

Each user receives a user identification code (UIC), on which the protection mechanism is based. The UIC is associated with each structure, file, volume, and device that the user owns. For example, when the user creates a file, the system associates the user's UIC with the file. The system also attaches a user-specified protection mask that permits and/or prohibits categories of access to the protected entity. Typically, a protection mask might permit all users to read a data file, while permitting only the owner to modify or delete it. The protection mechanism also permits users to associate in groups, such that access can be permitted to all members of the group and denied to all others.

Scavenge protection is also provided in three forms. File high-water marking prevents users from reading beyond the end of a file mark. Erase on delete insures that information in a file is zeroed before being returned to general use. Erase on extend prevents a user from reading information that may have been previously allocated to another file.

Security alarms are provided on both a file and a system-wide basis. A class of operations may be defined to receive security related messages.

Input/Output

I/O directives can be specified on the following levels:

- * **DCL commands** - Commands such as EDIT, CREATE, APPEND, COPY, PRINT, TYPE, SORT, and DELETE provide the nonprogramming user with the ability to manipulate files.
- * **High-level programs** - Programs written in COBOL, FORTRAN, BASIC, PASCAL, C, PL/I and other high-level languages can perform I/O using standard language elements. Application programmers will find this level the simplest and most efficient in most cases.
- * **VAX Record Management Services (RMS)** - VAX RMS consists of a set of routines that MACRO and high-level language programs can call for device-independent I/O. VAX MACRO and VAX BLISS-32 programmers will find VAX RMS the most efficient level in most cases.
- * **Queue I/O (QIO) system services** - The QIO system services are direct calls to the operating system. Programmers can use this level of I/O to perform special device dependent functions and to eliminate the system overhead involved in RMS; for example, to respond to interrupts from real-time devices as rapidly as possible.