

EXPERIÊNCIA NO PROJETO E IMPLEMENTAÇÃO
DA FEDE LOCAL REDPUC

Daniel A. Menascé
Departamento de Informática, PUC/RJ

Luiz Fernando Gomes Soares
Departamento de Engenharia Elétrica, PUC/RJ

Daniel Schwabe
Departamento de Informática, PUC/RJ
22453 Rio de Janeiro, RJ

1. Introdução

No início de 1981, teve início na PUC/RJ o projeto de uma rede local de microcomputadores. Esta rede, que seria confinada, isto é, localizada em um único "rack", serviria de base à implementação de uma central de comutação com arquitetura distribuída [1]. Após o término da especificação preliminar, chegou-se à conclusão que o projeto da rede local deveria ser transformado no objetivo principal, em detrimento do término da central de comutação. Esta mudança de ênfase pode ser justificada por três razões. Em primeiro lugar, o Brasil já estava importando tecnologia na área de comutação por pacotes e o CPqD da Telebrás já estava desenvolvendo uma central de comutação para redes públicas. Em segundo lugar, o projeto preliminar da rede local havia se mostrado bastante promissor e interessante. Finalmente, uma rede local seria um produto de maior utilidade e aplicabilidade; uma vez concluído, possibilitaria a implementação de uma rede local não confinada no campus da PUC/RJ.

A partir daquela época, os esforços voltaram-se para o projeto detalhado e implementação do hardware e software da rede local, que passou a chamar-se REDPUC.

Uma primeira decisão que foi adotada logo no início do projeto foi a de que toda a tecnologia do projeto seria desenvolvida na PUC/RJ, evitando-se assim copiar ou adaptar

tros tipos de redes desenvolvidas no exterior. O esforço culminou no desenvolvimento de todos os protocolos da REDPUC, e no projeto do hardware e software que os implementam.

A REDPUC e seus protocolos já foram descritos em artigos recentes dos mesmos autores [2,3]. Este artigo tenta apresentar um relato de experiência obtida na sua implementação, ressaltando alguns aspectos que podem vir a servir de subsídios a outros grupos que atuam na mesma área.

A seção dois revê brevemente a arquitetura da REDPUC e seus protocolos. A seção 3.1 descreve a implementação do nível dois do protocolo de acesso ao embarramento, evidenciando a relação entre o hardware e o software. A próxima subseção apresenta o Núcleo de Sistema Operacional do processador de comunicações (PC) responsável pela implementação do Protocolo de Acesso ao Embarramento (PAE).

A implementação do nível três é ilustrada na seção 3.3 e o ambiente de implementação é descrito na seção quatro. Finalmente, o artigo conclui com uma descrição dos planos de pesquisa para os próximos anos.

2. Arquitetura da REDPUC

REDPUC é uma rede local utilizando como sistema físico de interconexão até dois embaramentos globais seriais compartilhados pelos diversos processadores de comunicação (PC). Estes podem ser desligados de um ou ambos os embaramentos em condições de falhas irrecuperáveis, neles mesmos, ou na própria estação (MB+ME - vide figura 1). A estrutura da rede é apresentada na figura 1.

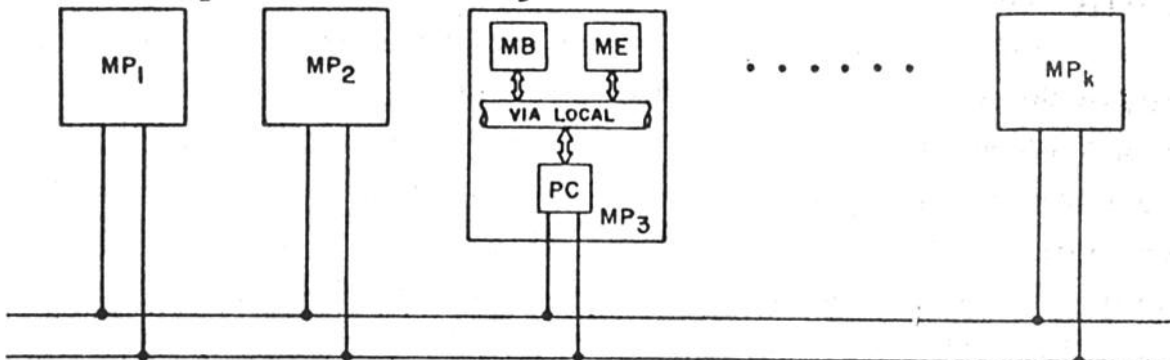


Figura 1 - A Estrutura da REDPUC

A comunicação entre os módulos é realizada por um protocolo de comunicações (Protocolo de Acesso ao Embarramento-PAE) que compreende três níveis independentes entre si.

