"UM MODELO DE SIMULAÇÃO PARA ALOCAÇÃO DE ARQUIVOS

DE UM BANCO DE DADOS TOTAL"

Martin Robert Rudolf Wieser

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENCÃO DO GRAU DE MESTRE EM CIÊNCIAS (M.Sc.).

Aprovada por:

Prof. Nelson Maculan Filho
(Presidente)

Prof.João Lizardo R.H.de Araujo

Prof. Sergio Ellery Girão Barroso

Prof. Jano Moreira de Souza

RIO DE JANEIRO,R.J.-BRASIL

SETEMBRO DE 1978

WIESER, MARTIN ROBERT RUDOLF

Um Modelo de Simulação para Alocação de Arquivos de um Banco de Dados TOTAL. Rio de Janeiro, 1977.

VI, p. 29,7cm (COPPE/UFRJ, M.Sc., Engenharia de Sistemas, 1977).

Tese - Universidade Federal do Rio de Janeiro. COPPE.

1. Simulação I. COPPE/UFRJ II. Título (série).

AGRADECIMENTOS

Em primeiro lugar agradeço sinceramente a Jacyra, minha amiga e companheira, cujo estímulo foi certamente o fator primordial para que este trabalho tivesse chegado a bom termo.

A seguir meus agradecimentos a todos aqueles que de alguma forma participaram para que este objetivo tivesse sido alcançado.

Por absoluta falta de espaço, limito-me a citar apenas alguns, que não foram, de forma alguma, os únicos:

- Marcelo Pardo Brown, Nelson Maculan Filho e Sergio E. Girão Barroso, meus orientadores, por seu valioso trabalho de orientação.
- Suzana Vaisman Pinto, por sua expontânea e preciosa contribuição técnica.
- Marília Gama Canetti e Sheila Gomes, pela incansável ajuda na parte de pesquisa bibliográfica.
- Marcia Brito Magnan, responsável pelo perfeito aspecto datilográfico do presente trabalho.

RESUMO

O presente trabalho apresenta um modelo de simulação, destinado a fornecer parâmetros que sirvam de elemento de diferenciação entre diversos esquemas propostos para alocação de arquivos de uma rede TOTAL de Banco de Dados.

Os parâmetros de diferenciação visam aquilatar o deslocamento físico dos braços dos diversos volumes de disco, contendo o Banco de Dados.

Os parâmetros são obtidos simulando o funcionamento do TOTAL sobre o arquivo log por ele fornecido.

Para que se possa obter idéia de distorções localizadas, os parâmetros são abertos por task, buffer e versão, arquivo e volume.

ABSTRACT

This work presents a simulation model, which has the purpose of giving parameters that will serve as differentiation element between several proposed schemes for file allocations in a TOTAL data base.

The differentiation parameters pretend evaluate the physical displacements of the arms of the DASD's containing the data base.

They are obtained by simulation of TOTAL's actions over the log file provided by the software.

By means of having an idea of local distortions, these parameters are givin by task, buffer version, file and volume.

ÍNDICE

I - INTRODUÇÃO

I.1 - Definição de Banco de Dados

Dada a grande expansão que experimenta atualmente a computação comercial, frequentemente, e cada vez mais, encontra-se referência ao conceito de Banco de Dados. Depois de alguma procura, talvez a melhor definição seja a que se segue:

> *Banco de Dados é uma coleção de dados interrelacionados, armazenados com um mínimo ou sem redundância, visando atender a uma ou mais aplicações, de forma otimizada.*

Dada a forma bastante geral da definição, é fácil verificar que ela abrange os mais diversos tipos de arquivos (sejam eletrônicos ou não), com acesso ou busca à informação por meios os mais variados (manuais, mecânicos ou eletrônicos). Para o alcance do presente trabalho porém, estaremos só interessados em arquivos e manipulação de informação no âmbito exclusivo do processamento eletrônico de dados, em equipamentos comerciais de computação.

Visto isto, vamos analisar detalhadamente o conceito visto, para compreendermos exatamente o seu alcance.

I.2 - Análise da Definição de Banco de Dados

> *Banco de Dados é uma coleção de dados interrelacionados... .*

Entende-se que os dados, as informações, ou seja, o que vai estar no Banco de Dados propriamente dito, deve ter algum grau de relacionamento entre si, de modo a formar uma certa unidade. E então surgem naturalmente, Bancos de Dados de Material, de Pessoal, de Contabilidade, de Faturamento, etc.

... armazenados com um mínimo ou sem redundância, ...

Um dos maiores problemas que surgem em sistemas de médio ou grande porte é o da redundância, ou seja, a repetição da informação em dois ou mais arquivos do sistema. Certamente nenhuma sistemática de armazenamento de dados será de utilidade se a redundância não for reduzida a um mínimo, ou simplesmente eliminada.

... visando atender a uma ou mais aplicações ... .

Certamente um Banco de Dados deve permitir o uso de suas informações por mais de um programa simultaneamente. Ou seja, as solicitações de diversos serviços concorrentes são atendidas sem que um tome conhecimento do outro. Evidentemente é normal se estabelecer algum tipo de resguardo, de modo que um serviço não seja prejudicado por outro ou o prejudique. Mas essa segurança deve ser estabelecida fora do Banco de Dados, como um complemento ao mesmo.

... de forma otimizada.

Em processamento eletrônico de dados estão consagrados três tipos de organização de arquivos: sequencial, indexado-sequencial e acesso direto.

Dos três, este último tipo é o mais complexo e que requer mão-de-obra de programação mais requintada, sendo porém, em contrapartida, o mais eficiente.

Todos os Bancos de Dados, em processamento eletrônico de dados, dispõem de sistemáticas padronizadas visando a utilização desse tipo de organização (acesso direto).

II - BANCO DE DADOS

II.2 - Características Gerais de um Banco de Dados

- Os dados devem independer dos programas que os acessam.
- A manipulação da informação deve permitir todo tipo de operações, ou seja, adição, modificação e deleção.
- O acesso físico às informações não é feito diretamente por cada programa, e sim através de uma sistemática padronizada e única, comum a todos eles. Esta sistemática é conhecida como Sistema de Administração do Banco de Dados (SABD).
- Estruturalmente podemos encontrar dois tipos de Bancos de Dados:
  - . de estrutura hierárquica
  - . estrutura em rede
- Num Banco de Dados com estrutura hierárquica, cada arquivo é associado a um nível lógico em relação aos outros arquivos. Nesse tipo de estrutura, o relacionamento entre registros de diferentes arquivos se faz através de arquivos auxiliares, de dois tipos:
  - . arquivos de referência cruzada
  - . arquivos de vínculos

Conforme o número de arquivos de informações cresça e os inter relacionamentos entre eles venham a ficar mais complexos, as necessidades de arquivos auxiliares tendem a crescer exponencialmente.

- Um Banco de Dados com estrutura em rede tem por premissa básica o fato de que somente quando um arquivo tem um relacionamento lógico com outro é que se estabelece um vínculo entre eles. Não há necessidade de arquivos auxiliares como no caso de estrutura hierárquica, sendo eliminados os inconvenientes destes.

II.2 - Objetivos de um Sistema de Administração de Banco de Dados

- O sistema deve permitir um acesso simples e padronizado aos dados.
- A segurança e a integridade dos dados no Banco de Dados devem ser absolutas.
- Deve permitir a expansão do Banco de Dados, sem destruir o conteúdo atual dos arquivos.
- Deve utilizar criteriosamente os recursos do sistema, principalmente memória e tempo de CPU.

Concluindo, existem no mercado inúmeros Sistemas de Administração de Banco de Dados, alguns exigindo uma estrutura hierárquica de arquivos, e a maioria trabalhando com uma estrutura em rede.

Na TELERJ é usado como Sistema de Administração um "software" conhecido como TOTAL, desenvolvido pela Cincom Systems, Inc., EUA.

O modelo desenvolvido no presente trabalho simula os Bancos de Dados TOTAL existentes na Empresa, visando obter parâmetros para uma política de alocação de arquivos nos diversos volumes de discos magnéticos.

III - TOTAL

TOTAL é um Sistema de Administração de Banco de Dados, trabalhando com os arquivos numa estrutura em rede.

TOTAL não possui uma linguagem de programação completa, ele é um hóspede na linguagem do programa do programa de aplicação. Explicando: este último é escrito em uma linguagem hospedeira qualquer (PL/I, COBOL, ASSEMBLER) e quando há necessidade de acesso aos dados dos arquivos do Banco de Dados, o TOTAL é invo

cado através de um comando do tipo:

CALL DATBAS (coleção de parâmetros);

(com variantes de sintaxe do comando decorrentes unicamente da linguagem hospedeira) e, executa a função desejada.

É portanto o programa de aplicação quem ativa o TOTAL, e este então executa inclusões, exclusões ou modificações de informações armazenadas no Banco de Dados.

III.1 - Arquivos de uma Rede TOTAL

O TOTAL enquadra os arquivos pertencentes a rede um Banco de Dados em um dos dois tipos abaixo:

- arquivos mestre
- arquivos variável

No linguajar habitual os dois tipos são chamados de mestre e variável, simplesmente. E é dessa forma simplificada que serão referidos no presente trabalho.

III.2 - Características de Mestre e Conceito de Chave

Mestres são arquivos nos quais os registros são recuperados por acesso direto. Assim sendo, associado a cada mestre, em cada registro existe um campo de tamanho definido, tamanho esse constante e padronizado para todo o mestre, chamado de chave. Existirão tantos registros com informações no mestre quantas forem as chaves distintas introduzidas no arquivo. A partir do conteúdo da chave, o TOTAL obtém, por um processo interno transparente ao usuário, um endereço que será o do registro correspondente. O processo consiste em randomizar a chave, ou seja, obter através de um algoritmo de hashing (1), um número aleatório, posteriormente ajustado ao tamanho do arquivo e usado diretamente como endereço relativo do registro. O termo

endereço relativo dá noção de ordem: se o resultado da randomização da chave nos fornecer o valor 83 por exemplo, o registro correspondente ocupará a 83a. posição contada a partir do início físico do arquivo.

O acesso direto tem pois a vantagem de permitir que cada registro do mestre, desde que se conheça sua chave, possa ser acessado independente e diretamente.

Pelo exposto acima é fácil verificar que, em termos de ocupação, o mestre tem "claros" (posições em branco) entre seus registros, e estes se espalham por toda extensão do arquivo. Conforme o número de registros for crescendo (por adição de novas chaves) o número e tamanho desses "claros" irá diminuindo, até, teoricamente, desaparecer quando o arquivo estiver cheio. Dizemos "teoricamente" por causa de um fenômeno conhecido por colisão que será estudado mais adiante, e que torna o trabalho com mestres muito densamente carregados e ante-econômico, pois o acesso aos registros se torna lento e ineficiente.

III.3 - Campo de Vínculo no Mestre

Além do campo de chave, obrigatório como vimos acima, cada registro de um mestre terá um campo de vínculo para cada variável a que ele (mestre) esteja ligado. Se o mestre existir independentemente, seus registros não terão nenhum campo de vínculo.

Para entendermos a necessidade do campo de vínculo no mestre, precisamos apresentar primeiro o conceito de cadeia de um variável. Para apoiar a explanação, vamos imaginar um exemplo. Suponhamos que se tenha um mestre de um Banco de Dados de Material, contendo itens de material, seu nome padronizado, sua descrição e sua unidade de estoque. A chave seria o número de estoque do material. Imaginemos agora a existência de outro mestre do mesmo Banco, contendo informações de fornecedores de itens de material, como por exemplo, a razão social do fornece

dor, o endereço e o CGC. A chave aqui seria uma sigla vinculada a cada fornecedor, um código de fornecedor.

Esses dois mestres poderiam existir separadamente, mas certamente seria de grande interesse se se pudesse fazer um cruzamento dessas informações:

- que fornecedores estão capacitados a fornecer determinado item de material ?
- que itens de material um determinado fornecedor pode fornecer ?

Esse cruzamento seria feito exatamente pela existência de um variável ligado aos dois mestres em questão. Esse variável teria tantos registros quantos fossem os pares item de material X fornecedor existentes.

O conceito de cadeia agora aparece claramente:

> *Cadeia de um variável em relação a determinado mestre é a coleção de registros desse variável que tem como elemento comum, o mesmo valor de chave do referido mestre.*

Assim sendo, todos os registros do variável referenciando o mesmo tiem de material, constituem uma cadeia daquele item. Como o variável em questão está vinculado também ao mestre de fornecedores, estabelecem-se cadeias de registros referentes, cada uma delas, a um código de fornecedor.

E então podemos desvendar a necessidade de um campo de vínculo no mestre, para cada variável a que ele esteja ligado. Prende-se ao próprio acesso aos registros do variável.

O variável, diferentemente do mestre, tem os seus registros gravados convencionalmente, numa sequência normal desde o início físico do arquivo (adições de novos registros num variável, a menos que existam "buracos" deixados por deleções anteriores, se fazem a partir do último registro gravado. Ou seja, um tratamento idêntico ao DISP=MOD do JCL). Consequentemente, o ende

reço relativo do registro num variável nada tem a ver com a chave (ou chaves) a que ele se refere. O que se faz para recuperar esta informação é guardar no mestre, nos registros de cada chave, no campo de vínculo, os endereços relativos do início e do fim das cadeias referentes a cada chave, no variável. Então para se iniciar a leitura de qualquer cadeia num dado variável, de posse da chave faz-se necessário primeiramente um acesso ao mestre, no registro respectivo, para descobrir os endereços relativos no variável de:

- início da cadeia, se a leitura da mesma deve ser conduzida do início para o fim.
- fim da cadeia, se a leitura deve ser processada de modo reverso, do fim para o início.

Feita a leitura inicial na cadeia, as subsequentes já dispensam o acesso ao mestre (mesmo porque seria inútil, visto não haver lá nenhuma informação adicional a esse respeito): os endereços relativos dos registros seguintes da cadeia são recuperados no último registro lido, no campo de vínculo do variável. Mas esse procedimento será explicado mais adiante.

III.4 - Campo de Raiz no Mestre

Todo registro de mestre tem como primeiro campo o campo de raiz (no release atual, 6.2, é obrigatório o root ser o primeiro campo, como também o campo de chave deve vir logo a seguir. Vínculos e dados é que podem estar em qualquer posição, respeitadas as duas primeiras). Para entendermos a necessidade do campo de raiz, temos que analisar antes os conceitos de sinônimos e colisão.

Sinônimos são registros com diferentes chaves que, após terem sido randomizadas, apontam para o mesmo endereço relativo. Colisão vem a ser tentativa de gravar dois sinônimos ao mesmo endereço relativo.

Essas imperfeições decorrem de deficiências no algoritmo gerador de endereços relativos. Quando se analisa o mecanismo interno desses algoritmos, torna-se claro que existem duas características importantes que são antagônicas: simplicidade e rapidez X número de sinônimos gerados. Explicando: geradores simples e rápidos (e consequentemente baratos e econômicos) tendem a gerar um número grande de sinônimos, ao contrário de algoritmos mais sofisticados, mas então lentos e caros.

Existem mesmo alguns algoritmos propostos que geram zero sinônimos, mas são de tal forma elaborados que se tornam lentos, consequentemente caros e inviáveis. O que se adota na prática é uma solução de compromisso: um algoritmo barato (ou seja, rápido) com um número de sinônimos pequenos. Atualmente o algoritmo que melhor atende a essas características é o da divisão (1), sendo por isso o mais usado.

O campo de raiz serve exatamente para o encadeamento de sinônimos. É um campo de oito bytes, com duas metades de quatro bytes cada. A primeira metade indica o endereço relativo do sinônimo anterior (na cadeia de sinônimos), a segunda metade, o do sinônimo posterior na cadeia. Vamos entender bem como os fatos se passam. Suponhamos que se esteja adicionando registros a um mestre, e que numa das adições já se encontre o lugar ocupado por outro registro. Supondo que houve uma colisão isto é, que os dois registros sejam sinônimos (mais adiante será explicado como pode acontecer colisão de registros não sinônimos), o registro já gravado permanece no lugar (o endereço relativo de um registro de mestre é também chamado de home-address) ao passo que o novo, que se tentou gravar neste mesmo endereço, será gravado no primeiro lugar vago dentro do bloco físico em questão (se isso for possível, visando otimizar o acesso físico ao disco) ou em um bloco anexo. E assim está iniciada uma cadeia de sinônimos no mestre. Os campos de raiz de cada registro da cadeia fazem o encadeamento dos mesmos, apontando respectivamente o anterior e o posterior.

É fácil verificar que em cada cadeia de sinônimos, só o primeiro da cadeia está no seu home-address. Os outros todos estarão como "intrusos" em endereços relativos que nada tem a ver com o seu home-address. Esse fato propicia eventualmente uma colisão entre dois registros não sinônimos. Esta ocorre quando o home-address de um registro que vai ser adicionado está ocupado por um registro de uma cadeia de sinônimos, (mas não sinônimos do registro que vai ser adicionado) e este último não é o primeiro da cadeia. Quando tal ocorre, o registro "intruso" é deslocado para o primeiro lugar vago disponível, e o dono do home-address ocupa o lugar que lhe é devido. E o TOTAL, automaticamente, faz a manutenção dos endereços no campo de raiz dos registros da cadeia, visto que um elemento da mesma foi deslocado.

Toda manutenção do campo de raiz é automática, como visto, e totalmente transparente para o usuário.

A ocorrência de sinônimos depende de um parâmetro, que mede o grau de utilização do arquivo, chamado de fator de ocupação(ou também, fator de carga):

$$\% \text{ F.U} = \frac{\text{número de registros lógicos gravados}}{\text{número de registros lógicos totais possíveis}} \times 100$$

Demonstra-se que para F.U. de até 80%, o número médio de acessos por chave é essencialmente igual a 1, para um fator de bloco $\geq$ 5. Ver anexo 8.

III.5 - Características de Variável

Algumas características de variável já foram vistas no item III.3, anteriormente. Vamos ver agora algumas características adicionais, em especial o conceito de vínculo no variável.

Uma das perguntas mais comuns e mais importantes que se faz ao conceber um Banco de Dados TOTAL é:

- Quando usar um mestre ?
- Quando usar um variável ?

A resposta pode ser encontrada na diferença de implicações quando se faz manutenção estrutural num ou noutro tipo de arquivo.

Vamos entender: manutenção estrutural vem a ser todo tipo de operação que implique em manutenção de vínculos (e eventualmente também de campos de raiz). Estão nessa categoria os vários tipos de add's e delete's existentes, tanto para mestre como para variável. Em relação a esse tipo de operação, mestres e variáveis diferem na medida em que os primeiros podem existir independentemente ou vinculados aos segundos, mas estes nunca podem existir independentemente. Ou seja, se vamos eliminar um registro no mestre, de determinada chave, temos que explicitamente eliminar primeiro todos os registros de variáveis vinculados daquela chave. Por esse tipo de trabalho adicional com mestres, vemos que estes devem conter informações que não tenham uma grande volatilidade, reservando-se para os variáveis as informações de maior mobilidade.

III.6 - Campos de Chave e Vínculo no Variável

Todo registro de variável terá tantos pares de campos chave - vínculo quantos forem os mestres a que ele estiver relacionado. No release atual, 6.2, os campos terão que vir pareados, e nessa ordem: chave-vínculo. E ainda, deverão preceder os campos de dados.

Visto que um variável não pode existir independentemente (se não seria impossível descobrir a posição inicial e final da cadeia de determinada chave, a não ser pelo ineficiente método da leitura serial do arquivo), pelo menos um par chave-vínculo existirá em cada registro do variável.

Vamos interpretar os campos em questão. O campo chave contém um valor de chave do mestre a que ele, variável, está vinculado. O campo de vínculo contém endereços relativos (à semelhança dos campos de raiz e de vínculo do mestre) que permitem que se percorra a cadeia de registros vinculados àquela chave. É também um campo de 8 bytes, interpretado como duas metades de 4. A primeira tem o endereço relativo do anterior da cadeia, a segunda metade, o do posterior.

O TOTAL guarda esses endereços na forma binária, e reserva para fim-de-cadeia a configuração "ØØØØ" (em hexadecimal: "40404040").

O caminho percorrido numa cadeia, seguindo os vínculos do variável, é chamado de "linkage path". Fica óbvio que se um variável está preso a dois mestres, existirão, para cada registro deste variável, dois linkage path's passíveis de serem percorridos, dependendo da chave de mestre com a qual se inicie a pesquisa.

III.7 - Análise do Relacionamento Mestre-Variável

Para completar a conceituação de mestres e variáveis, seus campos de chave e vínculo, e como se estabelece o relacionamento entre eles, vamos nos reportar ao anexo 1, que exemplifica a situação retratada no item III.3, envolvendo dois mestres (um de itens de material, outro de fornecedores) e um variável de relacionamento.

Vemos inicialmente que para os campos de vínculo é usada a palavra "link", bastante comum no trato com Banco de Dados. Além disso, por estar relacionado com dois mestres, o variável tem dois pares de campos chave-vínculo. Como não entramos no mérito do encadeamento de sinônimos nos mestres, foram suprimidos, por comodidade, os campos de raiz em ambos os arquivos. Também para os mestres não foram explicitados os endereços relativos dos seus registros, havendo um número variável

de lacunas entre eles. Deve-se observar somente que as chaves estão apresentadas por ordem crescente de endereço relativo e, consequentemente, não obedecem nos mestres a nenhuma ordem numérica ou alfabética.

Antes de prosseguir, vamos apresentar o conceito de RQLOC. Em verdade, RQLOC é a função do TOTAL pela qual é fornecida ao próprio uma chave de um mestre e ele então retorna o endereço relativo da mesma. Mas o uso comum fez confundir a função com o próprio endereço relativo. Então, a menos de ressalva explícita, quando falarmos em RQLOC estaremos nos referindo ao endereço relativo propriamente dito.

O anexo 1 mostra alguns fatos interessantes, relativos à carga e manutenção de um variável.

Carga de um variável vem a ser a operação de gravar um a um todos os registros que inicialmente o arquivo deve conter.

Manutenção de um variável vem a ser a operação de, periodicamente, alterar selecionadas informações nele contidas. Essas alterações, se forem ao nível de adições e deleções, recebem o nome de manutenção estrutural (conceito já visto no item III.5) diferentemente de quando forem simples gravações (alteração de informações em registros já existentes).

Normalmente, na carga de um variável, as cadeias são gravadas de modo a formarem um conjunto conexo (isto é, sem "buracos"), e mais, na ordem ascendente do RQLOC da chave (supondo estar o variável vinculado a um mestre somente).

Se for um variável ligado a mais de um mestre, como é o nosso caso, temos que escolher uma das chaves para ser a chave de carga, ou seja, a chave pela qual as cadeias do variável estarão ordenados por RQLOC ascendente. É fácil verificar que para chave de carga do variável em questão foi escolhida a chave do mestre de itens de material.

Ainda em relação à carga de um variável, podemos classificar os registros de uma mesma cadeia, segundo um critério qualquer e carregar o arquivo com essa característica. No caso de variáveis ligados a dois mestres, normalmente o critério é classificar, dentro das cadeias da chave de carga, por ordem ascendente da chave não de carga. Podemos verificar que também isso foi adotado no nosso variável.

Resta então analisarmos o variável quanto às modificações do tipo manutenção estrutural que tenham oco-rido (a manutenção não estrutural não será abordada aqui visto que só afeta campos de dados do arquivo e estes não são relevantes nesta análise).

Antes dessa apreciação vamos esclarecer o critério que o TOTAL adota quando da adição de registros em cadeias de variáveis de pelo menos dois caminhos de vínculos.

Se adicionarmos um registro numa cadeia de uma determinada chave de carga, a posição desse registro na cadeia desta chave já estará determinada pelo critério de classificação da chave não de carga. (Atenção! Esse controle não é do TOTAL e sim de quem está procedendo à manutenção!). Agora, em que posição será colocado o registro na cadeia não de carga (ou seja, a não controlada)? Aí a decisão é do TOTAL e é a seguinte: na cadeia da chave não controlada o registro é sempre adicionado ao final da mesma. Resumindo: sempre que temos adição de registro num variável preso a dois (ou mais) mestres, na cadeia da chave de carga a inclusão se fará sob controle do usuário e na posição que lhe convier; já na cadeia não de carga a adição se fará sempre ao final da mesma e sob controle do TOTAL.

Devemos notar que, apesar de no parágrafo anterior, temos referido sempre à chave de carga como a que está sendo controlada (é por ela que se está seguindo as cadeias do variável), porque esta é a situação mais comum, nada impede que o controle se faça numa chave não de carga, ficando a de carga (e as eventuais outras não de carga) submetidas ao critério de "adições ao fim da cadeia" visto.

Isto esclarecido, podemos analisar as cadeias que aparentemente sofreram manutenção estrutural.

. Chave 501 - Nada indica que tenha sofrido manutenção.

. Chave 123 - Inicialmente devem ter existido 5 registros na cadeia e, posteriormente um foi deletado (o do endereço relativo 42).

. Chave 913 - É um caso interessante de manutenção. Supondo que seja verdade a manutenção descrita para a chave 123, então havia inicialmente só um registro na chave 913 (o endereço 44). Posteriormente houve necessidade de aumentar esta cadeia, com a inclusão do registro com chave não de carga EFG (a do já existente, como pode ser visto, era RST). Pelo critério de classificação da chave não de carga, EFG vem antes de RST. No momento da adição, o endereço 42 estava vago e foi utilizado. O campo LINK1 mostra como ficou a cadeia da chave de carga: o registro que entrou (EFG) inicia a cadeia e o que já estava a termina. E LINK2, cadeia de cíncu - los da chave não de carga mostra o que foi dito anteriormente: EFG foi adicionado ao final de sua própria cadeia.

. Chave 739 - Nada a comentar. Registro único da cadeia. Conforme a análise da chave 700 pressupõem, ou essa cadeia (chave 739) tinha mais um registro (endereço 46) ou havia uma outra chave ali. O certo é que o endereço 46 não continha originalmente a chave 700, como atualmente está mostrado.

. Chave 700 - Originalmente só um registro, posteriormente adicionado mais um, que ocupou o endereço 46, vago naquele momento. Como podemos deduzir isto? Porque fisicamente a cadeia da chave 700 está invertida: ela cresce no sentido inverso dos endereços. Isto para que fosse respeitada a classificação da chave não de carga. Notar que, reforçando essa suposição (que o registro 46 veio posteriormente), a chave não de carga XYZ no endereço 46 está no final de sua própria cadeia.

. Chaves 487 e 108 - Situação semelhante ao que foi descrito para as chaves 123 e 913.

III.8 - Resumo do Relacionamento Mestre - Variável

Um mestre pode existir independentemente ou estar relacionado a um ou mais variáveis, mas não poderá nunca estar diretamente ligado a outro mestre.

Um variável não pode existir independentemente, devendo estar relacionado a pelo menos um mestre, mas nunca diretamente a outro variável.

III.9 - Número de Mestres Ligados por um só Variável

Não existe limitação para o número de variáveis que poderão estar ligados a um mesmo mestre. Mas, por causa da crescente complexidade na manutenção dos vínculos, não é recomendável que um só variável sirva de ligação entre vários mestres. O máximo viável é 3 mestres ligados por um só variável, sendo o ideal, 2.

III.10 - Redefinição de Registros de um Variável

Ao lado de outras características já vistas de arquivos variáveis, uma das mais interessantes é a possibilidade de redefinição dos registros (mestres não permitem essa facilidade).

Para mais facilmente acompanhar como funciona essa redefinição vamos analisar o exemplo do anexo 7 (as referências neste item se referem a este anexo).

O exemplo é de um variável de nome IOPS (todos os nomes de arquivos em TOTAL devem ter 4 dígitos, o primeiro necessariamente alfabético) que serve de referência cruzada entre 3 mestres: MSUB, MPGM, MEIO. (ref. 1).

O layout do variável propriamente dito tem uma definição toda ela contida entre as palavras-chave BASE-DATA e END-DATA (ref.2)

Por se tratar de um variável com redefinição, o layout obrigatoriamente começa com um campo de duas posições, conhecido por record code, que conterá um código numérico ou alfanumérico, característico de cada redefinição do registro. É chamado com a palavra chave xxxxCODE, onde xxxx é o nome do variável, no caso IOPS (ref. 3). Por exemplo, se o variável tivesse duas redefinições, cada registro receberia no campo de CODE os códigos 01 ou 02, conforme se tratasse de uma ou outra redefinição. No nosso caso real, os valores são alfabéticos: SU (de SUbsistema) e PG (de ProGrama). Obrigatoriamente, o registro tem que ter uma parte fixa, contendo um par chave-vínculo, e eventualmente dados. Aqui só temos a chave, que é a do mestre MEIO, codificada MEIOCTRL (padrão de nomes de chaves em mestres de TOTAL: xxxxCTRL, onde xxxx é o nome do mestre (convenção da TELERJ) e, CTRL é obrigatório) e, recebendo aqui o nome de IOPSMEIO (padrão de nomes de campos em geral no TOTAL (sugerido, mas não obrigatório): xxxxyyyy onde xxxx é o nome do arquivo e yyyy é o qualificador do campo, de codificação livre) de tamanho 9 (ref. 4), e o vínculo, codificado MEIOLKIO (padrão sugerido de nomes de vínculos em variáveis: xxxxLKyy onde xxxx é o mestre a cuja chave pertence o vínculo, LK é obrigatório, e yy é um qualificador do variável, livre. Sugere-se porém que se use para yy as duas primeiras posições do nome do variável, para efeitos mnemônicos, desde que com isso não surjam nomes de vínculos homônimos) (ref. 5). Depois é nomeada a parte que será redefinida, IOPSDADO, de tamnho 18 bytes (ref. 6).

Iniciando a primeira redefinição aparecem as palavras-chave RECORD-CODE e o nome dado a essa redefinição: SU (ref. 7). Seguem-se o par chave-vínculo da redefinição, no caso com relação ao mestre MSUB. Nesse mestre a chave é o código do subsistema (ref. 8). Seguem-se os campos de dados, IOPSTPSU e IOPSREST (ref. 9). Notar que no atual release 6.2, o tamanho dos campos nas redefinições tem que vir entre parenteses.

Uma estrutura em todo semelhante é apresentada para a outra redefinição: somente o mestre a que se referem o par chave-vínculo aqui é o MPGM. (ref. 10).

Não há necessidade de haver campos de vínculos nas áreas redefinidas. Mas se eles existem, vão permitir um acesso selecionado, a somente um tipo de registro (SU ou PG) conforme se siga um ou outro caminho de vínculos. Tal facilidade é conhecida como vínculos seletivos. Se, no entanto, desejarmos todas as ocorrências, seguimos o caminho de vínculos não seletivo, qual seja, o da parte fixa do registro, referente ao mestre MEIO.

III.11 - Mais de um Vínculo entre os mesmos Mestre e Variável

Normalmente envolvendo um mestre e um variável temos apenas um vínculo. Mas situações existem em que se fazem necessários mais de um vínculo (normalmente dois).

Suponhamos a seguinte situação: um mestre de datas (DDMMAA) e um variável com a informação da matrícula de empregados de uma empresa. Suponhamos que existam dois vínculos entre o mestre e o variável: um por data de admissão, outro por data de demissão. Cada registro do variável terá dois pares de campos chave-vínculo: um com a data de admissão, outro com a data de demissão, ao lado, é claro, da informação da matrícula do funcionário. Na verdade, tudo se passa como se o variável estivesse relacionando dois mestres distintos. Só que esses dois mestres estão fundidos num só. É evidente que o exemplo apresentado admite um terceiro vínculo (data de aniversário, por exemplo) e outros mais.

Existirão então no mestre tantos campos de vínculos quantos forem os relacionamentos com o variável, para que se possa acessar as respectivas cadeias neste último.

III.12 - Operações de Gravação

Sempre que houver uma operação de gravação (add's, write's ou delete's), o buffer, contendo a informação já pronta para ir para o arquivo, recebe uma marca de "write pendente", e a gravação fí

sica é simplesmente adiada.

Sempre que um buffer é liberado para receber novas informações, primeiramente é verificada a presença da marca de "write pendente" para o mesmo. Caso afirmativo, as informações atualmente existentes no buffer são efetivamente gravadas no arquivo a que elas se referem e só então se dá a liberação.

III.13 - Liberação de Buffers

Como vimos no item anterior, o TOTAL só traz informações do Banco de Dados para buffers que estejam liberados.

A política de liberação de buffers obedece ao seguinte critério:

- sempre que houver necessidade de se dipor de um buffer para receber novas informações de um arquivo, e não houver nenhum viável (ver item IV.1 mais adiante, para o conceito de viável) liberado, será escolhido dentre os buffers viáveis aquele que não tiver sido consultado a mais tempo. Este buffer, antes de sua liberação, poderá exigir uma operação física de E/S, se estiver com a marca de "write pendente" (ver item anterior).

O anexo 9 retrata, no seu fluxo, o procedimento descrito.

Eventualmente, o buffer escolhido para ser liberado pode conter uma informação retida (lida com 'END.' na lista de parâmetros , ver item VI.24, para uma explicação mais detalhada), para utilização futura.

Esta particularidade não é levada em conta pelo TOTAL, mas fica transparente para o usuário: quando essa utilização se tornar presente, o TOTAL trata de repor o registro liberado novamente em algum buffer viável, fazendo eventualmente uma nova pesquisa para liberação de buffers.

Quando estudarmos as funções do TOTAL (capítulo VI) veremos que além da liberação de buffers por necessidade de uso, existem operações físicas de E/S forçadas explicitamente por algumas funções. São elas:

- DEQUE, QUIET e ENDTO - todos os buffers com "write pendente" são gravados.
- CLOSP - todos os buffers viáveis do arquivo que está sendo fechado são gravados (só os que estiverem com a marca de "write pendente", é claro).

IV - DEFINIÇÃO DE ARQUIVOS

O TOTAL dispõe de uma linguagem toda própria e especial para a definição dos arquivos que ele manipula, a DBDL (Data Base Definition Language).

Alguma coisa desta linguagem já foi adiantada quando da análise da redefinição de registros de um variável, item III.10, mas cabe agora uma apresentação formal.

IV.1 - O Banco de Dados

Iniciando a definição do Banco de Dados, existe uma coleção de comandos, chamados prólogos de geração.

Vamos nos referir ao anexo 3, para explanação. De início, as palavras-chave... BEGIN-DATA-BASE-GENERATION se fazem necessárias para informar ao TOTAL que os comandos que se seguem vão definir um Banco de Dados.

Segue-se o nome que o Banco de Dados recebeu: CTBASE. Este nome é importante para alguns utilitários do TOTAL (como por exemplo o formatador do BD).

A opção de versão é para se escolher se o TOTAL vai funcionar batch somente, batch e multitask ou se ainda vai usar ENVIRON/1

(monitor de TP desenvolvido pela CINCOM). Na TELERJ o TOTAL é usado em versão central, isto é, só uma cópia do software está na memória e serve a todos os programas.

Para terminar há a indicação de SHARE-IO.

Como o nome indica, tem a ver com compartilhamento de áreas de buffer. Para o TOTAL todos os buffers de E/S tem nome, e mais, permite-se um fator de multiplicação n para cada buffer. No caso específico: o Banco de Dados reconhece três buffers distintos: MAS1, MAT1 e VAR1. E existirão quatro buffers MAS1, 1 MAT1 e 6 VAR1. Como veremos na definição de mestres e variáveis, o comando IOAREA=xxxx (desde que SHARE-IO tenha sido definido aqui no prólogo) aparece obrigatoriamente, apontando algum nome de buffer reconhecível. E esse buffer é partilhado (em suas n versões) por todos os arquivos em cuja definição o comando IOAREA aponte para ele.

Então introduzimos o conceito de buffer viável: para um determinado arquivo é o buffer cujo nome aparece no comando IOAREA , sendo o único buffer em que poderão ser feitas operações de E/S com dados daquele arquivo (respeitadas, evidentemente, suas n versões).

IV.2 - Definição de um Mestre

Examinemos o anexo 4, que mostra a definição do mestre de itens de material, MIDE.

Toda a definição de um mestre estará contida entre as palavras-chave BEGIN-MASTER-DATA-SET e END-MASTER-DATA-SET.

Inicialmente temos o enunciado do nome do arquivo (DATA-SET-NAME) e do nome do buffer que ele usa (IOAREA). Então este arquivo, MIDE, nas suas operações de E/S usará sempre qualquer das 4 versões do buffer MAS1, como visto no prólogo.

As palavras chaves MASTER-DATA e END-DATA emolduram todo o layout do mestre.

Iniciando o layout, o obrigatório campo de raiz, chamado mmmmROOT. O TOTAL sugere que mmmm seja o nome do mestre, para efeitos mnemônicos. Tamanho em bytes: 8, obrigatoriamente.

Segue-se o campo de chave, único no mestre: mmmmCTRL é o seu nome, e a mesma sugestão acima.

Logo a seguir devem aparecer todos os campos de vínculos, um para cada variável ao qual o mestre esteja ligado. O padrão de nome para estes campos é mmmmLKnn, onde o TOTAL sugere para mmmm o nome do mestre e nn as duas iniciais do variável. Deve-se aqui cuidar para não haver nomes de vínculos idênticos num mesmo mestre para diferentes variáveis. Tamanho do campo: 8 bytes, obrigatoriamente.

