



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA  
**INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS**

**INPE-9596-PRE/5224**

## **UM SISTEMA OPERACIONAL PARA O MCR**

Eliane Martins\*  
Maria de Fátima Mattiello-Francisco  
Antonio Ésio Marcondes Salgado  
Pedro Paula Santos Júnior

Instituto de Computação - UNICAMP

Trabalho apresentado no III Simpósio Brasileiro sobre Redes de Computadores –  
Rio de Janeiro, 1 a 3 de abril de 1985.

INPE  
São José dos Campos  
2003

- tação de um Protocolo de Transporte para a Rede CEPINNE, submetido ao 3o. Simpósio Brasileiro de Redes de Computadores, Janeiro, 1985.
- [IDEAT 83] DEATON Jr., G.A. - HIPPERT, Jr., R.O. - X.25 and Related Recommendations in IBM Products, IBM Systems Journal, 22 (1/2): 11-29, 1983.
- [IDEC 79] DIGITAL EQUIPMENT CORPORATION - Terminals and Communications Handbook - 1979, capítulo 7, p. 145-159.
- [IDEC 82] DIGITAL EQUIPMENT CORPORATION - Communication Between Processes: IFCF, Software Notebook 4, July 1982, p. 8.1 - 8.24.
- [MONT 84] MONTEIRO, J.A.S., JUREMA, M.A. - CUNHA, P.R.F. - Comporta para Conexão de um Computador DEC-10 à uma Rede Pública de comunicação de Dados, in: 2. Simpósio Brasileiro sobre Redes de Computadores, Anais, Campina Grande, 1984, p. 2.1 - 2.21.

## 3o SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES (3o SBRC)

### UM SISTEMA OPERACIONAL PARA O MCR

Eliane Martins  
 Maria de Fátima Mattiello  
 Antonio Esio Marcondes Salgado  
 Pedro de Paula Santos Júnior

Instituto de Pesquisas Espaciais - INPE  
 Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq  
 C.P. 515 - 12200 - São José dos Campos - SP

### RESUMO

O trabalho apresenta um Sistema Operacional de tempo real que apoia a execução de processos aplicativos sob demanda. Estes processos comunicam-se através de troca de mensagens e visam implementar os níveis mais baixos do protocolo de comunicação de dados - funções básicas de um nó de rede de computadores. O Sistema Operacional utiliza monitores para gerenciar os barramentos e linhas seriais do nó. Este sistema foi projetado para o multi processador de comunicação em Redes - MCR ( nó de uma sub-rede de comunicação de dados ) desenvolvido pelo INPE/CNPq.

## 1. INTRODUÇÃO

O Multiprocessador de Comunicação em Rede - MCR é um periférico que faz parte de um nó da sub-rede de comunicação de dados do Sistema REDACE/INPE.

Sua função é gerenciar a comunicação de dados em um grupo de linhas seriais implementando, para isto, os níveis mais baixos do protocolo de comunicação, responsáveis pelo transporte de dados e rotinas auxiliares de roteamento [2].

O Sistema Operacional que dará suporte à execução das funções acima citadas é essencialmente composto por núcleos e monitores modulares, e caracteriza-se como um Sistema Operacional Distribuído. É, basicamente, um S.O. de tempo real com capacidade de multiprocessamento, concorrência e gerenciamento de processos.

O principal objetivo do S.O. do MCR é garantir uma estrutura altamente modular para o sistema, além de:

- Prover mecanismos de detecção de erros, baseado em temporização;
- Tornar a estrutura física (hardware) transparente aos processos aplicativos, de modo que estes possam ser programados independentemente do processador em que serão executados.

## 2. ARQUITETURA DO MCR

A configuração de um nó básico da sub-rede de comunicação de dados em questão está descrita na Figura 2.1.

