

PROCESSADOR DE LISTAS SIMÉTRICAS (SLIP)

[IMPLANTACAO NO SISTEMA DEC-10]

PALTONIO DAUN FRAGA - IMECC - UNICAMP
NOVEMBRO DE 1973

ORIENTADOR: DR. ODELAR LEITE LINHARES

TRABALHO SUBMETIDO A COMISSAO DE POS-GRADUACAO
DO INSTITUTO DE MATEMATICA, ESTATISTICA E CIENCIA
DE COMPUTACAO DA UNIVERSIDADE ESTADUAL DE CAMPINAS
PARA OBTENCAO DO TITULO DE MESTRE EM CIENCIA DE
COMPUTACAO.

UNICAMP
BIBLIOTECA CENTRAL

AGRADECIMENTOS:

SENDO DIFICIL CITAR OS NOMES DE TODOS QUE DIRETA OU INDIRETAMENTE
TORNARAM POSSIVEL A EXECUCAO DESTE TRABALHO, AGRADECO:

AO DR. ODELAR LEITE LINHARES, PELA ORIENTACAO E CHANCES DESDE 1964.

AO DR. UBIRATAN DIAMBROSIO, GRANDE INCENTIVADOR DESSA NOVA FASE DE TRABALHO.

AO DR. NELSON MACHADO, PELAS FACILIDADES OBTIDAS COM OS AVANCADOS RECURSOS
DO CPD-UNICAMP.

A DALVA, PELO INCENTIVO E APOIO MORAL.

AOS MEUS PAIS, POR TUDO.

R E S U M O

SLIP É UM SISTEMA DE PROCESSAMENTO DE LISTAS EM QUE CADA CELULA CONTEM UM DADO E DOIS APONTADORES, UM PARA A ESQUERDA, OUTRO PARA A DIREITA (PODENDO SER PENSADOS COMO PARA CIMA E PARA BAIXO, OU PARA TRAZ E PARA A FRENTE)

O NOME PROCESSADOR DE LISTAS SIMETRICAS SE DEVE AO FATO DE QUE É IRRELEVANTE O SENTIDO DE PERCURSO NA LISTA.

O SISTEMA É COMPOSTO DE UMA SERIE DE SUBROTINAS E FUNCOES EM FORTRAN E ALGUMAS FUNCOES PRIMITIVAS EM MACRO-10, QUE PERMITEM MANIPULAR OS CAMPOS ESPECIFICOS DAS CELULAS.

C O N T E U D O

I - INTRODUCAO

II - LISTAS: CONCEITOS, DEFINICOES E MECANISMOS

III - PROCESSAMENTO DE LISTAS SIMETRICAS

- 1) CRIACAO DA LISTA DE ESPACOS DISPONIVEIS (LAVS) E AS LISTAS PUBLICAS (WIS)
- 2) CRIACAO DE LISTAS VAZIAS.
- 3) INSERCAO DE INFORMACAO NA LISTA.
- 4) O MECANISMO DA READER - PERCURSO DE LISTAS.
- 5) OUTROS MODOS DE RECUPERAR INFORMACAO DE LISTA.
- 6) FUNCOES DE TESTE
- 7) PROCESSOS MISTOS.
- 8) RECURSAO

IV - OPERACOES LOGICAS, MANIPULACAO DE BITS E CARACTERES.

V - CONSIDERACOES SOBRE O SISTEMA SLIP

VI - BIBLIOGRAFIA - REFERENCIAS

VII - APENDICE - LISTAGENS DOS SUBPROGRAMAS

VIII - REFERENCIAS CRUZADAS - LISTA EM ORDEM ALFABETICA DOS SIMBOLOS GLOBAIS.

IX - INDICE DAS ROTINAS EM ORDEM ALFABETICA.

X - GLOSSARIO

XI - APLICACOES.

I - INTRODUCAO:

O USO DE COMPUTADORES PARA MANIPULACAO DE DADOS NUMERICOS E' BASTANTE CONHECIDO, E HISTORICAMENTE FOI O PRIMEIRO USO DE COMPUTADORES DIGITAIS DE PROGRAMA ARMAZENADO. A MANIPULACAO DE SIMBOLOS E' UMA AREA DE PESQUISA RELATIVAMENTE NOVA E QUE SE DESENVOLVEU RAPIDAMENTE NO CAMPO DO COMPUTADOR.

A MANIPULACAO DE CADEIAS DE CARACTERES (STRING-MANIPULATION) E PROCESSAMENTO DE LISTAS (LIST-PROCESSING) EXIGIU UMA DAS ALTERNATIVAS:

- 1) CRIACAO DE UMA NOVA LINGUAGEM DE PROGRAMACAO COM A VANTAGEM DE SER PROJETADA PARA EVITAR AS RESTRICOES DAS LINGUAGENS EXISTENTES E COM A SERIA DESVANTAGEM DAS DIFICULDADES DE IMPLEMENTACAO NOS VARIOS COMPUTADORES E A FALTA DE CONTINUIDADE COM AS LINGUAGENS EXISTENTES.
- 2) EXTENSAO DE UMA LINGUAGEM DE PROGRAMACAO EXISTENTE, QUE PODE SER CLASSIFICADA EM EXTENSAO HORIZONTAL E EXTENSAO VERTICAL. NA EXTENSAO HORIZONTAL, OS COMPILADORES SAO REESCRITOS PARA INSERIR AS VARIAS INSTRUcoes, TENDO A VANTAGEM DE PODER SER INSERIDOS NOVOS TIPOS DE DECLARACOES E ESTRUTURAS DE DADOS, E' O CASO DE FORMULA-ALGOL E FORMAC, POREM A REESCRITA DE UM COMPILADOR PARA CADA MAQUINA E' ONEROSA. NA EXTENSAO VERTICAL, TAMBEM CHAMADA DE "EMBEDDING", E' USADA UMA LINGUAGEM EXISTENTE PARA ESCREVER UMA OUTRA. OQUE EXPANDE AS CAPACIDADES E AO MESMO TEMPO RETEM AQUELAS DA LINGUAGEM HOSPITALEIRA (HOST LANGUAGE). A VANTAGEM DO EMBEDDING E' A FACILIDADE DE IMPLEMENTACAO, PORQUE NAO EXIGE UM COMPILADOR PROPRIO, E' O CASO DO SLIP E DYSIAL ESCRITOS EM FORTRAN E FLIP ESCRITO EM LISP. A DESVANTAGEM DO EMBEDDING E' DEVIDO AS RESTRICOES DA LINGUAGEM HOSPITALEIRA.

ALGOL E FORTRAN SAO AS ESCOLHAS LOGICAS COMO LINGUAGENS HOSPITALEIRAS, DESDE QUE SAO AS LINGUAGENS DE ALTO-NIVEL MAIS USADAS.

O ALGOL TEM DUAS FRAQUEZAS COMO LINGUAGEM HOSPITALEIRA: EM MUITOS COMPILADORES, SEUS PROCEDURES NAO PODEM SER COMPILADOS SEPARADAMENTE; (*) OUTRA DESVANTAGEM E' A FALTA DE UM SISTEMA GERAL DE ENTRADA/SAIDA.

O FORTRAN NAO TEM ESTES PROBLEMAS, PORTANTO TEM MAIOR POTENCIALIDADE PARA INSERCAO DE LINGUAGENS ESPECIAIS, A DESPELTO DA ELEGANCIA DOS PROGRAMAS EM ALGOL. A MAIOR FALHA DO FORTRAN RECAI NO PROBLEMA DE ALOCACAO DINAMICA DE MEMORIA E CAPACIDADE DE PROCESSAMENTO DE LISTAS.

NO SISTEMA SLIP E' BASEADO NA ALOCACAO DE MEMORIA DO IPL-V, SENDO QUE A MAIOR VANTAGEM DO SLIP SOBRE O IPL-V E' O USO DAS CAPACIDADES ARITMETICAS E DE E/S DO FORTRAN, ALEM DO GRANDE NUMERO DE PROGRAMAS E SUBROTINAS QUE PODEM SER USADAS.

A PAR DISSO, O SLIP TEM SIDO IMPLEMENTADO COMO EXTENSAO DAS LINGUAGENS FORTRAN, MAD E ALGOL.

* OBS: NAO E' O CASO DO ALGOL DO PDP-10.

. O SISTEMA SLIP:

SLIP (SYMMETRIC LIST PROCESSOR) É UM SISTEMA DE PROCESSAMENTO DE LISTAS NO QUAL, NORMALMENTE, CADA CELULA DA LISTA TEM UM DADO E DOIS APONTADORES (PARA A ESQUERDA E PARA A DIREITA), A SIMETRIA DECORRE DO FATO DE QUE O SENTIDO DE PERCURSO DA LISTA É IRRELEVANTE, E QUALQUER OPERAÇÃO QUE PODE SER EFETUADA NO TOPO DA LISTA (CELULA MAIS-A-ESQUERDA, OU CELULA MAIS-ACIMA), PODE SER EFETUADA NO FUNDO DA LISTA (CELULA MAIS-A-DIREITA, OU CELULA MAIS-ABAIXO).

PELO FATO DE SER BASEADO EM POUCAS FUNÇÕES PRIMITIVAS EM LINGUAGEM DE BAIXO NÍVEL (MACRO-10) E VÁRIOS SUBPROGRAMAS EM LINGUAGEM DE ALTO-NÍVEL (FORTRAN), O SISTEMA SLIP É ESPECIALMENTE ORIENTADO PARA SER UTILIZADO POR PROGRAMAS EM LINGUAGEM FORTRAN, QUE SEM DUVIDA AINDA É A LINGUAGEM DE ALTO NÍVEL MAIS UTILIZADA PARA COMPUTADORES DIGITAIS.

II - LISTAS: CONCEITOS E DEFINIÇÕES:

. LISTAS SIMÉTRICAS:

O MÓDULO FUNDAMENTAL DE INFORMAÇÃO COM O QUAL SLIP TRABALHA É A "CELULA", QUE É UM PAR DE PALAVRAS CONSECUTIVAS.

. A PRIMEIRA PALAVRA DA CELULA É PENSADA COMO SENDO DIVIDIDA EM 3 CAMPOS PRINCIPAIS:

- O CAMPO DO IDENTIFICADOR COM 2 BITS (ID): ESPECIFICA A NATUREZA DO ITEM ARMAZENADO NO CAMPO DE DADO NA SEGUNDA PALAVRA DA CELULA.
- O CAMPO DE APONTADOR A ESQUERDA, COM 16 BITS (LNKL **)
- O CAMPO DE APONTADOR A DIREITA, COM 16 BITS (LNKR **)

. A SEGUNDA PALAVRA DA CELULA (CHAMADA DE CAMPO DE DADO OU CAMPO DE INFORMAÇÃO = INFO) É FREQUENTEMENTE PENSADA COMO SENDO DADO BRUTO, I.E., NÃO SENDO SUBDIVIDIDA EM SUBCOMPONENTES, MAS QUANDO ELA É DIVIDIDA, TEM AS MESMAS DEFINIÇÕES DE CAMPO DA PRIMEIRA PALAVRA DA CELULA.

** OBS: VER CONSIDERAÇÕES SOBRE ENDEREÇOS NA SEÇÃO V.

FUNCOES PRIMITIVAS:

AS FUNCOES PRIMITIVAS QUE EXTRAEM INFORMACAO CONTIDAS NOS CAMPOS DE UMA CELULA SAO FUNCOES COM VALOR INTEIRO DA INFORMACAO CONTIDA NO CAMPO:

0

1. ID(CELL) EXTRAI A PORCAO ID DA CELULA CELL.
2. LNKL(CELL) EXTRAI O ENDEREÇO DE MAQUINA ARMAZENADO NO CAMPO DE APONTADOR ESQUERDO DE CELL.
3. LNKR(CELL) EXTRAI O ENDEREÇO DE MAQUINA ARMAZENADO NO CAMPO DE APONTADOR DIREITO DE CELL.

DUAS FUNCOES PRIMITIVAS COM FUNCIONAMENTO IDENTICO, POREM COM MODO DIFERENTE, PARA EVITAR OS PROBLEMAS INHERENTES AS CONVENCÕES DE PONTO-FIXO/PONTO-FLUTUANTE DO FORTRAN.

4. CONT(A) SAO FUNCOES QUE TEM COMO VALOR A INFORMACAO ARMAZENADA NA PALAVRA CUJO ENDEREÇO ESTA EM A.
5. INHAL(A)

A FUNCAO PRIMITIVA:

6. MADOV(CELL) E' A FUNCAO INTEIRA QUE TEM COMO VALOR O ENDEREÇO DE MAQUINA DA CELULA CELL.

AS DUAS PRIMITIVAS QUE SERVEM PARA ARMAZENAR INFORMACAO DOS TRES CAMPOS NUMA PALAVRA, SAO:

- 7 *. SETDIR(I,L,R,CELL) QUE ARMAZENA I NO CAMPO ID, L NO CAMPO LNKL, R NO CAMPO LNKR, RESPECTIVAMENTE, DA CELULA CELL. SE ALGUM DESSES TRES PARAMETROS TEM O VALOR -1, O CAMPO CORRESPONDENTE DE CELL NAO E' ALTERADO.
- 8 *. SETIND(J,L,R,A) TEM O MESMO EFEITO DE SETDIR, EXCETO QUE A CELULA A SER ALTERADA E' AQUELA CUJO ENDEREÇO APARECE COMO UM INTEIRO EM A.

* OBS: I MOD 4, L MOD 65536, R MOD 65536.

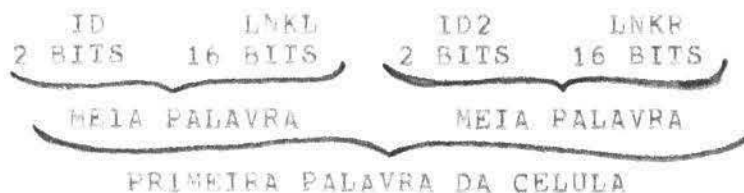
DUAS PRIMITIVAS QUE ARMAZENAM PALAVRAS COMPLETAS EM CELULAS ESPECIFICAS:

9. STRDIR(DATUM,CELL) QUE E' UMA FUNCAO QUE TEM DATUM COMO SEU VALOR, E ARMAZENA DATUM EM CELL. ESTA PRIMITIVA EXISTE PARA EVITAR DIFICULDADES DE CONVERSÃO INTEIRA/REAL DO FORTRAN, MAS TEM A VANTAGEM ADICIONAL QUE ELA PODE SER USADA ENCADEADA, COMO EM:
CALL STRDIR(STRDIR(0,N),X) QUE COLOCA 0 EM N E EM X.
10. STRIND(DATUM,A) QUE E' UMA FUNCAO QUE TEM DATUM COMO SEU VALOR, E QUE REALIZA A OPERACAO DE ARMAZENAR DATUM NA CELULA CUJO ENDEREÇO APARECE COMO UM INTEIRO EM A.

OBS: CONSIDERANDO-SE QUE A PALAVRA DO SISTEMA DEC-10 TEM 36 BITS, E EXISTEM VARIAS INSTRUÇÕES PARA TRATAMENTO DE MEIA PALAVRA, ADOTANDO O CAMPO ID COM 2 BITS, LNKL COM 16 BITS, E LNKR COM 16 BITS (POR COMPATIBILIDADE), SOBRA 2 BITS QUE PODEM SER USADOS, POR EXEMPLO, PARA MARCAR O TIPO DE DADO QUE ESTA CONTIDO NA SEGUNDA PALAVRA DE UMA CELULA DE DADO (COM ID=0, CELULA CHAMADA DE TIPO "E" OU ELEMENTO) SE SE DESEJA ARMAZENAR DADOS DE MODO MISTO NA LISTA:

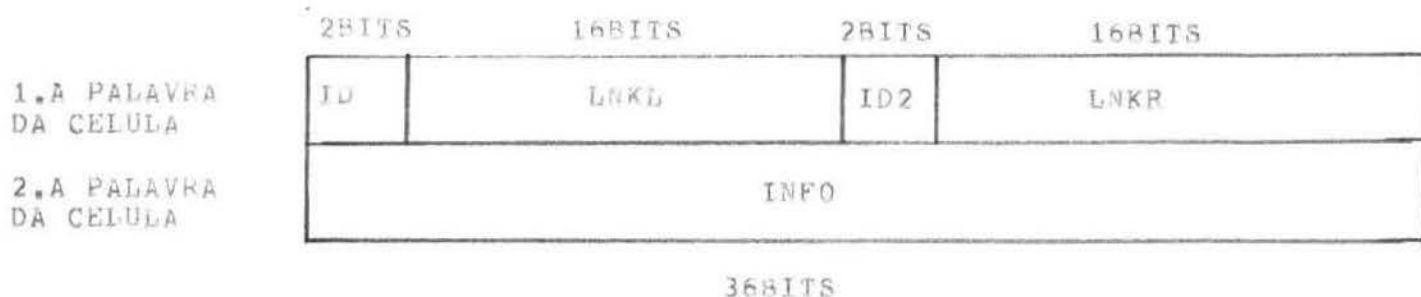
ID2=0 QUALQUER TIPO DE DADO
 ID2=1 DADO ALFANUMÉRICO
 ID2=2 DADO INTEIRO
 ID2=3 DADO REAL; OU

COMO "TAG'S", QUE SÃO MUITO ÚTEIS PARA MANIPULAÇÃO DE LISTAS, EM PARTICULAR, PARA TRATAMENTO DE ARVORES BINARIAS QUE PODEM SER CRIADAS COM A ESTRUTURA PROPOSTA; FORAM FEITAS VARIAS OUTRAS FUNÇÕES E SUBROTINAS PARA TER ACESSO AOS 2 BITS QUE SOBRA NA SEGUNDA METADE DA PRIMEIRA PALAVRA. TEMOS, ENTÃO, NA PRIMEIRA PALAVRA DA CELULA, OS SEGUINTE CAMPOS:



11. SUBROTINA SETID2(I,CELL)
 COLOCA O CONTEUDO DE I MOD 4 COMO CONTEUDO DO CAMPO ID2 EM CELL.
12. FUNÇÃO ID2(CELL)
 CUJO VALOR É UM NÚMERO INTEIRO CONTIDO NO CAMPO ID2 DA CELULA CELL.
13. SUBROTINAS SETLTAG(I,CELL) E SETRTAG(I,CELL)
 QUE ESTABELECEM, RESPECTIVAMENTE, LTAG E RTAG NA CELULA CELL.
14. FUNÇÕES RTAG(CELL) E LTAG(CELL)
 TEM COMO VALOR BOOLEANO O TAG CORRESPONDENTE:
 ID2(2BITS) = LTAG(1BIT),RTAG(1BIT)
 O VALOR INTEIRO: -1(10) = 777777777777(8) = TRUE
 0(10) = 000000000000(8) = FALSE

CAMPOS DE UMA CELULA NORMAL:



O ENDEREÇO DE UMA CELULA É SEMPRE O ENDEREÇO DA PRIMEIRA PALAVRA DA CELULA.
 O PROGRAMA PRINCIPAL - DECLARAÇÕES, INICIALIZAÇÃO.

O PROGRAMADOR FORTRAN QUE TENCIONA UTILIZAR O SISTEMA SLIP DEVE TER EM MENTE OS SEGUINTEs PONTOS:

1) O PROGRAMADOR DEVE DECLARAR NO PROGRAMA PRINCIPAL UM ARRAY UNI-DIMENSIONAL "SPACE" COM UM NUMERO N PAR DE PALAVRAS PARA SER CRIADA A LISTA DE ESPACOS DISPONIVEIS (LAVS). O ARRAY PODE SER DECLARADO POR UM COMANDO DIMENSION, INTEGER, OU REAL, O ARRAY PODE TER DIMENSAO ATE 65534 (2**16-2) O QUE FORNECE UMA AREA GRANDE DE TRABALHO PARA LISTAS, 32K PARA INFORMACAO E 32K PARA APONTADORES (VER CONSIDERACOES NA SECAO V).

2) O PROGRAMADOR DEVE DECLARAR OS SEGUINTEs COMANDOS (OU ANALOGOS):

```
COMMON /XTRCT/ CP(5),BLANK,NULL,ZERO,LP,RP
COMMON AVSL, N(100)
```

TODAS AS CELULAS SLIP PODEM SER REMOVIDAS E DEVOLVIDAS A LISTA DE ESPACOS DISPONIVEIS (LAVS) QUANDO NECESSARIO (FUNCOES NUCELL E RCELL).

NO INICIO DE EXECUCAO DE UM PROGRAMA, DEVE SER INICIALIZADA A LISTA DE ESPACOS DISPONIVEIS (LAVS) E AS LISTAS DE PUBLICAS (W'S) PELA SUBROTINA INITAS NAS ETAPAS INTERMEDIARIAS DE EXECUCAO DO PROGRAMA, CADA CELULA OU ESTA' EM USO ATIVO OU ESTA' NA LAVS.

UMA CELULA QUE ESTA' EM USO ATIVO É UMA READER, UMA HEADER DE LISTA, OU UM MEMBRO DE UMA LISTA UTIL.

UMA CELULA QUE ESTA' NA LAVS, PODE SER UM MEMBRO DIRETO DA LAVS OU SER UM ELEMENTO DE UMA LISIA QUE É SUBLISTA DA LAVS.

3) O PROGRAMADOR FORTRAN DEVE TOMAR CUIDADO COM A CONVERSÃO AUTOMÁTICA DE MODOS INTEIRO/REAL, NO CASO DE SE FAZER UMA ATRIBUIÇÃO DE VALOR, FICANDO PROIBIDO O SEGUINTE COMANDO: L=ADVLNR(R5) POIS A FUNÇÃO ADVLNR É UMA FUNÇÃO COM VALOR EXPLÍCITO REAL E "L" É UMA VARIÁVEL INTEIRA, NA TENTATIVA DE CONVERSÃO REAL-INTEIRA, PROVAVELMENTE HAVERÁ UM ESTOURO DE INTEIRO. POR OUTRO LADO, O PROGRAMADOR NÃO PRECISA (EM GERAL) SE PREOCUPAR COM OS MODOS DOS ARGUMENTOS DE SUBPROGRAMAS, POIS TODO CUIDADO JA FOI TOMADO PARA EVITAR PROBLEMAS DE CONVERSÃO AUTOMÁTICA. UMA VARIÁVEL PODE TER VALOR INTEIRO, E TER NOME REAL, E VICE-VERSA.

PARA EVITAR A MISTURA DE MODOS EXISTEM AS FUNÇÕES PRIMITIVAS: CONT, INHALI, STRDIR E STRIND, DESCRITAS NA SECAO II.

• TIPOS DE CELULAS:

EXISTEM BASICAMENTE 4 TIPOS DE CELULAS, DEPENDENDO DO CONTEUDO DO CAMPO DE INFORMACAO NA SEGUNDA PALAVRA DA CELULA.

- O CAMPO DE INFORMACAO PODE SER O NOME DE UMA LISTA.
- O CAMPO DE INFORMACAO PODE SER UM DADO.
- O CAMPO DE INFORMACAO PODE TER APONTADORES E CONTADORES.

O CAMPO ID DA PRIMEIRA PALAVRA DA CELULA, IDENTIFICA O TIPO DE CELULA:

ID	CONTEUDO DO CAMPO DE INFORMACAO
BIN, INT, 00 0	INFO E' DADO ARMAZENADO
01 1	INFO E' NOME DE LISTA
10 2	A CELULA E' HEADER DE UMA LISTA
11 3	A CELULA E' READER DE UMA LISTA.

CELULA TIPO 0: DADO NAO E' NOME. (CELULA CLASSE E : ELEMENTO)

0	LNK L	0	LNK R
D A D O			

ID=0
LNKL: APONTA PARA A CELULA A ESQUERDA
LNKR: APONTA PARA A CELULA A DIREITA
DADO: CAMPO DE INFORMACAO, DADO.

CELULA TIPO 1: CAMPO DE INFORMACAO E' NOME DE UMA LISTA.
(CELULA CLASSE E : NOME)

1	LNK L	0	LNK R
NOME-DE-LISTA			

ID=1
LNKL: APONTA PARA A CELULA A ESQUERDA
LNKR: APONTA PARA A CELULA A DIREITA
NOME-DE-LISTA NO FORMATO ESPECIAL

CELULA TIPO 2: CABECA-DE-LISTA (HEADER)

2	LNK L	0	LNK R
M	L.D E S C	0	L.C N T R

ID=2
LNKL: APONTA PARA O FUNDO DA LISTA
LNKR: APONTA PARA O TOPO DA LISTA
M: MARCA DA LISTA (0, 1, 2, OU 3)
L.DESC: APONTA PARA LISTA DESCRITIVA
LCNTR: CONTADOR DE REFERENCIAS

CELULA DO TIPO 3: LETTORA (READER)

3	LNTR	0	LNKR
0	LHEAD	0	LCNTR

ID=3
LNTR: APONTA PARA UMA CELULA
LNKR: 0 (PODE CONTER O ENDEREÇO DA PROXIMA READER NA PILHA DE READER'S
LHEAD: APONTA PARA A HEADER DA LISTA
LCNTR: CONTADOR DE NIVEL DA LETTORA NA ESTRUTURA DE LISTAS.

• DESCRICAO DO SISTEMA SLIP:

UMA LISTA SLIP (DORAVANTE CHAMADA SIMPLEMENTE DE LISTA) E' UMA COLECAO DE CELULAS LIGADAS, TAIS QUE, O CAMPO LNKR DE CADA CELULA APONTA PARA (I.E., CONTEM O ENDEREÇO DE MAQUINA DA) PROXIMA CELULA A DIREITA (OU A PROXIMA-ABAIXO) NA LISTA. O CAMPO LNKL DE CADA CELULA APONTA PARA A PROXIMA CELULA A ESQUERDA (OU A PROXIMA-ACIMA) NA LISTA.

EM CADA LISTA EXISTE UMA UNICA CELULA COM ID=2, ESTA CELULA E' CHAMADA CABECA-DE-LISTA (HEADER) DESTA LISTA.

O CORPO DA LISTA E' A COLECAO DE TODAS AS CELULAS PERTENCENTES A LISTA, EXCETO A CELULA HEADER.

UMA CELULA QUE CONTENHA NOS CAMPOS LNKL E LNKR O ENDEREÇO DA CELULA QUE E' HEADER DE UMA LISTA E' DITA CONTER O NOME DAQUELA LISTA. UMA VARIÁVEL SIMBOLICA DO PROGRAMA CONTENDO O NOME DE UMA LISTA PODE SER CONSIDERADA UM "ALIAS" PARA O NOME DA LISTA. UMA LISTA PODE TER UM NUMERO ARBITRARIO DE ALIAS'ES. (MAIORES DETALHES SOBRE ALIAS'ES NA SECAO III.2)

• A CABECA-DE-LISTA

O CAMPO LNKR DA PRIMEIRA PALAVRA DA CELULA HEADER E' DITO APONTAR PARA O TOPO, E O CAMPO LNKL DA MESMA PALAVRA E' DITO APONTAR PARA O FUNDO DA LISTA ASSOCIADA AQUELA HEADER.

O CAMPO LNKR DA SEGUNDA PALAVRA DA CELULA HEADER E' CHAMADO CONTADOR DE REFERENCIAS DA CABECA-DA-LISTA. SUA FUNCAO E' CONTAR O NUMERO DE LISTAS QUE COMPARTILHAM AQUELA LISTA (ASSOCIADA AQUELA HEADER) COMO SUBLISTA.

A CELULA CHAMADA CABECA-DE-LISTA OU HEADER TEM AS SEGUINTE INFORMACOES:

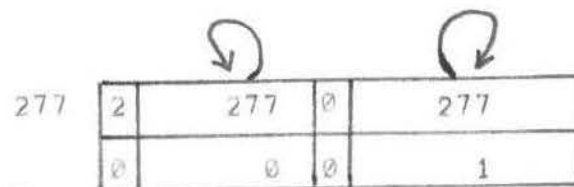
- 1.A PALAVRA DA CELULA:
 - ID=2
 - LNKL= ENDEREÇO DO FUNDO DA LISTA
 - LNKR=ENDEFEÇO DO TOPO DA LISTA.
- 2.A PALAVRA DA CELULA:
 - ID= MARCA DE LISTA (COMUMENTE 0)
 - LNKL= LISTA DESCRITIVA
 - LNKR= CONTADOR DE REFERENCIAS DA LISTA, E' A CONTAGEM DO NUMERO DE LISTAS DAS QUAIS A LISTA E' SUBLISTA (PARA PROPOSITOS DO GARBAGE COLLECTOR)

• A LISTA VAZIA:

NUMA LISTA VAZIA (CRIADA PELA FUNCAO LIST) A HEADER TEM A SEGUINTE INFORMACAO:

- 1.A PALAVRA DA CELULA
 - ID=2
 - LNKL=ENDEFEÇO DA HEADER
 - LNKR=ENDEFEÇO DA HEADER
- 2.A PALAVRA DA CELULA
 - ID=0
 - LNKL=0
 - LNKR=0 OU CONTADOR DE REFERENCIAS

EXEMPLO DE UMA LISTA VAZIA:

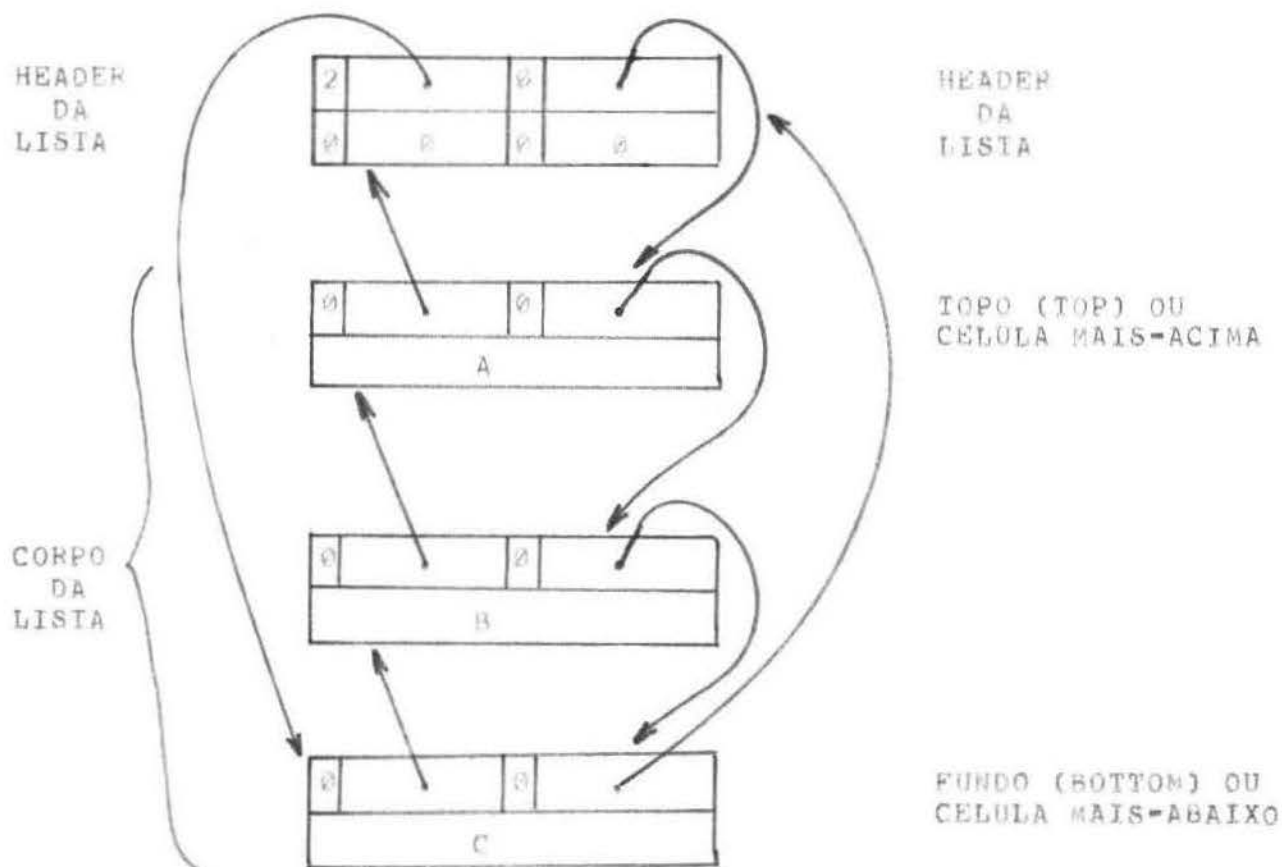


EXEMPLO DE UMA LISTA SIMPLES: (A, B, C)

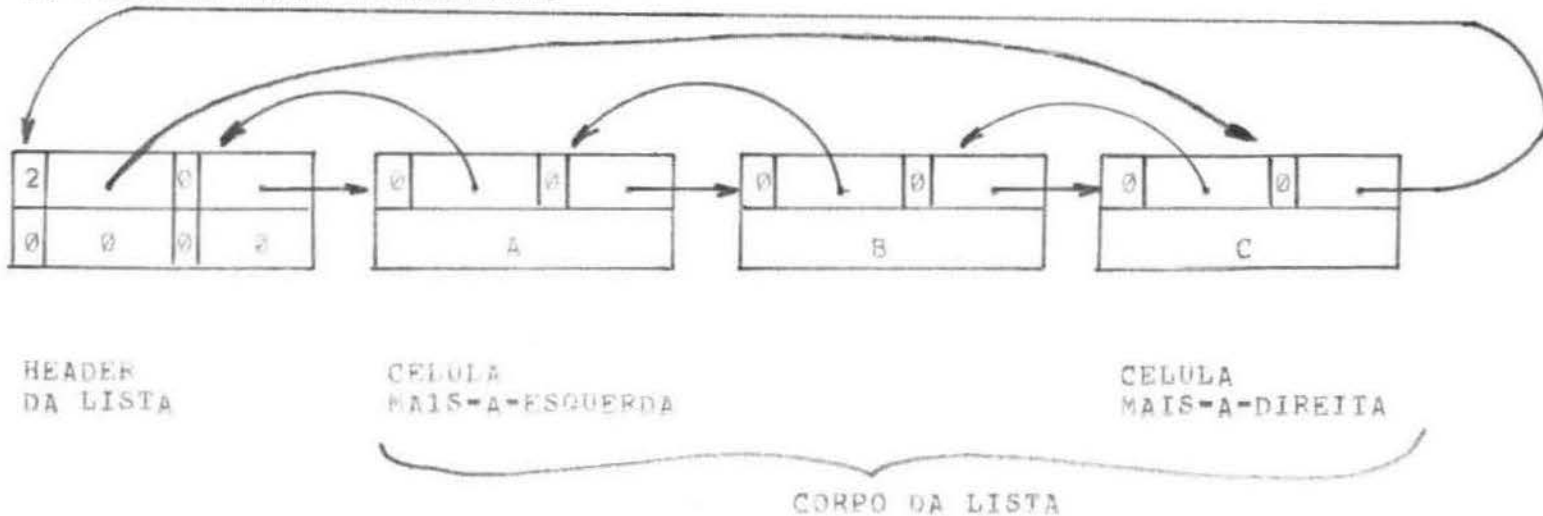
(A, B, C)

REPRESENTAÇÕES GRÁFICAS E NOMENCLATURA:

A) REPRESENTAÇÃO VERTICAL



B) REPRESENTAÇÃO HORIZONTAL:

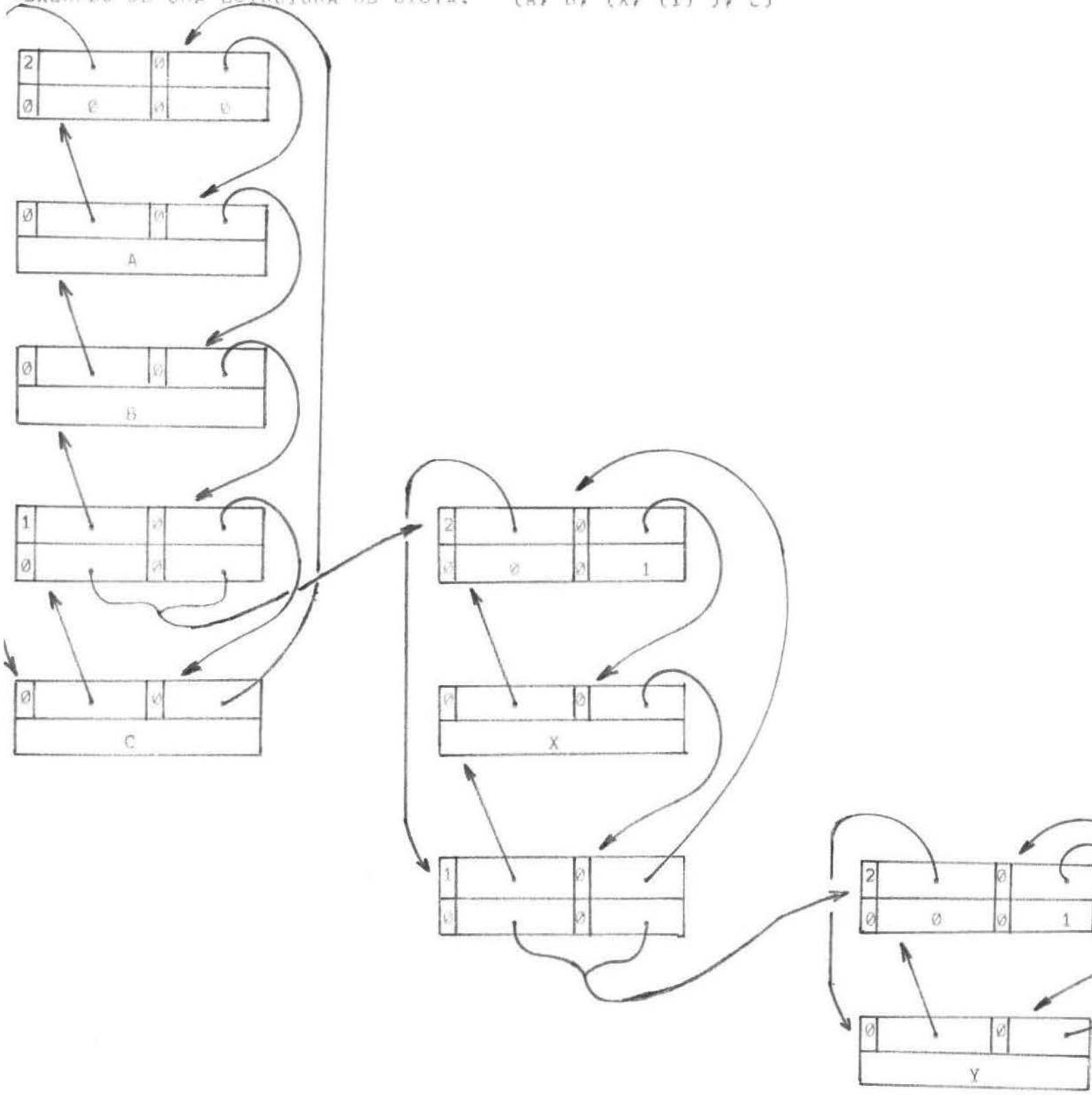


ESTRUTURA DE LISTAS:

UMA ESTRUTURA DE LISTAS É UM CONJUNTO DE LISTAS, TAIS QUE, UMA LISTA (CHAMADA LISTA PRINCIPAL) TEM TODAS AS OUTRAS LISTAS COMO SUBLISTAS, OU SUBLISTAS DE SUAS SUBLISTAS, ETC.

UMA LISTA S É DITA SFR SUBLISTA DE UMA LISTA L, SE O NOME DE S É CONTIDO NUM CAMPO DE DADO (NA SEGUNDA PALAVRA) DE UMA CELULA DA LISTA L E O CAMPO ID DAQUELA CELULA VALE 1.

EXEMPLO DE UMA ESTRUTURA DE LISTA: (A, B, (X, (Y)), C)



. RESPONSABILIDADE DO SISTEMA NA DEVOLUCAO DE CELULAS A LISTA LIVRE:

UMA DAS MAIORES POTENCIAS DOS SISTEMAS DE PROCESSAMENTO DE LISTAS E' O FATO DE SER POSSIVEL A CRIACAO E MANUTENSAO DE ESTRUTURAS DE LISTAS, I.E., INSERIR O NOME DE UMA LISTA EM OUTRA LISTA E MANIPULAR A ESTRUTURA RESULTANTE COMO UMA UNICA ENTIDADE. ENTRETANTO, DE UM MODO OU OUTRO, TODO PROCESSADOR DE LISTAS ENFRENTA O PROBLEMA DE ASSUMIR A RESPONSABILIDADE DE CONTROLAR A DEVOLUCAO DE CELULAS PARA A LISTA-LIVRE, PROBLEMA ESTE QUE APARECE NA TENTATIVA DE RESPONDER A QUESTAO:

"QUEM ESTA AUTORIZADO A DEVOLVER UM DADO PARA A LISTA-LIVRE, QUANDO UMA CELULA CONTENDO O DADO E' DEVOLVIDA ?"

