Memorandum on

Jones' Name Interpretation Paths and Protection

and

Some Aspects of TENEX Files that Allow "Type" Casting

Chuck Wall

Feburary 1, 1974

Technical Memo TM.74-3

Jones defines a name interpretation function f_E that "maps the name space associated with environment E to the set of all objects."

$$f_E: \{names\} \xrightarrow{\text{interpretation path}} \{objects\}$$

Jones then states that: "Because name interpretation necessarily accompanies each exercise of a right, correlating the performance of name interpretation and protection checking will guarantee that all requisite checks are made in support of the Enforcement Rule." Now as Jones specifies protection checking can be performed:

1. At the beginning of the name interpretation path, i.e., the site at which names are generated
2. At the termination of the interpretation path, i.e., the side of the referenced object, or
3. At an intermediate stage along the interpretation path by referencing data structures not needed for name interpretation.

The basic problem we face right now is relating these abstract ideas to specific objects in TENEX. This memo simply suggest a couple of possibilities.

A candidate for access site protection checking are the JSYS calls. If it is reasonable to consider each JSYS as an individual procedure (in a logical sense) then it is reasonable to associate an environment with each JSYS (which may also be passed rights in the form of parameters) in order to encapsulate it.

The TENEX file system lends itself to protection checking both at the object site and along the way to the object (i.e., along the interpretation path). The TENEX file designator has the following form:

device: <directory name> file name.extension;version number

There currently is some protection associated with directories and the file object itself. There are also some restriction on device usage, but this is because of their particular physical restriction rather than a protection consideration. I think it is reasonable to apply protection to each element in the name and when the JOB or process is given the handle on the file all other access to the file could be a right which is equivalent to the intersection of the rights encountered along the way to the object. For example, an extension may be one of the following:

Extension	Meaning
.MAC	MACRO-10 source program
.BAS	BASIC program
.F4	FORTRAN IV source program
.REL	relocatable object program
.SAV	an executable object program

So a read access from a text editor to a .SAV file should be stopped. Naturally, all this depends on the policies to be implemented. The format of file naming could possibly be the basis for the partitioning of files thus creating other types of objects.

This just represents a couple of ideas I wanted to get down on paper and think about a little bit.