

“MINHOCA PLUS”, UMA REDE LOCAL PARA FINS DIDÁTICOS

- * Antonio Mauro Barbosa de Oliveira
- ** José Cláudio Teixeira e Silva Junior
- *** Maria Heveline Bernardes Vieira
- **** José Maria Coelho Filho

O presente trabalho descreve a rede “Minhoca” em sua versão “PLUS”. A rede apresenta-se como um “KIT” de baixo custo para o ensino de redes locais de computadores. Aspectos didáticos do “KIT” e suas características técnicas serão abordadas.

ABSTRACT

This paper describes the “Minhoca” local area network (LAN), version PLUS. This LAN presents itself like a LAN teaching KIT. Its educational aspects and technical features will be presented.

1. INTRODUÇÃO

A necessidade cada vez maior de comunicação entre usuários de sistemas de computação, associada ao benefício proporcionado pelo compartilhamento de recursos caros de “HARDWARE” e de “SOFTWARE” tem tomado atraente o uso de redes locais de computadores em ambientes outrora destinados a sistemas centralizados de médio e grande porte.

Naturalmente, o desenvolvimento mais recente da tecnologia de redes e a tendência à adoção de padrões, têm facilitado a proliferação das redes locais a níveis nacional e Internacional, além da característica de modularidade (inerente às redes locais) aspecto importante na análise custo X benefício de um sistema.

Como consequência imediata da rápida intromissão de redes locais em escritórios (automação), indústrias (controle de processos), escolas (ensino e administração) tem-se constatado carência de recursos humanos nesta tecnologia. Em resposta a esta deficiência tanto de formação como de informação é que cursos, palestras e artigos técnicos sobre o tema têm acontecido com maior frequência, além da oferta de disciplinas afins em currículos de engenharia elétrica e informática.

Os recursos didáticos utilizados, em geral, nestes cursos e palestras são os recursos tradicionais de ensino: giz, transparência, “slide”. Na pedagogia, no entanto, o aprender — fazendo tem tido sua eficácia comprovada, sobretudo em se tratando de ciências exatas e tecnológicas. As teorias Piagetiana e Skinneriana, enquanto distoantes no trato da questão ensino-aprendizagem, são uníssonas quanto à importância da atividade prática neste processo.

Neste contexto predominantemente pedagógico nasceu, foi projetada e implementada a rede “Minhoca”, aqui apresentada na sua versão “PLUS”. A idéia maior por trás deste projeto (rede “Minhoca”) é dispor de um laboratório de baixo custo para que alguns conceitos (por vezes abstratos) sobre teoria de redes de computadores possam ser ensinados/aprendidos de uma forma mais eficiente.

2. HISTÓRICO

A rede “Minhoca” tem sua arquitetura baseada na rede “CAPENGA” (SILV87), desenvolvida como

* Eng.^o Eletricista, M. Sc em Sistema de Computação (PUC-RJ) Coord. do Curso de Informática Industrial da ETF-CE
** Estudante do Curso de Matemática da UFC
*** Estudante do Curso de Engenharia Elétrica da UFC
**** Estudante do Curso de Bacharelado em Computação da UFC

trabalho na disciplina de redes locais (SOAR86) ministrada no departamento de engenharia elétrica da PUC/RJ. A rede "Minhoca", como um "KIT" para ensino, foi idealizada ao se tentar implementar o projeto "Um sistema de computação interredes para aplicação em um ambiente educacional" (OLI287), o qual é um segmento de um outro projeto mais amplo denominado "INFO 2000" (OLIV 86). Este sistema interredes teve sua versão experimental (OLI187) implementada no laboratório de engenharia de sistemas de computação da PUC/RJ.

As figuras 1 e 2 mostram o sistema interredes e a versão implementada, respectivamente. A "RED-PUC" (IERU85) foi utilizada como a rede de alto desempenho; o protocolo interredes foi o especificado em (MENA84), adaptou-se um servidor de discos CPM/80 (OLIV80) para um ambiente CPM/86 e, como já citado, a rede "CAPENGA" serviu como a rede de baixo custo no sistema.

