MODULE *Database*

EXTENDS *Integers*, *Sequences*, *TLC*
CONSTANTS *Data*, *NULL*, *Clients*

Constants: *Clients* ← [model value] {*c*1, *c*2}
NULL ← [model value]
Data ← [model value] {*d*1, *d*2}

```
--algorithm database

variables
  query = [c ∈ Clients ↦ NULL] ;
  ghost_db_history = [c ∈ Clients ↦ NULL] ;


define
  Exists(val) ≜ val ≠ NULL
  RequestingClients ≜ {c ∈ Clients : Exists(query[c]) ∧ query[c].type = "request"}
end define ;

macro wait_for_response()begin
  await query[self].type = "response" ;
end macro ;

macro request(data)begin
  query[self] := [type ↦ "request"] @@ data ;
end macro ;

  The response is the value of the database at the time the request was processed.

process database = "Database"
variable db_value ∈ Data ;
begin
  DB:
    with client ∈ RequestingClients, q = query[client] do
      if q.request = "write" then
        db_value := q.data
       elsif q.request = "read" then
        skip ;
       else
        assert FALSE ;   what did we even pass in
      end if ;
      ghost_db_history[client] := db_value ;
      query[client] := [type ↦ "response", result ↦ db_value] ;
    end with ;
  goto DB ;
end process ;
```

```
process clients ∈ Clients
variables result = NULL;
begin

  Request:
    while TRUE do
      either   read
        request([request ↦ "read"]);
        Confirm:
          wait_for_response();
          result := query[self].result;
          assert result = ghost_db_history[self];
      or   write
        with d ∈ Data do
          request([request ↦ "write", data ↦ d]);
        end with ;
        Wait:
          wait_for_response();
      end either ;
    end while ;

end process ;

end algorithm ;
```

BEGIN TRANSLATION

VARIABLES $query$, $ghost\_db\_history$, pc

define statement

$Exists(val) \triangleq val \neq NULL$

$RequestingClients \triangleq \{c \in Clients : Exists(query[c]) \land query[c].type = \text{"request"}\}$

VARIABLES db_value, $result$

$vars \triangleq \langle query, ghost\_db\_history, pc, db\_value, result\rangle$

$ProcSet \triangleq \{\text{"Database"}\} \cup (Clients)$

$Init \triangleq$ Global variables
$\land\ query = [c \in Clients \mapsto NULL]$
$\land\ ghost\_db\_history = [c \in Clients \mapsto NULL]$
Process database
$\land\ db\_value \in Data$
Process clients
$\land\ result = [self \in Clients \mapsto NULL]$
$\land\ pc = [self \in ProcSet \mapsto \text{CASE } self = \text{"Database"} \to \text{"DB"}$
$\qquad\qquad\qquad\qquad\qquad\qquad \square\ \ self \in Clients \to \text{"Request"}]$

```
DB ≜ ∧ pc["Database"] = "DB"
     ∧ ∃ client ∈ RequestingClients :
         LET q ≜ query[client] IN
           ∧ IF q.request = "write"
                THEN ∧ db_value' = q.data
                ELSE ∧ IF q.request = "read"
                          THEN ∧ TRUE
                          ELSE ∧ Assert(FALSE,
                                        "Failure of assertion at line 38, column 9.")
                     ∧ UNCHANGED db_value
           ∧ ghost_db_history' = [ghost_db_history EXCEPT ![client] = db_value']
           ∧ query' = [query EXCEPT ![client] = [type ↦ "response", result ↦ db_value']]
     ∧ pc' = [pc EXCEPT !["Database"] = "DB"]
     ∧ UNCHANGED result

database ≜ DB

Request(self) ≜ ∧ pc[self] = "Request"
                ∧ ∨ ∧ query' = [query EXCEPT ![self] = [type ↦ "request"] @@ ([request ↦ "read"])]
                     ∧ pc' = [pc EXCEPT ![self] = "Confirm"]
                  ∨ ∧ ∃ d ∈ Data :
                        query' = [query EXCEPT ![self] = [type ↦ "request"] @@ ([request ↦ "write", data
                     ∧ pc' = [pc EXCEPT ![self] = "Wait"]
                ∧ UNCHANGED ⟨ghost_db_history, db_value, result⟩

Confirm(self) ≜ ∧ pc[self] = "Confirm"
                ∧ query[self].type = "response"
                ∧ result' = [result EXCEPT ![self] = query[self].result]
                ∧ Assert(result'[self] = ghost_db_history[self],
                         "Failure of assertion at line 57, column 11.")
                ∧ pc' = [pc EXCEPT ![self] = "Request"]
                ∧ UNCHANGED ⟨query, ghost_db_history, db_value⟩

Wait(self) ≜ ∧ pc[self] = "Wait"
             ∧ query[self].type = "response"
             ∧ pc' = [pc EXCEPT ![self] = "Request"]
             ∧ UNCHANGED ⟨query, ghost_db_history, db_value, result⟩

clients(self) ≜ Request(self) ∨ Confirm(self) ∨ Wait(self)

Next ≜ database
       ∨ (∃ self ∈ Clients : clients(self))

Spec ≜ Init ∧ □[Next]_vars

END TRANSLATION
```

\ * Modification History
\ * Last modified Sun May 05 14:55:42 *PDT* 2019 by *jasondebolt*
\ * Created Sun May 05 11:33:25 *PDT* 2019 by *jasondebolt*