00011          LDX   1 -5
0007 00 C5000000 00012    LOOP3 LD   L1 *-*        CARREGA NM(I)
0009 0  1808     00013          SRA     8
000A 0  E014     00014          AND     H3F
000B 01 D5000025 00015          STO  L1 IN+5
000D 0  7101     00016          MDX   1 +1
000E 0  70F8     00017          MDX     LOOP3
000F 0  10A0     00018          SLT     32           LIMPA ACC
0010 0  61FB     00019          LDX   1 -5
0011 0  18C6     00020    LOOP4 RTE     6
0012 01 ED000025 00021          OR   L1 IN+5
0014 0  7101     00022          MDX   1 1
0015 0  70FB     00023          MDX     LOOP4
0016 0  18D8     00024          RTE     24
0017 00 D6800001 00025          STO  I2 1
0019 0  1090     00026          SLT     16
001A 00 D6800002 00027          STO  I2 2
001C 00 4E000003 00028          BSC  L2 3
001E 0  0001     00029    UM    DC      1
001F 0  003F     00030    H3F   DC      /3F
0020    0005     00031    IN    BSS     5
0026              00032          END     WZCDM

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

// DUP

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


*STORE      WS  UA  WZCDM           0010
CART ID 0010   DB ADDR  02CD   DB CNT   0003

// EJECT
```

NOME DO PROGRAMA	CÓDIGO DE ATIVIDADE NUM. DO DOCUMENTO	CÓDIGO DO PROGRAMA
ROTINA DE COMPACTAÇÃO A1 A A4.		WZPK
	PROGRAMADOR	/ / DATA

LINGUAGEM USADA

FORTRAN [ ] ASSEMBLER [x] BASIC [ ] ALGOL [ ] .... [ ]

TIPO DE PROGRAMA

DRIVER [ ] SUBROUTINE [x] PRINCIPAL FUNCTION [ ] OUTROS [ ]

IDENTIFICAÇÃO DOS DISCOS (FITAS)	SISTEMA SUPORTE (SIST. OPERACIONAL E HARDWARE)
0010	

CHAVES DO PAINEL OU SAIDAS ALTERNATIVAS

NENHUMA

DESCRIÇÃO DO PROGRAMA

Este programa compacta 4 caracteres lidos em formato A1 em um caracter em A4, e um em A1.

OS valores fornecidos são destruídos, sendo os valores calculados retornados sobre os dois primeiros valores da matriz.

Ex:

C1 40	C3 40	C9 40	F4 40	F8 40	valores de entrada
C1 C3	C9 F4	F8 40	F4 40		valores de saída

1 (use o verso, se necessário)

Os valores iniciais da matriz são destruidos e os valor

CÓDIGO DO PROGRAMA

WZPK

SEQUÊNCIA DE CHAMADA E DESCRIÇÃO DAS VARIÁVEIS - ARGUMENTO

CALL WZPK (MATR)

MATR - matriz onde vem os caractres e onde retorna o valor compactado

SUB-ROTINAS USADAS

NENHUMA

OBSERVAÇÕES OU MENSAGENS DE ERROS

NENHUMA

 (use o verso, se necessário)

	CÓDIGO DE PROGRAMA
	WZPK

VARIÁVEIS PRINCIPAIS

Nome	Tipo (Dimension, Common, Real, Integer, Data, etc)	Função da Variável
RET	ROTULO	
M	I,	guarda o endereço do dado tratado
TEMP	I	guarda resultados intermediários dos calculos

ANEXOS:

DIAGRAMA DE BLOCOS [ ]

LISTAGEM DO PROGRAMA [X]

LISTAGEM DO EXEMPLO [ ]

GABARITO DOS CARTÕES DE ENTRADA OU DA IMPRESSORA [ ]

FOLHA DE ALTERAÇÃO VERSÃO 1 [ ]

VERSÃO = [ ]

```
0000    26A57480 00001          ENT     WZPK
                 00002    *
                 00003    *     ESTA RO TINA COMPACTA FORMATO A1 EM FORMATO A
                 00004    *
0000 0  0000     00005    WZPK  DC      0
0001 01 65800000 00006         LDX  I1 WZPK
0003 00 C5000000 00007         LD   L1 0     CARREGA END. DA MATRIZ
0005 0  D026     00008         STO     M
0006 01 C480002C 00009         LD   I  M     CARREGA M(1)
0008 0  D024     00010         STO     TEMP
0009 01 74FF002C 00011         MDX  L  M,-1
000B 01 C480002C 00012         LD   I  M     CARREGA M(2)
000D 0  1808     00013         SRA     8
000E 0  E81E     00014         OR      TEMP FAZ OR COM M(1)
000F 01 74010020 00015         MDX  L  M,1
0011 01 D480002C 00016         STO  I  M
0013 01 74FD002C 00017         MDX  L  M,-3
0015 01 C480002C 00018         LD   I  M     CARREGA M(4)
0017 0  1808     00019         SRA     8
0018 01 74010020 00020         MDX  L  M,+1
001A 01 EC80002C 00021         OR   I  M     FAZ OR COM M(3)
001C 01 74010020 00022         MDX  L  M,1
001E 01 D480002C 00023         STO  I  M
0020 01 74FD002C 00024         MDX  L  M,-3
0022 01 C480002C 00025         LD   I  M
0024 01 74020020 00026         MDX  L  M,2
0026 01 D480002C 00027         STO  I  M
0028 0  7101     00028         MDX   1 1
0029 0  6901     00029         STX   1 RET+1
002A 00 4C000000 00030    RET  BSC  L  0
002C 0  0000     00031    M    DC      0
002D 0  0000     00032    TEMP DC      0
002E             00033         END     WZPK

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

// DUP

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


*STORE      WS  UA  WZPK            0010
CART ID 0010   DB ADDR  02C9   DB CNT   0004

// EJECT
```

NOME DO PROGRAMA	CÓDIGO DE ATIVIDADE NUM. DO DOCUMENTO	CÓDIGO DO PROGRAMA
ROTINA DE C ONVERSÃO BINáRIO PARA HEXEDECIMAL.		WZITR
	PROGRAMADOR	/ / DATA

LINGUAGEM USADA

FORTRAN [ ] ASSEMBLER [x] BASIC [ ] ALGOL [ ]....[ ]

TIPO DE PROGRAMA PRINCIPAL

DRIVER [ ] SUBROUTINE [x] FUNCTION [ ] OUTROS [ ]

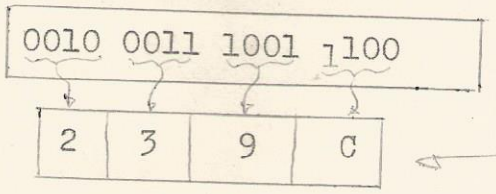
IDENTIFICAÇÃO DOS DISCOS (FITAS)	SISTEMA SUPORTE (SIST. OPERACIONAL E HARDWARE)
0010	

CHAVES DO PAINEL OU SAIDAS ALTERNATIVAS

DESCRIÇÃO DO PROGRAMA

Esta rotina recebe um número binário e o transforma em quatro caracteres hexadecimais . O valor a ser descompactado é recebido em uma variável, recebida como parâmetro, e retorna ao programa chamante, um resultado em uma matriz de saída.

Ex:



valor de entrada

matriz de saída

 (use o verso, se necessário)

CÓDIGO DO PROGRAMA

WZITR

SEQUÊNCIA DE CHAMADA E DESCRIÇÃO DAS VARIÁVEIS - ARGUMENTO

CALL WZITR (IBINA, NN,NM,NO,NP)

IBINA - valor binário

NN - primeiro caracter hexadecimal

NM - segundo caracter hexadecimal

NO - terceiro caracter hexadecimal

NP - quarto caracter hexadecimal

SUB-ROTINAS USADAS

NENHUMA

OBSERVAÇÕES ~~ERRICAMENTO~~ OU MENSAGENS DE ERROS

NENHUMA

 (use o verso, se necessário)

CÓDIGO DE PROGRAMA

WZITR

VARIÁVEIS PRINCIPAIS

TIPO
(Dimension, Common, Real, Integer, Data, etc)

Nome	Tipo	Função da Variável
FIM	ROTULO	
IOCT	I	armazena o valor binário a ser tratado
IOCTT	I	endereço de IOCT
AND1	I	máscara para calcular o primeiro caracter hexa /F000
AND2	I	máscara para calcular o segundo caracter hexa /0F00
AND3	I	máscara para calcular o terceiro caracter hexa /00F0
AND4	I	máscara para calcular o quarto caracter hexa /000F

ANEXOS:

DIAGRAMA DE BLOCOS [ ]

LISTAGEM DO PROGRAMA [X]

LISTAGEM DO EXEMPLO [ ]

GABARITO DOS CARTÕES DE ENTRADA OU DA IMPRESSORA [ ]

FOLHA DE ALTERAÇÃO VERSÃO 1 [ ]

VERSÃO = [ ]

```
0000     26A498D9 00001          ENT     WZITR
                  00002    *
                  00003    *    ROTINA DE CONVERSAO DE BINARIO PARA HEXADECI
                  00004    *
0000 0   0000     00005    WZITR DC       *-*
0001 01  65800000 00006          LDX  I1 WZITR
0003 00  C5000000 00007          LD   L1 0
0005 01  D4000042 00008          STO  L  IOCTT
0007 00  C5000001 00009          LD   L1 1
0009 0   D03B     00010          STO     I1
000A 00  C5000002 00011          LD   L1 2
000C 0   D039     00012          STO     I1+1
000D 00  C5000003 00013          LD   L1 3
000F 0   D037     00014          STO     I1+2
0010 00  C5000004 00015          LD   L1 4
0012 0   D035     00016          STO     I1+3
0013 01  C4800042 00017          LD   I  IOCTT
0015 0   D02D     00018          STO     IOCT
0016 0   E032     00019          AND     AND1
0017 0   180C     00020          SRA     12
0018 0   D02B     00021          STO     END
0019 01  66800044 00022          LDX  I2 END
001B 01  C6000040 00023          LD   L2 MAT
001D 01  D4800045 00024          STO  I  I1
001F 0   C023     00025          LD      IOCT
0020 0   E029     00026          AND     AND2
0021 0   1888     00027          SRT     8
0022 0   D021     00028          STO     END
0023 01  66800044 00029          LDX  I2 END
0025 01  C6000040 00030          LD   L2 MAT
0027 01  D4800046 00031          STO  I  I1+1
0029 0   C019     00032          LD      IOCT
002A 0   E020     00033          AND     AND3
002B 0   1884     00034          SRT     4
002C 0   D017     00035          STO     END
002D 01  66800044 00036          LDX  I2 END
002F 01  C6000040 00037          LD   L2 MAT
0031 01  66800044 00038          LDX  I2 END
0033 01  D4800047 00039          STO  I  I1+2
0035 0   C00D     00040          LD      IOCT
0036 0   E015     00041          AND     AND4
0037 0   D00C     00042          STO     END
0038 01  66800044 00043          LDX  I2 END
003A 01  C6000040 00044          LD   L2 MAT
003C 01  D4800048 00045          STO  I  I1+3
003E 0   7105     00046          MDX   1 5
003F 0   6901     00047          STX   1 FIM+1
0040 00  4C000000 00048    FIM   BSC  L  0
0042     0001     00049    IOCTT BSS     1
0043     0001     00050    IOCT  BSS     1
0044     0001     00051    END   BSS     1
0045     0004     00052    I1    BSS     4
0049 0   F000     00053    AND1  DC      /F000
004A 0   0F00     00054    AND2  DC      /0F00
004B 0   00F0     00055    AND3  DC      /00F0
004C 0   000F     00056    AND4  DC      /000F
004D 0   001C     00057    MAT   DC      28
```

056

```
004E 0  001D     00058          DC      29
004F 0  001E     00059          DC      30
0050 0  001F     00060          DC      31
0051 0  0020     00061          DC      32
0052 0  0021     00062          DC      33
0053 0  0022     00063          DC      34
0054 0  0023     00064          DC      35
0055 0  0024     00065          DC      36
0056 0  0025     00066          DC      37
0057 0  0016     00067          DC      22
0058 0  0017     00068          DC      23
0059 0  0018     00069          DC      24
005A 0  0019     00070          DC      25
005B 0  001A     00071          DC      26
005C 0  001B     00072          DC      27
005E             00073          END     WZITR

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

// DUP

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


*STORE      WS  UA  WZITR           0010
CART ID 0010   DB ADDR  02F8   DB CNT   0006

// EJECT
```

NOME DO PROGRAMA COMPACTAÇÃO A1 EM A4	CÓDIGO DE ATIVIDADE NUM. DO DOCUMENTO	CÓDIGO DO PROGRAMA WZCMP
	PROGRAMADOR	/ / DATA

LINGUAGEM USADA

FORTRAN [ ] ASSEMBLER [x] BASIC [ ] ALGOL [ ]....[ ]

TIPO DE PROGRAMA

DRIVER [ ] SUBROUTINE [x] PRINCIPAL FUNCTION [ ] OUTROS [ ]

IDENTIFICAÇÃO DOS DISCOS (FITAS)	SISTEMA SUPORTE (SIST. OPERACIONAL E HARDWARE)
0010	

CHAVES DO PAINEL OU SAIDAS ALTERNATIVAS

/

DESCRIÇÃO DO PROGRAMA

Esta rotina compacta 4 caracteres A1 em um caracter A4. Os caracteres A1 vem em quatro variáveis, e a variável compactada retorna em uma variável real.

Ex:

C1 40	C4 40	F5 40	D6 40

← entrada

C1 C4 F5 D6

← saída

 (use o verso, se necessário)

CÓDIGO DO PROGRAMA

WZCMP

SEQUÊNCIA DE CHAMADA E DESCRIÇÃO DAS VARIÁVEIS - ARGUMENTO

CALL WZCMP (F, MAT1, MAT2, MAT3, MAT4)

F - variável onde retorna o valor binário compactado

MAT1 - primeira variável a compactar

MAT2 - segunda variável a compactar

MAT3 - terceira variável a compactar

MAT4 - quarta variável a compactar

SUB-ROTINAS USADAS

NENHUMA

OBSERVAÇÕES [illegible] OU MENSAGENS DE ERROS

NENHUMA

 (use o verso, se necessário)

CÓDIGO DE PROGRAMA

VARIÁVEIS PRINCIPAIS

TIPO
(Dimension, Common, Real, Integer, Data, etc)

Nome	Tipo	Função da Variável

ANEXOS: DIAGRAMA DE BLOCOS [ ]
LISTAGEM DO PROGRAMA [ ]
LISTAGEM DO EXEMPLO [ ]
GABARITO DOS CARTÕES DE ENTRADA OU DA IMPRESSORA [ ]
FOLHA DE ALTERAÇÃO VERSÃO 1 [ ]
VERSÃO = [ ]

```
PAGE   2                        WZCMP

0000     26A43517 00001            ENT      WZCMP
                  00002     *
                  00003     *      ROTINA DE COMPACTACAO A1 EM A4
                  00004     *
0000 0   0000     00005     WZCMP DC       *-*
0001 01  65800000 00006           LDX   I1 WZCMP
0003 00  C5800002 00007           LD    I1 2
0005 0   1808     00008           SRA      8
0006 00  ED800001 00009           OR    I1 1
0008 0   D00D     00010           STD      F
0009 00  C5800004 00011           LD    I1 4
000B 0   1808     00012           SRA      8
000C 00  ED800003 00013           OR    I1 3
000E 0   D008     00014           STD      F+1
000F 0   C806     00015           LDD      F
0010 00  DD800000 00016           STD   I1 0
0012 0   7105     00017           MDX    1 5
0013 0   6901     00018           STX    1 FIM+1
0014 00  4C000000 00019     FIM   BSC   L  0
0016     0002     00020     F     BSS   E  2
0018              00021           END      WZCMP

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

// DUP

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


*STORE      WS  UA  WZCMP           0010
CART ID 0010   DB ADDR  02C6   DB CNT   0003

// EJECT
```

```
// *

// ASM
*LIST
0000     26A445D2 00001           ENT      WZDPK
0000 0   0000     00002    WZDPK DC       *-*
0001 01  65800000 00003           LDX  I1  WZDPK
0003 00  C5000000 00004           LD   L1  0
0005 0   8016     00005           A        UM
0006 0   D014     00006           STO      TEMP
0007 01  66800013 00007           LDX  I2  TEMP
0009 0   C2FD     00008           LD    2  -3
000A 0   D2FB     00009           STO   2  -5
000B 0   C2FE     00010           LD    2  -2
000C 0   18C8     00011           RTE      8
000D 0   1008     00012           SLA      8
000E 0   D2FD     00013           STO   2  -3
000F 0   1090     00014           SLT      16
0010 0   D2FC     00015           STO   2  -4
0011 0   C2FF     00016           LD    2  -1
0012 0   18C8     00017           RTE      8
0013 0   1008     00018           SLA      8
0014 0   D2FF     00019           STO   2  -1
0015 0   1090     00020           SLT      16
0016 0   D2FE     00021           STO   2  -2
0017 0   7101     00022           MDX   1  1
0018 0   6901     00023           STX   1  RET+1
0019 00  4C000000 00024    RET   BSC  L   0
001B 0   0000     00025    TEMP  DC       *-*
001C 0   0001     00026    UM    DC       1
001E              00027           END      WZDPK

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

// DUP

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


*STORE      WS  UA  WZDPK           0010
CART ID 0010   DB ADDR  02EB   DB CNT   0003

// EJECT
```

ROTINAS DE TRATAMENTO GERAL DO MONTADOR

WZPRC / WZPRO
WZTRZ / WZTRA
WZCNV / WZDEC

ROTINA DE BUSCA NA TABELA DE SIMBOLOS

WZMN

NOME DO PROGRAMA	CÓDIGO DE ATIVIDADE NUM. DO DOCUMENTO	CÓDIGO DO PROGRAMA
Rotina de busca e inserção de um simbolo na tabela de símbolos		WZPRC
	PROGRAMADOR	/ / DATA

LINGUAGEM USADA

FORTRAN [X] ASSEMBLER [ ] BASIC [ ] ALGOL [ ] .... [ ]

TIPO DE PROGRAMA

DRIVER [ ] SUBROUTINE [X] PRINCIPAL FUNCTION [ ] OUTROS [ ]

IDENTIFICAÇÃO DOS DISCOS (FITAS)	SISTEMA SUPORTE (SIST. OPERACIONAL E HARDWARE)

CHAVES DO PAINEL OU SAIDAS ALTERNATIVAS

DESCRIÇÃO DO PROGRAMA

Esta rotina busca e insere um símbolo na tabela de símbolos, desde que ele não conste ainda desta tabela

O símbolo é colocado na tabela compactado, isto é 2 caracteres por palavra.

A seguir o procedimento é determinado pelo valor de um dos parametros transmitidos a subrotina. Se o valor desta chave é menor que 1, o símbolo, é simplesmente marcado como usado. Se o valor é 1, o símbolo deve ser marcado como definido, e endereçado. Se porém ele já esta definido então é marcado como sobredefinido. Se o valor da chave é maior que 1, simplesmente a posição do simbolo é determinada.

 (use o verso, se necessário)

CÓDIGO DO PROGRAMA

WZPRC

SEQUÊNCIA DE CHAMADA E DESCRIÇÃO DAS VARIÁVEIS - ARGUMENTO

CALL WZPRC (ISW, K, ISTEP, LL? L2)

ISW : chave cujo valor determina o procedimento

< 0 - marca como usado

= 0 - define e endereça o simbolo

> 0 - determina a posição do símbolo na tabe la

K : posição do símbolo na tabela

ISTEP : próximo comando a ser montado deve ocupar esta posição vire

SUB-ROTINAS USADAS

WZPK - rotina de compactação de dois caracteres em uma palavra

OBSERVAÇÕES OU MENSAGENS DE ERROS

2 (use o verso, se necessário)

CÓDIGO DE PROGRAMA

WZPRC

VARIÁVEIS PRINCIPAIS

TIPO
(Dimension, Common, Real, Integer, Data, etc)

Nome	Tipo	Função da Variável
MM(5)	I	matriz de trabalho e compactação
IAPSI	I,C	Ponteiro da ~~max~~ tabela de símbolos
IENSI	I,C	Matriz de endereços
ITGSI	I,C	Matriz de marcação dos símbolos como usados
N NOSIM	I,C	Tabela de símbolos

ANEXOS: DIAGRAMA DE BLOCOS [ ]
LISTAGEM DO PROGRAMA [ ]
LISTAGEM DO EXEMPLO [ ]
GABARITO DOS CARTÕES DE ENTRADA OU DA IMPRESSORA [ ]
FOLHA DE ALTERAÇÃO VERSÃO 1 [ ]
VERSÃO = [ ]

```
// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
      SUBROUTINE WZPRC (ISW,K,ISTEP,LL,L2 )
C
C     ESTA ROTINA BUSCA UM SIMBOLO NA TABELA DE SIMBOLOS
C     SE ISW-0 MARCA O SIMBOLO COMO USADO
C     SE ISW = 1 INSERE NA TABELA
C     SE ISW = 2 SOMENTE BUSCA NA TABELA
C     LL - POSICAO INICIAL DO SIMBOLO EM ICOL
C     L2 - POSICAO FINAL
C     K - PONTEIRO PARA O SIMBOLO NA TABELA DE SIMBOLOS
C     ISTEP - ENDERECO DO SIMBOLO A SER INSERIDO NA TABELA
C
      DIMENSION  MM (5)
      COMMON IPAG,KONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEQ,NOEQU(100),NOSIM(400,5),IENSI(400)
      COMMON IORIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ICOL(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
      EQUIVALENCE (IBCO,INIC(38))
C
      L1=LL
      DO  100  J = 1 , 5
  100 MM(J) = IBCO
      NN=L2-L1+1
C
C     COMPLETA MM COM O SIMBOLO
C
      DO 150 J=1,NN
      MM (J) = ICOL (L1)
      L1 = L1 + 1
  150 CONTINUE
      CALL WZPK(MM)
C
C         PRINCIPIA A BUSCAR O SIMBOLO
C
      IF(IAPSI)200,301,200
  200 DO  300  K = 1 ,IAPSI
      DO 250 J=1,3
      IF(MM(J)-NOSIM(K,J))300,250,300
  250 CONTINUE
      GO TO 400
  300 CONTINUE
C
C         SIMBOLO NAO  FAZ  PARTE  DA  TABELA
C
  301 IAPSI=IAPSI+1
C
      DO 350 J=1,3
  350 NOSIM (IAPSI , J) = MM(J)
      NOSIM(IAPSI,4)=0
      NOSIM(IAPSI,5)=0
      K = IAPSI
C
C             SIMBOLO CONSTA DA TABELA
C
```

```
  400 IF(ISW-1)450,500,700
C
C          SIMBOLO FOI  DEFINIDO  COMO USADO
C
  450 RETURN
C
C          SIMBOLO  E 'TAGED' COMO  DEFINIDO
C
  500 IENSI(K)=ISTEP
      MS=NOSIM(K,4)
      IF(MS)510,700,510
  510 GO TO(520,520,520,520,530,530,520,520,520,520),MS
C
C          SIMBOLO  ESTA  SOBREDEFINIDO
  520 NOSIM(K,4)=10
      RETURN
  530 NOSIM(K,4)=MS+3
  700 RETURN
      END

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZPRC
 COMMON   3130  VARIABLES     12  PROGRAM    282

RELATIVE ENTRY POINT ADDRESS IS 0011 (HEX)

END OF COMPILATION

// DUP

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


*STORE      WS  UA  WZPRC            0010
CART ID 0010   DB ADDR  04B3   DB CNT   0012

// EJECT
```

```
// *
                                                                    068
// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
      SUBROUTINE WZPRO (ISW,K,ISTEP,LL,L2 )
C
C     ESTA ROTINA BUSCA UM SIMBOLO NA TABELA DE SIMBOLOS
C     SE ISW-0 MARCA O SIMBOLO COMO USADO
C     SE ISW = 1 INSERE NA TABELA
C     SE ISW = 2 SOMENTE BUSCA NA TABELA
C     LL - POSICAO INICIAL DO SIMBOLO EM ICOL
C     L2 - POSICAO FINAL
C     K - PONTEIRO PARA O SIMBOLO NA TABELA DE SIMBOLOS
C     ISTEP - ENDERECO DO SIMBOLO A SER INSERIDO NA TABELA
C
      DIMENSION ICOL(80)
      DIMENSION  MM (5)
      COMMON IPAG,KONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEQ,NOEQU(100),NOSIM(400,5),IENSI(400)
      COMMON IORIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ISAI(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
      EQUIVALENCE (IBCD,INIC(38))
      EQUIVALENCE (ICOL(1),IERRA(1))
C
      L1=LL
      DO  100  J = 1 , 5
  100 MM(J) = IBCD
      NN=L2-L1+1
C
C     COMPLETA MM COM O SIMBOLO
C
      DO 150 J=1,NN
      MM (J) = ICOL (L1)
      L1 = L1 + 1
  150 CONTINUE
      CALL WZPK(MM)
C
C         PRINCIPIA A BUSCAR O SIMBOLO
C
      IF(IAPSI)200,301,200
  200 DO  300  K = 1 ,IAPSI
      DO 250 J=1,3
      IF(MM(J)-NOSIM(K,J))300,250,300
  250 CONTINUE
      GO TO 400
  300 CONTINUE
C
C         SIMBOLO NAO  FAZ  PARTE  DA  TABELA
C
  301 IAPSI=IAPSI+1
C
      DO 350 J=1,3
  350 NOSIM (IAPSI , J) = MM(J)
      NOSIM(IAPSI,4)=0
      K = IAPSI
```

```
C
C           SIMBOLO CONSTA DA TABELA
C
      MS=NOSIM(K,4)
      IF(MS)510,700,510
  510 GO TO(520,520,520,520,530,530,520,520,520,520),MS
C
C         SIMBOLO  ESTA  SOBREDEFINIDO
  520 NOSIM(K,4)=10
      RETURN
  530 NOSIM(K,4)=MS+3
  400 IF(ISW-1)450,500,700
C
C         SIMBOLO FOI  DEFINIDO  COMO USADO
C
  450 RETURN
C
C         SIMBOLO  E 'TAGED' COMO  DEFINIDO
C
  500 IENSI(K)=ISTEP
  700 RETURN
      END

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZPRO
 COMMON   3130  VARIABLES     12  PROGRAM    272

RELATIVE ENTRY POINT ADDRESS IS 0011 (HEX)

