JFN Protection

Jack Freeman June 12, 1974

TM.74-23

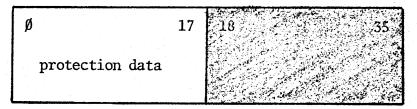
JFN Protection

I. Changes to Data Structures

As a minimal change to the current protection of JFNs, we suggest a simple extension of the "restricted to this fork" feature.

We add a "protection field" to the JFN data structure.

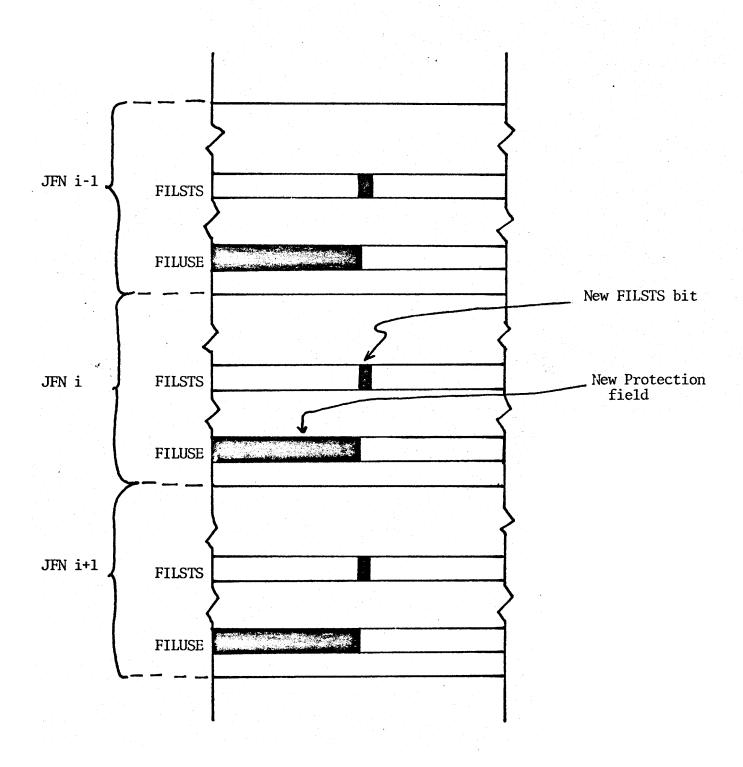
Assuming a limit of 18 forks per job, this requires one additional half-word of storage per JFN. The protection field thus looks like



and is interpreted as follows:

Fork i can use JFN j only if bit i is set in JFN j's protection field.

We could <u>always</u> consult the protection field, in series with the current protection algorithms, but it will make things easier if we allocate a bit (in FILSTS, perhaps) whose value determines whether or not the new protection data is to be looked at.



II. Changes to Code

- A. GTJFN is unchanged except that it must make sure the new FILSTS bit is set to \emptyset .
- B. A new JSYS called GTPJFN will be provided for getting a "private" JFN. It should be simple to implement this as another entry point into the GTJFN code. It works just like GTJFN except:
 - 1. It sets the new FILSTS bit to 1;
 - It sets the JFN's protection field to have exactly one bit set, namely the one corresponding to the calling fork.
- C. Two new JSYSes are required for reading/setting the protection field of a JFN:

RJFNP (Read JFN Protection) returns the protection field of a specified JFN. It fails if the designated JFN is not accessible by the calling fork or if the designated JFN is not a "private" JFN (i.e., if the new FILSTS bit is \emptyset .)

SJFNP (Set JFN Protection) sets the protection field of a specified JFN to a specified value. It fails if the designated JFN is not accessible to the calling fork and if it isn't a 'private' JFN. It is not allowed to set the protection field to \emptyset .

III. Use

The idea is that a "protected" program will use its

"private" directory number to get at files. To make sure these
files aren't also available to its superiors, it will use

GTPJFN. If the program likes, it can share these JFNs selectively
with other forks (e.g. inferiors that it creates), still keeping
them safe from use by "outsiders."