A RAZAO DISSO E' QUE CERTOS DADOS PODEM SER COMPARTILHADOS POR VARIAS LISTAS SIMULTANEAMENTE.

EM SLIP, O PROBLEMA APARECE SOMENTE NO FATO DE QUE UMA LISTA PODE SER UMA SUBLISTA DE VARIAS LISTAS.

. COMPARTILHAMENTO DE SUBLISTAS:

EM SLIP UMA LISTA PODE SER SUBLISTA DE VARIAS LISTAS. DIFERENTEMENTE DE OUTROS SISTEMAS DE PROCESSAMENTO DE LISTAS, O SLIP ARMAZENA DADO NO CAMPO DE INFORMACAO, AO INVES. DE ARMAZENAR O ENDEREÇO DO DADO. NO SLIP, SE O PROGRAMADOR DESEJA ARMAZENAR O ENDEREÇO DO SIMBOLO NO CAMPO DE DADO, ESTE ENDEREÇO E' CONSIDERADO UM DADO, PARA TODO EFEITO. POREM, QUANDO A LISTA E' "APAGADA", I.E., SUAS CELULAS SAO DEVOLVIDAS PARA LAVS, DEVE SER CONSIDERADO SE SUAS SUBLISTAS TAMBEM PODEM SER APAGADAS.

O MECANISMO DE GARBAGE COLLECTION - LAVS; LISTA DE ESPACOS DISPONIVEIS

A SUBROTINA INITAS(SPACE,N) ESTABELECE AS LIGACOES NECESSARIAS NO ARRAY SPACE DIMENSIONADO NO PROGRAMA PRINCIPAL COM UM NUMERO N PAR DE PALAVRAS TAIS QUE, O CAMPO LNKR DE CADA PALAVRA DE INDICE IMPAR, CONTEM O ENDEREÇO DA PROXIMA PALAVRA DE INDICE IMPAR, EXCETO A (N-1)-ESIMA PALAVRA QUE FICA COM LNKR=0 INDICANDO O FIM DA LAVS.

TODOS OS OUTROS CAMPOS DAS PALAVRAS NO ARRAY SAO ZERADOS. UMA POSICAO ESPECIAL AVSL, DECLARADA EM COMMON AVSL,W(100) NO PROGRAMA PRINCIPAL TEM LNKR CONTENDO O ENDEREÇO DA PRIMEIRA PALAVRA DA LAVS E LNKL CONTENDO O ENDEREÇO DA ULTIMA CELULA DE LAVS, I.E., O ENDEREÇO DA (N-1)-ESIMA PALAVRA DE SPACE. LAVS ASSIM CRIADA E' UMA LISTA LINEAR.

NUCELL E' A FUNCAO QUE RETIRA UMA CELULA DA LAVS PARA TORNA-LA DISPONIVEL, A PRIMEIRA CELULA DE LAVS TEM O ENDEREÇO CONTIDO NO LNKR DE AVSL, O VALOR DA FUNCAO NUCELL E' ESTE ENDEREÇO, E O NOVO LNKR DE AVSL E' O ENDEREÇO DA PROXIMA CELULA A DIREITA DA CELULA LIBERADA, I.E., E' O ANTIGO CONTEUDO DE SEU LNKR. A CELULA RETIRADA DA LAVS E' ZERADA PARA ASSEGURAR SUA BOA MANIPULACAO POR OUTRAS ROTINAS.

(UMA CELULA E' RETIRADA DO TOPO DA LISTA LAVS)

RCELL E' A FUNCAO QUE DEVOLVE UMA CELULA PARA LAVS, INSERINDO-A NO FUNDO DA LISTA, COLOCANDO A CELULA DEVOLVIDA COMO SUCESSORA DO ATUAL FUNDO DE LAVS, E ATUALIZANDO O LNKL DE AVSL, PARA APONTAR PARA O NOVO FUNDO DA LAVS ALEM DE ZERAR O CAMPO LNKR DA CELULA DEVOLVIDA PARA INDICAR O FIM-DA-LISTA. (A DEVOLUCAO DE UMA CELULA A COLOCA NO FUNDO DA LISTA LAVS)

A REMOCAO E DEVOLUCAO DE CELULAS PARA LAVS A MANTEM COMO UM CONJUNTO DE PARES DE PALAVRAS LIGADAS.

5 1

• DEVOLUCAO DE UMA LISTA PARA LAVS ("BULK ERASURE") - MTLIST :

PARA DEVOLVER TODA UMA LISTA PARA LAVS, O APONTADOR DE PROXIMO-ABAIXO DO FUNDO DE LAVS (QUE ERA ZERO) PASSA A APONTAR PARA A PRIMEIRA CELULA DA LISTA, E O LNKL DE AVSL QUE APONTAVA PARA ESTE FUNDO, PASSA A APONTAR PARA O FUNDO DA LISTA DEVOLVIDA, E FINALMENTE O LNKR DO NOVO FUNDO E' ZERADO, PARA INDICAR O FIM DA LISTA LAVS.

UMA VANTAGEM DESTES MECANISMO DE DEVOLUCAO DE LISTA PARA A LISTA LIVRE E' QUE O TEMPO DE DEVOLUCAO E' INDEPENDENTE DO TAMANHO DA LISTA APAGADA.

QUANDO UMA LISTA QUE CONTEM O NOME DE OUTRA LISTA, E' DEVOLVIDA PARA LAVS VEM O PROBLEMA DE SABER SE ESTA SUBLISTA TAMBEM PODE SER DEVOLVIDA A LAVS OU SE ELA E' COMPARTILHADA POR OUTRAS LISTAS, ESTA SUBLISTA PODE SER TAMBEM APAGADA?

NO SLIP, PARA A INDICACAO DE QUE UMA LISTA S E' SUBLISTA DE UMA LISTA L, HA' UMA CELULA NO CORPO DA LISTA PRINCIPAL L, QUE CONTEM NA SEGUNDA PALAVRA O NOME-DE-LISIA DA LISTA S (LNKL=LNKR=ENDERECO DO HEADER DA SUBLISTA). COMO A REMOCAO DE CELULAS DA LAVS E' FEITA UMA POR VEZ, NUCELL TEM A OPORTUNIDADE DE VERIFICAR ESTAS CONDICOOES. NUCELL TESTA O CAMPO ID, SE ID FOR 1, ENTAO ENCONTROU NA LAVS UMA CELULA CONTENDO O NOME DE UMA SUBLISTA, O CONTADOR DE REFERENCIAS CONTIDO NO HEADER DA SUBLISTA, INDICA O NUMERO DE LISTAS QUE COMPARTILHAM ESTA LISTA SENDO ENTAO DECREMENTADO DE UM E SE A CONTAGEM SE ANULA, A SUBLISTA E' DEVOLVIDA PARA LAVS PELO MECANISMO DE DEVOLUCAO DE LISTA INTEIRA. QUALQUER SUBLISTA DESTA LISTA DEVOLVIDA PROVAVELMENTE PODERA SER APAGADA PELO MESMO MECANISMO. PELO FATO DE QUE A DEVOLUCAO DE UMA LISTA A COLOCA NO FUNDO DA LISTA LIVRE, E A REMOCAO DE UMA CELULA E' DO TOPO DA LISTA LIVRE, O PROBLEMA DE DEVOLUCAO DE SUBLISTAS E' DEIXADO PARA MAIS TARDE POSSIVEL, EVITANDO PROBLEMAS DE DECISAO DE "DELECAO" ANTES DO TEMPO NECESSARIO.

O FUNCIONAMENTO CORRETO DA DEVOLUCAO DE SUBLISTAS DEPENDE DO CONTADOR DE REFERENCIAS DO HEADER DA LISTA, QUE E' INCREMENTADO QUANDO UMA LISTA SE TORNA SUBLISTA DE OUTRA.

AS DUAS FUNCOES DE INSERCAO DE DADOS NUMA LISTA (NXILFT E NXTRGT) TEM A RESPONSABILIDADE DE TESTAR SE ESTA CELULA A SER INSERIDA E' O NOME DE UMA LISTA, SE O FOR, DEVE INCREMENTAR O CONTADOR DE REFERENCIAS DE TAL LISTA.

O CONTADOR DE REFERENCIAS E' DECREMENTADO SOMENTE QUANDO HA NECESSIDADE DE SE REMOVER UMA CELULA TIPO 1 DA LAVS, QUE CONTEM O NOME DE TAL SUBLISTA.

• CONTROLE PELO PROGRAMADOR PARA EVITAR "DELECAO" DE LISTAS.

POR ESTA TECNICA DE "DELECAO", UMA LISTA INTEIRA PODE SER APAGADA, SEM UM COMANDO DIRETO DO PROGRAMADOR. O PROGRAMADOR PODERA CONTROLAR ESTA "DELECAO" EMPILHANDO E DESEMPILHANDO O NOME DESTA LISTA EM ALGUMA LISTA TEMPORARIA A FUNCAO LIST(K) TEM UM MECANISMO QUE RESOLVE ESTE PROBLEMA.

O OBJEITVO DA FUNCAO LIST(K) E' CRIAR UMA LISTA VAZIA, DEIXANDO O NOME DA LISTA CRIADA COMO VALOR DA FUNCAO. QUANDO O PARAMETRO K DE ENTRADA DA FUNCAO NAO TEM VALOR 9, O NOME DA LISTA CRIADA TAMBEM E' DEIXADO NO PARAMETRO K E O CONTADOR DE REFERENCIAS DA LISTA E' ESTABELECIDO 1, SE K=9, O CONTADOR DE REFERENCIA DA LISTA CRIADA E' ZERO E SEU NOME E' DEIXADO SOMENTE COMO VALOR DA FUNCAO.

EXEMPLO: CALL LIST(L5) PROVOCA QUE L5 SEJA O ALIAS SIMBOLICO DA LISTA VAZIA CRIADA E FICA ACESSIVEL AO PROGRAMADOR. COMO LCMTR=1, ESTA LISTA NAO SERA APAGADA DURANTE A "DELECAO" DE UMA LISTA ONDE O SEU NOME APARECE. FICANDO SOBRE TOTAL CONTROLE DO PROGRAMADOR A "DELECAO" EXPLICITA E INTENCIONAL DA LISTA L5 PARA LAYS, COM O USO DA FUNCAO IRALST(L5).

AS LISTAS PUBLICAS - RECURSAO.

ADICIONALMENTE A SUBROTINA INITAS CRIA UM CONJUNTO DE 100 LISTAS VAZIAS COM ALIAS'ES W(1),W(2),...W(100) CHAMADAS "LISTAS PUBLICAS", QUE POR MEIO DO COMANDO COMMON AVSL,W(100) SEUS ALIAS'ES PODEM SER ACESSADAS POR TODOS OS SUBPROGRAMAS QUE LHE FIZEREM REFERENCIA. AS LISTAS PUBLICAS TEM CONTADOR DE REFERENCIAS ELEVADO (4095) PARA EVITAR SUA "DELECAO" INADVERTIDAMENTE. O PRINCIPAL USO DAS LISTAS PUBLICAS E' NO PROCESSO DE RECURSAO, ONDE ELAS SERVEM COMO VEICULOS PARA A COMUNICACAO DE PARAMETROS.

O MECANISMO VISIT-TERM PARA POSSIBILITAR A RECURSAO.

AS SUBROTINAS BASICAS VISIT E TERM EM LINGUAGEM DE MONTAGEM MACRO-10 TORNAM POSSIVEL A RECURSAO. QUALQUER COMANDO DENTRO DO SUBPROGRAMA PARA O QUAL O CONTROLE E' PARA SER TRANSFERIDO RECURSIVAMENTE, I.E., POR MEIO DO MECANISMO DE VISIT-TERM, DEVE-SE ASSINALAR UM LABEL SIMBOLICO POR MEIO DE UM COMANDO ASSIGN DO FORTRAN.

FUNDAMENTALMENTE, VISIT EMPILHA O ENDEREÇO NORMAL DE RETORNO (QUE FORTRAN COMUNICA A VISIT) NUMA PILHA INTERNA, NUMA CELULA QUE E' FORNECIDA PELA NUCELL, E TRANSFERE O CONTROLE PARA O COMANDO CUJO NOME SIMBOLICO FOI FORNECIDO COMO PARAMETRO DE ENTRADA. TERM, POR OUTRO LADO, DESEMPILHA DA MESMA LISTA INTERNA E TRANSFERE O CONTROLE PARA O COMANDO CUJO ENDEREÇO DE MAQUINA ESTAVA NO TOPO DA PILHA. A CELULA DESEMPILHADA E' DEVOLVIDA PARA LAYS POR MEIO DA FUNCAO RCELL.

O PROBLEMA DA RECURSAO NAO E' SOMENTE O DO EMPILHAMENTO DO ENDEREÇO DE RETORNO, MAS TAMBEM O MEIO DE COMUNICACAO E O EMPILHAMENTO DOS PARAMETROS. VISIT E TERM SAO FUNCOES COM 2 PARAMETROS, SENDO QUE O SEGUNDO DELES, RESPECTIVAMENTE, EMPILHA E DESEMPILHA PARAMETROS NAS LISTAS PUBLICAS. A OPERACAO DE EMPILHAR PARAMETROS NAS LISTAS PUBLICAS E' REALIZADO PELAS SUBROTINAS PARMT2(A1,A2), PARMT4(A1,A2,A3,A4), ETC. ENQUANTO A OPERACAO DE DESEMPILHAMENTO E' REALIZADA PELA RESTOR(N), QUE RESTAURA AS CELULAS DO TOPO DE CADA UMA DAS N PRIMEIRAS LISTAS PUBLICAS. UM EXEMPLO DE USO DAS ROTINAS PARMTN E':

```
ASSIGN 10 TO LOCO
X=VISIT(LOCO,PARMT2(A,B))
```

PARMT2 EMPILHA "A" NO TOPO DE W(1), "B" NO TOPO DE W(2), TRANSFERINDO O CONTROLE AO COMANDO 10 QUE FOI LOGICAMENTE ASSIGNADO AO "LABEL" SIMBOLICO "LOCO" E EVENTUALMENTE RETORNANDO PARA A OPERACAO DE SUBSTITUICAO DE VALOR DE "X". O VALOR DE VISIT SERA DETERMINADO PELA SUBROTINA TERM. O PROCESSAMENTO QUE COMECA NO COMANDO ROTULADO LOCO, DEVERA' ACHAR OS PARAMETROS DE ENTRADA NO TOPO DE W(1) E W(2), RESPECTIVAMENTE.

TERM E' UMA FUNCAO COM DOIS PARAMETROS DE ENTRADA, O PRIMEIRO DOS QUAIS E' SEU VALOR, ENQUANTO O SEGUNDO E' SIMPLEMENTE UMA CHAMADA PARA A SUBROTINA RESTOR(N) PARA RECUPERACAO DAS CELULAS DOS TOPOS DAS LISTAS PUBLICAS W(1),W(2),...W(N). ASSIM, SE A PORCAO DO SUBPROGRAMA COMECANDO COM O COMANDO ROTULADO LOCO FOI VISITADA PELA CHAMADA DA FUNCAO VISIT ACIMA, E TERMINADO POR

CALL TERM(C,RESTOR(2))

ENTAO, O RESULTADO FINAL DE VISIT E DE X PELA
ASSIGNACAO DE VALOR IMPLICADA POR = SERIA O VALOR DE C.
AS 2 LISTAS PUBLICAS W(1) E W(2) SERIAM DEIXADAS NAS MESMAS CONDICoes
ANTERIORES A CHAMADA DE PART2.

.RESTRICoes NA RECURSAO = CICLOS DE DO'S

COMO VISIT E' ESSENCIALMENTE UMA OPERACAO DE DESVIO, AS REGRAS DE FORTRAN
SE APLICAM COMO A UM GOTO NORMAL, PARTICULARMENTE COM RESPEITO AOS CICLOS
DE DO'S, APESAR DE HAVER ESTAS RESTRICoes PARA A RECURSAO DENTRO DO SLIP
ESTE MECANISMO FUNCIONA VARIADAMENTE NAS FUNCOES LSTEGL(LA, LB) E LSSCPY(LA).
LSTEGL(LA, LB) TESTA A "IGUALDADE" DAS LISTAS CUJOS ALIAS'ES SAO LA E LB.
LSSCPY(LA) CRIA A COPIA DE UMA ESTRUTURA DE LISTAS CUJO ALIAS E'
SEU PARAMETRO DE ENTRADA.

. O MECANISMO DA READER = PERCURSO EM ESTRUTURA DE LISTAS.

PARA O PROGRAMADOR SENTIR A POTENCIA TOTAL DO PROCESSAMENTO DE LISTAS,
A HABILIDADE DE PERCURSO EM ESTRUTURAS DE LISTAS DEVE SER CLARO E SIMPLES.
EM PARTICULAR EM TRATAMENTO DE ESTRUTURAS TALS COMO ARVORES, GRAFOS, PLEXOS, ETC

A READER E' ESSENCIALMENTE UM DISPOSITIVO ANALOGO A INDEXACAO.
POREM, A INDEXACAO SOBRE ESTRUTURAS DE LISTAS E' MAIS COMPLEXA QUE
SOBRE "ARRAYS". SE UM REGISTRADOR-INDEXADOR ESTA' APONTANDO PARA UM
DADO PARTICULAR DENTRO DE UM ARRAY, ENTAO UMA COMPUTACAO (NA QUAL O
CONTEUDO DAQUELE REGISTRADOR-INDEXADOR E' UM PARAMETRO) PODE SER ESPECI-
FICADA TAL QUE O VALOR PRODUZIDO POR AQUELA COMPUTACAO E' O ENDERECO DE
MAQUINA DE UMA CELULA DENTRO DO ARRAY.
UM PONTO IMPORTANTE SOBRE TALS COMPUTACOES E' QUE (EM GERAL) ELAS NAO
ENVOLVEM QUALQUER IDEIA DA "HISTORIA" DO REGISTRADOR-INDEXADOR AFETADO.
ENTRETANTO, SE ALGUM REGISTRADOR ESTA' APONTANDO PARA UMA CELULA PROFUNDA
DENTRO DA ESTRUTURA DE LISTAS (I.E., NUM CELULA QUE PERTENCE A UMA LISTA
QUE E' SUBLISTA DE SUBLISTA, ETC. DA LISTA PRINCIPAL) E SE DESEJA
QUE AQUELE REGISTRADOR APONTE SUCESSIVAMENTE PARA OS NOS DA ESTRUTURA
(I.E., PARA AQUELAS CELULAS QUE CONTEM OS NOMES DAS VARIAS SUBLISTAS QUE
LIDERAM A SUBLISTA QUE ESTA' SENDO CORRENTEMENTE APONTADA) NUMA ORDEM INVERSA
DAQUELA EM QUE FORAM ENCONTRADAS DURANTE A DESCIDA NA ESTRUTURA, ENTAO,
A INFORMACAO HISTORICA DEVE ESTAR DISPONIVEL PARA EFETUAR O PROCEDIMENTO
DE RETORNO.

. A HISTORIA DO PERCURSO DE UMA ESTRUTURA DE LISTAS ESTA' CONTIDA NUMA
PILHA DE READER'S.

UMA PILHA DE READER'S ASSOCIADA A UMA ESTRUTURA DE LISTAS CONTENDO
INFORMACOES SUFICIENTES PARA POSSIBILITAR O PERCURSO DE IDA E VOLTA
NA ESTRUTURA INTEIRA. NENHUMA ANOTACAO ADICIONAL PRECISA SER EFETUADA
PELO PROGRAMADOR.

.CRIACAO DE UMA READER:

UMA CELULA E' TIRADA DA LANS, PARA CRIACAO DA READER DE UMA LISTA ESPECI-
FICA OU DE UMA ESTRUTURA DE LISTAS.
POR EXEMPLO:

LEITOR = DRDROV(LISTA)

ESTE COMANDO ASSUME QUE EXISTE UMA LISTA COM O ALIAS "LISTA", UMA CELULA E' RETIRADA DA LAVS E SEU ENDEREÇO E' DEIXADO COMO UM INTEIRO NA VARIÁVEL "LEITOR". OS CAMPOS DA CELULA SÃO PREENCHIDOS COM:

1. A PALAVRA DA CELULA DA READER:

ID=3 (CELULA TIPO 3: READER)
LNKL: ENDEREÇO DA HEADER DA LISTA, ESTE CAMPO E' CHAMADO LPNTR (LIST POINTER) DA READER.
LNKR=0

2. A PALAVRA DA CELULA DA READER:

ID=0
LNKL: ENDEREÇO DA HEADER DA LISTA.
LNKR=0 ESTE CAMPO E' CHAMADO DE LCNTR (LEVEL COUNTER) DA READER.

A READER NESTE ESTADO E' DITA ESTAR NO ESTADO INICIAL, APONTANDO PARA A CABECA-DE-LISTA DA QUAL ELA E' A LEITORA.

. OPERACOES DE AVANÇO :

EXISTE UM CONJUNTO DE OPERACOES DE AVANÇO QUE TEM COMO FUNÇÃO A MODIFICAÇÃO DO ESTADO DE UMA READER, QUE E' UM DE SEUS PARAMETROS DE ENTRADA. O AVANÇO PODE SER LINEAR OU ESTRUTURAL E EM VIRTUDE DA SIMETRIA DO SLIP, CADA CLASSE E' DIVIDIDA DE ACORDO COM A DIREÇÃO DO AVANÇO (ESQUERDO OU DIREITO).

UM AVANÇO LINEAR COMEÇA COM A CELULA CUJO ENDEREÇO ESTA' ESPECIFICADO PELO LPNTR DA READER QUE E' UM PARAMETRO DA FUNÇÃO DE AVANÇO. O AVANÇO SEGUIE OS APONTADORES (LNKR, NO CASO DE AVANÇO A DIREITA; LNKL, NO CASO DE AVANÇO A ESQUERDA) ATÉ QUE UM CRITÉRIO DE TERMINO DE AVANÇO SEJA SATISFEITO. O ENDEREÇO DA CELULA APONTADA SUBSTITUI O LPNTR DA READER.

. INICIALIZAÇÃO DE UMA READER:

UMA READER PODE SER INICIALIZADA POR MEIO DA FUNÇÃO INITRD(K) QUE SUBSTITUI O LPNTR (I.E., O CONTEÚDO DO CAMPO LNKL DA PRIMEIRA PALAVRA DA READER) PELO CONTEÚDO DO CAMPO LNKL DA SEGUNDA PALAVRA DA READER QUE SEMPRE APONTA PARA A CABECA-DA-LISTA RELACIONADA A ESTA READER.

OS AVANÇOS ESTRUTURAIS SÃO EFETUADOS DE MODO ANALOGO AO AVANÇO LINEAR, SOMENTE QUE DESDE O COMEÇO DO AVANÇO E EM QUALQUER ESTAGIO DO AVANÇO ANTES DE SEU TERMINO, A CELULA ENCONTRADA NÃO PODE CONTER UM NOME (ID=1) OU UMA HEADER (ID=2). NO CASO DO AVANÇO ESTRUTURAL TER ENCONTRADO UMA CELULA CONTENDO O NOME DE UMA LISTA, A READER ATUAL E' COPIADA NUMA CELULA REMOVIDA DA LAVS E OS CAMPOS DA CELULA READER ORIGINAL SÃO ESTABELECIDAS PARA SER UMA READER NO ESTADO INICIAL DA LISTA CUJO NOME ENCONTROU. O CAMPO LCNTR E' INCREMENTADO DE 1, REFLETINDO QUE A READER DESCEU MAIS UM NÍVEL NA ESTRUTURA. O ENDEREÇO DA CELULA TOMADA DA LAVS E' COLOCADO NO CAMPO LNKR DA PRIMEIRA PALAVRA DA READER. O MECANISMO CRIA UMA PILHA DE READER'S QUE PRESERVA A HISTÓRIA DE PERCURSO DA ESTRUTURA DE LISTAS ATÉ QUE A READER SEJA DESEMPILHADA.

. CRITÉRIOS DE PARADA DE AVANÇOS:

O ENCONTRO DE UMA HEADER NO PERCURSO DE UM AVANÇO LINEAR, E' UM CRITÉRIO DE PARADA DO AVANÇO LINEAR. ENTRETANTO, NUM AVANÇO ESTRUTURAL O LCNTR DA READER E' CONSULTADO PARA DETERMINAR SE O AVANÇO DEVE TERMINAR. SE O VALOR DO CONTADOR NÃO E' ZERO, E' DESEMPILHADA UMA LEITORA DA PILHA DE READER'S E O AVANÇO CONTINUA A PARTIR DA CELULA INDICADA PELO ESTADO DO NOVO LPNTR.

III. PROCESSOS:

1. CRIAÇÃO DA LISTA DE ESPAÇOS DISPONÍVEIS (LAVS) E AS LISTAS PÚBLICAS (W'S):

CALL INITAS(SPACE,N) (10-180)

"SPACE" É UM ARRAY LINEAR DE N (*) PALAVRAS QUE FOI DECLARADO NO PROGRAMA PRINCIPAL EM FORTRAN PELO COMANDO DIMENSION. A SUBROTINA INITAS CRIA A LISTA DE ESPAÇOS DISPONÍVEIS (LAVS), E CRIA ADICIONALMENTE UM CONJUNTO DE LISTAS VAZIAS COM ALIAS'ES W(1), W(2), W(3), ... , W(100), SENDO AS CABECAS-DE-LISTA DAS LISTAS PÚBLICAS.

O ESPAÇO EXIGIDO PELAS CABECAS-DE-LISTA DAS LISTAS PÚBLICAS E SEUS ALIAS'ES NÃO SÃO TIRADOS DA LISTA DE ESPAÇOS DISPONÍVEIS (LAVS). QUALQUER SUBPROGRAMA QUE REQUERIR ACESSO AS LISTAS PÚBLICAS, DEVEM DECLARAR-LAS POR MEIO DO COMANDO COMMON SEGUINTE:

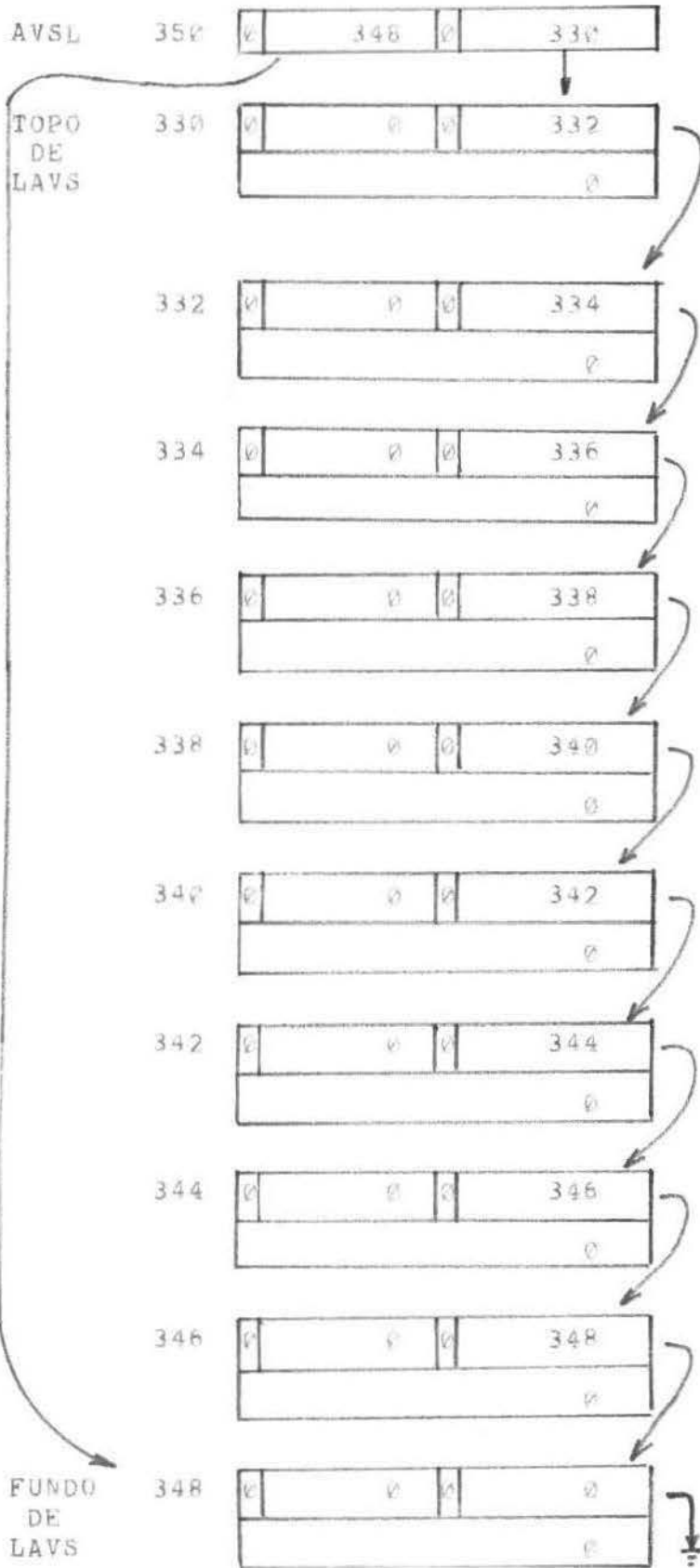
COMMON AVSL,W(100)

* OBS: N DEVE SER UM NÚMERO INTEIRO PAR, POIS CADA CELULA É UM PAR CONSECUTIVO DE PALAVRAS.

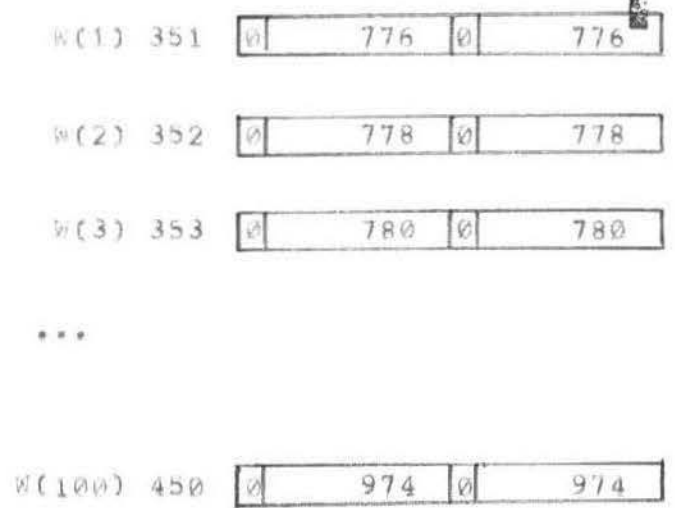
EXEMPLO DE INICIALIZAÇÃO DA LISTA LIVRE E DAS LISTAS PÚBLICAS:

```
DIMENSION SPACE(20)
COMMON AVSL, W(100)
N=20
CALL INITAS(SPACE,N)
. . .
END
```

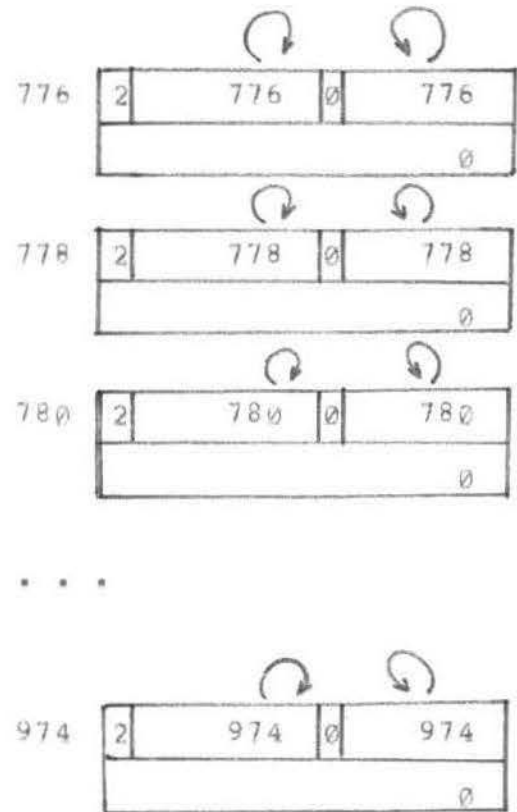
AVSL: NOME DA LISTA-LIVRE



ALIASES DAS LISTAS PUBLICAS



CABECAS-DE-LISTA DAS LISTAS PUBLICAS:



2. CRIACAO DE LISTAS:

LIST(K)

(440-540)

2

A FUNCAO LIST(K) CRIA UMA LISTA VAZIA E DEIXA SEU NOME, TANTO EM K COMO EM SEU VALOR. A CELULA K SE TORNA UM ALIAS PARA A LISTA CRIADA. SE K=9, ENTAO O CONTADOR DE REFERENCIA DA LISTA E' ZERO, SENAO E' UM. (**) DOIS ALIAS'ES PODEM SER ESTABELECIDOS, ESCRIVENDO-SE: LA=LIST(LB) QUE PROVOCA QUE LA E LB CONTENHAM O NOME DA LISTA CRIADA, I.E., SEJAM OS ALIAS'ES PARA AQUELA LISTA. COMO LIST E' UMA FUNCAO, PODE SER ENCADEADA.

** OBS: ESTE CONTADOR DE REFERENCIAS E' USADO NUM MECANISMO DE CONTROLE DE "DELECAO" DE LISTAS.

. ALIAS OU NOME-DE-LISTA

UMA LISTA (OU ESTRUTURA DE LISTAS) E' SEMPRE REFERIDA POR MEIO DE UMA PALAVRA CHAMADA ALIAS DA LISTA OU NOME-DE-LISTA.

UMA LISTA PODE TER VARIOS ALIAS'ES.

UM ALIAS DE UMA LISTA TEM O SEGUINTE FORMATO:

0	LNKL	0	LNKR
---	------	---	------

ONDE O CONTEUDO DOS CAMPOS LNKL E LNKR SAO IGUAIS E APONTAM PARA A HEADER DA LISTA.

C CRIACAO DE UMA LISTA VAZIA:

L=0

LA = LIST(L)

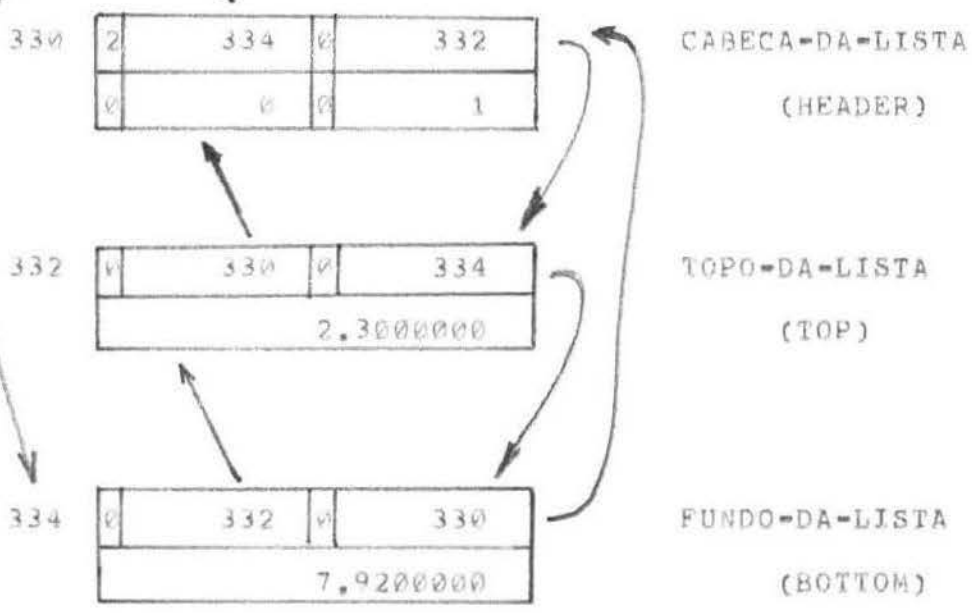
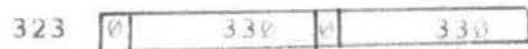
C INSERIR INFORMACOES:

N1= NEWTOP(7.92,L)

N2 = NEWTOP(2.3,L)

ALIAS L DA LISTA

QUIERO ALIAS LA DA LISTA



CORPO-DA-LISTA (BODY)

3. COLOCACAO DE INFORMACAO EM LISTAS:

A SEGUIR, DATUM (OU DADO) INDICA A INFORMACAO QUE DEVE SER COLOCADA NA LISTA. UM DADO PODE SER QUALQUER INFORMACAO VALIDA EM FORTRAN (DESDE QUE CAIBA NUMA PALAVRA DE MAQUINA):

- UM NUMERO EM PONTO-FIXO (OU INTEIRO)
- UM NUMERO EM PONTO-FLUTUANTE (OU REAL)
- UM CONJUNTO DE CARACTERES HOLLERITH (OU ALFANUMERICOS) - ATE 5 CARACTERES.
- UM ENDEPECO DE MAQUINA
- O NOME DE UMA LISTA, ETC.

A COLOCACAO DE NOMES-DE-LISTAS NUMA CELULA COM ID=1, QUANDO E' INSERIDA UMA SUBLISTA, E' UMA FUNCAO AUTOMATICA DO SISTEMA SLIP.

ALGUMAS DAS FUNCOES DE COLOCACAO DE INFORMACAO EM LISTA, RETORNAM COM O ENDEPECO DE MAQUINA DA CELULA QUE FOI RETIRADA DA LISTA DE ESPACOS DISPONIVEIS (LAVS), OUTRAS FUNCOES RETORNAM COM O CONTEUDO ANTIGO DA CELULA EM QUE OCORREU UMA SUBSTITUICAO DE INFORMACAO, EM CERTAS CONDICoes, ALGUMAS FUNCOES RETORNAM COM INDICADORES DE CONDICoes DE ERRO, QUE EM MUITOS CASOS POSSIBILITAM A RECUPERACAO DE ERROS.

• EMPILHAR DADO NO TOPO E NO FUNDO DA LISTA.

NEWTOP(P,Q) (1090-1130)

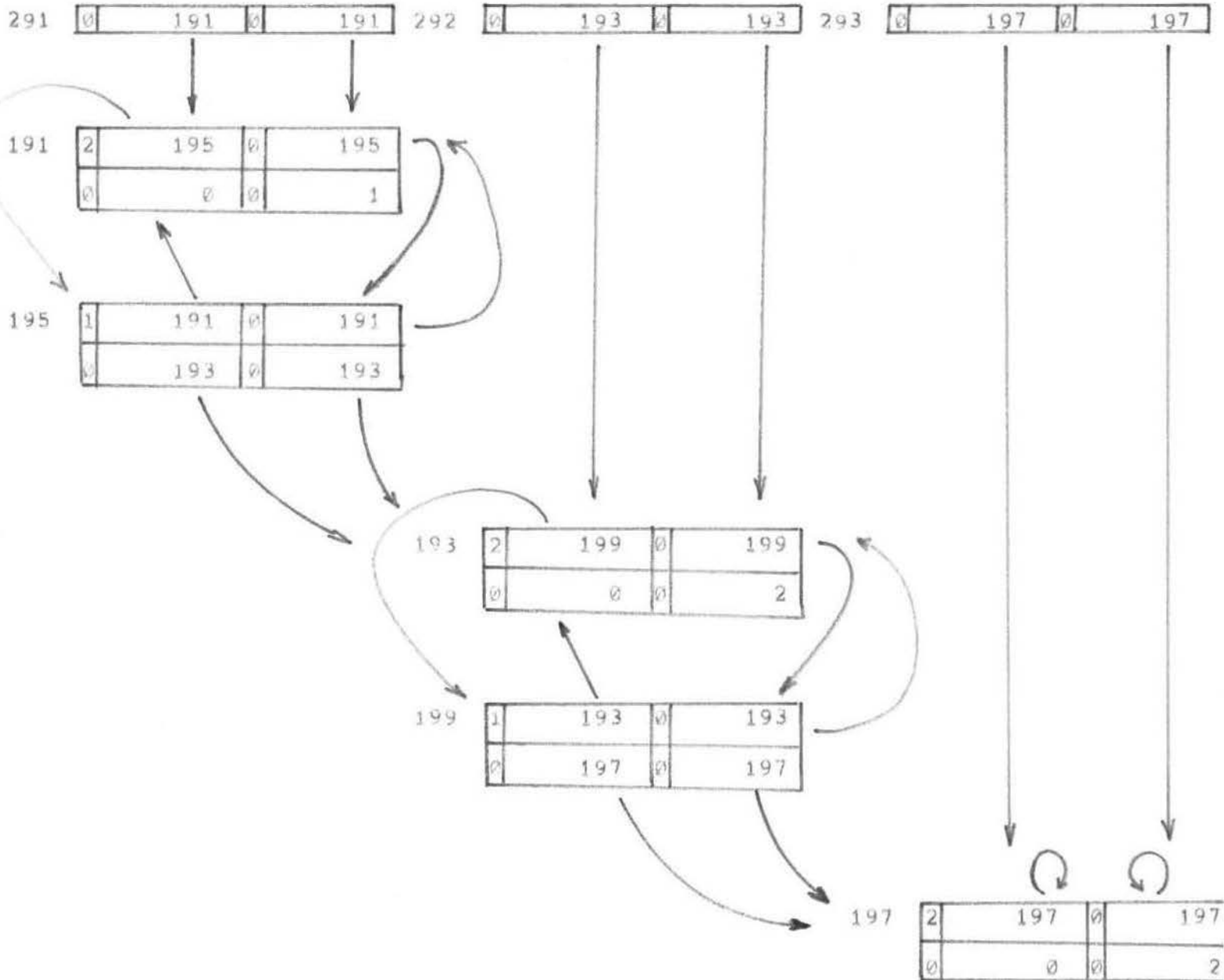
NEWTOP EMPILHA O DADO P NO TOPO DA LISTA Q. NEWTOP E' UMA FUNCAO QUE TEM COMO VALOR O ENDEPECO DE MAQUINA DA CELULA TOMADA DA LAVS PARA O ARMAZENAMENTO DO DADO E OS APONTADORES ASSOCIADOS.

