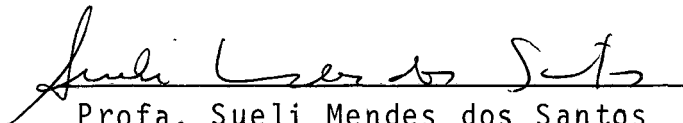


SUBSISTEMA DE GERÊNCIA E MANIPULAÇÃO DE
E/S EM VEÍCULOS DE ACESSO DIRETO PARA
UM MICROCOMPUTADOR

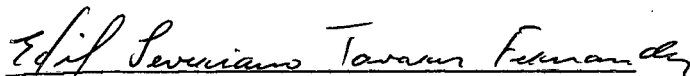
Henrique Mariano Costa do Amaral

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 OBTENÇÃO
DO GRAU DE MESTRE EM CIÊNCIAS (M.Sc.)

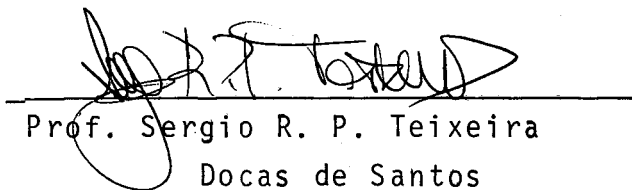
Aprovada por:



Profa. Sueli Mendes dos Santos
(Presidente)



Prof. Edil S. Tavares Fernandes
COPPE/UFRJ



Prof. Sergio R. P. Teixeira
Docas de Santos

RIO DE JANEIRO, RJ - BRASIL

ABRIL DE 1982

AMARAL, HENRIQUE MARIANO COSTA DO

Subsistema de Gerência de Manipulação de E/S em Veículos de Acesso Direto para um Microcomputador (Rio de Janeiro) 1982.

IX , 141 p. , 29,7 cm (COPPE-UFRJ, M.Sc., Engenharia de Sistemas e Computação, 1982).

Tese - Univ. Fed. Rio de Janeiro. Fac. Engenharia
1. Sistemas Operacionais: Gerência de Arquivos I.COPPE/
UFRJ II.Título (série).

À Celeste, Romulo e Ricardo
Aos meus pais e aos meus avós

AGRADECIMENTOS

Em primeiro lugar agradeço a minha esposa, amiga e companheira Celeste, aos meus filhos Romulo e Ricardo, que com sua compreensão e estímulo, foram fundamentais para a realização deste trabalho.

Aos meus pais Socorro e Francisco, aos meus avós Eulina e Mariano pela orientação e esforços à minha formação, e aos amigos Pedro, Assis Mário, Tereza, Edmilson e Clea também pelo que contribuíram.

A seguir agradeço à Professora Sueli Mendes dos Santos, pela sua orientação dedicada e valiosa, que possibilitou o bom andamento desta tese.

Ao Prof. Edil Severiano Tavares Fernandes pela contribuição.

Aos amigos José Lavaquial Breitiger e Pedro Luis Maltheiros pelo incentivo, discussões e sugestões que deram durante todo o desenvolvimento do trabalho.

Ao Professor José Maria Cabral Marques, Reitor da Universidade Federal do Maranhão.

Ao senador e amigo José Sarney pelo que me possibilitou a realização deste trabalho.

Ao casal amigo, Roberto e Eduarda pela colaboração.

A Suely Klajman, pela contribuição prestada nos trabalhos de datilografia.

A CAPES e UEMA pela ajuda financeira.

RESUMO

Descreve-se a implementação de um Gerente de Arquivos altamente flexível e poderoso, que fará parte de um Sistema Operacional de microcomputador que está em desenvolvimento no Programa de Engenharia de Sistemas da COPPE/UFRJ.

O Gerente de Arquivo, denominado SGMES, possibilita a implementação de sistemas de arquivos tanto em disco flexíveis como em outros tipos de disco (panela - fixo ou removível e cartucho), uma vez que estes são totalmente transparentes, devido a existência no microcomputador de uma interface inteligente para discos, que faz o tratamento local dos diversos tipos de disco.

Algumas características não usuais em tais tipos de sistemas foram implementados, tais como escrita e leitura parcial de registros, já a nível de operações primitivas do gerente.

Como o SGMES rodará em ambiente de multiprogramação o mesmo possibilita a utilização, concomitante, de um mesmo arquivo por mais de um processo (inclusive de diferentes usuários), utilizando um monitor (implementado a nível do núcleo do sistema operacional) para realizar a exclusão mútua.

Uma outra característica do SGMES, pouco comum nos sistemas operacionais usados em microcomputadores, é a possibilidade efetiva de uso de arquivos organizados de modo sequencial indexado, com tamanho da tabela de chaves, pelo menos teoricamente, sem limite (depende da área disponível para a mesma ser armazenada em disco e não da área disponível em memória).

O que é fundamental nos diversos tipos de tratamento dos registros e arquivos (como por exemplo escrita e leitura

parcial, blocagem de registros, registros de tamanho variável, etc) é o esquema de utilização dos buffers internos do gerente, para o compartilhamento dos mesmos com os diversos processos que concorrem para obtenção de recursos de entrada e saída.

Os compromissos que foram adotados para o desenvolvimento deste sistema, simplicidade, confiabilidade e portabilidade, tornam o mesmo altamente poderoso em relação a todos os sistemas até hoje desenvolvidos para microcomputadores do porte do Carcarã. Assim é que ele possibilita o uso de múltiplos canais de E/S com uso de múltiplas unidades, podendo alcançar capacidade de armazenamento superior a 64 MBytes.

ABSTRACT

The implementation of a powerful and flexible File Management System is presented, which is part of an Operating System of a microcomputer actually being developed at the Programa de Engenharia de Sistemas at COPPE/UFRJ (Federal University of Rio de Janeiro).

The File Management System allows the implementation of files on flexible disks and fixed or removable disk packs as well. Those are totally transparent to the user due to the interface for disks which makes a local treatment for the different types of disks.

Some features usually unavailable, like partial reading and partial writing of registers, have also been implemented.

The system which runs in a multiprogramming environment file by more than one process of different users, through the use of a monitor which does the mutual exclusion, implemented in the kernel of the Operating System.

Another uncommon feature in Operatins Systems for microcomputers, is the effective possibility of using indexed sequentially organized files with theoretically unlimited size of table keys.

The scheme of use of the File Management System internal buffers is basic to the treatment of registers and files of all the process concurring for I/O resources.

The File Management System is very powerful when compared with other microcomputer systems about the same size due to the policy adopted in its development: simplicity, reliability and portability. Thus it allows the use of the

multiple I/O channels servicing several drivers allowing storage capacity over 64 MB.

INDICE

	<u>páginas</u>
I. CAPÍTULO I - Apresentação	1
I.1. O Projeto Carcarã	1
I.2. O Sistema de Gerenciamento e Manuseio de Entradas e Saídas em Discos Flexíveis	3
II. CAPÍTULO II - Acesso ao SGMES	8
II.1. Acesso ao SGMES	8
II.2. Descrição das Operações do Núcleo que permitem Comunicação de processos	9
III. CAPÍTULO III - Sistemas de Arquivos e Suportes de Controle	14
III.1. Sistemas de Arquivo	14
III.2. Alocação Física e Lógica de espaço aos arquivos	16
III.3. Diretórios	18
III.4. Proteção	22
III.5. Chamadas de Entradas e Saídas	26
III.6. Organização dos Arquivos e seus modos de acesso	29
IV. CAPÍTULO IV - Manipulação de E/S	36
IV.1. Geral	36
IV.2. Organização Geral dos Veículos de E/S	37
IV.2.1. Geral	37
IV.2.2. Descrição da Área do Sistema	40
IV.2.2.1. Diretório Index - DIX	42
IV.2.2.2. Diretório de Rótulos-DLB	47
IV.2.2.3. Diretório de Controle de Alocação de Espaço-DCA	53
IV.2.2.4. Diretório de Identificação	56
IV.2.2.5. Arquivos de Diretórios do Sistema	57
IV.2.3. Descrição de Área de Dados	59
IV.3. Pedidos de Entradas e Saídas	60

	<u>páginas</u>
IV.4. Funções Operativas do SGMES	62
IV.5. Descritores para manipulação de E/S - DEMES .	64
IV.5.1. Descritor de Ocupação de Blocos - DESOB	65
IV.5.2. Descritor de Volumes Habilitados - DEVOH	67
IV.5.3. Descritor de Arquivos em uso - DESAU.	68
IV.5.4. Descritor de Buffer Requisitados - DEBUR	70
IV.5.5. Outros descritores	74
V. CAPÍTULO V - Descrição das Funções do SGMES	76
V.1. Chamadas ou Funções Operativas do SGMES	76
V.2. Funções de Categoria Ø1	77
V.3. Funções de Categoria Ø2	80
V.4. Funções de Categoria Ø3	83
V.5. Funções de Categoria Ø4	91
V.6. Descrição dos Atributos para chamar as funções do SGMES	97
VI. CAPÍTULO VI - Conclusão	103
VI.1. Geral	103
VI.2. Comparação com outros sistemas de gerenciame <u>n</u> to de arquivos	105
VI.3. Sugestões para desenvolvimentos futuros	110
BIBLIOGRAFIA	112
APÊNDICE 1 - Condições de Erro do SGMES	118
APÊNDICE 2 - Descrição da Organização Especial de Ar- quivos	120
APÊNDICE 3 - Estrutura do BCES	130
APÊNDICE 4 - Interface de Disco	135
APÊNDICE 5 - Códigos de Operação	139

CAPÍTULO IINTRODUÇÃO1.1. O Projeto Carcarã

O "Projeto Carcarã", é um projeto do Programa de Engenharia de Sistemas da COPPE, e tem como objetivo desenvolver todo o software básico de um microcomputador, baseado em um microprocessador 6809 de 4MHz.

O referido projeto consta de um conjunto de trabalhos interdependentes, que consistirá no sistema operacional do microcomputador. Esses trabalhos são:

- A - Gerente Geral - que contém o núcleo do sistema operacional do qual fazem parte o módulo de comunicação entre processos, módulo de gerência de escalonamento de processos e módulo de gerência de memória.
- B - Gerente de E/S - isto é o Sistema de Gerência de Manipulações de E/S (SGMES); consiste exatamente de todo o módulo projetado e descrito neste trabalho e tem como objetivo dotar o sistema operacional de um poderoso agente de Entrada e Saída de informações e dados em veículos de acesso direto.
- C - Gerente de Terminal e Linguagem Operativa - este módulo consiste em um conjunto de instruções e algoritmos que permite o usuário do microcomputador comunicar-se com bastante facilidade com o sistema. É responsável

pela implementação de E/S por teclado e vídeo.

- D - Editor de Texto e Processador de Palavra - são basicamente dois utilitários intimamente ligados às funções do sistema operacional e objetivam dotar os usuários de um maior poder de preparação, verificação, atualização e depuração de seus arquivos.
- E - Protocolos de Comunicação - consistem em um conjunto de rotinas e processos inteligentes que possibilitam a implementação de uma rede local ou remota composta de minicomputadores do tipo em desenvolvimento ou a utilização destes como processadores locais/remotos de uma rede de teleprocessamento cujo o processador "HOST" seja diferente.

Vemos assim, que o SGMES, consiste em um subprojeto de um projeto bastante amplo no qual se tem buscado sempre boas soluções de compromisso.

É importante salientar que, a simplicidade e facilidade de uso foram sempre critérios prioritários, uma vez que a filosofia básica do desenvolvimento do referido sistema operacional é fazer chegar à "era da informática", através de um hardware simples porém eficiente, aqueles que necessitam de informática sem no entanto poderem investir demasiadamente em hardware, software e treinamento de pessoal.

Vale salientar também , que todo o software e hardware empregado neste projeto é de inteira concepção nacional e que apesar de a princípio ser um projeto acadêmico é em síntese altamente industrializável, visto o descrito e a dedicação de todos aqueles que nele estão ou estiveram envolvidos.

I.2. O Sistema de Gerenciamento e Manuseio de Entradas e Saídas em Discos Flexíveis

O Sistema de Gerenciamento e Manuseio de Entradas e Saídas em Discos Flexíveis, denominado por nós simbolicamente de SGMES, é o sistema responsável por todo o tratamento e manuseio de pedidos de entrada e saídas e gerência de recursos de arquivos armazenados em disco. Seu desenvolvimento foi feito basicamente para sistemas de computadores providos de subsistemas de memória de massa suportados em discos magnéticos, em especial em discos flexíveis.

Dentro de sua filosofia de desenvolvimento, adotamos características bastante singulares, no sentido de otimizar o tratamento dos arquivos, independente do veículo (meio físico onde estão localizados os arquivos) onde estão armazenados os dados ou informações, e de forma a capacitá-lo a gerenciar de modo eficaz todos os pedidos de E/S. O SGMES suporta o uso de diversos canais independentes de E/S, permitindo com isso o uso simultâneo de diversos arquivos, localizados em unidades periféricas diferentes ou não; tal característica é devido, em primeiro lugar ao uso do SGMES como parte de um sis-

tema operacional multiprogramado, em segundo a utilização de processos que concorrem paralelamente ao uso dos recursos disponíveis de E/S, e por último a existência no microcomputador de uma interface inteligente que trata os pedidos de maneira única, independente do tipo de disco a que ela esteja ligada.

O SGMES é virtualmente composto de três módulos, formando um conjunto único de processos e rotinas:

- Módulo Geral
- Módulo de Gerência
- Módulo de Manuseio

O Módulo Geral é aquele composto por rotinas e processos responsáveis pelo controle da comunicação entre o SGMES e os processos simples ou requisitantes (os definidos pelos usuários) e os processos especiais (aqueles definidos dentro dos módulos do sistema operacional, como por exemplo, na Linguagem de Comandos - LICO, etc).

O Módulo de Gerência é aquele que é responsável pela gerência (como diz seu próprio nome) dos recursos e arquivos disponíveis. Toda a proteção ao acesso de E/S é função deste módulo, que além disto tem a finalidade de controlar todo o sistema de diretórios, descritores e rótulos dos arquivos, bem como atualizá-los quando necessário.

O Módulo de Manuseio tem como função a realização de tarefas desde os níveis mais básicos até os de níveis intermediários, tais como:

- criar arquivos;

- habilitar arquivos;
- fechar arquivos;
- ler registros;
- gravar registros;
- blocagem e desblocagem de registros;
- etc.

Dentro deste módulo tudo é feito de modo a permitir que o usuário não se preocupe com a alocação física de arquivos, mas tão somente com a alocação lógica.

A comunicação entre processos em cada módulo e entre os processos dos módulos, é realizada através de estruturas tipo monitores, definidos especialmente para tais fins dentro do NÚCLEO* do sistema operacional. A obtenção dos recursos de E/S, por exemplo, uma unidade de disco, é sempre obtida através de estruturas como a citada acima.

Nas figuras 1, 2 e 3 apresentamos esquemas do SGMES, com o objetivo de dar uma idéia da sua estrutura geral, e um outro com a finalidade de mostrar a atuação do SGMES sobre os recursos de E/S disponíveis:

*NÚCLEO - módulo que implementa processos e provê todo um mecanismo de comunicação entre eles. |Maiores detalhes reporta-se a |⁶|, |²³|, |²⁵| e |²⁶||.

SGMES

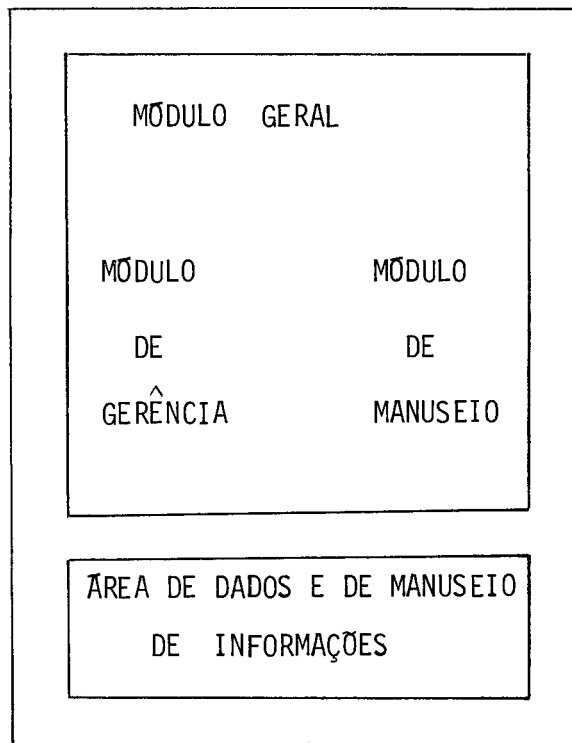
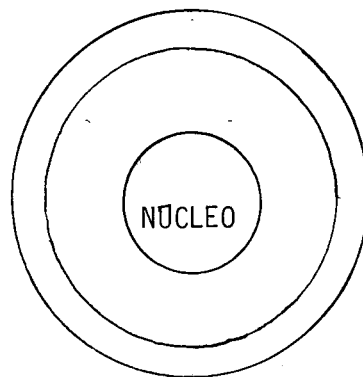


FIG.1 - ESQUEMA DO SGMES.



GERENTE DE ENTRADAS E SAÍDAS

GERENTE DE PROCESSOS E DE MEMÓRIA

FIG.2 - NÍVEL DO SGMES NO SISTEMA OPERACIONAL.

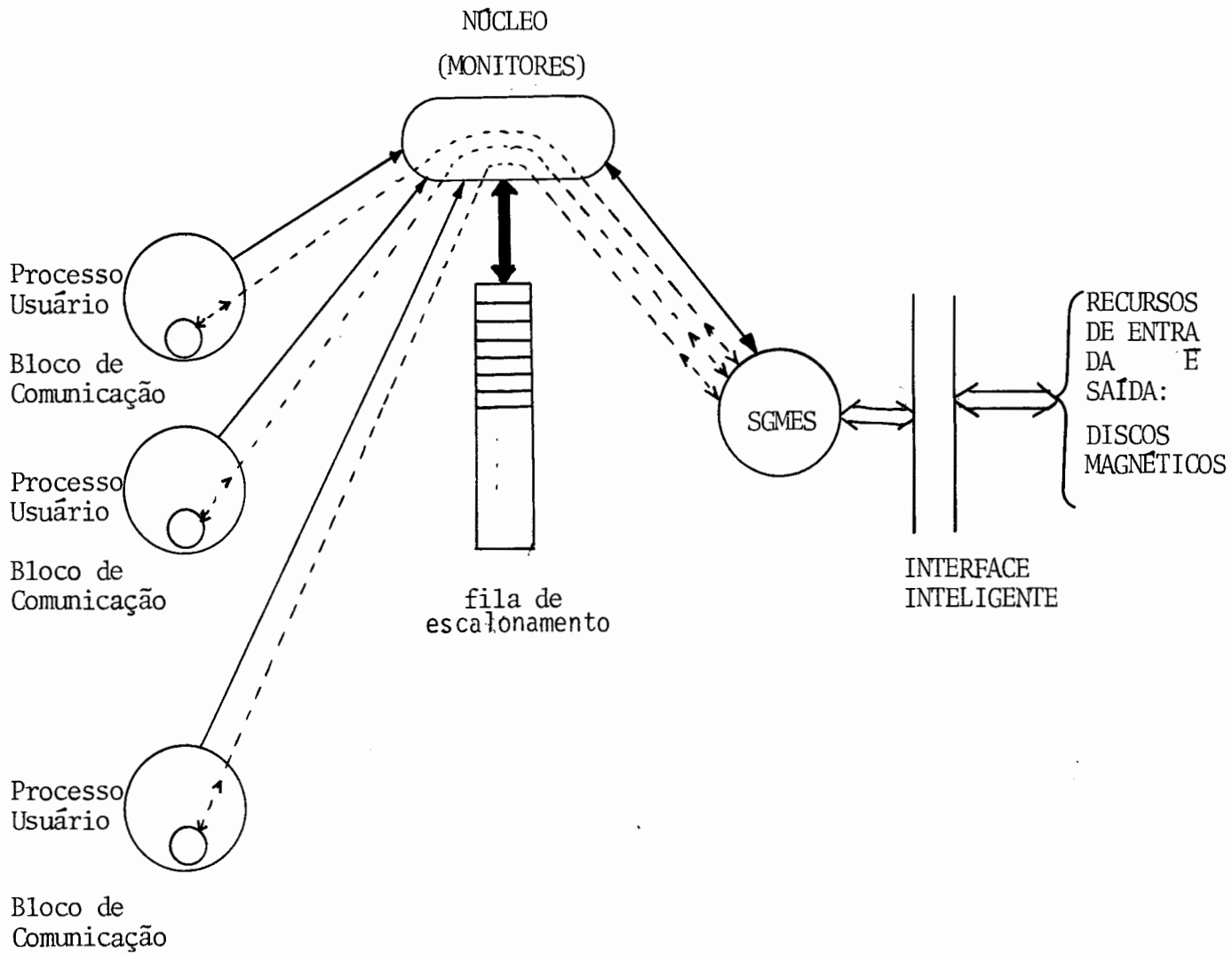


FIG. 3 - ATUAÇÃO DOS SEGMS E SUA LIGAÇÃO COM OS PROCESSOS NO AMBIENTE DO SISTEMA OPERACIONAL.

CAPÍTULO II
ACESSO AO SGMES

II.1. Acesso ao SGMES

O acesso ao SGMES, por processos simples ou não, é realizado sempre via uma estrutura tipo "monitor" |³|, |⁸| e |²| e através de uma estrutura complementar de envio de informações denominado de Bloco de Controle de Entradas e Saídas (BCES).

O BCES é um "buffer" definido pelo usuário, onde são formatadas as informações para um determinado pedido de E/S. Para cada arquivo usado o usuário deve definir esta estrutura chamada BCES, à qual ficam associados todos os pedidos de E/S desse determinado arquivo.

Desta forma, o acesso ao SGMES é realizado através de uma operação denominada DEPÓSITO, passando como parâmetro o endereço da BCES que contém as informações a serem analisadas e transformadas em execuções de tarefas pelo SGMES. A operação DEPÓSITO (rotina existente no NÚCLEO do sistema operacional) tem como parâmetros formais: a) o número do serviço solicitado, b) endereço da BCES; e tem a finalidade básica de enfileirar o pedido e esperar que o procedimento ou processo requisitado retire este pedido via uma operação RETIRADA, (também é uma rotina do NÚCLEO) a qual é utilizada somente pelos processos operadores (definidos como aqueles processos, dos diversos módulos do sistema operacional, responsáveis pela execução de serviços básicos, suportados pelo

sistema). Após o processo operador terminar o pedido solicitado, o mesmo se utiliza de outra operação, chamada FINALIZA, do Núcleo que tem como finalidade retirar o processo solicitante do estado de espera e retirar o seu pedido da fila.

É importante notar que o usuário antes de usar a operação DEPÓSITO deve preencher o BCES conforme o formato necessário para a execução do pedido a ser realizado via o Núcleo do Sistema Operacional.

II.2. Descrição das Operações do Núcleo que Permitem Comunicação de Processo com o SGMES

Antes de iniciar a descrição das operações do núcleo que permitem a solicitação de serviços e subsequente liberação dos mesmos, é interessante focalizar e descrever os diversos tipos de processos que compõem o sistema operacional do qual faz parte o SGMES.

O Núcleo, que é o núcleo do sistema operacional, consegue dotar o sistema de um ambiente de multiprogramação através do uso de filas implantadas via estruturas tipo "monitor".

Definimos como processo requisitante ou solicitante (aqui denotado apenas por PR) àquele processo que de-

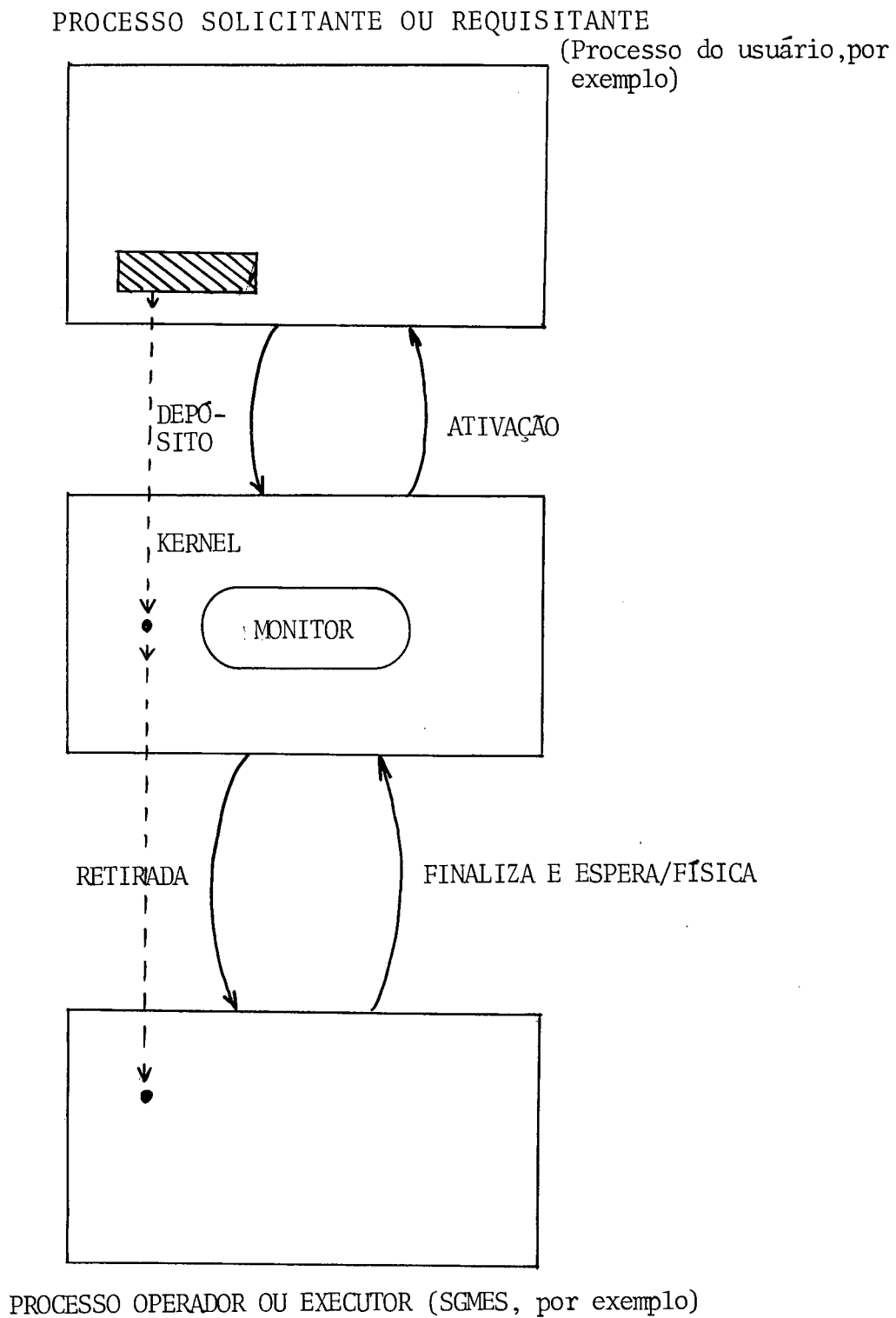


FIG.4. ESQUEMA DE COMUNICAÇÃO ENTRE PROCESSO E OPERAÇÕES DO NÚCLEO UTILIZADOS.

seja a execução de um serviço ou operação sobre determinados recursos aos quais não tem acesso direto. Definimos como processo operador ou executor (aqui denotado apenas por PE) aquele que é responsável pela execução dos serviços solicitados e que têm sob sua gerência os recursos sobre os quais serão feitas as operações.