END OF COMPILATION

// DUP

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


*STORE      WS  UA  WZPRO           0010
CART ID 0010   DB ADDR  05EB   DB CNT   0011

// EJECT
```

069

NOME DO PROGRAMA	CÓDIGO DE ATIVIDADE NUM. DO DOCUMENTO	CÓDIGO DO PROGRAMA
ROTINA DE TRATAMENTO DE UM SIMBOLO		WZTRZ/WZTRA
	PROGRAMADOR	/ / DATA

LINGUAGEM USADA

FORTRAN [x] ASSEMBLER [ ] BASIC [ ] ALGOL [ ] .... [ ]

TIPO DE PROGRAMA

DRIVER [ ] SUBROUTINE [x] PRINCIPAL FUNCTION [ ] OUTROS [ ]

IDENTIFICAÇÃO DOS DISCOS (FITAS)	SISTEMA SUPORTE (SIST. OPERACIONAL E HARDWARE)
0010	

CHAVES DO PAINEL OU SAIDAS ALTERNATIVAS

DESCRIÇÃO DO PROGRAMA

Este programa verifica a validade de um rótulo ou operando, determinando suas posições iniciais e finais. O rótulo ou operando pode ser seguido de um branco, vírgula, sinais de mais e menos.

O programa chamante deve indicar quais os símbolos que podem seguir um determinado rótulo ou operando. Isto é feito através de uma chave.

 (use o verso, se necessário)

CÓDIGO DO PROGRAMA
WZTRZ/ WZTRA

SEQUÊNCIA DE CHAMADA E DESCRIÇÃO DAS VARIÁVEIS - ARGUMENTO

CALL WZTRZ(IN, IFIM, ISW, ISW1, M)
IN - caracter inicial do rotulo/operando
IFIM - caracter final
ISW - indica quais os delimitadores permitidos
= 1 - branco , = 2 - mais e menos
= 3 - mais, menos e (, = 4 - mais, menos, (e vírgula
ISW1 - indica qual o delimitador encontrado
M - se diferente de zero indica erro.

SUB-ROTINAS USADAS

WZERR - para tratar erros

OBSERVAÇÕES ENCAMINHAMENTO OU MENSAGENS DE ERROS

Podem ocorrer os seguintes erros:
+ ERRO 1 - simbolo se inicia com caracter inválido
- ERRO 2 - simbolo comtém caracter inválido.

 (use o verso, se necessário)

CÓDIGO DE PROGRAMA

WZTRZ / WZTRA

VARIÁVEIS PRINCIPAIS

TIPO
(Dimension, Common, Real, Integer, Data, etc)

Nome	Tipo	Função da Variável
MX1	I	caracter $
MX2	I	caracter #
MX3	I	caracter @
K	I	ponteiro para o buffer
IBCO	I	caracter branco
		DEMAIS VARIAVEIS SÃO DO COMMON E PARAMETROS

ANEXOS:

DIAGRAMA DE BLOCOS [ ]

LISTAGEM DO PROGRAMA [ ]

LISTAGEM DO EXEMPLO [ ]

GABARITO DOS CARTÕES DE ENTRADA OU DA IMPRESSORA [ ]

FOLHA DE ALTERAÇÃO VERSÃO 1 [ ]

VERSÃO = [ ]

073

```
// JOB    0001 0010                                                    *     E102,MALIC

LOG DRIVE   CART SPEC   CART AVAIL  PHY DRIVE
  0000        0001        0001        0000
  0001        0010        0010        0001
                          0024        0002

V2 M09   ACTUAL 16K  CONFIG 16K

* LABORATORIO DE SISTEMAS DIGITAIS * EPUSP *  15/ 04/ 74  * 17 H  26 MIN *

// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
      SUBROUTINE WZTRZ(IN,IFIM,ISW,ISW1,M)
C
C     *      ESTA SUBROTINA DETETA O FIM DE UM OPERANDO OU                 *
C     *   LABEL, E TESTA OS SIMBOLOS PERMITIDOS                            *
C     IN - PRIMEIRA POSICAO
C     IFIM - ULTIMA POSICAO DA VARIAVEL
C     ISW - INDICA DELIMITADORES PERMITIDOS
C                ISW=1  SO BRANCO PODE SEGUIR O SIMBOLO
C                ISW=2  PESQUISA-SE + E -
C                ISW=3  PESQUISA-SE + - E (
C                ISW=4  PESQUISA-SE + - ( E ,
C     M - INDICA ERRO SE IGUAL A 1 , SENAO DEVE SER ZERO.
C     ISW1 - INDICA QUAL FOI O DELIMITADOR ENCONTRADO
C
C              ERRO1 - SIMBOLO SE INICIA COM CARACTER NAO PERMITIDO
C              ERRO2 - VARIAVEL CONTEM CARACTER NAO PERMITIDO
C
      DIMENSION M1(3)
      COMMON IPAG,KONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEQ,NOEQU(100),NOSIM(400,5),IENSI(400)
      COMMON IORIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ICOL(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
      EQUIVALENCE (IBCD,INIC(38))
      DATA MX1,MX2,MX3/'$','=','('/
      DATA M1 / 40,41,42/
      ISW1=38
      M=0
C
C     PROCURA PRIMEIRA LETRA
C
      DO 10 J=2,27
      IF(ICOL(IN)-INIC(J))10,20,10
   10 CONTINUE
      IF(ICOL(IN)-MX1) 11,20,12
   11 IF(ICOL(IN)-MX2) 12,20,12
   12 IF(ICOL(IN)-MX3) 13,20,13
   13 M=1
      CALL WZERR(1)
C
C     BUSCA OUTROS CARACTERES
C
   20 DO 30 K=1,4
```

```
      KK=IN+K
      IF(KK-73)25,50,50
 25   IF(ICOL(KK)-IBCO    )40,50,40
   40 DO 60 J=2,37
      IF(ICOL(KK)-INIC(J))60,30,60
   60 CONTINUE
      GO TO (80,200,200,200),ISW
   80 CALL WZERR(2)
      M=1
      RETURN
C
C     BUSCA DELIMITADORES DIFERENTE DE BRANCO
C
   70 DO 90 J=39,MM
      IF(ICOL(KK)-INIC(J))90,100,90
   90 CONTINUE
      GO TO 80
  100 ISW1=J
      GO TO 50
   30 CONTINUE
      IFIM=IN+4
      IF(ISW-1) 200,31,200
   31 IF(IBCO-ICOL(IFIM+1)) 80,32,80
C
C     FOI ENCONTRADO CARACTER 'BRANCO'
C
   32 ISW1=38
      RETURN
   50 IFIM=IN+K-1
      RETURN
  200 MM= M1(ISW-1)
      GO TO 70
      END

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZTRZ
 COMMON   3130  VARIABLES     12  PROGRAM    296

RELATIVE ENTRY POINT ADDRESS IS 0015 (HEX)

END OF COMPILATION

// DUP

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


*STORE      WS  UA  WZTRZ           0010
CART ID 0010   DB ADDR  1770   DB CNT   0014

// EJECT
```

```
// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
      SUBROUTINE WZTRA(IN,IFIM,ISW,ISW1,M)
C
C     *       ESTA SUBROTINA DETETA O FIM DE UM OPERANDO OU                 *
C     *   LABEL, E TESTA OS SIMBOLOS PERMITIDOS                             *
C     IN - PRIMEIRA POSICAO
C     IFIM - ULTIMA POSICAO DA VARIAVEL
C     ISW - INDICA DELIMITADORES PERMITIDOS
C                 ISW=1  SO BRANCO PODE SEGUIR O SIMBOLO
C                 ISW=2  PESQUISA-SE + E -
C                 ISW=3  PESQUISA-SE + - E (
C                 ISW=4  PESQUISA-SE + - ( E ,
C     M - INDICA ERRO SE IGUAL A 1 , SENAO DEVE SER ZERO.
C     ISW1 - INDICA QUAL FOI O DELIMITADOR ENCONTRADO
C
C              ERRO1 - SIMBOLO SE INICIA COM CARACTER NAO PERMITIDO
C              ERRO2 - VARIAVEL CONTEM CARACTER NAO PERMITIDO
C
      DIMENSION ICOL(80)
      DIMENSION M1(3)
      COMMON IPAG,KONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEQ,NOEQU(100),NOSIM(400,5),IENSI(400)
      COMMON IORIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ISAI(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
      EQUIVALENCE (IBCO,INIC(38))
      EQUIVALENCE (ICOL(1),IERRA(1))
      DATA MX1,MX2,MX3/'$','=','('/
      DATA M1 / 40,41,42/
      ISW1=38
      M=0
C
C     PROCURA PRIMEIRA LETRA
C
      DO 10 J=2,27
      IF(ICOL(IN)-INIC(J))10,20,10
   10 CONTINUE
      IF(ICOL(IN)-MX1) 11,20,12
   11 IF(ICOL(IN)-MX2) 12,20,12
   12 IF(ICOL(IN)-MX3) 13,20,13
   13 M=1
      CALL WZERR(1)
C
C     BUSCA OUTROS CARACTERES
C
   20 DO 30 K=1,4
      KK=IN+K
      IF(KK-73)25,50,50
 25   IF(ICOL(KK)-IBCO    )40,50,40
   40 DO 60 J=2,37
      IF(ICOL(KK)-INIC(J))60,30,60
   60 CONTINUE
      GO TO (80,200,200,200),ISW
   80 CALL WZERR(2)
      M=1
```

RETURN

076

```
C
C     BUSCA DELIMITADORES DIFERENTE DE BRANCO
C
   70 DO 90 J=39,MM
      IF(ICOL(KK)-INIC(J))90,100,90
   90 CONTINUE
      GO TO 80
  100 ISW1=J
      GO TO 50
   30 CONTINUE
      IFIM=IN+4
      IF(ISW-1) 200,31,200
   31 IF(IBCO-ICOL(IFIM+1)) 80,32,80
C
C     FOI ENCONTRADO CARACTER 'BRANCO'
C
   32 ISW1=38
      RETURN
   50 IFIM=IN+K-1
      RETURN
  200 MM= M1(ISW-1)
      GO TO 70
      END

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZTRA
 COMMON   3130  VARIABLES     12  PROGRAM    296

RELATIVE ENTRY POINT ADDRESS IS 0015 (HEX)

END OF COMPILATION

// DUP

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


*STORE      WS  UA  WZTRA            0010
CART ID 0010   DB ADDR  05D7   DB CNT   0014

// EJECT
```

NOME DO PROGRAMA	CÓDIGO DE ATIVIDADE NUM. DO DOCUMENTO	CÓDIGO DO PROGRAMA
PROGRAMA DE CONVERSÃO NUMÉRICA		WZCNV/WZDEC
	PROGRAMADOR	/ / DATA

LINGUAGEM USADA

FORTRAN [X] ASSEMBLER [ ] BASIC [ ] ALGOL [ ] .... [ ]

TIPO DE PROGRAMA PRINCIPAL

DRIVER [ ] SUBROUTINE [X] FUNCTION [ ] OUTROS [ ]

IDENTIFICAÇÃO DOS DISCOS (FITAS)	SISTEMA SUPORTE (SIST. OPERACIONAL E HARDWARE)
0010	

CHAVES DO PAINEL OU SAIDAS ALTERNATIVAS

DESCRIÇÃO DO PROGRAMA

Este programa converte um número lido em EBCDIC em um valor binário. O caracter inicial e final do número são devolvidos ao programa que chama a subrotina. A conversão pode ser decimal ou hexadecimal. A conversão é pedida através do programa que chama a subrotina, o qual deve determinar se o número a converter é decimal ou hexa. O programa principal deve indicar ainda quais os caracteres que podem seguir o número. Estes caracteres são:

- branco
- vírgula
- qualquer outro, não usado pelo Cross-Assembler, mas que pode ser indicado por um parâmetro.

Ex:

```
SAI     1Ø
         K2
        K1      branco
```

 (use o verso, se necessário)

CÓDIGO DO PROGRAMA

WZCNV/ WZDEC

SEQUÊNCIA DE CHAMADA E DESCRIÇÃO DAS VARIÁVEIS - ARGUMENTO

CALL WZCNV(I, N,K1, K2, ISW, M, MM)

I - ultima posição antes do inicio do número.
N - número convertido.
K1, K2 - respectivamente valores iniciais e finais.
ISW - chave que indica conversão decimal ou hexa.
= Ø - hexadecimal
= 1 - decimal
MM - número de tabela do caracter que pode seguir o número.
M - se diferente de zero indica ocorrência de erro.

SUB-ROTINAS USADAS

WZERR - rotina tratadora de erros

OBSERVAÇÕES [illegible] OU MENSAGENS DE ERROS

Erros apresentados

13 - número apresenta caracter não válido.

 (use o verso, se necessário)

CÓDIGO DE PROGRAMA

WZCNV / WZDEC

VARIÁVEIS PRINCIPAIS

TIPO
(Dimension, Common, Real, Integer, Data, etc)

Nome	Tipo	Função da Variável
K	I	ponteiro do buffer de entrada
J1	I	indica início da tabela de caracteres permitidos.
J2	I	indica fim da tabela de caracteres permitidos.
IC	I	fator de multiplicação do número
		DEMAIS VARIAVEIS SÃO DO COMMON OU OS PARÂMETROS VISTOS ATRÁS

ANEXOS:

DIAGRAMA DE BLOCOS [ ]

LISTAGEM DO PROGRAMA [ ]

LISTAGEM DO EXEMPLO [ ]

GABARITO DOS CARTÕES DE ENTRADA OU DA IMPRESSORA [ ]

FOLHA DE ALTERAÇÃO VERSÃO 1 [ ]

VERSÃO = [ ]

M 14 SYSTEM PROGRAM DETECTED MONITOR CONTROL RECORD

```
// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
      SUBROUTINE WZDEC(I,N,K1,K2,ISW,M,MM)
      DIMENSION ICOL(80)
      COMMON IPAG,IKONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEQ,NOEQU(100),NOSIM(400,5),IENSI(400)
      COMMON IORIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ISAI(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
      EQUIVALENCE (IESTR,INIC(45)),(IBAR,INIC(47)),(IBCD,INIC(38)),
     1     (ICOL(1),IERRA(1))
      M=0
      K2=0
      K1=I+1
      K=K1
      N=0
      IF(ISW-1) 10,20,10
C
C     CONVERSAO DECIMAL
C
   20 J1=28
      J2=37
      IC=10
      GO TO 30
C
C     CONVERSAO HEXADECIMAL
C
   10 J1=22
      J2=37
      IC=16
   30 IF(ICOL(K)-IBCD) 50,40,50
   40 CALL WZERR(13)
      GO TO 70
   50 K1=K
C
C     CONVERSAO
C
   80 DO 90 J=J1,J2
      IF(ICOL(K)-INIC(J)) 90,100,90
   90 CONTINUE
      GO TO 110
  100 N=N*IC+NUM(J-21)
      K=K+1
      IF(K-73) 120,150,120
   70 M=1
      RETURN
  110 IF(ICOL(K)-INIC(MM)) 40,150,40
  120 IF(ICOL(K)-IBCD) 80,150,80
  150 K2=K-1
      RETURN
      END
```

FEATURES SUPPORTED
 ONE WORD INTEGERS

```
CORE REQUIREMENTS FOR WZDEC
 COMMON   3130  VARIABLES      8  PROGRAM    216

RELATIVE ENTRY POINT ADDRESS IS 0011 (HEX)

END OF COMPILATION

// DUP

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


*STORE      WS  UA  WZDEC            0010
CART ID 0010   DB ADDR  03D3   DB CNT   000E

// EJECT
```

```
// *
```



```
// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
      SUBROUTINE WZCNV(I,N,K1,K2,ISW,M,MM)
C
C     ESTA ROTINA FAZ CONVERSAO ALFANUMERICO PARA NUMERICO
C     I - POSICAO ANTERIOR AO NUMERO
C     K1 - PRIMEIRA POSICAO DO NUMERO
C     K2 - ULTIMA POSICAO DO NUMERO
C     ISW - CHAVE DECIMAL/HEXADECIMAL
C          SE = 1 - DECIMAL
C          SE NAO 1 - HEXADECIMAL
C     MM - CARACTER QUE PODE SEGUIR O NUMERO
C     M - INDICADOR DE ERRO
C         ERRO13 - ERRO NA CONVERSAO
C
      COMMON IPAG,KONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEQ,NOEQU(100),NOSIM(400,5),IENSI(400)
      COMMON IORIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ICOL(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
      EQUIVALENCE  (IESTR,INIC(45)),(IBAR,INIC(47)),(IBCD,INIC(38))
      M=0
      K2=0
      I=I+1
      K=I
      N=0
      IF(ISW-1)10,20,10
C
C     ESCOLHA DE CONVERSAO DECIMAL
C
   20 J1=28
      J2=37
      IC=10
      GO TO 30
C
C     ESCOLHA DE CONVERSAO HEXADECIMAL
C
   10 J1=22
      J2=37
      IC=16
   30 IF(ICOL(K)-IBCD    )50,40,50
   40 CALL WZERR(13)
      GO TO 70
   50 K1=K
C
C     CONVERSAO
C
   80 DO 90 J=J1,J2
      IF(ICOL(K)-INIC(J))90,100,90
   90 CONTINUE
      GO TO 110
  100 N=N*IC+NUM(J-21)
      K=K+1
      IF(K-73)120,150,120
```

```
   70 M=1
      RETURN
  110 IF(ICOL(K)-INIC(MM))140,150,140
  140 CALL WZERR(13)
      GO TO 70
  120 IF(ICOL(K)-IBCD    )80,150,80
  150 K2=K-1
      RETURN
      END
```

FEATURES SUPPORTED
ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZCNV
COMMON 3130 VARIABLES 8 PROGRAM 220

RELATIVE ENTRY POINT ADDRESS IS 0011 (HEX)

END OF COMPILATION

```
// DUP

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


*STORE      WS  UA  WZCNV           0010
CART ID 0010   DB ADDR  02D6   DB CNT   000E

// EJECT
```

NOME DO PROGRAMA	CÓDIGO DE ATIVIDADE NUM. DO DOCUMENTO	CÓDIGO DO PROGRAMA
Rotina de busca na tabela de mnemônicos		WZMN
	PROGRAMADOR	/ / DATA

LINGUAGEM USADA

FORTRAN [X] ASSEMBLER [ ] BASIC [ ] ALGOL [ ] .... [ ]

TIPO DE PROGRAMA

DRIVER [ ] SUBROUTINE [X] PRINCIPAL FUNCTION [ ] OUTROS [ ]

IDENTIFICAÇÃO DOS DISCOS (FITAS)	SISTEMA SUPORTE (SIST. OPERACIONAL E HARDWARE)

CHAVES DO PAINEL OU SAIDAS ALTERNATIVAS

DESCRIÇÃO DO PROGRAMA

Este programa busca um mnemônico na tabela de mnemônicos. A busca é feita através de duas tabelas : a tabela de mnemônicos propriamente ditos e a tabela de pseudos. Em cada uma destas tabelas a busca é feita primeiramente a partir do último caracter do mnemônico. Pesquisa-se se este caracter é igual ou diferente de branco. Se é diferente a pesquisa se dá na primeira parte da tabela; se é igual a branco a pesquisa se dá na segunda parte da tabela. A pesquisa se processa primeiramente na tabela de mnemônicos, a seguir na tabela de pseudos. Dependendo da tabela em que o símbolo é encontrado, um código é retornado ao programa principal.

As tabelas fazem parte integrante desta rotina.

No caso de ter sido encontrado um mnemônico, verifica-se a existência de um rótulo, e , se este existir, ele é inserido na tabela de símbolos .

CÓDIGO DO PROGRAMA

WZMN

SEQUÊNCIA DE CHAMADA E DESCRIÇÃO DAS VARIÁVEIS - ARGUMENTO

CALL WZMN (IK,L1,L2,ISTEP,IFP,IFPX)

IK : grupo de tratamento do mnemônico ou pseudo
L1 : início do rótulo (primeiro caracter)
L2 : último caracter do rótulo
ISTEP : posição de memória a ser ocupada pelo co - mando
IFPX : tabela de externals
IFP : ponteiro de IFPX

SUB-ROTINAS USADAS

WZCMP : compacta as 4 posições do rotulo em uma variavel real
WZPRC : busca na tabela de símbolos o rótulo do comando
WZERR: tratamento do erro

OBSERVAÇÕES OU MENSAGENS DE ERROS

18- mnemônico, pseudo ou external inválido

2 (use o verso, se necessário)

CÓDIGO DE PROGRAMA

WZMN

VARIÁVEIS PRINCIPAIS

TIPO
(Dimension, Common, Real, Integer, Data, etc)

Nome	Tipo	Função da Variável
ICOL(80)	C, I	contem os 80 caracteres de um cartão
IBCO	I,C	caracter branco
LINIC	I	simbolo inicial na tabela de mnemonicos
LINIP	I	simbolo inicial na tabela de pseudos
LIFIM	I	simbolo final na tabela de mnemonicos
LIFIP	I	simbolo final na tabela de pseudos
XMNE (50)	R,D	tabela de mnemonicos
XPSEU(24)	R,D	tabela de pseudos
ISTEC(90)	I,C	lista de informação suplementar
IAPTC	I,C	POnteiro para a lista de informação suplementar indicando a primeira posição vaga.
NOSIM	I,C	tabela de simbolos.

ANEXOS: DIAGRAMA DE BLOCOS [ ]
LISTAGEM DO PROGRAMA [ ]
LISTAGEM DO EXEMPLO [ ]
GABARITO DOS CARTÕES DE ENTRADA OU DA IMPRESSORA [ ]
FOLHA DE ALTERAÇÃO VERSÃO 1 [ ]
VERSÃO = [ ]

```
// JOB    0001 0010                                          *     E102,MALIC

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  * 19 H  24 MIN *

// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
      SUBROUTINE WZMN(IK,L1,L2,ISTEP,IFP,IFPX)
C
C     ESTA SUBROUTINE BUSCA O MNEMONICO NA TABELA DE MNEMONICOS OU DE
C     PSEUDOS SE SE TRATAR DE CODIGO DE OPERACAO INCLUI TAMBEM O ROTU-
      LO NA TABELA DE SIMBOLOS
C         ERRO18 - MNEMONICO INVALIDO
C     L1 - PRIMEIRA POSICAO DO ROTULO
C     L2 - ULTIMA POSICAO DO ROTULO
C     ISTEP - CONTADOR DE INSTRUCOES
C
      REAL IFPX(50)
      DIMENSION XMA(24),XNA(50),XMNE(50),XPSEU(24),ICODE(50)
      COMMON IPAG,KONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEQ,NOEQU(100),NOSIM(400,5),IENSI(400)
      COMMON IORIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ICOL(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
      EQUIVALENCE (IBCD,INIC(38)),(IIS,INIC(4))
      EQUIVALENCE (XMNE(1),XNA(1)),(XPSEU(1),XMA(1))
      DATA XMA/'DEFC','BLOC','CALL','DEFE','LINE','PAGE','LIST','NLST',
     1  'CNTR','NCTR','COM ','EQU ','END ','ORG ','ENT ','ABS ','ISS ',
     2  'ASC ','CAB ','EBC ','DN  ','EXT',2*' '/
      DATA XNA/'CARI','CMP1','SOMI','NAND','PLAZ','PLAN','LIMP','CMP2',
     1  'PLAX','PARE','ARMX','CARX','SOMX','ENTR','PERM','SLTM','SLVM',
     2  'UNEG','INIB','PLA ','PUG ','ARM ','TRE ','CAR ','SLV ','TRI ',
     3  'XOR ','DD  ','INC ','UM  ','SOM ','PNL ','SAI ','SAL ','SUS ',
     4  'SLT ','DDS ','DDV ','DE  ','DEV ','GD  ','GE  ','FNC ','ESP ',
     5  'LIM ','NOP ','PUL ','GDV ','GEV ','    '/
      DATA ICODE/ 3,4,2*3,2*1,10,4,1,4,3*1,2,4,2*6,2*4,3*1,4,1,6,4,3,8,
     1  2*4,1,7,2,2,1,6,6*8,2,2*4,9,4,2*8,1/
      CALL WZCMP(F,ICOL(27),ICOL(28),ICOL(29),ICOL(30))
C
C     BUSCA ALFABETICA
C
      IF(ICOL(30)-IBCD)521,522,521
  521 LINIC=1
      LINIP=1
      LIFIM=19
      LIFIP=10
      GO TO 533
  522 LINIC=20
      LINIP=11
      LIFIM=49
      LIFIP=22
```

```
C
C     BUSCA NA TABELA DE MNEMONICOS
C
  533 DO 534 IK=LINIC,LIFIM
      IF(F-XMNE(IK))534,540,534
  534 CONTINUE
      GO TO 630
  540 CONTINUE
C
C     MNEMONICO ENCONTRADO
C
      ISTEC(IAPTC+3)=35
      ISTEC(IAPTC+1)=1
      ISTEC(IAPTC+2)=IK
      IK =ICODE(IK)
C
C     TRATAMENTO DO LABEL
C
  559 IF ( L1 )  560 ,  570  , 560
  560 CALL WZPRC(1,LK,ISTEP,L1,L2)
      MS=NOSIM(LK,4)
      IF(MS) 755,561,755
  561 NOSIM(LK,4)=1
  570 ISTEC(IAPTC+4)=0
      RETURN
C
C      NAO E MNEMONICO - BUSCA NA TABELA DE PSEUDOS
C
  630 CONTINUE
      DO 638 IK=LINIP,LIFIP
      IF(XPSEJ(IK)-F)638,700,638
  638 CONTINUE
      GO TO 800
  700 CONTINUE
C
C     PSEUDO ENCONTRADO
C
      ISTEC(IAPTC+3)=35
      ISTEC(IAPTC+1)=2
      ISTEC(IAPTC+2)=IK
      IF(IK-3)755,559,755
  755 RETURN
  800 CONTINUE
      DO 810 IK=1,IFP
      IF(F-IFPX(IK)) 810,820,810
  810 CONTINUE
C
C     TEM - SE ERRO &  MNEMONICO ILEGAL  -
C
      CALL WZERR(18)
      ISTEC(IAPTC+1)=6
      RETURN
  820 ISTEC(IAPTC+1)=3
      RETURN
      END
```

FEATURES SUPPORTED

089

```
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZMN
 COMMON   3130  VARIABLES    210  PROGRAM    342

RELATIVE ENTRY POINT ADDRESS IS 00DF (HEX)