O passo inicial na implementação do sistema interredes (OLI287) citado anteriormente foi o transporte da rede "CAPENGA" do ambiente CPM/86 para o MSDOS. Esta rede é praticamente toda em "software" e para este transporte fez-se necessária a adoção de inúmeras soluções que culminaram no abandono desta rede e no projeto e implementação de uma nova rede: a rede "Minhoca".

Os estudos que se sucederam na busca de soluções envolveram desde a estrutura de interrupção ao processador da estação, uso de primitivas de baixo nível do compilador na ativação de funções do sistema operacional e, naturalmente, protocolos de comunicação. Constatou-se, então, que se dispunha de muito material, a um baixo custo, para o ensino não só da teoria de redes de computadores, bem como de software básico.

A rede "Minhoca", assim, além de ser um produto para ensino se presta como um sistema de desenvolvimento de protocolos, programas aplicativos e utilitários de rede.

3. O USO DA REDE "MINHOCA"

A rede "Minhoca" é uma rede local praticamente toda em "software", característica que confere à mesma um baixo custo. Em contrapartida, ela possui um baixo desempenho haja visto todos os protocolos serem executados pela estação. Isto não se constitui um problema considerando-se os objetivos pedagógicos a que ela se propõe.

O "KIT" da rede "Minhoca" é constituído dos seguintes componentes:

- Três conectores DB-25 (padrão serial RS 232 C) ligados em um par trançado.
- Um (01) disco flexível de 5 1/4" contendo:
 - * Sistema operacional MSDOS - compatível
 - * Arquivo - fonte (MINHOCA.PAS)
 - * Arquivo - executável (MINHOCA.COM)
 - * Arquivo - texto (MINHOCA.DOC)
- Manual de experimentos contendo:
 - * Informações de instalação e uso
 - * Modificações sugeridas no arquivo fonte para análise e posterior compilação

A versão "PLUS" da rede "Minhoca", aqui apresentada possui além dos protocolos de baixo nível, os níveis interredes, transporte e um servidor de disco, executáveis no ambiente MSDOS - compatível.

As atividades pensadas, inicialmente para o "KIT" podem ser resumidas nas seguintes, fazendo-se uso de três (03) microcomputadores:

- Demonstração, a nível de usuário final, onde o professor/palestrante se limita a apresentar aplicativos de rede local. Ou ainda, o usuário faz uso destes aplicativos, orientado pelo manual.

- Laboratório, onde alunos de posse do programa fonte, dos protocolos e aplicativos fazem modificações, sugeridas ou não no manual, para posterior análise da rede.

- Sistema de desenvolvimento de aplicativos de rede local, protocolos de comunicação, etc.

4. PROTOCOLOS DA REDE "MINHOCA"

A rede "Minhoca" foi implementada na linguagem Pascal, uma vez que esta linguagem se adequa bem ao caráter didático ao qual o protótipo se propõe.

4.1 - CARACTERÍSTICAS

A rede "Minhoca", como citada acima, é uma rede local de baixíssimo custo, com fins didáticos, possuindo as seguintes características:

- Topologia: Barra comum;
- Meio de transmissão: Par trançado;
- Acesso ao meio: CSMA;
- Transmissão: Assíncrona a 9600 bps;

4.2 - "HARDWARE"

O "hardware" é extremamente simples, consistindo de uma interface RS-232-C ligada à saída serial padrão do microcomputador PC-compatível referida como COM1 (GOFT86).

4.3 - "SOFTWARE"

Como foi dito anteriormente, todos os protocolos foram implementados em "software", fato que confere à rede um baixíssimo custo tendo como contrapartida um "overhead" natural, dado o processamento necessário.

4.3.1 - ESTRUTURA DO PACOTE

O pacote utilizado na rede "Minhoca" tem a seguinte estrutura:

```
Tipo-pacote = record
destino      : byte;
tamanho     : byte;
fonte       : byte;
dados       : array (1..128) of byte;
end
```

Deste modo, o campo "dados" tanto poderá conter dados quanto informações de controle. Isso é

transparente neste nível de protocolo (nível 2) e só será tratado em níveis superiores.