NO SEGUINTE EXEMPLO, TEMOS A ESTRUTURA DE LISTAS ILUSTRADA A SEGUIR: ((()))
 1 2 3 3 2 1

```
DIMENSION SPACE(100), LST(4)
COMMON AVSL, W(100)
DATA N/100/
```

```
CALL INITAS(SPACE,N)
CALL LIST(LST(1))
DO 100 I=2,3
100 CALL NEWTOP(LIST(LST(I)),LST(I-1))
```

ALIAS'ES LST(1)



NEWBOT(P,Q)

(1140-1180)

NEWBOT É UMA FUNÇÃO QUE OPERA COMO A NEWTOP, EXCETO QUE O DADO É EMPILHADO NO FUNDO DA LISTA Q.

. INSERIR INFORMAÇÕES NA LISTA

NXTLFT(P,A)

(1190-1320)

NXTLFT INSERE O DADO P À ESQUERDA DA CELULA CUJO ENDEREÇO DE MAQUINA É ESPECIFICADO POR A. O VALOR DA FUNÇÃO É O ENDEREÇO DE MAQUINA DA CELULA TOMADA DA LAVS PARA ARMAZENAMENTO DO DADO E DOS APONTADORES ASSOCIADOS.

NXTRGT(P,A)

(1330-1460)

NXTRGT OPERA ANALOGAMENTE À FUNÇÃO NXTLFT, EXCETO QUE INSERE O DADO P À DIREITA DA CELULA CUJO ENDEREÇO DE MAQUINA É ESPECIFICADO POR A.

. INSERIR LISTA A ESQUERDA E DIREITA DE UMA CELULA:

INLSTL(M,A)

(1470-1600)

M DEVE SER O ALIAS DE UMA LISTA, A É O ENDEREÇO DE MAQUINA DE UMA CELULA. INLSTL INSERE O CORPO DA LISTA M (I.E., O CONJUNTO DE CELULAS LIGADAS QUE CONSTITUEM A LISTA M, EXCETO A CELULA HEADER DE M) À ESQUERDA DA CELULA A. AUMENTANDO ASSIM O COMPRIMENTO DA LISTA QUE CONTEM A CELULA A. A LISTA M É ESVAZIADA E O SEU NOME É LIBERADO COMO VALOR DA FUNÇÃO INLSTL.

INLSTR(M,A)

(1620-1740)

FUNÇÃO DE MODO ANALOGO A INLSTL, EXCETO QUE A LISTA "M" É INSERIDA A DIREITA DAQUELA CELULA CUJO ENDEREÇO É ESPECIFICADO POR "A".

. SUBSTITUIR O CONTEUDO DO CAMPO DE INFORMAÇÃO DE UMA CELULA:

SUBSTP(P,M)

(1820-1860)

SUBSBT(P,M)

(1870-1910)

O DADO P SUBSTITUI O DADO PRESENTEMENTE ARMAZENADO NO TOPO (FUNDO) DA LISTA M. O VALOR DESTA FUNÇÃO É O DADO QUE FOI SUBSTITUIDO, I.E., O CONTEUDO ANTERIOR DO CAMPO DE INFORMAÇÃO DA CELULA DE TOPO (FUNDO) DA LISTA M.

SUBST(P,M)

(1750-1810)

O DADO P SUBSTITUI O DADO ARMAZENADO NA CELULA CUJO ENDEREÇO É ESPECIFICADO POR A. O DADO SUBSTITUIDO É O VALOR DA FUNÇÃO.

4. O MECANISMO DA LEITURA (READER) - PERCURSO DE LISTAS E ESTRUTURA DE LISTAS.

OS DETALHES DE FUNCIONAMENTO INTERNO DO MECANISMO DA READER FORAM DISCUTIDOS NA SECCAO II ANTERIOR. NESTA SECCAO, O MESMO MECANISMO E SUAS OPERACOES SAO DESCRITAS DO PONTO DE VISTA DO PROGRAMADOR.

LRDROV(P)

(3960-4020)

A FUNCAO LRDROV PREPARA UMA READER PARA A LISTA CUJO ALIAS "P" E' DADO COMO SEU PARAMETRO DE ENTRADA. O VALOR DA FUNCAO E' O ENDERECO DA CELULA QUE ESTA' NO FORMATO DE READER (E QUE FOI TIRADA DA LAYS). NO SEU ESTADO INICIAL, A READER ESTA' APONTANDO PARA A CABECA-DA-LISTA PARA A QUAL FOI CRIADA.

LPNTR(R)

(4150-4190)

O PARAMETRO R DE ENTRADA DA FUNCAO LPNTR E' O ENDERECO DE UMA READER. O VALOR INTEIRO DA FUNCAO E' O ENDERECO DE MAQUINA ARMAZENADO NO CAMPO LPNTR DAQUELA READER, J.E., O ENDERECO DA CELULA PARA A QUAL AQUELA READER ESTA' APONTANDO ATUALMENTE.

LCNTR(R)

(4100-4140)

OPERA COMO A LPNTR, EXCETO QUE O CAMPO LCNTR (CONTADOR DE NIVEL) DA READER "R" E' O VALOR DA FUNCAO.

LOFRDR(R)

(4030-4090)

OPERA COMO LPNTR, EXCETO QUE O VALOR DA FUNCAO E' O NOME DA LISTA QUE A READER ESTA' APONTANDO ATUALMENTE. O VALOR E' LIBERADO NO FORMATO DE NOME-DE-LISTA.

• OS AVANCOS LINEARES:

EXISTEM SEIS FUNCOES DE AVANCO LINEAR, CADA UMA DAS QUAIS COM DOIS PARAMETROS DE ENTRADA, O PRIMEIRO DELES E' O ENDEREÇO DE UMA READER QUE E' O INSTRUMENTO PARA O AVANCO E UMA MARCA (FLAG) QUE E' ZERO OU NAO-ZERO DEPENDENDO DO SUCESSO DO AVANCO. AS OPERACOES DE AVANCO LINEAR TEM (APARENIENTEMENTE) AS SEGUINTEs TAREFAS:

- (A) PROVOCAR QUE O CAMPO LPNTR DA READER AVANCE PARA UMA NOVA POSICAO, I.E., QUE O LPNTR APONTE PARA UMA APOS OUTRA DAS CELULAS DA LISTA A QUAL A READER ESTA' ASSOCIADA, SEGUINDO OS APONTADORES DAS CELULAS DA LISTA, E FINALMENTE, RETORNAR QUANDO O CAMPO LPNTR ESTIVER APONTANDO PARA UMA CELULA PARTICULAR DE ACORDO COM A FUNCAO DE AVANCO ESCOLHIDA.
- (B) LIBERAR COMO RESULTADO DA FUNCAO DE AVANCO, O DADO CONTIDO NA CELULA QUE ESTA' SENDO APONTADA PELO ESTADO FINAL DE LPNTR.
- (C) ESTABELECEr COMO VALOR DA MARCA (CUJO NOME E' O SEGUNDO PARAMETRO DE ENTRADA DA FUNCAO DE AVANCO) PARA ZERO, SE A FUNCAO SATISFAZ COM SUCESSO O CRITERIO SELECIONADO, PARA NAO-ZERO, SE A READER APONIA PARA A CABECA-DA-LISTA. NESTE CASO, A OPERACAO DE AVANCO NAO ACHOU UMA CELULA COM AS PROPRIEDADES ESPECIFICADAS COMO CRITERIO DE PARADA SELECIONADO PELO PROGRAMADOR.

AS SEIS OPERACOES DE AVANCO LINEAR SAO:

- ADVLWR(R,F) - AVANCO LINEAR PARA UMA PALAVRA A DIREITA
- ADVLER(R,F) - AVANCO LINEAR PARA UM ELEMENTO A DIREITA
- ADVLNR(R,F) - AVANCO LINEAR PARA UM NOME A DIREITA
- ADVLWL(R,F) - AVANCO LINEAR PARA UMA PALAVRA A ESQUERDA.
- ADVLEL(R,F) - AVANCO LINEAR PARA UM ELEMENTO A ESQUERDA.
- ADVLNL(R,F) - AVANCO LINEAR PARA UM NOME A ESQUERDA.

PARA MEMORIZAR, AS QUATRO PRIMEIRAS LETRAS DO NOME DAS FUNCOES SAO ADVL, ABBREVIACAO DE AVANCO LINEAR ("ADVANCE LINEAR") A ULTIMA LETRA DETERMINA A DIRECAO DO AVANCO "L" PARA A ESQUERDA (LEFT) "R" PARA A DIREITA (RIGHT). (OS AVANCOS PARA ESQUERDA E PARA DIREITA PODEM SER INTERPRETADOS COMO PARA CIMA E PARA BAIXO, DEPENDENDO DA CONVENIENCIA)

A QUINTA LETRA DOS NOMES DAS FUNCOES DE AVANCO (W, E, OU N) DETERMINAM O CRITERIO DE TERMINO DO AVANCO.

PARA PROPOSITO DE AVANCOS, AS CELULAS SAO CLASSIFICADAS DE ACORDO COM O DADO. AS CELULAS QUE TEM UM NOME NO CAMPO DE INFORMACAO (I.E., ID=1) TEM A DESIGNACAO "N" (NOME), TODAS AS OUTRAS CELULAS SAO DESIGNADAS "E" (ELEMENTO). QUANDO A NATUREZA DO DADO E' IRRELEVANTE, A LETRA ASSOCIADA E' "W"(WORD). A CLASSE "W" E' PORTANTO A UNIAO DAS CLASSES "N" E "E". A IDENTIFICACAO DA HEADER E' A PARTE.

EXEMPLOS:

X = ADVLWR(R5, FLAG)

O CAMPO LPNTR DA LEITORA R5 APONTARÁ PARA A PROXIMA CELULA A DIREITA DA CELULA QUE ELA APONTAVA QUANDO A OPERACAO FOI INICIADA. SE AQUELA CELULA NAO E' A HEADER DA LISTA, ENTÃO O DADO ARMAZENADO SUBSTITUIRÁ O VALOR DE X, E A MARCA FLAG SERÁ NAO-ZERO (QUE PODE SER TESTADA POR UM COMANDO IF). SE AQUELA CELULA E' A HEADER DA LISTA, FLAG TERÁ VALOR NAO-NULO E X SERÁ ZERO.

X = ADVLFR(R5, FLAG)

O EFEITO DESTA OPERAÇÃO É IDENTICO AO DO EXEMPLO ANTERIOR, EXCETO QUE SE A PROXIMA CELULA ENCONTRADA TEM UM NOME COMO DADO, A PROXIMA CELULA A DIREITA SERÁ EXAMINADA PARA A CONDIÇÃO DE QUE ELA NÃO CONTENHA UM NOME COMO DADO. O CAMPO LPNTR APONTARÁ SUCESSIVAMENTE AS CELULAS NO AVANÇO A DIREITA ATÉ QUE ENCONTRE UMA CELULA "E" OU A HEADER DA LISTA, NO CASO DE ENCONTRAR UMA CELULA "E", A OPERAÇÃO DE AVANÇO LINEAR TERMINA COM SUCESSO, COM LPNTR APONTANDO PARA AQUELA CELULA "E" E RETORNA COMO RESULTADO O DADO ARMAZENADO NA CELULA, NO CASO DE ENCONTRAR A HEADER DA LISTA, O FLAG TERÁ VALOR NÃO-ZERO E O VALOR DA FUNÇÃO SERÁ ZERO.

X = ADVLNL(R5, FLAG)

NESTE CASO, O AVANÇO É A ESQUERDA, COMEÇANDO COM A CELULA PARA A QUAL O CAMPO LPNTR DA HEADER R5 APONTAVA. O AVANÇO A ESQUERDA CONTINUA ATÉ QUE SEJA ENCONTRADA UMA CELULA CONTENDO UM NOME OU UMA HEADER. SE UMA CELULA TIPO N É ACHADA, X É SUBSTITUÍDO PELO NOME NO FORMATO DE NOME, E FLAG TOMA VALOR ZERO (INDICANDO SUCESSO). SE UMA HEADER É ACHADA, X TOMA VALOR ZERO E FLAG TOMA VALOR NÃO-ZERO.

. AVANÇOS ESTRUTURAIIS:

A DIFERENÇA ENTRE OS AVANÇOS LINEAR E ESTRUTURAL É QUE O AVANÇO ESTRUTURAL PERMITE A LEITORA READER, DESCER E SUBIR NA ESTRUTURA DE LISTAS, I.E., PERCORRER SUBLISTAS DA LISTA. A OPERAÇÃO DE AVANÇO ESTRUTURAL TEM (APARENTEMENTE AO PROGRAMADOR) AS SEGUINTEs TAREFAS:

- (A), (B), E (C) DAS OPERAÇÕES DE AVANÇO LINEAR
- (D) MODIFICAR O LCNTR DA READER, PARA INDICAR A QUE NÍVEL A READER DESCEU NA ESTRUTURA (LCNTR=0 INDICA LISTA PRINCIPAL).
- (E) ATUALIZAR, SE EXIGIDO, AQUELE CAMPO DA READER QUE É RECUPERADO PELA FUNÇÃO LOPDR, TAL QUE SEMPRE CONTENHA O NOME DA LISTA PARA A QUAL A READER ESTÁ APONTANDO.

AS SEIS OPERAÇÕES DE AVANÇO ESTRUTURAL SÃO:

ADVSWR(R,F) - AVANÇO ESTRUTURAL PARA UMA PALAVRA A DIREITA.
ADVSNP(R,F) - AVANÇO ESTRUTURAL PARA UM NOME A DIREITA.
ADVSR(R,F) - AVANÇO ESTRUTURAL PARA UM ELEMENTO A DIREITA.
ADVSWL(R,F) - AVANÇO ESTRUTURAL PARA UMA PALAVRA A ESQUERDA.
ADVSNL(R,F) - AVANÇO ESTRUTURAL PARA UM NOME A ESQUERDA.
ADVSEL(R,F) - AVANÇO ESTRUTURAL PARA UM ELEMENTO A ESQUERDA.

ESTAS OPERAÇÕES DE AVANÇO ESTRUTURAL OPERAM COMO SEGUE:
SE O CAMPO ID DA CELULA QUE ESTÁ SENDO CORRENTEMENTE APONTADA PELO LPNTR DA READER "R", INDICA QUE A CELULA É DE CLASSE "N", ENTÃO A READER SE TORNA UMA READER DA LISTA CUJO NOME ESTÁ ARMAZENADO NAQUELA CELULA E AO LCNTR DA READER É ADICIONADO 1, SENDO PORTANTO CONVENIENTEMENTE MODIFICADO O CAMPO NA READER QUE CONTEM O NOME DA LISTA PARA A QUAL A READER É LEITORA.
SE A CELULA ATUALMENTE APONTADA PELO LPNTR DA READER, NÃO É DA CLASSE "N", ENTÃO A PROXIMA CELULA (A ESQUERDA OU A DIREITA, DEPENDENDO DA FUNÇÃO ESCOLHIDA) É EXAMINADA PELA CORRESPONDÊNCIA DO CRITÉRIO. A PROXIMA CELULA É UMA CANDIDATA PARA O AVANÇO. A CANDIDATA

PODE SATISFAZER O CRITERIO DE TERMINO ESPECIFICADO PELA FUNCAO DE AVANCO. NESTE CASO, O AVANCO TERMINA DE MODO ANALOGO AO TERMINO DO AVANCO LINEAR, I.E., O VALOR DA FUNCAO E' O DADO CONTIDO NA CELULA CANDIDATA FINAL, O FLAG TOMA VALOR ZERO.

SE A CELULA CANDIDATA NAO CORRESPONDE AO CRITERIO DE TERMINO, E NAO E' UMA HEADER, ENTAO TODO O PROCESSO REINICIA DO MESMO MODO.

SE A CANDIDATA E' UMA HEADER, O CAMPO LCNTR DA READER DETERMINA QUANDO O AVANCO TERMINA PELA FALHA DO CRITERIO, I.E., FLAG TOMA VALOR NAO-ZERO OU CONTINUA.

SE LCNTR=0 E' QUE A READER ESTA' LENDO A LISTA PRINCIPAL. O ENCONTRO DA HEADER NESTAS CIRCUNSTANCIAS SIGNIFICA QUE O FIM DE TODA A ESTRUTURA DE LISTA FOI ENCONTRADO. O OBJETIVO ESPECIFICADO PELO PROGRAMADOR NAO FOI ACHADO.

SE LCNTR NAO E' ZERO E UMA HEADER FOI ENCONTRADA NUM AVANCO, ENTAO, AQUELA HEADER E' A INDICACAO DE QUE O FIM DA SUBLISTA FOI ENCONTRADO A READER SOBE DE NIVEL PARA UMA CELULA DA LISTA DA QUAL ESTA E' UMA SUBLISTA. A PROXIMA CANDIDATA E' ENTAO DETERMINADA POR UM PASSO DE AVANCO LINEAR, MAS PROSSEGUE NO AVANCO ESTRUTURAL.

LCNTR DA READER E' DECREMENTADO DE 1 NA SUBIDA DA LISTA, E O NOME DA LISTA PARA A QUAL A READER SUBIU E' REGISTRADO PELO CAMPO APROPRIADO DA READER.

EXEMPLO:

SEJA UMA ESTRUTURA DE LISTAS

L1: (1, 2, 3, (41, (421, 422), 43), 5, (61, 62), 7)

UMA OUTRA DESCRICAO DA LISTA L1 PODERIA SER:

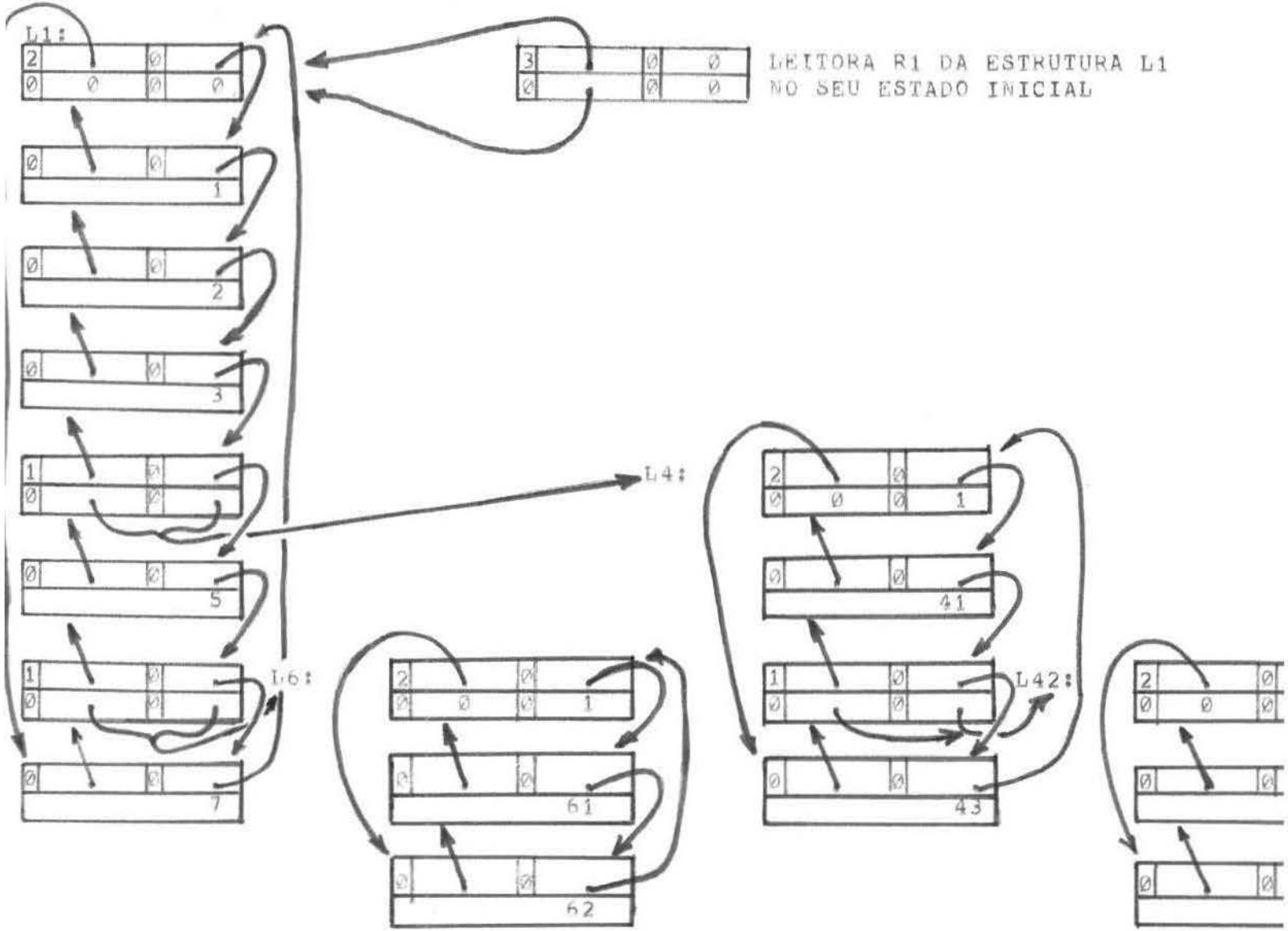
L1: (1, 2, 3, L4, 5, L6, 7)

COM L4: (41, L42, 43)

L42: (421, 422)

L6: (61, 62)

I.E., L1 E' UMA ESTRUTURA DE LISTAS COM OS ELEMENTOS 1, 2, 3, 5, 7
E AS DUAS SUBLISTAS L4 E L6.



SUPONHAMOS QUE A LEITORA R1 APONTE PARA A ESTRUTURA DE LISTAS L1
(R1 FOI CRIADA PELO COMANDO R1=LRDROV(L1))
R1 ESTA' ENTAO NO SEU ESTADO INICIAL (I.E., LENTR(R1) APONTE PARA A CABECA
DA LISTA (1)).
O EFEITO DO COMANDO

M = ADVSKD(R1, F)

E' QUE M TERIA VALOR 7 E F=0. O CAMPO LENTR DE R1 APONTARIA PARA O FUNDO
DA LISTA L1, I.E., O ELEMENTO "7".

SE O MESMO COMANDO FOSSE EXECUTADO NOVAMENTE, M CONTERIA O NOME DE L6
E F TERIA VALOR ZERO.

MAIS UMA EXECUCAO DEIXARIA M=62 E F=0
MAIS DUAS EXECUCOES DEIXARIA M=5 E F=0

VAMOS INICIALIZAR NOVAMENTE A READER R1:

M = INITFD(R1)

O EFEITO DO COMANDO

M=ADVSNF(R1, F)

E' QUE M CONTENHA O NOME DE L4 E F=M.

SE O COMANDO FOR EXECUTADO NOVAMENTE, O RESULTADO SERIA:

M=NOME DE L42 E F=0.

A REPETICAO DESTE COMANDO PRODUZIRIA M=NOME DE L6 E F=0, E
FINALMENTE M=M COM F NAO-NULO.

SE A READER ESTA' NO SEU ESTADO INICIAL, O COMANDO

M = ADVSER(R1, F)

FOSSE EXECUTADO REPECTIVAMENTE, M TOMARIA SUCESSIVAMENTE OS VALORES:
1, 2, 3, 41, 421, 422, 43, 5, 51, 62, 7 E 0

F TOMARIA VALOR ZERO (INDICANDO SUCESSO) EXCETO NO ULTIMO CASO, QUANDO
F TOMA VALOR NAO-ZERO (POIS A OPERACAO DE AVANCO ENCONTROU A READER DA
LISTA).

. OPERACOES DE PERCURSO REVERSO NUMA ESTRUTURA DE LISTAS.

LVLVRT(F)

(4200-4300)

O OBJETIVO DA FUNCAO LVLVRT E' PROVOCAR QUE A READER QUE E' SEU PARAMETRO
DE ENTRADA, SUBA DE NIVEL NA ESTRUTURA DE LISTAS QUE ESTA' SENDO APONTADA,
ATE' QUE A READER APONTE PARA A LISTA PRINCIPAL QUE ERA APONTADA ORIGINAL-
MENTE. SE A READER "R" JA ESTA APONTANDO PARA A LISTA PRINCIPAL, NADA
ACONTECE. DE OUTRA FORMA, O LENTR DA READER FICA APONTANDO PARA AQUELA
CELULA DA LISTA PRINCIPAL QUE CONTEM O NOME DA SUBESTRUTURA PARA A QUAL
A READER APONTAVA QUANDO A OPERACAO FOI INICIADA.

O VALOR DA FUNCAO LVLVRT E' O ENDEREÇO DA READER. CONSEQUENTEMENTE, ESTA
FUNCAO PODE SERVIR COMO PARAMETRO DE ENTRADA PARA QUALQUER OUTRA FUNCAO
QUE REQUEIRA UMA READER COMO UM DE SEUS PARAMETROS.

EXEMPLO: SEJA A ESTRUTURA DE LISTAS:

L1:(1, 2, 3, (41, (421, 422), 43), 5, (61, 62), 7)

SUPONHAMOS QUE A LEITORA "RA" ESTEJA APONTANDO PARA O ELEMENTO "421".

A CHAMADA DA FUNCAO

LVLVRT(RA)

DEIXARIA "RA" APONTANDO PARA A CELULA QUE CONTEM O NOME DA SUBLISTA L4.

LVLRV1(R)

(4310-4410)

ESTA FUNCAO TEM OPERACAO ANALOGA A LVLVRT EXCETO QUE, A READER "R" SOBRE SOMENTE UM NIVEL, SE POSSIVEL.

EXEMPLO: SEJA A ESTRUTURA

L1:(1,2,3,(41,(421,422),43),5,(61,62),7)

SUPONHAMOS QUE LPNTR DA LEITORA "RA" APONTE PARA O ELEMENTO "421" DA LISTA, A CHAMADA DA FUNCAO

LVLRV1(RA)

DEIXARIA "RA" APONTANDO PARA A CELULA CONTENDO O NOME DA SUBLISTA "L42" NA LISTA "L4".

. INICIALIZACAO DE UMA READER.

INITRD(R)

(4420-4470)

ESTA FUNCAO PROVOCA QUE O LPNTR DA READER "R" QUE E' SEU PARAMETRO DE ENTRADA APONTE PARA A HEADER DA LISTA PARA A QUAL A READER ESTA APONTANDO. A LEITORA "R" FICA, ENTAO, NO SEU "ESTADO INICIAL".

O VALOR DA FUNCAO E' O ENDEREÇO DA READER.

UMA READER PODE SER INICIALIZADA INDEPENDENTEMENTE DA PROFUNDIDADE QUE HAVIA PENETRADO NA ESTRUTURA DE LISTAS COM O COMANDO:

CALL INITRD(LVLVRT(R))

ONDE "R" E' A READER A SER INICIALIZADA.

. COPIA DE UMA READER:

LRDRCP(R)

(4590-4730)

O OBJETIVO DESTA FUNCAO E' FAZER UMA COPIA DA READER QUE E' SEU PARAMETRO DE ENTRADA E LIBERAR O ENDEREÇO DAQUELA COPIA COMO SEU VALOR. A UTILIDADE DE "LRDRCP" E' QUE A READER QUE ESTA' SENDO COPIADA CARREGA CONSIGO UM REGISTRO DA SUA HISTORIA. O PROGRAMADOR PODE QUERER DEIXAR A HISTORIA INTACTA E TAMBEM EFETUAR UMA OPERACAO DE AVANCO. A COPIA PRODUZIDA POR ESTA FUNCAO PODERIA SER AVANÇADA COMO SE ELA FOSSE A READER ORIGINAL, DEIXANDO A READER ORIGINAL INTACTA. POR OUTRO LADO, A READER ORIGINAL, PODERIA SER AVANÇADA, DEIXANDO A READER COPIADA INTACTA.

. LEITURA DO DADO DE UMA CELULA APONTADA POR UMA READER:

REED(R)

(3750-3790)

O PARAMETRO DE ENTRADA DESTA FUNCAO E' UMA READER.

O VALOR DA FUNCAO E' O DADO CONTIDO NA CELULA QUE O LPNTR DA READER ESTA' APONTANDO ATUALMENTE.

A READER NAO E' MODIFICADA.

. APAGAR UMA PFAZER (DEVOLVER PARA LAVS):

IRARDR(R)

(4480-4580)

3

ESTA FUNCAO TEM O EFEITO DE APAGAR A READER QUE E' SEU PARAMETRO DE ENTRADA, O VALOR DA FUNCAO E' O LCNTR DA READER, I.E., UM INTEIRO QUE INDICA A PROFUNDIDADE DENTRO DA ESTRUTURA DE LISTAS, QUE A LEITORA APAGADA ESTAVA APONTANDO ANTES DE SER APAGADA.

. OS MECANISMOS DA LEITORA SEQUENCIAL (SEQUENCE READER):

A LEITORA SEQUENCIAL E' UMA VERSAO MAIS SIMPLES DA READER. OS MECANISMOS ASSOCIADOS A ELA TORNAM POSSIVEL O PERCURSO DE UMA LISTA OU ESTRUTURA DE LISTAS DE UMA MANEIRA ANALOGA AOS MECANISMOS ASSOCIADOS COM A READER. POREM, A LEITORA SEQUENCIAL NAO ARMAZENA A INFORMACAO HISTORICA COM RESPEITO AS SUAS DESCIDAS NA ESTRUTURA, NAO SENDO CAPAZ DE SUBIR DE UMA SUBLISTA PARA A LISTA PRINCIPAL. SUA PRINCIPAL APLICACAO E' PARA OS PROCEDIMENTOS QUE REQUEREM UM PERCURSO SEQUENCIAL DE UMA LISTA OU ESTRUTURA DE LISTAS, SEM RETORNAR PARA AS LISTAS DE ORDEM MAIS ALTA. SUAS PRINCIPAIS VANTAGENS SAO QUE AS OPERACOES ASSOCIADAS COM ESTES MECANISMOS SAO EXECUTADAS MAIS RAPIDAMENTE QUE AS OPERACOES DE AVANCO CORRESPONDENTES DOS MECANISMOS DA READER, COM ECONOMIA DE MEMORIA E COMO A LEITORA SEQUENCIAL NAO PRECISA DE TOMAR CELULAS DE LAVS, NUNCA PRECISA APAGAR CELULAS.

. CRIACAO DE UMA LEITORA SEQUENCIAL (SEQUENCE READER)

SEQRDR(LST)

(7760-7790)

O COMANDO S=SEQRDR(LST) PROVOCA QUE "S" SEJA UMA LEITORA SEQUENCIAL DA LISTA CUJA ALIAS E' "LST". I.E., SEQRDR E' UMA FUNCAO CUJO VALOR E' UMA PALAVRA DE COMPUTADOR NO FORMATO DA LEITORA SEQUENCIAL,

. AVANCO LINEAR NA ESTRUTURA DE LISTAS:

SEQLR(S, F)

(7430-7490)

SEQLL(S, F)

(7360-7420)

ESTAS FUNCOES SAO EQUIVALENTES AS FUNCOES ADVLWR E ADVLWL, RESPECTIVAMENTE, COM RELACAO AOS VALORES PRODUZIDOS. EXECUCOES REPETIDAS RESULTAM NOS DADOS DAS SUCESSIVAS CELULAS DA LISTA PARA A QUAL "S" E' UMA READER SEQUENCIAL O FLAG "F" TOMA VALOR -1 SE O DADO LOCALIZADO NA CELULA E' UM ELEMENTO 0 SE O DADO DA CELULA APONTADA E' UM NOME-DE-LISTA, 1 SE A CELULA E' UMA HEADER.

A OPERACAO "SEQ??" PODE SER SEGUIDA POR UM COMANDO "IF" DA FORMA:

IF(F) N1, N2, N3

TAL QUE O CONTROLE E' TRANSFERIDO AO PROXIMO COMANDO DEPENDENDO DA CONDICAO RESULTANTE DA OPERACAO DE AVANCO.

O VALOR DAS FUNCOES E' O DADO APONTADO PELA LEITORA SEQUENCIAL.

SEQSL(S, F)

(7630-7750)

SEQSR(S, F)

(7500-7620)

ESTAS FUNCOES OPERAM COMO ADVSEL E ADVSER, RESPECTIVAMENTE, DO MESMO MODO COMO DESCRITO ACIMA. O ENCONTRO DE UMA HEADER EM QUALQUER NIVEL PROVOCA QUE O FLAG "F" TOME VALOR ZERO, MAS NAO PROVOCA NENHUMA ASCENSAO NA

ESTRUTURA DE LISTAS, ENTRETANTO, O ENCONTRO DE UM NOME-DE-LISTA PROVOCA A DESCIDA NA ESTRUTURA.
O VALOR DAS FUNÇÕES É O DADO CONTIDO NA CELULA APONTADA PELA LEITORA SEQUENCIAL.

• VALORES DO FLAG "F" RESULTANTES DAS OPERAÇÕES DE AVANÇO LINEAR COM A LEITORA SEQUENCIAL:

- F=-1 SE O DADO CONTIDO NA CELULA APONTADA PELA LEITORA SEQUENCIAL É UM ELEMENTO, I.E., NÃO É UM NOME-DE-LISTA.
- F= 0 SE O DADO CONTIDO NA CELULA APONTADA PELA LEITORA SEQUENCIAL É O NOME DE UMA LISTA.
- F= 1 SE A CELULA APONTADA PELA LEITORA SEQUENCIAL É UMA HEADER.

5. OUTROS MEIOS DE RECUPERAÇÃO DE INFORMAÇÃO DAS LISTAS:

. RECUPERAR DADO ARMAZENADO NA CELULA DE TOPO (FUNDO) DA LISTA:

TOP(L) (1920-1960)
O PARAMETRO L DE ENTRADA DA FUNÇÃO TOP É O NOME DE UMA LISTA. SEU VALOR É O DADO ARMAZENADO NO TOPO (CELULA MAIS-A-ESQUERDA) DAQUELA LISTA. A LISTA NÃO É MODIFICADA POR ESTA OPERAÇÃO.

BOT(L) (1970-2010)
ESTA FUNÇÃO OPERA COMO A TOP, EXCETO QUE O VALOR DA FUNÇÃO É O VALOR DO DADO ARMAZENADO NO FUNDO (CELULA MAIS-A-DIREITA) DA LISTA.

. DESEMPILHAR A CELULA DE TOPO (FUNDO), RECUPERANDO O DADO:

POPTOP(L) (2020-2060)
A CELULA DO TOPO DA LISTA CUJO NOME É O PARAMETRO DE ENTRADA PARA ESTA FUNÇÃO É DESEMPILHADA DA LISTA, I.E., A CELULA DO TOPO É DEVOLVIDA PARA LAVS E O DADO CONTIDO NESTA CELULA RETORNA COMO VALOR DA FUNÇÃO.

POPBOT(L) (2070-2110)
A CELULA DE FUNDO DA LISTA CUJO NOME É O PARAMETRO DE ENTRADA PARA ESTA FUNÇÃO É DESEMPILHADA DA LISTA, E O DADO CONTIDO NA CELULA LIBERADA É O VALOR DA FUNÇÃO.

. APAGAR UMA CELULA DA LISTA, RECUPERANDO O DADO:

DELETE(A) (3800-3950)
O PARAMETRO DE ENTRADA PARA ESTA FUNÇÃO É O ENDEREÇO DE MÁQUINA DE UMA CELULA. A CELULA INDICADA É "APAGADA" DA LISTA NA QUAL ELA APARECE E O DADO ARMAZENADO NAQUELA CELULA LIBERADO COMO VALOR DA FUNÇÃO. SE (INADVERTIDAMENTE) O ENDEREÇO DE UMA CABECA-DE-LISTA É DADO COMO PARAMETRO DE ENTRADA, ENTÃO O VALOR DA FUNÇÃO SERÁ ZERO E UMA MENSAGEM DE ERRO IMPRESSA. ENTRETANTO, O PROGRAMA CONTINUARÁ A SER EXECUTADO.

. CRIAÇÃO DE UMA NOVA LISTA, DIVIDINDO A LISTA DADA (SPLITTING):

NULSTL(A,L) (790-930)
OS PARAMETROS DE ENTRADA PARA ESTA FUNÇÃO SÃO: "A", O ENDEREÇO DE MÁQUINA DE UMA CELULA DA LISTA "L". O OBJETIVO DESTA FUNÇÃO É DIVIDIR A LISTA L CRIANDO UMA NOVA LISTA CONTENDO COMO SEUS MEMBROS TODAS AS CELULAS A ESQUERDA (OU ACIMA) DA CELULA INDICADA POR A, INCLUSIVE. AS CELULAS ASSIM ASSOCIADAS COM A NOVA LISTA SÃO REMOVIDAS DA LISTA L. O NOME DA NOVA LISTA CRIADA É O VALOR DESTA FUNÇÃO.

NULSTR(A,L) (940-1080)
ESTA FUNÇÃO É IDENTICA A FUNÇÃO NULSTL, EXCETO QUE TODAS AS CELULAS A DIREITA (OU ABAIXO) DA CELULA A, INCLUSIVE, SÃO "APAGADAS" DA LISTA L E COLOCADAS NA NOVA LISTA CRIADA. O VALOR DA FUNÇÃO É O NOME DA LISTA CRIADA.

6. FUNCOES DE TESTE:

- TESTE SE UMA VARIÁVEL ESTÁ NO FORMATO DE NOME DE LISTA, E APONTA PARA UMA CABEÇA-DE-LISTA:

NAMIST(K) (5080-5170)
O DADO EM K É EXAMINADO, SE O DADO É O NOME DE UMA LISTA, O VALOR DA FUNÇÃO É ZERO, SENÃO O VALOR DA FUNÇÃO É NÃO-ZERO.

- TESTAR SE UMA LISTA É VAZIA OU NÃO:

LISTMI(L) (5180-5260)
O PARÂMETRO L DE ENTRADA DESTA FUNÇÃO É O NOME DE UMA LISTA, SE AQUELA LISTA É VAZIA, O VALOR DA FUNÇÃO É ZERO, SENÃO O VALOR DA FUNÇÃO É NÃO-ZERO.

- TESTAR IGUALDADE DE DUAS ESTRUTURAS DE LISTAS:

LSTEQL(LA, LB) (5270-5510)

OS DOIS PARÂMETROS DE ENTRADA DESTA FUNÇÃO SÃO NOMES DE ESTRUTURAS DE LISTAS. O OBJETIVO DESTA FUNÇÃO É DETERMINAR SE AS DUAS ESTRUTURAS SÃO IGUAIS. SE ELAS SÃO IGUAIS, O VALOR DA FUNÇÃO É ZERO, SENÃO É NÃO-ZERO. DUAS ESTRUTURAS SÃO IGUAIS, SE ELAS TÊM ESTRUTURAS IDENTICAS, I.E., SE OS NOMES DAS SUBLISTAS QUE APARECEM NOS LUGARES CORRESPONDENTES DENTRO DE AMBAS ESTRUTURAS E SE OS ELEMENTOS CORRESPONDENTES DE AMBAS ESTRUTURAS SÃO IDENTICOS.

7. PROCESSOS MISTOS:

ALGUMAS FUNÇÕES DEPENDEM DE SEUS PARÂMETRO SEREM ENDEREÇOS DE MÁQUINA DE CELULA, COMO EM DELETE, NULSEL, NXTGT, ETC. CERTAS FUNÇÕES PRODUZEM TAIS ENDEREÇOS DE MÁQUINA COMO SEUS VALORES, EXEMPLOS: NXTGT, XTLEFT, NEWTOP, NEWBOT, ETC.

AS QUATRO FUNÇÕES ABAIXO SÃO MODOS EXPLICITOS DE PRODUZIR ENDEREÇOS:

MADLFT(A) (4940-5000)
MADRGT(A) (5010-5070)

O PARÂMETRO DE ENTRADA "A" DESTAS FUNÇÕES É O ENDEREÇO DE MÁQUINA DE UMA CELULA OU UMA HEADER. O VALOR DA FUNÇÃO É O ENDEREÇO DE MÁQUINA DA CELULA A ESQUERDA (DIREITA) DAQUELA ESPECIFICADA POR A. SE AQUELE ENDEREÇO DE MÁQUINA PRODUZIDO É O ENDEREÇO DE UMA HEADER ENTÃO O VALOR LIBERADO ESTÁ NO FORMATO DE NOME-DE-LISTA, SENÃO É UM NÚMERO INTEIRO.

