

**Feira
Internacional
de Informática
1981
São Paulo
Brasil**

**São Paulo
16 - 23 outubro/1981**

**Anais do
XIV Congresso
Nacional
de Informática**

**Patrocinio
SUCESU Nacional
Realizacao
SUCESU São Paulo
Patrocinio Oficial
Secretaria Especial
de Informática**

**004.06
C749a
1981**

XIV CONGRESSO NACIONAL DE INFORMÁTICA

ANAIS DO XIV CNI

São Paulo
Outubro de 1981

A ARQUITETURA DO COMUTADOR DE PACOTES DA PUC/RJ

Ronaldo Roenick; Daniel A. Menascé
Pontifícia Universidade Católica (Deptº de Informática)
CEP 22.453 – Rio de Janeiro-RJ

Palavras-chave: Sistemas distribuídos, protocolo interno, multiprocessador, embarramento global, controle do acesso ao embarramento, módulos de hardware, módulos de software, funções básicas de um comutador, processos, comunicação entre processos, controle da configuração.

Resumo: Este trabalho descreve a proposta inicial de arquitetura de hardware e de software do comutador de pacotes da PUC/RJ, segundo projeto contratado pelo CPqD/Telebrás. Trata-se de um sistema distribuído, baseado em múltiplos microprocessadores fracamente acoplados e com controle de configuração descentralizado, que pode ser dimensionado, dentro da mesma filosofia de funcionamento e utilizando os mesmos módulos de hardware e de software, para o atendimento dos requisitos mínimos de tráfego a baixo custo. Além da capacidade de configurar pequenas centrais com relação custo/desempenho atrativa, a arquitetura adotada permite a expansão, no campo, do sistema de maneira a ampliar o seu desempenho de modo simples e barato; e a adoção de redundância interna, que produz alto grau de confiabilidade.

São apresentados, ao longo do texto, os aspectos básicos das estruturas de hardware e de software e do protocolo interno de comunicação entre os vários módulos constituintes do sistema.

I. INTRODUÇÃO

Este trabalho pretende detalhar as arquiteturas de hardware e de software, da central de comutação de pacotes que vem sendo desenvolvida, em convênio com a Embratel e o Centro de Pesquisas da Telebrás, no Departamento de Informática da PUC/RJ.

A definição da arquitetura de hardware baseou-se numa seleção entre 12 arquiteturas possíveis, segundo 9 itens de avaliação (ROEN81), que apontou três arquiteturas como as mais adequadas, de acordo com critérios previamente estabelecidos, para a implementação do comutador:

- sistema com memória compartilhada via mapa de páginas (exemplo: TELENET TP4000);
- sistema com módulos independentes ligados em anel (exemplo: SESA DPS-25, FDTE/FEPASA);
- sistemas com módulos independentes ligados por meio de um embarramento global.

A partir de algumas definições iniciais, tais como:

- a arquitetura deveria empregar uma estrutura fracamente acoplada;
- o sistema deveria ser modular de modo a possibilitar, a custo compatível, uma configuração adequada para cada situação de tráfego;
- o sistema deveria ser expansível no campo a baixo custo e de modo simples;
- os esquemas de funcionamento e os componentes empregados deveriam ser simples e confiáveis;
- a arquitetura deveria favorecer o emprego de redundância e a realização de reconfigurações automáticas com o mínimo de influência sobre a

A definição da arquitetura básica de hardware convergiu para uma tentativa de conciliação dos aspectos mais positivos dos sistemas em anel e com embarramento global. Em especial, a comunicação direta do embarramento global e a simplicidade do controle do acesso ao embarramento e da configuração interna apresentados nos sistemas em anel.

II. ARQUITETURA BÁSICA

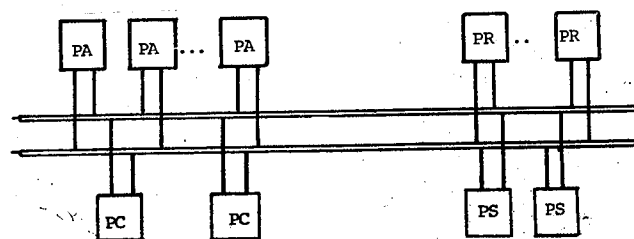


Figura 1. Arquitetura básica do comutador de pacotes.

A estrutura básica da arquitetura proposta consiste em dois embarramentos redundantes, e de quatro tipos de módulos, que se conectam aos embarramentos através de interfaces independentes. Embora a duplicidade de embarramentos se justifique por requisitos de confiabilidade, a operação nos embarramentos pode se dar independente e simultaneamente.

São as funções aplicativos básicas de cada tipo de módulo:

1) Processador de Acesso - PA

- Implementar X.25 nível 1 (físico);
- Implementar X.25 nível 2 (lógico);

reinicialização de chamadas e encaminhá-los aos PCs;
 - Manter, a partir de informações geradas pelos PCs, tabelas de encaminhamento de pacotes de dados;
 - Implementar a parte de transferência de dados do X.25 nível 3.

2) *Processador de Comutação - PC*

- Estabelecer, encerrar e reinicializar circuitos virtuais e transferir aos processadores responsáveis pelo encaminhamento dos pacotes de dados as informações referentes às tabelas de encaminhamento;
 - Processar a validação/negociação de facilidades no estabelecimento das chamadas.

3) *Processador de Rede - PR*

- Implementar o nível 1 do protocolo interno da rede;
 - Implementar o nível 2 do protocolo interno da rede;
 - Identificar pacotes de estabelecimento, encerramento e reinicialização de chamada e encaminhá-los aos PCs;
 - Manter, a partir de informações geradas pelos PCs, tabelas de encaminhamento de pacotes de dados;
 - Implementar a parte de transferência de dados do nível 3 do protocolo interno da rede.

4) *Processador de Supervisão - PS*

- Monitorar o estabelecimento, o encerramento e o tráfego de cada circuito virtual;
 - Enviar informações de tarifação ao centro de controle da rede (CCR);
 - Controlar a configuração do comutador;
 - Efetuar interfaceamento com periféricos;
 - Efetuar interfaceamento (remoto) com o CCR.

Certamente, funções relacionadas à comunicação entre os processos; à execução de testes, estatísticas, relatórios e alarmes; e ao próprio supervisor de cada módulo deverão ser previstas na definição da arquitetura de software do sistema.

