2. operations which are common to many user programs, especially programs which implement hospital applications —— this category includes routines for storage and retrieval of individual items in tree-structured hospital records, conversion of numbers in date or decimal format to internal code, and floating-point arithmetic;

3. linkage to two system "interpreter" routines, called Job Hunter and Syntax Verifier, which are used by all hospital-applications programs. Job Hunter is a questionnaire administrator that provides standard methods for formatting questions, correcting erroneous answers, and branching around questions made unnecessary by previous answers. Syntax Verifier provides a standard method of specifying the syntax of possible user input and determining which of a number of possible user inputs (including invalid input) was actually given.

A block of user core memory (registers 40_8-72_8) is reserved for temporary storage for the IOT routines. (These registers are not protected by hardware, as are registers 0_8-37_8, but protection is unnecessary since user-written code is not executed during the execution of an IOT.) With all temporary storage used by IOT's allocated to the user memory block, it is clear that the IOT's can be written as reenterable subroutines; thus a user program may be swapped out during the execution of an IOT. In actual practice, IOT's which have high execution speeds are executed out of user mode on interrupt level 16 (i.e., before debreaking), thereby preventing initiation of the Swapper which is on interrupt level 17 and, therefore, of lower priority. Slower IOT's, in particular Job Hunter and Syntax Verifier, are executed in user mode (i.e., after debreaking) so that swapping of programs using these IOT's is very likely to occur.

The execution of an IOT is performed according to the following steps.