MADNTP(L, N) (4740-4830)
MADNBT(L, N) (4840-4930)

OS PARÂMETROS DE ENTRADA DESTAS FUNÇÕES SÃO, RESPECTIVAMENTE, O NOME DE UMA LISTA "L" E UM INTEIRO "N". O OBJETIVO DESTA FUNÇÃO É LIBERAR O ENDEREÇO DE MÁQUINA DA N-ÉSIMA CELULA CONTANDO DO TOPO (FUNDO) DA LISTA L. SE AQUELA CELULA É A HEADER DA LISTA L, ENTÃO O VALOR É LIBERADO NO FORMATO DE NOME-DE-LISTA, SENÃO O VALOR É INTEIRO. SE N É MAIOR QUE O NÚMERO DE CELULAS DA LISTA L (INCLUINDO A CELULA HEADER), ENTÃO O EFEITO DA FUNÇÃO É COMO SE N MÓDULO M FOSSE O SEGUNDO PARÂMETRO DA FUNÇÃO, ONDE M É O NÚMERO ATUAL DE CELULAS DA LISTA L, INCLUSIVE A HEADER.

. ESVAZIAR / UMA LISTA, DEVOLVENDO SUAS CELULAS PARA LAVS:

MTLIST(L) (550-680)

A LISTA CUJO NOME APARECE COMO PARAMETRO DE ENTRADA PARA ESTA FUNCAO E' ESVAZIADA, I.E., SUAS CELULAS SAO DEVOLVIDAS PARA LAVS. O VALOR DA FUNCAO E' O NOME DA LISTA VAZIA L.

. APAGAR UMA LISTA, DEVOLVENDO TODAS SUAS CELULAS PARA LAVS:

IRALST(L) (690-780)

A LISTA CUJO NOME APARECE COMO PARAMETRO DE ENTRADA PARA ESTA FUNCAO E' APAGADA, I.E., TODAS SUAS CELULAS, INCLUSIVE A HEADER SAO DEVOLVIDAS PARA LAVS POR ESTA FUNCAO. ENTRETANTO ESTA LISTA (E SEU CONTEUDO) PODE SER UMA SUBLISTA DE OUTRA LISTA. SE ESTA CONDICAO E' DETETADA, NAO E' APAGADA A LISTA (NESTE CASO, O CONTADOR DE REFERENCIAS DA LISTA E' SUBTRAIDO DE UM). O VALOR DA FUNCAO E' UM INTEIRO QUE ESPECIFICA QUANTAS LISTAS COMPARTILHAM ESTA LISTA COMO SUBLISTA. SE AQUELE VALOR E' ZERO, A LISTA E' APAGADA.

. COPIAR UMA LISTA DADA:

LSSCPY(L) (5520-5690)

O PARAMETRO DE ENTRADA DESTA FUNCAO E' O NOME-DE-LISTA. SEU OBJETIVO E' CRIAR UMA LISTA QUE E' UMA COPIA DA LISTA (OU ESTRUTURA) DE ENTRADA. O VALOR DA FUNCAO E' O NOME DA NOVA LISTA CRIADA.

. TRANSMISSAO DE PARAMETROS PARA A RECURSAO - LISTAS PUBLICAS:

PRESRV(N) (5970-6030)

O PARAMETRO DE ENTRADA DESTA FUNCAO E' UM INTEIRO N (N<=100). ESTA FUNCAO "PRESERVA" OS TOPOS DAS N PRIMEIRAS DAS 100 LISTAS PUBLICAS: W(1), W(2), ... W(100). AQUI, PRESERVAR UMA LISTA SIGNIFICA EMPILHAR A INFORMACAO CONTIDA NO TOPO DAS LISTA, NOVAMENTE NO TOPO DAS LISTAS. O CONIEUDO DAS DUAS CELULAS DO TOPO DAS LISTAS PRESERVADAS SAO IDENTICOS.

RESTOR(N) (6040-6100)

A SUBROTINA RESTOR DEVOLVE OS TOPOS DAS N PRIMEIRAS LISTAS PUBLICAS PARA LAVS.

PARMT2(X,Y) (6210-6280)

ESTA SUBROTINA PROVOCA QUE OS DOIS PARAMETROS DE ENTRADA SEJAM EMPILHADOS NO TOPO DAS LISTAS PUBLICAS W(1) E W(2), RESPECTIVAMENTE. SUA UTILIDADE E', PRINCIPALMENTE, NA COMUNICACAO DE PARAMETROS PARA A RECURSAO. PARMT2 SERVE COMO MODELO PARA AS SUBROTINAS

PARMTN(X1,X2,X3, ... ,XN) QUE COMUNICA N PARAMETROS ATRAVEZ DAS LISTAS PUBLICAS W(1), W(2), ... W(N).

. RESTAURAR CELULA PARA LAVS E RETIRAR CELULA DE LAVS:

RCELL(A) (360-430)

ESTA SUBROTINA DEVOLVE PARA LAVS A CELULA CUJO ENDEREÇO É DADO POR A.
O PROGRAMADOR DEVE UTILIZAR ESTA FUNÇÃO FREQUENTEMENTE.

NUCELL(DUMMY)

(190-350)

ESTA FUNÇÃO TEM UM PARÂMETRO FALSO "DUMMY" E PROVOCA A RETIRADA DE UMA
CELULA DE LAVS E SEU VALOR É O ENDEREÇO DE MÁQUINA DA CELULA OBTIDA DE
LAVS.

O PROGRAMADOR NÃO PRECISA CHAMAR ESTA FUNÇÃO.

• MARCA DE LISTA:

MRKLST(M, L)

(8600-8640)

O OBJETIVO DESTA FUNÇÃO É COLOCAR UMA MARCA M=0, 1, 2, OU 3 NA LISTA L.
INICIALMENTE, TODAS AS LISTAS SÃO MARCADAS COM 0. O VALOR DA FUNÇÃO É O
NOME DA LISTA, I.E., O SEGUNDO PARÂMETRO DE ENTRADA.

MRKLSS(M, L)

(8650-8740)

ESTA FUNÇÃO É IDENTICA A FUNÇÃO MRKLST, EXCETO QUE TODAS AS LISTAS DA
ESTRUTURA DE LISTAS L SÃO MARCADAS COM O CONTEÚDO DE M.

LSIMRK(L)

(8750-8790)

O PARÂMETRO DE ENTRADA PARA ESTA FUNÇÃO É O NOME DA LISTA L, SEU VALOR
É A MARCA DAQUELA LISTA.

• OPERAÇÕES EM LISTAS DESCRITIVAS - (ATTRIBUTE-VALUE LISTS)

QUALQUER LISTA PODE TER UMA LISTA DESCRITIVA. AS LISTAS DESCRITIVAS SE
TORNAM PARTE INTEGRAL DAS LISTAS ASSOCIADAS. (NÃO SUBLISTAS NO SENTIDO
COMUM). QUANDO SE "APAGA" UMA LISTA, SUA LISTA DESCRITIVA ASSOCIADA
TAMBÉM É APAGADA. O ACESSO A UMA LISTA DESCRITIVA, GERALMENTE SE FAZ PELO
NOME DA LISTA ASSOCIADA.

• ATUALIZAR O VALOR DE UM ATRIBUTO OU INSERIR O ATRIBUTO E SEU VALOR
A LISTA DESCRITIVA:

NEWVAL(AT, VAL, L)

(7890-7970)

O OBJETIVO DESTA FUNÇÃO É DAR O VALOR "VAL" AO ATRIBUTO "AT" NA LISTA
DESCRITIVA DA LISTA "L". VAL SUBSTITUI O VALOR PREVIÓ DE AT, SE O
ATRIBUTO AT NÃO É ACHADO, ELE É ADICIONADO (JUNTAMENTE COM SEU VALOR)
NA LISTA DESCRITIVA.

SE L NÃO TEM LISTA DESCRITIVA, É CRIADA UMA LISTA DESCRITIVA E O ATRIBUTO
E SEU VALOR SÃO ARMAZENADOS NA LISTA DESCRITIVA CRIADA. O VALOR DA FUNÇÃO
É O VALOR ANTIGO DO ATRIBUTO, SE HAVIA, SENÃO SERÁ ZERO.

• APAGAR UM ATRIBUTO E SEU VALOR DA LISTA DESCRITIVA:

NOATVL(AT, L)

(7800-7880)

ESTA FUNÇÃO REMOVE O ATRIBUTO "AT" E SEU VALOR DA LISTA DESCRITIVA

ASSOCIADA COM A LISTA L. O VALOR DA FUNCAO E' O ANTIGO VALOR ASSOCIADO COM AQUELE ATRIBUTO.

. RECUPERAR O VALOR DO ATRIBUTO DA LISTA DESCRITIVA DE UMA LISTA:

ITSVAL(AT,L) (7980-8070)

ESTA FUNCAO TEM COMO SEU VALOR O VALOR DO ATRIBUTO "AT" ARMAZENADO NA LISTA DESCRITIVA DA LISTA "L". SE O ATRIBUTO "AT" NAO E' ACHADO, O VALOR DA FUNCAO E' ZERO.

. ESVAZIAR A LISTA DESCRITIVA DE UMA LISTA:

MTDLST(L) (8080-8150)

ESTA FUNCAO ESVAZIA A LISTA DESCRITIVA DA LISTA "L". SEU VALOR E' O NOME DA LISTA L.

. FORNECER O NOME-DE-LISTA DA LISTA DESCRITIVA DE UMA LISTA:

NAMEDL(L) (8250-8280)

O VALOR DESTA FUNCAO E' O NOME DA LISTA DESCRITIVA DA LISTA "L", QUE E' SEU PARAMETRO DE ENTRADA.

. ASSOCIAR UMA LISTA DESCRITIVA A UMA LISTA:

MAKEDL(L,M) (8160-8240)

ESTA FUNCAO FAZ COM QUE A LISTA "L" SEJA UMA LISTA DESCRITIVA DA LISTA "M". O VALOR DA FUNCAO E' O NOME DA LISTA "M".

VISIT(PLACE)
 VISIT(PLACE,PARMTN(X1,X2,X3,..,XN))

ESTA FUNCAO TEM 1 OU 2 ARGUMENTOS:

- O SEGUNDO ARGUMENTO, SE HOVER, E' UMA CHAMADA DE UMA FUNCAO PARMTN DE N ARGUMENTOS QUE EMPILHA OS N ARGUMENTOS NO TOPO DAS N PRIMEIRAS LISTAS PUBLICAS W(1),...W(N). (N<=100)
- O PRIMEIRO ARGUMENTO E' O ENDEREÇO DE DESVIO A SER EFETUADO POR VISIT OBTIDO PELO COMANDO DO FORTRAN: ASSIGN <NUMERO DE COMANDO> TO PLACE.

VISIT EMPILHA O ENDEREÇO DE RETORNO NORMAL APOS VISIT E DESVIA PARA O ENDEREÇO FORNECIDO POR PLACE. APOS O DESVIO, O PROGRAMA PROSEGUE NORMALMENTE, ATE QUE A ROTINA TERM SEJA CHAMADA, ESTA ROTINA DETERMINA O VALOR DA FUNCAO VISIT.

TERM(RESULT)
 TERM(RESULT,RESTOR(N))

ESTA ROTINA TEM 1 OU 2 ARGUMENTOS.

- O SEGUNDO ARGUMENTO, SE HOVER, E' UMA CHAMADA DA FUNCAO RESTOR(N) QUE RESTAURA OS TOPOS DAS N PRIMEIRAS LISTAS PUBLICAS PARA LAVS.
- O PRIMEIRO ARGUMENTO E' O VALOR DA FUNCAO VISIT.

TERM DESEMPILHA O ENDEREÇO DE RETORNO NORMAL QUE VISIT EMPILHOU, E RETORNA PARA AQUELA PARTE DO PROGRAMA QUE SERIA EXECUTADA IMEDIATAMENTE APOS A ULTIMA CHAMADA DA FUNCAO VISIT.

IV. OPERACOES LOGICAS:

AS SEGUINTE FUNCOES EXISTEM PRINCIPALMENTE PARA AJUDAR A EVITAR INCONVENIENCIAS CAUSADAS PELAS CONVENCOES DE CONVERSAO PONTO-FIXO/ PONTO-FLUTUANTE, ASSOCIADAS COM O FORTRAN.

EQUAL(A,B)

O VALOR DESTA FUNCAO E' ZERO SE OS DOIS PARAMETROS A E B SAO IGUAIS, SENAO, O VALOR DA FUNCAO E' DIFERENTE DE ZERO.

AND(A,B)

O VALOR DESTA FUNCAO E' O "E LOGICO", BIT A BIT DOS DOIS OPERANDOS A E B.

OR(A,B)

O VALOR DESTA FUNCAO E' A UNIAO LOGICA DOS DOIS OPERANDOS A E B.

NOT(A)

O VALOR DESTA FUNCAO E' A NEGACAO BIT A BIT DO PARAMETRO A.

INTGER(FUNCTION-CALL)

ESTA FUNCAO EVITA A TRANSFORMACAO AUTOMATICA DE MODO REAL PARA INTEIRO NUMA CHAMADA DE FUNCAO, CUJO NOME E' IMPLICITAMENTE REAL,

POREM COM VALOR INTEIRO. EX.: I=INTEGER(VISIT(LOCAL,PARMT2(A,B)))

STRDIR(A,B)

ESTA FUNCAO TEM O EFEITO DE B=A, IGNORANDO A POSSIVEL INCOMPATIBILIDADES DE MODOS ENTRE A E B (PONTO-FIXO/FLUTUANTE). O VALOR DA FUNCAO E' O PARAMETRO DE ENTRADA A. (VER DISCUSSAO DAS PRIMITIVAS -STRDIR E STRIND - NA SECCAO II)

. MANIPULACAO DE BITS E CARACTERES.

(DA AUAL IMPIANTACAO - ORIENTADA PARA CARATER)

SQOUT(FIELD,SOURCE)

ESTA FUNCAO TEM COMO OBJETIVO RETIRAR UM CAMPO ESCOLHIDO POR FIELD DO CAMPO SOURCE, O RESULTADO OBTIDO E' O VALOR DA FUNCAO. OS SELETORES DE CAMPO ESTAO EM COMMON /XTRCT/ CP(5), QUE SELECCIONAM OS 5 CAMPOS EM ASCII(7 BITS) DE UMA PALAVRA.

SQIN(FIELD,DATUM,DEST)

SQIN INSERE UM DADO DATUM NUM CAMPO SELECCIONADO DE DEST, POR FIELD, O RESULTADO VOLTA EM DEST E NO VALOR DA FUNCAO.

SHIN(N,DATUM,DEST)

SHIN DESLOCA O DEST DE N BITS A ESQUERDA, INSERINDO O DATUM NO ESPACO VAGADO DE DEST, O RESULTADO VOLTA COMO VALOR DA FUNCAO.

LANORM(WORD)

LANORM AJUSTA O PRIMEIRO CARATER NAO BRANCO DA PALAVRA WORD A ESQUERDA, POR SUCESSIVOS DESLOCAMENTOS A ESQUERDA. O RESULTADO VOLTA COMO VALOR DA FUNCAO, E O PARAMETRO DE ENTRADA NAO E' ALTERADO.

• MANIPULACAO DE BITS E CARACTERES
(DO SLIP CANONICO - ORIENTADA PARA BIT PATTERN)

SQOUT(FIELD, SOURCE)

O PARAMETRO FIELD E' UMA MASCARA. O VALOR DA FUNCAO E' AQUELE CAMPO EXTRAIDO DE SOURCE, E DESLOCADO A ESQUERDA TANTAS VEZES QUANTOS SAO OS BITS ZERO A ESQUERDA DO PRIMEIRO BIT 1 NA MASCARA FIELD. OS ZEROS A DIREITA SAO INTRODUZIDOS NO PROCESSO DE AJUSTAMENTO A ESQUERDA (NA VERSAO CANONICA DO SLIP, UM CONJUNTO DE "MASCARAS DE CARACTERES" SAO PREARMAZENADAS COMO PARTE DO SISTEMA. SEUS LABELS SAO: CP(1), CP(2), ... CP(5) ONDE CP(1) CONSISTE DE 7 BITS DE MAIOR ORDEM IGUAIS A 1. CP(1) EXTRAI O PRIMEIRO CARATER EM ASCII=7 DA PALAVRA.

SQIN(FIELD, DATUM, DEST)

DATUM ESPECIFICADO E' DESLOCADO A DIREITA, PELO NUMERO DE ZEROS A DIREITA NO FIELD ESPECIFICADO, ANTES DO PRIMEIRO BIT 1, DATUM E' INSERIDO NA PALAVRA DEST NO CAMPO ESPECIFICADO POR FIELD.

SHIN(N, DATUM, DEST)

SHIN PROVOCA QUE A PALAVRA DEST SEJA DESLOCADA DE N BITS A ESQUERDA E DATUM SEJA INSERIDO NA PORCAO DE MAIS BAIXA ORDEM VAGADA EM DEST.

LANORM(WORD)

SE CARACTERES ALFANUMERICOS SAO INTRODUZIDOS NUMA PALAVRA POR MEIO DE SHIN, QUANDO ESTA PALAVRA CONTINHA INICIALMENTE CARACTERES BRANCOS, ENTAO OS CARACTERES DE MAIS ALTA ORDEM DAQUELA PALAVRA PODIAM SER BRANCOS. LANORM DESLOCA A PALAVRA ATE' QUE O CARATER DE MAIS ALTA ORDEM DO RESULTADO E' UM CARATER NAO-BRANCO. O VALOR DA FUNCAO E' A PALAVRA DESLOCADA. O PARAMETRO DE ENTRADA NAO E' ALTERADO.

. CONSIDERACOES SOBRE O ARTIGO ORIGINAL [29] DE JOSEPH WEIZENBAUM:

O ARTIGO NO QUAL ESTE TRABALHO FOI BASEADO, E' MUITO BOM E DIDATICO, TENDO A VANTAGEM QUE O AUTOR INTRODUZ OS CONCEITOS NA ORDEM EM QUE ELAS SAO NECESSARIAS, EVITANDO UMA INTRODUCAO TEORICA MUITO EXAUSTIVA. QUANTO A NOMENCLATURA, NAO SEGUE OS PADROES QUE SO' MAIS TARDE FORAM DELINEADAS POR KNUTH [18]; TERMOS COMO TOPO E FUNDO, SAO RELATIVOS A UMA REPRESENTACAO GRAFICA VERTICAL DA LISTA, KNUTH DIZ TOPO DE PILHA E NAO TOPO DE LISTA, POREM CONSIDERANDO AS FACILIDADES DE EXPRESSAO, FOI AINDA MANTIDA.

UM FORTE DO SLIP E' O BOM PLANEJAMENTO QUE WEIZENBAUM FEZ NO TOCANTE AO VALOR QUE AS FUNCOES LIBERAM, FACILITANDO O ENCADEAMENTO DE FUNCOES. O AUTOR APROVEITOU TANTO QUANTO POSSIVEL AS QUALIDADES DO FORTRAN. INTERESSANTE E' O DESPREZO QUE O AUTOR DA' AO MODO DOS PARAMETROS, MISTURANDO MODOS INTEIRO-REAL O QUE SEMPRE E' VALIDO PARA COMPILADORES FORTRAN ONDE NAO HA' PROBLEMAS DE DIFERENCA DE ENDERECAMENTO DE INTEIRO E REAL, O PROGRAMADOR QUE UTILIZAR O SISTEMA SLIP NAO PRECISA PRATICAMENTE SE PREOCUPAR COM O MODO DAS VARIAVEIS E FUNCOES QUE SAO PARAMETROS DE SUBPROGRAMAS, A MENOS QUE ELE DESEJE FAZER UMA ASSIGNACAO DE VALOR.

NO TRABALHO DE WEIZENBAUM FALTAVA UM INDICE DE PROGRAMAS E UMA TABELA DE REFERENCIAS, QUE FORAM AQUI INTRODUIZIDAS PARA FACILITAR A CONSULTA A DESCRICAO DOS VARIOS SUBPROGRAMAS E SUAS RESPECTIVAS LISTAGENS.

AS FUNCOES LNKL, LNKR, ID PODERIAM SER MODIFICADAS, POIS ELAS EXIGEM UM ENCADEAMENTO COM A FUNCAO CONT (OU INHALT) NA MAIORIA DE SUAS CHAMADAS, SE SE CONSIDERA QUE SAO AS PRIMITIVAS MAIS REFERENCIADAS (VER SECAO VIII-TABELA DE REFERENCIAS DOS SIMBOLOS) COMPENSARIA FAZER SUAS MODIFICACOES.

O FATO DE HAVER UMA PERFEITA SIMETRIA, (I.E., OPERACOES QUE SAO EFETUADAS SEGUINDO-SE O APONTADOR LNKR TEM SUAS CORRESPONDENTES SEGUINDO-SE O APONTADOR LNKL) FACILITA A MANIPULACAO DESSAS ESTRUTURAS TANTO QUANTO O APRENDIZADO.

UMA ESTRUTURA DE LISTAS DEVE SER O TANTO QUANTO POSSIVEL ABSTRATA, POREM, A INTERPRETACAO GRAFICA REPRESENTANDO VERTICALMENTE OU HORIZONTALMENTE AS LISTAS, DEPENDENDO DA CONVENIENCIA DO PROGRAMADOR, TRAZ AS SEGUINTE ANALOGIAS:

CELULA DE TOPO : CELULA MAIS-A-ESQUERDA
CELULA DE FUNDO : CELULA MAIS-A-DIREITA
PROXIMA CELULA-ABAIXO : PROXIMA CELULA A DIREITA
PROXIMA CELULA-ACIMA : PROXIMA CELULA A ESQUERDA
APONTADOR PARA A ESQUERDA : APONTADOR PARA A CELULA ANTERIOR ;
APONTADOR PARA CIMA
APONTADOR PARA A DIREITA : APONTADOR PARA A CELULA SEGUINTE ;
APONTADOR PARA BAIXO
AVANCO PARA A DIREITA : AVANCO PARA FRENTE ; AVANCO PARA BAIXO
AVANCO PARA A ESQUERDA : AVANCO PARA TRAZ ; AVANCO PARA CIMA

EXISTEM MUITAS CRITICAS FAVORAVEIS AO SLIP, A PARTIR DE KNUTH, BOBROW, FOSTER, SAKODA, SAMMET E RAMANIS E OUTROS RECONHECIDOS AUTORES, TANTO

• A IMPLEMENTACAO:

• CONSIDERACOES SOBRE ENDERECEAMENTO:

A CAPACIDADE DE ENDERECEAMENTO COM OS CAMPOS LNKL E LNKR DE 16 BITS, RESTRINGE PARA UM ENDERECEAMENTO DE 64K, CONSIDERANDO QUE UMA CELULA TEM DUAS PALAVRAS E AO MENOS UMA DELAS E' DISPENHADA PARA ARMAZENAR APONTADORES, TEMOS MENOS DE 32K PARA INFORMACAO E MAIS DE 32K PARA APONTADORES. PARA PROPOSITOS DE ENSINO, E PEQUENAS APLICACOES, ESTA CAPACIDADE DE ENDERECEAMENTO E' MAIS QUE SUFICIENTE, POREM, PARA APLICACOES DE GRANDE PORTE, TEM-SE A AJUDA DO "GARBAGE COLLECTOR", E A POSSIBILIDADE DE SE GRAVAR LISTAS UTEIS EM DISCO (*), QUANDO LAVS E' ESGOTADA, COM A IDEIA DO "PAGING SLIP" IMPLANTADO NO IBM-1130 POR WEEKS, R. V. (1968) [27], TENDO A DESVANTAGEM DE DIMINUIR A EFICIENCIA DO PROCESSADOR SLIP, COM UM AUMENTO SUBSTANCIAL DE CAPACIDADE DE ARMAZENAMENTO DE INFORMACAO.

(*) OBS.: ESTA POSSIBILIDADE E' COMBATIDA POR MOOERS, C. NA IFIP WORKING CONFERENCE ON SYMBOL MANIPULATION LANGUAGES (1966).

• CONSIDERACOES SOBRE A FORMA E CONTEUDO DESTES TRABALHOS:

NAO TENCIONANDO SER COMPLETO, O PRINCIPAL OBJETIVO DESTES TRABALHOS E' PROVER FACILIDADES PARA FUTUROS TRABALHOS EM LINGUISTICA QUE EXIGEM UM TRABALHO COM ESTRUTURAS DE LISTAS, BEM COMO PARA MANIPULACAO FORMAL (POLINOMIOS, DIFERENCIACAO, INTEGRACAO, NUMEROS RACIONAIS, ETC.) QUE NUMA COMPUTACAO NUMERICA TEM TRATAMENTO DISTINTO DAS POSSIBILIDADES DA COMPUTACAO NAO-NUMERICA OU MISTA.

. COMENTARIOS SOBRE O SLIP:

KNUTH, D. E. (EM [18] PAG.413) DIZ SOBRE GARBAGE COLLECTION:

J. WEIZENBAUM SUGERIU UMA MODIFICACAO INTERESSANTE DA TECNICA DE CONTADOR DE REFERENCIAS. USANDO ESTRUTURAS DE LISTA DUPLAMENTE LIGADAS, ELE COLOCA UM CONTADOR DE REFERENCIAS SOMENTE NA HEADER DE CADA LISTA. ASSIM, QUANDO APONTADORES PERCORREM UMA LISTA, ELAS NAO SAO INCLUIDOS NOS CONTADORES DE REFERENCIAS PARA OS NO'S INDIVIDUAIS, MAS COMO O PROGRAMADOR CONHECE A REGRA PELA QUAL OS CONTADORES DE REFERENCIAS SAO MANTIDOS PARA LISTAS INTEIRAS, ELE CONHECE (TEORICAMENTE) COMO EVITAR REFERENCIAS AS LISTAS COM CONTADOR DE REFERENCIAS NULO. O PROGRAMADOR TAMBEM TEM A HABILIDADE DE EXPLICITAMENTE SOBREPUJAR OS CONTADORES DE REFERENCIAS E DEVOLVER CERTAS LISTAS PARA ARMAZENAMENTO DISPONIVEL (IRALST). ESTAS IDEIAS REQUEREM QUE O PROGRAMADOR TOBE CERTOS CUIDADOS. ELAS PROVAM SER PERIGOSAS NAS MAOS DE PROGRAMADORES INEXPERIENTES E TEM A TENDENCIA DE TORNAR A DEPURACAO DE PROGRAMAS MAIS DIFICIL, DEVIDO AS CONSEQUENCIAS DE REFERENCIAR NO'S QUE TENHAM SIDO APAGADOS. A PARTE MAIS INTERESSANTE DO SISTEMA DE WEIZENBAUM E' COMO ELE TRATA AS LISTAS CUJO CONTADOR DE REFERENCIAS SE ANULA : TAL LISTA E' DEVOLVIDA NO FUNDO DA ATUAL LISTA DE ESPACOS DISPONIVEIS - ISTO E' FACIL DE FAZER EM LISTAS DUPLAMENTE LIGADAS - E SERA' CONSIDERADA ESPACO DISPONIVEL, SOMENTE APOS TODAS AS CELULAS DISPONIVEIS ANTERIORES TEREM SIDO UTILIZADAS; ASSIM, QUANDO OS NO'S INDIVIDUAIS DESTA LISTA SE TORNAM DISPONIVEIS, OS CONTADORES DE REFERENCIAS DAS LISTAS A QUE ELAS FAZEM REFERENCIA, SAO DECREMENTADOS DE 1. ESTA ACAO "RETARDADA" DE APAGAR AS LISTAS E' BASTANTE EFICIENTE COM RESPEITO AO TEMPO DE EXECUCAO; MAS TENDE FAZER COM QUE PROGRAMAS INCORREITOS RODEM (APARENTEMENTE CORRETOS).

FOSTER, J. N. (EM [11] PAG. 39) DIZ DO GARBAGE COLLECTOR DO SLIP:

... ESTE METODO ELEGANTE TEM A VANTAGEM DE QUE UMA CERTA QUANTIDADE DE TEMPO E' ECONOMIZADA PELO FATO DE QUE A DEVOLUCAO DE LISTA PARA A LISTA LIVRE, SO' SE FAZ UMA VEZ, AQUELA GARBAGE COLLECTION SOMENTE E' REQUERIDA QUANDO O ESPACO E' EXIGIDO E FOR DESNECESSARIO, E DURANTE A OPERACAO DO PROGRAMA, E NAO EM PASSOS ISOLADOS. LISTAS CIRCULARES NAO PODEM SER APAGADAS (A MENOS QUE SE ELIMINE A CIRCULARIDADE).

SAKODA, J. M. (EM [6] PAG. 304) DIZ:

"SLIP FOI BEM SUCEDIDO NA INTERCALACAO DO PROCESSAMENTO DE LISTAS, MANIPULACAO DE CADEIAS DE CARACTERES E CAPACIDADES DE ARITMETICA DO FORTRAN, USANDO PALAVRAS EXPLICITAMENTE LIGADAS (A CELULA E' UM PAR DE PALAVRAS CONSECUTIVAS) COMO A FORMA PREDOMINANTE DE ORGANIZACAO DE DADOS, MAS TAMBEM FAZENDO USO DE ARRAYS DO FORTRAN QUANDO NECESSARIO."

RAPHAEL, B. ET AL (EM [6] PAG.18) DIZ:

UMA INTRUSAO BEM SUCEDIDA DE PROCESSAMENTO DE LISTAS EM UMA LINGUAGEM ALGEBRICA NUMERICA.

. VANTAGENS E DESVANTAGENS DO SLIP:

ALEM DAS VANTAGENS E JUSTIFICATIVAS APRESENTADAS SOBRE O SLIP NA INTRODUCAO (SECAO I), TEMOS ALGUMAS CONSIDERACOES:

FOSTER, J. M. (EM [11] PAG. 47-48) MENCIONA AS SEGUINTE VANTAGENS E DESVANTAGENS DO SLIP:

VANTAGENS:

- 1) TODA A POTENCIA DO FORTRAN PARA ARITMETICA ESTA' DISPONIVEL, ALIADA AO GRANDE NUMERO DE PROGRAMAS JA EXISTENTES.
- 2) AS ROTINAS DE E/S DO FORTRAN SAO DISPONIVEIS, INCLUSIVE O USO DE ARMAZENAMENTO AUXILIAR (DISCOS, FITAS, ETC.).
- 3) UMA LISTA PODE SER SUBLISTA DE VARIAS LISTAS (O CONTADOR DE REFERENCIAS NO HEADER CONTA O NUMERO DE VEZES QUE UMA LISTA E' USADA COMO SUBLISTA).
- 4) LISTAS CIRCULARES PODEM SER CRIADAS COM O USO DAS OPERACOES PRIMITIVAS, POREM, PARA SEREM RETORNADAS PARA LAVS, A CIRCULARIDADE DEVE SER ELIMINADA.

DESVANTAGENS:

- 1) O PROCESSAMENTO DE LISTAS NAO E' MUITO RAPIDO, PORQUE AS OPERACOES SAO EFETUADAS POR CHAMADAS DE SUBROTINAS, IMPLICANDO NUM ELEVADO NUMERO DE DESVIOS E TRANSMISSOES DE DADOS.
- 2) PROGRAMAS E DADOS SAO ESTRUTURAS DISTINTAS NO SLIP, IMPOSSIBILITANDO QUE UM PROGRAMA SE MODIFIQUE A SI PROPRIO.
- 3) UM GRANDE NUMERO DE SUBPROGRAMAS (APROXIMADAMENTE 80) SAO DISPONIVEIS, SENDO DE DIFICIL MEMORIZACAO.
- 4) A RECURSAO PODE SER FEITA, POREM O PROGRAMADOR TEM QUE MANIPULAR OS PARAMETROS E OS RESULTADOS.

. OUTRAS DESVANTAGENS DO SISTEMA SLIP:

- 1) UMA CELULA PARTICULAR SO' PODE PARTICIPAR DE UMA LISTA DEVIDO A EXISTENCIA DE SOMENTE DOIS APONTADORES.
- 2) COMPARADA COM OUTROS SISTEMAS DE PROCESSAMENTO DE LISTAS, O SLIP ARMAZENA O DADO NO CAMPO DE INFORMACAO, ONDE OUTROS SISTEMAS ARMAZENAM O ENDEREÇO DO DADO E OUTRAS INFORMACOES, TAIS COMO: INICIO DE UMA CADEIA DE CARACTERES E SEU COMPRIMENTO.

. OUTRAS VANTAGENS DO SISTEMA SLIP:

- 1) A GARBAGE COLLECTION PODE SER TOTALMENTE CONTROLADA PELO PROGRAMADOR.
- 2) COMO UMAS POUCAS FUNCOES PRIMITIVAS TEM QUE SER PROGRAMADAS EM LINGUAGEM DE BAIXO-NIVEL, E TODAS AS OUTRAS OPERACOES SAO PROGRAMADAS NUMA LINGUAGEM DE ALTO-NIVEL (FORTRAN), HA' UMA (QUASE) TOTAL COMPATIBILIDADE ENTRE AS VARIAS IMPLEMENTACOES.

. OUTRAS IMPLEMENTACOES DE SLIP:

IBM 7090, IBM 7094
CDC 1604, CDC 3200
IBM 7044, IBM 1620
PHILCO 2000, ATLAS
IBM 360, AN/F8Q-32
IBM 1130, IBM 370

VI - BIBLIOGRAFIA:

- [1] BALL, W. E. HERAS, R. I.
 AUTOMAST - AUTOMATIC MATHEMATICAL ANALYSIS AND SYMBOLIC TRANSLATION
 COMMUNICATIONS OF THE ACM
 VOLUME 9, NUMBER 8 PP 626-633
 AUGUST 1966 (PRESENTED AT SICSAM SYMPOSIUM)
 AUTOMAST SLIP POLYNOMIALS DIFFERENTIAL EQUATIONS
 THE PAPER FIRST DESCRIBES A PROCEDURE FOR NUMERICALLY SOLVING SYSTEMS
 OF ORDINARY DIFFERENTIAL EQUATIONS WHICH IS SHOWN TO ALSO
 GENERATE SYMBOLIC SOLUTIONS. THEN IT DESCRIBES A 7072 COMPUTER
 PROGRAM WHICH READS A SYSTEM OF EQUATIONS AND GENERATES THE
 EXPANSIONS FOR ALL THE DEPENDENT VARIABLES.
- [2] BARRIERI, R.
 COMPUTER LIST PROCESSING LANGUAGES
 IBM, DATA SYSTEMS DIVISION, POUGHKEEPSIE, NEW YORK
 TECHNICAL REPORT NO. TR00.1209 32 PAGES
 NOVEMBER 1964
 SLIP FLPL IPL-V LISP IMPLEMENTATION TECHNIQUES LIST
 THIS REPORT DESCRIBES SEVERAL LISP STRUCTURES AND SOME IMPORTANT ASPECTS
 OF THEIR IMPLEMENTATION. IT DESCRIBES AND COMPARES FLPL, IPL-V,
 LISP AND SLIP. IT ALSO GIVES EXAMPLES OF THE USE OF SLIP IN
 MANIPULATING POWER SERIES.
- [3] BARRON, D. W.
 RECURSIVE TECHNIQUES IN PROGRAMMING
 MC DONALD/ELSEVIER COMPUTER MONOGRAPHS
 1969 (3RD EDITION) LONDON
 RECURSION APPLICATION ITERATION IPL-V ALGOL LISP
 KDF-9 B5000
- [4] BOBROW, D. G. RAPHAEL, B.
 A COMPARISON OF LIST-PROCESSING LANGUAGES
 COMMUNICATIONS OF THE ACM
 VOLUME 7, NUMBER 4 PP 231-240
 APRIL 1964
 COMIT IPL-V LISP SLIP LIST STRING
 CONTAINS A DETAILED COMPARISON OF COMIT, IPL-V, LISP 1.5 AND SLIP
 WITH RESPECT TO DATA REPRESENTATIONS, STORAGE ALLOCATION, PROGRA-
 MMING NOTATION, SPECIAL PROCESSES AVAILABLE, AND USUABILITY.
 A DESCRIPTION OF COMMON FEATURES OF LIST-PROCESSING LANGUAGES IS
 GIVEN.
- [5] BOBROW, D. G. WEIZENBAUM, J.
 LIST PROCESSING AND EXTENSION OF LANGUAGE FACILITY BY IMBEDDING
 IEEE TRANSACTIONS ON ELECTRONIC COMPUTERS
 VOLUME EC-13, NUMBER 4 PP 395-400
 AUGUST 1964
 METEOR COMIT SLIP LIST
 TWO EXTENSIONS TO DO LIST PROCESSING ADDED TO EXISTING LANGUAGES ARE
 DESCRIBED, WITHIN THE FRAMEWORK OF A DISCUSSION OF THE PROBLEM
 OF LANGUAGE EXTENSIONS AND IMBEDDING. THE LANGUAGES DESCRIBED
 ARE METEOR AND SLIP.

- [6] BOBKOW, D. G. (ED.)
 SYMBOL MANIPULATION LANGUAGES AND TECHNIQUES
 PROCEEDINGS OF THE IFIP WORKING CONFERENCE ON SYMBOL MANIPULATION
 LANGUAGES.
 NORTH-HOLLAND PUBL. CO. - AMSTERDAM 1968 487 PAGES
- [7] COLLINS, G. E.
 REFCO III. A REFERENCE COUNT LIST PROCESSING SYSTEM FOR THE IBM 7094
 IBM, WATSON RESEARCH CENTER, YORKTOWN HEIGHTS, N. Y.
 RESEARCH REPORT RC-1436 33 PAGES
 MAY 1965
 REFCO LISP SLIP LIST TECHNIQUES
 THIS DESCRIBES A SERIES OF LIST PROCESSING SUBROUTINES DESIGNED FOR USE
 WITH FORTRAN II AND FMS. A COMPARISON BETWEEN REFCO III AND SLIP
 AND GARBAGE COLLECTION SCHEMES AND LISP IS GIVEN.
- [8] DIGITAL EQUIPMENT CORPORATION
 DEC-SYSTEM 10 MATHEMATICAL LANGUAGES HANDBOOK
 1972 (2ND EDITION) DEC-10-KRZB-D 388 PAG.
 FORTRAN BASIC ALGOL
- [9] DIGITAL EQUIPMENT CORPORATION
 DEC-SYSTEM10 ASSEMBLY LANGUAGE HANDBOOK
 1972 (2ND EDITION) DEC-10-NRZB-D
 SYSTEM-REFERENCE MACRO MONITOR-CALLS LOADER DOT UTILITIES
- [10] DIGITAL EQUIPMENT CORPORATION
 DEC-SYSTEM10 USERS HANDBOOK
 1972 (2ND EDITION) DEC-10-NGZB-D 836 PAGES
 SOFTWARE TIME-SHARING BATCH TECO LINED PIP COMMANDS
- [11] FOSTER, J. M.
 LIST PROCESSING
 MC DONALD/ELSEVIER COMPUTER MONOGRAPHS
 LONDON 1969 (5TH EDITION)
 LIST STRUCTURE SLIP IPL-V LISP FLPL COMIT
- [12] FRAGA, P. D.
 LINGUAGEM FORTRAN - PROGRAMACAO E EXERCICIOS
 DEPTO DE MATEMATICA - EESC - USP (1967)
- [13] FRAGA, P. D.
 ESTRUTURAS DE DADOS E SUA MANIPULACAO EM APL
 IMECC - UEC (1972)
 LISP-APL ARVORE LISTA LINEAR
- [14] FURTADO, A. L.
 TEORIA DOS GRAFOS - ALGORITMOS
 LIVROS TECNICOS E CIENTIFICOS, EDITORA, SA
 RIO DE JANEIRO, 1973 155 PAG.
 ALGORITMO GRAFO LISTA TEORIA DE PROGRAMACAO PL/I ALGOL
 LISP
- [15] GELEKNER, H. ET AL.
 A FORTRAN-COMPILER LIST-PROCESSING LANGUAGE
 JOURNAL OF THE ACM 7,2 (APRIL 1960)

- [22] NEWELL, A., ET AL.
 INFORMATION PROCESSING LANGUAGE V MANUAL
 RAND CORP. P191B (MAR 1960)
- [23] NOVAK, R. J.
 ALGORITHMS FOR ALGEBRAIC MANIPULATIONS - PROJECT REPORT
 INSTITUTE OF COMPUTER SCIENCE, UNIVERSITY OF TORONTO,
 TORONTO, CANADA
 SEPTEMBER 1965 17 PAGES
 (UNPUBLISHED)
 ALGEM SLIP IMPL POLY
 THIS IS A BRIEF DESCRIPTION AND STATUS REPORT ON A PROJECT AIMED AT
 MANIPULATING EXPRESSIONS WHICH USING THE FOUR ARITHMETIC
 OPERATORS, AND EXPONENTIATION, AND LEAVING THE RESULT IN A
 CANONICAL FORM. THE PROGRAM USES SLIP.
- [24] PERLIS, A. J. ET AL.
 SYMBOL MANIPULATION BY THREADED LISTS
 COMMUNICATIONS OF THE ACM 3(1960) PP 195-204
- [25] RAMANI, S.
 SLIP OPERATIONS ON TREES AND THEIR RELEVANCE TO PROBLEMS OF LINGUISTIC
 INTEREST
 SYMBOL MANIPULATION LANGUAGES AND TECHNIQUES, PROCEEDINGS OF THE IFIP
 WORKING CONFERENCE ON SYMBOL MANIPULATION LANGUAGES
 BOBROW, D. G. (ED.)
 NORTH HOLLAND PUBLISHING COMPANY, AMSTERDAM, NETHERLANDS
 1968 PP 312-339
 SLIP
 THIS PAPER DEALS A PROGRAMMING TECHNIQUE FOR DEALING WITH THEM.
- [26] RAPHAEL, B. BOBROW, D. G. FEIN, L.
 A BRIEF SURVEY OF COMPUTER LANGUAGES FOR SYMBOLIC AND ALGEBRAIC
 MANIPULATION
 SYMBOL MANIPULATION LANGUAGES AND TECHNIQUES, PROCEEDINGS OF THE IFIP
 WORKING CONFERENCE ON SYMBOL MANIPULATION LANGUAGES
 BOBROW, D. G. (ED.)
 NORTH HOLLAND PUBLISHING COMPANY, AMSTERDAM, NETHERLANDS
 1968 PP 1-54
 ALTRAN ANBIT COGENT COMIT CONVERT CORAL DYSTAL FLIP IPL-V
 FORMULA-ALGOL LISP LISP2 L6 PANON SLIP SNOBOL
 THIS PAPER (WHICH WAS DONE BY ACM SICSAM SUBCOMMITTEE ON LANGUAGE
 COMPARISON) CLASSIFIES 17 LANGUAGES INTO 6 GROUPS. COMMON
 PROPERTIES OF EACH GROUP ARE GIVEN, A VERY BRIEF DESCRIPTION
 AND PROGRAM EXERPT FOR EACH LANGUAGE IS GIVEN, AND SOME VERY
 BRIEF COMPARISONS ARE MADE. THIS PAPER WAS MADE VERY SHORT TO MEE
 PUBLISHING CONSTRAINTS.