A cada processo executor ou PE é associada uma fila do núcleo, dentro, no qual são enfileirados os pedidos dirigidos a tais processos executores.

Os processos tipo PR e PE, podem ser classificados em três tipos:

Processos Simples - ou do tipo 1, que são aqueles que não têm nenhum recurso computacional do sistema sob sua gerência direta, e para seu uso utiliza-se de processos PE.

Processos Especiais - ou do tipo 2, são aqueles que têm sob sua gerência recursos do sistema computacional e atendem requisições de processo PR ou do tipo 1, definido acima.

Processos Físicos - ou do tipo 3, são aqueles que lidam diretamente com o funcionamento físico de um recurso computacional (ver |¹⁹|).

Abaixo, descrevemos algumas das diversas operações que possibilitam comunicação entre processos via núcleo:

DEPÓSITO - operação que permite um PR solicitar ao núcleo a execução de um determinado serviço por um PE; os argumentos formais desta operação são: a) número do serviço a ser executado, b) endereço do BCES, c) booleana informando o estado de ESPERA ou não.

ESPERA - operação que permite um PR que solicitou serviços sem espera, possa mudar seu estado para espera.

RETIRADA - operação usada somente pelos PE e que torna disponível ao mesmo o primeiro pedido de serviços existente em sua respectiva fila.

FINALIZA - possibilita um PR sair do estado de espera, quando do término do seu serviço requisitado.

ESPERA-FÍSICA - usado por processos que esperam por eventos, para os quais não existem filas específicas de requisição. Quando um processo está em espera-física, ele é retirado da fila de prontos, sendo desativado.

ATIVACÃO - ativa processos desativados, colocando-os novamente na fila de prontos.

TRANSFIRA - esta rotina permite que um pedido de serviço passe de uma determinada fila para outra. Tem como parâmetros formais: a) número do serviço onde está o pedido, b) número do serviço para onde vai o pedido. É importante notar que esta operação somente age sobre o pedido que estiver no topo da fila original.

EXAMINE - determina o número de pedidos pendentes em determinada fila, de um determinado serviço.

CRIA-SERVIÇO - associa a um número de serviço livre um nome dado, criando uma fila de serviços para este número.

ABRA-SERVIÇO - verifica se, dado um nome, o mesmo existe na Tabela de Serviços, associado a um número, retornando o número.

ABRA-ARQUIVO - é o processo que torna acessível e/ou privatizável um determinado periférico ou arquivo físico (no caso de disco, habilita, ou melhor, associa à unidade física um nome lógico igual ao nome do volume que estiver contido na estação). Ele aloca na BCES dada o número do serviço e faz uma requisição à rotina de abertura específica do periférico pedido.

FECHA-ARQUIVO - desativa e/ou desaloca o arquivo ou um periférico indicado na BCES. É a operação inversa à descrita acima.

LEIA - Ler um elemento (caracter, registro, etc) de um arquivo previamente aberto, através de uma operação DEPÓSITO na fita especificada.

ESCREVA - análogo ao anterior, só que ao invés de ler, escreve.

CAPÍTULO III
SISTEMAS DE ARQUIVOS E
SUPORTES DE CONTROLE

III.1. Sistemas de Arquivos

O sistema de arquivos suportado pelo SGMES é baseado fundamentalmente em veículos de acesso direto tipo "disco flexível" (floppy disk ou diskette).

Os arquivos são tratados, todos, de forma análoga quanto à sua alocação física e lógica. Excessão deve ser feita, quanto à forma de acesso, apenas aos arquivos de gerenciamento dos recursos disponíveis, isto é, os diretórios do sistema, que apesar de serem arquivos comuns, têm características específicas.

Podemos classificar os arquivos, segundo sua finalidade, como:

- arquivos "diretório":

são arquivos especiais, manipulados apenas pelo SGMES, e cujo acesso é limitado a um pequeno número de processos especiais;

- arquivo de dados:

são arquivos comuns, cuja finalidade é armazenamento de dados e informações;

- arquivos simbólicos:

são arquivos comuns, destinados a conter programas fontes ou simbólicos, escritos em uma determinada linguagem, tal como o Pascal, Basic, Fortran, Coral, etc;

- arquivos de programas objeto:

são arquivos comuns, que contêm os códigos executáveis dos programas simbólicos;

Quanto à forma de acesso, os arquivos, neste trabalho, classificam-se em:

- arquivos sequenciais:

são aqueles em que o acesso por meio de escrita ou leitura é realizado de forma estritamente sequencial. Estes por sua vez classificam-se em:

- arquivos com registros de tamanho fixo;

- arquivos com registros de tamanho variável.

- arquivos de acesso direto relativo:

são aqueles em que a forma de acesso aos registros de tamanho fixo, pode se dar de forma semi-aleatória;

- arquivos de acesso especial:

são aqueles cujo tratamento e estrutura de acesso tem características especiais (para maiores detalhes ver o Apêndice 2).

Outras classificações poderiam ser feitas para melhor especificar a variedade de características dos arquivos,

mas para os nossos propósitos as classificações dadas acima são suficientes.

III.2. Alocação Física e Lógica de Espaço aos Arquivos

A alocação lógica dos arquivos aos quais o SGMES dá suporte realizada por um esquema de concessão de um bloco ou unidade monolítica de espaço (aqui denominada de Bloco Lógico de Alocação - BLA) ao arquivo à medida de sua necessidade de novos espaços.

Fisicamente, os arquivos estão localizados em veículos de acesso direto (discos magnéticos) cuja forma básica de organização, em geral é:

superfície
trilhas
setores

conforme pode ser visto na figura 5 abaixo:

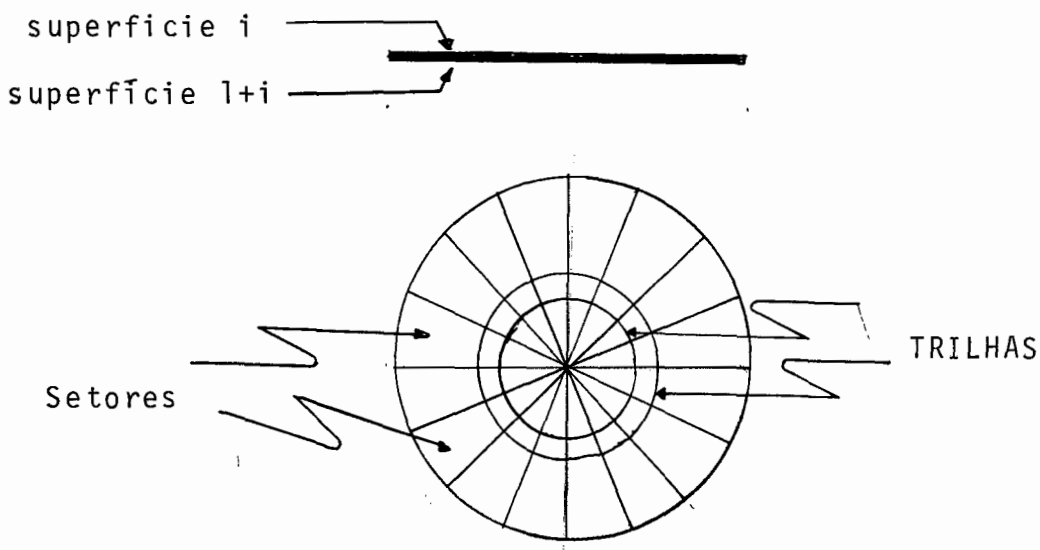


FIG.5 - ORGANIZAÇÃO FÍSICA DE UM DISCO.

O SGMES foi desenvolvido para dar suporte em princípio, a veículos de acesso direto denominados "discos flexíveis" que, em geral, independente de seu tamanho ou densidade, uma vez que isso é controlado pela interface do sistema. Isso leva também à possibilidade do SGMES, com poucas modificações sobre os tamanhos dos registros de controle, gerenciar discos dos tipos "panela" ou "cartucho" com mais de 10 MB de capacidade.

Uma trilha, dependendo da densidade de gravação permitida e dos mecanismos de sincronização e velocidade, pode ter uma quantidade de setores variável, sendo mais comum, nos discos flexíveis, ter 13, 16 ou 26 setores de 128 ou 256 bytes cada setor. O SGMES suportará, em sua versão original, trilhas com 256 bytes somente, independente da formatação a ser adotada pelo usuário. Tal tamanho é proveniente da construção da interface.

Um Bloco Lógico de Alocação - BLA (definido mais acima) é a unidade básica de alocação de espaço, e consiste basicamente de um conjunto de 4 (quatro) setores consecutivos, podendo ser esse número modificado pelo usuário, quando da inicialização do sistema.

A alocação física dar-se-á em qualquer parte do veículo onde haja BLA's disponíveis, não sendo necessário o usuário preocupar-se com o conhecimento local, uma vez que este controle lhe é totalmente transparente.

Na alocação lógica, é praticamente ilimitada a alocação de BLA's, desde que haja disponibilidade. A segmentação é permitida em grupos de BLA's, contudo sendo restrito a apenas 5 (cinco) grupos, isto é, só é possível ter até cinco

grupos (com BLA's contíguas) de segmentação alocados a um arquivo de qualquer espécie. Uma tentativa de requisição de espaço para mais de cinco blocos representa para o usuário um tipo errôneo de alocação, sendo-lhe informado que o disco está cheio.

Na Figura 6 mostramos um esquema da utilização e controle de espaço alocado a três arquivos hipotéticos em um veículo de E/S, também hipotético, utilizando os conceitos de trilhas, setores agrupados em BLA's e com a estrutura proposta pelo SGMES.

III.3. Diretórios

Os diretórios são entidades de informação que fazem a interface entre requisições de arquivos pelos processos de usuário e os arquivos propriamente ditos, e induzem sobre eles uma estrutura complexa, como um todo.

Cada usuário tem no diretório um subconjunto de informações referentes aos seus arquivos, tanto de forma local como global.

Um diretório comporta-se, exatamente como um arquivo comum, exceto que não pode ser criado, escrito ou lido diretamente pelo usuário, mas tão somente por processos privilegiados; seu controle é totalmente efetuado pelo SGMES. Apesar desta restrição, qualquer processo (privilegiado ou não) pode ter acesso (leitura somente) a seu diretório como se fosse um arquivo comum, via comandos específicos do SGMES

ESQUEMA REPRESENTANDO TODO O ESPAÇO UTILIZÁVEL DE UM VEÍCULO HIPOTÉTICO DE E/S, COMPARTILHANDO TRÊS ARQUIVOS PARA USO EM AMBIENTE DE MULTIPROGRAMAÇÃO

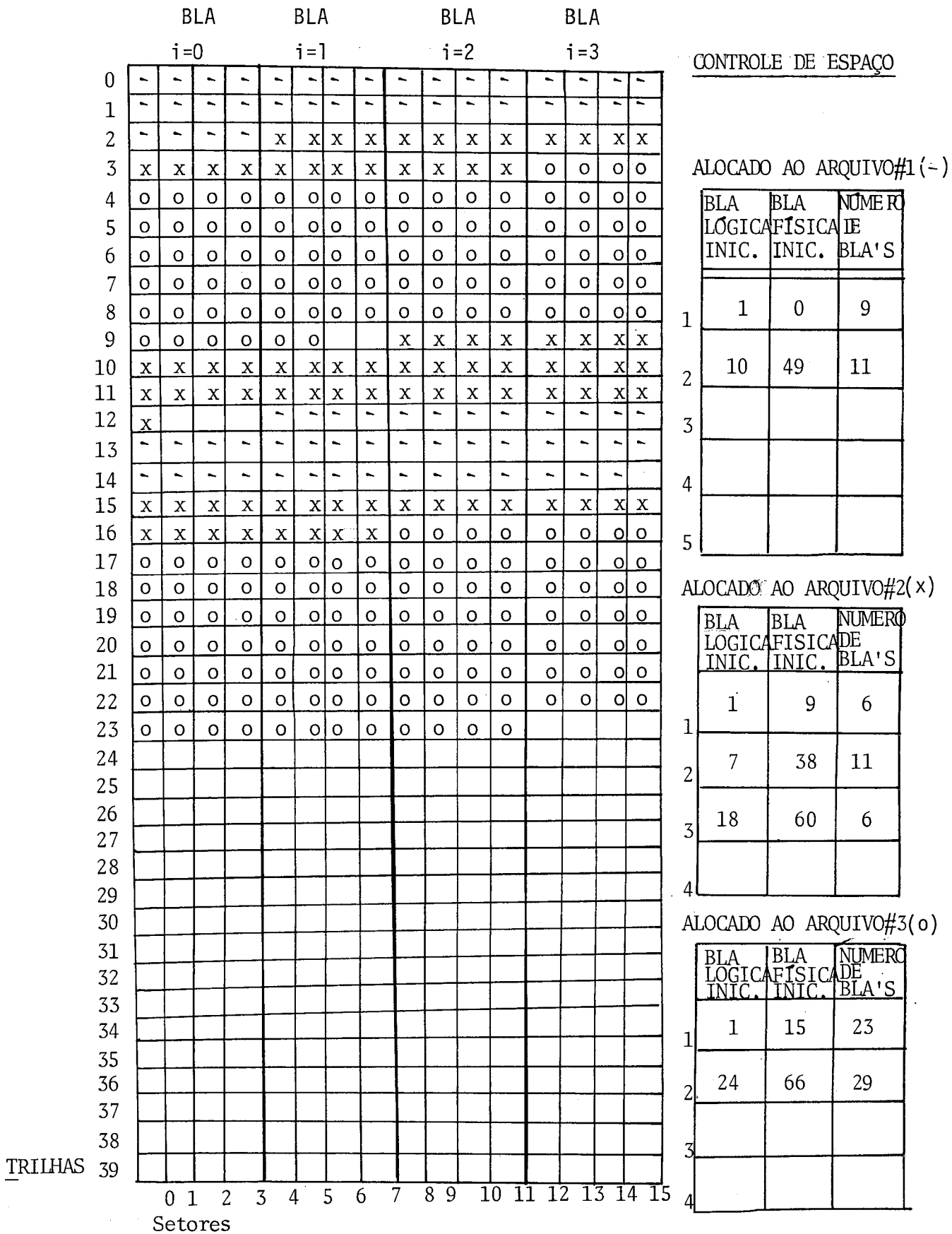


FIG.6. ESQUEMA DE ALOCAÇÃO DE ESPAÇO EM DISCO

no entanto a restrição continua válida quando se trata de áreas do diretório não contendo informações referentes a um determinado usuário.

O SGMES, mantém diversos diretórios para seu uso próprio, como por exemplo o diretório index, onde são catalogadas todas as raízes das informações pertinentes aos diversos usuários dos veículos de acesso direto.

Todos os arquivos de um usuário podem ser encontrados percorrendo-se uma cadeia de informações pertinentes a ele, partindo-se do diretório index, que é o ponto de partida de todo e qualquer acesso aos diretórios.

No diretório index o apontador é um número de 2 dígitos (ocupa 1 byte) que informa a localização do rótulo inicial dos arquivos de determinado usuário dentro do diretório de rótulos (ou diretórios de labels) - DLB: assim por exemplo,

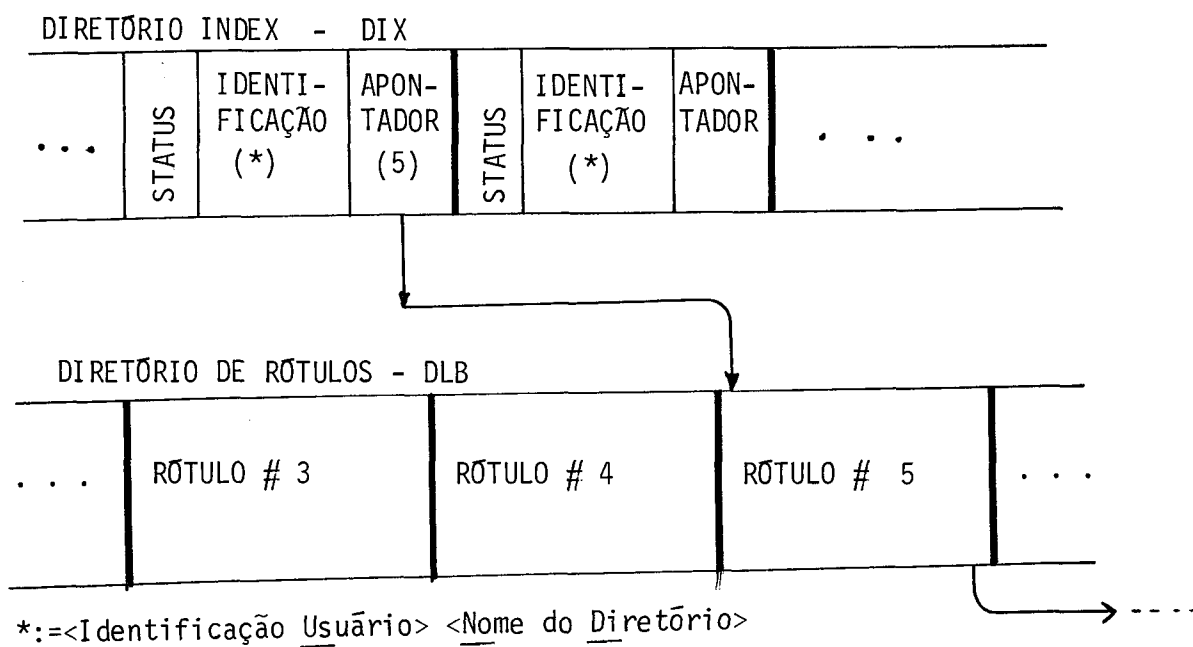


FIG.7. ESQUEMA DE LIGAÇÃO DIX COM DLB.

se o número contido no apontador for 5 (cinco), indica que é o quinto rótulo existente no DLB (ver figura 7), ou seja, o rótulo inicial do usuário está localizado fisicamente em:

endereço setor = base + $\text{int}(n/3, 1)$

endereço byte

dentro do setor = $15 + 80 \times \text{int}(n/B)$

base = número do setor dentro de uma determinada trilha onde será criado o DLB; este valor é determinado dinamicamente, quando da inicialização de cada disco (unidade flexível).

n = número corrente contido no apontador do DIX

B = endereço do setor - base - 1

$\text{int}(\cdot)$ = valor inteiro do valor numérico da expressão entre parênteses, truncando a parte decimal.

Outros diretórios suportados pelo SGMES são:

- DLB diretório de rótulos (LABEL'S)
- DCA diretório de controle de alocação de espaço em disco
- DID diretório de identificação de volume
- outros.

Outros detalhes podem ser vistos no capítulo IV.

III.4. Proteção

Embora o esquema de controle de acesso, projetado para o sistema, seja bastante simples, ele inclui diversas características bem interessantes.

Cada usuário dentro do sistema é designado ou reconhecido por um número que denominamos de "identificação do usuário", o qual tem um papel importante dentro do esquema de acesso a arquivos e sua proteção.

Quando um arquivo é criado, a ele é associada automaticamente a identificação do usuário sob qual está sendo criado. Também é associado ao arquivo, quando de sua criação, um registro (que faz parte do rótulo do arquivo) que especifica diversas características do arquivo que servem como padrões para proteção. Esse registro tem 8 bits (1 byte) assim utilizados:

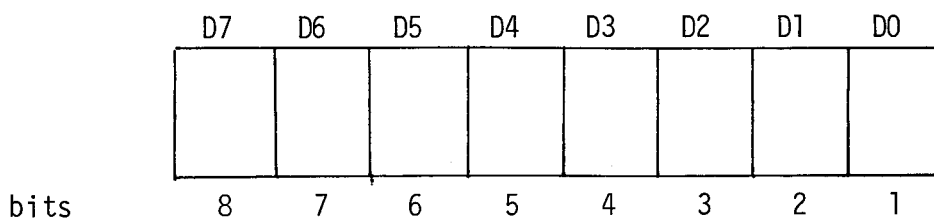


FIG.8. REGISTRO DE CARACTERÍSTICAS DO ARQUIVO.

bit 1 = ativo(1) - arquivo público, isto é, qualquer usuário pode acessá-lo.

restaurado(0) - arquivo privado: acesso permitido somente ao usuário criador.

- bit 2 = ativo(1) - restrito o acesso apenas ao usuário criador
e ao co-utilizador;
restaurado(0) - não restritivo;
- bit 3 = ativo(1) - só leitura é permitida;
restaurado(0) - leitura não é permitida;
- bit 4 = ativo(1) - só escrita é permitida;
restaurado(0) - escrita não é permitida;
- bit 5 = uso futuro;
- bit 6 = ativo(1) - arquivo temporário;
restaurado(0) - arquivo permanente;
- bit 7 = ativo(1) - alocação de novas BLA's, desde o início, criação do arquivo é feita dinamicamente;
restaurado(0) - alocação inicial de BLA's é solicitada e quantum de alocação de BLA's é especificado;
- bit 8 = ativo(1) - arquivo continuação de algum outro localizado ou iniciado em outro volume;
restaurado(0) - arquivo totalmente contido no volume corrente;

Outro registro, alocado em conjunto com o acima especificado, é feito para complementar a proteção em um primeiro nível (ver detalhe na descrição do diretório de rótulos - DLB).

O segundo nível de proteção está diretamente ligado à filosofia de alocação de espaço para os arquivos. O espaço é

alocado segundo um método de alocação mapeada, cujo mapa está localizado no diretório geral do sistema, ou mais precisamente no diretório de rótulos (DLB), cujas informações pertinentes às alocações realizadas ou feitas a um arquivo são registradas, a nível de gerência geral, também no diretório de controle de alocação de espaço (DCA).

No DLB é mantida, para cada arquivo existente, uma tabela com um máximo de cinco entradas, com a seguinte configuração (figura 9):

	1	2	3
	NÚMERO LÓGICO DO BLA INICIAL	NÚMERO FÍSICO DO BLA INICIAL	NÚMERO DE BLA'S ALOCADOS A
1			
2			
3			
4			
5			

BLOCOS DE
SEGMENTAÇÃO

FIG.9. TABELA DE SEGMENTAÇÃO DE ARQUIVO.

onde na coluna 1 é registrado o número lógico do BLA inicial de cada bloco de segmentação (representado por cada entrada da tabela) na coluna 2 é registrado o número físico da BLA correspondente àquela registrada na coluna 1 ; na coluna 3 é registrado o número de BLA's contíguas alocadas ao arquivo, a partir daquela indicada na coluna 1 , isto é, o número de BLA's do referido bloco de segmentação. Tal como aparece também na figura 6.

A tabela que mantém todas as informações acima descritas é denominada de Tabela de Alocação de Segmentos - TAS.

Conforme já definido, um BLA constitui-se, em geral, de quatro setores contíguos, e é alocado a um arquivo à medida de suas necessidades de expansão, e são desalocados à medida de sua liberação ou não uso pelo arquivo. Os BLA's são numerados (para efeitos físicos) de 1 a N, sendo N o número máximo de setores existente em um disco dividido por 4:

$$N = S.T.s/4$$

onde S é o número de superfícies existente no veículo;

T é o número de trilhas existentes por superfície;

s é o número de setores por trilha;

No DCA localizam-se os registros que fazem o controle dos BLA's em uso, disponíveis, e daqueles contidos em possíveis trilhas defeituosas.

O controle dos BLA's disponíveis e em uso é realizado por registros (um para cada espécie) de N bits, onde cada um desses bits representa um BLA, que através de seu estado, ativado ou restaurado, informa se o correspondente BLA está em uso ou disponível, respectivamente, como é visto na figura 10.