O primeiro nível, nível físico, especifica as características físicas e elétricas do PAE. As características importantes deste nível, vista pelo nível imediatamente acima são: transmissão serial, síncrona, full-duplex em um sistema de comunicação do tipo difusão (broadcast) usando técnica de codificação Manchester.

O segundo nível, nível de embarramento, é o responsável pela ligação e desligamento de um módulo na rede, bem como do controle de acesso ao embarramento e detecção de erros. Para tanto utiliza um protocolo do tipo passagem de permissão ("token passing") com o controle realizado por uma lista denominada lista de controle de ciclos [2, 4]. Este nível implementa um serviço do tipo datagrama não confiável [5] , pois não realiza controle de erro, apenas existe uma grande probabilidade que a mensagem chegue a seu destino sem erro.

O terceiro nível, nível de transporte, converte a linha física propensa a erros, em uma ligação lógica, relativamente livre de erros, para a transferência de dados entre o transmissor e o receptor. Mais ainda, possibilita ao usuário, caso deseje, a transformação da ligação lógica em vários canais lógicos, realizando a função de um concentrador. Este nível realiza o controle de erro e de fluxo nos vários canais lógicos, bem como a sequenciação de mensagens, se desejada pelo usuário, nos mesmos. Este nível vai implementar desta forma dois tipos de serviço: datagrama confiável e circuito virtual.

Cabe aos processadores de comunicação (PC's) a realização do protocolo de comunicação PAE, bem como a interface com a estação. Cada PC possui capacidade de processamento tornando-o adequado não só para ligação com estações inteligentes, mas também para outras funções tais como serviço específico para rede (p.ex., estação de impressão) ou ligação com dispositivos pouco inteligentes (p.ex., ligação de ter-

minais à rede), entre outras.

3. Arquitetura de Software da REDPUC

3.1 - PAE2 - Interação hardware/software

A figura 2 apresenta um diagrama simplificado do Processador de Comunicação (PC), responsável pela interface com a estação usuária e pela realização do protocolo de acesso ao embarramento

