```
CONTEXT
  context0
SETS
  pid_t  // set of applications
END
```

CONTEXT

context1

EXTENDS

context0

SETS

```
app_knobs_disc_t
app_knobs_cont_t
app_mons_disc_t
app_mons_cont_t
// set of application continuous knobs
app_mons_disc_t
app_mons_cont_t
// set of application discrete monitors
app_mons_cont_t
// set of application continuous monitors
```

END

CONTEXT

context2

EXTENDS

context1

CONSTANTS

MIN_INFINITY MAX_INFINITY

AXIOMS

 $a \times m1$: $MIN_INFINITY \in \mathbb{Z}$ $a \times m2$: $MAX_INFINITY \in \mathbb{Z}$

END

```
MACHINE
  machine0
SEES
  context0
EVENTS
  INITIALISATION ≜
  STATUS
   ordinary
  BEGIN
    skip
  END
  PRIME_API_APP_RTM =
  STATUS
    ordinary
  ANY
    proc_id
  WHERE
   grd1 : proc_id ∈ pid_t
  THEN
    skip
  END
  PRIME_API_DEV_RTM =
  STATUS
    ordinary
  BEGIN
   skip
  END
```

END

```
MACHINE
  machine1
                 // refining PRIME_API_APP_RTM
REFINES
  machine0
SEES
   context1
VARIABLES
                   // set of registered applications
  app_sockets
                     // set of registered application discrete knobs
  app knobs disc
  app_knobs_cont
                     // set of registered application continuous knobs
                     // set of registered application discrete monitors
  app_mons_disc
                     // set of registered application continuous monitors
  app_mons_cont
INVARIANTS
  inv1
             app sockets ⊆ pid t
  inv2 : app_knobs_disc ∈ app_sockets ↔ app_knobs_disc_t
  inv3 :
             app_knobs_cont ∈ app_sockets ↔ app_knobs_cont_t
         :
   inv4
             app_mons_disc ∈ app_sockets ↔ app_mons_disc_t
  inv5
        : app_mons_cont ∈ app_sockets ↔ app_mons_cont_t
EVENTS
   INITIALISATION
    extended
  STATUS
    ordinary
  BEGIN
    act1
              app\_sockets = \emptyset
         : app_knobs_disc ≔ ø
    act2
    act3
         : app_knobs_cont ≔ ø
    act4 : app_mons_disc = \emptyset
    act5
         : app_mons_cont ≔ ø
  END
  PRIME API APP REG
  STATUS
    ordinary
   REFINES
    PRIME_API_APP_RTM
  ANY
    proc id
               // application id
  WHERE
    grd1
         : proc_id ∈ pid_t ∧ proc_id ∉ app_sockets
    act1
         : app_sockets ≔ app_sockets u {proc_id}
   END
  PRIME_API_APP_KNOB_DISC_manage
  STATUS
    ordinary
  ANY
    proc_id
    knob
             // app knob id
    idd
  WHERE
    grd1 : proc_id ∈ pid_t ∧ proc_id ∈ app_sockets
              knob ∈ app_knobs_disc_t ∧ proc_id → idd ∉ app_knobs_disc
    grd2
   THEN
    act1
              app_knobs_disc ≔ app_knobs_disc u {proc_id → idd}
   END
```

```
PRIME API APP KNOB CONT manage
STATUS
 ordinary
ANY
 proc_id
 knob
 idd
WHERE
 grd1
             proc_id ∈ pid_t ∧ proc_id ∈ app_sockets
 grd2
             knob ∈ app_knobs_cont_t ∧ proc_id → idd ∉ app_knobs_cont
THEN
 act1
            app knobs cont ≔ app knobs cont u {proc id → idd}
END
{\bf PRIME\_API\_APP\_MON\_DISC\_manage}
STATUS
 ordinary
ANY
 proc_id
 mon
 idd
WHERE
 grd1
             proc_id ∈ pid_t ∧ proc_id ∈ app_sockets
 grd2
             mon ∈ app_mons_disc_t ∧ proc_id → idd ∉ app_mons_disc
       : app_mons_disc ≔ app_mons_disc ∪ {proc_id → idd}
 act1
END
PRIME_API_APP_MON_CONT_manage
STATUS
 ordinary
ANY
 proc_id
 mon
 idd
WHERE
 grd1
             proc_id ∈ pid_t ∧ proc_id ∈ app_sockets
 grd2
             mon ∈ app_mons_cont_t ∧ proc_id → idd ∉ app_mons_cont
THEN
 act1
        : app_mons_cont ≔ app_mons_cont ∪ {proc_id → idd}
END
PRIME API APP DEREG
STATUS
 ordinary
ANY
 proc_id
WHERE
             proc id ∈ pid t ∧ proc id ∈ app sockets
 grd1
             ran(app_knobs_disc) ⊆ app_knobs_disc_t ∧ app_knobs_disc[{proc_id}] = ø
 grd2
             ran(app_knobs_cont) ⊆ app_knobs_cont_t ∧ app_knobs_cont[{proc_id}] = ∅
 grd3
             ran(app\_mons\_disc) \subseteq app\_mons\_disc\_t \land app\_mons\_disc[\{proc\_id\}] = \emptyset
 grd4
 grd5
             ran(app\_mons\_cont) \subseteq app\_mons\_cont\_t \land app\_mons\_cont[\{proc\_id\}] = \emptyset
THEN
 act1
             app_sockets = app_sockets \ {proc_id}
             app_knobs_disc ≔ {proc_id} ⊲ app_knobs_disc
 act2
 act3
             app_knobs_cont ≔ {proc_id} ⊲ app_knobs_cont
 act4
             app_mons_disc ≔ {proc_id} ⊲ app_mons_disc
 act5
             app_mons_cont ≔ {proc_id} ⊲ app_mons_cont
END
```

```
PRIME_API_DEV_RTM
 extended
STATUS
 ordinary
REFINES
 PRIME_API_DEV_RTM
BEGIN
 skip
END
PRIME_API_APP_KNOB_DISC_DEREG
STATUS
 ordinary
ANY
 knob
 proc_id
 idd
WHERE
 grd1
            knob ∈ app_knobs_disc_t ∧ proc_id → idd ∈ app_knobs_disc
THEN
            app_knobs_disc = app_knobs_disc \ {proc_id → idd}
 act1
END
PRIME API APP KNOB CONT DEREG
STATUS
 ordinary
ANY
 proc_id
 knob
 idd
WHERE
            knob ∈ app_knobs_cont_t ∧ proc_id → idd ∈ app_knobs_cont
 grd1
THEN
       : app_knobs_cont ≔ app_knobs_cont \ {proc_id → idd}
 act1
PRIME_API_APP_MON_DISC_DEREG
STATUS
 ordinary
ANY
 proc_id
 mon
 idd
WHERE
       : mon ∈ app_mons_disc_t ∧ proc_id → idd ∈ app_mons_disc
 grd1
THEN
 act1
        : app_mons_disc ≔ app_mons_disc \ {proc_id → idd}
END
PRIME API APP MON CONT DEREG
STATUS
 ordinary
 proc_id
 \quad \text{mon} \quad
 idd
WHERE
        : mon ∈ app_mons_cont_t ∧ proc_id → idd ∈ app_mons_cont
 grd1
THEN
```