END OF COMPILATION

// DUP

*DELETE              WZMN      0010
CART ID 0010   DB ADDR  03B6   DB CNT   0030

*STORE      WS  UA  WZMN             0010
CART ID 0010   DB ADDR  1973   DB CNT   0030

// EJECT
```

SUBROTINAS DE IMPRESSÂO

Estas subrotinas destinam-se a impressão das listagens de saída do programa, bem como ao tratamento de erros.

WZPAG

WZPRN / WZIER / WZERR

WZWER

WZLIS

WZSSI

WZSAI

WZTES

WZRSL

NOME DO PROGRAMA ROTINA DE PAGINAÇÃO	CÓDIGO DE ATIVIDADE NUM. DO DOCUMENTO	CÓDIGO DO PROGRAMA WZPAG
	PROGRAMADOR	/ / DATA

LINGUAGEM USADA

FORTRAN [x] ASSEMBLER [ ] BASIC [ ] ALGOL [ ] .... [ ]

TIPO DE PROGRAMA

DRIVER [ ] SUBROUTINE [x] PRINCIPAL FUNCTION [ ] OUTROS [ ]

IDENTIFICAÇÃO DOS DISCOS (FITAS)	SISTEMA SUPORTE (SIST. OPERACIONAL E HARDWARE)
0010	

CHAVES DO PAINEL OU SAIDAS ALTERNATIVAS

DESCRIÇÃO DO PROGRAMA

Esta rotina inicializa uma página, escrevendo nela um cabeçalho, que pode ser brancos, ou um cabeçalho do programador transmitido através de common.

 (use o verso, se necessário)

CÓDIGO DO PROGRAMA

WZPAG

SEQUÊNCIA DE CHAMADA E DESCRIÇÃO DAS VARIÁVEIS - ARGUMENTO

CALL WZPAG

SUB-ROTINAS USADAS

NENHUMA

OBSERVAÇÕES [illegible] OU MENSAGENS DE ERROS

NENHUMA

 (use o verso, se necessário)

CÓDIGO DE PROGRAMA

WZPAG

VARIÁVEIS PRINCIPAIS

Nome	Tipo (Dimension, Common, Real, Integer, Data, etc)	Função da Variável
IPAG	I,C	Número da página
ICAB	I,C	Cabeçalho a ser escrito

ANEXOS: DIAGRAMA DE BLOCOS [ ]
LISTAGEM DO PROGRAMA [ ]
LISTAGEM DO EXEMPLO [ ]
GABARITO DOS CARTÕES DE ENTRADA OU DA IMPRESSORA [ ]
FOLHA DE ALTERAÇÃO VERSÃO 1 [ ]
VERSÃO = [ ]

```
// *
```



```
// FOR
*LIST ALL
*ONE WORD INTEGERS
      SUBROUTINE WZPGG
C     ESTA SUBROTINE INICIALIZA A PAGINA DE IMPRESSAO
      COMMON IPAG,KONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEQ,NOEQU(100),NOSIM(400,5),IENSI(400)
      COMMON IORIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ICOL(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
      KONT=0
      IPAG=IPAG+1
      WRITE(IPRIN,100) IPAG,ICAB
  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
     1 I G I T A I S   -   L.S.D.   -   PAG.',I4,//,41X,37A1,//)
      RETURN
      END
VARIABLE ALLOCATIONS
  IPAG(IC)=7FFF        KONT(IC)=7FFE       IPRIN(IC)=7FFD        ICAB(IC)
   ID7(IC)=7FD5          N5(IC)=7FD4          N6(IC)=7FD3          N7(IC)
 NOEQU(IC)=7FCF-7F6C NOSIM(IC)=7F6B-779C IENSI(IC)=779B-760C IORIG(IC)
   LST(IC)=7608       ISWER(IC)=7607         NUM(IC)=7606-75F7  INIC(IC)
 IAPTC(IC)=741E          IS(IC)=741D        IERR(IC)=741C       IAPER(IC)
 IAPIN(IC)=73C9        IRET(IC)=73C8         IAC(IC)=73C7       ICOMP(IC)

STATEMENT ALLOCATIONS
 100  =0002

FEATURES SUPPORTED
 ONE WORD INTEGERS

CALLED SUBPROGRAMS
 SWRT    SCOMP   SIOAI   SIOI

INTEGER CONSTANTS
     0=0000      1=0001

CORE REQUIREMENTS FOR WZPGG
 COMMON   3130  VARIABLES      0  PROGRAM     82

RELATIVE ENTRY POINT ADDRESS IS 003C (HEX)

END OF COMPILATION

// DUP

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


*STORE      WS  UA  WZPGG            0010
CART ID 0010   DB ADDR  243A   DB CNT   0006

// EJECT
```

095

```
// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
      SUBROUTINE WZPAG
C
C     ESTA SUBROTINE INICIALIZA A PAGINA DE IMPRESSAO
C
      COMMON IPAG,KONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEQ,NOEQU(100),NOSIM(400,5),IENSI(400)
      COMMON IORIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ICOL(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
      KONT=0
      IPAG=IPAG+1
      WRITE(IPRIN,100) IPAG,ICAB
  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
     1 I G I T A I S   -   L.S.D.   -   PAG.',I4,//,41X,37A1,//)
      WRITE(IPRIN,2543)
 2543 FORMAT(1H ,3X,'CC',4X,'CI',2X,'INSTRUCAO',33X,'COMANDO')
      RETURN
      END

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZPAG
 COMMON   3130  VARIABLES      0  PROGRAM    108

RELATIVE ENTRY POINT ADDRESS IS 0052 (HEX)

END OF COMPILATION

// DUP

*DELETE              WZPAG     0010
CART ID 0010   DB ADDR  154C   DB CNT   0006

*STORE      WS  UA  WZPAG           0010
CART ID 0010   DB ADDR  2432   DB CNT   0008

// EJECT
```

NOME DO PROGRAMA	CÓDIGO DE ATIVIDADE NUM. DO DOCUMENTO	CÓDIGO DO PROGRAMA
SUBROTINAS DE TRATAMENTO DE ERROS		WZPRN/WZIER/ WZERR
	PROGRAMADOR	/ / DATA

LINGUAGEM USADA

FORTRAN [x] ASSEMBLER [ ] BASIC [ ] ALGOL [ ] .... [ ]

TIPO DE PROGRAMA

DRIVER [ ] SUBROUTINE [x] PRINCIPAL FUNCTION [ ] OUTROS [ ]

IDENTIFICAÇÃO DOS DISCOS (FITAS)	SISTEMA SUPORTE (SIST. OPERACIONAL E HARDWARE)
0010	

CHAVES DO PAINEL OU SAIDAS ALTERNATIVAS

DESCRIÇÃO DO PROGRAMA

A primeira subrotina imprime uma mensagem de erros do tipo: ERRO NO. -----

A segunda subrotina trata um erro, quando este é detetado no segundo passo. O tratamento neste caso consiste de se colocar o erro (seu número) em uma pilha, para posterior impressão.

A terceira subrotina, trata erros tanto na primeira , como na segunda fase do programa. Na primeira fase, o tratamento do erros consiste em se guardar o erro em uma lista de erros, para posterior impressão ao fim desta fase. Na segunda fase , o tratamento consiste em se chamar WZIER.

 (use o verso, se necessário)

CÓDIGO DO PROGRAMA

WZPRN/WZIER/WZERR

SEQUÊNCIA DE CHAMADA E DESCRIÇÃO DAS VARIÁVEIS - ARGUMENTO

CALL WZPRN(K) onde K é o número do erro a ser impresso

CALL WZIER(K)

CALL WZERR(K)

SUB-ROTINAS USADAS

NENHUMA

OBSERVAÇÕES ERRO CANAL OU MENSAGENS DE ERROS

NENHUMA

 (use o verso, se necessário)

CÓDIGO DE PROGRAMA

WZPRN/WZIER/WZERR

VARIÁVEIS PRINCIPAIS

Nome	Tipo (Dimension, Common, Real, Integer, Data, etc)	Função da Variável
		WZIER
IAPER	I,C	contador de erros para o comando em análise
NOBQU	I,C,()	pilha onde são armazenados os erros desta análise
		WZERR
ISWER	I,C	indica em qual passo se encontra: se Ø indica segundo passo, se 1, indica primeiro.
IERR	I,C	contador de erros do primeiro passo.

ANEXOS: DIAGRAMA DE BLOCOS [ ]

LISTAGEM DO PROGRAMA [ ]

LISTAGEM DO EXEMPLO [ ]

GABARITO DOS CARTÕES DE ENTRADA OU DA IMPRESSORA [ ]

FOLHA DE ALTERAÇÃO VERSÃO 1 [ ]

VERSÃO = [ ]

```
// JOB    0001 0010                                          *     E102,

LOG DRIVE   CART SPEC   CART AVAIL  PHY DRIVE
  0000        0001        0001        0000
  0001        0010        0010        0001
                          0015        0002

V2 M09   ACTUAL 16K  CONFIG 16K

* LABORATORIO DE SISTEMAS DIGITAIS * EPUSP *  05/ 06/ 74  * 17 H  07

// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
      SUBROUTINE WZERR(N)
      COMMON IPAG,KONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEQ,NOEQU(100),NOSIM(400,5),IENSI(400)
      COMMON IORIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ICOL(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
      EQUIVALENCE (ID7,KERRR)
      IF(ISWER)200,10,200
   10 CALL WZWPT(N)
      IERR=IERR+1
      RETURN
  200 CALL WZIER(N)
      KERRR=KERRR+1
      RETURN
      END

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZERR
 COMMON   3130  VARIABLES      0  PROGRAM     34

RELATIVE ENTRY POINT ADDRESS IS 0001 (HEX)

END OF COMPILATION

// DUP

*DELETE              WZERR     0010
CART ID 0010   DB ADDR  15A4   DB CNT   0005

*STORE      WS  UA  WZERR           0010
CART ID 0010   DB ADDR  1B42   DB CNT   0003

// EJECT
```

```
// JOB    0001 0010                                          100      E1

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 *  05/ 11/ 74  * 17 H

// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
      SUBROUTINE WZIER(N)
      COMMON IPAG,KONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEQ,NOEQU(100),NOSIM(400,5),IENSI(400)
      COMMON ICRIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ICOL(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
      IAPER=IAPER+1
      NOEQU(IAPER)=N
      KERRR=KERRR+1
      RETURN
      END

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZIER
 COMMON   3130  VARIABLES      2  PROGRAM     30

RELATIVE ENTRY POINT ADDRESS IS 0003 (HEX)

END OF COMPILATION

// DUP

*DELETE              WZIER     0010
CART ID 0010   DB ADDR  1133   DB CNT   0003

*STORE      WS  UA  WZIER           0010
CART ID 0010   DB ADDR  242F   DB CNT   0003

// EJECT
```

```
// *

// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
      SUBROUTINE WZPRN(K)
      DATA IPRIN /5/
   20 WRITE(IPRIN,25) K
      RETURN
   25 FORMAT(41X,'ERRO NO.',I3)
      END

UNREFERENCED STATEMENTS
 20

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZPRN
 COMMON      0  VARIABLES      2  PROGRAM      22

RELATIVE ENTRY POINT ADDRESS IS 000A (HEX)

END OF COMPILATION

// DUP

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


*STORE      WS  UA  WZPRN           0010
CART ID 0010   DB ADDR  02EE   DB CNT   0003

// EJECT
```

NOME DO PROGRAMA	CÓDIGO DE ATIVIDADE NUM. DO DOCUMENTO	CÓDIGO DO PROGRAMA
SUBROTINA DE CONTROLE DE IMPRESSÃO DOS ERROS NA PRIMEIRA FASE		WZWER
	PROGRAMADOR	/ / DATA

LINGUAGEM USADA

FORTRAN [x] ASSEMBLER [ ] BASIC [ ] ALGOL [ ] .... [ ]

TIPO DE PROGRAMA

PRINCIPAL

DRIVER [ ] SUBROUTINE [x] FUNCTION [ ] OUTROS [ ]

IDENTIFICAÇÃO DOS DISCOS (FITAS)	SISTEMA SUPORTE (SIST. OPERACIONAL E HARDWARE)
0010	

CHAVES DO PAINEL OU SAIDAS ALTERNATIVAS

DESCRIÇÃO DO PROGRAMA

Este programa controla a impressão de erros na primeira fase do programa. O programa lê o arquivo de erros em buffer de 80 posições. A primeira posição do buffer indica quantas posições deste se encontram preenchidas. As posições seguintes contém números dos erros que ocorreram, ou então contém uma indicação de que não ocorreu erro. O programa verifica para cada comando a ocorrência de erro: se este ocorreu então uma mensagem é impressa. Se não ocorre erro, a rotina retorna ao ponto de chamada. O programa inicializa uma página sempre que necessário. O programa lê um novo buffer toda vez que o contador de palavras do buffer supera o valor na primeira posição.

 (use o verso, se necessário)

CÓDIGO DO PROGRAMA

WZwer

SEQUÊNCIA DE CHAMADA E DESCRIÇÃO DAS VARIÁVEIS - ARGUMENTO

CALL WZWER

SUB-ROTINAS USADAS

WZPRN - imprime o erro
WZPAG - inicializa página

OBSERVAÇÕES [illegible] OU MENSAGENS DE ERROS

NENHUMA

 (use o verso, se necessário)

CÓDIGO DE PROGRAMA

WZWER

VARIÁVEIS PRINCIPAIS

TIPO
(Dimension, Common, Real, Integer, Data, etc)

Nome	Tipo	Função da Variável
IAPER	I,C	contador do buffer de erros
IPAUT	I,C	indica que não ocorreu erro
IERRA	I,C,()	buffer dos erros
ID7	I,C	número do arquivo lógico de erros
KONT	I,C	contador de linhas impressas.

ANEXOS: DIAGRAMA DE BLOCOS [ ]

LISTAGEM DO PROGRAMA [ ]

LISTAGEM DO EXEMPLO [ ]

GABARITO DOS CARTÕES DE ENTRADA OU DA IMPRESSORA [ ]

FOLHA DE ALTERAÇÃO VERSÃO 1 [ ]

VERSÃO = [ ]

```
// *

// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
      SUBROUTINE WZWER
      COMMON IPAG,KONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEQ,NOEQU(100),NOSIM(400,5),IENSI(400)
      COMMON IORIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ICOL(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
      EQUIVALENCE (IPAUT,INIC(44))
  701 IF(IAPER-IERRA(1))710,710,850
  710 IF(IERRA(IAPER)-IPAUT   )900,1500,900
  850 READ(ID7'N7) IERRA
      IAPER=2
      GO TO 701
  900 CALL WZPRN(IERRA(IAPER))
  801 KONT=KONT+1
      IF(KONT-53) 811,811,810
  810 CALL WZPAG
  811 IAPER=IAPER+1
      GO TO 701
 1500 IAPER=IAPER+1
      RETURN
      END

UNREFERENCED STATEMENTS
 801

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZWER
 COMMON   3130  VARIABLES      2  PROGRAM     80

RELATIVE ENTRY POINT ADDRESS IS 0005 (HEX)

END OF COMPILATION

// DUP

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


*STORE      WS  UA  WZWER            0010
CART ID 0010   DB ADDR  1816   DB CNT   0006

// EJECT
```

NOME DO PROGRAMA ROTINA DE LISTAGEM DA FASE DE LOCAÇÃO	CÓDIGO DE ATIVIDADE NUM. DO DOCUMENTO	CÓDIGO DO PROGRAMA WZLIS
	PROGRAMADOR	/ / DATA

LINGUAGEM USADA

FORTRAN [X] ASSEMBLER [ ] BASIC [ ] ALGOL [ ] .... [ ]

TIPO DE PROGRAMA

PRINCIPAL

DRIVER [ ] SUBROUTINE [X] FUNCTION [ ] OUTROS [ ]

IDENTIFICAÇÃO DOS DISCOS (FITAS)	SISTEMA SUPORTE (SIST. OPERACIONAL E HARDWARE)
0010	

CHAVES DO PAINEL OU SAIDAS ALTERNATIVAS

DESCRIÇÃO DO PROGRAMA

Esta rotina lista o programa fonte, na ocorrência de erros após a primeira fase. A listagem é composta de: número de ordem de entrada do cartão, bem como do código fonte.

A fim de produzir esta listagem, programa fonte, arquivo de erros e arquivo de Informação Suplementar são lidos de disco. Inicia-se a listagem, imprimindo-se um cabeçalho de página, e iniciando-se o contador de linhas. Na ocorrência de um erro, este é impresso logo em seguida ao comando onde ocorreu.

O programa faz o controle de blocagem dos dados no arquivo de informação suplementar, pois o término do programa é determinado a partir deste.

 (use o verso, se necessário)

CÓDIGO DO PROGRAMA
WZlis

SEQUÊNCIA DE CHAMADA E DESCRIÇÃO DAS VARIÁVEIS - ARGUMENTO

CALL WZLIS (INREL, ISTEP)
INREL - controla a blocagem
ISTEP - contador de instruções

SUB-ROTINAS USADAS

WZPAG - controle de paginação
WZWER - controle de impressão de erros

OBSERVAÇÕES [illegible] OU MENSAGENS DE ERROS

NENHUMA

 (use o verso, se necessário)

CÓDIGO DE PROGRAMA

WZLIS

VARIÁVEIS PRINCIPAIS

TIPO
(Dimension, Common, Real, Integer, Data, etc)

Nome	Tipo	Função da Variável
IPAG	I,C	contador de páginas
KONT	I,C	contador de linhas
N5,N6,N7	I,C	varaiáveis de leitura dos arquivos de disco
IAPTC	I,C	ponteiro para o buffer do arquivo de inf. suplem.
IAPER	I,C	ponteiro para o buffer de erros
IERRA	I,C,()	buffer de erros
ISTRP	I,C,()	buffer de informação suplementar
ICOL	I,C,()	buffer de leitura de um comando
IVEL	I,	variável auxiliar para controle de blocagem do buffer de inf. suplem.
ID6,ID7	I,C	variáveis que determinam os arquivos lógicos de inf. suplem. e de erros, respectivamente.

ANEXOS: DIAGRAMA DE BLOCOS [ ]

LISTAGEM DO PROGRAMA [ ]

LISTAGEM DO EXEMPLO [ ]

CABARITO DOS CARTÕES DE ENTRADA OU DA IMPRESSORA [ ]

FOLHA DE ALTERAÇÃO VERSÃO 1 [ ]

VERSÃO = [ ]

```
// 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 *  23/ 05/ 74  * 20 H  01 MIN *

// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
      SUBROUTINE WZLIS(INREL,ISTEP)
C
      ******************************************************************
      *       ESTA SUBROTINA LISTA O PROGRAMA NO CASO DE OCORRENCIA    *
      *   DE ERROS NA PRIMEIRA PASSADA DO ASSEMBLER                    *
      ******************************************************************
C
C
      DIMENSION ISTRP(80)
      COMMON IPAG,KONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEQ,NOEQU(100),NOSIM(400,5),IENSI(400)
      COMMON IORIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ICOL(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
      EQUIVALENCE (ISTRP(1),ISTEC(1)),(INIC(44),  IPAUT)
C
      IPAG=0
      KONT=0
      CALL WZPAG
      N5=1
      N6=1
      N7=1
      IAPTC=1
      IAPER=2
      ISTEP=1
      READ(ID7'N7) IERRA
      READ(ID6'N6) ISTRP
C
  200 READ(ID5'N5)(ICOL(I),I=1,80)
      WRITE(IPRIN,1100) ISTEP,(ICOL(I),I=1,80)
      KONT=KONT+1
      IF(KONT-53) 501,501,400
  400 CALL WZPAG
  501 CALL WZWER
      IF(IAPTC-80) 641,950,641
  641 IF(ISTEC(IAPTC+1)-4) 650,700,650
  650 ISTEP=ISTEP+1
      IF(IAPTC) 670,660,670
  660 IAPTC=IVEL
      GO TO 200
  670 IF(IAPTC-ISTEC(IAPTC)) 671,671,680
  671 IAPTC=ISTEC(IAPTC)
      GO TO 200
  680 IAPTC=ISTEC(IAPTC)
```

```
      READ(ID6'N6) ISTRP
      GO TO 200
C
C     FINALIZACAO
C
  700 WRITE(IPRIN,1202)
 1202 FORMAT(1H ,//)
      CALL WZWER
      RETURN
  950 IVEL=ISTEC(IAPTC)
      READ(ID6'N6)ISTRP
      IAPTC=0
      GO TO 641
 1100 FORMAT(1X,I5,5X,80A1)
      END

FEATURES SUPPORTED
ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZLIS
COMMON   3130  VARIABLES      4  PROGRAM    236

RELATIVE ENTRY POINT ADDRESS IS 0015 (HEX)

END OF COMPILATION

// DUP

*DELETE              WZLIS     0010
CART ID 0010   DB ADDR  02EA   DB CNT   0012

*STORE      WS  UA  WZLIS           0010
CART ID 0010   DB ADDR  19C3   DB CNT   0010

// EJECT
```

NOME DO PROGRAMA	CÓDIGO DE ATIVIDADE NUM. DO DOCUMENTO	CÓDIGO DO PROGRAMA
ROTINAS PARA IMPRESSÃO DAS INSTRUÇÕES LONGAS		WZSSI
	PROGRAMADOR	/ / DATA

LINGUAGEM USADA

FORTRAN [X] ASSEMBLER [ ] BASIC [ ] ALGOL [ ] .... [ ]

TIPO DE PROGRAMA

DRIVER [ ] SUBROUTINE [X] PRINCIPAL FUNCTION [ ] OUTROS [ ]

IDENTIFICAÇÃO DOS DISCOS (FITAS)	SISTEMA SUPORTE (SIST. OPERACIONAL E HARDWARE)
0010	

CHAVES DO PAINEL OU SAIDAS ALTERNATIVAS

DESCRIÇÃO DO PROGRAMA

Este programa faz a impressão das instruções longas. A instrução é listada na forma : número de ordem do cartão, o contador de instruções (em hexa), a instrução codificada (em hexa), e o comando fonte. As instruções são também colocadas no buffer de saída em disco. Esta rotina não trata das instruções de endereçamento a memória, que requerem tratamento especial. A colocação no buffer de saída em disco é feita colocando- se as palavras encostadas a esquerda, um byte por palavra, isto é a palavra é decomposta em dois bytes. O contador de instruções é incrementado. O indicador de relocação é zero.

 (use o verso, se necessário)

CÓDIGO DE PROGRAMA

WZ991

VARIÁVEIS PRINCIPAIS

TIPO
(Dimension, Common, Real, Integer, Data, etc)

Nome	Tipo	Função da Variável
LST	I,C	variável de controle de listagem
ITEMP	I,	variável auxiliar

ANEXOS: DIAGRAMA DE BLOCOS [ ]
LISTAGEM DO PROGRAMA [ ]
LISTAGEM DO EXEMPLO [ ]
GABARITO DOS CARTÕES DE ENTRADA OU DA IMPRESSORA [ ]
FOLHA DE ALTERAÇÃO VERSÃO 1 [ ]
VERSÃO = [ ]

CÓDIGO DO PROGRAMA
WZSSI

SEQUÊNCIA DE CHAMADA E DESCRIÇÃO DAS VARIÁVEIS - ARGUMENTO

CALL WZSSI (IOCT,ISTEP,JJ,n2,IK)
IOCT - palavra montada
ISTEP - contador de instruções
JJ - ordem de entrada do comando
N2 - não tem significado, mas é sempre zero.
KI - indicador detratamento por WZTRT - deve ser 1

SUB-ROTINAS USADAS

WZITR - descompactação binário em hexadecimal
WZAND - exclusão do byte a direita de uma palavra
WZTRT - trata instrução no buffer de disco

OBSERVAÇÕES [illegible] OU MENSAGENS DE ERROS

NENHUMA

 (use o verso, se necessário)

```
// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
      SUBROUTINE WZSSI(IOCT,ISTEP,JJ,N2,IK)
      DIMENSION ICOL(80)
      COMMON IPAG,KONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEQ,NDEQU(100),NDSIM(400,5),IENSI(400)
      COMMON IORIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ISAI(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
      EQUIVALENCE (ICOL(1),IERRA(1))
      IF(LST) 7030,7060,7030
 7030 CALL WZITR(IOCT,IS1,IS2,IS3,IS4)
      CALL WZITR(ISTEP,K1,K2,K3,K4)
      WRITE(IPRIN,7500) JJ,INIC(K1),INIC(K2),INIC(K3),INIC(K4),INIC(IS1)
     1     , INIC(IS2),INIC(IS3),INIC(IS4),ICOL
 7500 FORMAT (1H ,I5,2X,4A1,2X,4A1,6X,80A1)
 7060 ITEMP=IOCT
      CALL WZAND(IOCT)
      KI=1
      CALL WZTRT(IOCT,N2,ISTEP,KI)
      ITEMP=ITEMP-IOCT
      IOCT=ITEMP*256
      CALL WZAND(IOCT)
      CALL WZTRT(IOCT,N2,ISTEP,KI)
      ISTEP=ISTEP+2
      RETURN
      END

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZSSI
 COMMON   3130  VARIABLES     18  PROGRAM    170

RELATIVE ENTRY POINT ADDRESS IS 0022 (HEX)

END OF COMPILATION

// DUP

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


*STORE      WS  UA  WZSSI            0010
CART ID 0010   DB ADDR  03B2   DB CNT   000C

// EJECT
```

NOME DO PROGRAMA	CÓDIGO DE ATIVIDADE NUM. DO DOCUMENTO	CÓDIGO DO PROGRAMA
ROTINA DE SAIDA DAS INSTRUÇÕES CURTAS? DAS LONGAS DE ENDEREÇAMENTO DA MEM; e DE PSEUDOS		WZSAI
	PROGRAMADOR	/ / DATA

LINGUAGEM USADA

FORTRAN [x] ASSEMBLER [ ] BASIC [ ] ALGOL [ ] .... [ ]

TIPO DE PROGRAMA

DRIVER [ ] SUBROUTINE [x] PRINCIPAL FUNCTION [ ] OUTROS [ ]

IDENTIFICAÇÃO DOS DISCOS (FITAS)	SISTEMA SUPORTE (SIST. OPERACIONAL E HARDWARE)
0010	

CHAVES DO PAINEL OU SAIDAS ALTERNATIVAS

DESCRIÇÃO DO PROGRAMA

Esta rotina trata da saída das instruções curtas, das longas de endereçamento da memória, e de certas pseudos.

São impressos: ordem de entrada do cartão, contador de instruções (em hexa), instrução montada (em hexa) e comando fonte. O contador de instruções é incrementado do número de instruções conveniente. A rotina providencia a colocação das instruções no buffer de disco.

No caso das instruções longas, somente a primeira palavra é colocada no buffer de disco. As demais são tratadas pela rotina tratadora de endereços. (uma a mais, que é zero).

 (use o verso, se necessário)

CÓDIGO DO PROGRAMA

WZSAI

SEQUÊNCIA DE CHAMADA E DESCRIÇÃO DAS VARIÁVEIS - ARGUMENTO

CALL WZSAI(IOCT,N2,ISTEP,IK,JJ,SWITC)

IOCT - insttrução montada

N2 - se diferente de zero, segunda palavra de um "name code" em chamada de subrotina

ISTEP - contador de instruções

IK - diz se a instrução é longa ou curta (1 ou 0)

JJ - ordem de entrada do cartão

SWITC - indica a ocorrência de uma pseudo EBC ou ASC quando então é o contador de palavras.

SUB-ROTINAS USADAS

WZITR - descompactação binário em hexadecimal

WZTRT - trata da saída no buffer de disco

OBSERVAÇÕES [illegible] OU MENSAGENS DE ERROS

 (use o verso, se necessário)

CÓDIGO DE PROGRAMA

WZSAI

VARIÁVEIS PRINCIPAIS

TIPO
(Dimension, Common, Real, Integer, Data, etc)

Nome	Tipo	Função da Variável
LST	I,C	controle de listagem
INIC	I,C,()	caracteres da linguagem

ANEXOS: DIAGRAMA DE BLOCOS [ ]

LISTAGEM DO PROGRAMA [ ]

LISTAGEM DO EXEMPLO [ ]

GABARITO DOS CARTÕES DE ENTRADA OU DA IMPRESSORA [ ]

FOLHA DE ALTERAÇÃO VERSÃO 1 [ ]

VERSÃO = [ ]