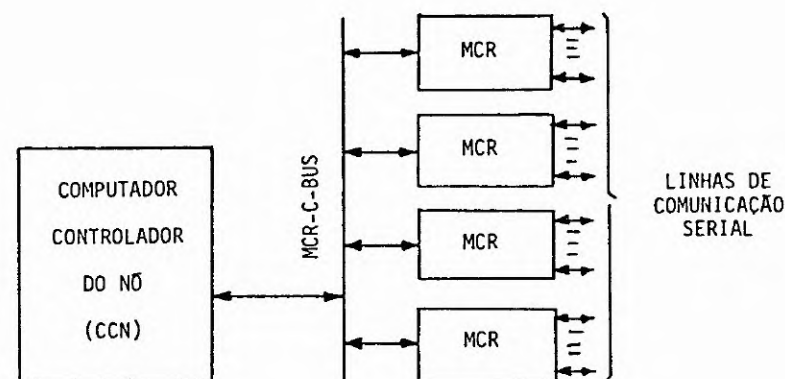


Fig. 2.1 - Nó básico da sub-rede.

As linhas seriais do nó estão conectadas a outros nós da sub-rede, e a máquinas de fim específico que são controladas remotamente, via sub-rede.

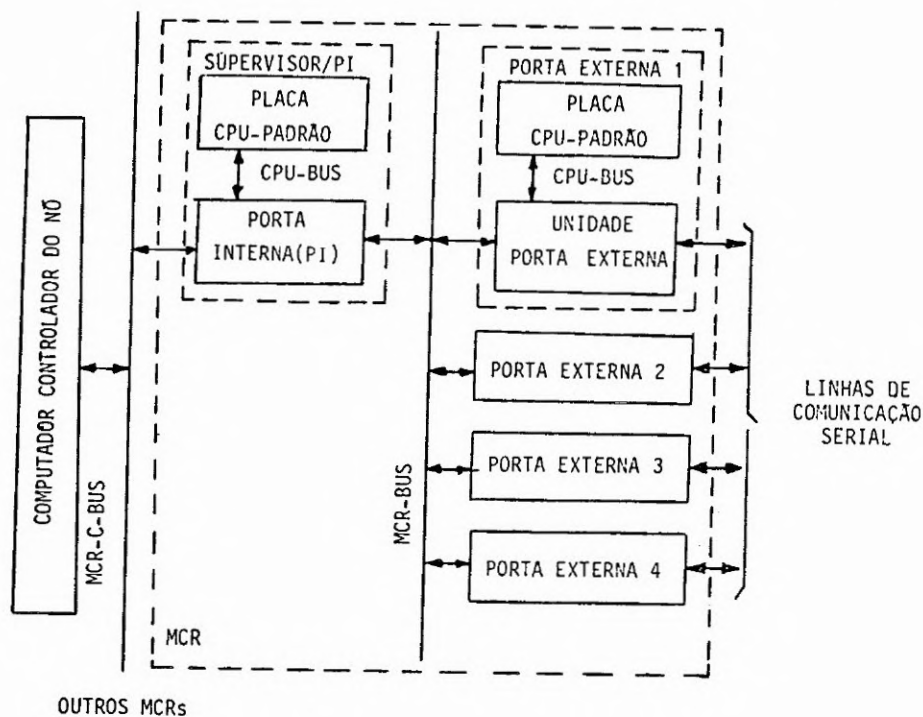
O Computador Controlador do Nó - CCN realiza estatística e controle do fluxo de dados através do nó, exercendo para isto a gerência sobre a utilização do barramento interno que interliga os MCR's e o próprio CCN. Através deste barramento chamado MCR-C-BUS, são efetuadas as trocas de mensagens MCR ↔ MCR e MCR ↔ CCN.

O protocolo de comunicação que opera na sub-rede tem seus níveis mais baixos, responsáveis pelo transporte dos dados, implementados no nó formado pelo conjunto MCR's + CCN, cabendo aos MCR's a implementação dos níveis que garantem a conexão física e a transferência de dados de forma adequada através das linhas seriais.

As mensagens enviadas através das linhas seriais conectadas aos MCR's são enviadas a seus destinos que podem ser: linha serial pertencente ao mesmo MCR ou, linha serial pertencente a outro MCR ou o CCN. Caso a mensagem recebida não possa ser enviada ao destino devido a problemas temporários na conexão com este, ela pode ser armazenada temporariamente no CCN, que possui unidades de memória de massa.