Para otimizar a pesquisa nesse registro de bits, existem também outros dois registros (além de outros) R1 e R2 que informam respectivamente o início da área do sistema (área onde ficam os diretórios) e a posição do primeiro BLA disponível.

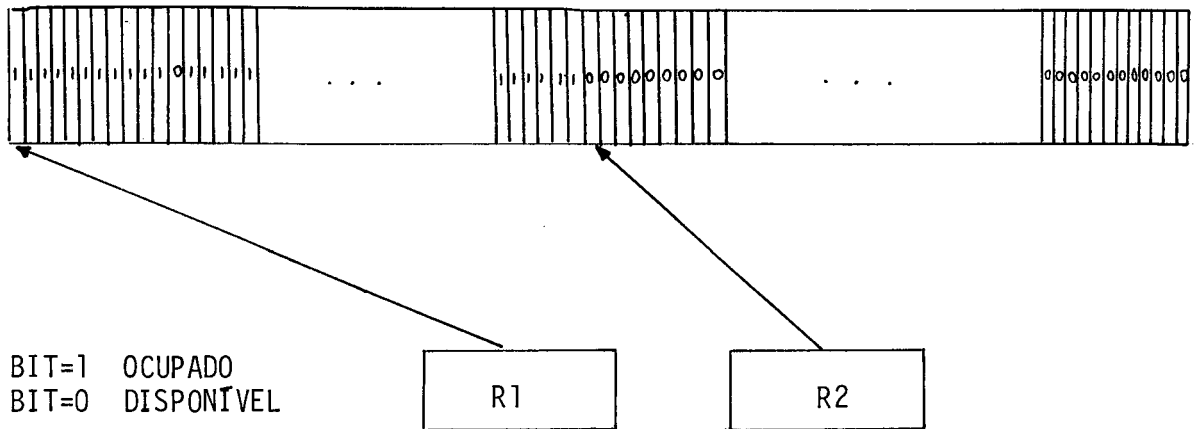


FIG.10. BLOCO DE CONTROLE DE BLA's.

Os blocos lógicos de alocação, BLA, no seu padrão, estão distribuídos pelas trilhas e setores consecutivamente a partir da trilha zero setor um, de quantum em quantum, sendo também, numerados sequencialmente de 1 a n, onde n é o número máximo de setores existentes no veículo dividido pelo quantum.

Da mesma forma que os BLA devido ao fato de que as características da interface de disco são implementadas em software, os setores serão virtualmente numerados em ordem cardinal de 1 a n, onde n é o número máximo de setores existentes no veículo de acesso direto (maiores detalhes, ver o apêndice 4).

III.5. Chamadas de Entradas e Saídas

As chamadas de entrada saída - (E/S), prevêm uma facilidade no sentido de eliminar a diferença entre os vários periféricos e os modos de acesso.

As chamadas, também, aproveitando uma característica dos microprocessadores permitem ao processo solicitante especificar se a mesma (entrada ou saída) é com ou sem espera, isto é, se o PR será suspenso até a E/S ser completa da ou se o mesmo continua executando enquanto a E/S está se processando.

A cada arquivo aberto é associado um descritor no qual são alocados três números que devem ser usados nas suas chamadas subsequentes, para qualquer tipo de manipulação sobre o mesmo.

Exceto quando indicado em contrário, uma leitura e uma gravação serão sempre sequenciais. Isto significa que se um determinado registro no arquivo foi o último lido ou escrito, a próxima chamada de E/S para esse mesmo arquivo referir-se-á ao próximo registro que segue aquele por último lido ou escrito.

Para cada arquivo aberto, o sistema manterá no descritor de "arquivos abertos" (o qual se localizará na memória principal, em parte reservada ao SGMES) uma estrutura de dados que indicará o próximo registro a ser lido ou escrito.

A detecção de fim de arquivo, é feita quando o endereço do registro a ser lido ou escrito for diferente de um endereço de setor alocado ao arquivo ou quando houver um comando do processo específico para tal procedimento.

O espaço alocado para um arquivo é representado por:

$$F = Z_j = \{(T,S,M)_j | 1 \leq j \leq N\}$$

onde Z_j é um BLA que é representado em uma notação algébrica, de uma forma genérica, como uma terna $(T, S, M)_j$ que representa uma partição do disco, e onde T indica a trilha iniciando, S o setor início, e M o número de setores contíguos, a contar de S . Assim, por exemplo, se ao arquivo tiverem sido alocados, os BLA Z_1 , Z_6 , Z_7 e Z_8 temos que:

$$\begin{aligned} F &= \{Z_1, Z_6, Z_7, Z_8\} = \\ &= \{(0,5,4), (0,25,4), (1,3,4), (1,7,4)\} \end{aligned}$$

Um endereço é em geral, na forma de uma terna (Su, T, Se) representando a superfície requerida por Su , a trilha por T e o setor por Se (a terna pode ser vazia, quando então será interpretada como sendo o próximo setor contíguo ao último lido ou escrito).

No caso de haver uma detecção de fim de área de forma não desejada por exemplo, o SGMES retorna a informação ao processo requisitante de E/S via BCES, devendo o "PR" tomar uma das alternativas abaixo:

- a - pedir alocação de novo BLA, e prosseguir normalmente a execução do serviço (quando a locação não for automática);
- b - pedir fechamento do arquivo; ou
- c - cancelar a tarefa via o núcleo do Sistema Operacional;

Uma chamada de E/S necessita sempre informar o arquivo desejado (via número correspondente alocado no BCES pelo SGMES).

As trilhas defeituosas (aquelas em que não é possível nenhuma operação) terão seus bits de ocupação ativados na

cadeia de disponibilidade (isto é, não as tornando disponíveis) e descrevendo-as no diretório geral do volume.

A alocação de novos BLA's aos arquivos pode ser também feita de forma automática, caso em que o bit de alocação automática do registro de proteção está ativado.

Essa alocação é feita em quantidades de n BLA's (default $n = 1$), devidamente especificados pelo usuário quando da criação ou habilitação de um arquivo.

III.6. Organização dos Arquivos e seus Modos de Acesso

O SGMES dará suporte a três tipos básicos de organização de arquivos, a saber:

- § organização sequencial;
- § organização direta relativa;
- § organização especial,

A organização especial não será descrita aqui (Maiores detalhes ver no Apêndice 2).

Nas organizações sequencial e relativa os registros podem tanto ser armazenados contíguamente, como de forma semi-aleatória, dado seu endereço pertencente à área alocada ao arquivo, independente de suas chaves de recuperação.

As características das organizações sequencial e relativa são bastante interessantes do ponto de vista do fator de utilização do veículo de armazenamento de dados.

Para a descrição detalhada destas características

vamos considerar os registros dos arquivos sendo de tamanho fixo ou de tamanho variável.

Registros de tamanho fixo são aqueles que têm seu tamanho predefinido pelo usuário, não podendo ser modificado durante a sua vida útil a não ser através de cópia para outro arquivo com registros de tamanho fixo porém de comprimento diferente do anterior.

Registros de tamanho variável, podem ser subdivididos em dois tipos:

- a - de tamanho limitado, isto é, aqueles cujo comprimento dos dados ou informações a serem armazenadas em um registro varia de um para outro, mas têm um limite físico pré-definido. Estes têm tratamento similar aos registros de tamanho fixo;
- b - de tamanho ilimitado, isto é, aqueles que não têm limites na sua forma ou comprimento.

Consideremos também, os seguintes modos de acesso:

- a - acesso sequencial -

Modo no qual o usuário não necessita especificar o registro a ser lido ou escrito, pois por definição o registro a ter acesso será sempre o próximo.

As operações básicas permitidas a esse tipo de acesso, são:

- ler próximo registro;
- gravar o próximo registro;
- retornar n registros (&); (permitindo só a arquivos contendo registros de tamanho fixo);

- avançar n registros (&);
- retornar ao início do arquivo;

além, evidentemente, das operações normais de abertura, habilitação, fechamento, etc sobre o arquivo. As operações assinaladas por (&) são na realidade pseudo-operações, pois o que é feito é um acréscimo ou decréscimo no contador de registros. Em ambos os casos, se $n \pm$ o conteúdo do contador de registros sair fora do espaço reservado ao arquivo, haverá uma condição de erro, e nestes casos o contador é zerado (retorna a 0).

b - acesso direto relativo -

Modo de acesso no qual o usuário necessita especificar o número lógico do seu registro ao qual quer ter acesso. As operações de E/S possíveis são:

- ler registro M;
- gravar registro M;

onde M é o número lógico dos registros dentro do arquivo (número cardinal), o qual varia de 1 até o número máximo de registros possíveis de armazenar em todos os BLA's alocados ao arquivo.

Naturalmente, este modo de acesso só é possível de ser realizado quando o tamanho dos registros é fixo ou variável limitado.

A solicitação de acesso a um registro fora do espaço alocado ao arquivo, gera uma condição de erro irrecuperável.

A seguir vamos procurar analisar as características da organização sequencial.

i - Blocação e Desblocação de registros transparente aos usuários.

Esta é uma das mais fortes características desta forma de organização de arquivos. A blocação de registros consiste em agrupar registros de tamanho fixo ou variável limitado em um único bloco. As condições para existência da blocação são:

- registros de tamanho pré-fixado menores que 256 byte;
- registros de tamanho variável limitado.

No primeiro caso acima, a blocação é feita agrupando tantos registros quanto for possível conter em um único setor de 256 bytes. Se por exemplo, o registro tiver comprimento fixo igual a 60 bytes, então a blocação consiste em agrupar quatro registros ($4 \times 60 = 240$ bytes) em um único setor. Tal característica é registrada nos descritores do arquivo.

A desblocação é o processo inverso, isto é, consiste em fazer o usuário ter acesso a, no exemplo acima, 4 registros (entregues pelo SGMES ao usuário um de cada vez à medida que for solicitado) lógicos com apenas um acesso físico ao disco. Esta característica permite uma maior taxa de utilização do veículo além de permitir uma otimização no tempo de acesso, uma vez que com um único acesso físico, são realizados diversos acessos lógicos.

ii - Possibilidade de acesso a caractere.

Esta característica possibilita a leitura e escrita a nível de caractere, sequencial e contiguamente. Ela permite ao usuário, tanto na escrita quando na leitura de um determinado arquivo, ter acesso caractere a caractere, independente da forma de organização do arquivo e do tamanho do registro.

Neste caso, quando da escrita por exemplo, a gravação física só é realizada, o que é totalmente transparente ao usuário, quando o buffer do SGMES associado ao arquivo estiver totalmente cheio. Quando da leitura, as leituras físicas são comandadas pelo SGMES e também totalmente transparentes ao usuário.

A entrada/saída a nível de caractere é uma entrada/saída normal, onde é solicitada a escrita ou leitura (via comandos especiais) de 1 ou mais caracteres de um mesmo registro.

iii - Utilização de arquivos com registros de tamanho variável

Os arquivos com registros de tamanho variável têm sua organização de uma forma um pouco diferente daquela padrão dos arquivos sequenciais com registros de tamanho fixo. No caso que se analisa aqui, aos registros lógicos são acrescentadas etiquetas de 4 bytes que são gravadas em cada registro lógico, tantas vezes quantas forem os setores ocupados, e que tem a seguinte configuração:

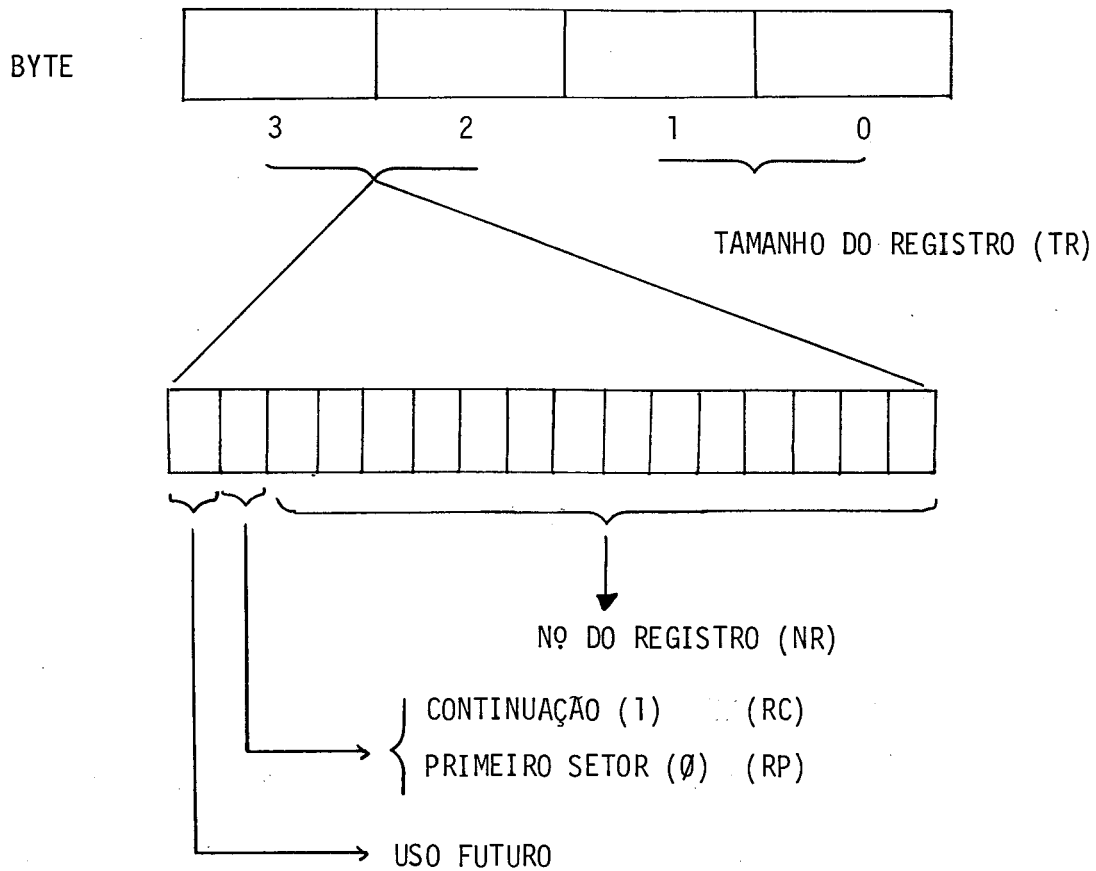


FIG.11. ETIQUETA DE REGISTRO DE TAMANHO VARIÁVEL.

Esta etiqueta é gravada sempre nos 4 primeiros bytes de cada setor continuação utilizado por um mesmo registro e no começo do registro lógico. Nos registros que ocuparem mais de 1 setor, nos setores seguintes ao primeiro, o campo tamanho do registro aponta para o próximo, continuação deste registro (isto é, naqueles em que o bit de continuação estiver ativado).

Nos arquivos com registros de tamanho variável aqueles registros que tiverem tamanho maior que 1 setor, deverão ser, sempre que possível, gravados contiguamente, porém não obrigatoriamente.

Aqui, uma dificuldade que deve ser contornada, é aquela em que nunca sabemos a priori o tamanho de cada regis-

tro, e cujo valor deve fazer parte do mesmo. Tal dificuldade, será por nós contornada da seguinte maneira:

- a - quando o registro for menor que 256 bytes, incluindo os quatro bytes da etiqueta, antes de ser efetivamente gravado, o SGMES alocará automaticamente o valor do tamanho do registro no campo TR da etiqueta.

- b - quando o registro for maior que 252 bytes (256 = 252 bytes lógicos usados pelo usuário + 4 bytes da etiqueta), os primeiros 256 bytes (alocados em um determinado buffer) somente serão gravados quando o SGMES receber a última parte do registro. No entanto, para os 252 bytes subsequentes aos primeiros 252 bytes, serão gravados normalmente, sendo alocado no TR o número relativo do setor no qual será gravado o próximo registro.

CAPÍTULO IV

MANIPULAÇÃO DE ENTRADAS/SAÍDAS EM

VEÍCULOS DE ACESSO DIRETO

IV. Manipulação de Entradas e Saídas

IV.1. Geral

A manipulação de entradas e saídas para veículos de acesso direto (em geral, discos magnéticos), será realizado por um conjunto de processos denominados processos manipuladores, entre os quais estão previstos aqueles que fazem as seguintes operações e tarefas:

- criação de arquivos;
- abertura de arquivos;
- gravação de registros;
- leitura de registros;
- fechamento de arquivos;
- blocagem/desblocagem de registros;
- etc.

O controle e atualização de diretórios e a geração de informações sobre as E/S, úteis aos sistemas de gerenciamento, serão realizados por um conjunto de processos e rotinas, compondo aquilo que chamamos de "Gerente", o qual poderá realizar, dentre outras, as seguintes tarefas e operações:

- manutenção e atualização de tabelas e descritores de diretórios e arquivos;

- remoção de arquivos;
- atualização dos diretórios do sistema;
- geração de mensagens de advertências e erros;
- etc.

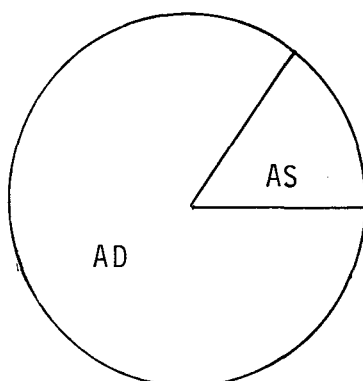
IV.2. Organização Geral dos Veículos de E/S

IV.2.1. Geral

A macro-organização dos veículos de entrada e saída é realizada virtualmente por duas áreas ou estruturas distintas, AS e AD, onde:

AS - área do sistema onde estarão localizados todos os diretórios do sistema e as áreas reservadas para o armazenamento de informações sobre o uso e desempenho do sistema de entrada e saída;

AD - área de dados onde estarão localizados todos os programas, arquivos e diretórios do usuário, inclusive o software básico do sistema, tais como rotinas utilitárias, compiladores, etc.



A área do sistema, AS, é organizada em arquivos de nominados "arquivos diretórios do sistema", os quais têm a finalidade de registrar os acontecimentos e controlar o uso e de desempenho dos veículos de entrada e saída. Dentre os diversos "diretórios que compõem os diversos arquivos de diretórios do sistema, podemos citar os seguintes:

= Diretório-Index	DIX
= Diretório-de-Rótulos	DLB
= Diretório-de-Controle-de Alocação-de-Espaço	DCA
= Diretório-Identificação	DID

Cada um dos diretórios do sistema tem como objetivo o controle de determinadas características de seu uso. O conjunto desses diretórios forma um complexo sistema de informações, capaz de permitir a gerencia de todas as formas de usos dos veículos de entrada/saída.

A pseudo-área de dados AD, com relação a estrutura e formas de acesso a arquivos, tem organização de certa forma livre, contendo registros de tamanho fixo ou variável. Uma

limitação de uso dessa área, no nosso sistema, é que ela pode conter no máximo 48 diretórios/arquivos de usuário. A alocação de espaço nessa área é realizada através da unidade de alocação conforme já visto: Bloco Lógico de Alocação - BLA, o qual compreende, em geral, um bloco de 4 setores contíguos.

Para o SGMES, os diversos veículos de acesso direto estão estruturados sempre em setores, não importando que na realidade eles estejam organizados fisicamente em superfícies, trilhas e setores. Dessa forma, se cada veículo tiver C superfícies, T trilhas por superfície, cada qual contendo S setores, o número total de setores será $C \times T \times S$. Se a capacidade de cada setor for K bytes, então a capacidade total de cada veículos será $C \times T \times S \times K$ bytes.

As informações referentes à estrutura do veículo de acesso direto em uso corrente estarão armazenadas em um diretório.

Existem muitos tipos de veículos de acesso direto, entre os quais podemos citar os discos cartuchos, discos painelas, discos flexíveis, etc. A estrutura do SGMES apesar de inicialmente voltada para os discos flexíveis, pode ser, no entanto, facilmente adaptada para discos tipo panela, fixo ou não. Existem no mercado de computadores três tipos básicos de discos flexíveis, os de 5.1/4, 8 e 10 polegadas de diâmetro, os quais dependendo de sua formatação (ou via hardware ou software/hardware) têm de 36 a 77 setores, cada qual com 13 a 26 setores de 128 a 1024 bytes, mas tudo isso nos é transparente, uma vez que a manipulação física direta dos equipamentos é feito pela interface de disco, que é inteligente (ver apêndice 4).

Uma observação deve ser feita aqui quanto ao tratamento tanto da AS como da AD; ambas têm tratamentos análogos tanto com relação a estrutura de dados quanto aos procedimentos; também são análogas às formas de manipulação e de acesso exceto em relação à área do sistema que tem acesso prioritário e reservado àqueles processos autorizados pelo sistema operacional.

IV.2.2. Descrição da Área do Sistema

Como definido no item anterior, a área do sistema é a área do veículo de E/S e pode se localizar em qualquer tripla do veículo. Nela são armazenadas todas as informações necessárias para a manipulação e gerenciamento de entrada e saídas.

As informações contidas na AS estão hierarquicamente estruturadas, em conjuntos denominados de "diretórios do sistema".

Os diretórios são arquivos especiais, aos quais apenas o sistema operacional e os processos autorizados pelo SGMES tem acesso.

No entanto, esses diretórios serão tratados, como já dito, de forma semelhante aos arquivos de dados dos usuários.

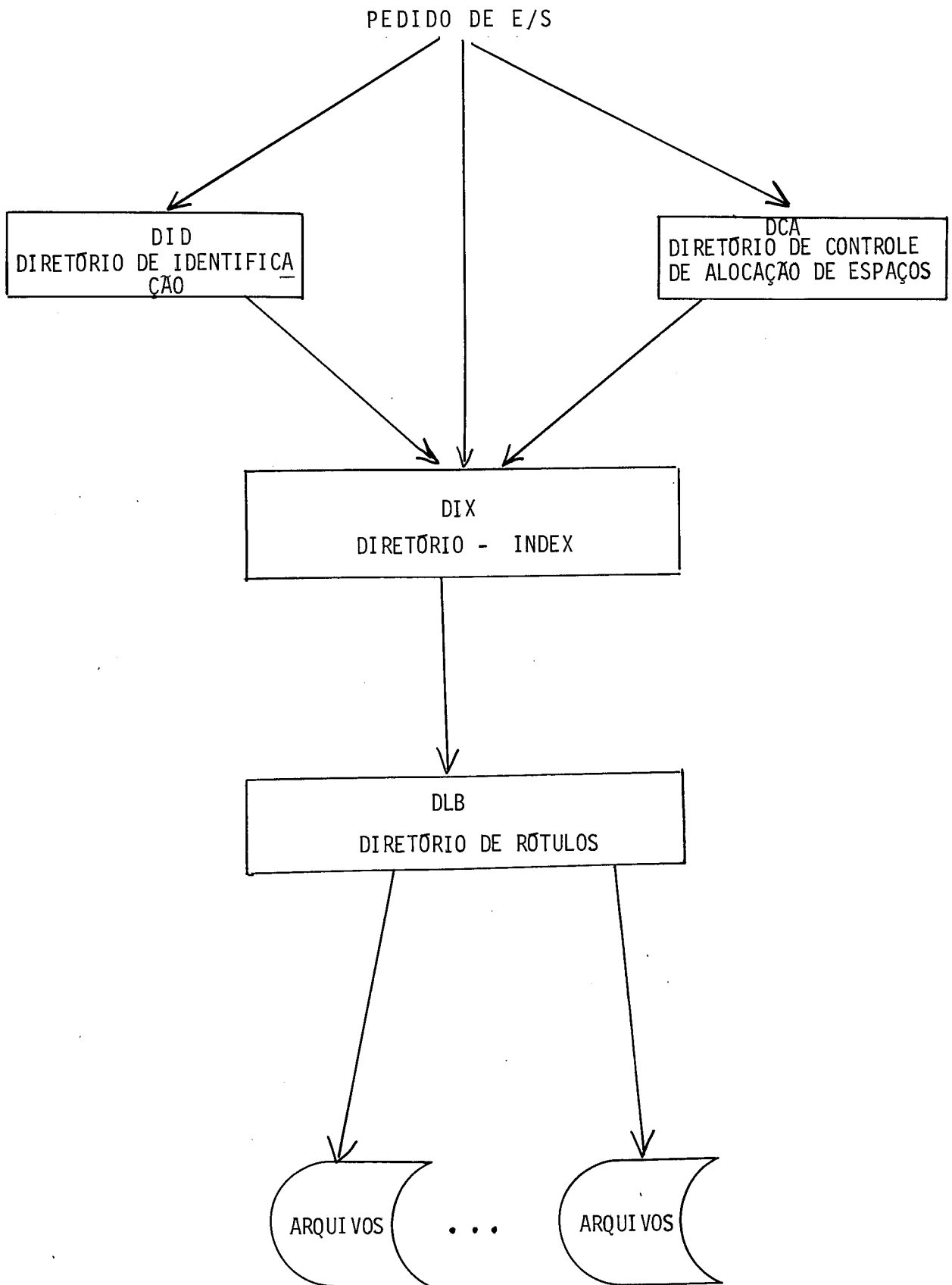


FIG.13. ESTRUTURAS DOS DIRETÓRIOS DO SISTEMA.

IV.2.2.1. Diretório-Index (DIX)

É um diretório do sistema cujo conteúdo são os índices de todos os diretórios existentes no sistema de entrada e saída, inclusive daqueles diretórios pertencentes ao sistema de controle de arquivos e controle dos próprios veículos. Assim, como o diretório-index é na realidade um subdiretório do Diretório Geral do SGMES nele contém o seu próprio índice.

