

1. Publicação nº <i>INPE-3491-PRE/728</i>	2. Versão	3. Data <i>Abril, 1985</i>	5. Distribuição ☐ Interna ☒ Externa ☐ Restrita
4. Origem <i>DCA/DCO</i>	Programa <i>REDACE</i>		
6. Palavras chaves - selecionadas pelo(s) autor(es) <i>SISTEMA OPERACIONAL DISTRIBUÍDO</i> <i>COMUNICAÇÃO/SINCRONIZAÇÃO ENTRE PROCESSOS</i> <i>TEMPO REAL</i> <i>GERÊNCIA DE BARRAMENTOS</i> <i>ESCALONAMENTO</i> <i>LINHAS SERIAIS DE UM NÓ</i>			
7. C.D.U.: <i>681.3.066</i>			
8. Título <i>UM SISTEMA OPERACIONAL PARA O MULTIPROCESSADOR DE COMUNICAÇÃO EM REDE</i>		10. Páginas: <i>25</i>	
		11. Última página: <i>18</i>	
		12. Revisada por <i>Mauro Hissao Hashioka</i>	
9. Autoria <i>Eliane Martins</i> <i>Maria de Fátima Mattiello</i> <i>Antonio Esio Marcondes Salgado</i> <i>Pedro de Paula Santos Júnior</i>		13. Autorizada por  <i>Marco Antônio Raupp</i> <i>Diretor Geral</i>	
Assinatura responsável <i>Eliane Martins</i>			
14. Resumo/Notas <i>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 Multiprocessador de Comunicação em Redes - MCR (nó de uma sub-rede de comunicação de dados) desenvolvido pelo INPE/CNPq.</i>			
15. Observações <i>Este trabalho foi apresentado no 3º Seminário Brasileiro de Redes de Computadores, realizado no Núcleo de Computação Eletrônica da Universidade Federal do Rio de Janeiro, de 1 a 3 de abril de 1985.</i>			

ABSTRACT

This work presents a real-time Operating System, that provides services to applications processes, which implement levels 2 and 3 of a communications protocol. The application processes are run by the operating system by request from other processes or external events; they use messages to communicate with each other. In the case of external events, such as node buses and serial lines, the interruptions are processed by Monitors. This system was designed for a Multiprocessor for Network Communication - MCH (node of a sub-network for data communications) developed by INPE/CNPq.



SUMÁRIO

	<u>Pág.</u>
LISTA DE FIGURAS	v
1. <u>INTRODUÇÃO</u>	1
2. <u>ARQUITETURA DO MCR</u>	1
3. <u>DESCRIÇÃO DOS MÓDULOS</u>	3
3.1 - Hierarquia dos módulos dentro do MCR	4
3.2 - O supervisor	4
3.3 - A porta interna	5
3.4 - A porta externa	5
4. <u>O SISTEMA OPERACIONAL</u>	6
4.1 - O núcleo	7
5. <u>CONCLUSÃO</u>	17
6. <u>REFERÊNCIAS BIBLIOGRÁFICAS</u>	18

LISTA DE FIGURAS

	<u>Pág.</u>
2.1 - Nō básico da sub-rede	2
3.1 - Diagrama de blocos do MCR	3
4.1 - Diagrama de estados de um processo	7
4.2 - Diagrama de fluxo de mensagens na PE	14
4.3 - Diagrama de fluxo de mensagens na PI	16

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200

1. INTRODUÇÃO

O Multiprocessador de Comunicação em Rede - MCR - é um periférico que faz parte de um nó na 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 - S.O. - 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 de 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") transparentes 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 é apresentada na Figura 2.1.