### 3. DESCRIÇÃO DOS MÓDULOS

Pode-se visualizar o MCR através do seguinte diagrama:



MCR: Multiprocessador de Comunicação em Rede

PI: Porta Interna

MCR-BUS: Barramento interno ao MCR

MCR-C-BUS: Barramento de interligação do Computador Controlador do Nó ↔ MCR's.

Fig. 3.1 - Diagrama de blocos do MCR.

#### 3.1 - HIERARQUIA DOS MÓDULOS DENTRO DO MCR

- Dentro da arquitetura de multiprocessamento, tem-se uma relação tipo mestre-escravo, onde o Supervisor controla todo e qualquer fluxo de dados no MCR, a saber:

A - Comunicação de dados com o Computador controlador do nó (CCN), via MCR-C-BUS através da Porta Interna (PI).

B - Comunicação de dados recebidos através da Porta Interna (PI) para uma das Portas Externas (PE), via MCR-BUS.

C - Permutação de dados entre duas Portas Externas (PE) através do MCR-BUS.

#### 3.2 - O SUPERVISOR (SV)

- Suas principais funções são:

a) controle do acesso ao barramento interno (MCR-BUS), nas comunicações tipo B e C;

b) controle da troca de mensagens entre PE's, entre PE e PI;

c) controle da troca de mensagens entre a PI e o Computador Controlador do nó (CCN), via o MCR-C-BUS.

O Supervisor não se destina ao armazenamento de pacotes de dados. A ele cabe arbitrar sobre o fluxo destes pacotes, atendendo aos diversos pedidos de transferência de pacotes;  $PE_i \leftrightarrow PE_j$ ,  $PE \leftrightarrow PI$  ou  $PI \leftrightarrow CCN$ .

Cabe ao SV a diagnose de todo o MCR, o que é feito através da emissão periódica de comandos às varias PE's. Em caso de falha o SV deve comunicar-se com o CCN para uma reconfiguração do sistema.

O SV mantém estatísticas sobre o tráfego de pacotes no MCR, e periodicamente estas são transmitidas ao CCN.

### 3.3 - A PORTA INTERNA (PI)

- Não tem processador próprio e funciona como um "buffer" intermediário para a troca de mensagens entre o meio interno (PE's) e externo (CCN ou outro MCR), sendo controlada pelo SV.

É uma memória compartilhada que tem três vias de acesso:

- a) CPU-BUS (Supervisor).
- b) MCR-C-BUS (Computador Controlador do Nó ou outro MCR).
- c) MCR-BUS (Porta Externa).

Cabe ao Supervisor a liberação do acesso à PI para um e somente um dos possíveis acessos acima citados.

O acesso à PI pelo Supervisor é feito através da liberação da conexão do seu barramento de dados/endereço, de forma que a PI passa a ser uma parte da memória do Supervisor.

Por outro lado, o acesso à PI através do MCR-BUS/MCR-C-BUS, requer uma consulta prévia ao Supervisor, o qual decidirá sobre a liberação da conexão com a PI.

Desta forma, quando uma PE deseja acessar a PI, é necessária uma consulta prévia ao Supervisor para uma posterior liberação da conexão através do MCR-BUS. De maneira análoga é realizado o mesmo procedimento para acesso à PI através do MCR-C-BUS por parte do Computador Controlador do Nó (ou outro MCR do Nó).

### 3.4 - A PORTA EXTERNA (PE)

- É o elemento que realiza a função principal do MCR: a comunicação de dados na rede.

É composta de duas partes:

- 1) CPU-padrão e
- 2) Unidade Porta Externa.

