MACHINE M_IPC
REFINES M_IPC_Conds
SEES Ctr_IPC
VARIABLES

      partition_mode

      processes

      processes_of_partition

      process_state

      processes_of_cores

      finished_core

      location_of_service

      create_process_parm

      periodtype_of_process

      process_wait_type

      locklevel_of_partition

      startcondition_of_partition

      basepriority_of_process

      currentpriority_of_process

      retainedpriority_of_process

      period_of_process

      timecapacity_of_process

      deadline_of_process

      deadlinetime_of_process

      releasepoint_of_process

      delaytime_of_process

      current_partition

      current_partition_flag

      current_processes

      current_processes_flag

      clock_tick

      need_reschedule

      need_procresch

      preempter_of_partition

      preemption_lock_mutex

      timeout_trigger

      errorhandler_of_partition

      process_call_errorhandler

      location_of_service2

      setnorm_wait_procs

      setnorm_susp_procs

      set_priority_parm

      suspend_self_timeout

      suspend_self_waitproc

      resume_proc

      stop_self_proc

      stop_proc

      start_aperiod_proc

      start_aperiod_innormal_proc

      start_period_instart_proc

      start_period_innormal_proc

      delay_start_ainstart_proc

      delay_start_ainnormal_proc

delay_start_ainnormal_delaytime

delay_start_instart_proc

delay_start_innormal_proc

delay_start_innormal_delaytime

req_busy_resource_proc

resource_become_avail_proc

finished_core2

resource_become_avail2

time_wait_proc

period_wait_proc

queuing_ports

sampling_ports

msgspace_of_samplingports

queue_of_queuingports

processes_waitingfor_queuingports

used_messages

send_queuing_message_port

wakeup_waitproc_on_srcqueports_port

location_of_service3

wakeup_waitproc_on_dstqueports_port

receive_queuing_message_port

buffers

MaxMsgNum_of_Buffers

queue_of_buffers

processes_waitingfor_buffers

buffers_of_partition

send_buffer_needwakeup

send_buffer_withfull

receive_buffer_needwake

receive_buffer_whenempty

blackboards

blackboards_of_partition

msgspace_of_blackboards

emptyindicator_of_blackboards

processes_waitingfor_blackboards

display_blackboard_needwake

read_blackboard_whenempty

semaphores

semaphores_of_partition

MaxValue_of_Semaphores

value_of_semaphores

processes_waitingfor_semaphores

wait_semaphore_whenzero

signal_semaphore_needwake

events

events_of_partition

state_of_events

processes_waitingfor_events

set_event_needwake

wait_event_whendown

mutexs

mutex_state

mutex_of_process

priority_of_mutex

mutex_of_count

processes_waitingfor_mutexs

create_of_mutex

acquire_mutex

release_mutex

reset_mutex

finished_core3

RefreshPeriod_of_SamplingPorts

needtrans_of_sourcesamplingport

quediscipline_of_queuingports

quediscipline_of_semaphores

quediscipline_of_mutexs

quediscipline_of_buffers

INVARIANTS

inv_refreshprd_of_sampports: $RefreshPeriod\_of\_SamplingPorts \in sampling\_ports \to \mathbb{N}_1$

inv_flag_sourcesampport: $needtrans\_of\_sourcesamplingport \in sampling\_ports \to BOOL$

inv_flag_means_msg: $\forall p \cdot (p \in sampling\_ports \land needtrans\_of\_sourcesamplingport(p) = TRUE \Rightarrow p \in dom(msgspace\_of\_samplingports))$

inv_quedisp_queuingports: $quediscipline\_of\_queuingports \in queuing\_ports \to QUEUING\_DISCIPLINE$

inv_quedisp_semaphores: $quediscipline\_of\_semaphores \in semaphores \to QUEUING\_DISCIPLINE$

inv_quedisp_mutexs: $quediscipline\_of\_mutexs \in mutexs \to QUEUING\_DISCIPLINE$

inv_quediscipline_of_buffers: $quediscipline\_of\_buffers \in buffers \to QUEUING\_DISCIPLINE$

EVENTS

Initialisation ⟨extended⟩

begin

act001: $partition\_mode := PARTITIONS \times \{PM\_COLD\_START\}$

act101: $processes := \varnothing$

act102: $processes\_of\_partition := \varnothing$

act103: $process\_state := \varnothing$

act104: $processes\_of\_cores := \varnothing$

act105: $finished\_core := CORES \times \{TRUE\}$

act106: $location\_of\_service := \varnothing$

act201: $periodtype\_of\_process := \varnothing$

act301: $process\_wait\_type := \varnothing$

act302: $locklevel\_of\_partition := PARTITIONS \times \{1\}$

act303: $startcondition\_of\_partition := \varnothing$

act304: $basepriority\_of\_process := \varnothing$

act305: $currentpriority\_of\_process := \varnothing$

act306: $retainedpriority\_of\_process := \varnothing$

act307: $period\_of\_process := \varnothing$

act308: $timecapacity\_of\_process := \varnothing$

act309: $deadline\_of\_process := \varnothing$

act310: $deadlinetime\_of\_process := \varnothing$

act311: $releasepoint\_of\_process := \varnothing$

act312: $delaytime\_of\_process := \varnothing$

act313: $current\_partition :\in PARTITIONS$

act314: $current\_partition\_flag := PARTITIONS \times \{FALSE\}$

act315: $current\_processes := CORES \times \varnothing$

act316: $current\_processes\_flag := CORES \times \{FALSE\}$

act317: $clock\_tick := 1$

act318: $need\_reschedule := FALSE$

act319: $need\_procresch := CORES \times \{FALSE\}$

act320: $preempter\_of\_partition := \varnothing$

act321: $preemption\_lock\_mutex := \varnothing$
act322: $timeout\_trigger := \varnothing$
act323: $errorhandler\_of\_partition := \varnothing$
act324: $process\_call\_errorhandler := \varnothing$
act325: $location\_of\_service2 := \varnothing$
act326: $setnorm\_wait\_procs := \varnothing$
act327: $setnorm\_susp\_procs := \varnothing$
act328: $set\_priority\_parm := \varnothing$
act329: $suspend\_self\_timeout := \varnothing$
act330: $suspend\_self\_waitproc := \varnothing$
act331: $resume\_proc := \varnothing$
act332: $stop\_self\_proc := \varnothing$
act333: $stop\_proc := \varnothing$
act334: $start\_aperiod\_proc := \varnothing$
act335: $start\_aperiod\_innormal\_proc := \varnothing$
act336: $start\_period\_instart\_proc := \varnothing$
act337: $start\_period\_innormal\_proc := \varnothing$
act338: $delay\_start\_ainstart\_proc := \varnothing$
act339: $delay\_start\_ainnormal\_proc := \varnothing$
act340: $delay\_start\_ainnormal\_delaytime := \varnothing$
act341: $delay\_start\_instart\_proc := \varnothing$
act342: $delay\_start\_innormal\_proc := \varnothing$
act343: $delay\_start\_innormal\_delaytime := \varnothing$
act344: $req\_busy\_resource\_proc := \varnothing$
act345: $resource\_become\_avail\_proc := \varnothing$
act346: $finished\_core2 := CORES \times \{TRUE\}$
act347: $resource\_become\_avail2 := \varnothing$
act348: $time\_wait\_proc := \varnothing$
act349: $period\_wait\_proc := \varnothing$
act401: $queuing\_ports := \varnothing$
act402: $sampling\_ports := \varnothing$
act403: $msgspace\_of\_samplingports := \varnothing$
act404: $queue\_of\_queuingports := \varnothing$
act405: $processes\_waitingfor\_queuingports := \varnothing$
act406: $used\_messages := \varnothing$
act407: $send\_queuing\_message\_port := \varnothing$
act408: $wakeup\_waitproc\_on\_srcqueports\_port := \varnothing$
act409: $location\_of\_service3 := \varnothing$
act410: $wakeup\_waitproc\_on\_dstqueports\_port := \varnothing$
act411: $receive\_queuing\_message\_port := \varnothing$
act412: $buffers := \varnothing$
act413: $MaxMsgNum\_of\_Buffers := \varnothing$
act414: $queue\_of\_buffers := \varnothing$
act415: $processes\_waitingfor\_buffers := \varnothing$
act416: $buffers\_of\_partition := \varnothing$
act417: $send\_buffer\_needwakeup := \varnothing$
act418: $send\_buffer\_withfull := \varnothing$
act419: $receive\_buffer\_needwake := \varnothing$
act420: $receive\_buffer\_whenempty := \varnothing$
act421: $blackboards := \varnothing$
act422: $blackboards\_of\_partition := \varnothing$
act423: $msgspace\_of\_blackboards := \varnothing$
act424: $emptyindicator\_of\_blackboards := \varnothing$
act425: $processes\_waitingfor\_blackboards := \varnothing$
act426: $display\_blackboard\_needwake := \varnothing$
act427: $read\_blackboard\_whenempty := \varnothing$
act428: $semaphores := \varnothing$
act429: $semaphores\_of\_partition := \varnothing$
act430: $MaxValue\_of\_Semaphores := \varnothing$

        act431: $value\_of\_semaphores := \varnothing$
        act432: $processes\_waiting for\_semaphores := \varnothing$
        act433: $wait\_semaphore\_whenzero := \varnothing$
        act434: $signal\_semaphore\_needwake := \varnothing$
        act435: $events := \varnothing$
        act436: $events\_of\_partition := \varnothing$
        act437: $state\_of\_events := \varnothing$
        act438: $processes\_waiting for\_events := \varnothing$
        act439: $set\_event\_needwake := \varnothing$
        act440: $wait\_event\_whendown := \varnothing$
        act441: $mutexs := \varnothing$
        act442: $mutex\_state := \varnothing$
        act443: $mutex\_of\_process := \varnothing$
        act444: $priority\_of\_mutex := \varnothing$
        act445: $mutex\_of\_count := \varnothing$
        act446: $processes\_waiting for\_mutexs := \varnothing$
        act447: $create\_of\_mutex := \varnothing$
        act448: $acquire\_mutex := \varnothing$
        act449: $release\_mutex := \varnothing$
        act450: $reset\_mutex := \varnothing$
        act451: $finished\_core3 := CORES \times \{TRUE\}$
        act500: $RefreshPeriod\_of\_SamplingPorts := \varnothing$
        act501: $needtrans\_of\_sourcesamplingport := \varnothing$
        act502: $quediscipline\_of\_queuingports := \varnothing$
        act503: $quediscipline\_of\_semaphores := \varnothing$
        act504: $quediscipline\_of\_mutexs := \varnothing$
        act505: $quediscipline\_of\_buffers := \varnothing$
    end

Event create_sampling_port $\langle ordinary \rangle \;\widehat{=}$
extends create_sampling_port
    any
        core
        port
        refresh
        part
    where
        grd001: $core \in CORES$
        grd002: $port \in SamplingPorts \wedge port \notin sampling\_ports$
        grd003: $finished\_core(core) = TRUE$
        grd201: $part = current\_partition$
        grd202: $Ports\_of\_Partition(port) = part$
        grd203: $partition\_mode(part) = PM\_COLD\_START \vee partition\_mode(part) = PM\_WARM\_START$

        grd204: $part \in dom(current\_partition\_flag)$
        grd205: $current\_partition\_flag(part) = TRUE$
        grd206: $partition\_mode(part) \neq PM\_NORMAL$
        grd207: $refresh \in \mathbb{N}_1$
    then
        act001: $sampling\_ports := sampling\_ports \cup \{port\}$
        act201: $RefreshPeriod\_of\_SamplingPorts(port) := refresh$
        act202: $needtrans\_of\_sourcesamplingport(port) := FALSE$
    end

Event write_sampling_message $\langle ordinary \rangle \;\widehat{=}$
extends write_sampling_message
    any
        core
        port
        msg
        t

            part
    where
            grd001: $core \in CORES$
            grd002: $port \in sampling\_ports$
            grd003: $Direction\_of\_Ports(port) = PORT\_SOURCE$
            grd004: $msg \in MESSAGES \wedge msg \notin used\_messages$
            grd005: $t \in \mathbb{N}$
            grd006: $finished\_core(core) = TRUE$
            grd201: $part = current\_partition$
            grd202: $Ports\_of\_Partition(port) = part$
            grd203: $t = clock\_tick * ONE\_TICK\_TIME$
    then
            act001: $msgspace\_of\_samplingports(port) := msg \mapsto t$
            act002: $used\_messages := used\_messages \cup \{msg\}$
            act201: $needtrans\_of\_sourcesamplingport(port) := TRUE$
    end
Event transfer_sampling_msg $\langle ordinary \rangle \;\widehat{=}$
extends transfer_sampling_msg
    any
            core
            port
            msg
            t
    where
            grd001: $core \in CORES$
            grd002: $port \in sampling\_ports$
            grd003: $msg \in MESSAGES$
            grd004: $port \in dom(msgspace\_of\_samplingports)$
            grd005: $t \in \mathbb{N}$
            grd006: $msg \mapsto t = msgspace\_of\_samplingports(port)$
            grd007: $Sampling\_Channels^{-1}[\{port\}] \subseteq sampling\_ports$
            grd008: $finished\_core(core) = TRUE$
            grd201: $t = clock\_tick * ONE\_TICK\_TIME$
    then
            act001: $msgspace\_of\_samplingports := msgspace\_of\_samplingports \vartriangleleft (Sampling\_Channels^{-1}[\{port\}] \times$
                $\{msg \mapsto t\})$
            act201: $needtrans\_of\_sourcesamplingport(port) := FALSE$
    end
Event read_sampling_message $\langle ordinary \rangle \;\widehat{=}$
extends read_sampling_message
    any
            core
            port
            part
            t
    where
            grd001: $core \in CORES$
            grd002: $port \in sampling\_ports$
            grd003: $Direction\_of\_Ports(port) = PORT\_DESTINATION$
            grd004: $port \in dom(msgspace\_of\_samplingports)$
            grd005: $finished\_core(core) = TRUE$
            grd201: $part = current\_partition$
            grd202: $Ports\_of\_Partition(port) = part$
            grd203: $t = clock\_tick * ONE\_TICK\_TIME$
    then
            skip
    end
Event get_sampling_port_id $\langle ordinary \rangle \;\widehat{=}$
    any

> port
> core
> part

where

> grd001: $port \in sampling\_ports$
> grd002: $core \in CORES$
> grd003: $part \in dom(current\_partition\_flag) \land current\_partition = part \land current\_partition\_flag(part) = TRUE$
> grd005: $Ports\_of\_Partition(port) = part$
> grd006: $finished\_core2(core) = TRUE$

then

> *skip*

end

Event get_sampling_port_status ⟨ordinary⟩ $\widehat{=}$

any

> part
> core
> port

where

> grd001: $port \in sampling\_ports$
> grd002: $core \in CORES$
> grd003: $part \in dom(current\_partition\_flag) \land current\_partition = part \land current\_partition\_flag(part) = TRUE$
> grd005: $Ports\_of\_Partition(port) = part$
> grd006: $finished\_core2(core) = TRUE$

then

> *skip*

end

Event create_queuing_port ⟨ordinary⟩ $\widehat{=}$

extends create_queuing_port

any

> *port*
> *core*
> part
> disc

where

> grd001: $port \in QueuingPorts \land port \notin queuing\_ports$
> grd005: $port \in dom(queue\_of\_queuingports)$
> grd002: $core \in CORES$
> grd004: $finite(queue\_of\_queuingports(port))$
> grd003: $finished\_core(core) = TRUE$
> grd201: $part = current\_partition$
> grd206: $part \in dom(current\_partition\_flag)$
> grd202: $current\_partition\_flag(part) = TRUE$
> grd203: $partition\_mode(part) = PM\_COLD\_START \lor partition\_mode(part) = PM\_WARM\_START$
>
> grd204: $Ports\_of\_Partition(port) = part$
> grd205: $disc \in QUEUING\_DISCIPLINE$

then

> act001: $queuing\_ports := queuing\_ports \cup \{port\}$
> act002: $queue\_of\_queuingports(port) := \varnothing$
> act003: $processes\_waiting for\_queuingports(port) := \varnothing$
> act201: $quediscipline\_of\_queuingports(port) := disc$

end

Event send_queuing_message ⟨ordinary⟩ $\widehat{=}$

extends send_queuing_message

any

> *core*
> *port*

msg
t
part

where

grd001: $core \in CORES$
grd002: $port \in queuing\_ports$
grd003: $Direction\_of\_Ports(port) = PORT\_SOURCE$
grd004: $msg \in MESSAGES \land msg \notin used\_messages$
grd005: $finite(queue\_of\_queuingports(port)) \land card(queue\_of\_queuingports(port)) < MaxMsgNum\_of\_QueuingP$

grd006: $processes\_waiting for\_queuingports(port) = \varnothing$
grd007: $t \in \mathbb{N}$
grd008: $finished\_core(core) = TRUE$
grd201: $part = current\_partition$
grd202: $Ports\_of\_Partition(port) = part$
grd203: $t = clock\_tick * ONE\_TICK\_TIME$

then

act001: $queue\_of\_queuingports(port) := queue\_of\_queuingports(port) \mathbin{\lhd\mkern-9mu-} \{msg \mapsto t\}$
act002: $used\_messages := used\_messages \cup \{msg\}$

end

Event transfer_queuing_msg $\langle ordinary \rangle \;\widehat{=}$
extends transfer_queuing_msg

any

core
p
m
t
q
que1
que2

where

grd001: $core \in CORES$
grd002: $p \in queuing\_ports \land q \in queuing\_ports \land p \in Source\_QueuingPorts$
grd003: $q = Queuing\_Channels(p)$
grd004: $m \in MESSAGES$
grd005: $m \mapsto t \in queue\_of\_queuingports(p)$
grd006:
$finite(queue\_of\_queuingports(p)) \land card(queue\_of\_queuingports(p)) \leq MaxMsgNum\_of\_QueuingPorts(p) \land$
$card(queue\_of\_queuingports(p)) > 0$
$\land processes\_waiting for\_queuingports(p) = \varnothing$
grd007: $finite(queue\_of\_queuingports(p)) \land finite(queue\_of\_queuingports(Queuing\_Channels(p))) \land$
$card(queue\_of\_queuingports(q)) < MaxMsgNum\_of\_QueuingPorts(q)$
grd008: $que1 \in queuing\_ports \to (MESSAGES \nrightarrow \mathbb{N})$
grd009: $que1 = queue\_of\_queuingports \mathbin{\lhd\mkern-9mu-} \{p \mapsto (queue\_of\_queuingports(p) \setminus \{m \mapsto t\})\}$
grd010: $que2 \in queuing\_ports \to (MESSAGES \nrightarrow \mathbb{N})$
grd011: $que2 = que1 \mathbin{\lhd\mkern-9mu-} \{q \mapsto (que1(q) \mathbin{\lhd\mkern-9mu-} \{m \mapsto t\})\}$
grd012: $finished\_core(core) = TRUE$
grd201: $\forall m1, t1 \cdot (m1 \mapsto t1 \in queue\_of\_queuingports(p) \Rightarrow t \leq t1)$

then

act001: $queue\_of\_queuingports := que2$

end

Event send_queuing_message_needwait_init $\langle ordinary \rangle \;\widehat{=}$
extends send_queuing_message_needwait_init

any

part
proc
newstate
core
port

where

    grd001: $part \in PARTITIONS$

    grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state) \cap dom(process\_wait\_type)$

    grd003: $newstate \in PROCESS\_STATES$

    grd004: $core \in CORES \wedge core \in dom(current\_processes\_flag)$

    grd005: $processes\_of\_partition(proc) = part$

    grd017: $finished\_core2(core) = TRUE$

    grd101: $partition\_mode(part) = PM\_NORMAL$

    grd102: $process\_state(proc) = PS\_Running$

    grd103: $newstate = PS\_Waiting$

    grd205: $proc \in dom(delaytime\_of\_process) \wedge proc \in dom(process\_wait\_type)$

    grd201: $part = current\_partition \wedge current\_partition \in dom(current\_partition\_flag)$

    grd202: $current\_partition\_flag(part) = TRUE$

    grd203: $current\_processes\_flag(core) = TRUE$

    grd204: $proc = current\_processes(core)$

    grd301: $port \in queuing\_ports$

    grd302: $Ports\_of\_Partition(port) = part$

    grd303: $Direction\_of\_Ports(port) = PORT\_SOURCE$

then

    act001: $process\_state(proc) := newstate$

    act002: $location\_of\_service2(core) := Req\_busy\_resource \mapsto loc\_i$

    act003: $finished\_core2(core) := FALSE$

    act004: $req\_busy\_resource\_proc(core) := proc$

    act005: $current\_processes\_flag(core) := FALSE$

    act006: $current\_processes := \{core\} \lhd current\_processes$

    act301: $location\_of\_service3(core) := Send\_Queuing\_Message\_Wait \mapsto loc\_i$

    act302: $send\_queuing\_message\_port(core) := port$

end

Event send_queuing_message_needwait_timeout $\langle ordinary \rangle \mathrel{\widehat{=}}$

extends send_queuing_message_needwait_timeout

    any

        part

        proc

        core

        timeout

        tmout_trig

        wt

        port

    where

        grd001: $part \in PARTITIONS$

        grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition)$

        grd003: $core \in CORES \cap dom(req\_busy\_resource\_proc) \wedge core \in dom(current\_processes\_flag) \wedge$
             $core \in dom(location\_of\_service2)$

        grd004: $proc = req\_busy\_resource\_proc(core)$

        grd005: $processes\_of\_partition(proc) = part$

        grd006: $part = current\_partition$

        grd018: $processes\_of\_partition(req\_busy\_resource\_proc(core)) \in dom(current\_partition\_flag)$

        grd007: $current\_partition\_flag(part) = TRUE$

        grd008: $current\_processes\_flag(core) = TRUE$

        grd009: $timeout \geq 0$

        grd010: $wt \in PROCESS\_WAIT\_TYPES \wedge (wt = PROC\_WAIT\_OBJ \vee wt = PROC\_WAIT\_TIMEOUT)$

        grd011: $tmout\_trig \in processes \nrightarrow (PROCESS\_STATES \times \mathbb{N}_1)$

        grd012:
             $(timeout = INFINITE\_TIME\_VALUE \Rightarrow tmout\_trig = \varnothing)$
             $\wedge (timeout > 0 \Rightarrow tmout\_trig = \{proc \mapsto (PS\_Ready \mapsto (timeout + clock\_tick * ONE\_TICK\_TIME))\})$

        grd013: $timeout > 0 \Rightarrow wt = PROC\_WAIT\_TIMEOUT$

        grd014: $timeout = INFINITE\_TIME\_VALUE \Rightarrow wt = PROC\_WAIT\_OBJ$

        grd015: $finished\_core2(core) = FALSE$

        grd016: $location\_of\_service2(core) = Req\_busy\_resource \mapsto loc\_i$

        grd017: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Req\_busy\_resource \mapsto loc\_i)$

        grd301: $core \in dom(send\_queuing\_message\_port)$

        grd302: $port \in queuing\_ports$

        grd303: $port = send\_queuing\_message\_port(core)$

        grd304: $Ports\_of\_Partition(port) = part$

        grd305: $location\_of\_service3(core) = Send\_Queuing\_Message\_Wait \mapsto loc\_i$

        grd306: $\neg(finished\_core(core) = FALSE \wedge location\_of\_service3(core) = Send\_Queuing\_Message\_Wait \mapsto loc\_i)$

then

        act001: $location\_of\_service2(core) := Req\_busy\_resource \mapsto loc\_1$

        act002: $timeout\_trigger := timeout\_trigger \lhd tmout\_trig$

        act003: $process\_wait\_type(proc) := wt$

        act301: $location\_of\_service3(core) := Send\_Queuing\_Message\_Wait \mapsto loc\_1$

end

Event send_queuing_message_needwait_insert ⟨ordinary⟩ $\widehat{=}$

extends send_queuing_message_needwait_insert

    any

        part

        proc

        core

        port

        msg

        t

    where

        grd001: $part \in PARTITIONS$

        grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition)$

        grd003: $core \in CORES \cap dom(send\_queuing\_message\_port) \cap dom(req\_busy\_resource\_proc) \cap dom(location\_of\_service3)$

        grd004: $proc = req\_busy\_resource\_proc(core)$

        grd005: $processes\_of\_partition(proc) = part$

        grd006: $part = current\_partition$

        grd019: $part \in dom(current\_partition\_flag)$

        grd007: $current\_partition\_flag(part) = TRUE$

        grd008: $current\_processes\_flag(core) = TRUE$

        grd009: $port \in queuing\_ports$

        grd010: $port = send\_queuing\_message\_port(core)$

        grd011: $Ports\_of\_Partition(port) = part$

        grd012: $Direction\_of\_Ports(port) = PORT\_SOURCE$

        grd013: $msg \in MESSAGES \wedge msg \notin used\_messages$

        grd014: $(finite(queue\_of\_queuingports(port)) \wedge card(queue\_of\_queuingports(port)) = MaxMsgNum\_of\_QueuingP$
               $processes\_waitingfor\_queuingports(port) \neq \varnothing$

        grd015: $t \in \mathbb{N}$

        grd016: $location\_of\_service3(core) = Send\_Queuing\_Message\_Wait \mapsto loc\_1$

        grd017: $finished\_core(core) = FALSE$

        grd018: $\neg(finished\_core(core) = FALSE \wedge location\_of\_service3(core) = Send\_Queuing\_Message\_Wait \mapsto loc\_1)$

        grd201: $t = clock\_tick * ONE\_TICK\_TIME$

    then

        act001: $location\_of\_service3(core) := Send\_Queuing\_Message\_Wait \mapsto loc\_2$

        act002: $processes\_waitingfor\_queuingports(port) := processes\_waitingfor\_queuingports(port) \lhd \{proc \mapsto (msg \mapsto t)\}$

        act003: $used\_messages := used\_messages \cap \{msg\}$

    end

Event send_queuing_message_needwait_schedule ⟨ordinary⟩ $\widehat{=}$

extends send_queuing_message_needwait_schedule

any
    part
    proc
    core
    port
where
    grd001: $part \in PARTITIONS$
    grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition)$
    grd003: $core \in CORES \cap dom(req\_busy\_resource\_proc) \wedge core \in dom(current\_processes\_flag) \wedge$
        $core \in dom(location\_of\_service2)$
    grd004: $proc = req\_busy\_resource\_proc(core)$
    grd005: $processes\_of\_partition(proc) = part$
    grd006: $part = current\_partition$
    grd012: $processes\_of\_partition(req\_busy\_resource\_proc(core)) \in dom(current\_partition\_flag)$
    grd007: $current\_partition\_flag(part) = TRUE$
    grd008: $current\_processes\_flag(core) = FALSE$
    grd009: $finished\_core2(core) = FALSE$
    grd010: $location\_of\_service2(core) = Req\_busy\_resource \mapsto loc\_1$
    grd011: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Req\_busy\_resource \mapsto$
        $loc\_1)$
    grd301: $core \in dom(send\_queuing\_message\_port)$
    grd302: $port \in queuing\_ports$
    grd303: $port = send\_queuing\_message\_port(core)$
    grd304: $Ports\_of\_Partition(port) = part$
    grd305: $finished\_core(core) = FALSE$
    grd306: $location\_of\_service3(core) = Send\_Queuing\_Message\_Wait \mapsto loc\_2$
    grd307: $\neg(finished\_core(core) = FALSE \wedge location\_of\_service3(core) = Send\_Queuing\_Message\_Wait \mapsto$
        $loc\_2)$
then
    act001: $location\_of\_service2(core) := Req\_busy\_resource \mapsto loc\_2$
    act002: $need\_reschedule := TRUE$
    act301: $location\_of\_service3(core) := Send\_Queuing\_Message\_Wait \mapsto loc\_3$
end