END

```
MACHINE
   machine2
                      refining PRIME_API_APP_KNOB_DISC_manage and PRIME_API_APP_KNOB_CONT_manage
REFINES
   machine1
SEES
   context1
VARIABLES
                     // set of registered applications
   app sockets
                      // set of registered application discrete knobs
   app_knobs_disc
                       // set of registered application continuous knobs
   app_knobs_cont
                       // set of registered application discrete monitors
   app_mons_disc
                       // set of registered application continuous monitors
   app_mons_cont
   app_knobs_disc_min
   app_knobs_cont_min
   app_knobs_disc_max
   app_knobs_cont_max
   app_knobs_disc_value
   app_knobs_cont_value
INVARIANTS
   inv1 \quad : \quad app\_knobs\_disc\_min \in app\_knobs\_disc \, \rightarrow \, \mathbb{Z}
   inv2 : app_knobs_cont_min \in app_knobs_cont \rightarrow \mathbb{Z}
   inv3 : app_knobs_disc_max \in app_knobs_disc \rightarrow \mathbb{Z}
   inv4 : app_knobs_cont_max \in app_knobs_cont \rightarrow \mathbb{Z}
   inv5 : app_knobs_disc_value \in app_knobs_disc \rightarrow \mathbb{Z}
   inv6 : app_knobs_cont_value \in app_knobs_cont \rightarrow \mathbb{Z}
EVENTS
   INITIALISATION ≜
    extended
   STATUS
    ordinary
   BEGIN
               app_sockets = ø
    act1
    act2
               app_knobs_disc = ø
    act3
           : app_knobs_cont = ø
    act4
           : app_mons_disc = ø
           : app_mons_cont = ø
    act5
    act6
           : app_knobs_disc_min ≔ ø
    act7
           : app_knobs_cont_min ≔ ø
    act8
           : app_knobs_disc_max ≔ ø
    act9 : app_knobs_cont_max = \emptyset
    act10 : app_knobs_disc_value = \emptyset
    act11 : app_knobs_cont_value = ø
   END
   PRIME API APP REG ≜
    extended
   STATUS
    ordinary
   REFINES
    PRIME_API_APP_REG
   ANY
    proc_id
                // application id
   WHERE
          : proc_id ∈ pid_t ∧ proc_id ∉ app_sockets
    grd1
   THEN
    act1 : app_sockets = app_sockets u {proc_id}
   END
   PRIME_API_APP_KNOB_DISC_REG
   STATUS
    ordinary
   REFINES
```

```
PRIME_API_APP_KNOB_DISC_manage
ANY
 proc_id
 knob
 idd
           // app knob id
 minn
 maxx
WHERE
 grd1
             proc_id ∈ pid_t ∧ proc_id ∈ app_sockets
 grd2
             knob ∈ app_knobs_disc_t ∧ proc_id → idd ∉ app_knobs_disc
 grd3
             knob ∈ app_knobs_disc_t ∧ minn < maxx
THEN
             app_knobs_disc ≔ app_knobs_disc ∪ {proc_id → idd}
 act1
             app_knobs_disc_min ≔ app_knobs_disc_min ∪ { (proc_id → idd) → minn }
 act2
             app\_knobs\_disc\_max \; = \; app\_knobs\_disc\_max \; \cup \; \{ \; (proc\_id \; \mapsto \; idd) \; \mapsto \; maxx \; \}
 act3
             app_knobs_disc_value(proc_id → idd) ≔ minn
 act4
END
PRIME_API_APP_KNOB_CONT_REG
STATUS
 ordinary
REFINES
 PRIME_API_APP_KNOB_CONT_manage
ANY
 proc_id
 knob
 idd
 minn
 maxx
WHERE
             proc_id ∈ pid_t ∧ proc_id ∈ app_sockets
 grd1
             knob ∈ app_knobs_cont_t ∧ proc_id → idd ∉ app_knobs_cont
 grd2
        :
            knob ∈ app_knobs_cont_t ∧ minn < maxx
 grd3
THEN
 act1
           app_knobs_cont ≔ app_knobs_cont u {proc_id → idd}
        :
 act2
           app_knobs_cont_min ≔ app_knobs_cont_min ∪ { (proc_id → idd) → minn }
 act3
            app_knobs_cont_max ≔ app_knobs_cont_max ∪ { (proc_id ↦ idd) ↦ maxx }
 act4
            app_knobs_cont_value(proc_id → idd) ≔ minn
PRIME API APP KNOB DISC MIN
STATUS
 ordinary
ANY
 proc_id
 knob
 idd
 minn
 maxx
 val
WHERE
 grd1
             knob ∈ app_knobs_disc_t ∧ proc_id → idd ∈ app_knobs_disc
             knob \in app_knobs_disc_t \land proc_id \mapsto idd \in app_knobs_disc \land maxx = app_knobs_disc_max
 grd2
             (proc_id → idd) ∧ minn < maxx
             knob \in app_knobs_disc_t \land proc_id \mapsto idd \in app_knobs_disc \land val = app_knobs_disc_value
 grd3
             (proc_id → idd) ∧ minn ≤ val
THEN
 act1
        .
           app_knobs_disc_min(proc_id → idd) ≔ minn
END
PRIME_API_APP_KNOB_CONT_MIN
STATUS
 ordinary
ANY
 proc_id
```

```
knob
 idd
 minn
 maxx
 val
WHERE
 grd1
            knob ∈ app_knobs_cont_t ∧ proc_id → idd ∈ app_knobs_cont
            knob ∈ app_knobs_cont_t ∧ proc_id → idd ∈ app_knobs_cont ∧ maxx = app_knobs_cont_max
 grd2
             (proc_id → idd) ∧ minn < maxx
            knob ∈ app_knobs_cont_t ∧ proc_id → idd ∈ app_knobs_cont ∧ val = app_knobs_cont_value
 grd3
            (proc_id → idd) ∧ minn ≤ val
THEN
            app knobs cont min(proc id → idd) ≔ minn
 act1
END
PRIME_API_APP_KNOB_DISC_MAX
STATUS
 ordinary
ANY
 proc id
               //
 knob
 idd
 minn
 maxx
 val
WHERE
 grd1
            knob ∈ app_knobs_disc_t ∧ proc_id → idd ∈ app_knobs_disc