A indicação do nome do variável, após o tamanho do campo de vínculo, é opcional e é tratada pelo TOTAL como comentário.

Terminada a apresentação dos vínculos (notar que não há nenhuma ordem pré-determinada para esta apresentação) vem os campos de dados, cujos nomes são de formação absolutamente livre mas há a sugestão de se usar a forma mnemônica mmmmyyyy onde mmmm é o nome do mestre e yyyy é livre.

Uma característica de todos os nomes de campos nos mestres (e também nos variáveis) é seu tamanho: sempre 8 caracteres alfanuméricos, o primeiro necessariamente alfabético.

Outra característica de todos os campos de mestres e variáveis: seu tamanho é sempre dado em bytes , qualquer que seja o formato dos dados neles contidos. Esse tamanho, como visto, sempre segue imediatamente o nome do campo.

O TOTAL necessita ainda algumas informações sobre o arquivo, do tipo DCB e outras. Algumas são normalmente definidas por quem cria o Banco de Dados, as restantes são calculadas pelo TOTAL .

Sempre que este entender que a informação dada não é a melhor, ele utiliza a calculada.

Então, ainda o Anexo 4:

- DEVICE =, normalmente 3330, se não informado o defaut é 2314.

- TOTAL-LOGICAL-RECORDS =, este comando e o TOTAL-TRACKS são mutuamente excludentes. Se um é incluido, o outro pode ser omitido. Normalmente os cálculos serão feitos usando o TOTAL-TRACKS e o TOTAL-LOGICAL-RECORDS será ignorado. Se nenhum dos dois for explicitado, o defaut é uma trilha. Se só TOTAL-LOGICAL-RECORDS for informado, a partir dele TOTAL-TRACKS é calculado, recalculando-se depois o valor informado.

- TOTAL-TRACKS =, ver comentário sobre TOTAL-LOGICAL-RECORDS.

- LOGICAL-RECORD-LENGTH =, é um comando desnecessário e é tratado como comentário. Prevalece o valor calculado pelo TOTAL através da soma dos tamanhos dos campos descritos no lay-out.

- LOGICAL-BLOCKS-PER-TRACK =, informa o valor que o TOTAL vai usar para calcular o fator de bloco e o tamanho do buffer. Com isso a informação de LOGICAL-RECORDS-PER-BLOCK, se dada, é tratada como comentário. Se for omitida a informação de LOGICAL-BLOCKS-PER-TRACK, o defaut é 1 bloco.

IV.3 - Definição de um variável

A definição de um variável é bastante semelhante a do mestre , com algumas diferenças que serão destacadas. O anexo 6 mostra o variável FOID que serve de relacionamento entre o mestre MIDE analisado anteriormente e o mestre de fornecedores, MFOR, retratado no Anexo 5.

Toda definição de um variável estará contida entre as palavras-chave BEGIN-VARIABLE-ENTRY-DATA-SET e END-VARIABLE-ENTRY-DATA-SET.

O nome do arquivo e do seu baffer viável se seguem. Notar que, pelo prólogo, a IOAREA VAR1 tem 6 cópias homônimas, qualquer uma delas podendo ser usada pelo variável FOID para suas operações de E/S.

O layout do variável estará todo contido entre as palavras chaves BASE-DATA e END-DATA.

Seguem-se os campos de informação do variável. O TOTAL deixa em aberto a codificação desses campos, III.10). A exceção é o campo de vínculo, que obrigatoriamente tem que ter as letras LK nas 5a. e 6a. posições do nome, e mais importante ainda, TEM QUE SER IGUAL AO NOME QUE O VÍNCULO TEM NO MESTRE RESPECTIVO. E é dessa forma que o TOTAL descobre, no variável, o par ou pares de campos chave-vínculo: a chave por defaut é o campo que precede o vínculo. E ambos devem vir no início do layout (limitação do release 6.2). O mestre a que se refere o par chave-vínculo está determinado pelo nome do vínculo,como visto. Como o FOID está vinculado a dois mestres, apresenta dois pares de campos chave-vínculo: um para o MIDE, outro para o MFOR.

As informações extra-layout (tamanho do arquivo, tamanho do registro lógico, fator de bloco, etc.) no variável são de todo idênticas às de um mestre, exceto que naquele temos o CILYNDER-LOAD-LIMIT, que vem a ser o limite superior de carga para cada cilindro componente do variável. Explicando melhor: um variável sempre tem sua carga crescendo do início para o fim do arquivo, suavemente (ou o mais suavemente possível). Sempre que o limite de carga de um cilindro é atingido, as novas cadeias começam a ser carregadas no início do cilindro seguinte. Somente prolongações de cadeias já existentes poderão invadir a área do cilindro acima do seu limite de carga. E assim os fatos vão se sucedendo, cilindro a cilindro. Até que em algum instante, o variável estará carregado, que TODOS os seus cilindros terão atingido o limite de carga. Deste momento em diante, qualquer gravação de novos registros em cadeias já existentes , provocará, por ocasião do CLOSE do arquivo variável em questão , uma mensagem de alerta (veremos mais adiante que é devolvido o STATUS "LOAD"), visando lembrar ao usuário que o arquivo já está

próximo de sua saturação. Se o CILYNDER-LOAD-LIMIT não for especificado, o TOTAL assume como defaut o valor de 80%.

Para terminar a apreciação de variáveis, restaria descrever a redefinição de variáveis. Mas tal já foi feito, na análise do anexo 7, item III.10.

V - O COMANDO DATBAS

Como visto no início da apreciação do TOTAL, item III, este é invocado, nos programas de aplicação, através de um comando do tipo:

CALL DATBAS (parâmetros);

OBS.: O comando está apresentado na sintaxe do PL/I. As variantes decorrerão somente, em nível de sintaxe quando da troca da linguagem de programação usada.

A análise dos parâmetros está contida no que é conhecida pelo TOTAL como DML (Data Management Language).

A lista completa de parâmetros é a que se segue, se bem que só os dois primeiros e o último aparecem obrigatoriamente em todos os comandos. Os demais aparecerão ou não, dependendo da função utilizada.

É importante lembrar que para o TOTAL a lista de parâmetros não é uma coleção de valores diretamente utilizáveis e sim, uma coleção de endereços onde, aí sim, estão os valores utilizáveis.

Eventualmente, por um vício de expressão, poderão aparecer, no presente trabalho, referências do tipo:

"... o valor do campo tal...", que deverão, pela conceituação vista acima, ser interpretadas como:

"... o valor apontado pelo campo tal...". Campo aqui, seria qualquer um da lista de parâmetros.

V.1 - Lista de Parâmetros

Função, status, arquivo, referência (ou refer), linkage path (ou link ou caminho de vínculos), chave, lista de dados (ou data-list), área de dados (ou data-area), dbmod, task, end.

V.2 - Função

Este parâmetro aponta para um campo de cinco posições, definido pelo usuário, no qual está codificada uma das funções abaixo:

- ADD-M, ADDVA, ADDVB, ADDVC, ADDVR, CLOSM, CLOSV, CLOSP, DEL-M, DELVD, DEQUE, ENDTO, ENDLG, FREEF, FREEX, LOGTR, MARKL, OPENM, OPENV, QMARK, QUEST, QUIET, READD, READM, READR, READV, RESTM, RESTV, RQLOC, SERVL, SEQRM, SEQRV, SEQWV, TOTAL, WRITD, WRITM, WRITV.

V.3 - Status

Este parâmetro aponta para um campo de quatro posições, definido pelo usuário, no qual o TOTAL retorna um código alfabético (na imensa maioria dos casos) indicativo do resultado da função solicitada. Alguns status mais comuns:

**** - comando bem sucedido
END. - fim de leitura serial
*PNR - provável erro de endereçamento
FNOP - arquivo não aberto
FULL - arquivo com capacidade esgotada
IEOJ - fim-de-serviço inválido
IPAR - lista de parâmetros inválida
LOAD - carga do variável atingiu limite pré-estabelecido
MRNF - chave não existente no mestre
NACT - não foi ativada a task, para o TOTAL
NHLD - registro não retido no buffer
NOTO - TOTAL não ativo

SNTF - erro na lista de dados

Outros códigos de status, se citados, serão explicados no decorrer do texto.

OBS.: O campo de status deve ser testado pelo programa de aplicação, após cada apuração de TOTAL!

V.4 - Arquivo

Este parâmetro aponta para um campo de quatro posições, definido pelo usuário, que contém o nome do arquivo no qual o TOTAL vai exercer a função.

V.5 - Referência ou Refer

Este parâmetro aponta para um campo de quatro posições, definido pelo usuário, no qual podemos encontrar, normalmente:

- LKxx, posto pelo usuário, antes da execução de um comando de leitura de início de cadeia num variável. Indica que o registro do início ou do fim da cadeia (lembrar que a leitura de uma cadeia de um variável pode ser feita do início para o fim ou no sentido contrário. O sentido será indicado pela função usada) de um variável deve ser lido. LKxx obrigatoriamente deve ser as quatro últimas posições do nome do campo de vínculo no variável.
- rrrr, um endereço relativo em binário, posto pelo TOTAL, durante a leitura de uma cadeia de um variável. Contém o endereço relativo do último registro lido na cadeia e indica que o próximo registro deve ser lido (respeitado o sentido lógico em que se processa a leitura da cadeia).
- END., posto pelo TOTAL, indicando que uma leitura na cadeia de um variável foi tentada após o fim da mesma. Serve para que o programa de aplicação, testando após cada operação de leitura de cadeia de um variável o refer, detecte exatamente

o fim da cadeia.

V.6 - Linkage Path (ou Link ou Caminho de Vínculos)

Este parâmetro aponta para um campo de oito posições, definido pelo usuário, que deve conter o nome de oito posições do vínculo, igual ao existente na definição do arquivo. Este parâmetro é o veículo através do qual o usuário identifica uma relação específica entre uma cadeia de um variável e um registro de mestre.

Os termos cadeia primária ou cadeia controladora se referem ao caminho de vínculos que aparece na Lista de Parâmetros. Qualquer outro, enunciado para o arquivo quando da definição deste é conhecido como cadeia secundária.

V.7 - Chave

Este parâmetro aponta para um campo de tamanho variável, definido pelo usuário, no qual deve ser colocada a chave de um registro de mestre a ser processado. O TOTAL randomiza o conteúdo do campo chave para localizar um registro de mestre ou para vinculá-lo a uma cadeia de variável. O tamanho do campo chave é definido quando da definição do mestre em questão. Sugere-se usar como nome do campo chave o nome definido na área de dados, para evitar eventuais discrepâncias entre ambos.

V.8 - Lista de Dados ou Data-List

Este parâmetro aponta para um campo de tamanho variável que tem que conter uma lista de nomes de campos elementares. Os nomes constantes nesta lista deverão formar um "character string" e, cada um deles deve ter sido enunciado na definição do arquivo.

A lista de dados deverá sempre ter o formato seguinte:

elem 1 elem 2 ...elem n END.

A lista de dados pode incluir:

. a - dados
. b - chaves de controles
. c - record codes (ver o conceito em III.10)

A lista de dados normalmente não inclui:

. raiz (ver conceito em III.4)
. vínculo de mestre
. vínculo de veriável

se bem que para o administrador do Banco de Dados existem instrumentos que lhe permitem ler e alterar também estes campos.

A lista de dados poderá não estar organizada na mesma ordem do layout do registro lógico, mas certamente deverá estar ordenada como a área de dados.

V.9 - Área de Dados ou Data-Area

Este parâmetro aponta para uma área na memória, definida pelo usuário, que funciona como área de E/S para os elementos de dados nomeados na lista de dados. A estrutura e características desta área devem ser coincidentes com os dados da lista de dados.

V.10 - DBMOD

Este parâmetro aponta para um campo de seis posições, definido pelo usuário, que contem o nome do Módulo Descritor do Banco de Dados a ser usado.

Explicando melhor: quando se usa o TOTAL batch, cada programa

de per si ativa uma cópia do TOTAL e um DBMOD próprio (para reconhecer o Banco de Dados em questão). Em operação multi-task, tal não é necessário: o TOTAL, ao ser ativado, ativa um DBMOD geral, que fica ativo durante toda a seção. Por uniformidade de linguagem, foi mantido o parâmetro mesmo em instalações com operação multi-task, quando então a sua informação é desprezada pelo TOTAL.

V.11 - Task

Este parâmetro aponta para um campo de oito posições, definido pelo usuário, contendo o nome da task a que pertence o programa.

V.12 - End

Este parâmetro aponta para um campo de quatro posições, definido pelo usuário, cuja função é delimitar a lista de parâmetros. Os únicos valores possíveis para este campo são:

- "END" : se um registro recuperado deve ser retido para posterior utilização.
- "RLSE" : caso contrário.

VI - AS FUNÇÕES DO TOTAL

Para podermos entender o mecanismo de funcionamento do TOTAL e consequentemente o modelo de simulação de que trata o presente trabalho, não podemos deixar de analisar as principais funções executadas pelo programa em questão.

O modelo, como veremos mais tarde, visa fornecer parâmetros que permitam otimizar a alocação de arquivos de um Banco de Dados TOTAL, em um número limitado de volumes de disco. O parâmetro principal é o tempo de deslocamento físico e posicionamento do braço do disco, pois ele será o fator limitante no tempo médio de acesso por informação recuperada. Com base nisso, o modelo

se detem em analisar somente funções que tenham possibilidade de atuar fisicamente no braço do disco. E são essas funções que descrevemos a seguir, analisando primeiro os formatos das listas de parâmetros.

VI.1 - Formato Geral das Listas de Parâmetros

As listas de parâmetros, dependendo da função executada, se enquadram em alguns padrões bem definidos de formato. Das funções que nos interessam, só as relativas à desativação de tasks e aquelas privativas da administração do Banco de Dados têm listas de parâmetros que não se enquadram em um dos dois formatos a seguir.

VI.1.1 - Formato 1

Função, status, arquivo, refer, cadeia de vínculos, chave, lista de dados, área de dados, end.

VI.1.2 - Formato 2

Função, status, arquivo, chave, lista de dados, área de dados, end.

VI.2 - ADD-M (Add Master)

Formato da lista de parâmetros: 2

Descrição: Opera adicionando logicamente da área de dados um registro a um mestre. Isso é conseguido por randomização da chave, para achar o endereço relativo e, selecionando na área de dados os elementos especificados na lista de dados.

VI.3 - ADDVA (Add Variable After)

Formato da lista de parâmetros: 1

Descrição: Opera adicionando logicamente um registro em uma cadeia de um variável, com dados da área de dados , após o registro cujo endereço relativo está no refer. A inserção lógica é feita somente na cadeia cujo caminho de vínculos esteja referido no campo respectivo da lista de parâmetros. Para todos os outros vínculos definidos para este registro, a adição se fará ao final das respectivas cadeias. O campo refer normalmente aponta um endereço relativo, após o qual deverá ser feita a adição lógica na cadeia. Todos outros valores são inválidos. Após um completamento satisfatório da função, o refer conterá o endereço relativo do registro adicionado.

VI.4 - ADDVB (Add Variable Before)

Formato da lista de parâmetros: 1

Descrição: Opera adicionando logicamente um registro em uma cadeia de um variável, com dados da área de dados, antes do registro cujo o endereço relativo está no refer. Como no caso do ADDVA, a inserção lógica só é feita na cadeia referida pelo caminho de vínculos da lista de parâmetros, sendo que para todos os outros eventuais vínculos do registro a inserção é feita ao final das respectivas cadeias. O campo refer normalmente aponta um endereço relativo, antes do qual deverá ser feita a adição lógica na cadeia. Todos os outros valores são inválidos. Após um completamento satisfatório da função, o refer conterá o endereço relativo do registro adicionado.

VI.5 - ADDVC (Add Variable Continue)

Formato da lista de parâmetros: 1

Descrição: Opera adicionando logicamente um registro em uma cadeia de um variável, com dados da área de dados, ao final de todas as cadeias de todos os vínculos definidos para o registro. O conteúdo do campo refer não é usado para a execução da função, mas é testado quanto à validade. Se for diferente de:

- LKxx
- endereço relativo;

a função não é executada e um status de erro é devolvido. Se a função teve sucesso, o refer contém o endereço relativo do registro adicionado.

VI.6 - ADDVR (Add Variable Replace)

Formato da lista de parâmetros: 1

Descrição: Opera revinculando um registro existente na cadeia de um variável, para outras cadeias sem que fisicamente sejam movidos dados da área de dados para o registro. Na verdade, o que o TOTAL faz é desvincular o registro da cadeia velha e revinculá-lo logicamente ao final da nova cadeia, controlada pela chave da área de dados do usuário.

O conteúdo do refer deve ser um endereço relativo, que é usado pelo TOTAL para recuperar um registro para dentro da área de E/S do próprio TOTAL (são os buffers viáveis, ver conceituação no item IV.1) se já lá não estava e posteriormente usado novamente, (este endereço relativo no refer) para regravar o registro. Após completar com sucesso a função, o refer conterá ainda o mesmo endereço relativo.

Em relação a esta função cabem as seguintes considerações adicionais:

- É a única função que pode ser usada para alteração de record code's (ver conceito no item III.10 - redefinição de registros de um variável). Se for este o caso, todos campos de controle da área redefinida para o novo record code terão que estar presentes, mesmo que já estejam na forma definitiva no registro original.
- ADDVR não permite manutenção no caminho de vínculos primário (ou seja, o vínculo pelo qual a cadeia do variável está sendo lida (ADDVR exige um prévio READV explícito do programa de aplicação para trazer o registro para o buffer), ou ainda, o vínculo que está na lista de parâmetros). Para este vínculo, a manutenção tem que ser do tipo "delete/add". Em verdade, o TOTAL mantém todos os vínculos que não o primário, executando, com efeito, uma eliminação e uma adição.

VI.7 - DEL-M (Delete Master)

Formato da lista de parâmetros: 2

Descrição: Opera determinando o endereço relativo do registro do mestre especificado, por randomização da chave. A eliminação consiste em branquear o registro do mestre, e por à disposição imediatamente para reutilização o espaço liberado.

Algumas considerações adicionais sobre esta função seriam:

- O TOTAL não elimina registros de mestre se existirem variáveis ainda vinculados a ele; o programa de aplicação deve providenciar uma prévia eliminação desses registros de variável (ver explanação a respeito no item III.5 - Características de Variável).

- Em instalações multi-task, o registro deve ser lido e retido (ver explicação no item V.12) antes de ser eliminado, em operações explícitas do programa de aplicação.

VI.8 - DELVD (Delete Variable Direct)

Formato da lista de parâmetros: 2

Descrição: Opera eliminando o registro do variável cujo o endereço relativo está no campo refer. O registro é branqueado e removido logicamente de todos os caminhos de vínculos. A área liberada é imediatamente posta à disposição para reutilização.

O campo refer inicialmente contem o endereço relativo do registro a ser eliminado. Após completada a função, o refer contem o endereço relativo do registro lógico anterior na cadeia. Se o registro eliminado for o primeiro da cadeia, o refer após a operação conterá LKxx.

Algumas considerações adicionais sobre esta função seriam:

- Se o DELVD for seguido imediatamente por um READV (será visto mais adinate), é recuperado o registro que está logicamente após o registro eliminado.
- Se o DELVD for seguido imediatamente por um READR (também será visto mais adinate) o registro logicamente anterior ao eliminado é abandonado e o registro seguinte (no mesmo sentido) é recuperado.
- Em instalações multi-task, o registro a ser eliminado deve previamente ser lido e retido (ver item V.12) explicitamente pelo programa de aplicação.

VI.9 - READD (Read Direct)

Formato da lista de parâmetros: 1

Descrição: Opera recuperando o registro lógico de um variável, cujo endereço relativo está no campo refer. A continuação do processamento pode se fazer ao longo de qualquer caminho de vínculos válido para o registro.

O campo refer, como vimos, deve conter o endereço relativo do registro de variável a ser recuperado. Após uma recuperação bem sucedida, o refer ainda conterá o mesmo endereço.

Algumas considerações adicionais sobre essa função seriam:

- O endereço relativo de um registro de variável não é, provavelmente, gerado por uma rotina do usuário. Com grande probabilidade, o endereço foi gerado pelo TOTAL, quando de uma interrupção nas leituras ao longo de uma cadeia, e deve ser usado para reinício de processamento, para recuperação do último registro lido.
- Este comando é de grande utilidade em processamento de estruturas hierárquicas, pois o programa de aplicação salva o refer retornado pelo TOTAL, em um nó da estrutura, e retorna posteriormente a este mesmo endereço relativo.
- Se o registro deve ser lido mas não atualizado, deve-se usar preferencialmente 'RLSE' em vez de 'END' no parâmetro END (ver política de liberação de buffers, mais adiante).

VI.10 - READM (Read Master)

Formato da lista de parâmetros: 2

Descrição: Opera randomizando o conteúdo do campo chave, trazendo o registro de mestre, existente no endereço obtido, para dentro da área de dados de acordo com a lista de dados.

Se o registro é para ser lido somente e não atualizado, usar 'RLSE', em vez de 'END.' no parâmetro END (ver observação ao final da função anterior).

VI.11 - READR (Read Reverse)

Formato da lista de parâmetros: 1

Descrição: Opera seguindo logicamente de trás para adiante uma cadeia de um variável, seguindo um determinado caminho de vínculos. Para se ler a cadeia inteira, o refer é inicializado com LKxx e emite-se o comando READR. Nessa primeira leitura, o TOTAL usa a chave para acessar o registro de mestre correspondente e obter o endereço relativo do final de cadeia. O último registro desta é então fornecido ao programa solicitante. Daí em diante, o processamento continua emitindo-se seguidos READR's. Como o campo refer sempre aponta para o endereço relativo do último registro lido na cadeia, a cadeia reversa é percorrida e os seus registros são recuperados em ordem reversa, até que o primeiro da cadeia seja lido. Após esta última leitur, o valor apontado pelo refer sera 'END.', indicando fim de cadeia. Se 'END.' for devolvido no refer logo no primeiro READR, é sinal que não existem registros na cadeia.

Algumas considerações adicionais sobre esta função seriam:

- Deve ser tomado cuidado com o uso de READR, imedia

tamente após um DLVD. Ver observação a respeito no item VI.8.

- Após processamento completo de uma cadeia de um variável (o refer conterá 'END.'), deve-se reposicionar o refer para LKxx antes do início de processamento de outra cadeia com READR. Caso contrário, esse READR devolverá como satatus o valor 'IPAR'.
- Se o registro deve ser lido mas não atualizado, usar 'RLSE' em vez de 'END.', no parâmetro END (ver última observação do item VI.8).

VI.12 - READV (Read Variable)

Formato da lista de parâmetros: 1

Descrição: Opera seguindo logicamente do início para o fim, uma cadeia de um variável, seguindo determinado caminho de vínculos. Para ler a cadeia inteira, o refer é posicionado inicialmente em LKxx e emite-se o comando READV. Nessa primeira leitura, o TOTAL usa a informação do campo chave para acessar o registro de mestre correspondente e obter o endereço relativo do registro do início da cadeia do variável. Este registro inicial é então passado para o programa de aplicação. Daí em diante, o processamento continua emitindo-se seguidos READV's. Como o campo refer sempre aponta para o endereço relativo do último registro lido na cadeia, a cadeia direta é percorrida e seus registros são recuperados nesta ordem, até que o último da cadeia seja lido. Após esta última leitura, o valor apontado pelo refer será 'END.', indicando fim de cadeia. Se este código foi devolvido logo no primeiro READV, é sinal que não existem registros na cadeia.

Prevalecem para o READV as mesmas duas últimas considerações adicionais vistas para o READR.

VI.13 - SEQRM (Serial Read Master)

Formato da lista de parâmetros: 2

Descrição: Opera por leitura serial de um arquivo mestre, isto é, leitura em sequência física. O comando usa o endereço relativo para recuperação de registros, sendo que inicialmente deve ter-se tido o cuidado de reposicioná-lo em zero (caso se tenha lido algum mestre serialmente, antes, na mesma task) com a função RESTM.

Sucessivos SEQRM's farão com que o TOTAL a cada vez incremente de 1 o endereço relativo, recuperando o respectivo registro. Mas ele tem o cuidado de só passar registros de dados para a área de dados: registros em branco e de controle são abandonados.

Considerações adicionais sobre esta função:

- O não uso da função RESTM após a leitura serial de um dado mestre, fará com que a primeira leitura serial em um segundo mestre comece pelo meio deste, ou devolva um STATUS de 'END.', indicando fim de arquivo. Isso porque o endereço relativo do primeiro registro recuperado do 2º arquivo será igual ao do último do 1º arquivo, mais 1.
- Por motivo de otimizações internas de alocação feitas pelo TOTAL por ocasião de ADD-M's e DEL-M's devem ser evitadas operações conjugadas destas funções com SEQRM: alguns registros poderiam ser lidos duplamente, outros, nenhuma vez.

O "porquê" disso se encontra na manutenção de raízes (cadeias de sinônimos). Rever explicação no item III.4, para melhor compreensão.

- O comando WRITM (será visto mais adiante) pode ser usado para atualizar um registro recuperado com SEQRM.

- Só um mestre pode ser processado por vez, por task (exatamente pelo controle centralizado de contagem de registros que o TOTAL faz).

VI.14 - SEQRV (Serial Read Variable)

Formato de lista de parâmetros: 2

Descrição: Opera por leitura sequencial de um variável; isto é leitura em sequência lógica, ou seja, seguindo cadeias do variável. Utiliza para tanto o seguinte procedimento:

Passo a - recupera e examina o primeiro registro lógico do arquivo.

Passo b - se for um registro de dados e mais, for o início de uma cadeia, considerando o caminho de vínculos que na definição do variável aparece em primeiro lugar, então o mesmo é colocado na área de dados, de acordo com a lista de dados.

Passo c - se o teste anterior (passo b) fracassar, o registro é abandonado e, tantos registros lógicos seguintes serão lidos até um deles atender integralmente ao teste.

Passo d - achado o início de uma cadeia, a mesma é seguida (através de sucessivos SEQRV's) seguindo o caminho de vínculos definido no passo b, até o seu fim.

Passo e - atingido o fim de uma cadeia, um novo SEQRV faz com que o TOTAL retorne ao início da mesma, e por leituras seriais, tente achar novo início de cadeia, retornando ao passo b.

Somente registros de dados são considerados: registros em branco e de controle são abandonados.

Algumas considerações adicionais para esta função seriam:

- O usuário deve se precaver e informar ao TOTAL que o primeiro registro a ser pesquisado é o primeiro do arquivo, através da função RESTV. Isso porque para a busca de cadeias, o TOTAL usa dois endereços relativos.

  Esses dois endereços relativos (ou contadores, como citados adiante) são respectivamente o do início da cadeia e o do registro atual.

  Não usar RESTV antes do início de um segundo ciclo de SEQRV's poderá causar resultados imprevisíveis, visto que os dois contadores usados pela função conterão valores que só dirão respeito ao primeiro ciclo de SEQRV's.

- A eliminação via DELVD de registro recuperado via SEQRV fará com que se perca a recuperação do resto da cadeia a que pertence o registro. Por tanto não se deve usar DELVD e SEQRV conjugadamente.

- SEQWV (será visto a seguir) pode ser usado para atualizar registros recuperados via SEQRV.

- Só um variável pode ser processado via SEQRV por vez, por task (por causa dos contadores internos).

VI.15 - SEQWV (Serial Write Variable)

Formato da lista de parâmetros: 2

Descrição: Opera regravando um registro previamente recuperado por SEQRV ou SERLV (será visto a seguir). Esta função não permite manutenção de vínculos.

VI.16 - SERLV (Serial Read Variable)

Formato da lista de parâmetros: 2

Descrição: Opera por leitura serial de um variável, isto é, por sequência física, usando para tanto somente um endereço relativo (o do registro que está sendo lido). Supondo que o início da leitura se tenha dado no início físico do arquivo (com eventual uso do comando RESTV para restaurar o endereço relativo para o primeiro registro, após uma varredura anterior do variável com ciclos de SERLV), sucessivos SERLV's farão com que o TOTAL incremente de uma unidade o endereço relativo do registro lido e recupere o seguinte. Somente dados são transferidos para a área de dados: registros em branco e de controle são abandonados.

Algumas considerações adicionais sobre esta função seriam:

- Usar o RESTV se necessário para restaurar o endereço relativo inicial de leitura. Semelhantemente ao SEQRV, após uma leitura inteira de um variável com SERLV, o contador interno estará apontando para o último registro lógico do arquivo. Um SERLV posterior tentará recuperar um registro de endereço relativo igual a este último, +1.
- SEQWV pode ser usado para atualizar um registro recuperado por SERLV.
- Somente um variável por vez por task pode ser lido via SERLV (justamente por causa do controle interno de registros do TOTAL).

VI.17 - WRITM (Write Master)

Formato da lista de parâmetros: 2

Descrição: Opera randomizando o conteúdo do campo chave de um mestre para recuperar o registro a ser atualizado. Os dados constantes na área de dados são movidos para o registro, que é então gravado. O programa de aplicação fica obrigado a explicitamente ler e reter (ver item V.12) o referido registro, em instalações multi-task.

VI.18 - WRITV (Write Variable)

Formato da lista de parâmetros: 1

Descrição: Opera gravando o registro de variável cujo endereço relativo se encontra no campo refer. Como esta função não executa manutenção de vínculos, estes últimos e os eventuais record codes do registro não podem ser alterados. Em instalações multi-task, o registro deve ser explicitamente lido e retido (item V.12) antes do WRITV, pelo programa de aplicação.

VI.19 - CLOSP (Close Physical)

Formato da lista de parâmetros

Descrição: Opera fechando lógica e fisicamente um arquivo mestre ou variável. Fechar logicamente um arquivo implica , entre outras coisas, em gravar qualquer buffer viável (ver item IV.1) com "write pendente" (ver item III.12) sendo pois operação que gera E/S. Fechar fisicamente um arquivo significa emitir um comando de CLOSE para o sistema operacional.

VI.20 - DEQUE

Formato da lista de parâmetros: DEQUE, STATUS, TASK, END.

Descrição: Opera desativando uma task, executando as seguintes funções:

- liberando todos registros retidos pela task.
- forçando a gravação de todos buffers com "write pendente" (ver item III.12). Isso é conseguido em instalações multi-task por meio da emissão automática de um comando QUIET (ver item VI.12).
- Fechando qualquer arquivo que tenha ficado aberto (só para instalações batch. Rever diferença entre batch e multi-task no item IV.1).
- Removendo, em instalações multi-task, esta task da tabela interna de tasks do TOTAL. Esta tabela é interna, mantida pelo TOTAL e contém a relação de todas as tasks ativas em determinado momento. Uma task se inscreve nesta tabela quando ela emite o comando TOTAL, passando a ser conhecida pelo TOTAL.

O interesse da função no modelo reside justamente na ação de E/S gerada eventualmente pela gravação dos buffers com "write pendente".

VI.21 - QUIET

Formato da lista de parâmetros: QUIET, STATUS, CONTADOR, END.

Descrição: Opera forçando a gravação de todos os buffers com "write pendente" (item III.12), e grava um registro indicativo de QUIET no arquivo log.

Uma explicação sobre esse arquivo log.

- É um arquivo que contem registros com todas as funções do TOTAL ou somente de alguns tipos selecionados, que tenham sido emitidas em uma determinada seção (seção aqui tem o sentido de intervalo de tempo em que o TOTAL esteve continuadamente ativo). O modelo de simulação de que trata o presente tra-

balho se baseia fundamentalmente nas informações contidas neste arquivo.

- O log poderá estar ativo ou não. Neste último caso (que é a situação normal de operação do TOTAL na TELERJ), evidentemente o registro de QUIET referido anteriormente não poderá ser gravado. Mas serão feitas as gravações dos buffers. Imediatamente após essa gravação, o Banco de Dados físico (isto é, os arquivos em disco) estará perfeitamente aderente aos dados que estão sendo manipulados na memória. A essa aderência chamamos de sincronização banco-buffer e, representa um ponto a partir do qual um reprocessamento pode ser feito. O comando QUIET, por suas características, só tem sentido em instalações multi-task.

Na lista de parâmetros aparece o campo de contador , que vamos explicar agora.

No início de uma seção, ao se por o TOTAL no ar, no PARM do cartão EXEC do programa ATTACH (este programa estabelece a comunicação entre as tasks dos usuários e o TOTAL), um dos valores especificados, mais precisamente o quinto, fornece um número que determina o número de atualizações de arquivos (add's, write's, delete's) entre dois QUIET sucessivos emitidos automaticamente pelo TOTAL (as funções citadas são supostas terem sido efetuadas somente nos buffers). Se no PARM este valor for omitido, um valor alternativo padrão de 200 atualizações é assumido.

Já na emissão explícita de um QUIET por um programa de aplicação, o valor deste contador (que é o mesmo que o anterior) poderá ser zeros binários ou um valor previamente definido pelo usuário. Em ambos os casos, o QUIET é executado, mas no segundo (contador $\neq$ zero), o valor inicial do cartão EXEC é restaurado para o valor do contador.

VI.22 - WRITD (Write Direct)

Formato da lista de parâmetros: WRITD, STATUS, ARQUIVO, ENDEREÇO RELATIVO, ÁREA DE DADOS, END

Descrição: Opera gravando um registro lógico inteiro em um endereço relativo especificado pelo usuário. Normalmente é usado em instalações multi-task, pelo administrador do Banco de Dados para repor a imagem "antes", no caso de recuperação do banco após uma destruição física ou lógica do mesmo. O endereço relativo é o que aparece na lista de parâmetros e deve estar em formato binário. Os dados provém da área de dados.

Este comando deve ser usado com extrema cautela, pois o TOTAL bypassa todas suas rotinas internas de segurança, verificando tão somente:

- se o arquivo existe.
- se o arquivo pode conter o endereço relativo especificado.
- se existe em verdade tal endereço no arquivo.

VI.23 - ENDTO (End TOTAL)

Formato da lista de parâmetros: ENDTO, STATUS, END

Descrição: Opera encerrando a operação do TOTAL.

Nenhuma instalação multi-task isso implica em:

- nenhuma função adicional, de nenhuma task, é aceita pelo TOTAL.
- todas pendências de funções anteriores são providenciadas.
- todos os buffers de E/S são fisicamente gravados , desde que tenham "write pendente".
- todos os arquivos são fechados.

- depois de todos os eventos anteriores terem ocorrido, o TOTAL cessa suas funções.

VI.24 - Considerações adicionais sobre Funções e Acessos

Quando do acesso a um registro de mestre poderá não ocorrer somente uma operação física de E/S (estamos supondo que o registro desejado não está em nenhuma versão do buffer viável e, que portanto será necessário buscá-lo fisicamente no arquivo): caso haja encadeamento de sinônimos e, o registro procurado esteja um bloco físico diferente daquele do seu home-address (ver conceito no item III.4), um novo acesso físico se fará necessário.

Quando do acesso a registro de um variável, o número de acessos estará condicionado ao tipo de função usada e, no caso de seguir uma cadeia, da posição relativa do registro recuperado na mesma. Explicando em detalhe os dois casos:

- No caso de um ciclo de SERLV's certamente a relação acessos físicos/registros lógicos recuperados será 1/n, onde n é o fator de bloco do arquivo.
- No caso de seguir cadeias com ciclos de READV's (um ciclo por chave do mestre associado à cadeia) teremos a seguinte situação:
  - . 1º READV - um acesso ao mestre para obter o endereço relativo do primeiro da cadeia, seguido de um acesso ao variável para obter o dito registro.
  - . READV's seguintes, até o final da cadeia - o endereço relativo é tirado do campo de vínculo do último registro lido na cadeia (o endereço relativo deste está no refer); normalmente a cadeia estando totalmente contida em um só bloco, a relação acessos físicos/registros lógicos recuperados será 1/m, onde m é o número de registros da cadeia -1 (o primeiro foi lido no READV inicial). Se esta se extender por mais de um bloco físico (p/blocos, por exemplo), a relação aumentaria para p/m.

- No caso de varredura do arquivo com SEQRV's seguidos, detectado o início de uma cadeia, a relação seria 1/m ou p/m, conforme o critério para os READV's. A detectação desse início de cadeia que poderá ou não ser (em geral é) mais eficiente que o do READV: é serial dentro do bloco em que está o início da última cadeia lida, como visto na descrição da função.
- Para garantir que a gravação seja bem sucedida (estamos limitados aqui somente a gravações sobre registros previamente existentes, isto é, não estão incluídas adições), o TOTAL exige que o programa de aplicação traga explicitamente para a memória o registro em questão, com a lista de parâmetros da função de leitura indicando 'END.' no parâmetro END. Desta maneira o registro ficará retido para a operação subsequente. Isso vale para mestres e variáveis, e inclui "writes" e "deletes".
- Determinadas funções, além dos acessos indicados anteriormente para chegar à informação, podem exigir eventualmente acessos adicionais após a função, a um ou mais arquivos, visando basicamente à atualização de vínculos. São elas:

    - Em mestres:

        . ADD-M e DEL-M - para manutenção de cadeias de sinônimos, no mestre em questão.

    - Em variáveis:

        . ADDVA, ADDVB, ADDVC, ADDVR e DELVD, para:
        manutenção de vínculos na própria cadeia do variável.

        manutenção dos vínculos em todos os mestres associados.