4.3.2 - "BUFFER" CIRCULAR

Após um pacote ser recebido por uma estação, ele é imediatamente colocado em um "buffer" circular para posterior utilização. Isso é feito pela rotina "atualiza-buffer".

```
procedure atualiza-buffer;
begin
  pt-buffer-circular = pt-buffer-circular mod
    T - BUFFER;
  buffer-circular (pt-buffer-circular) := buffer;
  pt-buffer-circular := pt-buffer-circular + 1;
end;
```

Onde temos:

- I) T-BUFFER é uma constante que indica o tamanho do "buffer" circular;
- II) as variáveis:
 - pt-buffer-circular : integer;
 - buffer : tipo-pacote
 - buffer-circular : array (D..T-BUFFER) of tipo-pacote

4.3.3 - INTERRUPTÕES

Na rede "Minhoca" foram introduzidas duas interrupções de "hardware". A primeira, a interrupção do relógio (10h, do BIOS), responsável por instalar um temporizador; e a segunda, a interrupção da porta serial principal (DC h, do BIOS), que ativa a rotina de recepção.

A alteração do vetor de interrupções do BIOS e a instalação da interrupção da porta serial são encontradas com detalhes em (LIND87).

4.3.3.1 - INTERRUPTÃO DA PORTA SERIAL

Ao instalar-se a rotina que trata a interrupção da porta serial e ao habilitar-se o registrador de interrupções do "chip" 8250 ("UART", no PC-compatível), tem-se preparado a rotina de recepção da rede "Minhoca".

Cada byte presente na linha provoca uma interrupção no micro para que este possa ler e analisar o determinado dado.

Considerando o afirmado acima vemos tratar-se de um problema, pois essa interrupção afeta consideravelmente o desempenho da estação. Uma solução para isto está em se desenvolver um hardware inteligente que interrompa a estação quando tiver um pacote pronto para ser tratado. Essa solução se tornará viável nas futuras versões da rede "Minhoca".

4.3.3.2 - INTERRUPTÃO DO RELÓGIO

A interrupção do relógio foi usada para solucionar dois problemas: determinar o estado do meio físico e abortar a recepção de um pacote com erro no tamanho.

Esta interrupção ativa um temporizador. Trata-se de um contador, o qual é incrementado a uma certa frequência pelo micro. Esse contador é representado pela variável "tempo", do tipo "integer", de escopo global.

Essa variável é usada para informar há quanto tempo não existe dado transitando no meio. Isto indica o estado do meio físico, ou seja, se este está livre ou ocupado para transmissão.

Caso o temporizador seja maior que uma constante TEMPO-MAXIMO (esta variável indica o tempo máximo na transmissão de dois caracteres do mesmo pacote), a linha está livre para transmissão, caso contrário está ocupada.

```
procedure int-relógio;
begin
  tempo := tempo + 1;
  if tempo > TEMPO-MAXIMO then
    meio := livre;
  else
    meio := ocupado;
end;
```

A utilização de um "hardware" inteligente, como discutido anteriormente, também evitaria esta interrupção no micro, pois o estado do meio seria detectado por este "hardware".

4.3.4 - ROTINAS DE COMUNICAÇÃO

A rotina de transmissão é chamada explicitamente quando se deseja enviar um pacote, ao passo que a rotina de recepção é executada automaticamente, pelo "hardware", sempre que houver dado transitando na linha, conforme explicado anteriormente.