```
// *

// FOR
* ONE WORD INTEGERS
*LIST SOURCE PROGRAM
      SUBROUTINE WZSAI(IOCT,N2,ISTEP,IK,JJ,SWITC)
      INTEGER SWITC
      DIMENSION ICOL(80)
      COMMON IPAG,KONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEQ,NOEQU(100),NOSIM(400,5),IENSI(400)
      COMMON IORIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ISAI(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
      EQUIVALENCE (ICOL(1),IERRA(1))
 7020 CALL WZTRT(IOCT,N2,ISTEP,IK)
      IF(LST)7030,7060,7030
 7030 CALL WZITR(IOCT,IS1,IS2,IS3,IS4)
      CALL WZITR(ISTEP,K1,K2,K3,K4)
      GO TO (7032,7031),IK
 7031 WRITE(IPRIN,7500)JJ,INIC(K1),INIC(K2),INIC(K3),INIC(K4),
     1                  INIC(IS1),INIC(IS2),INIC(IS3),INIC(IS4),
     2                  ICOL
 7500 FORMAT(1H ,I5,2X, 4A1,2X,4A1,6X,80A1)
      GO TO 7040
 7032 WRITE(IPRIN,7501) JJ,INIC(K1),INIC(K2),INIC(K3),INIC(K4),
     1                   INIC(IS1),INIC(IS2),ICOL
 7501 FORMAT(1H ,I5,2X,4A1,2X,2A1,8X,80A1)
 7040 IF(N2)7050,7055,7050
 7050 CALL WZITR(N2,IS1,IS2,IS3,IS4)
      WRITE (IPRIN,7600) INIC(IS1),INIC(IS2),INIC(IS3),INIC(IS4)
 7600 FORMAT(1H+,18X,4A1)
      ISTEP=ISTEP+2
 2105 RETURN
 7055 SWITC=SWITC-1
      IF(SWITC)7060,7060,7070
 7060 ISTEP=ISTEP+IK
      GO TO 2105
 7070 ISTEP=ISTEP+1
      GO TO 2105
      END

UNREFERENCED STATEMENTS
 7020

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZSAI
 COMMON   3130  VARIABLES     16  PROGRAM    290

RELATIVE ENTRY POINT ADDRESS IS 0032 (HEX)

END OF COMPILATION

// DUP

*DELETE              WZSAI     0010
```

D 26 NAME NOT FOUND IN LET/FLET

```
*STORE      WS  UA  WZSAI            0010
CART ID 0010   DB ADDR  0327   DB CNT   0014
```

// EJECT

NOME DO PROGRAMA	CÓDIGO DE ATIVIDADE NUM. DO DOCUMENTO	CÓDIGO DO PROGRAMA
ROTINA DE DETERMINAÇÃO DA DEFINIÇÃO, NÃO DEFINIÇÃO E SOBREDEFINIÇÃO DE UM SIMBOLO		WZTES
	PROGRAMADOR	/ / DATA

LINGUAGEM USADA

FORTRAN [x] ASSEMBLER [ ] BASIC [ ] ALGOL [ ] .... [ ]

TIPO DE PROGRAMA

PRINCIPAL

DRIVER [ ] SUBROUTINE [x] FUNCTION [ ] OUTROS [ ]

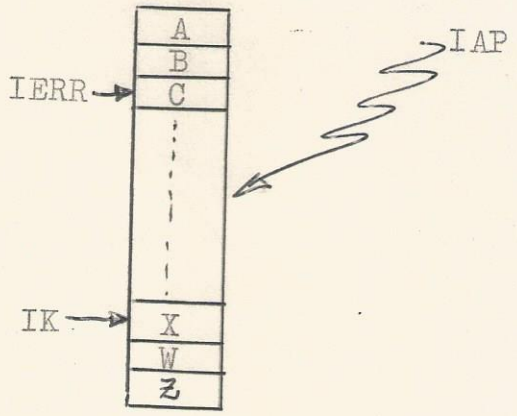
IDENTIFICAÇÃO DOS DISCOS (FITAS)	SISTEMA SUPORTE (SIST. OPERACIONAL E HARDWARE)
0010	

CHAVES DO PAINEL OU SAIDAS ALTERNATIVAS

NENHUMA

DESCRIÇÃO DO PROGRAMA

Esta rotina determina se um símbolo é definido, sobredefinido ou indefinido. Os símbolos não definidos e sobredefinidos são colocados em uma pilha, cuja configuração é mostrada abaixo:



 (use o verso, se necessário)

CÓDIGO DO PROGRAMA

WZTES

SEQUÊNCIA DE CHAMADA E DESCRIÇÃO DAS VARIÁVEIS - ARGUMENTO

CALL WZTES(N)

N - número de símbolos não definidos + número de símbolos sobredefinidos

SUB-ROTINAS USADAS

NENHUMA

OBSERVAÇÕES [illegible] OU MENSAGENS DE ERROS

NENHUMA

 (use o verso, se necessário)

CÓDIGO DE PROGRAMA

WZTES

VARIÁVEIS PRINCIPAIS

TIPO
(Dimension, Common, Real, Integer, Data, etc)

Nome	Tipo	Função da Variável
IK	I,C	Contador de símbolos sobredefinidos
IERR	I,C	Contador de símbolos não definidos
IAP	I,C,()	Pilha dos símbolos em condição irregular

ANEXOS: DIAGRAMA DE BLOCOS [ ]

LISTAGEM DO PROGRAMA [ ]

LISTAGEM DO EXEMPLO [ ]

GABARITO DOS CARTÕES DE ENTRADA OU DA IMPRESSORA [ ]

FOLHA DE ALTERAÇÃO VERSÃO 1 [ ]

VERSÃO = [ ]

```
// *

// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
      SUBROUTINE WZTES(N)
C
C     TESTA SUBDEFINICAO OU SOBREDEFINICAO
C
      DIMENSION IAP(100)
      COMMON IPAG,KONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEQ,NOEQU(100),NOSIM(400,5),IENSI(400)
      COMMON IORIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ICOL(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
      EQUIVALENCE (IAP(1),NOSIM(1,5))
      EQUIVALENCE (IK,IS)
C
      IK=101
      N=0
      IF(IAPSI) 5,50,5
    5 DO 20 I=1,IAPSI
      K=NOSIM(I,4)+1
      GO TO (10,20,20,20,20,10,10,20,20,20,30),K
   10 IERR=IERR+1
      IAP(IERR)=I
      GO TO 25
   30 IK=IK-1
      IAP(IK)=I
   25 N=N+1
   20 CONTINUE
   50 RETURN
      END

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZTES
 COMMON   3130  VARIABLES      4  PROGRAM    102

RELATIVE ENTRY POINT ADDRESS IS 0007 (HEX)

END OF COMPILATION

// DUP

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


*STORE      WS  UA  WZTES            0010
CART ID 0010   DB ADDR  02E4   DB CNT   0007

// EJECT
```

NOME DO PROGRAMA	CÓDIGO DE ATIVIDADE NUM. DO DOCUMENTO	CÓDIGO DO PROGRAMA
ROTINA PARA IMPRESSÂO DOS SIMBOLOS SIMBOLOS NÂO DEFINIDOS E SOBREDEFINIDOS		WZRSL
	PROGRAMADOR	/ / DATA

LINGUAGEM USADA

FORTRAN [X] ASSEMBLER [ ] BASIC [ ] ALGOL [ ] .... [ ]

TIPO DE PROGRAMA

PRINCIPAL

DRIVER [ ] SUBROUTINE [X] FUNCTION [ ] OUTROS [ ]

IDENTIFICAÇÃO DOS DISCOS (FITAS)	SISTEMA SUPORTE (SIST. OPERACIONAL E HARDWARE)
0010	

CHAVES DO PAINEL OU SAIDAS ALTERNATIVAS

NENHUMA

DESCRIÇÃO DO PROGRAMA

Esta rotina imprime os símbolos não definidos e sobre - definidos, que se encontram nas tabelas de não definição e sobredefinição.

 (use o verso, se necessário)

CÓDIGO DO PROGRAMA

WZRSL

SEQUÊNCIA DE CHAMADA E DESCRIÇÃO DAS VARIÁVEIS - ARGUMENTO

CALL WZRSL

SUB-ROTINAS USADAS

NENHUMA

OBSERVAÇÕES [illegible] OU MENSAGENS DE ERROS

Imprime as seguintes mensagens, seguidas dos simbolos em situação irregular:

SIMBOLOS NÃO DEFINIDOS e

SIMBOLOS DUPLICADOS

 (use o verso, se necessário)

CÓDIGO DE PROGRAMA

WZRSL

VARIÁVEIS PRINCIPAIS

TIPO
(Dimension, Common,Real,Integer,Data,etc)

Nome	Tipo	Função da Variável
IERR	I,C	Contador de símbolos não definidos
KONT	I,C	Contador de linhas impressas
IAP	I,C,()	Tabela de símbolos em situação irregular
NOSIM	I,C,()	Tabela de símbolos
IK	I,C	Contador de símbolos em sobredefinição

ANEXOS: DIAGRAMA DE BLOCOS [ ]

LISTAGEM DO PROGRAMA [ ]

LISTAGEM DO EXEMPLO [ ]

GABARITO DOS CARTÕES DE ENTRADA OU DA IMPRESSORA [ ]

FOLHA DE ALTERAÇÃO VERSÃO 1 [ ]

VERSÃO = [ ]

```
// JOB    0001 0010                                          *     E102,MA

LOG DRIVE   CART SPEC   CART AVAIL  PHY DRIVE
  0000        0001        0001        0000
  0001        0010        0010        0002
                          302F        0001

V2 M09   ACTUAL 16K  CONFIG 16K

* LABORATORIO DE SISTEMAS DIGITAIS * EPUSP *  15/ 04/ 74  * 16 H  10 MI

// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
      SUBROUTINE WZRSL
      DIMENSION IAP(100)
      COMMON IPAG,KONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEQ,NOEQU(100),NOSIM(400,5),IENSI(400)
      COMMON IORIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ICOL(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
      EQUIVALENCE (IAP(1),NOSIM(1,5)),(IS,IK)
C
C     SIMBOLOS NAO DEFINIDOS
C
      IF(IERR)  100,200,100
  100 WRITE(IPRIN,10)
   10 FORMAT(1H0,20X,'SIMBOLOS NAO DEFINIDOS')
      KONT=KONT+2
      II=1
  500 DO 550 J=1,IERR
      I=IAP(J)
      WRITE(IPRIN,38)(NOSIM(I,L),L=1,3)
   38 FORMAT(1H ,20X,3A2)
      GO TO 680
  550 CONTINUE
C
C     SIMBOLOS SOBREDEFINIDOS
C
  200 IF(IK-101)250,300,250
  250 WRITE(IPRIN,20)
   20 FORMAT (1H0,20X,'SIMBOLOS DUPLICADOS')
      KONT=KONT+2
      II=2
  600 J=100
  602 I=IAP(J)
      WRITE(IPRIN,38)(NOSIM(I,L),L=1,3)
      GO TO 680
  650 J=J-1
      IF(J-IK) 300,602,602
C
C     SIMBOLOS NAO USADOS
C
  300 RETURN
C
C     PAGINACAO
C
```

```
  680 KONT=KONT+1
      IF(KONT-53) 690,685,685
  685 CALL WZPAG
  690 GO TO (550,650,700),II
  700 RETURN
      END

UNREFERENCED STATEMENTS
 500     600

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZRSL
 COMMON   3130  VARIABLES      6  PROGRAM    208

RELATIVE ENTRY POINT ADDRESS IS 0031 (HEX)

END OF COMPILATION

// DUP

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


*STORE      WS  UA  WZRSL            0010
CART ID 0010   DB ADDR  05A3   DB CNT   000E

// EJECT
```

ROTINAS DE INICIAÇÃO

Estas rotinas destinam-se a iniciação do sistema, tanto no primeiro, como no segundo passo.

WZ2
WZ1
WZHDR
WZINI

NOME DO PROGRAMA	CÓDIGO DE ATIVIDADE NUM. DO DOCUMENTO	CÓDIGO DO PROGRAMA
ROTINA DE INICIALIZAÇÃO DO SEGUNDO PASSO		WZ2
	PROGRAMADOR	/ / DATA

LINGUAGEM USADA

FORTRAN [x] ASSEMBLER [ ] BASIC [ ] ALGOL [ ]....[ ]

TIPO DE PROGRAMA

DRIVER [ ] SUBROUTINE [x] PRINCIPAL FUNCTION [ ] OUTROS [ ]

IDENTIFICAÇÃO DOS DISCOS (FITAS)	SISTEMA SUPORTE (SIST. OPERACIONAL E HARDWARE)
0010	

CHAVES DO PAINEL OU SAIDAS ALTERNATIVAS

NENHUMA

DESCRIÇÃO DO PROGRAMA

Esta subrotina inicia ponteiros, indicadores, ponteiros de arquivos de disco, contadores, e lê o primeiro arquivo de disco.

O contador de erros é iniciado em zero..

O indicador de listagem de comandos é posicionado para que a listagem seja produzida. O indicador de supressão de comandos de controle é posicionado para que tais comandos sejam suprimidos.

 (use o verso, se necessário)

CÓDIGO DO PROGRAMA

WZ2

SEQUÊNCIA DE CHAMADA E DESCRIÇÃO DAS VARIÁVEIS - ARGUMENTO

CALL WZ2(ISTEP,IA,IB)

ISTEP- contador de instruções

IA - indicará se o programa é absoluto ou relativo

IB - ~~número da disk xxxx~~ próximo endereço da memória.

SUB-ROTINAS USADAS

WZHDR - inicia cabeçalho de disco para o programa

OBSERVAÇÕES ~~ENDEREÇAMENTO~~ OU MENSAGENS DE ERROS

NENHUMA

2 (use o verso, se necessário)

CÓDIGO DE PROGRAMA
WZ2

VARIÁVEIS PRINCIPAIS

Nome	TIPO (Dimension, Common, Real, Integer, Data, etc) Tipo	Função da Variável
ISTEP	I,C	Contador de instruções
ISTRT	I,C	Contador de instruções inicial para o bloco de dados
ICNT	I,C	Indicador de listagem de comandos de controle
LST	I,C	Indicador de listagem de comandos fonte
ISWER	I,C	Ponteiro para a lista de erros
IERR	I,C	Conatdor do total de erros no programa
IAC	I,C	Contador de dados no bloco de dados
IAPTC	I,C	Ponteiro para o buffer de informação suplementar

ANEXOS: DIAGRAMA DE BLOCOS [ ]

LISTAGEM DO PROGRAMA [ ]

LISTAGEM DO EXEMPLO [ ]

GABARITO DOS CARTÕES DE ENTRADA OU DA IMPRESSORA [ ]

FOLHA DE ALTERAÇÃO VERSÃO 1 [ ]

VERSÃO = [ ]

```
// *

// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
      SUBROUTINE WZ2(ISTEP,IABS,IGLU)
      DIMENSION ISTRP(80)
      COMMON IPAG,KONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEQ,NOEQU(100),NOSIM(400,5),IENSI(400)
      COMMON IORIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ISAI(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
      EQUIVALENCE (NR,N7),(IDISK,ID7)
      EQUIVALENCE (ISTRP(1),ISTEC(1))
C
C     INICIALIZACAO DO GERADOR
C
      ISTEP=0
      ISTRT=ISTEP
C
C     GERA CABECALHOS DO PROGRAMA
C
      CALL WZHDR(ISTEP,IABS,IGLU)
C
C     INICIALIZA CONTROLES DE IMPRESSAO
C
      IPAG=0
      ICNT=0
      LST=1
C
C     INICIALIZA INDICADORES DE ERROS
C
      ISWER=1
      IERR=0
C
C     INICIALIZA APONTADORES DO ARQUIVO EM DISCO
C
      N5=1
      NR=1
      N6=1
      IDISK=8
      READ(ID6'N6) ISTRP
      IAC=2
      IAPTC=1
      RETURN
      END

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZ2
 COMMON   3130  VARIABLES      0  PROGRAM     80

RELATIVE ENTRY POINT ADDRESS IS 0004 (HEX)

END OF COMPILATION
```

NOME DO PROGRAMA	CÓDIGO DE ATIVIDADE NUM. DO DOCUMENTO	CÓDIGO DO PROGRAMA
Inicialização do Primeiro Passo	WZ1	WZ1
	PROGRAMADOR	/ / DATA

LINGUAGEM USADA

FORTRAN [X] ASSEMBLER [ ] BASIC [ ] ALGOL [ ] .... [ ]

TIPO DE PROGRAMA

DRIVER [ ] SUBROUTINE [X] PRINCIPAL FUNCTION [ ] OUTROS [ ]

IDENTIFICAÇÃO DOS DISCOS (FITAS)	SISTEMA SUPORTE (SIST. OPERACIONAL E HARDWARE)

CHAVES DO PAINEL OU SAIDAS ALTERNATIVAS

DESCRIÇÃO DO PROGRAMA

Esta rotina inicializa uma série de indicadores do Primeiro Passo:

Cabeçalho de listagem - feito branco

Indicador de erro = 0

Ponteiros { tabela de símbolos
tabela de equivalência

Buffer de disco - inicializado em zero

1 (use o verso, se necessário)

CÓDIGO DO PROGRAMA

WZ1

SEQUÊNCIA DE CHAMADA E DESCRIÇÃO DAS VARIÁVEIS - ARGUMENTO

CALL WZ1

SUB-ROTINAS USADAS

WZINI

OBSERVAÇÕES [illegible] OU MENSAGENS DE ERROS

 (use o verso, se necessário)

CÓDIGO DE PROGRAMA

WZ1

VARIÁVEIS PRINCIPAIS

TIPO
(Dimension, Common, Real, Integer, Data, etc)

Nome	Tipo	Função da Variável
IPRIN	C	nº do dispositivo lógico de impressão
ID5	C	" " " " de leitura de cartões
ID6	C	" " " " do arquivo de inf. suplementar
ID7	C	" " " " do arquivo de erros
NOSIM(100,5)	C	tabela de símbolos
ITGSI(100)	C	indicador de símbolos ñ usados
ICAB (37)	C	cabecalho de páginas
ISWER	C	indicador de erros
IERR	C	contador de erros
NSTAT	C	indicador de programa absoluto

ANEXOS: DIAGRAMA DE BLOCOS [ ]
LISTAGEM DO PROGRAMA [ ]
LISTAGEM DO EXEMPLO [ ]
GABARITO DOS CARTÕES DE ENTRADA OU DA IMPRESSORA [ ]
FOLHA DE ALTERAÇÃO VERSÃO 1 [ ]
VERSÃO = [ ]

```
// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
      SUBROUTINE WZ1
C
C     ESTA ROTINA FAZ INICIALIZACAO DO COMMON
C
      COMMON IPAG,KONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEQ,NOEQU(100),NOSIM(400,5),IENSI(400)
      COMMON IORIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ICOL(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
      EQUIVALENCE (IBCD,INIC(38)),(IAC ,NSTAT)
C
      CALL WZINI(NUM,INIC)
      IPRIN=5
      ISWER=0
      NSTAT=0
C
C     ZERA BUFFER DE CABECALHO
C
      DO 1 I=1,37
    1 ICAB(I)=IBCD
C
C     ZERA INDICADORES DE DEFINICAO DE SIMBOLO
C
      DO 10 I=1,400
   10 NOSIM(I,4)=0
C
C     INICIALIZA INDICADORES DE DISCO
C
      ID5=9
      ID6=6
      ID7=7
      N5=1
      N6=1
      N7=1
      IAPER=2
      IERR=0
C
C     ZERA PONTEIROS
C
      IAPSI=0
      IAPEQ=0
      IAPTC=1
C
      RETURN
      END

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZ1
 COMMON   3130  VARIABLES      2  PROGRAM    116

RELATIVE ENTRY POINT ADDRESS IS 000B (HEX)
```

END OF COMPILATION


```
// DJP

*DELETE              WZ1        0010
D 26 NAME NOT FOUND IN LET/FLET


*STORE       WS  JA  WZ1                0010
CART ID 0010   DB ADDR  0477   DB CNT   0008

// EJECT
```

NOME DO PROGRAMA	CÓDIGO DE ATIVIDADE NUM. DO DOCUMENTO	CÓDIGO DO PROGRAMA
INICIALIZAÇÃO DO CABEÇALHO DE DISCO		WZHDR
	PROGRAMADOR	/ / DATA

LINGUAGEM USADA

FORTRAN [X] ASSEMBLER [ ] BASIC [ ] ALGOL [ ] .... [ ]

TIPO DE PROGRAMA

DRIVER [ ] SUBROUTINE [X] PRINCIPAL FUNCTION [ ] OUTROS [ ]

IDENTIFICAÇÃO DOS DISCOS (FITAS)	SISTEMA SUPORTE (SIST. OPERACIONAL E HARDWARE)
0010	

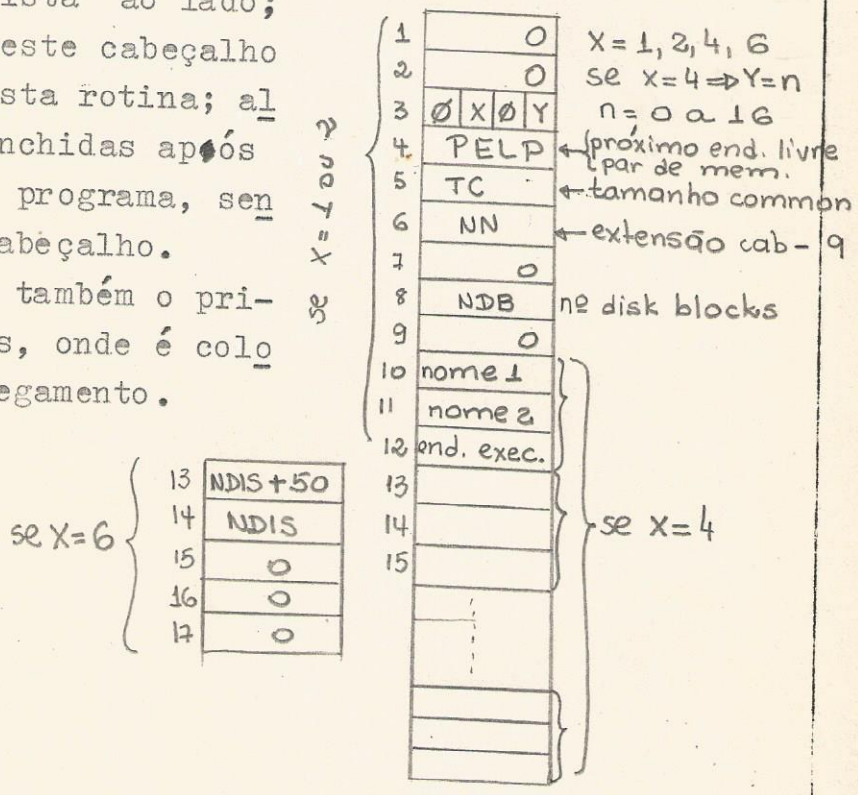
CHAVES DO PAINEL OU SAIDAS ALTERNATIVAS

NENHUMA

DESCRIÇÃO DO PROGRAMA

Este programa com auxílio das informações armazenadas na primeira fase, gera o cabeçalho da saída de disco. A configuração deste cabeçalho é vista ao lado; nem todas as posições deste cabeçalho são porém geradas por esta rotina; algumas só podem ser preenchidas após o término de geração do programa, sendo então completado o cabeçalho.

O programa inicia também o primeiro cabeçalho de dados, onde é colocado o endereço de carregamento.



 (use o verso, se necessário)

CÓDIGO DO PROGRAMA

WZHDR

SEQUÊNCIA DE CHAMADA E DESCRIÇÃO DAS VARIÁVEIS - ARGUMENTO

CALL WZHDR(ISTEP,IABS,IB)

ISTEP - contador de instruções

IABS - indica se o programa é absoluto ou relativo

IB - indicará qual o tipo de programa que se tem.

SUB-ROTINAS USADAS

WZDPK - desempacota o nome do programa em 5 caracteres

WZCDM - transforma os 5 caracteres do nome do programa em código M.

OBSERVAÇÕES OU MENSAGENS DE ERROS

NENHUMA

 (use o verso, se necessário)

	CÓDIGO DE PROGRAMA
	WZHDR

VARIÁVEIS PRINCIPAIS

TIPO
(Dimension, Common,Real,Integer,Data,etc)

Nome	Tipo	Função da Variável
N	I,C,()	matriz-buffer de saída - equivale a ISAI
NSTAT	I,C	indicador de programa subrotina de E/S se valer 1, tem-se este tipo de rotina
IENTR	I,C,	se diferente de zero indica que se trata de uma subrotina, indicando neste caso quantos pontos de entrada esta apresenta
IORIG	I,C	se diferente de zero, indica que se tem um programa absoluto, neste caso, contendo o valor do ponto de carregamento
MOSIM	I,C,()	tabela de símbolos
IAPSI	I,C	ponteiro para a tabela de símbolos
NAME	I	matriz auxiliar onde reside o nome de um programa principal
KR	I,C	indica na tabela de símbolos, a posição de um ponto de entrada,; se se tratar de subrotina de entrada/saída indica o seu entry - point; se não for, indica o primeiro entry--point de uma série de entry points encadeados
ISOMA	I,C	extensão do common do programa
ICOMP	I,C	total de posições vagas no buffer de saída
IS	I,C	extensão deste cabeçalho

ANEXOS: DIAGRAMA DE BLOCOS [ ]
LISTAGEM DO PROGRAMA [ ]
LISTAGEM DO EXEMPLO [ ]
GABARITO DOS CARTÕES DE ENTRADA OU DA IMPRESSORA [ ]
FOLHA DE ALTERAÇÃO VERSÃO 1 [ ]
VERSÃO = [ ]

```
// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
      SUBROUTINE WZHDR(ISTEP,IABS,IGLU)
C
      DIMENSION ICOL(80)
      DIMENSION N(80),NAME(5)
C
      COMMON IPAG,KONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEQ,NOEQU(100),NOSIM(400,5),IENSI(400)
      COMMON IORIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ISAI(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
      COMMON IXXX,IPENT
      EQUIVALENCE (ICOL(1),IERRA(1)),(IKONT,IORIG)
      EQUIVALENCE (N(1),ISAI(1)),(IAPIN,K)
      EQUIVALENCE (ICNT,KR),(LST,IENTR),(IAC,NSTAT)
      DATA NAME(1),NAME(2),NAME(3)/'MA','IN','  '/
      DATA NN,ISOMA/3,0/
C
C     INICIALIZACAO
C
      IGLU=2
      N(1)=0
      N(2)=0
      N(3)=2
      N(5)=0
      N(6)=3
      N(7)=0
      N(9)=0
      N(15)=0
      N(16)=0
      N(17)=0
C
      IF(NSTAT-1) 100,200,100
  100 IF(IENTR)150,130,150
C
C     MAIN PROGRAM
C
  130 IF(IORIG) 190,193,190
C
C     SUBROUTINE
C
  150 NN=IENTR*3
      IGLU=4
      N(6)=NN
      N(3)=4
      GO TO 400
C
C        ABSOLUTO
C
190   N(3)=1
      IGLU=1
  193 IF(IAPSI) 194,400,194
  194 DO 195 I=1,IAPSI
      IF(NOSIM(I,4)-7) 195,197,195
  195 CONTINUE
```