- [27] WEEKS, R. V.
SLIP - PROCESSADOR DE LISTAS SIMETRICAS
UFRJ-COPPE - RIO DE JANEIRO
OUTUBRO 1968
SLIP LISTA ESTRUTURA LAVS PAGING-SLIP 132 PAG.
- [28] WEIZENBAUM, J.
KNOTTED LIST STRUCTURES
COMMUNICATIONS OF THE ACM
VOLUME 5, NUMBER 3 PP 161-165
MARCH 1962
LIST TECHNIQUES
THE STATED MOTIVATION OF THE PAPER IS MACHINE ORGANIZATION, NOT LANGUAGE.
THE PAPER DESCRIBES A TYPE OF LIST STRUCTURE AND OPERATIONS WHICH
CAN BE PERFORMED ON IT. IT ALSO DISCUSSES THE PROBLEM OF
ERASING LISTS.
- [29] WEIZENBAUM, J.
SYMMETRIC LIST PROCESSOR
COMMUNICATIONS OF THE ACM
VOLUME 6, NUMBER 9 PP 524-544
SEPTEMBER 1963
SLIP LIST TECHNIQUES
THIS DESCRIBES IN COMPLETE DETAIL A SET OF SUBROUTINES TO BE ADDED TO
FORTRAN TO DO LIST PROCESSING. THE SUBROUTINES ARE THEMSELVES
CODED IN FORTRAN AND A LISTING IS INCLUDED. THIS SYSTEM (WHOSE
NAME IS ABBREVIATED AS SLIP) IS BRIEFLY COMPARED TO EARLIER SYSTE
- [30] WEIZENBAUM, J.
ELIZA - A COMPUTER PROGRAM FOR THE STUDY OF NATURAL LANGUAGE
COMMUNICATION BETWEEN MAN AND MACHINE.
COMMUNICATIONS OF THE ACM
VOLUME 9, NUMBER 1 PP 36-45
JANUARY 1966
SLIP ARTIFICIAL INTELLIGENCE
- [31] WEIZENBAUM, J.
CONTEXTUAL UNDERSTANDING BY COMPUTERS
COMMUNICATIONS OF THE ACM
VOLUME 10, NUMBER 8 PP 474-480
AUGUST 1967
SLIP ARTIFICIAL INTELLIGENCE
- [32] WEIZENBAUM, J.
RECOVERY OF REENTRANT LIST STRUCTURES IN SLIP
COMMUNICATIONS OF THE ACM
VOLUME 12, NUMBER PP 370-372
JULY 1969
SLIP LIST-PROCESSING GARBAGE-COLLECTION

	TITLE	PRIMITIVAS
ID:	ENTRY	ID
	Ø	
	MOVE	Ø,ØØ(16)
	LSH	Ø,-42
	JRA	16,1(16)
MADOV:	ENTRY	MADOV
	Ø	
	HPRZ	Ø,(16)
	JRA	16,1(16)
ID2:	ENTRY	ID2
	Ø	
	MOVE	Ø,ØØ(16)
	LSH	Ø,22
	LSH	Ø,-42
	JRA	16,1(16)
LNKR:	ENTRY	LNKR
	Ø	
	MOVE	Ø,ØØ(16)
	LSH	Ø,24
	LSH	Ø,-24
	JRA	16,1(16)
LNKL:	ENTRY	LNKL
	Ø	
	MOVE	Ø,ØØ(16)
	LSH	Ø,2
	LSH	Ø,-24
	JRA	16,1(16)
M1:	ENTRY	SETDIR
	OCT	777777777777
MI:	OCT	177777777777
ML:	OCT	6ØØØØØ777777
MR:	OCT	7777776ØØØØØ
	INTEGER	FIELD,AFIELD
SETDIR:	Ø	
	MOVE	Ø,3(16)
	HRRZM	Ø,AFIELD
	MOVE	Ø,ØAFIELD
	MOVEM	Ø,FIELD
SETI:	MOVE	1,ØØ(16)
	CAMN	1,M1
	JUMPA	SETL
	LSH	1,42
	MOVE	Ø,M1
	ANDM	Ø,FIELD
	IORM	1,FIELD
SETL:	MOVE	1,@1(16)
	CAMN	1,M1
	JUMPA	SETR
	LSH	1,24
	LSH	1,-2
	MOVE	Ø,ML
	ANDM	Ø,FIELD
	IORM	1,FIELD

```

SETR:  MOVE 1, R2(16)
        CAMM 1, R1
        JUMPA RET
        LSH 1, 24
        LSH 1, -24
        MOVE 0, MR
        ANDM 0, FIELD
        IORM 1, FIELD
RET:    MOVE 0, FIELD
        MOVEM 0, @AFIELD
        JRA 16, 4(16)

SETIND: ENTRY SETIND
        0
        MOVE 0, @3(16)
        HRRZM 0, AFIELD
        MOVE 0, @AFIELD
        MOVEM 0, FIELD
        JUMPA SETI

MI2:   ENTRY SETID2
        OCT 777777177777
SETID2: 0
        MOVE 0, @1(16)
        MOVEM 0, FIELD
SETI2:  MOVE 0, @2(16)
        LSH 0, 42
        LSH 0, -22
        MOVE 1, MI2
        ANDM 1, FIELD
        IOR 0, FIELD
        MOVEM 0, @1(16)
        JRA 16, 2(16)

CONT:  ENTRY CONT
        0
        MOVE 0, @0(16)
        LSH 0, 24
        LSH 0, -24
        HRRZM 0, AFIELD
        MOVE 0, @AFIELD
        JRA 16, 1(16)

INHALT: ENTRY INHALT
        0
        JUMPA CONT+1

INTGER: ENTRY INTGER
        0
        MOVE 0, @0(16)
        JRA 16, 1(16)

STRDIR: ENTRY STRDIR
        0
        MOVE 0, @0(16)
        MOVEM 0, @1(16)
        JRA 16, 2(16)

STRIND: ENTRY STRIND
        0

```

MOVE 0,81(16)
LSH 0,24
LSH 0,-24
HRRZN 0,AFTELO
MOVE 0,80(16)
MOVEM 0,8AFIELD
JRA 16,2(16)

```

SUBTTL RECURSÃO
ENTRY VISIT
INTEGER FIEL, AFIEL
EXTERN NEWTOP
ALIAS: XWD STACK, STACK
STACK: OCT 4000000000000,0
KFRST: DEC 1
NOPARG: OCT 457000000000
VISIT: Z

```

```

MOVE @,0(16)
HRRZM @,FIEL
HRRZM 16,AFIEL
AOS @,AFIEL
MOVE @,1(16)
AND @,NOPARG
JUMPN @,FIRST
AOS @,AFIEL
FIRST: MOVE @,KFRST
CALL @,1
JUMPA POPDWN
AOS @,KFRST
MOVE @,ALIAS
IURN @,STACK

```

```

;
POPDWN: JSA 16,@NEWTOP
ARG AFIEL
ARG ALIAS
JUMPA AFIEL

```

EMPILHAR ENDERECO DE RETORNO

; DESVIA PARA O ENDERECO DADO

```

TERM: ENTRY TERM
EXTERN POPTOP
MOVE @,0(16)
MOVEN @,FIEL
JSA 16,POPTOP
ARG ALIAS
MOVEN @,AFIEL
MOVE @,FIEL
JUMPA @AFIEL

```

; DESEMPILHAR ENDERECO DE RETORNO

; VALOR DA FUNCAO VISIT
; RETORNO PARA APOS VISIT CALL.

```
SUBTTL LOGICAS
ENTRY  EQUAL
EQUAL: 0
MOVE  0,0(16)
MOVEM 0,FIELD
MOVE  0,01(16)
ANDCM 0,FIELD
JRA   16,2(16)

ENTRY  AND
AND:   0
MOVE  0,0(16)
MOVEM 0,FIELD
MOVE  0,01(16)
AND   0,FIELD
JRA   16,2(16)

ENTRY  OR
OR:    0
MOVE  0,0(16)
MOVEM 0,FIELD
MOVE  0,01(16)
IOR   0,FIELD
JRA   16,2(16)

ENTRY  NOT
NOT:   0
MOVE  0,0(16)
SETCA 0
JRA   16,1(16)

END
```

LISTAGEM DOS SUBPROGRAMAS-FONTE EM FORTRAN IV
INCLUINDO ERRATA DEVIDA A:

RUSSELL, D. B. "ON THE IMPLEMENTATION OF SLIP" - (ICT-ATLAS-1)
IN CACM 8,5 (MAY 1965) PAGE 263.

```

00010          SUBROUTINE INITAS(M,N)
00011 C          INITAS CRIA A LISTA DE ESPACOS DISPONIVEIS NUM ARRAY M
00012 C          DE N PALAVRAS, DIMENSIONADO NO PROGRAMA PRINCIPAL.
00013 C          ALEM DE CRIAR OS ALIAS'ES DE 100 LISTAS PUBLICAS, QUE O
00014 C          PROGRAMADOR PODE TER ACESSO, USANDO COMMON AVSL,X(100)
00015 C          NOS SUBPROGRAMAS QUE NECESSITA DAS LISTAS PUBLICAS
00016 C          E NO PROGRAMA PRINCIPAL, PARA ALOCACAO DE MEMORIA.
00020 C
00030          COMMON AVSL,X(100)
00040          DIMENSION W(200),M(2)
00050          DO 1 I=1,100
00060              J=2*I-1
00070          CALL STRDIR(STRDIR(STRDIR(0,X(I)),W(J)),W(J+1))
00080          CALL SETDIR(0,MADOV(W(J)),MADOV(W(J)),X(I))
00090          CALL SETDIR(2,MADOV(W(J)),MADOV(W(J)),W(J))
00100 1          CALL SETDIR(-1,-1,4095,W(J+1))
00110          DO 2 I=1,N
00120 2              M(I)=0
00130              K=N-2
00140          DO 3 I=1,K,2
00150 3          CALL SETDIR(-1,-1,MADOV(M(I+2)),M(I))
00160          CALL SETDIR(0,MADOV(M(N-1)),MADOV(M(1)),AVSL)
00170          RETURN
00180          END
00190          FUNCTION NUCELL(X)
00191 C          NUCELL RETIRA UMA CELULA DA LISTA DE ESPACOS DISPONIVEIS,
00192 C          SE HOVER, A CELULA FICA COM CONTEUDO 0 EM AMBAS PALAVRAS, E
00193 C          SEU ENDEREÇO E' DISPONIVEL COMO VALOR DA FUNCAO NUCELL.
00194 C          A FUNCAO NUCELL TEM UM ARGUMENTO DUMMY X, NECESSARIO
00195 C          APENAS PARA DEFINICAO E UTILIZACAO DE FUNCAO EM FORTRAN.
00196 C          O PROGRAMADOR NAO PRECISA APELAR PARA NUCELL.
00200 C
00210          COMMON AVSL
00220          M = LNKR(AVSL)
00230          IF (M) 1,2,1
00240 2          PRINT 901
00250          STOP
00260 1          IF (ID(CONT(M))-1)3,4,3
00270 4          CALL IRALST(CONT(M+1))
00280 3          CALL SETDIR(-1,-1,LNKR(CONT(M)),AVSL)
00290          CALL STRIND(0,M)
00300          CALL STRIND(0,M+1)
00310          NUCELL = M
00320          RETURN
00330 901          FORMAT(1H1,6X,'A LISTA DE ESPACOS DISPONIVEIS (LAVS) ESTA ' '
00340          1,'ESGOIADA - PROGRAMA CANCELADO' )
00350          END
00360          SUBROUTINE RCELL(CELL)
00361 C          RCELL DEVOLVE PARA LAVS A CELULA CUJO ENDEREÇO ESTA EM CELL.
00362 C          POSSIBILITANDO AO PROGRAMADOR DEVOLVER CELULAS PARA
00363 C          A LISTA DE ESPACOS DISPONIVEIS, CELL PODE SER O ENDEREÇO DE
00364 C          DE UMA CABECA DE LISTA. RCELL POSSIBILITA QUE A
00365 C          GARBAGE COLLECTION SEJA CONTROLADA PELO PROGRAMADOR.
00370 C
00380          COMMON AVSL
00390          CALL SETIND(-1,-1,CELL,LNKL(AVSL))
00400          CALL SETDIR(-1,CELL,-1,AVSL)
00410          CALL SETIND(-1,-1,0,CELL)
00420          RETURN
00430          END

```



```

00440                                FUNCTION LIST(K)
00441 C    LIST(K) CRIA UMA LISTA VAZIA E DEIXA SEU NOME TANTO
00442 C    NA CELULA K COMO NO VALOR DA FUNCAO, A CELULA K SE TORNA UM
00443 C    ALIAS PARA ESTA LISTA VAZIA.
00444 C    LIST POR SER FUNCAO PODE SER ENCADEADA.
00445 C    SE K=9, O CONTADOR DE REFERENCIA PARA A LISTA E' 0, SENAO E' 1.
00450 C
00460 C
00470             LIST = NUCELL(Z)
00480 CALL SETDIR(0,LIST,LIST,LIST)
00490 CALL SETIND(2,LIST,LIST,LIST)
00500             IF( K-9 ) 2,1,2
00510 2    CALL SETIND(-1,-1,1,LIST+1)
00520             K = LIST
00530 1    RETURN
00540 END

00550                                FUNCTION MTLIST(P)
00551 C    A LISTA CUJO NOME ESTA NO PARAMETRO P E' ESVAZIADA, I.E.,
00552 C    SUAS CELULAS SAO RESTAURADAS A LAVS, O VALOR DA FUNCAO E'
00553 C    O NOME DA LISTA VAZIA P.
00560 C
00570 COMMON AVSL
00580 M = LOCI(P)
00590 IF(LISTMT(P))3,4,3
00600 3    LR = LNKR(CONT(M))
00610     LL = LNKL(CONT(M))
00620     CALL SETIND(-1,M,M,M)
00630     CALL SETIND(-1,-1,LR,LNKL(AVSL))
00640     CALL SETDIR(-1,LL,-1,AVSL)
00650             CALL SETIND(-1,-1,0,LNKL(AVSL))
00660 4    MTLIST = M
00670     RETURN
00680 END

00690                                FUNCTION IRALST(P)
00691 C    A LISTA CUJO NOME APARECE NO PARAMETRO DE ENTRADA DA FUNCAO
00692 C    E' APAGADA, I.E., TODAS SUAS CELULAS, INCLUSIVE O HEADER SAO
00693 C    RESTAURADAS A LAVS POR ESTA FUNCAO.
00694 C    ENTRETANTO, ESTA LISTA (E SEU CONTEUDO) PODE AINDA SER UMA
00695 C    SUBLISTA DE OUTRA LISTA, A DELECAO NAO E' EFETUADA
00696 C    SE ESTA CONDICAO E' DETETADA, NESTE CASO, O CONTADOR DE
00697 C    REFERENCIA E' DECREMENTADO DE 1.
00698 C    O VALOR DA FUNCAO IRALST E' A QUANTIDADE DE LISTAS DAS QUAIS
00699 C    A LISTA DADA E' SUBLISTA, E TEM VALOR 0 SE FOI APAGADA.
00700 C
00710             L = LOCI(P)
00720 CALL SETIND(-1,-1,LCNTR(L)-1,L+1)
00730             IRALST = LCNTR(L)
00740             IF(IRALST)1,2,1
00750 2    CALL MTLIST(P)
00751             N = LNKL(CONT(L+1))
00752             IF(N)3,4,3
00753 3    NEW = NUCELL(Z)
00754     CALL SETIND(1,-1,-1,NEW)
00755     CALL SETIND(-1,N,N,NEW+1)
00756     CALL RCELL(NEW)
00760 4    CALL RCELL(L)
00770 1    RETURN
00780 END

00790                                FUNCTION NULST(LNKP,LNKH)
00791 C    O PARAMETRO LNKP E' O ENDERECO DE UMA CELULA DA LISTA

```

```

00792 C      CUJO NOME ESTA EM LNKH. O OBJETIVO DA FUNCAO NULSTL E'
00793 C      DIVIDIR A LISTA LNKH CRIANDO UMA NOVA LISTA QUE TEM TODAS
00794 C      AS CELULAS A ESQUERDA DA CELULA LNKP, INCLUSIVE A CELULA
00795 C      LNKP E SEUS MEMBROS.
00796 C      AS CELULAS DA NOVA LISTA CRIADA SAO REMOVIDAS DA LISTA LNKH.
00797 C      O NOME DA NOVA LISTA CRIADA E' O VALOR DA FUNCAO.
00800 C
00810      NULSTL = LIST(9)
00820      IF(ID(CONT(LNKP))-2)1,2,1
00830 2      CALL SETIND(2,NULSTL,NULSTL,NULSTL)
00840      RETURN
00850 1      LTOP = LNKR (CONT(LNKH))
00860      LSUC = LNKR (CONT(LNKP))
00870      CALL SETIND(-1,-1,LSUC,LNKH)
00880      CALL SETIND(-1,LNKH,-1,LSUC)
00890      CALL SETIND(2,LNKP,LTOP,NULSTL)
00900      CALL SETIND(-1,-1,NULSTL,LNKP)
00910      CALL SETIND(-1,NULSTL,-1,LTOP)
00920      RETURN
00930      END
00940      FUNCTION NULSTR(LNKP,LNKH)
00941 C      A FUNCAO NULSTR E' ANALOGA 'A NULSTL, EXCETO QUE TODAS AS
00942 C      CELULAS A DIREITA DE LNKP , INCLUSIVE A CELULA LNKP E
00943 C      SEUS MEMBROS SAO REMOVIDAS DA LISTA LNKH,E
00944 C      COLOCADAS NA NOVA LISTA CRIADA.
00945 C      O NOME DA NOVA LISTA CRIADA E' O VALOR DA FUNCAO.
00950 C
00960      NULSTR = LIST(9)
00970      IF(ID(CONT(LNKP))-2)1,2,1
00980 2      CALL SETIND(2,NULSTR,NULSTR,NULSTR)
00990      RETURN
01000 1      LBOT = LNKL(CONT(LNKH))
01010      LPRE = LNKL(CONT(LNKP))
01020      CALL SETIND(-1,LPRE,-1,LNKH)
01030      CALL SETIND(-1,-1,LNKH,LPRE)
01040      CALL SETIND(2,LBOT,LNKP,NULSTR)
01050      CALL SETIND(-1,NULSTR,-1,LNKP)
01060      CALL SETIND(-1,-1,NULSTR,LBOT)
01070      RETURN
01080      END
01090      FUNCTION NEWTOP(P,Q)
01091 C      NEWTOP EMPILHA O DADO P NO TOPO DA LISTA Q.
01092 C      ESTA FUNCAO TEM COMO VALOR O ENDERECO DA CELULA QUE FOI
01093 C      RETIRADA DA LAYS, PARA SER ARMAZENADO O DADO P E OS
01094 C      APONTADORES ASSOCIADOS.
01100 C
01110      NEWTOP = NXTRGT(P,LOCT(Q))
01120      RETURN
01130      END
01140      FUNCTION NEWBOT(P,Q)
01141 C      NEWBOT E' ANALOGA A NEWTOP, EXCETO QUE O DADO P E'
01142 C      EMPILHADO NO FUNDO DA LISTA Q.
01150 C
01160      NEWBOT = NXTLFT(P,LOCT(Q))
01170      RETURN
01180      END
01190      FUNCTION NXTLFT(M,A)
01191 C      NXTLFT INSERE O DADO M A ESQUERDA DA CELULA CUJO ENDERECO
01192 C      DE MAQUINA E' ESPECIFICADO POR A. O VALOR DA FUNCAO E' O
01193 C      ENDERECO DE MAQUINA DA CELULA QUE FOI TIRADA PARA ARMAZENAR

```

```

01194 C O DADO E OS APONTADORES. OS ENDEREÇOS DE MAQUINA DE
01195 C CELULAS DE LISTA SAO CONHECIDOS, PRINCIPALMENTE, POR
01196 C OPERACOES DE INSERCAO DE DADOS, E POR ACESSO AO LPNTR
01197 C DAS READER'S.
01200 C
01210 IL = NUCELL(Z)
01220 NXLEFT = IL
01230 LL = LNKL(CONT(A))
01240 CALL SETIND(-1,-1,IL,LL)
01250 CALL SETIND(-1,IL,-1,A)
01260 CALL SETIND(0,LL,A,IL)
01270 IF(NAMTST(M))1,2,1
01280 2 CALL SETIND(1,-1,-1,IL)
01290 CALL SETIND(-1,-1,LCNTR(M)+1,M+1)
01300 1 CALL STRIND(M,IL+1)
01310 RETURN
01320 END
01330 FUNCTION NXTRGT(M,A)
01331 C INSERE O DADO M A DIREITA DA CELULA CUJO ENDEREÇO E'
01332 C ESPECIFICADO POR A, SENDO ANALOGA A FUNCAO NXLEFT.
01340 C
01350 IR = NUCELL(Z)
01360 NXTRGT = IR
01370 LR = LNKR(CONT(A))
01380 CALL SETIND(-1,IR,-1,LR)
01390 CALL SETIND(-1,-1,IR,A)
01400 CALL SETIND(0,A,LR,IR)
01410 IF(NAMTST(M))1,2,1
01420 2 CALL SETIND(1,-1,-1,IR)
01430 CALL SETIND(-1,-1,LCNTR(M)+1,M+1)
01440 1 CALL STRIND(M,IR+1)
01450 RETURN
01460 END
01470 FUNCTION INLSTL(M,N)
01471 C M DEVE SER O ALIAS DE UMA LISTA, E N O ENDEREÇO DE MAQUINA
01472 C DE UMA CELULA DE LISTA. INLSTL TOMA O CONJUNTO DE CELULAS
01473 C LIGADAS QUE CONSTITUEM O CORPO DA LISTA M (I.E., TODAS AS
01474 C CELULAS DA LISTA M EXCETO O HEADER), E INSERE A ESQUERDA DA
01475 C CELULA N, AUMENTANDO ASSIM A LISTA QUE N PERTENCE.
01476 C A LISTA M E' ESVAZIADA E SEU NOME VOLTA COMO VALOR DA FUNCAO.
01480 C
01490 L = LOCT(M)
01500 ITOP = LNKR(CONT(L))
01510 IBOT = LNKL(CONT(L))
01520 INLSTL = L
01530 CALL SETIND(-1,L,L,L)
01540 IPRE = LNKL(CONT(N))
01550 CALL SETIND(-1,IBOT,-1,N)
01560 CALL SETIND(-1,-1,ITOP,IPRE)
01570 CALL SETIND(-1,IPRE,-1,ITOP)
01580 CALL SETIND(-1,-1,N,IBOT)
01590 RETURN
01600 END
01610 FUNCTION INLSTR(M,N)
01611 C INLSTR E' ANALOGA A INLSTL, EXCETO QUE O CORPO DA LISTA M
01612 C E' INSERIDO A DIREITA DA CELULA CUJO ENDEREÇO ESTA EM N.
01620 C
01630 L = LOCT(M)
01640 ITOP = LNKR(CONT(L))
01650 IBOT = LNKL(CONT(L))

```

```

01660      ILLSTR = L
01670      CALL SETIND(-1,L,L,L)
01680          ISUC = LNKR(CONT(N))
01690      CALL SETIND(-1,-1,ITOP,N)
01700      CALL SETIND(-1,IBOT,-1,ISUC)
01710      CALL SETIND(-1,N,-1,ITOP)
01720      CALL SETIND(-1,-1,ISUC,IBOT)
01730      RETURN
01740      END

01750          FUNCTION SUBST(D,N)
01751      C      O DADO D SUBSTITUI O DADO ARMAZENADO NA CELULA CUJO
01752      C      ENDEREÇO ESTA ESPECIFICADO POR N. O DADO ANTIGO E' O VALOR
01753      C      QUE RETORNA DA FUNCAO.
01760      C
01770          LBACK = LNKL(CONT(N))
01780          SUBST = DELETE(N)
01790          CALL NXTGT(D,LBACK)
01800      RETURN
01810      END

01820          FUNCTION SUBSTP(DAT,LST)
01821      C      O DADO DAT SUBSTITUI O DADO PRESENTEMENTE ARMAZENADO NO
01822      C      TOPO DA LISTA LST. O VALOR DESTA FUNCAO E' O DADO SUBSTITUIDO,
01823      C      I.E., E' O CONTEUDO PREVIO DO TOPO DA LISTA LST.
01830      C
01840          SUBSTP = SUBST(DAT,LNKR(CONT(LST)))
01850      RETURN
01860      END

01870          FUNCTION SUBSBT(DAT,LST)
01871      C      SUBSBT E' ANALOGA A SUBSTP, EXCETO QUE OPERA COM A CELULA
01872      C      DO FUNDO DA LISTA (BOTTOM).
01880      C
01890          SUBSBT = SUBST(DAT,LNKL(CONT(LST)))
01900      RETURN
01910      END

01920          FUNCTION TOP(P)
01921      C      O PARAMETRO P DE ENTRADA DA FUNCAO TOP E' O NOME DE UMA
01922      C      LISTA, SEU VALOR E' O DADO ARMAZENADO NA CELULA DO TOPO
01923      C      (CELULA MAIS A ESQUERDA) DA LISTA.
01924      C      A LISTA NAO E' MODIFICADA POR ESTA OPERACAO.
01930      C
01940          TOP = CONT(LNKR(CONT(LOCT(P))+1))
01950      RETURN
01960      END

01970          FUNCTION BOT(P)
01971      C      O PARAMETRO P DE ENTRADA DA FUNCAO BOT E' O NOME DE UMA
01972      C      LISTA, SEU VALOR E' O DADO ARMAZENADO NA CELULA DE FUNDO
01973      C      (CELULA MAIS A DIREITA) DA LISTA.
01974      C      A LISTA NAO E' MODIFICADA POR ESTA OPERACAO.
01980      C
01990          BOT = CONT(LNKL(CONT(LOCT(P))+1))
02000      RETURN
02010      END

02020          FUNCTION POPTOP(P)
02021      C      A CELULA DO TOPO DA LISTA CUJO NOME E' O PARAMETRO P DE ENTRADA,
02022      C      E' "POPPED-OFF" DA LISTA, I.E., E' DEVOLVIDA PARA LAVS,
02023      C      E O VALOR DA FUNCAO E' O VALOR CONTIDO NO ANTIGO TOPO.
02030      C
02040          POPTOP = DELETE(LNKR(CONT(LOCT(P))))
02050          RETURN
02060          END

```

```

02070          FUNCTION POPBOT(P)
02071 C      A CELULA DO FUNDO DA LISTA CUJO NOME E' O PARAMETRO P,
02072 C      E' "POPPED-OFF" DA LISTA, I.E., E' DEVOLVIDA PARA LAVS,
02073 C      E O VALOR DA FUNCAO E' O VALOR CONTIDO NO ANTIGO BOTTOM.
02080 C
02090      POPBOT = DELETE(LNKL(CONT(LOCT(P))))
02100      RETURN
02110      END
02120          FUNCTION ADVLL(LR,J,K)
02130 C
02140      CLR = CONT(LR)
02150 5      LK = LNKL(CONT(LNKL(CLR)))
02160      CAND = CONT(LK)
02170      CALL SETDIR(-1,LK,-1,CLR)
02180      IF(ID(CAND)-2)1,2,1
02190 1      IF(ID(CAND)-J)3,4,3
02200 3      IF(ID(CAND)-K)5,4,5
02210 4      ADVLL = 0.
02220      GO TO 6
02230 2      ADVLL = -1.0
02240 6      CALL STRIND(CLR,LR)
02250      RETURN
02260      END
02270          FUNCTION ADVLR(LR,J,K)
02280      CLR = CONT(LR)
02290 5      LK = LNKR(CONT(LNKL(CLR)))
02300      CAND = CONT(LK)
02310      CALL SETDIR(-1,LK,-1,CLR)
02320      IF(ID(CAND)-2)1,2,1
02330 1      IF(ID(CAND)-J)3,4,3
02340 3      IF(ID(CAND)-K)5,4,5
02350 4      ADVLR = 0.
02360      GO TO 6
02370 2      ADVLR = -1.0
02380 6      CALL STRIND(CLR,LR)
02390      RETURN
02400      END
02410          FUNCTION ADVSR(L,J,K)
02420      R = CONT(L)
02425      LCP = LNKL(R)
02430      CAND = CONT(LNKL(R))
02440      IF(ID(CAND)-1)1,6,1
02450 1      LCP = LNKR(CAND)
02460      CALL SETDIR(-1,LCP,-1,R)
02470      CAND = CONT(LCP)
02480      IF(ID(CAND)-2)3,4,3
02490 3      IF(ID(CAND)-J)7,8,7
02500 7      IF(ID(CAND)-K)5,8,5
02510 5      IF(ID(CAND)-1)1,6,1
02520 6      M = NUCELL(Z)
02530      CALL STRIND(R,M)
02540      CALL STRIND(CONT(L+1),M+1)
02550      CALL SETIND(-1,INHALT(LCP+1),LCNTR(L)+1,L+1)
02560      CALL SETDIR(-1,-1,N,R)
02570      CAND = CONT(INHALT(LNKL(R)+1))
02580      GOTO 1
02590 4      IF (LCNTR(L))9,10,9
02600 10     ADVSR = -1.0
02610      GOTO 12
02620 9      LK = LNKR(R)

```

```

02630          R = CONT(LK)
02640      CALL STRIND(CONT(LK+1),L+1)
02650      CAND = CONT(LNK(LK))
02660      CALL RCELL(LK)
02670      GOTO 1
02680      8      ADVSR = 0.
02690      12     CALL STRIND(R,L)
02700      RETURN
02710      END

          FUNCTION ADVSL(L,J,K)
02720          R = CONT(L)
02730          LCP = LNKL(R)
02735          CAND = CONT(LNKL(R))
02740          IF(ID(CAND)-1)1,6,1
02750          1      LCP = LNKL(CAND)
02760          CALL SETDIR(-1,LCP,-1,R)
02770          CAND = CONT(LCP)
02780          IF(ID(CAND)-2)3,4,3
02790          3      IF(ID(CAND)-J)7,8,7
02800          7      IF(ID(CAND)-K)5,8,5
02810          5      IF(ID(CAND)-1)1,6,1
02820          6      M=NUCELL(Z)
02830          CALL STRIND(R,M)
02840          CALL STRIND(CONT(L+1),M+1)
02850          CALL SETIND(-1,INHALT(LCP+1),LCNTR(L)+1,L+1)
02860          CALL SETDIR(-1,-1,M,R)
02870          CAND = CONT(INHALT(LNKL(R)+1))
02880          GO TO 1
02890          4      IF(LCNTR(L))9,10,9
02900          10     ADVSL = -1.0
02910          GO TO 12
02920          9      LK = LNKR(R)
02930          R = CONT(LK)
02940          CALL STRIND(CONT(LK+1),L+1)
02950          CAND = CONT(LNKL(R))
02960          CALL RCELL(LK)
02970          GO TO 1
02980          8      ADVSL = 0.
02990          12     CALL STRIND(R,L)
03000          RETURN
03010          END
03011
03012      C
03013      C
03014      C      FUNCOES DE AVANCO LINEAR:
03015      C      ADVLNR : ADVANCE LINEAR NAME RIGHT
03016      C      ADVLER : ADVANCE LINEAR ELEMENT RIGHT
03017      C      ADVLWR : ADVANCE LINEAR WORD RIGHT
03018      C      ADVLNL : ADVANCE LINEAR NAME LEFT
03019      C      ADVLEL : ADVANCE LINEAR ELEMENT LEFT
03020      C      ADVLWL : ADVANCE LINEAR WORD LEFT
03021      C
03022      C      FUNCOES DE AVANCO ESTRUTURAL:
03023      C      ADVSNR : ADVANCE STRUCTURAL NAME RIGHT
03024      C      ADVSER : ADVANCE STRUCTURAL ELEMENT RIGHT
03025      C      ADVSWR : ADVANCE STRUCTURAL WORD RIGHT
03026      C      ADVSNL : ADVANCE STRUCTURAL NAME LEFT
03027      C      ADVSEL : ADVANCE STRUCTURAL ELEMENT LEFT
03028      C      ADVSWL : ADVANCE STRUCTURAL WORD LEFT
03029      C
03030          FUNCTION ADVLNR(LR,A)

```

```

03040      A = ADVLR(LR,1,1)
03050      IF(A)1,2,1
03060      2      ADVLWR = REED(LR)
03070      1      RETURN
03080      END
                                FUNCTION ADVLER(LR,A)
03090      A = ADVLR(LR,0,0)
03100      IF(A)1,2,1
03110      2      ADVLEF = REED(LR)
03120      1      RETURN
03130      1      END
                                FUNCTION ADVLWR(LR,A)
03140      A = ADVLR(LR,1,0)
03150      IF(A)1,2,1
03160      2      ADVLWR = REED(LR)
03170      1      RETURN
03180      1      END
                                FUNCTION ADVSNR(LR,A)
03190      A = ADVSR(LR,1,1)
03200      IF(A)1,2,1
03210      2      ADVSNR = REED(LR)
03220      1      RETURN
03230      1      END
                                FUNCTION ADVSER(LR,A)
03240      A = ADVSR(LR,0,0)
03250      IF(A)1,2,1
03260      2      ADVSER = REED(LR)
03270      1      RETURN
03280      1      END
                                FUNCTION ADVSWR(LR,A)
03290      A = ADVSR(LR,1,0)
03300      IF(A)1,2,1
03310      2      ADVSWR = REED(LR)
03320      1      RETURN
03330      1      END
                                FUNCTION ADVLNL(LR,A)
03340      A = ADVLL(LR,1,1)
03350      IF(A)1,2,1
03360      2      ADVLNL = REED(LR)
03370      1      RETURN
03380      1      END
                                FUNCTION ADVLEL(LR,A)
03390      A = ADVLL(LR,0,0)
03400      IF(A)1,2,1
03410      2      ADVLEL = REED(LR)
03420      1      RETURN
03430      1      END
                                FUNCTION ADVLEL(LR,A)
03440      A = ADVLL(LR,0,0)
03450      IF(A)1,2,1
03460      2      ADVLEL = REED(LR)
03470      1      RETURN
03480      1      END
03490      1      END
03500      END

```

```

03510                                FUNCTION ADVLWL(LR,A)
03520      A = ADVLWL(LR,1,0)
03530      IF(A)1,2,1
03540  2  ADVLWL = REED(LR)
03550  1  RETURN
03560      END
03570                                FUNCTION ADVSNL(LR,A)
03580      A = ADVSNL(LR,1,1)
03590      IF(A) 1,2,1
03600  2  ADVSNL = REED(LR)
03610  1  RETURN
03620      END
03630                                FUNCTION ADVSEL(LR,A)
03640      A = ADVSEL(LR,0,0)
03650      IF(A)1,2,1
03660  2  ADVSEL = REED(LR)
03670  1  RETURN
03680      END
03690                                FUNCTION ADVSWL(LR,A)
03700      A = ADVSWL(LR,1,0)
03710      IF(A) 1,2,1
03720  2  ADVSWL = REED(LR)
03730  1  RETURN
03740      END
03750                                FUNCTION REED(K)
03760  C
03770      REED = CONT(LNKD(CONT(K))+1)
03780      RETURN
03790      END
03800                                FUNCTION DELETE(K)
03801  C      O PARAMETRO K DE ENTRADA E' O ENDERECO DE MAQUINA DE UMA
03802  C      CELULA QUE DEVE SER APAGADA DA LISTA, NA QUAL ELA ESTA', O DADO
03803  C      CONTIDO NA CELULA DEVOLVIDA PARA LAVS VOLTA COMO VALOR
03804  C      DA FUNCAO. SE (INADVERTIDAMENTE) O ENDERECO DE UM HEADER E'
03805  C      DADO COMO PARAMETRO DE ENTRADA, ENTAO O VALOR DA FUNCAO SERA'
03806  C      ZERO E UMA MENSAGEM E' IMPRESSA E O PROGRAMA CONTINUA A EXECUCAO.
03810  C
03820      IF(ID(CONT(K))-2)1,2,1
03830  2  PRINT 901
03840      DELETE = 0.
03850      RETURN
03860  901  FORMAT(IH1,'TENTATIVA DE APAGAR UMA HEADER - ',
03870      1'A FUNCAO VOLTA COM RESULTADO NULO E O PROGRAMA CONTINUA.')
03880  1  DELETE = CONT(K+1)
03890      LL = LNKL(CONT(K))
03900      LR = LNKR(CONT(K))
03910      CALL RCELL(K)
03920      CALL SETIND(-1,-1,LR,LL)
03930      CALL SETIND(-1,LL,-1,LR)
03940      RETURN
03950      END
03951  C
03955  C      M E C A N I S M O      D A      R E A D E R
03956  C
03960                                FUNCTION LRDRGV(P)
03961  C      LRDRGV PREPARA UMA READER PARA A LISTA CUJO ALIAS E' DADO
03962  C      COMO PARAMETRO DE ENTRADA. O VALOR QUE RETORNA DA FUNCAO
03963  C      E' O ENDERECO DA CELULA QUE FOI TIRADA DA LAVS, I.E., O
03964  C      ENDERECO DA CELULA QUE ATUALMENTE ESTA NO FORMATO DE UMA READER.
03965  C      INICIALMENTE, A READER CRIADA APONTA PARA A HEADER DA LISTA

```



```

03966 C PARA A QUAL FOI CRIADA,
03970 C
03980 LRDROV = NUCELL(Z)
03990 CALL SETIND(3,LOCT(P),0,LRDROV)
04000 CALL SETIND(0,P,0,LRDROV+1)
04010 RETURN
04020 END
04030 FUNCTION LOFRDR(K)
04040 C
04050 L = LNKL(CONT(K+1))
04060 CALL SETDIR(0,L,L,L)
04070 LOFRDF = L
04080 RETURN
04090 END
04100 FUNCTION LCNTR(K)
04101 C O PARAMETRO K DE ENTRADA E' O ENDERECO DE UMA READER,
04102 C O VALOR DA FUNCAO E' O CONTADOR DE NIVEL CONTIDO NO
04103 C CAMPO LCNTR DA READER.
04110 C
04120 LCNTR = LNKR(CONT(K+1))
04130 RETURN
04140 END
04150 FUNCTION LPNTR(K)
04151 C O PARAMETRO K DE ENTRADA E' O ENDERECO DE UMA READER,
04152 C O VALOR DA FUNCAO E' O ENDERECO DE MAQUINA ARMazenADO NO
04153 C CAMPO LPNTR DA READER, I.E., O ENDERECO DA CELULA PARA
04154 C A QUAL AQUELA READER ESTA APONTANDO ATUALMENTE.
04160 C
04170 LPNTR = LNKL(CONT(K))
04180 RETURN
04190 END
04200 FUNCTION LVLRVT(K)
04210 C
04220 LVLRVT = K
04230 1 IF(CONT(LVLRVT+1))2,3,2
04240 3 RETURN
04250 2 L = LNKR(CONT(LVLRVT))
04260 CALL STRIND(CONT(L),LVLRVT)
04270 CALL STRIND(CONT(L+1),LVLRVT+1)
04280 CALL RCELL(L)
04290 GO TO 1
04300 END
04310 FUNCTION LVLRV1(K)
04320 C
04330 LVLRV1 = K
04340 IF(CONT(LVLRV1+1))2,3,2
04350 3 RETURN
04360 2 L = LNKR(CONT(LVLRV1))
04370 CALL STRIND(CONT(L),LVLRV1)
04380 CALL STRIND(CONT(L+1),LVLRV1+1)
04390 CALL RCELL(L)
04400 RETURN
04410 END
04420 FUNCTION INITRD(K)
04430 C
04440 CALL SETIND(-1,LNKL(CONT(K+1)),-1,K)
04450 INITRD = K
04460 RETURN
04470 END
04480 FUNCTION IRARDR(K)

