HONEYWELL INTEROFFICE CORRESPONDENCE

DATE: APRIL 2, 1971

TO: K. VAN VLANDREN

FROM: C. T. CLINGEN

DIVISION: CISL/PCO

cc: CW Dix

bcc: FJ Corbató
RA Freiburghouse
JW Gintell
JH Saltzer ✓

SUBJECT: NOTES ON ACCESS CONTROL IN A HIERARCHICAL FILE SYSTEM

Here are some thoughts that may help you satisfy your request for information on file protection. The style purports to be semi-popular and is not intended to be technically precise. I have not constrained myself to methods implemented in Multics; some GECOS techniques are mentioned as are some as-yet-unimplemented ideas.

Good luck and give me a call if this needs further clarification.

C. T. CLINGEN, Manager
Cambridge Information
Systems Laboratory

/ll
(enclosure)

Access control techniques like the ones described here allow each user of a computer facility to control the entire range of accessibility of his files -- from totally private to shared in a controlled manner to totally public.

It is this intermediate ability-- to control sharing -- which offers much promise for harnessing the problems of multiple-access computer data banks. These techniques, taken together with further refinements such as the Multics ring protection mechanism, are already beginning to provide useable solutions to many issues related to information protection and security in large computers.

C. T. CLINGEN
4/1/71

SOME ACCESS CONTROL MECHANISMS FOR FILE HIERARCHIES IN DIRECT ACCESS COMPUTER SYSTEMS

A simple and convenient way to classify files of information in a computer is to group them by common topic, subtopic, etc., much in the same way that the Dewey Decimal System or the format of an outline groups together common information. This is frequently accomplished by constructing hierarchies of catalogs which begin with high-level descriptions of the encompassed files and refine the classifications with increasing discrimination at each subsequent level. The files containing the actual information appear at the "bottom" of this hierarchy of descriptive catalogs. Figure 1 shows an example of such a hierarchy as an inverted tree of catalogs terminated by files.