CONTEÚDO DE CADA SETOR DO DIX:

- Número do subdiretório	}	1 byte	}	16 bytes
- Tipo Index(1 = sistema; 0 = usuário)				
- Status	10 bytes			
- Uso Futuro	5 bytes			
- Diretório			}	15
- Identificação Usuário	2 bytes			
- Nome diretório do Usuário	6 bytes			
- Apontador para o DLB	2 bytes			
- Registro de Segurança de Acesso e Temporalidade	2 bytes			
- Uso futuro	2 bytes			
- Apontador p/próximo Diretório Usuário no DIX	1 byte			
TOTAL				$16 + (16 \times 15) \times 3$

O campo Diretório, cujo tamanho é 15 bytes, é criado a cada novo diretório (isto é, de um usuário) aberto no VOLUME. No nosso caso, o SGMES suportará um máximo de 48 usuários. Assim, este diretório ocupará 768 bytes, isto é, 3 seto-

res (3/4 de um BLA). No primeiro setor estão localizados os diretórios 1 a 16, no segundo os de ordem 17 a 32 e no terceiro e último os de número 33 a 48.

O apontador para o DLB diz respeito ao endereço do diretório do usuário/arquivo, propriamente dito, o qual está contido no DIRETÓRIO-LABEL; o apontador na realidade indica a ordem cardinal em que está o rótulo no DLB.

O número do subdiretório é um número entre 0 e 15 que identifica o descritor.

O tipo index, é um indicador do tipo de volume, no qual o DIX está armazenado:

0 - volume do usuário

1 - volume do sistema

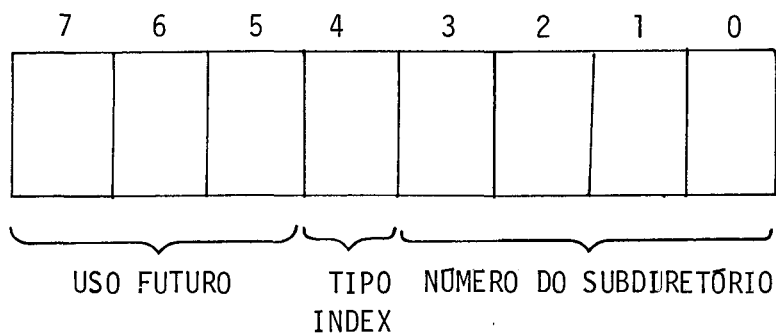


FIG.14. ESTRUTURA DO 1º BYTE DO DIX.

Status é um campo que contém as seguintes informações:

bytes	descrição
1 a 2	número do volume a qual se refere este DIX.
3 a 8	nome do volume
9 a 10	apontadores para (no mínimo) duas áreas livres do DLB.

Diretório é o campo que especifica o diretório do usuário e o primeiro dos seus arquivos no DLB (Ver Figura 15).

BYTES 15 e 14	BYTES 13	BYTES 12 e 11	BYTES 10 e 9	BYTES 8 a 3	BYTES 2 e 1
USO FUTURO	APONTADOR PARA O PRÓXIMO DIRETÓRIO DO USUÁRIO	REGISTRO DE SEGU-RANÇA DE ACESSO E TEMPORA-LIDADE	APONTADOR PARA O DLB	NOME DO DI-RETÓRIO DO USUÁRIO	IDENTIFICA-ÇÃO DO USUÁ-RIO

FIG.15 - ESTRUTURA DO CAMPO "DIRETÓRIO" DO DIX.

Código de Identificação do Usuário é um número entre 0 e 65536 que identifica univocamente o usuário dentro do sistema. Os códigos de 0 a 511 são reservados para diretórios do sistema (Exemplos: bibliotecas de rotinas do S.O.).

Apontador para o DLB é um apontador de valor relativo (número cardinal) que indica o primeiro diretório do usuário que está contido dentro do Diretório de Rótulos DLB.

A finalidade do DIX é, assim de distribuir os pedidos de habilitação de E/S e verificar a permissão de uso ou não do sistema pelo usuário corrente.

Registro de Segurança de Acesso e Temporalidade, tem a seguinte estrutura:

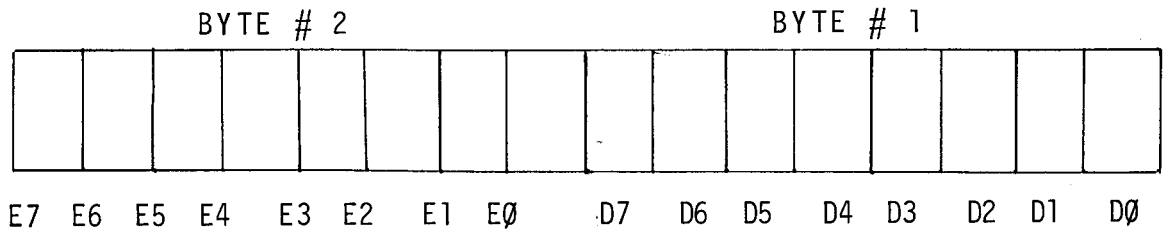


FIG.16. ESTRUTURA DO REGISTRO DE SEGURANÇA DE ACESSO E TEMPORALIDADE.

Onde:

D0 ::= Tipo de acesso I:

1 - acesso público;

0 - acesso privado;

D1 ::= Tipo de acesso II:

1 - restrito (o acesso é restrito a usuários do sistema local).

0 - não restrito (acesso é permitido de qualquer ponto de uma rede).

D2 ::= Tipo de acesso III:

1 - leitura permitida;

0 - leitura não é permitida;

D3 ::= Tipo de acesso IV:

1 - escrita permitida;

0 - escrita não permitida;

D4::= Uso Futuro - deve conter zeros, isto é, deve está desativado (restaurado).

D5::= Temporalidade:

1 - arquivo temporário (destruído quando do seu fechamento);

0 - arquivo permanente;

D6::= Alocação

1 - alocação de BLA's é automática, fazendo-se à medida da necessidade de crescimento do arquivo;

0 - alocação de BLA's é inicialmente solicitada - neste caso o usuário deve especificar o número de BLA's inicialmente desejados quando da criação do arquivo e o quantum de alocação de BLA's (quantos BLA's serão alocadas a cada nova necessidade);

D7::= Continuação I

1 - arquivo continuação de outro existente em volume, que não o corrente;

0 - arquivo totalmente contido no volume corrente.

E0 ::= Continuação II \rightarrow $\begin{cases} 0 & \text{- uso futuro} \\ 1 & \text{- arquivo que continua em outro volume.} \end{cases}$

E1 e E2 ::= Modo de organização do arquivo:

00 - uso futuro

01 - especial;

11 - direto relativo

10 - sequencial;

E3 e E4 ::= Modo de acesso: (atualizado dinamicamente a cada acesso ao arquivo)

- 00 - sequencial;
- 01 - uso futuro;
- 10 - direto relativo;
- 11 - especial.

E5 a E7 ::= uso futuro.

IV.2.2.2. Diretório de Rótulos - DLB

É um diretório do sistema cujo conteúdo é a descrição detalhada dos arquivos existentes em veículos de entrada e saída.

CONTEÚDO

- Descrição Grupal		16 bytes
- Número do Diretório	1 byte	
- Status	15 bytes	
- Diretório		
- Nome Identificação Arquivo		
- Nome Volume	6 bytes	
- Nome diretório/usua.	6 bytes	
- Nome arquivo	6 bytes	
- Extensão	3 bytes	
- Identificação do Usuário criador do arquivo	2 bytes	
- Identificação de usuários co-utilizadores possíveis	4 bytes	

- Data de Criação Arquivo	2 bytes	
- Expiração	2 bytes	
- Última utilização	2 bytes	
- Tamanho do registro	2 bytes	
- Número de BLA's solicitadas quando da alocação pedida	2 bytes	
- BLA's Alocadas		30 bytes
-Segmentação (atê 5)		
- N° de BLA Lógico Inicial	2 bytes	
- N° do BLA Físico Inicial	2 bytes	
- Número de BLA's contíguas alocadas a partir do BLA físico inicial		
- Apontador para o próximo diretório do usuário no DLB	1 byte	
- Uso futuro	2 bytes	
- N° de setores contendo a tabela de índices se arq.seq. indexado	1 byte	
- Tamanho da chave das tabelas de índice (de um a sete caracteres)	1 byte	
- Uso futuro	8 bytes	

Desta forma, vemos que cada elemento do Diretório Rótulo existente tem um comprimento de 80 bytes, além de 16 bytes de descrição geral para cada três elementos; assim o DLB, no caso do SGMES, suportará no máximo 96 elementos, quando então será necessário ocupar 8 BLA's, isto é 32 setores de 256 bytes formatados da seguinte maneira:

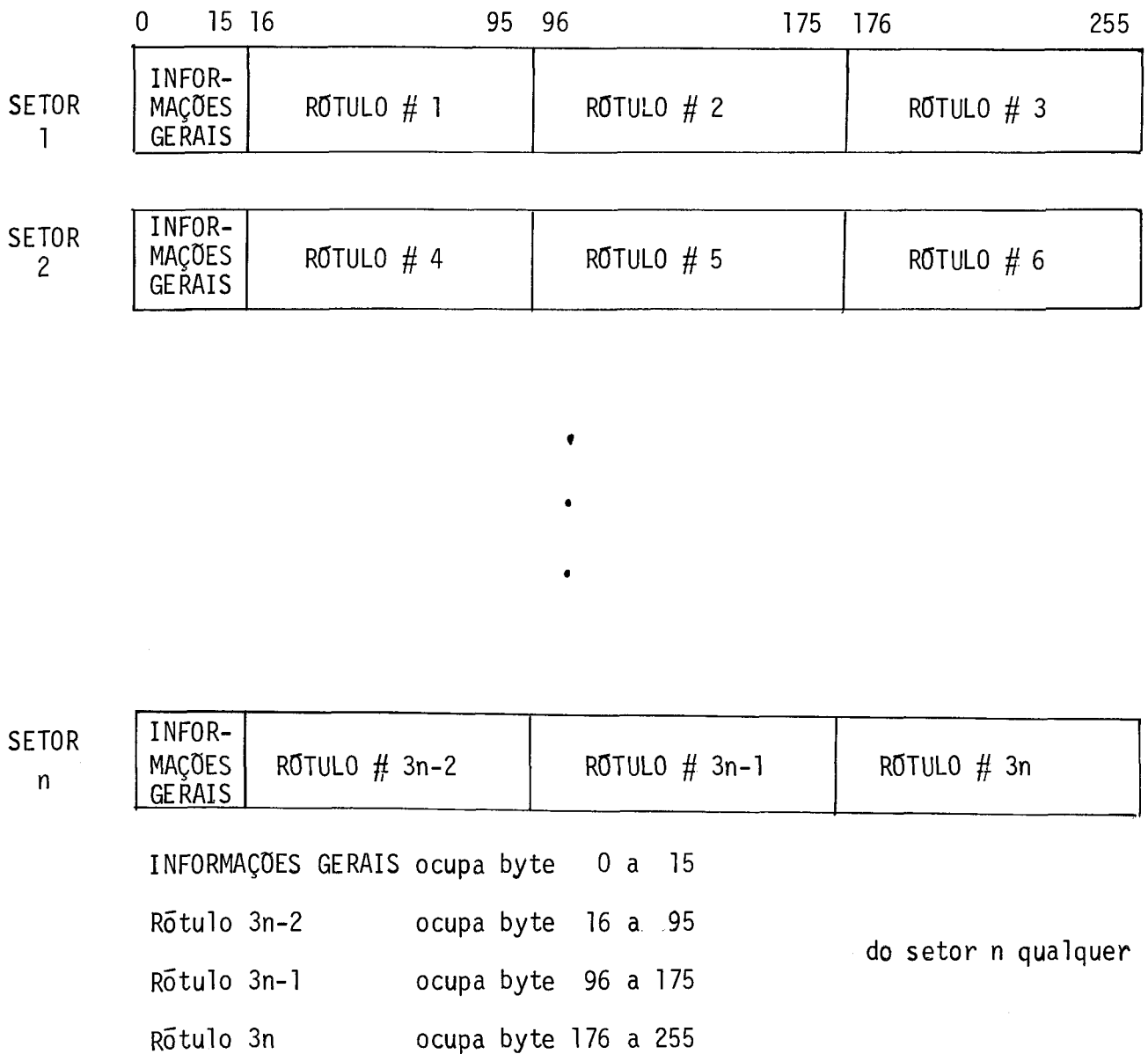


FIG.17. ESTRUTURA DO DLB.

Assim, com $n = 32$ no máximo, temos a seguinte estrutura do DLB:

	INFORMA- ÇÕES GERAIS	RÓTULO 3i-2 ou 3xINC+1	RÓTULO 3i-1 ou 3xINC+2	RÓTULO 3i ou 3xINC+3	BASE + INC
1		1	2	3	+ 0
2		4	5	6	+ 1
3		7	8	9	+ 2
4		10	11	12	+ 3
5		13	14	15	+ 4
6		16	17	18	+ 5
7		19	20	21	+ 6
8		22	23	24	+ 7
27		79	80	81	+ 26
28		82	83	84	+ 27
29		85	86	87	+ 28
30		88	89	90	+ 29
31		91	92	93	+ 30
32		94	95	96	+ 31

FIG.18. ESTRUTURA DLB.

O campo Status é dividido em três subcampos, cada qual com 5 bytes, referentes a cada um dos três diretório/rótulo descritos no 1/4 de BLA (isto é, um setor):

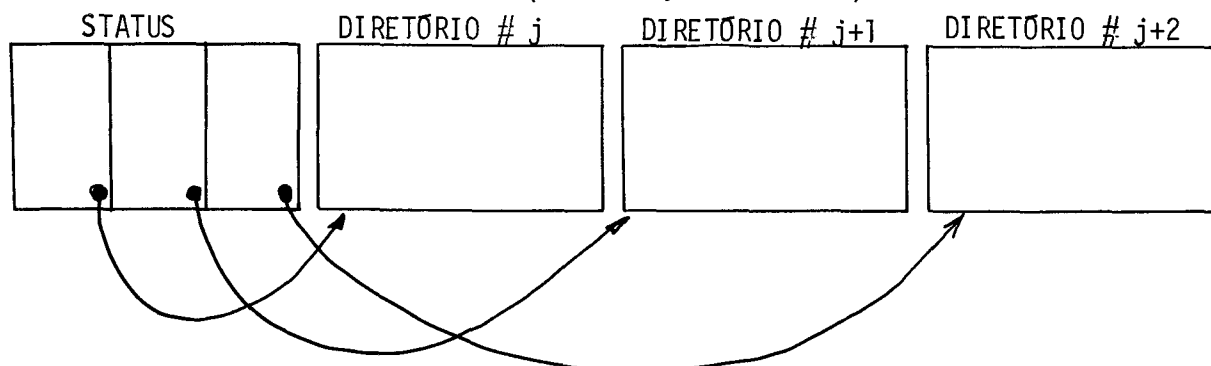


FIG.19. ESTRUTURA DE CADA REGISTRO DO DLB

Cada sub-campo citado acima contém:

- Número físico do BLA contendo este setor 2 bytes
- Número de BLA's alocados ao diretório 2 bytes
- Uso futuro 1 byte

O campo Nome do Arquivo que tem comprimento de 15 bytes, tem a seguinte estrutura:

<nome # 1> <nome # 2> <extensão>

que corresponde, respectivamente, aos nomes do diretório (6 bytes), nome do arquivo (6 bytes), e tipo do arquivo ou extensão (3 bytes).

Os tipos de arquivos são palavras reservadas do sistema operacional, entre as quais constam (maiores detalhes, ver na documentação da Linguagem de Comandos - LICO):

- DAT: arquivo de dados
- FOR: arquivos em simbólico FORTRAN

- PAS: arquivos em simbólico PASCAL
- BAS: arquivos em simbólico BASIC
- COB: arquivos em simbólico COBOL
- EXE: arquivos em linguagem objeto/executável
- DIR: arquivo diretório
- LIC: arquivo LICO
- LIB: arquivo LIBRARY
- SIX: arquivo SEQUENCIAL INDEXADO
- REL: arquivo RELOCÁVEL
- SIM: arquivo SIMBÓLICO
- HEX: arquivo em código hexadecimal
- BIN: arquivo em código binário
- BAK: arquivo backup
- etc.

No campo data de criação é alocada a data em que foi criado o arquivo, na forma Juliana. Os campos "Expiração" e "Última Utilização" contêm um número que corresponde, respectivamente, ao número de dias, a partir da "data de criação", que o arquivo terá validade e o número de dias a partir daquela data em que o mesmo foi utilizado mais recentemente.

O campo "tamanho de registro" contém um número que corresponde ao comprimento do registro em bytes. Se este campo contiver valor zero, significa que o comprimento do registro é variável.

O campo "BLA's Alocados" pode ser visto em maiores detalhes no capítulo III.

O campo "Apontador para o próximo Diretório" do

usuário no DLB" contém um valor que corresponde ao valor relativo (número cardinal) da localização do mesmo dentro do DLB:

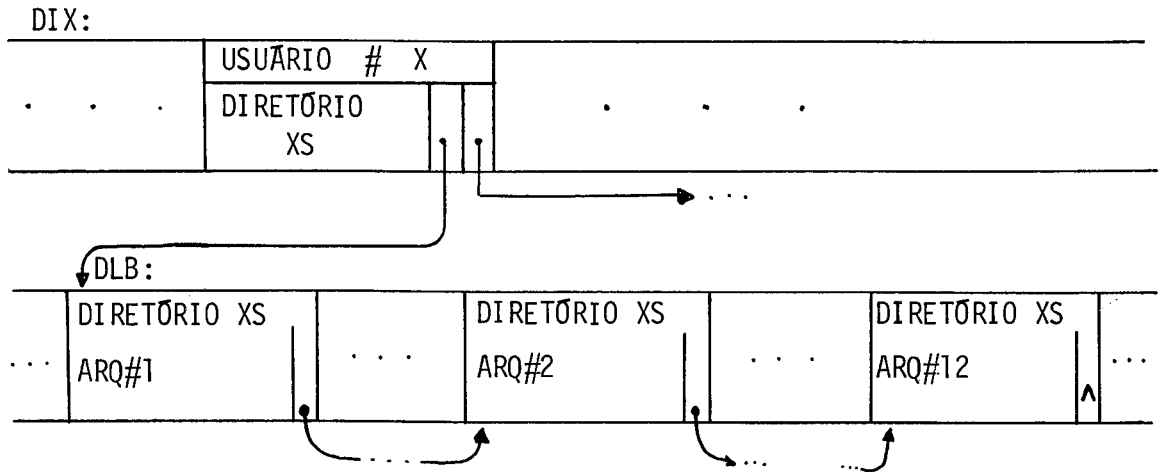


FIG.20. LIGAÇÃO DIRETÓRIO - ARQUIVOS.

Como os diretórios na área do sistema serão tratados da forma usual, de maneira análoga aos arquivos da área da dados, não é necessário alocar todo o espaço ao DLB de uma só vez; assim alocaremos inicialmente apenas dois BLA's, quando da inicialização do disco ou volume; os demais BLA's serão alocados à medida da necessidade.

Este diretório formará um arquivo sozinho.

IV.2.2.3. Diretório de Controle de Alocação de Espaço DCA

Consiste em um subdiretório do Diretório-do-Sistema que contém informações sobre os BLA's já alocados ou não ao sistema de arquivos em veículos de acesso direto e daqueles cujas trilhas a que pertencem estejam defeituosas. É representado em forma de uma cadeia de 1024 bits (128 bytes) nesta primeira versão, onde cada bit representa um BLA.

Esta cadeia informa a ocupação dos BLA's (1 - ocupada ou alocada a algum arquivo ou ainda com algum defeito físico; 0 - livre ou disponível) como visto na figura 10.

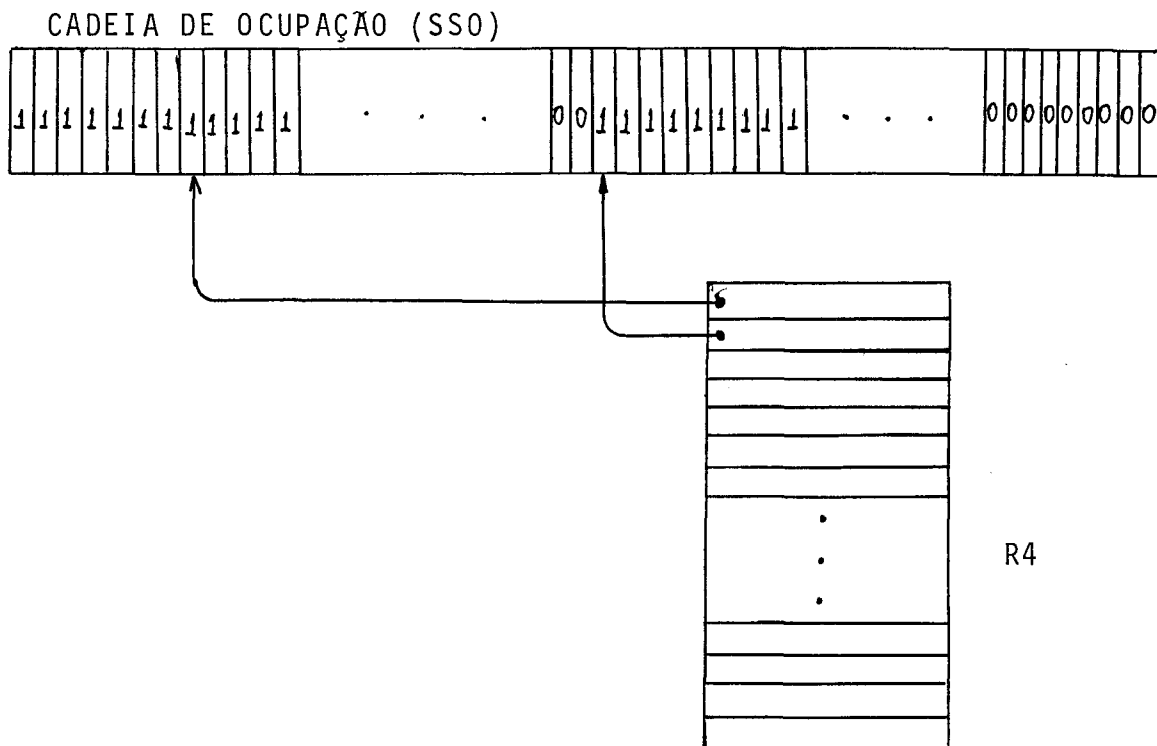


FIG.21. CONTROLE DE TRILHAS DEFEITUOSAS.

Quando uma trilha apresentar defeito, os bits da subcadeia de ocupação têm que ter ativados para "1" naqueles bits correspondentes aos BLA's pertencentes à trilha defeituosa (ver figura 21). Para efetivar este controle, é alocado no registro R4 o número do 1º BLA físico da pilha defeituosa; este registro permite o controle de até 28 trilhas defeituosas. Além desse limite o veículo será considerado pelo sistema como impróprio para uso.

Quanto a este procedimento, pode ser levantada a objeção de que leva ao desperdício de 1/2 BLA de uma trilha vizinha da trilha defeituosa, mas inicialmente isso será to-

lerado, de modo a que o critério de simplicidade se sobreponha a outros tais como de precisão, generalidade, etc; posteriormente poderemos tratar tal problema, incluindo um tratamento de recuperação da informação contida em uma trilha defeituosa. Uma outra forma de contornar esse problema é o uso no sistema de veículos formatados com trilhas contendo número de setores múltiplo do número de setores de um BLA.

Neste diretório também haverá os seguintes registros:

- R1 - Número do BLA físico a partir do qual inicia a área de sistemas; (2 bytes)
- R2 - Aponta para o primeiro BLA livre existente (2 bytes);
- R3 - Contador de BLA's disponíveis - 2 bytes;
- R4 - Este é um registro composto de 28 subregistros de 2 bytes, onde cada um deles aponta para o primeiro BLA de cada trilha defeituosa.

Os registros R2 e R3 devem ser atualizados automaticamente a cada alocação de novo BLA.

Como a cadeia de ocupação (SS0) tem 128 bytes conforme já citado, este subdiretório ocupará, então, no máximo 1 setor de 256 bytes.

Para uso futuro foram reservados neste subdiretório 68 bytes.

IV.2.2.4. Diretório Identificação - DID

Este sub-diretório contém a descrição do volume que o contém detalhando suas características físicas e sua identificação lógica. Seu conteúdo é:

Número do sub-diretório		1 byte
Número do volume		2 bytes
Nome do Volume		6 bytes
Data de criação do volume		2 bytes
Nome do usuário criador		30 bytes
Trilhas defeituosas	28 x 2 =	56 bytes
Código usuário criador		2 bytes
Quantidade de superfícies no volume		1 byte
Quantidade de trilhas por superfície		2 bytes
Quantidade de setores por trilha		1 byte
Quantidade de bytes por setor		2 bytes
Endereço do BLA físico inicial dos arquivos		
diretórios dos sistemas:		
- &DIR 99		2 bytes
- &DIR 98		2 bytes
- &DIR 97		2 bytes
Uso futuro		145 bytes

Data de criação do volume é a data em que o corrente volume foi inicializado, isto é, a data em que o veículo foi formatado e inicializado com as informações e diretórios básicos, de forma a torná-lo conhecido e manipulável pelo sistema.

Um campo composto de 28 subcampos de 2 bytes cada,

é denominado de trilhas defeituosas e nele está contida a informação referente ao número do BLA físico inicial da trilha defeituosa.

Todos os demais campos são auto-explicativos.

