In this paper, we propose a method that enables operating system calls from inside architecture simulators. The proposed framework provides support to file I/O and dynamic memory systems, and can be incorporated into any ISA simulator, at the instruction or cycle-accurate levels. It enables calls to POSIX-compatible system routines in a simulated application, without requiring any change to the application code. Since file I/O is performed transparently, the input and output data for the application program is read/written directly from/to the host filesystem, and all console operations are redirected to the host console. This framework was tested in ISA simulators synthesized from models written in the ArchC Architecture Description Language, but it can be incorporated into any ADL or hand-coded simulator. Instruction and cycle-accurate models, for both MIPS I and SPARC V8 architectures have been thoroughly evaluated by successfully running programs from the Mibench and MediaBench benchmarks.