            knob ∈ app_knobs_disc_t ∧ proc_id → idd ∈ app_knobs_disc ∧ minn = app_knobs_disc_min
 grd2
             (proc_id → idd) ∧ maxx > minn
            knob ∈ app_knobs_disc_t ∧ proc_id → idd ∈ app_knobs_disc ∧ val = app_knobs_disc_value
 grd3
            (proc_id → idd) ∧ maxx ≥ val
THEN
 act1
            app_knobs_disc_max(proc_id → idd) ≔ maxx
END
PRIME_API_APP_KNOB_CONT_MAX
STATUS
 ordinary
ANY
 proc_id
 knob
 idd
 minn
 maxx
 val
WHERE
 grd1
            knob ∈ app_knobs_cont_t ∧ proc_id → idd ∈ app_knobs_cont
            knob ∈ app_knobs_cont_t ∧ proc_id → idd ∈ app_knobs_cont ∧ minn = app_knobs_cont_min
 grd2
             (proc_id \mapsto idd) \land maxx > minn
            knob ∈ app_knobs_cont_t ∧ proc_id → idd ∈ app_knobs_cont ∧ val = app_knobs_cont_value
 grd3
            (proc_id → idd) ∧ maxx ≥ val
THEN
            app_knobs_cont_max(proc_id → idd) ≔ maxx
 act1
END
PRIME API APP KNOB DISC GET
                                                get value
STATUS
 ordinary
ANY
 proc_id
 knob
 idd
 minn
 maxx
 val
```

```
WHERE
             knob ∈ app_knobs_disc_t ∧ proc_id → idd ∈ app_knobs_disc
 grd1
             knob ∈ app_knobs_disc_t ∧ proc_id → idd ∈ app_knobs_disc ∧ minn = app_knobs_disc_min
 grd2
             (proc_id \mapsto idd) \land minn \le val
             knob \in app_knobs_disc_t \land proc_id \mapsto idd \in app_knobs_disc \land maxx = app_knobs_disc_max
 grd3
             (proc_id → idd) ∧ maxx ≥ val
THEN
            app_knobs_disc_value(proc_id → idd) = val
 act1
END
PRIME_API_APP_KNOB_CONT_GET =
                                               aet value
STATUS
 ordinary
ANY
 proc_id
 knob
 idd
 minn
 maxx
 val
WHERE
 ard1
             knob ∈ app_knobs_cont_t ∧ proc_id → idd ∈ app_knobs_cont
             knob ∈ app_knobs_cont_t ∧ proc_id → idd ∈ app_knobs_cont ∧ minn = app_knobs_cont_min
 grd2
             (proc_id → idd) ∧ minn ≤ val
             knob \in app knobs cont t \land proc id \mapsto idd \in app knobs cont \land maxx = app knobs cont max
 grd3
             (proc_id → idd) ∧ maxx ≥ val
THEN
           app_knobs_cont_value(proc_id → idd) ≔ val
 act1
END
PRIME API APP KNOB DISC DEREG
STATUS
 ordinary
REFINES
 PRIME_API_APP_KNOB_DISC_DEREG
 proc_id
 knob
 idd
WHERE
            knob ∈ app_knobs_disc_t ∧ proc_id → idd ∈ app_knobs_disc
 grd1
THEN
            app_knobs_disc ≔ app_knobs_disc \ {proc_id → idd}
 act1
            app_knobs_disc_min ≔ {proc_id → idd} ⊲ app_knobs_disc_min
 act2
            app_knobs_disc_max ≔ {proc_id → idd} ⊲ app_knobs_disc_max
 act3
 act4
            app_knobs_disc_value ≔ {proc_id → idd} ∢ app_knobs_disc_value
END
PRIME API APP KNOB CONT DEREG
STATUS
 ordinary
REFINES
 PRIME_API_APP_KNOB_CONT_DEREG
ANY
 proc_id
 knob
 idd
WHERE
 grd1
            knob ∈ app_knobs_cont_t ∧ proc_id → idd ∈ app_knobs_cont
THEN
 act1
            app_knobs_cont ≔ app_knobs_cont \ {proc_id → idd}
            app_knobs_cont_min = {proc_id → idd} ⊲ app_knobs_cont_min
 act2
 act3
            app_knobs_cont_max ≔ {proc_id → idd} ⊲ app_knobs_cont_max
 act4
            app_knobs_cont_value ≔ {proc_id → idd} ⊲ app_knobs_cont_value
END
```

```
PRIME_API_APP_MON_DISC_manage
 extended
STATUS
 ordinary
REFINES
 PRIME_API_APP_MON_DISC_manage
ANY
 proc_id
 mon
 idd
WHERE
 grd1 : proc id ∈ pid t ∧ proc id ∈ app sockets
 grd2 : mon ∈ app mons disc t ∧ proc id → idd ∉ app mons disc
 act1 : app mons disc ≔ app mons disc u {proc id → idd}
END
PRIME API APP MON CONT manage
 extended
STATUS
 ordinary
REFINES
 PRIME_API_APP_MON_CONT_manage
 proc_id
 mon
           //
 idd
WHERE
 grd1 : proc_id ∈ pid_t ∧ proc_id ∈ app_sockets
 grd2 : mon ∈ app_mons_cont_t ∧ proc_id → idd ∉ app_mons_cont
THEN
 act1 : app_mons_cont ≔ app_mons_cont u {proc_id → idd}
END
PRIME API APP DEREG
 extended
STATUS
 ordinary
REFINES
 PRIME_API_APP_DEREG
ANY
 proc_id
WHERE
 grd1 : proc_id \in pid_t \land proc_id \in app_sockets
 grd2 : ran(app\_knobs\_disc) \subseteq app\_knobs\_disc\_t \land app\_knobs\_disc[\{proc\_id\}] = \emptyset
 grd3 : ran(app\_knobs\_cont) \subseteq app\_knobs\_cont\_t \land app\_knobs\_cont[\{proc\_id\}] = \emptyset
 grd4
       : ran(app_mons_disc) ⊆ app_mons_disc_t ∧ app_mons_disc[{proc_id}] = Ø
 grd5
       : \quad ran(app\_mons\_cont) \subseteq app\_mons\_cont\_t \land app\_mons\_cont[\{proc\_id\}] = \varnothing
THEN
 act1 : app_sockets = app_sockets \ {proc_id}
        : app_knobs_disc ≔ {proc_id} ⊲ app_knobs_disc
 act2
        : app_knobs_cont ≔ {proc_id} ⊲ app_knobs_cont
 act3
 act4
           app_mons_disc ≔ {proc_id} ⊲ app_mons_disc
       : app_mons_cont ≔ {proc_id} ∢ app_mons_cont
 act5
END
PRIME_API_DEV_RTM
 extended
STATUS
 ordinary
REFINES
 PRIME_API_DEV_RTM
BEGIN
 skip
```

