MINHONIX, A DISTRIBUTED SYSTEM FOR TEACHING

A. Mauro B. **OLIVEIRA** (*)
e. mail : mbo@advance.ibp.fr

Université Paris 6
Laboratoire MASI
45, Avenue des Etats-Unis
78000 Versailles (France)

Jean-Pierre **PEYRIN**
e. mail : peyrin@imag.imag.fr

Université Joseph Fourier Grenoble 1
Laboratoire de Génie Informatique
B.P. 53x
38041 Grenoble cedex (France)

Abstract

In this paper we discuss the problems related to teaching Computer Networks and Distributed Systems. We present the Minhonix System, a tool to use in technical course concerning these subjects.

The Minhonix System has mainly two components: a teaching kit (a zero-slot local area network) and a method (the pedagogical steps proposed) for better use of this kit.

The Minhonix System is the outcome of the "marriage" between our experience using the Minhoca Network and MINIX, a didactic UNIX-like operating system.

While we were implementing this system, we discovered that the Minhonix System is also a good tool for the teaching of operating systems. The several concepts can be illustrated by the teacher and easily used by the students.

(*) On leave from Federal Technical School of Ceara (Brazil)

Introduction

There has been an explosive growth in the computer networks in the last years. Computer networks have revolutionized our use of computers.

There are many reasons for this : the technology has greatly reduced the cost of establishing computer networks and the proliferation of personnal computers and workstations during the 1980s. Another reason is that many computer manufacturers now package networking software as part of the basic operating system.

Today, one speaks about distributed systems. Everyone agrees that distributed systems are going to be very important in the future, but what does exactly mean a distributed system?

Unfortunately, not everyone agrees on what they mean by the term "distributed systems", but it is sure that the key concept is that of "transparency" environment. The transparency access to resources of the other machines has become increasingly desirable. In other words, the user views the system as a "virtual uniprocessor", not a collection of distinct machines [TANENBAUM 89a]. This is easier said than done.

Both the computer networks and distributed systems are the subjects treated (teaching) in the Minhonix System. This paper is organised as follow:

The first part describes our opinion about the problems related to the teaching of the subjects above, based on our experience in the technical education. It also proposes the Minhonix System as a tool to assist the traditional teaching of these subjects.

The second part presents generalities about the Minhonix System: the kit, the method (the pedagogical steps for the teacher to better use the Minhonix System), the features and advantages of the system.

The third part lists some versions implemented. Several examples of their use are described too.

The fourth part gives some informations about the current state of the system and its future profile.

1.Problematic

Today, teachers of computer networks and distributed systems still use a blackboard and a slide projector as the principle teaching tools. To clarify our viewpoint we are going to classify these tools as resources of the traditional teaching methods.

The efficiency of these resources (the blackboard and the slide) in teaching is at the present quite questionable. Their success is very dependent of the teacher's didactic.

Our experience in the technical courses during 15 years, reveals that there is a wide gap between the "teaching" and the "learning" when the teacher uses only the traditional teaching methods, mainly with the teenager students. In our opinion, this problem is more evident in the teaching of the technical subjects.

To improve this situation (to reduce this gap) in ours courses, we designed and implemented the Minhoca System [OLIVEIRA 89] that originated the Minhonix System presented here.

Generally speaking, the Minhonix System has the following characteristics that distinguish it : feasible, dynamic, modular and motivating.

Feasible: it means that the system is not expensive, it is easy to install and use. It runs on the popular existing machines (IBM-PC compatibles).

Dynamic: it expresses the capacity of the system to be used both as a demonstration tool of the classical concepts (for the teacher) and as a developement tool of communication softwares (for the students).

Modular: the system contains several softwares (versions) for each subject proposed. The teacher can choose the point of start and the versions more adequate for the course program.

Motivating: it means for us the aspects concerning the cognitive psychology. We had observed the good results and the acceptance among the students when we used this system, but a thorough analysis in this area is out of the scope of this paper.

2.Generality

The bases of the present system were developed during the realisation of a practical project for the LAN courses at the "PUC/RJ" university [SOARES 86] and are described in [OLIVEIRA 87]. The figure 1 shows this environment.

The Minhonix System is composed of the two elements : the Minhonix LAN (a teaching's kit) and the Minhonix Method (usage of this kit) .

2.1 About the Minhonix LAN

The kit of Minhonix System is simply a zero-slot LAN (local area network) for the microcomputers IBM-PC compatibles. That is, instead of taking up one of your microcomputer's expansion slots, it must be plugged into a serial port.

