CMU-ITC-83-031

ITC File System Goals

12 September 1983

File System Group

Information Technology Center
Carnegie-Mellon University
Schenley Park
Pittsburgh, PA 15213

CONTENTS

INTRODUCTION

This is the file system part of an overview of the ITC's project goals. Its purpose is to tell you what using the VICE file system will be like, so that you have a chance to influence our design now and we have a target to shoot at.

VICE provides communications and shared file storage for the ITC network of workstations. Communications, including the local area network and also the set of "gateways" to existing campus computers, is discussed elsewhere, as is the VIRTUE workstation. Here we describe the file system, including storage and movement of files, protection and authentication, accounting and space control, dealing with network failures and independent operation, and certain applications such as mail and bulletin boards.

Ordinarily you can write programs and manipulate files as if the file system were stored on a single very large computer system. The workstation will contain interfacing software whose job it is to insulate you from the details of network interfacing, copying files to and from the workstation, and so on. However, the network won't be completely invisible:

- You'll notice performance differences because locally stored files are easier to get to.
- You can, if you wish, control file placement, bypass the interfacing code, or even build your own workstation.
- The file system will attempt to provide (possibly limited) services even if the workstation is disconnected from the network or the network itself is partitioned.

Therefore, a little terminology about the structure of the network is in order.

The network will be organized into interconnected "clusters" of workstations. Each cluster connects some number of workstations and at least one "cluster server" machine, which stores files and runs other VICE applications. For security reasons, cluster servers run only VICE applications. The hardware and software of the attached workstations may differ, but they can all use the services provided by VICE if they support the underlying communication and application protocols.

Within this framework, there are two file system interfaces: the VIRTUE interface, through which programs and user commands running in the workstation see files stored in VICE and/or locally, and the VICE interface, used by the workstation (and potentially by any machine connected to the network) to store, share, and retrieve files. You will seldom deal directly with the VICE interface, which is really intended for machines, and will usually deal with the VIRTUE interface by running programs which read and write files.

What will using the file system be like? This section attempts to give you simple examples, with most of the detail suppressed, of what things you can do with the ITC file system.

ORDINARY FILE ACCESS

When you start using a workstation, you "log on" by providing a password and your name. The workstation goes through an authentication procedure to establish a secure connection with VICE. Since only your workstation can use the connection, and you're in control of the workstation, VICE treats the connection as your ticket to your files. (It will be up to you to "log off" when you're done, to prevent somebody else from misappropriating the connection and masquerading as you.)

When you first use a file, the workstation will copy it automatically to its local "cache" of files. Thereafter it will use the cached copy directly, checking with VICE to verify that nobody else has updated the master copy. If you modify the file, the workstation will copy it back to VICE automatically. If you stop using the file, the workstation will eventually discard the local copy in order to make space for other files.

Some widely-used files, such as compilers or popular applications programs, will be replicated at every cluster server. VICE will update such files periodically, for example once a day. You can force an update if you really must have the most recent version.

SHARING

You can use your files from any workstation, regardless of where they're actually stored. For example, you can log on to a public workstation anywhere on campus and use your files normally, although possibly with a performance penalty if the files must be moved a long distance.

Only you have access to your files to start out with. You can, however, allow other users or groups of users to perform selected operations on your files. For example, a research project can set up a shared set of files which any member of the project can read and update.

MAIL AND BULLETIN BOARDS

Mail and bulletin boards are a widely-used example of file sharing. VICE will support them through its tree-structured directory. Every user who wishes to receive mail will have a "mailbox" subdirectory into which other users are permitted to store messages in the form of files. Only you can remove and read messages from your mailbox.

Bulletin boards will resemble mail, except that a whole group of users can read the messages. Stale messages will be removed by the bulletin board's owner.

VIRTUE will provide application programs to generate, categorize, and read mail and bulletin boards; the file system simply stores them.

LOCAL FILES

You can force files to be stored only in your workstation by placing them in a special "/local" subdirectory of your workstation's file system. This provides some additional security for very sensitive files, but you won't be able to get at them from any other workstation. A more important application is to store the programs and data needed to get the workstation started and connected to the network when its power is turned on.

INDEPENDENT OPERATION

Assuming your workstation has sufficient local storage to hold your files and programs, you should be able to use it even if it is disconnected from VICE. This could happen if you take the workstation off-campus, or if there is some failure in the network or cluster server. This is a special case of what is called "partitioned" operation, meaning that the network has been divided into two or more independently operating parts which can't talk to each other.

Your workstation will retain file updates until reconnected to VICE, then merge your local changes with any that happened in the rest of the network. We expect that conflicting updates will be relatively rare, but will attempt to detect them by keeping some sort of update log.