IV.2.2.5. Arquivos de Diretórios do Sistema

Os arquivos de diretórios do sistema são arquivos mantidos pelo sistema SGMES, os quais formam a área do sistema. Esses arquivos agregam em si um ou mais diretórios. São os seguintes os arquivos diretórios:

&DIR99 - este arquivo é formado pelos seguintes diretórios:

DIX - Diretório Index (o qual ocupará no máximo três setores);

DID - Diretório de Identificação (este ocupará também no máximo 1 setor);

DCA - Diretório de Controle de Alocação de Espaço em Disco (ocupa 1 setor); e

DDD - é uma área que armazena uma cópia dos descritores de arquivos abertos e usados pertencentes a este volume.

Este arquivo ocupará 12 setores e é criado quando da inicialização do volume.

Estrutura do &DIR99:

DID	DCA	DIX	DDD
1º SETOR	2º SETOR	3º SETOR AO 5º SETOR	6º SETOR AO 12º SETOR

FIG.22 - ESTRUTURAS DO ARQUIVO DIR 99.

&DIR 98 - este arquivo é formado unicamente pelo DLB - Diretório de Labels ou Rótulos - o qual, quando da inicialização do volume ocupará dois BLA's, e quando do uso de todas as extensões com o mínimo de alocação ocupará 24 rótulos de arquivos no mínimo.

&DIR 97 - este arquivo ocupa parte da primeira trilha do volume, e nele haverá um descritor indicando o endereço físico de registros especiais da área do sistema. O restante do espaço disponível (alocado quando da inicialização do volume) conterá rotinas especiais do sistema operacional (quando o disco for caracterizado como "do sistema") carregadas no momento de "bootstrap".

Este diretório localizar-se-á nos primeiros setores das primeiras trilhas de cada volume.

Os endereços constantes são, em ordem (endereço = superfície/trilha/setor/byte) os seguintes:

- do DIR& 99
- do DIR& 98
- do "Bootstrap"
- do Programa de Inicialização
- das rotinas de manutenção

- das rotinas do Sistema Operacional carregáveis em tempo de "bootstrap".

IV.2.3. Descrição da Área de Dados

É a área do volume de informação onde o usuário tem acesso de forma direta para leitura e escrita de seus arquivos e dados.

A alocação de espaço para os arquivos a serem armazenados nessa área é realizada bloco-a-bloco, isto é, quando da criação do arquivo o usuário tem duas possibilidades de alocação:

- alocação inicial solicitada
- alocação dinâmica

Na alocação solicitada o usuário especifica o número de BLA's inicialmente requeridos (menor ou igual a 90 BLA's) pelo seu arquivo; se há disponibilidade de alocar a solicitação, o sistema o faz automaticamente. Quando do fechamento do arquivo, é feito um balanço entre o solicitado e o realmente utilizado, desalocando os BLA's não usados. Ainda nesse caso, se o arquivo necessitar de um número de BLA's maior que o inicialmente solicitado, é realizado internamente pelo SGMES uma mudança de atributo de alocação, tornando-a dinâmica. Se não há disponibilidade de BLA's para atender à solicitação inicial, o SGMES rejeita o pedido, retornando ao usuário a quantidade disponível no sistema.

Na alocação dinâmica ou automática, o sistema alo-

ca um quantum de BLA se arquivo tipo sequencial e 16 se arquivo tipo especial, quando da sua criação. Após esta alocação inicial, o SGMES vai alocando um quantum BLA à medida da necessidade de uso de novos espaços pelos arquivos.

Esse quantum é um número (0 a 255), especificado pelo usuário, que indica a quantidade de BLA's que devem ser alocados à cada nova necessidade de espaço no disco.

IV.3. Pedidos de Entrada e Saída

A solicitação de qualquer entrada e saída ao SGMES é encaminhada pelo processo do usuário através do Núcleo por meio de uma estrutura de dados denominada de BCES conforme já vimos anteriormente, a qual é responsável pela comunicação entre o usuário e o sistema.

Quando da abertura ou criação de um arquivo, o usuário informa o endereço do BCES e o número da unidade requerida, afim de tornar o BCES conhecido pelo sistema. Existirá um BCES para cada arquivo ativo no sistema. Todo este procedimento é realizado pelo Núcleo, onde se informa o endereço do BCES e a identificação do volume, diretório, arquivo, montando tais informações no BCES, depois, então, o Núcleo chama o SGMES, se necessário, para executar a criação física do arquivo.

Para cada canal o SGMES manterá um registro de 8 bits, para controle de seu uso:

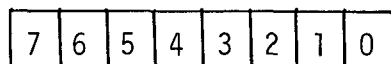


FIG.23. REGISTRO DE CONTROLE DE CANAL.

onde: bit 7 - uso do canal; sendo que:

1 - canal de E/S em uso

0 - canal livre

bit 6 a 4 - uso futuro

bit 3 a 0 - indica quais unidades estão em uso,
conforme o correspondente bit estiver
ativado ou não:

bit 3 - unidade 1 deste canal

bit 2 - unidade 2 deste canal

bit 1 - unidade 3 deste canal

bit 0 - unidade 4 deste canal

Quando do pedido inicial de acesso a um dispositivo de entrada e saída, é associado ao arquivo físico do usuário um "nome lógico" através de diretivas do núcleo do sistema operacional. Esse "nome lógico" é o nome pelo qual o processo se referênciava para ter acesso a entradas e saídas através de um periférico. Isso nos permite independência entre as unidades físicas (periféricos propriamente ditos) onde se realizam as operações de E/S, solicitadas por um determinado programa, isto é, uma entrada e saída de um determinado programa depende única e exclusivamente do "nome lógico" associado aos seus arquivos, e não de uma determinada unidade física periférica.

A associação do "nome lógico" a uma determinada unidade física pode realizar-se de duas formas básicas:

- 1: via comando CONECTE da linguagem de comandos;
- 2: pelo núcleo do sistema operacional.

Quando da solicitação de abertura de um arquivo, o SGMES retorna ao usuário um número, o qual a partir desse momento, deve fazer parte do BCES, para todas as solicitações de E/S. Esse número refere-se ao número da entrada na tabela de descrição de arquivos e unidades em uso.

Ver outros detalhes nos itens II-1 e II-2.

IV.4. Funções Operativas, Chamadas ou Diretivas do SGMES

As funções operativas do SGMES são classificadas em quatro categorias:

- CAT§1 - Funções de Entrada - são aquelas que executam operações apenas de leitura sobre os volumes;
- CAT§2 - Funções de Saída - são aquelas que executam somente operações de escrita sobre os volumes;
- CAT§3 - Funções de Habilitação - são aquelas que operam sobre os volumes de forma a permitir ou habilitar o seu uso pelo sistema;
- CAT§4 - Funções de Gerência - aquelas que executam operações de verificação, atualização e modificação sobre os diretórios e arquivos, bem como toda a organização do próprio volume de dados.

No capítulo VI daremos a descrição detalhada de cada função pertencente às categorias de operações definidas acima.

Para a execução de uma dessas funções, após a solicitação por um processo requisitante (PR), um processo executor (PE) denominado "gerente" recebe o pedido e o avalia, afim de tomar as providências necessárias à execução do pedido, distribuindo as tarefas às diversas funções das diversas categorias existentes.

As funções pertencentes a cada categoria são:

CAT§1:

- F0101 - Ler próximo registro
- F0102 - Ler registro especificado
- F0103 - Ler "n" caracteres a partir do apontador corrente
- F0104 - Salte para o início próximo registro; usado quando de leitura parcial de registros.

CAT§2:

- F0201 - Grave próximo registro
- F0202 - Grave registro especificado
- F0203 - Grave "n" caracteres a partir do apontador corrente
- F0204 - Grave fim de registro.

CAT§3:

- F0301 - Retorne um registro (backspace)
- F0302 - Avance "n" registros
- F0303 - Volte ao início do arquivo (Rewind)
- F0304 - Inicialize Volume (rotina Stand-Alone)

- F0305 - Habilite Volume
- F0306 - Abra arquivo (novo) especificado
- F0307 - Habilite arquivo (abra arquivo já existente)
- F0308 - Feche arquivo especificado
- F0309 - Feche arquivos abertos pelo processo "X"
- F0310 - Feche todos arquivos abertos no sistema

CAT§4:

- F0401 - Mostre diretórios do Sistema
- F0402 - Mostre diretório especificado
- F0403 - Remova arquivo especificado
- F0404 - Altere atributos de arquivo
- F0405 - Verifique e mostre arquivos existentes no volume
- F0406 - Verifique e mostre BLA's usados e/ou disponíveis
- F0407 - Mostre status corrente de arquivo

IV.5. Descritores Para Manipulação de Entrada e Saída DEMES

São diversos os descritores mantidos pelo SGMES para controle dos acesso de entradas e saídas ao sistema de discos flexíveis. Esses descritores contêm informações sobre o estado do sistema no instante presente.

Os descritores são entidades ou estruturas de dados armazenadas na memória principal na parte reservada ao SGMES, que a ela tem acesso exclusivo.

Devido ao fato de que a multiprogramação do sistema do qual faz parte o SGMES é realizada por "ocupação-total" do programa na memória, não há possibilidade de um número razoável de programas executando. Desta forma, após diversas análises decidimos (devido às restrições de memória e otimização na comunicação de processos) restringir o número máximo de arquivos abertos e habilitados, para uso simultâneo por um ou vários processos, a 24 arquivos. Essa restrição dependerá do número de canais e será a seguinte:

- 12 - quando apenas um canal de E/S for usado para disco;
- 16 - quando dois canais forem usados;
- 20 - quando três canais forem usados e
- 24 - se quatro canais de E/S estiverem ocupados com unidades de disco flexível.

Nessa quantidade especificada estão incluídas as possíveis aberturas de arquivos pertencentes ao sistema operacional.

IV.5.1. Descritor de Ocupação de Blocos - DESOB -

Este é um descritor que é criado quando da inicialização ou habilitação de um determinado volume. É na realidade uma cópia do diretório DCA (Controle de Alocação de Espaço em Disco), que consiste de uma cadeia de bits, representando os BLA alocados e os BLA's contidos em trilhas defeituosas, além de 4(quatro) registros R1, R2, R3 e R4 indicando respectiva-

mente:

- número da BLA física a partir da qual se inicia a área de diretórios;
- apontador indicando a primeira BLA disponível;
- número de BLA's disponíveis;
- apontadores que indicam o número da 1a. BLA de cada trilha defeituosa.

Este descritor é atualizado concomitantemente com qualquer operação que utilize funções de categorias CAT2, CAT3 e CAT4 e terá um máximo de 4 entradas, representando 4 volumes abertos.

Ocupará, na versão corrente, 516 bytes, com a seguinte configuração para cada entrada:

Entrada i

Nº DO VOLUME	_____	2 bytes
CANAL/UNIDADE	_____	1 byte
SS0	_____	64 bytes
R1	_____	2 bytes
R2	_____	2 bytes
R3	_____	2 bytes
R4	_____	56 bytes

		SUBTOTAL = 129 bytes

Como, existirão até 4 entradas, a máxima ocupação deste descritor será 516 bytes.

Para entendimento de nomenclatura usada na descri-

ção da entrada i acima, ver "Diretório de Controle de Alocação de Espaço - DCA".

IV.5.2. Descriptor de Volumes Habilitados - DEVOH

Este descritor controla os volumes que estão habilitados no sistema, bem como informa que usuários estão capacitados para usá-los.

É importante notar aqui algumas características da função do sistema operacional que têm influência sobre a habilitação de volumes.

Quando um processo qualquer vai habilitar um periférico ou arquivo (que corresponde no nosso caso ao volume) é invocada inicialmente uma operação do núcleo denominada "ABRARQUIVO" cuja função é alocar na BCES especificada pelo usuário (importante lembrar que deve existir sempre uma BCES por arquivo usado, cuja criação é de competência do usuário), os campos NOMEVL.NO, NOMEVL.VOLUME, NOMEVL.DIRETOR, NUMSER, REDE1, REDE2, NUMPRO, FISICA e IDUSU, conforme descrição dada no apêndice 3.

Assim, após a execução da rotina "ABRE-ARQUIVO" do núcleo, é que é, então, ativado o SGMES para a abertura e habilitação física do volume ou arquivo. Como uma das características do SGMES é dar suporte a um sistema operacional que permita ambiente computacional de multiprogramação, o DEVOH é o elemento responsável, para o controle de quais usuários podem ter acesso a quais volumes e arquivos.

Cada entrada deste descritor conterá as seguintes informações:

- 1 - NVOLDG - NÚMERO LÓGICO DO VOLUME NO SISTEMA(SGMES) - 2 bytes
- 2 - NUMSEV - NÚMERO DO SERVIÇO - 2 bytes
- 3 - NOMVOL - NOME DO VOLUME - 6 bytes
- 4 - FÍSICA - Nº DA UNIDADE FÍSICA - 1 byte
- 5 - USUHAB - USUÁRIOS HABILITADORES (4 NO MÁXIMO) - 8 bytes
- 6 - APTDIR - APONTADORES PARA DIR DO USUÁRIO(4 NO MAX, isto é, um dir por usuário) - 4 bytes
- 7 - USOFUT - USO FUTURO - 2 bytes

Assim, cada entrada tem 25 bytes; como será permitido um máximo de (quando do sistema completo) 16 volumes o DEVOH terá 400 bytes. No entanto, na primeira versão, só serão permitidas 4 volumes por sistema, devido a restrições de memória. Assim o DEVOH corrente ocupará 100 bytes.

IV.5.3. Descritor de Arquivos em Uso - DESAU

Este descritor controla e mantém informações ou catálogo de informações sobre os arquivos abertos e em uso pelos diversos processos ativados pelo sistema.

As informações contidas aqui são complementadas por aquelas contidas nos diversos BCES (lembramos aqui que, para cada arquivo aberto por um determinado processo, existirá um BCES a ele vinculado ou associado).

São as seguintes as informações contidas no DESAU para cada arquivo aberto:

1 - NLOGAB - Número Lógico da abertura _____	1 byte
2 - EDBCES - Endereço do BCES correspondente _____	2 bytes
3 - BLALOC - BLA's alocadas _____	30 bytes
4 - BLALOC - Número do Buffer alocados _____	1 byte
5 - TOMREG - Tamanho do Registro _____	2 bytes
6 - BLTUA - BLA atual	

Nº BLA	SETOR	3 bytes
2 bytes	1 byte	

7 - FORMAC - Forma de acesso corrente _____	1 byte
8 - WUFUT - Uso futuro _____	1 byte
9 - FATBLO - Fator de Bloco _____	1 byte
10 - APINTB - Apontador de posição dentro do buffer ____	1 byte
11 - NPROHA - Número do processo habilitador (atê 2) ____	2 bytes
	45 bytes

Cada entrada neste descritor (que se refere a cada arquivo aberto existente no sistema) ocupa 45 bytes . De acordo com o estabelecido quando do detalhamento do descritor DEMES, a área total a ser ocupada pelo "descritor DESAU" será:

CANAIS HABILITADOS	ÁREA OCUPADA (EM BYTES)
1	540 referente a 12 entradas
2	720 referente a 16 entradas
3	900 referente a 20 entradas
4	1080 referente a 24 entradas

IV.5..4. Descritor de Buffers Requisitados - DEBUR -

Este descritor mantém informações cruzadas sobre os buffers existentes com os processos que os estão utilizando. Isso deve-se ao fato de que limitamos os buffers para entrada e saída a um número de vinte e quatro (24), sendo que a sua utilização no aspecto global, embora dual (isto é, serve para receber e enviar mensagens) tem finalidade única para cada arquivo aberto, isto é, não pode ser usada concomitantemente para enviar e receber mensagens. Como o número máximo de arquivos abertos simultâneos é 24 e cada um deles usa no mínimo um buffer, deveríamos ter 24 buffers, e se ainda todos os arquivos abertos estiveram em uso lendo e gravando, seriam necessários, então, 48 buffers. Assim, devido à restrição de espaço de memória e à melhor utilização do uso dos buffers, a sua alocação será realizada dinamicamente.

O esquema ou rede de estados dos buffers usando o conceito de Redes de Petri, é o seguinte:

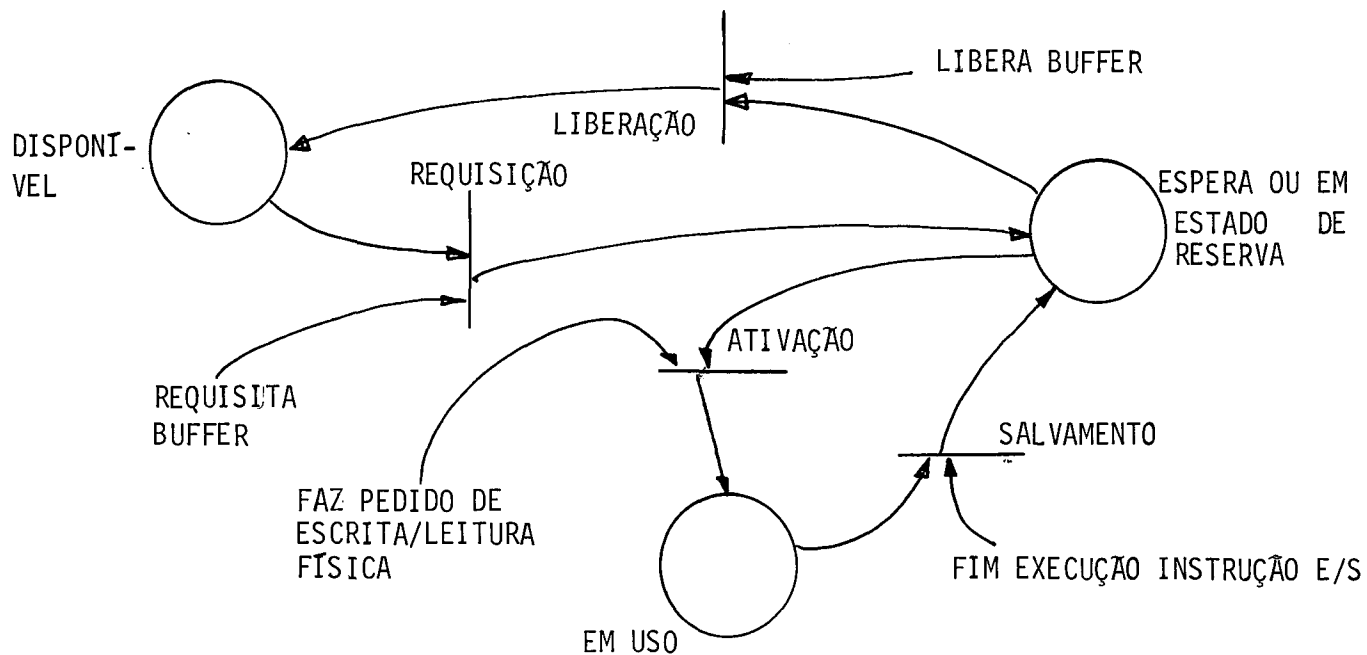


FIG.24. REDE DE ESTUDO DOS BUFFERS DO SISTEMA.

No diagrama da Figura 24 acima, vemos um esquema bastante geral da sistemática adotada para a cessão, uso e liberação de buffers internos ao SGMES aos processos de usuário utilizando entrada e saída. Os buffers são associados a um determinado processo, pelo seu número, pela espécie de uso (recepção ou envio) e pelo arquivo que o utiliza (aqui representado pelo número da abertura do arquivo no DESAU).

No princípio todos os buffers alocados ao sistema estarão em estado "disponível" até que haja por parte de algum processo do usuário um pedido de E/S, quando então o SGMES fará um pedido de buffer (quando não existir nenhum ligado aquele processo requisitante-PR), deslocando-o para o estado de "espera" ou de "reserva". A partir de então sempre que o PR fizer solicitação de E/S o buffer será usado, continuando no mesmo estado. O buffer é passado para o estado de em uso sempre que o SGMES detectar uma situação de escrita ou leitura física que o utilize; após completar essa operação o mesmo é retornado ao estado "reserva". O buffer será reposto na condição "disponí-

vel" sempre que o pedido corrente de E/S do PR tenha sido completado com êxito. Ele continuaria no estado "reserva" se no pedido de E/S executado tiver implícita a continuação de envio de informações para o buffer.

A monitorização ou melhor, o controle da exclusão mútua e de sincronização será realizado pela estrutura de monitor disponível no núcleo do sistema operacional.

As informações contidas no DEBUR (para cada entrada) são:

1 - NUB - Número do buffer	-	1 byte
2 - NAB - Número de abertura do arquivo	-	1 byte
3 - EBS - Endereço do Buffer do sistema	-	2 bytes
4 - NPR - Número do processo usando	-	2 bytes
5 - ETX - Estado	-	2 bytes
		<hr/>
		8 bytes

Os campos acima numerados de 1 a 4 são auto explicativos; o campo Estado (2 bytes) tem a estrutura vista na figura abaixo:

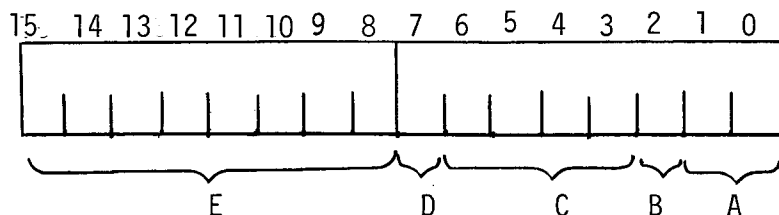


FIG.25. ESTRUTURA DO CAMPO "ESTADO".

A - estado

- = 00 Disponível
- = 01 Espera
- = 10 Uso futuro (não usado)
- = 11 Em uso

B - tipo de uso do buffer

0 - buffer de recepção

1 - buffer de transmissão ou envio

C - número da unidade lógica associada a este buffer

D - se igual a 0 indica que quando um cancelamento de um pedido de E/S o buffer será colocado em estado "espera"; se igual a 1, indica que, na mesma situação anterior, o buffer será colocado no estado "disponível".

E - apontador interno = posição do 1º byte dentro do buffer para próxima leitura e/ou gravação.

Assim, cada entrada no DEBUR tem 8 bytes; como são possíveis coexistirem 24 arquivos abertos (caso em que a cada arquivo está associado apenas um buffer) ou no máximo 24 buffers conforme foi estabelecido na descrição do descritor DEMES, DEBUR ocupará no máximo 192 bytes:

CANAIS DE E/S HABILITADOS	Nº DE BYTES OCUPADOR P/DEBUR
1	96
2	128
3	160
4	192

Além disso, existem três pilhas denominadas PDISPO, PESPER, PEMUSO, usadas para implementar o esquema de compartilhamento dos buffers e sincronização de seu uso

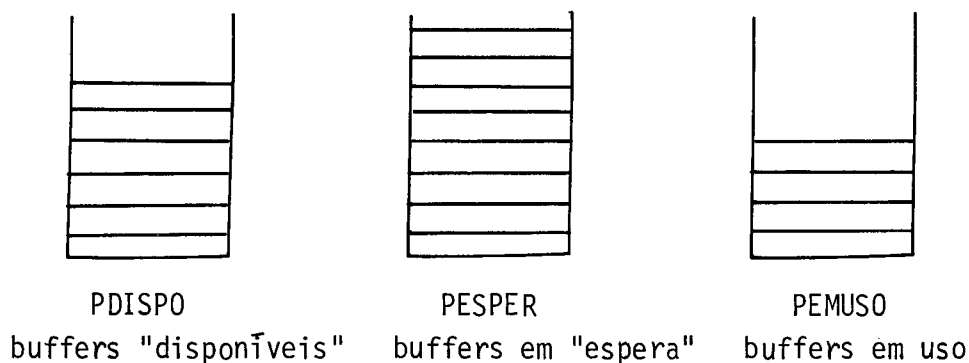


FIG.26. PILHAS DE CONTROLE DE ESTADO DE BUFFER'S.

PDISPO e PESPER tem 48 bytes, isto é, 24 posições (para alocar os números dos buffers) de 2 bytes.

PEMUSO tem 8 bytes, isto é, 4 posições (para alocar os números dos buffers) de 2 bytes, uma referente a cada canal de E/S habilitado para uso em disco (máximo 4).

IV.5.5. Outros Descritores

Existem, ainda, na memória específica do SGMES outros descritores que são na realidade uma série de registros que caracterizam os diversos estados do sistema de entrada e saída em disco e uma série de buffers que são alocados, à medida da necessidade, aos arquivos abertos para as operações de leitura e/ou gravação.

Os registros e buffers referidos acima serão des-

critos com detalhes quando da definição das diversas diretivas (funções operativas) suportadas pelo SGMES, o que será feito no capítulo VI.

O próprio BCES (Bloco de Controle de Entrada e Saída) é considerado pelo SGMES como um descritor, no qual são alocadas aquelas informações que o processo requisitante tem acesso sem necessidade de fazer pedido específico ao sistema operacional.

CAPÍTULO V
DESCRIÇÃO DE FUNÇÕES OPERATIVAS OU
DIRETIVAS DO SGMES

V.1. Funções do SGMES

Neste capítulo faremos uma descrição, a mais detalhada possível das diversas funções do SGMES. Como vimos no capítulo anterior; as funções se dividem em quatro categorias: de escrita, de leitura, de habilitação e de gerência de arquivos. É importante, também, ressaltar que a execução de qualquer função é sempre realizada pelo encaminhamento do BCES ao sistema operacional, via uma operação de "depósito" do núcleo.

Cada BCES é composto de duas partes, uma fixa que é comum a qualquer solicitação de entrada e saída independente do equipamento, isto é, contém informações que em geral são comuns tanto a discos, teclado, vídeo, impressora, etc, e de uma parte variável que conterá as informações específicas requeridas por cada periférico em cada tipo de operação ou função que irá executar. Para maiores detalhes sobre a estrutura do BCES, ver apêndice 3.