Therefore, the system has not any electronic device. All of its communication protocols are implemented in software. This provides the principle pedagogical feature of the system.

The Minhonix LAN has a bus topology and uses a twisted-pair cable as data physical transmission medium. It is composed of 4 connectors DB-25 (the RS-232c standard) and several softwares that run under MINIX [TANENBAUM 89a] and MSDOS [DUNCAN 86] operating systems. The software of the Minhonix LAN is commented in the item 3.1 and detailed in [OLIVEIRA 90].

The Minhoca Network [COELHO 89] had been the first prototype of the Minhonix LAN. It was implemented with the Turbo-Pascal compiler under MSDOS. For the versions developed under MINIX operating system, we used the C compiler available in this system. We have a heterogeneous environment of communications. This situation expresses the reality of the computer networks and so it is really rich, pedagogicaly speaking.

2.2 About the Minhonix Method

The Minhonix Method is a set of laboratory's activities proposed for better use of the Minhonix LAN.

These activities were grouped in three pedagogical steps, as shown bellow:

Sensibilisation: in this first step, the principle idea is to motivate the students. So, the teacher must undertake some experiments where the students use the Minhonix LAN rather as end-users. Several application programs are available (simple electronic mail, games, simple file transfer, etc.).

Analysis: in the second step, the teacher discusses with the students the architecture of the communication protocols and application programs of the Minhonix LAN. One very important activity to start this step is to explain the low-level functions as, for exemple, the code source of the SendPacket and ReceivePacket procedures (see also the item 3.2.2).

Synthesis: in the third step, it is suposed that the students have acquired enough knowledge about the Minhonix LAN. Now, the theacher must ask them (individually or in groups) to develop a program. This program can be one application program (games, simple mail, etc.), perhaps a protocol to improve an existing version or to simulate the classical situations of the communication systems.

It is important to note that the above pedagogical steps are not necessarily in a sequential order. That is, it is the task of the teacher to decide the best time to start these steps and to choose the versions of the Minhonix LAN to use in each step according to the "pedagogical restrictions" (the objective of the course, the level of the students, the time available, and so on).

Another remark (necessary for the understanding of the next part of this paper) is to do not forget that the availability of the whole source code of the Minhonix LAN is the "heart" of the Analysis and Synthesis Pedagogical Steps described above.

Nothing is hidden in the Minhonix LAN. The teacher can illustrate from the low-level functions (as the interrupt hardware when a process sends or receives a message), up to high-level concepts as the client-server model, remote procedure call, and so on.

The details of these steps, with the more adequate versions of the Minhonix LAN, are presented in [OLIVEIRA 90].

2.3 Features and advantages

The Minhonix System is not a complete course by itself, but a tool to assist the traditional teaching. As we said above, it has a affordable kit (Minhonix LAN), very easy to install and use (it require only to plug the connectors to the microcomputers and load the softwares).

Perhaps, the principle feature of the Minhonix System is its aspect of the open system, in the pedagogical sense. That is, the student can look at all of the parts of the architecture and protocols of the system (to analyse and maybe to modify it), step by step.

The teacher can choose the versions available in the system that seem more adequate for his course program. If he wants, he can also modify both the versions and the method proposed to use the system.

With the Minhonix System, it is possible to simulate the classical topics in the computer networks and distributed systems subjects, as we have commented above in the Synthesis Step. For example: the several politics of the medium access control, the deadlock between two processes in a distributed application. This seems for us one very high level feature of the system.

The first prototype of the Minhonix System (Minhoca Network) had been used for 2 years in the courses of several levels (high school and undergraduate levels) in the Industrial Computer Science Course at the "Escola Tecnica Federal do Ceara" and in the Department of Electrical Engineering at the "Universidade Federal do Ceara" (Brazil). It proved to be very efficient in relation to the traditional teaching methods.

3. The Minhonix System

3.1 The Versions (Minhonix LAN)

We propose 10 classical subjects in the Minhonix LAN. They were based on the concepts presented in two known books in France: Computer Networks [TANENBAUM 89b] and "Systèmes Informatiques Répartis" [CORNAFION 81].