Na CPU-padrão tem-se o programa que executará funções dos níveis mais baixos do protocolo especificado para a comunicação de dados, o qual pode ser particular a cada uma das PE's, dependendo da aplicação. Também cabe a CPU-padrão o armazenamento temporário de mensagens, até que a PE consiga autorização do SV para transmitir o pacote armazenado, via o MCR-BUS, para a PI ou PE. A Unidade Porta Externa é composta de uma interface de comunicação serial e uma unidade de acesso direto à memória - DMA. A Unidade de DMA opera por roubo de ciclo da CPU durante a transmissão/recepção de dados pela linha serial. No caso das transmissões via MCR-BUS (para a PI ou outra PE) a Unidade de DMA opera por avalanche, colocando a CPU (da placa CPU-padrão) em "hold".

### 4. O SISTEMA OPERACIONAL

O "software" a ser implementado no MCR é constituído de duas partes:

- *Processos Aplicativos*: realizam as funções para as quais o processador foi destinado.
- *Sistema Operacional*: fornece a infra-estrutura necessária para o apoio aos Processos Aplicativos, tornando o funcionamento interno do processador transparente ao nível da aplicação.

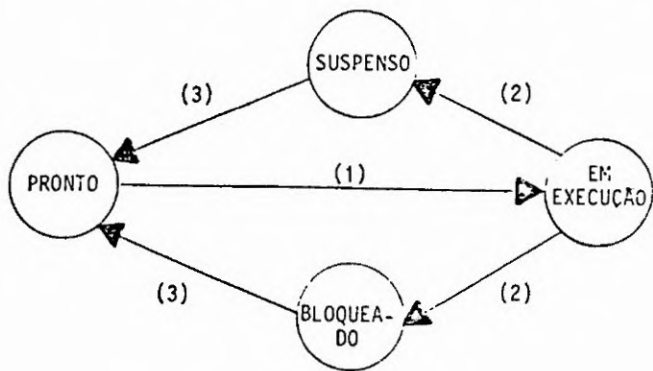
O sistema operacional é multi-tarefa, para tempo real, e permite a comunicação entre processos através da troca de mensagens. É constituído por um núcleo - residente em cada processador do MCR - e por Monitores especificamente projetados para realizarem a interface com os dispositivos de E/S (bem como), com os barramentos interno (MCR-BUS) e externo (MCR-C-BUS).

#### 4.1 - O NÚCLEO

O núcleo é constituído por um conjunto de rotinas e estruturas de dados que provêm mecanismos para:

- escalonamento de processos;
- comunicação e sincronização entre processos;
- controle de tempo;
- gerenciamento de memória.

As entidades básicas suportadas pelo núcleo são os *processos* e as *mensagens*. A Figura 4.1 mostra os possíveis estados de um processo.



- (1) escalonamento do processo
- (2) espera de um evento (uso de primitiva)
- (3) ocorrência de um evento.

Fig. 4.1 - Diagrama de Estados de um Processo.

#### ESCALONAMENTO DE PROCESSOS

Cada processo tem uma prioridade pré-definida, atribuída pelo usuário, conforme a aplicação. A prioridade do processo é utilizada pelo núcleo para o escalonamento: o processo mais prioritário, no estado de pronto, é escolhido para execução. O escalonamento é sob demanda, ou seja, uma vez alocada a CPU a um determinado processo, este permanece em execução até terminar sua tarefa ou requerer os serviços do Sistema Operacional ou de outro processo. Apenas os monitores caracterizam-se como exceção a este mecanismo, pois após sua ativação, pela rotina de atendimento de interrupção, é efetuado um reescalonamento. Cabe notar que, neste sistema, os monitores são considerados processos de alta prioridade, o que implica na possibilidade de ocorrência de preempção.

Os estados de um processo são representados por filas que fazem parte da estrutura de dados do núcleo. A fila de prontos é mantida pela ordem da prioridade para a execução dos processos. Dentre os processos de mesma prioridade, o atendimento se dá na forma FIFO. [1]

#### COMUNICAÇÃO E SINCRONIZAÇÃO ENTRE PROCESSOS

A comunicação entre processos é feita através de troca de mensagens [1], pois utiliza a estrutura do ambiente de comunicação. O processo fonte não necessita saber onde reside o destinatário: caso este resida em um processador diferente, a mensagem é encaminhada através do monitor do barramento.