Event send_queuing_message_needwait_return ⟨ordinary⟩ $\widehat{=}$

extends send_queuing_message_needwait_return
any
    part
    proc
    core
    port
where
    grd001: $part \in PARTITIONS$
    grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition)$
    grd003: $core \in CORES \cap dom(req\_busy\_resource\_proc) \wedge core \in dom(current\_processes\_flag) \wedge$
        $core \in dom(location\_of\_service2)$
    grd004: $proc = req\_busy\_resource\_proc(core)$
    grd005: $processes\_of\_partition(proc) = part$
    grd006: $part = current\_partition$
    grd012: $processes\_of\_partition(req\_busy\_resource\_proc(core)) \in dom(current\_partition\_flag)$
    grd007: $current\_partition\_flag(part) = TRUE$
    grd008: $current\_processes\_flag(core) = FALSE$
    grd009: $finished\_core2(core) = FALSE$
    grd010: $location\_of\_service2(core) = Req\_busy\_resource \mapsto loc\_2$
    grd011: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Req\_busy\_resource \mapsto$
        $loc\_2)$
    grd301: $port \in queuing\_ports$
    grd307: $core \in dom(location\_of\_service3)$
    grd302: $core \in dom(send\_queuing\_message\_port)$
    grd303: $port = send\_queuing\_message\_port(core)$

grd304: $finished\_core(core) = FALSE$

grd305: $location\_of\_service3(core) = Send\_Queuing\_Message\_Wait \mapsto loc\_3$

grd306: $\neg(finished\_core(core) = FALSE \wedge location\_of\_service3(core) = Send\_Queuing\_Message\_Wait \mapsto$
$loc\_3)$

then

act001: $location\_of\_service2(core) := Req\_busy\_resource \mapsto loc\_r$

act002: $finished\_core2(core) := TRUE$

act003: $req\_busy\_resource\_proc := \{core\} \lessdot req\_busy\_resource\_proc$

act301: $location\_of\_service3(core) := Send\_Queuing\_Message\_Wait \mapsto loc\_r$

act302: $send\_queuing\_message\_port := \{core\} \lessdot send\_queuing\_message\_port$

end

Event wakeup_waitproc_on_srcqueports_init ⟨ordinary⟩ ≙

extends wakeup_waitproc_on_srcqueports_init

any

    part

    proc

    newstate

    core

    port

where

grd001: $part \in PARTITIONS$

grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state)$

grd003: $newstate \in PROCESS\_STATES$

grd004: $core \in CORES$

grd005: $processes\_of\_partition(proc) = part$

grd017: $finished\_core2(core) = TRUE$

grd101: $partition\_mode(part) = PM\_NORMAL$

grd102: $process\_state(proc) = PS\_Waiting \vee process\_state(proc) = PS\_WaitandSuspend$

grd103: $process\_state(proc) = PS\_Waiting \Rightarrow newstate = PS\_Ready$

grd104: $process\_state(proc) = PS\_WaitandSuspend \Rightarrow newstate = PS\_Suspend$

grd201: $part = current\_partition$

grd203: $processes\_of\_partition(proc) \in dom(current\_partition\_flag)$

grd202: $current\_partition\_flag(part) = TRUE$

grd301: $port \in queuing\_ports$

grd302: $Direction\_of\_Ports(port) = PORT\_SOURCE$

grd303: $finite(queue\_of\_queuingports(port)) \wedge card(queue\_of\_queuingports(port)) < MaxMsgNum\_of\_QueuingP$

grd304: $proc \in dom(processes\_waitingfor\_queuingports(port))$

then

act001: $process\_state(proc) := newstate$

act201: $location\_of\_service2(core) := Resource\_become\_avail \mapsto loc\_i$

act202: $finished\_core2(core) := FALSE$

act203: $resource\_become\_avail\_proc(core) := proc$

act204: $timeout\_trigger := \{proc\} \lessdot timeout\_trigger$

act301: $location\_of\_service3(core) := Wakeup\_Waitproc\_on\_Srcqueports \mapsto loc\_i$

act302: $wakeup\_waitproc\_on\_srcqueports\_port(core) := port$

end

Event wakeup_waitproc_on_srcqueports_timeout_trig ⟨ordinary⟩ ≙

extends wakeup_waitproc_on_srcqueports_timeout_trig

any

    part

    proc

    core

    port

where

grd001: $part \in PARTITIONS$

grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition) \wedge proc \in dom(process\_wait\_type)$

grd003: $core \in CORES \cap dom(resource\_become\_avail\_proc) \wedge core \in dom(location\_of\_service2)$

grd004: $proc = resource\_become\_avail\_proc(core)$

grd005: $processes\_of\_partition(proc) = part$

grd006: $partition\_mode(part) = PM\_NORMAL$

grd007: $part = current\_partition$

grd013: $processes\_of\_partition(proc) \in dom(current\_partition\_flag)$

grd008: $current\_partition\_flag(part) = TRUE$

grd009: $process\_wait\_type(proc) = PROC\_WAIT\_OBJ$

grd010: $finished\_core2(core) = FALSE$

grd011: $location\_of\_service2(core) = Resource\_become\_avail \mapsto loc\_i$

grd012: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Resource\_become\_avail \mapsto loc\_i)$

grd301: $core \in dom(wakeup\_waitproc\_on\_srcqueports\_port)$

grd302: $port \in queuing\_ports$

grd303: $port = wakeup\_waitproc\_on\_srcqueports\_port(core)$

grd304: $proc \in dom(processes\_waiting for\_queuingports(port))$

grd305: $location\_of\_service3(core) = Wakeup\_Waitproc\_on\_Srcqueports \mapsto loc\_i$

grd306: $\neg(finished\_core(core) = FALSE \wedge location\_of\_service3(core) = Wakeup\_Waitproc\_on\_Srcqueports \mapsto loc\_i)$

then

act001: $location\_of\_service2(core) := Resource\_become\_avail \mapsto loc\_1$

act002: $process\_wait\_type := \{proc\} \lhd process\_wait\_type$

act301: $location\_of\_service3(core) := Wakeup\_Waitproc\_on\_Srcqueports \mapsto loc\_1$

end

Event wakeup_waitproc_on_srcqueports_delport ⟨ordinary⟩ ≙

extends wakeup_waitproc_on_srcqueports_delport

any

    part

    proc

    core

    port

    msg

    t

where

grd001: $part \in PARTITIONS \wedge part \in dom(current\_partition\_flag)$

grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition) \wedge proc \in dom(process\_wait\_type)$

grd003: $core \in CORES \cap dom(resource\_become\_avail\_proc) \cap dom(wakeup\_waitproc\_on\_srcqueports\_port) \cap dom(location\_of\_service3)$

grd004: $proc = resource\_become\_avail\_proc(core)$

grd005: $port \in queuing\_ports \wedge port \in ran(wakeup\_waitproc\_on\_srcqueports\_port)$

grd007: $t \in \mathbb{N}$

grd008: $processes\_of\_partition(proc) = part$

grd009: $partition\_mode(part) = PM\_NORMAL$

grd010: $part = current\_partition$

grd011: $current\_partition\_flag(part) = TRUE$

grd012: $process\_wait\_type(proc) = PROC\_WAIT\_OBJ$

grd013: $port = wakeup\_waitproc\_on\_srcqueports\_port(core)$

grd014: $Direction\_of\_Ports(port) = PORT\_SOURCE$

grd015: $finite(queue\_of\_queuingports(port)) \wedge card(queue\_of\_queuingports(port)) < MaxMsgNum\_of\_QueuingP$

grd016: $(proc \mapsto (msg \mapsto t)) \in processes\_waiting for\_queuingports(port)$

grd017: $finished\_core(core) = FALSE$

grd018: $location\_of\_service3(core) = Wakeup\_Waitproc\_on\_Srcqueports \mapsto loc\_1$

grd019: $\neg(finished\_core(core) = FALSE \wedge location\_of\_service3(core) = Wakeup\_Waitproc\_on\_Srcqueports \mapsto loc\_1)$

grd201: $quediscipline\_of\_queuingports(port) = QUEUE\_FIFO \Rightarrow (\forall p1, t1, m \cdot ((p1 \mapsto (m \mapsto t1)) \in processes\_waiting for\_queuingports(port) \Rightarrow t \le t1))$

grd202: $quediscipline\_of\_queuingports(port) = QUEUE\_PRIORITY \Rightarrow (\forall p1, t1, m \cdot ((p1 \mapsto (m \mapsto t1)) \in processes\_waiting for\_queuingports(port) \Rightarrow currentpriority\_of\_process(proc) \ge currentpriority\_of\_proces$

then

$\qquad$ act001: $location\_of\_service3(core) := Wakeup\_Waitproc\_on\_Srcqueports \mapsto loc\_2$

$\qquad$ act002: $processes\_waitingfor\_queuingports(port) := \{proc\} \lhd processes\_waitingfor\_queuingports(port)$

$\qquad$ act003: $queue\_of\_queuingports(port) := queue\_of\_queuingports(port) \Leftarrow \{msg \mapsto t\}$

end

Event wakeup_waitproc_on_srcqueports_schedule $\langle ordinary \rangle \;\widehat{=}$

extends wakeup_waitproc_on_srcqueports_schedule

$\quad$ **any**

$\qquad$ *part*

$\qquad$ *proc*

$\qquad$ *core*

$\qquad$ *resch*

$\qquad$ *port*

$\quad$ **where**

$\qquad$ grd001: $part \in PARTITIONS$

$\qquad$ grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition)$

$\qquad$ grd003: $core \in CORES \cap dom(resource\_become\_avail\_proc) \wedge core \in dom(location\_of\_service2)$

$\qquad$ grd004: $proc = resource\_become\_avail\_proc(core)$

$\qquad$ grd005: $processes\_of\_partition(proc) = part$

$\qquad$ grd006: $partition\_mode(part) = PM\_NORMAL$

$\qquad$ grd007: $part = current\_partition$

$\qquad$ grd013: $processes\_of\_partition(proc) \in dom(current\_partition\_flag)$

$\qquad$ grd008: $current\_partition\_flag(part) = TRUE$

$\qquad$ grd009: $resch \in BOOL$

$\qquad$ grd010: $finished\_core2(core) = FALSE$

$\qquad$ grd011: $location\_of\_service2(core) = Resource\_become\_avail \mapsto loc\_1$

$\qquad$ grd012: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Resource\_become\_avail \mapsto loc\_1)$

$\qquad$ grd301: $port \in queuing\_ports$

$\qquad$ grd302: $core \in dom(wakeup\_waitproc\_on\_srcqueports\_port)$

$\qquad$ grd303: $port = wakeup\_waitproc\_on\_srcqueports\_port(core)$

$\qquad$ grd304: $proc \in dom(processes\_waitingfor\_queuingports(port))$

$\qquad$ grd305: $location\_of\_service3(core) = Wakeup\_Waitproc\_on\_Srcqueports \mapsto loc\_2$

$\qquad$ grd306: $\neg(finished\_core(core) = FALSE \wedge location\_of\_service3(core) = Wakeup\_Waitproc\_on\_Srcqueports \mapsto loc\_2)$

$\quad$ **then**

$\qquad$ act001: $location\_of\_service2(core) := Resource\_become\_avail \mapsto loc\_2$

$\qquad$ act002: $need\_reschedule := resch$

$\qquad$ act301: $location\_of\_service3(core) := Wakeup\_Waitproc\_on\_Srcqueports \mapsto loc\_3$

$\quad$ **end**

Event wakeup_waitproc_on_srcqueports_return $\langle ordinary \rangle \;\widehat{=}$

extends wakeup_waitproc_on_srcqueports_return

$\quad$ **any**

$\qquad$ *part*

$\qquad$ *proc*

$\qquad$ *core*

$\qquad$ *port*

$\quad$ **where**

$\qquad$ grd001: $part \in PARTITIONS$

$\qquad$ grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition)$

$\qquad$ grd003: $core \in CORES \cap dom(resource\_become\_avail\_proc) \wedge core \in dom(location\_of\_service2)$

$\qquad$ grd004: $proc = resource\_become\_avail\_proc(core)$

$\qquad$ grd005: $processes\_of\_partition(proc) = part$

$\qquad$ grd006: $partition\_mode(part) = PM\_NORMAL$

$\qquad$ grd007: $part = current\_partition$

$\qquad$ grd012: $processes\_of\_partition(proc) \in dom(current\_partition\_flag)$

$\qquad$ grd008: $current\_partition\_flag(part) = TRUE$

$\qquad$ grd009: $finished\_core2(core) = FALSE$

$\qquad$ grd010: $location\_of\_service2(core) = Resource\_become\_avail \mapsto loc\_2$

grd011: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Resource\_become\_avail \mapsto loc\_2)$

grd301: $port \in queuing\_ports$

grd302: $core \in dom(wakeup\_waitproc\_on\_srcqueports\_port)$

grd303: $port = wakeup\_waitproc\_on\_srcqueports\_port(core)$

grd304: $proc \in dom(processes\_waitingfor\_queuingports(port))$

grd305: $location\_of\_service3(core) = Wakeup\_Waitproc\_on\_Srcqueports \mapsto loc\_3$

grd306: $\neg(finished\_core(core) = FALSE \wedge location\_of\_service3(core) = Wakeup\_Waitproc\_on\_Srcqueports \mapsto loc\_3)$

then

act001: $location\_of\_service2(core) := Resource\_become\_avail \mapsto loc\_r$

act002: $finished\_core2(core) := TRUE$

act003: $resource\_become\_avail\_proc := \{core\} \lhd resource\_become\_avail\_proc$

act301: $location\_of\_service3(core) := Wakeup\_Waitproc\_on\_Srcqueports \mapsto loc\_r$

act302: $wakeup\_waitproc\_on\_srcqueports\_port := \{core\} \lhd wakeup\_waitproc\_on\_srcqueports\_port$

end

Event wakeup_waitproc_on_dstqueports_init ⟨ordinary⟩ $\widehat{=}$

extends wakeup_waitproc_on_dstqueports_init

any

  part
  proc
  newstate
  core
  port

where

grd001: $part \in PARTITIONS$

grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state)$

grd003: $newstate \in PROCESS\_STATES$

grd004: $core \in CORES$

grd005: $processes\_of\_partition(proc) = part$

grd017: $finished\_core2(core) = TRUE$

grd101: $partition\_mode(part) = PM\_NORMAL$

grd102: $process\_state(proc) = PS\_Waiting \vee process\_state(proc) = PS\_WaitandSuspend$

grd103: $process\_state(proc) = PS\_Waiting \Rightarrow newstate = PS\_Ready$

grd104: $process\_state(proc) = PS\_WaitandSuspend \Rightarrow newstate = PS\_Suspend$

grd201: $part = current\_partition$

grd203: $processes\_of\_partition(proc) \in dom(current\_partition\_flag)$

grd202: $current\_partition\_flag(part) = TRUE$

grd301: $port \in queuing\_ports$

grd302: $Direction\_of\_Ports(port) = PORT\_DESTINATION$

grd303: $proc \in dom(processes\_waitingfor\_queuingports(port))$

grd304: $queue\_of\_queuingports(port) \neq \varnothing$

then

act001: $process\_state(proc) := newstate$

act201: $location\_of\_service2(core) := Resource\_become\_avail \mapsto loc\_i$

act202: $finished\_core2(core) := FALSE$

act203: $resource\_become\_avail\_proc(core) := proc$

act204: $timeout\_trigger := \{proc\} \lhd timeout\_trigger$

act301: $location\_of\_service3(core) := Wakeup\_Waitproc\_on\_Dstqueports \mapsto loc\_i$

act302: $wakeup\_waitproc\_on\_dstqueports\_port(core) := port$

end

Event wakeup_waitproc_on_dstqueports_timeout_trig ⟨ordinary⟩ $\widehat{=}$

extends wakeup_waitproc_on_dstqueports_timeout_trig

any

  part
  proc
  core
  port

where

    grd001: $part \in PARTITIONS$

    grd002: $proc \in processes \land proc \in dom(processes\_of\_partition) \land proc \in dom(process\_wait\_type)$

    grd003: $core \in CORES \cap dom(resource\_become\_avail\_proc) \land core \in dom(location\_of\_service2)$

    grd004: $proc = resource\_become\_avail\_proc(core)$

    grd005: $processes\_of\_partition(proc) = part$

    grd006: $partition\_mode(part) = PM\_NORMAL$

    grd007: $part = current\_partition$

    grd013: $processes\_of\_partition(proc) \in dom(current\_partition\_flag)$

    grd008: $current\_partition\_flag(part) = TRUE$

    grd009: $process\_wait\_type(proc) = PROC\_WAIT\_OBJ$

    grd010: $finished\_core2(core) = FALSE$

    grd011: $location\_of\_service2(core) = Resource\_become\_avail \mapsto loc\_i$

    grd012: $\neg(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Resource\_become\_avail \mapsto loc\_i)$

    grd301: $core \in dom(wakeup\_waitproc\_on\_dstqueports\_port)$

    grd302: $port \in queuing\_ports$

    grd303: $port = wakeup\_waitproc\_on\_dstqueports\_port(core)$

    grd304: $proc \in dom(processes\_waitingfor\_queuingports(port))$

    grd307: $queue\_of\_queuingports(port) \neq \varnothing$

    grd305: $location\_of\_service3(core) = Wakeup\_Waitproc\_on\_Dstqueports \mapsto loc\_i$

    grd306: $\neg(finished\_core2(core) = FALSE \land location\_of\_service3(core) = Wakeup\_Waitproc\_on\_Dstqueports \mapsto loc\_i)$

then

    act001: $location\_of\_service2(core) := Resource\_become\_avail \mapsto loc\_1$

    act002: $process\_wait\_type := \{proc\} \lhd process\_wait\_type$

    act301: $location\_of\_service3(core) := Wakeup\_Waitproc\_on\_Dstqueports \mapsto loc\_1$

end

Event wakeup_waitproc_on_dstqueports_delport ⟨ordinary⟩ $\widehat{=}$

extends wakeup_waitproc_on_dstqueports_delport

any

    part

    proc

    core

    port

    msg

    t

    t1

where

    grd001: $part \in PARTITIONS \land part \in dom(current\_partition\_flag)$

    grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_wait\_type)$

    grd003: $core \in CORES \cap dom(wakeup\_waitproc\_on\_dstqueports\_port) \cap dom(location\_of\_service3)$

    grd005: $port \in queuing\_ports$

    grd006: $t \in \mathbb{N}$

    grd007: $processes\_of\_partition(proc) = part$

    grd008: $partition\_mode(part) = PM\_NORMAL$

    grd009: $part = current\_partition$

    grd010: $current\_partition\_flag(part) = TRUE$

    grd011: $process\_wait\_type(proc) = PROC\_WAIT\_OBJ$

    grd012: $port = wakeup\_waitproc\_on\_dstqueports\_port(core)$

    grd013: $Direction\_of\_Ports(port) = PORT\_DESTINATION$

    grd014: $queue\_of\_queuingports(port) \neq \varnothing$

    grd015: $(proc \mapsto (msg \mapsto t)) \in processes\_waitingfor\_queuingports(port)$

    grd016: $finished\_core2(core) = FALSE$

    grd017: $location\_of\_service3(core) = Wakeup\_Waitproc\_on\_Dstqueports \mapsto loc\_1$

    grd018: $\neg(finished\_core2(core) = FALSE \land location\_of\_service3(core) = Wakeup\_Waitproc\_on\_Dstqueports \mapsto loc\_1)$

    grd201: $quediscipline\_of\_queuingports(port) = QUEUE\_FIFO \Rightarrow (\forall p1, tt, m \cdot (p1 \mapsto (m \mapsto tt) \in processes\_waitingfor\_queuingports(port) \Rightarrow t \leq tt))$

    grd202: $quediscipline\_of\_queuingports(port) = QUEUE\_PRIORITY \Rightarrow (\forall p1, tt, m \cdot (p1 \mapsto (m \mapsto$
        $tt) \in processes\_waitingfor\_queuingports(port) \Rightarrow currentpriority\_of\_process(proc) \geq currentpriority\_of\_process$

    grd203: $msg \mapsto t1 \in queue\_of\_queuingports(port)$
    grd204: $(\forall tt, mm \cdot (mm \mapsto tt \in queue\_of\_queuingports(port) \Rightarrow t1 \leq tt))$

then
    act001: $location\_of\_service3(core) := Wakeup\_Waitproc\_on\_Dstqueports \mapsto loc\_2$
    act002: $processes\_waitingfor\_queuingports(port) := \{proc\} \lhd processes\_waitingfor\_queuingports(port)$

    act003: $queue\_of\_queuingports(port) := queue\_of\_queuingports(port) \setminus \{msg \mapsto t\}$
end

Event wakeup_waitproc_on_dstqueports_schedule ⟨ordinary⟩ $\widehat{=}$
extends wakeup_waitproc_on_dstqueports_schedule
    any
        part
        proc
        core
        resch
        port
    where
        grd001: $part \in PARTITIONS$
        grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition)$
        grd003: $core \in CORES \cap dom(resource\_become\_avail\_proc) \wedge core \in dom(location\_of\_service2)$
        grd004: $proc = resource\_become\_avail\_proc(core)$
        grd005: $processes\_of\_partition(proc) = part$
        grd006: $partition\_mode(part) = PM\_NORMAL$
        grd007: $part = current\_partition$
        grd013: $processes\_of\_partition(proc) \in dom(current\_partition\_flag)$
        grd008: $current\_partition\_flag(part) = TRUE$
        grd009: $resch \in BOOL$
        grd010: $finished\_core2(core) = FALSE$
        grd011: $location\_of\_service2(core) = Resource\_become\_avail \mapsto loc\_1$
        grd012: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Resource\_become\_avail \mapsto$
          $loc\_1)$
        grd301: $port \in queuing\_ports$
        grd302: $core \in dom(wakeup\_waitproc\_on\_dstqueports\_port)$
        grd303: $port = wakeup\_waitproc\_on\_dstqueports\_port(core)$
        grd304: $proc \in dom(processes\_waitingfor\_queuingports(port))$
        grd305: $location\_of\_service3(core) = Wakeup\_Waitproc\_on\_Dstqueports \mapsto loc\_2$
        grd306: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service3(core) = Wakeup\_Waitproc\_on\_Dstqueports \mapsto$
          $loc\_2)$
    then
        act001: $location\_of\_service2(core) := Resource\_become\_avail \mapsto loc\_2$
        act002: $need\_reschedule := resch$
        act301: $location\_of\_service3(core) := Wakeup\_Waitproc\_on\_Dstqueports \mapsto loc\_3$
    end

Event wakeup_waitproc_on_dstqueports_return ⟨ordinary⟩ $\widehat{=}$
extends wakeup_waitproc_on_dstqueports_return
    any
        part
        proc
        core
        port
    where
        grd001: $part \in PARTITIONS$
        grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition)$
        grd003: $core \in CORES \cap dom(resource\_become\_avail\_proc) \wedge core \in dom(location\_of\_service2)$
        grd004: $proc = resource\_become\_avail\_proc(core)$
        grd005: $processes\_of\_partition(proc) = part$

        grd006: $partition\_mode(part) = PM\_NORMAL$

        grd007: $part = current\_partition$

        grd012: $processes\_of\_partition(proc) \in dom(current\_partition\_flag)$

        grd008: $current\_partition\_flag(part) = TRUE$

        grd009: $finished\_core2(core) = FALSE$

        grd010: $location\_of\_service2(core) = Resource\_become\_avail \mapsto loc\_2$

        grd011: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Resource\_become\_avail \mapsto loc\_2)$

        grd301: $port \in queuing\_ports$

        grd302: $core \in dom(wakeup\_waitproc\_on\_dstqueports\_port)$

        grd303: $port = wakeup\_waitproc\_on\_dstqueports\_port(core)$

        grd304: $proc \in dom(processes\_waiting for\_queuingports(port))$

        grd305: $location\_of\_service3(core) = Wakeup\_Waitproc\_on\_Dstqueports \mapsto loc\_3$

        grd306: $\neg(finished\_core(core) = FALSE \wedge location\_of\_service3(core) = Wakeup\_Waitproc\_on\_Dstqueports \mapsto loc\_3)$

then

        act001: $location\_of\_service2(core) := Resource\_become\_avail \mapsto loc\_r$

        act002: $finished\_core2(core) := TRUE$

        act003: $resource\_become\_avail\_proc := \{core\} \lhd resource\_become\_avail\_proc$

        act301: $location\_of\_service3(core) := Wakeup\_Waitproc\_on\_Dstqueports \mapsto loc\_r$

        act302: $wakeup\_waitproc\_on\_dstqueports\_port := \{core\} \lhd wakeup\_waitproc\_on\_dstqueports\_port$

end

Event receive_queuing_message ⟨ordinary⟩ ≘

extends receive_queuing_message

    any

        core

        port

        msg

        t

        part

    where

        grd001: $core \in CORES$

        grd002: $port \in queuing\_ports$

        grd003: $Direction\_of\_Ports(port) = PORT\_DESTINATION$

        grd004: $msg \in MESSAGES$

        grd005: $queue\_of\_queuingports(port) \neq \varnothing$

        grd006: $(msg \mapsto t) \in queue\_of\_queuingports(port)$

        grd007: $finished\_core2(core) = TRUE$

        grd201: $part = current\_partition$

        grd205: $part \in dom(current\_partition\_flag)$

        grd202: $current\_partition\_flag(part) = TRUE$

        grd203: $Ports\_of\_Partition(port) = part$

        grd204: $(msg \mapsto t) \in queue\_of\_queuingports(port) \wedge (\forall m, t1 \cdot (m \mapsto t1 \in queue\_of\_queuingports(port) \Rightarrow t \leq t1))$

    then

        act001: $queue\_of\_queuingports(port) := queue\_of\_queuingports(port) \setminus \{msg \mapsto t\}$

    end

Event receive_queuing_message_needwait_init ⟨ordinary⟩ ≘

extends receive_queuing_message_needwait_init

    any

        part

        proc

        newstate

        core

        port

    where

        grd001: $part \in PARTITIONS$

        grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state) \cap dom(process\_wait\_type)$

$grd003$: $newstate \in PROCESS\_STATES$

$grd004$: $core \in CORES \land core \in dom(current\_processes\_flag)$

$grd005$: $processes\_of\_partition(proc) = part$

$grd017$: $finished\_core2(core) = TRUE$

$grd101$: $partition\_mode(part) = PM\_NORMAL$

$grd102$: $process\_state(proc) = PS\_Running$

$grd103$: $newstate = PS\_Waiting$

$grd205$: $proc \in dom(delaytime\_of\_process) \land proc \in dom(process\_wait\_type)$

$grd201$: $part = current\_partition \land current\_partition \in dom(current\_partition\_flag)$

$grd202$: $current\_partition\_flag(part) = TRUE$

$grd203$: $current\_processes\_flag(core) = TRUE$

$grd204$: $proc = current\_processes(core)$

$grd301$: $port \in queuing\_ports$

$grd302$: $Direction\_of\_Ports(port) = PORT\_DESTINATION$

$grd303$: $queue\_of\_queuingports(port) = \varnothing$

then

$act001$: $process\_state(proc) := newstate$

$act002$: $location\_of\_service2(core) := Req\_busy\_resource \mapsto loc\_i$

$act003$: $finished\_core2(core) := FALSE$

$act004$: $req\_busy\_resource\_proc(core) := proc$

$act005$: $current\_processes\_flag(core) := FALSE$

$act006$: $current\_processes := \{core\} \ntriangleleft current\_processes$

$act301$: $location\_of\_service3(core) := Receive\_Queuing\_Message\_Wait \mapsto loc\_i$

$act302$: $receive\_queuing\_message\_port(core) := port$

end

Event receive_queuing_message_needwait_timeout $\langle$ordinary$\rangle$ $\widehat{=}$

extends receive_queuing_message_needwait_timeout

any

$part$

$proc$

$core$

$timeout$

$tmout\_trig$

wt

$port$

where

$grd001$: $part \in PARTITIONS$

$grd002$: $proc \in processes \land proc \in dom(processes\_of\_partition)$

$grd003$: $core \in CORES \cap dom(req\_busy\_resource\_proc) \land core \in dom(current\_processes\_flag) \land$ $core \in dom(location\_of\_service2)$

$grd004$: $proc = req\_busy\_resource\_proc(core)$

$grd005$: $processes\_of\_partition(proc) = part$

$grd006$: $part = current\_partition$

$grd018$: $processes\_of\_partition(req\_busy\_resource\_proc(core)) \in dom(current\_partition\_flag)$

$grd007$: $current\_partition\_flag(part) = TRUE$

$grd008$: $current\_processes\_flag(core) = TRUE$

$grd009$: $timeout \geq 0$

$grd010$: $wt \in PROCESS\_WAIT\_TYPES \land (wt = PROC\_WAIT\_OBJ \lor wt = PROC\_WAIT\_TIMEOUT)$

$grd011$: $tmout\_trig \in processes \nrightarrow (PROCESS\_STATES \times \mathbb{N}_1)$

$grd012$:

    $(timeout = INFINITE\_TIME\_VALUE \Rightarrow tmout\_trig = \varnothing)$

    $\land (timeout > 0 \Rightarrow tmout\_trig = \{proc \mapsto (PS\_Ready \mapsto (timeout + clock\_tick * ONE\_TICK\_TIME))\})$

$grd013$: $timeout > 0 \Rightarrow wt = PROC\_WAIT\_TIMEOUT$

$grd014$: $timeout = INFINITE\_TIME\_VALUE \Rightarrow wt = PROC\_WAIT\_OBJ$

$grd015$: $finished\_core2(core) = FALSE$

$grd016$: $location\_of\_service2(core) = Req\_busy\_resource \mapsto loc\_i$

$grd017$: $\neg (finished\_core2(core) = FALSE \land location\_of\_service2(core) = Req\_busy\_resource \mapsto loc\_i)$

      grd301: $core \in dom(receive\_queuing\_message\_port)$

      grd302: $port \in queuing\_ports$

      grd303: $port = receive\_queuing\_message\_port(core)$

      grd304: $queue\_of\_queuingports(port) = \varnothing$

      grd305: $location\_of\_service3(core) = Receive\_Queuing\_Message\_Wait \mapsto loc\_i$

      grd306: $\neg(finished\_core2(core) = FALSE \land location\_of\_service3(core) = Receive\_Queuing\_Message\_Wait \mapsto loc\_i)$

then

      act001: $location\_of\_service2(core) := Req\_busy\_resource \mapsto loc\_1$

      act002: $timeout\_trigger := timeout\_trigger \lhd tmout\_trig$

      act003: $process\_wait\_type(proc) := wt$

      act301: $location\_of\_service3(core) := Receive\_Queuing\_Message\_Wait \mapsto loc\_1$

end

Event receive_queuing_message_needwait_insert ⟨ordinary⟩ $\widehat{=}$

extends receive_queuing_message_needwait_insert

    any

      part

      proc

      core

      port

      msg

      t

    where

      grd001: $part \in PARTITIONS \land part \in dom(current\_partition\_flag)$

      grd002: $proc \in processes \cap dom(processes\_of\_partition)$

      grd003: $core \in CORES \cap dom(receive\_queuing\_message\_port) \cap dom(req\_busy\_resource\_proc)$

      grd004: $processes\_of\_partition(proc) = part$

      grd016: $proc = req\_busy\_resource\_proc(core)$

      grd005: $part = current\_partition$

      grd006: $current\_partition\_flag(part) = TRUE$

      grd007: $current\_processes\_flag(core) = TRUE$

      grd008: $port \in queuing\_ports$

      grd009: $port = receive\_queuing\_message\_port(core)$

      grd010: $Direction\_of\_Ports(port) = PORT\_DESTINATION$

      grd011: $queue\_of\_queuingports(port) = \varnothing$

      grd012: $(msg \mapsto t) \in queue\_of\_queuingports(port)$

      grd013: $finished\_core2(core) = FALSE$

      grd014: $location\_of\_service3(core) = Receive\_Queuing\_Message\_Wait \mapsto loc\_1$

      grd015: $\neg(finished\_core2(core) = FALSE \land location\_of\_service3(core) = Receive\_Queuing\_Message\_Wait \mapsto loc\_1)$

      grd201: $locklevel\_of\_partition(part) = 0$

    then

      act001: $location\_of\_service3(core) := Receive\_Queuing\_Message\_Wait \mapsto loc\_2$

      act002: $processes\_waitingfor\_queuingports(port) := processes\_waitingfor\_queuingports(port) \lhd \{proc \mapsto (msg \mapsto t)\}$

    end

Event receive_queuing_message_needwait_schedule ⟨ordinary⟩ $\widehat{=}$

extends receive_queuing_message_needwait_schedule

    any

      part

      proc

      core

      port

    where

      grd001: $part \in PARTITIONS$

      grd002: $proc \in processes \land proc \in dom(processes\_of\_partition)$

      grd003: $core \in CORES \cap dom(req\_busy\_resource\_proc) \land core \in dom(current\_processes\_flag) \land core \in dom(location\_of\_service2)$

      grd004: $proc = req\_busy\_resource\_proc(core)$

   grd005: $processes\_of\_partition(proc) = part$
   grd006: $part = current\_partition$
   grd012: $processes\_of\_partition(req\_busy\_resource\_proc(core)) \in dom(current\_partition\_flag)$
   grd007: $current\_partition\_flag(part) = TRUE$
   grd008: $current\_processes\_flag(core) = FALSE$
   grd009: $finished\_core2(core) = FALSE$
   grd010: $location\_of\_service2(core) = Req\_busy\_resource \mapsto loc\_1$
   grd011: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Req\_busy\_resource \mapsto$
    $loc\_1)$
   grd301: $core \in dom(receive\_queuing\_message\_port)$
   grd302: $port \in queuing\_ports$
   grd303: $port = receive\_queuing\_message\_port(core)$
   grd304: $queue\_of\_queuingports(port) = \varnothing$
   grd305: $location\_of\_service3(core) = Receive\_Queuing\_Message\_Wait \mapsto loc\_2$
   grd306: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service3(core) = Receive\_Queuing\_Message\_Wait \mapsto$
    $loc\_2)$
  then
   act001: $location\_of\_service2(core) := Req\_busy\_resource \mapsto loc\_2$
   act002: $need\_reschedule := TRUE$
   act301: $location\_of\_service3(core) := Receive\_Queuing\_Message\_Wait \mapsto loc\_3$
  end