O relacionamento existente entre os vários tipos de módulos está descrito na figura a seguir:

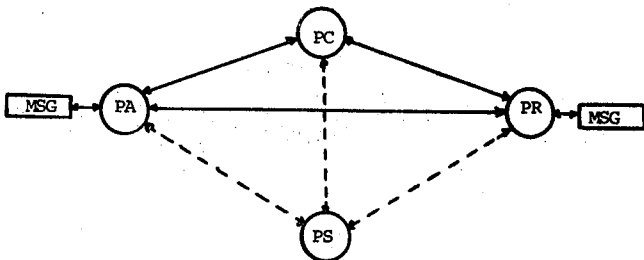


Figura 2. Relacionamento entre os módulos.

III. PROTOCOLO INTERNO DE COMUNICAÇÃO

Toda a comunicação entre os processos executados nos módulos é realizada por um protocolo de comunicação em três níveis: nível de interface, nível de ligação e nível de processo.

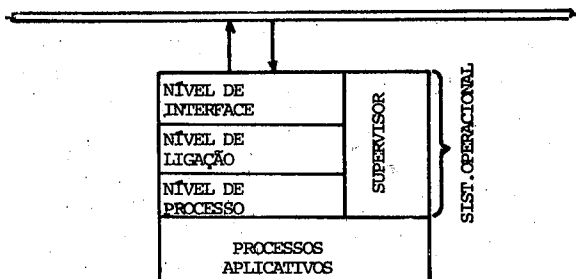


Figura 3.

No nível de interface são realizadas as funções relativas aos aspectos físicos da transmissão síncrona serial entre os módulos, destacam-se: transparência; endereçamento; temporização; validação das mensagens; verificação do estado do embarramento; deferência para transmissões; tratamento da transferência do controle do embarramento; detecção de interferência e recepção de mensagens.

No nível de ligação são executados os procedimentos de controle do acesso - cíclico - ao embarramento, em especial: gerenciamento da lista de controle do ciclo; inicialização/reinicialização; tratamento de omissões de transmissão e tratamento do intervalo de contenção.

Finalmente, no nível de processo, são executados os procedimentos de intercomunicação e sincronização dos processos: tratamento das funções de controle; reconhecimento e controle do fluxo.

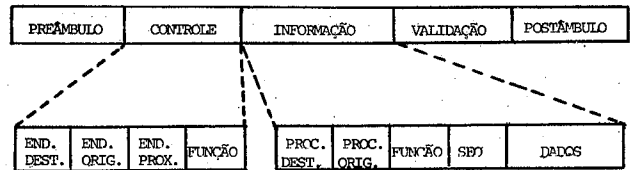


Figura 4. Formato das mensagens.

A troca de mensagens e o controle do ciclo estão baseados em três registradores de cada módulo (RTK, RID e RSQ) e em três campos do cabeçalho das mensagens que transitam pelo embarramento (END PROX, END ORIG e END DEST).

A figura 5 descreve funcionalmente o módulo, destacando a interface de comunicação e o seu relacionamento com o processador e com o embarramento.

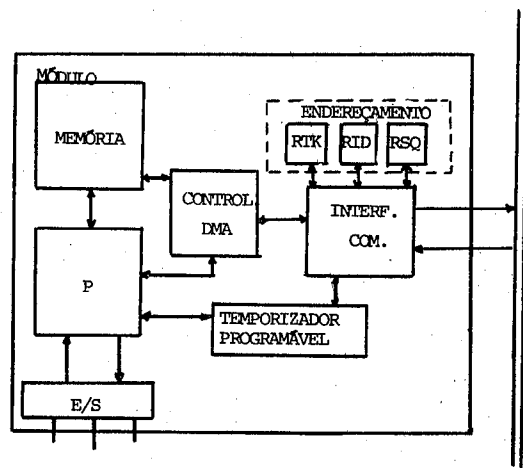


Figura 5. Diagrama esquemático de um módulo.

O nível de ligação estabelece, juntamente com o nível de interface, uma estrutura básica de comunicação entre os vários módulos. Por motivos de simplicidade e de modo a evitar a execução de funções redundantes por níveis distintos, o nível de ligação não executa qualquer procedimento de confirmação de recepção de mensagens, nem de controle de seqüenciação ou fluxo de mensagens, assumindo que a estrutura de transporte das

são tratados no nível de processo.

O controle do acesso ao embarramento por parte dos módulos está baseado no conceito de *lista de controle do ciclo*, que é o elemento responsável pela seqüenciação lógica das transmissões. A lista de controle do ciclo contém todos os endereços possíveis (64 ao todo) e associa a cada endereço de módulo o seu número de ordem no ciclo e o seu estado operacional (ativo ou inativo).

Gerenciamento da lista de controle do ciclo:

Entende-se como gerenciamento da lista de controle do ciclo a sua criação e a manutenção de sua atualidade e consistência, seja por inclusões ou exclusões de módulos, de modo que ela sempre indique, em todos os módulos ativos, quantos e quais são os módulos ativos em um dado instante e qual o seu interrelacionamento lógico para fins de seqüenciação do acesso ao embarramento. As inclusões/exclusões são processadas alterando-se, em cada lista, o estado do módulo de inativo/ativo para ativo/inativo e reordenando-se a lista; isto é, calculando-se os novos números de ordem dos módulos com endereço superior ao do incluído/excluído.

Inicialização: Cada módulo, ao ser ligado ou ao recuperar-se de uma falha interna, executa um procedimento de inicialização que consiste, basicamente, da observação da existência, ou não, de mensagens no embarramento, tomando por base uma lista de controle do ciclo internamente criada com todos os módulos (endereços possíveis) ativos. Neste ponto, cada módulo fica aguardando que algum dos módulos anteriores (de menor número de ordem que o seu) transmita, sugerindo uma configuração de ciclo e ativando a ciclagem conforme o procedimento de reinicialização descrito a seguir. Caso ocorra alguma transmissão, o módulo em questão entende que já há um ciclo estabelecido e aguarda que o controle lhe seja passando diretamente, por um endereçamento específico, ou indiretamente, via um intervalo de contenção — a ser discutido em item específico.

Reinicialização: Nas situações em que parem dúvidas sobre a real situação do ciclo de acesso ao embarramento — por exemplo, após a inicialização do sistema, ou após a ocorrência de uma omissão de transmissão, ou após a ocorrência de interferência entre duas ou mais transmissões — cada módulo passa a executar um procedimento de reinicialização; isto é, aguarda que algum dos módulos anteriores transmita uma mensagem, denominada *voto*, sugerindo uma configuração do ciclo. A figura 6 apresenta o formato de uma mensagem votando uma configuração do ciclo. Nesta mensagem o campo *endereço de destino* designa um endereçamento global, o campo de controle com o valor hexadecimal '01' indicando tratar-se o campo de informação de um voto, e o campo de informação composto de 64 bits, onde cada bit indica o estado 1, ativo; 0 inativo — do módulo associado a cada um dos 64 endereços possíveis de serem formados com os 6 bits do campo de endereçamento.