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

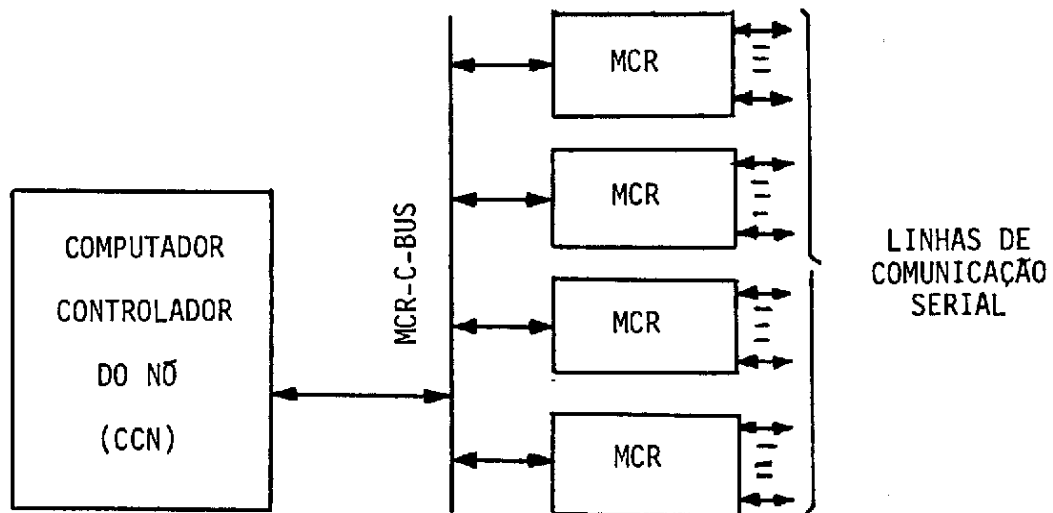


Fig. 2.1 - Nô básico da sub-rede.

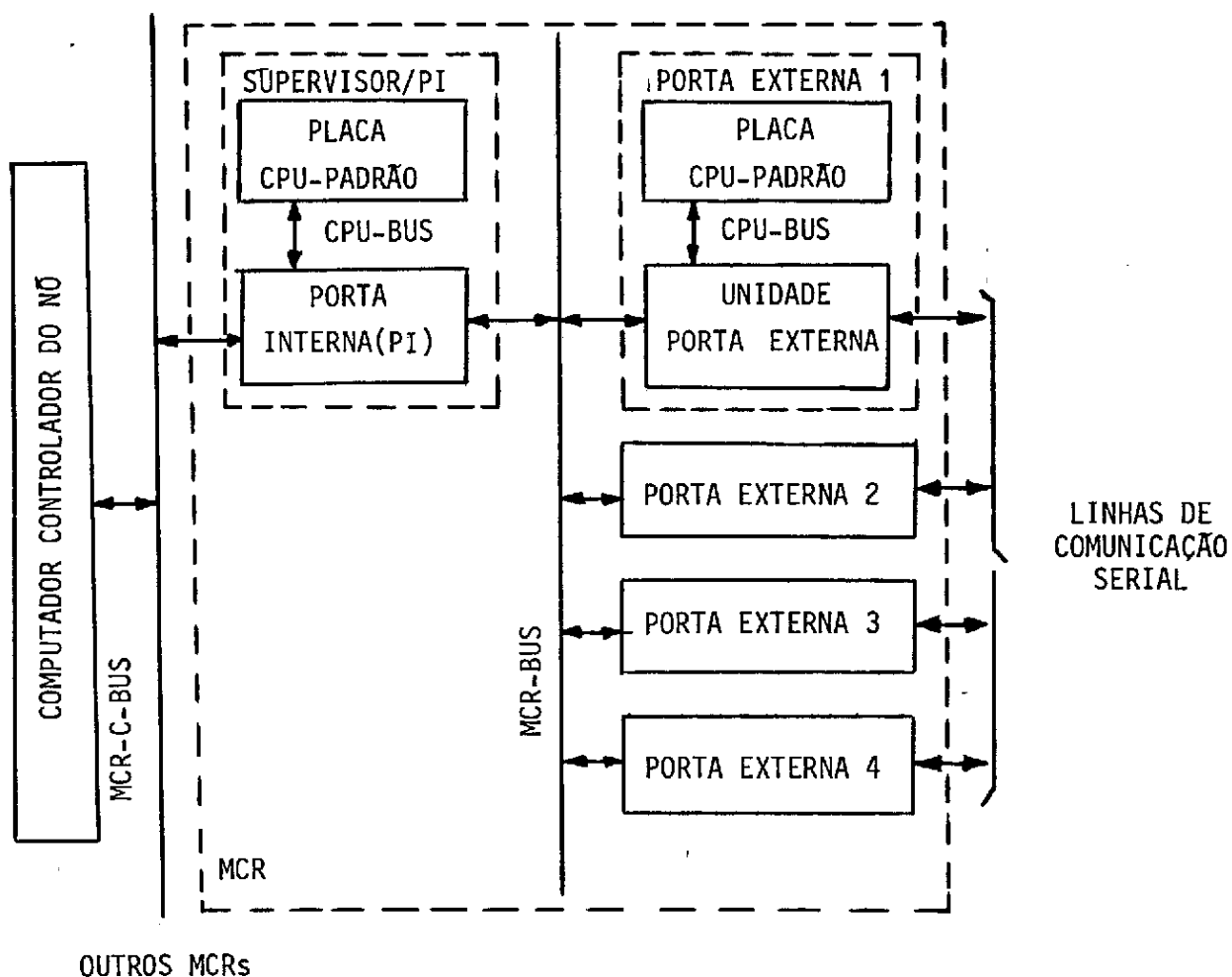
O Computador Controlador do Nó - CCN - realiza a estatística e o controle do fluxo de dados através do nó, exercendo para isto a gerência sobre a utilização do barramento interno que interliga os MCRs e o próprio CCN. Através deste barramento chamado MCR-C-BUS são efetuadas as trocas de mensagens $MCR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