VII - O MODELO

VII.1 - Introdução

Chegamos finalmente ao ponto em que podemos abordar o objetivo

do presente trabalho:

> *Um modelo de simulação para alocação de arquivos de um Banco de Dados em um número limitado de volumes de DASD's.*

A que em verdade se propõe o modelo?

A proposição básica do modelo se resume em obter parâmetros que sirvam de elemento de diferenciação entre diferentes alocações de arquivos.

O que são os parâmetros de diferenciação?

Basicamente são parâmetros que visam aquilatar o deslocamento físico dos braços dos diversos volumes de disco, contendo o Banco de Dados.

Como são obtidos os parâmetros de diferenciação?

Os parâmetros são obtidos simulando o funcionamento do TOTAL sobre o arquivo log por ele fornecido. Este arquivo fornece, a cada CALL DATBAS emitido pelos programas de aplicação, um registro log com as seguintes informações (entre outras):

- função
- task
- arquivo
- refer
- caminho de vínculos
- status
- tamanho do registro
- imagem do registro no buffer, após a função

VII.2 - Os Parâmetros

Os parâmetros seriam:

- total de registros do log
- total de funções do tipo "de acesso"
- total de deslocamentos
- total de não-deslocamentos
- total de acessos com deslocamento
- total de acessos sem deslocamento
- total de cilindros deslocados
- total de bytes transmitidos
- total de tempo de deslocamento em mseg
- cilindros deslocados por acesso
- tempo de deslocamento por acesso

Para dar idéia de distorções localizadas os parâmetros deverão ser dados por:

- task
- volume
- buffer
- arquivo

VII.3 - Análise dos Parâmetros

Total de registros do log: é o número de registros lidos no log.

Total de funções do tipo "de acesso": é o número de registros nos quais a função é uma das analisadas no capítulo VI, ou seja, capaz de, eventualmente, provocar acesso físico ao disco.

Total de deslocamentos: é o número de deslocamentos do braço realmente executados, eventualmente mais de um por função lida (por causa da manutenção de raizes e vínculos e, das funções de sincronização de buffers).

Total de não-deslocamentos: é o número de deslocamentos do braço que eram previstos ocorrer por necessidade da informação e, não ocorreram devido a mesma já se encontrar em algum buffer, por causa de uma função previamente executada. Eventualmente não coincide com o número de acessos sem deslocamentos, por causa das manutenções de vínculos.

Total de acessos sem deslocamento: é o número de funções do tipo "de acesso" em que o acesso físico não foi necessário, visto que a informação procurada já se encontrava em algum buffer.

Total de acessos com deslocamentos: é o número de funções do tipo "de acesso" em que o acesso físico foi necessário, visto que a informação procurada não se encontrava em nenhum buffer. A soma de "com" e "sem" deve dar igual ao total de acessos. O total "com" pode ser igual ou menor que o total de deslocamentos, visto que este último pode ser mais de um por acesso com deslocamento, nos casos de manutenção de vínculos e raízes e sincronização de buffers.

Total de cilindros deslocados: é o número de cilindros "viajados" pelo braço do DASD, no total de deslocamentos.

Total de bytes transmitidos: é o número de bytes de todos os blocos físicos transmitidos fisicamente.

Total de tempo de deslocamento: o modelo considera dois tempos envolvidos num acesso físico ao disco: delay posicional, que é o tempo gasto pelo braço para se deslocar de um determinado cilindro para outro, no mesmo volume. Latência rotacional, que é o tempo gasto para que, uma vez o braço parado sobre determinado cilindro, o bloco físico venha a se colocar sob o cabeçote de leitura-gravação, por rotação do disco, para então se efetuar a leitura ou gravação propriamente dita.

Cilindros deslocados por acesso: é a relação total de cilindros deslocados/total de funções do tipo "de acesso".

Tempo de deslocamento por acesso: é a relação tempo total de deslocamento/total de funções do tipo "de acesso".

VII.4 - Funcionamento do Modelo

O modelo, de posse do log do TOTAL, lê registro a registro, simulando a cada leitura a função indicada. Ele parte da pressuposição que existem as seguintes informações:

Detalhamento da alocação de arquivos, contendo:

- nome do arquivo

- em quantos extents ele se divide
- em que volume está cada extent
- o cilindro e a trilha de início de cada extent
- o cilindro e a trilha de fim de cada extent

Detalhamento da especificação de vínculos entre mestres e variáveis, contendo:

- nome do variável
- nome do mestre
- quantos vínculos existem entre este mestre e este variável (é possível existir mais de um? Reveja item III.11) e o nome de cada um deles.
- para cada vínculo, onde se inicia a chave no layout do variável, qual o tamanho médio das cadeias desse vínculo e se a carga do variável foi feita ou não respeitando este vínculo (para esse último conceito rever item III.7).

Detalhamento das características de cada arquivo, contendo:

- nome do arquivo
- indicação se é mestre ou variável
- qual o buffer viável para este arquivo
- tamanho da chave em bytes (só para mestres)
- capacidade total de registros lógicos
- comprimento em bytes do registro lógico
- número de registros lógicos por bloco
- número de blocos físicos por trilha
- total de trilhas
- fator de ocupação (em percentagem, reveja este conceito, válido somente para mestres, no final do item III.4)

Detalhamento de algumas características das funções, contendo:

- nome da função
- se atua em mestres ou variáveis
- se implica em acesso
- se é de leitura ou gravação
- sendo de gravação, se é de manutenção estrutural (implicando em manutenção de vínculos, tipo "add" ou "delete") ou não (tipo "write").

- se força sincronização de buffers (funções do tipo DEQUE, QUIET, etc.)
- quais os status válidos para aquela função ou seja, mesmo não sendo '****' a função ainda tenha sido executada.

De posse dessas informações, o modelo tem condições de determinar:

- se a função do log é do tipo "de acesso".
- sendo "de acesso", se é de sincronização de buffers ou não. Sendo, a função tem seus parâmetros computados em procedures específicas.

Não sendo de sincronização, são feitos os testes de:

- validade de status. É logo o teste inicial, pois não há sentido em analisar uma função que eventualmente não foi nem executada (o status devolvido é uma segura indicação de tal condição).
- necessidade de acesso (a informação pode estar nos buffers). Se sim, os parâmetros são computados.
- necessidade de manutenção de vínculos (este teste é feito em sequência ao anterior). Se sim, os parâmetros são computados em procedures específicas.

Algumas funções auxiliares contidas no modelo seriam:

- determinação do buffer a ser usado na função: utiliza os critérios analisados no III.13.
- liberação do buffer escolhido: se este buffer contem a marca de "write pendente", deve ser contabilizado o acesso físico que será feito, precedendo a liberação.

Cálculo do endereço relativo, de posse da chave

- utiliza diretamente a função RQLOC (REQUEST LOCATION) existente no TOTAL.

Geração de endereços relativos

- quando da necessidade de se obter o endereço relativo do registro inicial de uma cadeia no variável , não podendo consultar o vínculo no mestre, deve-se simular a obtenção deste endereço. Essa simulação

é feita de forma aproximada para o vínculo de carga do variável e de forma aleatória para qualquer outro vínculo.

Cálculo do cilindro-alvo

- é obtido através do endereço relativo, levando-se em conta detalhes do arquivo e distribuição dos seus extents.

Simulação da operação de E/S

- nesta rotina são simplesmente incrementados os parâmetros tais como: número de bytes transmitidos, cilindros deslocados e tempo de deslocamento.

Atualização de buffer após uma operação de E/S

- vários parâmetros indicativos da condição atual do buffer são atualizados, tais como: nome do arquivo, volume, endereço relativo inicial, endereço relativo final, ordem do bloco físico, trilha, cilindro, marca de tempo indicadora do momento exato do último uso (é a informação básica na liberação de buffers).

Uso aqui significa fazer referência a alguma informação contida no bloco, independente da necessidade ou não de acesso físico.

Estatísticas

- cômputo dos diversos parâmetros, com impressão dos mesmos ao final de toda simulação.

VIII - CRITÉRIOS DE ALOCAÇÃO

Vimos no capítulo anterior que o modelo exige que uma determinada alocação seja pré-estabelecida para todos os arquivos do Banco de Dados. O que o modelo faz é fornecer parâmetros que mostram o acerto de uma ou outra alocação.

O ideal seria usar o simulador como parte componente de um modelo maior, este então de otimização de alocação. O problema que surge é que um tal modelo demandaria um tempo de dedicação que o autor do presente trabalho infelizmente não poderia dispor.

O que se pode fazer é anunciar uma série de critérios de alocação, sugeridos pela experiência e pelo bom-senso, e utilizar o simulador para detectar desotimizações localizadas.

Levando-se em consideração tudo o que foi visto anteriormente, podemos enunciar algumas diretrizes para alocação de arquivos.

VIII.1 - Processamento Paralelo

Por processamento paralelo entende-se a seguinte situação: um mesmo programa acessando simultaneamente dois arquivos do Banco de Dados.

Uma boa maneira de se obter informações sobre processamento paralelo é fazer um levantamento dos programas de aplicação, sumariando seu processamento.

Uma sugestão seria algo como demonstrado no anexo 10, com alguns programas do Sistemas de Informações de Pessoal (SIP).

Se processamento paralelo implica em um único programa processar simultaneamente dois arquivos do Banco de Dados, então, numa primeira aproximação, esses dois arquivos não deverão ficar no mesmo volume.

Situação semelhante seria a de dois programas acessando cada um deles um arquivo do Banco de Dados (arquivos distintos) e, por alguma razão de ordem técnica ou administrativa, os dois programas devem rodar simultaneamente. Seria também um caso de processamento paralelo, e os dois arquivos novamente não deveriam ocupar o mesmo volume.

Em relação ao processamento paralelo porém cabem algumas considerações que podem mudar o critério inicial de arquivos em volumes separados:

- ponderar este programa em relação a outros que processem os mesmos arquivos, mas que não rodem simultaneamente. A ponderação deve levar em conta características como:

- frequência de rodada
- tempo gasto de CPU
- grau de responsabilidade (esta última característica avaliaria situações do tipo: atualização tem maior responsabilidade que consulta).

Se o programa em questão tivesse "peso" muito menor que os outros poderia eventualmente ser mais interessante manter os dois arquivos no mesmo volume, pois a limitação imposta por ele passaria a um segundo plano.

Se em um processamento paralelo, a frequência de uso dos dois arquivos, for muito diferente um do outro (por exemplo: um é 100 vezes mais acessado que o outro), a exigência de volumes separados também passa para um segundo plano.

VIII.2 - Processamento Serial

Esse tipo de processamento ocorre quando um programa acessa dois arquivos serialmente (isto é, primeiro um inteiro, depois o outro) ou quando dois programas acessam dois arquivos distintos, os dois programas não rodando simultaneamente. É evidente que, a menos de restrições externas ao processamento, os dois arquivos poderão estar no mesmo volume sem que isso implique numa degradação da performance.

VIII.3 - Tamanho dos Arquivos

Uma restrição elementar para a alocação de arquivos diz respeito à capacidade do volume: o somatório dos tamanhos de todos os arquivos alocados a um volume não pode ultrapassar à capacidade deste.

VIII.4 - Sincronização Mestre-Variável

Não sendo propriamente um critério de alocação, o sincronismo mestre-variável é sempre desejável em termos de minimização de E/S.

Em que consiste esse sincronismo? A explicação está diretamente vinculada ao conceito de chave de carga de um variável (ver item III.7).

Uma situação muito comum em programas que manipulam arquivos de uma rede TOTAL é um processamento do tipo:

- DO WHILE ("até o fim do mestre");
- SEQRM no mestre (notar que os registros serão recuperados em ordem ascendente de RQLOC);
- LOOP de leituras de variável (READV's), percorrendo toda a cadeia da chave recuperada no
- SEQRM;
- END.

Para a situação proposta o ideal seria, em termos de otimização do número de acessos físicos:

- mestre e variável em volumes diferentes
- cadeias do variável ordenadas, pelo RQLOC do mestre (ou seja, ordenadas pelo endereço relativo da chave pela qual as cadeias estão sendo lidas).

Dessa maneira, ao avanço progressivo e suave do braço do volume do mestre, corresponderá um avanço também suave e progressivo do braço do volume do variável (estamos supondo que não haja interferência externa).

Esse é pois o conceito de sincronismo mestre-variável. Depende, como visto, que o variável tenha sido carregado com seus regis - tros ordenados pelo RQLOC do mestre, (ou seja, que a chave de carga seja a desse mestre), e que, posteriormente, eventuais manutenções estruturais (ver item III.7), não tenham destruído essa ordenação inicial.

Caso o variável esteja vinculado a mais de um mestre, só uma chave será chave de carga (e só em relação a este mestre é que o variável estará sincronizado). O problema agora é escolher qual será a chave de carga. Apresentamos o critério mais utilizado e que é também o mais lógico.

- A sincronização de variável preso a mais de um mestre deve ser feita com o mestre pelo qual ele, variável, é pesquisado com mais frequência. Este é o critério principal e normalmente decisivo.

VIII.5 - Compartilhamento de Buffers

Como vimos nas definições de mestre e variável (itens IV.2 e IV.3 respectivamente), a cada arquivo está associado um nome de buffer que é então o buffer viável para aquele arquivo: toda operação de E/S do mesmo só pode ser feita utilizando uma das cópias homônimas do referido buffer.

Cabe agora a pergunta: Quantos e quais arquivos devem compartilhar um mesmo buffer?

Damos a seguir uma coleção de critérios que normalmente orientam essa decisão.

- Não se deve compartilhar buffer entre arquivos mestres e variáveis vinculados a ele, que sofram manutenção estrutural. Mesmo que a menutenção ocorra só no variável, acessos ao mestre se farão necessários para manutenção de vínculos.

- Não se deve compartilhar buffer entre arquivos que tenham seus registros processados simultaneamente por algum programa.

- Não se deve compartilhar buffer entre arquivos com fatores de bloco muito diferentes. O TOTAL exigirá que o buffer tenha o tamanho do bloco físico de maior tamanho.

- Deve-se criar múltiplas versões do Banco de Dados, diferindo

em nome (ver item IV.1) e principalmente em buffers, mas idênticas em tudo o mais. As diferentes combinações de buffers devem atender às diferentes necessidades (e probabilidades) de compartilhamento dos mesmos. Assim consegue-se otimizar a alocação de buffers para diversos programas em diferentes seções de TOTAL, somente alterando o JCL para pô-lo no ar.

- Quando no prólogo da definição de um Banco de Dados (item IV.1) for especificado o comando SHARE-IO, especial atenção deve ser dada à tentação de se especificar um n muito grande no comando IOAREA=XXXX=n (ver item IV.1), pois isso poderia causar um aumento inesperado (e indesejável) do número de movimentos físicos do braço do disco (onde devem estar os arquivos para os quais o buffer XXXX é viável).

O fenômeno é assim explicado: Se o número de tasks é pequeno, é bem provável que vários blocos físicos de um dos arquivos em questão estejam espalhados pelas múltiplas cópias homônimas do buffer viável para o mesmo. Supondo que seja um processamento do tipo lê - modifica na memória - grava, todas essas versões do buffer estão com a marca de "write pendente". Mas o grande número delas retarda sua liberação (com consequente gravação), possibilitando que nesse interim, ocorra algum evento que retire o braço do cilindro onde estavam os blocos físicos em questão. Certamente, quando afinal esses blocos forem devolvidos, modificados, ao arquivo, mais um movimento terá que ser feito pelo braço do disco, para posicioná-lo novamente no cilindro original.

IX - LIMITAÇÕES ATUAIS DO SIMULADOR

Dadas as limitações de tempo e exclusividade de dedicação a que esteve submetido o autor do presente trabalho durante a confecção do simulador de TOTAL, a versão atual do programa possui algumas limitações de menor porte, que serão apresentadas a seguir, ao lado de restrições do modelo resultantes de hipóteses simplificadoras assumidas.

IX.1 - Limitações Atuais (maio 1978)

Todas as eventuais ações de E/S das funções relacionadas no capítulo VI estão devidamente computadas, a menos daquelas resultantes de manutenção de vínculos e mesmo assim, só para as seguintes funções:

- ADDVA (item VI.3)
- ADDVB (item VI.4)
- ADDVR (item VI.6)

IX.2 - Hipóteses Simplificadoras

O modelo admite que em operações de manutenção estrutural em mestres, não existem E/S adicionais para manutenção de vínculos (raízes): a eventual cadeia de sinônimos estará toda contida no mesmo blobo físico.

O anexo 2 corrobora perfeitamente esta suposição.

Na TELERJ, o fator de carga (ou fator de ocupação) máximo dos arquivos mestres de produção está em torno de 60%, ao passo que o fator de bloco médio dos arquivos é superior a 20%.

Temos pois que o percentual médio do tamanho do arquivo que são sinônimos fora do bloco é essencialmente zero.

Na avaliação do parâmetro TOTAL DE TEMPO DE DESLOCAMENTO são usados valores médios tanto para o delay posicional bem como para a latência rotacional (rever estes conceitos no item VII.3).

Os valores foram tirados do Auerbach - 1972, para o modelo 3330 da IBM (disco magnético) e são respectivamente 30.0 e 8.4 mseg.

X - ANEXOS

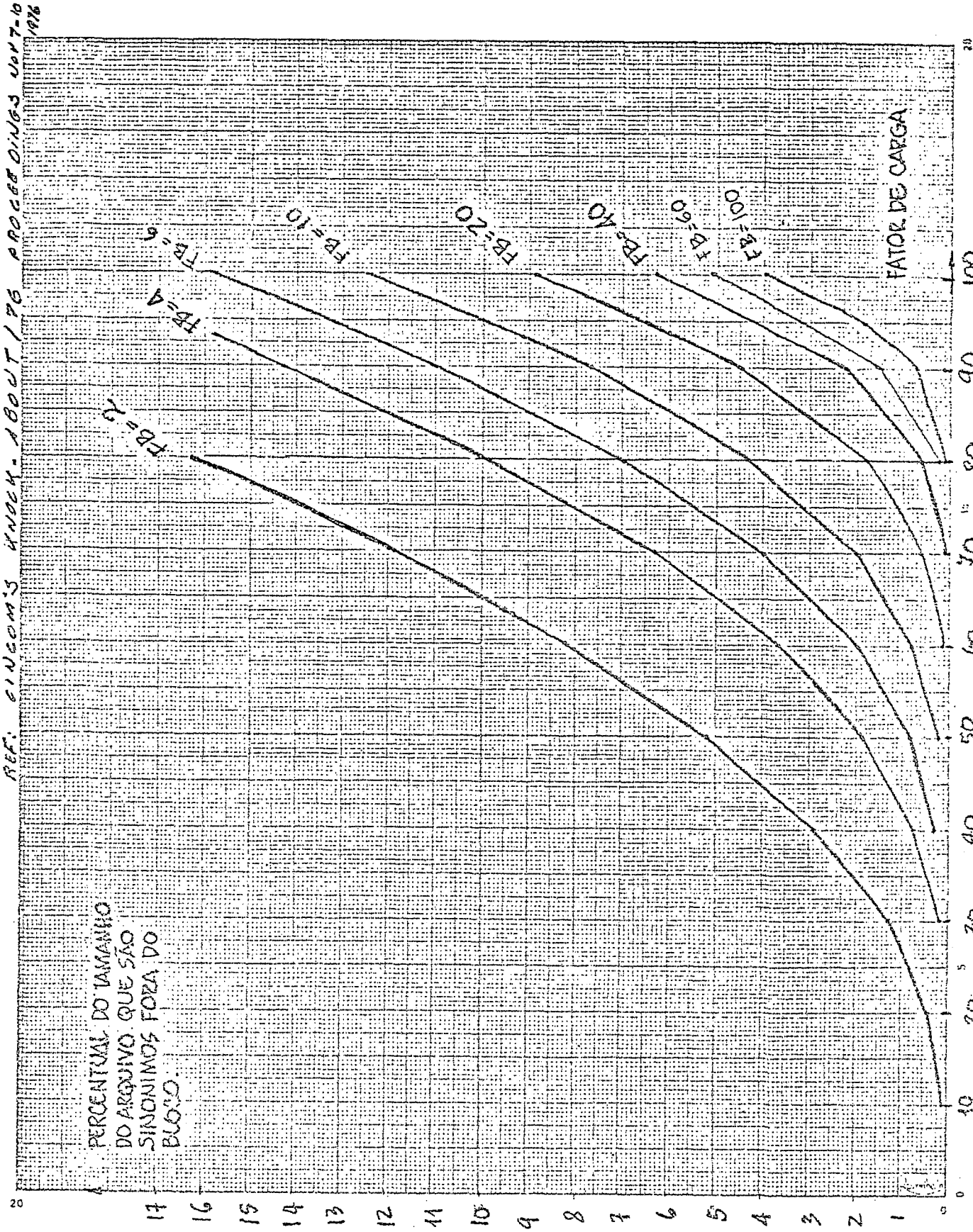
CHAVE	LINK	DADOS
501	37	38	----
123	39	43	----
913	42	44	----
739	45	45	----
700	47	46	----
487	48	51	----
108	50	54	----

ENDER. RELATIVO	CHAVE1	LINK1		CHAVE2	LINK2		DADOS
37	501	↻	38	ABC	↻	39	----
38	501	37	↻	BCD	↻	40	----
39	123	↻	40	ABC	37	48	----
40	123	39	41	BCD	38	45	----
41	123	40	43	EFG	↻	49	----
42	913	↻	44	EFG	49	↻	----
43	123	41	↻	XYZ	↻	54	----
44	913	42	↻	RST	↻	↻	----
45	739	↻	↻	BCD	40	50	----
46	700	47	↻	XYZ	54	↻	----
47	700	↻	46	MNO	↻	52	----
48	487	↻	49	ABC	39	↻	----
49	487	48	51	EFG	41	42	----
50	108	↻	52	BCD	45	↻	----
51	487	49	↻	MNO	52	↻	----
52	108	50	53	MNO	47	51	----
53	108	52	54	PQR	↻	↻	----
54	108	53	↻	XYZ	43	46	----

CHAVE	LINK		DADOS
XYZ	43	46	----
EFG	41	42	----
RST	44	44	----
BCD	38	50	----
MNO	47	51	----
ABC	37	48	----
PQR	53	53	----

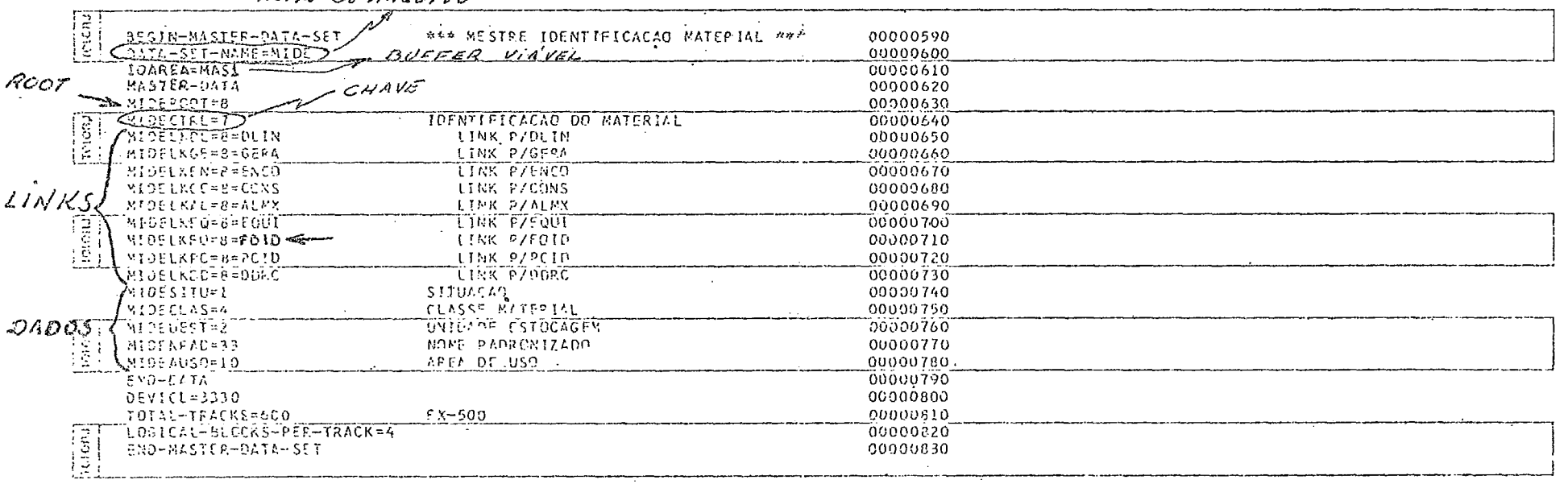
ANEXO 1

ANEXO 2


REF: EINSOM'S KNOCK-ABOUT / PG PROCEEDINGS VOL 7-10 1976
PERCENTUAL DO TAMANHO DO ARQUIVO QUE SÃO SINONIMOS FORA DO BLOCO.
FATOR DE CARGA
TB=10
TB=20
TB=40
TB=60
TB=100

ANEXO 3

```
BEGIN-DATA-BASE-GENERATION                                    00000240
DATA-BASE-NAME=ETRASE                                         00000250
OPTIONS VERSION=CENTRAL                                       00000260
    SHARE=10                                                  00000270
    IOAREA=MAS1=4                                             00000290
    IOAREA=MAT1=4                                             00000290
    IOAREA=VAR1=6                                             00000300
```



	GIVEN	CALCULATED	
TOTAL-LOGICAL-RECORDS	0	55200	*
LOGICAL-RECORD-LENGTH	0	137	*
LOGICAL-RECORDS-PER-BLOCK	0	23	*
LOGICAL-BLOCKS-PER-TRACK	4	4	
TOTAL-TRACKS	600	600	

VOL=SER=

TEL801

MESTRE DE ITENS DE MATERIAL

ANEXO 4

```
BSSIN-MASTER-DATA-SET          *** MESTRE DE FORNECEDORES ***          00000440
DATA-SET-NAME=MFOR                                                     00000450
  IOAREA=MAS1                                                          00000460
  MASTER-DATA                                                          00000470
  MFORROOT=8                                                           00000480
  MFORCIPL=5                   CODIGO DO FORNECEDOR                    00000490
  MFORLKFO=8=FOID                 LINK P/FOID                          00000500
  MFORLKPC=8=PCOD                 LINK P/PCOD                          00000510
  MFORRSOC=70                  RAZAO SOCIAL                            00000520
  MFORSITU=1                   SITUACAO                                00000530
  END-DATA                                                             00000540
  DEVICE=3330                                                          00000550
  TOTAL-TRACKS=60              FX-41                                   00000560
  LOGICAL-BLOCKS-PER-TRACK=4                                           00000570
  END-MASTER-DATA-SET                                                  00000580
```

	GIVEN	CALCULATED	
TOTAL-LOGICAL-RECORDS	0	7440	*
LOGICAL-RECORD-LENGTH	0	100	*
LOGICAL-RECORDS-PER-BLOCK	0	31	*
LOGICAL-BLOCKS-PER-TRACK	4	4	
TOTAL-TRACKS	60	60	

VOL=SER=

TELBDL

MESTRE DE FORNECEDORES

ANEXO 5

```
BEGIN-VARIABLE-ENTRY-DATA-SET                                   00000970
DATA-SET-NAME=FCID                                              00000980
SOAREA=VAR1                                                     00000990
BASE-DATA                                                       00001000
FOIONIDE=7=MIDECTRL                                             00001010
MEDELKFO=8                                                      00001020
FOIOFOSN=5=MFORCTPL                                             00001030
MFORELKFO=8                                                     00001040
FOIONPFC=18                                                     00001050
FOIOINDI=10                                                     00001060
FOIOIMDC=1                                                      00001070
FOIOINOS=1                                                      00001080
END-DATA                                                        00001090
DEVICE=3330                                                     00001100
TOTAL-TRACKS=290                                                00001110
LOGICAL-BLOCKS-PER-TRACK=6                                      00001120
END-VARIABLE-ENTRY-DATA-SET                                     00001130
```

PAR CHAVE-VÍNCULO

PAR CHAVE-VÍNCULO

DADOS

LAYOUT

	GIVEN	CALCULATED	
TOTAL-LOGICAL-RECORDS	0	50900	*
LOGICAL-RECORD-LENGTH	0	58	*
LOGICAL-RECORDS-PER-BLOCK	0	35	*
LOGICAL-BLOCKS-PER-TRACK	6	6	
TOTAL-TRACKS	290	290	
CYLINDER-LOAD-LIMIT	0	80	*

VOL=SER=

TEL802

VARIÁVEL DE RELACIONAMENTO

/

ITENS DE MATERIAL X FORNECEDORES

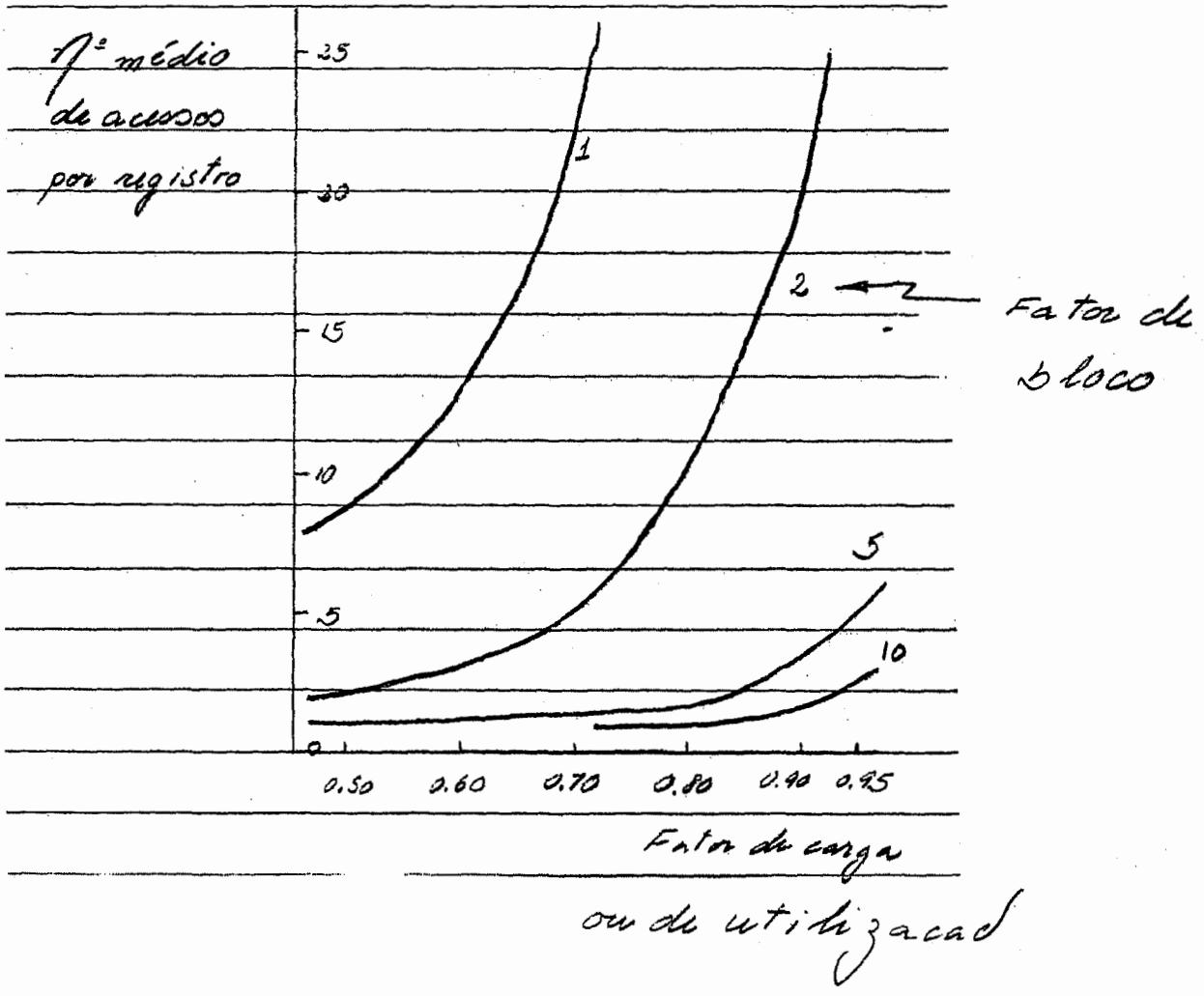
ANEXO 6

```
SEGIA-VARIABLE-ENTRY-DATA-SET
DATA-SET-NAME=IOPS      *** REF.CRUZADA ENTRE MSUB - MPGM - MEIO ***
ICTRBA=X'01'
BASE-DATA
IOPSCODS=2                      PARTE FIXA DO REGISTRO
IOPSMEIO=9=MEIOCTRL             CODIGO DA ENTRADA          (C)
MEIOLKIO=8                      LINK COM MEIO              (B)
IOPSDADO=16                     PARTE QUE SERA REDEFINIDA
RECORD-CODE=SU                  REDEFINICAO COM SUBST
IOPSMSUB=(6)                    CODIGO DO SUBSISTEMA       (C)
MSUBLKIO=(8)=MSUB               LINK COM MSUB              (B)
IOPSTPSU=(1)                    TIPO DO ELEMENTO (I/O/U)   (C)
IOPSAEST=(3)                    LIXO
RECORD-CODE=PG                  REDEFINICAO COM PROGRAMA
IOPSMPGM=(9)                    CODIGO DO PROGRAMA         (C)
MPGMLKIO=(8)=MPGM               LINK COM MPGM              (B)
IOPSTPPG=(1)                    TIPO DO ELEMENTO (I/O/U)   (C)
END-DATA
DEVICE=2330
TOTAL-TRACKS=1
LOGICAL-BLOCKS-PER-TRACK=6
END-VARIABLE-ENTRY-DATA-SET
```

	GIVEN	CALCULATED	
TOTAL-LOGICAL-RECORDS	0	330	*
LOGICAL-RECORD-LENGTH	0	37	*
LOGICAL-RECORDS-PER-BLOCK	0	55	*
LOGICAL-BLOCKS-PER-TRACK	6	6	
TOTAL-TRACKS	1	1	
CYLINDER-LOAD-LIMIT	0	80	*

ANEXO 7

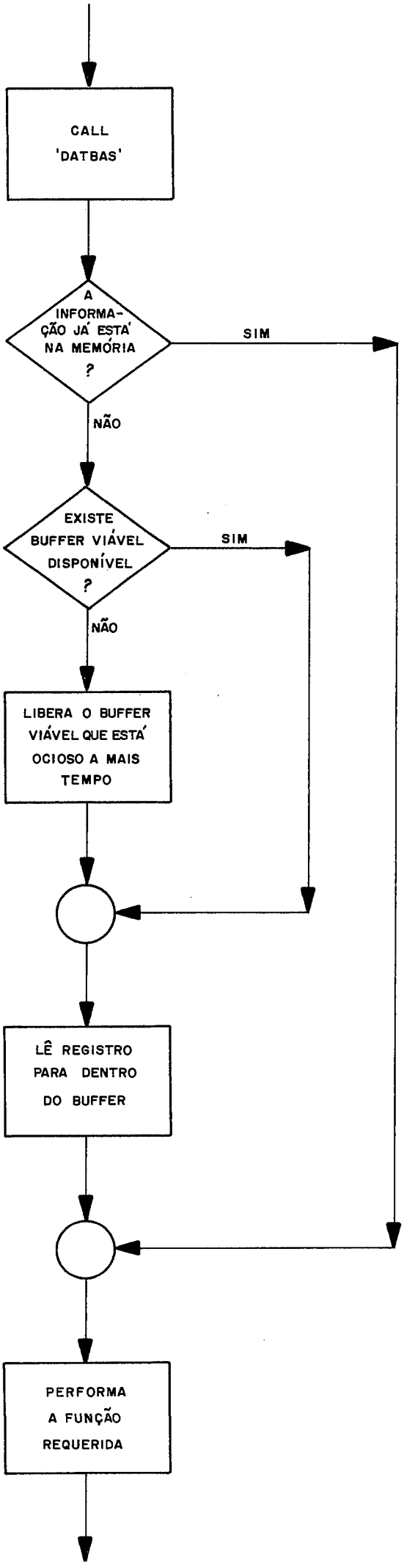
ANEXO 8



Curvas empíricas mostrando o nº médio de acessos por registro como função do fator de bloco e do fator de utilização

Ref ( ), fig 23-11

ANEXO 9


CALL
'DATBAS'
A INFORMAÇÃO JÁ ESTÁ NA MEMÓRIA ?
SIM
NÃO
EXISTE BUFFER VIÁVEL DISPONÍVEL ?
SIM
NÃO
LIBERA O BUFFER VIÁVEL QUE ESTÁ OCIOSO A MAIS TEMPO
LÊ REGISTRO PARA DENTRO DO BUFFER
PERFORMA A FUNÇÃO REQUERIDA

ANEXO 10

SISTEMA SIP (PESSOAL)

1 - Nome do programa - SIP1153Ø
Linguagem - PL/I
Arquivos do banco que acessa
MATR, PESS, MLOT
Número aproximado de registros acessados em cada arquivo
22000, 15000, 180
Frequência de rodada - 1 vez por mês
Breve descrição do pgm

```
DO WHILE (EXISTAM REGISTROS);
    LE MATR;
    SE SELECIONADO
        ENTÃO LE PESS;
            SE SELECIONADO
                ENTÃO LE MLOT;
                    GRAVA REGISTRO;
END;
SORT POR LOTAÇÃO;
IMPRIME RELATÓRIO;
```

Simultaneidade com outros programas

1. Programa simultâneo
SIP11XXØ
Arquivos acessados pelo programa simultâneo:
ABCD, EFGH
Probabilidade de simultaneidade (%): (*)
28,3
2. Programa simultâneo:
SIP11YYØ
Arquivos acessados pelo programa simultâneo
ABCD, MATR, WXYZ
Probabilidade de simultaneidade (%): (*)
41,9

(*) Fonte: Análise do log da console, ou análise dos relatórios do SMF, ou uso de algum utilitário, etc.