As mensagens são transmitidas entre processos através de *bloques de mensagens*, de tamanho fixo, obtidos de uma *área comum* gerenciada pelo núcleo.

A cada processo estará associada uma fila de mensagens. As mensagens que são enviadas por outros processos são inseridas nessa fila. Normalmente as mensagens são retiradas da fila na forma FIFO; no entanto um processo tem flexibilidade de atender somente as mensagens provenientes de um determinado processo fonte. Um processo pode ficar suspenso ou bloqueado a espera de uma mensagem - qualquer ou específica - na sua fila.



Para enviar uma mensagem, um processo deve primeiramente requerer um bloco na área comum. Em seguida esse bloco é preenchido com o texto da mensagem e inserida na fila do processo destino. Após ter sido utilizado, o bloco é devolvido à área comum.

O número de mensagens que um processo pode manter na sua fila é limitado. Com isso evita-se que um processo produtor "muito veloz" possa esvaziar a área comum em detrimento de outros processos, caso o consumidor de suas mensagens seja "mais lento".

#### GERENCIAMENTO DE MEMÓRIA

O núcleo irá gerenciar o uso da memória RAM, na qual estarão alocadas as estruturas de dados (área comum, filas, descritores de processos), as pilhas e variáveis tanto do núcleo quanto dos Processos Aplicativos. O código das rotinas ficará em EPROM.

#### CONTROLE DE TEMPO

É permitido aos processos: 1) suspender sua execução por um tempo determinado ou 2) aguardar a ocorrência de um evento por um tempo especificado.

Cada processador conta com um temporizador programável, o qual dispõe de contadores decrescentes. No primeiro caso, um processo fica suspenso até que o contador chegue a zero, quando então ele passa ao estado de pronto. No segundo caso, o processo pode ser ativado quando da ocorrência do evento, sendo interrompida a contagem de tempo; ou quando esgotar o tempo, significando que o evento esperado não ocorreu.

Os processos suspensos são inseridos em uma fila, organizada em função do momento em que o processo deve ser ativado. A temporização é relativa, ou seja, a contagem de tempo é feita em relação ao momento em que se inicializa o contador. Todas as contagens devem ser múltiplas inteiras da base de tempo do sistema.

#### PRIMITIVAS

As primitivas constituem a interface entre o núcleo e os processos aplicativos. Através delas os processos têm acesso aos serviços fornecidos pelo núcleo. O uso de primitivas corresponde portanto a chamadas para diversas rotinas que executam tarefas específicas.

As seguintes primitivas estão disponíveis aos processos:

##### 1) SUSPENDE

O processo passa ao estado de suspenso até que seja esgotado um tempo de espera, quando então ele é ativado.

##### 2) RETORNE

Libera o processador. O processo que usou a primitiva passa ao estado de pronto ou bloqueado, no caso de sua execução depender de um determinado evento.

##### 3) PEDE

O processo pede à área comum um bloco de mensagem. Caso não haja nenhum disponível, o processo é avisado.

##### 4) LIBERA

Devolve um bloco de mensagem para a área comum. O contador do número de mensagens alocadas ao processo é decrementado de 1.

##### 5) PEDEPI

Análoga à PEDE, só que o bloco é alocado a partir da área comum localizada na PI.

Esta primitiva só é utilizada pelo núcleo do SV (supervisor do MCR).

## 6) LIBERAPI

Análoga à LIBERA, só que o bloco é devolvido à área comum da PI. Esta primitiva só é utilizada pelo núcleo do SV.

## 7) ENVIA

Completa informações do bloco de mensagem e insere-o na fila do processo destino. Caso o processo destino resida em outro processador, o bloco é inserido na fila de mensagens do monitor que trata da transferência de dados pelo MCRBUS.

Caso o número de mensagens na fila do destinatário já esteja esgotado, a mensagem não é enviada e o processo origem é avisado do fato.

Se o processo destino estiver suspenso ou bloqueado a espera dessa mensagem, ele deve ser ativado.

## 8) RECEBE

Retira uma mensagem da fila do processo.

Em geral a primeira da fila é fornecida, a menos que o processo requeira uma mensagem específica.