Temos as seguintes definições das variáveis usadas:

```
estado : (espera-pct, toma-tam-pct, toma-fonte, guarda-pct, transmitindo);
meio : (livre, ocupado);
dado, tam-pct, tam-lido, pt-buffer : byte;
Além disso, existe a função "recebe-dado", que retorna um byte disponível da porta e a função "transmite-dado", que envia um byte para a porta.
```

4.3.4.1 - TRANSMISSÃO

Esta rotina implementa o protocolo de acesso ao meio-CSMA, na primeira versão.

```
procedure transmite-pacote (pacote : tipo-pacote);
var
  l : byte;
  pct-transmitido : boolean;
begin
  pct-transmitido := false;
  repeat
    espera-aleatoria;
  if meio = livre then
  begin
    transmite-dado (ord(pacote . destino));
    transmite-dado (pacote . tamanho);
    transmite-dado (ord(pacote . fonte));
    for l := 1 to length (pacote . dados) do
```

```

    transmite-dado(ord(pacote . dados (i));
    pct-transmitido := true;
  end;
until pct-transmitido;
end;

```

4.3.4.2. RECEPÇÃO

A rotina a seguir implementa a recepção na rede.

```

procedure recepção;
begin
  dado := recebe-dado;
  case estado of
    espera-pct:
      begin
        if dado = ord (ESTAÇÃO) then
          begin
            tam-lido := 1;
            pt-buffer := 0;
            buffer destino := chr (dado);
            estado := toma-tam-pct;
          end;
        end;
        toma-tam-pct;
      begin
        tam-pct := dado;
        buffer . tamanho := dado;
        tam-lido := tam-lido + 1;
        estado := toma-fonte
      end;
        toma-fonte;
      begin
        buffer.fonte = chr(dado);
        tam-lido = tam-lido + 1;
        estado = guarda-pct;
      end;
        guarda-pct;
      begin
        pt-buffer := pt-buffer + 1;
        tam-lido := tam-lido + 1;
        buffer . dados (pt-buffer) := chr (dado);
        If tam-lido = tam-pct then
          begin
            estado := espera-pct;
            atualiza-buffer;
          end;
        end;
      end;
    end;
  end;
end;

```

4.4 — OPERAÇÃO TÍPICA

A primeira versão da rede "Minhoca" (versão 1.1) opera através de "menu", logo após a rede ser carregada são apresentados os serviços oferecidos pela mesma (troca de mensagens, transferência de arquivos). Isto gera o inconveniente de existir um programa ocupando o micro sem que se possa usar aplicativos na rede, já que o mesmo utiliza um sistema operacional monousuário.

Foi desenvolvida uma outra versão (1.2) da rede "Minhoca", onde este problema foi contornado.

Inicialmente pensou-se na instalação de núcleo multitarefa (LIND87) com escalonamento circular

(ROUND-ROBIN) para os processos usuários e a rotina de recepção. Finalmente, optou-se por deixar a rotina de recepção residente fazendo uso da Interrupção 27h do BIOS.

4.5 — PROTOCOLO INTERREDES

A filosofia por trás das interredes consiste em considerar cada rede como um meio de comunicação por onde deverão transitar os pacotes até as estações intermediárias especiais ou à estação final na rede destino. As estações intermediárias tratam-se de vias especiais que têm como função a interligação física e lógica entre duas ou mais redes; são as chamadas passarelas ("gateways")

Ao pretender-se enviar uma mensagem interrede, uma estação deverá encaminhar seu pacote à passarela pertencente à sua rede local, utilizando o protocolo local de comunicação. Sendo recebido pela passarela, o pacote é traduzido e encaminhado a uma outra passarela intermediária, de acordo com protocolo da segunda rede, e assim por diante, até que a última passarela o encaminhe à estação final de destino.

O nível interredes implementado segue a especificação proposta em (MENA84). Dada as características já citadas da rede "Minhoca" uma série de simplificações desta especificação foram realizadas. A estrutura abaixo ilustra o serviço oferecido por este nível:

Type

rede = byte;

modulo = byte;

datag = RECORD

end-dtg: ponteiro;

tam-dtg: Integer

END;

dtg-mod = RECORD

tipo-dtg : byte;

red-origem : rede;

mod-origem : modulo;

red-destino : rede;

mod-destino : modulo;

tipo-protocolo : byte;

datagrama : datag;

END;

4.6 — NÍVEL DE TRANSPORTE

Com o objetivo de prover uma rede cada vez mais transparente para o usuário, e de evitar uma série de problemas em aplicativos nas versões anteriores da rede "Minhoca", foi implementado o nível de transporte.