XI - O PROGRAMA SIMULADOR

SOURCE LISTING

```
STMT LEV NT

                    /* ANALISE E ESTATISTICAS - LOG DO TOTAL - MARTIN - MAR-77 */      0010

                    /*****************************************************************     0020
                     * ESTE PROGRAMA SE DESTINA A ANALISAR O LOG DO TOTAL, VI- *      0030
                     * SANDO OBTER ESTATISTICAS RELACIONADAS COM O MOVIMENTO - *      0040
                     * MECANICO DOS BRACOS DOS DISCOS QUE CONTENHAM ARQUIVOS - *      0050
                     * DO BANCO DE DADOS. PARA OS EFEITOS DESSE PROGRAMA, A A- *      0060
                     * LOCACAO DOS ARQUIVOS NOS DIVERSOS VOLUMES, A POSICAO - *       0070
                     * RELATIVA DOS SEUS 'EXTENTS', O NUMERO DE VOLUMES E DE - *      0080
                     * BUFFERS, O USO DESTES ULTIMOS E OS VINCULOS EXISTENTES *       0090
                     * ENTRE MESTRES E VARIAVEIS JA FORAM DEFINIDOS ANTERIOR - *      0100
                     * MENTE.                                          *              0110
                     *****************************************************************/    0120

  1      0  ABD3010..  PROC OPTIONS (MAIN),.                                          0130

  2   1  0             DCL LOG_DO_TOTAL_REDEF CHAR (648) VAR,.                       0140

  3   1  0             DCL 1 LOG_DO_TOTAL BASED (PLOG),                              0150
                             2 LL                              BIN FIXED,           0160
                             2 FUNCAO                           CHAR (8),           0170
                             2 TASK                             CHAR (8),           0180
                             2 NUMERO_DO_REGISTRO          BIN FIXED (31),          0190
                             2 NUMERO_DO_LOG               BIN FIXED (31),          0200
                             2 ARQUIVO                          CHAR (4),           0210
                             2 REFER_BIN                   BIN FIXED (31),          0220
                             2 LINKPATH_CHAR                    CHAR (8),           0230
                             2 STATUS_CHAR                      CHAR (4),           0240
                             2 TAMANHO_DO_REGISTRO         BIN FIXED (31),          0250
                             2 IMAGEM_DO_REGISTRO               CHAR (600),.        0260

  4   1  0             PLOG = ADDR (LOG_DO_TOTAL_REDEF),.                            0270

  5   1  0             DCL REFER_CHAR            CHAR (4) BASED (PREF),.             0280

  6   1  0             PREF = ADDR (REFER_BIN),.                                     0290

  7   1  0             DCL LINKPATH_CHAR_NA_IMAGEM              CHAR (8),.           0300

  8   1  0             DCL 1 LINKPATH_BIN_NA_IMAGEM      BASED (PLINK),              0310
                             2 LINK_ANTES_BIN              BIN FIXED (31),          0320
                             2 LINK_DEPOIS_BIN             BIN FIXED (31),.         0330

  9   1  0             PLINK = ADDR (LINKPATH_CHAR_NA_IMAGEM),.                      0340

 10   1  0             DCL STATUS_BIN  BIN FIXED (31) BASED (PSTATUS),.              0350

 11   1  0             PSTATUS = ADDR (STATUS_CHAR),.                                0360
```

```
STMT LEV NT

 12   1  0          DCL 1 DETALHES_DE_ALOCACAO (V) CTL,                          0370

                       2 ARQUIVO                    CHAR (4),                  0380
                       2 ORDEM                      PIC '99',                  0390
                       2 TOTAL                      PIC '99',                  0400
                       2 VOLUME                     CHAR (6),                  0410
                       2 CILINDRO_INICIAL           PIC '999',                 0420
                       2 TRILHA_INICIAL             PIC '99',                  0430
                       2 CILINDRO_FINAL             PIC '999',                 0440
                       2 TRILHA_FINAL               PIC '99',.                 0450

 13   1  0          DCL 1 ESPECIFICACAO_DE_VINCULOS (W) CTL,                     0460

                       2 NOME_DO_VARIAVEL           CHAR (4),                  0470
                       2 NOME_DO_MESTRE             CHAR (4),                  0480
                       2 ORDEM                      PIC '9',                   0490
                       2 TOTAL                      PIC '9',                   0500
                       2 LINKPATH                   CHAR (8),                  0510
                       2 INICIO_DA_CHAVE            PIC '999',                 0520
                       2 TAMANHO_MEDIO_DA_CADEIA    PIC '99V99',               0530
                       2 PRIMARIO_OU_SECUNDARIO     CHAR (1),.                 0540

                         /* 'P' - E' O LINK PRIMARIO DE CARREGAMENTO DO         0550
                                                              VARIAVEL */       0560
                         /* 'S' - E' UM LINK SECUNDARIO DE CARREGAMENTO DO      0570
                                                              VARIAVEL */       0580

 14   1  0          DCL 1 CONDICAO_ATUAL_DO_BRACO (X) CTL,                       0590

                       2 VOLUME                     CHAR (6),                  0600
                       2 CILINDRO                   PIC '999',.                0610

 15   1  0          DCL 1 CONDICAO_ATUAL_DO_BUFFER (Y)       CTL,                0620

                       2 BUFFER                     CHAR (4),                  0630
                       2 VERSAO                     PIC '99',                  0640
                       2 ARQUIVO                    CHAR (4),                  0650
                       2 VOLUME                     CHAR (6),                  0660
                       2 ENDERECO_INICIAL_BIN   BIN FIXED (31),                0670
                       2 ENDERECO_FINAL_BIN     BIN FIXED (31),                0680
                       2 BLOCO_FISICO               PIC '(5)9',                0690
                       2 TRILHA                     PIC '(4)9',                0700
                       2 CILINDRO                   PIC '999',                 0710
                       2 TEM_WRITE_PENDENTE         CHAR (1),                  0720
                       2 MOMENTO_DO_ULTIMO_USO      CHAR (15),.                0730

                         /*       DATE CAT TIME = YYMMDDHHMMSSTTT        */    0740
```

```
STMT LEV NT

  16   1  0          DCL 1 DETALHES_DE_ARQUIVOS (Z) CTL,                                  0750

                       2 ARQUIVO                        CHAR (4),                      0760
                       2 INDICACAO_M_OU_V               CHAR (1),                      0770
                       2 BUFFER                         CHAR (4),                      0780
                       2 TAMANHO_DA_CHAVE               PIC '99',                      0790
                         /*    ARQUIVO VARIAVEL - TAMANHO_DA_CHAVE = 00    */        0800
                       2 CAPACIDADE_REG_LOGICOS       PIC '(7)9',                      0810
                       2 COMPRIMENT_REG_LOGICO        PIC '(4)9',                      0820
                       2 REG_LOGICOS_POR_BLOCO        PIC '(3)9',                      0830
                       2 BLOCOS_FISICOS_POR_TRILHA      PIC '99',                      0840
                       2 TOTAL_DE_TRILHAS             PIC '(5)9',                      0850
                       2 FATOR_DE_UTILIZACAO            PIC '99',.                     0860

                         /* EXPRESSO EM PERCENTAGEM, VALIDO SOMENTE PARA               0870
                            ARQUIVOS MESTRE, SENDO ZERADO PARA VARIAVEIS   */         0880


  17   1  0          DCL STATUS_OK_BIN (5)          BIN FIXED (31) INIT                 0890
                            (0B, 1B, 1000B, 1111B, 10000B),.                          0900

                         /* STATUS DEVOLVIDOS NO LOG EQUIVALENTES A '****' */         0910


  18   1  0          DCL 1 VETOR_DE_STATUS_ACEITAVEIS (41),                            0920
                       2 MASCARA_DE_STATUS_ACEITAVEIS CHAR (8) INIT                    0930

                     ('        ','        ','        ','        ','MRNFNSMR',         0940
                      'MRNFNSMR','MRNFNSMR','MRNFNSMR','MRNF    ','*PNR    ',         0950
                      '*PNRMRNF','*PNRMRNF','****    ','****    ','****    ',         0960
                      '        ','        ','MRNF    ','        ','DUPM    ',         0970
                      'MRNF    ','        ','MRNF    ','        ','        ',         0980
                      '        ','        ','LOAD    ','        ','        ',         0990
                      '        ','        ','        ','        ','        ',         1000
                      '        ','        ','        ','        ','        ',         1010
                      '        ','        ','        ','        '),.                  1020

                         /* ESTES STATUS, SE DEVOLVIDOS PELAS RESPECTIVAS              1030
                            FUNCOES, INDICAM QUE APESAR DO INSUCESSO,   O              1040
                            I / O FOI PERFORMADO E DEVE SER COMPUTADO      */         1050


  19   1  0          DCL I_O                                CHAR (1),.                 1060

                     /* 'I' - A ROTINA DE PERFORMA_I_O TRARAH INFORMACOES DA REDE       1070
                            PARA O BUFFER                                             1080
                        'O' - A ROTINA DE PERFORMA_I_O LEVARAH INFORMACOES DO BUF-     1090
                            FER PARA A REDE                                  */       1100


  20   1  0          DCL 1 PARAMETROS,                                                 1110

                       2 ARQUIVO                        CHAR (4),                     1120
                       2 ENDERECO_CHAR                  CHAR (4),.                    1130
```

```
STMT LEV NT

 21   1  0          DCL 1 PARAMETROS_1 BASED (PPARM),                              1140

                       2 ARQUIVO                    CHAR (4),                    1150
                       2 ENDERECO_BIN          BIN FIXED (31),.                  1160

 22   1  0          PPARM = ADDR (PARAMETROS),.                                    1170

 23   1  0          DCL (CHFIMLOG,          /* 1 - ACABOU O ARQUIVO FITALOG    */ 1180
                         CHFIMALOC,        /* 1 - ACABOU O ARQUIVO ALOCA      */ 1190
                         CHFIMVINC,        /* 1 - ACABOU O ARQUIVO VINCULO    */ 1200
                         CHFIMVOL,         /* 1 - ACABOU O ARQUIVO VOLUMES    */ 1210
                         CHFIMBUF,         /* 1 - ACABOU O ARQUIVO BUFFERS    */ 1220
                         CHFIMARQ,         /* 1 - ACABOU O ARQUIVO ARQS       */ 1230
                         CHAVE_SINCRONISMO /* 1 - NA VARREDURA DE BUFFERS NAS    1240
                                              FUNCOES DE SINCRONISMO, FOI       1250
                                              ENCONTRADO ALGUM COM MARCA        1260
                                              DE 'WRITE PENDENTE'               1270
                                           2 - APROPRIANDO A FUNCAO EM SI  */   1280
                                         ) PIC '9' INIT (0),.                   1290

 24   1  0          DCL CHAVE_ESTATISTICAS        PIC '9' INIT (1),.            1300
                     /* 1 - VAI PERMITIR QUE A FUNCAO TENHA SUAS ESTATISTICAS    1310
                            DETALHADAS POR TASK, BUFFER, ARQUIVO E VOLUME  */    1320

 25   1  0          DCL MENOR_MOMENTO          CHAR (15),.                      1330

 26   1  0          DCL CILINDRO_ALVO          PIC '999' INIT (0),.             1340

 27   1  0          DCL (REGISTROS_POR_TRILHA,                                  1350
                         TRACK_INICIAL,                                         1360
                         TRACK_FINAL,                                           1370
                         ENDERECO_EM_TRILHAS,                                   1380
                         TAMANHO_DO_EXTENT_EM_TRILHAS,                          1390
                         TRILHAS_SOMADAS_DOS_EXTENTS                            1400
                                       ) PIC '(6)9',.                           1410

 28   1  0          DCL CHAVE_DO_ARQUIVO       CHAR (18) VAR,.                  1420
                     /* RECEBE O VALOR DA CHAVE DO MESTRE DE UM VARIAVEL NA      1430
                        FUNCAO SERAH_NECESSARIO_O_ACESSO E NAS ROTINAS DE        1440
                        MANUTENCAO_OU_SINCRONISMO                          */    1450

 29   1  0          DCL SEMENTE           BIN FIXED (31) INIT (75139),.          1460

 30   1  0          DCL (DATE, TIME, FLOOR, SUBSTR, PLIDUMP, ADDR, VERIFY       1470
                                                    ) BUILTIN,.                 1480

 31   1  0          DCL MASCARA_STATUS         CHAR (28) INIT                   1490
                                  ('ABCDEFGHIJKLMNOPQRSTUVWXYZ*.'),.            1500

 32   1  0          DCL (PFUNC, PLOG, PPARM, PREF, PLINK, PSTATUS) POINTER,.     1510

 33   1  0          DCL TAMANHO_VETOR_TASKS /* TAMANHO DO VETOR DE CONTADORES    1520
                                               POR TASK                  */     1530
                                    BIN FIXED (31) INIT (0),.                   1540
```

```
STMT LEV NT

                  /*************************************************************      1550
                   *                ACUMULADORES DE ESTATISTICAS                *      1560
                   *************************************************************/     1570

 34   1  0        DCL (TOTAL_FUNCOES_NAO_RECONHECIDAS,                                 1580
                       TOTAL_ARQS_NAO_RECONHECIDOS,                                   1590
                       TOTAL_VINCULOS_NAO_RECONHECIDOS,                               1600
                       TOTAL_BUFFERS_NAO_RECONHECIDOS,                                1610
                       TOTAL_VOLS_NAO_RECONHECIDOS,                                   1620
                       TOTAL_ERROS_EM_ENDERECOS                                       1630
                            ) BIN FIXED (31) INIT (0),.                               1640

 35   1  0        DCL 1 CONTADORES_POR_TASK (TAMANHO_VETOR_TASKS) CTL,                 1650

                      2 NOME_DA_TASK                          CHAR (8),               1660
                      2 TOTAL_REGISTROS                  BIN FIXED (31),              1670
                      2 TOTAL_ACESSOS                    BIN FIXED (31),              1680
                      2 TOTAL_DESLOCAMENTOS              BIN FIXED (31),              1690
                      2 TOTAL_NAO_DESLOCAMENTOS          BIN FIXED (31),              1700
                      2 TOTAL_ACESSOS_SEM_DESLOCAMENTO   BIN FIXED (31),              1710
                      2 TOTAL_ACESSOS_COM_DESLOCAMENTO   BIN FIXED (31),              1720
                      2 TOTAL_CILINDROS_DESLOCADOS       BIN FIXED (31),              1730
                      2 TOTAL_BYTES_TRANSMITIDOS         BIN FIXED (31),              1740
                      2 TOTAL_TEMPO_DE_DESLOCAMENTO      DEC FIXED (10,1),.           1750

 36   1  0        DCL 1 CONTADORES_POR_BUFFER (Y) CTL,                                1760

                      2 NOME_DO_BUFFER                        CHAR (4),               1770
                      2 VERSAO_DO_BUFFER                      PIC '99',               1780
                      2 TOTAL_REGISTROS                  BIN FIXED (31),              1790
                      2 TOTAL_ACESSOS                    BIN FIXED (31),              1800
                      2 TOTAL_DESLOCAMENTOS              BIN FIXED (31),              1810
                      2 TOTAL_NAO_DESLOCAMENTOS          BIN FIXED (31),              1820
                      2 TOTAL_ACESSOS_SEM_DESLOCAMENTO   BIN FIXED (31),              1830
                      2 TOTAL_ACESSOS_COM_DESLOCAMENTO   BIN FIXED (31),              1840
                      2 TOTAL_CILINDROS_DESLOCADOS       BIN FIXED (31),              1850
                      2 TOTAL_BYTES_TRANSMITIDOS         BIN FIXED (31),              1860
                      2 TOTAL_TEMPO_DE_DESLOCAMENTO      DEC FIXED (10,1),.           1870

 37   1  0        DCL 1 CONTADORES_POR_ARQUIVO (Z) CTL,                               1880

                      2 NOME_DO_ARQUIVO                       CHAR (4),               1890
                      2 TOTAL_REGISTROS                  BIN FIXED (31),              1900
                      2 TOTAL_ACESSOS                    BIN FIXED (31),              1910
                      2 TOTAL_DESLOCAMENTOS              BIN FIXED (31),              1920
                      2 TOTAL_NAO_DESLOCAMENTOS          BIN FIXED (31),              1930
                      2 TOTAL_ACESSOS_SEM_DESLOCAMENTO   BIN FIXED (31),              1940
                      2 TOTAL_ACESSOS_COM_DESLOCAMENTO   BIN FIXED (31),              1950
                      2 TOTAL_CILINDROS_DESLOCADOS       BIN FIXED (31),              1960
                      2 TOTAL_BYTES_TRANSMITIDOS         BIN FIXED (31),              1970
```

```
STMT LEV NT

                    2 TOTAL_TEMPO_DE_DESLOCAMENTO        DEC FIXED (10,1),.    1980

 38   1  0       DCL 1 CONTADORES_POR_VOLUME (X) CTL,                         1990

                    2 NOME_DO_VOLUME                          CHAR (6),       2000
                    2 TOTAL_REGISTROS                   BIN FIXED (31),       2010
                    2 TOTAL_ACESSOS                     BIN FIXED (31),       2020
                    2 TOTAL_DESLOCAMENTOS               BIN FIXED (31),       2030
                    2 TOTAL_NAO_DESLOCAMENTOS           BIN FIXED (31),       2040
                    2 TOTAL_ACESSOS_SEM_DESLOCAMENTO    BIN FIXED (31),       2050
                    2 TOTAL_ACESSOS_COM_DESLOCAMENTO    BIN FIXED (31),       2060
                    2 TOTAL_CILINDROS_DESLOCADOS        BIN FIXED (31),       2070
                    2 TOTAL_BYTES_TRANSMITIDOS          BIN FIXED (31),       2080
                    2 TOTAL_TEMPO_DE_DESLOCAMENTO       DEC FIXED (10,1),.    2090

 39   1  0       DCL 1 CONTADORES_GERAIS,                                     2100

                    2 TOTAL_REGISTROS                   BIN FIXED (31),       2110
                    2 TOTAL_ACESSOS                     BIN FIXED (31),       2120
                    2 TOTAL_DESLOCAMENTOS               BIN FIXED (31),       2130
                    2 TOTAL_NAO_DESLOCAMENTOS           BIN FIXED (31),       2140
                    2 TOTAL_ACESSOS_SEM_DESLOCAMENTO    BIN FIXED (31),       2150
                    2 TOTAL_ACESSOS_COM_DESLOCAMENTO    BIN FIXED (31),       2160
                    2 TOTAL_CILINDROS_DESLOCADOS        BIN FIXED (31),       2170
                    2 TOTAL_BYTES_TRANSMITIDOS          BIN FIXED (31),       2180
                    2 TOTAL_TEMPO_DE_DESLOCAMENTO       DEC FIXED (10,1),.    2190
```

```
STMT LEV NT

  40   1  0          DCL (A,      /* INCREMENTO NO TOTAL DE FUNCOES DO TIPO          2200
                                                          'DE ACESSO' */          2210
                          B,       /* INCREMENTO NO TOTAL DE DESLOCAMENTOS */     2220

                          C      /* INCREMENTO NO TOTAL DE CILINDROS DESLOCADOS */ 2230

                               ) BIN FIXED (31) INIT (0),.                       2240

  41   1  0          DCL (D       /* INCREMENTO NO TEMPO TOTAL DE DESLOCAMENTO */  2250

                               ) DEC FIXED (10,1) INIT (0),.                     2260

  42   1  0          DCL (E,      /* INCREMENTO NO TOTAL DE BYTES TRANSMITIDOS */  2270

                          F,       /* INDEXADOR QUE VARRE, PARA UM DADO VARIAVEL,  2280
                                      ESPECIFICACOES_DE_VINCULOS. USADO NA MANU-   2290
                                      TENCAO DE LINKS EM ADDS E DELETES.       */ 2300
                          G,       /* CONTADOR USADO NA APROPRIACAO DOS I_O FOR-   2310
                                      CADOS PELAS FUNCOES QUIET - ENDTO - CLOSP -  2320
                                      DEQUE                                    */ 2330
                          H,       /* RECEBE O MENOR MOMENTO NA ROTINA            2340
                                             DETERMINA_O_BUFFER                */ 2350
                          I,       /* APONTA EM VETOR_DE_FUNCOES PARA A FUNCAO    2360
                                      DO LOG DO TOTAL, POSICIONA TAMBEM -          2370
                                             - DETALHES_DE_FUNCOES                 2380
                                             - MANUTENCAO_OU_SINCRONISMO           2390
                                      USADO TAMBEM COMO CONTADOR NA ROTINA DE      2400
                                      IMPRESSAO DE TOTAIS FINAIS E NAS ROTINAS     2410
                                      DE INICIALIZACAO DOS VETORES INTERNOS    */ 2420
                          J,       /* APONTA EM CONDICAO_ATUAL_DO_BUFFER PARA O    2430
                                      BUFFER QUE RECEBERA' O REGISTRO DO TOTAL.    2440
                                      USADO TAMBEM COMO CONTADOR NA ROTINA DE      2450
                                      IMPRESSAO DE TOTAIS FINAIS.              */ 2460
                          K,       /* APONTA EM DETALHES_DE_ARQUIVOS PARA O AR-    2470
                                      QUIVO DO LOG DO TOTAL                    */ 2480
                          L,       /* APONTA EM CONDICAO_ATUAL_DO_BUFFER PARA O    2490
                                      ARQUIVO PASSADO COMO PARAMETRO DA PROCEDU-   2500
                                      RE ROTINA_SIM_OU_NAO                     */ 2510
                          M,       /* APONTA EM DETALHES_DE_ARQUIVOS PARA O AR-    2520
                                      QUIVO MESTRE PRESENTE NO LINKAGE PATH DO     2530
                                      VARIAVEL NO LOG DO TOTAL                 */ 2540
                          N,       /* APONTA EM ESPECIFICACAO_DE_VINCULOS PARA     2550
                                      O VINCULO MESTRE-VARIAVEL, QUANDO O LOG DO   2560
                                      TOTAL SE REFERE A UM VARIAVEL            */ 2570
                          O,       /* APONTA EM CONDICAO_ATUAL_DO_BRACO PARA O     2580
                                      VOLUME QUE CORRESPONDE AO QUE ESTA' NO -     2590
                                      BUFFER SELECIONADO                       */ 2600
                          P,       /* APONTA EM DETALHES_DE_ALOCACAO PARA O AR-    2610
                                      QUIVO REFERIDO AO CILINDRO_ALVO          */ 2620
                          Q,       /* APONTA EM DETALHES_DE_ARQUIVOS PARA O AR-    2630
                                      QUIVO QUE ESTA' NO BUFFER SELECIONADO PARA   2640
                                      SER LIBERADO MAS TEM UM WRITE PENDENTE   */ 2650
                          R,       /* USADO COMO CONTADOR NA FUNCAO STATUS_OK  */  2660
```