The ability to operate independently requires that your workstation have a local disk with reasonable performance. We may also provide for diskless workstations (which would not be able to run independently) in order to reduce the minimum cost of a workstation. This decision depends on technological and performance considerations not yet available. Diskless workstations would depend on the cluster server to store almost all of their files.

The following sections deal in somewhat more detail with the various major components of the file system.

VIRTUE INTERFACE

The VIRTUE file system looks and behaves like a single-site Unix file system, with a tree-structured directory. The VIRTUE user interface is expected to provide a convenient and easily understood set of user-level commands to manipulate files. The file system interface proper is in the form of "system calls" through which application programs can create, locate, read, write, close, and control files.

With one exception, the VIRTUE file name space is identical to that of VICE. The exception is the subtree "/local," which contains files strictly local to the workstation. Symbolic links from the VICE name space to files in "/local" may be used to achieve workstation-sensitive file name interpretation.

The VIRTUE system calls are intended to resemble standard Unix system calls closely enough that it is easy to port Unix application programs to the workstation. Exactly how compatible the interface is depends on which version of Unix you're talking about; there's no universal standard. Some incompatibilities known at the present time are:

1. Symbolic links only.

   Most Unix systems provide "physical links" between files. The key property of physical links is that if somebody else replaces or deletes a file to which you have a physical link, you retain access to the *old* version of the file. By contrast, a symbolic link is no more than an alternate name for a file. Changes to the file change it under both names. We feel that physical links between cluster servers will be too hard to implement, so we will provide only symbolic links.

2. Protection.

   While maintaining reasonable compatibility with Unix, we may decide to implement a more flexible and/or robust protection system than that of Unix.

3. Directory Structure.

   Programs which read directories as files, expecting a certain format, may fail or run slowly.

4. Concurrent Updating.

   It may not be possible for two programs to update or append to the same file concurrently.

VICE INTERFACE

The VICE interface is designed for simple and efficient communication with VIRTUE and possibly other programs or workstations. Thus it's of interest only if you intend to bypass the VIRTUE interface and write a program which talks directly to the network.

The VICE file system stores files and identifies them with a Unix-like hierarchy of directories. Naming conventions and file formats follow Unix; for example file names are case sensitive, and a file is treated as an arbitrary sequence of 8-bit bytes with no pre-specified internal structure.

Our strategy is to transfer whole files between the workstation and VICE, so the interface is based on file Fetch and Store operations employing an as-yet unspecified file transfer protocol. In addition to these basic operations, a workstation will be able to lock and unlock files, to read and write file descriptor information, and to create and destroy empty subdirectories, files, and symbolic links between files.

At a later time we may extend the VICE interface to include not only whole file transfer but also a way to open a remote file and then read or write pieces of it. Such a facility may be required for efficient access to large files, such as data bases. It is also one of way to implement diskless workstations.

PROTECTION

NOTE

This section has not yet been agreed to.

Since we feel that the standard Unix protection mechanism is inadequate for a large, diverse user community, we propose to replace it with an access-list system.

In such a system you associate with the objects you wish to protect a list of other users (or groups of users) and the operations each is allowed to perform. In the VICE file system, the protected objects are directories, and the operations are:

create store new files in the directory

read read files already in the directory (this includes running programs)

list obtain a list of the files in the directory and their properties (but not their contents)

update modify and delete files in the directory, and also update the directory's access list.

For example, you would grant only "create" access for your mailbox to the group of all users. Access controls apply to all files stored in a directory; if you want a particular file treated specially you must put it in a special directory.

There is one exception to the access lists: the "owner" of a file always has full access to it. This avoids paying to store a file you can't delete.

ACCOUNTING

One of the conveniences of using a workstation is that you needn't bother accounting for its use. Unfortunately VICE provides shared services which must be accounted for.

THe VICE accounting system will be based on a "banker", or accounting server, which will maintain a working balance for each user. Use of system services will debit this balance. For example, disk space will be accounted for by periodically scanning the disks and charging the account for the space used; communication costs will be charged when sessions are closed. Other VICE services, such as printing, will operate similarly.

Above and beyond accounting for VICE usage, the accounting server should provide way for you to transfer part of your working balance to another user. You can use this mechanism to allocate computing resources within an overall group budget, and possibly to "pay" for services rendered by a non-VICE server.

RELIABILITY

We intend that VICE be at least as reliable as a conventional time sharing system, and we hope we can make it much more so. Our strategy for accomplishing this is based on redundancy:

- There will be more than one cluster server and a way to bypass and/or replace any one which fails.
- Key files will be stored at least twice on different disks (or other media.)

- VICE servers will run only VICE code, not user programs, and so should be more reliable.

- Workstations can continue to operate with cached data if disconnected from VICE.

In other words, we plan a "suspenders and belt" strategy in which individually reliable cluster servers back each other up both for storing files and getting to them.