Event receive_queuing_message_needwait_return ⟨ordinary⟩ $\widehat{=}$

extends receive_queuing_message_needwait_return

  any
   $part$
   $proc$
   $core$
   $port$
  where
   grd001: $part \in PARTITIONS$
   grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition)$
   grd003: $core \in CORES \cap dom(req\_busy\_resource\_proc) \wedge core \in dom(current\_processes\_flag) \wedge$
    $core \in dom(location\_of\_service2)$
   grd004: $proc = req\_busy\_resource\_proc(core)$
   grd005: $processes\_of\_partition(proc) = part$
   grd006: $part = current\_partition$
   grd012: $processes\_of\_partition(req\_busy\_resource\_proc(core)) \in dom(current\_partition\_flag)$
   grd007: $current\_partition\_flag(part) = TRUE$
   grd008: $current\_processes\_flag(core) = FALSE$
   grd009: $finished\_core2(core) = FALSE$
   grd010: $location\_of\_service2(core) = Req\_busy\_resource \mapsto loc\_2$
   grd011: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Req\_busy\_resource \mapsto$
    $loc\_2)$
   grd301: $core \in dom(receive\_queuing\_message\_port)$
   grd302: $port \in queuing\_ports$
   grd303: $port = receive\_queuing\_message\_port(core)$
   grd304: $queue\_of\_queuingports(port) = \varnothing$
   grd305: $location\_of\_service3(core) = Receive\_Queuing\_Message\_Wait \mapsto loc\_3$
   grd306: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service3(core) = Receive\_Queuing\_Message\_Wait \mapsto$
    $loc\_3)$
  then
   act001: $location\_of\_service2(core) := Req\_busy\_resource \mapsto loc\_r$
   act002: $finished\_core2(core) := TRUE$
   act003: $req\_busy\_resource\_proc := \{core\} \lhd req\_busy\_resource\_proc$
   act301: $location\_of\_service3(core) := Receive\_Queuing\_Message\_Wait \mapsto loc\_r$
   act302: $receive\_queuing\_message\_port := \{core\} \lhd receive\_queuing\_message\_port$
  end

Event get_queuing_port_id ⟨ordinary⟩ $\widehat{=}$

  any

   part
   core
   port

where

   grd001: $part = current\_partition$

   grd002: $port \in queuing\_ports$

   grd003: $part \in dom(current\_partition\_flag) \wedge current\_partition = part \wedge current\_partition\_flag(part) = TRUE$

   grd004: $Ports\_of\_Partition(port) = part$

   grd005: $core \in CORES$

   grd006: $finished\_core2(core) = TRUE$

then

   skip

end

Event get_queuing_port_status ⟨ordinary⟩ $\widehat{=}$

 any

   part
   core
   port

 where

   grd001: $part = current\_partition$

   grd002: $port \in queuing\_ports$

   grd003: $part \in dom(current\_partition\_flag) \wedge current\_partition = part \wedge current\_partition\_flag(part) = TRUE$

   grd004: $Ports\_of\_Partition(port) = part$

   grd005: $core \in CORES$

   grd006: $finished\_core2(core) = TRUE$

 then

   skip

 end

Event clear_queuing_port ⟨ordinary⟩ $\widehat{=}$

extends clear_queuing_port

 any

   core
   port
   part

 where

   grd001: $core \in CORES$

   grd002: $port \in queuing\_ports$

   grd003: $Direction\_of\_Ports(port) = PORT\_DESTINATION$

   grd004: $finished\_core(core) = TRUE$

   grd201: $part \in dom(current\_partition\_flag) \wedge current\_partition = part \wedge current\_partition\_flag(part) = TRUE$

   grd203: $Ports\_of\_Partition(port) = part$

 then

   act001: $queue\_of\_queuingports(port) := \varnothing$

 end

Event create_buffer ⟨ordinary⟩ $\widehat{=}$

extends create_buffer

 any

   part
   core
   buf
   max_msg_size
   disc

 where

   grd001: $core \in CORES$

   grd002: $buf \in BUFFERS \wedge buf \notin buffers$

   grd003: $finished\_core2(core) = TRUE$

grd004: $max\_msg\_size \in \mathbb{N}_1$

grd005: $part \in PARTITIONS$

grd008: $buf \in dom(queue\_of\_buffers)$

grd007: $finite(queue\_of\_buffers(buf))$

grd006: $part = current\_partition$

grd201: $disc \in QUEUING\_DISCIPLINE$

grd202: $current\_partition\_flag(part) = TRUE$

grd204: $part \in dom(current\_partition\_flag)$

grd203: $(partition\_mode(current\_partition) = PM\_COLD\_START \lor partition\_mode(current\_partition) = PM\_WARM\_START)$

then

act001: $buffers := buffers \cup \{buf\}$

act002: $MaxMsgNum\_of\_Buffers(buf) := max\_msg\_size$

act003: $queue\_of\_buffers(buf) := \varnothing$

act004: $buffers\_of\_partition(buf) := part$

act005: $processes\_waitingfor\_buffers(buf) := \varnothing$

act201: $quediscipline\_of\_buffers(buf) := disc$

end

Event send_buffer ⟨ordinary⟩ $\widehat{=}$

extends send_buffer

any

    core

    buf

    msg

    t

    part

where

grd001: $core \in CORES$

grd002: $buf \in buffers$

grd003: $msg \in MESSAGES \land msg \notin used\_messages$

grd004: $t \in \mathbb{N}$

grd005: $finite(queue\_of\_buffers(buf)) \land card(queue\_of\_buffers(buf)) < MaxMsgNum\_of\_Buffers(buf)$

grd006: $finished\_core2(core) = TRUE$

grd201: $part \in dom(current\_partition\_flag) \land current\_partition = part \land current\_partition\_flag(part) = TRUE$

grd203: $buffers\_of\_partition(buf) = part$

grd204: $t = clock\_tick * ONE\_TICK\_TIME$

then

act001: $queue\_of\_buffers(buf) := queue\_of\_buffers(buf) \lhd\!\!\!- \{msg \mapsto t\}$

act002: $used\_messages := used\_messages \cup \{msg\}$

end

Event send_buffer_needwakeuprecvproc_init ⟨ordinary⟩ $\widehat{=}$

extends send_buffer_needwakeuprecvproc_init

any

    part

    proc

    newstate

    core

    buf

where

grd001: $part \in PARTITIONS$

grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state)$

grd003: $newstate \in PROCESS\_STATES$

grd004: $core \in CORES$

grd005: $processes\_of\_partition(proc) = part$

grd017: $finished\_core2(core) = TRUE$

grd101: $partition\_mode(part) = PM\_NORMAL$

grd102: $process\_state(proc) = PS\_Waiting \lor process\_state(proc) = PS\_WaitandSuspend$

grd103: $process\_state(proc) = PS\_Waiting \Rightarrow newstate = PS\_Ready$
grd104: $process\_state(proc) = PS\_WaitandSuspend \Rightarrow newstate = PS\_Suspend$
grd201: $part = current\_partition$
grd203: $processes\_of\_partition(proc) \in dom(current\_partition\_flag)$
grd202: $current\_partition\_flag(part) = TRUE$
grd301: $buf \in buffers$
grd302: $finite(queue\_of\_buffers(buf)) \wedge card(queue\_of\_buffers(buf)) < MaxMsgNum\_of\_Buffers(buf)$

grd303: $processes\_waitingfor\_buffers(buf) \neq \varnothing$
grd304: $proc \in dom(processes\_waitingfor\_buffers(buf))$

then

act001: $process\_state(proc) := newstate$
act201: $location\_of\_service2(core) := Resource\_become\_avail \mapsto loc\_i$
act202: $finished\_core2(core) := FALSE$
act203: $resource\_become\_avail\_proc(core) := proc$
act204: $timeout\_trigger := \{proc\} \lhd timeout\_trigger$
act301: $location\_of\_service3(core) := Send\_Buffer\_NeedWakeup \mapsto loc\_i$
act302: $send\_buffer\_needwakeup(core) := buf$

end

Event send_buffer_needwakeuprecvproc_timeout_trig $\langle ordinary \rangle$ $\widehat{=}$

extends send_buffer_needwakeuprecvproc_timeout_trig

any

    part
    proc
    core
    buf

where

grd001: $part \in PARTITIONS$
grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition) \wedge proc \in dom(process\_wait\_type)$
grd003: $core \in CORES \cap dom(resource\_become\_avail\_proc) \wedge core \in dom(location\_of\_service2)$
grd004: $proc = resource\_become\_avail\_proc(core)$
grd005: $processes\_of\_partition(proc) = part$
grd006: $partition\_mode(part) = PM\_NORMAL$
grd007: $part = current\_partition$
grd013: $processes\_of\_partition(proc) \in dom(current\_partition\_flag)$
grd008: $current\_partition\_flag(part) = TRUE$
grd009: $process\_wait\_type(proc) = PROC\_WAIT\_OBJ$
grd010: $finished\_core2(core) = FALSE$
grd011: $location\_of\_service2(core) = Resource\_become\_avail \mapsto loc\_i$
grd012: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Resource\_become\_avail \mapsto loc\_i)$
grd301: $core \in dom(send\_buffer\_needwakeup)$
grd302: $buf \in buffers$
grd303: $buf = send\_buffer\_needwakeup(core)$
grd304: $proc \in dom(processes\_waitingfor\_buffers(buf))$
grd305: $location\_of\_service3(core) = Send\_Buffer\_NeedWakeup \mapsto loc\_i$
grd306: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service3(core) = Send\_Buffer\_NeedWakeup \mapsto loc\_i)$

then

act001: $location\_of\_service2(core) := Resource\_become\_avail \mapsto loc\_1$
act002: $process\_wait\_type := \{proc\} \lhd process\_wait\_type$
act301: $location\_of\_service3(core) := Send\_Buffer\_NeedWakeup \mapsto loc\_1$

end

Event send_buffer_needwakeuprecvproc_wakeupproc $\langle ordinary \rangle$ $\widehat{=}$

extends send_buffer_needwakeuprecvproc_wakeupproc

any

    part
    proc
    core

   buf

   msg

   t

   m

  where

    grd001: $part \in PARTITIONS$

    grd002: $proc \in processes \cap dom(processes\_of\_partition)$

    grd003: $core \in CORES \cap dom(send\_buffer\_needwakeup) \cap dom(resource\_become\_avail\_proc) \cap$
     $dom(location\_of\_service3)$

    grd004: $proc = resource\_become\_avail\_proc(core)$

    grd005: $buf \in buffers$

    grd006: $msg \in MESSAGES \wedge msg \notin used\_messages$

    grd007: $processes\_of\_partition(proc) = part$

    grd008: $partition\_mode(part) = PM\_NORMAL$

    grd009: $buf = send\_buffer\_needwakeup(core)$

    grd010: $finished\_core2(core) = FALSE$

    grd011: $location\_of\_service3(core) = Send\_Buffer\_NeedWakeup \mapsto loc\_1$

    grd012: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service3(core) = Send\_Buffer\_NeedWakeup \mapsto$
     $loc\_1)$

    grd201: $t \in \mathbb{N} \wedge m \in MESSAGES$

    grd202: $processes\_waitingfor\_buffers(buf) \neq \varnothing \wedge (proc \mapsto (m \mapsto WAITING\_R \mapsto t)) \in$
     $processes\_waitingfor\_buffers(buf)$

    grd203: $quediscipline\_of\_buffers(buf) = QUEUE\_FIFO \Rightarrow (\forall p1, m1, t1 \cdot (p1 \mapsto (m1 \mapsto WAITING\_R \mapsto$
     $t1) \in processes\_waitingfor\_buffers(buf) \Rightarrow t \leq t1))$

    grd204: $quediscipline\_of\_buffers(buf) = QUEUE\_PRIORITY \Rightarrow (\forall p1, m1, t1 \cdot (p1 \mapsto (m1 \mapsto$
     $WAITING\_R \mapsto t1) \in processes\_waitingfor\_buffers(buf) \Rightarrow currentpriority\_of\_process(proc) \geq$
     $currentpriority\_of\_process(p1)))$

  then

    act001: $location\_of\_service3(core) := Send\_Buffer\_NeedWakeup \mapsto loc\_2$

    act002: $used\_messages := used\_messages \cup \{msg\}$

    act003: $processes\_waitingfor\_buffers(buf) := \{proc\} \lhd processes\_waitingfor\_buffers(buf)$

  end

Event send_buffer_needwakeuprecvproc_schedule $\langle ordinary \rangle$ $\widehat{=}$

extends send_buffer_needwakeuprecvproc_schedule

  any

   $part$

   $proc$

   $core$

   $resch$

   buf

  where

    grd001: $part \in PARTITIONS$

    grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition)$

    grd003: $core \in CORES \cap dom(resource\_become\_avail\_proc) \wedge core \in dom(location\_of\_service2)$

    grd004: $proc = resource\_become\_avail\_proc(core)$

    grd005: $processes\_of\_partition(proc) = part$

    grd006: $partition\_mode(part) = PM\_NORMAL$

    grd007: $part = current\_partition$

    grd013: $processes\_of\_partition(proc) \in dom(current\_partition\_flag)$

    grd008: $current\_partition\_flag(part) = TRUE$

    grd009: $resch \in BOOL$

    grd010: $finished\_core2(core) = FALSE$

    grd011: $location\_of\_service2(core) = Resource\_become\_avail \mapsto loc\_1$

    grd012: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Resource\_become\_avail \mapsto$
     $loc\_1)$

    grd301: $buf \in buffers$

    grd302: $core \in dom(send\_buffer\_needwakeup)$

    grd303: $buf = send\_buffer\_needwakeup(core)$

    grd304: $location\_of\_service3(core) = Send\_Buffer\_NeedWakeup \mapsto loc\_2$

        grd305: $\neg(finished\_core2(core) = FALSE \land location\_of\_service3(core) = Send\_Buffer\_NeedWakeup \mapsto loc\_2)$

then

        act001: $location\_of\_service2(core) := Resource\_become\_avail \mapsto loc\_2$

        act002: $need\_reschedule := resch$

        act301: $location\_of\_service3(core) := Send\_Buffer\_NeedWakeup \mapsto loc\_3$

end

Event send_buffer_needwakeuprecvproc_return $\langle ordinary \rangle \;\widehat{=}$

extends send_buffer_needwakeuprecvproc_return

    any

        part

        proc

        core

        buf

    where

        grd001: $part \in PARTITIONS$

        grd002: $proc \in processes \land proc \in dom(processes\_of\_partition)$

        grd003: $core \in CORES \cap dom(resource\_become\_avail\_proc) \land core \in dom(location\_of\_service2)$

        grd004: $proc = resource\_become\_avail\_proc(core)$

        grd005: $processes\_of\_partition(proc) = part$

        grd006: $partition\_mode(part) = PM\_NORMAL$

        grd007: $part = current\_partition$

        grd012: $processes\_of\_partition(proc) \in dom(current\_partition\_flag)$

        grd008: $current\_partition\_flag(part) = TRUE$

        grd009: $finished\_core2(core) = FALSE$

        grd010: $location\_of\_service2(core) = Resource\_become\_avail \mapsto loc\_2$

        grd011: $\neg(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Resource\_become\_avail \mapsto loc\_2)$

        grd301: $buf \in buffers$

        grd302: $core \in dom(send\_buffer\_needwakeup)$

        grd303: $buf = send\_buffer\_needwakeup(core)$

        grd304: $location\_of\_service3(core) = Send\_Buffer\_NeedWakeup \mapsto loc\_3$

        grd305: $\neg(finished\_core2(core) = FALSE \land location\_of\_service3(core) = Send\_Buffer\_NeedWakeup \mapsto loc\_3)$

    then

        act001: $location\_of\_service2(core) := Resource\_become\_avail \mapsto loc\_r$

        act002: $finished\_core2(core) := TRUE$

        act003: $resource\_become\_avail\_proc := \{core\} \lhd resource\_become\_avail\_proc$

        act301: $location\_of\_service3(core) := Send\_Buffer\_NeedWakeup \mapsto loc\_r$

        act302: $send\_buffer\_needwakeup := \{core\} \lhd send\_buffer\_needwakeup$

    end

Event send_buffer_withfull_init $\langle ordinary \rangle \;\widehat{=}$

extends send_buffer_withfull_init

    any

        part

        proc

        newstate

        core

        buf

    where

        grd001: $part \in PARTITIONS$

        grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state) \cap dom(process\_wait\_type)$

        grd003: $newstate \in PROCESS\_STATES$

        grd004: $core \in CORES \land core \in dom(current\_processes\_flag)$

        grd005: $processes\_of\_partition(proc) = part$

        grd017: $finished\_core2(core) = TRUE$

        grd101: $partition\_mode(part) = PM\_NORMAL$

        grd102: $process\_state(proc) = PS\_Running$

> grd103: $newstate = PS\_Waiting$
> grd205: $proc \in dom(delaytime\_of\_process) \wedge proc \in dom(process\_wait\_type)$
> grd201: $part = current\_partition \wedge current\_partition \in dom(current\_partition\_flag)$
> grd202: $current\_partition\_flag(part) = TRUE$
> grd203: $current\_processes\_flag(core) = TRUE$
> grd204: $proc = current\_processes(core)$
> grd301: $buf \in buffers$
> grd302: $buffers\_of\_partition(buf) = part$
> grd303: $finite(queue\_of\_buffers(buf)) \wedge card(queue\_of\_buffers(buf)) = MaxMsgNum\_of\_Buffers(buf)$

then

> act001: $process\_state(proc) := newstate$
> act002: $location\_of\_service2(core) := Req\_busy\_resource \mapsto loc\_i$
> act003: $finished\_core2(core) := FALSE$
> act004: $req\_busy\_resource\_proc(core) := proc$
> act005: $current\_processes\_flag(core) := FALSE$
> act006: $current\_processes := \{core\} \lhd current\_processes$
> act301: $location\_of\_service3(core) := Send\_Buffer\_Withfull \mapsto loc\_i$
> act302: $send\_buffer\_withfull(core) := buf$

end

Event send_buffer_withfull_timeout ⟨ordinary⟩ $\widehat{=}$

extends send_buffer_withfull_timeout

any

> *part*
> *proc*
> *core*
> *timeout*
> *tmout_trig*
> *wt*
> *buf*

where

> grd001: $part \in PARTITIONS$
> grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition)$
> grd003: $core \in CORES \cap dom(req\_busy\_resource\_proc) \wedge core \in dom(current\_processes\_flag) \wedge$
> $core \in dom(location\_of\_service2)$
> grd004: $proc = req\_busy\_resource\_proc(core)$
> grd005: $processes\_of\_partition(proc) = part$
> grd006: $part = current\_partition$
> grd018: $processes\_of\_partition(req\_busy\_resource\_proc(core)) \in dom(current\_partition\_flag)$
> grd007: $current\_partition\_flag(part) = TRUE$
> grd008: $current\_processes\_flag(core) = TRUE$
> grd009: $timeout \geq 0$
> grd010: $wt \in PROCESS\_WAIT\_TYPES \wedge (wt = PROC\_WAIT\_OBJ \vee wt = PROC\_WAIT\_TIMEOUT)$
>
> grd011: $tmout\_trig \in processes \nrightarrow (PROCESS\_STATES \times \mathbb{N}_1)$
> grd012:
> $(timeout = INFINITE\_TIME\_VALUE \Rightarrow tmout\_trig = \varnothing)$
> $\wedge (timeout > 0 \Rightarrow tmout\_trig = \{proc \mapsto (PS\_Ready \mapsto (timeout + clock\_tick * ONE\_TICK\_TIME))\})$
>
> grd013: $timeout > 0 \Rightarrow wt = PROC\_WAIT\_TIMEOUT$
> grd014: $timeout = INFINITE\_TIME\_VALUE \Rightarrow wt = PROC\_WAIT\_OBJ$
> grd015: $finished\_core2(core) = FALSE$
> grd016: $location\_of\_service2(core) = Req\_busy\_resource \mapsto loc\_i$
> grd017: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Req\_busy\_resource \mapsto$
> $loc\_i)$
> grd301: $buf \in buffers$
> grd302: $core \in dom(send\_buffer\_withfull)$
> grd303: $buf = send\_buffer\_withfull(core)$
> grd304: $location\_of\_service3(core) = Send\_Buffer\_Withfull \mapsto loc\_i$

$grd305$: $\neg(finished\_core2(core) = FALSE \land location\_of\_service3(core) = Send\_Buffer\_Withfull \mapsto loc\_i)$

then

$act001$: $location\_of\_service2(core) := Req\_busy\_resource \mapsto loc\_1$

$act002$: $timeout\_trigger := timeout\_trigger \ensuremath{\lhd\mkern-14mu-} tmout\_trig$

$act003$: $process\_wait\_type(proc) := wt$

$act301$: $location\_of\_service3(core) := Send\_Buffer\_Withfull \mapsto loc\_1$

end

Event send_buffer_withfull_waiting ⟨ordinary⟩ $\widehat{=}$

extends send_buffer_withfull_waiting

any

    part

    proc

    core

    buf

    msg

    t

where

$grd001$: $part \in PARTITIONS$

$grd002$: $proc \in processes \cap dom(processes\_of\_partition)$

$grd003$: $core \in CORES \cap dom(req\_busy\_resource\_proc) \cap dom(send\_buffer\_withfull) \cap dom(location\_of\_service3)$

$grd004$: $proc = req\_busy\_resource\_proc(core)$

$grd005$: $processes\_of\_partition(proc) = part$

$grd006$: $buf \in buffers$

$grd007$: $buf = send\_buffer\_withfull(core)$

$grd008$: $msg \in MESSAGES \land msg \notin used\_messages$

$grd009$: $buffers\_of\_partition(buf) = part$

$grd010$: $finite(queue\_of\_buffers(buf)) \land card(queue\_of\_buffers(buf)) = MaxMsgNum\_of\_Buffers(buf)$

$grd014$: $t \in \mathbb{N}$

$grd011$: $finished\_core(core) = FALSE$

$grd012$: $location\_of\_service3(core) = Send\_Buffer\_Withfull \mapsto loc\_1$

$grd13$: $\neg(finished\_core2(core) = FALSE \land location\_of\_service3(core) = Send\_Buffer\_Withfull \mapsto loc\_1)$

$grd201$: $t = clock\_tick * ONE\_TICK\_TIME$

then

$act001$: $location\_of\_service3(core) := Send\_Buffer\_Withfull \mapsto loc\_2$

$act002$: $used\_messages := used\_messages \cup \{msg\}$

$act003$: $processes\_waitingfor\_buffers(buf) := processes\_waitingfor\_buffers(buf) \ensuremath{\lhd\mkern-14mu-} \{proc \mapsto (msg \mapsto WAITING\_W \mapsto t)\}$

end

Event send_buffer_withfull_schedule ⟨ordinary⟩ $\widehat{=}$

extends send_buffer_withfull_schedule

any

    part

    proc

    core

    buf

where

$grd001$: $part \in PARTITIONS$

$grd002$: $proc \in processes \land proc \in dom(processes\_of\_partition)$

$grd003$: $core \in CORES \cap dom(req\_busy\_resource\_proc) \land core \in dom(current\_processes\_flag) \land core \in dom(location\_of\_service2)$

$grd004$: $proc = req\_busy\_resource\_proc(core)$

$grd005$: $processes\_of\_partition(proc) = part$

$grd006$: $part = current\_partition$

$grd012$: $processes\_of\_partition(req\_busy\_resource\_proc(core)) \in dom(current\_partition\_flag)$

$grd007$: $current\_partition\_flag(part) = TRUE$

        grd008: $current\_processes\_flag(core) = FALSE$
        grd009: $finished\_core2(core) = FALSE$
        grd010: $location\_of\_service2(core) = Req\_busy\_resource \mapsto loc\_1$
        grd011: $\neg(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Req\_busy\_resource \mapsto loc\_1)$
        grd301: $buf \in buffers$
        grd302: $buf = send\_buffer\_withfull(core)$
        grd303: $buffers\_of\_partition(buf) = part$
        grd304: $location\_of\_service3(core) = Send\_Buffer\_Withfull \mapsto loc\_2$
        grd305: $\neg(finished\_core(core) = FALSE \land location\_of\_service3(core) = Send\_Buffer\_Withfull \mapsto loc\_2)$

then
        act001: $location\_of\_service2(core) := Req\_busy\_resource \mapsto loc\_2$
        act002: $need\_reschedule := TRUE$
        act301: $location\_of\_service3(core) := Send\_Buffer\_Withfull \mapsto loc\_3$
end

Event send_buffer_withfull_return $\langle ordinary \rangle \;\widehat{=}$

extends send_buffer_withfull_return

    any
        part
        proc
        core
        buf
    where
        grd001: $part \in PARTITIONS$
        grd002: $proc \in processes \land proc \in dom(processes\_of\_partition)$
        grd003: $core \in CORES \cap dom(req\_busy\_resource\_proc) \land core \in dom(current\_processes\_flag) \land core \in dom(location\_of\_service2)$
        grd004: $proc = req\_busy\_resource\_proc(core)$
        grd005: $processes\_of\_partition(proc) = part$
        grd006: $part = current\_partition$
        grd012: $processes\_of\_partition(req\_busy\_resource\_proc(core)) \in dom(current\_partition\_flag)$
        grd007: $current\_partition\_flag(part) = TRUE$
        grd008: $current\_processes\_flag(core) = FALSE$
        grd009: $finished\_core2(core) = FALSE$
        grd010: $location\_of\_service2(core) = Req\_busy\_resource \mapsto loc\_2$
        grd011: $\neg(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Req\_busy\_resource \mapsto loc\_2)$
        grd301: $buf \in buffers$
        grd302: $buf = send\_buffer\_withfull(core)$
        grd303: $buffers\_of\_partition(buf) = part$
        grd304: $location\_of\_service3(core) = Send\_Buffer\_Withfull \mapsto loc\_3$
        grd305: $\neg(finished\_core2(core) = FALSE \land location\_of\_service3(core) = Send\_Buffer\_Withfull \mapsto loc\_3)$

    then
        act001: $location\_of\_service2(core) := Req\_busy\_resource \mapsto loc\_r$
        act002: $finished\_core2(core) := TRUE$
        act003: $req\_busy\_resource\_proc := \{core\} \lhd req\_busy\_resource\_proc$
        act301: $location\_of\_service3(core) := Send\_Buffer\_Withfull \mapsto loc\_r$
        act302: $send\_buffer\_withfull := \{core\} \lhd send\_buffer\_withfull$
    end