END

```
PRIME_API_APP_MON_DISC_DEREG
STATUS
 ordinary
REFINES
 PRIME_API_APP_MON_DISC_DEREG
 proc_id
             //
 mon
 idd
WHERE
 grd1 : mon ∈ app_mons_disc_t ∧ proc_id → idd ∈ app_mons_disc
act1 : app_mons_disc ≔ app_mons_disc \ {proc_id → idd}
END
PRIME API APP MON CONT DEREG
STATUS
 ordinary
REFINES
 PRIME_API_APP_MON_CONT_DEREG
ANY
 proc_id
 mon
 idd
WHERE
 \verb|grdl| : mon \in app_mons_cont_t \land proc_id \mapsto idd \in app_mons_cont|
THEN
 act1 : app_mons_cont ≔ app_mons_cont \ {proc_id → idd}
END
```

END

```
MACHINE
   machine3
                      refining PRIME_API_APP_MON_DISC_manage and PRIME_API_APP_MON_CONT_manage
REFINES
   machine2
SEES
   context2
VARIABLES
                     // set of registered applications
   app sockets
                      // set of registered application discrete knobs
   app knobs disc
                       // set of registered application continuous knobs
   app_knobs_cont
                       // set of registered application discrete monitors
   app_mons_disc
                       // set of registered application continuous monitors
   app_mons_cont
   app_knobs_disc_min
   app_knobs_cont_min
   app_knobs_disc_max
   app_knobs_cont_max
   app_knobs_disc_value
   app_knobs_cont_value
   app_mons_disc_min
   app_mons_cont_min
   app_mons_disc_max
   app_mons_cont_max
   app_mons_disc_weight
   app_mons_cont_weight
   app_mons_disc_value
   app_mons_cont_value
INVARIANTS
   inv1 : app mons disc min \in app mons disc \rightarrow \mathbb{Z}
   inv2 : app mons cont min \in app mons cont \rightarrow \mathbb{Z}
   inv3 : app_mons_disc_max \in app_mons_disc \rightarrow \mathbb{Z}
   inv4 : app_mons_cont_max \in app_mons_cont \rightarrow \mathbb{Z}
   inv8 : app_mons_disc_weight \in app_mons_disc \rightarrow \mathbb{Z}
   inv7
         : app_mons_cont_weight ∈ app_mons_cont → Z
   inv5
         : app_mons_disc_value \in app_mons_disc \rightarrow \mathbb{Z}
         : app_mons_cont_value \in app_mons_cont \rightarrow \mathbb{Z}
   inv6
EVENTS
   INITIALISATION ≜
    extended
   STATUS
    ordinary
   BEGIN
    act1 :
              app_sockets = ø
    act2 : app_knobs_disc ≔ ø
    act3 : app_knobs_cont ≔ ø
    act4 : app_mons_disc ≔ ø
    act5 : app_mons_cont ≔ ø
    act6 : app_knobs_disc_min = ø
    act7 :
                app_knobs_cont_min = ø
    act8 :
                app_knobs_disc_max = Ø
    act9 :
                app_knobs_cont_max := ø
    act10 : app_knobs_disc_value = ø
    act11 :
                app_knobs_cont_value = ø
    act12 :
                app_mons_disc_min ≔ ø
    act13 :
                 app_mons_cont_min = \emptyset
    act14 :
                app_mons_disc_max = ø
    act15 :
                 app_mons_cont_max = \emptyset
    act19 :
                 app_mons_disc_weight = \emptyset
    act18 :
                 app_mons_cont_weight = \emptyset
    act16 :
                 app_mons_disc_value = \emptyset
    act17 :
                 app_mons_cont_value = ø
```

