Digital Objects and Directories

Peter J. Denning

5/20/19

A *digital object* is a digital representation of something in the world. The representation is inscribed into a medium, where it can be moved, copied, and manipulated. Examples abound -- books, papers, articles, photos, videos, movies, soundtracks, albums, contact books, parts catalogs, 3D part production plans, and much more. Digital objects are proliferating as we learn to digitize almost everything. Since operating systems are intended to help us manage our digital resources, it comes as no surprise that operating systems include many functions for defining and interacting with digital objects.

Operating systems treat four kinds of digital objects as primitive objects because they are so central to the creation, storage and manipulation of other digital objects. They are: *file*, *device*, *pipe*, and *directory*. Every OS kernel has highly optimized subsystems for these types and kernel calls to do the manipulations.

To add another type of digital object, such as mp3 recordings, we design a service process (daemon) for those objects. The service process would create and delete mp3 objects, and manipulate mp3 objects. The capabilities pointing to mp3 objects would be tagged "mp3". The service process would employ primitive digital objects, such as files to hold mp3-formatted data.

Stream Objects

The file, pipe, and device digital objects are called *stream objects* because their content takes the form of bit streams. A stream can be a definite size, such as the content of a file, or it can be an ongoing series of bits, such as the signal played on a digital loudspeaker or the signal output by a video camera. The three primitive stream objects recognized by the kernel are:

- A *file* is a named container of a stream, of definite length.
- A *pipe* is a conveyor channel for a stream. It incorporates a synchronized producer-consumer relationship between the transmitter and the receiver – each transmitted bit is received exactly once and in the same order of transmission. The receiver must stop and wait until the sender sends more bits.
- A *device* can be a *source* or a *sink* for a stream (e.g. keyboard / speaker).

The stream objects can be accessed by a common interface in the kernel. The main kernel calls are `READ` and `WRITE`. Reading a digital object pointed to by stream object capability `c` is done by the kernel call `READ(c)`, which copies the stream from `c` into the calling process's workspace. The `READ(c)` program calls one of

three other kernel programs – `READ_FILE(c)`, `READ_PIPE(c)`, or `READ_DEVICE(c)` – depending on the type indicated in the capability `c`. The same is true for the kernel call `WRITE(c)`.

Directories

A *directory* is a table associating one or more symbolic names (chosen by a user) with a capability (chosen by the OS). For example, if the user chooses symbolic string `N` for capability `c`, the directory has an entry `(N,c)`. The user can get a capability by searching the directory for an entry with a given `N,` then retrieving the associated entry `c.`

A directory is a digital object, but is not itself a stream of bits; directories do not interact with streams, they point to objects that do. Directories cannot be connected to standard input or output of a process. Although directories are often implemented and stored in a filesystem, looking much like a file with a special directory format, directories are not file objects: do not confuse the two.

Directories are almost always arranged in trees: some entries in a directory point to other directories. The interior nodes of these trees are directories. The leaves are either directories or stream objects. The root of the tree is a single node, common across all users on a system. In Unix, for example, the root is designated "/" and contains a directory `/usr` that lists the roots of all the individual user trees.

A *pathname* is the sequence of names from the root of a directory tree to a particular digital object. The component names are separated by a standard symbol, such as backslash in Windows or forward slash in MacOS and Unix. Because there is only one path from root to any other node in a tree, pathnames are unique names for digital objects. As we have seen, the Web URL embeds a pathname to uniquely specify a file on a host.

A process – implemented by virtual machines at the next higher level of the OS – contains two environment variables referring to directories. One is `CD`, a capability pointing to the current directory and the other is `PATH`, a list of one or more directory pathnames used to organize a search for a particular object. The process uses these variable as parameters to the downward calls on the directory manager.