Event receive_buffer $\langle ordinary \rangle \;\widehat{=}$

extends receive_buffer

    any
        core
        buf
        msg
        t
    where

        grd001: $core \in CORES$
        grd002: $buf \in buffers$
        grd003: $queue\_of\_buffers(buf) \neq \varnothing$
        grd004: $(msg \mapsto t) \in queue\_of\_buffers(buf)$
        grd005: $finished\_core2(core) = TRUE$
        grd201: $msg \mapsto t \in queue\_of\_buffers(buf) \wedge (\forall m1, t1 \cdot (m1 \mapsto t1 \in queue\_of\_buffers(buf) \Rightarrow t \leq t1))$
        grd202: $processes\_waitingfor\_buffers(buf) = \varnothing$

then

        act001: $queue\_of\_buffers(buf) := queue\_of\_buffers(buf) \setminus \{msg \mapsto t\}$

end

Event receive_buffer_needwakeupsendproc_init ⟨ordinary⟩ $\widehat{=}$

extends receive_buffer_needwakeupsendproc_init

any

        part
        proc
        newstate
        core
        buf

where

        grd001: $part \in PARTITIONS$
        grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state)$
        grd003: $newstate \in PROCESS\_STATES$
        grd004: $core \in CORES$
        grd005: $processes\_of\_partition(proc) = part$
        grd017: $finished\_core2(core) = TRUE$
        grd101: $partition\_mode(part) = PM\_NORMAL$
        grd102: $process\_state(proc) = PS\_Waiting \vee process\_state(proc) = PS\_WaitandSuspend$
        grd103: $process\_state(proc) = PS\_Waiting \Rightarrow newstate = PS\_Ready$
        grd104: $process\_state(proc) = PS\_WaitandSuspend \Rightarrow newstate = PS\_Suspend$
        grd201: $part = current\_partition$
        grd203: $processes\_of\_partition(proc) \in dom(current\_partition\_flag)$
        grd202: $current\_partition\_flag(part) = TRUE$
        grd301: $buf \in buffers$
        grd302: $queue\_of\_buffers(buf) \neq \varnothing$
        grd303: $processes\_waitingfor\_buffers(buf) \neq \varnothing$

then

        act001: $process\_state(proc) := newstate$
        act201: $location\_of\_service2(core) := Resource\_become\_avail \mapsto loc\_i$
        act202: $finished\_core2(core) := FALSE$
        act203: $resource\_become\_avail\_proc(core) := proc$
        act204: $timeout\_trigger := \{proc\} \lhd timeout\_trigger$
        act301: $location\_of\_service3(core) := Receive\_Buffer\_NeedWakeup \mapsto loc\_i$
        act302: $receive\_buffer\_needwake(core) := buf$

end

Event receive_buffer_needwakeupsendproc_timeout_trig ⟨ordinary⟩ $\widehat{=}$

extends receive_buffer_needwakeupsendproc_timeout_trig

any

        part
        proc
        core
        buf

where

        grd001: $part \in PARTITIONS$
        grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition) \wedge proc \in dom(process\_wait\_type)$
        grd003: $core \in CORES \cap dom(resource\_become\_avail\_proc) \wedge core \in dom(location\_of\_service2)$
        grd004: $proc = resource\_become\_avail\_proc(core)$
        grd005: $processes\_of\_partition(proc) = part$
        grd006: $partition\_mode(part) = PM\_NORMAL$

$\quad$ grd007: $\quad part = current\_partition$

$\quad$ grd013: $\quad processes\_of\_partition(proc) \in dom(current\_partition\_flag)$

$\quad$ grd008: $\quad current\_partition\_flag(part) = TRUE$

$\quad$ grd009: $\quad process\_wait\_type(proc) = PROC\_WAIT\_OBJ$

$\quad$ grd010: $\quad finished\_core2(core) = FALSE$

$\quad$ grd011: $\quad location\_of\_service2(core) = Resource\_become\_avail \mapsto loc\_i$

$\quad$ grd012: $\quad \neg(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Resource\_become\_avail \mapsto loc\_i)$

$\quad$ grd301: $\quad buf \in buffers$

$\quad$ grd305: $\quad buf = receive\_buffer\_needwake(core)$

$\quad$ grd302: $\quad queue\_of\_buffers(buf) \neq \varnothing$

$\quad$ grd303: $\quad processes\_waitingfor\_buffers(buf) \neq \varnothing$

$\quad$ grd304: $\quad location\_of\_service3(core) = Receive\_Buffer\_NeedWakeup \mapsto loc\_i$

$\quad$ grd306: $\quad \neg(finished\_core2(core) = FALSE \land location\_of\_service3(core) = Receive\_Buffer\_NeedWakeup \mapsto loc\_i)$

then

$\quad$ act001: $location\_of\_service2(core) := Resource\_become\_avail \mapsto loc\_1$

$\quad$ act002: $process\_wait\_type := \{proc\} \lhd process\_wait\_type$

$\quad$ act301: $location\_of\_service3(core) := Receive\_Buffer\_NeedWakeup \mapsto loc\_1$

end

Event receive_buffer_needwakeupsendproc_insert $\langle$ordinary$\rangle$ $\widehat{=}$

extends receive_buffer_needwakeupsendproc_insert

any

$\quad part$

$\quad proc$

$\quad core$

$\quad buf$

$\quad msg$

$\quad t$

$\quad m\_$

$\quad t\_$

where

$\quad$ grd001: $\quad part \in PARTITIONS \land part \in dom(current\_partition\_flag)$

$\quad$ grd002: $\quad proc \in processes \land proc \in dom(processes\_of\_partition)$

$\quad$ grd003: $\quad core \in CORES \cap dom(resource\_become\_avail\_proc) \cap dom(location\_of\_service3) \cap dom(receive\_buffer\_need$

$\quad$ grd004: $\quad proc = resource\_become\_avail\_proc(core)$

$\quad$ grd005: $\quad processes\_of\_partition(proc) = part$

$\quad$ grd006: $\quad partition\_mode(part) = PM\_NORMAL$

$\quad$ grd007: $\quad part = current\_partition$

$\quad$ grd008: $\quad current\_partition\_flag(part) = TRUE$

$\quad$ grd009: $\quad buf \in buffers$

$\quad$ grd010: $\quad buf = receive\_buffer\_needwake(core)$

$\quad$ grd011: $\quad msg \in MESSAGES \land m\_ \in MESSAGES \land t \in \mathbb{N} \land t\_ \in \mathbb{N}$

$\quad$ grd012: $\quad queue\_of\_buffers(buf) \neq \varnothing$

$\quad$ grd013: $\quad processes\_waitingfor\_buffers(buf) \neq \varnothing \land (proc \mapsto (m\_ \mapsto WAITING\_W \mapsto t\_)) \in processes\_waitingfor\_buffers(buf)$

$\quad$ grd014: $\quad (msg \mapsto t) \in queue\_of\_buffers(buf)$

$\quad$ grd015: $\quad finished\_core2(core) = FALSE$

$\quad$ grd016: $\quad location\_of\_service3(core) = Receive\_Buffer\_NeedWakeup \mapsto loc\_1$

$\quad$ grd017: $\quad \neg(finished\_core2(core) = FALSE \land location\_of\_service3(core) = Receive\_Buffer\_NeedWakeup \mapsto loc\_1)$

$\quad$ grd201: $\quad processes\_waitingfor\_buffers(buf) \neq \varnothing \land (proc \mapsto (msg \mapsto WAITING\_W \mapsto t\_)) \in processes\_waitingfor\_buffers(buf)$

$\quad$ grd202: $\quad quediscipline\_of\_buffers(buf) = QUEUE\_FIFO \Rightarrow (\forall p1, m1, t1 \cdot (p1 \mapsto (m1 \mapsto WAITING\_R \mapsto t1) \in processes\_waitingfor\_buffers(buf) \Rightarrow t \leq t1))$

$\quad$ grd203: $\quad quediscipline\_of\_buffers(buf) = QUEUE\_PRIORITY \Rightarrow (\forall p1, m1, t1 \cdot (p1 \mapsto (m1 \mapsto WAITING\_R \mapsto t1) \in processes\_waitingfor\_buffers(buf) \Rightarrow currentpriority\_of\_process(proc) \geq currentpriority\_of\_process(p1)))$

      then
          act001: $location\_of\_service3(core) := Receive\_Buffer\_NeedWakeup \mapsto loc\_2$
          act002: $queue\_of\_buffers(buf) := queue\_of\_buffers(buf) \setminus \{msg \mapsto t\}$
          act003: $processes\_waitingfor\_buffers(buf) := \{proc\} \lhd processes\_waitingfor\_buffers(buf)$
      end

Event receive_buffer_needwakeupsendproc_schedule $\langle ordinary \rangle \;\widehat{=}$
extends receive_buffer_needwakeupsendproc_schedule
      any
          part
          proc
          core
          resch
          buf
      where
          grd001: $part \in PARTITIONS$
          grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition)$
          grd003: $core \in CORES \cap dom(resource\_become\_avail\_proc) \wedge core \in dom(location\_of\_service2)$
          grd004: $proc = resource\_become\_avail\_proc(core)$
          grd005: $processes\_of\_partition(proc) = part$
          grd006: $partition\_mode(part) = PM\_NORMAL$
          grd007: $part = current\_partition$
          grd013: $processes\_of\_partition(proc) \in dom(current\_partition\_flag)$
          grd008: $current\_partition\_flag(part) = TRUE$
          grd009: $resch \in BOOL$
          grd010: $finished\_core2(core) = FALSE$
          grd011: $location\_of\_service2(core) = Resource\_become\_avail \mapsto loc\_1$
          grd012: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Resource\_become\_avail \mapsto loc\_1)$
          grd301: $buf \in buffers$
          grd302: $buf = receive\_buffer\_needwake(core)$
          grd304: $location\_of\_service3(core) = Receive\_Buffer\_NeedWakeup \mapsto loc\_2$
          grd305: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service3(core) = Receive\_Buffer\_NeedWakeup \mapsto loc\_2)$
      then
          act001: $location\_of\_service2(core) := Resource\_become\_avail \mapsto loc\_2$
          act002: $need\_reschedule := resch$
          act301: $location\_of\_service3(core) := Receive\_Buffer\_NeedWakeup \mapsto loc\_3$
      end

Event receive_buffer_needwakeupsendproc_return $\langle ordinary \rangle \;\widehat{=}$
extends receive_buffer_needwakeupsendproc_return
      any
          part
          proc
          core
          buf
      where
          grd001: $part \in PARTITIONS$
          grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition)$
          grd003: $core \in CORES \cap dom(resource\_become\_avail\_proc) \wedge core \in dom(location\_of\_service2)$
          grd004: $proc = resource\_become\_avail\_proc(core)$
          grd005: $processes\_of\_partition(proc) = part$
          grd006: $partition\_mode(part) = PM\_NORMAL$
          grd007: $part = current\_partition$
          grd012: $processes\_of\_partition(proc) \in dom(current\_partition\_flag)$
          grd008: $current\_partition\_flag(part) = TRUE$
          grd009: $finished\_core2(core) = FALSE$
          grd010: $location\_of\_service2(core) = Resource\_become\_avail \mapsto loc\_2$
          grd011: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Resource\_become\_avail \mapsto loc\_2)$

        grd301: $buf \in buffers$

        grd302: $buf = receive\_buffer\_needwake(core)$

        grd303: $location\_of\_service3(core) = Receive\_Buffer\_NeedWakeup \mapsto loc\_3$

        grd304: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service3(core) = Receive\_Buffer\_NeedWakeup \mapsto loc\_3)$

then

        act001: $location\_of\_service2(core) := Resource\_become\_avail \mapsto loc\_r$

        act002: $finished\_core2(core) := TRUE$

        act003: $resource\_become\_avail\_proc := \{core\} \lhd resource\_become\_avail\_proc$

        act301: $location\_of\_service3(core) := Receive\_Buffer\_NeedWakeup \mapsto loc\_r$

        act302: $receive\_buffer\_needwake := \{core\} \lhd receive\_buffer\_needwake$

end

Event receive_buffer_whenempty_init ⟨ordinary⟩ $\widehat{=}$

extends receive_buffer_whenempty_init

    any

        part

        proc

        newstate

        core

        buf

    where

        grd001: $part \in PARTITIONS$

        grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state) \cap dom(process\_wait\_type)$

        grd003: $newstate \in PROCESS\_STATES$

        grd004: $core \in CORES \wedge core \in dom(current\_processes\_flag)$

        grd005: $processes\_of\_partition(proc) = part$

        grd017: $finished\_core2(core) = TRUE$

        grd101: $partition\_mode(part) = PM\_NORMAL$

        grd102: $process\_state(proc) = PS\_Running$

        grd103: $newstate = PS\_Waiting$

        grd205: $proc \in dom(delaytime\_of\_process) \wedge proc \in dom(process\_wait\_type)$

        grd201: $part = current\_partition \wedge current\_partition \in dom(current\_partition\_flag)$

        grd202: $current\_partition\_flag(part) = TRUE$

        grd203: $current\_processes\_flag(core) = TRUE$

        grd204: $proc = current\_processes(core)$

        grd301: $buf \in buffers$

        grd302: $buffers\_of\_partition(buf) = part$

        grd303: $queue\_of\_buffers(buf) = \varnothing$

    then

        act001: $process\_state(proc) := newstate$

        act002: $location\_of\_service2(core) := Req\_busy\_resource \mapsto loc\_i$

        act003: $finished\_core2(core) := FALSE$

        act004: $req\_busy\_resource\_proc(core) := proc$

        act005: $current\_processes\_flag(core) := FALSE$

        act006: $current\_processes := \{core\} \lhd current\_processes$

        act301: $location\_of\_service3(core) := Receive\_Buffer\_Whenempty \mapsto loc\_i$

        act302: $receive\_buffer\_whenempty(core) := buf$

    end

Event receive_buffer_whenempty_timeout ⟨ordinary⟩ $\widehat{=}$

extends receive_buffer_whenempty_timeout

    any

        part

        proc

        core

        timeout

        tmout_trig

        wt

        buf

where

  grd001: $part \in PARTITIONS$

  grd002: $proc \in processes \land proc \in dom(processes\_of\_partition)$

  grd003: $core \in CORES \cap dom(req\_busy\_resource\_proc) \land core \in dom(current\_processes\_flag) \land$
    $core \in dom(location\_of\_service2)$

  grd004: $proc = req\_busy\_resource\_proc(core)$

  grd005: $processes\_of\_partition(proc) = part$

  grd006: $part = current\_partition$

  grd018: $processes\_of\_partition(req\_busy\_resource\_proc(core)) \in dom(current\_partition\_flag)$

  grd007: $current\_partition\_flag(part) = TRUE$

  grd008: $current\_processes\_flag(core) = TRUE$

  grd009: $timeout \geq 0$

  grd010: $wt \in PROCESS\_WAIT\_TYPES \land (wt = PROC\_WAIT\_OBJ \lor wt = PROC\_WAIT\_TIMEOUT)$

  grd011: $tmout\_trig \in processes \nrightarrow (PROCESS\_STATES \times \mathbb{N}_1)$

  grd012:
   $(timeout = INFINITE\_TIME\_VALUE \Rightarrow tmout\_trig = \varnothing)$
   $\land (timeout > 0 \Rightarrow tmout\_trig = \{proc \mapsto (PS\_Ready \mapsto (timeout + clock\_tick * ONE\_TICK\_TIME))\})$

  grd013: $timeout > 0 \Rightarrow wt = PROC\_WAIT\_TIMEOUT$

  grd014: $timeout = INFINITE\_TIME\_VALUE \Rightarrow wt = PROC\_WAIT\_OBJ$

  grd015: $finished\_core2(core) = FALSE$

  grd016: $location\_of\_service2(core) = Req\_busy\_resource \mapsto loc\_i$

  grd017: $\neg(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Req\_busy\_resource \mapsto$
   $loc\_i)$

  grd301: $buf \in buffers$

  grd304: $buf = receive\_buffer\_whenempty(core)$

  grd302: $buffers\_of\_partition(buf) = part$

  grd303: $queue\_of\_buffers(buf) = \varnothing$

  grd305: $location\_of\_service3(core) = Receive\_Buffer\_Whenempty \mapsto loc\_i$

  grd306: $\neg(finished\_core2(core) = FALSE \land location\_of\_service3(core) = Receive\_Buffer\_Whenempty \mapsto$
   $loc\_i)$

then

  act001: $location\_of\_service2(core) := Req\_busy\_resource \mapsto loc\_1$

  act002: $timeout\_trigger := timeout\_trigger \mathbin{\lhd\mkern-9mu-} tmout\_trig$

  act003: $process\_wait\_type(proc) := wt$

  act301: $location\_of\_service3(core) := Receive\_Buffer\_Whenempty \mapsto loc\_1$

end

Event receive_buffer_whenempty_wait ⟨ordinary⟩ $\widehat{=}$

extends receive_buffer_whenempty_wait

any

  part

  proc

  core

  buf

  msg

  t

where

  grd001: $part \in PARTITIONS$

  grd002: $proc \in processes \cap dom(processes\_of\_partition)$

  grd003: $core \in CORES \cap dom(req\_busy\_resource\_proc) \cap dom(location\_of\_service3)$

  grd004: $proc = req\_busy\_resource\_proc(core)$

  grd005: $processes\_of\_partition(proc) = part$

  grd006: $part = current\_partition$

  grd007: $buf \in buffers$

  grd008: $buffers\_of\_partition(buf) = part$

  grd009: $queue\_of\_buffers(buf) = \varnothing$

  grd010: $msg \in MESSAGES$

  grd011: $t \in \mathbb{N}$

          grd012: $finished\_core2(core) = FALSE$

          grd013: $location\_of\_service3(core) = Receive\_Buffer\_Whenempty \mapsto loc\_1$

          grd14: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service3(core) = Receive\_Buffer\_Whenempty \mapsto loc\_1)$

          grd201: $t = clock\_tick * ONE\_TICK\_TIME$

      then

          act001: $location\_of\_service3(core) := Receive\_Buffer\_Whenempty \mapsto loc\_2$

          act002: $processes\_waitingfor\_buffers(buf) := processes\_waitingfor\_buffers(buf) \vartriangleleft \{proc \mapsto (msg \mapsto WAITING\_R \mapsto t)\}$

      end

Event receive_buffer_whenempty_schedule ⟨ordinary⟩ $\widehat{=}$

extends receive_buffer_whenempty_schedule

      any

          part

          proc

          core

          buf

      where

          grd001: $part \in PARTITIONS$

          grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition)$

          grd003: $core \in CORES \cap dom(req\_busy\_resource\_proc) \wedge core \in dom(current\_processes\_flag) \wedge core \in dom(location\_of\_service2)$

          grd004: $proc = req\_busy\_resource\_proc(core)$

          grd005: $processes\_of\_partition(proc) = part$

          grd006: $part = current\_partition$

          grd012: $processes\_of\_partition(req\_busy\_resource\_proc(core)) \in dom(current\_partition\_flag)$

          grd007: $current\_partition\_flag(part) = TRUE$

          grd008: $current\_processes\_flag(core) = FALSE$

          grd009: $finished\_core2(core) = FALSE$

          grd010: $location\_of\_service2(core) = Req\_busy\_resource \mapsto loc\_1$

          grd011: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Req\_busy\_resource \mapsto loc\_1)$

          grd301: $buf \in buffers$

          grd306: $buf = receive\_buffer\_whenempty(core)$

          grd302: $buffers\_of\_partition(buf) = part$

          grd303: $queue\_of\_buffers(buf) = \varnothing$

          grd304: $location\_of\_service3(core) = Receive\_Buffer\_Whenempty \mapsto loc\_2$

          grd305: $\neg(finished\_core(core) = FALSE \wedge location\_of\_service3(core) = Receive\_Buffer\_Whenempty \mapsto loc\_2)$

      then

          act001: $location\_of\_service2(core) := Req\_busy\_resource \mapsto loc\_2$

          act002: $need\_reschedule := TRUE$

          act301: $location\_of\_service3(core) := Receive\_Buffer\_Whenempty \mapsto loc\_3$

      end

Event receive_buffer_whenempty_return ⟨ordinary⟩ $\widehat{=}$

extends receive_buffer_whenempty_return

      any

          part

          proc

          core

          buf

      where

          grd001: $part \in PARTITIONS$

          grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition)$

          grd003: $core \in CORES \cap dom(req\_busy\_resource\_proc) \wedge core \in dom(current\_processes\_flag) \wedge core \in dom(location\_of\_service2)$

          grd004: $proc = req\_busy\_resource\_proc(core)$

          grd005: $processes\_of\_partition(proc) = part$

          grd006: $part = current\_partition$

grd012: $processes\_of\_partition(req\_busy\_resource\_proc(core)) \in dom(current\_partition\_flag)$
grd007: $current\_partition\_flag(part) = TRUE$
grd008: $current\_processes\_flag(core) = FALSE$
grd009: $finished\_core2(core) = FALSE$
grd010: $location\_of\_service2(core) = Req\_busy\_resource \mapsto loc\_2$
grd011: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Req\_busy\_resource \mapsto$
           $loc\_2)$
grd301: $buf \in buffers$
grd302: $buf = receive\_buffer\_whenempty(core)$
grd303: $buffers\_of\_partition(buf) = part$
grd304: $queue\_of\_buffers(buf) = \varnothing$
grd305: $location\_of\_service3(core) = Receive\_Buffer\_Whenempty \mapsto loc\_3$
grd306: $\neg(finished\_core(core) = FALSE \wedge location\_of\_service3(core) = Receive\_Buffer\_Whenempty \mapsto$
           $loc\_3)$

then

act001: $location\_of\_service2(core) := Req\_busy\_resource \mapsto loc\_r$
act002: $finished\_core2(core) := TRUE$
act003: $req\_busy\_resource\_proc := \{core\} \lhd req\_busy\_resource\_proc$
act301: $location\_of\_service3(core) := Receive\_Buffer\_Whenempty \mapsto loc\_r$
act302: $receive\_buffer\_whenempty := \{core\} \lhd receive\_buffer\_whenempty$

end

Event get_buffer_id ⟨ordinary⟩ $\widehat{=}$

any

     part
     core
     buf

where

grd001: $part \in dom(current\_partition\_flag) \wedge current\_partition = part \wedge current\_partition\_flag(part) =$
           $TRUE$
grd002: $buf \in buffers$
grd004: $buffers\_of\_partition(buf) = part$
grd006: $core \in CORES$
grd005: $finished\_core2(core) = TRUE$

then

     skip

end

Event get_buffer_status ⟨ordinary⟩ $\widehat{=}$

any

     part
     core
     buf

where

grd001: $part \in dom(current\_partition\_flag) \wedge current\_partition = part \wedge current\_partition\_flag(part) =$
           $TRUE$
grd002: $buf \in buffers$
grd004: $buffers\_of\_partition(buf) = part$
grd005: $core \in CORES$
grd006: $finished\_core2(core) = TRUE$

then

     skip

end

Event create_blackboard ⟨ordinary⟩ $\widehat{=}$

refines create_blackboard

any

     core
     bb
     part

where

grd001: $core \in CORES$

   grd002: $bb \in BLACKBOARDS \wedge bb \notin blackboards$

   grd003: $finished\_core(core) = TRUE$

   grd004: $part \in PARTITIONS$

   grd201: $part \in dom(current\_partition\_flag) \wedge current\_partition = part \wedge current\_partition\_flag(part) = TRUE$

   grd202: $(partition\_mode(current\_partition) = PM\_COLD\_START \vee partition\_mode(current\_partition) = PM\_WARM\_START)$

  then

   act001: $blackboards := blackboards \cup \{bb\}$

   act002: $emptyindicator\_of\_blackboards(bb) := BB\_EMPTY$

   act003: $blackboards\_of\_partition(bb) := part$

   act004: $processes\_waitingfor\_blackboards(bb) := \varnothing$

  end

Event display_blackboard ⟨ordinary⟩ $\widehat{=}$

extends display_blackboard

  any

   core

   bb

   msg

   part

  where

   grd001: $core \in CORES$

   grd002: $bb \in blackboards$

   grd003: $msg \in MESSAGES \wedge msg \notin used\_messages$

   grd004: $processes\_waitingfor\_blackboards(bb) = \varnothing$

   grd005: $finished\_core(core) = TRUE$

   grd201: $part \in dom(current\_partition\_flag) \wedge current\_partition = part \wedge current\_partition\_flag(part) = TRUE$

   grd203: $current\_processes\_flag(core) = TRUE$

   grd204: $blackboards\_of\_partition(bb) = part$

  then

   act001: $msgspace\_of\_blackboards(bb) := msg$

   act002: $used\_messages := used\_messages \cup \{msg\}$

   act003: $emptyindicator\_of\_blackboards(bb) := BB\_OCCUPIED$

  end

Event display_blackboard_needwakeuprdprocs_init ⟨ordinary⟩ $\widehat{=}$

extends display_blackboard_needwakeuprdprocs_init

  any

   part

   procs

   newstates

   core

   bb

  where

   grd001: $part \in PARTITIONS$

   grd002: $procs \subseteq processes \cap dom(process\_state)$

   grd003: $newstates \in procs \rightarrow PROCESS\_STATES$

   grd004: $core \in CORES$

   grd005: $procs \subseteq processes\_of\_partition^{-1}[\{part\}]$

   grd101: $partition\_mode(part) = PM\_NORMAL$

   grd102: $\forall proc \cdot (proc \in procs \Rightarrow process\_state(proc) = PS\_Waiting \vee process\_state(proc) = PS\_WaitandSuspend)$

   grd103: $\forall proc \cdot (proc \in procs \wedge process\_state(proc) = PS\_Waiting \Rightarrow newstates(proc) = PS\_Ready)$

   grd104: $\forall proc \cdot (proc \in procs \wedge process\_state(proc) = PS\_WaitandSuspend \Rightarrow newstates(proc) = PS\_Suspend)$

   grd301: $part = current\_partition$

   grd303: $part \in dom(current\_partition\_flag)$

   grd302: $current\_partition\_flag(part) = TRUE$

        grd304: $finished\_core2(core) = TRUE$
        grd401: $bb \in blackboards$
        grd402: $blackboards\_of\_partition(bb) = part$
        grd403: $processes\_waitingfor\_blackboards(bb) \neq \varnothing$
        grd404: $procs = processes\_waitingfor\_blackboards(bb)$

then

        act001: $process\_state := process\_state \Leftarrow newstates$
        act301: $location\_of\_service2(core) := Resource\_become\_avail2 \mapsto loc\_i$
        act302: $finished\_core2(core) := FALSE$
        act303: $resource\_become\_avail2(core) := procs$
        act304: $timeout\_trigger := procs \lhd timeout\_trigger$
        act401: $location\_of\_service3(core) := Display\_Blackboard\_NeedWakeup \mapsto loc\_i$
        act402: $display\_blackboard\_needwake(core) := bb$

end

Event display_blackboard_needwakeuprdprocs_timeout_trig $\langle ordinary \rangle \ \widehat{=}$

extends display_blackboard_needwakeuprdprocs_timeout_trig

    any

        part
        procs
        core
        bb

    where

        grd001: $part \in PARTITIONS$
        grd002: $procs \subseteq (processes \cap dom(process\_state))$
        grd003: $core \in CORES \wedge core \in dom(location\_of\_service2) \wedge core \in dom(resource\_become\_avail2)$

        grd004: $procs = resource\_become\_avail2(core)$
        grd005: $part = current\_partition$
        grd006: $partition\_mode(part) = PM\_NORMAL$
        grd007: $\forall proc \cdot (proc \in procs \wedge proc \in dom(process\_wait\_type) \Rightarrow process\_wait\_type(proc) = PROC\_WAIT\_OBJ)$
        grd008: $finished\_core2(core) = FALSE$
        grd009: $location\_of\_service2(core) = Resource\_become\_avail2 \mapsto loc\_i$
        grd010: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Resource\_become\_avail2 \mapsto loc\_i)$
        grd301: $bb \in blackboards$
        grd302: $core \in dom(display\_blackboard\_needwake)$
        grd303: $bb = display\_blackboard\_needwake(core)$
        grd304: $blackboards\_of\_partition(bb) = part$
        grd305: $processes\_waitingfor\_blackboards(bb) \neq \varnothing$
        grd306: $procs = processes\_waitingfor\_blackboards(bb)$
        grd307: $location\_of\_service3(core) = Display\_Blackboard\_NeedWakeup \mapsto loc\_i$
        grd308: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service3(core) = Display\_Blackboard\_NeedWakeup \mapsto loc\_i)$

    then

        act001: $location\_of\_service2(core) := Resource\_become\_avail2 \mapsto loc\_1$
        act002: $process\_wait\_type := procs \lhd process\_wait\_type$
        act301: $location\_of\_service3(core) := Display\_Blackboard\_NeedWakeup \mapsto loc\_1$
        act302: $emptyindicator\_of\_blackboards(bb) := BB\_OCCUPIED$

    end

Event display_blackboard_needwakeuprdprocs_insert $\langle ordinary \rangle \ \widehat{=}$

extends display_blackboard_needwakeuprdprocs_insert

    any

        part
        procs
        core
        bb
        msg

    where

grd001: $part \in PARTITIONS$

grd002: $procs \subseteq (processes \cap dom(process\_state))$

grd003: $core \in CORES \wedge core \in dom(location\_of\_service3) \wedge core \in dom(display\_blackboard\_needwake) \cap$
$dom(resource\_become\_avail2)$

grd004: $procs = resource\_become\_avail2(core)$

grd005: $part = current\_partition$

grd006: $partition\_mode(part) = PM\_NORMAL$

grd007: $bb \in blackboards$

grd008: $bb = display\_blackboard\_needwake(core)$

grd009: $blackboards\_of\_partition(bb) = part$

grd010: $msg \in MESSAGES \wedge msg \notin used\_messages$

grd011: $processes\_waitingfor\_blackboards(bb) \neq \varnothing$

grd012: $procs = processes\_waitingfor\_blackboards(bb)$

grd013: $finished\_core2(core) = FALSE$

grd014: $location\_of\_service3(core) = Display\_Blackboard\_NeedWakeup \mapsto loc\_1$

grd015: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service3(core) = Display\_Blackboard\_NeedWakeup \mapsto loc\_1)$

grd201: $processes\_waitingfor\_blackboards(bb) \neq \varnothing$

grd202: $current\_partition\_flag(part) = TRUE$

grd203: $current\_processes\_flag(core) = TRUE$

grd204: $part \in dom(current\_partition\_flag)$

then

act001: $location\_of\_service3(core) := Display\_Blackboard\_NeedWakeup \mapsto loc\_2$

act002: $msgspace\_of\_blackboards(bb) := msg$

act003: $processes\_waitingfor\_blackboards(bb) := processes\_waitingfor\_blackboards(bb) \setminus procs$

act004: $used\_messages := used\_messages \cup \{msg\}$

end

Event display_blackboard_needwakeuprdprocs_schedule ⟨ordinary⟩ $\widehat{=}$

extends display_blackboard_needwakeuprdprocs_schedule

any

    part

    procs

    core

    resch

    bb

where

grd001: $part \in PARTITIONS$

grd002: $procs \subseteq (processes \cap dom(process\_state))$

grd003: $core \in CORES \wedge core \in dom(location\_of\_service2) \wedge core \in dom(resource\_become\_avail2)$

grd004: $procs = resource\_become\_avail2(core)$

grd005: $part = current\_partition$

grd006: $partition\_mode(part) = PM\_NORMAL$

grd008: $resch \in BOOL$

grd009: $finished\_core2(core) = FALSE$

grd010: $location\_of\_service2(core) = Resource\_become\_avail2 \mapsto loc\_1$

grd011: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Resource\_become\_avail2 \mapsto loc\_1)$