PRÉAMBULO	77 ₈	END. PROX.	01 ₁₆	VOTO	VALID.	POSTÁMBULO
-----------	-----------------	------------	------------------	------	--------	------------

64 bits

Figura 6. Formato de um voto.

A cada ocorrência de um esgotamento da temporização, o módulo de número de ordem mais baixo é excluído da lista, de modo que, após a ocorrência de n_k esgotamentos da temporização — onde n_k é o endereço em binário do módulo acrescido de uma unidade — o módulo em questão transmitirá uma mensagem composta de 64 bits votando numa configuração de tabela onde cada bit definirá o estado do módulo ao qual sua posição no voto está associada, conforme a tabela corrente na sua memória. Todos os demais módulos criarão, a partir dessa mensagem, tabelas idênticas nas suas memórias que servirão de base comum para o início da ciclagem.

O anexo 1 apresenta o fluxograma dos procedimentos de inicialização/reinicialização.

Como medidas de agilização do estabelecimento de uma lista de controle definitiva e comum a todos os módulos, (1) os primeiros endereços devem ser reservados aos processadores de supervisão de modo que eles sejam os primeiros a efetuarem votos e (2) os processadores de supervisão devem ser dotados de informações sobre a real configuração do sistema (endereços dos módulos instalados gravados em ROM, disco flexível ou definido por 'straps') visando ao estabelecimento de uma lista inicial que não considere endereços de módulos que não estejam instalados.

Manutenção da integridade do ciclo: Cada módulo testa continuamente a sua permanência na lista de controle do ciclo que está sendo utilizada no sistema. Sempre que algum módulo perceber a ocorrência de dois ciclos consecutivos — através do intervalo de contenção — sem que, neste período, ele tenha recebido explicitamente o controle do embarramento, o módulo em questão assume que foi retirado indevidamente da lista e executa o procedimento descrito no anexo 1.

Tratamento de omissões e transmissões: Sempre que ocorrer um esgotamento da temporização, significando que algum módulo não transmitiu após ter recebido o controle do embarramento, o nível de ligação é ativado em todos os nós para a execução do procedimento de reinicialização, após a exclusão da lista do módulo cujo endereço estiver contido no registrador RTK. No módulo em que os conteúdos dos registradores RTK e RSQ forem idênticos, o procedimento alterará o conteúdo do RSQ carregando o endereço do novo próximo módulo da lista (ver anexo 2). Caso ocorra uma transmissão com erro ou que se encerre com uma seqüência de aborto, ela é descartada e o mesmo procedimento é executado, com a diferença de que o módulo cujo endereço está contido no registrador RTK não é retirado da lista. A distinção entre ausência de transmissão abortada é realizada na recepção dos módulos por uma palavra de "sense/status".

Tratamento do intervalo de contenção: O intervalo de contenção é um mecanismo projetado para possibilitar a inclusão de módulos num ciclo já estabelecido. Toda vez que um módulo, ao executar um procedimento de inicialização ou de reinicialização, detectar a existência de transmissões no embarramento assumirá que já há um ciclo estabelecido e aguardará uma oportunidade de integrar-se ao ciclo através de um intervalo desse tipo. O intervalo de contenção é o último intervalo de cada ciclo e segue as seguintes regras (ver anexo 3):

1) É ativado pelo último módulo do ciclo por

meio de uma mensagem com o conteúdo do campo endereço do próximo com o valor 778;

2) O intervalo é encerrado quando:

2.a) a temporização se esgota sem que ocorra qualquer transmissão;

2.b) uma única transmissão ocorre, sem que sofra qualquer interferência, fazendo com que um único módulo seja incluído em todas as listas;

2.c) dois, ou mais, módulos tentam a inclusão gerando interferência entre as mensagens.

3) Sempre que ocorre uma interferência durante o intervalo de contenção, é executado um procedimento de reinicialização.

O nível de processo engloba as funções de controle da comunicação entre os vários processos residentes nos diversos módulos do sistema e sua função básica é isolar, dos processos de aplicação, as atividades de seqüenciação das mensagens, controle de fluxo, controle de erros e recuperação por retransmissão. Deste modo o relacionamento entre os processos de aplicação e o protocolo de comunicação (sistema operacional) se reduz a alguns primitivos do tipo TRANSMITA, RECEBA, INFORME ESTADO, etc; e fica independente do funcionamento dos esquemas físicos e lógicos de intercomunicação dos módulos.

Uma preocupação existente na definição do nível de processo foi a de prover um alto grau de flexibilidade visando a atender uma ampla gama de características de tráfego. Isto foi implementado através do emprego de dois tipos de datagrama: as cartas, sujeitas a seqüenciação e controle de fluxo por janela; e os telegramas, que são mensagens auto-contidas, numeradas e sem controle de fluxo por janela. As várias mensagens trocadas entre processos ativos em dois módulos distintos são multiplexados por meio do par de códigos de processo, que equivale à identificação de um canal lógico, através de um protocolo do tipo de datagrama. Deste modo, não há qualquer procedimento de estabelecimento ou de encerramento das ligações entre os processos/módulos uma vez que há um código de circuito lógico previamente estabelecido de modo estático para cada par de processos.

Este fato exige certo conhecimento prévio, por parte de cada processo, dos códigos dos processos com os quais ele mantém algum relacionamento (troca de mensagens). Trata-se de uma restrição perfeitamente tolerável para a aplicação específica considerada, de um computador de pacotes. As características de operação são programáveis pelos processos aplicativos e aparecem explicitadas no campo de função das mensagens do nível de processo.

MODO	TIPO	REINIC.	ACEIT./REJ.	LIVRE/OCUP.	REQ. ESTADO	INF. ESTADO	FIM DE TEXTO
------	------	---------	-------------	-------------	-------------	-------------	--------------

Figura 7. Bits que compõem o campo de função.

IV. DESCRIÇÃO BÁSICA DE HARDWARE E DE SOFTWARE

Os anexos 4 e 5, respectivamente, apresentam as descrições básicas de hardware e de software para o computador da PUC/RJ. Destacam-se como características marcantes da arquitetura proposta;

quanto ao hardware:

confiabilidade e eficiência apresentado pelos mecanismos e componentes empregados;

2) A alta modularidade e a grande facilidade de reconfiguração, tanto física como lógica.

quanto ao software:

1) A modularidade funcional, que permite a atualização e a inclusão de novos módulos de software de maneira simples e eficiente;

2) O esquema de distribuição de funções que facilita a reconfiguração automática e possibilita as várias configurações do sistema de acordo com os requisitos de tráfego.