```
// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
      SUBROUTINE WZHDR(ISTEP,IABS,IGLU)
C
      DIMENSION ICOL(80)
      DIMENSION N(80),NAME(5)
C
      COMMON IPAG,KONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEQ,NOEQU(100),NOSIM(400,5),IENSI(400)
      COMMON IORIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ISAI(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
      COMMON IXXX,IPENT
      EQUIVALENCE (ICOL(1),IERRA(1)),(IKONT,IORIG)
      EQUIVALENCE (N(1),ISAI(1)),(IAPIN,K)
      EQUIVALENCE (ICNT,KR),(LST,IENTR),(IAC,NSTAT)
      DATA NAME(1),NAME(2),NAME(3)/'MA','IN','  '/
      DATA NN,ISOMA/3,0/
C
C     INICIALIZACAO
C
      IGLU=2
      N(1)=0
      N(2)=0
      N(3)=2
      N(5)=0
      N(6)=3
      N(7)=0
      N(9)=0
      N(15)=0
      N(16)=0
      N(17)=0
C
      IF(NSTAT-1) 100,200,100
  100 IF(IENTR)150,130,150
C
C     MAIN PROGRAM
C
  130 IF(IORIG) 190,193,190
C
C     SUBROUTINE
C
  150 NN=IENTR*3
      IGLU=4
      N(6)=NN
      N(3)=4
      GO TO 400
C
C        ABSOLUTO
C
190   N(3)=1
      IGLU=1
  193 IF(IAPSI) 194,400,194
  194 DO 195 I=1,IAPSI
      IF(NOSIM(I,4)-7) 195,197,195
  195 CONTINUE
```

```
      GO TO 400
  197 DO 199 J=1,3
  199 NAME(J)=NOSIM(I,J)
      GO TO 400
C
C     ISS
C
  200 N(3)=6
      IGLU=6
      N(6)=8
      DO 350 J=1,3
  350 NAME(J)= NOSIM(KR,J)
      IPENT=IENSI(KR)
  400 N(3)=N(3)*256
C
C     EXISTIRA COMMON&
C
  480 IF(IAPSI)481,491,481
  481 DO 490 J=1,IAPSI
      IF(NOSIM(J,4)-3) 490,2000,490
  490 CONTINUE
  491 N(5)=ISOMA
C
C     PRINCIPAL
      IF(IENTR) 1400,1050,1400
 1050 IF(NSTAT-1)1100,1300,1100
 1100 K=13
 1149 CALL WZDPK(NAME)
      CALL WZCDM(NAME,N1,N2)
      N(10)=N1
      N(11)=N2
      N(12)=IPENT
      GO TO 1600
C
C     ISS
C
 1300 N(14)=NOSIM(KR,5)
      N(13)=N(14)+50
      K=18
      IORIG=0
      GO TO 1149
C
C     SUBROUTINE
C
1400  K=10
 1499 DO 1500 I=1,3
 1500 NAME(I)=NOSIM(KR,I)
      N(K+2)= IENSI(KR)
      CALL WZDPK(NAME)
      CALL WZCDM(NAME,N1,N2)
      N(K)=N1
      N(K+1)=N2
      K=K+3
      KR=NOSIM(KR,5)
      IF(KR) 1499, 1600, 1499
 1600 K=K+1
      ICOMP=320-K
```

```
      IKONT=0
      IS=K
      RETURN
 2000 ISOMA=ISOMA+NOSIM(J,5)
      GO TO 490
      END

UNREFERENCED STATEMENTS
 480

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZHDR
 COMMON   3132  VARIABLES     22  PROGRAM    506

RELATIVE ENTRY POINT ADDRESS IS 0024 (HEX)

END OF COMPILATION

// DUP

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


*STORE      WS  UA  WZHDR           0010
CART ID 0010   DB ADDR  05B1   DB CNT   0020

// EJECT
```

NOME DO PROGRAMA	CÓDIGO DE ATIVIDADE NUM. DO DOCUMENTO	CÓDIGO DO PROGRAMA
INICIALIZACAO DA MATRIZ DE CARACTERES E NÚMEROS		WZINI
	PROGRAMADOR	/ / DATA

LINGUAGEM USADA

FORTRAN [ ] ASSEMBLER [X] BASIC [ ] ALGOL [ ] .... [ ]

TIPO DE PROGRAMA

PRINCIPAL

DRIVER [ ] SUBROUTINE [X] FUNCTION [ ] OUTROS [ ]

IDENTIFICAÇÃO DOS DISCOS (FITAS)	SISTEMA SUPORTE (SIST. OPERACIONAL E HARDWARE)

CHAVES DO PAINEL OU SAIDAS ALTERNATIVAS

DESCRIÇÃO DO PROGRAMA

Inicializa matrizes do common na 1ª fase: a matriz de caracteres e a de números

 (use o verso, se necessário)

CÓDIGO DO PROGRAMA

W2INI

SEQUÊNCIA DE CHAMADA E DESCRIÇÃO DAS VARIÁVEIS - ARGUMENTO

CALL W2INI

SUB-ROTINAS USADAS

—//—

OBSERVAÇÕES [illegible] OU MENSAGENS DE ERROS

—//—

(use o verso, se necessário)

CÓDIGO DE PROGRAMA

W2INI

VARIÁVEIS PRINCIPAIS

TIPO
(Dimension, Common, Real, Integer, Data, etc)

Nome	Tipo	Função da Variável
INUMI	I	matriz de números
INICI	I	matriz de caracteres.

ANEXOS:
- DIAGRAMA DE BLOCOS [ ]
- LISTAGEM DO PROGRAMA [ ]
- LISTAGEM DO EXEMPLO [ ]
- GABARITO DOS CARTÕES DE ENTRADA OU DA IMPRESSORA [ ]
- FOLHA DE ALTERAÇÃO VERSÃO 1 [ ]
  VERSÃO = [ ]

```
0000     26A49549 00001          ENT     WZINI
                  00002     *
                  00003     *     ROTINA DE INICIALIZACAO DAS MATRIZES DE CA
                  00004     *     TERES E DE NUMEROS
                  00005     *     OS NUMEROS VAO DE 0 A 15 PARA COMPREENDER
                  00006     *     CONJUNTO HEXADECIMAL,
                  00007     *
0000 0   0000     00008     WZINI DC      0
0001 0   691E     00009           STX   1 SAVE1+1
0002 0   6A1F     00010           STX   2 SAVE2+1
0003 01  65800000 00011           LDX  I1 WZINI
0005 00  C5000000 00012           LD   L1 0
0007 01  84000025 00013           A    L  UM
0009 0   D010     00014           STO     VORTA+3
000A 00  C5000001 00015           LD   L1 1
000C 01  84000025 00016           A    L  UM
000E 0   D004     00017           STO     VOLTA+3
000F 0   62C2     00018           LDX   2 -62
0010 01  C6000074 00019     VOLTA LD   L2 INICI+62
0012 00  D6000000 00020           STO  L2 0
0014 0   7201     00021           MDX   2 1
0015 0   70FA     00022           MDX     VOLTA
0016 0   62F0     00023           LDX   2 -16
0017 01  C6000036 00024     VORTA LD   L2 INUMI+16
0019 00  D6000000 00025           STO  L2 0
001B 0   7201     00026           MDX   2 1
001C 0   70FA     00027           MDX     VORTA
001D 0   7102     00028           MDX   1 2
001E 0   6905     00029           STX   1 DONE+1
001F 00  65000000 00030     SAVE1 LDX  L1 0
0021 00  66000000 00031     SAVE2 LDX  L2 0
0023 00  4C000000 00032     DONE  BSC  L  0
0025 0   0001     00033     UM    DC      1
0026 0   0009     00034     INUMI DC      9
0027 0   0008     00035           DC      8
0028 0   0007     00036           DC      7
0029 0   0006     00037           DC      6
002A 0   0005     00038           DC      5
002B 0   0004     00039           DC      4
002C 0   0003     00040           DC      3
002D 0   0002     00041           DC      2
002E 0   0001     00042           DC      1
002F 0   0000     00043           DC      0
0030 0   000F     00044           DC      15
0031 0   000E     00045           DC      14
0032 0   000D     00046           DC      13
0033 0   000C     00047           DC      12
0034 0   000B     00048           DC      11
0035 0   000A     00049           DC      10
0036     0001     00050     INICI EBC     . .
0037     0001     00051           EBC     . .
0038     0001     00052           EBC     . .
0039     0001     00053           EBC     .'.
003A     0001     00054           EBC     . .
003B     0001     00055           EBC     . .
003C     0001     00056           EBC     . .
003D     0001     00057           EBC     .+.
```

```
003E    0001    00058    EBC    . .
003F    0001    00059    EBC    . .
0040    0001    00060    EBC    . .
0041    0001    00061    EBC    .'.
0042    0001    00062    EBC    .$.
0043    0001    00063    EBC    . .
0044    0001    00064    EBC    .).
0045    0001    00065    EBC    ./.
0046    0001    00066    EBC    .=.
0047    0001    00067    EBC    .*.
0048    0001    00068    EBC    .=.
0049    0001    00069    EBC    .).
004A    0001    00070    EBC    .,.
004B    0001    00071    EBC    .(.
004C    0001    00072    EBC    .-.
004D    0001    00073    EBC    .+.
004E    0001    00074    EBC    . .
004F    0001    00075    EBC    .9.
0050    0001    00076    EBC    .8.
0051    0001    00077    EBC    .7.
0052    0001    00078    EBC    .6.
0053    0001    00079    EBC    .5.
0054    0001    00080    EBC    .4.
0055    0001    00081    EBC    .3.
0056    0001    00082    EBC    .2.
0057    0001    00083    EBC    .1.
0058    0001    00084    EBC    .0.
0059    0001    00085    EBC    .F.
005A    0001    00086    EBC    .E.
005B    0001    00087    EBC    .D.
005C    0001    00088    EBC    .C.
005D    0001    00089    EBC    .B.
005E    0001    00090    EBC    .A.
005F    0001    00091    EBC    .Z.
0060    0001    00092    EBC    .Y.
0061    0001    00093    EBC    .X.
0062    0001    00094    EBC    .W.
0063    0001    00095    EBC    .V.
0064    0001    00096    EBC    .U.
0065    0001    00097    EBC    .T.
0066    0001    00098    EBC    .S.
0067    0001    00099    EBC    .R.
0068    0001    00100    EBC    .Q.
0069    0001    00101    EBC    .P.
006A    0001    00102    EBC    .O.
006B    0001    00103    EBC    .N.
006C    0001    00104    EBC    .M.
006D    0001    00105    EBC    .L.
006E    0001    00106    EBC    .K.
006F    0001    00107    EBC    .J.
0070    0001    00108    EBC    .I.
0071    0001    00109    EBC    .H.
0072    0001    00110    EBC    .G.
0073    0001    00111    EBC    ...
0074            00112    END    WZINI

    000 OVERFLOW SECTORS SPECIFIED
```

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

// DUP

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


*STORE      WS  JA  WZINI           0010
CART ID 0010   DB ADDR  039E   DB CNT   0008

// EJECT
```

ROTINAS TRATADORAS DE PSEUDOS

Estas rotinas destinam-se ao tratamento das pseudo-instruções, na primeira ou na segunda fase.

WZENT
WZCNT
WZCOD
WZEQU
WZISS

NOME DO PROGRAMA Rotina de tratamento dos pontos de entrada	CÓDIGO DE ATIVIDADE NUM. DO DOCUMENTO	CÓDIGO DO PROGRAMA WZENT
	PROGRAMADOR	/ / DATA

LINGUAGEM USADA

FORTRAN [X] ASSEMBLER [ ] BASIC [ ] ALGOL [ ]....[ ]

TIPO DE PROGRAMA

PRINCIPAL

DRIVER [ ] SUBROUTINE [X] FUNCTION [ ] OUTROS [ ]

IDENTIFICAÇÃO DOS DISCOS (FITAS)	SISTEMA SUPORTE (SIST. OPERACIONAL E HARDWARE)

CHAVES DO PAINEL OU SAIDAS ALTERNATIVAS

DESCRIÇÃO DO PROGRAMA

Este programa trata dos pontos de entrada de uma subrotina. O número máximo de "entry-points" é 12.

A rotina trata destes pontos de entrada, inserindo-os na tabela de símbolos, e definindo-os como tal. Se o ponto de entrada constar da tabela, ele recebe um indicador de 8, o que indicará a sua definição. Se não constar, recebe um indicador 5, e estará indefinido.

Antes de se incluir o símbolo, ~~nie~~ testa-se o aparecimento de um indicador de programa absoluto, ou de um programa tipo ISS.

Se for permitida, a definição deste ponto de entrada, o procedimento é o seguinte:

- coloca-se o símbolo na tabela, define-se, e se for a primeiro entry point, recebe um outro indicador de valor Zero
- se não for o primeiro entry point, recebe este indicador com o valor do ponteiro para o entry point imediatamente antes definido.

 (use o verso, se necessário)

CÓDIGO DO PROGRAMA
WZENT

SEQUÊNCIA DE CHAMADA E DESCRIÇÃO DAS VARIÁVEIS - ARGUMENTO

CALL WZENT (KR, IENTR, NS)
KR - posição do entry point na tabela se símbolos.
IENTR - número de entry points definidos
NS - indicador de definição de ISS.
≠ 0 - ISS definida

SUB-ROTINAS USADAS

WZTRZ - traz o entry point testando sua validade como símbolo
WZPRC - busca se o simbolo, já consta da tabela de simbolos.

OBSERVAÇÕES OU MENSAGENS DE ERROS

ERRO 34 - Ocorreram mais de 12 pontos de entrada da subrotina
Programa foi definido como absoluto através de um comando ABS ou ORG.
Programa foi definido como subrotina ISS.
ERRO 29 - Símbolo foi definido , mas não como rótulo, ou através de EQU.

 (use o verso, se necessário)

CÓDIGO DE PROGRAMA

WZENT

VARIÁVEIS PRINCIPAIS

Nome	TIPO (Dimension, Common, Real, Integer, Data, etc) Tipo	Função da Variável

ANEXOS:

DIAGRAMA DE BLOCOS [ ]

LISTAGEM DO PROGRAMA [ ]

LISTAGEM DO EXEMPLO [ ]

GABARITO DOS CARTÕES DE ENTRADA OU DA IMPRESSORA [ ]

FOLHA DE ALTERAÇÃO VERSÃO 1 [ ]

VERSÃO = [ ]

```
// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
      SUBROUTINE WZENT(KR,IENTR,NS)
C
C     ESTA ROTINA TRATA DA PSEUDO ENTR
C          CODIGO DE ENTRY POINT = 5
C          ERRO 29 - ENTRADA DA TABELA JA DEFINIDA
C          ERRO 34 - MAIS DE 12 ENTRY-POINTS DE ROTINA,OU DEFINIDA ISS
C
      COMMON IPAG,KONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEQ,NOEQU(100),NOSIM(400,5),IENSI(400)
      COMMON IORIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ICOL(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
      COMMON IABS
      EQUIVALENCE (INIC(38),IBCO)
      IF(IABS-1) 5,70,5
    5 IF(NS)70,10,70
   10 L1=ISTEC(IAPTC+3)
      CALL WZTRZ(L1,L2,1,ISW1,M)
      IF(M) 15,25,15
   15 RETURN
   25 CALL WZPRC(2,K,ISTEP,L1,L2)
      IF(NOSIM(K,4))30,40,30
   30 IF(NOSIM(K,4)-2) 42,42,41
   41 CALL WZERR(29)
   35 RETURN
   42 NOSIM(K,4)=8
      GO TO 43
   40 NOSIM(K,4)=5
   43 NOSIM(K,5)=KR
      IENTR=IENTR+1
      KR=K
      IF(IENTR-12) 35,35,70
   70 CALL WZERR(34)
      RETURN
      END
```

IF(NOSIM(K,4)-4) 44, 42, 44

44

```
FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZENT
 COMMON   3132  VARIABLES      8  PROGRAM    134

RELATIVE ENTRY POINT ADDRESS IS 000F (HEX)

END OF COMPILATION

// DUP

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


*STORE      WS  UA  WZENT            0010
```

CART ID 0010 DB ADDR 046E DB CNT 0009

// EJECT

NOME DO PROGRAMA	CÓDIGO DE ATIVIDADE NUM. DO DOCUMENTO	CÓDIGO DO PROGRAMA
Rotina de tratamento de definição de constante		WZ@NT
	PROGRAMADOR	/ / DATA

LINGUAGEM USADA

FORTRAN [X] ASSEMBLER [ ] BASIC [ ] ALGOL [ ]....[ ]

TIPO DE PROGRAMA PRINCIPAL

DRIVER [ ] SUBROUTINE [X] FUNCTION [ ] OUTROS [ ]

IDENTIFICAÇÃO DOS DISCOS (FITAS)	SISTEMA SUPORTE (SIST. OPERACIONAL E HARDWARE)

CHAVES DO PAINEL OU SAIDAS ALTERNATIVAS

DESCRIÇÃO DO PROGRAMA

Este programa trata dos comandos tipo "DC"

DC /n - hexadecimal

DC n - decimal

DC *-* - zero

com ou sem rótulo

1 (use o verso, se necessário)

CÓDIGO DO PROGRAMA

W2@NT

SEQUÊNCIA DE CHAMADA E DESCRIÇÃO DAS VARIÁVEIS - ARGUMENTO

@ALL W2@NT (L1, L2, ISTEP)

L1 - posição inicial do operando

L2 - posição final do operando

ISTEP - contador de instruções

SUB-ROTINAS USADAS

W2@NT - converte o numero do operando a binário

W2PRC - traz o rotulo, testando a validade

W2ERR - trata erro se encontrado.

OBSERVAÇÕES [illegible] OU MENSAGENS DE ERROS

ERRO 31 - constante > 255 ou menor que "0"

(use o verso, se necessário)

CÓDIGO DE PROGRAMA

W2CNT

VARIÁVEIS PRINCIPAIS

TIPO
(Dimension, Common, Real, Integer, Data, etc)

Nome	Tipo	Função da Variável
IBAR	Data	caracter "/"
IESTR	Data	caracter "*"
ISW	I	chave p/ rotina de conversão: se = 0 indica hexadecimal; se = 1 ⇒ decimal
L1	Param.	primeira posição do rotulo
L2	Param	ultima posição do rotulo
M	I	≠0 indica erro em alguma rotina auxiliar
N	I	nº binario convertido
IAPTC	I	ponteiro primeira posição arquivo de informações suplementar
ISTEC(80)	DIM, I	arquivo de informação suplementar (buffer)
NOSIM	DIM	tabela de simbolos
MENOS	Data	caracter "-"
MAIS	Data	caracter "+"
ICOL	DIM, COM	buffer do comando de entrada
IBCO	Data	caracter "b"

ANEXOS: DIAGRAMA DE BLOCOS [ ]

LISTAGEM DO PROGRAMA [ ]

LISTAGEM DO EXEMPLO [ ]

GABARITO DOS CARTÕES DE ENTRADA OU DA IMPRESSORA [ ]

FOLHA DE ALTERAÇÃO VERSÃO 1 [ ]

VERSÃO = [ ]

```
// *

// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
      SUBROUTINE WZCNT(L1,L2,ISTEP)
C
C     ESTA SUBROUTINE TRATA DAS CONSTANTES DEFINIDAS
C     TIPOS DE CONSTANTES
C         10  /10
C         ERRO 31 - CONSTANTE MENOR QUE ZERO OU MAIOR QUE 255
C
      COMMON IPAG,KONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEQ,NOEQU(100),NOSIM(400,5),IENSI(400)
      COMMON IORIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ICOL(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
      EQUIVALENCE (IBCO,INIC(38)),(IBAR,INIC(47)),(IESTR,INIC(45)),
     1            (MAIS,INIC(39)),(MENOS,INIC(40))
C
      K=ISTEC(IAPTC+3)
      ISW=1
  255 IF(ICOL(K)-IBAR)260,270,260
  260 IF(ICOL(K)-IESTR)280,600,280
C
C     CONSTANTE HEXADECIMAL
C
  270 ISW=0
      GO TO 350
C
C     CONSTANTE DECIMAL
C
  280 K=K-1
  350 CALL WZCNV(K,N,K1,K2,ISW,M,38)
      IF(M)410 ,400,410
  400 IF(N-256)410,440,440
C
  440 CALL WZERR(31)
  410 ISTEC(IAPTC+5)=N
      ISTEC(IAPTC+4)=1
      IF(L1) 20,1000,20
   20 CALL WZPRC(1,IK,ISTEP,L1,L2)
      IF(NOSIM(IK,4))1000,21,1000
   21 NOSIM(IK,4)=1
 1000 RETURN
C
C
C
  600 IF(ICOL(K+1)-MENOS   )440,620,440
  620 I3=-1
  624 IF(ICOL(K+2)-IESTR   )440,650,440
  628 N=ISTEP+I3*N
      GO TO 400
C
  650 N=ISTEP
      IF(ICOL(K+3)-IBCO)440,628,440
      END
```

```
UNREFERENCED STATEMENTS
 255     624

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZCNT
 COMMON   3130  VARIABLES     10  PROGRAM    194

RELATIVE ENTRY POINT ADDRESS IS 000F (HEX)

END OF COMPILATION

// DUP

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


*STORE      WS  UA  WZCNT           0010
CART ID 0010   DB ADDR  0357   DB CNT   000D

// EJECT
```

NOME DO PROGRAMA	CÓDIGO DE ATIVIDADE NUM. DO DOCUMENTO	CÓDIGO DO PROGRAMA
Tratamento do comando EQUATE		WZEQU
	PROGRAMADOR	/ / DATA

LINGUAGEM USADA

FORTRAN [X] ASSEMBLER [ ] BASIC [ ] ALGOL [ ] .... [ ]

TIPO DE PROGRAMA

PRINCIPAL

DRIVER [ ] SUBROUTINE [X] FUNCTION [ ] OUTROS [ ]

IDENTIFICAÇÃO DOS DISCOS (FITAS)	SISTEMA SUPORTE (SIST. OPERACIONAL E HARDWARE)

CHAVES DO PAINEL OU SAIDAS ALTERNATIVAS

DESCRIÇÃO DO PROGRAMA

Esta rotina faz o tatamento de comando do tipo:

```
SIMB   EQU   SIMBX
SIMB   EQU   SIMBX ± 1
SIMB   EQU   /nnn
SIMB   EQU   nnn
SIMB   EQU   * ± 1
```

O tratamento do primeiro tipo de comando é feito inserindo-se ambos os símbolos na tabela de símbolos em equivalência. Os símbolos que ainda não constarem da tabela de símbolos são nela inseridos. Se um dos símbolos já estiver definido , o outro símbolo é também definido, e endereçado.

No caso do segundo tipo de comando, SIMBX deve já ter sido definido ou então ter-se-á erro. SIMB é definido e endereçado.

Nos demais casos, SIMB é endereçado e definido. O endereço é calculado pela expressão no segundo membro do comando . A "/" significa um endereço em hexadecimal. O "*" é interpretado como o valor corrente do contador de instruções.

 (use o verso, se necessário)

CÓDIGO DO PROGRAMA

WZEQU

SEQUÊNCIA DE CHAMADA E DESCRIÇÃO DAS VARIÁVEIS - ARGUMENTO

CALL WZEQU (L1,L2,ISTEP)

L1 : posição inicial do símbolo SIMB

L2 : posição final do símbolo SIMB

ISTEP : valor do contador de instruções

SUB-ROTINAS USADAS

WZPRC - busca e insere símbolo na tabela de símbolos

WZTRZ - determina extensão do símbolo.

WZCNV - converte expressões numéricas a binário.

OBSERVAÇÕES [illegible] OU MENSAGENS DE ERROS

2 (use o verso, se necessário)

	CÓDIGO DE PROGRAMA
	WZEQU

VARIÁVEIS PRINCIPAIS

TIPO
(Dimension, Common, Real, Integer, Data, etc)

Nome	Tipo	Função da Variável
ICOL(80)	I,C	matriz que contém o cartão perfurado
IBAR	I,C	caracter "/"
IESTR	I,C	caracter "*"
IBCO	I,C	caracter branco
MAAIS	I,C	caracter "+"
MENOS	I,C	caracter "-"
NOEQU	I,C	matriz de equivalências
IAPEQ	I,C	ponteiro para a última posição da matriz de equivalências
NOSIM	I,C	matriz tabela de símbolos
IENSI	I,C	endereços dos símbolos
M	I	indicador de erro. Ocorrência de erros se M é diferente de zero.