grd301: $bb \in blackboards$

grd302: $core \in dom(display\_blackboard\_needwake)$

grd303: $bb = display\_blackboard\_needwake(core)$

grd304: $blackboards\_of\_partition(bb) = part$

grd305: $location\_of\_service3(core) = Display\_Blackboard\_NeedWakeup \mapsto loc\_2$

grd306: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service3(core) = Display\_Blackboard\_NeedWakeup \mapsto loc\_2)$

then

act001: $location\_of\_service2(core) := Resource\_become\_avail2 \mapsto loc\_2$

act002: $need\_reschedule := resch$

act301: $location\_of\_service3(core) := Display\_Blackboard\_NeedWakeup \mapsto loc\_3$

end

Event display_blackboard_needwakeuprdprocs_return ⟨ordinary⟩ ≙

extends display_blackboard_needwakeuprdprocs_return

    any

        part

        procs

        core

        bb

    where

        grd001: $part \in PARTITIONS$

        grd002: $procs \subseteq (processes \cap dom(process\_state))$

        grd003: $core \in CORES \land core \in dom(location\_of\_service2) \land core \in dom(resource\_become\_avail2)$

        grd004: $procs = resource\_become\_avail2(core)$

        grd005: $part = current\_partition$

        grd006: $partition\_mode(part) = PM\_NORMAL$

        grd007: $finished\_core2(core) = FALSE$

        grd008: $location\_of\_service2(core) = Resource\_become\_avail2 \mapsto loc\_2$

        grd009: $\neg(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Resource\_become\_avail2 \mapsto loc\_2)$

        grd301: $bb \in blackboards$

        grd302: $core \in dom(display\_blackboard\_needwake)$

        grd303: $bb = display\_blackboard\_needwake(core)$

        grd304: $blackboards\_of\_partition(bb) = part$

        grd305: $location\_of\_service3(core) = Display\_Blackboard\_NeedWakeup \mapsto loc\_3$

        grd306: $\neg(finished\_core2(core) = FALSE \land location\_of\_service3(core) = Display\_Blackboard\_NeedWakeup \mapsto loc\_3)$

    then

        act001: $location\_of\_service2(core) := Resource\_become\_avail2 \mapsto loc\_r$

        act002: $finished\_core2(core) := TRUE$

        act003: $resource\_become\_avail2 := \{core\} \lessdot resource\_become\_avail2$

        act301: $location\_of\_service3(core) := Display\_Blackboard\_NeedWakeup \mapsto loc\_r$

        act302: $display\_blackboard\_needwake := \{core\} \lessdot display\_blackboard\_needwake$

    end

Event read_blackboard ⟨ordinary⟩ ≙

extends read_blackboard

    any

        core

        bb

        msg

        part

    where

        grd001: $core \in CORES$

        grd002: $bb \in blackboards$

        grd003: $msg \in MESSAGES$

        grd004: $emptyindicator\_of\_blackboards(bb) = BB\_OCCUPIED$

        grd201: $part \in dom(current\_partition\_flag) \land current\_partition = part \land current\_partition\_flag(part) = TRUE$

        grd203: $current\_processes\_flag(core) = TRUE$

        grd204: $blackboards\_of\_partition(bb) = part$

    then

        skip

    end

Event read_blackboard_whenempty_init ⟨ordinary⟩ ≙

extends read_blackboard_whenempty_init

    any

        part

        proc

   newstate
   core
   bb
  where
   grd001: $part \in PARTITIONS$
   grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state) \cap dom(process\_wait\_type)$

   grd003: $newstate \in PROCESS\_STATES$
   grd004: $core \in CORES \wedge core \in dom(current\_processes\_flag)$
   grd005: $processes\_of\_partition(proc) = part$
   grd017: $finished\_core2(core) = TRUE$
   grd101: $partition\_mode(part) = PM\_NORMAL$
   grd102: $process\_state(proc) = PS\_Running$
   grd103: $newstate = PS\_Waiting$
   grd205: $proc \in dom(delaytime\_of\_process) \wedge proc \in dom(process\_wait\_type)$
   grd201: $part = current\_partition \wedge current\_partition \in dom(current\_partition\_flag)$
   grd202: $current\_partition\_flag(part) = TRUE$
   grd203: $current\_processes\_flag(core) = TRUE$
   grd204: $proc = current\_processes(core)$
   grd301: $bb \in blackboards$
   grd302: $blackboards\_of\_partition(bb) = part$
   grd303: $emptyindicator\_of\_blackboards(bb) = BB\_EMPTY$
  then
   act001: $process\_state(proc) := newstate$
   act002: $location\_of\_service2(core) := Req\_busy\_resource \mapsto loc\_i$
   act003: $finished\_core2(core) := FALSE$
   act004: $req\_busy\_resource\_proc(core) := proc$
   act005: $current\_processes\_flag(core) := FALSE$
   act006: $current\_processes := \{core\} \vartriangleleft current\_processes$
   act301: $location\_of\_service3(core) := Read\_Blackboard\_Whenempty \mapsto loc\_i$
   act302: $read\_blackboard\_whenempty(core) := bb$
  end
Event read_blackboard_whenempty_timeout ⟨ordinary⟩ $\widehat{=}$
extends read_blackboard_whenempty_timeout
  any
   part
   proc
   core
   timeout
   tmout_trig
   wt
   bb
  where
   grd001: $part \in PARTITIONS$
   grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition)$
   grd003: $core \in CORES \cap dom(req\_busy\_resource\_proc) \wedge core \in dom(current\_processes\_flag) \wedge$
    $core \in dom(location\_of\_service2)$
   grd004: $proc = req\_busy\_resource\_proc(core)$
   grd005: $processes\_of\_partition(proc) = part$
   grd006: $part = current\_partition$
   grd018: $processes\_of\_partition(req\_busy\_resource\_proc(core)) \in dom(current\_partition\_flag)$
   grd007: $current\_partition\_flag(part) = TRUE$
   grd008: $current\_processes\_flag(core) = TRUE$
   grd009: $timeout \geq 0$
   grd010: $wt \in PROCESS\_WAIT\_TYPES \wedge (wt = PROC\_WAIT\_OBJ \vee wt = PROC\_WAIT\_TIMEOUT)$

   grd011: $tmout\_trig \in processes \nrightarrow (PROCESS\_STATES \times \mathbb{N}_1)$
   grd012:
    $(timeout = INFINITE\_TIME\_VALUE \Rightarrow tmout\_trig = \varnothing)$

$$\wedge(timeout > 0 \Rightarrow tmout\_trig = \{proc \mapsto (PS\_Ready \mapsto (timeout+clock\_tick*ONE\_TICK\_TIME))\})$$

    grd013: $timeout > 0 \Rightarrow wt = PROC\_WAIT\_TIMEOUT$

    grd014: $timeout = INFINITE\_TIME\_VALUE \Rightarrow wt = PROC\_WAIT\_OBJ$

    grd015: $finished\_core2(core) = FALSE$

    grd016: $location\_of\_service2(core) = Req\_busy\_resource \mapsto loc\_i$

    grd017: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Req\_busy\_resource \mapsto loc\_i)$

    grd301: $bb \in blackboards$

    grd302: $core \in dom(read\_blackboard\_whenempty)$

    grd303: $bb = read\_blackboard\_whenempty(core)$

    grd304: $blackboards\_of\_partition(bb) = part$

    grd305: $emptyindicator\_of\_blackboards(bb) = BB\_EMPTY$

    grd306: $location\_of\_service3(core) = Read\_Blackboard\_Whenempty \mapsto loc\_i$

    grd307: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service3(core) = Read\_Blackboard\_Whenempty \mapsto loc\_i)$

then

    act001: $location\_of\_service2(core) := Req\_busy\_resource \mapsto loc\_1$

    act002: $timeout\_trigger := timeout\_trigger \Leftarrow tmout\_trig$

    act003: $process\_wait\_type(proc) := wt$

    act301: $location\_of\_service3(core) := Read\_Blackboard\_Whenempty \mapsto loc\_1$

end

Event read_blackboard_whenempty_wait $\langle ordinary \rangle \;\widehat{=}$

extends read_blackboard_whenempty_wait

    any

        part

        proc

        core

        bb

    where

        grd001: $part \in PARTITIONS$

        grd002: $proc \in processes \cap dom(processes\_of\_partition)$

        grd003: $processes\_of\_partition(proc) = part$

        grd004: $partition\_mode(part) = PM\_NORMAL$

        grd005: $core \in CORES \cap dom(req\_busy\_resource\_proc) \cap dom(location\_of\_service3)$

        grd006: $proc = req\_busy\_resource\_proc(core)$

        grd007: $part = current\_partition$

        grd008: $part \in dom(current\_partition\_flag)$

        grd009: $current\_partition\_flag(part) = TRUE$

        grd010: $current\_processes\_flag(core) = TRUE$

        grd011: $bb \in blackboards$

        grd012: $core \in dom(read\_blackboard\_whenempty)$

        grd013: $bb = read\_blackboard\_whenempty(core)$

        grd014: $blackboards\_of\_partition(bb) = part$

        grd015: $emptyindicator\_of\_blackboards(bb) = BB\_EMPTY$

        grd016: $finished\_core2(core) = FALSE$

        grd017: $location\_of\_service3(core) = Read\_Blackboard\_Whenempty \mapsto loc\_1$

        grd018: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service3(core) = Read\_Blackboard\_Whenempty \mapsto loc\_1)$

        grd201: $locklevel\_of\_partition(part) = 0$

    then

        act001: $location\_of\_service3(core) := Read\_Blackboard\_Whenempty \mapsto loc\_2$

        act002: $processes\_waitingfor\_blackboards(bb) := processes\_waitingfor\_blackboards(bb) \cup \{proc\}$

    end

Event read_blackboard_whenempty_schedule $\langle ordinary \rangle \;\widehat{=}$

extends read_blackboard_whenempty_schedule

    any

        part

        proc

   core
   bb

where

  grd001: $part \in PARTITIONS$

  grd002: $proc \in processes \land proc \in dom(processes\_of\_partition)$

  grd003: $core \in CORES \cap dom(req\_busy\_resource\_proc) \land core \in dom(current\_processes\_flag) \land$
    $core \in dom(location\_of\_service2)$

  grd004: $proc = req\_busy\_resource\_proc(core)$

  grd005: $processes\_of\_partition(proc) = part$

  grd006: $part = current\_partition$

  grd012: $processes\_of\_partition(req\_busy\_resource\_proc(core)) \in dom(current\_partition\_flag)$

  grd007: $current\_partition\_flag(part) = TRUE$

  grd008: $current\_processes\_flag(core) = FALSE$

  grd009: $finished\_core2(core) = FALSE$

  grd010: $location\_of\_service2(core) = Req\_busy\_resource \mapsto loc\_1$

  grd011: $\neg(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Req\_busy\_resource \mapsto$
    $loc\_1)$

  grd301: $bb \in blackboards$

  grd302: $core \in dom(read\_blackboard\_whenempty)$

  grd303: $bb = read\_blackboard\_whenempty(core)$

  grd304: $blackboards\_of\_partition(bb) = part$

  grd305: $emptyindicator\_of\_blackboards(bb) = BB\_EMPTY$

  grd306: $location\_of\_service3(core) = Read\_Blackboard\_Whenempty \mapsto loc\_2$

  grd307: $\neg(finished\_core2(core) = FALSE \land location\_of\_service3(core) = Read\_Blackboard\_Whenempty \mapsto$
    $loc\_2)$

then

  act001: $location\_of\_service2(core) := Req\_busy\_resource \mapsto loc\_2$

  act002: $need\_reschedule := TRUE$

  act301: $location\_of\_service3(core) := Read\_Blackboard\_Whenempty \mapsto loc\_3$

end

Event read_blackboard_whenempty_return $\langle ordinary \rangle \;\widehat{=}$

extends read_blackboard_whenempty_return

 any

   part
   proc
   core
   bb

 where

  grd001: $part \in PARTITIONS$

  grd002: $proc \in processes \land proc \in dom(processes\_of\_partition)$

  grd003: $core \in CORES \cap dom(req\_busy\_resource\_proc) \land core \in dom(current\_processes\_flag) \land$
    $core \in dom(location\_of\_service2)$

  grd004: $proc = req\_busy\_resource\_proc(core)$

  grd005: $processes\_of\_partition(proc) = part$

  grd006: $part = current\_partition$

  grd012: $processes\_of\_partition(req\_busy\_resource\_proc(core)) \in dom(current\_partition\_flag)$

  grd007: $current\_partition\_flag(part) = TRUE$

  grd008: $current\_processes\_flag(core) = FALSE$

  grd009: $finished\_core2(core) = FALSE$

  grd010: $location\_of\_service2(core) = Req\_busy\_resource \mapsto loc\_2$

  grd011: $\neg(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Req\_busy\_resource \mapsto$
    $loc\_2)$

  grd301: $bb \in blackboards$

  grd302: $core \in dom(read\_blackboard\_whenempty)$

  grd303: $bb = read\_blackboard\_whenempty(core)$

  grd304: $blackboards\_of\_partition(bb) = part$

  grd305: $emptyindicator\_of\_blackboards(bb) = BB\_EMPTY$

  grd306: $location\_of\_service3(core) = Read\_Blackboard\_Whenempty \mapsto loc\_3$

  grd307: $\neg(finished\_core2(core) = FALSE \land location\_of\_service3(core) = Read\_Blackboard\_Whenempty \mapsto$
    $loc\_3)$

then

        act001: $location\_of\_service2(core) := Req\_busy\_resource \mapsto loc\_r$

        act002: $finished\_core2(core) := TRUE$

        act003: $req\_busy\_resource\_proc := \{core\} \vartriangleleft req\_busy\_resource\_proc$

        act301: $location\_of\_service3(core) := Read\_Blackboard\_Whenempty \mapsto loc\_r$

        act302: $read\_blackboard\_whenempty := \{core\} \vartriangleleft read\_blackboard\_whenempty$

end

Event clear_blackboard ⟨ordinary⟩ $\widehat{=}$

extends clear_blackboard

    any

        core

        bb

        part

    where

        grd001: $core \in CORES$

        grd002: $bb \in blackboards$

        grd201: $part = current\_partition$

        grd202: $part \in dom(current\_partition\_flag)$

        grd203: $current\_partition\_flag(part) = TRUE$

        grd204: $current\_processes\_flag(core) = TRUE$

        grd205: $part \in dom(current\_partition\_flag)$

    then

        act001: $emptyindicator\_of\_blackboards(bb) := BB\_EMPTY$

        act002: $msgspace\_of\_blackboards := \{bb\} \vartriangleleft msgspace\_of\_blackboards$

    end

Event get_blackboard_id ⟨ordinary⟩ $\widehat{=}$

    any

        part

        core

        bb

    where

        grd001: $part \in dom(current\_partition\_flag) \wedge current\_partition = part \wedge current\_partition\_flag(part) = TRUE$

        grd002: $bb \in blackboards$

        grd003: $blackboards\_of\_partition(bb) = part$

        grd004: $core \in CORES$

        grd005: $finished\_core2(core) = TRUE$

    then

        skip

    end

Event get_blackboard_status ⟨ordinary⟩ $\widehat{=}$

    any

        part

        core

        bb

    where

        grd001: $part \in dom(current\_partition\_flag) \wedge current\_partition = part \wedge current\_partition\_flag(part) = TRUE$

        grd002: $bb \in blackboards$

        grd003: $blackboards\_of\_partition(bb) = part$

        grd004: $core \in CORES$

        grd005: $finished\_core2(core) = TRUE$

    then

        skip

    end

Event create_semaphore ⟨ordinary⟩ $\widehat{=}$

refines create_semaphore

    any

    part
    core
    sem
    maxval
    currentval
    disc

where

    grd001: $core \in CORES$

    grd002: $sem \in SEMAPHORES \land sem \notin semaphores$

    grd003: $maxval \in \mathbb{N}_1$

    grd004: $currentval \in \mathbb{N}$

    grd008: $currentval \leq maxval$

    grd005: $part \in PARTITIONS$

    grd007: $finished\_core2(core) = TRUE$

    grd201: $part \in dom(current\_partition\_flag) \land current\_partition = part \land current\_partition\_flag(part) = TRUE$

    grd202: $(partition\_mode(current\_partition) = PM\_COLD\_START \lor partition\_mode(current\_partition) = PM\_WARM\_START)$

    grd203: $disc \in QUEUING\_DISCIPLINE$

then

    act001: $semaphores := semaphores \cup \{sem\}$

    act002: $value\_of\_semaphores(sem) := currentval$

    act003: $MaxValue\_of\_Semaphores(sem) := maxval$

    act004: $semaphores\_of\_partition(sem) := part$

    act005: $processes\_waiting for\_semaphores(sem) := \varnothing$

    act201: $quediscipline\_of\_semaphores(sem) := disc$

end

Event wait_semaphore ⟨ordinary⟩ $\hat{=}$

extends wait_semaphore

any

    core
    sem
    part

where

    grd001: $core \in CORES$

    grd002: $sem \in semaphores$

    grd003: $value\_of\_semaphores(sem) > 0$

    grd201: $part \in dom(current\_partition\_flag) \land current\_partition = part \land current\_partition\_flag(part) = TRUE$

    grd203: $current\_processes\_flag(core) = TRUE$

    grd204: $semaphores\_of\_partition(sem) = part$

then

    act001: $value\_of\_semaphores(sem) := value\_of\_semaphores(sem) - 1$

end

Event wait_semaphore_whenzero_init ⟨ordinary⟩ $\hat{=}$

extends wait_semaphore_whenzero_init

any

    part
    proc
    newstate
    core
    sem

where

    grd001: $part \in PARTITIONS$

    grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state) \cap dom(process\_wait\_type)$

    grd003: $newstate \in PROCESS\_STATES$

    grd004: $core \in CORES \land core \in dom(current\_processes\_flag)$

    grd005: $processes\_of\_partition(proc) = part$

$\quad$ grd017: $\quad finished\_core2(core) = TRUE$

$\quad$ grd101: $\quad partition\_mode(part) = PM\_NORMAL$

$\quad$ grd102: $\quad process\_state(proc) = PS\_Running$

$\quad$ grd103: $\quad newstate = PS\_Waiting$

$\quad$ grd205: $\quad proc \in dom(delaytime\_of\_process) \wedge proc \in dom(process\_wait\_type)$

$\quad$ grd201: $\quad part = current\_partition \wedge current\_partition \in dom(current\_partition\_flag)$

$\quad$ grd202: $\quad current\_partition\_flag(part) = TRUE$

$\quad$ grd203: $\quad current\_processes\_flag(core) = TRUE$

$\quad$ grd204: $\quad proc = current\_processes(core)$

$\quad$ grd301: $\quad sem \in semaphores$

$\quad$ grd302: $\quad semaphores\_of\_partition(sem) = part$

$\quad$ grd303: $\quad value\_of\_semaphores(sem) = 0$

then

$\quad$ act001: $process\_state(proc) := newstate$

$\quad$ act002: $location\_of\_service2(core) := Req\_busy\_resource \mapsto loc\_i$

$\quad$ act003: $finished\_core2(core) := FALSE$

$\quad$ act004: $req\_busy\_resource\_proc(core) := proc$

$\quad$ act005: $current\_processes\_flag(core) := FALSE$

$\quad$ act006: $current\_processes := \{core\} \lhd current\_processes$

$\quad$ act301: $location\_of\_service3(core) := Wait\_Semaphore\_Whenzero \mapsto loc\_i$

$\quad$ act302: $wait\_semaphore\_whenzero(core) := sem$

end

Event wait_semaphore_whenzero_timeout $\langle ordinary \rangle \;\widehat{=}$

extends wait_semaphore_whenzero_timeout

$\quad$ **any**

$\qquad part$

$\qquad proc$

$\qquad core$

$\qquad timeout$

$\qquad tmout\_trig$

$\qquad wt$

$\qquad sem$

$\quad$ **where**

$\quad$ grd001: $\quad part \in PARTITIONS$

$\quad$ grd002: $\quad proc \in processes \wedge proc \in dom(processes\_of\_partition)$

$\quad$ grd003: $\quad core \in CORES \cap dom(req\_busy\_resource\_proc) \wedge core \in dom(current\_processes\_flag) \wedge$
$\qquad core \in dom(location\_of\_service2)$

$\quad$ grd004: $\quad proc = req\_busy\_resource\_proc(core)$

$\quad$ grd005: $\quad processes\_of\_partition(proc) = part$

$\quad$ grd006: $\quad part = current\_partition$

$\quad$ grd018: $\quad processes\_of\_partition(req\_busy\_resource\_proc(core)) \in dom(current\_partition\_flag)$

$\quad$ grd007: $\quad current\_partition\_flag(part) = TRUE$

$\quad$ grd008: $\quad current\_processes\_flag(core) = TRUE$

$\quad$ grd009: $\quad timeout \geq 0$

$\quad$ grd010: $\quad wt \in PROCESS\_WAIT\_TYPES \wedge (wt = PROC\_WAIT\_OBJ \vee wt = PROC\_WAIT\_TIMEOUT)$

$\quad$ grd011: $\quad tmout\_trig \in processes \nrightarrow (PROCESS\_STATES \times \mathbb{N}_1)$

$\quad$ grd012:

$\qquad (timeout = INFINITE\_TIME\_VALUE \Rightarrow tmout\_trig = \varnothing)$
$\qquad \wedge (timeout > 0 \Rightarrow tmout\_trig = \{proc \mapsto (PS\_Ready \mapsto (timeout + clock\_tick * ONE\_TICK\_TIME))\})$

$\quad$ grd013: $\quad timeout > 0 \Rightarrow wt = PROC\_WAIT\_TIMEOUT$

$\quad$ grd014: $\quad timeout = INFINITE\_TIME\_VALUE \Rightarrow wt = PROC\_WAIT\_OBJ$

$\quad$ grd015: $\quad finished\_core2(core) = FALSE$

$\quad$ grd016: $\quad location\_of\_service2(core) = Req\_busy\_resource \mapsto loc\_i$

$\quad$ grd017: $\quad \neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Req\_busy\_resource \mapsto loc\_i)$

$\quad$ grd301: $\quad sem \in semaphores$

$\quad$ grd302: $\quad core \in dom(wait\_semaphore\_whenzero)$

          grd303: $sem = wait\_semaphore\_whenzero(core)$
          grd304: $semaphores\_of\_partition(sem) = part$
          grd305: $location\_of\_service3(core) = Wait\_Semaphore\_Whenzero \mapsto loc\_i$
          grd306: $\neg(finished\_core2(core) = FALSE \land location\_of\_service3(core) = Wait\_Semaphore\_Whenzero \mapsto loc\_i)$

then

          act001: $location\_of\_service2(core) := Req\_busy\_resource \mapsto loc\_1$
          act002: $timeout\_trigger := timeout\_trigger \mathbin{\lhd\mkern-14mu-} tmout\_trig$
          act003: $process\_wait\_type(proc) := wt$
          act301: $location\_of\_service3(core) := Wait\_Semaphore\_Whenzero \mapsto loc\_1$

end

Event wait_semaphore_whenzero_waiting ⟨ordinary⟩ $\widehat{=}$

extends wait_semaphore_whenzero_waiting

    any

        part
        proc
        core
        sem
        t

    where

          grd001: $part \in PARTITIONS$
          grd002: $proc \in processes \cap dom(processes\_of\_partition)$
          grd003: $core \in CORES \cap dom(req\_busy\_resource\_proc) \cap dom(wait\_semaphore\_whenzero) \cap dom(location\_of\_service3)$
          grd004: $proc = req\_busy\_resource\_proc(core)$
          grd005: $processes\_of\_partition(proc) = part$
          grd006: $sem \in semaphores$
          grd007: $t \in \mathbb{N}$
          grd008: $semaphores\_of\_partition(sem) = part$
          grd009: $sem = wait\_semaphore\_whenzero(core)$
          grd010: $value\_of\_semaphores(sem) = 0$
          grd011: $finished\_core2(core) = FALSE$
          grd012: $location\_of\_service3(core) = Wait\_Semaphore\_Whenzero \mapsto loc\_1$
          grd013: $\neg(finished\_core2(core) = FALSE \land location\_of\_service3(core) = Wait\_Semaphore\_Whenzero \mapsto loc\_1)$
          grd201: $t = clock\_tick * ONE\_TICK\_TIME$

    then

          act001: $location\_of\_service3(core) := Wait\_Semaphore\_Whenzero \mapsto loc\_2$
          act002: $processes\_waitingfor\_semaphores(sem) := processes\_waitingfor\_semaphores(sem) \mathbin{\lhd\mkern-14mu-} \{proc \mapsto t\}$

    end

Event wait_semaphore_whenzero_schedule ⟨ordinary⟩ $\widehat{=}$

extends wait_semaphore_whenzero_schedule

    any

        part
        proc
        core
        sem

    where

          grd001: $part \in PARTITIONS$
          grd002: $proc \in processes \land proc \in dom(processes\_of\_partition)$
          grd003: $core \in CORES \cap dom(req\_busy\_resource\_proc) \land core \in dom(current\_processes\_flag) \land core \in dom(location\_of\_service2)$
          grd004: $proc = req\_busy\_resource\_proc(core)$
          grd005: $processes\_of\_partition(proc) = part$
          grd006: $part = current\_partition$
          grd012: $processes\_of\_partition(req\_busy\_resource\_proc(core)) \in dom(current\_partition\_flag)$
          grd007: $current\_partition\_flag(part) = TRUE$
          grd008: $current\_processes\_flag(core) = FALSE$

$grd009$: $finished\_core2(core) = FALSE$

$grd010$: $location\_of\_service2(core) = Req\_busy\_resource \mapsto loc\_1$

$grd011$: $\neg(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Req\_busy\_resource \mapsto loc\_1)$

$grd301$: $sem \in semaphores$

$grd302$: $core \in dom(wait\_semaphore\_whenzero)$

$grd303$: $sem = wait\_semaphore\_whenzero(core)$

$grd304$: $semaphores\_of\_partition(sem) = part$

$grd305$: $value\_of\_semaphores(sem) = 0$

$grd306$: $location\_of\_service3(core) = Wait\_Semaphore\_Whenzero \mapsto loc\_2$

$grd307$: $\neg(finished\_core2(core) = FALSE \land location\_of\_service3(core) = Wait\_Semaphore\_Whenzero \mapsto loc\_2)$

then

$act001$: $location\_of\_service2(core) := Req\_busy\_resource \mapsto loc\_2$

$act002$: $need\_reschedule := TRUE$

$act301$: $location\_of\_service3(core) := Wait\_Semaphore\_Whenzero \mapsto loc\_3$

end

Event wait_semaphore_whenzero_return ⟨ordinary⟩ $\widehat{=}$

extends wait_semaphore_whenzero_return

any

    part

    proc

    core

    sem

where

$grd001$: $part \in PARTITIONS$

$grd002$: $proc \in processes \land proc \in dom(processes\_of\_partition)$

$grd003$: $core \in CORES \cap dom(req\_busy\_resource\_proc) \land core \in dom(current\_processes\_flag) \land core \in dom(location\_of\_service2)$

$grd004$: $proc = req\_busy\_resource\_proc(core)$

$grd005$: $processes\_of\_partition(proc) = part$

$grd006$: $part = current\_partition$

$grd012$: $processes\_of\_partition(req\_busy\_resource\_proc(core)) \in dom(current\_partition\_flag)$

$grd007$: $current\_partition\_flag(part) = TRUE$

$grd008$: $current\_processes\_flag(core) = FALSE$

$grd009$: $finished\_core2(core) = FALSE$

$grd010$: $location\_of\_service2(core) = Req\_busy\_resource \mapsto loc\_2$

$grd011$: $\neg(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Req\_busy\_resource \mapsto loc\_2)$

$grd301$: $sem \in semaphores$

$grd302$: $core \in dom(wait\_semaphore\_whenzero)$

$grd303$: $sem = wait\_semaphore\_whenzero(core)$

$grd304$: $semaphores\_of\_partition(sem) = part$

$grd305$: $value\_of\_semaphores(sem) = 0$

$grd306$: $location\_of\_service3(core) = Wait\_Semaphore\_Whenzero \mapsto loc\_3$

$grd307$: $\neg(finished\_core2(core) = FALSE \land location\_of\_service3(core) = Wait\_Semaphore\_Whenzero \mapsto loc\_3)$

then

$act001$: $location\_of\_service2(core) := Req\_busy\_resource \mapsto loc\_r$

$act002$: $finished\_core2(core) := TRUE$

$act003$: $req\_busy\_resource\_proc := \{core\} \lhd req\_busy\_resource\_proc$

$act301$: $location\_of\_service3(core) := Wait\_Semaphore\_Whenzero \mapsto loc\_r$

$act302$: $wait\_semaphore\_whenzero := \{core\} \lhd wait\_semaphore\_whenzero$

end

Event signal_semaphore ⟨ordinary⟩ $\widehat{=}$

extends signal_semaphore

any

    core

    sem

part
where
> grd001: $core \in CORES$
> grd005: $sem \in semaphores$
> grd002: $value\_of\_semaphores(sem) \neq MaxValue\_of\_Semaphores(sem)$
> grd003: $processes\_waitingfor\_semaphores(sem) = \varnothing$
> grd004: $finished\_core2(core) = TRUE$
> grd201: $part \in dom(current\_partition\_flag) \wedge current\_partition = part \wedge current\_partition\_flag(part) = TRUE$
> grd203: $current\_processes\_flag(core) = TRUE$
> grd204: $semaphores\_of\_partition(sem) = part$

then
> act001: $value\_of\_semaphores(sem) := value\_of\_semaphores(sem) + 1$

end

Event signal_semaphore_needwakeupproc_init $\langle ordinary \rangle \;\widehat{=}$

extends signal_semaphore_needwakeupproc_init

> **any**
> > part
> > proc
> > newstate
> > core
> > sem

> **where**
> > grd001: $part \in PARTITIONS$
> > grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state)$
> > grd003: $newstate \in PROCESS\_STATES$
> > grd004: $core \in CORES$
> > grd005: $processes\_of\_partition(proc) = part$
> > grd017: $finished\_core2(core) = TRUE$
> > grd101: $partition\_mode(part) = PM\_NORMAL$
> > grd102: $process\_state(proc) = PS\_Waiting \vee process\_state(proc) = PS\_WaitandSuspend$
> > grd103: $process\_state(proc) = PS\_Waiting \Rightarrow newstate = PS\_Ready$
> > grd104: $process\_state(proc) = PS\_WaitandSuspend \Rightarrow newstate = PS\_Suspend$
> > grd201: $part = current\_partition$
> > grd203: $processes\_of\_partition(proc) \in dom(current\_partition\_flag)$
> > grd202: $current\_partition\_flag(part) = TRUE$
> > grd301: $sem \in semaphores$
> > grd302: $value\_of\_semaphores(sem) \neq MaxValue\_of\_Semaphores(sem)$
> > grd303: $processes\_waitingfor\_semaphores(sem) \neq \varnothing$