Os serviços adicionados à rede com o nível de transporte são relacionados a seguir:

1. Conexão lógica:

Com a instalação do protocolo de transporte ocorre uma conexão lógica entre as estações que estão se comunicando. Deste modo, qualquer transmissão destinada a estas estações não podem ser efetuadas até que a conexão seja desfeita.

Esta decisão foi tomada tendo em vista a utilização de estações monotarefas.

II. Correção de erros.:

O protocolo de transporte permite a correção de erros na transmissão no sentido de que quando um erro é detectado, o pacote é retransmitido.

III. Fragmentação:

A fragmentação das mensagens nas versões anteriores era realizada pela própria aplicação que se encarregava de dividir a mensagem em vários pacotes.

Com o protocolo de transporte instalado este problema não mais ocorre, uma vez que é o próprio protocolo de transporte quem efetua a fragmentação na estação origem e a remontagem na estação destino.

5 – SERVIDOR DA REDE MINHOCA

Conforme comentado no Item 2, que traça a evolução da rede "Minhoca", o servidor de disco utilizado na versão experimental do sistema interredes (OLI187) fora implementado no ambiente CPM/86 (OLIV84).

As estações da rede, usuários deste servidor, tinham o BIOS alterado de modo a permitir que um programa gerenciador de recursos (GREC) montasse discos localmente (caso existissem unidades físicas) ou remotamente (servidor) para o usuário. O servidor, por sua vez, associava a cada disco virtual um arquivo em seu ambiente CPM/86.

O transporte deste servidor para o ambiente da rede (MSDOS - compatível) enfrentou alguns problemas, a começar pela dificuldade de alteração do BIOS residente (funções próximas do "hardware") no MSDOS-compatível. Além disso este sistema operacional utiliza o mecanismo chamado "DEVICE DRIVER", mais precisamente "BLOCK DEVICE DRIVER", no tratamento de dispositivos de disco.

A alternativa inicial foi a instalação de um redirecionador para acesso ao disco remoto, utilizando-se esta característica de "DEVICE DRIVER", citada anteriormente. O redirecionador, denominado RED.SYS, foi implementado, sendo posteriormente descartado devido a problemas de reentrância do sistema operacional nativo, certamente provocados por características intrínsecas dos protocolos de baixo nível da rede "Minhoca" que para contornar a inexistência de hardware na rede, abusa das estruturas de interrupção. O redirecionador implementado fica no aguardo de uma versão evoluída da rede que execute os protocolos de baixo nível em "hardware"/"firmware".

A segunda alternativa foi implementar funções de manipulação de arquivos para acesso a discos remotos ficando por conta destas funções todo controle de alocação no disco virtual. Naturalmente, o servidor na estação remota cria arquivos que emulam discos com a estrutura padrão MSDOS-compatível.

Assim, um comando RDIR pede ao servidor informações sobre diretório do seu disco virtual, sem

se fazer o "login" do disco virtual, procedimento pertinente à alternativa inicial.

RCOPY, por exemplo, ao copiar um arquivo do servidor para a estação analisa o "ROOT DIR" e a "FAT" no disco (arquivo emulador) servidor, solicitando ao mesmo os "clusters" alocados para o arquivo a ser copiado. O processo inverso é mais interessante pois o próprio RCOPY faz a análise de onde irá alocar espaço dentro do disco servidor e depois de copiado ele próprio analisa a "FAT".

Percebe-se que foi criado um "FILE SYSTEM" próprio, onde comandos tipo COPY, DEL, MD podem ser analisados pelo estudante, num ambiente compatível com o MSDOS.

