plete their calculations at about the same time. This is not always desirable, since certain hospital services should be completed faster than program-development computations. For example, the compilation of a laboratory report should take precedence over a systems programmer's assembly. Accordingly, the Swapper recognizes a priority system (not to be confused with the interrupt priority system) which allows certain programs to fall to lower queues more slowly than normal. Programs with a priority of zero (or one) are handled exactly as described above. If an application program, by direction of the system administrator, is assigned a priority P ($2\leq P\leq 31$), then its queue counter will be decremented by 1 for every P quanta of running time. Thus, for a priority of 14, 14 time quanta must be given to the program before its queue counter is decremented by 1. It is also possible for a program to be assigned a negative priority; in this case, the program's queue counter is decremented by P for every single time quantum which it receives.

Any priority system can be abused, and the system described here is no exception. The user, however, must use the Executive system to change his program's priority, and the Executive could have been written to require special passwords in order to allow priority change. Instead, it was decided that the highest priority used by a program should be included in the system statistics that are kept on every user run. These statistics are available to the system administrator, and with them he can minimize the misuse of priority.

C. Fastrand Swapping

Ordinarily, active nonrunning programs are stored on the swapping