> **then**
> > act001: $process\_state(proc) := newstate$
> > act201: $location\_of\_service2(core) := Resource\_become\_avail \mapsto loc\_i$
> > act202: $finished\_core2(core) := FALSE$
> > act203: $resource\_become\_avail\_proc(core) := proc$
> > act204: $timeout\_trigger := \{proc\} \lhd timeout\_trigger$
> > act301: $location\_of\_service3(core) := Signal\_Semaphore\_NeedWakeup \mapsto loc\_i$
> > act302: $signal\_semaphore\_needwake(core) := sem$

> **end**

Event signal_semaphore_needwakeupproc_timeout_trig $\langle ordinary \rangle \;\widehat{=}$

extends signal_semaphore_needwakeupproc_timeout_trig

> **any**
> > part
> > proc
> > core
> > sem

> **where**
> > grd001: $part \in PARTITIONS$
> > grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition) \wedge proc \in dom(process\_wait\_type)$
> > grd003: $core \in CORES \cap dom(resource\_become\_avail\_proc) \wedge core \in dom(location\_of\_service2)$

$\quad$ grd004: $\quad proc = resource\_become\_avail\_proc(core)$

$\quad$ grd005: $\quad processes\_of\_partition(proc) = part$

$\quad$ grd006: $\quad partition\_mode(part) = PM\_NORMAL$

$\quad$ grd007: $\quad part = current\_partition$

$\quad$ grd013: $\quad processes\_of\_partition(proc) \in dom(current\_partition\_flag)$

$\quad$ grd008: $\quad current\_partition\_flag(part) = TRUE$

$\quad$ grd009: $\quad process\_wait\_type(proc) = PROC\_WAIT\_OBJ$

$\quad$ grd010: $\quad finished\_core2(core) = FALSE$

$\quad$ grd011: $\quad location\_of\_service2(core) = Resource\_become\_avail \mapsto loc\_i$

$\quad$ grd012: $\quad \neg(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Resource\_become\_avail \mapsto loc\_i)$

$\quad$ grd301: $\quad sem \in semaphores$

$\quad$ grd302: $\quad core \in dom(signal\_semaphore\_needwake)$

$\quad$ grd303: $\quad sem = signal\_semaphore\_needwake(core)$

$\quad$ grd304: $\quad value\_of\_semaphores(sem) \neq MaxValue\_of\_Semaphores(sem)$

$\quad$ grd305: $\quad processes\_waitingfor\_semaphores(sem) \neq \varnothing$

$\quad$ grd306: $\quad location\_of\_service3(core) = Signal\_Semaphore\_NeedWakeup \mapsto loc\_i$

$\quad$ grd307: $\quad \neg(finished\_core2(core) = FALSE \land location\_of\_service3(core) = Signal\_Semaphore\_NeedWakeup \mapsto loc\_i)$

then

$\quad$ act001: $location\_of\_service2(core) := Resource\_become\_avail \mapsto loc\_1$

$\quad$ act002: $process\_wait\_type := \{proc\} \lhd process\_wait\_type$

$\quad$ act301: $location\_of\_service3(core) := Signal\_Semaphore\_NeedWakeup \mapsto loc\_1$

end

Event signal_semaphore_needwakeupproc_insert $\langle ordinary \rangle \; \widehat{=}$

extends signal_semaphore_needwakeupproc_insert

any

$\qquad part$

$\qquad proc$

$\qquad core$

$\qquad sem$

$\qquad$ t

where

$\quad$ grd001: $\quad part \in PARTITIONS$

$\quad$ grd002: $\quad proc \in processes \land proc \in dom(processes\_of\_partition)$

$\quad$ grd003: $\quad core \in CORES \cap dom(resource\_become\_avail\_proc) \cap dom(location\_of\_service3)$

$\quad$ grd004: $\quad proc = resource\_become\_avail\_proc(core)$

$\quad$ grd005: $\quad processes\_of\_partition(proc) = part$

$\quad$ grd006: $\quad partition\_mode(part) = PM\_NORMAL$

$\quad$ grd007: $\quad sem \in semaphores$

$\quad$ grd008: $\quad core \in dom(signal\_semaphore\_needwake)$

$\quad$ grd009: $\quad sem = signal\_semaphore\_needwake(core)$

$\quad$ grd010: $\quad value\_of\_semaphores(sem) \neq MaxValue\_of\_Semaphores(sem)$

$\quad$ grd011: $\quad processes\_waitingfor\_semaphores(sem) \neq \varnothing$

$\quad$ grd012: $\quad finished\_core2(core) = FALSE$

$\quad$ grd013: $\quad location\_of\_service3(core) = Signal\_Semaphore\_NeedWakeup \mapsto loc\_1$

$\quad$ grd014: $\quad \neg(finished\_core2(core) = FALSE \land location\_of\_service3(core) = Signal\_Semaphore\_NeedWakeup \mapsto loc\_1)$

$\quad$ grd201: $\quad part = current\_partition$

$\quad$ grd202: $\quad current\_partition\_flag(part) = TRUE$

$\quad$ grd203: $\quad current\_processes\_flag(core) = TRUE$

$\quad$ grd204: $\quad processes\_waitingfor\_semaphores(sem) \neq \varnothing \land (proc \mapsto t) \in processes\_waitingfor\_semaphores(sem)$

$\quad$ grd205: $\quad quediscipline\_of\_semaphores(sem) = QUEUE\_FIFO \Rightarrow (\forall p1, t1 \cdot (p1 \mapsto t1 \in processes\_waitingfor\_semaph$ $t \leq t1))$

$\quad$ grd207: $\quad part \in dom(current\_partition\_flag)$

$\quad$ grd206: $\quad quediscipline\_of\_semaphores(sem) = QUEUE\_PRIORITY \Rightarrow (\forall p1, t1 \cdot (p1 \mapsto t1 \in processes\_waitingfor\_semaphores(sem) \Rightarrow currentpriority\_of\_process(proc) \geq currentpriority\_of\_process(p1)))$

then

    act001: $location\_of\_service3(core) := Signal\_Semaphore\_NeedWakeup \mapsto loc\_2$

    act002: $processes\_waiting\,for\_semaphores(sem) := \{proc\} \mathbin{\lhd\mkern-8mu-} processes\_waiting\,for\_semaphores(sem)$

end

Event signal_semaphore_needwakeupproc_schedule ⟨ordinary⟩ $\widehat{=}$

extends signal_semaphore_needwakeupproc_schedule

    any

        part

        proc

        core

        resch

        sem

    where

    grd001: $part \in PARTITIONS$

    grd002: $proc \in processes \land proc \in dom(processes\_of\_partition)$

    grd003: $core \in CORES \cap dom(resource\_become\_avail\_proc) \land core \in dom(location\_of\_service2)$

    grd004: $proc = resource\_become\_avail\_proc(core)$

    grd005: $processes\_of\_partition(proc) = part$

    grd006: $partition\_mode(part) = PM\_NORMAL$

    grd007: $part = current\_partition$

    grd013: $processes\_of\_partition(proc) \in dom(current\_partition\_flag)$

    grd008: $current\_partition\_flag(part) = TRUE$

    grd009: $resch \in BOOL$

    grd010: $finished\_core2(core) = FALSE$

    grd011: $location\_of\_service2(core) = Resource\_become\_avail \mapsto loc\_1$

    grd012: $\neg(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Resource\_become\_avail \mapsto loc\_1)$

    grd301: ⟨theorem⟩ $sem \in semaphores$

    grd302: $core \in dom(signal\_semaphore\_needwake)$

    grd303: $sem = signal\_semaphore\_needwake(core)$

    grd304: $value\_of\_semaphores(sem) \neq MaxValue\_of\_Semaphores(sem)$

    grd305: $location\_of\_service3(core) = Signal\_Semaphore\_NeedWakeup \mapsto loc\_2$

    grd306: $\neg(finished\_core2(core) = FALSE \land location\_of\_service3(core) = Signal\_Semaphore\_NeedWakeup \mapsto loc\_2)$

    then

    act001: $location\_of\_service2(core) := Resource\_become\_avail \mapsto loc\_2$

    act002: $need\_reschedule := resch$

    act301: $location\_of\_service3(core) := Signal\_Semaphore\_NeedWakeup \mapsto loc\_3$

    end

Event signal_semaphore_needwakeupproc_return ⟨ordinary⟩ $\widehat{=}$

extends signal_semaphore_needwakeupproc_return

    any

        part

        proc

        core

        sem

    where

    grd001: $part \in PARTITIONS$

    grd002: $proc \in processes \land proc \in dom(processes\_of\_partition)$

    grd003: $core \in CORES \cap dom(resource\_become\_avail\_proc) \land core \in dom(location\_of\_service2)$

    grd004: $proc = resource\_become\_avail\_proc(core)$

    grd005: $processes\_of\_partition(proc) = part$

    grd006: $partition\_mode(part) = PM\_NORMAL$

    grd007: $part = current\_partition$

    grd012: $processes\_of\_partition(proc) \in dom(current\_partition\_flag)$

    grd008: $current\_partition\_flag(part) = TRUE$

    grd009: $finished\_core2(core) = FALSE$

    grd010: $location\_of\_service2(core) = Resource\_become\_avail \mapsto loc\_2$

>> **grd011**: $\neg(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Resource\_become\_avail \mapsto loc\_2)$
>> **grd301**: $sem \in semaphores$
>> **grd302**: $core \in dom(signal\_semaphore\_needwake)$
>> **grd303**: $sem = signal\_semaphore\_needwake(core)$
>> **grd304**: $value\_of\_semaphores(sem) \neq MaxValue\_of\_Semaphores(sem)$
>> **grd305**: $location\_of\_service3(core) = Signal\_Semaphore\_NeedWakeup \mapsto loc\_3$
>> **grd306**: $\neg(finished\_core2(core) = FALSE \land location\_of\_service3(core) = Signal\_Semaphore\_NeedWakeup \mapsto loc\_3)$

> **then**
>> **act001**: $location\_of\_service2(core) := Resource\_become\_avail \mapsto loc\_r$
>> **act002**: $finished\_core2(core) := TRUE$
>> **act003**: $resource\_become\_avail\_proc := \{core\} \lhd resource\_become\_avail\_proc$
>> **act301**: $location\_of\_service3(core) := Signal\_Semaphore\_NeedWakeup \mapsto loc\_r$
>> **act302**: $signal\_semaphore\_needwake := \{core\} \lhd signal\_semaphore\_needwake$

> **end**

Event get_semaphore_id ⟨ordinary⟩ $\widehat{=}$

> **any**
>> part
>> sem
>> core

> **where**
>> **grd001**: $part \in dom(current\_partition\_flag) \land current\_partition = part \land current\_partition\_flag(part) = TRUE$
>> **grd003**: $sem \in semaphores$
>> **grd004**: $semaphores\_of\_partition(sem) = part$
>> **grd005**: $core \in CORES$
>> **grd006**: $finished\_core2(core) = TRUE$

> **then**
>> *skip*

> **end**

Event get_semaphore_status ⟨ordinary⟩ $\widehat{=}$

> **any**
>> part
>> core
>> sem

> **where**
>> **grd001**: $part \in dom(current\_partition\_flag) \land current\_partition = part \land current\_partition\_flag(part) = TRUE$
>> **grd003**: $sem \in semaphores$
>> **grd004**: $semaphores\_of\_partition(sem) = part$
>> **grd005**: $core \in CORES$
>> **grd006**: $finished\_core2(core) = TRUE$

> **then**
>> *skip*

> **end**

Event create_event ⟨ordinary⟩ $\widehat{=}$

extends create_event

> **any**
>> *core*
>> *ev*
>> part

> **where**
>> **grd001**: $core \in CORES$
>> **grd002**: $ev \in EVENTS \land ev \notin events$
>> **grd003**: $finished\_core2(core) = TRUE$
>> **grd201**: $part \in dom(current\_partition\_flag) \land current\_partition = part \land current\_partition\_flag(part) = TRUE$

$grd203$: $partition\_mode(current\_partition) = PM\_COLD\_START \lor partition\_mode(current\_partition) = PM\_WARM\_START$

then

    $act001$: $events := events \cup \{ev\}$

    $act002$: $state\_of\_events(ev) := EVENT\_DOWN$

    $act003$: $events\_of\_partition(ev) := current\_partition$

    $act004$: $processes\_waitingfor\_events(ev) := \varnothing$

end

Event set_event ⟨ordinary⟩ $\widehat{=}$

extends set_event

    any

        core

        ev

        part

    where

        $grd001$: $core \in CORES$

        $grd002$: $ev \in events$

        $grd003$: $processes\_waitingfor\_events(ev) = \varnothing$

        $grd004$: $finished\_core2(core) = TRUE$

        $grd201$: $part \in dom(current\_partition\_flag) \land current\_partition = part \land current\_partition\_flag(part) = TRUE$

        $grd203$: $events\_of\_partition(ev) = part$

        $grd204$: $current\_processes\_flag(core) = TRUE$

    then

        $act001$: $state\_of\_events(ev) := EVENT\_UP$

    end

Event set_event_needwakeupprocs_init ⟨ordinary⟩ $\widehat{=}$

extends set_event_needwakeupprocs_init

    any

        part

        procs

        newstates

        core

        ev

    where

        $grd001$: $part \in PARTITIONS$

        $grd002$: $procs \subseteq processes \cap dom(process\_state)$

        $grd003$: $newstates \in procs \rightarrow PROCESS\_STATES$

        $grd004$: $core \in CORES$

        $grd005$: $procs \subseteq processes\_of\_partition^{-1}[\{part\}]$

        $grd101$: $partition\_mode(part) = PM\_NORMAL$

        $grd102$: $\forall proc \cdot (proc \in procs \Rightarrow process\_state(proc) = PS\_Waiting \lor process\_state(proc) = PS\_WaitandSuspend)$

        $grd103$: $\forall proc \cdot (proc \in procs \land process\_state(proc) = PS\_Waiting \Rightarrow newstates(proc) = PS\_Ready)$

        $grd104$: $\forall proc \cdot (proc \in procs \land process\_state(proc) = PS\_WaitandSuspend \Rightarrow newstates(proc) = PS\_Suspend)$

        $grd301$: $part = current\_partition$

        $grd303$: $part \in dom(current\_partition\_flag)$

        $grd302$: $current\_partition\_flag(part) = TRUE$

        $grd304$: $finished\_core2(core) = TRUE$

        $grd401$: $ev \in events$

        $grd402$: $processes\_waitingfor\_events(ev) \neq \varnothing$

    then

        $act001$: $process\_state := process\_state \Leftarrow newstates$

        $act301$: $location\_of\_service2(core) := Resource\_become\_avail2 \mapsto loc\_i$

        $act302$: $finished\_core2(core) := FALSE$

        $act303$: $resource\_become\_avail2(core) := procs$

        $act304$: $timeout\_trigger := procs \lhd timeout\_trigger$

$act401$: $location\_of\_service3(core) := Set\_Event\_NeedWakeup \mapsto loc\_i$

$act402$: $set\_event\_needwake(core) := ev$

end

Event set_event_needwakeupprocs_timeout_trig $\langle ordinary \rangle \;\widehat{=}$

extends set_event_needwakeupprocs_timeout_trig

    any

        part

        procs

        core

        ev

    where

$grd001$: $part \in PARTITIONS$

$grd002$: $procs \subseteq (processes \cap dom(process\_state))$

$grd003$: $core \in CORES \wedge core \in dom(location\_of\_service2) \wedge core \in dom(resource\_become\_avail2)$

$grd004$: $procs = resource\_become\_avail2(core)$

$grd005$: $part = current\_partition$

$grd006$: $partition\_mode(part) = PM\_NORMAL$

$grd007$: $\forall proc \cdot (proc \in procs \wedge proc \in dom(process\_wait\_type) \Rightarrow process\_wait\_type(proc) = PROC\_WAIT\_OBJ)$

$grd008$: $finished\_core2(core) = FALSE$

$grd009$: $location\_of\_service2(core) = Resource\_become\_avail2 \mapsto loc\_i$

$grd010$: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Resource\_become\_avail2 \mapsto loc\_i)$

$grd301$: $ev \in events$

$grd302$: $processes\_waiting for\_events(ev) \neq \varnothing$

$grd303$: $core \in dom(set\_event\_needwake)$

$grd304$: $ev = set\_event\_needwake(core)$

$grd305$: $location\_of\_service3(core) = Set\_Event\_NeedWakeup \mapsto loc\_i$

$grd306$: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service3(core) = Set\_Event\_NeedWakeup \mapsto loc\_i)$

    then

$act001$: $location\_of\_service2(core) := Resource\_become\_avail2 \mapsto loc\_1$

$act002$: $process\_wait\_type := procs \lhd process\_wait\_type$

$act301$: $location\_of\_service3(core) := Set\_Event\_NeedWakeup \mapsto loc\_1$

    end

Event set_event_needwakeupprocs_insert $\langle ordinary \rangle \;\widehat{=}$

extends set_event_needwakeupprocs_insert

    any

        part

        procs

        core

        ev

    where

$grd001$: $part \in PARTITIONS$

$grd002$: $procs \subseteq processes$

$grd003$: $core \in CORES \wedge core \in dom(location\_of\_service3) \wedge core \in dom(set\_event\_needwake) \cap dom(resource\_become\_avail2)$

$grd004$: $procs = resource\_become\_avail2(core)$

$grd005$: $part = current\_partition$

$grd006$: $partition\_mode(part) = PM\_NORMAL$

$grd007$: $ev \in events$

$grd008$: $ev = set\_event\_needwake(core)$

$grd009$: $processes\_waiting for\_events(ev) \neq \varnothing$

$grd010$: $finished\_core2(core) = FALSE$

$grd011$: $location\_of\_service3(core) = Set\_Event\_NeedWakeup \mapsto loc\_1$

$grd012$: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service3(core) = Set\_Event\_NeedWakeup \mapsto loc\_1)$

$grd201$: $current\_partition\_flag(part) = TRUE$

        grd202: $current\_processes\_flag(core) = TRUE$
        grd203: $partition\_mode(part) = PM\_NORMAL$
        grd204: $part \in dom(current\_partition\_flag)$
then
        act001: $location\_of\_service3(core) := Set\_Event\_NeedWakeup \mapsto loc\_2$
        act002: $state\_of\_events(ev) := EVENT\_UP$
        act003: $processes\_waiting for\_events(ev) := processes\_waiting for\_events(ev) \setminus procs$
end

Event set_event_needwakeupprocs_schedule ⟨ordinary⟩ $\widehat{=}$
extends set_event_needwakeupprocs_schedule
    any
        part
        procs
        core
        resch
        ev
    where
        grd001: $part \in PARTITIONS$
        grd002: $procs \subseteq (processes \cap dom(process\_state))$
        grd003: $core \in CORES \wedge core \in dom(location\_of\_service2) \wedge core \in dom(resource\_become\_avail2)$

        grd004: $procs = resource\_become\_avail2(core)$
        grd005: $part = current\_partition$
        grd006: $partition\_mode(part) = PM\_NORMAL$
        grd008: $resch \in BOOL$
        grd009: $finished\_core2(core) = FALSE$
        grd010: $location\_of\_service2(core) = Resource\_become\_avail2 \mapsto loc\_1$
        grd011: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Resource\_become\_avail2 \mapsto loc\_1)$
        grd301: $ev \in events$
        grd302: $core \in dom(set\_event\_needwake)$
        grd303: $ev = set\_event\_needwake(core)$
        grd304: $location\_of\_service3(core) = Set\_Event\_NeedWakeup \mapsto loc\_2$
        grd305: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service3(core) = Set\_Event\_NeedWakeup \mapsto loc\_2)$
    then
        act001: $location\_of\_service2(core) := Resource\_become\_avail2 \mapsto loc\_2$
        act002: $need\_reschedule := resch$
        act301: $location\_of\_service3(core) := Set\_Event\_NeedWakeup \mapsto loc\_3$
    end

Event set_event_needwakeupprocs_return ⟨ordinary⟩ $\widehat{=}$
extends set_event_needwakeupprocs_return
    any
        part
        procs
        core
        ev
    where
        grd001: $part \in PARTITIONS$
        grd002: $procs \subseteq (processes \cap dom(process\_state))$
        grd003: $core \in CORES \wedge core \in dom(location\_of\_service2) \wedge core \in dom(resource\_become\_avail2)$

        grd004: $procs = resource\_become\_avail2(core)$
        grd005: $part = current\_partition$
        grd006: $partition\_mode(part) = PM\_NORMAL$
        grd007: $finished\_core2(core) = FALSE$
        grd008: $location\_of\_service2(core) = Resource\_become\_avail2 \mapsto loc\_2$
        grd009: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Resource\_become\_avail2 \mapsto loc\_2)$

grd301: $ev \in events$
grd302: $core \in dom(set\_event\_needwake)$
grd303: $ev = set\_event\_needwake(core)$
grd304: $location\_of\_service3(core) = Set\_Event\_NeedWakeup \mapsto loc\_3$
grd305: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service3(core) = Set\_Event\_NeedWakeup \mapsto loc\_3)$

then

act001: $location\_of\_service2(core) := Resource\_become\_avail2 \mapsto loc\_r$
act002: $finished\_core2(core) := TRUE$
act003: $resource\_become\_avail2 := \{core\} \lhd resource\_become\_avail2$
act301: $location\_of\_service3(core) := Set\_Event\_NeedWakeup \mapsto loc\_r$
act302: $set\_event\_needwake := \{core\} \lhd set\_event\_needwake$

end

Event reset_event ⟨ordinary⟩ $\widehat{=}$

extends reset_event

any

    core
    ev
    part

where

grd001: $core \in CORES$
grd002: $ev \in events$
grd003: $finished\_core2(core) = TRUE$
grd201: $part \in dom(current\_partition\_flag) \wedge current\_partition = part \wedge current\_partition\_flag(part) = TRUE$
grd203: $current\_processes\_flag(core) = TRUE$
grd204: $events\_of\_partition(ev) = part$

then

act001: $state\_of\_events(ev) := EVENT\_DOWN$

end

Event wait_event ⟨ordinary⟩ $\widehat{=}$

extends wait_event

any

    core
    ev
    part

where

grd001: $core \in CORES$
grd002: $ev \in events$
grd003: $finished\_core2(core) = TRUE$
grd201: $part \in dom(current\_partition\_flag) \wedge current\_partition = part \wedge current\_partition\_flag(part) = TRUE$
grd203: $current\_processes\_flag(core) = TRUE$
grd204: $events\_of\_partition(ev) = part$

then

    skip

end

Event wait_event_whendown_init ⟨ordinary⟩ $\widehat{=}$

extends wait_event_whendown_init

any

    part
    proc
    newstate
    core
    ev

where

grd001: $part \in PARTITIONS$
grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state) \cap dom(process\_wait\_type)$

        grd003: $newstate \in PROCESS\_STATES$

        grd004: $core \in CORES \land core \in dom(current\_processes\_flag)$

        grd005: $processes\_of\_partition(proc) = part$

        grd017: $finished\_core2(core) = TRUE$

        grd101: $partition\_mode(part) = PM\_NORMAL$

        grd102: $process\_state(proc) = PS\_Running$

        grd103: $newstate = PS\_Waiting$

        grd205: $proc \in dom(delaytime\_of\_process) \land proc \in dom(process\_wait\_type)$

        grd201: $part = current\_partition \land current\_partition \in dom(current\_partition\_flag)$

        grd202: $current\_partition\_flag(part) = TRUE$

        grd203: $current\_processes\_flag(core) = TRUE$

        grd204: $proc = current\_processes(core)$

        grd301: $ev \in events$

        grd302: $events\_of\_partition(ev) = part$

        grd303: $state\_of\_events(ev) = EVENT\_DOWN$

then

        act001: $process\_state(proc) := newstate$

        act002: $location\_of\_service2(core) := Req\_busy\_resource \mapsto loc\_i$

        act003: $finished\_core2(core) := FALSE$

        act004: $req\_busy\_resource\_proc(core) := proc$

        act005: $current\_processes\_flag(core) := FALSE$

        act006: $current\_processes := \{core\} \lhd current\_processes$

        act301: $location\_of\_service3(core) := Wait\_Event\_Whendown \mapsto loc\_i$

        act302: $wait\_event\_whendown(core) := ev$

end

Event wait_event_whendown_timeout ⟨ordinary⟩ $\widehat{=}$

extends wait_event_whendown_timeout

    any

        part

        proc

        core

        timeout

        tmout_trig

        wt

        ev

    where

        grd001: $part \in PARTITIONS$

        grd002: $proc \in processes \land proc \in dom(processes\_of\_partition)$

        grd003: $core \in CORES \cap dom(req\_busy\_resource\_proc) \land core \in dom(current\_processes\_flag) \land$
           $core \in dom(location\_of\_service2)$

        grd004: $proc = req\_busy\_resource\_proc(core)$

        grd005: $processes\_of\_partition(proc) = part$

        grd006: $part = current\_partition$

        grd018: $processes\_of\_partition(req\_busy\_resource\_proc(core)) \in dom(current\_partition\_flag)$

        grd007: $current\_partition\_flag(part) = TRUE$

        grd008: $current\_processes\_flag(core) = TRUE$

        grd009: $timeout \geq 0$

        grd010: $wt \in PROCESS\_WAIT\_TYPES \land (wt = PROC\_WAIT\_OBJ \lor wt = PROC\_WAIT\_TIMEOUT)$

        grd011: $tmout\_trig \in processes \nrightarrow (PROCESS\_STATES \times \mathbb{N}_1)$

        grd012:
           $(timeout = INFINITE\_TIME\_VALUE \Rightarrow tmout\_trig = \varnothing)$
           $\land (timeout > 0 \Rightarrow tmout\_trig = \{proc \mapsto (PS\_Ready \mapsto (timeout + clock\_tick * ONE\_TICK\_TIME))\})$

        grd013: $timeout > 0 \Rightarrow wt = PROC\_WAIT\_TIMEOUT$

        grd014: $timeout = INFINITE\_TIME\_VALUE \Rightarrow wt = PROC\_WAIT\_OBJ$

        grd015: $finished\_core2(core) = FALSE$

        grd016: $location\_of\_service2(core) = Req\_busy\_resource \mapsto loc\_i$

        grd017: $\neg(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Req\_busy\_resource \mapsto$
           $loc\_i)$

$\quad$ grd301: $\quad ev \in events$

$\quad$ grd302: $\quad core \in dom(wait\_event\_whendown)$

$\quad$ grd303: $\quad ev = wait\_event\_whendown(core)$

$\quad$ grd304: $\quad events\_of\_partition(ev) = part$

$\quad$ grd305: $\quad state\_of\_events(ev) = EVENT\_DOWN$

$\quad$ grd306: $\quad location\_of\_service3(core) = Wait\_Event\_Whendown \mapsto loc\_i$

$\quad$ grd307: $\quad \neg(finished\_core2(core) = FALSE \wedge location\_of\_service3(core) = Wait\_Event\_Whendown \mapsto loc\_i)$

then

$\quad$ act001: $location\_of\_service2(core) := Req\_busy\_resource \mapsto loc\_1$

$\quad$ act002: $timeout\_trigger := timeout\_trigger \lhd tmout\_trig$

$\quad$ act003: $process\_wait\_type(proc) := wt$

$\quad$ act301: $location\_of\_service3(core) := Wait\_Event\_Whendown \mapsto loc\_1$

end

Event wait_event_whendown_waiting ⟨ordinary⟩ $\widehat{=}$

extends wait_event_whendown_waiting

$\quad$ **any**

$\qquad$ *part*

$\qquad$ *proc*

$\qquad$ *core*

$\qquad$ *ev*

$\quad$ **where**

$\quad$ grd001: $\quad part \in PARTITIONS$

$\quad$ grd002: $\quad proc \in processes \cap dom(processes\_of\_partition)$

$\quad$ grd003: $\quad core \in CORES \wedge core \in dom(req\_busy\_resource\_proc) \wedge core \in dom(wait\_event\_whendown) \cap dom(location\_of\_service3)$

$\quad$ grd004: $\quad proc = req\_busy\_resource\_proc(core)$

$\quad$ grd005: $\quad processes\_of\_partition(proc) = part$

$\quad$ grd006: $\quad ev \in events$

$\quad$ grd007: $\quad ev = wait\_event\_whendown(core)$

$\quad$ grd008: $\quad events\_of\_partition(ev) = part$

$\quad$ grd009: $\quad state\_of\_events(ev) = EVENT\_DOWN$

$\quad$ grd012: $\quad finished\_core2(core) = FALSE$

$\quad$ grd010: $\quad location\_of\_service3(core) = Wait\_Event\_Whendown \mapsto loc\_1$

$\quad$ grd011: $\quad \neg(finished\_core2(core) = FALSE \wedge location\_of\_service3(core) = Wait\_Event\_Whendown \mapsto loc\_1)$