The versions developed for these subjects were grouped as follows:

1. **Network Group**

1.1 MHIX1 :	point-to-point connection
1.2 MHIX2 :	network architectures
1.3 MHIX3 :	medium access control (LAN)
1.4 MHIX4 :	data link layer (ISO)
1.5 MHIX5 :	transport protocol

2. **Distributed Group**

2.1 MHIX6 :	client-server model
2.2 MHIX7 :	problems of the distribution
2.3 MHIX8 :	execution of the distributed application
2.4 MHIX9 :	scheduling & synchronization
2.5 MHIX10 :	internetworking

Of course, other subjects about the computer networks and distributed systems can be illustrated with the Minhonix System. The criterion for the choice of the topics above was rather the "ad hoc" manner.

Most versions of the Minhonix LAN use the same procedures to send and receive packets :

The SendPacket procedure uses a classical access control of the communication physical medium : CSMA/NP.

The ReceivePacket procedure is less standard. It was developed taking into account the special characteristics (the zero slot LAN) of the system. Figure 2 shows the state machine that describes the behaviour of this procedure.

Each one of the subjects above has some versions. The details about these versions (subject, description, objective, hypothesis and the scenario) are described in [OLIVEIRA 90].

There is also in [OLIVEIRA 90] a relationship between a check-list [NORA 85] and a code that allows the user to know how subjects can be taught with the Minhonix System and what version is the best to use.

3.2 The Examples (Minhonix Method)

It is important here to remember that there are several versions about the same subject. For example, the "MHIX1a" version is a very simple program that sends and receives a message in a sequential manner (as a monoprogramming environment). The "MHIX1b" version is the same application, but implemented as two concurrent processes (to send / to receive) in the MINIX environment.

There are many forms (Minhonix Method) to use the Minhonix LAN. We are going to present here some examples that seem more interesting and illustrate the power of the system as a teaching tool.

3.2.1 What's a Computer Network?

In general, we begin a course with a set of experiments about the use of the system (end-user mode), as we described in the Sensibilisation Step (item 2.2) : first, the students see the connectors of the Minhonix LAN while being plugged to the stand-alone microcomputers. Later on, the teacher makes the logical connection of the machines in the LAN (he loads a "MHIX" version). Before running the experiments ("MHIX6" version, for example) the teacher must mention the heterogeneous aspect of the system (MSDOS & MINIX operating systems). It is sure that the students will have several questions. One very frequent question is "what is the difference between a LAN and a set of terminals connected to a main frame ?".

"Here, the student is a simple end-user".

3.2.2 A Simple Network Application Program

This example is almost indispensable. It still must be done in the Sensibilisation Step (item 2.2). The teacher asks the students to implement in the laboratory a simple program which sends and receives messages between two stations (It's their first electronic mail) using the procedures SendPacket and ReceivePacket (already available in the versions MHIX4) which should be analysed in the classroom.

"Here, the student is an application programmer".

The following listing shows a possible interface used by the students in this last activity. This interface corresponds to the primitives of the Minhoca LAN versions [COELHO 89] that run under the MSDOS.

```
const
      INTER_TX    = $A0;
      INTER_RX    = $A1;

type regs = record
              case integer of
                1: (ax,bx,cx,dx,bp,si,di,ds,es,flags : integer);
                2: (al,ah,bl,bh,cl,ch,dl,dh          : byte);
              end;

     packet = record
                destination : byte;
                taille      : byte;
                source      : byte;
                data        : array[1..128] of byte;
              end;

 var
      reg           : regs;
      PaquetRx      : packet;
      PaquetTx      : packet;
      message       : string[128];
      option        : char;

 { ************************  SendPacket  ************************

 procedure SendPacket(var PaquetTx: packet);

 begin
   reg.dx := Dseg;
   reg.bx := ofs(PaquetTx);
   INTR(INTER_TX,reg);
 end; { SendPacket }

 {************************  RecievePacket  ********************

 procedure RecievePacket(var PaquetRx: packet);
 begin
   reg.dx := Dseg;
   reg.bx := ofs(PaquetRx);
   INTR(INTER_RX, reg);
 end; { RecievePacket }
```

3.2.3 Access Medium Protocol

The idea here is to demonstrate the need of the access control protocol (to the communication physical medium of the LAN). For this, first the teacher uses a simple version that only sends and receives the messages ("MHIX3a" version). So, the users (the teacher or/and the students) send several messages with this version from the differents stations, in the same time. It is expected that they will see a blend of messages (collision of packets) on the screen. Later on, they should repeat the same experiment with another version ("MHIX3b" version) that has an access medium control protocol. Now, the messages must arrive correctly at the station.

