The common run-time procedure library provides run-time support for the VAX native-mode, high-level languages BASIC, COBOL, FORTRAN, PL/I, RPG, and PASCAL. All routines in the common run-time library follow standard call and condition handling conventions and most are contained within a shareable image.

At a lower level, programs can call the system directly for event flag, asynchronous system trap, logical name, record and file I/O, process control, timer, conversion, condition handling, lock management, and memory management services. Again, the system uses standard call and condtiton handling conventions.

Programmers using native mode, high-level languages need not worry about the mechanics of calling the common run-time procedures or the base system services. The necessary functions are built into the VAX languages. However, the various levels of detail are available for those with special requirements, for example, programmers constructing real-time applications. In addition, the user can add procedures to the common run-time procedure library and can write new system services.

A system utylity, sort/merge, creates sorted files from unsorted binary or character input and merges several sorted files into one, in ascending or descending sequence based on one or more keys.

The sort/merge routines can be invoked as a utility through DCL commands or can be called as subroutines by user programs. In addition to the usual record sort, the sort/merge routines provide tag, address, and index sorts.

A librarian utility permits the programmer to efficiently store object modules, macros, help text, or any general record-oriented information in central, easily accassible files. Object modules are automatically incorporated in images by the linker as they are referenced by calling programs. Macros are automatically included in the source code of VAX MACRO programs as the programs are assembled. Programs can call librarian routines to access help text or general information stored in a library file.

A symbolic debugger permits programmers to trace program execution and to display and modify program contents using the same symbols that are in the source code. Working interactively with DCL-like commands, the programmer can:

* **Set breakpoints** - Specify locations in the program where execution will temporarily halt

* **Set tracepoints** - Specify locations or operation codes which, if used, will cause a message to be displayed during program execution

* **Set watchpoints** - Specify locations, that, if modified, will cause a message to be displayed during program execution