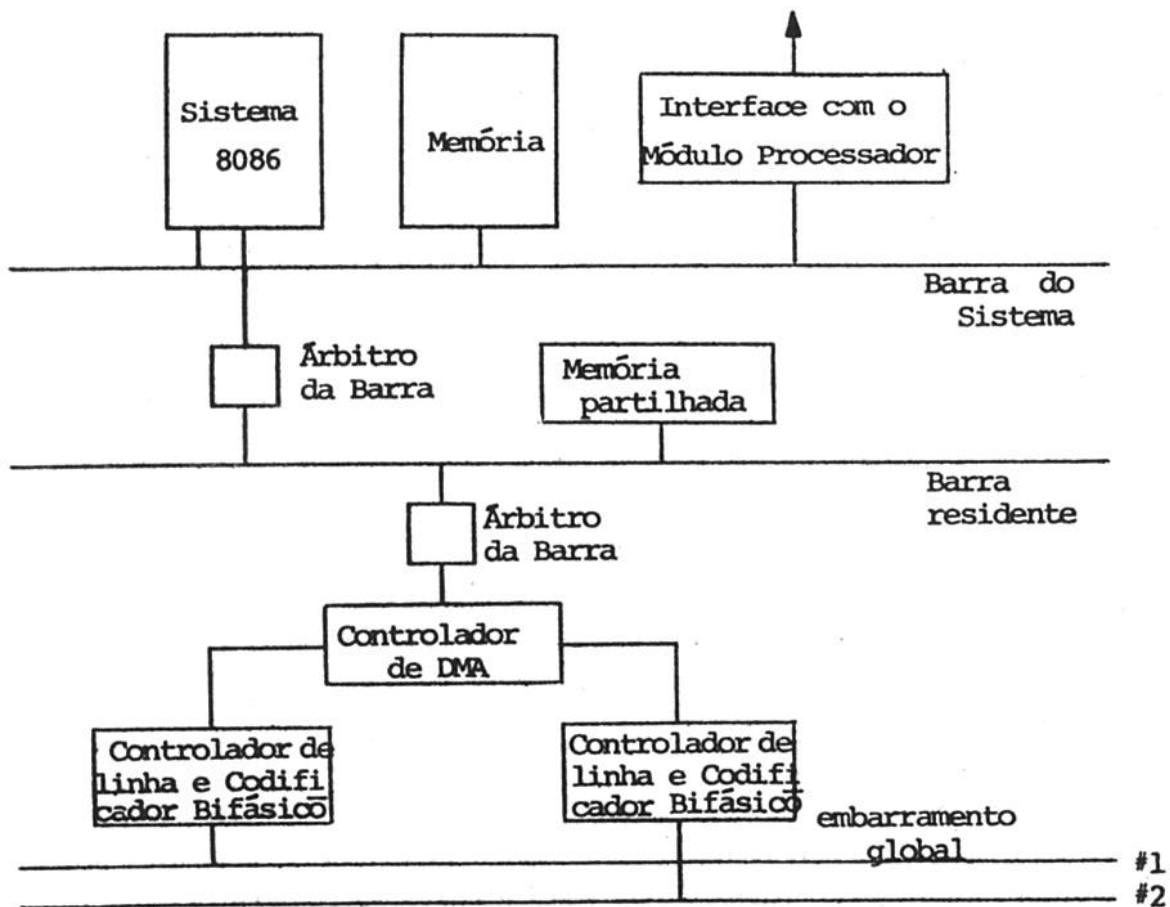


Figura 2 - Diagrama simplificado do Processador de Comunicação

A primeira dificuldade no projeto do PC foi a determinação de que tarefas seriam realizadas por software e quais seriam realizadas por hardware. Esta dificuldade resultou principalmente pela falta de circuitos integrados adequados ao projeto, e pelas altas taxas de transferência exigidas,

pois cada embaraço deveria poder aceitar uma taxa de até 10 Mbps.

A ligação ao embaraço foi realizada através do controlador de comunicação 2652-1 da Signetics e de circuitos para sincronismo, codificação e decodificação bifásica. A escolha deste controlador se deu pelo fato dele ser na época do projeto, o de maior velocidade encontrado no mercado (2 Mbps). Esta escolha trouxe como consequência um hardware adicional complexo e a introdução de um preâmbulo desnecessário ao quadro de nível 2 do PAE.

Um outro problema que se apresentou foi o da escolha do processador responsável, entre outras tarefas, pela realização do nível 3 e parte do nível 2 do PAE. A ligação do controlador de comunicação (2652-1) a este sistema poderia ser realizada de três modos: memória chaveada, por DMA e memória partilhada.

Na memória chaveada, um banco de memória ficou alocado a um só sistema por vez (2652-1 ou processador) podendo ser chaveada de um sistema para outro. Esta solução foi descartada pelo grande número de circuitos integrados que ela exigiria para sua realização.

A solução por DMA é inviável uma vez que a latência no atendimento ao pedido de um ciclo, bem como a realização deste ciclo de DMA, poderia gastar um tempo bem maior do que a chegada de dois caracteres a serem transferidos.

A solução de memória partilhada foi a adotada, uma vez que praticamente elimina o tempo de latência no atendimento de um DMA. Tal solução influenciou diretamente na escolha do processador. Sendo esta memória partilhada pelo 2652 através de DMA e pelo processador, ver figura 2, é necessário que o tempo de acesso à mesma seja o menor possível. A escolha de um processador de 16 bits teve aí seu fundamento e recaiu no 8086 da INTEL por três motivos: primeiro que o sistema 8086 foi desenvolvido para uma arquitetura "multibus" que facilita a realização da memória partilhada, segundo porque se sabia que em breve estaria no mercado o microprocessador 80186 que reduziria em muito o empacotamento do siste

ma; finalmente, porque vários pacotes de software desenvolvidos e que se utilizariam na implementação (p.ex., compilador C) existiam para o sistema 8086.

A realização de parte do nível 2 do PAE no processador 8086 trouxe implicações no software. As rotinas de atendimento das interrupções do PAE2 deveriam ser bastante eficientes e curtas, devido aos requisitos do protocolo. O problema surge do fato que nestas rotinas vários testes devem ser realizados para a escolha do procedimento adequado. O sistema 8086 só pode realizar estes testes sequencialmente o que ocasionaria um dispêndio de tempo precioso. A solução encontrada foi a de separar as rotinas de atendimento das interrupções em co-rotinas, selecionadas não pelo teste de variáveis, mas por modificações no vetor de interrupção.