Neste capítulo, após a explicação de cada função, principalmente daquelas menos triviais, apresentaremos um algoritmo escrito em linguagem de especificação de nível bastante alto, para tornar mais claro o funcionamento do sistema.

V.2. Funções da Categoria 01

As funções pertencentes a esta categoria são de leitura, apenas, isto é, são funções de saída de informações do veículo e não envolvem qualquer espécie de manipulação de "descriptores", com exceção de atualização de apontadores e contadores. As funções classificadas nesta categoria são:

#F0101 - Ler próximo registro

Esta função se aplica a arquivos que são compostos tanto de registros de tamanho fixo como de tamanho variável. Ela também, só é permitida quando a forma de acesso é sequencial ou direto relativo.

INICIO

```

Se BCES.ENTRAD.DESAU = A
  então
    erro;
Se DESAU.FORMAC = seq ou direto
  então
    se DESAU.NUMBUF = A
      então
        pede buffer;
        ativa buffer;
        decodifica endereço disco;
        ler fisicamente;
        salvar buffer;
        transfere registro lógico;
      se último registro ou fim de buffer
        então
          libera buffer;
        atualiza descritores;
    senão
      erro,

```

FIM

#F0102 - Ler registro especificado

Esta função faz a leitura de um registro qualquer especificado. Ela implementa o acesso direto relativo e o sequencial indexado. A etiqueta do registro é o seu número cardinal dentro do arquivo.

Após decodificar o número do registro em trilha/setor (e superfície se for o caso), faz a leitura física ou tratamento lógico do registro.

#F0103- Ler n caracteres a partir do apontador corrente, no registro especificado.

Esta função lê n caracteres a partir da posição corrente do apontador do buffer associado ao arquivo.

Os n caracteres devem estar contidos totalmente dentro de um registro. Se um registro de tamanho fixo contiver apenas, m, $m < n$, caracteres disponíveis no registro corrente, só serão transmitidos m caracteres e enviado, via BCES, código de advertência.

Esta função pode tanto ser aplicada a registros de tamanho fixo quanto os de tamanho variável.

#F0104 -Salte próximo registro especificado - usada para posicionar no início do próximo registro, normalmente usada em conjunto com F0103 (pode ser um modificador de F0103)

Na realidade, posiciona o cursor ou apontador no início do próximo registro, mesmo se necessitando de leitura física, isto é, se no buffer ainda tiver outro registro (caso fator de bloco > 1) posiciona no início desse registro; caso contrário ler fisicamente um novo setor (imediatamente subsequente ao anterior) e posiciona o apontador no início do primeiro registro lógico, sem no entanto fazer transferência de informação.

Abaixo apresentamos um quadro com referências cruzadas entre os tipos de leitura e as diversas formas de acesso e organização de arquivos (onde X significa que pode ser utilizada a função e - não pode):

FUNÇÃO	ORGANIZAÇÃO:								
	SEQUENCIAL			DIRETA(RELATIVA)			SEQ INDEXADA		
	ACESSO								
	SEQ	REL	SEQX	SEQ	REL	SEQX	SEQ	REL	SEQX
F0101	X	X	-	X	X	-	-(1)	-(1)	-
F0102	-	X	-	X	X	-	-(1)	-(1)	X
F0103	X	X	-	X	X	X	-(1)	-(1)	X
F0104	X	X	-	X	X	-	-	-	X

(1) pode ser usada, mas não deve face as incertezas que surgirão no tratamento dos registros dos arquivos.

QUADRO DE REFERÊNCIAS DE TIPO DE LEITURAS X FORMAS DE ACESSO E ORGANIZAÇÃO DE ARQUIVOS.

V.3. Funções da Categoria 02

Esta categoria engloba as funções que fazem escrita nos veículos de entrada e saída, isto é, fazem gravações. São as seguintes as funções desta categoria.

#F0201 - Grave registro

Esta função recebe o registro contido no buffer do usuário e faz sua escrita física (se não houver necessidade de blocagem ou se este for o último registro do bloco físico) ou lógica (quando for permitida blocagem e não for o último do bloco corrente).

A blocagem é feita automaticamente sempre que o tamanho do registro for menor ou igual a 128 bytes e é totalmente transparente ao usuário, no caso registros de tamanho fixo.

#F0202 - Grave Registro Especificado

A especificação do registro deve ser o número cardinal do mesmo dentro do seu arquivo. Esta função envolve a leitura do bloco onde se localizará o registro especificado se ele não estiver na memória. Após esta leitura, o registro é incluído logicamente no bloco e, fica aguardando o próximo pedido de gravação (se há blocagem); caso contrário, após a inclusão, faz a gravação física.

Esta função é aplicada unicamente a arquivos de tamanho fixo.

#F0203 - Grave n caracteres a partir do apontador corrente no registro corrente

De forma análoga à função F0103, esta função grava n caracteres (bytes) a partir da posição corrente do apontador associado ao buffer vinculado ao arquivo especificado.

Ela é aplicada tanto a arquivos de tamanho fixo quanto variável.

ALGORITMO

INICIO

```

Se arq.pedido esta no DESAU
  então decodifica pedido;
    se não existe, buffer associado
      então pede buffer;
    se primeiro = 1 então escreve tag;
    se n° de caracteres > posições no buffer
      então
        transfere caracteres possíveis;
        se primeiro=1 então prende buffer;
          senão grava fisicamente buffer,
            pede buffer;
        transfere caracteres;
      senão
        retorna advertência;

```

FIM;

#F0204 - Grave Fim de Registro

Marca fim de registro para os dados gravados com a função 0203, escrevendo neste momento o tamanho do registro e fazendo a gravação física no caso de registros de tamanho variável (ilimitado).

INICIO

Toma buffer preso;
 Escreve tamanho registro;
 Faz primeiro:=1;
 Grava fisicamente o buffer;
 libera buffer;

FIM,

Abaixo mostramos uma tabela com referências cruzadas entre as funções do tipo 2 e as formas de organização e acesso de arquivos:

FUNÇÃO	SEQUENCIAL		RELATIVA		SEQ INDEXADA		
	SEQ	REL	SEQ	REL	SEQ	REL	SEQX
F0101	X	X	X	X	-	-	-
F0202	-	X	-	X	-	-	X
F0303	X	X	X	X	-	-	X
F0304	X	X	X	X	-	-	X

V.4. Funções da Categoria 03

As funções classificadas nesta categoria são aquelas cuja finalidade é a manipulação dos arquivos, quando não envolve qualquer ação de escrita e leitura física sobre os arquivos, mas tão somente sobre os diretórios.

#F0301 - Retorne "n" registros

Tem a finalidade de atualizar o contador de registros gravados/lidos, subtraindo do seu conteúdo o valor "n". Esta função aplica-se essencialmente à forma de acesso estritamente sequencial.

Se o valor de n for maior que o conteúdo do registro, a função equivale à F0303. Se "n" igual a zero, a função equivale a uma função de restaurar o apontador de posição interna ao registro. Seja o exemplo a baixo, representado esquematicamente na figura 27,

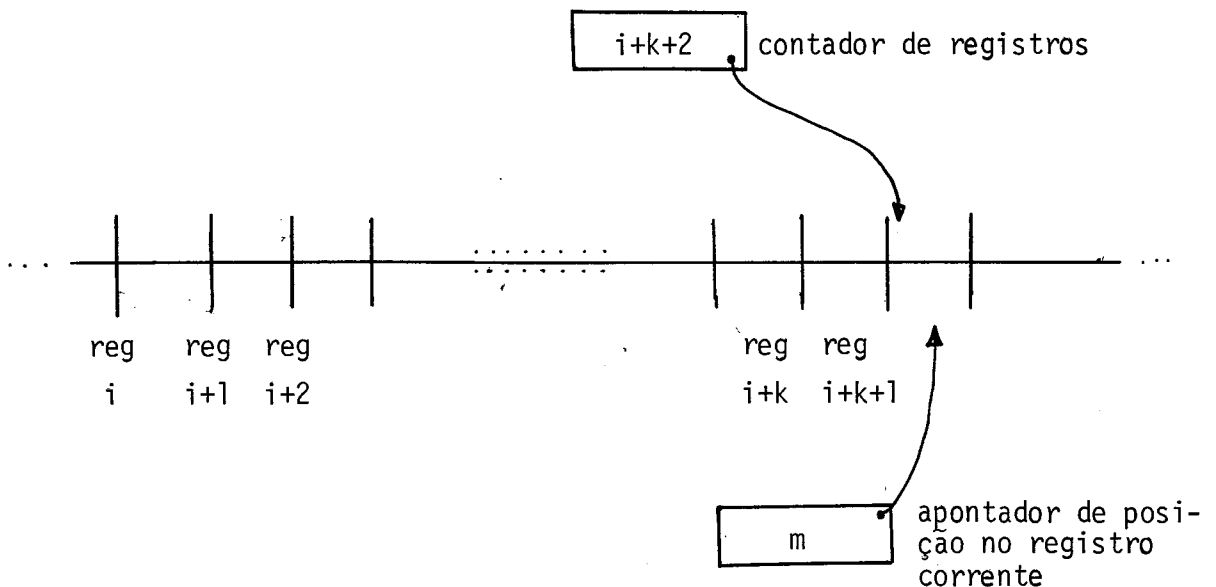


FIG.27. ESTADO DE UM ARQUIVO FICTICIO.

após comandos "retorne K registros" e "retorne 0 registros", temos as seguintes situações, respectivamente: vistas nas figuras 28 e 29 abaixo:

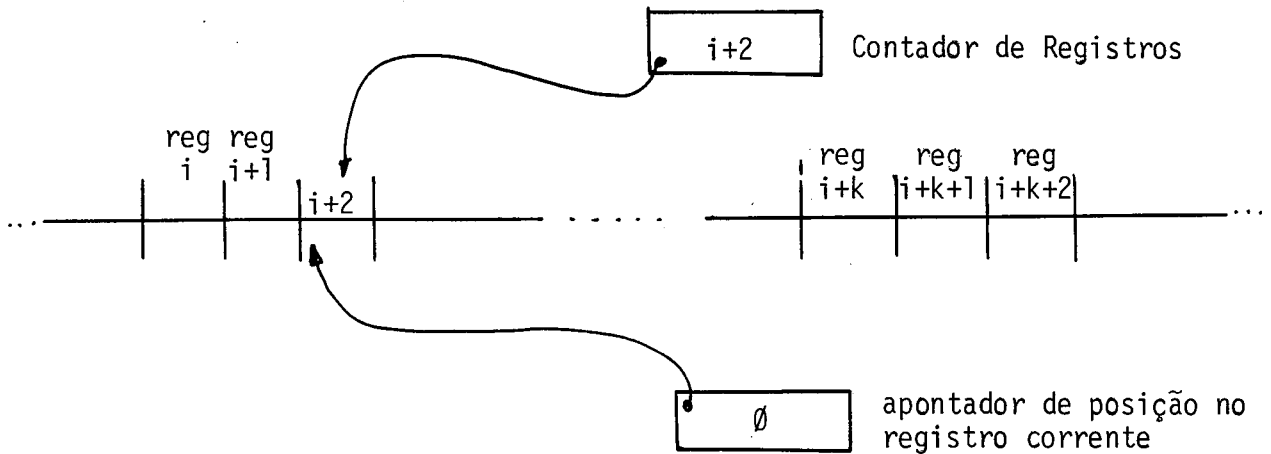


FIG.28. ESTADO DO ARQUIVO DA FIG.27 APÓS COMANDO "RETORNE K REGISTROS".

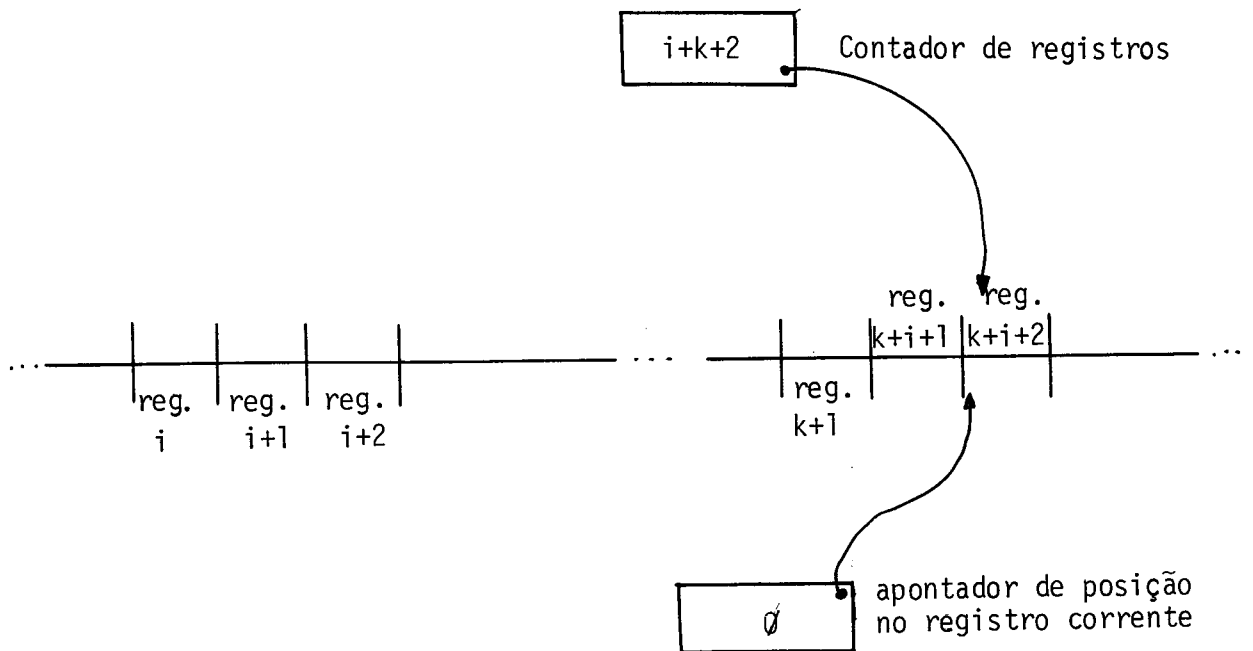


FIG.29. ESTADO DO ARQ. DA FIG.27 APÓS O COMANDO "RETORNE 0 REGISTRO".

INICIO

```

Se modo.acesso=sequencial
  então
    Se contador.registros  $\geq$  n
      então
        contador.registros:=contador.registros - n;
      senão
        contador.registros:=0;
        posição.corrente:=0;
    senão
      advertência;

```

FIM

#F0302 - Avance n registros (Forward)

Esta função é a inversa da F0301, isto é, adiciona o valor n ao conteúdo do registro contador de registros lidos/gravados do arquivo. As restrições quanto ao tamanho do registro e forma de acesso, são semelhantes às descritas na função F0301.

INICIO

```

Se modo.acesso=sequencial
  então
    Se contador.registros+n  $\leq$  n0 de registros
      então
        contador.registros:=contador.registros+n;
      senão
        contador.registros:=n0 de registros
        posição.corrente:=0
    senão
      advertência;

```

FIM

#F0303 - Volte ao início do arquivo

Posiciona o arquivo no seu primeiro registro, isto é, retorna ao início do arquivo, considerando o mesmo de acesso estritamente sequencial, com registros de tamanho fixo ou não.

INICIO

SE modo.acesso=sequencial

então

contador registros:=1;

posição.corrente:=0;

senão

FIM;

#F0304 - Inicialize Volume

Esta função é responsável pela inicialização de um volume qualquer de entrada e saída, isto é, faz a criação dos diversos arquivos de diretórios e faz uma visita global sobre as diversas trilhas e setores, detectando as defeituosas, anotando-as para evitar uso futuro.

Esta função será "stand-alone".

#F0305 - Habilite Volume

O objetivo desta função é a de tornar um determinado volume conhecido pelo sistema. Consiste em criar no DEVOH (Descritor de Volumes Habilitados) toda a descrição do volume, caso este ainda não esteja habilitado.

INICIO

EXISTE:=0

Para I=1 até N FazerSe VOL.PEDIDO = DEVOH VOL|I|então

EXISTE :=1;

INDICE:I

Se EXISTE ≠ 1então Pede buffer

Ler DIX # %Ler DIX no drive #

Se VOL.PEDIDO ≠ DIX.VOLentão

advertência;

Monta DEVOH;

Libera buffer;

FIM;

#F0306 - Abrir Arquivo (novo) Especificado

Esta função tem como objetivo criar um novo arquivo, isto é, alocar nos diretórios os seus atributos e criar sua descrição do DESAU (Descritor de arquivos em uso). Se já existir no diretório do usuário um arquivo com o mesmo nome, removerá este arquivo e no seu lugar abrirá um novo com o mesmo nome e com os novos atributos. É importante notar que a <extensão> faz parte do nome do arquivo que é:

<nó>.<volume>.<diretório>.<arquivo>.<extensão>

INICIO

EXISTE:=0

INDICE:=Λ

Para I:=0 atē N fazerSe NOME PEDIDO = BCES.NOME(I)Então

EXISTE:=1

INDICE:=I

Pedebuffer

Se EXISTE ≠ 1então

Ler DIX;

Se EXISTE ≠ 1então

Ler DIX;

Se Diretório existeentão

Ler DLB;

Se Arquivo.Extensão existeentão

Advertência;% não for aberta

senão

Monta DESAU|I|

senão

Criadiretório

grava DIX

Cria arquivo.extensão

grava DLB

Monta DESAU

senão

J:=0

Fazer

J:=J+1

atē DESAU.USUHAB(I,J)=ΛSe DESAU.USUHAB(I,J)=Λentão

DESAU.USUHAB:=CONDIGO.USUARIO;

DESAU.APTADLB:=POSICAO DLB

senão

advertência % já existe 2 usuários usando

% o arquivo pedido

Libera buffer,

FIM;

#F0307 - Habilite Arquivo

Esta função cria no DESAU a descrição de um ... arquivo pré-existente. Caso o arquivo pedido não exista, não é criado um novo, mas apenas retorna uma advertência indicando que o arquivo solicitado não existe no sistema.

O algoritmo desta função é semelhante ao da função anterior com a diferença que, tão logo seja detectada a não existência do diretório ou do arquivo, o usuário será advertido de que o arquivo solicitado não existe, e nenhuma outra operação será realizada.

#F0308 - Feche Arquivo Especificado

É a função responsável por gravar o último "buffer" de dados que está na memória, se houver, atualizar o diretório no DLB e desalocar a habilitação do arquivo no DESAU.

Quando o arquivo pedido para ser fechado estiver sendo usado por mais alguém, então, a função opera somente sobre os campos que habilitam o usuário solicitante a usar o arquivo, a fim de que no DESAU conste a informação de que a partir deste momento o mesmo não está mais habilitado para acesso pelo usuário.

INICIO

A:=ENDEREÇO BCES; I:=J:=EXISTE:=0;

B:=BCES,PROCESSO

Enquanto DESAU.MIN LOG ABR ≠ Λ fazer

I:=I+1;

Se A = DESAU END BCES

então EXISTE:=1

Se Existe buffer associado ao processo

então

grava fisicamente;

limpar buffer;

ler DLB |I|

atualiza DLB|I|

grava fisicamente

ler DCA |I|

atualizad DCA|I|;

grava fisicamente;

libera buffer;

Se DESAU.PROCHAB1|I| = B e DESAUPROCHAB2|I| = Λ

então

DESAU.X|I|:=0; % retira descrição do
% DESAU

senão

DESAU.PROC HAB1|I|:=DESAU.PROC HAB2|I|;

DESAU.PROCHDB2|I|:=0

Se Existe ≠ 1

então

Advertência % não existe arquivo aberto

FIM;

#F0309 - Fecha arquivos abertos pelo processo X

Tem a finalidade de auxiliar o gerente de processos, no sentido de que quando algum processo for suspenso, por qualquer motivo, se tenha condição de salvar todos os seus arquivos abertos e em uso até o momento de sua suspensão.

Na realidade, esta função recebe do núcleo o número do processo, com ele verifica no descritor DESAU se existe algum arquivo aberto associado ao mesmo; se existe, usa a função F0308 para fechar o arquivo, após isso, volta a procurar novos arquivos associados ao processo, e sempre que encontrar usa a função citada acima para fechar o arquivo.

Ao número do processo estão associados um ou mais BCES, assim há necessidade de se varrer todas as BCES para verificar quais as que estão associadas a ele; após esta pesquisa é que utilizamos o procedimento descrito acima.

V.5. Funções da Categoria 04

Estas funções permitem gerenciar os arquivos, possibilitando verificar; modificar e remover atributos dos diretórios e rótulos dos arquivos.

São 8 as funções classificadas nesta categoria a saber:

- Mostra Diretório especificado

- Mostre Rótulo especificado
- Remova arquivo especificado
- Altere atributos de arquivo
- Mostre arquivos existentes no volume
- Mostre BLA's usadas e disponíveis
- Mostre status corrente de arquivo especificado

É importante ressaltar que estas funções são um poderoso auxílio para a linguagem de comandos do computador, e que além disso, a visualização das informações passadas pelas funções, acima, tanto na sua forma de apresentação quanto na sua interpretação é atribuição específica da Linguagem de Comandos - Lico, ou do usuário que as utilizar.

#F0401 - Mostre Diretório Especificado

Esta função passa para o usuário todo um diretório(especificado), composto do conteúdo do diretório propriamente dito e o conteúdo de todos os rótulos de arquivos associados a este diretório.

Com o Nome do Volume e o Nome do Diretório associado a um determinado usuário, a função percorre toda a cadeia de informações ligadas ao diretório, passando-as ao solicitante.

INICIO

```

Se VOLPEDIDO estã no DEVOH
  entã I:=0;
    pede buffer;
    ler DIX|1|;
    pede buffer;
    ler DIX|2|;
    Enquanto DIX.NOMEDIR  $\neq$  DIRETORIO fazer
      I:=I + 1;
    Se Dix.REGSEGDIRE  $\neq$  & ENTÃO advertência;
    TRANSFERE DIX|I|;
    Libera Buffer
    A:=DIX APONTA DLB|I|
LABEL: K:=DECODIFICA DLB(A) % determina 1º rótulo
  Se k  $\neq$  z entã Ler DLB|K|;
  Se CODUSUARIO  $\leq$  DLB.IDN|K|
    E CODUSUARIO  $\geq$  DLB.IDN|K|
    entã %  $\bar{e}$  possível transferir
      transfere DLB|K|
      A:=DLB.APONTADOR
      Z:=K
      GOTO LABEL
    senã
      advertência; % Usuário não autorizado a
      % ver diretório.

```

FIM;

A função DECOFICADLB(X) toma X que \bar{e} a ordem de localização do rótulo do arquivo no DLB e determina qual o setor em que ela se encontra.

#F0402 - Mostre Rótulo Especificado

É uma função semelhante à anterior, com a diferença de que só transfere ao usuário o rótulo especificado.

#F0403 - Remova Arquivo Especificado

Esta função age tanto sobre os descritores como sobre os diretórios. Quando o arquivo é especificado (caso contrário, temos uma condição de erro) a função verifica se o mesmo está em uso, se tiver; remove sua entrada do DESAU; posteriormente retira do DLB o rótulo correspondente ao arquivo, e atualiza o diretório DIX, no seu campo DIX.APONTADBLIVRE, uma vez que este campo aponta sempre as duas primeiras áreas livres no DLB.

INICIO

I:=0; Se ARQUIVO está no DESAU então Remove DESAU[ARQUIVO] ;

Pede Buffer;

Ler DIX[1];

Pede Buffer;

Ler DIX2;

Enquanto DIX.NOMEDIR≠DIRETÓRIO fazer

I:=I+1

Libera buffer que não contém DIX[I];

A:=DIX APONTADLB [I];

K:=DECODIFICADLB [A];


```

Se K≠Z então Pede buffer;
                Ler DLB[K];
Se DLB.NOMEARQ[K] = ARQUIVO
então
DIX.APONTADLB[I] = DLB.APONTADIR[K]
Remove DLB K
senão
A: DLB.APONTADIR|K|
Z:=K
K:=DECODIFICADLB|A|
Se K≠Z então Pede buffer
                Ler DLB[K]
Se DLB.NOMEARQ|K| = ARQUIVO
então
                DLB.APONTADOR Z :=DLBAPONTADIR[K]
                Remove DLB[K]

```

FIM;

#F0404 - Altera Atributo de Arquivo

Esta função age tão somente sobre os diretórios. Os arquivos que estiverem em uso, não podem ter seus atributos modificados, mas apenas quando forem salvos (ou fechados).

Os atributos modificáveis são somente aqueles existentes no DLB e no DIX, mas não todos. São os seguintes:

No DLB:

- 1 - Nome do arquivo;
- 2 - Limite inferior do intervalo do número de usuários habilitados;
- 3 - Limite superior do intervalo do número de usuários habilitados;
- 4 - Expiração;
- 5 - Incremento BLA;
- 6 - Tamanho da chave.

No DIX:

- 7 - Registro de Segurança

Todos os demais atributos não podem ser modificados, sob pena de causar danos e incertezas quando do tratamento futuro dos arquivos associados ao diretório e mesmo no manuseio do arquivo cujo rótulo foi alterado.

#F0405 - Mostra arquivos existentes no Volume Especificado

Consiste em ler todo o DLB e passá-lo ao usuário, ao qual compete interpretá-lo

INICIO

Pede buffer;

Fazer

Ler DLB;

Transfere DLB;

atê fim

Libera Buffer;

FIM;

#F0406 - Mostre BLA's usadas e disponíveis

É uma função, tão simples quanto a anterior, que consiste em ler o diretório DCA e transferi-lo ao usuário.

#F0407 - Mostre Estado Atual do Arquivo Especificado.