O protocolo de comunicação que opera na sub-rede tem seus níveis mais baixos, os quais são responsáveis pelo transporte dos dados e implementados no nó formado pelo conjunto MCRs + CCN, cabendo aos MCRs 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 MCRs 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 seu 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 da Figura 3.1:



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 ↔ MCRs.

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 do tipo mestre-escravo, onde o Supervisor controla todo e qual quer fluxo de dados no MCR, a saber:
 - a) Comunicação de dados com o Computador Controlador do Nô (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

- As principais funções do Supervisor (SV) são:
 - a) controle do acesso ao barramento interno (MCR-BUS) nas comunicações do tipo B e C;
 - b) controle da troca de mensagens entre PEs e entre PE e PI;
 - c) controle da troca de mensagens entre a PI e o Computador Controlador do Nô (CCN), via o MCR-C-BUS.

O SV 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 várias PEs. 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

- Não tem processador próprio e funciona como um "buffer" intermediário para a troca de mensagens entre o meio interno (PEs) e o 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 Nô 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 o acesso à PI através do MCR-C-BUS por parte do Computador Controlador do Nô (ou outro MCR do Nô).

3.4 - A PORTA EXTERNA

- 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 PEs, 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.

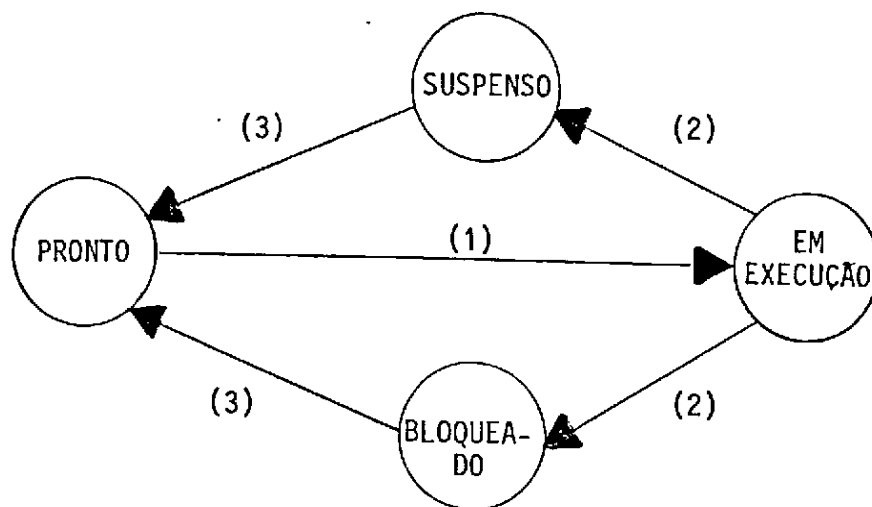
O sistema operacional é multitarefa para tempo real e permite a comunicação entre processos através da troca de mensagens. É constituído por um núcleo - residente em cada processador do MCR - e por Monitores especificamente projetados para realizar a interface com os dispositivos de E/S, com os barramentos 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 prédefinida, 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 a 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 *blocos 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 nesta fila. Normalmente as mensagens são retiradas da fila na forma FIFO; no entanto, um processo tem flexibilidade para atender somente as men

sagens provenientes de um determinado processo fonte. Um processo pode ficar suspenso ou bloqueado à 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, este 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 isto, 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 as variáveis, tanto do núcleo dos Processos Aplicativos. O código das rotinas será 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 na ocorrência do evento, sendo interrompida a contagem de tempo, ou quando esgotar o tempo, o que significa 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, inserindo-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 à espera desta mensagem, ele deve ser ativado.