V. CONCLUSÃO

Atualmente o grupo envolvido no projeto e implementação da central de pacotes acha-se realizando tarefas de modelagem e reavaliação da arquitetura proposta visando, basicamente, a estudar a possibilidade da adoção de novas soluções, através da simplificação de alguns mecanismos; e estimar melhor o desempenho do sistema diante das situações de tráfego previstas para a rede experimental do CPqD/Telebrás. Paralelamente, estão sendo estudadas arquiteturas de produtos já existentes para confrontação mais detalhada com a apresentada e iniciadas as atividades de implementação de placas básicas, de sistema de desenvolvimento e de preparação do pessoal envolvido no projeto.

Os dados iniciais de avaliação de desempenho foram satisfatórios, apontando uma pequena perda de eficiência pelo protocolo interno, e estão sendo estudadas as simplificações possíveis para o nó da PUC/RJ, procurando-se eliminar funções não-essenciais, concentrar funções em poucos processadores, etc.

Maiores detalhes sobre o sistema proposto podem ser obtidos em (ROEN81).

BIBLIOGRAFIA

(ROEN81)

Roenick, Ronaldo "Computador de Pacotes com Arquitetura Distribuída: Considerações e Especificação Preliminar" Dissertação de Mestrado - Departamento de Informática - PUC/RJ - Março 1981.

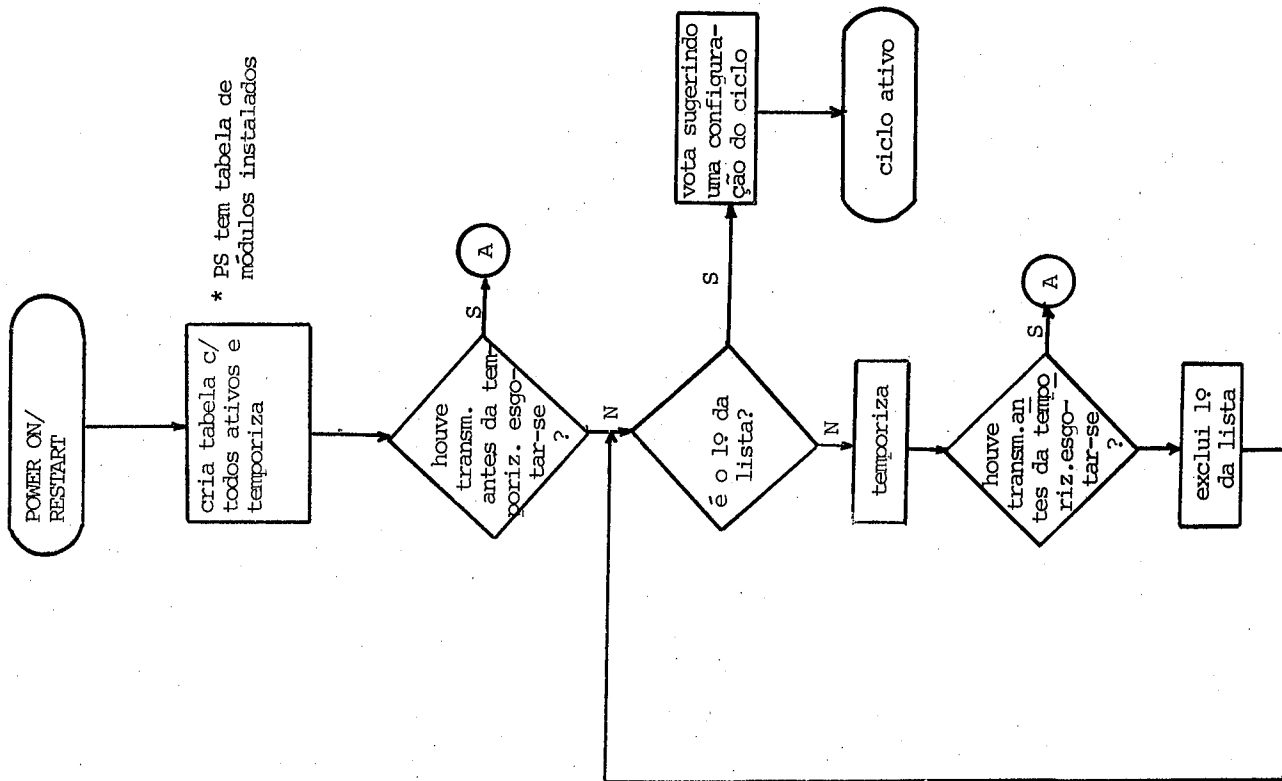
(WIET80)

Wietzman, C. "Distributed Micro/Minicomputer Systems", Prentice-Hall, New Jersey, 1980.