$\quad$ grd201: $\quad part = current\_partition$

$\quad$ grd202: $\quad current\_partition\_flag(part) = TRUE$

$\quad$ grd203: $\quad current\_processes\_flag(core) = TRUE$

$\quad$ grd204: $\quad events\_of\_partition(ev) = part$

$\quad$ grd205: $\quad part \in dom(current\_partition\_flag)$

$\quad$ **then**

$\quad$ act001: $location\_of\_service3(core) := Wait\_Event\_Whendown \mapsto loc\_2$

$\quad$ act002: $processes\_waitingfor\_events(ev) := processes\_waitingfor\_events(ev) \cup \{proc\}$

$\quad$ **end**

Event wait_event_whendown_schedule ⟨ordinary⟩ $\widehat{=}$

extends wait_event_whendown_schedule

$\quad$ **any**

$\qquad$ *part*

$\qquad$ *proc*

$\qquad$ *core*

$\qquad$ *ev*

$\quad$ **where**

$\quad$ grd001: $\quad part \in PARTITIONS$

$\quad$ grd002: $\quad proc \in processes \wedge proc \in dom(processes\_of\_partition)$

$\quad$ grd003: $\quad core \in CORES \cap dom(req\_busy\_resource\_proc) \wedge core \in dom(current\_processes\_flag) \wedge core \in dom(location\_of\_service2)$

$\quad$ grd004: $\quad proc = req\_busy\_resource\_proc(core)$

$\quad$ grd005: $\quad processes\_of\_partition(proc) = part$

        grd006: $part = current\_partition$

        grd012: $processes\_of\_partition(req\_busy\_resource\_proc(core)) \in dom(current\_partition\_flag)$

        grd007: $current\_partition\_flag(part) = TRUE$

        grd008: $current\_processes\_flag(core) = FALSE$

        grd009: $finished\_core2(core) = FALSE$

        grd010: $location\_of\_service2(core) = Req\_busy\_resource \mapsto loc\_1$

        grd011: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Req\_busy\_resource \mapsto loc\_1)$

        grd301: $ev \in events$

        grd302: $core \in dom(wait\_event\_whendown)$

        grd303: $events\_of\_partition(ev) = part$

        grd304: $state\_of\_events(ev) = EVENT\_DOWN$

        grd305: $location\_of\_service3(core) = Wait\_Event\_Whendown \mapsto loc\_2$

        grd306: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service3(core) = Wait\_Event\_Whendown \mapsto loc\_2)$

then

        act001: $location\_of\_service2(core) := Req\_busy\_resource \mapsto loc\_2$

        act002: $need\_reschedule := TRUE$

        act301: $location\_of\_service3(core) := Wait\_Event\_Whendown \mapsto loc\_3$

end

Event wait_event_whendown_return $\langle ordinary \rangle \,\widehat{=}$

extends wait_event_whendown_return

    any

        part

        proc

        core

        ev

    where

        grd001: $part \in PARTITIONS$

        grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition)$

        grd003: $core \in CORES \cap dom(req\_busy\_resource\_proc) \wedge core \in dom(current\_processes\_flag) \wedge core \in dom(location\_of\_service2)$

        grd004: $proc = req\_busy\_resource\_proc(core)$

        grd005: $processes\_of\_partition(proc) = part$

        grd006: $part = current\_partition$

        grd012: $processes\_of\_partition(req\_busy\_resource\_proc(core)) \in dom(current\_partition\_flag)$

        grd007: $current\_partition\_flag(part) = TRUE$

        grd008: $current\_processes\_flag(core) = FALSE$

        grd009: $finished\_core2(core) = FALSE$

        grd010: $location\_of\_service2(core) = Req\_busy\_resource \mapsto loc\_2$

        grd011: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Req\_busy\_resource \mapsto loc\_2)$

        grd301: $ev \in events$

        grd302: $core \in dom(wait\_event\_whendown)$

        grd303: $events\_of\_partition(ev) = part$

        grd304: $state\_of\_events(ev) = EVENT\_DOWN$

        grd305: $location\_of\_service3(core) = Wait\_Event\_Whendown \mapsto loc\_3$

        grd306: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service3(core) = Wait\_Event\_Whendown \mapsto loc\_3)$

    then

        act001: $location\_of\_service2(core) := Req\_busy\_resource \mapsto loc\_r$

        act002: $finished\_core2(core) := TRUE$

        act003: $req\_busy\_resource\_proc := \{core\} \lhd req\_busy\_resource\_proc$

        act301: $location\_of\_service3(core) := Wait\_Event\_Whendown \mapsto loc\_r$

        act302: $wait\_event\_whendown := \{core\} \lhd wait\_event\_whendown$

    end

Event get_event_id $\langle ordinary \rangle \,\widehat{=}$

    any

        part

        core

        ev

where

    grd001: $part \in dom(current\_partition\_flag) \land current\_partition = part \land current\_partition\_flag(part) = TRUE$

    grd003: $ev \in events$

    grd004: $events\_of\_partition(ev) = part$

    grd005: $core \in CORES$

    grd006: $finished\_core2(core) = TRUE$

then

    skip

end

Event get_event_status ⟨ordinary⟩ $\widehat{=}$

    any

        part

        core

        ev

    where

        grd001: $part \in dom(current\_partition\_flag) \land current\_partition = part \land current\_partition\_flag(part) = TRUE$

        grd003: $ev \in events$

        grd004: $events\_of\_partition(ev) = part$

        grd005: $core \in CORES$

        grd006: $finished\_core2(core) = TRUE$

    then

        skip

    end

Event create_mutex_init ⟨ordinary⟩ $\widehat{=}$

extends create_mutex_init

    any

        part

        core

        mutex

        disc

    where

        grd001: $part = current\_partition$

        grd002: $core \in CORES$

        grd003: $mutex \in MUTEXS \land mutex \notin mutexs$

        grd004: $finished\_core3(core) = TRUE$

        grd201: $disc \in QUEUING\_DISCIPLINE$

    then

        act001: $mutexs := mutexs \cup \{mutex\}$

        act002: $create\_of\_mutex(core) := mutex$

        act003: $finished\_core3(core) := FALSE$

        act004: $location\_of\_service3(core) := Create\_Mutex \mapsto loc\_i$

        act201: $quediscipline\_of\_mutexs(mutex) := disc$

    end

Event create_mutex_priority ⟨ordinary⟩ $\widehat{=}$

extends create_mutex_priority

    any

        part

        core

        mutex

        pri

    where

        grd001: $part = current\_partition$

        grd002: $core \in CORES \land core \in dom(create\_of\_mutex) \land core \in dom(location\_of\_service3)$

        grd003: $mutex \in mutexs$

        grd004: $mutex = create\_of\_mutex(core)$

   grd005: $pri \in \mathbb{N}_1$
   grd006: $finished\_core3(core) = FALSE$
   grd007: $location\_of\_service3(core) = Create\_Mutex \mapsto loc\_i$
   grd008: $\neg(finished\_core3(core) = FALSE \land location\_of\_service3(core) = Create\_Mutex \mapsto loc\_i)$
  then
   act001: $priority\_of\_mutex(mutex) := pri$
   act002: $location\_of\_service3(core) := Create\_Mutex \mapsto loc\_1$
  end

Event create_mutex_lock_count ⟨ordinary⟩ $\widehat{=}$

extends create_mutex_lock_count

  any
   part
   core
   mutex
  where
   grd001: $part = current\_partition$
   grd002: $core \in CORES \land core \in dom(create\_of\_mutex) \land core \in dom(location\_of\_service3)$
   grd003: $mutex \in mutexs$
   grd004: $mutex = create\_of\_mutex(core)$
   grd005: $finished\_core2(core) = FALSE$
   grd006: $location\_of\_service3(core) = Create\_Mutex \mapsto loc\_1$
   grd007: $\neg(finished\_core2(core) = FALSE \land location\_of\_service3(core) = Create\_Mutex \mapsto loc\_1)$
  then
   act001: $mutex\_of\_count(mutex) := 0$
   act002: $location\_of\_service3(core) := Create\_Mutex \mapsto loc\_2$
  end

Event create_mutex_state ⟨ordinary⟩ $\widehat{=}$

extends create_mutex_state

  any
   part
   core
   mutex
  where
   grd001: $part = current\_partition$
   grd002: $core \in CORES \land core \in dom(create\_of\_mutex) \land core \in dom(location\_of\_service3)$
   grd003: $mutex \in mutexs$
   grd004: $mutex = create\_of\_mutex(core)$
   grd005: $finished\_core2(core) = FALSE$
   grd006: $location\_of\_service3(core) = Create\_Mutex \mapsto loc\_2$
   grd007: $\neg(finished\_core2(core) = FALSE \land location\_of\_service3(core) = Create\_Mutex \mapsto loc\_2)$
  then
   act001: $mutex\_state(mutex) := MUTEX\_AVAILABLE$
   act002: $location\_of\_service3(core) := Create\_Mutex \mapsto loc\_3$
  end

Event create_mutex_return ⟨ordinary⟩ $\widehat{=}$

extends create_mutex_return

  any
   part
   core
  where
   grd001: $part = current\_partition$
   grd002: $core \in CORES \land core \in dom(location\_of\_service3)$
   grd003: $finished\_core2(core) = FALSE$
   grd004: $location\_of\_service3(core) = Create\_Mutex \mapsto loc\_3$
   grd005: $\neg(finished\_core2(core) = FALSE \land location\_of\_service3(core) = Create\_Mutex \mapsto loc\_3)$

then

    act001: $create\_of\_mutex := \{core\} \lhd create\_of\_mutex$

    act002: $finished\_core2(core) := TRUE$

    act003: $location\_of\_service3(core) := Create\_Mutex \mapsto loc\_r$

end

Event acquire_mutex_init ⟨ordinary⟩ $\widehat{=}$

extends acquire_mutex_init

    any

        part

        core

        mutex

        proc

    where

        grd001: $part = current\_partition$

        grd002: $core \in CORES$

        grd003: $mutex \in mutexs$

        grd004: $proc \in processes$

        grd005: $mutex\_state(mutex) = MUTEX\_AVAILABLE$

        grd009: $mutex \notin dom(mutex\_of\_process)$

        grd006: $proc \notin ran(mutex\_of\_process)$

        grd007: $processes\_waiting for\_mutexs(mutex) = \varnothing$

        grd008: $finished\_core3(core) = TRUE$

        grd201: $part \in dom(current\_partition\_flag) \land current\_partition = part \land current\_partition\_flag(part) = TRUE$

        grd203: $current\_processes\_flag(core) = TRUE$

    then

        act001: $mutex\_state(mutex) := MUTEX\_OWNED$

        act002: $mutex\_of\_process(mutex) := proc$

        act003: $acquire\_mutex(core) := mutex$

        act005: $finished\_core3(core) := FALSE$

        act004: $location\_of\_service3(core) := Acquire\_Mutex \mapsto loc\_i$

    end

Event acquire_mutex_lock_count ⟨ordinary⟩ $\widehat{=}$

extends acquire_mutex_lock_count

    any

        part

        core

        mutex

        count

    where

        grd001: $part = current\_partition$

        grd002: $core \in CORES \land core \in dom(acquire\_mutex) \land core \in dom(location\_of\_service3)$

        grd003: $mutex \in mutexs$

        grd004: $mutex\_state(mutex) = MUTEX\_OWNED$

        grd005: $processes\_waiting for\_mutexs(mutex) = \varnothing$

        grd009: $count = mutex\_of\_count(mutex) + 1$

        grd010: $mutex = acquire\_mutex(core)$

        grd006: $finished\_core2(core) = FALSE$

        grd007: $location\_of\_service3(core) = Acquire\_Mutex \mapsto loc\_i$

        grd008: $\neg(finished\_core2(core) = FALSE \land location\_of\_service3(core) = Acquire\_Mutex \mapsto loc\_i)$

    then

        act001: $mutex\_of\_count(mutex) := count$

        act002: $location\_of\_service3(core) := Acquire\_Mutex \mapsto loc\_1$

    end

Event acquire_mutex_retain_priority ⟨ordinary⟩ $\widehat{=}$

extends acquire_mutex_retain_priority

    any

        part
        core
        proc
        mutex
        pri

where

  grd001: $part = current\_partition$
  grd002: $core \in CORES \land core \in dom(acquire\_mutex) \land core \in dom(location\_of\_service3)$
  grd003: $mutex \in mutexs$
  grd004: $mutex\_state(mutex) = MUTEX\_OWNED$
  grd005: $mutex = acquire\_mutex(core)$
  grd006: $processes\_waitingfor\_mutexs(mutex) = \varnothing$
  grd007: $proc = mutex\_of\_process(mutex)$
  grd008: $pri = currentpriority\_of\_process(proc)$
  grd009: $finished\_core2(core) = FALSE$
  grd010: $location\_of\_service3(core) = Acquire\_Mutex \mapsto loc\_1$
  grd011: $\neg(finished\_core3(core) = FALSE \land location\_of\_service3(core) = Acquire\_Mutex \mapsto loc\_1)$

then

  act001: $retainedpriority\_of\_process(proc) := pri$
  act002: $location\_of\_service3(core) := Acquire\_Mutex \mapsto loc\_2$

end

Event acquire_mutex_current_priority ⟨ordinary⟩ $\widehat{=}$

extends acquire_mutex_current_priority

any

        part
        core
        proc
        mutex
        pri

where

  grd001: $part = current\_partition$
  grd002: $core \in CORES \land core \in dom(acquire\_mutex) \land core \in dom(location\_of\_service3)$
  grd003: $mutex \in mutexs$
  grd004: $mutex\_state(mutex) = MUTEX\_OWNED$
  grd005: $mutex = acquire\_mutex(core)$
  grd006: $processes\_waitingfor\_mutexs(mutex) = \varnothing$
  grd007: $proc = mutex\_of\_process(mutex)$
  grd008: $pri = priority\_of\_mutex(mutex)$
  grd009: $finished\_core3(core) = FALSE$
  grd010: $location\_of\_service3(core) = Acquire\_Mutex \mapsto loc\_2$
  grd011: $\neg(finished\_core3(core) = FALSE \land location\_of\_service3(core) = Acquire\_Mutex \mapsto loc\_2)$

then

  act001: $currentpriority\_of\_process(proc) := pri$
  act002: $location\_of\_service3(core) := Acquire\_Mutex \mapsto loc\_3$

end

Event acquire_mutex_return ⟨ordinary⟩ $\widehat{=}$

extends acquire_mutex_return

any

        part
        core

where

  grd001: $part = current\_partition$
  grd002: $core \in CORES \land core \in dom(acquire\_mutex) \land core \in dom(location\_of\_service3)$
  grd003: $finished\_core3(core) = FALSE$
  grd004: $location\_of\_service3(core) = Acquire\_Mutex \mapsto loc\_3$
  grd005: $\neg(finished\_core3(core) = FALSE \land location\_of\_service3(core) = Acquire\_Mutex \mapsto loc\_3)$

> **then**
>> act001: $acquire\_mutex := \{core\} \lhd acquire\_mutex$
>> act002: $finished\_core3(core) := TRUE$
>> act003: $location\_of\_service3(core) := Acquire\_Mutex \mapsto loc\_r$
> **end**

Event release_mutex_init ⟨ordinary⟩ ≙

extends release_mutex_init

> **any**
>> $part$
>> $core$
>> $mutex$
>> $proc$
>> $count$
> **where**
>> grd001: $part = current\_partition$
>> grd002: $core \in CORES$
>> grd003: $mutex \in mutexs$
>> grd004: $proc \in processes$
>> grd005: $mutex\_state(mutex) = MUTEX\_OWNED$
>> grd006: $mutex \in dom(mutex\_of\_process)$
>> grd007: $proc = mutex\_of\_process(mutex)$
>> grd008: $mutex\_of\_count(mutex) \geq 1$
>> grd010: $count = mutex\_of\_count(mutex) - 1$
>> grd009: $finished\_core3(core) = TRUE$
>> grd201: $part \in dom(current\_partition\_flag) \land current\_partition = part \land current\_partition\_flag(part) = TRUE$
>> grd203: $current\_processes\_flag(core) = TRUE$
> **then**
>> act001: $mutex\_of\_count(mutex) := count$
>> act002: $release\_mutex(core) := mutex$
>> act003: $finished\_core3(core) := FALSE$
>> act004: $location\_of\_service3(core) := Release\_Mutex \mapsto loc\_i$
> **end**

Event release_mutex_avail ⟨ordinary⟩ ≙

extends release_mutex_avail

> **any**
>> $part$
>> $core$
>> $mutex$
>> $proc$
>> pri
> **where**
>> grd001: $part = current\_partition$
>> grd002: $core \in CORES \land core \in dom(release\_mutex) \land core \in dom(location\_of\_service3)$
>> grd003: $mutex \in mutexs$
>> grd004: $proc \in processes$
>> grd006: $mutex = release\_mutex(core)$
>> grd005: $mutex\_state(mutex) = MUTEX\_OWNED$
>> grd007: $proc = mutex\_of\_process(mutex)$
>> grd008: $mutex\_of\_count(mutex) = 0$
>> grd009: $pri = retainedpriority\_of\_process(proc)$
>> grd010: $finished\_core3(core) = FALSE$
>> grd011: $location\_of\_service3(core) = Release\_Mutex \mapsto loc\_i$
>> grd012: $\neg(finished\_core3(core) = FALSE \land location\_of\_service3(core) = Release\_Mutex \mapsto loc\_i)$
> **then**
>> act001: $mutex\_state(mutex) := MUTEX\_AVAILABLE$
>> act002: $currentpriority\_of\_process(proc) := pri$
>> act003: $mutex\_of\_process := \{mutex\} \lhd mutex\_of\_process$

     act004: $location\_of\_service3(core) := Release\_Mutex \mapsto loc\_1$

   end

Event release_mutex_return ⟨ordinary⟩ $\widehat{=}$

extends release_mutex_return

   any

     core

     part

   where

     grd001: $part = current\_partition$

     grd002: $core \in CORES \land core \in dom(location\_of\_service3)$

     grd003: $finished\_core3(core) = FALSE$

     grd004: $location\_of\_service3(core) = Release\_Mutex \mapsto loc\_1$

     grd005: $\neg(finished\_core3(core) = FALSE \land location\_of\_service3(core) = Release\_Mutex \mapsto loc\_1)$

   then

     act001: $release\_mutex := \{core\} \lhd release\_mutex$

     act002: $finished\_core3(core) := TRUE$

     act003: $location\_of\_service3(core) := Release\_Mutex \mapsto loc\_r$

   end

Event reset_mutex_init ⟨ordinary⟩ $\widehat{=}$

extends reset_mutex_init

   any

     part

     core

     mutex

     proc

   where

     grd001: $part = current\_partition$

     grd002: $core \in CORES$

     grd003: $mutex \in mutexs$

     grd004: $mutex \in dom(mutex\_of\_process)$

     grd005: $proc = mutex\_of\_process(mutex)$

     grd006: $finished\_core3(core) = TRUE$

     grd201: $part \in dom(current\_partition\_flag) \land current\_partition = part \land current\_partition\_flag(part) = TRUE$

     grd203: $current\_processes\_flag(core) = TRUE$

   then

     act001: $mutex\_of\_count(mutex) := 0$

     act004: $reset\_mutex(core) := mutex$

     act002: $finished\_core3(core) := FALSE$

     act003: $location\_of\_service3(core) := Reset\_Mutex \mapsto loc\_i$

   end

Event reset_mutex_avail ⟨ordinary⟩ $\widehat{=}$

extends reset_mutex_avail

   any

     part

     core

     mutex

     proc

     pri

   where

     grd001: $part = current\_partition$

     grd002: $core \in CORES \land core \in dom(reset\_mutex) \land core \in dom(location\_of\_service3)$

     grd003: $mutex \in mutexs$

     grd004: $proc \in processes$

     grd005: $mutex = reset\_mutex(core)$

     grd006: $mutex\_state(mutex) = MUTEX\_AVAILABLE$

     grd007: $proc = mutex\_of\_process(mutex)$

grd008: $mutex\_of\_count(mutex) = 0$

grd009: $pri = retainedpriority\_of\_process(proc)$

grd010: $finished\_core3(core) = FALSE$

grd011: $location\_of\_service3(core) = Reset\_Mutex \mapsto loc\_i$

grd012: $\neg(finished\_core3(core) = FALSE \wedge location\_of\_service3(core) = Reset\_Mutex \mapsto loc\_i)$

then

act001: $mutex\_state(mutex) := MUTEX\_AVAILABLE$

act002: $currentpriority\_of\_process(proc) := pri$

act003: $mutex\_of\_process := \{mutex\} \lhd mutex\_of\_process$

act004: $location\_of\_service3(core) := Reset\_Mutex \mapsto loc\_1$

end

Event reset_mutex_return ⟨ordinary⟩ $\widehat{=}$

extends reset_mutex_return

    any

        part

        core

    where

grd001: $part = current\_partition$

grd002: $core \in CORES \wedge core \in dom(location\_of\_service3)$

grd003: $finished\_core3(core) = FALSE$

grd004: $location\_of\_service3(core) = Reset\_Mutex \mapsto loc\_1$

grd005: $\neg(finished\_core3(core) = FALSE \wedge location\_of\_service3(core) = Reset\_Mutex \mapsto loc\_i)$

    then

act001: $reset\_mutex := \{core\} \lhd reset\_mutex$

act002: $finished\_core3(core) := TRUE$

act003: $location\_of\_service3(core) := Reset\_Mutex \mapsto loc\_r$

    end

Event get_mutex_id ⟨ordinary⟩ $\widehat{=}$

    any

        part

        mutex

        core

    where

grd001: $part \in dom(current\_partition\_flag) \wedge current\_partition = part \wedge current\_partition\_flag(part) = TRUE$

grd003: $mutex \in mutexs$

grd004: $core \in CORES$

grd005: $finished\_core2(core) = TRUE$

    then

skip

    end

Event get_mutex_status ⟨ordinary⟩ $\widehat{=}$

    any

        part

        mutex

        core

    where

grd001: $part \in dom(current\_partition\_flag) \wedge current\_partition = part \wedge current\_partition\_flag(part) = TRUE$

grd003: $mutex \in mutexs$

grd004: $core \in CORES$

grd005: $finished\_core2(core) = TRUE$

    then

skip

    end

Event get_process_mutex_status ⟨ordinary⟩ $\widehat{=}$

    any

        part

        mutex

        core

    where

        grd001: $part \in dom(current\_partition\_flag) \wedge current\_partition = part \wedge current\_partition\_flag(part) = TRUE$

        grd003: $mutex \in mutexs$

        grd004: $core \in CORES$

        grd005: $finished\_core2(core) = TRUE$

    then

        skip

    end

Event ticktock ⟨ordinary⟩ ≙

extends ticktock

    begin

        act001: $clock\_tick := clock\_tick + 1$

        act002: $need\_reschedule := TRUE$

    end

Event partition_schedule ⟨ordinary⟩ ≙

extends partition_schedule

    any

        part

    where

        grd001: $part \in PARTITIONS$

        grd002: $partition\_mode(part) = PM\_NORMAL \vee partition\_mode(part) = PM\_COLD\_START \vee partition\_mode(part) = PM\_WARM\_START$

        grd101: $need\_reschedule = TRUE$

        grd102: $\exists offset, dur \cdot part\_sched\_list(partition2num(part)) = (offset \mapsto dur) \wedge clock\_tick \bmod majorFrame \geq offset \wedge clock\_tick \bmod majorFrame < offset + dur$

    then

        act101: $need\_reschedule := FALSE$

        act102: $current\_partition := part$

        act103: $need\_procresch := need\_procresch \Leftarrow (Cores\_of\_Partition(part) \times \{TRUE\})$

    end

Event process_schedule ⟨ordinary⟩ ≙

extends process_schedule

    any

        part

        proc

        core

        errproc

    where

        grd001: $part \in PARTITIONS$

        grd002: $proc \in processes \cap dom(process\_state) \cap dom(processes\_of\_cores) \cap dom(processes\_of\_partition)$

        grd003: $core \in CORES$

        grd004: $processes\_of\_partition(proc) = part$

        grd005: $core \in Cores\_of\_Partition(part)$

        grd006: $processes\_of\_cores(proc) = core$

        grd007: $partition\_mode(part) = PM\_NORMAL$

        grd008: $process\_state(proc) = PS\_Ready \vee process\_state(proc) = PS\_Running$

        grd208: $errproc \in processes$

        grd210: $part \in dom(errorhandler\_of\_partition)$

        grd209: $errorhandler\_of\_partition(part) = errproc$

        grd212: $core \in ran(processes\_of\_cores)$

        grd213: $core \in dom(need\_procresch)$

        grd206: $proc \in dom(currentpriority\_of\_process)$

        grd207: $part \in dom(locklevel\_of\_partition)$

        grd211: $proc \in ran(errorhandler\_of\_partition)$

grd201: $need\_procresch(core) = TRUE$

grd202: $part \in dom(current\_partition\_flag) \land current\_partition = part \land current\_partition\_flag(part) = TRUE$

grd203: $(current\_partition \notin dom(errorhandler\_of\_partition) \lor process\_state(errproc) = PS\_Dormant) \land locklevel\_of\_partition(current\_partition) = 0$

grd204: $\forall p \cdot (p \in processes\_of\_partition^{-1}[\{part\}] \land p \in dom(currentpriority\_of\_process) \Rightarrow currentpriority\_of\_process(p) \leq currentpriority\_of\_process(proc))$

then

act201: $process\_state := (process\_state \Leftarrow \{current\_processes(core) \mapsto PS\_Ready\}) \Leftarrow \{proc \mapsto PS\_Running\}$

act202: $current\_processes(core) := proc$

act203: $current\_processes\_flag(core) := TRUE$

act204: $need\_reschedule := FALSE$

act205: $need\_procresch(core) := FALSE$

end

Event get_partition_status ⟨ordinary⟩ $\widehat{=}$

extends get_partition_status

any

    part

    core

where

grd001: $part \in PARTITIONS$

grd002: $part \in dom(current\_partition\_flag) \land current\_partition = part \land current\_partition\_flag(part) = TRUE$

grd003: $core \in CORES$

grd004: $finished\_core(core) = TRUE$

then

    skip

end

Event set_partition_mode_to_idle ⟨ordinary⟩ $\widehat{=}$

extends set_partition_mode_to_idle

any

    part

    newm

    procs

    cores

where

grd001: $part \in PARTITIONS$

grd002: $newm \in PARTITION\_MODES$

grd101: $procs = processes\_of\_partition^{-1}[\{part\}]$

grd102: $cores \in \mathbb{P}_1(CORES)$

grd103: $partition\_mode(part) = PM\_COLD\_START \lor partition\_mode(part) = PM\_WARM\_START \lor partition\_mode(part) = PM\_NORMAL$

grd104: $newm = PM\_IDLE$

grd105: $cores = Cores\_of\_Partition(part)$

grd106: $\forall core \cdot (core \in (Cores\_of\_Partition(part) \cap dom(finished\_core)) \Rightarrow finished\_core(core) = TRUE)$

grd202: $\forall core \cdot (core \in cores \land core \in dom(current\_processes) \land core \in dom(current\_processes\_flag))$

grd203: $current\_partition \in dom(current\_partition\_flag)$

grd201: $part \in dom(current\_partition\_flag) \land current\_partition = part \land current\_partition\_flag(part) = TRUE$

then

act001: $partition\_mode(part) := newm$

act101: $processes := processes \setminus procs$

act102: $process\_state := procs \vartriangleleft process\_state$

act103: $processes\_of\_partition := procs \vartriangleleft processes\_of\_partition$

act104: $processes\_of\_cores := procs \vartriangleleft processes\_of\_cores$

act201: $periodtype\_of\_process := procs \vartriangleleft periodtype\_of\_process$

act301: $process\_wait\_type := procs \lhd process\_wait\_type$

act302: $locklevel\_of\_partition(part) := 1$

act303: $basepriority\_of\_process := procs \lhd basepriority\_of\_process$

act304: $currentpriority\_of\_process := procs \lhd currentpriority\_of\_process$

act305: $retainedpriority\_of\_process := procs \lhd retainedpriority\_of\_process$

act306: $period\_of\_process := procs \lhd period\_of\_process$

act307: $timecapacity\_of\_process := procs \lhd timecapacity\_of\_process$

act308: $deadline\_of\_process := procs \lhd deadline\_of\_process$

act309: $deadlinetime\_of\_process := procs \lhd deadlinetime\_of\_process$

act310: $releasepoint\_of\_process := procs \lhd releasepoint\_of\_process$

act311: $delaytime\_of\_process := procs \lhd delaytime\_of\_process$

act312: $current\_partition\_flag(part) := FALSE$

act313: $current\_processes\_flag := current\_processes\_flag \lhd (cores \times \{FALSE\})$

act314: $preempter\_of\_partition := \{part\} \lhd preempter\_of\_partition$

act315: $preemption\_lock\_mutex := procs \lhd preemption\_lock\_mutex$

act316: $timeout\_trigger := procs \lhd timeout\_trigger$

act317: $errorhandler\_of\_partition := \{part\} \lhd errorhandler\_of\_partition$

act318: $process\_call\_errorhandler := procs \lhd process\_call\_errorhandler$

act319: $setnorm\_wait\_procs := cores \lhd setnorm\_wait\_procs$

act320: $setnorm\_susp\_procs := cores \lhd setnorm\_susp\_procs$

act321: $set\_priority\_parm := cores \lhd set\_priority\_parm$

act322: $suspend\_self\_timeout := cores \lhd suspend\_self\_timeout$

act323: $suspend\_self\_waitproc := cores \lhd suspend\_self\_waitproc$

act324: $resume\_proc := cores \lhd resume\_proc$

act325: $stop\_self\_proc := cores \lhd stop\_self\_proc$

act326: $stop\_proc := cores \lhd stop\_proc$

act327: $start\_aperiod\_proc := cores \lhd start\_aperiod\_proc$

act328: $start\_aperiod\_innormal\_proc := cores \lhd start\_aperiod\_innormal\_proc$

act329: $start\_period\_instart\_proc := cores \lhd start\_period\_instart\_proc$

act330: $start\_period\_innormal\_proc := cores \lhd start\_period\_innormal\_proc$

act331: $delay\_start\_ainstart\_proc := cores \lhd delay\_start\_ainstart\_proc$