END

```
PRIME_API_APP_REG =
 extended
STATUS
 ordinary
REFINES
 PRIME_API_APP_REG
 proc_id
             // application id
 grd1 : proc_id ∈ pid_t ∧ proc_id ∉ app_sockets
act1 : app sockets = app sockets u {proc id}
END
PRIME API APP KNOB DISC REG
 extended
STATUS
 ordinary
REFINES
 PRIME API APP KNOB DISC REG
 proc_id
 knob
 idd
          // app knob id
 minn
 maxx
WHERE
 grd1 : proc_id \in pid_t \land proc_id \in app\_sockets
 grd2 : knob ∈ app_knobs_disc_t ∧ proc_id → idd ∉ app_knobs_disc
 grd3 : knob ∈ app_knobs_disc_t ∧ minn < maxx
THEN
 act1 : app\_knobs\_disc = app\_knobs\_disc \ \textit{u} \ \{proc\_id \ \Rightarrow \ idd\}
       : app_knobs_disc_min ≔ app_knobs_disc_min ∪ { (proc_id → idd) → minn }
 act2
        : app_knobs_disc_max = app_knobs_disc_max v { (proc_id → idd) → maxx }
 act3
       : app_knobs_disc_value(proc_id → idd) ≔ minn
 act4
END
PRIME_API_APP_KNOB_CONT_REG
 extended
STATUS
 ordinary
REFINES
 PRIME_API_APP_KNOB_CONT_REG
ANY
 proc_id
 knob
 idd
 minn
 maxx
WHERE
 grd1
        : proc_id ∈ pid_t ∧ proc_id ∈ app_sockets
 grd2
           knob ∈ app_knobs_cont_t ∧ proc_id → idd ∉ app_knobs_cont
 grd3
       : knob ∈ app_knobs_cont_t ∧ minn < maxx
THEN
 act1
       : app_knobs_cont ≔ app_knobs_cont u {proc_id → idd}
 act2
        : app_knobs_cont_min ≔ app_knobs_cont_min u { (proc_id → idd) → minn }
 act3
        : app_knobs_cont_max ≔ app_knobs_cont_max u { (proc_id → idd) → maxx }
 act4
       : app_knobs_cont_value(proc_id → idd) = minn
PRIME API APP KNOB DISC MIN
 extended
STATUS
 ordinary
REFINES
```

```
PRIME_API_APP_KNOB_DISC_MIN
ANY
 proc_id
 knob
 idd
 minn
 maxx
 val
WHERE
 grd1
              knob ∈ app_knobs_disc_t ∧ proc_id → idd ∈ app_knobs_disc
              knob \; \in \; app\_knobs\_disc\_t \; \land \; proc\_id \; \nrightarrow \; idd \; \in \; app\_knobs\_disc \; \land \; maxx \; = \; app\_knobs\_disc\_max
 grd2
              (proc_id → idd) ∧ minn < maxx
              knob ∈ app_knobs_disc_t ∧ proc_id → idd ∈ app_knobs_disc ∧ val = app_knobs_disc_value
 grd3
              (proc id → idd) ∧ minn ≤ val
THEN
            app_knobs_disc_min(proc_id → idd) ≔ minn
 act1
END
PRIME API APP KNOB CONT MIN
 extended
STATUS
 ordinary
REFINES
 PRIME_API_APP_KNOB_CONT_MIN
 proc id
 knob
 idd
 minn
 maxx
 val
WHERE
 grd1
              knob \in app\_knobs\_cont\_t \land proc\_id \Rightarrow idd \in app\_knobs\_cont
              knob \in app\_knobs\_cont\_t \land proc\_id \mapsto idd \in app\_knobs\_cont \land maxx = app\_knobs\_cont\_max
 grd2
              (proc_id → idd) ∧ minn < maxx
              knob \in app\_knobs\_cont\_t \land proc\_id \mapsto idd \in app\_knobs\_cont \land val = app\_knobs\_cont\_value
 grd3
              (proc_id → idd) ∧ minn ≤ val
THEN
            app_knobs_cont_min(proc_id → idd) ≔ minn
 act1
END
PRIME_API_APP_KNOB_DISC_MAX
 extended
STATUS
 ordinary
REFINES
 PRIME API APP KNOB DISC MAX
 proc id
                 //
 knob
 idd
 minn
 maxx
 val
WHERE
 grd1
              knob \in app\_knobs\_disc\_t \land proc\_id \mapsto idd \in app\_knobs\_disc
              knob ∈ app_knobs_disc_t ∧ proc_id → idd ∈ app_knobs_disc ∧ minn = app_knobs_disc_min
 grd2
              (proc_id → idd) ∧ maxx > minn
              knob ∈ app_knobs_disc_t ∧ proc_id → idd ∈ app_knobs_disc ∧ val = app_knobs_disc_value
 grd3
              (proc id → idd) ∧ maxx ≥ val
THEN
 act1
            app_knobs_disc_max(proc_id → idd) ≔ maxx
END
PRIME API APP KNOB CONT MAX
```

```
extended
STATUS
 ordinary
REFINES
 PRIME_API_APP_KNOB_CONT_MAX
ANY
 proc_id
 knob
 idd
 minn
 maxx
 val
WHERE