Another experiment, that logically follows the one above, concerns the activity where two processes in different stations change a file while two users in two other stations play a game for example.

3.2.4 Logical Connection

The objective in this example is to show the logical connection concept. The processes in the stations A and B simultaneously, send files to the station C without opening a logical connection ("MHIX5a" version). So, the students will see a blend of the files on the screen station of C (but without collision of packets). Afterwards, they should repeat the same experiment with the version ("MHIX5b") that opens a logical connection before starting the file transfer. Now, the files must arrive correctly on the station's screen.

3.2.5 File Server

In this example, a process makes a remote access ("MHIX6b" version) to the LAN file server station. The students can follow this operation in several manners. The analysis of this source code is very rich because the file transfer is a very popular application in the computer networks and also it embeds various important concepts about operating systems (system calls, interprocess communication), computer networks (OSI protocols) and distributed systems (server-client model).

3.2.6 Internetworking

There is a version, in the Minhonix System ("MHIX10d" version), where the user in the MINIX station can use the "lpr" command (lpr <file>) to send a file to the printer branched on an Ethernet Network (of course, for this experiment we need a station connected to an existing Ethernet System).

The figure 3 presents this operation. The gateway concept is easily illustrated in this example.

4. Minhonix System Utilisations

The Minhonix System has been presented at some universities in France :

- Grenoble3 : Stendhal University (September 89)
- Grenoble1 : Joseph Fourier University (June 90)
- Paris : Institut National d'Agronomie (November 90)
- Lyon 1 : Claude Bernard University (January91)
- Paris 6 : Pierre et Marie Curie University (March 91)

A course (24 hrs) based upon the kit and the method described in this paper was given last April in "IUT Informatique", University of "Lyon 1". The result was excelent and the Minhonix System was adopted for use for the next Network System courses.

In June, in the MASI laboratory, University of Paris VI, this system will be used for the Network post-graduate courses. Five DESS students (3rd cycle diploma) would work on this system to realise their project.

5. Conclusion

We have presented in this paper the Minhonix System, a tool to use in technical course about computer networks and distributed systems. It is composed of the Minhonix LAN (teaching kit) and a Minhonix Method.

An important feature of the Minhonix System is that, prior to the analysis of a complex application program, the student can examine (and modify, if he wants) various simpler programs. Basically, each version has the same contents of the previous one, adding some new functions.

There are several ideas to the future of the Minhonix System. The first idea is to integrate the Minhonix protocols as a task (a system process) in the MINIX operating system, whose source code is easily available.

In the present, we are working with the goal of offering a more complex environment and a more user-friendly interface for the students. We are analysing the possibilities and advantages for the integration of the SUN workstation and the X Window System environment [NEY 88] in the Minhonix System. The X Window System is a network-oriented windowing system. An application need not be running on the same system that currently supports the display. So, perhaps we may take advantage of this feature in the "philosophy" of the Minhonix System. The first prototype is ready.

To conclude, it must be mentioned that if we take into account the strict definition of the distributed systems, we cannot say that the Minhonix System is one of them, but we are sure that it is a true and efficient system for teaching the concepts of computer networks and distributed systems.

Acknowledgement

The authors would like to thanks Jose Maria Coelho Filho (who has developed the first version of the Minhoca LAN at the UFC), Marly Rodrigues Gurgel (who has managed the Minhoca Project at the ETFCe) and Jose Neuman de Sousa, Joaquim Celestino Jr, Fouad Georges, (From the MASI Laboratory) for their critical reading of this paper.

Special thanks for Jacques Briat at the Joseph Fourier University (IMAG Laboratory). His ideas, comments and suggestions added greatly to realization of the Minhonix System.

A. Mauro B. Oliveira is supported by CNPq (Conselho Nacional de Desenvolvimento Cientifico e Tecnologico - Brazil) / CESELEC (Centre d'Etudes Superieures en Electricite, Electronique et Informatique - France) contract.

