CENTRO DE INFORMÁTICA

DA UNIVERSIDADE NOVA DE LISBOA

ONLINE: A PORTABLE PROGRAM TO GIVE A PERSONAL COMPUTER FULL ACESS TO ANOTHER COMPUTER.

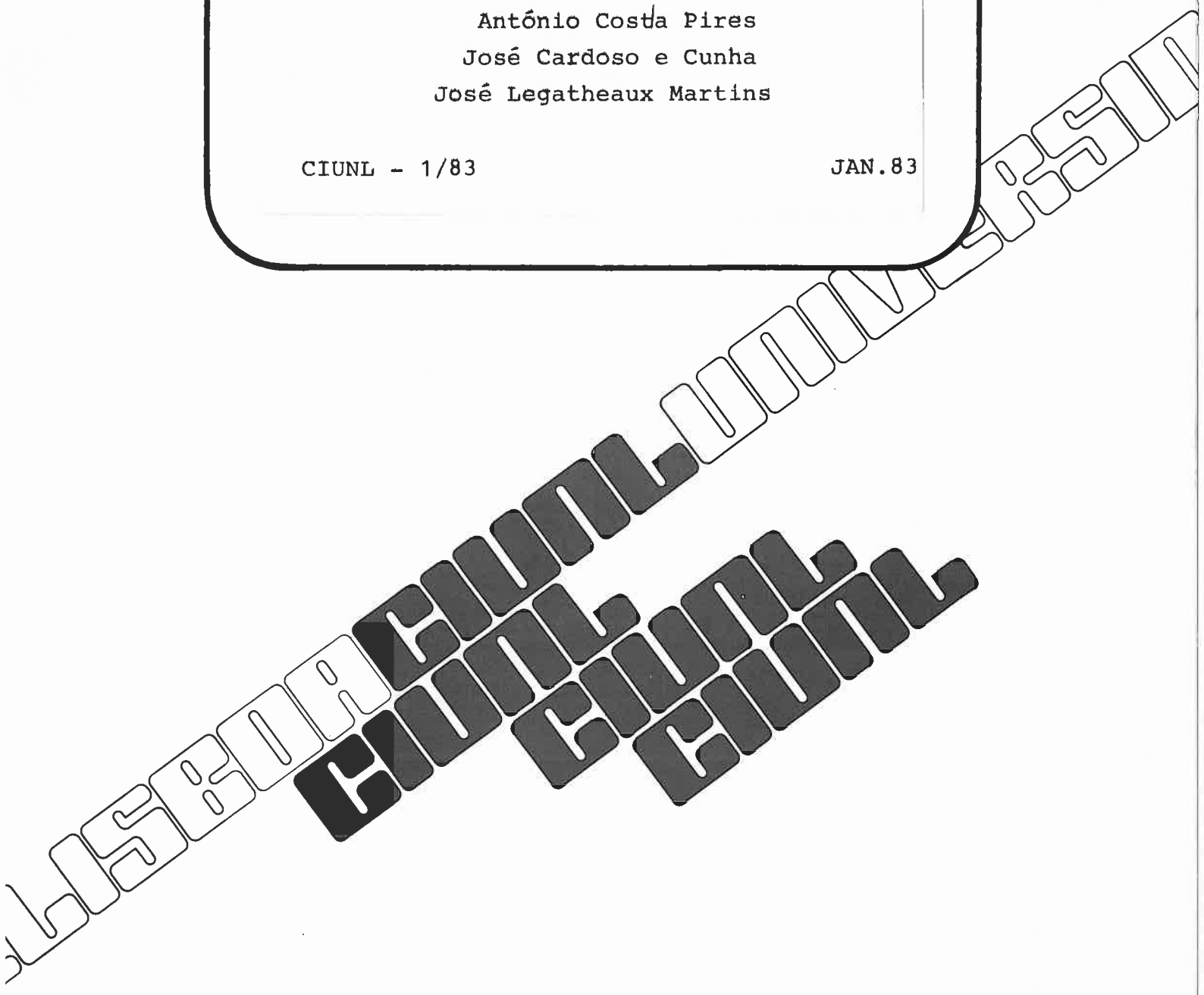
por

Antônio Costa Pires

José Cardoso e Cunha

José Legatheaux Martins

CIUNL - 1/83 JAN.83



ONLINE: A PORTABLE PROGRAM TO GIVE A PERSONAL COMPUTER FULL ACESS TO ANOTHER COMPUTER.

por

António Costa Pires

José Cardoso e Cunha

José Legatheaux Martins

CIUNL - 1/83 JAN.83

ONLINE: A PORTABLE PROGRAM TO GIVE A PERSONAL COMPUTER FULL ACCESS TO ANOTHER COMPUTER.

by

Antonio Costa Pires

Jose Cardoso e Cunha

Jose Legatheaux Martins

Centro de Informatica da Universidade

Nova de Lisboa.

Portugal

Abstract : 'ONLINE' is a portable program to make a personal computer look like a terminal of another computer system. Besides allowing file transfer between systems, 'ONLINE' allows the emulation of the environment, at the local system, of a typical host system user working at an interactive terminal. The strategy of abstract machine specification and design was followed with success, without sacrificing the efficiency requirements.

CR Categories and Subject Descriptors: D.4.4 [Software]: Operating Systems - Communications management.

General Terms: Design.

Additional key words and phrases: Computer communications, abstract machines, portability.