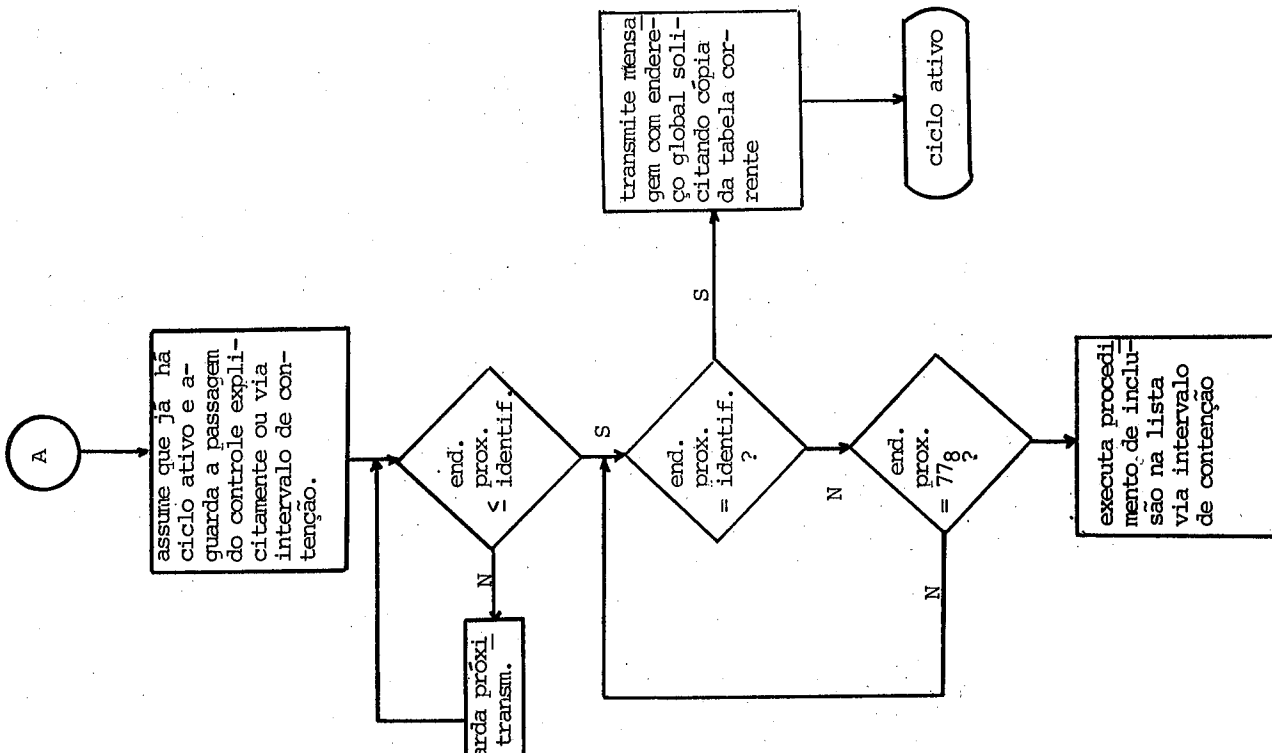
(ADAM78)

Adams, G.; Roiander, T. "Design Motivations for Multiple Processor Microcomputer Systems" Computer Design, March 1978, pp. 81-89.

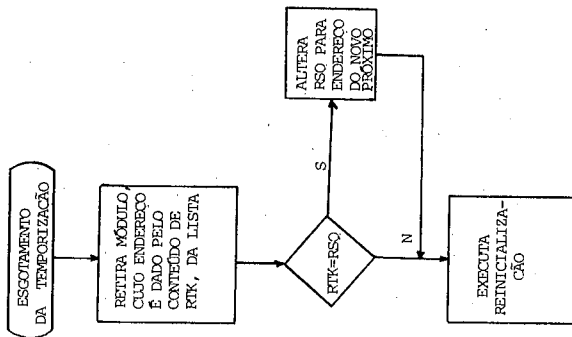
Anexo 1. Fluxograma dos procedimentos de inicialização/reinicialização



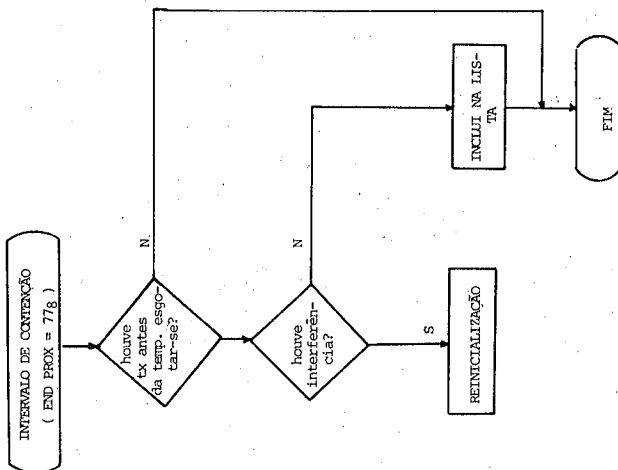
Continuação do fluxograma.



Anexo 2. Procedimento de tratamento de omissões de transmissão.



Anexo 3. Fluxograma do tratamento do intervalo de contenção.



Anexo 4. Descrição Preliminar de Hardware.

Podemos definir como um módulo, o hardware básico capaz de executar todas as funções de execução, supervisão e inter-comunicação dos processos existentes num comutador de pacotes. Um módulo é composto de seis placas básicas.

- (1) Fonte de Alimentação - FA
- (1) Processador Central - PC
- (1) Memória Principal - MP
- (1) Interface com o Embarramento - IE
- (2) Interfaces de Comunicação - IC.

Fonte de Alimentação. Cada módulo possui alimentação independente, visando a aumentar a confiabilidade do sistema e a diminuir a complexidade – provavelmente o custo – dos circuitos em relação ao que seria necessário para a implementação de uma fonte dual de alta capacidade.

São funções básicas desta placa:

- retificação da alimentação da rede;
- regulação do(s) nível(is) de tensão necessário(s);
- proteção contra curto-circuito;
- proteção contra sobre-carga;
- ativação de alarme de mau-funcionamento.

Processador Central. Neste cartão são executados os processos aplicativos do módulo, bem como todos os procedimentos de controle/supervisão da execução de tais processos através de um monitor de tempo real.

São funções básicas desta placa:

- processamento dos programas aplicativos;

- geração dos sinais de sincronismo;
- geração de temporização em tempo real;
- interfaceamento com o console do operador (linguagem homem-máquina);
- interfaceamento com dispositivo de memória secundário (disco flexível).

Memória Principal. Neste cartão são armazenados os dados e os programas do processador central. A memória principal também serve de meio de comunicação entre placas dos tipos IE, PC e IC, segundo um esquema de mailbox. São suas funções básicas:

- armazenar dados e programas do processador central;
- arbitrar os vários pedidos de acesso à memória;
- geração de alarme por falha (paridade de dados e de endereços);
- armazenamento – em ROM – do carregador de programas gravados no disco flexível.

Interface com o Embarramento. Este cartão é responsável pela comunicação entre os vários módulos constituintes de um sistema. Suas funções básicas são:

- processamento do nível de processo do protocolo de comunicação entre módulos;
- execução dos procedimentos do nível de interface do protocolo de comunicação entre módulos.

Interface de Comunicações. Há três opções de cartões de comunicação:

- cartão 2AVS
- cartão 8MVSA
- cartão 16BVA

O cartão 2AVS, serve como porta de rede e permite a ligação de até duas linhas síncronas de alta velocidade, protocolo HDLC, até uma vazão global máxima de aproximadamente 72 kbps.

O cartão 8MVSA serve como porta de acesso para 8 linhas de assinantes com terminais pacote (síncronos) ou não-pacote síncronos e assíncronos, até 9600 bps em cada linha.

O cartão 16BVA serve como porta de acesso para 16 linhas assíncronas de até 1200 bps para terminais não-pacote.