Caso não seja encontrada a mensagem — a fila está vazia ou a mensagem requerida não foi encontrada —, o processo é avisado.

## 9) ESPERA

Análoga à RECEBE mas em caso de fila vazia ou não encontrar a mensagem especificada, o processo fica bloqueado ou suspenso — espera temporizada — aguardando o aparecimento da mensagem.

## ESTRUTURAS DE DADOS

### 1) Tabela de Descritores de Processos

O núcleo mantém um descritor para cada processo contendo as seguintes informações: estado atual, prioridade, contador de tempo de espera, área de contexto, endereço inicial de execução, número de mensagens, condição de bloqueio, ponteiros para o início e fim da fila de mensagens, endereço da área de dados e da base da pilha, ponteiro para o anterior e o próximo processo no mesmo estado, identificação do processo.

### 2) Filas de Mensagens

Há uma fila para cada processo, contendo as mensagens que lhe são destinadas. Cada mensagem estará contida em um bloco contendo as informações:

- tipo da mensagem
- identificação dos processos fonte e destino
- texto da mensagem
- ponteiros para as mensagens anterior e a próxima.

A fila de mensagens é encabeçada a partir do descritor do processo.

### 3) Filas de Processos

Cada estado de um processo é representado por uma fila.

A fila de prontos é organizada por prioridade: os processos mais prioritários encabeçam a fila.

Cada fila tem um *cabeça* contendo um ponteiro para o primeiro processo da fila.



#### 4) Área comum

A área comum é constituída de blocos de mensagens disponíveis para os processos. Cada bloco tem um tamanho fixo e faz parte de uma lista duplamente ligada.

Quando é pedido um bloco, o primeiro da lista é fornecido, mantendo no entanto uma quantidade mínima de blocos disponíveis para atender às rotinas de interrupção.

A lista de blocos disponíveis tem um cabeça contendo um ponteiro para o início da lista e o número total de blocos disponíveis.

#### 4.2 - OS MONITORES

Os monitores são processos de alta prioridade que interagem diretamente com as rotinas de atendimento de interrupção, tornando o ambiente físico transparente ao núcleo e aos processos aplicativos.[3].

Basicamente os monitores servem de interface para os seguintes recursos:

- linhas de comunicação, nas PEs
- barramento interno
- barramento externo (ligação com o Controlador).

#### MONITORES DA PE

São três os monitores que controlam o fluxo de dados em uma PE: Monitor de Entrada, Monitor do MCRBUS e Monitor de Saída, como pode ser visto na Figura 4.2.

Cada monitor é ativado por uma rotina de Tratamento de Interrupção, havendo prioridade decrescente: Monitor de Entrada, MCRBUS e Saída, no seu atendimento.