5.1 Rxxxxxx.EXE

A idéia desses comandos e que eles analisem o disco virtual através de sua estrutura, conhecida "a priori", e o modifiquem (se for o caso) dentro da expectativa do usuário. Por exemplo, consideremos a cópia de um arquivo no diretório raiz do disco h, (virtual) no servidor para o disco b, (real) na estação do usuário. O comando RCOPY irá executar os seguintes passos:

- Obter o "BOOT SECTOR" do disco h: (no servidor)
- Localizar a "FAT" e o "ROOT DIR" e então ler os setores indicados
- Procurar um arquivo com o nome especificado no "ROOT DIR" identificando o "STARTING CLUSTER" deste arquivo
- A partir do "STARTING CLUSTER", obter os "CLUSTERS" restante (FAT)
- Abrir um arquivo no drive b: e copiar os "CLUSTERS" do arquivo em h: para o arquivo em b:

Dentro deste espírito de controle total do processo estão sendo implementadas outras funções do MSDOS. Ao se criar uma biblioteca de funções o desenvolvimento de sistemas fica mais rápido propiciando a interessados um ambiente favorável a testes de funções de um sistema operacional. Um exemplo de uma função de biblioteca: dado um "CLUSTER", obter o próximo "CLUSTER" daquele arquivo.

5.2 – SERVIDOR.EXE

O servidor da rede "Minhoca" é um programa que trata os comandos vindos da estação servidora, via primitivas envia-pacote, recebe-pacote. Como ele foi desenvolvido paralelamente ao nível de transporte, sua interface se dá com o nível de enlace. Brevemente ele será integrado a rede "Minhoca" juntamente como os níveis de transporte e interredes.

Quando os comandos Rxxxxxx.EXE acessam o servidor, eles o fazem com a estrutura de dados tipo-pacote.

```
typedef struct (  
    unsigned char destino;  
    unsigned char tamanho;
```

```

unsigned char fonte;
unsigned char tamanho-dados;
char dados (128);
) tipo-pacote;

```

que é enviada para a rede para o servidor que a decodifica em:

- tipo-pacote dados(D) = comando
- tipo-pacote dados(1) e (2) = número do setor
- tipo-pacote dados() = nome do disco

(a transmissão ocorre através de chamadas à rede que fica residente e utiliza as interrupções AOH E A1H)

Os comandos podem ser:

- criação de disco "C"
- leitura de setor "S"
- gravação de setor "G"

O corpo principal do programa consiste de:

```

while(1) (
recebe-comando( );
switch(comando . dados(D))(
case LER;
ler-setor( );
enviar-setor( );
break;
case GRAVAR;
recebe-setor( );
grava-setor( );
break;
case CRIAR;
criadisco( );
break;
)
)

```

A leitura e a gravação em disco são executadas pelas rotinas ler-setor e grava-setor, respectivamente, que recebem o nome do disco e o setor para, em seguida realizarem a operação solicitada.

Para se criar um disco, o processo é mais complexo. As estruturas de dados a seguir, ilustram bem este fato:

```

typedef struct (
.
.
.
) VOLUME LABEL
typedef struct (
.
.
.
) BPB
typedef struct (
.
.
.
) BOOT-SECTOR

```

A função CRIADISCO() consiste basicamente de:

```

CRIADISCO( )
(
criar( );
boot( );
fat( );
)

```

diretório();

Este servidor não possui funções de concorrência, proteção e outras características desejáveis, seja por tratar-se de uma versão primeira seja também, por limitações do ambiente disponível em sua implementação. No entanto, como já comentado, ele se apresenta não como um serviço e sim como um farto laboratório de software básico.

6 – CONCLUSÃO

A rede "Minhoca", como já bastante enfatizado no texto, não se propõe a ser uma rede comercial, dados as limitações de distanciamento das estações impostas pelo padrão RS232C e o baixo desempenho provocado pela inexistência de "hardware". Ela é um recurso didático de baixo custo para o ensino da teoria de redes de computadores e disciplinas afins (linguagens, sistemas operacionais, etc) podendo ser utilizado como sistema de desenvolvimento.