              knob \in app\_knobs\_cont\_t \land proc\_id \mapsto idd \in app\_knobs\_cont
 grd1
              knob ∈ app_knobs_cont_t ∧ proc_id → idd ∈ app_knobs_cont ∧ minn = app_knobs_cont_min
 grd2
              (proc\_id \mapsto idd) \land maxx > minn
              knob \in app\_knobs\_cont\_t \land proc\_id \mapsto idd \in app\_knobs\_cont \land val = app\_knobs\_cont\_value
 grd3
              (proc_id → idd) ∧ maxx ≥ val
THEN
 act1
        : app_knobs_cont_max(proc_id → idd) = maxx
END
PRIME API APP KNOB DISC GET ≜
                                               // get value
 extended
STATUS
 ordinary
REFINES
 PRIME API APP KNOB DISC GET
ANY
 proc_id
 knob
 idd
 minn
 maxx
 val
WHERE
              knob \in app\_knobs\_disc\_t \land proc\_id \mapsto idd \in app\_knobs\_disc
 grd1
              knob ∈ app_knobs_disc_t ∧ proc_id → idd ∈ app_knobs_disc ∧ minn = app_knobs_disc_min
 grd2
              (proc\_id \Rightarrow idd) \land minn \leq val
              knob ∈ app_knobs_disc_t ∧ proc_id → idd ∈ app_knobs_disc ∧ maxx = app_knobs_disc_max
 grd3
              (proc_id → idd) ∧ maxx ≥ val
THEN
         : app_knobs_disc_value(proc_id → idd) = val
 act1
END
PRIME API APP KNOB CONT GET ≜
                                                     get value
 extended
STATUS
 ordinary
REFINES
 PRIME API APP KNOB CONT GET
ANY
 proc id
 knob
 idd
 minn
 maxx
 va1
WHERE
              knob \in app\_knobs\_cont\_t \land proc\_id \Rightarrow idd \in app\_knobs\_cont
 grd1
              knob \in app\_knobs\_cont\_t \ \land \ proc\_id \ \nrightarrow \ idd \in app\_knobs\_cont \ \land \ minn = app\_knobs\_cont\_min
 grd2
              (proc\_id \Rightarrow idd) \land minn \leq val
              knob \in app\_knobs\_cont\_t \land proc\_id \mapsto idd \in app\_knobs\_cont \land maxx = app\_knobs\_cont\_max
 grd3
              (proc_id → idd) ∧ maxx ≥ val
THEN
```

```
act1 : app_knobs_cont_value
            (proc_id \mapsto idd) = val
END
PRIME_API_APP_KNOB_DISC_DEREG
 extended
STATUS
 ordinary
REFINES
 PRIME_API_APP_KNOB_DISC_DEREG
 proc_id
 knob
 idd
WHERE
 grd1
            knob ∈ app_knobs_disc_t ∧ proc_id → idd ∈ app_knobs_disc
 act1
       : app_knobs_disc ≔ app_knobs_disc \ {proc_id → idd}
       : app_knobs_disc_min ≔ {proc_id → idd} ∢ app_knobs_disc_min
 act2
       : app_knobs_disc_max ≔ {proc_id → idd} ∢ app_knobs_disc_max
 act3
 act4
       : app_knobs_disc_value ≔ {proc_id → idd} ∢ app_knobs_disc_value
END
PRIME API APP KNOB CONT DEREG
 extended
STATUS
 ordinary
REFINES
 PRIME_API_APP_KNOB_CONT_DEREG
 proc_id
 knob
 idd
WHERE
 grd1
           knob ∈ app_knobs_cont_t ∧ proc_id → idd ∈ app_knobs_cont
THEN
 act1 : app_knobs_cont = app_knobs_cont \ {proc_id → idd}
        : app_knobs_cont_min ≔ {proc_id → idd} ∢ app_knobs_cont_min
 act2
 act3
            app_knobs_cont_max = {proc_id → idd} d app_knobs_cont_max
 act4
            app_knobs_cont_value = {proc_id → idd} ⊲ app_knobs_cont_value
END
PRIME_API_APP_MON_DISC_REG
STATUS
 ordinary
REFINES
 PRIME_API_APP_MON_DISC_manage
ANY
 proc_id
 mon
           // app knob id
 idd
 minn
 maxx
WHERE
 grd1
            proc_id ∈ pid_t ∧ proc_id ∈ app_sockets
 grd2
            mon ∈ app_mons_disc_t ∧ proc_id → idd ∉ app_mons_disc
 grd3
           mon ∈ app_mons_disc_t ∧ proc_id → idd ∈ app_mons_disc ∧ minn < maxx
THEN
           app_mons_disc ≔ app_mons_disc ∪ {proc_id → idd}
 act1
 act2
            app_mons_disc_min ≔ app_mons_disc_min ∪ { (proc_id → idd) → minn }
 act3
            app\_mons\_disc\_max = app\_mons\_disc\_max \cup \{ (proc\_id \mapsto idd) \mapsto maxx \}
END
PRIME_API_APP_MON_CONT_REG
STATUS
```

```
ordinary
REFINES