Vários outros pequenos problemas, principalmente na geração das interrupções do nível 2 do PAE, foram resolvidos com introdução de hardware adicional, por exemplo, o fato do controlador de interrupção 8259 mascarar as interrupções e não inibi-las ocasionou mudanças na implementação do protocolo, bem como a utilização de circuitos integrados extras. Devido a todos os problemas encontrados a conclusão a que se chegou é que uma melhor solução seria a de se fazer uma interface com o embarramento microprogramada, sendo esta responsável por toda a realização do nível 2, liberando o processador 8086 para a realização do nível 3 e/ou aplicativos. Tal interface, em fase de projeto, vai atingir uma velocidade de de 10 Mbps por embarramento, e está sendo realizada de forma a poder ser facilmente integrada em um ou poucos circuitos integrados a curto prazo.

3.2 - O Núcleo de Sistema Operacional

Fornecendo a estrutura básica de software para a implementação dos diversos protocolos na REDPUC, encontra-se o núcleo do sistema operacional. O nível 2, como foi visto, é formado por um conjunto de co-rotinas que são ativadas essencialmente pelas interrupções de hardware.

O nível 3, que será descrito com mais detalhes na próxima seção, é formado por um conjunto de processos que executam em paralelo, que se comunicam entre si através de mensagens.

O núcleo do sistema operacional é responsável pelo escalonamento destes processos, além de oferecer as primitivas para a troca de mensagens e para gerência de memória. Além disso, o núcleo permite a multiplexação dos relógios de hardware em vários relógios lógicos que podem ser usados pelos vários processos.

As mensagens que podem ser trocadas pelos processos são de tamanho fixo (4 bytes), e podem ser de dois tipos: normais e do núcleo (urgentes). As mensagens do núcleo são utilizadas para comunicar aos processos eventos urgentes, tipicamente resultantes de interrupções de hardware.

Devido ao fato de que os processos que implementam o nível 3 serem fixos, pequenos e bem conhecidos, torna-se desnecessária a multiprogramação de processos através de "time-slices". Em vez disso, cada processo executa sem ser interrompido pelo núcleo (as interrupções do nível 2 são transparentes) e é responsável por devolver o controle ao núcleo em intervalos de tempo razoáveis.

O escalonamento de processos é feito de acordo com um esquema de prioridades. A prioridade de um processo é dada pela soma de uma prioridade estática e uma prioridade dinâmica¹. A prioridade estática tem três componentes: uma prioridade estática inicial, que reflete a importância relativa dos diversos processos; um modificador referente ao número de mensagens normais que o processo tem na sua fila de mensagens normais (quanto maior a fila, maior a prioridade); e um outro modificador, referente às mensagens urgentes.

A prioridade dinâmica de um processo pronto para executar é aumentada cada vez que um outro processo é escalonado. Desta forma, impede-se que um processo de baixa prioridade estática fique sem executar indefinidamente.

1 - O dinamismo da prioridade aqui é visto em relação ao processo de escalonamento.

Para a troca de mensagens o núcleo oferece primitivas que permitem o envio e recebimento de mensagens de forma bloquead e não-bloqueada. Quando um processo envia uma mensagem, esta é copiada para uma área interna do núcleo. Se não houver espaço no núcleo, o processo fica ou não bloqueado dependendo da primitiva executada.

Analogamente, quando um processo recebe uma mensagem, esta é copiada da área do núcleo para a área do processo. O processo pode ficar bloqueado se não houver mensagem para ele, dependendo da primitiva executada.

O núcleo oferece também primitivas para a alocação e liberação de buffers de tamanho fixo; no momento a gerência de memória é feita da forma mais simples possível.

O hardware do PC oferece um relógio para uso do software. Para permitir uma melhor utilização deste relógio, o núcleo permite que os processos programem vários relógios lógicos, que são multiplexados no relógio físico. Estes relógios são usados essencialmente como um serviço de "despertador".

3.3 - Arquitetura de Software do Nível 3 do PAE

O nível 3 do protocolo de acesso ao embaramento é implementado através dos seis processos ilustrados na figura, cuja descrição será feita a seguir..