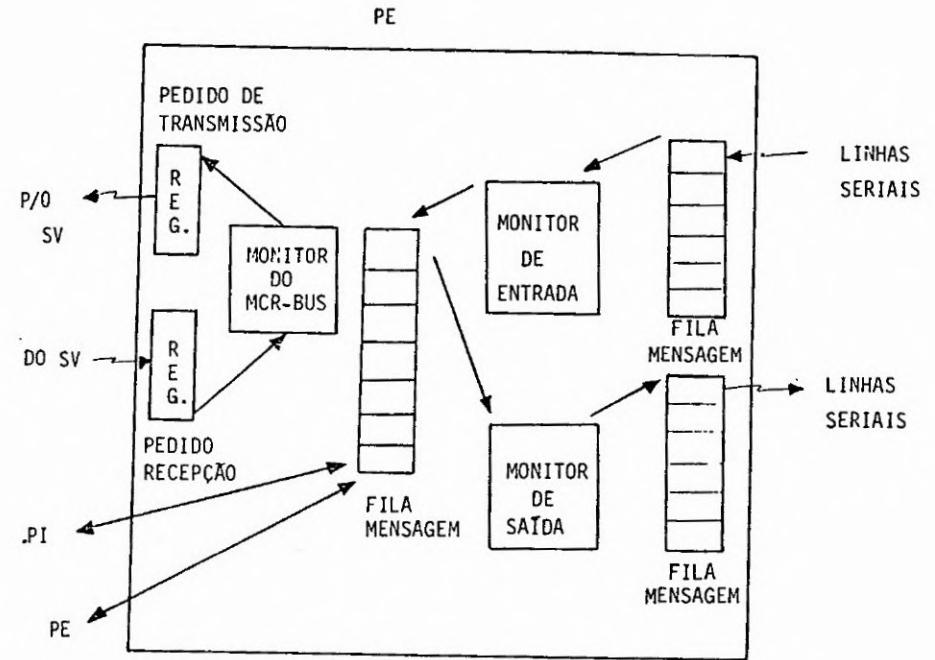


Fig. 4.2 - Diagrama de fluxo de mensagens na PE.

- Monitor de Entrada: Ativado pela rotina de atendimento da interrupção da interface serial, indicando a chegada de dados. O DMA é programado para a recepção dos dados e o monitor passa ao estado de suspenso/bloqueado aguardando nova interrupção:
- Caso a PE tenha somente uma linha de comunicação, o Monitor de Entrada fica a espera da interrupção do DMA que indica final de transferência.
- Senão, além da interrupção já citada, vinda do canal que iniciou a recepção, o monitor pode ser ativado pela chegada de dados em outro canal. Neste caso, o monitor monta uma tabela com o estado atual de cada canal para que quando ativado da próxima vez verifique o evento e atualize a tabela.

- Monitor de Saída: Fica bloqueado a espera de uma mensagem na sua fila. Da mesma forma que o Monitor de Entrada mantém uma tabela com o estado de cada canal. Várias transmissões podem ser ativadas por canais diferentes; o monitor fica a espera da interrupção do final de transmissão por um dado canal.

- Monitor de MCR-BUS: Responsável pela comunicação das PEs via barramento interno. Este monitor é ativado (passa ao estado pronto) nas seguintes condições:

- chegada de um pedido de recepção de SV
- mensagem da própria PE para ser transmitida a outra PE ou à PI.

#### MONITORES DO SV

O SV tem basicamente dois monitores:

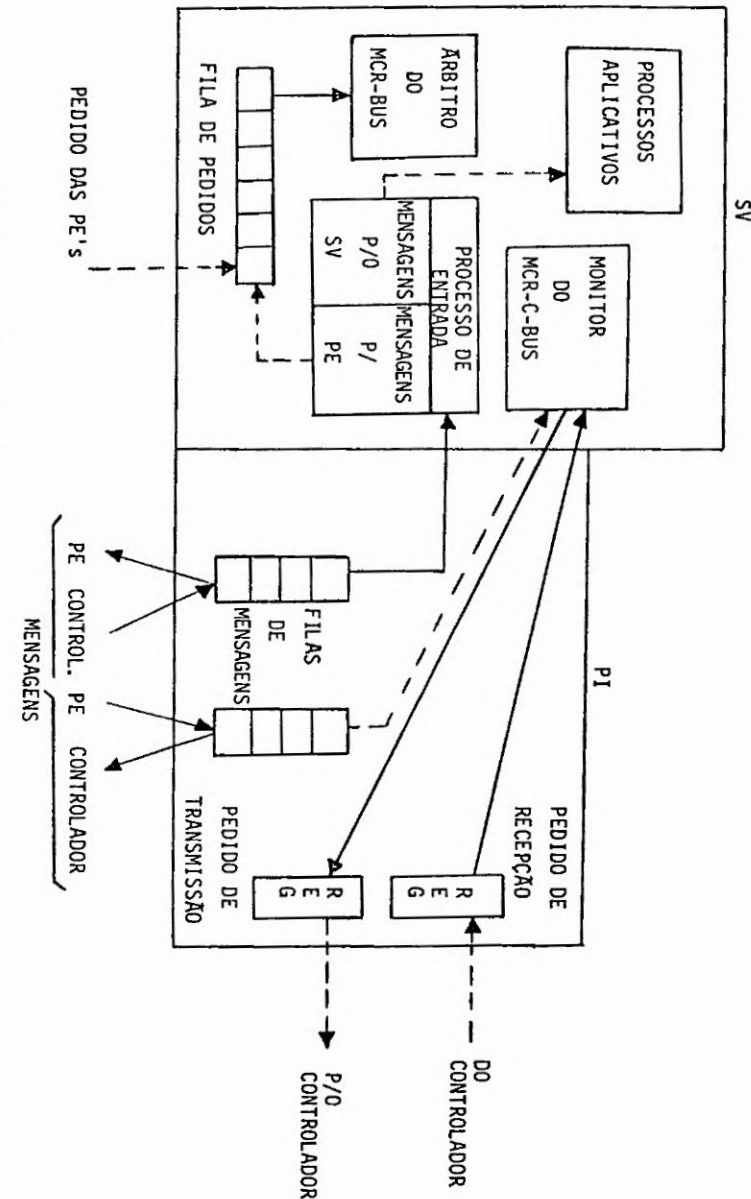
- Árbitro do MCRBUS: controla o acesso ao barramento interno;
- Monitor do MCR-C-BUS: trata da comunicação entre o SV e o controlador de MCRS. O SV se comporta, com relação ao controlador, como se fosse uma PE. Este monitor tem prioridade menor que o Árbitro do MCR-BUS.