```

```

04490 C
04500     LRDRCP = LRDRCP(K)
04510     M = K
04520 3     N = LNKR(CONT(M))
04530     CALL RCELL(M)
04540     IF(N)1,2,1
04550 1     M = N
04560     GO TO 3
04570 2     RETURN
04580     END
04590     FUNCTION LRDRCP(K)
04600 C
04610     LRDRCP = NUCELL(Z)
04620     NEWR = LRDRCP
04630     NOW = K
04640 3     CALL STRIND(CONT(NOW),NEWR)
04650     CALL STRIND(CONT(NOW+1),NEWR+1)
04660     NOW = LNKR(CONT(NOW))
04670     IF(NOW)1,2,1
04680 1     NEW = NUCELL(Z)
04690     CALL SETIND(-1,-1,NEW,NEWR)
04700     NEWR = NEW
04710     GOTO 3
04720 2     RETURN
04730     END
04740     FUNCTION MADNTP(P,N)
04750 C
04760     L = LOCT(P)
04770     DO 1 I = 1,N
04780 1     L = LNKR(CONT(L))
04790     IF(ID(CONT(L))-2)2,3,2
04800 3     CALL SETDIP(0,L,L,L)
04810 2     MADNTP = L
04820     RETURN
04830     END
04840     FUNCTION MADNBT(P,N)
04850 C
04860     L = LOCT(P)
04870     DO 1 I=1,N
04880 1     L = LNKL(CONT(L))
04890     IF(ID(CONT(L))-2)2,3,2
04900 3     CALL SETDIR(0,L,L,L)
04910 2     MADNBT = L
04920     RETURN
04930     END
04940     FUNCTION MADLFT(K)
04950 C
04960     MADLFT = LNKL(CONT(K))
04970     IF(ID(CONT(MADLFT))-2)1,2,1
04980 2     CALL SETDIR(0,MADLFT,MADLFT,MADLFT)
04990 1     RETURN
05000     END
05010     FUNCTION MADRGT(K)
05020 C
05030     MADRGT = LNKR(CONT(K))
05040     IF(ID(CONT(MADRGT))-2)1,2,1
05050 2     CALL SETDIR(0,MADRGT,MADRGT,MADRGT)
05060 1     RETURN
05070     END
05080     FUNCTION NAMTST(K)

```

```

05090 C
05100 IF (LNKL(K)=LNKR(K))1,4,1
05110 4 IF (ID(CONT(K))-2)1,2,1
05120 2 IF (CONT(LNKR(CONT(LNKL(CONT(K)))))-CONT(K))1,3,1
05130 3 NAMIST = 0
05140 RETURN
05150 1 NAMIST = -1
05160 RETURN
05170 END
05180 FUNCTION LISTMT(P)
05190 C
05200 L = LOCT(P)
05210 IF (EQUAL(CONT(L),CONT(LNKR(CONT(L))))3,4,3
05220 4 LISTMT = 0
05230 RETURN
05240 3 LISTMT = -1
05250 RETURN
05260 END
05270 FUNCTION LSTEQL(LA,LB)
05280 C
05290 COMMON A,W(100)
05300 ASSIGN 100 TO LOCO
05310 LSTEQL = INTEGER(VISIT(LOCO,PARM12(LRDROV(LA),LRDROV(LB))))
05320 RETURN
05330 100 LRA = INTEGER(TOP(W(1)))
05340 LRB = INTEGER(TOP(W(2)))
05350 8 XA = ADVLWR(LRA,KA)
05360 XB = ADVLWR(LRB,KB)
05370 IF (KA)1,2,1
05380 1 IF (KB)3,4,3
05390 2 IF (KB)4,6,4
05400 6 IF (EQUAL(XA,XB))7,8,7
05410 7 IF (NAMTST(XA))4,9,4
05420 9 IF (NAMTST(XB))4,10,4
05430 10 LSTEQL = INTEGER(VISIT(LOCO,PARM12(LRDROV(XA),LRDROV(XB))))
05440 IF (LSTEQL)4,100,4
05450 3 CALL RCELL(LRA)
05460 CALL RCELL(LRB)
05470 CALL TERM(0,RESTOR(2))
05480 4 CALL RCELL(LRA)
05490 CALL RCELL(LRB)
05500 CALL TERM(-1,RESTOR(2))
05510 END
05520 FUNCTION LSSCPY(LA)
05530 C
05540 COMMON A,W(100)
05550 ASSIGN 100 TO LOCO
05560 LSSCPY = INTEGER(VISIT(LOCO,PARM12(LRDROV(LA),LIST(9))))
05570 RETURN
05580 100 LC = INTEGER(TOP(W(2)))
05590 LR = INTEGER(TOP(W(1)))
05600 5 X = ADVLWR(LR,K)
05610 IF (K)1,2,1
05620 1 CALL RCELL(LR)
05630 CALL TERM(LC,RESTOR(2))
05640 2 IF (NAMTST(X))3,4,3
05650 3 CALL NEWBOT(X,LC)
05660 GOTO 5
05670 4 CALL NEWBOT(VISIT(LOCO,PARM12(LRDROV(X),LIST(9))),TOP(W(2)))
05680 GOTO 100

```

```

05690      END
05700      FUNCTION LSTPRO(L,K)
05710  C
05720      NEXT = K
05730  3      IF(LNK(L,NEXT+1)-LNKR(L))1,2,1
05740  1      NEXT = LNKR(CONT(NEXT))
05750      IF(NEXT)3,4,3
05760  2      LSTPRO = 0
05770      RETURN
05780  4      LSTPRO = -1
05790      RETURN
05800      END
05810      FUNCTION LPURGE(LST)
05820  C
05830      K = LPDROV(LST)
05840      LPURGE = 0
05850  3      X = ADVSWR(K,J)
05860  6          IF(J)2,1,2
05870  1      IF(NAMTST(X))3,4,3
05880  4      IF(LSTPRO(X,K))3,5,3
05890  5          L = LPNTR(K)
05900          X = ADVLWR(K,J)
05910          CALL DELETE(K)
05920          LPURGE = LPURGE+1
05930      GO TO 6
05940  2      CALL IRAPDR(K)
05950      RETURN
05960      END
05970      SUBROUTINE PRESRV(N)
05980  C
05990      COMMON AVSL,W(100)
06000      DO 1 I=1,N
06010  1      CALL NEWTOP(TOP(W(I)),W(I))
06020      RETURN
06030      END
06040      SUBROUTINE RESTOR(N)
06050  C
06060      COMMON AVSL,W(100)
06070      DO 1 I=1,N
06080  1      CALL POPTOP(W(I))
06090      RETURN
06100      END
06110      FUNCTION LOCT(K)
06120  C
06130      IF(NAMTST(K))1,2,1
06140  2      LOCT = K
06150      RETURN
06160  1      PRINT 901
06165      CALL IMPCELL(MADOV(K))
06170      STOP
06180  901      FORMAT('10 ARGUMENTO DE LOCT DEVERIA SER UMA CABECA DE ',
06190      '1'LISTA = 0 PROGRAMA TERMINA.')
06200      END
06210      FUNCTION PARMT2(A,B)
06220  C
06230      COMMON X,W(100)
06240      CALL NEWTOP(A,W(1))
06250      CALL NEWTOP(B,W(2))
06260      PARMT2 = A
06270      RETURN

```



```

06290                                FUNCTION RDLSTA(Z)
06300 C
06310 DIMENSION CRDBUF(16)
06320 COMMON /XIRCT/ CP(5),BLANK,NULL,ZERO,LP,RP
06325 DATA MXWRD,MXCRD/5,16/
06330 80 FORMAT(16A5)
06340 ASSIGN 13 TO START
06350 ASSIGN 20 TO NEWLST
06360 CALL LIST(STACK)
06370 IS = 1
06380 WORD = BLANK
06390 KOUNT = 0
06400 PLACE = BLANK
06410 12 READ 80, CRDBUF
06420 PRINT 80, (CRDBUF(J),J=1,MXCRD)
06430 IW = 1
06440 10 IC = 1
06450 9 SYMBOL = SQIN(CP(1),SQOUT(CP(IC),CRDBUF(IW)),PLACE)
06460 IF(EQUAL(SYMBOL,BLANK))1,2,1
06470 1 IF(EQUAL(SYMBOL,LP ))3,4,3
06480 3 IF(EQUAL(SYMBOL,RP ))5,6,5
06490 13 IF(IC=MXWRD) 7,8,7
06500 7 IC = IC+1
06510 GOTO 9
06520 8 IF(IW=MXCRD)11,12,11
06530 11 IW = IW + 1
06540 GOTO 10
06550 4 IF(KOUNT)40,44,40
06560 44 CALL NXTTRGT(LIST(NEW),STACK)
06570 KOUNT = 1
06580 CALL VISIT(START)
06585 C FIM DA LISTA PRINCIPAL
06590 RDLSTA = POPTOP(STACK)
06600 CALL MTLIST(STACK)
06610 CALL RCELL(STACK)
06620 RETURN
06630 40 IF(EQUAL(WORD,BLANK))41,42,41
06640 41 CALL NXTLEFT(LANORM(WORD),TOP(STACK))
06650 WORD = BLANK
06660 IS = 1
06665 C SUBLISTA
06670 42 CALL VISIT(NEWLST)
06675 C FIM DE SUBLISTA
06680 CALL POPTOP(STACK)
06690 GOTO 13
06700 C NEWLST
06710 20 CALL NXTLEFT(LIST(NEW),TOP(STACK))
06720 CALL NXTTRGT(NEW,STACK)
06730 GOTO 13
06740 C NON-BLANK SYMBOL NOT LP OR RP
06750 5 CALL SHIN(7,SYMBOL,WORD)
06760 IF(IS = MXWRD)51,52,51
06770 51 IS = IS+1
06780 GOTO 13
06790 52 IS = 1
06800 21 CALL NXTLEFT(LANORM(WORD),TOP(STACK))
06810 WORD = BLANK
06820 IS = 1
06830 GOTO 13
06840 2 IF(EQUAL(WORD,BLANK))21,13,21

```

```

06850 C RIGHT PARENTHESIS
06860 6 IF(EQUAL(WORD,BLANK))61,62,61
06870 61 CALL NXLEFT(LANORM(WORD),TOP(STACK))
06880 WORD = BLANK
06890 IS = 1
06900 62 CALL TERM(Z)
06910 END
06920 SUBROUTINE PRLSTS(OUTLST,I)
06930 C
06940 EQUIVALENCE (KOUT,OUT)
06950 900 FORMAT(1H1,20X,'INICIO DE LISTA')
06960 901 FORMAT( 21X,'FIM DE LISTA')
06970 902 FORMAT(21X,I15)
06980 903 FORMAT(24X,'INICIO DE SUBLISTA')
06990 904 FORMAT(24X,'FIM DE SUBLISTA')
07000 905 FORMAT(27X,A5)
07010 906 FORMAT(21X,F15,7)
07020 907 FORMAT(24X,'SUBLISTA VAZIA')
07030 PRINT 900
07040 LR = LRDRGV(OUTLST)
07050 LEVEL = 0
07060 7 X = ADVSWR(LR,K)
07070 IF(K)1,2,1
07080 2 IF(LEVEL = LCNTR(LR))21,22,23
07090 22 IF(NAMTST(X))3,4,3
07100 4 IF(LISTMT(X))5,6,5
07110 6 PRINT 907
07120 GOTO 7
07130 5 PRINT 903
07140 LEVEL = LEVEL + 1
07150 GOTO 7
07160 3 GOTO(12,11,13),I
07170 11 OUT = X
07180 PRINT 902,KOUT
07190 GOTO 7
07200 12 OUT = X
07210 PRINT 905,KOUT
07220 GOTO 7
07230 13 PRINT 906,X
07240 GOTO 7
07250 23 PRINT 904
07260 LEVEL = LEVEL -1
07270 GOTO 2
07280 1 IF(LEVEL = LCNTR(LR))21,32,33
07290 33 PRINT 904
07300 LEVEL = LEVEL -1
07310 GOTO 1
07320 32 PRINT 901
07330 CALL RCELL(LR)
07340 21 RETURN
07350 END
07360 FUNCTION SEQLL(Z,N)
07370 L = LNKL(Z)
07380 Z = CONT(L)
07390 SEQLL = CONT(L+1)
07400 N = ID(Z)-1
07410 RETURN
07420 END
07430 FUNCTION SEQLR(Z,N)
07440 L = LNKR(Z)

```

```

07450      Z = CONT(L)
07460      SEQDR = CONT(L+1)
07470      N = ID(Z)-1
07480      RETURN
07490      END
07500      FUNCTION SEQSR(Z,N)
07510      IF(ID(Z)-1)4,5,4
07520      L = LNKR(CONT(CONT(LNKL(CONT(LNKR(Z))))+1))
07530      5      GOTO 3
07540      4      L = LNKR(Z)
07550      3      IF(ID(CONT(L))-1)1,2,1
07560      1      SEQSR = CONT(L+1)
07570      Z = CONT(L)
07580      N = ID(Z) - 1
07590      RETURN
07600      2      L = LNKR(CONT(CONT(L+1)))
07610      GOTO 3
07620      END
07630      FUNCTION SEQSL(Z,N)
07640      IF(ID(Z)-1)4,5,4
07650      L = LNKL(CONT(CONT(LNKL(CONT(LNKR(Z))))+1))
07660      5      GOTO 3
07670      4      L = LNKL(Z)
07680      3      IF(ID(CONT(L))-1)1,2,1
07690      1      SEQSL = CONT(L+1)
07700      Z = CONT(L)
07710      N = ID(Z) - 1
07720      RETURN
07730      2      L = LNKL(CONT(CONT(L+1)))
07740      GOTO 3
07750      END
07760      FUNCTION SEQRDR(LST)
07770      SEQRDR = CONT(LOC(LST))
07780      RETURN
07790      END
07800      FUNCTION NOATVL(AT,LST)
07810      M = MADATR(AT,LST)
07820      IF(M+1)2,1,2
07830      2      NOATVL = INTEGER(DELETE(LNKR(CONT(M))))
07840      CALL DELETE(M)
07850      RETURN
07860      1      NOATVL = 0
07870      RETURN
07880      END
07890      FUNCTION NEWVAL(AT,VAL,LST)
07900      M = MADATR(AT,LST)
07910      IF(M+1)2,1,2
07920      2      NEWVAL = INIGER(SUBST(VAL,LNKR(CONT(M))))
07930      RETURN
07940      1      CALL LOATVL(AT,VAL,LST)
07950      NEWVAL = 0
07960      RETURN
07970      END
07980      FUNCTION ITSVAL(AT,LST)
07990      IF(LNKL(CONT(LST+1)))3,4,3
08000      3      M = MADATR(AT,LST)
08010      IF(M+1)1,2,1
08020      1      ITSVAL = INHALI(LNKR(CONT(M))+1)
08030      RETURN
08040      4      CALL DERROR(LST)

```



```

08050 2      LTVSVAL = 0
08060      RETURN
08070      END
08080      FUNCTION MIDLST(LST)
08090          MIDLST = LST
08100          K = LNKL(CONT(LOCT(LST)+1))
08110          IF(K)1,2,1
08120 1      CALL SETDIR(0,K,K,K)
08130      CALL MLIST(K)
08140 2      RETURN
08150      END
08160      FUNCTION MAKEDL(L,M)
08170      CALL MIDLST(M)
08180      MAKEDL = M
08190      N = LOCT(M)
08200      K = LOCT(L)
08210      CALL SETIND(-1,K,-1,N+1)
08220      CALL SETIND(-1,-1,LCNTR(L)+1,K+1)
08230      RETURN
08240      END
08250      FUNCTION NAMEDL(L)
08260      NAMEDL = LNKL(LOCT(L)+1)
08270      RETURN
08280      END
08290      FUNCTION LDATVL(AT,VL,LST)
08300          IF(LNKL(CONT(LST+1)))1,2,1
08310 2      LDATVL = LISTAV(LST)
08320 1      CALL NXTRGT(VL,NXTLEFT(AT,LNKL(CONT(LST+1))))
08330      RETURN
08340      END
08350      FUNCTION LISTAV(LST)
08360      LISTAV = LIST(0)
08370      CALL SETIND(-1,LNKR(LISTAV),-1,LST+1)
08380      RETURN
08390      END
08400      FUNCTION MADATR(AT,LST)
08410      LSTDES = LNKL(CONT(LST+1))
08420      IF(LSTDES)1,4,1
08430 1      MADATR = LNKR(CONT(LSTDES))
08440 8      IF(ID(CONT(MADATR))-2)3,4,3
08450 3      IF(EQUAL(CONT(MADATR+1),AT))5,6,5
08460 5      M = LNKR(CONT(MADATR))
08470      IF(ID(CONT(M))-2)7,4,7
08480 7      MADATR = LNKR(CONT(M))
08490      GOTO 8
08500 4      MADATR = -1
08510 6      RETURN
08520      END
08530      SUBROUTINE DERROR(LST)
08540      PRINT 900,LST
08550      PRINT 901
08560      RETURN
08570 900  FORMAT(1H1,20X,I5)
08580 901  FORMAT(20X'LISTA DE ATRIBUTOS EXIGIDA MAS NAO FOI ACHADA')
08590      END
08600      FUNCTION MRKLST(M,LST)
08610      MRKLST = LST
08620      CALL SETIND(M,-1,-1,LOCT(LST)+1)
08630      RETURN
08640      END

```

```

08650                                FUNCTION MKKISS(M,LST)
08660    MKKISS = LST
08670    LR = LRDROV(MPKLST(M,LST))
08680    3    X = ADVSNR(LR,K)
08690        IF(K)1,2,1
08700    2    CALL SETIND(N,-1,-1,LNKR(X)+1)
08710        GOTO 3
08720    1    CALL PCELL(LR)
08730        RETURN
08740        END
08750                                FUNCTION LSTMRK(L)
08760    C
08770        LSTMRK = ID (CONF(LOCT(L)+1))
08780        RETURN
08790        END

```

```

FUNCTION SQOUT(FIELD,SOURCE)
DIMENSION A1(5)
IFIELD=FIELD
GOTO(2,2,2,2,2),IFIELD
SQOUT =0.0
RETURN
2 DECODE(5,1,SOURCE),A1
1 FORMAT(5A1)
SQOUT = A1(IFIELD)
RETURN
END

```

```

FUNCTION SQIN(FIELD,DATUM,DEST)
DIMENSION A1(5)
IFIELD=FIELD
GOTO(2,2,2,2,2),IFIELD
SQIN=0.0
RETURN
2 DECODE(5,1,DEST),A1
A1(IFIELD)=DATUM
ENCODE(5,1,DEST),A1
1 FORMAT(5A1)
SQIN = DEST
RETURN
END

```

```

FUNCTION SHIN(N,DATUM,DEST)
DIMENSION A1(5)
DECODE(5,1,DEST),A1
1 FORMAT(5A1)
DO 100 I=1,4
100 A1(I) = A1(I+1)
A1(5)=DATUM
ENCODE(5,1,DEST),A1
SHIN = DEST
RETURN
END

```

```

FUNCTION LANORM(WORD)
DIMENSION A1(5),BL(5)
DO 100 I=1,5
100 BL(I)=' '

DECODE(5,1,WORD),A1
1 FORMAT(5A1)

DO 200 I=1,5
IF(A1(I).NE.' ') GO TO 201
200 CONTINUE
LANORM=0
RETURN

201 K=1
DO 300 J=I,5
BL(K)=A1(J)
K=K+1
300 CONTINUE

ENCODE(5,1,1),BL
LANORM=1

```

RETURN

END

BLOCKDATA

COMMON /XTRCT/ CP(5),BLANK,NULL,ZERO,LP,RP

COMMON AVSL,W(100)

DATA CP/1.,2.,3.,4.,5./

DATA BLANK,NULL,ZERO,LP,RP/' ',0,'0','(1,1)'/

DATA AVSL,W/101*0.0/

END

SUBROUTINE IMPCELL(N)

IN=INHALT(N)

IID=ID(IN)

ILNKL=LNKL(IN)

IID2=ID2(IN)

ILNKR=LNKR(IN)

JN=INHALT(N+1)

XN=CONT(N+1)

JID=ID(JN)

JLNKL=LNKL(JN)

JID2=ID2(JN)

JLNKR=LNKR(JN)

IF(IID.GT.0) GO TO 24

IID2=0 NORMAL

IID2=1 ALFA

IID2=2 INTEIRO

IID2=3 REAL

IF(IID2.GT.0) GO TO 231

PRINT 22, N, IID, ILNKL, IID2, ILNKR, I=1, 4)

PRINT 23, JID, JLNKL, JID2, JLNKR, JN, XN, JN

FORMAT(///, 4(I8, I3, I7, I3, I7))

FORMAT(/8X, I3, I7, I3, I7, I28, F28.7, 20X, A5)

RETURN

IF(IID2.GE.1) GO TO 233

PRINT 22, N, IID, ILNKL, IID2, ILNKR

PRINT 232, JN

FORMAT(/20X, A5)

RETURN

IF(IID2.EQ.3) GO TO 235

PRINT 22, N, IID, ILNKL, IID2, ILNKR

PRINT 234, JN

FORMAT(/I28)

RETURN

PRINT 22, N, IID, ILNKL, IID2, ILNKR

PRINT 236, XN

FORMAT(/F28.7)

RETURN

PRINT 22, N, IID, ILNKL, IID2, ILNKR

PRINT 23, JID, JLNKL, JID2, JLNKR

RETURN

END

SUBROUTINE IMPDIR(N)

IN=N

IID=ID(IN)

ILNKL=LNKL(IN)

IID2=ID2(IN)

ILNKR=LNKR(IN)

XN=CONT(N)

PRINT 22, IID, ILNKL, IID2, ILNKR, N, XN, N

FORMAT(///I8, I7, I3, I7, I17, F17.7, 5X, A5)

RETURN

END

SUBROUTINE IMPLIST(ALIAS)

COMMON AVSL

M=INHALT(MADOV(ALIAS))

I=ID(M)

```
J=LKRL(M)
K=LO2(M)
L=LKKR(M)
PRINT 24, I, J, K, L
24  FORMAT(///8X, I3, I7, I3, I7)
    IF (ALIAS.EQ.AVSL) GO TO 777
    M=LOCT(ALIAS)
777  CALL IMPCELL(L)
    LR=LKKR(CONT(L))
10   IF ( LR.EQ.0 )RETURN
    IF ( LR.EQ.L.AND.ID(CONT(L)).EQ.2 ) RETURN
    CALL IMPCELL(LR)
    LR=LKKR(CONT(LR))
    GO TO 10
END
```

```
SUBROUTINE OUTLIST(OUTLST)
DIMENSION SAIDA(240)
DATA MAX,AP,FP,SAIDA/240,'(,)',240*' /,VIRGU/!,!/
DATA VAZIO/(' )'/
```

```

LAST=1
IF (NAMTST(OUTLST))30,40,30
40 IF (LISTMT(OUTLST))50,60,50
C      LISTA VAZIA
60 SAIDA(LAST)=VAZIO
GO TO 240
50 SAIDA(LAST)=AP
LR = LRDPOV(OUTLST)
LEVEL = 0
7 X = ADVSWR(LR,K)
IF(K)1,2,1
2 IF (LEVEL - LCNTR(LR))24,22,23
22 IF (NAMTST(X))3,4,3
4 IF (LISTMT(X))5,6,5
C      SUBLISTA VAZIA
6 IF (LAST+2.GE.MAX) GO TO 24
LAST=LAST+1
SAIDA(LAST)=VAZIO
LAST=LAST+1
SAIDA(LAST)=VIRGU
GOTO 7
C      INICIO DE SUBLISTA
5 LAST=LAST+1
IF (LAST.GE.MAX) GOTO 24
SAIDA(LAST)=AP
LEVEL = LEVEL + 1
GOTO 7
C      ALFA, INT, REAL
3 I = ID2(CONT(LPNTR(LR)))
IF (I.EQ.0) I=1
GOTO(12,11,13),I
C      ELEMENTO INTEIRO
11 IF (ABS(X).GE.10000.) GO TO 111
IF (LAST+2.GE.MAX) GO TO 24
ENCODE(1,2410,SAIDA(LAST)),X
2410 FORMAT(I5)
LAST=LAST+2
SAIDA(LAST)=VIRGU
GOTO 7
111 IF (LAST+4.GE.MAX) GO TO 24
ENCODE(3,241,SAIDA(LAST+1)),X
241 FORMAT(J15)
LAST=LAST+4
SAIDA(LAST)=VIRGU
GOTO 7
C      ALFANUMERICO
12 IF (LAST+2.GE.MAX) GO TO 24
LAST=LAST+1
SAIDA(LAST)=X
LAST=LAST+1
SAIDA(LAST)=VIRGU
GOTO 7
C      REAL
13 IF (LAST+4.GE.MAX) GO TO 24
ENCODE(242,3,SAIDA(LAST+1)),X
```

```

242  FORMAT(F15.7)
      LAST=LAST+4
      SAIDA(LAST)=VIRGU
      GOTO 7

C          FIM DE SUBLISTA
23  IF(LAST+1.GE.MAX) GO TO 24
      SAIDA(LAST)=FP
      LAST=LAST+1
      SAIDA(LAST)=VIRGU
      LEVEL = LEVEL -1
      GOTO 2
1    IF(LEVEL = LCNTR(LR))24,32,33
C          FIM DE SUBLISTA
33  IF(LAST+1.GE.MAX) GO TO 24
      SAIDA(LAST)=FP
      LAST=LAST+1
      SAIDA(LAST)=VIRGU
      LEVEL = LEVEL -1
      GOTO 1

C          FIM DE LISTA
32  SAIDA(LAST)=FP
24  CALL RCELL(LR)
240  PRINT 243,(SAIDA(K),K=1, LAST)
243  FORMAT(/(/1X,24A5))
      RETURN
30  PRINT 900,OUTLST
900  FORMAT(I,' NAO E' LISTA, O PROGRAMA CONTINUA')
      RETURN
      END

```


VIII - TABELA DE REFERENCIAS DE TODOS OS SIMBOLOS GLOBAIS:

SIMBOLO	REFERENCIADO EM:
ADVLEL	
ADVLER	
ADVLL	ADVLLN, ADVLEL, ADVLWL
ADVLN	
ADVLR	ADVLRN, ADVLER, ADVLWR
ADVLWL	
ADVLP	LSTEQ, LSSCPY, LPURGE
ADVSEL	
ADVSR	
ADVSL	ADVSNL, ADVSEL, ADVSWL
ADVSNL	
ADVSNR	MRKLSS
ADVSR	ADVSNR, ADVSER, ADVSWR
ADVSWL	
ADVSWR	LPURGE, PRLSTS
ALPHI.	RDLSTA, PRLSTS, SQOUT, SQIN, SHIN, LANORM, IMPCEL
ALPHO.	RDLSTA, PRLSTS, SQOUT, SQIN, SHIN, LANORM, IMPCEL
AND	
BOT	
CONT	NUCELL, MTLIST, IRALST, NULSTL, NULSTR, NXTLFT, NXTRGT, INLSTL, INLSTR, SUBST, SUBSTP, SUBSBT, TOP, BOT, POPTOP, POPBOT, ADVLL, ADVLR, ADVSR, ADVSL, REED, DELETE, LOFRDR, LCNTR, LPNTR, LVLVTV, LVLVVI, INITRD, IRARDR, BRDRCP, MADNTP, MADNBI, MADLFT, MADRGT, NAMTST, LISTMT, SEQLL, SEQLR, SEQSR, SEQSL, SEQRDR, NOATVL, NEWVAL, ITSVAL, MIDLST, LDATVL, MADATR, IMPCEL, LSTMRK
DAT.	
DELETE	SUBST, POPTOP, POPBOT, LPURGE, NOATVL
DERROR	ITSVAL
EQUAL	LISTMT, LSTEQ, RDLSTA, MADATP
EXIT	NUCELL, LOCT
EXP1,2	SHIN
FLIRT.	PRLSTS, IMPCEL
FLOUT.	PRLSTS, IMPCEL
ID	NUCELL, NULSTL, NULSTR, ADVLL, ADVLR, ADVSR, ADVSL, DELETE, MADNTP, MADNBT, MADLFT, MADRGT, NAMTST, SEQLL, SEQLR, SEQSR, SEQSL, MADATR, IMPCEL, LSTMRK
ID2	IMPCEL
IMPCEL	
INHAI	ADVSR, ADVSL, ITSVAL, IMPCEL
INITAS	
INITRD	
INLSTL	
INLSTR	
INTGER	LSTEQ, LSSCPY, NOATVL, NEWVAL
INTI.	PRLSTS, DERROR, IMPCEL
INTO.	PRLSTS, DERROR, IMPCEL
IRALST	NUCELL
IRARDR	LPURGE
ITSVAL	
LANORM	RDLSTA

SINBOLO	REFERENCIADO EM:
LCNTR	IRALST, NXTLFT, NXTTRGT, ADVSR, ADVSL, IRARDR, PRLSTS, MAKEDL
LDATVL	NEWVAL
LIST	NULSTL, NULSTR, LSSCPY, RDLSTA, LISTAV
LISTAV	LDATVL
LISTMT	MTLIST, PRLSTS
LNKL	RCELL, MTLIST, IRALST, NULSTR, NXTLFT, INLSTL, INLSTR, SUBST, SUBSBT, BOT, POPBOT, ADVLL, ADVLR, ADVSR, ADVSL, REED, DELETE, LOFRDR, LPNTR, INITRD, MADNBT, MADLFT, NAMTST, LSTPRO, SEQLL, SEQSR, SEQSL, ITSVAL, MTDLST, NAMEDL, LDATVL, MADATR, IMPCEL
LNKR	NUCELL, MTLIST, NULSTL, NXTTRGT, INLSTL, INLSTR, SUBSTP, TOP, POPTOP, ADVLR, ADVSR, ADVSL, DELETE, LCNTR, LVLVRT, LVLRV1, IRARDR, LRDRCP, MADNTP, MADRGT, NAMTST, LISMT, LSTPRO, SEQLR, SEQSR, SEQSL, NOATVL, NEWVAL, ITSVAL, LISTAV, MADATR, MRKLSS, IMPCEL
LOCT	MTLIST, IRALST, NEWTOP, NEWBOT, INLSTL, INLSTR, TOP, BOT, POPTOP, POPBOT, LRDRCV, MADNTP, MADNBT, LISTMT, SEQRDR, MTDLST, MAKEDL, NAMEDL, MRKLST, LSTMRK
LOFRDR	
LPNTR	LPURGE
LPURGE	
LRDRCP	
LRDRCV	LSTEQL, LSSCPY, LPURGE, PRLSTS, MRKLSS
LSSCPY	
LSTEQL	
LSTPRO	LPURGE
LVLVRT	
LVLRV1	
LVLRV1	
MADATR	NOATVL, NEWVAL, ITSVAL
MADLFT	
MADNET	
MADNTP	
MADOV	INITAS
MADRGT	
MAKEDL	
MRKLSS	MRKLSS
MRKLST	MAKEDL
MTDLST	IRALST, RDLSTA, MTDLST
MTLIST	
NAMEDL	
NAMTST	NXTLFT, NXTTRGT, LSTEQL, LSSCPY, LPURGE, LOCT, PRLSTS
NEWBOT	LSSCPY
NEWTOP	PRIMIT, PRESRV, PARMT2
NEWVAL	
NOATVL	
NOT	
NUCELL	LIST, IRALST, NXTLFT, NXTTRGT, ADVSR, ADVSL, LRDRCV, LRDRCP
NULSTL	
NULSTR	
NXTLFT	NEWBOT, RDLSTA, LDATVL
NXTTRGT	NEWTOP, SUBST, RDLSTA, LDATVL
OR	
PARMT2	LSTEQL, LSSCPY
POPBOT	

SIMBOLO	REFERENCIADO EM:
POPTOP	PRIMIT, RESTOR, RDLSTA
PRESRV	
PRLSTS	
RCELL	IRALST, ADVSR, ADVSL, DELETE, LVLRVT, LVLRV1, IRARDR, LSTEQL, LSSCPY, RDLSTA, PRLSTS, MRKLSS
RDLSTA	
REED	ADVLR, ADVLER, ADVLWR, ADVSNR, ADVSER, ADVSWR, ADVLNL, ADVDEL, ADVLWL, ADVSNL, ADVSEL, ADVSWL
RESTOR	LSTEQL, LSSCPY
SEQLL	
SEQLR	
SEQRDR	
SEQSL	
SEQSR	
SETDIR	INITAS, NUCELL, RCELL, LIST, MTLIST, ADVLL, ADVLR, ADVSR, ADVSL, LQFRDR, MADHTP, MADNBT, MADLFT, MADRGT, MTDLST
SETID2	
SETIND	RCELL, LIST, MTLIST, IRALST, NULSTL, NULSTR, NXTLFT, NXTRGT, INLSTL, INLSTR, ADVSR, ADVSL, DELETE, LRDRV, INITRD, LRDRCP, MAKEDL, LISTAV, MRKLST, MRKLSS
SHIN	RDLSTA
SQIN	RDLSTA
SQOUT	RDLSTA
SIRDIR	INITAS
STRING	NUCELL, NXTLFT, NXTRGT, ADVLL, ADVLR, ADVSR, ADVSL, LVLRVT, LVLRV1, LRDRCP
SUBSBT	
SUBST	SUBSTP, SUBSBT, NEWVAL
SUBSTP	
TERM	LSTEQL, LSSCPY, RDLSTA
TOP	LSTEQL, LSSCPY, PRESRV, RDLSTA
VADDR,	SQOUT, SQIN, SHIN, LANORM
VISIT	LSTEQL, LSSCPY, RDLSTA
XIRCT	DAT.
.COMM,	NUCELL, RCELL, MTLIST, LSTEQL, LSSCPY, PRESRV, RESTOR, PARM12, DAT.
.JBFF	DAT.

OBS.: ESTA TABELA FOI OBTIDA COM A UTILIZACAO DO PROGRAMA UTILITARIO "GLOB" DO SISTEMA DEC-10 E FINALMENTE EDITADO PELO EDITOR-CORRETOR "TECO".

IX - INDICE DAS ROTINAS EM ORDEM ALFABETICA,
TIPO DE ROTINA (FUNCAO/SUBROTINA)
NUMERO DE PARAMETROS,
NUMERO DAS LINHAS DE INICIO E FIM DA LISTAGEM-FONTE.

ROTINA	F/S/H(*)#PARAM.	LISTAGEM
ADVLEL	F 2	3450-3500
ADVLER	F 2	3090-3140
ADVLL	F 3	2120-2260
ADVLNL	F 2	3390-3440
ADVLNR	F 2	3030-3080
ADVLR	F 3	2270-2400
ADVLWL	F 2	3510-3560
ADVLWR	F 2	3150-3200
ADVSEL	F 2	3630-3680
ADVSEK	F 2	3270-3320
ADVSL	F 3	2720-3011
ADVSML	F 2	3570-3620
ADVSME	F 2	3210-3260
ADVSR	F 3	2410-2710
ADVSWL	F 2	3690-3740
ADVSWR	F 2	3330-3380
AND		
BOT	F 1	1970-2010
CONT		
DELETE	F 1	3800-3950
DERROR	S 1	8530-8590
EQUAL		
ID		
ID2		
IMPCEL		
INHALT		
INITAS	S 2	0010-0180
INITRD	F 1	4420-4470
INLSTL	F 2	1470-1600
INLSTR	F 2	1610-1740
IRALST	F 1	0690-0780
IRARDR	F 1	4480-4580
ITSVAL	F 2	7980-8070
LANORM		
LCNTR	F 1	4100-4140
LDATVL	F 3	8290-8340
LIST	F 1	0440-0540
LISTAV	F 1	8350-8390
LISTMT	F 1	5180-5260
LNKL		
LNKR		
LOCT	F 1	6110-6200
LOFRDR	F 1	4030-4090
LPNTR	F 1	4150-4190
LPURGE	F 1	5810-5960
LRDRCP	F 1	4590-4730
LRDROV	F 1	3960-4020
LSSCPY	F 1	5520-5690
LSTEQL	F 2	5270-5510

ROUTINA	F/S/M(*)#PARAM.	LISTAGEM
LSTMRR	F 1	8750-8790
LSTPRO	F 2	5700-5800
LVLRV1	F 1	4310-4410
LVLRV1	F 1	4200-4300
MADATR	F 2	8400-8520
MADLFT	F 1	4940-5000
MADNBT	F 2	4840-4930
MADNTP	F 2	4740-4830
MADOV		
MADRGT	F 1	5010-5070
MAKEDL	F 2	8160-8240
MRKLSS	F 2	8650-8740
MRKLST	F 2	8600-8640
ITDLST	F 1	8080-8150
ITLIST	F 1	0550-0680
IAMEDL	F 1	8250-8280
IAMTST	F 1	5080-5170
NEWBOT	F 2	1140-1180
NEWTOP	F 2	1090-1130
NEWVAL	F 3	7890-7970
NOATVL	F 2	7800-7880
NOT		
NUCELL	F 1 (DUMMY)	190-0350
NUSTL	F 2	0790-0930
NUSTR	F 2	0940-1080
NXTLFT	F 2	1190-1320
NXTRGT	F 2	1330-1460
CR		
FARMT2	F 2	6210-6280
FOPBOT	F 1	2070-2110
FOPTOP	F 1	2020-2060
FRESRV	S 1	5970-6030
FRLSTS	S 2	6930-7350
FCELL	S 1	0360-0430
FDLSTA	F 1	6290-6910
FEEED	F 1	3750-3790
FESTOR	S 1	6040-6100
FEQLL	F 2	7360-7420
FEQLR	F 2	7430-7490
SEQRDR	F 1	7760-7790
SEQSL	F 2	7630-7750
SEQSR	F 2	7500-7620
SETDIR		
SETID2		
SETIND		
SHIN		
SQIN		
SQOUT		
STRDIR		
STRIND		
SUBSBI	F 2	1870-1910
SUBST	F 2	1750-1810
SUBSTP	F 2	1820-1860
TERM		
TOP	F 1	1920-1960
VISIT		

* OBS: F FUNCAO EM FORTRAN
S SUBROTINA EM FORTRAN
M SUBPROGRAMA EM MACRO-10, GERALMENTE PODE
SER CHAMADO EM FORTRAN COMO FUNCAO OU SUBROTINA.

#PARAM. REPRESENTA A NUMERO DE PARAMETROS DO SUBPROGRAMA.

OBS.: ESTA TABELA FOI OBTIDA COMO SAIDA DO PROGRAMA UTILITARIO "FUDGE2", NA IMPRESSAO DOS "ENTRY-POINTS" DE UM PROGRAMA RELOCATAVEL (NA ORDEM DE ENTRADA), ORDENADO ALFABETICAMENTE PELO UTILITARIO "SORT", E COMPLETADO COM INFORMACOES COM O EDITOR-CORRETOR DE TEXTOS "TECO", QUE TAMBEM FORNECEU UM INDICE DE FUNCOES E SUBROTINAS NOS ARQUIVOS DO PROGRAMA-FONTE.

X - GLOSSARIO (ORIENTADO PARA O CONTEXTO DO TRABALHO).

ACHAR: RESULTADO DE UMA OPERACAO DE BUSCA OU RECUPERACAO DE INFORMACAO.

ALFANUMERICO: CARATER ALFANUMERICO; CONSISTE DA REPRESENTACAO DE SIMBOLOS ALFABETICOS, NUMERICOS E CARACTERES ESPECIAIS.

ALGOL: ALGORITHMIC LANGUAGE; LINGUAGEM CRIADA ESPECIALMENTE PARA DEFINICAO E DESCRICAO DE ALGORITMOS.

ALIAS: NOME-DE-LISTA; VARIAVEL DO FORTRAN, CUJO CONTEUDO ESTA' NUM FORMATO ESPECIAL (CHAMADO FORMATO DE NOME), QUE APONTA PARA A CABECA-DE-LISTA DE UMA LISTA.

APAGAR: DEVOLVER PARA LAVS.

APAGAR CELULA: DEVOLVER UMA CELULA PARA LAVS.

APAGAR LISTA: DEVOLVER UMA LISTA PARA LAVS.