O processo TX3 (Transmissor do Nível 3) ao ser ativado pelo Núcleo, varre a Fila de Transmissão do Nível 3 (FTX3) à procura da primeira mensagem a ser enviada através de um canal lógico, x cuja janela permita a transmissão (a janela de cada canal lógico tem tamanho igual a 1). Esta mensagem é então colocada na Fila de Transmissão do Nível 2 (FTX2). Ao mesmo tempo, é ligado o temporizador correspondente ao canal lógico x e um ponteiro para o "buffer" que contém a mensagem é guardado para uma eventual necessidade de retransmissão por esgotamento da temporização.

O processo RX3 (Receptor do Nível 3) retira a primeira mensagem da Fila de Recepção do Nível 2 (FRX2) e utiliza

o seu cabeçalho para atualizar as variáveis de controle de erro e controle de fluxo, e desligar a temporização dos canais lógicos que receberam confirmações de recebimento. Caso a mensagem recebida seja de informação ela é colocada na Fila de Recepção do Nível 3 (FRX3), sem o cabeçalho do Nível 3, porém precedida do número de canal lógico ao qual ela se destina. Caso a mensagem seja de supervisão, o processo SUP3 recebe uma mensagem de RX3 indicando o "buffer" onde se encontra a mensagem.

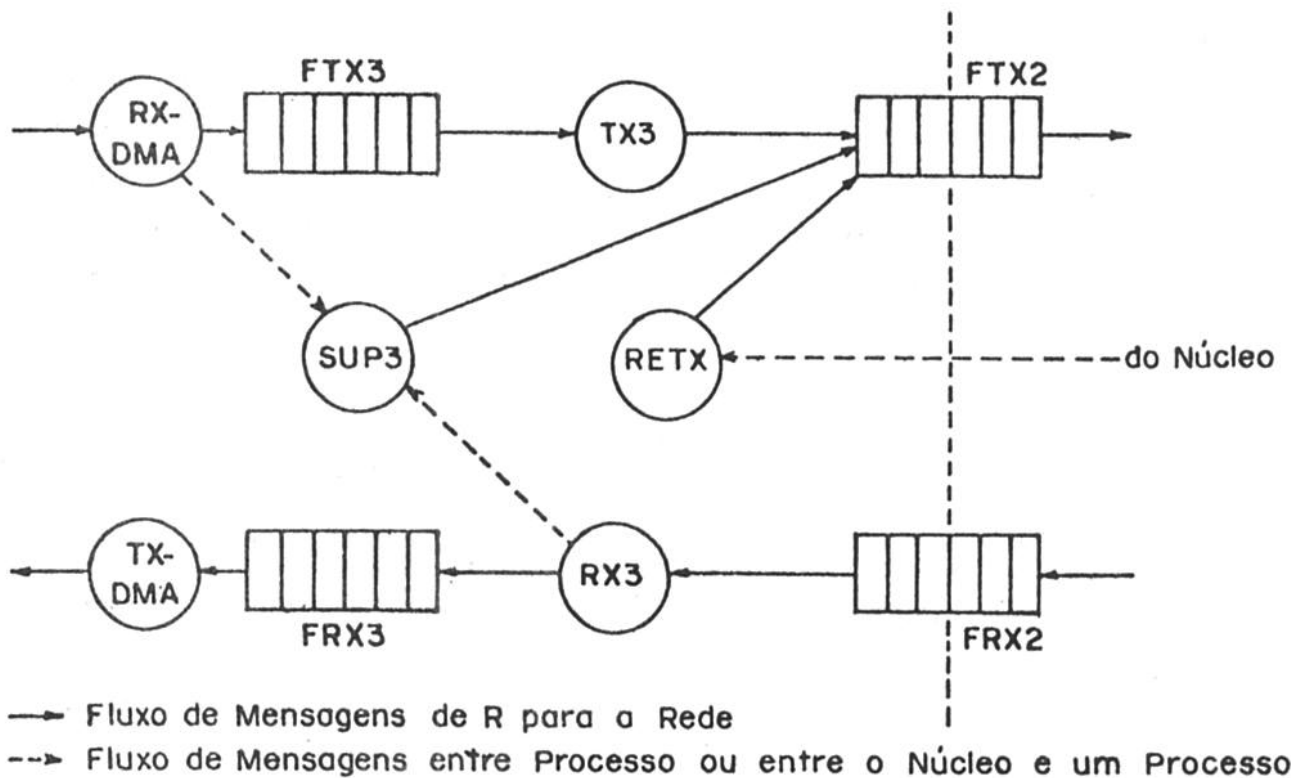


FIGURA 3 - PROCESSOS QUE IMPLMENTAM O NÍVEL 3

O processo SUP3 (Supervisor do Nível 3) se encarrega de realizar os procedimentos de estabelecimento e término