 PRIME_API_APP_MON_CONT_manage
ANY
 proc_id
 mon
            //
 idd
 minn
 maxx
WHERE
 grd1
              proc_id ∈ pid_t ∧ proc_id ∈ app_sockets
 grd2
             mon ∈ app_mons_cont_t ∧ proc_id → idd ∉ app_mons_cont
             mon ∈ app_mons_cont_t ∧ proc_id → idd ∈ app_mons_cont ∧ minn < maxx
 grd3
THEN
             app_mons_cont ≔ app_mons_cont u {proc_id → idd}
 act1
              app_mons_cont_min ≔ app_mons_cont_min ∪ { (proc_id → idd) → minn }
 act2
             app_mons_cont_max ≔ app_mons_cont_max u { (proc_id → idd) → maxx }
 act3
END
PRIME_API_APP_MON_DISC_MIN
STATUS
 ordinary
ANY
 proc_id
 \quad \text{mon} \quad
 idd
 minn
 maxx
 val
WHERE
              mon ∈ app_mons_disc_t ∧ proc_id → idd ∈ app_mons_disc
 grd1
              mon ∈ app_mons_disc_t ∧ proc_id → idd ∈ app_mons_disc ∧ maxx = app_mons_disc_max
 grd2
              (proc_id → idd) ∧ minn < maxx
              mon ∈ app_mons_disc_t ∧ proc_id → idd ∈ app_mons_disc ∧ val = app_mons_disc_value
 grd3
              (proc_id \mapsto idd) \land minn \le val
THEN
 act1
             app_mons_disc_min(proc_id → idd) ≔ minn
END
PRIME API APP MON CONT MIN
STATUS
 ordinary
ANY
 proc_id
 mon
 idd
 minn
 maxx
 val
WHERE
              \texttt{mon} \; \in \; \mathsf{app\_mons\_cont\_t} \; \; \land \; \; \mathsf{proc\_id} \; \mapsto \; \mathsf{idd} \; \in \; \mathsf{app\_mons\_cont}
 grd1
              mon ∈ app_mons_cont_t ∧ proc_id → idd ∈ app_mons_cont ∧ maxx = app_mons_cont_max
 grd2
              (proc_id → idd) ∧ minn < maxx
              mon ∈ app_mons_cont_t ∧ proc_id → idd ∈ app_mons_cont ∧ val = app_mons_cont_value
 grd3
              (proc_id → idd) ∧ minn ≤ val
THEN
            app_mons_cont_min(proc_id → idd) ≔ minn
 act1
END
PRIME_API_APP_MON_DISC_MAX
STATUS
 ordinary
ANY
 proc_id
 mon
```

```
idd
 minn
 maxx
 val
WHERE
             mon ∈ app_mons_disc_t ∧ proc_id → idd ∈ app_mons_disc
 grd1
             mon ∈ app_mons_disc_t ∧ proc_id → idd ∈ app_mons_disc ∧ minn = app_mons_disc_min
 grd2
             (proc_id → idd) ∧ maxx > minn
             mon ∈ app_mons_disc_t ∧ proc_id → idd ∈ app_mons_disc ∧ val = app_mons_disc_value
 grd3
             (proc_id → idd) ∧ maxx ≥ val
THEN
             app mons disc max(proc id \rightarrow idd) = maxx
 act1
END
PRIME_API_APP_MON_CONT_MAX
STATUS
 ordinary
ANY
 proc id
 mon
 idd
 minn
 maxx
 val
WHERE
             \texttt{mon} \; \in \; \mathsf{app\_mons\_cont\_t} \; \; \land \; \; \mathsf{proc\_id} \; \mapsto \; \mathsf{idd} \; \in \; \mathsf{app\_mons\_cont}
 grd1
             mon ∈ app_mons_cont_t ∧ proc_id → idd ∈ app_mons_cont ∧ minn = app_mons_cont_min
 grd2
             (proc_id → idd) ∧ maxx > minn
             mon ∈ app_mons_cont_t ∧ proc_id → idd ∈ app_mons_cont ∧ val = app_mons_cont_value
 grd3
             (proc_id → idd) ∧ maxx ≥ val
THEN
             app_mons_cont_max(proc_id → idd) ≔ maxx
 act1
END
PRIME_API_APP_MON_DISC_WEIGHT
STATUS
 ordinary
ANY
 proc_id
 mon
 idd
 minn
 maxx
 weight
WHERE
 grd1
             mon ∈ app_mons_disc_t ∧ proc_id → idd ∈ app_mons_disc
             mon ∈ app_mons_disc_t ∧ proc_id → idd ∈ app_mons_disc ∧ minn = app_mons_disc_min
 grd2
             (proc id \mapsto idd) \land maxx = app mons disc max
             (proc id → idd) ∧ minn = MIN INFINITY ∧ maxx = MAX INFINITY ∧ weight ≠ 0
THEN
             app_mons_disc_weight(proc_id → idd) ≔ weight
 act1
END
PRIME API APP MON CONT WEIGHT
STATUS
 ordinary
ANY
 proc id
 mon
 idd
 minn
 maxx
 weiaht
WHERE
 grd1
             mon ∈ app_mons_cont_t ∧ proc_id → idd ∈ app_mons_cont
```

```
grd2
              mon ∈ app_mons_cont_t ∧ proc_id → idd ∈ app_mons_cont ∧ minn = app_mons_cont_min