Esta função transfere para o usuário as seguintes informações:

- a - entrada no descritor DESAU;
- b - Estado do arquivo e buffers a ele associados.

A "entrada no descritor DESAU" tem todas as informações nele contidas associadas ao arquivo especificado.

V.6. Descrições dos Atributos para chamar as funções do SGMES

O objetivo aqui é descrever de uma forma geral as informações que o usuário deve alocar na BCES a cada solicitação ao SGMES. Vale ressaltar que para cada arquivo a ser utilizado pelo usuário, além das áreas de trabalho necessárias para manipulá-lo, o usuário deve reservar uma área de 120 bytes, que será o BCES associado ao arquivo. Essa área deve ser utilizada especificamente com a finalidade de comunicação e controle de entradas e saídas. Qualquer alteração inadequada no BCES pode gerar os problemas mais diversos possíveis desde a não realização das funções pedidas até o cancelamento do programa.

Ressalva-se que, esta área, é de inteira responsabilidade do usuário em criá-la e protegê-la.

A seguir apresentamos uma tabela que indica para cada função, das diversas categorias, quais as informações que precisam ser colocadas na parte variável do BCES (ver estrutura do BCES no apêndice 3). Lembramos que dos campos de informações da parte fixa de BCES, alguns são de responsabilidade do usuário e outros do sistema operacional; os primeiros são:

- 1 - identificação do veículo
 - 1.1. - N° (para uso em redes de teleprocessamento);
 - 1.2. - Nome do Volume;
 - 1.3. - Nome do Diretório;
 - 1.4. - Nome do Arquivo;
 - 1.5. - Nome Extensão;
- 2 - Código da operação ou função a ser efetuada;
- 3 - Registro de Segurança e Acesso;
- 4 - Endereço do Buffer do Usuário;
- 5 - Modo de acesso;

Na tabela abaixo está explicitada quando devem ser colocados na BCES esses atributos.

Função	Descrição dos parâmetros da BCES parte variável (OBS: a parte fixa deve ser atualizada se necessário)
Categoria 1	
F0101	
F0102	Se acesso direto relativo: <ul style="list-style-type: none"> a - Número do registro dentro do arquivo; Se acesso sequencial indexado: <ul style="list-style-type: none"> a - Tamanho da Chave; b - Chave;

F0103

Se acesso sequencial:
 a - número de caracteres;
 Se acesso direto relativo:
 a - número de registro;
 b - número de caracteres;
 Se acesso sequencial indexado:
 a - tamanho da chave;
 b - chave;
 c - número de caracteres;

F0104

Se acesso sequencial: <nihil>
 Se acesso direto relativo:
 a - número do registro;
 Se acesso sequencial indexado:
 a - tamanho chave;
 b - chave

Categoria 02

F0201

F0202

Se acesso sequencial: <nihil>
 Se acesso direto relativo:
 a - número do registro
 Se acesso sequencial indexado:
 a - tamanho chave
 b - chave

F0203

Se acesso sequencial:
 a - número de caracteres;
 Se acesso direto relativo:
 a - número do registro;
 b - número de caracteres;
 Se acesso sequencial indexado:
 a - tamanho chave;
 b - chave;
 c - número de caracteres;

F0204 Se acesso sequencial: <nihil>

Se acesso direto relativo:

- número do registro;

Se acesso sequencial indexado:

- tamanho chave
chave

Categoria 3

F0301 - número de registros

F0302 - número de registros

F0303

F0304 - Função "stand-alone"

F0305 - Não;
Volume;
Unidade ("drive").

F0306 - Diretório;
Arquivo;
Registro de Segurança e Acesso
Extensão;
Número de BLAS a serem alocadas ao arquivo
Número de BLAS para tabela de índices (se arq.seq.indexado)

F0307 - Diretório;
- Arquivo;
- Extensão;

F0308 - Diretório;
Arquivo;
Extensão;

F0309 - Número do processo;

Categoria 4

As funções desta categoria não operam com os descritores apenas com os diretórios dos veículos, com da F0412

F0401

- Nō; (opcional)
- Volume;
- Diretório;
- Unidade (opcional);

F0402

- Nō (opcional);
- Volume;
- Diretório;
- Arquivo;
- Extensão;
- Unidade (opcional);

³
F0403

- Nō (opcional);
- Volume;
- Diretório;
- Arquivo;
- Extensão;
- Unidade (opcional)

⁴
F0404

- Nō (opcional);
- Volume;
- Diretório;
- Arquivo;
- Extensão;
- Unidade;
- Código do atributo;
- Nova denominação ou valor do atributo;

⁵
F0405

- Nō (opcional);
- Volume;
- Unidade (opcional);

06
F0416

- Nō (opcional);
- Volume;
- Unidade;

07
F0417

- Nō;
- Volume;
- Diretório;
- Arquivo;
- Extensão;

CAPÍTULO VICONCLUSÕESVI.1. Geral

O gerenciamento de arquivos é dentro de um sistema operacional um ponto tão importante que na avaliação de desempenho de um equipamento, como por exemplo para selecionar computadores, é visto como o segundo item de maior peso, ^[46],^[47]. Isto é devido ao fato que a manipulação de arquivos tem uma função disciplinadora no desenvolvimento de software que muito influi no desempenho dos programas, principalmente daqueles que necessitam de um maior volume de entradas/saídas e de facilidades de recuperação e manutenção das informações.

O SGMES caracteriza-se por adotar alguns compromissos, tendo em vista seu uso futuro ou o mercado que se pretende atingir. Entre esses compromissos destacam-se o de simplicidade, segurança e adaptabilidade. Isto tudo é graças a um sistema operacional bastante modular de um lado, e do outro à existência de interface inteligente para os discos, que faz com que o SGMES seja transparente ao tipo de unidades usadas.

É importante salientar, que a formatação livre dos registros permitida pelo SGMES possibilita ao usuário a utilização total (nas formas de organização sequencial e direta) do espaço em disco disponível a ele, o que não acontece com

gerenciadores que se utilizam de uma formatação tipo IBM, no qual o aproveitamento é de 65% aproximadamente, tanto usando discos flexíveis de densidade simples ou dupla.

As funções básicas de um gerente, de uma forma bastante ampla, consistem em: a) Métodos de acesso a arquivos; e b) Serviços de manutenção de arquivos. As funções tipo a), consistem em primitivas que possibilitam o armazenamento e recuperação de dados dos arquivos, da forma mais simples possível, dentro de cada filosofia de organização adotada; as do segundo tipo, tem a finalidade de suprimir, modificar atributos, remover, registros e arquivos, reorganizar arquivos, etc. Devido à característica de adaptabilidade e simplicidade, foram implementadas no SGMES somente as essenciais dentre estas últimas. Aquelas que por um motivo ou outro, podiam ser transformadas em utilitários (as de reorganização de arquivos, por exemplo), assim foram consideradas.

A característica do sistema operacional de permitir uso de redes de teleprocessamento induziu no projeto do SGMES a possibilidade de se ter acesso a arquivos de qualquer ponto de uma rede, desde que o mesmo esteja habilitado. Com isso, uma das três classes de problemas de manipulação de arquivos em rede definidas em [51] (identificação global de arquivos) está resolvida; as demais, "representação dos dados dentro do arquivo" que consiste em definir uma organização padrão de arquivos é resolvida pela implementação de três formas padrões de organização no SGMES, onde a sequencial é tomada como "default"; "mecanismo de transferência de dados" que é constituído de operações de forma "padrão" ou "transparente"; e

de "conversão de dados na transmissão", que não são do escopo do SGMES, mas ~~são plenamente resolvidos~~ pelo módulo de rede que faz parte do sistema operacional do qual o SGMES também é parte.

Devido à forma de controle de espaço em disco ser feita em blocos de tamanho prefixado pelo usuário, dependendo do tamanho do bloco adotado, podem ser ligados discos (até 4 por canal) com capacidade variável:

Tamanho Bloco	Capacidade Disco
4 setores	1Mbyte
8 setores	2Mbytes
16 setores	4Mbytes
20 setores	5Mbytes

FIG.31. RELAÇÃO CAPACIDADE DISCO POR TAMANHO BLA.

Como é possível ter até quatro canais para disco, se usarmos um bloco de tamanho igual a 16 setores de 256 bytes, e 4 discos por canal, o SGMES pode controlar até 62MBytes sem nenhuma modificação.

VI.2. Comparação com outros sistemas de gerenciamento de arquivos

É difícil fazer uma comparação de sistemas de gerenciamento de arquivos de uma forma isolada sem levar em consideração o ambiente operacional em que está inserido. No entanto, determinadas características podem ser comparadas, tais

como os modos de acesso disponíveis, formas de organização de arquivos, etc.

Se tomarmos como referência do sistema operacional CP/M da Digital Research, em seu módulo FDOS o qual trata as primitivas encarregadas de realizar o acesso às unidades de disco e da construção dinâmica de arquivos, entre outras funções, vemos que todas essas funções também existem no SGMES, que além disso dá suporte a uma organização de arquivo que não é permitida no CP/M, que é a sequencial indexada, sem falar do modo de leitura e escrita byte a byte (caractere a caractere). O mesmo acontece com relação à gerência de arquivos do MP/M (que é um sistema operacional multiprogramado, semelhante ao daquele que o SGMES faz parte), que é uma extensão do CP/M, com o aumento da capacidade de armazenamento em disco, chegando a um máximo de 32MB quando em uso as 16 unidades em disco, o mesmo a que chega o SGMES.

Com relação a outros gerente de arquivos, tais como do recém lançado micro-computador da IBM -Worm, que é uma versão "scale-down" do UNIX, que tem funções semelhante ao CP/M e é limitado a gerir no máximo quatro unidades de discos flexíveis de $5 \frac{1}{4}$ " , sendo que a versão atual permite só duas unidades com 160KBytes cada, e do microcomputador COBRA TD200 ou TD300 que só permite o uso de quatro unidades e aceita modos de acesso sequencial, relativo indexado, o SGMES é um gerente muito mais flexível e poderoso.

A seguir apresentamos um quadro comparativo entre diversos sistemas operacionais e o sistema operacional do qual, faz parte o SGMES dando ênfase evidentemente aos aspec-

GERENTE DE DADOS

ORGANIZAÇÃO ARQUIVOS:

1 Sequencial	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim
2 Direta	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim
<u>3 Seq.Indexada</u>	Sim	Não	Não	-	Não	Sim	Não	Não
<u>Blocagem</u>	Sim	Não	Não	Não	Não	Não	Não	Não
<u>Intercâmbio(Swapping)</u>	Sim	Não	Não	Sim	Não	Sim	Não	Não
Segmentação de Arquivos	Sim	Não	Sim	Sim	Não	Sim	Não	Não
Capacidade Máxima em Discos	64MB	2MG	-	-	2MB OU 32MB (MP/M)	-	-	-
Diretórios Hierárquicos	Sim	Não	Sim	Sim	Não	Não	Não	Não
Acessos								
Sequencial (SAM)	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim
<u>Sequencial Caractere a Caracteres</u>	Sim	Não	Não	Não	Não	Não	Não	Não
Direto	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Sequencial Indexado	Sim	Sim	Não	-	Não	Sim	Não	Não

MP = multi-processos

TR - tempo real

- (não se tem informação)

FIG.32. QUADRO COMPARATIVO DE SISTEMAS OPERACIONAIS.

Como vemos, as características do Sistema Operacional do qual faz parte o SGMES e deste propriamente dito, são bastante avançadas e como um todo, é um sistema mais completo que qualquer um dos existentes, inclusive pelos tipos de soluções dadas a essas características.

O SGMES apresenta facilidades que nenhum dos sistemas de gerência de dados dos sistemas operacionais, citados na tabela anterior tais como alteração de muitos dos atributos de arquivos, estrutura dos diretórios ligada, escrita e leitura parcial de registros (com acesso de caractere-a-caractere ou conjunto de caracteres), controle dos arquivos por associa-

ção a processo, implementando com isso a possibilidade de mais de um usuário utilizar um mesmo arquivo simultaneamente (exclusão mútua é realizado via uma estrutura monitor implementada no kernel do sistema operacional).

Funções	MP/P	SGMES	CP/M	SOM
Restaura Sistema de Disco	X	-	X	-
Selecione Unidade	X	X	X	X
Crie arquivo	X	X	X	X
Abre arquivo	X	X	X	X
Feche arquivo	X	X	X	X
Mostre Diretório	X	X	-	-
Destrua arquivo	X	X	X	X
Altera nome arquivo	X	X	X	X
Ler Sequencial	X	X	X	X
Ler Direto	X	X	X	X
Escreve Seq	X	X	X	X
Escreve Direto	X	X	X	X
Mostre Unidades Disponíveis	X	-	X	X
Restaura Unidade	X	X	-	X
Mostre Indicadores Arquivo	X	X	X	-
Ler Seq.Indexado	-	X	-	X(*)
Leitura Parcial	-	X	-	-
Escreve Parcial	-	X	-	-
Escreve Seq.Indexado	-	X	-	X(*)
Bloqueie	-	X	-	-
Modifique atributos arquivos	-	X	-	-
Retorne n registros	-	X	-	-
Avance n registros	-	X	-	-
Retorne início registro	-	X	-	-
Feche arquivos processo X	-	X	-	-
Arquivos existentes volume V	X	X	X	X
Mostra Áreas disponíveis no disco	X	X	X	X

(*) de difícil uso e indicação de tabela cheia com a utilização de poucas chaves

FIG.33. FUNÇÕES DE MANIPULAÇÃO DE DISCO DE DIVERSOS SISTEMAS OPERACIONAIS.

VI.3. Sugestões para desenvolvimentos futuros

Dadas as características do SGMES, dentro da arquitetura atual do microcomputador para que foi desenvolvido pouco há a ser acrescentado. No entanto há toda uma área que pode ser desenvolvida para, dotá-lo de maior portabilidade que o projeto atual.

Uma outra facilidade que poderia ser desenvolvida e acoplada ao SGMES, é a que permite que múltiplos usuários possam usar (hoje é possível até dois, apenas) arquivos criando condições para evitar bloqueio perpétuo a nível do próprio SGMES. Isso evitaria o uso da estrutura de monitor implementado no núcleo do sistema operacional, possibilitando com isso uma maior portabilidade e segurança do SGMES, quando implementado em máquinas que não as tenham implementadas.

As formas de organização de arquivo poderão também ser ampliadas de modo a permitir uso de estruturas ligadas (listas) e estruturas em árvore, implementando os respectivos modos de acesso.

Para um ambiente de multiprocessamento, o SGMES, pouco precisa ser alterado, a não ser a inclusão de maiores informações nos diretórios e descritores (que estão na memória, os últimos), que possibilite a identificação do processador para evitar que processos iguais rodando em processadores diferentes tenham acesso a arquivos de forma alterada.

Condições de maior segurança de arquivos podem também ser desenvolvidas, principalmente para aquelas aplicações e usuários que necessitem proteger os conteúdos das in-

formações contra alterações danosas e criminosas, através de processos que permitam todo um sistema de codificação e transcodificação, estática ou dinâmica, de criptografias.

BIBLIOGRAFIA

1. REDELL, David D. - PILOT - An Operating System for a Personal Computer. Comm of ACM , 23(2) - fev.1980.
2. Holt, R.C. e outros - Structured Concurrent Programming with Operating System Application, Massachusetts, Addison Wesley Pub.Co. - 1978.
3. Brinch, Hansen - The Architecture of Concurrent Programs, Prentice-Hall, 1965.
4. While, George M. - Using Interrupts on the Apple II System. BYTE , maio 1981.
5. Shave, M.J.R. - Data Structures , McGraw-Hill, 1975.
6. Babad, Yair M. e Hoffer, Jeffrey A. - Data Element Security and Its Effects on File Segmentation, IEEE Transaction on Software Engineering - vol SE-6, set. 1980.
7. Howard, John H. - Proving Monitors - Comm. of ACM - 19(5) may 1976.
8. Hansen, Brinch - Operating System Principles . Prentice Hall, Inc. Englewood, New Jersey, 1973.
9. Codell, R.A. e Coffman, E.J. - Record Allocation for minimizing Expected Retrieval Cost on Drum-Like Storage Devices. Journal of ACM 25(1):103-105, jan.1976.

10. Lampson, Butter W. e Redell, David D. - Experience with Process and Monitors in MESA. Comm of ACM , 23(2), fev. 1980.
11. Cichelli, Richard J. - Minimal Perfect Hash Functions Made Simple - Con. of ACM , 23(1), jan.1980.
12. Fritzson, Richard - Data on Disk - Implementing File System - Micro computing , jan.1981.
13. Macedo, R.M. Juckowsky - Software para controle de Discos Flexíveis . Tese M.Sc. UFRJ/COPPE, 1979.
14. COBRA - Manual de Referência do Sistema Operacional Mono-programável - SOM - COBRA 300/TD . maio 1980.
15. Richie, Denis, M. e Thompson, Keni - The UNIX Time-Sharing System - Comm. of ACM , 17(7), july 1974.
16. Fuller, Samuel H. - Minimal - Total Processing Time Drum and Disk Scheduling Disciplines. Comm ACM , 17(7), july 1974.
17. Laventhal, Lance A. - 6502 - Assembly Language Programming - Osborne/McGraw-Hill, 1979.
18. Seanlon, Leo J. - 6502 Software Design - Howard W. Sams & Co. Inc - 1980.
19. Breitiger, J.L. - Tese de Mestrado em Engenharia de Sistemas - (a ser defendida).
20. Guimarães, Pedro L.M.- LICO - Uma Linguagem de Comandos para minicomputador - Tese de Mestrado Sistema/COPPE (a ser defendida).

21. Maurer, Hermann A. - Data Structures and Programming Techniques . Prentice-Hall, 1977.
22. Larson, James - The INS and OUTS of PC/M. BYTE , june 1981.
23. Peatman, John B. - Microcomputer - Based Design , Int. Student Edition, McGraw-Hill/Kogakusha Ltd., 1977.
24. Nicholson, James e Camp, Roger - Build a Super Simple Floppy - Disk Interface - BYTE , june 1981.
25. Shen, Kenneth K. e Peterson, James L. - A weighted Buddy Method for Dynamic Storage Allocation - Comm.ACM , 17(10) od 1974.
26. Organick, Elliott I. - Organização de Sistemas de Computação - Série B5700/B6700 - Bib. Pioneira de Adm. e Negócios, 1975.
27. Averbach, The Function of an Operating System - Relatório 13-02-01 Computer Programming Management - Auerbach Publishers Inc - 1975.
28. Brown, M.P. - Técnica de Pesquisa em tabelas . Edgard Blucher Ed., 1979.
29. Garnett, N.H. e Needhan, R.M. - An Asynchronous Garbage Colletor for the Cambridge File Server. Operatins System Review 14(4):36-40, oct.1980.
30. Dion, Jeremy - The Cambridge File Server. Operating System Review , 14(4):26:35, oct.1980.
31. Scholl, Michel - Performance Analysis of new file organizations base on dynamic Hash-Coding. Laboratoire de Recherche en informatique et automatique do IRIA,Repport de Recherche nº 347, mars 1979.

32. Bauer, F.L. e outros - Language Hierarchies and Interface - Lectures Notes in Computer Sciences Series - Spring-Verlag Berlim, 1976.
33. Brent, Richard P. - Reducing the Retrieval Time of Scatter Storage Techniques. Comm of the ACM , 16(2):105-109, fev. 1973.
34. Lum, V.Y., Yuen, P.S.T. e Dodd, M. - Key-to-Address Transform Techniques: A fundamental Performance Study on Large Existing Formatted Files. Comm of the ACM , 14(4): 228-239, april 1971.
35. Scholl, Michel - Analysis of two hash-coded file organization policies under steady-state conditions. Laboratoire de Recherche en Informatique et automatique do IRIA. Rapport de Recherche n° 354, June 1979.
36. Flynn, M.J., Gray, J.N. e outros - Operating Systems - An Advanced Course. Editado por Bayer, R. e outros - Lectures Notes in Computer Science Series - Berlim, Spring-Verlag, 1978.
37. Freeman, Donald E. e Perry, Olney R. - I/O Design: Data Management in Operatins Systems . New Jersey, Hayden Book Co.Inc., 1977.
38. Souza, Jano M. - Algoritmos de Hashing para problemas específicos. Tese de Mestrado, COPPE/UFRJ, fev.1978.

39. Sipala, Paolo - Optimum Cell Size for Storage Message.
IEEE Trans. on Software Engineering - SE-7(1) - Jan.1981.
40. Lefkovitz, David - File Structures for on-line Systems,
New Jersey, Hayden Book Company Inc., 1978.
41. Gudes, Ehud - The Desing of a Cryptography Based Secure
File System. IEEE Trans. on Software Engineering , SE-6
(5), set.1980.
42. Hansen, David R. - A Portable File Directory System,
Software - Praticce and Experience , 10(11):623-634, 1980.
43. Smith, Alan Jay - Analysis of Long Term File Reference
Patterns for Application to file Migration Algorithms,
IEEE Trans. of Software Engineering , SE-7(4), july 1981.
44. Lazowska, Edward D. e Zahorjan, John - Analytic Modelling
of Disk I/O Subsystems: A tutorial, ACM/SIGMETRICS
proceedings Conference on Measurements and Modeling of
Computer System, pp.33-35, set.1981.
45. Artis, H.P. - A Log File Design for Analysing Secondary
Storage Occupancy - ACM/SIGMETRIC Prof. of Conference on
Measurement and Modeling of Computer System, pp.129-135,
set.1981.
46. Teixeira, Sergio R.P., - Como Selecionar e Utilizar Mini-
computadores , Compucenter do Brasil Ltda, 1981.
47. Oliver, Robin T., "A Technique for Selecting Smal
Computer, Datamation , pp.93-97, jan.1970.

48. Severance, Dennis Gordon, Some Generalized Modelling Structures for Use in Design of File Organizations. Tese de Doutorado - University of Michigan, 1972.
49. Schlabit, F.F. e Corso, Thadeu B., Um Sistema Multiprogramado para minicomputadores - Tese de Mestrado - UFRS - abril 1978.
50. Wolochow, Peter I. - "Micro Computer Operating System Trends, Digital Design , 11(12):51-56, Dec.1981.
51. Kirner, Claudio, Computação Distribuída: Proposta de um Sistema com Facilidades de Manipulação de arquivos dispersos numa rede - Exame de Qualificação para Doutorado em Engenharia de Sistemas - (não publicado).
52. Digital Research - CP/M 2.2 Interface Guide - 1979.
53. Digital Research - MP/M Multi-Programming Monitor Control Program - User Guide - 1981.

APENDICE 1

Condições de Erro no SGMES

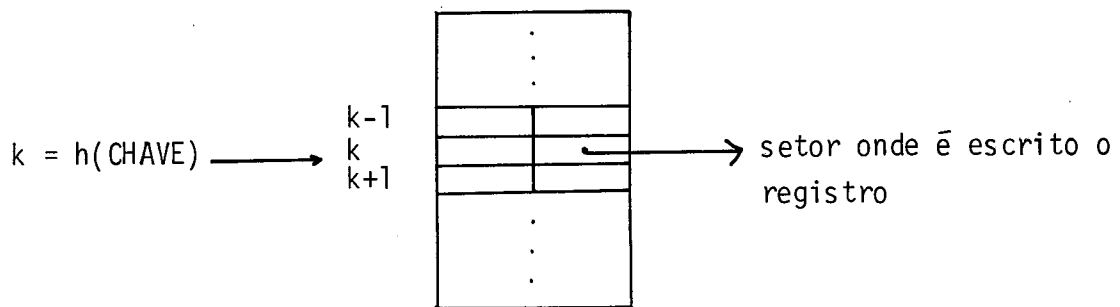
CONDIÇÕES DE ERRO DO SGMES

<u>CÓDIGO</u>	<u>DESCRIÇÃO</u>	<u>STATUS</u>
01	UNIDADE NÃO EXISTENTE	RECUPERÁVEL
02	ARQUIVO NÃO EXISTENTE	RECUPERÁVEL
03	DIRETÓRIO NÃO EXISTENTE	RECUPERÁVEL
04	VOLUME NÃO EXISTENTE	RECUPERÁVEL
05	EXTENSÃO DESCONHECIDA	RECUPERÁVEL
06	LIMITE SEGMENTAÇÃO ESGOTADA	FATAL
07	ARQUIVO NÃO ABERTO	RECUPERÁVEL
08	VOLUME NÃO HABILITADO	RECUPERÁVEL
09	CÓDIGO DESCONHECIDO	RECUPERÁVEL
10	CHAVE INEXISTENTE	RECUPERÁVEL
11	DIRETÓRIO PRIVADO	FATAL
12	ARQUIVO PRIVADO	RECUPERÁVEL
13	TABELA CHEIA	FATAL
14	MAIS DE 7 COLISÕES PARA A CHAVE	RECUPERÁVEL
15	REGISTRO PEDIDO NÃO EXISTENTE (Tentativa de acessar área não autorizada)	FATAL
16	NÃO HÁ MAIS ESPAÇO NO DISCO	FATAL
17	NÚMERO DE BLA'S ALOCADAS MENOR QUE A SOLICITADA	TRANSPARENTE
18	ESCRITA NÃO PERMITIDA	FATAL
19	EXCEDEU NÚMERO DE PROCESSOS USANDO MESMO ARQUIVO	FATAL
20	NÃO HÁ BUFFER DISPONÍVEL	RECUPERÁVEL
21	REGISTRO COM TAMANHO MAIOR QUE 65535 bytes	FATAL
22	ERRO DE ESCRITA APÓS n TENTATIVAS	FATAL
23	ERRO DE LEITURA APÓS n TENTATIVAS	FATAL

APÊNDICE 2

DESCRIÇÃO DA ORGANIZAÇÃO ESPECIAL
DE ARQUIVOS

O objetivo deste apêndice é fazer uma descrição de talhada do modo de organização especial de arquivos que é adotado pelo SGMES. Tal modo de organização compreende um esquema de acesso de forma sequencial indexada ou relativa indexada. A indexação é feita via uma função hash, cujo esquema simplificado seria:

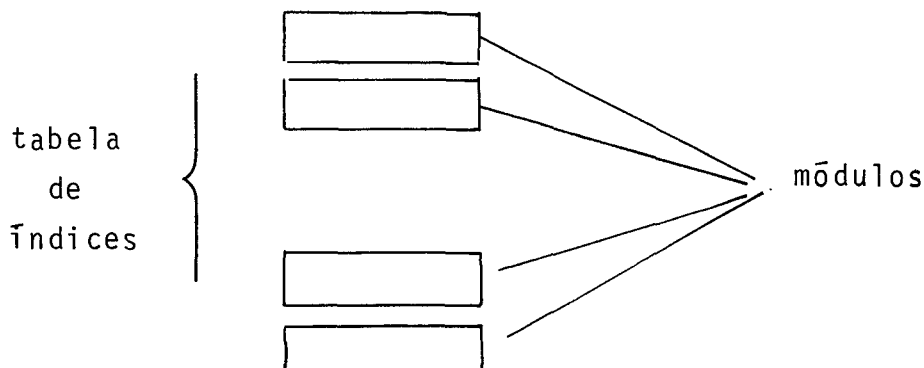


Os registros a que esse tipo de organização dá suporte, podem tanto ser de tamanho fixo como de tamanho variável.