STMT LEV NT

```
S,       /* INCREMENTO NO TOTAL DE NAO-DESLOCAMENTOS */      2670
T,   /* INCREMENTO NO TOTAL ACESSOS COM DESLOCAMENTO */      2680
U,   /* INCREMENTO NO TOTAL ACESSOS SEM DESLOCAMENTO */      2690
V,          /* TAMANHO DO VETOR DETALHES_DE_ALOCACAO */      2700
W,     /* TAMANHO DO VETOR ESPECIFICACAO_DE_VINCULOS */      2710
X,       /* TAMANHO DO VETOR CONDICAO_ATUAL_DO_BRACO */      2720
Y,      /* TAMANHO DO VETOR CONDICAO_ATUAL_DO_BUFFER */      2730
Z,          /* TAMANHO DO VETOR DETALHES_DE_ARQUIVOS */      2740

GUARDA_P,         /* USADO PARA UM POSSIVEL REPOSICIO-       2750
                     NAMENTO DE P NA ROTINA CORRIGE_P */      2760
CHAVE_VINCULO,    /* POSICIONADO NAS ROTINAS DE MANU -       2770
                     TENCAO PARA INDICAR NA IMPLEMEN -       2780
                     TACAO DE PARAMETROS QUE O ARQUIVO       2790
                     NAO E' O DO LOG E SIM UM DOS MES-       2800
                     TRES ASSOCIADOS QUE ESTA' SOFREN-       2810
                     DO MANUTENCAO                */         2820
CHAVE_PENDENTE,   /* POSICIONADO NAS ROTINAS DE LIBE -       2830
                     RACAO DE BUFFERS PARA INDICAR NA        2840
                     IMPLEMENTACAO DE PARAMETROS QUE O       2850
                     ARQUIVO NAO E' O DO LOG E SIM UM        2860
                     CUJO BUFFER FOI LIBERADO E ESTA'        2870
                     SENDO SINCRONIZADO COM ELE   */         2880
INDICE,           /* USADO EM PERFORMA_I_O PARA RECE-        2890
                     BER O INDICE DO ARQUIVO, CONFOR-        2900
                     ME ESTE SEJA O DO LOG DO TOTAL ,        2910
                     UM QUE ESTA' SENDO MANTIDO OU QUE       2920
                     ESTA' SENDO SINCRONIZADO     */         2930
INDICE1,          /* RECEBE O INDICE DO BUFFER ASSOCI-       2940
                     ADO AO ARQUIVO APONTADO PELA VA -       2950
                     RIAVEL ANTERIOR              */         2960
CONTA_PAGINA,     /* CONTADOR DAS PAGINAS DE ESTATIS -       2970
                     TICAS FINAIS                 */         2980
IND_T,            /* INDICE DO VETOR DE CONTADORES POR       2990
                     TASK, USADO NA ROTINA POSICIONA_        3000
                     INDICES                      */         3010
IND_B,            /* INDICE DO VETOR DE CONTADORES POR       3020
                     BUFFER, USADO NA ROTINA POSICIONA_      3030
                     INDICES                      */         3040
IND_A,            /* INDICE DO VETOR DE CONTADORES POR       3050
                     ARQUIVO, USADO NA ROTINA POSICIONA_     3060
                     INDICES                      */         3070
IND_V,            /* INDICE DO VETOR DE CONTADORES POR       3080
                     VOLUME, USADO NA ROTINA POSICIONA_      3090
                     INDICES                      */         3100
TREG              /* INCREMENTO NO TOTAL DE REGISTROS */      3110
                  ) BIN FIXED (31) INIT (0),.                3120
```

```
STMT LEV NT

 43   1  0          DCL MANUTENCAO_OU_SINCRONISMO (41) ENTRY VARIABLE INIT          3130
                        (QUIET, LOGTR, MARKL, QMARK, ADDVC,                       3140
                         ADDVB, ADDVA, ADDVR, DELVD, READD,                       3150
                         READR, READV, SERLV, SEQRV, SEQWV,                       3160
                         WRITV, WRITD, READM, SEQRM, ADD_M,                       3170
                         DEL_M, RQLOC, WRITM, OPENM, OPENV,                       3180
                         CLOSM, CLOSV, CLOSP, FREEF, FREEX,                       3190
                         ENDTO, ENDLG, TOTAL, QUEST, DEQUE,                       3200
                         RESTM, RESTV,                                            3210
                         SIGNON, SIGNOFF, RECORDB, RECORDA)                       3220

 44   1  0          DCL VETOR_DE_FUNCOES (41) CHAR (8) INIT                        3230
                    ('QUIET   ','LOGTR   ','MARKL   ','QMARK   ','ADDVC   ',     3240
                     'ADDVB   ','ADDVA   ','ADDVR   ','DELVD   ','READD   ',     3250
                     'READR   ','READV   ','SERLV   ','SEQRV   ','SEQWV   ',     3260
                     'WRITV   ','WRITD   ','READM   ','SEQRM   ','ADD-M   ',     3270
                     'DEL-M   ','RQLOC   ','WRITM   ','OPENM   ','OPENV   ',     3280
                     'CLOSM   ','CLOSV   ','CLOSP   ','FREEF   ','FREEX   ',     3290
                     'ENDTO   ','ENDLG   ','TOTAL   ','QUEST   ','DEQUE   ',     3300
                     'RESTM   ','RESTV   ',                                       3310
                     'SIGNON  ','SIGNOFF ','RECORDB ','RECORDA '),.               3320

 45   1  0          DCL 1 DETALHES_DE_FUNCOES (41) BASED (PFUNC),                  3330
                        2 IMPLICA_EM_ACESSO              CHAR (1),                3340
                          /* S - SIM        N - NAO                        */     3350
                        2 LEITURA_OU_GRAVACAO            CHAR (1),                3360
                          /* L - LEITURA    G - GRAVACAO    I - IRRELEVANTE */    3370
                        2 TIPO_DE_GRAVACAO               CHAR (1),                3380
                          /* W - TIPO 'WRITL'   M - TIPO 'ADD' OU 'DELETE'       3390
                             N - EVENTUALMENTE (VAI DEPENDER EXCLUSIVAMENTE       3400
                                 DO VALOR DA CHAVE SINCRONISMO) NAO PERMITE       3410
                                 NAS ESTATISTICAS DETALHAMENTO POR TASK, BUF-     3420
                                 FER, ARQUIVO E VOLUME, SENDO A FUNCAO COMPU-     3430
                                 TADA SOMENTE NOS CONTADORES GERAIS.       */     3440
                        2 MESTRE_OU_VARIAVEL             CHAR (1),                3450
                          /* M - MESTRE     V - VARIAVEL    I - IRRELEVANTE */    3460
                        2 FUNCAO_FORCA_I_O               CHAR (1),.               3470
                          /* S - SIM (QUIET, ENDTO E DEQUE - TODOS OS BUF-  */    3480
                          /*                          FERS COM WRITE PENDENTE  */ 3490
                          /*          CLOSP - TODAS AS VERSOES DO BUFFER DO */    3500
                          /*                  ARQUIVO COM WRITE PENDENTE)   */    3510
                          /* N - NAO (TODAS AS FUNCOES RESTANTES)          */    3520

 46   1  0          DCL DETALHES_DE_FUNCOES_REDEF (41) CHAR (5) INIT               3530
                    ('SINIS','  N  ','  N  ','  N  ','SGMVN',                     3540
                     'SGMVN','SGMVN','SGMVN','SGMVN','SL VN',                     3550
                     'SL VN','SL VN','SL VN','SL VN','SGWVN',                     3560
                     'SGWVN','SGWVN','SL MN','SL MN','SGMMN',                     3570
                     'SGMMN','  N  ','SGWMN','  N  ','  N  ',                     3580
                     '  N  ','  N  ','SI IS','  N  ','  N  ',                     3590
                     'SINIS','  N  ','  N  ','  N  ','SINIS',                     3600
                     '  N  ','  N  ',                                           3610
                     '  N  ','  N  ','  N  ','  N  '),.                         3620
```

```
STMT LEV NT

 47   1  0          PFLAG = ADDR (DETALHES_DE_FUNCOES_REDEF) ;.                    3630

 48   1  0          DCL (FITALOG, ALOCA, VINCULO, VOLUMES, BUFFERS, ARQS           3640
                                                  ) FILE RECORD INPUT,            3650
                        SYSIN FILE ,                                              3660
                        SYSPRINT FILE PRINT ;.                                    3670

 49   1  0          DCL ENTRADA CHAR (80) INIT (' '),.      /* AREA GERAL DE      3680
                                                              L E I T U R A */    3690

                    %INCLUDE ABDROT1;.*****************************************  3700
```

```
STMT LEV NT

                        /*              *****   ABORTI   ****                    */ 00000010
                        /* TRATAMENTO DE ERRO PARA PROGRAMAS QUE USAM O -TOTAL      00000020

                          NIRILO P LOPES  MAIO DE 1974                            00000030
                          ALTERADA EM OUT 1976 - SUZANA                           */ 00000040

 50   1  0              DCL 1 TERRO,                                                00000050
                            2 T1 CHAR(28) INIT (' ERRO DE -TOTAL- NO ARQUIVO '),  00000060
                            2 TARQ CHAR(4),                                       00000070
                            2 T2 CHAR(11) INIT (' NA FUNCAO '),                   00000080
                            2 TFUNC CHAR(5),                                      00000090
                            2 T3 CHAR(15) INIT ('  COM O STATUS '),               00000100
                            2 TSTATUS CHAR(4),                                    00000110
                            2 T4 CHAR(22) INIT (' NO REGISTRO DE CHAVE '),        00000120
                            2 TCHAVE CHAR(22) VAR INIT (' '),                     00000130
                          TERRO1 CHAR(67) BASED (TPT),                            00000140
                          TERRO2 CHAR(20) INIT(' '),                              00000150
                          NOME_TASK CHAR(8),                                      00000160
                          LISTA_DE_ARQUIVOS CHAR(96) VAR,                         00000165
                          TOTALPR FILE STREAM OUTPUT,                             00000170
                          TCHERRO BIT(1) INIT ('0'B),,                            00000180
 51   1  0              DCL DATBAS EXTERNAL ENTRY OPTIONS(ASSEMBLER),,            00000190


                        /* ERRO DE EXECUCAO DO PROGRAMA PL/I     */                00000200
 52   1  0              ON ERROR BEGIN,,                                          00000210
 53   2  0                      IF TCHERRO = '1'B                                 00000220
                               THEN DO,.     /* HOUVE ERRO ANTERIOR */           00000230
 54   2  1                           PUT FILE(TOTALPR) EDIT('*** - ATENCAO - ',   00000240
                    'TERMINO ANORMAL DE  PROCESSAMENTO - ***') (SKIP(20),2 A),,   00000250
 55   2  1                           CALL PLIDUMP,.                               00000260
 56   2  1                           STOP ,.                                      00000270
 57   2  1                           END,.                                        00000280
 58   2  0                      ELSE DO,.                                         00000290
 59   2  1                           TCHERRO = '1'B,.                             00000300
 60   2  1                           CALL ESTAT,.    /* ESTATISTICA DE REGISTROS */ 00000310
 61   2  1                           TERRO2 = ' ERRO DE PL/I',.                   00000320
 62   2  1  /* IMPR. ERRO */         PUT FILE(TOTALPR) EDIT (TERRO2) (SKIP(15),A),. 00000330
 63   2  1  /*  FIM  */             CALL ABDFIM(NOME_TASK,TCHERRO),.             00000340
 64   2  1                           STOP,.                                       00000350
 65   2  1                           END,.                                        00000360
 66   2  0                      END ,.                                            00000370

 67   1  0              TP1 = ADDR(TERRO),.                                       00000380
```

```
STMT LEV NT

                        /* GRAVACAO DOS BUFFERS DO -TOTAL- PARA PROGRAMAS  */          00000390
                        /* QUE ATUALIZAM OS ARQUIVOS                       */          00000400
  68   1  0  TGRBUF.. PROC,.                                                          00000410
  69   2  0           DCL TCON  CHAR(4),                                             00000420
                           TCON1 BIN FIXED (31,0) BASED (TPTCON),.                   00000430
  70   2  0           TPTCON = ADDR(TCON),.                                          00000440
  71   2  0           TCON1 = 0,.                                                    00000450
  72   2  0           CALL DATBAS('QUIET',TSTATUS,TCON,'END.'),.                     00000460
  73   2  0           IF TSTATUS NE '****'                                           00000470
                      THEN DO,.                                                     00000480
  74   2  1                TARQ = ' ',.                                             00000490
  75   2  1                TFUNC = 'QUIET',.                                        00000500
  76   2  1                CALL TIMPER,.                                            00000510
  77   2  1                END,.                                                    00000520
  78   2  0           END TGRBUF,.                                                   00000530


                     /* IMPRESSAO DE MENSAGEM DE ERRO DE -TOTAL-     */            00000540
  79   1  0  TIMPER.. PROC,.                                                         00000550
  80   2  0           TCHERRO = '1'B,.                                             00000560
  81   2  0           CALL ESTAT,.            /* ESTATISTICA DE REGISTROS */ .     00000570
  82   2  0           PUT FILE(TOTALPR) EDIT (TERRO1,T4,TCHAVE) (SKIP(15),3 A),.   00000580
  83   2  0           CALL  ABOFIM(NOME_TASK,TCHERRO),. /* ENCERRAMENTO  */        00000590
  84   2  0           END TIMPER,.                                                 00000600
                  ****************************                                      3700

                    %INCLUDE ABOROT2,.*********************************************    3710
```

```
STMT LEV NT

                    /*              ****   ABDROT2   ****              */                00000010
                    /*  ROTINAS GERAIS PARA PROGRAMAS QUE USAM TOTAL  */                00000020

                    /*       ROTINA DE INICIO DE TASK                 */                00000030
                    /*       E  ABERTURA DE ARQUIVOS                  */                00000040
                    /*                                SUZANA - NOV 1976*/                00000050
 85   1  0    ABDINI..                                                                   00000060
                 PROC(NOME_TASK,LISTA_DE_ARQUIVOS,TCHERRO),.                           00000070

                     /* ESTA ROTINA ABRE ATEH 24 ARQUIVOS */                           00000080
                     /*    SIMULTANEAMENTE               */                           00000090

 86   2  0           DCL (LENGTH,MOD,SUBSTR) BUILTIN,                                  00000100

                     ENDP CHAR(4) INIT('END.'),                                        00000110
                     ASTER CHAR(4) INIT('****'),                                       00000120
                     STATUS CHAR(4) INIT('XXXX'),                                      00000130
                     NOME_TASK CHAR(8),                                                00000140
                     LISTA_DE_ARQUIVOS CHAR(96) VAR,                                   00000150
                     TCHERRO BIT(1),                                                   00000160

                     COMPRIMENTO     DEC FIXED(2),                                     00000170
                     CONT            DEC FIXED(2) INIT(1),                             00000180
                     ARQ             CHAR(4),.                                         00000190

 87   2  0           PUT FILE(TOTALPR) EDIT(' TELERJ   -     ',                        00000200
                     'UTILITARIOS DA ADMINISTRACAO DE BANCO DE DADOS   ')              00000210
                     (SKIP,A,A),.                                                      00000220
 88   2  0           PUT FILE(TOTALPR) EDIT(' ABD8000L -     ',                        00000230
                     'MENSAGENS DO TOTAL       ')                                      00000240
                     (SKIP,A,A),.                                                      00000250
 89   2  0           CALL DATBAS('TOTAL',STATUS,'XXXXXX',NOME_TASK,ENDP),.             00000260
 90   2  0           PUT FILE(TOTALPR) EDIT('STATUS NO SIGNON = ',STATUS)              00000270
                                  (SKIP(4),COL(20),A,A),.                            00000280
 91   2  0           IF STATUS NE ASTER                                                00000290
                     THEN DO,.                                                         00000300
 92   2  1                TCHERRO='1'B,.                                               00000310
 93   2  1                CALL ABDFIM(NOME_TASK,TCHERRO),.                             00000320
 94   2  1                END,.                                                        00000330
 95   2  0           COMPRIMENTO=LENGTH(LISTA_DE_ARQUIVOS),.                           00000340

 96   2  0           IF MOD(COMPRIMENTO,4) NE 0                                        00000350
                     THEN DO,.                                                         00000360
 97   2  1                PUT FILE(TOTALPR) EDIT('LISTA DE ARQUIVOS INVALIDA',         00000370
                                  LISTA_DE_ARQUIVOS) (SKIP(4),A,X(2),A),.            00000380
 98   2  1                TCHERRO='1'B,.                                               00000390
 99   2  1                CALL ABDFIM(NOME_TASK,TCHERRO),.                             00000400
100   2  1                END,.                                                        00000410

101   2  0    ABRE.. DO WHILE(CONT LT COMPRIMENTO),.                                   00000420
```

```
STMT LEV NT

 102   2  1          ARQ=SUBSTR(LISTA_DE_ARQUIVOS,CONT,4),.                    00000430
 103   2  1          CALL DATBAS('OPENM',STATUS,ARQ,ENDP),.                    00000440
 104   2  1          PUT FILE(TOTALPR) EDIT('STATUS NO OPEN DE',ARQ,' = ',     00000450
                          STATUS)(COL(20),A,X(1),A,A,A),.                     00000460
 105   2  1          IF STATUS NE ASTER                                        00000470
                       THEN DO,.                                              00000480
 106   2  2             TCHERRO='1'B,.                                         00000490
 107   2  2             CALL ABDFIM(NOME_TASK,TCHERRO),.                       00000500
 108   2  2             END,.                                                  00000510
 109   2  1          CONT=CONT+4,.                                             00000520
 110   2  1          END ABRE,.                                                00000530
 111   2  0          TCHERRO='0'B,.                                            00000540
 112   2  0          END ABDINI,.                                              00000550
```

```
STMT LEV NT

                    /*  ROTINA DE FIM DE TASK TOTAL  */                          00000560
  113   1  0  ABDFIM,,                                                           00000570
               PROC(NOME_TASK,TCHERRO),.                                        00000580

  114   2  0       DCL ENDP CHAR(4) INIT('END.'),                               00000590
                    STATUS CHAR(4) INIT('XXXX'),                                00000600
                    TCHERRO BIT(1),                                             00000610
                    NOME_TASK CHAR(8),.                                         00000620
  115   2  0       IF TCHERRO='0'B                                              00000630
                   THEN CALL ESTAT,.                                            00000640
  116   2  0       CALL DATBAS('DEQUE',STATUS,NOME_TASK,ENDP),.                 00000650
  117   2  0       PUT FILE(TOTALPR) SKIP EDIT('STATUS NO DEQUE = ',            00000660
                                                STATUS)(COL(20),(2)A),.         00000670
  118   2  0       IF STATUS NE '****'                                          00000680
                   THEN TCHERRO='1'B,.                                          00000690
  119   2  0       IF TCHERRO='1'B                                              00000700
                   THEN DO,.                                                    00000710
  120   2  1            PUT FILE(TOTALPR) EDIT('*** -ATENCAO - ',               00000720
       'TERMINO ANORMAL DO PROCESSAMENTO - ***') (SKIP(20),2 A),.               00000730
  121   2  1            CALL PLIDUMP,.                                          00000740
  122   2  1            STOP,.                                                  00000750
  123   2  1            END,.                                                   00000760
  124   2  0       ELSE DO,.                                                    00000770
  125   2  1            PUT FILE(TOTALPR) EDIT('*** -TERMINO NORMAL ',          00000780
                       'DO PROCESSAMENTO - ***')(SKIP(20),2 A),.                00000790
  126   2  1            END,.                                                   00000800
  127   2  0  END ABDFIM,.                                                      00000810
           ***************************                                          3710
```

```
STMT LEV NT

                        /*****************************************************      3720
                        *                  INICIO DO PROGRAMA                *      3730
                        *****************************************************/     3740

 128   1  0             ON ENDFILE (FITALOG) CHFIMLOG  = 1,.                       3750
 129   1  0             ON ENDFILE (ALOCA)   CHFIMALOC = 1,.                       3760
 130   1  0             ON ENDFILE (VINCULO) CHFIMVINC = 1,.                       3770
 131   1  0             ON ENDFILE (VOLUMES) CHFIMVOL  = 1,.                       3780
 132   1  0             ON ENDFILE (BUFFERS) CHFIMBUF  = 1,.                       3790
 133   1  0             ON ENDFILE (ARQS)    CHFIMARQ  = 1,.                       3800

 134   1  0             OPEN FILE (FITALOG),                                       3810
                             FILE (ALOCA)  ,                                       3820
                             FILE (VINCULO),                                       3830
                             FILE (VOLUMES),                                       3840
                             FILE (BUFFERS),                                       3850
                             FILE (ARQS)   ,                                       3860
                             FILE (SYSPRINT) LINESIZE (132),.                      3870

 135   1  0             GET SKIP EDIT (V,       /* NUMERO TOTAL DE EXTENTS      */ 3880
                                      W,       /* NUMERO DE LINKS MESTRE-VARIAVEL */ 3890
                                      X,       /* NUMERO DE VOLUMES DO BANCO   */ 3900
                                      Y,       /* NUMERO DE BUFFERS UTILIZADOS */ 3910
                                      Z,       /* NUMERO DE ARQUIVOS DO BANCO  */ 3920
                                      TAMANHO_VETOR_TASKS)                        3930
                                      /* NUMERO DE TASKS (ESTIMADO)   */          3940
                                                 ((6) F(5)),.                     3950

 136   1  0             CALL ABDINI ('ABD30100','',TCHERRO),.                      3960

 137   1  0             ALLOCATE DETALHES_DE_ALOCACAO,                             3970
                                ESPECIFICACAO_DE_VINCULOS,                        3980
                                CONDICAO_ATUAL_DO_BRACO,                          3990
                                CONDICAO_ATUAL_DO_BUFFER,                         4000
                                DETALHES_DE_ARQUIVOS,                             4010
                                CONTADORES_POR_TASK,                              4020
                                CONTADORES_POR_BUFFER,                            4030
                                CONTADORES_POR_ARQUIVO,                           4040
                                CONTADORES_POR_VOLUME,.                           4050
```

```
STMT LEV NT

                        /*************************************************************     4060
                        *                LEITURA DOS DADOS DE ALOCACAO                *     4070
                        *************************************************************/    4080

138   1  0              DO I = 1 TO V WHILE (CHFIMALOC = 0),.                              4090

139   1  I              READ FILE (ALOCA) INTO (ENTRADA),.                                 4100
140   1  1              DETALHES_DE_ALOCACAO.ARQUIVO                          (I) =        4110
                                                   SUBSTR (ENTRADA,  1, 4),.              4120
141   1  1              DETALHES_DE_ALOCACAO.ORDEM                            (I) =        4130
                                                   SUBSTR (ENTRADA,  5, 2),.              4140
142   1  1              DETALHES_DE_ALOCACAO.TOTAL                            (I) =        4150
                                                   SUBSTR (ENTRADA,  7, 2),.              4160
143   1  1              DETALHES_DE_ALOCACAO.VOLUME                           (I) =        4170
                                                   SUBSTR (ENTRADA,  9, 6),.              4180
144   1  1              DETALHES_DE_ALOCACAO.CILINDRO_INICIAL                 (I) =        4190
                                                   SUBSTR (ENTRADA, 15, 3),.              4200
145   1  1              DETALHES_DE_ALOCACAO.TRILHA_INICIAL                   (I) =        4210
                                                   SUBSTR (ENTRADA, 18, 2),.              4220
146   1  1              DETALHES_DE_ALOCACAO.CILINDRO_FINAL                   (I) =        4230
                                                   SUBSTR (ENTRADA, 20, 3),.              4240
147   1  1              DETALHES_DE_ALOCACAO.TRILHA_FINAL                     (I) =        4250
                                                   SUBSTR (ENTRADA, 23, 2),.              4260
148   1  1              END,.                                                              4270

                        /*************************************************************     4280
                        *                LEITURA DOS DADOS DE VINCULOS                *     4290
                        *************************************************************/    4300

149   1  0              DO I = 1 TO W WHILE (CHFIMVINC = 0),.                              4310

150   1  1              READ FILE (VINCULO) INTO (ENTRADA),.                               4320

151   1  1              ESPECIFICACAO_DE_VINCULOS.NOME_DO_VARIAVEL            (I) =        4330
                                                   SUBSTR (ENTRADA,  1, 4),.              4340
152   1  1              ESPECIFICACAO_DE_VINCULOS.NOME_DO_MESTRE              (I) =        4350
                                                   SUBSTR (ENTRADA,  5, 4),.              4360
153   1  1              ESPECIFICACAO_DE_VINCULOS.ORDEM                       (I) =        4370
                                                   SUBSTR (ENTRADA,  9, 1),.              4380
154   1  1              ESPECIFICACAO_DE_VINCULOS.TOTAL                       (I) =        4390
                                                   SUBSTR (ENTRADA, 10, 1),.              4400
155   1  1              ESPECIFICACAO_DE_VINCULOS.LINKPATH                    (I) =        4410
                                                   SUBSTR (ENTRADA, 11, 8),.              4420
156   1  1              ESPECIFICACAO_DE_VINCULOS.INICIO_DA_CHAVE             (I) =        4430
                                                   SUBSTR (ENTRADA, 19, 3),.              4440
157   1  1              ESPECIFICACAO_DE_VINCULOS.TAMANHO_MEDIO_DA_CADEIA     (I) =        4450
                                                   SUBSTR (ENTRADA, 22, 5),.              4460
158   1  1              ESPECIFICACAO_DE_VINCULOS.PRIMARIO_OU_SECUNDARIO      (I) =        4470
                                                   SUBSTR (ENTRADA, 27, 1),.              4480
159   1  1              END,.                                                              4490
```

```
STMT LEV NT

                    /*************************************************************      4500
                    *                LEITURA DOS DADOS DE VOLUMES                 *      4510
                    *************************************************************/     4520

160   1  0          DO I = 1 TO X WHILE (CHFIMVOL = 0),.                              4530

161   1  1          READ FILE (VOLUMES) INTO (ENTRADA),.                              4540

162   1  1          CONDICAO_ATUAL_DO_BRACO.VOLUME                        (I) =       4550
                                                  SUBSTR (ENTRADA,  1, 6),.           4560
163   1  1          CONDICAO_ATUAL_DO_BRACO.CILINDRO                   (I) = 0,.      4570

164   1  1          END,.                                                             4580

                    /*************************************************************      4590
                    *                LEITURA DOS DADOS DE BUFFERS                 *      4600
                    *************************************************************/     4610

165   1  0          DO I = 1 TO Y WHILE (CHFIMBUF = 0),.                              4620

166   1  1          READ FILE (BUFFERS) INTO (ENTRADA),.                              4630

167   1  1          CONDICAO_ATUAL_DO_BUFFER.BUFFER                       (I) =       4640
                                                  SUBSTR (ENTRADA,  1, 4),.           4650
168   1  1          CONDICAO_ATUAL_DO_BUFFER.VERSAO                       (I) =       4660
                                                  SUBSTR (ENTRADA,  5, 2),.           4670
169   1  1          CONDICAO_ATUAL_DO_BUFFER.ARQUIVO                      (I),        4680

                    CONDICAO_ATUAL_DO_BUFFER.VOLUME                       (I),        4690

                    CONDICAO_ATUAL_DO_BUFFER.TEM_WRITE_PENDENTE           (I),        4700

                    CONDICAO_ATUAL_DO_BUFFER.MOMENTO_DO_ULTIMO_USO  (I) = ' ',.       4710

170   1  1          CONDICAO_ATUAL_DO_BUFFER.ENDERECO_INICIAL_BIN         (I),        4720

                    CONDICAO_ATUAL_DO_BUFFER.ENDERECO_FINAL_BIN           (I),        4730

                    CONDICAO_ATUAL_DO_BUFFER.BLOCO_FISICO                 (I),        4740

                    CONDICAO_ATUAL_DO_BUFFER.TRILHA                       (I),        4750

                    CONDICAO_ATUAL_DO_BUFFER.CILINDRO                  (I) = 0 ,.     4760

171   1  1          END,.                                                             4770
```

```
STMT LEV NT

                        /*************************************************************     4780
                        *                LEITURA DOS DADOS DE ARQUIVOS                *     4790
                        *************************************************************/    4800

 172   1  0             DO I = 1 TO Z WHILE (CHFIMARQ = 0),.                              4810

 173   1  1             READ FILE (ARCS) INTO (ENTRADA),.                                 4820

 174   1  1             DETALHES_DE_ARQUIVOS.ARQUIVO                          (I) =       4830
                                                    SUBSTR (ENTRADA,  1, 4),.            4840
 175   1  1             DETALHES_DE_ARQUIVOS.INDICACAO_M_OU_V                 (I) =       4850
                                                    SUBSTR (ENTRADA,  5, 1),.            4860
 176   1  1             DETALHES_DE_ARQUIVOS.BUFFER                           (I) =       4870
                                                    SUBSTR (ENTRADA,  6, 4),.            4880
 177   1  1             DETALHES_DE_ARQUIVOS.TAMANHO_DA_CHAVE                 (I) =       4890
                                                    SUBSTR (ENTRADA, 10, 2),.            4900
 178   1  1             DETALHES_DE_ARQUIVOS.CAPACIDADE_REG_LOGICOS           (I) =       4910
                                                    SUBSTR (ENTRADA, 12, 7),.            4920
 179   1  1             DETALHES_DE_ARQUIVOS.COMPRIMENT_REG_LOGICO            (I) =       4930
                                                    SUBSTR (ENTRADA, 19, 4),.            4940
 180   1  1             DETALHES_DE_ARQUIVOS.REG_LOGICOS_POR_BLOCO            (I) =       4950
                                                    SUBSTR (ENTRADA, 23, 3),.            4960
 181   1  1             DETALHES_DE_ARQUIVOS.BLOCOS_FISICOS_POR_TRILHA        (I) =       4970
                                                    SUBSTR (ENTRADA, 26, 2),.            4980
 182   1  1             DETALHES_DE_ARQUIVOS.TOTAL_DE_TRILHAS                 (I) =       4990
                                                    SUBSTR (ENTRADA, 28, 5),.            5000
 183   1  1             DETALHES_DE_ARQUIVOS.FATOR_DE_UTILIZACAO              (I) =       5010
                                                    SUBSTR (ENTRADA, 33, 2),.            5020
 184   1  1             END,.                                                             5030
```

```
STMT LEV NT

                        /*************************************************************      5040
                         *     INICIALIZACAO DOS ACUMULADORES DE ESTATISTICAS         *      5050
                         *************************************************************/     5060

  185   1  0            CONTADORES_POR_TASK    (*) = 0;.                                    5070
  186   1  0            DO I = 1 TO TAMANHO_VETOR_TASKS;.                                   5080
  187   1  1            CONTADORES_POR_TASK.NOME_DA_TASK (I) = '        ';.                 5090
  188   1  1            END;.                                                               5100
  189   1  0            CONTADORES_POR_BUFFER  (*) = 0;.                                    5110
  190   1  0            DO I = 1 TO Y;.                                                     5120
  191   1  1            CONTADORES_POR_BUFFER.NOME_DO_BUFFER (I) = '    ';.                 5130
  192   1  1            END;.                                                               5140
  193   1  0            CONTADORES_POR_ARQUIVO (*) = 0;.                                    5150
  194   1  0            DO I = 1 TO Z;.                                                     5160
  195   1  1            CONTADORES_POR_ARQUIVO.NOME_DO_ARQUIVO (I) = '    ';.               5170
  196   1  1            END;.                                                               5180
  197   1  0            CONTADORES_POR_VOLUME  (*) = 0;.                                    5190
  198   1  0            DO I = 1 TO X;.                                                     5200
  199   1  1            CONTADORES_POR_VOLUME.NOME_DO_VOLUME (I) = '      ';.               5210
  200   1  1            END;.                                                               5220
  201   1  0            CONTADORES_GERAIS          = 0;.                                    5230
```

```
STMT LEV NT

                        /*************************************************************      5240
                         *         ROTINAS ESPECIFICAS DAS DIVERSAS FUNCOES          *      5250
                         *************************************************************/     5260

                        /*************************************************************      5270
                         *                      Q   U   I   E   T                    *      5280
                         *************************************************************/     5290

 202   1  0  QUIET..    PROC,.                                                               5300

 203   2  0             A, TREG = 0,.   /* PARA NAO ATRAPALHAR EVENTUAL LIBERACAO */         5310

 204   2  0             DO G = 1 TO Y,.                                                      5320

 205   2  1             IF TEM_WRITE_PENDENTE (G) = 'S'                                      5330
                        THEN DO,.                                                           5340

 206   2  2                  B = B + 1,.  /* INCREMENTO NO TOTAL DE DESLOCAMENTOS */          5350
 207   2  2                  IF CHAVE_SINCRONISMO = 0                                       5360
                             THEN CHAVE_SINCRONISMO = 1,.                                   5370
 208   2  2                  CALL LIBERA_BUFFER (G),.                                       5380
 209   2  2                  END,.                                                          5390

 210   2  1             END,.                                                                5400

 211   2  0             A, TREG = 1,. /* PARA REPOR O INCREMENTO NO VALOR INICIAL */         5410

 212   2  0             IF CHAVE_SINCRONISMO = 1                                           5420
                        THEN T = T + 1,.  /* INCREMENTO ACESSOS COM DESLOCAMENTO  */        5430

 213   2  0             ELSE DO,.                                                          5440

 214   2  1                  S = S + 1,.  /* INCREMENTO TOTAL NAO-DESLOCAMENTOS   */        5450
 215   2  1                  U = U + 1,.  /* INCREMENTO ACESSOS SEM DESLOCAMENTO  */        5460
 216   2  1                  END,.                                                          5470

 217   2  0             IF CHAVE_SINCRONISMO = 1       /* 0, APROPRIA SO' NOS GE- */     5480
                        THEN CHAVE_SINCRONISMO = 2,.   /* RAIS, 1, GERAIS E TASK. */     5490

 218   2  0             CALL ESTATISTICAS,.                                                5500

 219   2  0             CHAVE_SINCRONISMO = 0,.                                            5510

 220   2  0  END QUIET,.                                                                    5520
```

```
STMT LEV NT

                        /*************************************************************      5530
                        *                        A   D   D   V   C                  *      5540
                        *************************************************************/     5550

 221   1  0  ADDVC..    PROC,.                                                               5560

 222   2  0              DCL RESPOSTA CHAR (3),.                                            5570

                       /* ACHANDO O VARIAVEL NO VETOR DE ESPECIFICACAO DE VINCULOS */       5580

 223   2  0              DO N = 1 TO W WHILE (LOG_DO_TOTAL.ARQUIVO NE                       5590
                              ESPECIFICACAO_DE_VINCULOS.NOME_DO_VARIAVEL (N)),.           5600
 224   2  1              END,.                                                              5610

 225   2  0              IF N = W + 1                                                       5620
                        THEN DO,.    /* NAO ACHOU O VARIAVEL NO VETOR DE VINCULOS */       5630

 226   2  1                   PUT SKIP EDIT ('VARIAVEL SEM ENTRADA NO VETOR DE ESPEC'      5640
                                    CAT 'IFICACAO DE VINCULOS - '                         5650
                                    CAT LOG_DO_TOTAL.ARQUIVO)          (A),.              5660
 227   2  1                   TOTAL_VINCULOS_NAO_RECONHECIDOS =                          5670
                                    TOTAL_VINCULOS_NAO_RECONHECIDOS + 1,.                 5680
 228   2  1                   CALL PLIDUMP,.                                             5690
 229   2  1                   CALL ESTAT,.                                               5700
 230   2  1                   STOP,.                                                     5710
 231   2  1                   END,.                                                      5720

 232   2  0              ELSE DO,.         /* ACHOU O PRIMEIRO VINCULO DO VARIAVEL */     5730

 233   2  1                   DO F = (N - ESPECIFICACAO_DE_VINCULOS.ORDEM (N) + 1) TO     5740
                                  (N - ESPECIFICACAO_DE_VINCULOS.ORDEM (N) +              5750
                                       ESPECIFICACAO_DE_VINCULOS.TOTAL (N)),.             5760

                       /* SEMPRE HAVERA' MANUTENCAO DE MESTRE, QUER SEJA PRIMEIRO DA      5770
                          CADEIA OU NAO. VERIFIQUE A VERACIDADE DESTA AFIRMACAO.    */     5780

 234   2  2                   PARAMETROS.ARQUIVO = NOME_DO_MESTRE (F),.                  5790

 235   2  2                   DO M = 1 TO Z WHILE (PARAMETROS.ARQUIVO NE                 5800
                                    DETALHES_DE_ARQUIVOS.ARQUIVO (M)),.                   5810
 236   2  3                   END,.                                                      5820

 237   2  2                   IF M = Z + 1                                               5830
                              THEN DO,.                                                  5840

 238   2  3                        PUT SKIP EDIT ('MESTRE NO LINK DO VARIAVEL NAO RE'    5850
                                       CAT 'CONHECIDO - ' CAT LOG_DO_TOTAL.ARQUIVO      5860