de conexões de transporte. As mensagens de supervisão recebidas da rede, via RX3, são analisadas por SUP3 e eventuais respostas são colocadas por SUP3 em FTX2. Os pedidos de estabelecimento ou término da conexão, localmente gerados, são entregues pelo processo RX-DMA a SUP3, que se encarrega de processá-los.

O processo RFTX (Retransmissor do Nível 3), ao receber o controle do Núcleo, examina se a temporização para algum canal lógico já expirou. Em caso afirmativo, ele recoloca a mensagem adequada em FTX2 para retransmissão e religa o temporizador.

Finalmente, os processos RX-DMA (Receptor de DMA) e TX-DMA (Transmissor de DMA) são responsáveis pela programação dos canais de DMA para recepção e envio de dados entre o MP e o PC.

4. Ambiente de Desenvolvimento

O software para a REDPUC está sendo desenvolvido em um módulo básico configurado com 128 k de memória, dois disquetes, um terminal de vídeo e uma impressora. No momento, outros sistemas de desenvolvimento com configuração semelhante estão sendo construídos.

A linguagem utilizada na codificação de 99% dos programas é C [6]. Apenas pequenos trechos de programa, no núcleo do nível 3 e partes do nível 2 foram escritos em linguagem assembler do 8086.

Em termos de espaço de memória consumido, o código implementando o nível 2 ocupa aproximadamente 2k bytes, o código do núcleo ocupa em torno de 3k bytes (mais 3 a 4k bytes de dados), e o código do nível 3 ocupa aproximadamente 7k bytes. Devemos ressaltar o fato de que estes números são obtidos com um compilador que produz código pouco otimizado; acreditamos que estes números possam ser reduzidos em torno de 20% por uma otimização mais cuidadosa.

5. Pesquisas Futuras

Como consequência do trabalho descrito neste artigo, foram iniciados diversos projetos na área de redes locais e automação de escritórios. Estes projetos visam consolidar a rede local desenvolvida na PUC/RJ, incluindo o projeto de interfaces para diversos microcomputadores brasileiros.

Além disso, estão sendo projetados diversos servidores especializados, tais como servidores de disco, servidor de impressora, e uma comporta X.25.

Com base nesta infraestrutura serão desenvolvidas aplicações do tipo transferência de arquivos, correio eletrônico, pastas eletrônicas, agendas e lembretes.

Finalmente, o grupo de redes da PUC/RJ está trabalhando na especialização de uma proposta de protocolo padrão de acesso a redes locais no nível imediatamente abaixo do nível de transporte.

Bibliografia

- [1] GOMES SOARES, L.F.; MENASCÉ, D.A. "Sobre a Evolução da Arquitetura de Centrais de Comutação de Pacotes". Anais do XV Congresso Nacional de Informática, outubro de 1982.
- [2] GOMES SOARES, L.F.; MENASCÉ, D.A. "Um Protocolo para Redes Locais do Tipo Difusão". Anales de la IX Conferencia Latinoamericana de Informatica, vol.2, PANEL INFO'82, Lima-Peru, agosto de 1982.
- [3] MENASCÉ, D.A.; GOMES SOARES, L.F.; JEFFERSON, F., CORREA C.H., "Descrição de Hardware e Software da Rede Local, REDPUC". Anais do XV Congresso Nacional de Informática, outubro de 1982.
- [4] MENASCÉ, D.A.; GOMES SOARES, L.F., "Arquitetura de uma Máquina Distribuída". Relatório Técnico, PUC/TELEBRÁS, Documento p 2, Laboratório de Engenharia e Sistemas de Computação, outubro de 1981.

- [5] MENASCÉ, D.A.; SCHWABE, D. "Redes de Computadores - Aspectos Técnicos e Operacionais". Terceira Escola de Computação, PUC/RJ, 1982.
- [6] KERNIJHAN, B.W.; RITCHIE, D. "The C Programming Language". Prentice Hall, 1978.