Author's Present Address: Centro de Informatica da Universidade Nova de Lisboa, Quinta da Torre, 2825 Monte da Caparica, Portugal.

1. Introduction.

Following the trends towards communicating computer systems, we have designed a program that will enable the users of several (possibly heterogeneous) computer systems to easily access other environments.

The communications link required is just a serial line connecting the standard serial interface of the personal computer to a terminal interface of another system.

The distinctive features of this program are its portability and its flexible user interface.

The proliferation of personal computer systems with specific architectures and operating systems makes it difficult to construct communications software that is both easily transportable and

efficient. As a matter of fact, most of the terminal emulating programs developped for microcomputer|systems do not satisfy, to our knowledge, the portability requirement.

When developping this program we have tried to clearly identify the machine and operating system dependent features, and we did isolated them on well defined points. We claim that the strategy of defining an abstract machine to support the low level interface with the i/o system is highly successfull for portable systems.

It also proved to be compatible with efficiency requirements.

The only requesites the host computer must satisfy are as follows:

i) a full duplex serial transmission line;

ii) the host (non local) operating system must be interactive and it must recognize a Start/Stop protocol (TTY-like).

There is no dependence on that host system, and some of its intrinsic features (refering mainly to control character conventions) may be easily parameterized and adjusted when installing 'ONLINE' on a particular environment.

2. The user interface.

The specification of the user interface was guided by the following requirements:

i) the local computer user must have a quick way of switching between two distinct logical environments, i.e. the one supported by the local operating system and language processing facilities, and the environment of the host system;

ii) the environment switching must be easily performed by the user, invoking simple commands, and he must be informed of wich of the environments he is currently working on;

iii) the user must not be aware of the low level details of the implementation, but he must be able to verify the coherence and correctness of the actions required (for instance, when transfering files between systems he must have the possibility of checking by himself the correcteness of the transmission), and eventually to abort them.

The program supports the following modes of operation:

i) 'online' mode: in this mode the local computer is viewed as a terminal of the host system, having full access to its facilities. Any character typed in the local keyboard will be sent to the host system, and any character coming

from the host computer will be displayed on the local screen. By typping a special control character the user may leave this mode and enter a 'menu mode' to select other functions. Besides displaying a brief text on its several options ,'ONLINE' also constantly displays its current state and the actions it is expecting from the user.

ii) 'Disktohost' and 'Screentodisk' modes: these modes provide a file transfer facility between systems. 'Screentodisk' is functionally equivalent to 'online' mode with an additional feature - all the characters received by the local system will be written into a previously opened local file, besides being echoed on the screen. 'Disktohost' mode lets the user start the transmission of a local file to the host system.

3. The implementation.

'ONLINE' is a program, written in Pascal, designed to achieve a reasonable (high) degree of portability. It runs standalone on a local computer, allowing a 'quick' return to the local operating system environmemt whenever the user wants to go back to local mode.

We have considered the portability problem from both aspects of programming language and program structure.

Wherever possible the program was conceived to use Pascal constructs with no portability problems. The nonportable features (

e.g. string manipulation, file opening and closing) amount to only a few lines of source text and are clearly identified.

Another concern in 'ONLINE' design was to support its adaptability to some intrinsic features specific to the local and host systems. So, the program is easily adjusted (parameterized) to support several features, namely:

* internal representation of the control characters required by 'ONLINE' (e.g. start/stop characters, endofline character as recognized/sent by the host, etc.);

* specific features to the local terminal (e.g. number of nulls after carriage return, reverse video, bell, backspace echoing. etc.).

One of the critical problems, when producing portable software is the i/o system defined by the language. This was agravatted, in our case, because of the specific objectives of the program.

We had two critical points, regarding the i/o problem, namely:

* the communications (serial) line i/o system;

* the local user terminal interactive dialogue for 'ONLINE' operation.

We have followed the approach of abstract machine modelling to deal with all the i/o, during 'ONLINE' operation (except for disk access, where we use the Pascal file access primitives).

This enabled us to clearly identify the machine dependent features, and also to avoid some intrinsic features of the i/o systems.

We have defined an abstract machine to model the low level i/o of the serial line and the user terminal (tty). It supports two logical devices, accessed via several primitives, such as 'deviceon' / 'deviceoff' (to activate / disactivate the device), 'deviceok' (giving the current status of the device) and 'getdevice', 'putdevice' to receive and send characters, one at a time.

The procedures for line and tty control perform their actions by cyclically testing the device status. This avoids the intrinsicacies that would appear if we had used an interrupt driven system.

An asynchronous communication scheme with buffers is used to support any mismatch between the line and the terminal transmission rates. Buffer overflow is prevented, and a start / stop protocol is used to regulate the flow of characters, in a smooth way, providing no data loss. Several features of the buffering scheme are parameterized and may easily be adjusted at installation time (e.g. buffer size).

4. Conclusions.

'ONLINE' has been installed on several personal computer systems, with a reduced implementation effort (just the writing of the low level i/o primitives for each hardware configuration, plus some minor adjustments). It is currently working on several 8085, LSI11, and Z80 based systems. With 8085 (6 Mhz) and LSI11 it supports a transmission line working at 9600 baud. With Z80 (2 Mhz) based systems it limits the transmission speed to 4800 baud.

It is our feeling that we have succeeded in producing a reasonably portable program without sacrificing the efficiency requirements.

------o------