                                       CAT PARAMETROS.ARQUIVO)              (A),.        5870
 239   2  3                        TOTAL_ARQS_NAO_RECONHECIDOS =                         5880
                                       TOTAL_ARQS_NAO_RECONHECIDOS + 1,.                  5890
 240   2  3                        CALL PLIDUMP,.                                        5900
 241   2  3                        CALL ESTAT,.                                          5910
 242   2  3                        STOP,.                                                5920
 243   2  3                        END,.                                                 5930
```

```
STMT LEV NT

 244   2  2              ELSE DO;.                                                5940

 245   2  3                  CHAVE_DO_ARQUIVO = SUBSTR (IMAGEM_DO_REGISTRO,       5950
                                INICIO_DA_CHAVE (F), TAMANHO_DA_CHAVE (M));..   5960
 246   2  3                  CALL CALCULA_RQLOC (PARAMETROS.ARQUIVO,              5970
                               CHAVE_DO_ARQUIVO, PARAMETROS_1.ENDERECO_BIN);..   5980
 247   2  3                  CALL ROTINA_SIM_OU_NAO (PARAMETROS.ARQUIVO;          5990
                                 PARAMETROS_1.ENDERECO_BIN, RESPOSTA);..        6000
 248   2  3                 IF RESPOSTA = 'SIM'                                   6010
                           THEN DO;.                                             6020

 249   2  4                      I = 18;.              /* FORCAMOS UM READM */    6030
 250   2  4                      B = B + 1;.     /* TOTAL DESLOCAMENTOS + 1 */    6040
 251   2  4                      CHAVE_VINCULO = 1;.                            6050
 252   2  4                      CALL IMPLEMENTACAO_DE_PARAMETROS;.             6060
 253   2  4                      TEM_WRITE_PENDENTE (J) = 'S';.                 6070
 254   2  4                      CHAVE_VINCULO = 0;.                            6080
 255   2  4                      END;.                                          6090

 256   2  3                  ELSE DO;.                                          6100

 257   2  4                      I = 18;.                                       6110
 258   2  4                      S = S + 1;. /* TOTAL_NAO_DESLOCAMENTOS + 1 */  6120
 259   2  4                      INDICE = M;.                                   6130
 260   2  4                      INDICE1 = L;.                                  6140
 261   2  4                      CALL POSICIONA_P;.                             6150
 262   2  4                      CALL CORRIGE_P;.                               6160
 263   2  4                      CALL ATUALIZA_BUFFER;.                         6170
 264   2  4                      TEM_WRITE_PENDENTE (L) = 'S';.                 6180
 265   2  4                      END;.                                          6190
 266   2  3                  END;.                                              6200

                        /* EM ADDVC SO' HA' MANUTENCAO DE LINK DO PROPRIO VARIAVEL  6210
                              SE A ADICAO NAO FOR A PRIMEIRA DA CADEIA       */  6220

 267   2  2                IF SUBSTR (REFER_CHAR, 1, 2) NE 'LK'                   6230
                           THEN DO;./* TEMOS QUE SIMULAR O ENDERECO DO ANTERIOR */ 6240

 268   2  3                  CALL GERA_ENDERECO_RELATIVO (K, M, F, ENDERECO_BIN);.. 6250
 269   2  3                    PARAMETROS.ARQUIVO = NOME_DO_VARIAVEL (F);.          6260
 270   2  3                    CALL ROTINA_SIM_OU_NAO (PARAMETROS.ARQUIVO;          6270
                                                      ENDERECO_BIN, RESPOSTA);..  6280
 271   2  3                  IF RESPOSTA = 'SIM'                                  6290
                             THEN DO;.                                          6300

 272   2  4                      I = 12;.              /* FORCAMOS UM READV */    6310

 273   2  4                      B = B + 1;.     /* TOTAL DESLOCAMENTOS + 1 */    6320
 274   2  4                      CALL IMPLEMENTACAO_DE_PARAMETROS;.             6330
 275   2  4                      TEM_WRITE_PENDENTE (J) = 'S';.                 6340
 276   2  4                      END;.                                          6350

 277   2  3                  ELSE DO;.                                          6360
```

```
STMT LEV NT

 278   2  4                          I = 12,.                                    6370
 279   2  4                          S = S + 1,. /* TOTAL_NAO_DESLOCAMENTOS + 1 */ 6380
 280   2  4                          INDICE = K,.                                6390
 281   2  4                          INDICE1 = L,.                               6400
 282   2  4                          CALL POSICIONA_P,.                          6410
 283   2  4                          CALL CORRIGE_P,.                            6420
 284   2  4                          CALL ATUALIZA_BUFFER,.                      6430
 285   2  4                          TEM_WRITE_PENDENTE (L) = 'S',.              6440
 286   2  4                          END,.                                       6450
 287   2  3                       END,.                                          6460
 288   2  2                    END,.                                             6470
 289   2  1                    END,.                                             6480

 290   2  0  END ADDVC,.                                                        6490
```

```
STMT LEV NT

                        /*************************************************************      6500
                         *                        D   E   L   V   D                  *      6510
                         *************************************************************/     6520

 291   1  0  DELVD..     PROC,.                                                               6530

 292   2  0              DCL RESPOSTA CHAR (3),.                                              6540

                       /* ACHANDO O VARIAVEL NO VETOR DE ESPECIFICACAO DE VINCULOS */        6550

 293   2  0              DO N = 1 TO W WHILE (LOG_DO_TOTAL.ARQUIVO NE                         6560
                                 ESPECIFICACAO_DE_VINCULOS.NOME_DO_VARIAVEL (N)),.          6570
 294   2  1              END,.                                                                6580

 295   2  0              IF N = W + 1                                                         6590
                        THEN DO,.    /* NAO ACHOU O VARIAVEL NO VETOR DE VINCULOS */         6600

 296   2  1                   PUT SKIP EDIT ('VARIAVEL SEM ENTRADA NO VETOR DE ESPEC'       6610
                                      CAT 'IFICACAO DE VINCULOS - '                         6620
                                      CAT LOG_DO_TOTAL.ARQUIVO)          (A),.              6630
 297   2  1                   TOTAL_VINCULOS_NAO_RECONHECIDOS =                            6640
                                        TOTAL_VINCULOS_NAO_RECONHECIDOS + 1,.               6650
 298   2  1                   CALL PLIDUMP,.                                               6660
 299   2  1                   CALL ESTAT,.                                                 6670
 300   2  1                   STOP,.                                                       6680
 301   2  1                   END,.                                                        6690

 302   2  0              ELSE DO,.          /* ACHOU O PRIMEIRO VINCULO DO VARIAVEL */       6700

 303   2  1                   DO F = (N - ESPECIFICACAO_DE_VINCULOS.ORDEM (N) + 1) TO       6710
                                 (N - ESPECIFICACAO_DE_VINCULOS.ORDEM (N) +                 6720
                                      ESPECIFICACAO_DE_VINCULOS.TOTAL (N)),.                6730

                       /* SO HAVERA' MANUTENCAO DE MESTRE, SE O REGISTRO DELETADO FOR      6740
                          O PRIMEIRO OU O  ULTIMO DA CADEIA. VERIFIQUE A VERACIDADE.*/       6750

 304   2  2                   PARAMETROS.ARQUIVO = NOME_DO_MESTRE (F),.                    6760

 305   2  2                   DO M = 1 TO Z WHILE (PARAMETROS.ARQUIVO NE                   6770
                                        DETALHES_DE_ARQUIVOS.ARQUIVO (M)),.                 6780
 306   2  3                   END,.                                                        6790

 307   2  2                   IF M = Z + 1                                                 6800
                              THEN DO,.                                                    6810

 308   2  3                        PUT SKIP EDIT ('MESTRE NO LINK DO VARIAVEL NAO RE'      6820
                                     CAT 'CONHECIDO - ' CAT LOG_DO_TOTAL.ARQUIVO           6830
                                     CAT PARAMETROS.ARQUIVO)                 (A),.         6840
 309   2  3                        TOTAL_ARQS_NAO_RECONHECIDOS =                          6850
                                           TOTAL_ARQS_NAO_RECONHECIDOS + 1,.              6860
 310   2  3                        CALL PLIDUMP,.                                         6870
 311   2  3                        CALL ESTAT,.                                           6880
 312   2  3                        STOP,.                                                 6890
 313   2  3                        END,.                                                  6900
```

```
STMT LEV NT

 314   2  2              ELSE DO,.                                                  6910

 315   2  3                   IF SUBSTR (IMAGEM_DO_REGISTRO, INICIO_DA_CHAVE (F)    6920
                           + TAMANHO_DA_CHAVE (M),     4) = '    '                  6930
                           OR SUBSTR (IMAGEM_DO_REGISTRO, INICIO_DA_CHAVE (F)    6940
                           + TAMANHO_DA_CHAVE (M) + 4, 4) = '    '                6950
                           THEN DO,.                                              6960

 316   2  4                      CHAVE_DO_ARQUIVO =                               6970
                                SUBSTR (IMAGEM_DO_REGISTRO,                     6980
                                INICIO_DA_CHAVE (F), TAMANHO_DA_CHAVE (M)),.    6990
 317   2  4                      CALL CALCULA_RQLOC (PARAMETROS.ARQUIVO,          7000
                                CHAVE_DO_ARQUIVO,PARAMETROS_1.ENDERECO_BIN),.   7010
 318   2  4                      CALL ROTINA_SIM_OU_NAO (PARAMETROS.ARQUIVO,      7020
                                      PARAMETROS_1.ENDERECO_BIN, RESPOSTA),.    7030
 319   2  4                      IF RESPOSTA = 'SIM'                              7040
                                THEN DO,.                                       7050

 320   2  5                           I = 18,.          /* FORCAMOS UM READM */  7060
 321   2  5                           B = B + 1,./* TOTAL DESLOCAMENTOS + 1 */   7070
 322   2  5                           CHAVE_VINCULO = 1,.                        7080
 323   2  5                           CALL IMPLEMENTACAO_DE_PARAMETROS,.         7090
 324   2  5                           TEM_WRITE_PENDENTE (J) = 'S',.             7100
 325   2  5                           CHAVE_VINCULO = 0,.                        7110
 326   2  5                           END,.                                      7120

 327   2  4                      ELSE DO,.                                        7130

 328   2  5                           I = 18,.                                   7140
 329   2  5                           S = S + 1,.  /* NAO_DESLOCAMENTOS + 1 */   7150
 330   2  5                           INDICE = M,.                               7160
 331   2  5                           INDICE1 = L,.                              7170
 332   2  5                           CALL POSICIONA_P,.                         7180
 333   2  5                           CALL CORRIGE_P,.                           7190
 334   2  5                           CALL ATUALIZA_BUFFER,.                     7200
 335   2  5                           TEM_WRITE_PENDENTE (L) = 'S',.             7210
 336   2  5                           END,.                                      7220
 337   2  4                      END,.                                           7230
 338   2  3                 END,.                                                7240

                        /* EM DELVD SO' HA' MANUTENCAO DE LINK DO PROPRIO VARIAVEL   7250
                           NA DIRECAO EM QUE NAO HOUVER '    ' NO LINKKAGE PATH.  */ 7260

 339   2  2                 LINKPATH_CHAR_NA_IMAGEM = SUBSTR (IMAGEM_DO_REGISTRO,  7270
                           INICIO_DA_CHAVE (F) + TAMANHO_DA_CHAVE (M), 8),.       7280

 340   2  2                 IF SUBSTR (LINKPATH_CHAR_NA_IMAGEM, 1, 4) NE '    '    7290
                           THEN DO,.          /* MANUTENCAO DA CADEIA A MONTANTE */ 7300

 341   2  3                      ENDERECO_BIN = LINK_ANTES_BIN,.                   7310
 342   2  3                      PARAMETROS.ARQUIVO = NOME_DO_VARIAVEL (F),.       7320
 343   2  3                      CALL ROTINA_SIM_OU_NAO (PARAMETROS.ARQUIVO,       7330
                                                  ENDERECO_BIN, RESPOSTA),.     7340
 344   2  3                      IF RESPOSTA = 'SIM'                               7350
```

```
STMT LEV NT

                              THEN DO,.                                          7360

 345   2  4                        I = 12,.              /* FORCAMOS UM READV */  7370
 346   2  4                        B = B + 1,.     /* TOTAL DESLOCAMENTOS + 1 */  7380
 347   2  4                        CALL IMPLEMENTACAO_DE_PARAMETROS,.             7390
 348   2  4                        TEM_WRITE_PENDENTE (J) = 'S',.                 7400
 349   2  4                        END,.                                          7410

 350   2  3                    ELSE DO,.                                          7420

 351   2  4                        I = 12,.                                       7430
 352   2  4                        S = S + 1,. /* TOTAL_NAO_DESLOCAMENTOS + 1 */  7440
 353   2  4                        INDICE = K,.                                   7450
 354   2  4                        INDICE1 = L,.                                  7460
 355   2  4                        CALL POSICIONA_P,.                             7470
 356   2  4                        CALL CORRIGE_P,.                               7480
 357   2  4                        CALL ATUALIZA_BUFFER,.                         7490
 358   2  4                        TEM_WRITE_PENDENTE (L) = 'S',.                 7500
 359   2  4                        END,.                                          7510
 360   2  3                    END,.                                              7520

 361   2  2                IF SUBSTR (LINKPATH_CHAR_NA_IMAGEM, 5, 4) NE '    '    7530
                           THEN DO,.          /* MANUTENCAO DA CADEIA A JUSANTE */ 7540

 362   2  3                    ENDERECO_BIN = LINK_DEPOIS_BIN,.                   7550
 363   2  3                    PARAMETROS.ARQUIVO = NOME_DO_VARIAVEL (F),.        7560
 364   2  3                    CALL ROTINA_SIM_OU_NAO (PARAMETROS.ARQUIVO,        7570
                                                      ENDERECO_BIN, RESPOSTA),.  7580
 365   2  3                    IF RESPOSTA = 'SIM'                                7590
                               THEN DO,.                                          7600

 366   2  4                        I = 12,.              /* FORCAMOS UM READV */  7610
 367   2  4                        B = B + 1,.     /* TOTAL DESLOCAMENTOS + 1 */  7620
 368   2  4                        CALL IMPLEMENTACAO_DE_PARAMETROS,.             7630
 369   2  4                        TEM_WRITE_PENDENTE (J) = 'S',.                 7640
 370   2  4                        END,.                                          7650

 371   2  3                    ELSE DO,.                                          7660

 372   2  4                        I = 12,.                                       7670
 373   2  4                        S = S + 1,. /* TOTAL_NAO_DESLOCAMENTOS + 1 */  7680
 374   2  4                        INDICE = K,.                                   7690
 375   2  4                        INDICE1 = L,.                                  7700
 376   2  4                        CALL POSICIONA_P,.                             7710
 377   2  4                        CALL CORRIGE_P,.                               7720
 378   2  4                        CALL ATUALIZA_BUFFER,.                         7730
 379   2  4                        TEM_WRITE_PENDENTE (L) = 'S',.                 7740
 380   2  4                        END,.                                          7750
 381   2  3                    END,.                                              7760
 382   2  2                END,.                                                  7770
 383   2  1                END,.                                                  7780

 384   2  0  END DELVD,.                                                         7790
```

```
STMT LEV NT

                        /*************************************************************      7800
                        *                        C   L   O   S   P                  *      7810
                        *************************************************************/     7820

385   1   0  CLOSP..    PROC,.                                                               7830

386   2   0            A, TREG = 0,.   /* PARA NAO ATRAPALHAR EVENTUAL LIBERACAO */          7840

387   2   0            DO Q = 1 TO Z WHILE (DETALHES_DE_ARQUIVOS.ARQUIVO (Q)                7850
                                                 NE LOG_DO_TOTAL.ARQUIVO),.                 7860
388   2   1            END,.                                                                7870

389   2   0            DO G = 1 TO Y,.                                                      7880

390   2   1            IF ((CONDICAO_ATUAL_DO_BUFFER.ARQUIVO (G) =                          7890
                           LOG_DO_TOTAL.ARQUIVO) AND TEM_WRITE_PENDENTE (G) = 'S')        7900
                        THEN DO,.                                                           7910

391   2   2                 IF CHAVE_SINCRONISMO = 0                                       7920
                            THEN CHAVE_SINCRONISMO = 1,.                                    7930
392   2   2                 I_O = 'O',.                                                    7940
393   2   2                 CHAVE_PENDENTE = 1,.                                           7950
394   2   2                 INDICE1 = G,.                                                  7960
395   2   2                 B = B + 1,.  /* INCREMENTO NO TOTAL DE DESLOCAMENTOS */        7970
396   2   2                 CALL CALCULA_CILINDRO_ALVO (G, CILINDRO_ALVO),.                7980
397   2   2                 CALL PERFORMA_I_O (G, CILINDRO_ALVO),.                         7990
398   2   2                 CALL ESTATISTICAS,.                                            8000

                          /* A FUNCAO CLOSP FAZ COM QUE O TOTAL NAO TENHA MAIS A-          8010
                             CESSO A'S INFORMACOES DO BUFFER (POR SINAL E' A UNI-          8020
                             CA FUNCAO DE SINCRONIZACAO COM ESTA CARACTERISTICA).          8030
                             LOGO, BRANQUEIA-SE TUDO...                          */       8040

399   2   2                 CONDICAO_ATUAL_DO_BUFFER.ARQUIVO (G),                          8050
                            CONDICAO_ATUAL_DO_BUFFER.VOLUME (G),                           8060
                            CONDICAO_ATUAL_DO_BUFFER.TEM_WRITE_PENDENTE (G),               8070
                            CONDICAO_ATUAL_DO_BUFFER.MOMENTO_DO_ULTIMO_USO (G) =           8080
                                                                          ' ',.            8090
400   2   2                 CONDICAO_ATUAL_DO_BUFFER.ENDERECO_INICIAL_BIN (G),             8100
                            CONDICAO_ATUAL_DO_BUFFER.ENDERECO_FINAL_BIN (G),               8110
                            CONDICAO_ATUAL_DO_BUFFER.BLOCO_FISICO (G),                     8120
                            CONDICAO_ATUAL_DO_BUFFER.TRILHA (G),                           8130
                            CONDICAO_ATUAL_DO_BUFFER.CILINDRO (G) = 0,.                    8140

401   2   2                 END,.                                                          8150
402   2   1            END,.                                                               8160

403   2   0            A, TREG = 1,. /* PARA REPOR O INCREMENTO NO VALOR INICIAL */        8170

404   2   0            IF CHAVE_SINCRONISMO = 1                                           8180
                        THEN T = T + 1,.  /* INCREMENTO ACESSOS COM DESLOCAMENTO  */       8190

405   2   0            ELSE DO,.                                                          8200
```

```
STMT LEV NT

406   2  1                 S = S + 1;  /* INCREMENTO TOTAL NAO-DESLOCAMENTOS   */     8210
407   2  1                 U = U + 1;  /* INCREMENTO ACESSOS SEM DESLOCAMENTO  */     8220
408   2  1                 END;                                                     8230

409   2  0           IF CHAVE_SINCRONISMO = 1       /* 0; APROPRIA SO' NOS GE= */ 8240
                    THEN CHAVE_SINCRONISMO = 2;    /* RAIS; 1; GERAIS E TASK. */ 8250

410   2  0           CALL ESTATISTICAS;                                           8260

411   2  0           CHAVE_SINCRONISMO = 0;                                       8270

412   2  0     END CLOSP;                                                         8280
```

```
STMT LEV NT

                        /*************************************************************.      8290
                        *                        E   N   D   T   O                    *      8300
                        *************************************************************/      8310

413   1  0  ENDTO..     PROC,.                                                                 8320

414   2  0              A, TREG = 0,.   /* PARA NAO ATRAPALHAR EVENTUAL LIBERACAO */           8330

415   2  0              DO G = 1 TO Y,.                                                        8340

416   2  1              IF TEM_WRITE_PENDENTE (G) = 'S'                                        8350
                        THEN DO,.                                                             8360

417   2  2                   B = B + 1,.  /* INCREMENTO NO TOTAL DE DESLOCAMENTOS */           8370
418   2  2                   IF CHAVE_SINCRONISMO = 0                                         8380
                              THEN CHAVE_SINCRONISMO = 1,.                                    8390
419   2  2                   CALL LIBERA_BUFFER (G),.                                         8400
420   2  2                   END,.                                                            8410

421   2  1              END,.                                                                  8420

422   2  0              A, TREG = 1,. /* PARA REPOR O INCREMENTO NO VALOR INICIAL */           8430

423   2  0              IF CHAVE_SINCRONISMO = 1                                               8440
                        THEN T = T + 1,.  /* INCREMENTO ACESSOS COM DESLOCAMENTO  */           8450

424   2  0              ELSE DO,.                                                              8460

425   2  1                   S = S + 1,.  /* INCREMENTO TOTAL NAO-DESLOCAMENTOS   */           8470
426   2  1                   U = U + 1,.  /* INCREMENTO ACESSOS SEM DESLOCAMENTO  */           8480
427   2  1                   END,.                                                            8490

428   2  0              IF CHAVE_SINCRONISMO = 1       /* 0, APROPRIA SO' NOS GE- */        8500
                        THEN CHAVE_SINCRONISMO = 2,.   /* RAIS, 1, GERAIS E TASK. */        8510

429   2  0              CALL ESTATISTICAS,.                                                    8520

430   2  0              CHAVE_SINCRONISMO = 0,.                                                8530

431   2  0  END ENDTO,.                                                                       8540
```

```
STMT LEV NT

                        /*************************************************************      8550
                        *                      D  E  Q  U  E                        *      8560
                        *************************************************************/     8570

432   1   0  DEQUE..    PROC,.                                                              8580

433   2   0              A, TREG = 0,.   /* PARA NAO ATRAPALHAR EVENTUAL LIBERACAO */        8590

434   2   0              DO G = 1 TO Y,.                                                     8600

435   2   1              IF TEM_WRITE_PENDENTE (G) = 'S'                                     8610
                         THEN DO,.                                                         8620

436   2   2                   B = B + 1,.  /* INCREMENTO NO TOTAL DE DESLOCAMENTOS */       8630
437   2   2                   IF CHAVE_SINCRONISMO = 0                                      8640
                              THEN CHAVE_SINCRONISMO = 1,.                                 8650
438   2   2                   CALL LIBERA_BUFFER (G),.                                      8660
439   2   2                   END,.                                                         8670

440   2   1              END,.                                                               8680

441   2   0              A, TREG = 1,. /* PARA REPOR O INCREMENTO NO VALOR INICIAL */       8690

442   2   0              IF CHAVE_SINCRONISMO = 1                                          8700
                         THEN T = T + 1,.  /* INCREMENTO ACESSOS COM DESLOCAMENTO  */      8710

443   2   0              ELSE DO,.                                                         8720

444   2   1                   S = S + 1,.  /* INCREMENTO TOTAL NAO-DESLOCAMENTOS   */      8730
445   2   1                   U = U + 1,.  /* INCREMENTO ACESSOS SEM DESLOCAMENTO  */      8740
446   2   1                   END,.                                                         8750

447   2   0              IF CHAVE_SINCRONISMO = 1       /* 0, APROPRIA SO' NOS GE- */   8760
                         THEN CHAVE_SINCRONISMO = 2,.   /* RAIS, 1, GERAIS E TASK. */   8770

448   2   0              CALL ESTATISTICAS,.                                               8780

449   2   0              CHAVE_SINCRONISMO = 0,.                                           8790

450   2   0  END DEQUE,.                                                                   8800
```

```
STMT LEV NT

                        /*************************************************************      8810
                         *   FUNCOES DE MANUTENCAO ESTRUTURAL AINDA NAO ATIVADAS   *      8820
                         *************************************************************/     8830

451   1  0  ADC_M..   PROC,.                                                                 8840

                        /* TEORICAMENTE HAVERIA MANUTENCAO DE LINKS NO MESTRE, POIS          8850
                           E' UMA FUNCAO DO TIPO MANUTENCAO ESTRUTURAL, MAS EM TER-          8860
                           MOS PRATICOS ELA NAO E' CONTABILIZADA, COM BASE NA SUPO-          8870
                           SICAO DE QUE A CADEIA EVENTUAL DE SINONIMOS ESTA' TODA -          8880
                           ELA CONTIDA NUM SO' BLOCO FISICO.                    */          8890

452   2  0  END ADC_M,.                                                                      8900

453   1  0  DEL_M..   PROC,.                                                                 8910

                        /* TEORICAMENTE HAVERIA MANUTENCAO DE LINKS NO MESTRE, POIS          8920
                           E' UMA FUNCAO DO TIPO MANUTENCAO ESTRUTURAL, MAS EM TER-          8930
                           MOS PRATICOS ELA NAO E' CONTABILIZADA, COM BASE NA SUPO-          8940
                           SICAO DE QUE A CADEIA EVENTUAL DE SINONIMOS ESTA' TODA -          8950
                           ELA CONTIDA NUM SO' BLOCO FISICO.                    */          8960

454   2  0  END DEL_M,.                                                                      8970

455   1  0  ADDVB..   PROC,. END ADDVB,.                                                     8980
457   1  0  ADDVA..   PROC,. END ADDVA,.                                                     8990
459   1  0  ADDVR..   PROC,. END ADDVR,.                                                     9000
```

```
STMT LEV NT

                        /*************************************************************      9010
                         *     FUNCOES SEM MANUTENCAO DE LINKS NEM SINCRONISMO       *      9020
                         *************************************************************/     9030

 461   1  0  LOGTR..    PROC,. END LOGTR,.                                                   9040
 463   1  0  MARKL..    PROC,. END MARKL,.                                                   9050
 465   1  0  QMARK..    PROC,. END QMARK,.                                                   9060
 467   1  0  READD..    PROC,. END READD,.                                                   9070
 469   1  0  READR..    PROC,. END READR,.                                                   9080
 471   1  0  READV..    PROC,. END READV,.                                                   9090
 473   1  0  SERLV..    PROC,. END SERLV,.                                                   9100
 475   1  0  SEQRV..    PROC,. END SEQRV,.                                                   9110
 477   1  0  SEQWV..    PROC,. END SEQWV,.                                                   9120
 479   1  0  WRITV..    PROC,. END WRITV,.                                                   9130
 481   1  0  WRITD..    PROC,. END WRITD,.                                                   9140
 483   1  0  READM..    PROC,. END READM,.                                                   9150
 485   1  0  SEQRM..    PROC,. END SEQRM,.                                                   9160
 487   1  0  RQLOC..    PROC,. END RQLOC,.                                                   9170
 489   1  0  WRITM..    PROC,. END WRITM,.                                                   9180
 491   1  0  OPENM..    PROC,. END OPENM,.                                                   9190
 493   1  0  OPENV..    PROC,. END OPENV,.                                                   9200
 495   1  0  CLOSM..    PROC,. END CLOSM,.                                                   9210
 497   1  0  CLOSV..    PROC,. END CLOSV,.                                                   9220
 499   1  0  FREEF..    PROC,. END FREEF,.                                                   9230
 501   1  0  FREEX..    PROC,. END FREEX,.                                                   9240
 503   1  0  TOTAL..    PROC,. END TOTAL,.                                                   9250
 505   1  0  QUEST..    PROC,. END QUEST,.                                                   9260
 507   1  0  RESTM..    PROC,. END RESTM,.                                                   9270
 509   1  0  RESTV..    PROC,. END RESTV,.                                                   9280
 511   1  0  SIGNON..   PROC,. END SIGNON,.                                                  9290
 513   1  0  SIGNOFF..  PROC,. END SIGNOFF,.                                                 9300
 515   1  0  RECCRDB..  PROC,. END RECORDB,.                                                9310
 517   1  0  RECCRDA..  PROC,. END RECORDA,.                                                9320
 519   1  0  ENDLG..    PROC,. END ENDLG,.                                                   9330
```

```
STMT LEV NT

                        /*************************************************************      0010
                         *            PROCESSAMENTO DA FITA LOG DO TOTAL             *      0020
                         *************************************************************/     0030

521   1  0              DO WHILE (CHFIMLOG = 0),. /* ENQUANTO HOUVER REG'S NO LOG */      0040

522   1  1              READ FILE (FITALOG) INTO (LOG_DO_TOTAL_REDEF),.                      0050

523   1  1              IF CHFIMLOG = 0                                                      0060
                        THEN DO,.                          /* NAO E' FIM DE ARQUIVO */   0070

524   1  2                   TREG = TREG + 1,.           /* TOTAL DE REGISTROS + 1 */      0080

525   1  2                   DO I = 1 TO 41 WHILE (VETOR_DE_FUNCOES (I) NE                  0090
                                                  LOG_DO_TOTAL.FUNCAO),.                   0100
526   1  3                   END,.                                                          0110

527   1  2                   IF I = 42                                                      0120
                              THEN DO,.                                                    0130

528   1  3                        PUT SKIP EDIT                                           0140
                                 ('FUNCAO NO LOG DO TOTAL NAO RECONHECIDA - '             0150
                                   CAT LOG_DO_TOTAL.FUNCAO) (A),.                         0160
529   1  3                        TOTAL_FUNCOES_NAO_RECONHECIDAS =                        0170
                                          TOTAL_FUNCOES_NAO_RECONHECIDAS + 1,.           0180
530   1  3                        CALL PLIDUMP,.                                          0190
531   1  3                        CALL ESTAT,.                                            0200
532   1  3                        STOP,.                                                  0210
533   1  3                        END,.                                                   0220

534   1  2                   ELSE DO,.                       /* FUNCAO DO LOG EXISTE */   0230

535   1  3                        IF DETALHES_DE_FUNCOES.IMPLICA_EM_ACESSO (I) = 'S'      0240

                                 THEN DO,.          /* FUNCAO E' DO TIPO DE ACESSO */    0250

536   1  4                             A = A + 1,.  /* A - INCREMENTO NO TOTAL DE       0260
                                                        ACESSOS                  */      0270

                                   /* O 'IF' ABAIXO OBRIGA O LOG SER DO TIPO ...         0280
                                      'REFER AFTER'. VERIFIQUE A VERACIDADE.     */      0290

537   1  4                             IF  LEITURA_OU_GRAVACAO (I) = 'L' AND              0300
                                        ((MESTRE_OU_VARIAVEL (I) = 'V' AND               0310
                                          REFER_CHAR = '    ')      OR                   0320
                                          LOG_DO_TOTAL.STATUS_CHAR = 'END.')             0330

                                        THEN DO,.       /* SAO REGISTROS DO LOG QUE     0340
                                                           NAO DEVEM SER CONSIDERADOS -  0350
                                                           FIM DE CADEIA OU DE ARQUIVO */ 0360
538   1  5                                  A = A - 1,.                                   0370
539   1  5                                  CHAVE_ESTATISTICAS = 0,.                      0380
540   1  5                                  CALL ESTATISTICAS,.                           0390
```

```
STMT LEV NT

541   1  5                        CHAVE_ESTATISTICAS = 1;.                    0400
542   1  5                        END;.                                       0410

543   1  4                    ELSE DO;.          /* NAO=FIM DE CADEIA/ARQ */  0420

544   1  5                        IF STATUS_OK = 'SIM'                        0430
                                 THEN DO;.          /* STATUS VALIDO */     0440

545   1  6                            IF FUNCAO_FORCA_I_O (I) = 'S'           0450

                                     THEN CALL                               0460
                                         MANUTENCAO_OU_SINCRONISMO (I);.     0470

                                       /* SAO OS CASOS DE SINCRONISMO DE     0480
                                          BUFFERS, TRATADOS A PARTE.  */     0490

546   1  6                            ELSE IF SERAH_NECESSARIO_O_ACESSO       0500
                                                          = 'SIM'            0510
                                         THEN DO;.                           0520
547   1  7                                 B = B + 1;.                       0530
548   1  7                                 CALL
                                         IMPLEMENTACAO_DE_PARAMETROS ;.      0550
549   1  7                                 END;.                             0560

550   1  6                            IF TIPO_DE_GRAVACAO (I) = 'M'           0570

                                     THEN CALL MANUTENCAO_OU_SINCRONISMO     0580
                                                                  (I);.      0590
                                   /* E' COMPUTADA AQUI A MANUTENCAO DE      0600
                                      LINKS DECORRENTE DE UMA OPERACAO DE    0610
                                      MANUTENCAO ESTRUTURAL EM ARQUIVOS      0620
                                      TIPO MESTRE OU VARIAVEL.          */   0630

551   1  6                            END;.      /* FIM - STATUS VALIDO */   0640

552   1  5                        ELSE DO;.                                   0650
                                         /* SAO REGISTROS DO LOG QUE        0660
                                         NAO DEVEM SER CONSIDERADOS -        0670
                                                       STATUS INVALIDO */    0680
553   1  6                            A = A - 1;.                             0690
554   1  6                            CHAVE_ESTATISTICAS = 0;.                0700
555   1  6                            CALL ESTATISTICAS;.                     0710
556   1  6                            CHAVE_ESTATISTICAS = 1;.                0720
557   1  6                            END;.                                   0730
558   1  5                        END;.  /* FIM - NAO-FIM DE CADEIA/ARQ */    0740
559   1  4                    END;. /* FIM - FUNCAO E' DO TIPO DE ACESSO */    0750

560   1  3                ELSE CALL ESTATISTICAS;.                            0760
                             /* PARA COMPUTAR O REGISTRO NAO-DE-ACESSO       0770
                                E ZERAR TREG.                          */    0780
561   1  3                END;.              /* FIM - FUNCAO DO LOG EXISTE */ 0790
562   1  2            END;.                  /* FIM - NAO E' FIM DE ARQUIVO */ 0800
563   1  1        END;.                  /* FIM - ENQUANTO HOUVER REG'S NO LOG */ 0810
```

```
STMT LEV NT

                    /*************************************************************      0820
                     *    FUNCAO QUE RETORNA UMA DAS DUAS RESPOSTAS ABAIXO -    *      0830
                     *     SIM - O STATUS DEVOLVIDO PELA FUNCAO ESTA' OK        *      0840
                     *     NAO - O STATUS DEVOLVIDO PELA FUNCAO NAO ESTA' OK    *      0850
                     *************************************************************/     0860

 564   1  0  STATUS_OK.. PROC RETURNS (CHAR (3)),.                                      0870

 565   2  0            DCL RESPOSTA CHAR (3),.                                         0880

 566   2  0            RESPOSTA = ' ',.                                                0890

 567   2  0            IF VERIFY (STATUS_CHAR, MASCARA_STATUS) NE 0                    0900
                       THEN DO R = 1 TO 5 WHILE (RESPOSTA NE 'SIM'),.                  0910

                                 /* O STATUS E' NUMERICO (OU MELHOR, BINARIO) */       0920

 568   2  1                 IF STATUS_BIN = STATUS_OK_BIN (R)                          0930
                            THEN RESPOSTA = 'SIM',.                                   0940
 569   2  1                 ELSE RESPOSTA = 'NAO',.                                    0950
 570   2  1                 END,.                                                      0960

 571   2  0            ELSE DO,.                                                       0970

                                 /* O STATUS E' ALFABETICO, '****' OU 'END.' */        0980

 572   2  1                 IF STATUS_CHAR =                                           0990
                              SUBSTR (MASCARA_DE_STATUS_ACEITAVEIS (I), 1, 4)         1000
                            OR STATUS_CHAR =                                          1010
                               SUBSTR (MASCARA_DE_STATUS_ACEITAVEIS (I), 5, 4)        1020
                            THEN RESPOSTA = 'SIM',.                                   1030
 573   2  1                 ELSE RESPOSTA = 'NAO',.                                    1040
 574   2  1                 END,.                                                      1050

 575   2  0            RETURN (RESPOSTA),.                                             1060

 576   2  0  END STATUS_OK,.                                                           1070
```