Um processo aplicativo, denominado Processo de Entrada, responde pelo controle das mensagens vindas do controlador. Este processo tem a função de identificar o destino destas mensagens, isto é se elas devem ser encaminhadas para uma PE ou se devem ser tratadas pelo próprio SV (ex.: mensagens para estatística de fluxo ou mesmo para atualização de tabelas de roteamento - Processos Aplicativos do SV). Vide Figura 4.3.

No caso da mensagem ser endereçada a uma PE, cabe ainda ao Processo de Entrada encaminhar, para o Árbitro do MCR-BUS, o pedido de transmissão desta mensagem à PE destino.

- Árbitro do MCRBUS: ativado caso haja mensagem na PI a ser transferida para uma das PEs; ou através da rotina de interrupção dos registros de escrita das PEs, quando houver pedido de transferência pelo barramento interno.

Fig. 4.3 - Diagrama de fluxo de mensagens na PI.



- Monitor do MCR-C-BUS: suas funções são análogas às do Monitor do MCRBUS, das PEs. É ativado nas seguintes condições:

- Pela rotina que atende os registros de comunicação da PI, indicando que o controlador do MCR deseja transmitir;
- Caso haja mensagem na sua fila (Monitor do MCR-C-BUS) que deve ser transmitida ao controlador de MCR.

## 5. CONCLUSÃO

O Sistema Operacional apresentado encontra-se em fase de implementação. Objetiva-se com esta 1ª versão, a validação conjunta do "hardware" do MCR bem como a análise do real desempenho dos mecanismos e estruturas de dados adotados.

Todos os cuidados foram tomados para que o Sistema Operacional tivesse como características principais: a simplicidade e modularidade. Assim, da mesma forma que estas propriedades poderão facilitar a distribuição eficiente dos processos aplicativos, o conjunto poderá apresentar um comportamento aquém das necessidades mínimas desejadas. Como consequência, isto implicaria em revisões futuras onde a introdução de mecanismos, tais como "time-slice", tornar-se-ão indispensáveis.

Técnicas de tolerância a falha por "hardware" e "software" deverão ser implementadas na versão de campo do MCR.

## 6. REFERÊNCIAS BIBLIOGRÁFICAS

[1] BRINCH HANSEN, P. Operating System Principles. New Jersey, Prentice Hall, 1973.

[2] HASHIOKA, M.H. Modelo e análise de uma interface de comunicação distribuída para aplicação em rede de comunicação por comutação de pacotes. Dissertação de Mestrado. São José dos Campos, INPE, 1983.

[3] MARTINS, R.C.O.; DE PAULA, A.R. - A Fault-Tolerant Multiprocessing Unit For On-Board Satellite Applications. Technical Report - Spar Aerospace Limited.