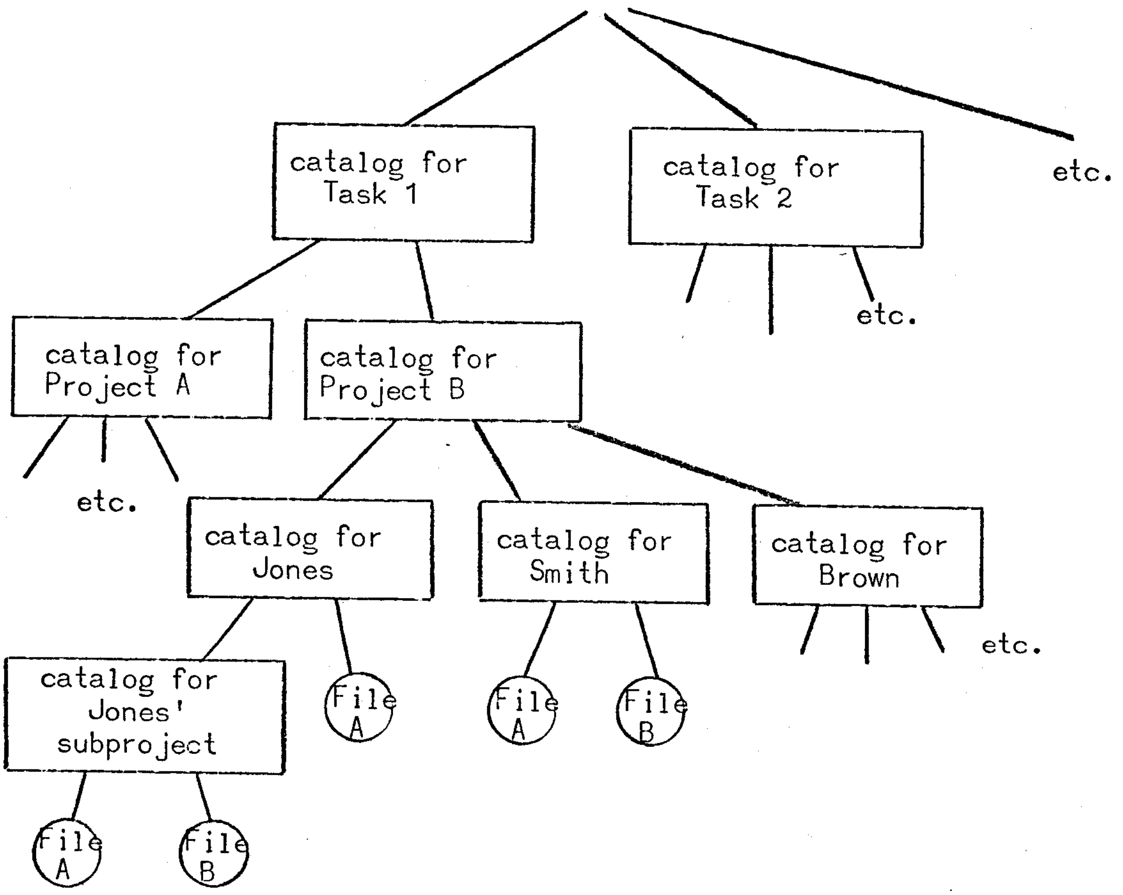

FIG 1. - CLASSIFICATION OF FILES BY TASK, PROJECT, PERSON AND SUBPROJECT

Availability of such descriptive structures in computer systems encourages the use of effective but simple conventions for locating information. Figure 1 illustrates one possible scheme in which a large number of on-line files may be classified by subproject, owner, project and finally by task.

Once such conventions have been established it becomes a relatively simple matter to specify and access one of the many files which reside in the system. This is done by specifying the path from the top of the hierarchy to the desired file. For example, File A in subproject Jones is uniquely specified by the following ordered sequence (starting from the top):

Catalog for Task 1
Catalog for Project B
Catalog for Jones
Catalog for Jones' Subproject
File A

This type of naming convention makes it easy for each user of a direct access computing facility to locate and share information placed in the system by other users of the system. The ability to share files has been proved extremely valuable to the users of such systems as Multics; however, this sharing capability immediately introduces the requirement for effective control of the access to each file in the hierarchy. The determination of which users may access which files in the system is called access control.

Most access control systems are based upon descriptive information contained in the catalog immediately superior to the file or catalog being accessed. In this way each catalog and file in the system may have access control information associated with it. Typically the access control information consists of a list of names of people who may access the associated catalog or file together with the exact type of access to be permitted for each name. Simple access types for a file are read and write and list and change for a catalog.

When a user wishes to access a file or catalog, the access control mechanism first searches the access control information for that file or catalog, looking for his name. If his name does not appear, he is not given any access to the specified information; otherwise, he is allowed to perform just those operations specified and no others.

A user's authority is in some sense proportional to the level in the hierarchy of the highest catalog which he has permission to modify. For example, in the hierarchy depicted in Figure 1, Jones may have change permission for catalog Jones implying that he can alter access control information for his files and catalogs. The user with change permission for the Task 1 catalog has a much greater implied authority, however, as he can modify access to the directories and files for all inferior projects, users, subtasks, etc. This hierarchical structuring of access authority is useful as it corresponds closely with the usual structure of organizations doing work on the system.

This approach presumes that the true identity of the user has been accurately determined before the first file or catalog access is attempted on his behalf. The identity of each user is normally verified by a sequence of names and passwords as he first logs into the system.

If a user's identity is not considered sufficient to determine the access to a file or catalog, additional protection can be afforded by extensions to the access control mechanism described above. One extension permits a password to be associated with the access control information for a file or catalog. This allows the owner of highly sensitive information to place a series of "traps" along the path of catalogs leading to his files. As a potential user approaches these files he must provide the proper passwords in the proper sequence before access is granted. Another variation replaces the file passwords with programs which may perform arbitrarily complex access requirement checking as specified by the owner of the information to be protected.