APONTADOR: UMA POSICAO, VARIAVEL OU CAMPO CUJO CONTEUDO E' O ENDEREÇO DE MAQUINA DE UMA CELULA ESPECIFICA.

ARGUMENTO DE SUBPROGRAMA: DIZ-SE DOS PARAMETROS TRANSMITIDOS NUMA CHAMADA DE SUBPROGRAMA.

ARMAZENAMENTO AUXILIAR: ARMAZENAMENTO SECUNDARIO; UNIDADE DE ARMAZENAMENTO DE DADOS, GERALMENTE DE ALTA CAPACIDADE, COM COMPONENTES ELETRO-MECANICOS.

ARRAY: ESTRUTURA DE DADOS DAS MAIS SIMPLES, QUE O FORTRAN POSSUI; UMA REFERENCIA A UM ELEMENTO DO ARRAY E' FEITA ATRAVEZ DE INDICES.

ARRAY UNI-DIMENSIONAL: TAMBEM CHAMADO DE VETOR; E' UMA ESTRUTURA DE DADOS ONDE A REFERENCIA A UM ELEMENTO ESPECIFICO E' FEITA ATRAVEZ DE UM INDICE.

ARVORE: ESTRUTURA DE DADOS QUE CONSISTE DE NO'S, UM DELES CHAMADO RAIZ DA ARVORE, OS OUTROS NO'S ESTAO CONTIDOS EM CONJUNTOS DISJUNTOS CADA UM DOS QUAIS E' UMA ARVORE.

ASCENSAO NA ESTRUTURA: NO PERCURSO DE UMA ESTRUTURA, ESTA OPERACAO CONSISTE EM PARTIR DE UM ELEMENTO DE UMA SUBLISTA, PARA A LISTA QUE A TEM COMO SUBLISTA, NESTE CASO DIZ-SE QUE SE SUBIU DE NIVEL NA ESTRUTURA.

ASSEMBLER: LINGUAGEM MONTADORA; LINGUAGEM DE BAIXO-NIVEL; LINGUAGEM ORIENTADA PARA A MAQUINA.

ASSIGN: COMANDO ASSIGN DO FORTRAN.

ATRIBUTO: CODIGO ASSOCIADO A QUALIDADES DA ESTRUTURA DE LISTAS. OS ATRIBUTOS E SEUS RESPECTIVOS VALORES NUMA LISTA DESCRITIVA DE UMA LISTA DADA, PODEM SERVIR PARA ROTULAR AS LISTAS E SUBLISTAS DA ESTRUTURA.

ATTRIBUTE-VALUE LIST: LISTA DESCRITIVA DE UMA LISTA.

ATUALIZAR: NO CONTEXTO DO TRABALHO, ATUALIZAR E' RENOVAR, OU SUBSTITUIR O VALOR DE UM ATRIBUTO DA LISTA DESCRITIVA.

AVANCAR UMA READER: OPERACAO EM QUE UM APONTADOR (LPNTR-LIST POINTER) E' ATUALIZADO NUM PERCURSO DE UMA ESTRUTURA DE LISTAS.

AVANCO A DIREITA: OPERACAO EM QUE O PERCURSO E' EFETUADO SEGUINDO-SE O APONTADOR LNKR DAS CELULAS DA ESTRUTURA.

AVANCO A ESQUERDA: OPERACAO EM QUE O PERCURSO E' EFETUADO SEGUINDO-SE O APONTADOR LNKL DAS CELULAS DA ESTRUTURA.

AVANCO ESTRUTURAL: OPERACAO DE PERCURSO NUMA ESTRUTURA, ONDE AS CELULAS QUE CONTEM O NOME-DE-LISTA DAS SUBLISTAS QUANDO SAO ENCONTRADAS NO PERCURSO, POSSIBILITAM QUE SE DESCA PARA A SUBLISTA DAQUELA LISTA, DIZENDO-SE ASSIM QUE SE DESCEU MAIS UM NIVEL NO PERCURSO DE UMA ESTRUTURA. NO CAMINHO INVERSO, O AVANCO ESTRUTURAL PERMITE SUBIR DE UM NIVEL DA ESTRUTURA PARA UM NIVEL SUPERIOR.

AVANCO LINEAR: OPERACAO DE PERCURSO NUMA ESTRUTURA, ONDE HA O PERCURSO DE UMA LISTA DA ESTRUTURA, SEM SUBIR OU DESCER DE NIVEL.

BINARIO: SISTEMA BINARIO DE NUMERACAO; SISTEMA NUMERICO DE BASE 2; SISTEMA DE NUMERACAO COM DOIS DIGITOS (0 E 1).

BIT: BINARY DIGIT; DIGITO BINARIO; TOMA VALOR 0 OU 1.

BIT PATTERN; MASCARA CONTENDO 0'S E 1'S, DEFININDO OS CAMPOS DE INTERESSE PARA COPIA DE BITS ASSOCIADOS AOS BITS 1'S DA MASCARA. BIT MANIPULATION; CHARACTER MANIPULATION.

BOTTOM; FUNDO; CELULA DE FUNDO DE UMA LISTA; CELULA MAIS-A-DIREITA.

BULK ERASURE; TECNICA DE DEVOLUCAO DE LISTAS INTEIRA PARA A LISTA LIVRE, ADOTADA NO SLIP, PELAS FACILIDADES OBTIDAS COM OS DOIS APONTADORES LNKL E LNKR DE CADA CELULA.

CABECA-DE-LISTA: UMA CELULA ESPECIAL DE CADA LISTA, QUE CONTEM O APONTADOR PARA A CELULA DE FUNDO DA LISTA, PARA A CELULA DE TOPO, CONTEM UMA POSSIVEL MARCA DE LISTA, APONTADOR PARA UMA LISTA DESCRITIVA DA LISTA, E UM CONTADOR DE REFERENCIAS QUE PROTEGE A LISTA DA OPERACAO DE "DELECAO", PARA FINS DE GARBAGE COLLECTION.

CALL-RETURN; MECANISMO UTILIZADO EM SUBPROGRAMAS FECHADOS, ONDE UMA REFERENCIA AO SUBPROGRAMA (CALL) PROVOCA UM DESVIO PARA O PONTO DE ENRADA DO SUBPROGRAMA, E O RETORNO (RETURN) PROVOCA UM DESVIO PARA A PROXIMA INSTRUCAO APOS O CALL.

CAMPO; SUBITEM DE UMA CELULA, CONTENDO UMA INFORMACAO ESPECIFICA, OU UM CODIGO.

CANDIDATA: CELULA CANDIDATA.

CELULA: UM MODULO BASICO DE INFORMACAO; NO SLIP, A CELULA CONSISTE DE UM PAR DE PALAVRAS CONSECUTIVAS.

CELULA CANDIDATA; NUMA OPERACAO DE AVANCO DA LEITORA DE UMA LISTA, UMA CELULA CANDIDATA E' A CELULA QUE ESTA SENDO APONTADA PELA

LEITORA, ELA É CANDIDATA A SATISFAZER COM SUCESSO O CRITÉRIO DE TÉRMINO DA OPERAÇÃO DE AVANÇO ESCOLHIDO PELO PROGRAMADOR.

LISTA

CELULA MAIS-A-DIREITA: NO SLIP, É UMA CELULA DE UMA LISTA, CUJO LNKR APONTA PARA A HEADER DA LISTA; CELULA DE FUNDO DA LISTA.

CELULA MAIS-A-ESQUERDA: NO SLIP, É UMA CELULA DE UMA LISTA, CUJO LNKL APONTA PARA A HEADER DA LISTA; CELULA DE TOPO DA LISTA.

CELULA MAIS-ABAIXO: CELULA DE FUNDO DA LISTA; BOTTOM

CELULA MAIS-ACIMA: CELULA DE TOPO DA LISTA; TOP.

CELULA SLIP: UM PAR CONSECUTIVO DE PALAVRAS; MODULO BASICO DE INFORMACAO DO SLIP.

CHAMADA DE SUBPROGRAMA: DESVIO PARA UM PONTO DE ENTRADA DE UM SUBPROGRAMA.

CHAMADA-RETORNO: CALL-RETURN; MECANISMO DE ACESSO A UM SUBPROGRAMA.

CICLO: LOOP; UM CICLO CONSISTE DE UM CONJUNTO DE COMANDOS QUE SAO EXECUTADOS REPETIDAMENTE.

CICLO DE DO: CICLO CONTROLADO POR UM CONTADOR.

CIRCULAR LIST: LISTA CIRCULAR.

CLASSE DE CELULA: NO SLIP, PARA OPERACOES DE AVANÇO, AS CELULAS SAO CLASSIFICADAS EM CELULAS "N", "E" E "W".

COMANDO ASSIGN: COMANDO DO FORTRAN QUE PERMITE ASSINALAR UM "LABEL" SIMBOLICO A UM NUMERO DE COMANDO.

COMANDO COMMON: COMANDO DO FORTRAN QUE PERMITE DEFINIR UMA AREA COMUM DE DADOS ENTRE PROGRAMAS E SUBPROGRAMAS.

COMANDO DIMENSION: COMANDO DO FORTRAN QUE PERMITE RESERVAR AREA PARA ARRAYS, ALEM DE FORNECER INFORMACOES PARA O CALCULO DO ENDEREÇO DE UM ELEMENTO DE UM ARRAY, DADOS SEUS INDICES.

COMANDO DO: COMANDO DO FORTRAN QUE PERMITE A EXECUCAO REPETITIVA DE UM CONJUNTO DE COMANDOS CONTIDOS NO CHAMADO DOMINIO DO "DO", ALEM DISSO, O COMANDO DO, GERALMENTE PROVIDENCIA UMA CONTAGEM OU UM INDICE PARA ITERACAO.

COMANDO FORMAT: COMANDO DO FORTRAN QUE PERMITE A EDICAO DE DADOS, PARA CODIFICACAO OU DECODIFICACAO DA INFORMACAO, TORNANDO-A LEGIVEL PELO HOMEM.

COMANDO GO TO: COMANDO DO FORTRAN QUE POSSIBILITA UM DESVIO INCONDICIONAL.

COMANDO IF: COMANDO DO FORTRAN, QUE POSSIBILITA UM DESVIO CONDICIONAL.

COMANDOS DE E/S: COMANDOS DO FORTRAN, QUE POSSIBILITAM A COMUNICACAO DO PROGRAMA COM UM OU VARIOS DISPOSITIVOS DE E/S.

COMPARTILHAMENTO DE DADOS: POSSIBILIDADE DE ARMAZENAMENTO, ONDE ALGUNS DADOS SAO COMUNS A VARIAS ESTRUTURAS.

COMPARTILHAMENTO DE SUBLISTAS: NO SLIP, HA O COMPARTILHAMENTO DE SUBLISTAS,

I.E., UMA SUBLISTA PODE SER SUBLISTA DE VARIAS LISTAS SIMULTANEAMENTE.

COMPUTACAO: UMA OPERACAO ENVOLVENDO OPERACOES ARITMETICAS E FUNCIONAIS SOBRE ALGUNS DADOS, FORNECENDO UM RESULTADO.

CONSTANTE: NO FORTRAN, UMA CONSTANTE E' UM NUMERO INTEIRO, UM NUMERO REAL, OU COMPLEXO, OU DE PRECISAO-DUPLA, OU UM ALFANUMERICO, OU BOOLEANO.

CONT: FUNCAO PRIMITIVA DO SLIP, QUE FORNECE COMO RESULTADO O CONTEUDO DO CAMPO CUJO ENDEREÇO ESTA' CONTIDO NO SEU PARAMETRO. O VALOR DESTA FUNCAO E' "REAL".

CONTADOR DE REFERENCIAS: CAMPO DE UMA HEADER DE UMA LISTA, QUE E' A CONTAGEM DE VEZES QUE UMA LISTA E' SUBLISTA DE OUTRAS LISTAS. ESTE CAMPO E' UTILIZADO NA PROTECAO DE "DELECAO" DA LISTA PARA FINS DE GARBAGE-COLLECTION.

CONTEUDO: VALOR DE UM CAMPO ESPECIFICO, OU DE UMA VARIAVEL.

COPIAR: OPERACAO QUE CONSISTE EM DUPLICAR UMA CELULA OU LISTA.

CORPO-DE-LISTA: SAO TODAS AS CELULAS DE UMA LISTA, EXCETO A SUA HEADER.

CRIAR: A OPERACAO DE CRIACAO DE UMA LISTA CONSISTE EM RETIRAR UMA CELULA DO TOPO DE LAVS, TRANSFORMANDO-A NA CABECA-DE-LISTA DE UMA LISTA VAZIA, ESTABELECENDO ID=2, LNKL=LNKR=ENDERECO DA CELULA TOMADA.

CRITERIO DE TERMINO: NUM PERCURSO DE UMA ESTRUTURA DE LISTAS, UM AVANCO E' BEM SUCEDIDO QUANDO FOI SATISFEITO UM CRITERIO DE TERMINO ESCOLHIDO PELO PROGRAMADOR.

DADOS: INFORMACAO; UM PROCESSO COMPUTACIONAL, GERALMENTE, PROCESSA DADOS, QUE SAO CONSIDERADOS ENTRADA PARA O PROCESSO.

DATA: DADOS; INFORMACAO.

DECIMAL: BASE DECIMAL; SISTEMA NUMERICO DE BASE 10; CODIGO DE ARMAZENAMENTO DE DADOS, ONDE CADA DIGITO E' CODIFICADO EM BINARIO.

DELECAO: OPERACAO DE APAGAR. (NEOLOGISMO TECNICO)

DELET: APAGAR.

DESCIDA NA ESTRUTURA: OPERACAO DE PERCURSO NUMA ESTRUTURA DE LISTAS QUE NO ENCONTRO DE UMA SUBLISTA, A LEITORA DA ESTRUTURA GUARDA A HISTORIA DO PERCURSO, SENDO PROVIDENCIADA UMA NOVA LEITORA QUE PROSSEGUE O PERCURSO NA SUBLISTA ENCONTRADA, NESTE CASO, DIZ-SE QUE SE DESCEU MAIS UM NIVEL NA ESTRUTURA.

DESCRIPTIVE LIST: LISTA DESCRITIVA.

DESCRIPTOR LIST: LISTA DESCRITIVA.

DESEMPILHAR: OPERACAO QUE CONSISTE EM RECUPERAR UMA INFORMACAO NO TOPO (FUNDO) DE UMA LISTA (PILHA), DEVOLVENDO AQUELA CELULA PARA LAVS.

DESIGNACAO DE CELULA: NONE DE UMA CELULA; CLASSIFICACAO AS VEZES NECESSARIA PARA HOMOGENIZAR (EXEMPLO: CLASSIFICAR AS CELULAS

EM CELULAS N, E, W PARA AS OPERACOES DE AVANCO).

DEVOLUCAO DE CELULA: OPERACAO DE ATUALIZACAO DOS APONTADORES DA CELULA PARA QUE ELA FAÇA PARTE DE LAVS.

DEVOLVER: DEVOLUCAO DE UMA CELULA OU LISTA.

DIMENSION: COMANDO DIMENSION DO FORTRAN; RESERVA DE POSICOES DE MEMORIA.

DIRECAO DE AVANCO: SENTIDO DE AVANCO; AVANCO PARA A DIREITA OU PARA A ESQUERDA SEGUINDO OS APONTADORES DAS CELULAS DA LISTA.

DO: COMANDO DO DO FORTRAN.

DOMINIO DO COMANDO DO: CONJUNTO DE COMANDOS FORTRAN QUE SERAO EXECUTADOS TANTAS VEZES DEPENDENDO DO CONTADOR DO "DO", E DE DESVIOS INTERNOS AO DOMINIO.

ELEMENTO: MEMBRO: CELULA: CELULA DE ^{uma} ~~UMA~~ LISTA QUE NAO CONTENHA UM NOME-DE-LISTA.

EMBEDDING: TECNICA DE EXTENSAO DE UMA LINGUAGEM EM QUE A EXPANSAO FICA CONTIDA NO SEIO DA LINGUAGEM DE BASE, QUE TEM O NOME DE LINGUAGEM HOSPITALAIREIRA.

EMPILHAR: COLOCAR UMA CELULA NO TOPO DE UMA LISTA.

EMPTY LIST: LISTA VAZIA.

ENDERECO: UM DADO (OU INSTRUCAO) ARMAZENADO NA MEMORIA TEM REFERENCIA POR UM ENDERECO UNICO.

ENDERECO DA CELULA: ENDERECO DA PRIMEIRA PALAVRA DA CELULA SLIP.

ENDERECO DE MAQUINA: ENDERECO NA MEMORIA DE UM DADO OU INSTRUCAO.

ENDERECO DE RETORNO: NUM SUBPROGRAMA O ENDERECO DE RETORNO E' O ENDERECO DA PROXIMA INSTRUCAO APOS A CHAMADA DO SUBPROGRAMA.

ENDERECO ESPECIFICADO: NO TEXTO, ENDERECO ESPECIFICADO POR UMA VARIAVEL SIGNIFICA: O ENDERECO CUJO VALOR INTEIRO E' O CONTEUDO DA VARIAVEL

ENTIDADE: ENTE; UNO; TRATAMENTO DE UMA ESTRUTURA DE LISTAS COMO UMA ENTIDADE; SIGNIFICA TRATAR A ESTRUTURA COMO UM TODO.

ESTADO FINAL DE UMA READER: NUMA OPERACAO DE AVANCO, E' O ESTADO EM QUE A READER PERMANESCE APOS O TERMINO DA OPERACAO.

ESTADO INICIAL DE UMA READER: E' O ESTADO EM QUE A READER ESTÁ APOS A SUA CRIACAO, OU SUA INICIALIZACAO (LOFRDR, INITRD) I.E., APONTANDO PARA A CABECA-DE-LISTA DA LISTA A ELA ASSOCIADA.

ESTRUTURA DE DADOS: METODO DE ORGANIZACAO DOS DADOS NUMA COMPUTACAO, QUE FACILITA A MANIPULACAO DOS DADOS, E EM MUITOS CASOS PERMITE UMA ECONOMIA DE MEMORIA E TEMPO DE PROCESSAMENTO.

ESTRUTURA DE LISTAS: E' UM CONJUNTO DE LISTAS, DAS QUAIS, UMA (CHAMADA LISTA PRINCIPAL DA ESTRUTURA, OU LISTA DE NIVEL 0), TEM AS OUTRAS COMO SUBLISTAS OU SUBLISTAS DE SUAS SUBLISTAS, ETC.

FALSE: VALOR BOOLEANO; FALSO.

FIELD: CAMPO.

FILA: ESTRUTURA DE INFORMACAO ONDE A INSERCAO DE INFORMACAO E' FEITA NUMA EXTREMIDADE E A RETIRADA DE INFORMACAO E' DA OUTRA EXTREMIDADE (FIFO).

FLAG: MARCA.

FORMAT: COMANDO FORMAT DO FORTRAN.

FORMATO: PADRAO.

FORMATO DA LEITORA SEQUENCIAL: NO SEU ESTADO INICIAL, E' UMA PALAVRA CONTENDO 2, LBOT, 0, LTOP, I.E., E' O CONTEUDO DA PRIMEIRA PALAVRA DA CELULA DA HEADER DA LISTA. NOS ESTAGIOS INTERMEDIARIOS CONTEM SEMPRE O CONTEUDO DA PRIMEIRA PALAVRA DA CELULA PARA A QUAL A LEITORA SEQUENCIAL APONTAVA.

FORMATO DE NOME: UMA PALAVRA DE COMPUTADOR ESTA' NO FORMATO-DE-NOME SE OS CAMPOS DAQUELA POSICAO SAO:

ID=0

LNKL=ENDERECO DE MAQUINA DA HEADER DE UMA LISTA

ID2=0

LNKR=LNKL

FORMATO DE NOME-DE-LISTA: FORMATO DE NOME

FORMATO DE HEADER: NO FORMATO DE READER, UMA CELULA TEM:
NA PRIMEIRA PALAVRA DA CELULA: ID=3, LNKL=LPNTR, LNKR=APONTADOR PARA A PROXIMA READER NA PILHA DE READER'S.
NA SEGUNDA PALAVRA DA CELULA: LNKL=ENDERECO DA HEADER DA LISTA ASSOCIADA. LNKR=LCNTR

FORTRAN: FORMULA TRANSLATION.

FREE LIST: LISTA LIVRE.

FUNCAO: SUBPROGRAMA QUE FORNECE (AO MENOS) UM VALOR.

FUNCOES PRIMITIVAS FUNCOES DE BASE; OPERACOES SOBRE AS QUAIS TODO O SISTEMA SLIP ESTA' CRIADO, SAO FUNCOES DE MANIPULACAO DE CAMPOS DE CELULA E ENDERECOS.

FUNCTION: FUNCAO.

FUNCTION CALL: CHAMADA DE FUNCAO.

FUNDO: CELULA DE FUNDO; E' A CELULA QUE ESTA' SENDO APONTADA PELO LNKL DA HEADER DA LISTA SLIP.

GARBAGE COLLECTOR: MECANISMO QUE PROVIDENCIA AREA LIVRE PARA LAVS, EM TODA SITUACAO EM QUE UMA CELULA UTIL (OU LISTA UTIL) DEIXA DE TER UTILIDADE. NO SLIP, A LISTA LIVRE E' INICIALIZADA (INITAS). A DEVOUCAO DE CELULA A COLOCA NO FUNDO DE LAVS (RCCELL), NA RETIRADA DE CELULA POR NUCELL, HA UMA VERIFICACAO SE A CELULA DO TOPO DE LAVS E' UMA CELULA QUE PODE SER RETIRADA DE LAVS, (UMA HEADER NEM SEMPRE PODE SER RETIRADA, BASTANDO QUE SEJA UMA HEADER DE UMA LISTA QUE E' SUBLISTA DE OUTRAS LISTAS),

OUTRA ROTINA ASSOCIADA AO GARBAGE-COLLECTOR DO SLIP É A
IRALST, QUE DEVOLVE UMA LISTA INTEIRA PARA LAVS.

GO TO: COMANDO GO TO DO FORTRAN.

GOTO: COMANDO GO TO DO FORTRAN.

GRAFO: ESTRUTURA DE DADOS EM QUE ALGUNS NO'S SÃO INTERRELACIONADOS.
OS GRAFOS PODEM SER DIRIGIDOS OU NÃO. UM GRAFO TEM NO'S
ESPECIAIS CHAMADOS FONTES E OUTROS CHAMADO SUMIDOUROS.

HARDWARE: PARTES FÍSICAS, ELETRÔNICAS E ELETRO-MECÂNICAS DE UM SISTEMA
DE COMPUTAÇÃO.

HEADER: CABEÇA-DE-LISTA; UMA CELULA ESPECIAL DE UMA LISTA QUE CONTEM
ALGUMAS INFORMAÇÕES SOBRE A LISTA.

HISTORIA DO PERCURSO NA ESTRUTURA: NO PERCURSO DE UMA ESTRUTURA DE LISTAS,
UMA PILHA DE READER'S CONTEM A HISTORIA DO PERCURSO, I.E.,
AS INFORMAÇÕES QUE PERMITEM A SUBIDA DE NÍVEL DE UMA SUBLISTA PARA
A LISTA QUE A TEM COMO SUBLISTA ESTÃO CONTIDAS NA PILHA DE READERS.

HOLLERITH: CÓDIGO DE INFORMAÇÃO

ID: FUNÇÃO PRIMITIVA DO SLIP QUE FORNECE UM VALOR INTEIRO QUE É O
CONTEÚDO DO CAMPO DE IDENTIFICAÇÃO DA CELULA.

ID2: FUNÇÃO PRIMITIVA DO SLIP QUE FORNECE COMO VALOR INTEIRO O CONTEÚDO
DOS DOIS PRIMEIROS BITS DA SEGUNDA METADE DA PRIMEIRA PALAVRA DE
UMA CELULA.

IF: COMANDO IF DO FORTRAN.

IMPLICITAMENTE LIGADAS: AS DUAS PALAVRAS DE UMA CELULA SLIP SÃO IMPLICITA-
MENTE LIGADAS POIS AS DUAS PALAVRAS SÃO CONSECUTIVAS.

INDEXAÇÃO: DISPOSITIVO DE LINGUAGENS DE ALTO NÍVEL (COMO O FORTRAN) QUE
PERMITE REFERENCIAR UM ELEMENTO ESPECÍFICO DE UM ARRAY.

INFO: ABREVIACÃO DE INFORMAÇÃO; CONTEÚDO DO CAMPO DE INFORMAÇÃO.

INFORMAÇÃO: DADO; ENTRADA PARA UM PROCESSAMENTO OU COMPUTAÇÃO.

INFORMATION: INFORMAÇÃO

INHALT: FUNÇÃO PRIMITIVA DO SLIP CUJO VALOR É O CONTEÚDO DA CELULA
QUE É SEU PARÂMETRO. INHALT TEM MODO INTEIRO E CONT COM A MESMA
FUNÇÃO TEM MODO REAL, PARA EVITAR OS PROBLEMAS DE CONVERSÃO
AUTOMÁTICA DE MODOS DO FORTRAN.

INICIALIZAÇÃO: OPERAÇÃO GERALMENTE NECESSÁRIA ANTES DA EXECUÇÃO DE UMA
COMPUTAÇÃO OU PROCESSAMENTO.

INSERÇÃO: OPERAÇÃO DE INSERIR.

INSERIR: COLOCAR UM DADO (UMA CELULA) NA LISTA.

INTEGER: INTEIRO.

INTEIRO: TIPO DE DADO NUMERICO DO FORTRAN

LABEL: ROTULO

LABEL SIMBOLICO: ROTULO SIMBOLICO

LAVS: LIST OF AVAILABLE SPACES; LISTA DE ESPACOS DISPONIVEIS.

LCNTR: LEVEL COUNTER; CAMPO DE UMA READER QUE DIZ QUE NIVEL DA ESTRUTURA A READER ESTA' NO PERCURSO DA ESTRUTURA.

LEFT: ESQUERDA

LEFT-POINTER: APONTADOR PARA A ESQUERDA

LEFTMOST CELL: CELULA MAIS-A-ESQUERDA

LEITORA: UMA CELULA COM FORMATO ESPECIAL, QUE APONTA PARA UMA CELULA DA LISTA A QUAL A LEITORA ESTA' ASSOCIADA.

LEITORA SEQUENCIAL: UMA POSICAO DE MAQUINA COM APONTADORES PARA A PROXIMA CELULA A ESQUERDA E A DIREITA DE UMA CELULA ESPECIFICA (CONTEM O CONTEUDO DA PRIMEIRA PALAVRA DA CELULA ESPECIFICADA).

LEVEL: NIVEL

LINEAR LIST: LISTA LINEAR

LINGUAGEM: LINGUAGEM DE PROGRAMACAO OU LINGUAGEM NATURAL.

LINGUAGEM DE ALTO NIVEL: LINGUAGEM ORIENTADA PARA O PROBLEMA OU PARA O USUARIO (PROGRAMADOR)

LINGUAGEM DE BAIXO NIVEL: LINGUAGEM ORIENTADA PARA A MAQUINA; ASSEMBLER

LINGUAGEM DE PROGRAMACAO: LINGUAGEM DE COMUNICACAO HOMEM-COMPUTADOR.

LISP: LINGUAGEM DE PROCESSAMENTO DE LISTAS COM ENFASE NO FORMALISMO E RECURSAO.

LIST: LISTA.

LIST OF AVAILABLE SPACES: LISTA DE ESPACOS DISPONIVEIS; LISTA LIVRE.

LIST POINTER: APONTADOR PARA UMA HEADER DE UMA LISTA.

LIST STRUCTURE: ESTRUTURA DE LISTA

LIST-HEADER: CABECA-DE-LISTA

LISTA: UMA ESTRUTURA DE DADOS CONSISTINDO DE CELULAS, CADA CELULA CONTEM DADOS E/OU APONTADORES PARA OUTRAS CELULAS DA LISTA.
UMA LISTA TEM UMA CELULA ESPECIAL QUE E' CHAMADA CABECA-DE-LISTA (HEADER) DA LISTA, AS OUTRAS CELULAS DA LISTA CONSISTEM DO CHAMADO CORPO-DA-LISTA.
UMA LISTA QUE SO' TEM HEADER E' DITA SER UMA LISTA VAZIA.

LISTA CIRCULAR: E' UMA LISTA NA QUAL CADA CELULA APONTA PARA A PROXIMA, E A ULTIMA APONTA PARA A PRIMEIRA, UMA CELULA ESPECIAL CHAMADA HEADER DA LISTA, APONTA PARA A PRIMEIRA CELULA DA LISTA.

LISTA DE ATRIBUTOS: LISTA DESCRITIVA; LISTA QUE CONTEM ATRIBUTOS E SEUS RESPECTIVOS VALORES

LISTA DE ESPACOS DISPONIVEIS: LISTA LIVRE; LISTA QUE CONTEM AS CELULAS QUE NUM DADO INSTANTE PODEM SER REQUISITADAS PARA UTILIZACAO.

LISTA DESCRITIVA: LISTA QUE CONTEM ATRIBUTOS E SEUS RESPECTIVOS VALORES.

LISTA DUPLAMENTE LIGADA: LISTA SIMETRICA; LISTA ONDE CADA CELULA TEM DOIS APONTADORES, UM APONTA PARA A CELULA ANTERIOR, OUTRO APONTA PARA A CELULA SEGUINTE NA LISTA.

LISTA LINEAR: LISTA EM QUE CADA CELULA TEM AO MENOS DOIS CAMPOS: UM PODE CONTER DADO OU APONTADOR, O OUTRO E' UM APONTADOR PARA A PROXIMA CELULA DA LISTA.

LISTA LIVRE: LISTA DE ESPACOS DISPONIVEIS; LAVS

LISTA PRINCIPAL: NUMA ESTRUTURA DE LISTA E' A LISTA QUE LIDERA A ESTRUTURA I.E., E' A LISTA QUE TEM TODAS AS OUTRAS LISTAS COMO SUAS SUBLISTAS OU SUBLISTAS DE SUAS SUBLISTAS.

LISTA PRIVADA: LISTA COM USO RESTRITO.

LISTA PUBLICA: LISTAS ESPECIAIS DO SISTEMA SLIP, CUJAS HEADER'S SAO VARIAVEIS EM COMMON, PODENDO PORTANTO SEREM USADAS POR TODOS PROGRAMAS E SUBPROGRAMAS QUE LHE FIZEREM REFERENCIA.

LISTA SIMETRICA: LISTA EM QUE CADA CELULA TEM DOIS APONTADORES, UM PARA A CELULA ANTERIOR E UM PARA A CELULA SEGUINTE NA LISTA; NO SLIP AS LISTAS SAO ESSENCIALMENTE SIMETRICAS.

LISTA SLIP: LISTA SIMETRICA DO SLIP.

LISTA UTIL: UMA LISTA PODE CONTER INFORMACAO UTIL, OU ESTAR PENDURADA NA LISTA LIVRE.

LISTA VAZIA: LISTA QUE SO TEM HEADER.

LNKL: CAMPO DE UMA CELULA SLIP QUE CONTEM UM APONTADOR PARA A CELULA A ESQUERDA.

LNKR: CAMPO DE UMA CELULA SLIP QUE CONTEM UM APONTADOR PARA A CELULA A DIREITA.

LPNTR: LIST POINTER; NUMA READER E' O CAMPO QUE APONTA PARA UMA CELULA DA LISTA ASSOCIADA A READER.

LTAG: FUNCAO CUJO VALOR BOOLEANO E' VERDADEIRO OU FALSO DEPENDENDO DO CORRESPONDENTE BIT DE MARCA DE ID2 SER 1 OU 0.

MACRO: LINGUAGEM MONTADORA COM RECURSOS DE MACRO-MONTAGEM.

MADOV: FUNCAO PRIMITIVA DO SLIP QUE FORNECE COMO VALOR INTEIRO O ENDEREÇO DE MAQUINA DE UMA VARIÁVEL FORTRAN.

MAIN LIST: LISTA PRINCIPAL

MANIPULACAO DE LISTAS: PROCESSAMENTO DE LISTAS; OPERACOES DE CRIACAO,

ATUALIZACAO, DELECAO DE LISTAS.

MASCARA: PALAVRA QUE CONTEM ALGUNS BITS 1, QUE SERVEM PARA SELECIONAR BITS ESPECIAIS DE UMA PALAVRA MASCARADA.

MASK: MASCARA.

MECANISMO: DISPOSITIVO QUE PERMITE (OU FACILITA) UMA SERIE DE OPERACOES.

MEMBRO: UMA CELULA E' MEMBRO DE UMA LISTA QUANDO SEUS APONTADORES APONTAM PARA OUTRAS CELULAS DA LISTA.

MODIFICACAO DE ESTADO: OPERACAO QUE CONSISTE EM ATUALIZAR OS CAMPOS DE UMA READER PARA QUE ELA APONTE PARA CELULAS SUCESSIVAS, DIZENDO-SE QUE EM CADA PASSO A LEITORA MUDOU DE ESTADO.

MODO: NO FORTRAN, AS VARIAVEIS PODEM TER MODO IMPLICITO REAL OU INTEIRO, DEPENDENDO DA PRIMEIRA LETRA DO NOME DA VARIAVEL.

MODULO: ELEMENTO BASICO, MENOR ELEMENTO.

NAME: NOME

NEXT-ABOVE: PROXIMO-ACIMA

NEXT-BELOW: PROXIMO-ABAIXO

NEXT-TO-THE-LEFT: PROXIMO-A-ESQUERDA

NEXT-TO-THE-RIGHT: PROXIMO-A-DIREITA

NIVEL: NUMA ESTRUTURA, CADA SUBLISTA TEM NIVEL SUPERIOR A LISTA DA QUAL ELA E' SUBLISTA, A LISTA PRINCIPAL TEM NIVEL 0.

NOME DE VARIAVEL: NO FORTRAN, UMA VARIAVEL TEM UM NOME, A UMA VARIAVEL AINDA ESTA ASSOCIADOS SEU ENDEREÇO, SEU VALOR E SEU MODO.

NOME-DE-LISTA: UMA VARIAVEL PODE CONTER UM NOME-DE-LISTA, ESTANDO NO FORMATO ESPECIAL CHAMADO FORMATO DE NOME.

OCTAL: SISTEMA DE NUMERACAO DE BASE 8.

OPERACAO: INSTRUCAO.

OPERACAO DE AVANCO: DIZ-SE DE UMA OPERACAO DE PERCURSO NUMA ESTRUTURA DE LISTAS. NO SLIP, OS MECANISMOS DA READER E DA LEITORA-SEQUENCIAL PERMITEM O PERCURSO NUMA ESTRUTURA DE LISTAS.

OPERACAO DE DESVIO: INSTRUCAO DE DESVIO; NUM PROGRAMA FORTRAN, AS INSTRUCOES SAO EXECUTADAS SEQUENCIALMENTE, UMA APOS A OUTRA, A MENOS DAS INSTRUCOES DE DESVIO E DOS CICLOS. AS INSTRUCOES DE DESVIO PODEM SER CONDICIONAIS OU INCONDICIONAIS.

PAGINACAO: SISTEMA QUE PERMITE A UTILIZACAO DE ARMAZENAMENTO SECUNDARIO COMO UMA EXTENSAO NATURAL DO ARMAZENAMENTO PRIMARIO; UM PROGRAMA FICA SEGMENTADO EM PAGINAS. UMA PAGINA PODE ESTAR RESIDENTE NA MEMORIA PRINCIPAL OU NUMA UNIDADE DE ARMAZENAMENTO SECUNDARIO, NESTE CASO, SENDO REQUERIDA UMA PAGINA ELA E' CARREGADA NA MEMORIA, SE HOVER DISPONIBILIDADE DE ESPACO.

PAGING: PAGINACAO.

PALAVRAS CONSECUTIVAS: SAO PALAVRAS DA MEMORIA PRINCIPAL COM ENDERECOS CONSECUTIVOS.

PAR DE PALAVRAS: UM CONJUNTO DE 2 PALAVRAS (GERALMENTE CONSECUTIVAS).

PARAMETRO DE SUBPROGRAMA: DADO QUE E' TRANSMITIDO NUMA CHAMADA DE UM SUBPROGRAMA, UMA TRANSMISSAO DE DADOS PARA SUBPROGRAMAS PODE SER POR VALOR, POR NOME OU POR REFERENCIA, NO FORTRAN, A TRANSMISSAO E' SEMPRE POR REFERENCIA, I.E., ESTA' DISPONIVEL NA CHAMADA DO SUBPROGRAMA O(S) ENDERECO(S) DO(S) PARAMETRO(S).

PATTERN: MODELO

PERCURSO DE UMA ESTRUTURA DE LISTAS: E' UMA OPERACAO DE CAMINHAMENTO (POR MEIO DE APONTADORES) POR CELULAS DA ESTRUTURA, EM BUSCA DE UMA CELULA COM UMA CARACTERISTICA ESPECIAL (PELO TIPO, DADO, ETC)

PILHA: ESTRUTURA DE DADOS, ONDE A INSERCAO DE INFORMACAO E A RETIRADA SAO EFETUADAS NA MESMA EXTREMIDADE, CHAMADA TOPO DA PILHA (LIFO).

PILHA DE READER'S: MECANISMO QUE PRESERVA A HISTORIA DE PERCURSO NUMA ESTRUTURA DE LISTAS.

PL/I: LINGUAGEM DE PROGRAMACAO DE MULTIPLOS PROPOSITOS.

PLEXO: COLECAO DE DADOS ARMAZENADOS EM UMA OU MAIS POSICOES DE MEMORIA COM AS SEGUINTE CARACTERISTICAS: (1) UM DADO PODE SER UM APONTADOR PARA UM PLEXO, (2) PLEXOS PODEM SER CRIADOS OU ABANDONADOS DURANTE A EXECUCAO.

POINTER: APONTADOR

PONTEIRO: APONTADOR.

PONTO-FIXO: INTEIRO; NA ANTIGA NOTACAO DO FORTRAN, NUMERO DE PONTO FIXO PODIA SER UM NUMERO INTEIRO OU UM NUMERO REAL SEM EXPOENTE

PONTO-FLUTUANTE: REAL; NA ANTIGA NOTACAO DO FORTRAN, NUMERO DE PONTO FLUTUANTE ERA UM NUMERO REAL COM EXPOENTE, I.E., CODIFICADO NA NOTACAO CHAMADA EXPONENCIAL, NORMALIZADA OU CIENTIFICA.

POP-DOWN: PUSH-DOWN; EMPILHAR.

POP-UP: DESEMPILHAR

PRIMITIVA: FUNCAO BASICA; FUNCAO DE BASE; CONJUNTO DE FUNCOES QUE PERMITE A CONSTRUCAO DE UM SISTEMA.

PRIVATE LIST: LISTA PRIVADA.

PROCEDURE: PROCEDIMENTO; SUBPROGRAMA; ROTINA.

PROCESSADOR: PROGRAMA OU CONJUNTO DE PROGRAMAS E ROTINAS COM O OBJETIVO DE FAZER UM PROCESSAMENTO ESPECIFICO.

PROFUNDIDADE: NIVEL.

PROGRAMA: PROCEDIMENTO: SEQUENCIA DE COMANDOS PARA EXECUTAR UM CONJUNTO DE COMPUTACOES.

PROGRAMA UTILITARIO: PROGRAMA GENERICO PARA EFETUAR AS OPERACOES PADRONIZADAS E MAIS COMUNS.

PROCESSAMENTO: COMPUTACAO; PROCESSO.

PROCESSAMENTO DE BITS: MANIPULACAO DE BITS.

PROCESSAMENTO DE CARACTERES: MANIPULACAO DE CARACTERES; MANIPULACAO DE SIMBOLOS.

PROCESSAMENTO DE LISTAS: OPERACOES TIPICAS ENVOLVIDAS COM O PROCESSAMENTO DE LISTAS SAO: CRIACAO DE LISTA VAZIA, INSERIR UM DADO NO TOPO DA LISTA, INSERIR DADO NO FUNDO DA LISTA, SUBSTITUIR O DADO DE UMA CELULA, CONSEGUIR O CONTEUDO DE UM CAMPO DE UMA CELULA, APAGAR UMA LISTA, DIVIDIR UMA LISTA, COPIAR UMA LISTA, REMOVER CELULAS DA LISTA, ETC.

PROCESSAMENTO DE PLEXOS: PROCESSAMENTO DE UMA ESTRUTURA DE PLEXOS.

PROCESSAMENTO DE STRINGS: PROCESSAMENTO DE CADEIAS DE CARACTERES.

PROXIMO ELEMENTO A DIREITA: CELULA CUJO ENDEREÇO ESTA CONTIDO NO CAMPO LNKE DE UMA CELULA ESPECIFICA.

PROXIMO ELEMENTO A ESQUERDA: CELULA CUJO ENDEREÇO ESTA CONTIDO NO CAMPO LNKL DE UMA CELULA ESPECIFICA.

PROXIMO-ABAIXO: PROXIMO ELEMENTO A DIREITA.

PROXIMO-ACIMA: PROXIMO ELEMENTO A ESQUERDA.