STMT LEV NT

```
                    /*************************************************************      1080
                    *    FUNCAO QUE RETORNA UMA DAS DUAS RESPOSTAS ABAIXO -     *      1090
                    *       SIM - SERAH NECESSARIO O ACESSO                     *      1100
                    *       NAO - NAO SERAH NECESSARIO O ACESSO                 *      1110
                    *************************************************************/     1120

577   1  0  SERAH_NECESSARIO_O_ACESSO.. PROC RETURNS (CHAR (3)),.                      1130

578   2  0            DCL RESPOSTA CHAR (3),.                                         1140

579   2  0            DO K = 1 TO Z WHILE (LOG_DO_TOTAL.ARQUIVO NE                    1150
                                        DETALHES_DE_ARQUIVOS.ARQUIVO (K)),.          1160
580   2  1            END,.                                                           1170

581   2  0            IF K = Z + 1                                                    1180
                      THEN DO,.                                                       1190

582   2  1                 PUT SKIP EDIT                                              1200
                          ('ARQUIVO NO LOG DO TOTAL NAO RECONHECIDO - '              1210
                            CAT LOG_DO_TOTAL.ARQUIVO) (A),.                          1220
583   2  1                 TOTAL_ARQS_NAO_RECONHECIDOS =                              1230
                                        TOTAL_ARQS_NAO_RECONHECIDOS + 1,.            1240
584   2  1                 CALL PLIDUMP,.                                             1250
585   2  1                 CALL ESTAT,.                                               1260
586   2  1                 STOP,.                                                     1270
587   2  1                 END,.                                                      1280

588   2  0            ELSE DO,.                          /* LEITURA OU GRAVACAO */   1290

589   2  1                 IF DETALHES_DE_FUNCOES.LEITURA_OU_GRAVACAO (1) = 'L'       1300
                          THEN DO,.                                  /* LEITURA */   1310

590   2  2                      IF DETALHES_DE_FUNCOES.MESTRE_OU_VARIAVEL (1)         1320
                                                                      = 'V'          1330
                               THEN DO,.                 /* LEITURA DE VARIAVEL */   1340

591   2  3                           PARAMETROS.ARQUIVO =                           1350
                                        LOG_DO_TOTAL.ARQUIVO,.                       1360

                                /* EM CASO DE LEITURA DE VARIAVEL, TEMOS QUE LE-     1370
                                   VAR EM CONTA A POSSIBILIDADE DO RETORNO DE UM     1380
                                   'LKXX' NO REFER, NO CASO DE UM 'READV' COM -      1390
                                   STATUS 'MRNF'. ESTE REGISTRO DEVE SER CONTA -     1400
                                   BILIZADO POIS O ACESSO REALMENTE OCORREU, MAS     1410
                                   AO MESTRE, AO VARIAVEL NAO, EM FACE DO STATUS     1420
                                   RETORNADO.                                */      1430

592   2  3                           IF SUBSTR (REFER_CHAR, 1, 2) = 'LK'            1440
                                    THEN DO,.                                        1450

593   2  4                                DO M = 1 TO Z WHILE                       1460
                                             (SUBSTR (LINKPATH_CHAR, 1, 4) NE        1470
                                             DETALHES_DE_ARQUIVOS.ARQUIVO (M)),.     1480
```

```
STMT LEV NT

 594   2  5                        END,.                                    1490

 595   2  4                        IF M = Z + 1                             1500
                                  THEN DO,.                                1510

 596   2  5                  PUT SKIP EDIT ('MESTRE NO LINK DO VARIAVEL NAO RE'   1520
                            CAT 'CONHECIDO - ' CAT LOG_DO_TOTAL.ARQUIVO          1530
                            CAT ' - ' CAT SUBSTR (LINKPATH_CHAR, 1, 4))  (A),.   1540
 597   2  5                        TOTAL_ARQS_NAO_RECONHECIDOS =            1550
                                    TOTAL_ARQS_NAO_RECONHECIDOS + 1,.      1560
 598   2  5                        CALL PLIDUMP,.                           1570
 599   2  5                        CALL ESTAT,.                             1580
 600   2  5                        STOP,.                                   1590
 601   2  5                        END,.                                    1600

 602   2  4                        ELSE DO,.                                1610

 603   2  5                        DO N = 1 TO W WHILE                      1620
                                                (LINKPATH_CHAR NE         1630
                                  ESPECIFICACAO_DE_VINCULOS.LINKPATH       1640
                                                          (N)),.           1650
 604   2  6                        END,.                                    1660

 605   2  5                        IF N = W + 1                             1670
                                  THEN DO,.                                1680

                              /* NAO ACHOU O VARIAVEL NO VETOR DE VINCULOS */   1690

 606   2  6                  PUT SKIP EDIT ('VARIAVEL SEM ENTRADA NO VETOR DE ESPEC'   1700
                                  CAT 'IFICACAO DE VINCULOS - '            1710
                                  CAT LOG_DO_TOTAL.ARQUIVO)     (A),.      1720
 607   2  6                    TOTAL_VINCULOS_NAO_RECONHECIDOS =            1730
                             TOTAL_VINCULOS_NAO_RECONHECIDOS + 1,.         1740
 608   2  6                        CALL PLIDUMP,.                           1750
 609   2  6                        CALL ESTAT,.                             1760
 610   2  6                        STOP,.                                   1770
 611   2  6                        END,.                                    1780

 612   2  5                        ELSE DO,.                                1790

                                  /* ACHOU O VINCULO DO VARIAVEL */        1800

 613   2  6                        CHAVE_DO_ARQUIVO = SUBSTR                1810
                                          (IMAGEM_DO_REGISTRO,             1820
                                           INICIO_DA_CHAVE (N),            1830
                                         TAMANHO_DA_CHAVE (M)),.           1840

 614   2  6                        CALL CALCULA_RQLOC                       1850
                                  (DETALHES_DE_ARQUIVOS.ARQUIVO (M),       1860
                                                  CHAVE_DO_ARQUIVO,        1870
                                    PARAMETROS_1.ENDERECO_BIN),.           1880

                        /* O ACESSO QUE EVENTUALMENTE SERA' APROPRIADO E' O AO MESTRE   1890
                           E ENTAO PASSAMOS DE UM READV PARA UM READM. PARA TANTO MU-   1900
```

```
STMT LEV NT

                 CAMPOS K PARA M, I DE READV PARA READM (12 PARA 18) E TRO =    1910
                 CAMPOS O NOME DO VARIAVEL PELO DO MESTRE NA ESTRUTURA DE       1920
                 PARAMETROS.                                                */  1930

 615   2  6                         K = M;.                                     1940
 616   2  6                         I = 18;.        /* VIROU READM */           1950
 617   2  6                         PARAMETROS.ARQUIVO =                        1960
                                 DETALHES_DE_ARQUIVOS.ARQUIVO (K);.             1970

 618   2  6                         END;.                                       1980
 619   2  5                      END;.                                          1990
 620   2  4                   END;.                                             2000

 621   2  3                ELSE DO;.        /* APROVEITA DIRETO O REFER */      2010

 622   2  4                   PARAMETROS_1.ENDERECO_BIN =                       2020
                                       LOG_DO_TOTAL.REFER_BIN;.                 2030
 623   2  4                   END;.                                             2040
 624   2  3                CALL ROTINA_SIM_OU_NAO (PARAMETROS.ARQUIVO,          2050
                                    PARAMETROS_1.ENDERECO_BIN,                  2060
                                        RESPOSTA);.                             2070

 625   2  3                IF RESPOSTA = 'NAO'                                  2080
                           THEN DO;.                                            2090

 626   2  4                   S = S + 1;.                                       2100
 627   2  4                   U = U + 1;.                                       2110
 628   2  4                   END;.                                             2120

 629   2  3                ELSE T = T + 1;.                                     2130

 630   2  3                END;.          /* FIM DE LEITURA DE VARIAVEL */      2140

 631   2  2             ELSE DO;. /* LEITURA DE MESTRE - READM OU SEQRM */      2150

 632   2  3                PARAMETROS.ARQUIVO =                                 2160
                                  LOG_DO_TOTAL.ARQUIVO;.                        2170
 633   2  3                CHAVE_DO_ARQUIVO = SUBSTR                            2180
                          (LOG_DO_TOTAL.IMAGEM_DO_REGISTRO,9,                   2190
                           DETALHES_DE_ARQUIVOS.TAMANHO_DA_CHAVE (K));.         2200
 634   2  3                CALL CALCULA_RELOC (PARAMETROS.ARQUIVO,              2210
                           CHAVE_DO_ARQUIVO,PARAMETROS_1.ENDERECO_BIN);.        2220
 635   2  3                CALL ROTINA_SIM_OU_NAO (PARAMETROS.ARQUIVO,          2230
                                    PARAMETROS_1.ENDERECO_BIN,                  2240
                                              RESPOSTA);.                       2250

 636   2  3                IF RESPOSTA = 'NAO'                                  2260
                           THEN DO;.                                            2270

 637   2  4                   S = S + 1;.                                       2280
 638   2  4                   U = U + 1;.                                       2290
 639   2  4                   END;.                                             2300

 640   2  3                ELSE T = T + 1;.                                     2310
```

```
STMT LEV NT

641   2  3                        END,.          /* FIM DE LEITURA DE MESTRE */      2320
642   2  2                    END,.                          /* FIM DE LEITURA */      2330

643   2  1                ELSE DO,.                          /* NAO E' LEITURA */      2340

644   2  2                    IF DETALHES_DE_FUNCOES.LEITURA_OU_GRAVACAO (I)          2350
                                                                        = 'G'         2360
                           THEN DO,.                                 /* GRAVACAO */   2370

645   2  3                        IF DETALHES_DE_FUNCOES.MESTRE_OU_VARIAVEL (I)       2380
                                                                        = 'M'         2390
                               THEN DO,.          /* GRAVACAO DE MESTRE */          2400

646   2  4                            PARAMETROS.ARQUIVO =                           2410
                                          LOG_DO_TOTAL.ARQUIVO,.                     2420
647   2  4                            CHAVE_DO_ARQUIVO = SUBSTR                      2430
                                     (LOG_DO_TOTAL.IMAGEM_DO_REGISTRO,9,             2440
                                   DETALHES_DE_ARQUIVOS.TAMANHO_DA_CHAVE (K)),.      2450
648   2  4                            CALL CALCULA_RQLOC                             2460
                                        (PARAMETROS.ARQUIVO,                         2470
                                              CHAVE_DO_ARQUIVO,                      2480
                                              PARAMETROS_1.ENDERECO_BIN),.           2490
649   2  4                            CALL ROTINA_SIM_OU_NAO                         2500
                                                 (PARAMETROS.ARQUIVO,                2510
                                               PARAMETROS_1.ENDERECO_BIN,            2520
                                                   RESPOSTA),.                       2530

650   2  4                            IF RESPOSTA = 'NAO'                            2540
                                      THEN DO,.                                      2550

651   2  5                                S = S + 1,.                                2560
652   2  5                                U = U + 1,.                                2570
653   2  5                                END,.                                      2580

654   2  4                            ELSE T = T + 1,.                               2590

655   2  4                            END,.    /* FIM DE GRAVACAO DE MESTRE */    2600

656   2  3                        ELSE DO,.          /* GRAVACAO DE VARIAVEL */     2610

657   2  4                            PARAMETROS.ARQUIVO =                           2620
                                              LOG_DO_TOTAL.ARQUIVO,.                 2630

                       /* APROVEITAMOS AQUI UMA INCOERENCIA DO LOG. MESMO QUE ELE SE-  2640
                          JA DO TIPO 'REFER AFTER', SE A FUNCAO FOR UM ADDVC PRIMEI -  2650
                          RO DE CADEIA, O REFER ESTARA' COM 'LKXX'.              */    2660

658   2  4                            IF SUBSTR (REFER_CHAR, 1, 2) = 'LK'            2670
                                      THEN DO,.                                      2680

659   2  5                                DO M = 1 TO Z WHILE                        2690
                                       (SUBSTR (LINKPATH_CHAR, 1, 4) NE              2700
                                        DETALHES_DE_ARQUIVOS.ARQUIVO (M)),.          2710
660   2  6                                END,.                                      2720
```

```
STMT LEV NT

661   2  5                        IF M = Z + 1                                  2730
                                  THEN DO;.                                     2740

662   2  6                  PUT SKIP EDIT ('MESTRE NO LINK DO VARIAVEL NAO RE'   2750
                           CAT 'CONHECIDO - ' CAT LOG_DO_TOTAL.ARQUIVO          2760
                           CAT ' - ' CAT SUBSTR (LINKPATH_CHAR, 1, 4)) (A),.    2770
663   2  6                              TOTAL_ARQS_NAO_RECONHECIDOS =            2780
                                    TOTAL_ARQS_NAO_RECONHECIDOS + 1,.           2790
664   2  6                              CALL PLIDUMP,.                           2800
665   2  6                              CALL ESTAT,.                             2810
666   2  6                              STOP,.                                   2820
667   2  6                              END,.                                    2830

668   2  5                        ELSE DO,.                                     2840

669   2  6                             DO N = 1 TO W WHILE                      2850
                                      (LINKPATH_CHAR NE                         2860
                                  ESPECIFICACAO_DE_VINCULOS.LINKPATH            2870
                                                          (N)),.                2880
670   2  7                             END,.                                    2890

671   2  6                             IF N = W + 1                             2900
                                      THEN DO,.                                 2910

                              /* NAO ACHOU O VARIAVEL NO VETOR DE VINCULOS */   2920

672   2  7                  PUT SKIP EDIT ('VARIAVEL SEM ENTRADA NO VETOR DE ESPEC'  2930
                                 CAT 'IFICACAO DE VINCULOS - '                  2940
                                 CAT LOG_DO_TOTAL.ARQUIVO)          (A),.       2950
673   2  7                           TOTAL_VINCULOS_NAO_RECONHECIDOS =          2960
                                  TOTAL_VINCULOS_NAO_RECONHECIDOS + 1,.         2970
674   2  7                                 CALL PLIDUMP,.                       2980
675   2  7                                 CALL ESTAT,.                         2990
676   2  7                                 STOP,.                               3000
677   2  7                                 END,.                                3010

678   2  6                             ELSE DO,.                                3020

                                    /* ACHOU O VINCULO DO VARIAVEL */           3030
                                                                                .
679   2  7                                   CHAVE_DO_ARQUIVO = SUBSTR          3040
                                         (IMAGEM_DO_REGISTRO,                   3050
                                          INICIO_DA_CHAVE (N),                  3060
                                          TAMANHO_DA_CHAVE (M)),.               3070

680   2  7                                   CALL CALCULA_RQLOC                 3080
                                  (DETALHES_DE_ARQUIVOS.ARQUIVO (M),            3090
                                          CHAVE_DO_ARQUIVO,                     3100
                                       PARAMETROS_1.ENDERECO_BIN),.             3110

681   2  7                                    CALL GERA_ENDERECO_RELATIVO       3120
                                             (K, M, N, ENDERECO_BIN),.          3130
682   2  7                                    END,.                             3140
683   2  6                              END,.                                    3150
```

```
STMT LEV NT

 684   2  5                              END,.                                  3160

 685   2  4                          ELSE DO,. /* APROVEITA DIRETO O REFER */  3170

 686   2  5                              PARAMETROS_1.ENDERECO_BIN =            3180
                                            LOG_DO_TOTAL.REFER_BIN,.           3190
 687   2  5                              END,.                                  3200

 688   2  4                          CALL ROTINA_SIM_OU_NAO                     3210
                                              (PARAMETROS.ARQUIVO,             3220
                                         PARAMETROS_1.ENDERECO_BIN,            3230
                                               RESPOSTA),.                     3240

 689   2  4                          IF RESPOSTA = 'NAO'                        3250
                                    THEN DO,.                                  3260

 690   2  5                              S = S + 1,.                            3270
 691   2  5                              U = U + 1,.                            3280
 692   2  5                              END,.                                  3290

 693   2  4                          ELSE T = T + 1,.                           3300

 694   2  4                          END,.  /* FIM DE GRAVACAO DE VARIAVEL */  3310
 695   2  3                    END,.                 /* FIM DE GRAVACAO */     3320

 696   2  2                ELSE RESPOSTA = 'NAO',.                              3330

                               /* E' UMA IMPOSSIBILIDADE LOGICA - O CAMPO */   3340
                               /* LEITURA_OU_GRAVACAO NAO E' 'G' NEM 'L', */   3350
                               /* OU SEJA, E' ' ' OU 'I' E EM AMBOS OS CA-*/   3360
                               /* SOS ESTA FUNCAO NAO E' ATIVADA.         */   3370

 697   2  2                    END,.                 /* FIM DE NAO E' LEITURA */ 3380

 698   2  1              IF RESPOSTA = 'NAO'         /* TRATAMOS DE APROPRIAR */ 3390
                        THEN DO,.                   /* AQUI OS SEGUINTES CA- */ 3400
 699   2  2                   INDICE = K,.           /* SOS (PARA NAO TERMOS  */ 3410
 700   2  2                   INDICE1 = L,.          /* QUE NOS PREOCUPAR FO- */ 3420
 701   2  2                   CALL POSICIONA_P,.     /* RA DA FUNCAO) -       */ 3430
 702   2  2                   CALL CORRIGE_P,.       /*  - READ, WRITE OU DE- */ 3440
 703   2  2                   CALL ATUALIZA_BUFFER,. /*    LETE EMITIDO APOS  */ 3450
 704   2  2                   END,.                  /*    UM READ.           */ 3460

 705   2  1              RETURN (RESPOSTA),.                                     3470

 706   2  1              END,.                 /* FIM DE LEITURA OU GRAVACAO */ 3480

 707   2  0 END SERAH_NECESSARIO_O_ACESSO,.                                     3490
```

```
STMT LEV NT

                        /*************************************************************      3500
                        *       ROTINA PARA APROPRIAR CORRETAMENTE O ACESSO          *      3510
                        *         IMPLEMENTANDO DEVIDAMENTE OS PARAMETROS            *      3520
                        *************************************************************/     3530

 708   1  0  IMPLEMENTACAO_DE_PARAMETROS.. PROC,.                                           3540

 709   2  0            CALL DETERMINA_O_BUFFER,.                                           3550

 710   2  0            IF TEM_WRITE_PENDENTE (J) = 'S'                                     3560
                       THEN CALL LIBERA_BUFFER (J),.                                       3570

 711   2  0            CALL TRAZ_O_BLOCO,.                                                 3580

 712   2  0  END IMPLEMENTACAO_DE_PARAMETROS,.                                             3590

                        /*************************************************************      3600
                        * ROTINA PARA APROPRIAR E/S DE BUFFERS QUE FORAM LIBERA -  *      3610
                        *             DOS MAS TEM UM WRITE PENDENTE                *      3620
                        *************************************************************/     3630

 713   1  0  LIBERA_BUFFER.. PROC (J1),.                                                   3640

 714   2  0            DCL J1                    BIN FIXED (31),.                          3650

 715   2  0            DO C = 1 TO Z WHILE                                                 3660
                              (DETALHES_DE_ARQUIVOS.ARQUIVO (Q) NE                        3670
                                   CONDICAO_ATUAL_DO_BUFFER.ARQUIVO (J1)),.               3680
 716   2  1            END,.                                                               3690

 717   2  0            I_O = 'O',.                                                         3700
 718   2  0            CHAVE_PENDENTE = 1,.                                                3710
 719   2  0            CALL CALCULA_CILINDRO_ALVO (J1, CILINDRO_ALVO),.                    3720
 720   2  0            CALL PERFORMA_I_O (J1, CILINDRO_ALVO),.                             3730
 721   2  0            INDICE1 = J1,.                                                      3740
 722   2  0            CALL ESTATISTICAS,.                                                 3750
 723   2  0            TEM_WRITE_PENDENTE (J1) = ' ',.                                     3760

 724   2  0  END LIBERA_BUFFER,.                                                           3770
```

```
STMT LEV NT

                    /*************************************************************      3780
                     * ESTA ROTINA DEVOLVE NA VARIAVEL RESPOSTA OS VALORES -   *      3790
                     *       SIM - NECESSARIO O ACESSO, O BLOCO NAO ESTA' EM ME-*      3800
                     *                                                   MORIA *      3810
                     *       NAO - NAO EH NECESSARIO O ACESSO, O BLOCO JA ESTA' *      3820
                     *                                              EM MEMORIA *      3830
                     *************************************************************/     3840

 725   1  0  ROTINA_SIM_OU_NAO.% PROC (ARQUIVO1,ENDERECO1,RESPOSTA),.                    3850

 726   2  0            DCL ARQUIVO1              CHAR (4),                             3860
                        ENDERECO1       BIN FIXED (31),                               3870
                        RESPOSTA                CHAR (3),.                          3880

 727   2  0            DCL ENDERECO1_CHAR        CHAR (4) BASED (PEND),.               3890

 728   2  0            DCL PEND                   POINTER,.                            3900

 729   2  0            PEND = ADDR (ENDERECO1),.                                       3910

 730   2  0            RESPOSTA = '   ',.                                              3920

 731   2  0            DO L = 1 TO Y WHILE (RESPOSTA = '    '),.                       3930

 732   2  1            IF CONDICAO_ATUAL_DO_BUFFER.ARQUIVO (L) = ARQUIVO1              3940
                       THEN DO,.                                                       3950

 733   2  2                 IF ENDERECO1_CHAR = '    '                                 3960
                           OR (ENDERECO1 GE                                          3970
                                    ENDERECO_INICIAL_BIN (L)                         3980
                           AND ENDERECO1 LE                                          3990
                                    ENDERECO_FINAL_BIN (L))                          4000
                           THEN RESPOSTA = 'NAO',.                                   4010
 734   2  2                 END,.                                                     4020

 735   2  1            END,.                                                         4030

 736   2  0            IF RESPOSTA = '   '                                           4040
                       THEN RESPOSTA = 'SIM',.                                       4050
 737   2  0            ELSE L = L - 1,.                                              4060

 738   2  0  END ROTINA_SIM_OU_NAO,.                                                  4070
```

```
STMT LEV NT

                    /*************************************************************      4080
                     *  ROTINA PARA GERAR DE FORMA ALEATORIA OU APROXIMADA O   *      4090
                     *    ENDERECO INICIAL DA CADEIA DE UM ARQUIVO VARIAVEL    *      4100
                     *************************************************************/     4110

 739   1  0  GERA_ENDERECO_RELATIVO.. PROC (K1, M1, N1, RESPOSTA),.                     4120

 740   2  0           DCL (K1, M1, N1, RESPOSTA)          BIN FIXED (31),              4130
                        ACABOU_BUSCA                     BIN FIXED (31) INIT (0),     4140
                       (FATOR_DE_UTILIZACAO_BIN,                                      4150
                        TAMANHO_MEDIO_BIN)              BIN FIXED (31,7) INIT (0),.  4160

 741   2  0           TAMANHO_MEDIO_BIN =            TAMANHO_MEDIO_DA_CADEIA (N1),.  4170
 742   2  0           FATOR_DE_UTILIZACAO_BIN =  FATOR_DE_UTILIZACAO (M1) * 0.01,.   4180

 743   2  0           DO WHILE (ACABOU_BUSCA = 0),.                                   4190

 744   2  1           IF PRIMARIO_OU_SECUNDARIO (N1) = 'P'                            4200
                      THEN DO,.                                                       4210

 745   2  2                RESPOSTA = FLOOR (ENDERECO_BIN * TAMANHO_MEDIO_BIN *       4220
                                                FATOR_DE_UTILIZACAO_BIN) + 1,.        4230
 746   2  2                IF RESPOSTA GE CAPACIDADE_REG_LOGICOS (K1)                 4240
                           THEN TAMANHO_MEDIO_BIN =        TAMANHO_MEDIO_BIN - 1,.   4250
 747   2  2                ELSE ACABOU_BUSCA = 1,.                                    4260
 748   2  2                END,.                                                      4270

 749   2  1           ELSE DO,.                                                       4280

 750   2  2                CALL ENDERECO_ALEATORIO                                    4290
                                (1, CAPACIDADE_REG_LOGICOS (K1), RESPOSTA),.        4300
 751   2  2                ACABOU_BUSCA = 1,.                                         4310
 752   2  2                END,.                                                      4320

 753   2  1           IF TAMANHO_MEDIO_BIN LT 0                                       4330
                      THEN DO,.                                                       4340

 754   2  2                RESPOSTA =    FLOOR (CAPACIDADE_REG_LOGICOS (K1) / 2),.   4350
 755   2  2                ACABOU_BUSCA = 1,.                                         4360
 756   2  2                END,.                                                      4370

 757   2  1           END,.            /* FIM DO 'DO WHILE (ACABOU_BUSCA = 0)' */     4380
```

```
STMT LEV NT

                    /*************************************************************      4390
                     *        ROTINA PARA GERACAO DE ENDERECO ALEATORIO           *      4400
                     *************************************************************/     4410

758   2  0  (NOFIXEDOVERFLOW).. ENDERECO_ALEATORIO.. PROC (END_INI,END_FIM,ALEA),.      4420

759   3  0        DCL (END_INI, END_FIM, ALEA, TRABALHO) BIN FIXED (31,0),              4430
                       ALEATORIO                        DEC FLOAT   (10),.            4440

760   3  0        TRABALHO = 65539 * SEMENTE,.                                         4450

761   3  0        IF TRABALHO LT 0                                                     4460
                  THEN TRABALHO = TRABALHO + 2147483647 + 1,.                          4470

762   3  0        ALEATORIO = TRABALHO,.                                               4480
763   3  0        ALEATORIO = ALEATORIO / 2147483647,.                                 4490
764   3  0        SEMENTE = TRABALHO,.                                                 4500
765   3  0        ALEA = END_INI + (END_FIM - END_INI) * ALEATORIO,.                   4510

766   3  0  END ENDERECO_ALEATORIO,.                                                   4520

767   2  0  END GERA_ENDERECO_RELATIVO,.                                               4530

                    /*************************************************************      4540
                     *                    CALCULO DO RQLOC                       *      4550
                     *************************************************************/     4560

768   1  0  CALCULA_RQLOC.. PROC (ARQUIVO2,CHAVE,RQLOC),.                             4570

769   2  0        DCL ARQUIVO2          CHAR (4),                                      4580
                     CHAVE              CHAR (*),                                      4590
                     RQLOC       BIN FIXED (31),.                                     4600

770   2  0        CALL DATBAS ('RQLOC',TSTATUS,ARQUIVO2,CHAVE,RQLOC,'END.'),.          4610

771   2  0        IF TSTATUS NE '****'                                                 4620
                  THEN DO,.                                                            4630
772   2  1             TARQ   = ARQUIVO2,.                                             4640
773   2  1             TFUNC  = 'RQLOC',.                                              4650
774   2  1             TCHAVE = CHAVE,.                                                4660
775   2  1             CALL TIMPER,.                                                   4670
776   2  1             END,.                                                           4680

777   2  0  END CALCULA_RQLOC,.                                                       4690
```

```
STMT LEV NT

                    /*************************************************************      4700
                     * ESTA PROCEDURE ESCOLHE QUAL O BUFFER QUE VAI SER LIBERA-*      4710
                     * DO PARA RECEBER UM NOVO BLOCO FISICO DA REDE. A ESCOLHA *      4720
                     * SE FAZ SEGUNDO O CRITERIO DO 'OCIOSO A MAIS TEMPO'.     *      4730
                     *************************************************************/     4740

 778   1  0  DETERMINA_O_BUFFER.. PROC,.                                               4750

 779   2  0            MENOR_MOMENTO = (15)'9',.                                      4760

 780   2  0            IF CHAVE_VINCULO = 1                                           4770
                      THEN INDICE = M,.                                              4780
 781   2  0            ELSE IF CHAVE_PENDENTE = 1                                     4790
                           THEN INDICE = Q,.                                         4800
 782   2  0                ELSE INDICE = K,.                                         4810

 783   2  0            DO J = 1 TO Y,.                                                4820

 784   2  1            IF CONDICAO_ATUAL_DO_BUFFER.BUFFER (J) =                       4830
                                        DETALHES_DE_ARQUIVOS.BUFFER (INDICE)        4840
                      THEN DO,.                                                      4850

 785   2  2                IF MOMENTO_DO_ULTIMO_USO (J) LT MENOR_MOMENTO              4860
                           THEN DO,.                                                 4870
 786   2  3                     H = J,.                                              4880
 787   2  3                     MENOR_MOMENTO = MOMENTO_DO_ULTIMO_USO (J),.          4890
 788   2  3                     END,.                                                4900
 789   2  2                END,.                                                     4910
 790   2  1            END,.                                                          4920

 791   2  0            IF J = Y + 1 AND MENOR_MOMENTO = (15)'9'                       4930
                      THEN DO,.                                                      4940

 792   2  1                PUT SKIP EDIT                                              4950
                           ('BUFFER REFERENCIADO PELO ARQUIVO '    CAT                4960
                             DETALHES_DE_ARQUIVOS.ARQUIVO (INDICE) CAT                4970
                            ' NAO RECONHECIDO - '                  CAT                4980
                             DETALHES_DE_ARQUIVOS.BUFFER  (INDICE)) (A),.             4990
 793   2  1                TOTAL_BUFFERS_NAO_RECONHECIDOS =                           5000
                                        TOTAL_BUFFERS_NAO_RECONHECIDOS + 1,.        5010
 794   2  1                CALL PLIDUMP,.                                             5020
 795   2  1                CALL ESTAT,.                                               5030
 796   2  1                STOP,.                                                     5040
 797   2  1                END,.                                                      5050

 798   2  0            ELSE J = H,.                                                   5060

 799   2  0  END DETERMINA_O_BUFFER,.                                                5070
```

```
STMT LEV NT

                    /*************************************************************      5080
                     * ESTA FUNCAO DETERMINA O CILINDRO PARA O QUAL O BRACO DE-*      5090
                     *                          VE SE DESLOCAR                *      5100
                     *************************************************************/     5110

 800   1  0  CALCULA_CILINDRO_ALVO.. PROC (J, RESPOSTA),.                              5120

 801   2  0            DCL RESPOSTA                  PIC '999',                        5130
                          REGISTROS_POR_CILINDRO   PIC '(6)9',                        5140
                          J                        BIN FIXED (31),.                   5150

 802   2  0            RESPOSTA = 0,.                                                  5160

 803   2  0            IF I_O = 'O'                                                    5170
                       THEN RESPOSTA = CONDICAO_ATUAL_DO_BUFFER.CILINDRO (J),.         5180
 804   2  0            ELSE DO,.                                                       5190

 805   2  1                 IF CHAVE_VINCULO = 1                                       5200
                           THEN INDICE = M,.                                          5210
 806   2  1                 ELSE INDICE = K,.                                          5220

                           /* AQUI NAO EXISTE A POSSIBILIDADE DE CHAVE_PENDENTE       5230
                              SER 1 (E INDICE = Q), POIS ENTAO I_O SERIA = 'O'  */    5240

 807   2  1                 CALL POSICIONA_P,.                                         5250

 808   2  1                 IF P = V + 1                                               5260
                           THEN DO,.                                                  5270

 809   2  2                      PUT SKIP EDIT                                         5280
                                ('ARQUIVO NO LOG DO TOTAL NAO ALOCADO - '             5290
                                  CAT DETALHES_DE_ARQUIVOS.ARQUIVO (INDICE)) (A),.    5300
 810   2  2                      TOTAL_ARQS_NAO_RECONHECIDOS =                         5310
                                          TOTAL_ARQS_NAO_RECONHECIDOS + 1,.           5320
 811   2  2                      CALL PLIDUMP,.                                        5330
 812   2  2                      CALL ESTAT,.                                          5340
 813   2  2                      STOP,.                                                5350
 814   2  2                      END,.                                                 5360

 815   2  1                 ELSE DO,.                                                  5370

 816   2  2                      CALL CORRIGE_P,.                                      5380
 817   2  2                      REGISTROS_POR_CILINDRO = 19 *                         5390
                                          REG_LOGICOS_POR_BLOCO (INDICE) *            5400
                                          BLOCOS_FISICOS_POR_TRILHA (INDICE),.        5410
 818   2  2                      RESPOSTA = CILINDRO_INICIAL (P) + FLOOR               5420
                                     (ENDERECO_BIN / REGISTROS_POR_CILINDRO),.        5430
 819   2  2                      END,.                                                 5440
 820   2  1                 END,.                                                      5450

 821   2  0  END CALCULA_CILINDRO_ALVO,.                                              5460
```

```
STMT LEV NT

                    /*************************************************************      5470
                     * ROTINA PARA POSICIONAR EM DETALHES_DE_ALOCACAO O MESMO  *      5480
                     *         ARQUIVO QUE ESTA' EM DETALHES_DE_ARQUIVOS       *      5490
                     *************************************************************/     5500

 822   1  0  POSICIONA_P.. PROC,.                                                       5510

 823   2  0            DO P = 1 TO V WHILE (DETALHES_DE_ARQUIVOS.ARQUIVO (INDICE)      5520
                                         NE DETALHES_DE_ALOCACAO.ARQUIVO  (P)),.     5530
 824   2  1            END,.                                                          5540

 825   2  0  END POSICIONA_P,.                                                        5550

                    /*************************************************************      5560
                     *   ROTINA PARA CALCULAR O TAMANHO DO EXTENT EM TRILHAS   *      5570
                     *    A PARTIR DAS TRILHAS DE INICIO E DE FIM DO EXTENT    *      5580
                     *************************************************************/     5590

 826   1  0  CALCULA_TAMANHO_DO_EXTENT.. PROC,.                                       5600

 827   2  0            TRACK_INICIAL                 = 19 * CILINDRO_INICIAL (P) +   5610
                                                           TRILHA_INICIAL (P),.      5620
 828   2  0            TRACK_FINAL                   = 19 * CILINDRO_FINAL (P)   +   5630
                                                           TRILHA_FINAL (P)  ,.      5640
 829   2  0            TAMANHO_DO_EXTENT_EM_TRILHAS = TRACK_FINAL - TRACK_INICIAL
                                                                          + 1,.      5650
                                                                                     5660

 830   2  0  END CALCULA_TAMANHO_DO_EXTENT,.                                         5670
```

```
STMT LEV NT

                    /*************************************************************      5680
                     * ROTINA PARA CORRIGIR P, DEVIDO AO ARQUIVO ESTAR ALOCADO *      5690
                     *                  EM MAIS DE UM EXTENT                   *      5700
                     *************************************************************/     5710

 831   1  0  CORRIGE_P.. PROC,.                                                         5720

 832   2  0        TRILHAS_SOMADAS_DOS_EXTENTS = 0,.                                   5730
 833   2  0        REGISTROS_POR_TRILHA = REG_LOGICOS_POR_BLOCO (INDICE) *             5740
                                          BLOCOS_FISICOS_POR_TRILHA (INDICE),.       5750
 834   2  0        ENDERECO_EM_TRILHAS =                                               5760
                            FLOOR (ENDERECO_BIN / REGISTROS_POR_TRILHA),.            5770

 835   2  0        GUARDA_P = P,.     /* PARA POSSIVEL REPOSICIONAMENTO DE P */        5780

 836   2  0        DO P = P TO (P + DETALHES_DE_ALOCACAO.TOTAL (P) - 1)                5790
                   WHILE (ENDERECO_EM_TRILHAS GT TRILHAS_SOMADAS_DOS_EXTENTS),.      5800

 837   2  1        CALL CALCULA_TAMANHO_DO_EXTENT,.                                    5810
 838   2  1        TRILHAS_SOMADAS_DOS_EXTENTS =                                       5820
                           TRILHAS_SOMADAS_DOS_EXTENTS +                             5830
                                    TAMANHO_DO_EXTENT_EM_TRILHAS,.                   5840
 839   2  1        END,.                                                               5850

 840   2  0        IF P GT GUARDA_P                                                    5860
                   THEN P = P - 1,.  /* REPOSICIONAMENTO DO P DEVIDO AO 'DO' */        5870

 841   2  0        IF ENDERECO_EM_TRILHAS GT TRILHAS_SOMADAS_DOS_EXTENTS               5880
                   THEN DO,.                                                         5890

 842   2  1             PUT SKIP EDIT                                                5900
                        ('REGISTRO COM ENDERECO NAO COMPATIVEL COM'                  5910
                          CAT ' TAMANHO DO ARQUIVO. FUNCAO - '                       5920
                          CAT LOG_DO_TOTAL.FUNCAO CAT '. TASK - '                    5930
                          CAT LOG_DO_TOTAL.TASK CAT '. ARQUIVO - '                   5940
                          CAT LOG_DO_TOTAL.ARQUIVO CAT '.') (A),.                    5950
 843   2  1             PUT SKIP EDIT ('IMAGEM DO REGISTRO - '                       5960
                          CAT SUBSTR (IMAGEM_DO_REGISTRO, 1,                         5970
                                (TAMANHO_DO_REGISTRO - 48)))          (A),.        5980
 844   2  1             PUT SKIP EDIT ('ENDERECO EM TRILHAS = '                      5990
                          CAT ENDERECO_EM_TRILHAS)          (A, P 'ZZZ.ZZ9'),.     6000
 845   2  1             PUT SKIP EDIT ('TRILHAS SOMADAS DOS EXTENTS = '              6010
                          CAT TRILHAS_SOMADAS_DOS_EXTENTS)   (A, P 'ZZZ.ZZ9'),.     6020
 846   2  1             TOTAL_ERROS_EM_ENDERECOS =                                   6030
                                    TOTAL_ERROS_EM_ENDERECOS + 1,.                   6040
 847   2  1             CALL PLIDUMP,.          /*    REATIVADO EM 26/09/78    */  6050
 848   2  1             CALL ESTAT,.            /*    REATIVADO EM 26/09/78    */  6060
 849   2  1             STOP,.                  /*    REATIVADO EM 26/09/78    */  6070
 850   2  1             END,.                                                       6080

 851   2  0  END CORRIGE_P,.                                                        6090
```

```
STMT LEV NT

                        /*************************************************************     6100
                         * ROTINA QUE COMPUTA A CADA I/O O TEMPO GASTO, CILINDROS  *     6110
                         *          DESLOCADOS E BYTES TRANSMITIDOS                *     6120
                         *************************************************************/    6130

852   1  0  PERFORMA_I_O.. PROC (J1, CILINDRO_ALVO1),.                                     6140

853   2  0            DCL  J1                 BIN FIXED (31),                             6150
                           CILINDRO_ALVO1          PIC '999',.                           6160

854   2  0            IF CHAVE_PENDENTE = 1                                               6170

                      THEN INDICE = 0,.                                                   6180

855   2  0            ELSE IF CHAVE_VINCULO = 1                                           6190

                           THEN INDICE = M,.                                              6200

856   2  0                ELSE INDICE = K,.                                              6210

857   2  0            E = E + DETALHES_DE_ARQUIVOS.COMPRIMENT_REG_LOGICO (INDICE),.       6220

858   2  0            IF CONDICAO_ATUAL_DO_BUFFER.VOLUME (J1) = ' ' OR I_O = 'I'          6230

                        /* PRIMEIRA UTILIZACAO DO BUFFER OU OPERACAO DE LEITURA */        6240

                      THEN DO,.                                                           6250

859   2  1                DO O = 1 TO X WHILE (CONDICAO_ATUAL_DO_BRACO.VOLUME (O)         6260
                                    NE DETALHES_DE_ALOCACAO.VOLUME (P)),.                 6270
860   2  2                END,.                                                          6280
861   2  1                END,.                                                          6290

862   2  0            ELSE DO,.     /* GRAVACAO DE UM BUFFER COM WRITE PENDENTE */        6300

863   2  1                DO O = 1 TO X WHILE (CONDICAO_ATUAL_DO_BRACO.VOLUME (O)         6310
                                 NE CONDICAO_ATUAL_DO_BUFFER.VOLUME (J1)),.             6320
864   2  2                END,.                                                          6330
865   2  1                END,.                                                          6340

866   2  0            IF O = X + 1                                                        6350
                      THEN DO,.                                                           6360

867   2  1                PUT SKIP EDIT                                                  6370
                          ('VOLUME NO LOG DO TOTAL NAO RECONHECIDO - '                    6380
                            CAT CONDICAO_ATUAL_DO_BUFFER.VOLUME (J1)) (A),.               6390
868   2  1                TOTAL_VOLS_NAO_RECONHECIDOS =                                  6400
                                          TOTAL_VOLS_NAO_RECONHECIDOS + 1,.               6410
869   2  1                CALL PLIDUMP,.                                                 6420
870   2  1                CALL ESTAT,.                                                   6430
871   2  1                STOP,.                                                         6440
872   2  1                END,.                                                          6450
```

```
STMT LEV NT

 873   2  0            ELSE DO;.                                                    6460

 874   2  1                 IF CONDICAO_ATUAL_DO_BRACO.CILINDRO (0) NE               6470
                                                              CILINDRO_ALVO1        6480
                           THEN D = D + 30.0;.                                      6490

 875   2  1                 D = D + 8.4;.                                            6500

 876   2  1                 C = C + CILINDRO_ALVO1 -                                 6510
                                   CONDICAO_ATUAL_DO_BRACO.CILINDRO (0);.           6520
 877   2  1                 IF C LT 0                                                6530
                           THEN C = -C;.                                            6540
 878   2  1                 CONDICAO_ATUAL_DO_BRACO.CILINDRO (0) = CILINDRO_ALVO1;.  6550
 879   2  1                 END;.                                                    6560

 880   2  0            CHAVE_PENDENTE = 0;.                                         6570
 881   2  0            I_O = ' ';.                                                  6580

 882   2  0  END PERFORMA_I_O;.                                                     6590
```

```
STMT LEV NT

                    /*************************************************************      6600
                    * ROTINA PARA COMPUTAR I/O DA REDE PARA O BUFFER, E ATUA- *      6610
                    *                LIZAR AS INFORMACOES DESTE                *      6620
