PROTECTED PROGRAMS

A SIMPLE IMPLEMENTATION

Jack Freeman
June 14, 1974

Technical Memo

TM.74-26

PROTECTED PROGRAMS

A SIMPLE IMPLEMENTATION

I. PSAV Files

We imagine a new type of SAVE file called a PSAV file. The format of a PSAV file is the same as that of a TENEX SAV file. PSAV files differ in that:

(a) A bit in the FDB of a file marks it as a PSAV file.

(b) PSAV files are created by a new JSYS called PSAVE, which is the only JSYS which will set the bit mentioned in (a). Except for this setting of the bit, PSAVE works just like SSAVE.

II. PGET JSYS

A new JSYS called PGET will provide the means of running PSAV files. This JSYS will do all of the operations needed to establish a fork, place a program in it, and start the program running. Thus, it will be a sort of combination of CFORK, GET, and SFRKV.

The arguments to PGET are:

1. JFN of the PSAV file;
2. Entry point (specified as an index into the file's entry vector);
3. Address and length of a "parameter block"; and
4. Address of an AC block.

The calling fork must have access to the JFN and must have execute access to the file. The file must be one created by PSAVE.

All that is required of the entry point is that it be a valid index into the file's entry vector.

The "parameter block" is a (possibly null) list of job-local objects (i.e., JFNs and directory numbers) to which the called program is to be given access. An element in the list will be one word with a type-code in its left half and a JFN or DIRTAB index in its right half. Of course, the calling fork must have access to the objects it "passes." All PGET does is set the appropriate protection fields on the objects to allow access to them by the called program.

The AC block is to be used for actually passing arguments to the called program. Typically, it might parallel the "parameter block," with entry i in the parameter block specifying that the called program is to be given access to some JFN, say, and ACi specifying the same JFN to the called program as an argument for it to use.

PGET creates a fork. It sets this fork's protection field (see Fork Protection memo) to allow no access to the fork by its superior. It sets up the fork's memory from the PSAV file in the normal way.

PGET finds a free entry in DIRTAB and puts into its right half the directory number of the directory from which the PSAV file comes. It puts the index of this DIRTAB entry into the right half of the new fork's FRKDIR word, and copies the left half of the calling fork's FRKDIR word into the left half of the new fork's FRKDIR word.

Finally, PGET starts the new fork at the specified place in its entry vector. Note that since the subsystem is associated with a different directory than the other forks in the job that it can bring with it and enable a subset of the capabilities possible in that directory. That is the DDBPRV word from the directory can be used to set up user capabilities. The subsystem could insist on enabling certain job capabilities (like ↑C).