              (proc_id → idd) ∧ maxx = app_mons_cont_max
              (proc_id \mapsto idd) \land minn = MIN_INFINITY \land maxx = MAX_INFINITY \land weight \neq 0
THEN
              app_mons_cont_weight(proc_id \mapsto idd) = weight
 act1
END
PRIME_API_APP_MON_DISC_SET
STATUS
 ordinary
ANY
 proc id
 mon
 idd
 minn
 maxx
 val
WHERE
              mon \in app mons disc t \land proc id \mapsto idd \in app mons disc
 grd1
              mon \ \in \ app\_mons\_disc\_t \ \land \ proc\_id \ \mapsto \ idd \ \in \ app\_mons\_disc \ \land \ minn \ = \ app\_mons\_disc\_min
 grd2
              (proc_id → idd) ∧ minn ≤ val
              mon ∈ app_mons_disc_t ∧ proc_id → idd ∈ app_mons_disc ∧ maxx = app_mons_disc_max
 grd3
              (proc_id \mapsto idd) \land maxx \ge val
THEN
 act1
              app_mons_disc_value(proc_id → idd) = val
END
PRIME_API_APP_MON_CONT_SET
STATUS
 ordinary
ANY
 proc_id
 idd
 minn
 maxx
 val
WHERE
              mon ∈ app_mons_cont_t ∧ proc_id → idd ∈ app_mons_cont
 grd1
              mon ∈ app_mons_cont_t ∧ proc_id → idd ∈ app_mons_cont ∧ minn = app_mons_cont_min
 grd2
              (proc_id → idd) ∧ minn ≤ val
              mon ∈ app_mons_cont_t ∧ proc_id → idd ∈ app_mons_cont ∧ maxx = app_mons_cont_max
 grd3
              (proc_id \mapsto idd) \land maxx \ge val
THEN
              app_mons_cont_value(proc_id → idd) ≔ val
 act1
END
PRIME_API_APP_MON_DISC_DEREG
STATUS
 ordinary
REFINES
 PRIME API APP MON DISC DEREG
 proc id
 mon
 idd
WHERE
              mon ∈ app_mons_disc_t ∧ proc_id → idd ∈ app_mons_disc
 ard1
THEN
              app\_mons\_disc \; \vdash \; app\_mons\_disc \; \setminus \; \{proc\_id \; \mapsto \; idd\}
 act1
              app\_mons\_disc\_min \; = \; \{proc\_id \; \mapsto \; idd\} \; \triangleleft \; app\_mons\_disc\_min
 act2
              app\_mons\_disc\_max := \{proc\_id \mapsto idd\} \triangleleft app\_mons\_disc\_max
 act3
 act4
              app_mons_disc_weight ≔ {proc_id → idd} ∢ app_mons_disc_weight
 act5
              app_mons_disc_value ≔ {proc_id → idd} ∢ app_mons_disc_value
END
```

```
PRIME_API_APP_MON_CONT_DEREG
STATUS
 ordinary
REFINES
 PRIME_API_APP_MON_CONT_DEREG
 proc_id
 mon
 idd
WHERE
           mon ∈ app_mons_cont_t ∧ proc_id → idd ∈ app_mons_cont
 grd1 :
 act1 : app_mons_cont ≔ app_mons_cont \ {proc_id → idd}
 act2 : app_mons_cont_min ≔ {proc_id → idd} ⊲ app_mons_cont_min
 act3 : app_mons_cont_max = {proc_id → idd} ⊲ app_mons_cont_max
 act4 : app_mons_cont_weight ≔ {proc_id → idd} ⊲ app_mons_cont_weight
 act5 : app_mons_cont_value ≔ {proc_id → idd} ⊲ app_mons_cont_value
END
PRIME API APP DEREG
 extended
STATUS
 ordinary
REFINES
 PRIME_API_APP_DEREG
 proc_id
WHERE
 grd1
       : proc_id ∈ pid_t ∧ proc_id ∈ app_sockets
 grd2
        : ran(app\_knobs\_disc) \subseteq app\_knobs\_disc\_t \land app\_knobs\_disc[\{proc\_id\}] = \emptyset
 grd3
        : \quad ran(app\_knobs\_cont) \subseteq app\_knobs\_cont\_t \land app\_knobs\_cont[\{proc\_id\}] = \emptyset
        : \quad ran(app\_mons\_disc) \subseteq app\_mons\_disc\_t \land app\_mons\_disc[\{proc\_id\}] = \varnothing
 grd4
        : \quad ran(app\_mons\_cont) \subseteq app\_mons\_cont\_t \land app\_mons\_cont[\{proc\_id\}] = \varnothing
 grd5
THEN
 act1
       : app_sockets = app_sockets \ {proc_id}
        : app_knobs_disc ≔ {proc_id} ∢ app_knobs_disc
 act2
           app_knobs_cont ≔ {proc_id} ∢ app_knobs_cont
 act3
 act4
             app_mons_disc ≔ {proc_id} ⊲ app_mons_disc
 act5
             END
PRIME_API_DEV_RTM
 extended
STATUS
 ordinary
REFINES
 PRIME_API_DEV_RTM
BEGIN
 skip
END
```

END