8) RECEBE

Retira uma mensagem da fila do processo. Em geral a primeira da fila é fornecida, a não ser que o processo requeira uma mensagem específica.

Caso não tenha sido encontrada a mensagem (fila vazia ou mensagem requerida não-encontrada), o processo é avisado.

9) ESPERA

Análoga à RECEBE, mas em caso de fila vazia ou mensagem requerida não-encontrada, 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 com 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 com as mensagens que lhe são destinadas. Cada mensagem estará contida em um bloco com 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* com 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 uma cabeça com 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.

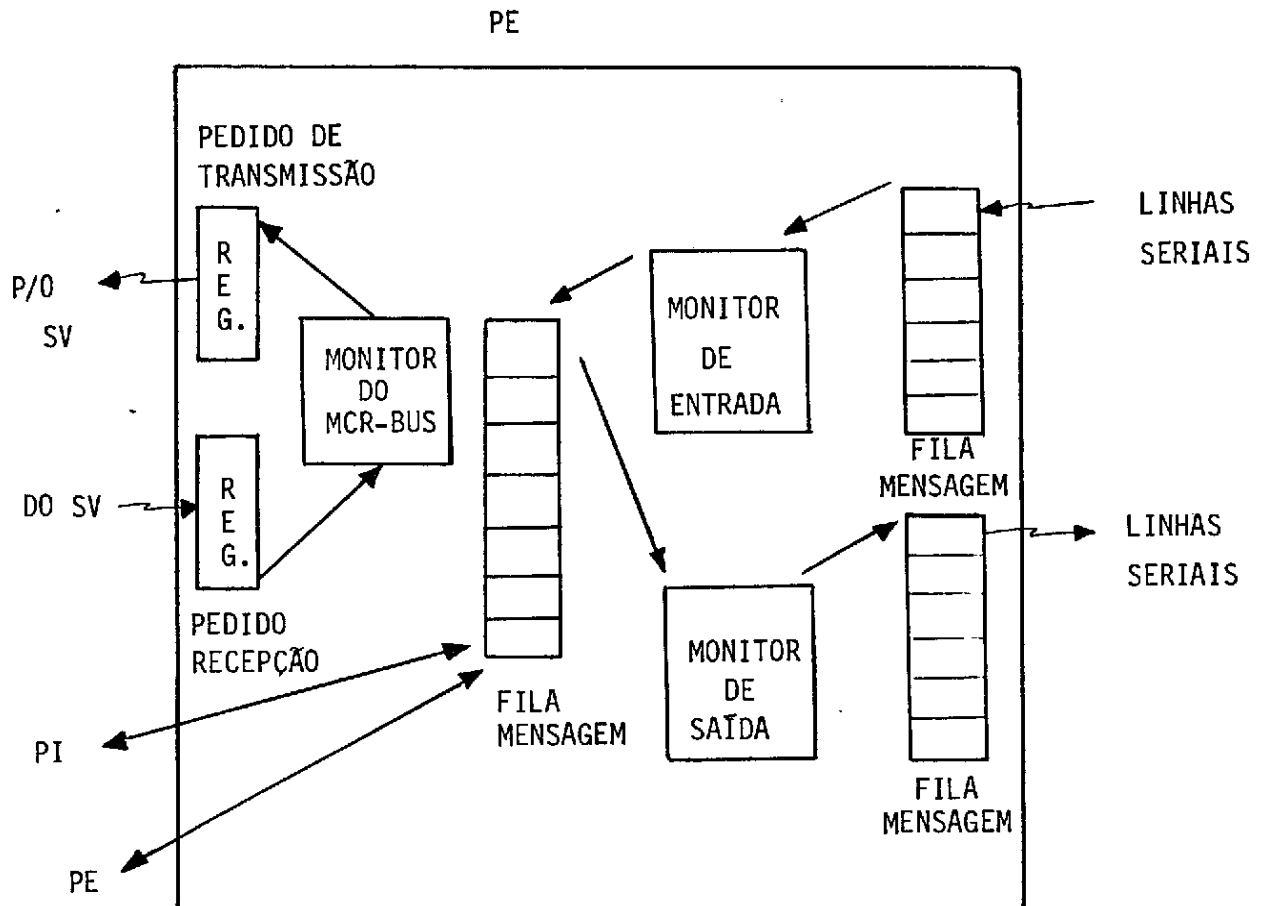


Fig. 4.2 - Diagrama de fluxo de mensagens na PE.