VERSÃO	DISCRIMINAÇÃO
0.0	Níveis 1/2: Comutação por mensagem, sem detenção de erro Primitivas: envia_m, recebe_m (Polling)
1.0	Níveis 1/2: CSMA, sem detenção de erro Primitivas: env_pac, rec_pac (Polling) Aplicativos: envia/recebe mensagem
1.1	Níveis 1/2: CSMA/CD com detenção de erro (CHECK-SUM) Primitivas: env_pac, rec_pac (interrupção) Aplicativos: envia/recebe mensagem, conversação Jogos: Jogo da velha, senha transferência de arquivos Interface: menu
1.2	Níveis 1/2: Idem à versão 1.0 Interface: Janelas
2.0	Níveis 1/2/4: CSMA/CD com detenção de erro (CHECK-SUM) transporte (fragmentação/remontagem de mensagens, conexão lógica, correção de erros). Primitivas: ABRECON, FECHACON, TRANSFER (Intra-redes) Aplicativas: Idem à versão 1.0
3.0	Níveis 1/2/3/4: CSMA/CD com detenção de erro (CHECK-SUM) Interredes (roteamento de datagramas não confiáveis) transporte: Idem à versão 2.0 Primitivas: ABRECON, FECHACON, TRANSFER (Interredes) Aplicativas: Idem à versão 1.0

Versão "PLUS: Idem à versão 1.x com o servidor de disco..

Através da análise e/ou apresentação das diversas versões da rede "minhoca", torna-se fácil, por exemplo, ilustrar a diferença entre uma rede de mensagens e uma rede de pacotes, protocolos de acesso ao meio, problemas causados pela ausência de um nível de transporte, etc., além de aspectos de "software" básico e de "software" de alto nível utilizados na implementação dos protocolos.

Interfaces que permitem o distanciamento das estações (raio de 100m) — não pertencentes ao "KIT" — foram desenvolvidas já como embrião para uma

futura versão de "hardware" da rede "Minhoca", utilizando componentes "standard", por motivos didáticos.

Versões para microcomputadores de 8 bits da linha Apple e Cobra (ambiente CPM/80) estão sendo desenvolvidas.

A rede "Minhoca" sub-produto do projeto (OLI287), vem sendo utilizado em cursos técnicos (ETFCE), em cursos de graduação e pós-graduação (UFC), dentro do seu propósito detalhado neste trabalho. Atualmente a rede possui cerca de 50 usuários cadastrados, espalhados em praticamente todas as capitais brasileiras que recebem versões atualizadas, além de uns poucos usuários no exterior.