O arquivo gerado na forma especial é logicamente separado em duas partes: uma onde ficará armazenada a tabela de índices e outra onde ficarão todos os registros.

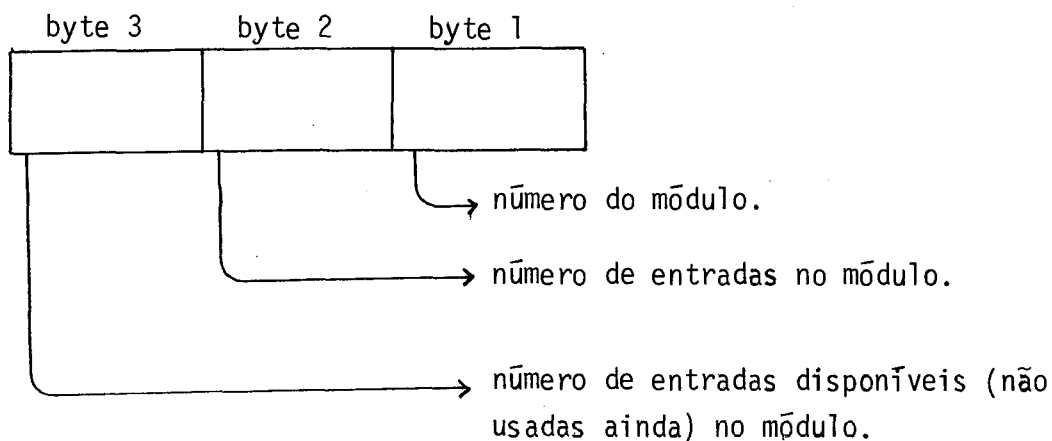
Devido a problemas de limitação de memória, este tipo de organização poderia não ser possível na prática, dado o tamanho da tabela de índices. Com o intuito de contornar tal dificuldade, adotamos uma solução nova no tratamento da tabela de índices, apesar de tal tratamento diminuir o desempenho do sistema, quando a tabela de índices cresce muito, devido a velocidade de acesso dos veículos usados correntemente.

Apesar da área de memória que ficará disponível para armazenamento da tabela de índices ser de 1Kbyte, a solução adotada, prevê uma tabela de índices de tamanho ilimitado, em teoria, subdividida em módulos de 256 bytes:



A quantidade de BLA's máximo que pode ser alocada para a tabela de Índices, na primeira versão do SGMES, será trinta e dois. A alocação de espaço em disco para a área de registros deve ser indicada; caso contrário, serão alocadas no mínimo, 32 BLA's.

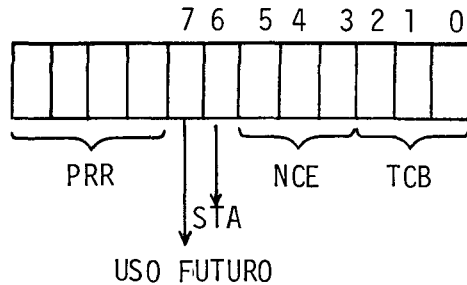
Cada módulo da tabela de Índices terá um número variável de entradas, dependendo do tamanho máximo da chave a ser usada para acessar o arquivo. Em cada módulo haverá no mínimo, um descritor de 3 bytes, ocupando as últimas posições do mesmo onde serão armazenadas as seguintes informações:



Cada chamada ou entrada na tabela de Índices conterá as seguintes informações:

1. KEY Chave _____ até 7 bytes

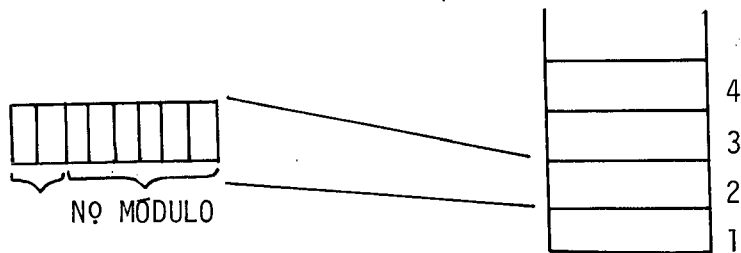
2. PTC Apontador para a entrada colidida, isto é, o número da entrada onde está armazenada a chave colidida - 12 bits
3. Controle - 12 bits



- 3.1. TCB - Tamanho, em bytes, da chave corrente. Deve ter no mínimo um byte. Ocupa 3 bits, assim, suporta no máximo 7 bytes de tamanho.
- 3.2. NCE - Número de colisões havidas nessa entrada. Ocupa 3 bits. Só será permitido um máximo de 7 colisões por lugar.
- 3.3. STA - Bit indicador, que informa se a chave ocupante é dona (bit ativado) ou não (bit restaurado) do lugar.
- 3.4. PRR - Posição relativa do registro dentro do setor apontado por NST

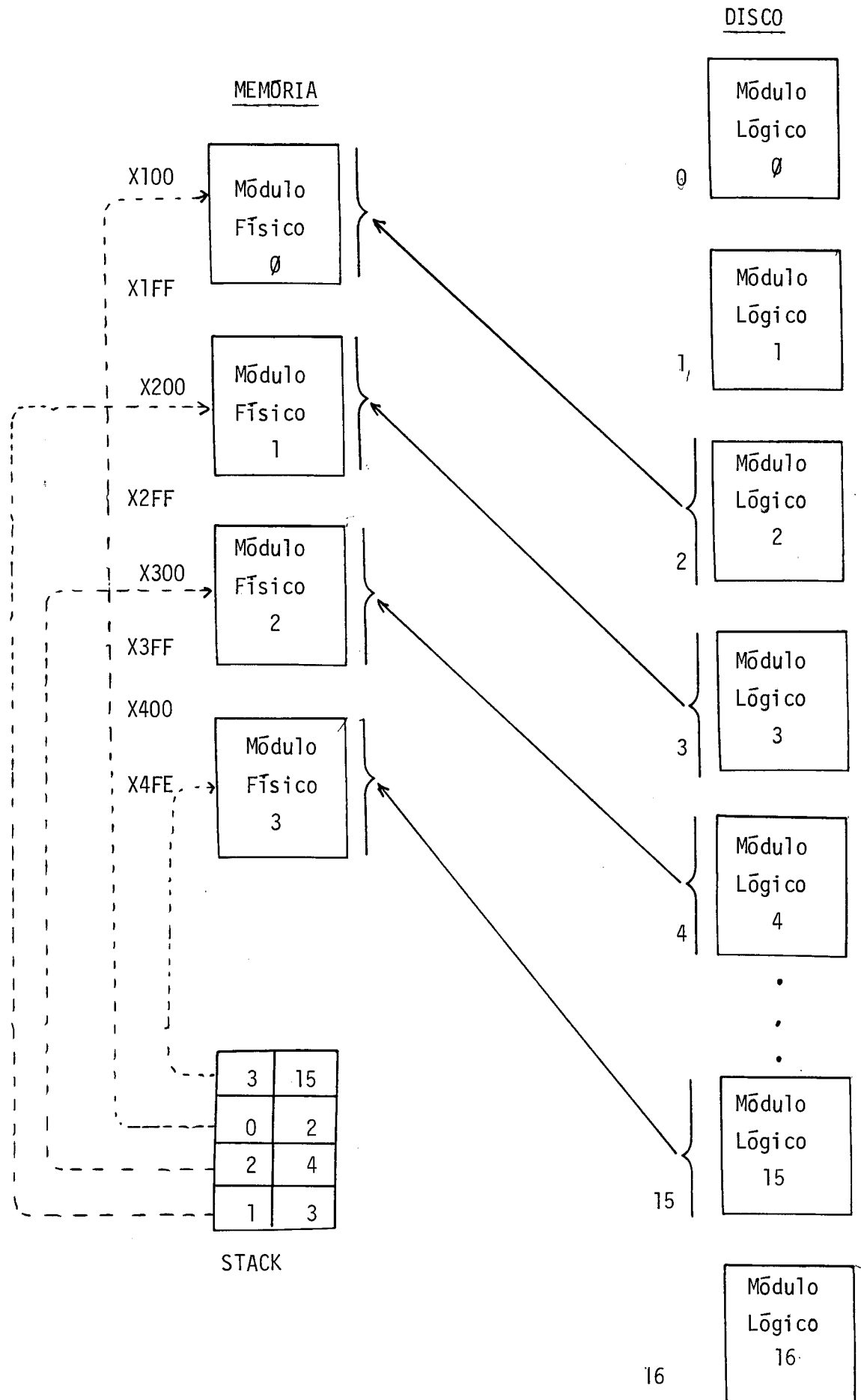
4. PTR - Apontador. Aponta para o número do setor no qual está armazenado o registro ___ 2 bytes

Quando da criação (abertura) ou habilitação de um arquivo que tenha organização sequencial indexada, os módulos contidos nos quatro primeiros setores do disco que contêm a tabela de índice, serão armazenados na memória. Ao ser feita uma entrada de chave, e se a aplicação da função hash sobre ela indicar uma entrada que não esteja nos módulos residentes na memória, então será realizada uma substituição (semelhante ao esquema adotado na memória virtual com paginação) do módulo menos recentemente usado (método LRU) por aquele cuja a entrada pedida esteja presente. Para a implementação desse esquema, será mantido na memória um stack ou pilha que armazena quatro entradas no máximo; cada valor alocado na pilha correspondente ao



número do módulo usado (6 bits) e que reside na memória, e a posição de início do módulo lógico no módulo físico (0,1,2 ou 3).

Na base da pilha estará sempre o módulo menos recentemente utilizado.



É importante notar que o campo TCB aqui indicado não é o mesmo que o campo do rótulo do arquivo que contém o tamanho máximo da chave a ser usada (o qual por simplificação chamaremos de TMC), apesar de seus conteúdos poderem ser iguais. A verdade é que:

$$TCB \leq TMC$$

onde TCB é o número de caracteres (ou bytes) de cada entrada, que pode ser no máximo igual a TMC. É com TMC que, na criação do arquivo, é calculada a quantidade de entradas existentes em cada módulo. Essa quantidade, aqui simbolizada por NEPM é:

$$NEPM = \frac{253}{5+TMC}$$

Abaixo, mostramos uma tabela onde indicamos para cada tamanho de TCB, o número de entradas NEPM e o número de bytes finais de cada módulo (excluindo os 3 bytes de identificação de módulo) que podem ser usados para armazenar informações quaisquer, tanto pelo usuário quanto pelo SGMES.

TCB	NEPM	DISPONÍVEIS
1	42	1
2	36	1
3	31	5
4	28	1
5	25	3
6	23	0
7	21	1

Como o NEPM é então, possível calcular o número da

entrada inicial de cada módulo pela seguinte fórmula:

$$NEI(M) = m \times NEPM \quad m = 0,1,2,\dots$$

onde $NEI(n)$ é o número da entrada inicial do m -ésimo módulo lógico e m é a ordem deste.

Como já foi dito, a determinação da entrada na tabela de índices é feita via uma função hash sobre a chave dada. A seguir damos uma explicação detalhada da função hash e do tratamento que é dado para as colisões.

Dada uma chave qualquer, geramos a partir dela, um número de 2 bytes, isto é, até 65535, da seguinte forma: tomam-se os dois primeiros bytes da chave, da direita para a esquerda, e somamos binariamente seu conteúdo ao daqueles dos dois bytes seguintes, no mesmo sentido, e assim fazemos sucessivamente até os últimos dois bytes da chave. Se a chave contém apenas 1 byte, então o conteúdo binário deste byte é transformado em decimal, o qual será o número procurado.

Esse número que denominamos de chave reduzida dividimos por DIVISOR, onde:

$$\text{DIVISOR} := P(X_i):$$

$$X_i := \text{N}^\circ \text{ DE SETORES POR BLA} \times \text{NEPM}_i \times$$

$$\text{N}^\circ \text{ DE BLA'S ALOCADAS PARA A TABELA DE ÍNDICES} = \text{N}^\circ \text{ DE ENTRADAS NA TABELA DE ÍNDICES}$$

$$P(X_i) = \text{n}^\circ \text{ primo mais próximo, por excesso, de } X_i$$

Tomamos a parte decimal do quociente referido acima e multiplicamos por X_i ; a parte inteira deste produto corresponde a entrada na tabela.

Abaixo apresentamos uma tabela de $P(X_i)$ para alguns X_i

X_i	$P(X_i)$
84	89
92	97
100	101
112	113
124	127
144	149
168	173
184	191
200	211
224	227
248	251
252	257
276	277
288	293
300	307
336	337
368	373
372	379
400	401
432	433
448	449
496	499
504	509
576	577
672	673

As colisões, no máximo de 7 por entrada, serão resolvidas da seguinte forma:

Dada uma colisão será feita uma pesquisa linear, com passo δ , para frente, nos módulos existentes na memória, até encontrar uma posição vazia, caso não encontre, faz substituição do módulo menos recentemente utilizado e continua a pesquisa até encontrar uma posição vazia ou encontrar fim de tabela, quando então será reiniciada a pesquisa pelo início da tabela até encontrar novamente fim de tabela, ou a chave colidida. Ao encontrar o lugar vazio, coloca-se a chave e atualizam-se os registros da tabela, tanto no lugar colidido como no lugar onde foi alocada a chave.

O passo δ de deslocamento será igual ao conteúdo dos três primeiros bits do 1º byte da "chave reduzida".

APÊNDICE 3

ESTRUTURA DO BCES

ESTRUTURA DO BLOCO DE COMUNICAÇÃO DE ENTRADAS E SAÍDAS

A comunicação de qualquer processo com qualquer servidor do sistema operacional, que gerencie recursos de entrada e saída, é realizada por intermédio de uma estrutura de dados, na área do processo, cujas informações são geridas tanto pelo usuário como pelo sistema operacional. Essa área que se denomina Bloco de Comunicação de Entradas e Saídas - BCES, tem a finalidade de armazenar informações coerentes e necessárias à operação, manipulação e gerência de arquivo. Para cada arquivo aberto pelo processo do usuário, necessita-se reservar 100 bytes para o seu BCES (O tamanho normal do BCES, em geral, será 256 bytes).

O BCES, é logicamente dividido em duas áreas: uma de interesse comum a todos os serviços de entrada e saída (impressora, disco, vídeo, etc) e uma outra que é de interesse específico a cada uma desses serviços.

A parte comum, por nós chamada de "parte fixa", conterá aquelas informações que são necessárias, independentemente do tipo de entrada e saída. A outra parte, (SGMES, LICO, REDE, etc) define suas necessidades em função das operações que irá realizar.

No nosso caso, o SGMES, necessita que algumas informações sejam armazenadas nesta última parte, em função de cada tipo de operação que for realizar.

A seguir, apresentamos o conteúdo do BCES:

Parte Fixa:

1. NUMSER - Números de Serviço - consiste em três números (cada um de tamanho de 1 byte) que indicam respectivamente o número associado à fila de serviço de cada serviço do sistema operacional: de escrita, da leitura e especial. - 3 bytes
(OBS: Números colocados pelo núcleo)
2. REDE1 - Para uso em trabalhos e processos que utilizam rede de teleprocessamento. - 4 bytes
3. NUMPRO - Número do processo do usuário, responsável por este BCES. - 2 bytes
- 4 - IDIUSU - Identificação do usuário, criador do processo acima descrito. - 3 bytes
5. NOMEVL - Identificação do Veículo, que consiste de:
 - Nº - nº do nº da rede - 2 bytes
 - VOLUME - nome do volume - 6 bytes
 - DIRETOR - nome do diretório - 6 bytes
 - ARQUIVO - nome do arquivo - 6 bytes
 - EXTENS - extensão - 3 bytes
6. REDE2 - Uso futuro - 2 bytes
7. ENTRAD - Entrada nos descritores - são os índices que indicam a ordem da entrada do arquivo corrente nos diversos descritores e diretórios mantidos pelos serviços de E/S. No caso do SGMES temos:

- DEVOH - 1 byte
 DESAU - 1 byte
 DEBUR - 1 byte
 SIX - 1 byte
 DIX - 1 byte
 DLB - 1 byte
 uso futuro - 4 bytes
8. RECEIVE - Endereço do Buffer de Recepção de dados, quando for operação leitura - 2 bytes
9. SEND - Endereço do Buffer de Envio, de dados, quando for operação de escrita - 2 bytes
10. ACESSO - Qualificação do Acesso - No caso do SGMES, significa o tipo de acesso requerido ao arquivo:
 - sequencial
 - direto relativo
 - sequencial indexado
 - 1 byte
11. CÕDIGO - Código de Operação - 2 bytes
12. STATUS1 - Condição de término. Recebe um código indicando o tipo de término que aconteceu com a operação solicitada - 2 bytes
13. CONTAL - Contador de registros lidos - 2 bytes
14. CONTAE - Contador de registros escritos - 2 bytes
15. FISICA - Nº da unidade física - 1 byte
16. FUTURO - Uso futuro - 10 bytes

Parte específica (SGMES)

- 17 - SEGURA - Registro de Segurança e Acesso - 2 bytes
- 18 - NUMREG - Número do Registro - 2 bytes
ou nº de BLAS do ar.seq.indexado
19. NUMCAR - Número de Caracteres - 2 bytes
ou nº de BLAS para tabela de índices
de arquivos seq.indexado
20. TAMCHV - Tamanho Chave - 1 byte
ou Código atributo
21. CHAVE - Chave do registro em arquivos sequencial indexado ou novo atributo - 7 bytes
22. POSIC1 - Ponteiro do buffer de recepção - 1 byte
23. POSIC2 - Ponteiro do buffer de envio - 1 byte
24. CONTAF - Contador regressivo do número de bytes faltando para o fim do registro (usado quando da leitura parcial) ou contador progressivo de caracteres escritos no registro (usado quando de escrita parcial) - 2 bytes
25. INCREM - Número de BLA's a serem alocadas a cada nova alocação de espaço em disco - 1 byte
26. STATUS2 - Condição de Interface (uso futuro) - 6 bytes

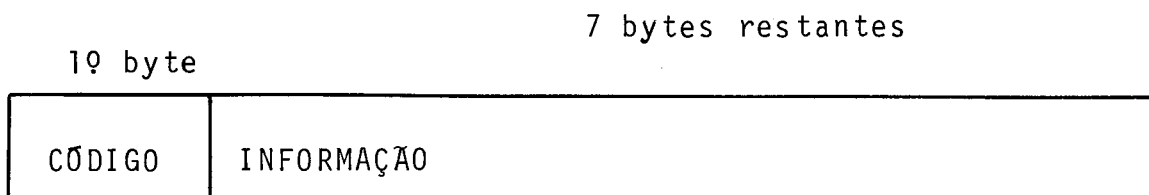
Como vemos, a parte fixa, contém 70 bytes de tamanho, e a parte variável ou específica, no caso do SGMES, ocupará 25 bytes apenas, dando um total de 95 bytes apenas.

APENDICE 4

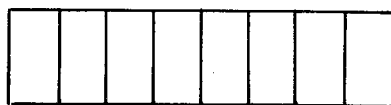
INTERFACE DE DISCO

A interface de disco na qual foi baseado todo o desenvolvimento do SGMES, tem características bastante avançadas, tanto de hardware e software, o que permite uma grande independência do software de gerência e manipulação de arquivos do hardware ou dos tipos de disco que estejam sendo usados; esta transparência dos equipamentos em relação ao software de gerência (SGMES), é vital para a simplificação do tratamento físico dos arquivos.

A comunicação do SGMES com a interface é dinâmica e feita através de mensagens. Cada mensagem pode ter no máximo um comprimento de 8 bytes, tendo a seguinte estrutura:



O primeiro byte de qualquer MID (Mensagem de Interface de Disco) é sempre um código de operação de interface dentre as seguintes:



código quantos bytes de inf. são transp.

- LER SETOR SEM VERIFICAÇÃO
- GRAVE SETOR SEM VERIFICAÇÃO
- LEIA SETOR COM VERIFICAÇÃO
- GRAVE SETOR COM VERIFICAÇÃO
- PEGUE CARACTERÍSTICAS DO DISCO DA UNIDADE "K"

- ATIVE CARACTERÍSTICAS DADAS PARA O DISCO DA UNIDADE "K"
- INICIALIZA DISCO DA UNIDADE "K"
- BOOTSTRAP
- CARREGUE PROGRAMA DE CANAL
- BUSQUE SETOR "S" NO DISCO DA UNIDADE "K"

Os 7 bytes seguintes da MID devem ter a seguinte estrutura:

1º byte = Nº DO UNIDADE

2º e 3º byte = ENDEREÇO DO SETOR

4º ao 7º bytes = informações necessárias às operações, quando for o caso.

O "endereço do setor" corresponde sempre ao número do setor em relação ao início do disco.

Após cada interrupção dada pela interface, o microprocessador alocará no registro bidirecional um código de 8 bits que representa o status da operação pedida à interface via MID.

Quando for o caso de uma leitura o SGMES guardará o status indicando tudo bem para iniciar a recuperação dos dados lidos no registro bidirecional, de 8 em 8 bits a cada 19 µs.

A mesma situação, acontece na gravação, com exceção de que em vez de retirar dados, deve-se por dados no "registro bidirecional".

Vemos assim que a interface é inteligente, tendo

seu próprio software e processador, o que permite uma total flexibilidade do sistema no tratamento de informações, com relação aos equipamentos que estão sendo utilizados.

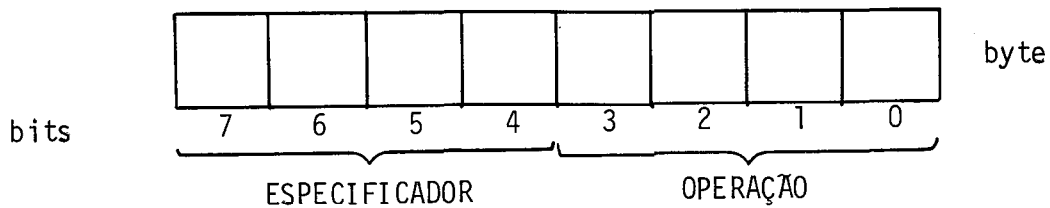
APÊNDICE 5

ESTRUTURA DOS CÓDIGOS DE OPERAÇÃO

Os c3digos de opera3o3o s3o valores hexadecimais (2 caracteres) que s3o utilizados na BCES como parte da mensagem de intercomunica3o3o entre os processos.

A estrutura do c3digo de opera3o3o 3 a seguinte:

1 # caracteres 2 # caractere



O especificador 3 o caractere que caracteriza o tipo de grupo a que pertence a opera3o3o. Os quatro bits dispon3veis para o "especificador" tem as seguintes configura3o3es poss3veis com as respectivas identifica3o3es:

bits

7	6	5	4	HEX	
0	0	0	0	0	- opera3o3es do n3cleo
0	0	0	1	1	- opera3o3es do n3cleo
0	0	1	0	2	- opera3o3es da LICO
0	0	1	1	3	- opera3o3es da LICO
0	1	0	0	4	- opera3o3o do SGMES
0	1	0	1	5	- opera3o3o do SGMES
0	1	1	0	6	- opera3o3o do SGMES
0	1	1	1	7	- opera3o3o do SGMES
1	0	0	0	8	- opera3o3es de perif3ricos lentos (impressora, cart3es, etc)
1	0	0	1	9	- uso futuro
1	0	1	0	A	- uso futuro
1	0	1	1	B	- uso futuro

1	1	0	0	C	- operações de comunicação de dados
1	1	0	1	D	- operações de comunicação de dados
1	1	1	0	E	- operações especiais
1	1	1	1	F	- operações especiais

De cada um dos 16 grupos acima são permitidos 16 operações, isto é, ao todo são possíveis na estrutura apresentada acima, até 256 operações, no máximo.

Ao SGMES são reservados 4 caracteres "especificador", os quais têm a seguinte significação:

<u>caractere</u>		<u>tipo de operação</u>
4(0100)	-	operações de escrita
5(0101)	-	operações de leitura
6(0110)	-	operações de habilitação
7(0111)	-	operações de gerência