We give a new denotational semantics for a shared-variable parallel programming language and prove full abstraction: the semantics gives identical meanings to commands if and only if they induce the same behavior in all program contexts. The meaning of a command is a set of "transition traces," which record the ways in which a command may interact with and be affected by its environment. We show how to modify the semantics to incorporate new program constructs, to allow for different levels of granularity or atomicity, and to model fair infinite computation, in each case achieving full abstraction with respect to an appropriate notion of program behavior. 