The roles of each of these components in system software:

1. **Editor**:
    - **Role**: An editor is a tool used to write and modify source code. It provides a user interface to input and manage code efficiently.
    - **Functionality**: Features include syntax highlighting, code formatting, and sometimes basic debugging tools. Examples include Visual Studio Code, Sublime Text, and Vim.
2. **Compiler**:
    - **Role**: A compiler translates source code written in a high-level programming language (like C, C++, or Java) into machine code or an intermediate code (like bytecode for Java).
    - **Functionality**: The compiler performs syntax checking, semantic analysis, optimization, and code generation. The output is usually an object file or intermediate representation that is not yet executable.
3. **Assembler**:
    - **Role**: An assembler translates assembly language, which is a low-level human-readable representation of machine code, into machine code or binary code.
    - **Functionality**: The assembler converts the symbolic instructions and labels in assembly language into the actual binary instructions that a CPU can execute. The output is an object file containing machine code.
4. **Linker**:
    - **Role**: A linker combines multiple object files produced by the compiler (or assembler) into a single executable or a library.
    - **Functionality**: The linker resolves references between different object files and libraries, adjusts addresses, and handles symbol resolution. It produces the final executable file that can be run by the operating system.
5. **Loader**:
    - **Role**: A loader loads the executable file into memory so that it can be executed by the CPU.
    - **Functionality**: The loader allocates memory, initializes the execution environment, and sets up the program's execution. It might also handle dynamic linking of libraries, address relocation, and other setup tasks.

In summary, these components work together to transform source code into an executable program. The editor is used for code creation and modification, the compiler and assembler convert the code into machine-readable formats, the linker combines these formats into a single executable, and the loader prepares the executable for running.