UM SISTEMA DE COMPUTAÇÃO INTERREDES PARA APLICAÇÃO EM UM AMBIENTE EDUCACIONAL

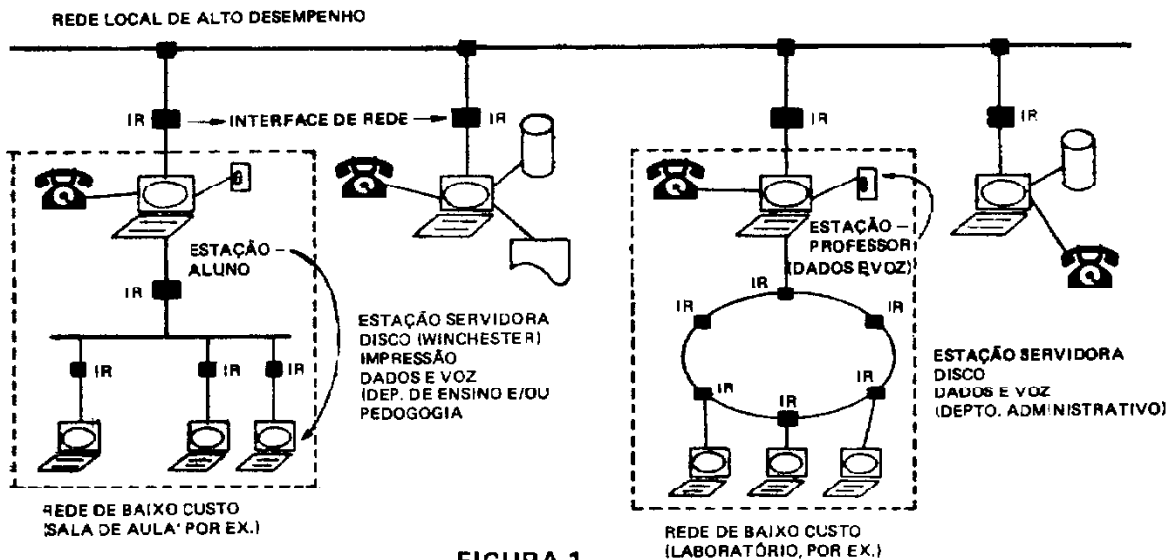


FIGURA 1

SISTEMA IMPLEMENTADO NO LABORATÓRIO DE ENGENHARIA DE SISTEMAS DE COMPUTAÇÃO DO DEPTº DE ENGENHARIA ELÉTRICA DA PUC-RJ

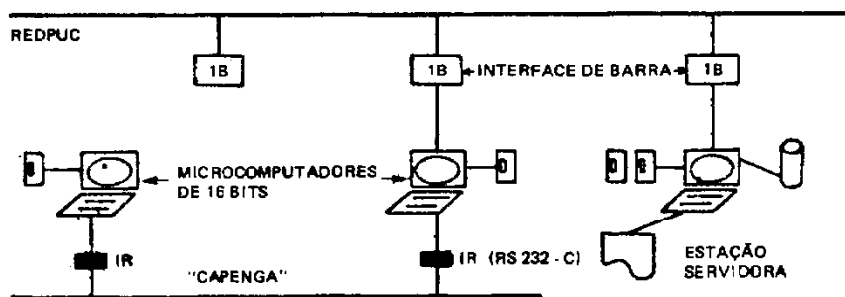


FIGURA 2

7 – AGRADECIMENTOS

Gostaríamos de agradecer ao prof. Alexandre D. Diógenes, diretor do Centro de Tecnologia da UFC e ao prof. José Evangelista C. Moreira, responsável pelo Laboratório de Instrumentação Eletrônica do Departamento de Física da UFC, pelo apoio recebido durante a realização deste trabalho. Agradecemos, também, a colaboração dos profs. Giovanni Cordeiro Barroso e José Barreto Júnior.

8 – REFERÊNCIAS BIBLIOGRÁFICAS

- (GOFT86) GOFTON, P.W. Mastering Serial Communications. SYBEX, California. 1986.
- (HOST83) HOSTE, F. Les reseaux locaux d'entreprises. Editests, Paris. 1983
- (IOFS79) International Organization for standardization (ISO). Refence Model of Open Systems Interconnection. 1979.
- (IERU85) IERUSALIMSCHY, R. Interface de barra microprogramada para a rede local REDPUC. Tese de mestrado, Departamento de Informática, PUC/RJ. Dezembro de 1985.
- (LIND87) LINDLAY, C.A. Multitasking With Turbo Pascal, Dr. Dobbs Journal of Software Tools for the professional programmer, Julho de 1987.
- (MENA84) MENASCE, D.A. Especificação de um protocolo para interligação de redes locais. Anais do XI Seminário Integrado de Software e Hardware, Viçosa-MG. Julho de 1984.
- (OLI287) OLIVEIRA, A.M.B. Um sistema de computação interredes para aplicação em um ambiente educacional. Tese de Mestrado, Departamento de Eng. Elétrica, PUC/RJ. Junho de 1987.
- (OLI187) OLIVEIRA, A.M.B., SOARES, L. F. G Um sistema de Computação interredes aplicado a um ambiente educacional. XIV SEMISH. Salvador (BA). Julho de 1987.
- (OLI86) OLIVEIRA, A.M.B, GURGEL, M. Projeto Info 2000. Anais do II Seminário de Informática da Amazônia. 1986.
- (OLIV84) OLIVEIRA, F.J.; SCHWABE, D.; VIEIRA, E.A. P. Como estender o seu ambiente de computação atual usando uma rede local. Anais do XVII Congresso Nacional de Informática – SUCESSU. Rio de Janeiro, 1984.
- (SILV87) SILVA, R.P.; ROCHA, C.A.; ALVES JR., O.C; RIBEIRO, N.R. Documentação da rede CAPENGA. Trabalho de curso. Departamento de Engenharia Elétrica. PUC/RJ. 1987.
- (SOAR87) SOARES, L.F.G. Redes Locais. Campus, Rio de Janeiro, 1987.