

---

MODULE *DEFRemote*

---

The default root finish imeplementation See *FinishState.RemoteFinish* for the actual implementation

EXTENDS *Sequences, Integers*

VARIABLES *fid, fstates, msgs, thrds, mseq, p0adoptSet*

CONSTANTS *PLACE, MXFINISHES, PROG\_HOME, MXTHREADS, NBLOCKS, MXSTMTS*

INSTANCE *Commons*

---

*Alloc*(*type, here, parent, root*)  $\triangleq$  parent not used here  
 $\wedge fstates[fid].status \in \{\text{"unused"}, \text{"forgotten"}\}$   
 $\wedge fstates' = [fstates \text{ EXCEPT } ![fid].id = fid,$   
 $\phantom{\wedge fstates' = [fstates \text{ EXCEPT } } ![fid].count = 0,$   
 $\phantom{\wedge fstates' = [fstates \text{ EXCEPT } } ![fid].status = \text{"waiting"},$   
 $\phantom{\wedge fstates' = [fstates \text{ EXCEPT } } ![fid].type = type,$   
 $\phantom{\wedge fstates' = [fstates \text{ EXCEPT } } ![fid].here = here,$   
 $\phantom{\wedge fstates' = [fstates \text{ EXCEPT } } ![fid].root = root]$

*LastActivity*  $\triangleq$   
 $\wedge fstates[fid].count = 1$

*NotifySubActivitySpawn*(*dst*)  $\triangleq$   
 $\wedge fstates' = [fstates \text{ EXCEPT } ![fid].remActs[dst] = @ + 1]$

*NotifySubActivitySpawnError*(*dst*)  $\triangleq$  FALSE

*NotifyRemoteActivityCreation*(*src, activity, inMsg*)  $\triangleq$   
 $\wedge fstates' = [fstates \text{ EXCEPT } ![fid].count = @ + 1]$   
 $\wedge RecvMsg(inMsg)$

*NotifyLocalActivitySpawnAndCreation*(*here, activity*)  $\triangleq$   
 $\wedge \text{IF } fstates[fid].here = here$   
 $\phantom{\wedge \text{IF } fstates[fid].here = here} \text{ THEN } fstates' = [fstates \text{ EXCEPT } ![fid].count = @ + 1,$   
 $\phantom{\wedge \text{IF } fstates[fid].here = here} \phantom{\text{ THEN } fstates' = [fstates \text{ EXCEPT } } ![fid].remActs[here] = @ + 1]$   
 $\phantom{\wedge \text{IF } fstates[fid].here = here} \text{ ELSE } fstates' = fstates$

*NotifyActivityTermination*  $\triangleq$   
 $\wedge fstates[fid].count > 0$   
 $\wedge \text{LET } here \triangleq fstates[fid].here$   
 $\text{IN } \text{IF } LastActivity$   
 $\phantom{\text{IN } \text{IF } LastActivity} \text{ THEN } fstates' = [fstates \text{ EXCEPT } ![fid].count = @ - 1,$   
 $\phantom{\text{IN } \text{IF } LastActivity} \phantom{\text{ THEN } fstates' = [fstates \text{ EXCEPT } } ![fid].remActs[here] = @ - 1,$   
 $\phantom{\text{IN } \text{IF } LastActivity} \phantom{\text{ THEN } fstates' = [fstates \text{ EXCEPT } } ![fid].status = \text{"finished"}$   
 $\phantom{\text{IN } \text{IF } LastActivity} \text{ ELSE } fstates' = [fstates \text{ EXCEPT } ![fid].count = @ - 1,$   
 $\phantom{\text{IN } \text{IF } LastActivity} \phantom{\text{ ELSE } fstates' = [fstates \text{ EXCEPT } } ![fid].remActs[here] = @ - 1]$

*PushException*(*e*)  $\triangleq$   
 $\wedge fstates' = [fstates \text{ EXCEPT } ![fid].excs = Append(@, e)]$

```

SendTermMsg  $\triangleq$ 
  LET pid  $\triangleq$  fstates[fid].root
    pidHome  $\triangleq$  GetFinishHome(pid)
    here  $\triangleq$  fstates[fid].here
  IN    $\wedge$  pidHome  $\neq$  here
       $\wedge$  fstates' = [fstates EXCEPT ![fid].status = "forgotten"]
       $\wedge$  SendMsg([mid  $\mapsto$  mseq,
                  src  $\mapsto$  here,
                  dst  $\mapsto$  pidHome,
                  type  $\mapsto$  "asyncTerm",
                  fid  $\mapsto$  pid,
                  remActs  $\mapsto$  fstates[fid].remActs,
                  excs  $\mapsto$  fstates[fid].excs])
       $\wedge$  mseq' = mseq + 1

ProcessChildTermMsg(msg)  $\triangleq$  FALSE  remote does't need this action

```

---

```

\ * Modification History
\ * Last modified Mon Nov 06 19:13:20 AEDT 2017 by u5482878
\ * Last modified Tue Sep 26 23:12:25 AEST 2017 by shamouda
\ * Created Wed Sep 13 12:17:03 AEST 2017 by u5482878

```