                    *************************************************************/     6630

 883   1   0  TRAZ_O_BLOCO.. PROC,.                                                    6640

 884   2   0            IF DETALHES_DE_FUNCOES.LEITURA_OU_GRAVACAO (I) = 'L'          6650
                      THEN DO,.                                                       6660

 885   2   1               I_O = 'I',.                                                6670
 886   2   1               CALL CALCULA_CILINDRO_ALVO (J, CILINDRO_ALVO),.            6680
 887   2   1               CALL PERFORMA_I_O (J, CILINDRO_ALVO),.                     6690
 888   2   1               END,.                                                      6700

 889   2   0            INDICE1 = J,.                                                 6710

 890   2   0  ATUALIZA_BUFFER.. ENTRY,.                                              6720

 891   2   0            CONDICAO_ATUAL_DO_BUFFER.ARQUIVO          (INDICE1) =         6730
                                                                PARAMETROS.ARQUIVO,.  6740
 892   2   0            CONDICAO_ATUAL_DO_BUFFER.VOLUME           (INDICE1) =         6750
                                                   DETALHES_DE_ALOCACAO.VOLUME (P),.  6760
 893   2   0            ENDERECO_INICIAL_BIN                      (INDICE1) =         6770
                                                    REG_LOGICOS_POR_BLOCO (INDICE) *  6780
                          FLOOR (ENDERECO_BIN / REG_LOGICOS_POR_BLOCO (INDICE)),.     6790
 894   2   0            ENDERECO_FINAL_BIN                        (INDICE1) =         6800
                                                     ENDERECO_INICIAL_BIN (INDICE1) + 6810
                                                 REG_LOGICOS_POR_BLOCO (INDICE) - 1,. 6820
 895   2   0            CONDICAO_ATUAL_DO_BUFFER.BLOCO_FISICO     (INDICE1) =         6830
                                                       1 + FLOOR (ENDERECO_BIN /     6840
                                                    REG_LOGICOS_POR_BLOCO (INDICE)),. 6850
 896   2   0            CONDICAO_ATUAL_DO_BUFFER.TRILHA           (INDICE1) =         6860
                                                     FLOOR (BLOCO_FISICO.(INDICE1) / 6870
                                                BLOCOS_FISICOS_POR_TRILHA (INDICE)),. 6880
 897   2   0            CONDICAO_ATUAL_DO_BUFFER.CILINDRO         (INDICE1) =         6890
                                                                CILINDRO_ALVO,.       6900
 898   2   0            CONDICAO_ATUAL_DO_BUFFER.MOMENTO_DO_ULTIMO_USO                6910
                                                               (INDICE1) =           6920
                                                                    DATE CAT TIME,.   6930
 899   2   0            IF DETALHES_DE_FUNCOES.LEITURA_OU_GRAVACAO    (I) = 'G'       6940
                      THEN CONDICAO_ATUAL_DO_BUFFER.TEM_WRITE_PENDENTE              6950
                                                               (INDICE1) = 'S',.     6960

 900   2   0            CALL ESTATISTICAS,.                                           6970

 901   2   0  END TRAZ_O_BLOCO,.                                                     6980
```

```
STMT LEV NT

                    /*************************************************************      0010
                     *         ROTINA PARA IMPLEMENTACAO DOS CONTADORES           *      0020
                     *************************************************************/     0030

902   1  0  ESTATISTICAS.. PROC,.                                                        0040

903   2  0     IF CHAVE_ESTATISTICAS = 1 THEN                                           0050
               DO,.          /* ISSO PORQUE QUANDO A CHAVE ESTIVER DESLIGADA, SO'      0060
                               °SERAO FEITAS ESTATISTICAS NOS CONTADORES GERAIS */     0070

904   2  1     IF (DETALHES_DE_FUNCOES.TIPO_DE_GRAVACAO (I) NE 'N' OR                   0080
                                      CHAVE_SINCRONISMO NE  0) THEN                    0090
               DO,.               /* O MESMO EFEITO PARA SINCRONISMOS INVALIDOS */     0100

905   2  2     CALL POSICIONA_INDICES (IND_T, IND_B, IND_A, IND_V),.                    0110

906   2  2     IF CHAVE_SINCRONISMO LT 2 THEN                                           0120
               DO,.     /* 0 - FUNCAO DE LEITURA OU GRAVACAO                  */       0130
                        /* 1 - FUNCAO DE SINCRONISMO APROPRIANDO LIBERACAO DE  */       0140
                        /*     BUFFER                                         */       0150
                        /* 2 - FUNCAO DE SINCRONISMO APROPIANDO A FUNCAO EM SI */       0160
                        /*     - SO' APROPRIA NOS CONTADORES GERAIS E POR TASK */       0170

907   2  3     CONTADORES_POR_BUFFER.TOTAL_REGISTROS (IND_B)                  =          0180
               CONTADORES_POR_BUFFER.TOTAL_REGISTROS (IND_B)                  +          0190
                                                                             TREG,.     0200
908   2  3     CONTADORES_POR_BUFFER.TOTAL_ACESSOS (IND_B)                    =          0210
               CONTADORES_POR_BUFFER.TOTAL_ACESSOS (IND_B)                    + A,.      0220
909   2  3     CONTADORES_POR_BUFFER.TOTAL_DESLOCAMENTOS (IND_B)              =          0230
               CONTADORES_POR_BUFFER.TOTAL_DESLOCAMENTOS (IND_B)              + B,.      0240
910   2  3     CONTADORES_POR_BUFFER.TOTAL_NAO_DESLOCAMENTOS (IND_B)          =          0250
               CONTADORES_POR_BUFFER.TOTAL_NAO_DESLOCAMENTOS (IND_B)          + S,.      0260
911   2  3     CONTADORES_POR_BUFFER.TOTAL_ACESSOS_COM_DESLOCAMENTO (IND_B)   =          0270
               CONTADORES_POR_BUFFER.TOTAL_ACESSOS_COM_DESLOCAMENTO (IND_B)   + T,.      0280
912   2  3     CONTADORES_POR_BUFFER.TOTAL_ACESSOS_SEM_DESLOCAMENTO (IND_B)   =          0290
               CONTADORES_POR_BUFFER.TOTAL_ACESSOS_SEM_DESLOCAMENTO (IND_B)   + U,.      0300
913   2  3     CONTADORES_POR_BUFFER.TOTAL_CILINDROS_DESLOCADOS (IND_B)       =          0310
               CONTADORES_POR_BUFFER.TOTAL_CILINDROS_DESLOCADOS (IND_B)       + C,.      0320
914   2  3     CONTADORES_POR_BUFFER.TOTAL_TEMPO_DE_DESLOCAMENTO (IND_B)      =          0330
               CONTADORES_POR_BUFFER.TOTAL_TEMPO_DE_DESLOCAMENTO (IND_B)      + D,.      0340
915   2  3     CONTADORES_POR_BUFFER.TOTAL_BYTES_TRANSMITIDOS (IND_B)         =          0350
               CONTADORES_POR_BUFFER.TOTAL_BYTES_TRANSMITIDOS (IND_B)         + E,.      0360

916   2  3     CONTADORES_POR_ARQUIVO.TOTAL_REGISTROS (IND_A)                 =          0370
               CONTADORES_POR_ARQUIVO.TOTAL_REGISTROS (IND_A)                 +          0380
                                                                             TREG,.     0390
917   2  3     CONTADORES_POR_ARQUIVO.TOTAL_ACESSOS (IND_A)                   =          0400
               CONTADORES_POR_ARQUIVO.TOTAL_ACESSOS (IND_A)                   + A,.      0410
918   2  3     CONTADORES_POR_ARQUIVO.TOTAL_DESLOCAMENTOS (IND_A)             =          0420
               CONTADORES_POR_ARQUIVO.TOTAL_DESLOCAMENTOS (IND_A)             + B,.      0430
919   2  3     CONTADORES_POR_ARQUIVO.TOTAL_NAO_DESLOCAMENTOS (IND_A)         =          0440
               CONTADORES_POR_ARQUIVO.TOTAL_NAO_DESLOCAMENTOS (IND_A)         + S,.      0450
920   2  3     CONTADORES_POR_ARQUIVO.TOTAL_ACESSOS_COM_DESLOCAMENTO (IND_A) =           0460
```

```
STMT LEV NT

                    CONTADORES_POR_ARQUIVO.TOTAL_ACESSOS_COM_DESLOCAMENTO (IND_A) + T,.      0470
 921    2  3        CONTADORES_POR_ARQUIVO.TOTAL_ACESSOS_SEM_DESLOCAMENTO (IND_A) =          0480
                    CONTADORES_POR_ARQUIVO.TOTAL_ACESSOS_SEM_DESLOCAMENTO (IND_A) + U,.      0490
 922    2  3        CONTADORES_POR_ARQUIVO.TOTAL_CILINDROS_DESLOCADOS (IND_A)     =          0500
                    CONTADORES_POR_ARQUIVO.TOTAL_CILINDROS_DESLOCADOS (IND_A)     + C,.      0510
 923    2  3        CONTADORES_POR_ARQUIVO.TOTAL_TEMPO_DE_DESLOCAMENTO (IND_A)    =          0520
                    CONTADORES_POR_ARQUIVO.TOTAL_TEMPO_DE_DESLOCAMENTO (IND_A)    + D,.      0530
 924    2  3        CONTADORES_POR_ARQUIVO.TOTAL_BYTES_TRANSMITIDOS (IND_A)       =          0540
                    CONTADORES_POR_ARQUIVO.TOTAL_BYTES_TRANSMITIDOS (IND_A)       + E,.      0550

 925    2  3        CONTADORES_POR_VOLUME.TOTAL_REGISTROS (IND_V)                 =          0560
                    CONTADORES_POR_VOLUME.TOTAL_REGISTROS (IND_V)                 +          0570
                                                                                  TREG,.     0580
 926    2  3        CONTADORES_POR_VOLUME.TOTAL_ACESSOS (IND_V)                   =          0590
                    CONTADORES_POR_VOLUME.TOTAL_ACESSOS (IND_V)                   + A,.      0600
 927    2  3        CONTADORES_POR_VOLUME.TOTAL_DESLOCAMENTOS (IND_V)             =          0610
                    CONTADORES_POR_VOLUME.TOTAL_DESLOCAMENTOS (IND_V)             + B,.      0620
 928    2  3        CONTADORES_POR_VOLUME.TOTAL_NAO_DESLOCAMENTOS (IND_V)         =          0630
                    CONTADORES_POR_VOLUME.TOTAL_NAO_DESLOCAMENTOS (IND_V)         + S,.      0640
 929    2  3        CONTADORES_POR_VOLUME.TOTAL_ACESSOS_COM_DESLOCAMENTO (IND_V)  =          0650
                    CONTADORES_POR_VOLUME.TOTAL_ACESSOS_COM_DESLOCAMENTO (IND_V)  + T,.      0660
 930    2  3        CONTADORES_POR_VOLUME.TOTAL_ACESSOS_SEM_DESLOCAMENTO (IND_V)  =          0670
                    CONTADORES_POR_VOLUME.TOTAL_ACESSOS_SEM_DESLOCAMENTO (IND_V)  + U,.      0680
 931    2  3        CONTADORES_POR_VOLUME.TOTAL_CILINDROS_DESLOCADOS (IND_V)      =          0690
                    CONTADORES_POR_VOLUME.TOTAL_CILINDROS_DESLOCADOS (IND_V)      + C,.      0700
 932    2  3        CONTADORES_POR_VOLUME.TOTAL_TEMPO_DE_DESLOCAMENTO (IND_V)     =          0710
                    CONTADORES_POR_VOLUME.TOTAL_TEMPO_DE_DESLOCAMENTO (IND_V)     + D,.      0720
 933    2  3        CONTADORES_POR_VOLUME.TOTAL_BYTES_TRANSMITIDOS (IND_V)        =          0730
                    CONTADORES_POR_VOLUME.TOTAL_BYTES_TRANSMITIDOS (IND_V)        + E,.      0740
 934    2  3        END,.                                                                    0750

 935    2  2        CONTADORES_POR_TASK.TOTAL_REGISTROS (IND_T)                   =          0760
                    CONTADORES_POR_TASK.TOTAL_REGISTROS (IND_T)                   +          0770
                                                                                  TREG,.     0780
 936    2  2        CONTADORES_POR_TASK.TOTAL_ACESSOS (IND_T)                     =          0790
                    CONTADORES_POR_TASK.TOTAL_ACESSOS (IND_T)                     + A,.      0800
 937    2  2        CONTADORES_POR_TASK.TOTAL_DESLOCAMENTOS (IND_T)               =          0810
                    CONTADORES_POR_TASK.TOTAL_DESLOCAMENTOS (IND_T)               + B,.      0820
 938    2  2        CONTADORES_POR_TASK.TOTAL_NAO_DESLOCAMENTOS (IND_T)           =          0830
                    CONTADORES_POR_TASK.TOTAL_NAO_DESLOCAMENTOS (IND_T)           + S,.      0840
 939    2  2        CONTADORES_POR_TASK.TOTAL_ACESSOS_COM_DESLOCAMENTO (IND_T)    =          0850
                    CONTADORES_POR_TASK.TOTAL_ACESSOS_COM_DESLOCAMENTO (IND_T)    + T,.      0860
 940    2  2        CONTADORES_POR_TASK.TOTAL_ACESSOS_SEM_DESLOCAMENTO (IND_T)    =          0870
                    CONTADORES_POR_TASK.TOTAL_ACESSOS_SEM_DESLOCAMENTO (IND_T)    + U,.      0880
 941    2  2        CONTADORES_POR_TASK.TOTAL_CILINDROS_DESLOCADOS (IND_T)        =          0890
                    CONTADORES_POR_TASK.TOTAL_CILINDROS_DESLOCADOS (IND_T)        + C,.      0900
 942    2  2        CONTADORES_POR_TASK.TOTAL_TEMPO_DE_DESLOCAMENTO (IND_T)       =          0910
                    CONTADORES_POR_TASK.TOTAL_TEMPO_DE_DESLOCAMENTO (IND_T)       + D,.      0920
 943    2  2        CONTADORES_POR_TASK.TOTAL_BYTES_TRANSMITIDOS (IND_T)          =          0930
                    CONTADORES_POR_TASK.TOTAL_BYTES_TRANSMITIDOS (IND_T)          + E,.      0940

 944    2  2        END,.                                                                    0950

 945    2  1        END,.                                                                    0960
```

```
STMT LEV NT

 946   2  0     CONTADORES_GERAIS.TOTAL_REGISTROS                =          0970
               CONTADORES_GERAIS.TOTAL_REGISTROS                +          0980
                                                                TREG,.     0990
 947   2  0     CONTADORES_GERAIS.TOTAL_ACESSOS                  =          1000
               CONTADORES_GERAIS.TOTAL_ACESSOS                  + A,.      1010
 948   2  0     CONTADORES_GERAIS.TOTAL_DESLOCAMENTOS            =          1020
               CONTADORES_GERAIS.TOTAL_DESLOCAMENTOS            + B,.      1030
 949   2  0     CONTADORES_GERAIS.TOTAL_NAO_DESLOCAMENTOS        =          1040
               CONTADORES_GERAIS.TOTAL_NAO_DESLOCAMENTOS        + S,.      1050
 950   2  0     CONTADORES_GERAIS.TOTAL_ACESSOS_COM_DESLOCAMENTO =          1060
               CONTADORES_GERAIS.TOTAL_ACESSOS_COM_DESLOCAMENTO + T,.      1070
 951   2  0     CONTADORES_GERAIS.TOTAL_ACESSOS_SEM_DESLOCAMENTO =          1080
               CONTADORES_GERAIS.TOTAL_ACESSOS_SEM_DESLOCAMENTO + U,.      1090
 952   2  0     CONTADORES_GERAIS.TOTAL_CILINDROS_DESLOCADOS     =          1100
               CONTADORES_GERAIS.TOTAL_CILINDROS_DESLOCADOS     + C,.      1110
 953   2  0     CONTADORES_GERAIS.TOTAL_TEMPO_DE_DESLOCAMENTO    =          1120
               CONTADORES_GERAIS.TOTAL_TEMPO_DE_DESLOCAMENTO    + D,.      1130
 954   2  0     CONTADORES_GERAIS.TOTAL_BYTES_TRANSMITIDOS       =          1140
               CONTADORES_GERAIS.TOTAL_BYTES_TRANSMITIDOS       + E,.      1150

 955   2  0     A, B, C, D, E, S, T, U, TREG                     = 0,.      1160

 956   2  0   END ESTATISTICAS,.                                           1170
```

```
STMT LEV NT

                    /*************************************************************      1180
                     * ROTINA PARA POSICIONAR CORRETAMENTE OS INDICES DOS VETO-*      1190
                     * RES DE ACUMULADORES DE ESTATISTICAS, INCREMENTADOS AO - *      1200
                     *    FINAL DE CADA PROCESSAMENTO DE UM REGISTRO DO LOG.   *      1210
                     *************************************************************/     1220

 957   1  0  POSICIONA_INDICES.. PROC (IND_T1, IND_B1, IND_A1, IND_V1),.                1230

 958   2  0            DCL (IND_T1, IND_B1, IND_A1, IND_V1)       BIN FIXED (31),.     1240

 959   2  0            DO IND_T1 = 1 TO TAMANHO_VETOR_TASKS WHILE                      1250
                       (LOG_DO_TOTAL.TASK NE CONTADORES_POR_TASK.NOME_DA_TASK          1260
                                                              (IND_T1)),.              1270
 960   2  1            IF CONTADORES_POR_TASK.NOME_DA_TASK (IND_T1) = '       '        1280
                       THEN DO,.                                                       1290

 961   2  2                 CONTADORES_POR_TASK.NOME_DA_TASK (IND_T1) =                1300
                                                          LOG_DO_TOTAL.TASK,.          1310
 962   2  2                 IND_T1 = IND_T1 - 1,.  /* PARA SAIR CORRETAMENTE POSI-     1320
                                                     CIONADO DO DO-WHILE       */     1330
 963   2  2                 END,.                                                      1340

 964   2  1            END,.                                                           1350

 965   2  0            IF IND_T1 = TAMANHO_VETOR_TASKS + 1                             1360
                       THEN DO,.                                                       1370

 966   2  1                 PUT SKIP EDIT ('O NUMERO DE TASKS E'' MAIOR QUE O VALO'    1380
                           CAT 'R ESTIMADO PARA O TAMANHO DO VETOR DE CONTADORES '    1390
                           CAT 'POR TASKS. ') (A),.                                   1400
 967   2  1                 PUT SKIP EDIT ('PROGRAMA ABD30100 ESTA'' CANCELADO. RE'    1410
                           CAT 'SSUBMETA AUMENTANDO O VALOR DO TAMANHO DO VETOR.')    1420
                                                                    (A),.             1430
 968   2  1                 CALL PLIDUMP,.                                             1440
 969   2  1                 CALL ESTAT,.                                               1450
 970   2  1                 STOP,.                                                     1460
 971   2  1                 END,.                                                      1470

 972   2  0            DO IND_B1 = 1 TO Y WHILE                                        1480
                      ((CONDICAO_ATUAL_DO_BUFFER.BUFFER (INDICE1) NE                  1490
                        CONTADORES_POR_BUFFER.NOME_DO_BUFFER (IND_B1)) OR             1500
                       (CONDICAO_ATUAL_DO_BUFFER.VERSAO (INDICE1) NE                  1510
                        CONTADORES_POR_BUFFER.VERSAO_DO_BUFFER (IND_B1))),.           1520
 973   2  1            IF CONTADORES_POR_BUFFER.NOME_DO_BUFFER (IND_B1) = '     '      1530
                       THEN DO,.                                                      1540

 974   2  2                 CONTADORES_POR_BUFFER.NOME_DO_BUFFER (IND_B1) =            1550
                                 CONDICAO_ATUAL_DO_BUFFER.BUFFER (INDICE1),.        1560
 975   2  2                 CONTADORES_POR_BUFFER.VERSAO_DO_BUFFER (IND_B1) =          1570
                                 CONDICAO_ATUAL_DO_BUFFER.VERSAO (INDICE1),.        1580
 976   2  2                 IND_B1 = IND_B1 - 1,.  /* PARA SAIR CORRETAMENTE POSI-     1590
                                                     CIONADO DO DO-WHILE       */     1600
 977   2  2                 END,.                                                      1610
```

```
STMT LEV NT

 978   2  1            END,.                                                    1620

 979   2  0            DO IND_A1 = 1 TO Z WHILE                                 1630
                       (CONDICAO_ATUAL_DO_BUFFER.ARQUIVO (INDICE1)              1640
                            NE CONTADORES_POR_ARQUIVO.NOME_DO_ARQUIVO (IND_A1)),.  1650
 980   2  1            IF CONTADORES_POR_ARQUIVO.NOME_DO_ARQUIVO (IND_A1) = '    '  1660
                       THEN DO,.                                                1670

 981   2  2                 CONTADORES_POR_ARQUIVO.NOME_DO_ARQUIVO (IND_A1) =   1680
                                 CONDICAO_ATUAL_DO_BUFFER.ARQUIVO (INDICE1),.  1690
 982   2  2                 IND_A1 = IND_A1 - 1,.  /* PARA SAIR CORRETAMENTE POSI-  1700
                                                      CIONADO DO DO-WHILE     */ 1710
 983   2  2                 END,.                                               1720

 984   2  1            END,.                                                    1730

 985   2  0            DO IND_V1 = 1 TO X WHILE                                 1740
                       (CONDICAO_ATUAL_DO_BUFFER.VOLUME (INDICE1)               1750
                              NE CONTADORES_POR_VOLUME.NOME_DO_VOLUME (IND_V1)),.  1760
 986   2  1            IF CONTADORES_POR_VOLUME.NOME_DO_VOLUME (IND_V1) = '      '  1770
                       THEN DO,.                                                1780

 987   2  2                 CONTADORES_POR_VOLUME.NOME_DO_VOLUME (IND_V1) =     1790
                                 CONDICAO_ATUAL_DO_BUFFER.VOLUME (INDICE1),.   1800
 988   2  2                 IND_V1 = IND_V1 - 1,.  /* PARA SAIR CORRETAMENTE POSI-  1810
                                                      CIONADO DO DO-WHILE     */ 1820
 989   2  2                 END,.                                               1830

 990   2  1            END,.                                                    1840

 991   2  0       END POSICIONA_INDICES,.                                       1850
```

```
STMT LEV NT

                        /*************************************************************      1860
                        *              C A B E C A L H O   G E R A L                *      1870
                        *************************************************************/     1880

 992   1  0  CABECALHO_GERAL.. PROC,.                                                       1890

 993   2  0            CONTA_PAGINA = CONTA_PAGINA + 1,.                                    1900
 994   2  0            PUT PAGE EDIT ((132)'*') (COL (1), A),.                              1910
 995   2  0            PUT SKIP (2) EDIT ('TELECOMUNICACOES DO RIO DE JANEIRO - TE'         1920
                                  CAT 'LERJ', 'PAGINA - ', CONTA_PAGINA)                   1930
                                     (COL(19), A, COL (102), A, P'ZZZ9') ,.                1940
 996   2  0            PUT SKIP (2) EDIT ('MODELO DE SIMULACAO PARA ALOCACAO DE'            1950
                                  CAT ' ARQUIVOS DE UMA REDE TOTAL EM UM NO'               1960
                                  CAT 'MERO LIMITADO DE VOLUMES')                          1970
                                     (COL(19), A) ,.                                       1980
 997   2  0            PUT SKIP (2) EDIT ('* * *  E S T A T I S T I C A S  * * *')          1990
                                    (COL(19), A) ,.                                       2000
 998   2  0            PUT SKIP (2) EDIT ((132) '*') (COL (1), A),.                         2010

 999   2  0  END CABECALHO_GERAL,.                                                         2020

                        /*************************************************************      2030
                        *              IMPRESSAO DOS TOTAIS DE ERROS                *      2040
                        *************************************************************/     2050

1000   1  0  IMPRIME_TOTAIS_DE_ERROS.. PROC,.                                             2060

1001   2  0            PUT SKIP(5) EDIT('TOTAL DE FUNCOES NAO RECONHECIDAS  - '  ,        2070
                       TOTAL_FUNCOES_NAO_RECONHECIDAS)  (COL(10),A,P'Z.ZZZ.ZZ9') ,.       2080
1002   2  0            PUT SKIP(2) EDIT('TOTAL DE ARQUIVOS NAO RECONHECIDOS - '  ,        2090
                       TOTAL_ARQS_NAO_RECONHECIDOS)     (COL(10),A,P'Z.ZZZ.ZZ9') ,.       2100
1003   2  0            PUT SKIP(2) EDIT('TOTAL DE VINCULOS NAO RECONHECIDOS - '  ,        2110
                       TOTAL_VINCULOS_NAO_RECONHECIDOS) (COL(10),A,P'Z.ZZZ.ZZ9') ,.       2120
1004   2  0            PUT SKIP(2) EDIT('TOTAL DE BUFFERS NAO RECONHECIDOS  - '  ,        2130
                       TOTAL_BUFFERS_NAO_RECONHECIDOS)  (COL(10),A,P'Z.ZZZ.ZZ9') ,.       2140
1005   2  0            PUT SKIP(2) EDIT('TOTAL DE VOLUMES NAO RECONHECIDOS  - '  ,        2150
                       TOTAL_VOLS_NAO_RECONHECIDOS)     (COL(10),A,P'Z.ZZZ.ZZ9') ,.       2160
1006   2  0            PUT SKIP(2) EDIT('TOTAL DE ERROS EM ENDERECOS        - '  ,        2170
                       TOTAL_ERROS_EM_ENDERECOS)        (COL(10),A,P'Z.ZZZ.ZZ9') ,.       2180

1007   2  0  END IMPRIME_TOTAIS_DE_ERROS,.                                               2190
```

```
STMT LEV NT

                        /*************************************************************      2200
                         *                IMPRESSAO DOS TOTAIS POR TASK                *      2210
                         *************************************************************/     2220

1008   1  0  (NOZERODIVIDE).. IMPRIME_TOTAIS_POR_TASK.. PROC (I1),.                          2230

1009   2  0            DCL I1                                    BIN FIXED (31),.          2240

1010   2  0            PUT SKIP(5) EDIT('TOTAL DE REGISTROS LIDOS NO LOG    - '  ,          2250
                       CONTADORES_POR_TASK.TOTAL_REGISTROS (I1))                            2260
                                                 (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.          2270
1011   2  0            PUT SKIP(2) EDIT('TOTAL DE FUNCOES TIPO ''DE ACESSO''  - ',          2280
                       CONTADORES_POR_TASK.TOTAL_ACESSOS (I1))                              2290
                                                 (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.          2300
1012   2  0            PUT SKIP(2) EDIT('TOTAL DE ACESSOS COM DESLOCAMENTO  - '  ,          2310
                       CONTADORES_POR_TASK.TOTAL_ACESSOS_COM_DESLOCAMENTO (I1))             2320
                                                 (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.          2330
1013   2  0            PUT SKIP(2) EDIT('TOTAL DE ACESSOS SEM DESLOCAMENTO  - '  ,          2340
                       CONTADORES_POR_TASK.TOTAL_ACESSOS_SEM_DESLOCAMENTO (I1))             2350
                                                 (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.          2360
1014   2  0            PUT SKIP(2) EDIT('TOTAL DE DESLOCAMENTOS             - '  ,          2370
                       CONTADORES_POR_TASK.TOTAL_DESLOCAMENTOS (I1))                        2380
                                                 (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.          2390
1015   2  0            PUT SKIP(2) EDIT('TOTAL DE NAO-DESLOCAMENTOS         - '  ,          2400
                       CONTADORES_POR_TASK.TOTAL_NAO_DESLOCAMENTOS (I1))                    2410
                                                 (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.          2420
1016   2  0            PUT SKIP(2) EDIT('TOTAL DE BYTES TRANSMITIDOS        - '  ,          2430
                       CONTADORES_POR_TASK.TOTAL_BYTES_TRANSMITIDOS (I1))                   2440
                                                 (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.          2450
1017   2  0            PUT SKIP(2) EDIT('TOTAL DE CILINDROS DESLOCADOS      - '  ,          2460
                       CONTADORES_POR_TASK.TOTAL_CILINDROS_DESLOCADOS (I1))                 2470
                                                 (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.          2480
1018   2  0            PUT SKIP(2) EDIT('CILINDROS DESLOCADOS POR ACESSO    - '  ,          2490
                       CONTADORES_POR_TASK.TOTAL_CILINDROS_DESLOCADOS (I1) /                2500
                       CONTADORES_POR_TASK.TOTAL_ACESSOS (I1))                              2510
                                                 (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.          2520
1019   2  0            PUT SKIP(2) EDIT('TEMPO TOTAL DE DESLOCAMENTO (MSEG) - '  ,          2530
                       CONTADORES_POR_TASK.TOTAL_TEMPO_DE_DESLOCAMENTO (I1))                2540
                                                 (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9V,9') ,.        2550
1020   2  0            PUT SKIP(2) EDIT('TEMPO DE DESLOCAMENTO POR ACESSO   - '  ,          2560
                       CONTADORES_POR_TASK.TOTAL_TEMPO_DE_DESLOCAMENTO (I1) /               2570
                       CONTADORES_POR_TASK.TOTAL_ACESSOS (I1))                              2580
                                                 (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9V,9') ,.        2590

1021   2  0  END IMPRIME_TOTAIS_POR_TASK,.                                                  2600
```

```
STMT LEV NT

                        /*************************************************************      2610
                         *              IMPRESSAO DOS TOTAIS POR BUFFER              *      2620
                         *************************************************************/     2630

1022   1  0  (NOZERODIVIDE).. IMPRIME_TOTAIS_POR_BUFFER.: PROC (I1),.                        2640

1023   2  0            DCL I1                                        BIN FIXED (31),.       2650

1024   2  0            PUT SKIP(5) EDIT('TOTAL DE REGISTROS LIDOS NO LOG    - ' ,          2660
                       CONTADORES_POR_BUFFER.TOTAL_REGISTROS (I1))                          2670
                                                  (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.       2680
1025   2  0            PUT SKIP(2) EDIT('TOTAL DE FUNCOES TIPO ''DE ACESSO'' - ',          2690
                       CONTADORES_POR_BUFFER.TOTAL_ACESSOS (I1)) ,                          2700
                                                  (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.       2710
1026   2  0            PUT SKIP(2) EDIT('TOTAL DE ACESSOS COM DESLOCAMENTO  - ' ,          2720
                       CONTADORES_POR_BUFFER.TOTAL_ACESSOS_COM_DESLOCAMENTO (I1))           2730
                                                  (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.       2740
1027   2  0            PUT SKIP(2) EDIT('TOTAL DE ACESSOS SEM DESLOCAMENTO  - ' ,          2750
                       CONTADORES_POR_BUFFER.TOTAL_ACESSOS_SEM_DESLOCAMENTO (I1))           2760
                                                  (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.       2770
1028   2  0            PUT SKIP(2) EDIT('TOTAL DE DESLOCAMENTOS             - ' ,          2780
                       CONTADORES_POR_BUFFER.TOTAL_DESLOCAMENTOS (I1))                      2790
                                                  (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.       2800
1029   2  0            PUT SKIP(2) EDIT('TOTAL DE NAO-DESLOCAMENTOS         - ' ,          2810
                       CONTADORES_POR_BUFFER.TOTAL_NAO_DESLOCAMENTOS (I1))                  2820
                                                  (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.       2830
1030   2  0            PUT SKIP(2) EDIT('TOTAL DE BYTES TRANSMITIDOS        - ' ,          2840
                       CONTADORES_POR_BUFFER.TOTAL_BYTES_TRANSMITIDOS (I1))                 2850
                                                  (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.       2860
1031   2  0            PUT SKIP(2) EDIT('TOTAL DE CILINDROS DESLOCADOS      - ' ,          2870
                       CONTADORES_POR_BUFFER.TOTAL_CILINDROS_DESLOCADOS (I1))               2880
                                                  (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.       2890
1032   2  0            PUT SKIP(2) EDIT('CILINDROS DESLOCADOS POR ACESSO    - ' ,          2900
                       CONTADORES_POR_BUFFER.TOTAL_CILINDROS_DESLOCADOS (I1) /          2910
                       CONTADORES_POR_BUFFER.TOTAL_ACESSOS (I1))                            2920
                                                  (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.       2930
1033   2  0            PUT SKIP(2) EDIT('TEMPO TOTAL DE DESLOCAMENTO (MSEG) - ' ,          2940
                       CONTADORES_POR_BUFFER.TOTAL_TEMPO_DE_DESLOCAMENTO (I1))          2950
                                                  (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9V,9') ,.    2960
1034   2  0            PUT SKIP(2) EDIT('TEMPO DE DESLOCAMENTO POR ACESSO   - ' ,          2970
                       CONTADORES_POR_BUFFER.TOTAL_TEMPO_DE_DESLOCAMENTO (I1) /         2980
                       CONTADORES_POR_BUFFER.TOTAL_ACESSOS (I1))                            2990
                                                  (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9V,9') ,.    3000

1035   2  0  END IMPRIME_TOTAIS_POR_BUFFER,.                                                3010
```

```
STMT LEV NT

                        /*************************************************************      3020
                        *          IMPRESSAO DOS TOTAIS POR ARQUIVO                 *      3030
                        *************************************************************/     3040

1036   1   0  (NOZERODIVIDE).. IMPRIME_TOTAIS_POR_ARQUIVO.: PROC (I1),.                     3050

1037   2   0            DCL I1                                    BIN FIXED (31),.         3060

1038   2   0            PUT SKIP(5) EDIT('TOTAL DE REGISTROS LIDOS NO LOG    - '  ,        3070
                        CONTADORES_POR_ARQUIVO.TOTAL_REGISTROS (I1))                        3080
                                                  (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.           3090
1039   2   0            PUT SKIP(2) EDIT('TOTAL DE FUNCOES TIPO ''DE ACESSO''  - ',        3100
                        CONTADORES_POR_ARQUIVO.TOTAL_ACESSOS (I1),                          3110
                                                  (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.           3120
1040   2   0            PUT SKIP(2) EDIT('TOTAL DE ACESSOS COM DESLOCAMENTO  - '  ,        3130
                        CONTADORES_POR_ARQUIVO.TOTAL_ACESSOS_COM_DESLOCAMENTO (I1))         3140
                                                  (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.           3150
1041   2   0            PUT SKIP(2) EDIT('TOTAL DE ACESSOS SEM DESLOCAMENTO  - '  ,        3160
                        CONTADORES_POR_ARQUIVO.TOTAL_ACESSOS_SEM_DESLOCAMENTO (I1))         3170
                                                  (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.           3180
1042   2   0            PUT SKIP(2) EDIT('TOTAL DE DESLOCAMENTOS             - '  ,        3190
                        CONTADORES_POR_ARQUIVO.TOTAL_DESLOCAMENTOS (I1))                    3200
                                                  (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.           3210
1043   2   0            PUT SKIP(2) EDIT('TOTAL DE NAO-DESLOCAMENTOS         - '  ,        3220
                        CONTADORES_POR_ARQUIVO.TOTAL_NAO_DESLOCAMENTOS (I1))                3230
                                                  (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.           3240
1044   2   0            PUT SKIP(2) EDIT('TOTAL DE BYTES TRANSMITIDOS        - '  ,        3250
                        CONTADORES_POR_ARQUIVO.TOTAL_BYTES_TRANSMITIDOS (I1))               3260
                                                  (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.           3270
1045   2   0            PUT SKIP(2) EDIT('TOTAL DE CILINDROS DESLOCADOS      - '  ,        3280
                        CONTADORES_POR_ARQUIVO.TOTAL_CILINDROS_DESLOCADOS (I1))             3290
                                                  (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.           3300
1046   2   0            PUT SKIP(2) EDIT('CILINDROS DESLOCADOS POR ACESSO    - '  ,        3310
                        CONTADORES_POR_ARQUIVO.TOTAL_CILINDROS_DESLOCADOS (I1) /            3320
                        CONTADORES_POR_ARQUIVO.TOTAL_ACESSOS (I1))                          3330
                                                  (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.           3340
1047   2   0            PUT SKIP(2) EDIT('TEMPO TOTAL DE DESLOCAMENTO (MSEG) - '  ,        3350
                        CONTADORES_POR_ARQUIVO.TOTAL_TEMPO_DE_DESLOCAMENTO (I1))            3360
                                                  (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9V,9') ,.        3370
1048   2   0            PUT SKIP(2) EDIT('TEMPO DE DESLOCAMENTO POR ACESSO   - '  ,        3380
                        CONTADORES_POR_ARQUIVO.TOTAL_TEMPO_DE_DESLOCAMENTO (I1) /           3390
                        CONTADORES_POR_ARQUIVO.TOTAL_ACESSOS (I1))                          3400
                                                  (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9V,9') ,.        3410

1049   2   0  END IMPRIME_TOTAIS_POR_ARQUIVO,.                                             3420
```

```
STMT LEV NT

                        /*************************************************************      3430
                         *              IMPRESSAO DOS TOTAIS POR VOLUME              *      3440
                         *************************************************************/     3450

1050   1  0  (NOZERODIVIDE).. IMPRIME_TOTAIS_POR_VOLUME.. PROC (I1),.                       3460

1051   2  0            DCL I1                                        BIN FIXED (31),.       3470

1052   2  0            PUT SKIP(5) EDIT('TOTAL DE REGISTROS LIDOS NO LOG    - ' ,         3480
                       CONTADORES_POR_VOLUME.TOTAL_REGISTROS (I1))                         3490
                                                  (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.          3500
1053   2  0            PUT SKIP(2) EDIT('TOTAL DE FUNCOES TIPO ''DE ACESSO'' - ',          3510
                       CONTADORES_POR_VOLUME.TOTAL_ACESSOS (I1))                           3520
                                                  (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.          3530
1054   2  0            PUT SKIP(2) EDIT('TOTAL DE ACESSOS COM DESLOCAMENTO  - ' ,          3540
                       CONTADORES_POR_VOLUME.TOTAL_ACESSOS_COM_DESLOCAMENTO (I1))          3550
                                                  (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.          3560
1055   2  0            PUT SKIP(2) EDIT('TOTAL DE ACESSOS SEM DESLOCAMENTO  - ' ,          3570
                       CONTADORES_POR_VOLUME.TOTAL_ACESSOS_SEM_DESLOCAMENTO (I1))          3580
                                                  (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.          3590
1056   2  0            PUT SKIP(2) EDIT('TOTAL DE DESLOCAMENTOS             - ' ,          3600
                       CONTADORES_POR_VOLUME.TOTAL_DESLOCAMENTOS (I1))                     3610
                                                  (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.          3620
1057   2  0            PUT SKIP(2) EDIT('TOTAL DE NAO-DESLOCAMENTOS         - ' ,          3630
                       CONTADORES_POR_VOLUME.TOTAL_NAO_DESLOCAMENTOS (I1))                 3640
                                                  (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.          3650
1058   2  0            PUT SKIP(2) EDIT('TOTAL DE BYTES TRANSMITIDOS        - ' ,          3660
                       CONTADORES_POR_VOLUME.TOTAL_BYTES_TRANSMITIDOS (I1))                3670
                                                  (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.          3680
1059   2  0            PUT SKIP(2) EDIT('TOTAL DE CILINDROS DESLOCADOS      - ' ,          3690
                       CONTADORES_POR_VOLUME.TOTAL_CILINDROS_DESLOCADOS (I1))              3700
                                                  (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.          3710
1060   2  0            PUT SKIP(2) EDIT('CILINDROS DESLOCADOS POR ACESSO    - ' ,          3720
                       CONTADORES_POR_VOLUME.TOTAL_CILINDROS_DESLOCADOS (I1) /             3730
                       CONTADORES_POR_VOLUME.TOTAL_ACESSOS (I1))                           3740
                                                  (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.          3750
1061   2  0            PUT SKIP(2) EDIT('TEMPO TOTAL DE DESLOCAMENTO (MSEG) - ' ,          3760
                       CONTADORES_POR_VOLUME.TOTAL_TEMPO_DE_DESLOCAMENTO (I1))             3770
                                                (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9V,9') ,.        3780
1062   2  0            PUT SKIP(2) EDIT('TEMPO DE DESLOCAMENTO POR ACESSO   - ' ,          3790
                       CONTADORES_POR_VOLUME.TOTAL_TEMPO_DE_DESLOCAMENTO (I1) /            3800
                       CONTADORES_POR_VOLUME.TOTAL_ACESSOS (I1))                           3810
                                                (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9V,9') ,.        3820

1063   2  0  END IMPRIME_TOTAIS_POR_VOLUME,.                                               3830
```

```
. STMT LEV NT

                        /*************************************************************      3840
                         *                  IMPRESSAO DOS TOTAIS GERAIS              *      3850
                         *************************************************************/     3860

1064   1  0  (NOZERODIVIDE).. IMPRIME_TOTAIS_GERAIS.. PROC,.                                 3870

1065   2  0            PUT SKIP(5) EDIT('TOTAL DE REGISTROS LIDOS NO LOG    - '  ,          3880
                       CONTADORES_GERAIS.TOTAL_REGISTROS)                                   3890
                                                 (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.          3900
1066   2  0            PUT SKIP(2) EDIT('TOTAL DE FUNCOES TIPO ''DE ACESSO'' - ',          3910
                       CONTADORES_GERAIS.TOTAL_ACESSOS)                                     3920
                                                 (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.          3930
1067   2  0            PUT SKIP(2) EDIT('TOTAL DE ACESSOS COM DESLOCAMENTO  - '  ,          3940
                       CONTADORES_GERAIS.TOTAL_ACESSOS_COM_DESLOCAMENTO)                    3950
                                                 (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.          3960
1068   2  0            PUT SKIP(2) EDIT('TOTAL DE ACESSOS SEM DESLOCAMENTO  - '  ,          3970
                       CONTADORES_GERAIS.TOTAL_ACESSOS_SEM_DESLOCAMENTO)                    3980
                                                 (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.          3990
1069   2  0            PUT SKIP(2) EDIT('TOTAL DE DESLOCAMENTOS             - '  ,          4000
                       CONTADORES_GERAIS.TOTAL_DESLOCAMENTOS)                               4010
                                                 (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.          4020
1070   2  0            PUT SKIP(2) EDIT('TOTAL DE NAO-DESLOCAMENTOS         - '  ,          4030
                       CONTADORES_GERAIS.TOTAL_NAO_DESLOCAMENTOS)                           4040
                                                 (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.          4050
1071   2  0            PUT SKIP(2) EDIT('TOTAL DE BYTES TRANSMITIDOS        - '  ,          4060
                       CONTADORES_GERAIS.TOTAL_BYTES_TRANSMITIDOS)                          4070
                                                 (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.          4080
1072   2  0            PUT SKIP(2) EDIT('TOTAL DE CILINDROS DESLOCADOS      - '  ,          4090
                       CONTADORES_GERAIS.TOTAL_CILINDROS_DESLOCADOS)                        4100
                                                 (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.          4110
1073   2  0            PUT SKIP(2) EDIT('CILINDROS DESLOCADOS POR ACESSO    - '  ,          4120
                       CONTADORES_GERAIS.TOTAL_CILINDROS_DESLOCADOS /                       4130
                       CONTADORES_GERAIS.TOTAL_ACESSOS)                                     4140
                                                 (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9') ,.          4150
1074   2  0            PUT SKIP(2) EDIT('TEMPO TOTAL DE DESLOCAMENTO (MSEG) - '  ,          4160
                       CONTADORES_GERAIS.TOTAL_TEMPO_DE_DESLOCAMENTO)                       4170
                                              (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9V,9') ,.          4180
1075   2  0            PUT SKIP(2) EDIT('TEMPO DE DESLOCAMENTO POR ACESSO   - '  ,          4190
                       CONTADORES_GERAIS.TOTAL_TEMPO_DE_DESLOCAMENTO /                      4200
                       CONTADORES_GERAIS.TOTAL_ACESSOS)                                     4210
                                              (COL(10),A,P'Z.ZZZ.ZZZ.ZZ9V,9') ,.          4220

1076   2  0  END IMPRIME_TOTAIS_GERAIS,.                                                    4230
```

```
STMT LEV NT

                        /*************************************************************      4240
                        *              IMPRESSAO DOS RESULTADOS FINAIS                *      4250
                        *************************************************************/     4260

1077   1  0  ESTAT..     PROC,.                                                              4270

1078   2  0               CALL CABECALHO_GERAL,.                                            4280
1079   2  0               PUT SKIP (5) EDIT                                                 4290
                        ('***  T O T A I S   D E   E R R O S   ***') (COL (10),A,A),.        4300
1080   2  0               CALL IMPRIME_TOTAIS_DE_ERROS,.                                    4310

1081   2  0               DO I = 1 TO TAMANHO_VETOR_TASKS WHILE                             4320
                        (CONTADORES_POR_TASK.NOME_DA_TASK (I) NE '        '),.              4330

1082   2  1               CALL CABECALHO_GERAL,.                                            4340
1083   2  1               PUT SKIP (5) EDIT ('T A S K   -   ',                              4350
                                  CONTADORES_POR_TASK.NOME_DA_TASK (I))                    4360
                                                        (COL (10), A, A) ,.                4370
1084   2  1               CALL IMPRIME_TOTAIS_POR_TASK (I),.                                4380

1085   2  1               END,.                                                             4390

1086   2  0               DO J = 1 TO Y,.                                                   4400

1087   2  1               DO I = 1 TO Y,.                                                   4410
1088   2  2               IF ((CONTADORES_POR_BUFFER.NOME_DO_BUFFER (I) =                   4420
                                  CONDICAO_ATUAL_DO_BUFFER.BUFFER (J)) AND                 4430
                          (CONTADORES_POR_BUFFER.VERSAO_DO_BUFFER (I) =                    4440
                                  CONDICAO_ATUAL_DO_BUFFER.VERSAO (J)))                    4450
                                THEN DO,.                                                  4460

1089   2  3                    CALL CABECALHO_GERAL,.                                      4470
1090   2  3                    PUT SKIP (5) EDIT ('B U F F E R   -   ',                    4480
                                  CONTADORES_POR_BUFFER.NOME_DO_BUFFER (I),                4490
                                                'V E R S A O   -   ',                      4500
                                  CONTADORES_POR_BUFFER.VERSAO_DO_BUFFER (I))              4510
                                                (COL (10), A, A, COL (40), A, A),.         4520
1091   2  3                    CALL IMPRIME_TOTAIS_POR_BUFFER (I),.                        4530

1092   2  3                    END,.                                                       4540

1093   2  2               END,.                                                           4550

1094   2  1               END,.                                                           4560

1095   2  0               DO I = 1 TO Z WHILE                                             4570
                        (CONTADORES_POR_ARQUIVO.NOME_DO_ARQUIVO (I) NE '    '),.          4580

1096   2  1               CALL CABECALHO_GERAL,.                                          4590
1097   2  1               PUT SKIP (5) EDIT ('A R Q U I V O   -   ',                      4600
                                  CONTADORES_POR_ARQUIVO.NOME_DO_ARQUIVO (I))            4610
                                                        (COL (10), A, A) ,.                4620
1098   2  1               CALL IMPRIME_TOTAIS_POR_ARQUIVO (I),.                           4630
```

```
STMT LEV NT

1099   2  1            END,.                                                    4640

1100   2  0            DO I = 1 TO X WHILE                                      4650
                       (CONTADORES_POR_VOLUME.NOME_DO_VOLUME (I) NE '      '),.  4660

1101   2  1            CALL CABECALHO_GERAL,.                                   4670
1102   2  1            PUT SKIP (5) EDIT ('V O L U M E  -  ',                   4680
                                CONTADORES_POR_VOLUME.NOME_DO_VOLUME (I))       4690
                                                    (COL (10), A, A) ,.         4700
1103   2  1            CALL IMPRIME_TOTAIS_POR_VOLUME (I),.                     4710

1104   2  1            END,.                                                    4720

1105   2  0            CALL CABECALHO_GERAL,.                                   4730
1106   2  0            PUT SKIP (5) EDIT ('***  T O T A I S  G E R A I S  ***')  4740
                                                    (COL (10), A, A) ,.         4750
1107   2  0            CALL IMPRIME_TOTAIS_GERAIS,.                             4760

1108   2  0  END ESTAT,.                                                        4770


1109   1  0            CALL ABDFIM ('ABD30100',TCHERRO),.                       4780


1110   1  0  END ABD3C10,.                                                      4790
```

XII - BIBLIOGRAFIA

|1| MARTIN, James. Computer data-base organization. Englewood Cliffs, Prentice-Hall, 1975. 558 p. (Prentice-Hall Series in Automatic Computation)

|2| CINCOM SYSTEMS INC. OS TOTAL reference manual; application programming. Cincinnati, 1976. 1 v. (Publ. No. PO2-1236-00)