MODULE *LinearSnapshot*

This module is part of the *AfekSimplified* example in Section 6 of "Auxiliary Variables in TLA+". The other modules in that example are *Linearizability*, *NewLinearSnapshot*, *NewLinearSnapshotPS*, *AfekSimplified*, and *AfekSimplifiedH*.

In their 1993 *JACM* paper "Atomic Snapshots of Shared Memory", *Afek*, Attiya, *Dolev*, *Gafni*, *Merritt*, and *Shavit* defined what they called a single-writer atomic snapshot object. We call it simply a snapshot object. A snapshot object is a data object with two kinds of processes, readers and writers, and whose state is a memory consisting of an array of registers, one register for each writer. A writer i can execute a command that sets the value of register value i, returning a "null value"; a reader can execute a command that returns the current value of the object, leaving its state unchanged. They present an algorithm that implements a linearizable snapshot object. Here, we define precisely what a linearizable snapshot object is. We specify it as an instance of the specification of linearizability in module *Linearizability*, which should be read before this module.

We declare the constants *Readers*, *Writers*, and *RegVals* to be the sets of readers, writers, and register values, and we declare the initial value *InitRegVal* of the registers. Readers and writers should be thought of as roles rather than processes, since a physical process can act as both a reader and a writer. We assume that the sets *Readers* and Writers are disjoint.

```
CONSTANTS Readers, Writers, RegVals, InitRegVal

ASSUME  ∧ Readers ∩ Writers = {}
        ∧ InitRegVal ∈ RegVals
```

We define *Procs* to be the value to be substituted for the constant parameter of that name in the *Linearizability* module.

```
Procs ≜ Readers ∪ Writers
```

MemVals is the set of memory values, and *InitMem* is its initial value. They are used to instantiate the *ObjValues* and *InitObj* constants of the *Linearizability* module.

```
MemVals ≜ [Writers → RegVals]
InitMem  ≜ [i ∈ Writers ↦ InitRegVal]
```

NotMemVal is the value returned by a write operation, and *NotRegVal* is the command issued by a reader.

```
NotMemVal ≜ CHOOSE v : v ∉ MemVals
NotRegVal   ≜ CHOOSE v : v ∉ RegVals
```

We now define the values to be instantiated for the remaining constant paramters of module *Linearizability*.

```
Commands(i) ≜ IF i ∈ Readers THEN {NotMemVal}
                              ELSE RegVals

Outputs(i) ≜ IF i ∈ Readers THEN MemVals
                            ELSE {NotRegVal}

InitOutput(i) ≜ IF i ∈ Readers THEN InitMem ELSE NotRegVal

Apply(i, cmd, obj) ≜ IF i ∈ Readers
                       THEN [newState ↦ obj, output ↦ obj]
                       ELSE [newState ↦ [obj EXCEPT ![i] = cmd],
                                  output     ↦ NotRegVal]
```

We prefer to use a variable named *mem* to instantiate the variable object of module *Linearizability*.

```
VARIABLES mem, interface, istate
```

Remember that parameters of an instantiated module whose instantiated values are not given in the WITH statement are instantiated by identifiers with the same name in the current module.

```
INSTANCE Linearizability WITH ObjValues ← MemVals, InitObj ← InitMem,
                              object ← mem
```

TLC does not automatically check assumptions in instantiated modules.

```
ASSUME LinearAssumps
```

* Modification History
* Last modified *Thu Nov* 08 17:10:01 *CST* 2018 by *hengxin*
* Last modified Sat *Oct* 22 01:36:13 *PDT* 2016 by *lamport*
* Created *Tue Oct* 04 02:26:20 *PDT* 2016 by *lamport*