ANEXOS: DIAGRAMA DE BLOCOS [ ]
LISTAGEM DO PROGRAMA [ ]
LISTAGEM DO EXEMPLO [ ]
GABARITO DOS CARTÕES DE ENTRADA OU DA IMPRESSORA [ ]
FOLHA DE ALTERAÇÃO VERSÃO 1 [ ]
VERSÃO = [ ]

```
// *

// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
      SUBROUTINE WZEQU(L1,L2,ISTEP)
C
C             ESTA SUBROTINA TRATA DAS EQUIVALENCIAS
C     L1 - INICIO DO CAMPO DE ROTULO
C     L2 - FIM DO CAMPO DE ROTULO
C     ISTEP - ENDERECO DA INSTRUCAO SENDO EXECUTADA
C
C
C     COMANDOS VALIDOS
C         A  EQU B    * TIPO 1
C         C  EQU 10   * TIPO 2
C         C  EQU /10   * TIPO 3
C         C  EQU LABEL+1  - LABEL JA DEFINIDO  *  TIPO 4
C         C  EQU *+1   * TIPO 5
C         C  EQU *-1   * TIPO 6
C         ERRO28 - COMANDO EQUATE INVALIDO
      COMMON IPAG,KONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEQ,NOEQU(100),NOSIM(400,5),IENSI(400)
      COMMON IORIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ICOL(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
      EQUIVALENCE (IBCO,INIC(38)),(IBAR,INIC(47)),(IESTR,INIC(45)),
     1            (MAIS,INIC(39)),(MENOS,INIC(40))
C
      K = ISTEC(IAPTC+3)
      NJ=2
      CALL WZPRC(2,J,ISTEP,L1,L2)
C
  200 IF(ICOL(K)-IBAR     ) 210,510,210
  210 IF(ICOL(K)-IESTR    )220,600,220
C
C     VERIFICA SE COMANDO E TIPO 2
C
  220 DO 230 JC=28,37
      IF(ICOL(K)-INIC(JC))230,550,230
  230 CONTINUE
C
C     COMANDO E TIPO 1 OU TIPO 4
C
      CALL WZTRZ(K,IFIM,2,ISW1,M)
      IF(M)150,300,150
C
  300 IF(ISW1-38)400,310,400
C
C     COMANDO E TIPO 1
C
  310 CALL WZPRC(2,KV,ISTEP,K,IFIM)
      NOEQJ(IAPEQ+1)=J
      NOEQJ(IAPEQ+2)=KV
      IAPEQ=IAPEQ+2
      RETURN
C     COMANDO E TIPO 4
```

```
C
C
  400 IF(ISW1-39) 410,420,410
  410 IF(ISW1-40) 100,430,100
  420 IX=+1
      GO TO 440
  430 IX=-1
  440 CALL WZPRC(2,KV,ISTEP,K,IFIM)
  450 NX=NOSIM(KV,4)+1
      GO TO(500,460,460,460,460,500,500,460,460,460,460),NX
C
  460 CALL WZCNV(IFIM+1,N,K1,K2,ISW1,M,38)
      IF(M) 490,470,490
  470 MK = IENSI(KV)+IX*N
      IF(MK)100,480,480
  480 IF(MK-4095) 481,481,100
  481 IENSI(J) = MK
      NX=NOSIM(J,4)+1
      GO TO (482,483,483,483,483,484,484,483,483,483,483),NX
  482 GO TO (486,487),NJ
  486 NOSIM(J,4)=2
      RETURN
  487 NOSIM(J,4)=1
      RETURN
  483 NOSIM(J,4)=10
      RETURN
  484 GO TO (500,485),NJ
  485 NOSIM(J,4)=NOSIM(J,4)+3
  490 RETURN
C
  500 CALL WZERR(28)
      RETURN
C
C     COMANDO TIPO 2
C
  510 ISW1=0
  520 CALL WZCNV(K,MK,K1,K2,ISW1,M,38)
      NJ=1
      IF(M) 490,480,490
C
C     COMANDO E TIPO 3
C
  550 ISW1=1
      K=K-1
      GO TO 520
C
C     COMANDO E TIPO 5 OU 6
C
  600 IF(ICOL(K+1)-IBCD) 601,650,601
  601 IF(ICOL (K+1)-MAIS) 610,630,610
  610 IF(ICOL(K+1)-MENOS   ) 500,640,500
  620 KV=IAPSI+2
      IFIM=K
      IENSI(KV)=ISTEP
      GO TO 460
C
  630 IX=1
```

```
      GO TO 620
  640 IX=-1
      GO TO 620
  650 IENSI(J)=ISTEP
      GO TO 487
  100 CALL WZERR(16)
  150 RETURN
      END

UNREFERENCED STATEMENTS
 200     450

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZEQU
 COMMON   3130  VARIABLES     18  PROGRAM    470

RELATIVE ENTRY POINT ADDRESS IS 001E (HEX)

END OF COMPILATION

// DUP

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


*STORE      WS  UA  WZEQU           0010
CART ID 0010   DB ADDR  03BE   DB CNT   001D

// EJECT
```

NOME DO PROGRAMA ROTINA DE TRATAMENTO DA PSEU PSEUDO CALL	CÓDIGO DE ATIVIDADE NUM. DO DOCUMENTO	CÓDIGO DO PROGRAMA WZCOD
	PROGRAMADOR	/ / DATA

LINGUAGEM USADA

FORTRAN [X] ASSEMBLER [ ] BASIC [ ] ALGOL [ ] .... [ ]

TIPO DE PROGRAMA

DRIVER [ ] SUBROUTINE [X] PRINCIPAL FUNCTION [ ] OUTROS [ ]

IDENTIFICAÇÃO DOS DISCOS (FITAS)	SISTEMA SUPORTE (SIST. OPERACIONAL E HARDWARE)
0010	

CHAVES DO PAINEL OU SAIDAS ALTERNATIVAS
NENHUMA

DESCRIÇÃO DO PROGRAMA

Esta rotina trata a pseudo CALL, na segunda fase. Na primeira fase, esta pseudo praticamente não sofre nenhum tratamento especial.

O operando da pseudo CALL, que deve ser um nome que não consta da tabela de símbolos, é procurado; determinada sua validade, ele é convertido em name code, e colocado em duas ~~xxxxxxx~~ variáveis, que servem como parâmetros, sendo assim transmitidos ao programa principal.

1 (use o verso, se necessário)

CÓDIGO DO PROGRAMA

WZCOD

SEQUÊNCIA DE CHAMADA E DESCRIÇÃO DAS VARIÁVEIS - ARGUMENTO

CALL WZCOD(NL1, NL2)

NL1 - primeira palavra do name code

NL2 - segunda palavra do name code

SUB-ROTINAS USADAS

WZTRA - busca o nome testando sua validade

WZCDM - transforma o nome em "name code."

OBSERVAÇÕES [illegible] OU MENSAGENS DE ERROS

NENHUMA

 (use o verso, se necessário)

CÓDIGO DE PROGRAMA

WZCOD

VARIÁVEIS PRINCIPAIS

Nome	Tipo (Dimension, Common, Real, Integer, Data, etc)	Função da Variável
ISTEC	I,C	Buffer de informação suplementar
NAME	I,()	Matriz auxiliar, que deve reter o nome da rotina sendo chamada
M	I	indica quando diferente de zero, que ocorreu erro
K	I	posição inicial do operando
IFIM	I	posição final do operando
IBCO	I	caracter branco
ICOL	I,C	buffer de entrada para os comandos fonte

ANEXOS: DIAGRAMA DE BLOCOS [ ]

LISTAGEM DO PROGRAMA [ ]

LISTAGEM DO EXEMPLO [ ]

GABARITO DOS CARTÕES DE ENTRADA OU DA IMPRESSORA [ ]

FOLHA DE ALTERAÇÃO VERSÃO 1 [ ]

VERSÃO = [ ]

```
// *

// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
      SUBROUTINE WZCOD(N1,N2)
C
C     ESTA ROTINA TRATA DO COMANDO 'CALL'
C
C     N1 - PRIMEIRA PALAVRA DO COMANDO TRADUZIDO
C     N2 - SEGUNDA PALAVRA DO COMANDO TRADUZIDO
C
      DIMENSION  ICOL(80)
      DIMENSION NAME(5)
      COMMON IPAG,KONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEQ,NOEQU(100),NOSIM(400,5),IENSI(400)
      COMMON IORIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ISAI(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
      EQUIVALENCE (ICOL(1),IERRA(1))
      EQUIVALENCE (IBCO,INIC(38))
C
C
      K=ISTEC(IAPTC+3)
      CALL WZTRA(K,IFIM,1,ISW,M)
      IF(M)350,300,350
C
  300 DO 310 I=1,5
  310 NAME(I)=IBCO
C
      IFIM=IFIM-K+1
      DO 320 I=1,IFIM
      NAME(I)=ICOL(K)
      K=K+1
  320 CONTINUE
      CALL WZCOM(NAME,N1,N2)
C
  350 RETURN
      END

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZCOD
 COMMON   3130  VARIABLES     12  PROGRAM    102

RELATIVE ENTRY POINT ADDRESS IS 000E (HEX)

END OF COMPILATION

// DUP

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


*STORE      WS  UA  WZCOD           0010
```

CART ID 0010 DB ADDR 02B7 DB CNT 0007

// EJECT

NOME DO PROGRAMA ROTINA DE TRATAMENTO DA PSEUDO ISS	CÓDIGO DE ATIVIDADE NUM. DO DOCUMENTO	CÓDIGO DO PROGRAMA WZISS
	PROGRAMADOR	/ / DATA

LINGUAGEM USADA

FORTRAN [x] ASSEMBLER [ ] BASIC [ ] ALGOL [ ] .... [ ]

TIPO DE PROGRAMA PRINCIPAL

DRIVER [ ] SUBROUTINE [x] FUNCTION [ ] OUTROS [ ]

IDENTIFICAÇÃO DOS DISCOS (FITAS)	SISTEMA SUPORTE (SIST. OPERACIONAL E HARDWARE)
0010	

CHAVES DO PAINEL OU SAIDAS ALTERNATIVAS
NENHUMA

DESCRIÇÃO DO PROGRAMA

Esta rotina, determina a validade de um comando ISS, isto é definição de uma rotina de entrada e saída.

Esta pseudo quando aparecer deve ser a primeira do "deck" fonte, excluindo pseudos de controle, e cartões de comentários.

O comando tem a seguinte sintaxe:

ISS NN,NOME , onde NN é um número decimal, e nome, é o nome do "entry point".

Alguns testes de erros são realizados:

- verificação se o comando é o primeiro do "deck" fonte, salvo as exceções apontadas.
- verificação se o dispositivo de entrada e saída é permitido
- verificação de sintaxe correta do programa.

 (use o verso, se necessário)

CÓDIGO DO PROGRAMA
WZISS

SEQUÊNCIA DE CHAMADA E DESCRIÇÃO DAS VARIÁVEIS - ARGUMENTO

CALL WZISS(IKCRD)

IKCRD - indica a ordem de entrada do cartão. Não são contados cartões de comentários e de pseudos de controle de listagem.

SUB-ROTINAS USADAS

WZERR - Tratamento de erros
WZCNV - conversão do número decimal do comando
WZTRZ - testa a validade do nome
WZPRC - insere o nome na tabela de símbolos.

OBSERVAÇÕES [illegible] OU MENSAGENS DE ERROS

Podem ocorrer os seguintes erros.
- ERRO 36 - a pseudo não é a primeira do programa
- ERRO 32 - dispositivo de entrada e saída inválido
- ERRO 37 - sintaxe inválida do comando.

 (use o verso, se necessário)

	CÓDIGO DE PROGRAMA
	WZISS

VARIÁVEIS PRINCIPAIS

Nome	TIPO (Dimension, Common, Real, Integer, Data, etc) Tipo	Função da Variável
ISTEC	I,C	Buffer do arquivo de inf. suplem.
IAPTC	I,C	Ponteiro do buffer de inf. suplem.
K1,K2	I,I	Respectivamente posições iniciais e finais do número.
N	I	Valor convertido do número
ICOL	I,C,( )	Buffer de comandos fonte
M	I	Se diferente de zero, indica a ocorrência de erros
IVIRG	I	Caracter vírgula
IBCO	I	Caracter branco
ISTEP	I	Contador de instruções- não tem significado
NOSIM	I,C	Tabela de símbolos
NSTAT	I,C	Indica quando igual a um a ocorrência de ISS
ICNT	I,C	Indica a posição na tabela de símbolos do nome do ponto de entrada.

ANEXOS: DIAGRAMA DE BLOCOS [ ]

LISTAGEM DO PROGRAMA [ ]

LISTAGEM DO EXEMPLO [ ]

GABARITO DOS CARTÕES DE ENTRADA OU DA IMPRESSORA [ ]

FOLHA DE ALTERAÇÃO VERSÃO 1 [ ]

VERSÃO = [ ]

```
// *

// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
      SUBROUTINE WZISS(IKCRD)
C
C     ESTA ROTINA TRATA DOS COMANDOS ISS
C         ERRO36- COMANDO 'ISS' NAO E O PRIMEIRO COMANDO DO PROGRAMA
C         ERRO39- NUMERO DE DISP. INVALIDO
C         ERRO37- SINTAXE INVALIDA EM PSEUDO 'ISS'
C     IKCRD - CONTADOR DE CARTOES
C
      COMMON IPAG,KONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEQ,NOEQU(100),NOSIM(400,5),IENSI(400)
      COMMON IORIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ICOL(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
      EQUIVALENCE(IAC,NSTAT),(IVIRG,INIC(42)),(IBCO,INIC(38))
C
C
      I=ISTEC(IAPTC+3)
      IF(IKCRD-1)4,6,4
    4 CALL WZERR(36)
      RETURN
C
    6 CALL WZCNV(I-1,N,K1,K2,1,M,42)
      IF(M)40,5,40
    5 IF(N-16) 20,10,10
   10 CALL WZERR(32)
      RETURN
   20 K =K2+1
C
      IF(ICOL(K)- IVIRG   )35,90,35
   35 CALL WZERR(37)
   40 RETURN
C
   90 K = K+ 1
      IF(ICOL(K)-IBCO    )92,35 ,92
   92 CALL WZTRZ(K,IFIM,1,ISW1,M)
      IF(M)40,120,40
  120 CALL WZPRC(2,IK,ISTEP,K,IFIM)
      NOSIM(IK,4)=6
      NOSIM(IK,5)=N
      NSTAT=1
      ICNT=IK
C
      RETURN
C
      END

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZISS
 COMMON   3130  VARIABLES     14  PROGRAM    148
```

```
// *

// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
      SUBROUTINE WZISS(IKCRD)
C
C     ESTA ROTINA TRATA DOS COMANDOS ISS
C         ERR036- COMANDO 'ISS' NAO E O PRIMEIRO COMANDO DO PROGRAMA
C         ERR039- NUMERO DE DISP. INVALIDO
C         ERR037- SINTAXE INVALIDA EM PSEUDO 'ISS'
C     IKCRD - CONTADOR DE CARTOES
C
      COMMON IPAG,KONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEQ,NOEQU(100),NOSIM(400,5),IENSI(400)
      COMMON IORIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ICOL(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
      EQUIVALENCE(IAC,NSTAT),(IVIRG,INIC(42)),(IBCD,INIC(38))
C
C
      I=ISTEC(IAPTC+3)
      IF(IKCRD-1)4,6,4
    4 CALL WZERR(36)
      RETURN
C
    6 CALL WZCNV(I-1,N,K1,K2,1,M,42)
      IF(M)40,5,40
    5 IF(N-16) 20,10,10
   10 CALL WZERR(32)
      RETURN
   20 K =K2+1
C
      IF(ICOL(K)- IVIRG   )35,90,35
   35 CALL WZERR(37)
   40 RETURN
C
   90 K = K+ 1
      IF(ICOL(K)-IBCD    )92,35 ,92
   92 CALL WZTRZ(K,IFIM,1,ISW1,M)
      IF(M)40,120,40
  120 CALL WZPRC(2,IK,ISTEP,K,IFIM)
      NOSIM(IK,4)=6
      NOSIM(IK,5)=N
      NSTAT=1
      ICNT=IK
C
      RETURN
C
      END

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZISS
 COMMON   3130  VARIABLES     14  PROGRAM    148
```

```
RELATIVE ENTRY POINT ADDRESS IS 0016 (HEX)

END OF COMPILATION

// DUP

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


*STORE      WS  JA  WZISS            0010
CART ID 0010   DB ADDR  034D   DB CNT   000A

// EJECT
```

ROTINA DE TRATAMENTO DE BUFFERS

Destinam-se estas rotinas ao tratamento e gerenciamento dos buffers de gravação e leitura de disco.

WZWPT

WZINS

WZFIM

WZTRT

WZDTB

NOME DO PROGRAMA	CÓDIGO DE ATIVIDADE NUM. DO DOCUMENTO	CÓDIGO DO PROGRAMA
PROGRAMA DE GERENCIAMENTO DO BUFFER DE ERROS NA PRIMEIRA FASE		WZWPT
	PROGRAMADOR	/ / DATA

LINGUAGEM USADA

FORTRAN [x] ASSEMBLER [ ] BASIC [ ] ALGOL [ ]....[ ]

TIPO DE PROGRAMA

PRINCIPAL

DRIVER [ ] SUBROUTINE [x] FUNCTION [ ] OUTROS [ ]

IDENTIFICAÇÃO DOS DISCOS (FITAS)	SISTEMA SUPORTE (SIST. OPERACIONAL E HARDWARE)
0010	

CHAVES DO PAINEL OU SAIDAS ALTERNATIVAS

NENHUMA

DESCRIÇÃO DO PROGRAMA

Esta rotina coloca no buffer de erros, uma marca indicando fim de cadeia de erros, ou então simplesmente a marca de não erro. A rotina trata o fim de buffer, e reinicializa novo buffer, gravando em disco o velho.

 (use o verso, se necessário)

CÓDIGO DO PROGRAMA

WZWPT

SEQUÊNCIA DE CHAMADA E DESCRIÇÃO DAS VARIÁVEIS - ARGUMENTO

CALL WZWPT(IPAUT)

IPAUT - caracter a ser inserido no buffer; pode também ser o número de um erro, isto é, um elemento da cadeia.

SUB-ROTINAS USADAS

NENHUMA

OBSERVAÇÕES [illegible] OU MENSAGENS DE ERROS

NENHUMA

 (use o verso, se necessário)

CÓDIGO DE PROGRAMA

WZWPT

VARIÁVEIS PRINCIPAIS

TIPO
(Dimension, Common, Real, Integer, Data, etc)

Nome	Tipo	Função da Variável
IAPER	I,C	Ponteiro para o buffer de erros
IERRA	I,C,()	Buffer de erros
ID7	número	do dispositivo lógico correspondente ao arquivo de erros
N7	I,C	variável associada ao dispositivo lógico ID7, correspondente ao ponteiro para o próximo registro

ANEXOS: DIAGRAMA DE BLOCOS [ ]

LISTAGEM DO PROGRAMA [ ]

LISTAGEM DO EXEMPLO [ ]

GABARITO DOS CARTÕES DE ENTRADA OU DA IMPRESSORA [ ]

FOLHA DE ALTERAÇÃO VERSÃO 1 [ ]

VERSÃO = [ ]

```
// *

// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
COMPILATION DISCONTINUED

M 14 SYSTEM PROGRAM DETECTED MONITOR CONTROL RECORD

// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
        SUBROUTINE WZWPT(IPAUT)
      COMMON IPAG,KONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEQ,NOEQU(100),NOSIM(400,5),IENSI(400)
      COMMON IORIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ICOL(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
C
C     VERIFICA SE O VETOR ERROS ESTA COMPLETO
C
      IF(IAPER-80)4201,4201,4202
 4201 IERRA(IAPER)=IPAUT
      IAPER=IAPER+1
      RETURN
 4202 IERRA(1)=IAPER-1
      WRITE(ID7'N7) IERRA
      IAPER=2
      GO TO 4201
      END

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZWPT
 COMMON   3130  VARIABLES      2  PROGRAM     52

RELATIVE ENTRY POINT ADDRESS IS 0005 (HEX)

END OF COMPILATION

// DUP

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


*STORE      WS  UA  WZWPT            0010
CART ID 0010   DB ADDR  180C   DB CNT   0005
```

NOME DO PROGRAMA	CÓDIGO DE ATIVIDADE	CÓDIGO DO PROGRAMA
WZINS- ROTINA QUE INSERE UMA PALAVRA JUNTAMENTE COM SEU INDICADOR DE RELOCAÇÂO NO BUFFER DE DISCO	NUM. DO DOCUMENTO NO	WZINS
	PROGRAMADOR	/ / DATA

LINGUAGEM USADA

FORTRAN [X] ASSEMBLER [ ] BASIC [ ] ALGOL [ ]....[ ]

TIPO DE PROGRAMA

DRIVER [ ] SUBROUTINE [X] PRINCIPAL FUNCTION [ ] OUTROS [ ]

IDENTIFICAÇÃO DOS DISCOS (FITAS)	SISTEMA SUPORTE (SIST. OPERACIONAL E HARDWARE)
0010	

CHAVES DO PAINEL OU SAIDAS ALTERNATIVAS

NENHUMA

DESCRIÇÃO DO PROGRAMA

Esta rotina, coloca no buffer de saidan uma palavra de código, juntamente com seu indicador de relocação.

O ponteiro do buffer de saída de disco aponta sempre a última posição preenchida. Os contadores de palavras no bloco de dados ~~xx~~ e de palavras restantes no buffer são respectivamente incrementados e decrementados.

 (use o verso, se necessário)

CÓDIGO DO PROGRAMA

WZINS

SEQUÊNCIA DE CHAMADA E DESCRIÇÃO DAS VARIÁVEIS - ARGUMENTO

CALL WZINS(IOCT)

IOCT - palavra a ser inserida

SUB-ROTINAS USADAS

NENHUMA

OBSERVAÇÕES [illegible] OU MENSAGENS DE ERROS

NENHUMA

 (use o verso, se necessário)

CÓDIGO DE PROGRAMA

WZINS

VARIÁVEIS PRINCIPAIS

TIPO
(Dimension, Common, Real, Integer, Data, etc)

Nome	Tipo	Função da Variável
IAPIN	I,C	ponteiro para o buffer de saída de disco
IKONT	I,C	contador de palavras no bloco de dados
ICOMP	I,C	palavra que indica o número de posições ainda vagas no buffer de disco
ISAI	I,C,()	buffer de saída em disco
IND	I,C	indicador derelocação
IRET	I,C	palavra onde vão sendo montados os indicadores de relocação

ANEXOS: DIAGRAMA DE BLOCOS [ ]
LISTAGEM DO PROGRAMA [ ]
LISTAGEM DO EXEMPLO [ ]
GABARITO DOS CARTÕES DE ENTRADA OU DA IMPRESSORA [ ]
FOLHA DE ALTERAÇÃO VERSÃO 1 [ ]
VERSÃO = [ ]

```
// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
      SUBROUTINE WZINS(IOCT)
      COMMON IPAG,KONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEQ,NOEQU(100),NOSIM(400,5),IENSI(400)
      COMMON IORIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ISAI(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
      EQUIVALENCE (IORIG,IKONT)
      IAPIN=IAPIN+1
      IKONT=IKONT+1
      ICOMP=ICOMP-1
      ISAI (IAPIN)=IOCT
      IAC=IAC+1
   30  IRET=IRET*4+IND
      RETURN
      END

UNREFERENCED STATEMENTS
 30

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZINS
 COMMON   3130  VARIABLES      2  PROGRAM     52

RELATIVE ENTRY POINT ADDRESS IS 0004 (HEX)

END OF COMPILATION

// DUP

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


*STORE      WS  UA  WZINS           0010
CART ID 0010   DB ADDR  046A   DB CNT   0004

// EJECT
```

NOME DO PROGRAMA	CÓDIGO DE ATIVIDADE NUM. DO DOCUMENTO	CÓDIGO DO PROGRAMA
ROTINA DE FINALIZAÇÃO DE UM BLOCO DE DADOS		WZFIM
	PROGRAMADOR	/ / DATA

LINGUAGEM USADA

FORTRAN [x] ASSEMBLER [ ] BASIC [ ] ALGOL [ ] .... [ ]

TIPO DE PROGRAMA

PRINCIPAL

DRIVER [ ] SUBROUTINE [x] FUNCTION [ ] OUTROS [ ]

IDENTIFICAÇÃO DOS DISCOS (FITAS)	SISTEMA SUPORTE (SIST. OPERACIONAL E HARDWARE)
0010	

CHAVES DO PAINEL OU SAIDAS ALTERNATIVAS

NENHUMA

DESCRIÇÃO DO PROGRAMA

Esta rotina finaliza um bloco de dados. Esta finalização consiste no término de preenchimento do buffer de dados, e iniciação de uma série de indicadores.

Cuidados especiais são tomados no caso de ocorrer fim de ~~xxxx~~ buffer. Neste caso testa-se a possibilidade de se iniciar um outro bloco neste mesmo buffer ou a necessidade de se transferir este buffer para disco, iniciando-se um novo buffer.

 (use o verso, se necessário)

CÓDIGO DO PROGRAMA

WZFIM

SEQUÊNCIA DE CHAMADA E DESCRIÇÃO DAS VARIÁVEIS - ARGUMENTO

CALL WZFIM(ISTEP,K)

ISTEP : contador de instrução

K : indica quando igual a zero que um novo buffer foi iniciado

SUB-ROTINAS USADAS

WZDTB - faz a montagem final do cabeçalho de dados e inicia alguns ponteiros.

OBSERVAÇÕES [illegible] OU MENSAGENS DE ERROS

NENHUMA

 (use o verso, se necessário)

CÓDIGO DE PROGRAMA

WZFIM

VARIÁVEIS PRINCIPAIS

TIPO
(Dimension, Common, Real, Integer, Data, etc)

Nome	Tipo	Função da Variável
ICOMP	I,C	indica quantas palavras disponíveis ainda existem ~~dixp~~ no buffer de saída
IKONT	I,C	contador de palavras no bloco de dados
IAPIN	I,C	ponteiro do buffer de saída
ISAI	I,C	buffer de saída
IRET	I,C	palavra onde são montados os contadores de relocação

ANEXOS: DIAGRAMA DE BLOCOS [ ]
LISTAGEM DO PROGRAMA [ ]
LISTAGEM DO EXEMPLO [ ]
GABARITO DOS CARTÕES DE ENTRADA OU DA IMPRESSORA [ ]
FOLHA DE ALTERAÇÃO VERSÃO 1 [ ]
VERSÃO = [ ]

```
// *

// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
      SUBROUTINE WZFIM (ISTEP,K)
      DIMENSION ICOL(80)
      COMMON IPAG, KONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEQ,NOEQU(100),NOSIM(400,5),IENSI(400)
      COMMON IORIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ISAI(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
      EQUIVALENCE (ICOL(1),IERRA(1))
      EQUIVALENCE (IDISK,ID7),(NR,N7)
      EQUIVALENCE (IKONT,IORIG)
      ICOMP=ICOMP-2
      IF(IKONT)20,5,20
    5 IF(ICOMP)1,1,2
    1 ICOMP=ICOMP+2
      IAC=IAC+ICOMP
      IAPIN=IAPIN+ICOMP
      CALL WZDTB(ISTEP)
      WRITE(IDISK'NR) ISAI
      IAPIN=2
      ICOMP=318
      K=1
      RETURN
    2 CALL WZDTB(ISTEP)
      IAPIN=IAPIN+2
      K=0
      RETURN
   20 INK=IAPIN-IKONT+1
      ISAI (INK)=IRET*2**(18-2*IKONT)
      IKONT=0
      GO TO 5
      END

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZFIM
 COMMON   3130  VARIABLES      4  PROGRAM    124

RELATIVE ENTRY POINT ADDRESS IS 0009 (HEX)

END OF COMPILATION

// DUP

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


*STORE      WS  UA  WZFIM            0010
CART ID 0010   DB ADDR  0364   DB CNT   0009

// EJECT
```

NOME DO PROGRAMA	CÓDIGO DE ATIVIDADE NUM. DO DOCUMENTO	CÓDIGO DO PROGRAMA
GERENCIAMENTO DO BUFFER DE SAIDA		WZTRT
	PROGRAMADOR	/ / DATA

LINGUAGEM USADA

FORTRAN [X] ASSEMBLER [ ] BASIC [ ] ALGOL [ ] .... [ ]

TIPO DE PROGRAMA

DRIVER [ ] SUBROUTINE [X] PRINCIPAL FUNCTION [ ] OUTROS [ ]

IDENTIFICAÇÃO DOS DISCOS (FITAS)	SISTEMA SUPORTE (SIST. OPERACIONAL E HARDWARE)
0010	

CHAVES DO PAINEL OU SAIDAS ALTERNATIVAS

NENHUMA

DESCRIÇÃO DO PROGRAMA

Esta rotina faz o gerenciamento das saídas de buffers de disco do programa. A rotina deve determinar quando mais de uma palavra deve ser colocada no buffer de saída de disco (caso das subrotinas). A rotina também deve determinar a ocasião em que um bloco de dados deve ser terminado, devido a fim de buffer de saída, providenciando então, a sua gravação em disco, e a iniciação de um novo bloco.

