```
MODULE DistFinish
Resilient
                                                                                      PPoPP14
                distributed
                                   finish
                                                        implemented
                                                                                                        See
                                                                             in
Finish State. Finish Resilient Distributed \\
EXTENDS Integers, Sequences
CONSTANTS PLACE, MXFINISHES, PROG_HOME, BACKUP
VARIABLES fid, fstates, msgs, thrds, pstate, waitForMsgs, killed, fbackups, seq
INSTANCE Commons
Terminated \triangleq
     \land fstates[fid].status = "forgotten"
Running \triangleq
     \land fstates[fid].status = "waiting"
IsRoot \triangleq
     \land \mathit{fstates}[\mathit{fid}].\mathit{type} = \mathit{``distroot''}
LastActivity \triangleq
     \land fstates[fid].count = 1
SendMasterTransit(dst) \triangleq
     \land dst \neq fstates[fid].here
     \land LET parentId \stackrel{\triangle}{=} fstates[fid].parent
              here \stackrel{\triangle}{=} fstates[fid].here \\ root \stackrel{\triangle}{=} fstates[fid].root
              rootPlace \triangleq GetFinishHome(fstates[fid].root)
              \land SendMsg([mid \mapsto seq.mseq,
                               src \mapsto here.
                               dst \mapsto rootPlace,
                             target \mapsto dst,
                                 fid \mapsto root,
                     adoptedFID \mapsto NotID,
                            type \mapsto \text{``masterTransit''}]
               \land waitForMsgs' = waitForMsgs \cup \{[src \mapsto rootPlace,
                                                              dst \mapsto here,
                                                             target \mapsto dst,
                                                                fid \mapsto root,
                                                               type \mapsto "masterTransitDone" ]}
               \wedge IncrMSEQ(1)
SendMasterTransitToLive(src, actId, inMsg, here, root) \stackrel{\triangle}{=}
    LET rootPlace \triangleq GetFinishHome(root)
           \land ReplaceMsg(inMsg,
                     [mid \mapsto seq.mseq,
                      src \mapsto here,
                    source \mapsto src,
```

```
target \mapsto here,
                       dst \mapsto rootPlace,
                       fid \mapsto root,
                       aid \mapsto actId,
                      type \mapsto \text{``masterLive''}])
           \land waitForMsgs' = waitForMsgs \cup \{[src \mapsto rootPlace,
                                                        dst \mapsto here,
                                                                \mapsto root,
                                                        fid
                                                               \mapsto actId,
                                                       source \mapsto src,
                                                          type \mapsto "masterLiveDone" \}
SendMasterLiveToCompleted(finishEnd) \stackrel{\Delta}{=}
    Let root \triangleq fstates[fid].root
          rootPlace \triangleq GetFinishHome(fstates[fid].root)
          here \triangleq fstates[fid].here
          \land SendMsg([mid \mapsto seq.mseq,
                           src \mapsto here,
                           dst \mapsto rootPlace,
                         target \mapsto here,
                            fid \mapsto root,
                  finishEnd \mapsto finishEnd,
                        type \mapsto \text{``masterCompleted''}])
           \land waitForMsgs' = waitForMsgs \cup \{[src
                                                                  \mapsto rootPlace,
                                                                  \mapsto here.
                                                      target
                                                                  \mapsto here,
                                                         fid
                                                                  \mapsto root,
                                                     finishEnd \mapsto finishEnd,
                                                           type \mapsto "masterCompletedDone" \}
           \wedge IncrMSEQ(1)
Alloc(type, here, parent, root) \triangleq
    \texttt{LET} \ encRoot \ \stackrel{\triangle}{=} \ \ GetEnclosingRoot(parent, fid)
           encRootPlace \triangleq \text{if } fid = FIRST\_ID \text{ THEN } PROG\_HOME \text{ ELSE } fstates[encRoot].here
          \land fstates[fid].status = "unused"
           \land fstates' = [fstates \ EXCEPT \ ![fid].id = fid,
                                                ![fid].count = 1,
                                                ![fid].status = "waiting",
                                                ![fid].type = type,
                                                ![fid].here = here,
                                                ![fid].parent = parent,
                                                ![fid].root = root,
                                                ![fid].eroot = encRoot,
                                                ![fid].isGlobal = IF type = "distremote"
```

```
THEN TRUE ELSE FALSE
```

```
needed for the local path of Runtime.runAsync
NotifyLocalActivitySpawnAndCreation(here, act) \stackrel{\Delta}{=}
     \land fstates[fid].status = "waiting"
    \land fstates' = [fstates \ EXCEPT \ ![fid].count = @ + 1]
NotifySubActivitySpawn(dst) \triangleq
    \land fstates[fid].status = "waiting"
    \land fstates' = [fstates \ EXCEPT \ ![fid].isGlobal = TRUE]
    \land SendMasterTransit(dst)
AllocRemoteAndNotifyRemoteActivityCreation(src, act, inMsg, type, here, parent, root) \triangleq
    \land SendMasterTransitToLive(src, act.aid, inMsg, here, root)
    \land here \neq NotPlace
    \land type = "distremote"
                                               create and notify
    \land Alloc(type, here, parent, root)
NotifyActivityTermination(finishEnd) \stackrel{\Delta}{=}
    \land fstates[fid].status = "waiting"
    \land fstates[fid].count > 0
    \land IF LastActivity \land \neg fstates[fid].isGlobal
        THEN \wedge fstates' = [fstates EXCEPT ! [fid].count = @ - 1,
                                                  ![fid].status = "forgotten"]
                \land \mathit{msgs'} = \mathit{msgs}
                \wedge seq' = seq
                \land waitForMsqs' = waitForMsqs
        ELSE IF LastActivity \land fstates[fid].isGlobal
                THEN \land SendMasterLiveToCompleted(finishEnd)
                        \land fstates' = [fstates \ EXCEPT \ ![fid].count = @ -1,
                                                           ![fid].status = IF fstates[fid].type = "distremote"
                                                                             THEN
                                                                                      "forgotten"
                                                                                      "pendingRelease"
                ELSE \land fstates' = [fstates \ EXCEPT \ ! [fid].count = @ - 1]
                        \land \mathit{msgs'} = \mathit{msgs}
                        \wedge seq' = seq
                        \land \ waitForMsgs' = waitForMsgs
```

^{\ *} Modification History

^{*} Last modified Wed Dec 13 14:45:49 AEDT 2017 by u5482878

^{*} Last modified Tue Dec 05 17:41:47 AEDT 2017 by shamouda

^{*} Created Tue Nov 07 17:50:59 AEDT 2017 by u5482878