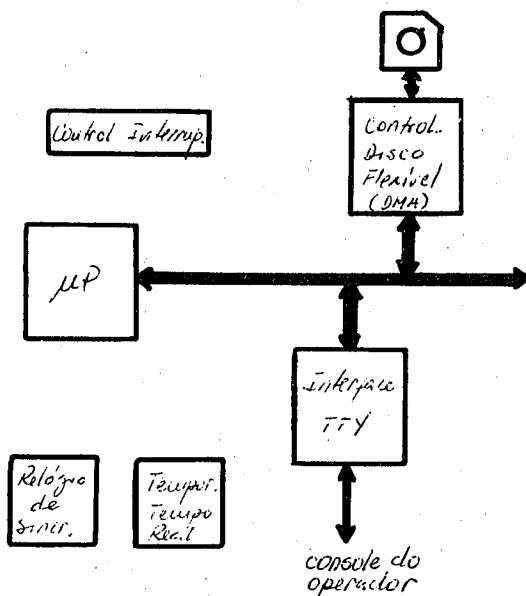




Figura 2. Cartão MP.

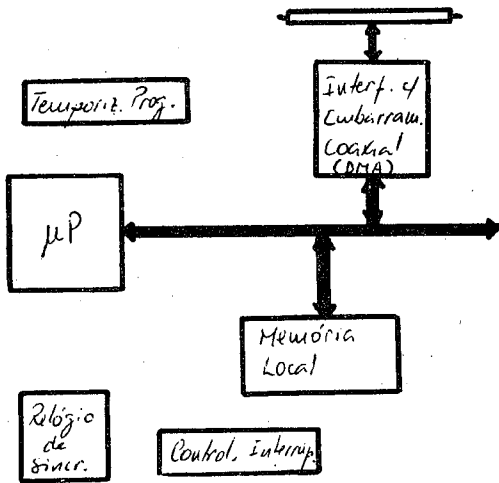


Figura 3. Cartão IE.

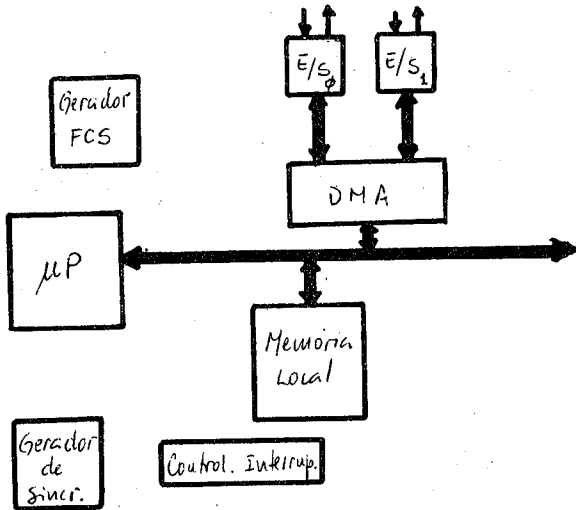


Figura 4. Cartão IC/2AVS.

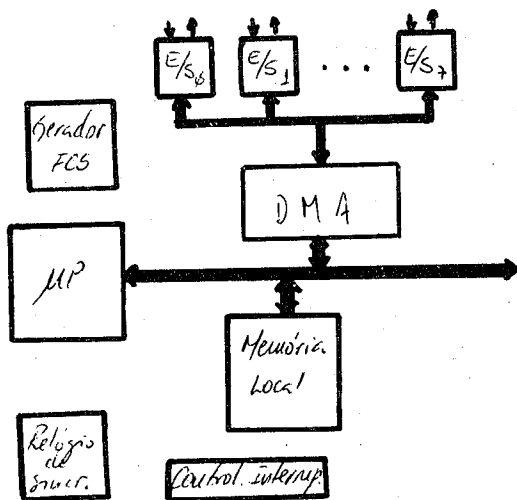


Figura 5. Cartão IC/8MVSA.

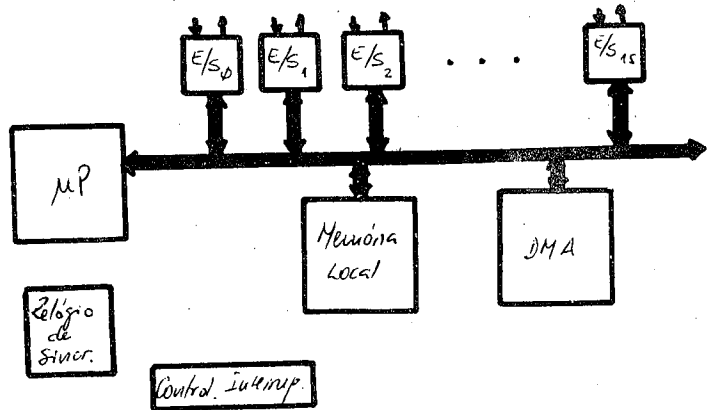


Figura 6. Cartão IC/16 BVA.

É importante destacar que os cartões IE e IC transferem dados entre sua memória local/seus periféricos e a memória principal através de um esquema de DMA. Há, portanto, dois níveis de processamento num módulo: o processamento em "front-end" executado pelo IE e pelos ICs; e o processamento em "back-end" executado pelo PC.

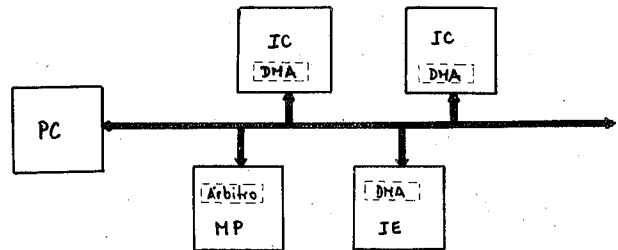


Figura 7. Interligação das placas de um módulo.

As figuras a seguir apresentam as configurações dos vários módulos previstos.

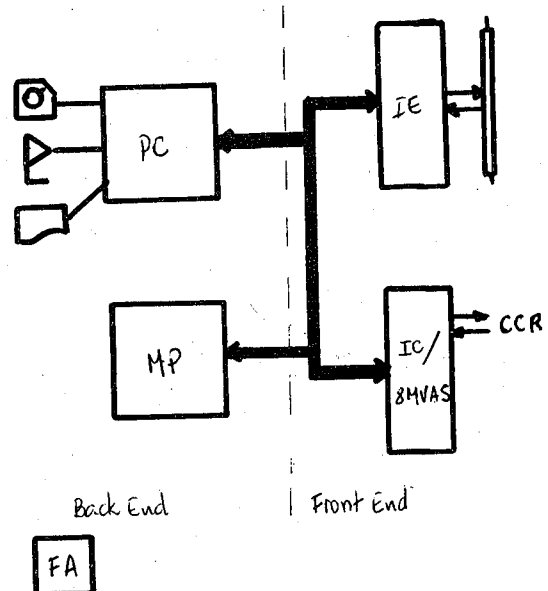


Figura 8. Módulo PS.