 (use o verso, se necessário)

CÓDIGO DO PROGRAMA

WZTRT

SEQUÊNCIA DE CHAMADA E DESCRIÇÃO DAS VARIÁVEIS - ARGUMENTO

CALL WZTRT(IOCT,N2?ISTEP,IK)

IOCT - palavra a ser colocada no buffer

N2 - eventualmente uma segunda palavra a ser colocada no buffer

ISTEP - contador de instruções

IK - indica quando igual a 2, que a instrução é relocável e de endereçamento da memória, devendo então receber tratamento especial.

SUB-ROTINAS USADAS

WZINS - insere uma palavra no buffer de saída

WZDTB - trata a necessidade de finalização de um bloco de dados

OBSERVAÇÕES [illegible] OU MENSAGENS DE ERROS

NENHUMA

 (use o verso, se necessário)

	CÓDIGO DE PROGRAMA
	WZTRT

VARIÁVEIS PRINCIPAIS

TIPO
(Dimension, Common, Real, Integer, Data, etc)

Nome	Tipo	Função da Variável
IND	I,C	indicador de relocação
IKONT	I,C	contador de palavras no grupo
IABLE	I,C,()	buffer de saída- equivale a ISAI
IAPIN	I,C	ponteiro para o buffer de saída
ICOMP	I,C	contador de palavras disponíveis no buffer de saída
IAC	I,C	contador de palavras no buffer de saída
IRET	I,C	palavra onde vão sendo montados os indicadores de relocação

ANEXOS: DIAGRAMA DE BLOCOS [ ]
LISTAGEM DO PROGRAMA [ ]
LISTAGEM DO EXEMPLO [ ]
GABARITO DOS CARTÕES DE ENTRADA OU DA IMPRESSORA [ ]
FOLHA DE ALTERAÇÃO VERSÃO 1 [ ]
VERSÃO = [ ]

```
RELATIVE ENTRY POINT ADDRESS IS 000D (HEX)

END OF COMPILATION

// DUP

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


*STORE      WS  JA  WZIME           0010
CART ID 0010   DB ADDR  02F1   DB CNT   0007

// EJECT
```

NOME DO PROGRAMA	CÓDIGO DE ATIVIDADE	CÓDIGO DO PROGRAMA
SUBROTINA DE TRATAMENTO DAS INSTRUÇÕES DE REFERENCI A MEMORIA (2ª fase)	NUM. DO DOCUMENTO	WZADR
	PROGRAMADOR	/ / DATA

LINGUAGEM USADA

FORTRAN [x] ASSEMBLER [ ] BASIC [ ] ALGOL [ ] .... [ ]

TIPO DE PROGRAMA

DRIVER [ ] SUBROUTINE [x] PRINCIPAL FUNCTION [ ] OUTROS [ ]

IDENTIFICAÇÃO DOS DISCOS (FITAS)	SISTEMA SUPORTE (SIST. OPERACIONAL E HARDWARE)
0010	

CHAVES DO PAINEL OU SAIDAS ALTERNATIVAS

NENHUMA

DESCRIÇÃO DO PROGRAMA

Esta rotina trata o operando das instruções de referência a memória na segunda fase do programa. Nestas condições determinou-se já o tipo do operando: simbólico, absoluto, relativo, indireto, estando estas informações contidas no buffer de informação suplementar. O tratamento das instruções nesta fase consiste em determinar-se o endereço correto da instruções, e sua colocação na ~~buffer~~ palavra, onde está sendo montada a instrução. Este cálculo, pode envolver ~~as~~ buscas na tabela de símbolo, conversões numéricas e avaliação de expressões compreendendo símbolos e números. No caso de endereços indiretos, ocorre a geração de mais uma instrução NOP antes da instrução sendo gerada afim de provir a montagem correta deste tipo de endereçamento. A rotina provê também a listagem desta instrução, quando o programa está produzindo listagem.

 (use o verso, se necessário)

CÓDIGO DO PROGRAMA

WZADR

SEQUÊNCIA DE CHAMADA E DESCRIÇÃO DAS VARIÁVEIS - ARGUMENTO

CALL WZADR(IOCT,ISTEP,IABS)

IOCT - palavra onde está sendo montada a instrução

IABS - indica se o programa é absoluto ou relativo

ISTEP - contador de instruções,

SUB-ROTINAS USADAS

WZDEC - converte operandos decimais e hexadecimais

WZIER - tratamento de erros

WZTRT - coloca no buffer a palavra de código

WZITR - converte um numero binário em 4 caracteres hexadecimais

OBSERVAÇÕES [illegible] OU MENSAGENS DE ERROS

Podem ocorrer os seguintes erros:

ERRO 15 - operando não é seguido de branco, ou sua sintaxe está errada.

ERRO 33 - endereçamento fora dos limites da memória

(use o verso, se necessário)

	CÓDIGO DE PROGRAMA
	WZADR

VARIÁVEIS PRINCIPAIS

TIPO
(Dimension, Common,Real,Integer,Data,etc)

Nome	Tipo	Função da Variável
jkjjk	I	código de instrução NOP
IOPTG	I,C	indica próxima posição do operando a ser analisad
IOPTI	I,C	tipo de operando
ISTEC	I,C	buffer de informação suplementar
IAPTC	I,C	ponteiro do buffer de informação suplementar
IND	I,C	indicador de relocação
IEND	I	endereço sendo montado
NOSIM	I,C	tabela de símbolos
IENSI	I,C	endereço dos símbolos na tabela
N	I	endereço numérico convertido
M	I	indica se diferente de zero que ocorreu erro
IESTR	I	caracter asterístico
IBCO	I	caracter branco

ANEXOS: DIAGRAMA DE BLOCOS [ ]
LISTAGEM DO PROGRAMA [ ]
LISTAGEM DO EXEMPLO [ ]
GABARITO DOS CARTÕES DE ENTRADA OU DA IMPRESSORA [ ]
FOLHA DE ALTERAÇÃO VERSÃO 1 [ ]
VERSÃO = [ ]

```
// *

// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
      SUBROUTINE WZADR(IOCT,ISTEP,IABS)
C     ******************************************************************
C     *   CALCULA O ENDERECO E MONTA A INSTRUCAO
C     ******************************************************************
C     ERRO15 - SINTAXE ERRADA EM COMANDO SELF.RELATIVE
C     ERRO15- CARACTER INVALIDO DENTRO DE UMA EXPRESSAO DE ENDERECO
C     ERRO40 - ENDERECO ULTRAPASSA 4095 OU E MENOR QUE ZERO
      DIMENSION  ICOL(80)
      COMMON IPAG,KONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEQ,NOEQU(100),NOSIM(400,5),IENSI(400)
      COMMON IORIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ISAI(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
      EQUIVALENCE (IBCO,INIC(38)),(MAIS,INIC(39)),(MENOS,INIC(40)),
     1            (IESTR,INIC(45))
      EQUIVALENCE (ICOL(1),IERRA(1))
      DATA JKJJK /Z9F00/
C
      IOPTG=ISTEC(IAPTC+3)
      IOPTI=ISTEC(IAPTC+5)-4
      IF(IABS-1) 1,2,1
    1 IND=1
    3 NV=38
      IEND=0
      MX=1
      IF(IOPTI) 50,50,1200
   50 IOPTI=IOPTI+4
   55 GO TO(1100,440,150,2000),IOPTI
C
C     SIMBOLICO
C
  150 NUOPE=ISTEC(IAPTC+6)
      IVX=1
      II=1
      INK=IAPTC+6
  180 INK=INK+1
      KEND=ISTEC(INK)
C
C     VERIFICA SE E COMMON - SE FOR O ENDERECO E ABSOLUTO
C
      IF(NOSIM(KEND,4)-3) 190,1400,190
  190 IF(NOSIM(KEND,4)-2) 191,1400,191
  191 IEND=IEND+MX*IENSI(KEND)
      IF(II-NUOPE) 200,1000,1000
  200 II=II+1
      INK=INK+1
      KEND=ISTEC(INK)
      IF(KEND-39)350,300,350
300   MX=1
      GO TO 400
350   MX=-1
  400 IF(II-NUOPE) 410,445,445
```

```
  410 II=II+1
      GO TO 180
C
C     ENDERECO ABSOLUTO HEXADECIMAL
C
 2000 IND=0
      IVX=0
      GO TO 445
C
C     ENDERECAMENTO ABSOLUTO
C
440   IND=0
      IVX=1
  445 IOPTG=IOPTG-1
  450 CALL WZDEC(IOPTG,N,K1,K2,IVX,M,NV)
  451 IF(M) 1000,600,1000
600   IEND=IEND+MX*N
      IF(IEND-4095)700,700,1500
  700 IF(IEND)1500,1000,1000
C
C
 1000 CALL WZIOR(IOCT,IEND)
      RETURN
C
C     SELF-RELATIVE
C
 1100 KV=45
      K=IOPTG+1
      IEND=ISTEC(IAPTC+7)
      IOPTG=IOPTG+2
      IF(ISTEC(IAPTC+6)-1)1114,1000,1114
 1114 KK=ISTEC(IAPTC+8)-38
      GO TO (1140,1160),KK
 1115 CALL WZIER(15)
      RETURN
 1140 MX=1
 1170 IF(ICOL(IOPTG)-IESTR)1171,1180,1171
 1171 CALL WZDEC(K,N,K1,K2,1,M,38)
      GO TO 451
 1180 IF(ICOL(IOPTG+1)-IBCO) 1115,1181,1115
 1181 N=ISTEP
      GO TO 600
 1160 MX=-1
      IF(ICOL(IOPTG)-IESTR)1171,3000,1171
 3000 IF(ICOL(IOPTG+1)-IBCO) 1115,3100,1115
 3100 IND=0
      RETURN
C
C     INDIRETO
C
 1200 IND=0
      ITEMP=IOCT
      IOCT=JKJJK
      N2=0
      KG=1
      CALL WZTRT(IOCT,N2,ISTEP,KG)
      CALL WZITR(IOCT,IS1,IS2,IS3,IS4)
```

```
      CALL WZITR(ISTEP,K1,K2,K3,K4)
      ISTEP=ISTEP+1
      WRITE(IPRIN,7500)  INIC(K1),INIC(K2),INIC(K3),INIC(K4),INIC(IS1)
     1                   INIC(IS2)
 7500 FORMAT(1H ,7X,4A1,2X,4A1)
      IOCT=ITEMP
      IF(IABS-1)4,5,4
    4 IND=1
      GO TO 55
 1400 IND=0
      GO TO 191
 1500 CALL WZIER(33)
      RETURN
    5 IND=0
      GO TO 55
    2 IND=0
      GO TO 3
      END

UNREFERENCED STATEMENTS
 450     1170

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZADR
 COMMON   3130  VARIABLES     34  PROGRAM    560

RELATIVE ENTRY POINT ADDRESS IS 0037 (HEX)

END OF COMPILATION

// DUP

*DELETE              WZADR     0010
CART ID 0010   DB ADDR  1175   DB CNT   0023

*STORE      WS  UA  WZADR           0010
CART ID 0010   DB ADDR  2420   DB CNT   0023

// EJECT
```

NOME DO PROGRAMA	CÓDIGO DE ATIVIDADE NUM. DO DOCUMENTO	CÓDIGO DO PROGRAMA
ROTINA DE TRATAMENTO DAS INSTRUÇÕES DE DESLOCAMENTO		WZDES
	PROGRAMADOR	/ / DATA

LINGUAGEM USADA

FORTRAN [X] ASSEMBLER [ ] BASIC [ ] ALGOL [ ] .... [ ]

TIPO DE PROGRAMA PRINCIPAL

DRIVER [ ] SUBROUTINE [X] FUNCTION [ ] OUTROS [ ]

IDENTIFICAÇÃO DOS DISCOS (FITAS)	SISTEMA SUPORTE (SIST. OPERACIONAL E HARDWARE)
0010	

CHAVES DO PAINEL OU SAIDAS ALTERNATIVAS

NENHUMA

DESCRIÇÃO DO PROGRAMA

Esta rotina trata das instruções de deslocamento. Estas instruções apresentam um rótulo, um código e o número de deslocamentos pedido. O número de deslocamentos pode ser um número entre 0 e 4.

A rotina trata justamente o operando da instrução. Verifica-se se é um operando válido e se a sintaxe da instrução está correta. Se o teste se verificar satisfatório, o operando é montado na instrução.

 (use o verso, se necessário)

CÓDIGO DO PROGRAMA

WZDES

SEQUÊNCIA DE CHAMADA E DESCRIÇÃO DAS VARIÁVEIS - ARGUMENTO

CALL WZDES(IOCT)

IOCT - palavra de código sendo montada.

SUB-ROTINAS USADAS

WZIOR - insere campo de deslocamento na palavra de código

WZIER - tratamento do erro.

OBSERVAÇÕES OU MENSAGENS DE ERROS

Pode ocorrer o seguinte erro:

ERRO 8 - operando ou sintaxe do comando inválidos.

 (use o verso, se necessário)

CÓDIGO DE PROGRAMA

WZDES

VARIÁVEIS PRINCIPAIS

TIPO
(Dimension, Common, Real, Integer, Data, etc)

Nome	Tipo	Função da Variável
IOPTG	I,C	indica no buffer qual a primeira posição de oper
ISTEC	I,C	~~xxxx~~ buffer de informação suplementar
IAPTC	I,C	ponteiro para o buffer de informação suplem.
ICOL	I,C	buffer de cartões
INIC	I,C	caracteres iniciais permitidos

ANEXOS: DIAGRAMA DE BLOCOS [ ]

LISTAGEM DO PROGRAMA [ ]

LISTAGEM DO EXEMPLO [ ]

GABARITO DOS CARTÕES DE ENTRADA OU DA IMPRESSORA [ ]

FOLHA DE ALTERAÇÃO VERSÃO 1 [ ]

VERSÃO = [ ]

```
  140 N = NUM (K-21)
      I=I+2
      GO TO 85
C
  190 CALL WZIER(25)
      RETURN
  320 IF(ICOL(I-1)-IVIRG    )190,50,190
  800 IF(ICOL(I-1)-IBCD     )90,810,90
  810 CALL WZIOR(IOCT,N)
      RETURN
      END

UNREFERENCED STATEMENTS
 88

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZIIO
 COMMON   3130  VARIABLES     12  PROGRAM    326

RELATIVE ENTRY POINT ADDRESS IS 001A (HEX)

END OF COMPILATION

// DUP

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


*STORE      WS  UA  WZIIO           0010
CART ID 0010   DB ADDR  0382   DB CNT   0015

// EJECT
```

NOME DO PROGRAMA	CÓDIGO DE ATIVIDADE NUM. DO DOCUMENTO	CÓDIGO DO PROGRAMA
TRATAMENTO DAS INSTRUÇÕES DE ENDEREÇAMENTO DA MEMÓRIA NA PRIMEIRA FASE		WZOPE
	PROGRAMADOR	/ / DATA

LINGUAGEM USADA

FORTRAN [x] ASSEMBLER [ ] BASIC [ ] ALGOL [ ] .... [ ]

TIPO DE PROGRAMA

PRINCIPAL

DRIVER [ ] SUBROUTINE [x] FUNCTION [ ] OUTROS [ ]

IDENTIFICAÇÃO DOS DISCOS (FITAS)	SISTEMA SUPORTE (SIST. OPERACIONAL E HARDWARE)
0010	

CHAVES DO PAINEL OU SAIDAS ALTERNATIVAS

NENHUMA

DESCRIÇÃO DO PROGRAMA

Esta rotina trata dos operandos das instruções de endereçamento da memória na primeira fase. Este tratamento é constituido por:

- determinação do tipo de operando.
- tratamento parcial dos operandos simbólicos, incluindo ~~xxxxxxxx~~ inclusão do operando na tabela de símbolos, ~~xxxxxx~~ colocação da sua posição nesta tabela no buffer de informação suplementar, bem como informação sobre o tipo de operando.
- determinação dos delimitadores existentes entre os operandos no caso de operandos relativos e de expressões que envolvem endereços.
- cuidados especiais no tratamento do buffer de informação suplementar.

 (use o verso, se necessário)

CÓDIGO DO PROGRAMA
WZOPE

SEQUÊNCIA DE CHAMADA E DESCRIÇÃO DAS VARIÁVEIS - ARGUMENTO

CALL WZOPE(ISTEP,JIND)
ISTEP - contador de instruções
JIND - indica quando diferente de zero, operando indireto

SUB-ROTINAS USADAS

WZTRZ - trata operando simbólico determinando sua validade
WZPRC - busca e insere operando na tabela de simbolos

OBSERVAÇÕES ENTRELAÇAMENTO OU MENSAGENS DE ERROS

NENHUMA

 (use o verso, se necessário)

CÓDIGO DE PROGRAMA

WZOPE

VARIÁVEIS PRINCIPAIS

Nome	Tipo (Dimension, Common, Real, Integer, Data, etc)	Função da Variável
ISTEP	I,C	contador de instruções
ICOL	I,C	buffer de cartões
IESTR	I	caracter asteristico
IBAR	I	caracter barra
IAPTC	I,C	ponteiro para o buffer de inf. suplem.
M	I	indica quando diferente de 0, que ocorreu erro
NUOPE	I	número de palavras na parte corresp. ao operando
IOPTG	I	proxima posição de tratamento de op. na seg. fas
INIC	I,C	matriz de caracteres permitidos
MAIS	I	caracter mais
MENOS	I	caracter menos
ISW1	I	indica qual o deleimitador encontrado

ANEXOS: DIAGRAMA DE BLOCOS [ ]

LISTAGEM DO PROGRAMA [ ]

LISTAGEM DO EXEMPLO [ ]

GABARITO DOS CARTÕES DE ENTRADA OU DA IMPRESSORA [ ]

FOLHA DE ALTERAÇÃO VERSÃO 1 [ ]

VERSÃO = [ ]

```
// JOB    0001 0010                                          *     E102,MA

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 *  16/ 04/ 74  * 09 H  52 MI

// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
      SUBROUTINE WZOPE(ISTEP,JIND)
C
C      OS TIPOS DE OPERANDO SAO
C         1 - SELF-RELATIVE DIRETO
C         2 - ABSOLUTO DECIMAL DIRETO
C         3 - SIMBOLICO DIRETO
C         4 - ABSOLUTO HEXA DIRETO
C         5 - SELF-RELATIVE INDIRETO
C         6 - ABSOLUTO DECIMAL INDIRETO
C         7 - SIMBOLICO INDIRETO
C         8 - ABSOLUTO HEXA INDIRETO
C
      COMMON IPAG,KONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEQ,NOEQU(100),NOSIM(400,5),IENSI(400)
      COMMON IORIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ICOL(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
      EQUIVALENCE  (IESTR,INIC(45)),(IBAR,INIC(47)),(IBCO,INIC(38))
      EQUIVALENCE (INIC(39),MAIS),(INIC(40),MENOS)
      NJOPE=0
      K=ISTEC(IAPTC+3)
C
      IF(ICOL(K)-IESTR   )100,800,100
  100 IF(ICOL(K)-IBAR    )200,700,200
  200 DO 300 JJ=28,37
      IF(ICOL(K)-INIC(JJ))300,600,300
  300 CONTINUE
C
C     OPERANDO SIMBOLICO
C
  330 ISTEC(IAPTC+5)=3
  350 CALL WZTRZ(K,IFIM,2,ISW1,M)
      IF(M) 1777,400,1777
  400 CALL WZPRC(0,KV,ISTEP,K,IFIM)
      NJOPE=NJOPE+1
      IND=IAPTC+NJOPE+6
      ISTEC(IND)=KV
      IF(ISW1-38) 500,1777,500
  500 NJOPE=NJOPE+1
      IND=IAPTC+NJOPE+6
      ISTEC(IND)=ISW1
      IOPTG=IFIM+2
```

```
// *

// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
      SUBROUTINE WZDES(IOCT)
C     ***************************************************************DES
C     *     ESTA SUBROTINA TRATA DOS DESLOCAMENTOS                    DES
C     ***************************************************************DES
      DIMENSION ICOL(80)
      DIMENSION NK(5)
      COMMON IPAG,KONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEQ,NOEQU(100),NOSIM(400,5),IENSI(400)
      COMMON IORIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ISAI(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
      EQUIVALENCE (ICOL(1),IERRA(1))
      DATA NK,IBCD/0,8,12,14,15,' '/
C
C                                                                     NI
      IOPTG=ISTEC(IAPTC+3)
      IF(ICOL(IOPTG+1)-IBCD) 220,200,220
  200 DO 210 I=28,32
      IF(ICOL(IOPTG)-INIC(I))210,250,210
  210 CONTINUE
  220 CALL WZIER(   8)
      RETURN
  250 I=I-27
      N=NK(I)
C                                                                     NI
  880 CALL WZIOR(IOCT,N)
      RETURN
      END

UNREFERENCED STATEMENTS
 880

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZDES
 COMMON   3130  VARIABLES     12  PROGRAM     88

RELATIVE ENTRY POINT ADDRESS IS 0010 (HEX)

END OF COMPILATION

// DUP

*STORE      WS  UA  WZDES           0010
CART ID 0010   DB ADDR  02BE   DB CNT   0008

// EJECT
```

NOME DO PROGRAMA	CÓDIGO DE ATIVIDADE NUM. DO DOCUMENTO	CÓDIGO DO PROGRAMA
TRATAMENTO DAS INSTRUÇÕES DE ENTRADA E SAÍDA		WZIIO
	PROGRAMADOR	/ / DATA

LINGUAGEM USADA

FORTRAN [x] ASSEMBLER [ ] BASIC [ ] ALGOL [ ] .... [ ]

TIPO DE PROGRAMA

PRINCIPAL

DRIVER [ ] SUBROUTINE [x] FUNCTION [ ] OUTROS [ ]

IDENTIFICAÇÃO DOS DISCOS (FITAS)	SISTEMA SUPORTE (SIST. OPERACIONAL E HARDWARE)
0010	

CHAVES DO PAINEL OU SAIDAS ALTERNATIVAS

NENHUMA

DESCRIÇÃO DO PROGRAMA

Esta instrução rotina trata as instruções de entrada e saída, cujo formato é o seguinte:

rótulo { SAI / ENTR } n

onde "n" é um número decimal, ou então:

rótulo { SAI / ENTR } operando

nesse caso operando é um símbolo que deve ter sido já definido, na primeira fase.

Pertencem ainda a este grupo as instruções: SAL e FNC, cujo formato é o seguinte:

rótulo { SAL / FNC } n,m

onde "n" e "m" são números decimais

Vale ainda as formas:

rótulo { SAL / FNC } operando, operando2

O tratamento desta instruções consiste em:

- determinação do tipo do operando (numérico ou simbólico)
- se numérico, conversão do operando e sua inclusão na instrução.
- se simbólico, determinação do endereço deste operando.
- em ambos os casos, determinação da validade do operando.
- se a instrução for do tipo 2 apresentado, tratamento identico para o segundo operando.

 (use o verso, se necessário)

CÓDIGO DO PROGRAMA

WZIIO

SEQUÊNCIA DE CHAMADA E DESCRIÇÃO DAS VARIÁVEIS - ARGUMENTO

CALL WZIIO(IOCT,ISTEP)

IOCT - palavra de código sendo montada.

ISTEP - contador deinstruções.

SUB-ROTINAS USADAS

WZIER - tratamento dos erros.

WZTRA - trata o operando simbólico, determinando sua validade

WZPRO - buca o operando simbólico na tabela de símbolos.

WZIOR - inclui o operando já na forma binária, na palavra de código.

OBSERVAÇÕES OU MENSAGENS DE ERROS

Podem ocorrer os seguintes erros:

ERRO 10 - instrução tipo 1, com sintaxe errada; ou dispositivo fora dos limites permitidos

ERRO 24 - tipo de endereço simbólico proibido.

ERRO 25 - comando tipo 2 tem sintaxe errada.