FIGURE 1 : Internetwork System for a Teaching Environment

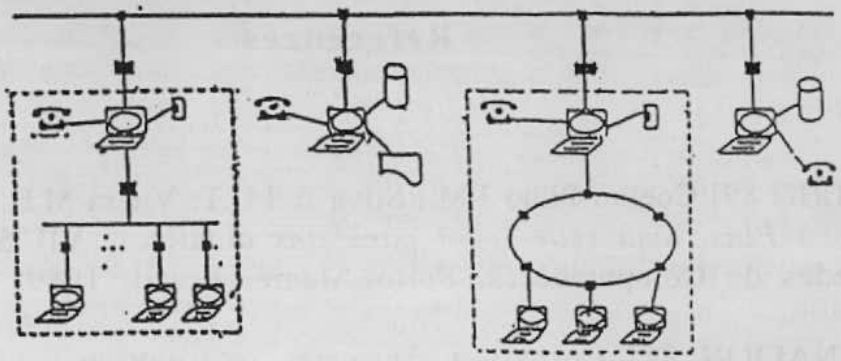
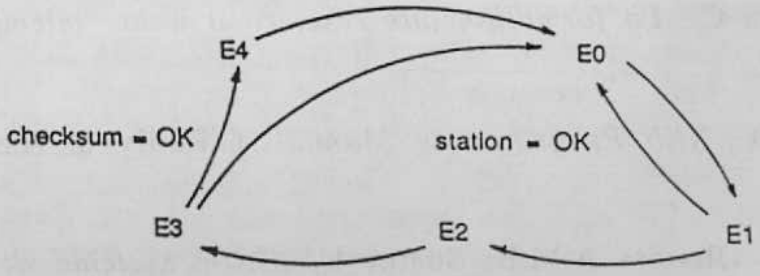
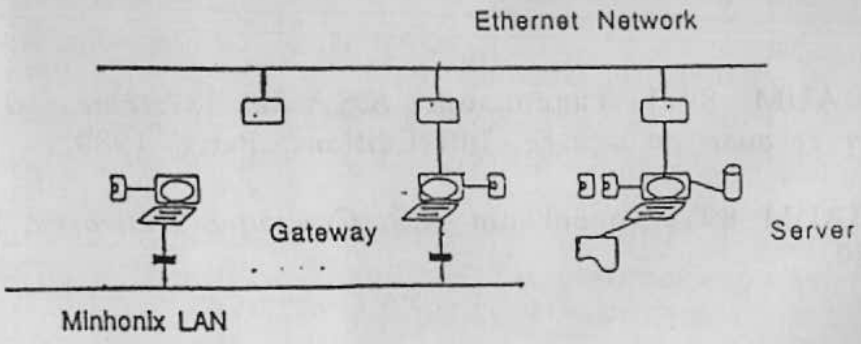


FIGURE 2 : The state machine of the ReceivePacket procedure



E0 = Idle state.
E1 = Reception of the character from the physical medium.
E2 = Reception of the packet size byte.
E3 = Reception of the rest (characters of the packet).
E4 = Filling of the reception buffer (if checksum = OK).

FIGURE 3: The internetworking



References

[COELHO 89] Coelho Filho J.M.; Silva Jr J.C.T; Vieira M.H.B.; Oliveira A.M.B, *Minhoca Plus, uma rede local para fins didaticos,* VII Simposio Brasileiro de Redes de Computadores. Porto Alegre, Brazil, 1989.

[CORNAFION 81] Cornafion, *Systèmes informatiques répartis,* concepts et techniques, Dunop. Paris, 1981.

[DUNCAN 86] Duncan R., *Advanced MSDOS*, Microsoft Corporation. USA, 1989.

[NORA 85] Nora C., *La formation aux réseaux et à la télématique.* Paris, 1985.

[NEY 88] Ney A., *Xlib Programming Manual,* O'Reilly & Associates, Inc. USA, 1988.

[OLIVEIRA 87] Oliveira A.M.B; Soares L.F.G,*Um systema de computacao interredes para aplicacao em um ambiente educational*, XIV SEMISH, Salvador, Brazil,1987. The thesis of the Master of Science, Department of Electrical Engineering, PUC/RJ. Rio de Janeiro, 1987.

[OLIVEIRA 88] Oliveira A.M.B, *Minhoca un reseau local didactique,* CUEF, Universite Stendhal. Grenoble, France, 1988.

[OLIVEIRA 90] Oliveira A.M.B, *Minhonix un système distribué didactique,* DEA informatique, IMAG, INPG. Grenoble, France,1990.

[SOARES 86] Soares L.F.G.; *Redes Locais*, Editora Campus. Rio de Janeiro, 1986.

[TANENBAUM 89a] Tanenbaum A.S, *Les systèmes d'exploitation, conception et mise en oeuvre,* InterEditions. Paris, 1989.

[TANENBAUM 89b] Tanenbaum A.S, *Computer Networks,* Prentice Hall. USA, 1989.