

# IBM-PCs in the VICE/VIRTUE Environment

*M. Satyanarayanan*  
*Information Technology Center*  
*Carnegie-Mellon University*  
*12 January 1984*

IBM PCs, in varying configurations, are currently proliferating the CMU campus. There is a perceived need to attach these machines in some manner to VICE. This memo proposes a particular attachment strategy and examines its merits and shortcomings.

Germane to this discussion are the following observations:

- The exact manner of PC attachment is a degree of freedom available to us. There are no externally imposed constraints on the level of transparency achieved, or on the subset of VICE facilities that are to be made available to PC users.
- The design of the VICE file system is tuned to the problem of efficient access by a very large number of users to physically distributed data. Strategies such as whole-file transfer and caching are critical pieces in the solution to the problem. However, such a design assumes a workstation that is considerably more sophisticated in hardware and software than a PC running MS-DOS.
- Being forced to cater to the needs of a relatively unsophisticated workstation will burden and slow the implementation of the VICE/VIRTUE prototype. *It already has.* The attachment impacts network hardware, transport protocols, application protocols, and VICE file system capabilities. A priori constraints on any of these cannot but slow our prototype implementation.

## The Proposal

What is proposed here is a strategy that balances the need for PC attachment with the need to decouple this attachment from the VICE/VIRTUE prototype development.

The key elements of the proposal are as follows:

- Write an application program, called *PCServer*, on Unix.
- *PCServer* allows a number of PCs to attach to the Unix file system on the machine on which the program is run. This is a single-site file system, NOT a distributed file system. The number of client PCs is small, say 10, and static.
- The basic Unix file system operations (Open, Read, Write, Close, and Seek) are the functions provided to the PCs by *PCServer*. It is effectively providing remote open, with byte-at-a-time access to the Unix file system.
- The connection to the PC may use any of the standard network protocols, *or even a 9600*

*entered 7/27/84*

*baud RS-232 line.* The application protocol may be designed to suit the needs and capabilities of PCs.<sup>1</sup>

- A simple authentication strategy is used: the PC looks like a user logging in (a la uucp).
- The PC file system is modified to trap calls to a particular device. This device fulfils the function of "current working directory" in Unix.
- A new DOS system call allows this special device to be bound to a Unix pathname. Future accesses to files on this device are prefixed by this pathname and submitted to PCServer, which performs the appropriate operation. Bytes are read or written on demand from PCServer.
- All of the above steps can be done *completely independently of VICE/VIRTUE development.* When the VIRTUE prototype is available, run PCServer on it. Since PCServer is an application program, it transparently accesses files in VICE. PCServer and the PCs it is serving merely see an enlarged Unix file system; they do not have to deal with the distributed nature of VICE.

## Merits

The main benefits of pursuing this strategy are as follows:

- The development of VICE/VIRTUE and PCServer can be carried out independently of each other. Neither serves as a constraint on the other. We have reduced the problem of attaching PCs to VICE to two subproblems. One of these, VIRTUE, is already being solved. The other is a much simpler subproblem because it deals with a single-site file system rather than a distributed one.
- We can start development today on our Suns. Sources are not critical since no Unix intercepts nor kernel changes are necessary. In fact it is desirable that no such changes are made, since that would enhance the portability of PCServer.
- The approach provides transparent, byte-by-byte access of remote files for PCs, not just whole file transfer. This is important because of the many small PC configurations in existence.
- Application protocols can be tailored to suit the PC's needs and capabilities. The optimizations relevant to the PC may be different from those of VICE/VIRTUE.
- PCServer is useful even with just simple Unix systems. This provides us with a cushion if VICE/VIRTUE runs into heavy weather.

---

<sup>1</sup>A version of the VICE/VIRTUE RPC protocol, without bulk-transfer, exists and can be used if it seems acceptable.

## Demerits

Playing the role of devil's advocate, I present below potential criticisms of this approach, along with rebuttals. No claim is made that this is an exhaustive list: only that these objections are the ones apparent to me at present.

*PCServer is a completely separate task, requiring additional manpower*

Absolutely true. However, this very independence is an advantage since it decouples the VICE/VIRTUE development from the PC attachment problem.

I estimate that an experienced Unix programmer (not necessarily a kernel hacker) can put together PCServer in about 2 months. At the PC end, this strategy is simpler than a direct VICE attachment because one does not have to deal with multiple custodians, and the myriad other problems of a distributed file system. A working version using RS232 lines can be used initially, thus decoupling from the PC network attachment problem.

*This approach does not demonstrate that many different machines can directly attach to VICE*

Also true. But I question whether such a proof need be simultaneous with development of the VICE/VIRTUE prototype. It can come after the first prototype is built. The fundamental purpose of the prototype is to debug our design and validate our assumptions. Every additional system that we are forced to interface with from the beginning will constrain us when we try to rectify our design errors.

The PC running DOS is a less sophisticated system than the potential future VIRTUE systems. I believe that it is reasonable to require a minimum threshold of functionality from systems that desire to attach directly to VICE.

*This strategy treats the PC as a second-class citizen.*

This is an emotional, rather than a rational concern. In any case it is not accurate. The extent to which an attaching system is able to use VICE functions, and the manner in which it does so, depends on its capabilities. What we obtain by this approach is transparent access to a Unix file system, and incidentally to VICE files. What we are missing (at least initially) is direct access to VICE facilities. Is it really essential that PC users have full access to VICE? The software, more than the hardware, complicates the PC attachment: perhaps PC/IX may be attachable as a full-fledged VIRTUE.

*Requires additional hardware for deployment*

Also a valid objection. But I do not believe the cost to be excessive. An IWS should easily be able to serve at least 10 PCs. I believe that this is an acceptable tradeoff, in view of the benefits of the approach. It should also be noted that this approach allows threadbare PC configurations to use VICE, whereas direct VICE attachment may require a richer PC configuration.

## Conclusion

In summary, it is proposed that we completely separate the problems of the VICE/VIRTUE prototype and the single-site Unix file server for PCs. What we gain is an important reduction of interdependencies in our task and a reduction in our design and implementation constraints. We pay for this in two ways: an additional person to implement PCServer and additional deployment cost. It is my belief that this is a bargain.