(use o verso, se necessário)

CÓDIGO DE PROGRAMA

WZIIO

VARIÁVEIS PRINCIPAIS

TIPO
(Dimension, Common, Real, Integer, Data, etc)

Nome	Tipo	Função da Variável
ISTEC	I,C	buffer de informação suplementar
IAPTC	I,C	ponteiro para o buffer de inf. suplem.
EENMN	I,C	código da instrução
ICOL	I,C	buffer de leitura de cartões
INIC	I,C	caracteres iniciais permitidos
M	I,C	indica quando diferente de zero, ocor. de erro
NOSIM	I,C	tabela de símbolos
IENSI	I,C	endereços dos símbolos na tabela de símbolos
IT	I	posição do símbolo na tabela de símbolos
IBCO	I	caracter branco
IVIRG	I	caracter vírgula
NUM	I,C	números decimais e hexadecimais
N	I	operando convertido

ANEXOS: DIAGRAMA DE BLOCOS [ ]

LISTAGEM DO PROGRAMA [ ]

LISTAGEM DO EXEMPLO [ ]

GABARITO DOS CARTÕES DE ENTRADA OU DA IMPRESSORA [ ]

FOLHA DE ALTERAÇÃO VERSÃO 1 [ ]

VERSÃO = [ ]

```
// *

// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
      SUBROUTINE WZIIO(IOCT,ISTEP)
C     ******************************************************************
C     *          ESTA SUBROTINA TRATA DA ENTRADA E DA SAIDA            *
C     ******************************************************************
      DIMENSION  ICOL(80)
      COMMON IPAG,KONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEQ,NOEQU(100),NOSIM(400,5),IENSI(400)
      COMMON IORIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ISAI(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
      EQUIVALENCE (ICOL(1),IERRA(1))
      EQUIVALENCE (IVIRG,INIC(42)),(IBCO,INIC(38))
C
      I=ISTEC(IAPTC+3)
      IENMN=ISTEC(IAPTC+2)
      MV = 2
   50 DO 55 K=28,37
      IF(ICOL(I)-INIC(K))55,100,55
   55 CONTINUE
      CALL WZTRA(I,IFIM,4,ISW,M)
      IF(M) 60,70,60
   60 RETURN
C
   70 CALL WZPRO(2,IT,ISTEP,I,IFIM)
      K=NOSIM(IT,4)+1
      GO TO (95,75,75,75,75,95,95,75,75,75,95),K
   75 IF(IENSI(IT)-15) 80,80,90
   80 N = IENSI(IT)
      I = IFIM + 2
   85 IF (MV-1) 87,800,87
   87 N = N*256
   88 CALL WZIOR(IOCT,N)
      MV = 1
      IF(IENMN-33)188,200,188
  188 IF(IENMN-14)320,200,320
  200 IF(ICOL(I-1)-IBCO) 90,60,90
C
   90 CALL WZIER(10)
      RETURN
C
   95 CALL WZIER(24)
      RETURN
C
  100 IF(K-29)140,110,140
  110 DO 120 J=28,33
      IF(ICOL(I+1)-INIC(J))120,130,120
  120 N = 1
      I=I+2
      GO TO 85
  130 N = 10 + NUM(J-21)
      I=I+3
      GO TO 85
```

```
      K=IOPTG
      ISTEC(IAPTC+3)=IOPTG
      DO 520 JJ=28,37
      IF(ICOL(K)-INIC(JJ)) 520,1777,520
  520 CONTINUE
      GO TO 350
C
C     OPERANDO E ABSOLUTO DECIMAL
C
  600 ISTEC(IAPTC+5)=2
      ISTEC(IAPTC+3)=K
      GO TO 1777
C
C     OPERANDO E ABSOLUTO HEXADECIMAL
C
  700 ISTEC(IAPTC+3)=K+1
      ISTEC(IAPTC+5)=4
      GO TO 1777
C
C     SELF-RELATIVE
C
  800 ISTEC(IAPTC+5)=1
      NUOPE=NUOPE+1
      IND=NUOPE+IAPTC+6
      ISTEC(IND)=ISTEP
      IF(ICOL(K+1)-MAIS) 850,900,850
  850 IF(ICOL(K+1)-MENOS) 1777,905,1777
  905 ISW1=40
      GO TO 910
  900 ISW1=39
  910 NUOPE=NUOPE+1
      IND=NUOPE+IAPTC+6
      ISTEC(IND)=ISW1
 1777 IF(JIND-1) 1778,1779,1778
 1779 ISTEC(IAPTC+5)=ISTEC(IAPTC+5)+4
 1778 IND=IAPTC+NUOPE+7
      IF(IND-80) 1800,1800,2000
 1800 ISTEC(IAPTC)=IND
 1801 ISTEC(IAPTC+4)=2
      ISTEC(IAPTC+6)=NUOPE
      IAPTC=IAPTC+NUOPE+7
      RETURN
 2000 ISTEC(IAPTC)=IND-80
      GO TO 1801
      END

UNREFERENCED STATEMENTS
 330

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZOPE
 COMMON   3130  VARIABLES     12  PROGRAM    432

RELATIVE ENTRY POINT ADDRESS IS 0019 (HEX)
```

```
END OF COMPILATION

// DJP

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


*STORE      WS  JA  WZOPE            0010
CART ID 0010   DB ADDR  17F1   DB CNT   001B

// EJECT
```

NOME DO PROGRAMA	CÓDIGO DE ATIVIDADE NUM. DO DOCUMENTO	CÓDIGO DO PROGRAMA
Rotina de geração dos comandos "SAL" e "LIMP"		W2MIC
	PROGRAMADOR	/ / DATA

LINGUAGEM USADA

FORTRAN [X] ASSEMBLER [ ] BASIC [ ] ALGOL [ ] .... [ ]

TIPO DE PROGRAMA

DRIVER [ ] SUBROUTINE [X] PRINCIPAL FUNCTION [ ] OUTROS [ ]

IDENTIFICAÇÃO DOS DISCOS (FITAS)	SISTEMA SUPORTE (SIST. OPERACIONAL E HARDWARE)

CHAVES DO PAINEL OU SAIDAS ALTERNATIVAS

DESCRIÇÃO DO PROGRAMA

Gera os comandos SLT, SLV, SLTM, SLVM e LIMP.

Estas instruções tem o formato:

SLT Ø
SLT 1
LIMP Ø
LIMP 1

1 (use o verso, se necessário)

CÓDIGO DO PROGRAMA

WZMIC

SEQUÊNCIA DE CHAMADA E DESCRIÇÃO DAS VARIÁVEIS - ARGUMENTO

CALL WZMIC (IOCT, ISW)

IOCT - palavra a ser preenchida com o valor gerado.

ISW - chave que indica:
= Ø → LIMP
= 1 → saltos.

SUB-ROTINAS USADAS

WZIOR - mascara os instruções convenientemente

WZIER - rotina tratadora de erros na segunda fase.

OBSERVAÇÕES OU MENSAGENS DE ERROS

ERRO 9 - operando é diferente de 1 ou ZERO.

 (use o verso, se necessário)

CÓDIGO DE PROGRAMA

VARIÁVEIS PRINCIPAIS

Nome	TIPO (Dimension, Common, Real, Integer, Data, etc) Tipo	Função da Variável

ANEXOS:

DIAGRAMA DE BLOCOS [ ]

LISTAGEM DO PROGRAMA [ ]

LISTAGEM DO EXEMPLO [ ]

GABARITO DOS CARTÕES DE ENTRADA OU DA IMPRESSORA [ ]

FOLHA DE ALTERAÇÃO VERSÃO 1 [ ]

VERSÃO = [ ]

```
// *

// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
      SUBROUTINE WZMIC(IOCT,ISW)
C     ******************************************************************
C     *         ESTA SUBROUTINE TRATA DAS INSTRUCOES SALTOS E LIMP.
C     *         ISW = 1  -  SALTOS.
C     *         ISW = 0  -  MIC LIMP.
C     ******************************************************************
      DIMENSION ICOL(80)
      COMMON IPAG,KONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEQ,NOEQU(100),NOSIM(400,5),IENSI(400)
      COMMON IORIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ISAI(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
      EQUIVALENCE (ICOL(1),IERRA(1))
C
C
      DATA IZERO,IUM,IBCO,I1,I0/'0','1',' ',512,1792/
      IOPTG=ISTEC(IAPTC+3)
      IF(ICOL(IOPTG+1)-IBCO)550,200,550
  200 IF(ICOL(IOPTG)-IUM)540,300,540
  300 IF(ISW-1) 400,350,400
C
C     SALTOS
C
  350 CALL WZIOR(IOCT,I1)
      RETURN
C
C     MIC LIMP
C
  400 CALL WZIOR(IOCT,I0)
      RETURN
C
  540 IF(ICOL(IOPTG)-IZERO)550,560,550
  550 CALL WZIER(9)
  560 RETURN
      END

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZMIC
 COMMON   3130  VARIABLES      8  PROGRAM     76

RELATIVE ENTRY POINT ADDRESS IS 000A (HEX)

END OF COMPILATION

// DUP

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

```
*STORE       WS  JA  WZMIC            0010
CART ID 0010   DB ADDR  0397   DB CNT   0007

// EJECT
```

```
// *

// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
      SUBROUTINE WZTRT(IOCT,N2,ISTEP,IK)
C
C     TRATADORA DAS SAIDAS
C     IND - INDICADOR DE RELOCACAO
C     IAPIN - APONTADOR DA TABELA DE GERACAO
C     IABLE - TABELA DE GERACAO
C     IAC - ACUMULADOR DE CONTAGEM DE PALAVRAS NO MODULO
C     ICOMP - COMPLEMENTO PARA O FIM DE SETOR DO MODULO
C     N2 - SEGUNDA PALAVRA DE GERACAO NO CASO DE SUBROUTINES
C
      DIMENSION IABLE(320)
      COMMON IPAG,KONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEQ,NOEQU(100),NOSIM(400,5),IENSI(400)
      COMMON IORIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ISAI(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
      EQUIVALENCE(IDISK,ID7),(NR,N7)
      EQUIVALENCE (IABLE(1),ISAI(1)),(IKONT,IORIG)
C
      NA=IK
      MA = IND
      IF(IKONT)10,200,10
  200 CALL WZINS(0)
   10 CALL WZINS(IOCT)
   30 IF(IKONT-9)160,40,160
C
C
   40 IABLE(IAPIN-8)=IRET
      IKONT=0
      IF(ICOMP) 60,50,60
C
C     DATA BREACK CAUSADA POR FIM DE SETOR
C
   50 CALL WZDTB(ISTEP)
      IAPIN=2
      ICOMP=318
      WRITE(IDISK'NR) IABLE
      IAC=2
C
C     INICIALIZA A TABELA
C
   60 IRET=0
  130 IF(MA-3)140,150,140
  140 IF(NA-2) 145,190,145
  145 RETURN
  150 IF(IKONT)153,152,153
  152 CALL WZINS(0)
  153 IND=0
      CALL WZINS(N2)
      NA=1
      MA=0
      GO TO 30
```

```
  160 IF(ICOMP)130,170,130
C
C     DATA BREAK CAUSADA POR FIM DE SETOR
C
  170 INK=IAPIN-IKONT+1
      IABLE(INK)=IRET*2**(18-2*IKONT)
      IKONT=0
      GO TO 50
  190 N2=0
      IND=0
      NA=1
      MA=1
      GO TO 150
      END

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZTRT
 COMMON   3130  VARIABLES      6  PROGRAM    186

RELATIVE ENTRY POINT ADDRESS IS 000D (HEX)

END OF COMPILATION

// DUP

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


*STORE      WS  UA  WZTRT           0010
CART ID 0010   DB ADDR  03A6   DB CNT   000C

// EJECT
```

NOME DO PROGRAMA	CÓDIGO DE ATIVIDADE NUM. DO DOCUMENTO	CÓDIGO DO PROGRAMA
ROTINA AUXILIAR DE FINALIZAÇÃO DE UM BLOCO E INICIAÇÃO DO PROXIMO		WZDTB
	PROGRAMADOR	/ / DATA

LINGUAGEM USADA

FORTRAN [x] ASSEMBLER [ ] BASIC [ ] ALGOL [ ] .... [ ]

TIPO DE PROGRAMA

PRINCIPAL

DRIVER [ ] SUBROUTINE [x] FUNCTION [ ] OUTROS [ ]

IDENTIFICAÇÃO DOS DISCOS (FITAS)	SISTEMA SUPORTE (SIST. OPERACIONAL E HARDWARE)
0010	

CHAVES DO PAINEL OU SAIDAS ALTERNATIVAS

NENHUMA

DESCRIÇÃO DO PROGRAMA

Esta rotina, gera o cabeçalho de dados do bloco de dados que está sendo finalizado neste instante. O cabeçalho de dados se compõe de um endereço de carregamento e do número de palavras que integram o bloco. O endereço do bloco que está sendo iniciado é guardado em seguida; o contador de palavras é iniciado.

1 (use o verso, se necessário)

CÓDIGO DO PROGRAMA

WZDTB

SEQUÊNCIA DE CHAMADA E DESCRIÇÃO DAS VARIÁVEIS - ARGUMENTO

CALL WZDTB(ISTEP)

ISTEP - contador de instruções

SUB-ROTINAS USADAS

NENHUMA

OBSERVAÇÕES [illegible] OU MENSAGENS DE ERROS

NENHUMA

 (use o verso, se necessário)

	CÓDIGO DE PROGRAMA
	WZDTB

VARIÁVEIS PRINCIPAIS

TIPO
(Dimension, Common, Real, Integer, Data, etc)

Nome	Tipo	Função da Variável
IAPIN	I,C	ponteiro para o buffer de saída de disco
IAC	I,C	contador de palavras em um bloco de dados
IABLE	I,C	buffer de disco- equivale a ISAI
IS	I,C	total de palavras ocupadas no buffer
ISTRT	I,C	endereço de carregamento de um bloco de dados

ANEXOS: DIAGRAMA DE BLOCOS [ ]
LISTAGEM DO PROGRAMA [ ]
LISTAGEM DO EXEMPLO [ ]
GABARITO DOS CARTÕES DE ENTRADA OU DA IMPRESSORA [ ]
FOLHA DE ALTERAÇÃO VERSÃO 1 [ ]
VERSÃO = [ ]

```
// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
      SUBROUTINE WZDTB(ISTEP)
C
C        DATA BREAK CAUSADO POR -
C          - ORG
C          - INICIO DE UM NOVO SETOR
C          - FIM DO PROGRAMA
C
      DIMENSION IABLE(320)
      COMMON IPAG,KONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEQ,NOEQU(100),NOSIM(400,5),IENSI(400)
      COMMON IORIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ISAI(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
      EQUIVALENCE (ISAI(1),IABLE(1))
C
      INK=IAPIN-IAC+1
      IABLE(INK)=ISTRT
      IABLE(INK+1)=IAC
      IS=IS+IAC
      ISTRT=ISTEP
      IAC=2
      RETURN
      END

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZDTB
 COMMON   3130  VARIABLES      4  PROGRAM     50

RELATIVE ENTRY POINT ADDRESS IS 0006 (HEX)

END OF COMPILATION

// DUP

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


*STORE      WS  UA  WZDTB           0010
CART ID 0010   DB ADDR  04AF   DB CNT   0004

// EJECT
```

NOME DO PROGRAMA	CÓDIGO DE ATIVIDADE NUM. DO DOCUMENTO	CÓDIGO DO PROGRAMA
ROTINA DE TRATAMENTO DE ENDEREÇOS NO FIM DA PRIMEIRA FASE		WZEND
	PROGRAMADOR	/ / DATA

LINGUAGEM USADA

FORTRAN [X] ASSEMBLER [ ] BASIC [ ] ALGOL [ ] .... [ ]

TIPO DE PROGRAMA

DRIVER [ ] SUBROUTINE [X] PRINCIPAL FUNCTION [ ] OUTROS [ ]

IDENTIFICAÇÃO DOS DISCOS (FITAS)	SISTEMA SUPORTE (SIST. OPERACIONAL E HARDWARE)
0010	

CHAVES DO PAINEL OU SAIDAS ALTERNATIVAS

NENHUMA

DESCRIÇÃO DO PROGRAMA

Esta rotina trata do endereçamento dos simbolos de área comum e daqueles que se encontram em equivalência, na lista de símbolos equivalentes.

A rotina trata primeiramente do endereçamento dos símbolos que se encontram em área comum, endereçando-os a partir do fim da memória principal.

Em seguida a rotina trata os símbolos equivalentes. Para isso faz-se uma tentativa para esvaziar a lista de símbolos equivalentes. No caso da tentativa malograr, não se podendo endereçar todos os símbolos, mensagens de erros são impressas.

 (use o verso, se necessário)

CÓDIGO DO PROGRAMA

WZEND

SEQUÊNCIA DE CHAMADA E DESCRIÇÃO DAS VARIÁVEIS - ARGUMENTO

CALL WZEND(IERRO)

IERRO - parâmetro de erro. Indica erro se diferente de zero.

SUB-ROTINAS USADAS

WZERR - tratamento dos erros

OBSERVAÇÕES [illegible] OU MENSAGENS DE ERROS

São detectados os seguintes erros:

ERRO 21 - dois símbolos equivalentes, endereçados em posições diferentes da memória.

ERRO 22 - a lista não pode ser esvaziada: restam ainda símbolos não enderçados

 (use o verso, se necessário)

CÓDIGO DE PROGRAMA

WZEND

VARIÁVEIS PRINCIPAIS

TIPO
(Dimension, Common, Real, Integer, Data, etc)

Nome	Tipo	Função da Variável
IAPSI	I,C	ponteiro para a tabela de símbolos
IEND	I,C	endereço do símbolo
NOSIM	I,C	tabela de símbolos
IENSI	I,C	endereço de um símbolo da tabela de símbolos
KONT	I,C	número de pares de simb. ainda não endereçados
NOEQU	I,C	tabela de equivalencia

ANEXOS: DIAGRAMA DE BLOCOS [ ]
LISTAGEM DO PROGRAMA [ ]
LISTAGEM DO EXEMPLO [ ]
GABARITO DOS CARTÕES DE ENTRADA OU DA IMPRESSORA [ ]
FOLHA DE ALTERAÇÃO VERSÃO 1 [ ]
VERSÃO = [ ]

```
// *

// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
      SUBROUTINE WZEND(IERRO)
      COMMON IPAG,KONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEQ,NOEQU(100),NOSIM(400,5),IENSI(400)
      COMMON IORIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ICOL(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
C
C               ENDERECA COMMON
C
      IF(IAPSI) 500,4100,500
  500 IEND=4016
      DO 550 J=1,IAPSI
      IF(NOSIM(J,4)-3 ) 550,530,550
  530 IEND=IEND-NOSIM(J,5)
      IENSI(J)=IEND
  550 CONTINUE
      L=0
C
C               SIMBOLOS EQUIVALENTES
C
      KONT=0
      IF(IAPEQ) 700,4100,700
  700 DO 1500 J=1,IAPEQ,2
      K=NOEQU(J)
      K1=NOEQU(J+1)
      KK=NOSIM(K,4)+1
      KK1=NOSIM(K1,4)+1
      GO TO (6000,6001,6001,6001,6001,6000,6000,6001,6001,6001,6001),KK
  899 IF(N1) 900,1200,900
  900 IF(N2)1000,1100,1000
 1000 IF(IENSI(K)-IENSI(K1))1050,1500,1050
 1050 CALL WZERR(21)
      IERRO=1
      GO TO 4100
 1100 IENSI(K1)=IENSI(K)
      NOSIM(K1,4)=NOSIM(K,4)
      GO TO 1500
 1200 IF(N2)1300,1400,1300
 1300 IENSI(K)=IENSI(K1)
      NOSIM(K,4)=NOSIM(K1,4)
      GO TO 1500
 1400 KONT=KONT+1
 1500 CONTINUE
      IF(KONT)1600,2100,1600
 1600 IF(KONT-L)1700,1800,1700
 1700 L=KONT
      KONT=0
      IF(IERRO)2100,700,2100
 1800 CALL WZERR(22)
      IERRO=1
 2100 CONTINUE
 4100 RETURN
```

```
 6000 N1=0
 6002 GO TO (7000,7001,7001,7001,7001,7000,7000,7001,7001,7001,7001),KK1
 6001 N1=1
      GO TO 6002
 7000 N2=0
      GO TO 899
 7001 N2=1
      GO TO 899
      END

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZEND
 COMMON   3130  VARIABLES     14  PROGRAM    332

RELATIVE ENTRY POINT ADDRESS IS 0015 (HEX)

END OF COMPILATION

// DUP

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


*STORE      WS  UA  WZEND           0010
CART ID 0010   DB ADDR  036D   DB CNT   0015

// EJECT
```

ROTINAS DE TRATAMENTO DAS INSTRUÇÕES

Estas rotinas destinam-se ao tratamento das instruções.

WZPNL
WZIME
WZADR
WZDES
WZIIO
WZOPE
WZMIC

NOME DO PROGRAMA	CÓDIGO DE ATIVIDADE NUM. DO DOCUMENTO	CÓDIGO DO PROGRAMA
TRATAMENTO DAS INSTRUÇÕES DE PAINEL		WZPNL
	PROGRAMADOR	/ / DATA

LINGUAGEM USADA

FORTRAN [X] ASSEMBLER [ ] BASIC [ ] ALGOL [ ] .... [ ]

TIPO DE PROGRAMA PRINCIPAL

DRIVER [ ] SUBROUTINE [X] FUNCTION [ ] OUTROS [ ]

IDENTIFICAÇÃO DOS DISCOS (FITAS)	SISTEMA SUPORTE (SIST. OPERACIONAL E HARDWARE)
0010	

CHAVES DO PAINEL OU SAIDAS ALTERNATIVAS

NENHUMA

DESCRIÇÃO DO PROGRAMA

Esta subroutine trata as instruções de painel. Estas instruções tem a seguinte forma em assembler:

rótulo PNL n

onde "n" é um número decimal, de 0 a 7.

A subrotina trata o operando, determinando a sua validade, e em caso afirmativo, o operando é inserido na instrução sendo montada.

 (use o verso, se necessário)

CÓDIGO DO PROGRAMA

WZPNL

SEQUÊNCIA DE CHAMADA E DESCRIÇÃO DAS VARIÁVEIS - ARGUMENTO

CALL WZPNL(IOCT)

IOCT - palavra de código sendo montada.

SUB-ROTINAS USADAS

WZIER - tratadora de erro.

WZIOR - insere o operando na instrução.

OBSERVAÇÕES OU MENSAGENS DE ERROS

Pode ocorrer o seguinte erro:

ERRO 30 - operando inválido, ou fora da faixa permitida.

 (use o verso, se necessário)

CÓDIGO DE PROGRAMA

WZPNL

VARIÁVEIS PRINCIPAIS

TIPO
(Dimension, Common, Real, Integer, Data, etc)

Nome	Tipo	Função da Variável
IOPTG	I,C	primeira posição do operando
ISTEC	I,C	buffer de informação suplementar
IAPTC	I,C	ponteiro do buffer de informação suplementar
IBCO	I,C	caracter branco
ICOL	I,C	buffer de cartões
IOCT	I,C	palavra de código sendo montada
NUM	I,C	tabela de números

ANEXOS: DIAGRAMA DE BLOCOS [ ]

LISTAGEM DO PROGRAMA [ ]

LISTAGEM DO EXEMPLO [ ]

GABARITO DOS CARTÕES DE ENTRADA OU DA IMPRESSORA [ ]

FOLHA DE ALTERAÇÃO VERSÃO 1 [ ]

VERSÃO = [ ]

```
// *

// FOR
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
      SUBROUTINE WZPNL(IOCT)
C
C     ******************************************************************
C     *        ESTA SUBROUTINE TRATA DAS INSTRUCOES DO PAINEL.         *
C     ******************************************************************
C
      DIMENSION ICOL(80)
      COMMON IPAG,KONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEQ,NOEQU(100),NOSIM(400,5),IENSI(400)
      COMMON IORIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ISAI(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
      EQUIVALENCE (ICOL(1),IERRA(1))
      EQUIVALENCE (IBCO,INIC(38))
      IOPTG=ISTEC(IAPTC+3)
      IF(ICOL(IOPTG+1)-IBCO)85,70,85
   70 DO 80 I=28,35
      IF(ICOL(IOPTG)-INIC(I)) 80,90,80
   80 CONTINUE
   85 CALL WZIER(   30)
      RETURN
C
   90 N = NUM(I-21)*256
      CALL WZIOR(IOCT,N)
      RETURN
      END

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZPNL
 COMMON   3130  VARIABLES      6  PROGRAM     86

RELATIVE ENTRY POINT ADDRESS IS 000A (HEX)

END OF COMPILATION

// DUP

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


*STORE      WS  UA  WZPNL            0010
CART ID 0010   DB ADDR  02FE   DB CNT   0006

// EJECT
```

NOME DO PROGRAMA	CÓDIGO DE ATIVIDADE NUM. DO DOCUMENTO	CÓDIGO DO PROGRAMA
ROTINA DE TRATAMENTO DAS INSTRUÇÕES IMEDIATAS		WZIME
	PROGRAMADOR	/ / DATA

LINGUAGEM USADA

FORTRAN [X] ASSEMBLER [ ] BASIC [ ] ALGOL [ ] .... [ ]

TIPO DE PROGRAMA

DRIVER [ ] SUBROUTINE [X] PRINCIPAL FUNCTION [ ] OUTROS [ ]

IDENTIFICAÇÃO DOS DISCOS (FITAS)	SISTEMA SUPORTE (SIST. OPERACIONAL E HARDWARE)
0010	

CHAVES DO PAINEL OU SAIDAS ALTERNATIVAS

NENHUMA

DESCRIÇÃO DO PROGRAMA

Esta rotina trata da montagem das instruções imediatas, que tem a forma:

$$\text{rótulo} \left\{ \begin{array}{l} \text{SOMI} \\ \text{NAND} \\ \text{XOR} \\ \text{CARI} \end{array} \right\} \left(\begin{array}{c} /n \\ n \end{array} \right)$$

onde o operando é facultativo.

O tratamento consiste em se determinar o operando, quando este existe, testar o seu tipo e sua validade, e em caso dos testes se mostrarem válidos, colocar o operando na instrução correspondente.

 (use o verso, se necessário)

CÓDIGO DO PROGRAMA

WZIME

SEQUÊNCIA DE CHAMADA E DESCRIÇÃO DAS VARIÁVEIS - ARGUMENTO

CALL WZIME (IOCT,I)
IOCT - palavra de codigo, sendo montada
I - tamanho da instrução.

SUB-ROTINAS USADAS

WZIOR - monta o operando em IOCT
WZIER - tratamento de erro.
WZDEC - conversão do operando.

OBSERVAÇÕES [illegible] OU MENSAGENS DE ERROS

Pode ocorrer o seguinte erro:
ERRO 11 - operando é maior que 255.

 (use o verso, se necessário)

CÓDIGO DE PROGRAMA

WZIME

VARIÁVEIS PRINCIPAIS

TIPO
(Dimension, Common, Real, Integer, Data, etc)

Nome	Tipo	Função da Variável
ISTEC	I,C	buffer de informação suplementar
IBCO	I	caracter branco
IBAR	I	caracter barra
IAPTC	I,C	ponteiro para o buffer de informação suplementar
ICOL		buffer de leitura de cartões
ISW	I	chave que indica conversão dec.(1) ou hexa(0)
M	I	indica se diferente de zero a ocorrência de erro
N	I	operando convertido
K1	I	posição de início do operando
K2	I	posição final do operando

ANEXOS: DIAGRAMA DE BLOCOS [ ]

LISTAGEM DO PROGRAMA [ ]

LISTAGEM DO EXEMPLO [ ]

GABARITO DOS CARTÕES DE ENTRADA OU DA IMPRESSORA [ ]

FOLHA DE ALTERAÇÃO VERSÃO 1 [ ]

VERSÃO = [ ]

```
// 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 *  15/ 04/ 74  * 10 H  29 MIN

// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
      SUBROUTINE WZIME(IDCT,JST)
C     ******************************************************************
C     *       ESTA SUBROUTINE MONTA OS OPERANDOS IMEDIATOS             *
C     ******************************************************************
      DIMENSION ICOL(80)
      COMMON IPAG,KONT,IPRIN,ICAB(37),ID5,ID6,ID7,N5,N6,N7
      COMMON IAPSI,IAPEQ,NOEQU(100),NOSIM(400,5),IENSI(400)
      COMMON IORIG,ISTRT,ICNT,LST,ISWER
      COMMON NUM(16),INIC(62),ISAI(320),ISTEC(90),IAPTC
      COMMON IS,IERR,IAPER,IERRA(80),IND,IAPIN,IRET,IAC,ICOMP
      EQUIVALENCE (INIC(38),IBCD),(INIC(47),IBAR)
      EQUIVALENCE (ICOL(1),IERRA(1))
C
      K=ISTEC(IAPTC+3)
      IF(ICOL(K)-IBCD) 350,150,350
  150 RETURN
C
  350 IF(IBAR-ICOL(K))370,450,370
  370 K=K-1
      ISW=1
      GO TO 495
C
  450 ISW=0
C
C
  495 CALL WZDEC(K,N,K1,K2,ISW,M,38)
      IF(M)150,500,150
  500 IF(N-255)550,550,650
  550 CALL WZIOR(IDCT,N)
      RETURN
C
  650 CALL WZIER(11)
      GO TO 150
C
C
      END

FEATURES SUPPORTED
 ONE WORD INTEGERS

CORE REQUIREMENTS FOR WZIME
 COMMON   3130  VARIABLES      8  PROGRAM     90
```