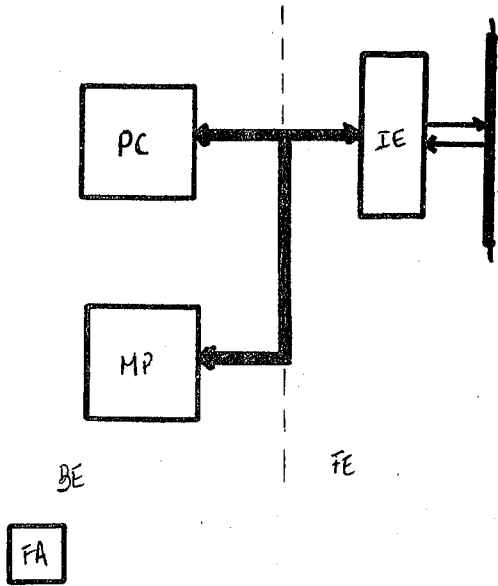


Figura 9. Módulo PC.

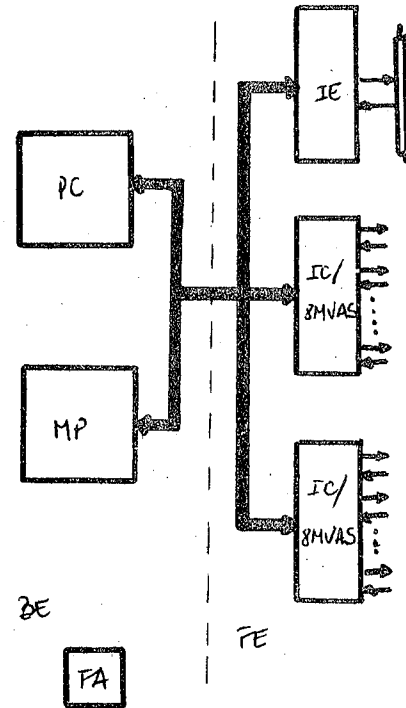


Figura 11. Módulo PA.

Anexo 5. Descrição Preliminar de Software.

a) Organização e Distribuição das Funções.

A organização e a distribuição das funções do comutador de pacotes seguem aquelas encontradas nas centrais tradicionais de telecomunicações onde controle (PS), comutação (PC) e acesso de linhas (PA e PR) são, de modo geral, partes independentes e distintas do equipamento.

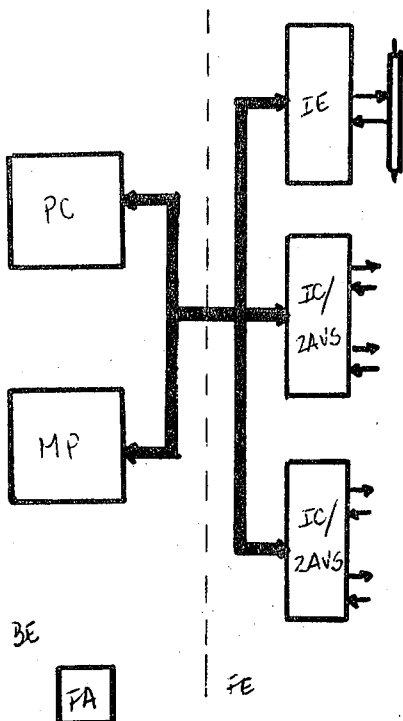


Figura 10. Módulo PR.

PROTOCOLO INTERNO

FUNÇÃO DE SUPERVISÃO (PS)	FUNÇÃO DE COMUTAÇÃO (PC)	FUNÇÃO DE ACESSO (PA, PR)
- Recuperação	- Roteamento	- Acessos de rede: X.75 ou protocolo próprio
- Protocolo de comunicação com	- Estatísticas e medidas	- Acessos de assinantes modo pacote: X.25 LAPB ou multilínea
- Tarifação	- Gerência dos circuitos	- Acessos de assinantes assíncronos: (não considerados no projeto) X.3, X.28, X.29
- Gerência dos Circuitos Virtuais		- Acessos de assinantes síncronos não-pacote: protocolo específico, módulo de conexão, estação de transporte. (idem)

Figura 1. Distribuição das funções no comutador.

As funções podem ser classificadas em (1) centrais, (2) distribuídas e (3) locais. As funções centrais acham-se, por sua própria natureza, concentradas num único módulo — por exemplo, as funções de supervisão. Aquelas funções que podem ser associadas a vários processadores — por exemplo, transferência de dados entre assinantes — são ditas distribuídas e cooperam

Finalmente, as funções que só têm significado no contexto interno de um módulo e não produzem qualquer efeito externo — por exemplo, gerência da memória local ao módulo — são ditas locais.

O comportamento de um módulo é determinado pela função principal que ele executa (supervisão, comutação ou acesso) e pela sua condição operacional (em operação ou reserva). O estado de um módulo é definido pelo estado do hardware (ativo ou inativo).

É atribuição da função de supervisão controlar a configuração do sistema, atribuindo comportamentos aos módulos de acordo com o seu estado e com a ocorrência de eventos em tempo real.

b) Módulos de Software.

Monitor de Tempo Real.

1. Controle da Execução Distribuída

- escalonamento dos processos do comutador;
- ativação/despacho dos processos do comutador;
- processamento do relógio de tempo real;
- controle do uso concorrente dos recursos;
- comunicação entre os processos.

2. Controle das Interfaces de Linha

- tratamento das interrupções;
- transferência das mensagens entre memória principal e linhas de comunicação;
- tratamento dos protocolos de linha.

3. Controle de Entrada e Saída

- tratamento das interrupções;
- transferência das mensagens entre memória principal e periféricos;
- controle dos periféricos específicos (disco flexível, terminal console e impressora).

4. Interface com o Operador

- processamento da linguagem de controle;
- interface homem-máquina (comandos, mensagens, alarmes, estatísticas e demais facilidades de operação e de supervisão).

5. Processamento e Diagnóstico de Falhas

- execução de testes sobre hardware e software básicos;
- detecção de falhas de hardware e software básicos e processamento de rotinas específicas de tratamento de falhas.
- gravação de registro (log) de falhas e ocorrências anormais.