- *Monitor de Entrada:* é 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 à espera da interrupção do DMA que indica o 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 à espera de uma mensagem na sua fila. Da mesma forma que o Monitor de Entrada, este mantém uma tabela com o estado de cada canal. Várias transmissões podem ser ativadas por canais diferentes; o monitor fica à espera da interrupção do final de transmissão por um dado canal.
- *Monitor de MCR-BUS:* é responsável pela comunicação das PEs via barramento interno. Este monitor é ativado (passa ao estado de pronto) nas seguintes condições:
 - chegada de um pedido de recepção do SV,
 - mensagem da própria PE para ser transmitida a outra PE ou à PI.

MONITORES DO SV

O SV tem basicamente dois monitores:

- a) *Árbitro do MCRBUS:* controla o acesso ao barramento interno.
- b) *Monitor do MCR-C-BUS:* trata da comunicação entre o SV e o controlador de MCRS. O SV comporta-se, 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), veja a Figura 4.3.

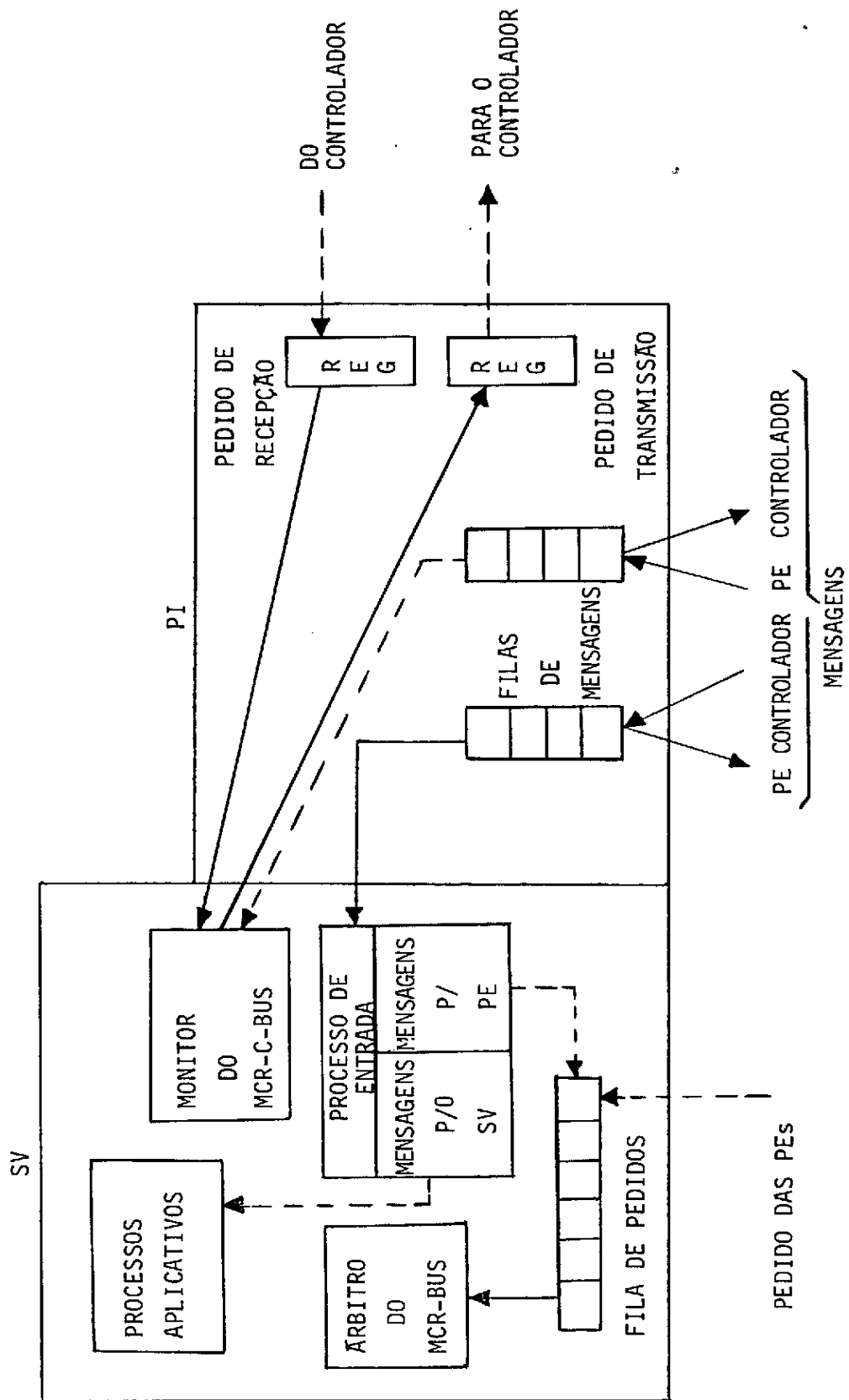


Fig. 4.3 - Diagrama de fluxo de mensagens na PI.

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

O *Árbitro do MCRBUS* é 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 transferên
cia pelo barramento interno.

As funções do *Monitor do MCR-C-BUS* são análogas às do Mo
nitor do MCRBUS das PEs. É ativado nas seguintes condições:

- pela rotina que atende os registros de comunicação da PI, indican
do que o controlador do MCR deseja transferir;
- caso haja mensagem em sua fila (Monitor do MCR-C-BUS), esta deve ser transmitida ao controlador de MCR.

5. CONCLUSÃO

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

Todos os cuidados foram tomados para que o Sistema Opera
cional tivesse como características principais: a simplicidade e a mo
dularidade. Assim, da mesma forma que estas propriedades poderão faci
litar a distribuição eficiente dos processos aplicativos, o conjunto
poderá apresentar um comportamento aquê das necessidades mínimas dese
jadas. Como consequência, isto implicará revisões futuras onde a intro