PUBLIC LIST: LISTA PUBLICA

PUNCH: PERFURAR

PUSH-DOWN: EMPILHAR

QUEUE: FILA

READER: LEITORA

REAL: NO FORTRAN, E' O TIPO DE DADO MAIS SIMPLES, SEUS VALORES ESTAO CONTIDOS NUM SUBCONJUNTO DOS NUMEROS REAIS.

RECURSAO: DISPOSITIVO QUE PERMITE QUE UMA FUNCAO SE REFERENCIE A SI PROPRIA. NO FORTRAN, NAO E' POSSIVEL FAZER UMA RECURSAO PURA, MAS UM MECANISMO PERMITE QUE SEJA EMPILHADO O ENDEREÇO DE RETORNO DA FUNCAO NUMA CHAMADA, E CADA RETORNO DESEMPILHA UM ENDEREÇO DA PILHA.

REFERENCE COUNTER: CONTADOR DE REFERENCIAS.

REGISTRADOR-INDEXADOR: INDICE; REGISTRADOR DE MAQUINA.

REMOVED: RETIRAR UMA CELULA DE LACS.

RESTAURAR: DEVOLVER UMA CELULA PARA DAVS.

RESTORE: RESTAURAR

RESULTADO: VALOR OBTIDO DE UMA COMPUTACAO.

RETORNO DE SUBPROGRAMA: DESVIO DO SUBPROGRAMA PARA A PROXIMA INSTRUCAO APOS A ULTIMA CHAMADA DO SUBPROGRAMA.

RETRACING: PERCURSO INVERSO DE UMA ESTRUTURA.

RETRIEVAL: RECUPERACAO

RIGHT: DIREITA

RIGHT-POINTER: APONTADOR PARA A DIREITA

RIGHTMOST CELL: CELULA MAIS-A-DIREITA

ROTINA: SUBPROGRAMA; PROCEDIMENTO; CONJUNTO DE COMANDOS QUE EXECUTAM OPERACOES ESPECIAIS SOBRE DADOS DE ENTRADAS.

ROTULADO: UMA INSTRUCAO ROTULADA PODE SER REFERENCIADA POR UM COMANDO DE DESVIO. NO FORTRAN, O ROTULO DE UM COMANDO E' UM NUMERO DE COMANDO.

ROTULO: UMA INSTRUCAO PODE TER UM ROTULO, PODENDO ASSIM SER REFERENCIADA POR UM COMANDO DE DESVIO.

RTAG: FUNCAO CUJO VALOR BOOLEANO E' VERDADEIRO OU FALSO, DEPENDENDO DO CORRESPONDENTE BIT DE MARCA DO CAMPO ID2 SER 1 OU 0.

SENTIDO DE AVANCO: AVANCO A DIREITA (OU A ESQUERDA) DEPENDENDO DA FUNCAO DE AVANCO ESCOLHIDA PELO PROGRAMADOR. O AVANCO SEMPRE SEGUE O APONTADOR APROPRIADO DEPENDENDO DO SENTIDO DE PERCURSO QUE O PROGRAMADOR ESCOLHEU.

SEQUENCE READER: LEITORA SEQUENCIAL.

SETDIR: FUNCAO PRIMITIVA DO SLIP QUE ATUALIZA CAMPOS DE UMA CELULA SLIP.

SETIND: FUNCAO PRIMITIVA DO SLIP QUE ATUALIZA CAMPOS DE UMA CELULA SLIP.

SHIFT: DESLOCAMENTO.

SIMETRIA: NO SLIP, A SIMETRIA DECORRE DO FATO QUE OPERACOES QUE PODEM SER EFETUADAS NUM SENTIDO TEM SUAS CORRESPONDENTES NO SENTIDO OPOSTO, OPERACOES NA CELULA DE TOPO TEM SUAS CORRESPONDENTES NA CELULA DE FUNDO DA LISTA.

SISTEMA DE E/S: SISTEMA DE PROGRAMACAO QUE FACILITA AS OPERACOES DE E/S, PROVIDENCIANDO A MANIPULACAO DE FORMATO, EDICAO DE DADOS, INTERRUPCAO DE E/S, BUFFERIZACAO, ETC.

SOFTWARE: SISTEMA DE PROGRAMACAO; CONJUNTO DE PROGRAMAS QUE TORNAM UM SISTEMA DE COMPUTACAO OPERATIVO.

SORT: ORDENACAO

SPLIT: DIVISAO; PARTILHA

STACK: PILHA

STRDIR: FUNCAO PRIMITIVA DO SLIP, UTIL PRINCIPALMENTE PARA EVITAR OS PROBLEMAS DE CONVERSAO AUTOMATICA DE MODO REAL-INTEIRO DO FORTRAN.

STRIND: FUNCAO PRIMITIVA DO SLIP, UTIL PRINCIPALMENTE PARA EVITAR OS PROBLEMAS DE CONVERSAO AUTOMATICA DE MODO REAL-INTEIRO DO FORTRAN.

STRING MANIPULATION: MANIPULACAO DE CADEIAS DE CARACTERES.

SUBIDA NA ESTRUTURA: OPERACAO QUE CONSISTE EM PARTIR DE UMA CELULA DE UMA SUBLISTA E ALCANCAR UMA CELULA DA LISTA QUE A TEM COMO SUBLISTA, E ASSIM SUCESSIVAMENTE, ATE A LISTA PRINCIPAL, SUBINDO UM NIVEL DE CADA VEZ.

SUBLIST: SUBLISTA

SUBLISTA: UMA LISTA CUJO NOME ESTA CONTIDO NO CAMPO DE DADO DE UMA CELULA DE AO MENOS UMA LISTA.

SUBPROGRAMA: PROCEDIMENTO; ROTINA; PROCEDURE; NO FORTRAN, UM SUBPROGRAMA PODE SER UMA FUNCAO OU SUBROTINA.

SUBROTINA: SUBPROGRAMA QUE OPERA SOBRE OS SEUS PARAMETROS DE ENTRADA, RETORNANDO OS VALORES SOBRE OS MESMOS PARAMETROS.

SUBROTINA FECHADA: AS SUBROTINAS DO FORTRAN SAO SUBROTINAS FECHADAS, SENDO COMPILADAS SEPARADAMENTE E O MECANISMO CALL-RETURN PROVIDENCIA OS DESVIOS PARA O COMEÇO DA SUBROTINA E O RETORNO PARA O PROGRAMA QUE CHAMOU.

SUBSTITUIR: TROCAR O VALOR DO CAMPO DE DADO DE UMA CELULA, GERALMENTE RECUPERANDO O ANTIGO VALOR.

SUCCESSO DE UMA OPERACAO DE AVANÇO: PARA O PERCURSO DE UMA ESTRUTURA DE LISTAS O PROGRAMADOR ESCOLHE UMA FUNCAO DE AVANÇO. O SUCESSO DA OPERACAO, DEPENDE DE QUE O MECANISMO DE AVANÇO (READER, PILHA DE READER'S) TENHA ALCANCADO UMA CELULA QUE SATISFAZ O CRITERIO DE TERMINO DE AVANÇO. OS CRITERIOS SELECIONADOS PELO PROGRAMADOR SAO: (1) ACHAR UMA CELULA TIPO "N" (CELULA CONTENDO NOME-DE-LISTA) (2) ACHAR UMA CELULA TIPO "E" (CELULA QUE NAO SEJA UMA HEADER E NAO CONTENHA UM NOME-DE-LISTA) (3) ACHAR A PROXIMA CELULA QUE NAO SEJA UMA HEADER.

SYMBOL MANIPULATION: MANIPULACAO DE CARACTERES; MANIPULACAO DE SIMBOLOS; PROCESSAMENTO DE SIMBOLOS.

TAG: MARCA

TIRAR: RETIRAR UMA CELULA DE LAVS.

TOMA VALOR: ASSIGNACAO DE VALOR; OPERACAO DE SUBSTITUICAO DO VALOR DE UMA VARIAVEL, GERALMENTE COM PERDA DO ANTIGO VALOR.

TOMAR DE LAVS: OPERACAO DE RETIRAR UMA CELULA DE LAVS (FUNCAO NUCELL)

TOP: TOPO; CELULA DE TOPO DE UMA LISTA; TOPO DE UMA PILHA.

TOPO: CELULA DE TOPO; CELULA APONTADA PELO CAMPO LNKR DA HEADER DE
UMA LISTA SLIP

TRACING: PERCURSO DE ESTRUTURA DE LISTAS.

TRAVERSAL: PERCURSO DE UMA LISTA OU ESTRUTURA DE LISTAS.

TRUE: VALOR BOOLEANO VERDADEIRO.

UTILITARIO: PROGRAMA UTILITARIO

VALOR: CONSTANTE, GERALMENTE RESULTANTE DE UMA OPERACAO DE ASSIGNACAO,
DE UMA COMPUTACAO, OU DE UMA OPERACAO DE LEITURA.

VALOR DA EXPRESSAO: VALOR RESULTANTE DO CALCULO DE UMA EXPRESSA ARITMETICA.

VALOR DA FUNCAO: VALOR LIBERADO NUMA COMPUTACAO EFETUADA PELA FUNCAO.

VALOR DA VARIABEL: CONTEUDO DA POSICAO DE MEMORIA ASSOCIADA AQUELA
VARIABEL.

VALOR DO ATRIBUTO: VALOR ASSOCIADO A UM ATRIBUTO NUMA LISTA DESCRITIVA.

VALOR ESPECIFICADO POR UMA VARIABEL: VALOR DA VARIABEL.

VALOR INTEIRO: CONSTANTE INTEIRA.

VARIABEL: NO FORTRAN, UMA VARIABEL TEM UM NOME E NUM DADO MOMENTO UM
VALOR QUE PODE SER MODIFICADO POR UMA LEITURA, POR UMA ASSIGNACAO
OU POR UMA CHAMADA DE SUBPROGRAMA.

VISITACAO: OPERACAO DE VISITAR.

VISITAR: NO SLIP, VISITAR ESTA NO SENTIDO DE DESVIAR PARA UM COMANDO
CUJO ENDEREÇO ESTA ESPECIFICADO, EMPILHANDO O ENDEREÇO DE
RETORNO, PROVIDENCIANDO ASSIM UM MECANISMO DE RECURSAO.

XI - APLICACOES:

ALGUMAS LINGUAGENS DE MANIPULACAO DE SIMBOLOS E PROGRAMAS DE APLICACAO:

- AED: UM SISTEMA DE PROPOSITO GERAL BASEADO EM ALGOL COM FACILIDADES PODEROSAS DE PROCESSAMENTO DE CADEIAS E LISTAS.
- ALGAN: UM CONJUNTO DE ROTINAS FORTRAN PARA RESOLVER SISTEMAS DE EQUACOES LINEARES E FATORACAO DE POLINOMIOS.
- ALGEM: UM CONJUNTO DE ROTINAS SLIP CODIFICADAS PARA MANIPULAR POLINOMIOS COM EXPOENTES VARIAVEIS.
- ALGY: UM SISTEMA PARA MANIPULAR EXPRESSOES.
- ALMS: UM SISTEMA PARA FAZER DIFERENCIACAO.
- ALPAK: UM SISTEMA DE SUBROTINAS PARA MANIPULAR POLINOMIOS E FUNCOES RACIONAIS.
- ALTRAN: UMA EXTENSAO DO FORTRAN QUE INCORPORA ALGUMAS DAS CAPACIDADES DE ALPAK.
- AMBIT: UMA LINGUAGEM PARA EXPRESSAR UM CERTO TIPO DE MANIPULACAO DE STRINGS.
- AUTOMAST: UM SISTEMA PARA RESOLVER SISTEMAS DE EQUACOES DIFERENCIAIS ORDINARIAS.
- AXLE: UMA LINGUAGEM PARA EXPRESSAR UM CERTO TIPO DE MANIPULACAO DE STRINGS.
- COMIT: UMA LINGUAGEM DE PROCESSAMENTO DE STRINGS, ORIGINALMENTE PROJETADA PARA TRADUCAO MECANICA, MAS TAMBEM TEM SIDO USADA PARA OUTRAS APLICACOES.
- CONVERT: UMA LINGUAGEM (BASEADA EM LISP) E ROTINAS PARA MANIPULAR REGRAS DE TRANSFORMACAO, COMO EM COMIT E SNOBOL.
- CORAL: UM SISTEMA DE PROCESSAMENTO DE LISTAS PARA MANIPULACAO DE ESTRUTURAS DE DADOS GRAFICOS.
- DYSTAL: UM SISTEMA DE SUBROTINAS ADICIONADAS AO FORTRAN PARA FAZER UM TIPO LIMITADO DE PROCESSAMENTO DE LISTAS.
- FLIP: UMA NOTACAO E LINGUAGEM (E ROTINAS EM LISP) PARA EXPRESSAR TRANSFORMACOES DE STRINGS, COMO AS DE COMIT E SNOBOL.
- FLPL: FORTRAN-LIST PROCESSING LANGUAGE
UMA SERIE DE SUBROTINAS ADICIONADAS AO FORTRAN EM CONEXAO COM TRABALHO EM TEOREMAS DE GEOMETRIA.
- FORMAC: UMA EXTENSAO DO FORTRAN QUE FAZ MUITOS TIPOS DE MANIPULACAO ALGEBRICA E CAPACIDADES DE PROCESSAMENTO DE STRINGS E LISTAS.
- FORMULA-ALGOL: UMA EXTENSAO DO ALGOL QUE PERMITE TIPOS DE DADOS FORMAIS E SUAS MANIPULACOES, TEM CAPACIDADES DE MANIPULACAO DE CADEIAS E LISTAS.

GPM: UM MACRO-GERADOR QUE INCORPORA FACILIDADES DE MANIPULACAO DE STRINGS.

GRAD-ASSISTANT: UM PROGRAMA LISP PARA FAZER MANIPULACAO ALGEBRICA.

IPL-V: UM SISTEMA DE PROCESSAMENTO DE LISTAS CUJOS COMANDOS SAO SIMILARES AOS DE UMA LINGUAGEM ASSEMBLER, MAS PARA MANIPULACAO DE LISTAS.

L6: UM SISTEMA DE PROCESSAMENTO DE LISTAS QUE POSSIBILITA AO USUARIO ALGUM CONTROLE SOBRE OS FORMATOS DA LISTA.

LIPL: VERSAO LINEARIZADA DO IPL-V.

LISP: UM SISTEMA DE PROCESSAMENTO DE LISTAS COM ENFASE EM RECURSAO E FORMALISMO.

LISP2: UM SISTEMA BASEADO EM ALGOL, MAS TAMBEM CONTENDO MUITAS FACILIDADES E CONCEITOS DO SISTEMA LISP 1.5 ORIGINAL.

MAGIC-PAPER: UM SISTEMA QUE PERMITE A INTERACAO HOMEM-MAQUINA NA MANIPULACAO DE EXPRESSOES MATEMATICAS.

MANIP: UM CONJUNTO DE PROGRAMAS FORTRAN PARA MANIPULAR EXPRESSOES.

MATHLAB: UM SISTEMA QUE PERMITE A INTERACAO HOMEM-MAQUINA NA MANIPULACAO DE EXPRESSOES MATEMATICAS.

METEOR: UM CONJUNTO DE FUNCOES LISP PARA REALIZAR ALGUMAS MANIPULACOES FEITAS PELO COMIT.

NU-SPEAK: UM CONJUNTO DE ROTINAS DE PROCESSAMENTO DE LISTAS ADICIONADAS AO FORTRAN OU FAP.

PANON: UMA LINGUAGEM PARA MANIPULACAO DE SIMBOLOS BASEADA NUMA EXTENSAO PARTICULAR DOS ALGORITMOS DE MARKOV.

PM: UM SISTEMA DE SUBROTINAS PARA MANIPULAR POLINOMIOS COM COEFICIENTES INTEIROS VARIAVEIS.

REFCO-III: UM CONJUNTO DE SUBROTINAS PARA FAZER PROCESSAMENTO DE LISTAS.

SAINT: UM PROGRAMA PARA FAZER INTEGRACAO FORMAL.

SLIP: UM SISTEMA DE SUBROTINAS ADICIONADO AO FORTRAN PARA FAZER PROCESSAMENTO DE LISTAS.

SNOBOL: UM SISTEMA DE MANIPULACAO DE STRINGS.

SYMBAL: UMA EXTENSAO DO ALGOL PARA FAZER MANIPULACAO ALGEBRICA FORMAL.

SYMBOLANG: UM CONJUNTO DE SUBROTINAS UTEIS NA MANIPULACAO DE POLINOMIOS E FUNCOES RACIONAIS.

TIPL: UM PROGRAMA PARA TESTAR PROGRAMAS IPL-V.

TRAC: UMA LINGUAGEM HOMEM-MAQUINA PARA MANIPULACAO DE STRINGS.

TREET: UM SISTEMA DE PROCESSAMENTO DE LISTAS.
VPRPAK: UM CONJUNTO DE SUBROTINAS PARA FAZER ARITMETICA DE
PRECISAO VARIAVEL E COM NUMEROS RACIONAIS.
WISP: UMA LINGUAGEM DE PROCESSAMENTO DE LISTAS SIMPLES,
BASEADA NO LISP.

. RESUMOS DE ALGUNS ARTIGOS QUE DESCREVEM APLICACOES DO SLIP:

SOME EXPERIMENTS IN ALGEBRAIC MANIPULATION BY COMPUTER

LAPIDUS, A. GOLDSTEIN, M.
NEW YORK UNIVERSITY
CACM 8,8 (AUG 65) PP 501-508

UM CONJUNTO DE SUBROTINAS PARA MANIPULACAO ALGEBRICA NO COMPUTADOR IBM 7094 ESCRITAS COM O PROCESSADOR DE LISTAS SLIP. UMA SERIE DE 4 PROGRAMAS DE DIFICULDADES CRESCENTES SAO RESOLVIDOS USANDO ESTAS ROTINAS.

ALGEM - AN ALGEBRAIC MANIPULATOR

GOTLIEB, C. C. NOVAK, R. J.
UNIVERSITY OF TORONTO
ABSTRACT IN CACM 9,8 (AUG 66) PP 549

ALGEM E' UM PACKAGE DE SUBROTINAS ESCRITAS EM SLIP, FORTRAN IV E MAP PARA O IBM 7094-II PARA MANIPULACAO DE EXPRESSOES ARITMETICAS. AS OPERACOES ALGEBRICAS BASICAS DE ALGEM SAO ADICAO, SUBTRACAO, MULTIPLICACAO, DIVISAO E EXPONENCIACAO. ELE E' CAPAZ DE MANIPULAR QUALQUER NUMERO DE VARIAVEIS DE UMA SO' LETRA, EXPOENTES VARIAVEIS, E ACHAR O MAXIMO DIVISOR COMUM DE DOIS POLINOMIOS. TAMBEM SAO INCLUIDAS FUNCOES DE SUBSTITUICAO E DIFERENCIACAO, DETERMINACAO DE COEFICIENTES DE VARIAVEIS ESPECIFICADAS, RESOLUCAO DE EQUACOES LINEARES, ROTINAS BASICAS DE E/S E OUTRAS ROTINAS ARITMETICAS E DE PROPOSITO GERAL. A PRINCIPAL INOVACAO DE ALGEM SOBRE OUTROS MANIPULADORES E' A ASSIGNACAO DE TIPOS DE EXPRESSOES E O USO DE PROCEDIMENTOS PADRAO DE ORDENACAO.

AN ALGEBRAIC MANIPULATOR USING SLIP

LAPIDUS, A. GOLDSTEIN, M. GREENSPAN, S.
NEW YORK UNIVERSITY
ABSTRACT IN CACM 9,8 (AUG 66) PP 549

CONTEM EM DETALHES A DESCRICAO DE UMA NOTACAO PARA A REPRESENTACAO DE EXPRESSOES ALGEBRICAS PARA SER MANIPULADAS POR UM PROCESSADOR DE LISTAS DO TIPO DO SLIP. ESTA NOTACAO EM GERAL, PERMITE QUE EXPRESSOES ALGEBRICAS SEJAM REPRESENTADAS NA FORMA DE ESTRUTURAS DE LISTAS PARA MANIPULACAO ALGEBRICA.

SLIP OPERATIONS ON TREES AND THEIR RELEVANCE TO PROBLEMS OF LINGUISTIC INTEREST

RAMANI, S. (EM [25])

OS DISPOSITIVOS ESSENCIAIS USADOS NOS PROCESSOS DE DESCRICAO E INVESTIGACAO EM LINGUISTICA, TAIS COMO ANALISE SINTATICA, ESTAO PRESENTES NAS LINGUAGENS DE PROCESSAMENTO DE LISTAS. ESTES DISPOSITIVOS INCLUEM A DISPONIBILIDADE DE ARMAZENAMENTO EM PILHAS E RECURSIVIDADE. UM DISPOSITIVO IMPORTANTE DE PROCESSADORES DE LISTAS E' QUE REALIZAM COMPUTACOES SOBRE OPERANDOS ESTRUTURADOS. O SLIP TEM MOSTRADO TER MUITAS FUNCOES BASICAS QUE SAO IDENTIFICAVEIS COM AS OPERACOES BASICAS DE LINGUISTICA. PROCEDIMENTOS PARA GERACAO E ANALISE DE SENTENCAS EM GRAMATICAS SENSIVEIS-AO-CONTEXTO E UM PROCEDIMENTO PARA GERACAO TRANSFORMACIONAL DE SENTENCAS SAO DESCRITOS NO ARTIGO. SAO APRESENTADOS AINDA RESULTADOS DE UMA QUANTIDADE LIMITADA DE EXPERIMENTACOES.

ELIZA - A COMPUTER PROGRAM FOR THE STUDY OF NATURAL LANGUAGE COMMUNICATION
BETWEEN MAN AND MACHINE.

1

WEIZENBAUM, J.

CACM 9,1 (JAN 66) PP 36-45

ELIZA É UM PROGRAMA OPERANDO EM TIME-SHARING NO PROJETO MAC-MIT QUE TORNA POSSÍVEL UMA ESPÉCIE DE CONVERSACÃO NATURAL ENTRE O HOMEM E O COMPUTADOR. AS SENTENÇAS DE ENTRADA SÃO ANALIZADAS COM BASE NAS REGRAS DE DECOMPOSIÇÃO DE ACORDO COM PALAVRAS-CHAVES QUE APARECEM NO TEXTO DE ENTRADA. AS RESPOSTAS SÃO GERADAS POR REGRAS DE REMONTAGEM ASSOCIADAS COM AS REGRAS DE DECOMPOSIÇÃO. OS PROBLEMAS TÉCNICOS FUNDAMENTAIS RELACIONADOS COM ELIZA SÃO:

- (1) A IDENTIFICAÇÃO DA PALAVRA-CHAVE
 - (2) A DESCOBERTA DO CONTEXTO MINIMAL
 - (3) A ESCOLHA DAS TRANSFORMAÇÕES APROPRIADAS
 - (4) GERAÇÃO DE RESPOSTAS NA AUSÊNCIA DE PALAVRAS-CHAVE
 - (5) PROVIDENCIAR AS FACILIDADES DE EDIÇÃO PARA O "SCRIPT" DE ELIZA.
- SÃO DISCUTIDOS ALGUNS ASPECTOS PSICOLÓGICOS E FUTUROS DESENVOLVIMENTOS.

CONTEXTUAL UNDERSTANDING BY COMPUTERS

WEIZENBAUM, J. (MIT)

CACM 10,8 (AUG 67) PP 474-480

É DISCUTIDO UM DESENVOLVIMENTO DO PROGRAMA ELIZA CAPAZ DE CONVERSAR NUMA LINGUAGEM NATURAL. É RESSALTADA A IMPORTÂNCIA DO CONTEXTO DOS ENTENDIMENTOS DA MÁQUINA E DO HOMEM. UM ARGUMENTO É QUE O NÍVEL DE COMPREENSÃO CONSEGUÍDO NUMA CONVERSACÃO PARTICULAR DEPENDE DO PROPOSITO DA CONVERSACÃO E QUE A COMPREENSÃO ABSOLUTA POR AMBAS PARTES É IMPOSSÍVEL.

EXEMPLO DE UTILIZACAO DO DISPOSITIVO VISIT-TERM PARA RECURSAO:

CALCULO DE FATORIAL DE UM NUMERO INTEIRO POSITIVO

```
FUNCTION NFACT(N)
COMMON AVSL,W(100)
ASSIGN 5 TO K
NFACT=INTGER(VISIT(K,PARMT2(N,N)))
RETURN
5 IF(EQUAL(1, TOP(W(1)))) 6,9,6
6 CALL VISIT(K,PARMT2(INTGER(TOP(W(1)))-1,N))
NF = NF*INTGER(TOP(W(1)))
CALL TERM(NF,RESTOR(2))
9 NF = 1
CALL TERM(NF,RESTOR(2))
END

COMMON AVSL, W(100)
DIMENSION SPACE(100)
N=100
CALL INITAS(SPACE,N)
DO 100 I=1,7
N=NFACT(I)
100 PRINT 24, I,N
24 FORMAT(' FATORIAL DE',I3,' =',I)
STOP
END
```

RESULTADO OBTIDO EM CHAMADAS SUCESSIVAS DA FUNCAO NFACT:

FATORIAL DE	1 =	1
FATORIAL DE	2 =	2
FATORIAL DE	3 =	6
FATORIAL DE	4 =	24
FATORIAL DE	5 =	120
FATORIAL DE	6 =	720
FATORIAL DE	7 =	5040

I N D I C E

I - INTRODUCAO	4
• NOVA LINGUAGEM DE PROGRAMACAO ESPECIAL PARA PROCESSAMENTO DE LISTAS VERSUS EXTENSAO DE LINGUAGEM EXISTENTE.	
• LINGUAGEM HOSPITALEIRA - ALGOL X FORTRAN (PROS-E-CONTRAS)	
• O SISTEMA SLIP	5
A CELULA, OS CAMPOS DA CELULA, SIMETRIA AS FUNCOES PRIMITIVAS	
II - LISTAS: CONCEITOS E DEFINICOES	5
• LISTAS SIMETRICAS	
• CELULA: IDENTIFICADOR, APONTADORES, INFORMACAO	
• FUNCOES PRIMITIVAS PARA MANIPULACAO DOS CAMPOS DA CELULA:	6
• EXTRACAO DE INFORMACAO DOS CAMPOS ID, LNKL, LNKR, CONT, INHALT	
• ENDERECO DA CELULA: MADOV	
• ARMAZENAMENTO DE INFORMACAO NOS CAMPOS: SETDIR, SETIND	
• ARMAZENAMENTO DE DADOS NAS PALAVRAS: STRDIR, STRIND	
• O CAMPO ID2 (MODO DE DADO OU TAG)	7
SETID2, ID2, SETLTAG, SETRTAG, RTAG, LTAG	
• CAMPOS DA CELULA NORMAL, ILUSTRACAO	
• ENDERECO DA CELULA	
• RECOMENDACOES, PROGRAMA PRINCIPAL = DECLARACOES E INICIALIZACAO CUIDADOS: CONVERSAO AUTOMATICA DO FORTRAN	8
CONT, INHALT, STRDIR, STRIND, INTGER	
• TIPOS DE CELULAS:	9
CELULA CONTEM DADO	
CELULA CONTEM NOME-DE-LISTA	
CELULA E' CABECA-DE-LISTA (HEADER)	
CELULA E' LEITORA (READER)	
ILUSTRACOES.	
• DESCRICAO DO SISTEMA SLIP	10
DEFINICOES: LISTAS, CABECA-DE-LISTA (HEADER), CORPO-DE-LISTA, NOME-DE-LISTA, ALIAS	
A CABECA-DE-LISTA OU HEADER	
A LISTA VAZIA	
EXEMPLO DE LISTA SIMPLES	11

REPRESENTACAO GRAFICA E NOMENCLATURA
A) REPRESENTACAO VERTICAL
B) REPRESENTACAO HORIZONTAL

ESTRUTURA DE LISTAS

12

LISTA PRINCIPAL, SUBLISTA

EXEMPLO DE ESTRUTURA DE LISTAS

GARBAGE-COLLECTION: A RESPONSABILIDADE DO SISTEMA NA DEVOLUCAO DE CELULAS PARA A LISTA LIVRE 13

COMPARTILHAMENTO DE SUBLISTAS

O DADO NO CAMPO DE INFORMACAO

MECANISMO DE GARBAGE COLLECTION: LAVS-LISTA DE ESPACOS DISPONIVEIS

INICIALIZACAO DE LAVS: INITAS

FUNCOES QUE RETIRAM E DEVOLVEM CELULA A LISTA LIVRE:
NUCELL, RCELL

DEVOLUCAO DE LISTA PARA LAVS: "BULK-ERASURE" - MTLIST 14

DEVOLUCAO DE SUBLISTAS COMPARTILHADAS

O CONTADOR DE REFERENCIAS

INSERCAO DE DADOS NUMA LISTA : NXTLFT, NXTRGT

CONTROLE PELO PROGRAMADOR PARA EVITAR "DELECAO" DE LISTAS: PRESRV, RESTOR

CRIACAO DE LISTA VAZIA: LIST

15

AS LISTAS PUBLICAS, INICIALIZACAO: INITAS
ACESSO: COMMON AVSL, W(100)

RECURSAO:

O MECANISMO VISIT-TERM PARA POSSIBILITAR A RECURSAO

COMUNICACAO E EMPILHAMENTO DOS PARAMETROS: PARMIN, RESTOR

RESTRICOES NA RECURSAO: CICLO DE DOIS

16

PERCURSO DE ESTRUTURA DE LISTAS

O MECANISMO DA READER
DESCIDA E SUBIDA NA ESTRUTURA

A "HISTORIA" DO PERCURSO DE UMA ESTRUTURA DE LISTAS
ESTA' CONTIDA NUMA PILHA DE READER'S.

CRIACAO DE UMA READER: LRDR0V

A READER NO SEU ESTADO INICIAL

OPERACOES DE AVANCO

17

AVANÇO LINEAR
AVANÇO ESTRUTURAL

INICIALIZAÇÃO DE UMA READER: INITRD

CRITERIOS DE PARADA DE AVANÇO,

III - PROCESSOS	18
1. CRIAÇÃO DA LISTA DE ESPAÇOS DISPONÍVEIS (LAVS) E DAS LISTAS PÚBLICAS (W'S): INITAS	
EXEMPLO DE INICIALIZAÇÃO, ILUSTRAÇÃO	19
2. CRIAÇÃO DE LISTAS: LIST	20
O CONTADOR DE REFERÊNCIAS	
O NOME-DA-LISTA (ALIAS)	
EXEMPLO DE CRIAÇÃO DE LISTAS	
ILUSTRAÇÃO	21
3. COLOCAÇÃO DE INFORMAÇÃO NA LISTA	
EMPILHAR DADOS NO TOPO E NO FUNDO DA LISTA: NEWTOP, NEWBOT	
EXEMPLO	22
INSERIR INFORMAÇÃO NA LISTA: NXTLFT, NXTRGT	23
INSERIR LISTA À ESQUERDA E À DIREITA DE UMA CELULA: INLSTL, INLSTR	
SUBSTITUIR O CONTEÚDO DO CAMPO DE INFORMAÇÃO: SUBSTP, SUBSBT, SUBST	
4. O MECANISMO DA LEITORA - PERCURSO DE LISTAS E ESTRUTURAS DE LISTAS LRDRÖV, LPNTR, LCNTR, LOFRDR	24
OS AVANÇOS LINEARES ADVLWR, ADVLER, ADVLNR, ADVLWL, ADVLEL, ADVLNL	25
DESIGNAÇÃO OU CLASSIFICAÇÃO DE CELULAS PARA OPERAÇÕES DE AVANÇO	
OS AVANÇOS ESTRUTURAIS ADVSWR, ADVSER, ADVSNR, ADVSWL, ADVSEL, ADVSNL	26
EXEMPLOS	28
SUBIDA DE NÍVEL (PERCURSO REVERSO) LRLRVT, LVLRV1	29
INICIALIZAÇÃO DE UMA READER: INITRD	30
COPIA DE UMA READER: LRDRCP	
LEITURA DO DADO CONTIDO NA CELULA APONTADA POR UMA READER:	

APAGAR UMA READER: IRARDR	31
A LEITORA SEQUENCIAL - "SEQUENCE-READER"	
VANTAGENS	
CRIACAO DE UMA LEITORA SEQUENCIAL: SEQRDR	
AVANCO LINEAR NA ESTRUTURA DE LISTAS: SEQLR, SEQLL	
AVANCO ESTRUTURAL: SEQSR, SEQSL	
VALOR DO FLAG NUM AVANCO DA LEITORA SEQUENCIAL	32
5. OUTROS MEIOS DE RECUPERACAO DE INFORMACAO DE LISTAS	33
RECUPERAR DADO ARMAZENADO NA CELULA DE TOPO E DE FUNDO DA LISTA: TOP, BOT	
DESEMPILHAR A CELULA DE TOPO (FUNDO), RECUPERANDO O DADO: POPTOP, POPBOT	
APAGAR UMA CELULA DA LISTA, RECUPERANDO O DADO: DELETE	
CRIACAO DE UMA NOVA LISTA, DIVIDINDO UMA LISTA DADA (SPLITTING) NULSTR, NULSTL	
6. FUNCOES DE TESTE	34
TESTE DE FORMATO DE NOME-DE-LISTA: NAMTST	
TESTE SE UMA LISTA E' VAZIA: LISTMT	
TESTE DE IGUALDADE DE ESTRUTURAS DE LISTAS: LSTEQI	
TESTE SE DUAS ESTRUTURAS DE LISTA SAO REVERSAS: LSTRVR	
7. PROCESSOS MISTOS	34
ENDERECOS DE CELULAS: MADLFT, MADRGT, MADNTP, MADNBT	
ESVAZIAR UMA LISTA: MLIST	35
APAGAR UMA LISTA: IRALST	
COPIAR UMA LISTA DADA: LSSCPY	
CRIAR UMA ESTRUTURA DE LISTAS REVERSA DE OUTRA DADA: LSTREV	
TRANSMISSAO DE PARAMETROS PARA A RECURSAO=LISTAS PUBLICAS PRESRV, RESTOR, PARMT2, PARMTN	
RESTAURAR CELULA PARA LAVS E RETIRAR CELULA DE LAVS: RCELL, NUCELL	

MARCA DE LISTA: MRKLST, LSTMRK

36

OPERACOES COM LISTA DESCRITIVA (ATTRIBUTE=VALUE LIST)

- ATUALIZAR O VALOR DE UM ATRIBUTO OU INSERCAO DE UM ATRIBUTO E SEU VALOR NA LISTA DESCRITIVA
NEWVAL
- APAGAR UM ATRIBUTO E SEU VALOR DA LISTA DESCRITIVA:
NOAIVL
- RECUPERAR O VALOR DO ATRIBUTO DA LISTA DESCRITIVA: ITSVAL
- ESVAZIAR A LISTA DESCRITIVA DE UMA LISTA: MTDLST
- FORNECER O NOME-DE-LISTA DA LISTA DESCRITIVA DE UMA LISTA:
NAMEDL
- ASSOCIAR UMA LISTA DESCRITIVA A UMA LISTA: MAKEDL

8. RECURSAO:

38

- MECANISMO VISIT/TERM
- MECANISMO LNK/BACK

IV - OPERACOES LOGICAS: EQUAL, AND, OR, NOT

FUNCOES PARA EVITAR CONVERSOES AUTOMATICAS DE MODO:
INTGER, STRDIR, STRIND

MANIPULACAO DE BITS E CARACTERES:

39

CHARACTER ORIENTED

BIT-PATTERN ORIENTED

40

SQOUT, SQIN, SHIN, LANORM

ROTINAS DE LEITURA E IMPRESSAO:

40,5

IMPRESSAO DOS CAMPOS DE UMA CELULA:
IMPCELL, PRCELL

IMPRESSAO DO CONTEUDO DE UMA PALAVRA: IMPDIR, PRDIR

IMPRESSAO DE UMA PILHA: PRSTACK

IMPRESSAO DE UMA LISTA: IMPLIST, PRLIST, PRLSTS

LEITURA DE UMA LISTA PARENTIZADA: RDLSTA

IMPRESSAO DE UMA LISTA PARENTIZADA: OUTLIST, TYPLIST

IMPRESSAO DOS ELEMENTOS DE UMA LISTA: OUTELEM, TYPELEM

COMPACTACAO DE ALFANUMERICOS: STORE, STORET

V - CONSIDERACOES SOBRE O SISTEMA SLIP - VANTAGENS E DESVANTAGENS	41
, CONSIDERACOES SOBRE O ARTIGO ORIGINAL DE J. WEIZENBAUM	
, A IMPLEMENTACAO - CONSIDERACOES SOBRE ENDERECAMENTO	42
- ASPECTO DIDACTICO, ASPECTO PRATICO	
O GARBAGE-COLLECTOR, O PAGING-SLIP	
, CONSIDERACOES SOBRE A FORMA E CONTEUDO DESTES TRABALHOS.	
, COMENTARIOS SOBRE O SLIP: KNUTH, FOSTER, SAKODA, RAPHAEL	43
, VANTAGENS E DESVANTAGENS DO SLIP	44
, OUTRAS IMPLEMENTACOES DO SLIP	

VI - BIBLIOGRAFIA:	45
--------------------	----

BALL, BARBIERI, BARRON, BOBROW, COLLINS, DEC, FENICHEL, FOSTER, FRAGA, FURTADO, GELERTER, GOTLIEB, HOPGOOD, KNUTH, LAPIDUS, LUCENA, NEWELL, NOVAK, PERLIS, RAMANIS, RAPHAEL, WEEKS, WEIZENBAUM

A D E N D O S

VII - LISTAGENS DE SUBPROGRAMAS	50
, ROTINAS EM MACRO-10	
FUNCOES PRIMITIVAS: ID, MADOV, ID2, LNKR, LNKL, SETDIR,	
SETIND, SETID2, CONT, INHALT, INTGER, STRDIR, STRIND	
, RECURSAO: VISIT, TERM, LNK, BACK	53
, MANIPULACAO LOGICA: EQUAL, AND, OR, NOT	54
, SUBPROGRAMAS-FONTE FORTRAN IV	55
INITAS, NUCELL, RCELL, LIST, MLIST, IRALST, NULSTL,	56
NULSTR, NEWTOP, NEWBOT, NXLEFT, NXTRGT, INLSTL, INLSTR,	58
SUBST, SUBSTP, SUBSBT, TOP, BOT, POPTOP, POPBOT,	60
ADVLL, ADVLR, ADVSR, ADVSL,	61
ADVLNR, ADVLER, ADVLWR, ADVSNR, ADVSER, ADVSWR,	62
ADVNL, ADVLEL, ADVLWL, ADVSNL, ADVSEL, ADVSWL, REED,	63
DELETE,	64
LRDROV, LOFRDR, LCNTR, LPNTR, LVLRVT, LVLRV1,	64
INITRD, IRARDR, LRDRCP,	65
MADNTP, MADNBT, MADLFT, MADRGT,	66
LISTMT, LSTEQL, LSSCPY, LSTPRO, LPURGE,	67
PRESRV, RESTOR, LOCT, PARM2	68
RDLSTA, PRLSTS,	70
SEQLL, SEQLR, SEQSR, SEQSL, SEQRDR,	71
NOATVL, NEWVAL, ITSVAL, MIDLST, MAKEDL, NAMEDL, LISTAV,	72
MADATR, DERROR,	73
MRKLST, LSTMRK,	73
LSTREV, LSTRVR,	
SQOUT, SQIN, SHIN, LANORM,	75
BLOCKDATA,	76
IMPCELL, IMPDIR, IMPLIST, PRCELL, PRDIR, PRLIST, PRSTACK,	77
OUTLIST, STORE, OUFELEMENTOS, TYPLIST, STORET, TYPELEMENTOS,	
LIST3, LISTOF,	

VIII - TABELA DE REFERENCIA DE TODOS OS SIMBOLOS GLOBAIS EM ORDEM ALFABETICA (GLOB)	81
IX- INDICE DAS LISTAGENS DAS ROTINAS EM ORDEM ALFABETICA	84
X - GLOSSARIO (ORIENTADO PARA O CONTEXTO DO TRABALHO)	87
XI - APLICACOES:	102
• ALGUMAS LINGUAGENS DE MANIPULACAO DE SIMBOLO E PROGRAMAS DE APLICACAO	
• RESUMOS DE ALGUNS ARTIGOS QUE DESCREVEM APLICACOES DO SLIP	105
MANIPULACAO ALGEBRICA: LAPIDUS, GOLDSTEIN, GREENSPAN ALGEM; GOTLIEB, NOVAK OPERACOES EM ARVORES, APLICACAO EM LINGUISTICA: RAMANI ELIZA; WEIZENBAUM	106
• EXEMPLO DE UTILIZACAO DO DISPOSITIVO VISIT-TERM PARA RECURSAO (CALCULO DE FATORIAL DE NUMEROS INTEIROS POSITIVOS)	107
• EXEMPLO DE UTILIZACAO DO DISPOSITIVO LNK/BACK PARA RECURSAO (CALCULO DE FATORIAL DE NUMEROS INTEIROS POSITIVOS)	107,5