6. Controle de Comunicação

- processamento do protocolo interno de comunicação entre módulos.

7. Utilitários e Programas de Suporte e Depuração

- bootstrapping do comutador;
- carregador para o software do comutador;
- editor;
- dump/trace/patch;
- linkage-editor;
- biblioteca de módulos.

Programas de Aplicação

1. Supervisão

- teste dos componentes do comutador;
- monitoração das comunicações entre os processos do comutador;
- supervisão da reconfiguração do nó;
- supervisão da operação do nó.

2. Recuperação

- reconfiguração do comutador;
- controle da comutação das funções;

- geração do software do comutador (escolher dos programas e inicialização das tabelas de aplicação — parâmetros de assinante inclusive).

3. Gerência dos Circuitos

- Gerência dos circuitos virtuais — permanentes e comutados — do nó;
- Gerência dos circuitos internos da rede.

4. Tarificação

- execução dos registros necessários ao cálculo da tarificação.

5. Interface com o Operador

- execução dos comandos do operador local;
- apresentação de dados, respostas, mensagens, alarmes e estatísticas ao operador local;
- processamento da linguagem de controle da operação.

6. Interface com o Centro de Controle da Rede.

- processamento do protocolo de comunicação com o CCR;
- processamento da linguagem de controle da operação;
- execução dos comandos do operador do CCR;
- apresentação de dados, respostas, mensagens, alarmes e estatísticas ao operador do CCR.

7. Registro de Ocorrências

- gravação (em disco flexível) de arquivo formado por todas as mensagens trocadas com o CCR;
- gravação (em disco flexível) de arquivo formado por todas as mensagens de erro, alarmes, estatísticas, etc.

8. Comutação

- processamento dos pacotes de sinalização (estabelecimento e encerramento de chamadas);
- implementação da comutação dos pacotes estabelecendo um circuito virtual.

9. Negociação de Facilidades

- gerência das tabelas dos usuários locais (contêm características de cada acesso/usuário).

10. Roteamento

- atualização da tabela de roteamento, com base nos dados de estabelecimento de chamada e de situação dos circuitos internos da rede;
- determinação da rota (próximo nó);
- detecção de “loops” de roteamento.

11. Controle dos circuitos

- controle dos circuitos virtuais do nó;
- controle dos circuitos internos da rede.

12. Medidas e Estatísticas

- aquisição de dados necessários à operação de nó (roteamento, retardo, etc.).

13. Protocolo de Rede (N.3)

- processamento do nível de pacote do protocolo interno da rede, sob o aspecto de transferência de pacotes de dados.

14. Protocolo de Rede (N.2)

- processamento de todo o nível lógico do protocolo interno da rede (controle de fluxo, controle de erro, transferência de informação, etc.);
- controle da comutação de interfaces;
- teste do ECD (interface; modem, controlador de linhas, etc.).

15. X.25 N.3/Multiline

- processamento do nível de pacote do protocolo de acesso X.25, sob o aspecto de transferência de pacotes de dados, e do procedimento “multiline” para usuários que, por motivos de maior confiabilidade, utilizarem mais de um circuito físico.

16. X.25 N2/LAPB

- processamento de todo o nível lógico do protocolo de acesso X.25, procedimento LAPB;
- controle da comutação de interfaces;
- teste de ECD (interface, modem, controlador de linhas, etc.).

17. PAD

- conversão de mensagens/pacotes em pacotes/mensagens;
- controle das características do assinante (X.3, X.28, X.29).

18. Start/Stop

- processamento de acesso assíncrono em protocolo teletipo.

c) Arquitetura de Software

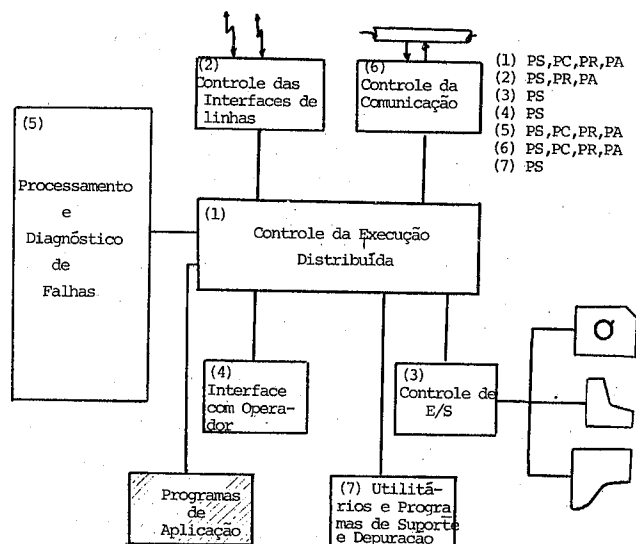


Figura 2. Sistema Operacional.

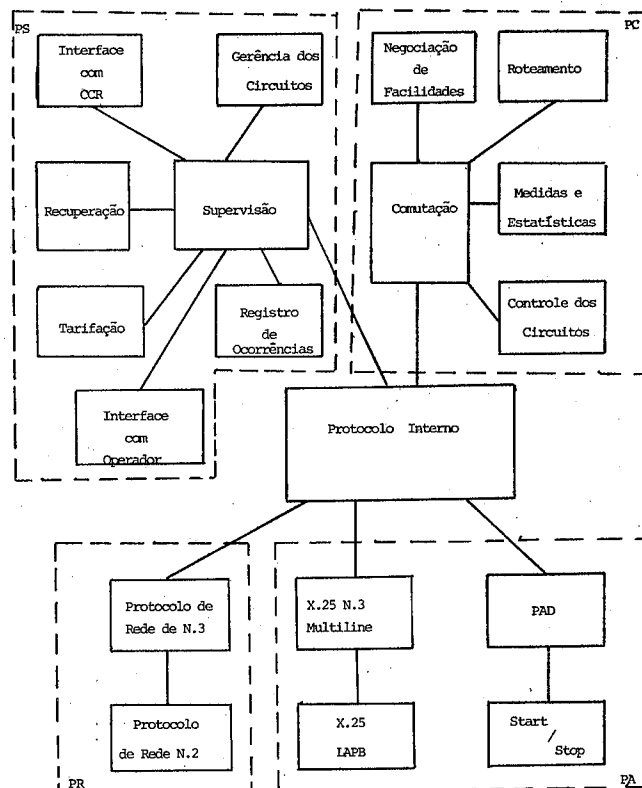


Figura 3. Programa de Aplicação.