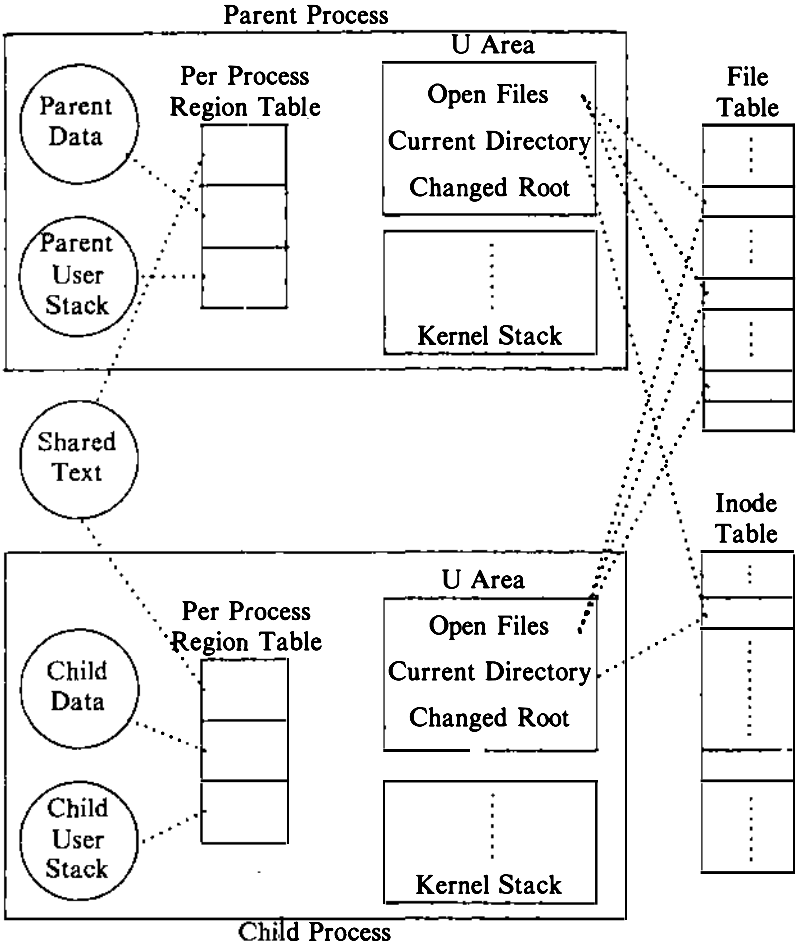

Figure 7.3. Fork Creating a New Process Context

existing file, *creat*s the new file, and — assuming it encounters no errors — *fork*s and creates a child process. Internally, the kernel makes a copy of the parent context for the child process, and the parent process executes in one address space and the child process executes in another. Each process can access private copies of the global variables *fdrd*, *fdwt*, and *c* and private copies of the stack variables *argc* and *argv*, but neither process can access the variables of the other process. However, the kernel copied the *u area* of the original process to the child process during the *fork*, and the child thus inherits access to the parent files (that is, the files the parent originally *open*ed and *creat*ed) using the same file descriptors.