dução de mecanismos, tal como "time-slice", tornar-se-á indispensável.

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. April 20, 1984.

texto de um Protocolo de Transporte para a Rede CEPINNE, submetido ao 30. Simpósio Brasileiro de Redes de Computadores, Janeiro, 1985.

[DEAT 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.

[DEC 79] DIGITAL EQUIPMENT CORPORATION - Terminals and Communications Handbook - 1979, capítulo 7, p. 145-159.

[DEC 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, K.A., CUNHA, F.R.F. - Computa para Conexão de um Computador DEC-10 a uma Rede Pública de comunicação de Dados, in: 2. Simpósio Brasileiro sobre Redes de Computadores, Anais, Campina Grande, 1984, p. 2.1 - 2.21.

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 multiprocessador de comunicação em Redes - MCR (nó de uma sub-rede de comunicação de dados) desenvolvido pelo INPE/CNPq.



Título

Um Sistema Operacional para o MCR

Autor

Eliane MARTINS, Maria de Fátima MATTIELLO-FRANCISCO, Antonio Ésio Marcondes SALGADO, Pedro de Paula SANTOS-JUNIOR

Tradutor

Editor

INPE-5596-PRE/5224

Origem

ETE/DSS

Projeto

Série

No. de Páginas

19

No. de Fotos

No. de Mapas

Tipo

☐ RPQ ☒ PRE ☐ NTC ☐ PRP ☐ MAN ☐ PUD ☐ TAE

Divulgação

☒ Externa ☐ Interna ☐ Reservada ☐ Lista de Distribuição Anexa

Periódico / Evento

III Simpósio Brasileiro sobre Redes de Computadores

Rio de Janeiro, 10 a 12, 1985

Convênio

Autorização Preliminar

___/___/___
Data

Edenilson F.E. Orlandi

Chefe da Unidade de

Desenv. de Sistemas de Solo

Revisão Técnica

☐ Solicitada

☒ Dispensada

Recebida

___/___/___

Devolvida

___/___/___

Leonilso Fernando Perondi
Eng. Titular de Nível "A" - Área de Especial

Assinatura do Revisor

Revisão de Linguagem

☐ Solicitada

☒ Dispensada

Recebida

___/___/___

Devolvida

___/___/___

Edenilson F.E. Orlandi
Titular de Nível "A"

Assinatura do Revisor

Autorização Final

___/___/___
Data

Edenilson F.E. Orlandi

Chefe da Unidade de

Desenv. de Sistemas de Solo

Palavras Chave

sistema operacional - tempo real - protocolo de comunicação - rede - filas de processos - filas de mensagens



Folha 2/2

Observações	