act332: $delay\_start\_ainnormal\_proc := cores \lhd delay\_start\_ainnormal\_proc$

act333: $delay\_start\_ainnormal\_delaytime := cores \lhd delay\_start\_ainnormal\_delaytime$

act334: $delay\_start\_instart\_proc := cores \lhd delay\_start\_instart\_proc$

act335: $delay\_start\_innormal\_proc := cores \lhd delay\_start\_innormal\_proc$

act336: $delay\_start\_innormal\_delaytime := cores \lhd delay\_start\_innormal\_delaytime$

act337: $req\_busy\_resource\_proc := cores \lhd req\_busy\_resource\_proc$

act338: $resource\_become\_avail\_proc := cores \lhd resource\_become\_avail\_proc$

act339: $resource\_become\_avail2 := cores \lhd resource\_become\_avail2$

act340: $time\_wait\_proc := cores \lhd time\_wait\_proc$

act341: $period\_wait\_proc := cores \lhd period\_wait\_proc$

act401: $queuing\_ports := queuing\_ports \setminus Ports\_of\_Partition^{-1}[\{part\}]$

act402: $sampling\_ports := sampling\_ports \setminus Ports\_of\_Partition^{-1}[\{part\}]$

act403: $msgspace\_of\_samplingports := Ports\_of\_Partition^{-1}[\{part\}] \lhd msgspace\_of\_samplingports$

act404: $queue\_of\_queuingports := Ports\_of\_Partition^{-1}[\{part\}] \lhd queue\_of\_queuingports$

act406: $processes\_waitingfor\_queuingports := Ports\_of\_Partition^{-1}[\{part\}] \lhd processes\_waitingfor\_queuingports$

act405: $buffers := buffers \setminus buffers\_of\_partition^{-1}[\{part\}]$

act407: $MaxMsgNum\_of\_Buffers := buffers\_of\_partition^{-1}[\{part\}] \lhd MaxMsgNum\_of\_Buffers$

act408: $queue\_of\_buffers := buffers\_of\_partition^{-1}[\{part\}] \lhd queue\_of\_buffers$

act409: $processes\_waitingfor\_buffers := buffers\_of\_partition^{-1}[\{part\}] \lhd processes\_waitingfor\_buffers$

act410: $blackboards := blackboards \setminus blackboards\_of\_partition^{-1}[\{part\}]$

act411: $msgspace\_of\_blackboards := blackboards\_of\_partition^{-1}[\{part\}] \lhd msgspace\_of\_blackboards$

act413: $emptyindicator\_of\_blackboards := blackboards\_of\_partition^{-1}[\{part\}] \lhd emptyindicator\_of\_blackboards$

act414: $processes\_waitingfor\_blackboards := blackboards\_of\_partition^{-1}[\{part\}] \lhd processes\_waitingfor\_blackboard$

act412: $semaphores := semaphores \setminus semaphores\_of\_partition^{-1}[\{part\}]$

act415: $MaxValue\_of\_Semaphores := semaphores\_of\_partition^{-1}[\{part\}] \lhd MaxValue\_of\_Semaphores$

act416: $value\_of\_semaphores := semaphores\_of\_partition^{-1}[\{part\}] \lhd value\_of\_semaphores$

act417: $processes\_waitingfor\_semaphores := semaphores\_of\_partition^{-1}[\{part\}] \lhd processes\_waitingfor\_semaphor$

act418: $events := events \setminus events\_of\_partition^{-1}[\{part\}]$

act419: $state\_of\_events := events\_of\_partition^{-1}[\{part\}] \lhd state\_of\_events$

act420: $processes\_waitingfor\_events := events\_of\_partition^{-1}[\{part\}] \lhd processes\_waitingfor\_events$

act421: $buffers\_of\_partition := buffers\_of\_partition \rhd \{part\}$

act422: $blackboards\_of\_partition := blackboards\_of\_partition \rhd \{part\}$

act423: $semaphores\_of\_partition := semaphores\_of\_partition \rhd \{part\}$

act424: $events\_of\_partition := events\_of\_partition \rhd \{part\}$

act438: $send\_queuing\_message\_port := cores \lhd send\_queuing\_message\_port$

act425: $wakeup\_waitproc\_on\_srcqueports\_port := cores \lhd wakeup\_waitproc\_on\_srcqueports\_port$

act426: $wakeup\_waitproc\_on\_dstqueports\_port := cores \lhd wakeup\_waitproc\_on\_dstqueports\_port$

act427: $receive\_queuing\_message\_port := cores \lhd receive\_queuing\_message\_port$

act428: $send\_buffer\_needwakeup := cores \lhd send\_buffer\_needwakeup$

act429: $send\_buffer\_withfull := cores \lhd send\_buffer\_withfull$

act430: $receive\_buffer\_needwake := cores \lhd receive\_buffer\_needwake$

act431: $receive\_buffer\_whenempty := cores \lhd receive\_buffer\_whenempty$

act432: $display\_blackboard\_needwake := cores \lhd display\_blackboard\_needwake$

act433: $read\_blackboard\_whenempty := cores \lhd read\_blackboard\_whenempty$

act434: $wait\_semaphore\_whenzero := cores \lhd wait\_semaphore\_whenzero$

act435: $signal\_semaphore\_needwake := cores \lhd signal\_semaphore\_needwake$

act436: $set\_event\_needwake := cores \lhd set\_event\_needwake$

act437: $wait\_event\_whendown := cores \lhd wait\_event\_whendown$

act501: $RefreshPeriod\_of\_SamplingPorts := Ports\_of\_Partition^{-1}[\{part\}] \lhd RefreshPeriod\_of\_SamplingPorts$

act502: $needtrans\_of\_sourcesamplingport := Ports\_of\_Partition^{-1}[\{part\}] \lhd needtrans\_of\_sourcesamplingport$

act503: $quediscipline\_of\_queuingports := Ports\_of\_Partition^{-1}[\{part\}] \lhd quediscipline\_of\_queuingports$

act504: $quediscipline\_of\_buffers := buffers\_of\_partition^{-1}[\{part\}] \lhd quediscipline\_of\_buffers$

act505: $quediscipline\_of\_semaphores := semaphores\_of\_partition^{-1}[\{part\}] \lhd quediscipline\_of\_semaphores$

end

Event set_partition_mode_to_coldstart ⟨ordinary⟩ ≙

extends set_partition_mode_to_coldstart

any

    part

    newm

    procs

    cores

where

grd001: $part \in PARTITIONS$

grd002: $newm \in PARTITION\_MODES$

grd101: $cores \in \mathbb{P}_1(CORES)$

grd102: $newm = PM\_COLD\_START$

grd103: $partition\_mode(part) = PM\_COLD\_START \lor partition\_mode(part) = PM\_WARM\_START \lor$
$partition\_mode(part) = PM\_NORMAL$

grd107: $part \in ran(processes\_of\_partition)$

grd104: $procs = processes\_of\_partition^{-1}[\{part\}]$

grd105: $cores = Cores\_of\_Partition(part)$

grd106: $\forall core \cdot (core \in (Cores\_of\_Partition(part) \cap dom(finished\_core)) \Rightarrow finished\_core(core) = TRUE)$

grd202: $\forall core \cdot (core \in cores \land core \in dom(current\_processes) \land core \in dom(current\_processes\_flag))$

grd201: $current\_partition \in dom(current\_partition\_flag)$

grd203: $part \in dom(current\_partition\_flag) \land current\_partition = part \land current\_partition\_flag(part) = TRUE$

then

act001: $partition\_mode(part) := newm$

act101: $processes := processes \setminus procs$

act102: $process\_state := procs \ntriangleleft process\_state$

act103: $processes\_of\_partition := procs \ntriangleleft processes\_of\_partition$

act104: $processes\_of\_cores := procs \ntriangleleft processes\_of\_cores$

act201: $periodtype\_of\_process := procs \ntriangleleft periodtype\_of\_process$

act301: $process\_wait\_type := procs \ntriangleleft process\_wait\_type$

act302: $locklevel\_of\_partition(part) := 1$

act303: $basepriority\_of\_process := procs \ntriangleleft basepriority\_of\_process$

act304: $currentpriority\_of\_process := procs \ntriangleleft currentpriority\_of\_process$

act305: $retainedpriority\_of\_process := procs \ntriangleleft retainedpriority\_of\_process$

act306: $period\_of\_process := procs \ntriangleleft period\_of\_process$

act307: $timecapacity\_of\_process := procs \ntriangleleft timecapacity\_of\_process$

act308: $deadline\_of\_process := procs \ntriangleleft deadline\_of\_process$

act309: $deadlinetime\_of\_process := procs \ntriangleleft deadlinetime\_of\_process$

act310: $releasepoint\_of\_process := procs \ntriangleleft releasepoint\_of\_process$

act311: $delaytime\_of\_process := procs \ntriangleleft delaytime\_of\_process$

act312: $current\_processes\_flag := current\_processes\_flag \ovelay (cores \times \{FALSE\})$

act313: $preempter\_of\_partition := \{part\} \ntriangleleft preempter\_of\_partition$

act314: $preemption\_lock\_mutex := procs \ntriangleleft preemption\_lock\_mutex$

act315: $timeout\_trigger := procs \ntriangleleft timeout\_trigger$

act316: $errorhandler\_of\_partition := \{part\} \ntriangleleft errorhandler\_of\_partition$

act317: $process\_call\_errorhandler := procs \ntriangleleft process\_call\_errorhandler$

act318: $setnorm\_wait\_procs := cores \ntriangleleft setnorm\_wait\_procs$

act319: $setnorm\_susp\_procs := cores \ntriangleleft setnorm\_susp\_procs$

act320: $set\_priority\_parm := cores \ntriangleleft set\_priority\_parm$

act321: $suspend\_self\_timeout := cores \ntriangleleft suspend\_self\_timeout$

act322: $suspend\_self\_waitproc := cores \ntriangleleft suspend\_self\_waitproc$

act323: $resume\_proc := cores \ntriangleleft resume\_proc$

act324: $stop\_self\_proc := cores \ntriangleleft stop\_self\_proc$

act325: $stop\_proc := cores \ntriangleleft stop\_proc$

act326: $start\_aperiod\_proc := cores \ntriangleleft start\_aperiod\_proc$

act327: $start\_aperiod\_innormal\_proc := cores \ntriangleleft start\_aperiod\_innormal\_proc$

act328: $start\_period\_instart\_proc := cores \ntriangleleft start\_period\_instart\_proc$

act329: $start\_period\_innormal\_proc := cores \ntriangleleft start\_period\_innormal\_proc$

act330: $delay\_start\_ainstart\_proc := cores \ntriangleleft delay\_start\_ainstart\_proc$

act331: $delay\_start\_ainnormal\_proc := cores \ntriangleleft delay\_start\_ainnormal\_proc$

act332: $delay\_start\_ainnormal\_delaytime := cores \ntriangleleft delay\_start\_ainnormal\_delaytime$

act333: $delay\_start\_instart\_proc := cores \ntriangleleft delay\_start\_instart\_proc$

act334: $delay\_start\_innormal\_proc := cores \ntriangleleft delay\_start\_innormal\_proc$

act335: $delay\_start\_innormal\_delaytime := cores \ntriangleleft delay\_start\_innormal\_delaytime$

act336: $req\_busy\_resource\_proc := cores \ntriangleleft req\_busy\_resource\_proc$

act337: $resource\_become\_avail\_proc := cores \ntriangleleft resource\_become\_avail\_proc$

act338: $resource\_become\_avail2 := cores \ntriangleleft resource\_become\_avail2$

act339: $time\_wait\_proc := cores \ntriangleleft time\_wait\_proc$

act340: $period\_wait\_proc := cores \ntriangleleft period\_wait\_proc$

act401: $queuing\_ports := queuing\_ports \setminus Ports\_of\_Partition^{-1}[\{part\}]$

act402: $sampling\_ports := sampling\_ports \setminus Ports\_of\_Partition^{-1}[\{part\}]$

act403: $msgspace\_of\_samplingports := Ports\_of\_Partition^{-1}[\{part\}] \ntriangleleft msgspace\_of\_samplingports$

act404: $queue\_of\_queuingports := Ports\_of\_Partition^{-1}[\{part\}] \ntriangleleft queue\_of\_queuingports$

act405: $processes\_waitingfor\_queuingports := Ports\_of\_Partition^{-1}[\{part\}] \ntriangleleft processes\_waitingfor\_queuingports$

act406: $buffers := buffers \setminus buffers\_of\_partition^{-1}[\{part\}]$

act407: $MaxMsgNum\_of\_Buffers := buffers\_of\_partition^{-1}[\{part\}] \mathbin{\lhd\mkern-14mu-} MaxMsgNum\_of\_Buffers$

act408: $queue\_of\_buffers := buffers\_of\_partition^{-1}[\{part\}] \mathbin{\lhd\mkern-14mu-} queue\_of\_buffers$

act409: $processes\_waitingfor\_buffers := buffers\_of\_partition^{-1}[\{part\}] \mathbin{\lhd\mkern-14mu-} processes\_waitingfor\_buffers$

act410: $blackboards := blackboards \setminus blackboards\_of\_partition^{-1}[\{part\}]$

act411: $msgspace\_of\_blackboards := blackboards\_of\_partition^{-1}[\{part\}] \mathbin{\lhd\mkern-14mu-} msgspace\_of\_blackboards$

act412: $emptyindicator\_of\_blackboards := blackboards\_of\_partition^{-1}[\{part\}] \mathbin{\lhd\mkern-14mu-} emptyindicator\_of\_blackboards$

act413: $processes\_waitingfor\_blackboards := blackboards\_of\_partition^{-1}[\{part\}] \mathbin{\lhd\mkern-14mu-} processes\_waitingfor\_blackboard$

act414: $semaphores := semaphores \setminus semaphores\_of\_partition^{-1}[\{part\}]$

act415: $MaxValue\_of\_Semaphores := semaphores\_of\_partition^{-1}[\{part\}] \mathbin{\lhd\mkern-14mu-} MaxValue\_of\_Semaphores$

act416: $value\_of\_semaphores := semaphores\_of\_partition^{-1}[\{part\}] \mathbin{\lhd\mkern-14mu-} value\_of\_semaphores$

act417: $processes\_waitingfor\_semaphores := semaphores\_of\_partition^{-1}[\{part\}] \mathbin{\lhd\mkern-14mu-} processes\_waitingfor\_semaphor$

act418: $events := events \setminus events\_of\_partition^{-1}[\{part\}]$

act419: $state\_of\_events := events\_of\_partition^{-1}[\{part\}] \mathbin{\lhd\mkern-14mu-} state\_of\_events$

act420: $processes\_waitingfor\_events := events\_of\_partition^{-1}[\{part\}] \mathbin{\lhd\mkern-14mu-} processes\_waitingfor\_events$

act421: $buffers\_of\_partition := buffers\_of\_partition \mathbin{\rhd} \{part\}$

act422: $blackboards\_of\_partition := blackboards\_of\_partition \mathbin{\rhd} \{part\}$

act423: $semaphores\_of\_partition := semaphores\_of\_partition \mathbin{\rhd} \{part\}$

act424: $events\_of\_partition := events\_of\_partition \mathbin{\rhd} \{part\}$

act438: $send\_queuing\_message\_port := cores \mathbin{\lhd\mkern-14mu-} send\_queuing\_message\_port$

act425: $wakeup\_waitproc\_on\_srcqueports\_port := cores \mathbin{\lhd\mkern-14mu-} wakeup\_waitproc\_on\_srcqueports\_port$

act426: $wakeup\_waitproc\_on\_dstqueports\_port := cores \mathbin{\lhd\mkern-14mu-} wakeup\_waitproc\_on\_dstqueports\_port$

act427: $receive\_queuing\_message\_port := cores \mathbin{\lhd\mkern-14mu-} receive\_queuing\_message\_port$

act428: $send\_buffer\_needwakeup := cores \mathbin{\lhd\mkern-14mu-} send\_buffer\_needwakeup$

act429: $send\_buffer\_withfull := cores \mathbin{\lhd\mkern-14mu-} send\_buffer\_withfull$

act430: $receive\_buffer\_needwake := cores \mathbin{\lhd\mkern-14mu-} receive\_buffer\_needwake$

act431: $receive\_buffer\_whenempty := cores \mathbin{\lhd\mkern-14mu-} receive\_buffer\_whenempty$

act432: $display\_blackboard\_needwake := cores \mathbin{\lhd\mkern-14mu-} display\_blackboard\_needwake$

act433: $read\_blackboard\_whenempty := cores \mathbin{\lhd\mkern-14mu-} read\_blackboard\_whenempty$

act434: $wait\_semaphore\_whenzero := cores \mathbin{\lhd\mkern-14mu-} wait\_semaphore\_whenzero$

act435: $signal\_semaphore\_needwake := cores \mathbin{\lhd\mkern-14mu-} signal\_semaphore\_needwake$

act436: $set\_event\_needwake := cores \mathbin{\lhd\mkern-14mu-} set\_event\_needwake$

act437: $wait\_event\_whendown := cores \mathbin{\lhd\mkern-14mu-} wait\_event\_whendown$

act501: $RefreshPeriod\_of\_SamplingPorts := Ports\_of\_Partition^{-1}[\{part\}] \mathbin{\lhd\mkern-14mu-} RefreshPeriod\_of\_SamplingPorts$

act502: $needtrans\_of\_sourcesamplingport := Ports\_of\_Partition^{-1}[\{part\}] \mathbin{\lhd\mkern-14mu-} needtrans\_of\_sourcesamplingport$

act503: $quediscipline\_of\_queuingports := Ports\_of\_Partition^{-1}[\{part\}] \mathbin{\lhd\mkern-14mu-} quediscipline\_of\_queuingports$

act504: $quediscipline\_of\_buffers := buffers\_of\_partition^{-1}[\{part\}] \mathbin{\lhd\mkern-14mu-} quediscipline\_of\_buffers$

act505: $quediscipline\_of\_semaphores := semaphores\_of\_partition^{-1}[\{part\}] \mathbin{\lhd\mkern-14mu-} quediscipline\_of\_semaphores$

end

Event coldstart_partition_from_idle ⟨ordinary⟩ ≙

extends coldstart_partition_from_idle

  any

    part

    newm

    cores

  where

    grd001: $part \in PARTITIONS$

    grd002: $newm \in PARTITION\_MODES$

        grd101: $cores \in \mathbb{P}_1(CORES)$

        grd102: $newm = PM\_COLD\_START$

        grd103: $partition\_mode(part) = PM\_IDLE$

        grd104: $cores = Cores\_of\_Partition(part)$

        grd105: $\forall core \cdot (core \in (Cores\_of\_Partition(part) \cap dom(finished\_core)) \Rightarrow finished\_core(core) = TRUE)$

then

        act001: $partition\_mode(part) := newm$

        act201: $locklevel\_of\_partition(part) := 1$

end

Event set_partition_mode_to_warmstart ⟨ordinary⟩ ≙

extends set_partition_mode_to_warmstart

any

        part

        newm

        procs

        cores

where

        grd001: $part \in PARTITIONS$

        grd002: $newm \in PARTITION\_MODES$

        grd101: $cores \in \mathbb{P}_1(CORES)$

        grd102: $newm = PM\_WARM\_START$

        grd103: $partition\_mode(part) = PM\_WARM\_START \lor partition\_mode(part) = PM\_NORMAL$

        grd104: $procs = processes\_of\_partition^{-1}[\{part\}]$

        grd105: $cores = Cores\_of\_Partition(part)$

        grd106: $\forall core \cdot (core \in (Cores\_of\_Partition(part) \cap dom(finished\_core)) \Rightarrow finished\_core(core) = TRUE)$

        grd203: $\forall core \cdot (core \in cores \land core \in dom(current\_processes) \land core \in dom(current\_processes\_flag))$

        grd201: $current\_partition \in dom(current\_partition\_flag)$

        grd202: $part \in dom(current\_partition\_flag) \land current\_partition = part \land current\_partition\_flag(part) = TRUE$

then

        act001: $partition\_mode(part) := newm$

        act101: $processes := processes \setminus procs$

        act102: $process\_state := procs \lhd process\_state$

        act103: $processes\_of\_partition := procs \lhd processes\_of\_partition$

        act104: $processes\_of\_cores := procs \lhd processes\_of\_cores$

        act201: $periodtype\_of\_process := procs \lhd periodtype\_of\_process$

        act301: $process\_wait\_type := procs \lhd process\_wait\_type$

        act302: $locklevel\_of\_partition(part) := 1$

        act303: $basepriority\_of\_process := procs \lhd basepriority\_of\_process$

        act304: $currentpriority\_of\_process := procs \lhd currentpriority\_of\_process$

        act305: $retainedpriority\_of\_process := procs \lhd retainedpriority\_of\_process$

        act306: $period\_of\_process := procs \lhd period\_of\_process$

        act307: $timecapacity\_of\_process := procs \lhd timecapacity\_of\_process$

        act308: $deadline\_of\_process := procs \lhd deadline\_of\_process$

        act309: $deadlinetime\_of\_process := procs \lhd deadlinetime\_of\_process$

        act310: $releasepoint\_of\_process := procs \lhd releasepoint\_of\_process$

        act311: $delaytime\_of\_process := procs \lhd delaytime\_of\_process$

        act312: $current\_processes\_flag := current\_processes\_flag \lhdplus (cores \times \{FALSE\})$

        act313: $preempter\_of\_partition := \{part\} \lhd preempter\_of\_partition$

        act314: $preemption\_lock\_mutex := procs \lhd preemption\_lock\_mutex$

        act315: $timeout\_trigger := procs \lhd timeout\_trigger$

        act316: $errorhandler\_of\_partition := \{part\} \lhd errorhandler\_of\_partition$

        act317: $process\_call\_errorhandler := procs \lhd process\_call\_errorhandler$

        act318: $setnorm\_wait\_procs := cores \lhd setnorm\_wait\_procs$

        act319: $setnorm\_susp\_procs := cores \lhd setnorm\_susp\_procs$

        act320: $set\_priority\_parm := cores \lhd set\_priority\_parm$

act321: $suspend\_self\_timeout := cores \lhd suspend\_self\_timeout$

act322: $suspend\_self\_waitproc := cores \lhd suspend\_self\_waitproc$

act323: $resume\_proc := cores \lhd resume\_proc$

act324: $stop\_self\_proc := cores \lhd stop\_self\_proc$

act325: $stop\_proc := cores \lhd stop\_proc$

act326: $start\_aperiod\_proc := cores \lhd start\_aperiod\_proc$

act327: $start\_aperiod\_innormal\_proc := cores \lhd start\_aperiod\_innormal\_proc$

act328: $start\_period\_instart\_proc := cores \lhd start\_period\_instart\_proc$

act329: $start\_period\_innormal\_proc := cores \lhd start\_period\_innormal\_proc$

act330: $delay\_start\_ainstart\_proc := cores \lhd delay\_start\_ainstart\_proc$

act331: $delay\_start\_ainnormal\_proc := cores \lhd delay\_start\_ainnormal\_proc$

act332: $delay\_start\_ainnormal\_delaytime := cores \lhd delay\_start\_ainnormal\_delaytime$

act333: $delay\_start\_instart\_proc := cores \lhd delay\_start\_instart\_proc$

act334: $delay\_start\_innormal\_proc := cores \lhd delay\_start\_innormal\_proc$

act335: $delay\_start\_innormal\_delaytime := cores \lhd delay\_start\_innormal\_delaytime$

act336: $req\_busy\_resource\_proc := cores \lhd req\_busy\_resource\_proc$

act337: $resource\_become\_avail\_proc := cores \lhd resource\_become\_avail\_proc$

act338: $resource\_become\_avail2 := cores \lhd resource\_become\_avail2$

act339: $time\_wait\_proc := cores \lhd time\_wait\_proc$

act340: $period\_wait\_proc := cores \lhd period\_wait\_proc$

act401: $queuing\_ports := queuing\_ports \setminus Ports\_of\_Partition^{-1}[\{part\}]$

act402: $sampling\_ports := sampling\_ports \setminus Ports\_of\_Partition^{-1}[\{part\}]$

act403: $msgspace\_of\_samplingports := Ports\_of\_Partition^{-1}[\{part\}] \ndres msgspace\_of\_samplingports$

act404: $queue\_of\_queuingports := Ports\_of\_Partition^{-1}[\{part\}] \ndres queue\_of\_queuingports$

act405: $processes\_waitingfor\_queuingports := Ports\_of\_Partition^{-1}[\{part\}] \ndres processes\_waitingfor\_queuingports$

act406: $buffers := buffers \setminus buffers\_of\_partition^{-1}[\{part\}]$

act407: $MaxMsgNum\_of\_Buffers := buffers\_of\_partition^{-1}[\{part\}] \ndres MaxMsgNum\_of\_Buffers$

act408: $queue\_of\_buffers := buffers\_of\_partition^{-1}[\{part\}] \ndres queue\_of\_buffers$

act409: $processes\_waitingfor\_buffers := buffers\_of\_partition^{-1}[\{part\}] \ndres processes\_waitingfor\_buffers$

act410: $blackboards := blackboards \setminus blackboards\_of\_partition^{-1}[\{part\}]$

act411: $msgspace\_of\_blackboards := blackboards\_of\_partition^{-1}[\{part\}] \ndres msgspace\_of\_blackboards$

act412: $emptyindicator\_of\_blackboards := blackboards\_of\_partition^{-1}[\{part\}] \ndres emptyindicator\_of\_blackboards$

act413: $processes\_waitingfor\_blackboards := blackboards\_of\_partition^{-1}[\{part\}] \ndres processes\_waitingfor\_blackboards$

act414: $semaphores := semaphores \setminus semaphores\_of\_partition^{-1}[\{part\}]$

act415: $MaxValue\_of\_Semaphores := semaphores\_of\_partition^{-1}[\{part\}] \ndres MaxValue\_of\_Semaphores$

act416: $value\_of\_semaphores := semaphores\_of\_partition^{-1}[\{part\}] \ndres value\_of\_semaphores$

act417: $processes\_waitingfor\_semaphores := semaphores\_of\_partition^{-1}[\{part\}] \ndres processes\_waitingfor\_semaphores$

act418: $events := events \setminus events\_of\_partition^{-1}[\{part\}]$

act419: $state\_of\_events := events\_of\_partition^{-1}[\{part\}] \ndres state\_of\_events$

act420: $processes\_waitingfor\_events := events\_of\_partition^{-1}[\{part\}] \ndres processes\_waitingfor\_events$

act421: $buffers\_of\_partition := buffers\_of\_partition \rhd \{part\}$

act422: $blackboards\_of\_partition := blackboards\_of\_partition \rhd \{part\}$

act423: $semaphores\_of\_partition := semaphores\_of\_partition \rhd \{part\}$

act424: $events\_of\_partition := events\_of\_partition \rhd \{part\}$

act438: $send\_queuing\_message\_port := cores \lhd send\_queuing\_message\_port$

act425: $wakeup\_waitproc\_on\_srcqueports\_port := cores \lhd wakeup\_waitproc\_on\_srcqueports\_port$

act426: $wakeup\_waitproc\_on\_dstqueports\_port := cores \lhd wakeup\_waitproc\_on\_dstqueports\_port$

act427: $receive\_queuing\_message\_port := cores \lhd receive\_queuing\_message\_port$

act428: $send\_buffer\_needwakeup := cores \lhd send\_buffer\_needwakeup$

act429: $send\_buffer\_withfull := cores \lhd send\_buffer\_withfull$

act430: $receive\_buffer\_needwake := cores \lhd receive\_buffer\_needwake$

act431: $receive\_buffer\_whenempty := cores \lhd receive\_buffer\_whenempty$

act432: $display\_blackboard\_needwake := cores \lhd display\_blackboard\_needwake$

act433: $read\_blackboard\_whenempty := cores \lhd read\_blackboard\_whenempty$

act434: $wait\_semaphore\_whenzero := cores \lhd wait\_semaphore\_whenzero$

act435: $signal\_semaphore\_needwake := cores \lhd signal\_semaphore\_needwake$

act436: $set\_event\_needwake := cores \lhd set\_event\_needwake$

act437: $wait\_event\_whendown := cores \lhd wait\_event\_whendown$

act501: $RefreshPeriod\_of\_SamplingPorts := Ports\_of\_Partition^{-1}[\{part\}] \lhd RefreshPeriod\_of\_SamplingPorts$

act502: $needtrans\_of\_sourcesamplingport := Ports\_of\_Partition^{-1}[\{part\}] \lhd needtrans\_of\_sourcesamplingport$

act503: $quediscipline\_of\_queuingports := Ports\_of\_Partition^{-1}[\{part\}] \lhd quediscipline\_of\_queuingports$

act504: $quediscipline\_of\_buffers := buffers\_of\_partition^{-1}[\{part\}] \lhd quediscipline\_of\_buffers$

act505: $quediscipline\_of\_semaphores := semaphores\_of\_partition^{-1}[\{part\}] \lhd quediscipline\_of\_semaphores$

end

Event warmstart_partition_from_idle ⟨ordinary⟩ $\widehat{=}$

extends warmstart_partition_from_idle

    any

        part

        newm

        cores

    where

        grd001: $part \in PARTITIONS$

        grd002: $newm \in PARTITION\_MODES$

        grd101: $cores \in \mathbb{P}_1(CORES)$

        grd102: $newm = PM\_WARM\_START$

        grd103: $partition\_mode(part) = PM\_IDLE$

        grd104: $cores = Cores\_of\_Partition(part)$

        grd105: $\forall core \cdot (core \in (Cores\_of\_Partition(part) \cap dom(finished\_core)) \Rightarrow finished\_core(core) = TRUE)$

    then

        act001: $partition\_mode(part) := newm$

        act201: $locklevel\_of\_partition(part) := 1$

    end

Event set_partition_mode_to_normal_init' ⟨ordinary⟩ $\widehat{=}$

extends set_partition_mode_to_normal_init'

    any

        part

        core

        service

    where

        grd001: $part \in PARTITIONS$

        grd002: $core \in CORES$

        grd003: $service \in Services$

        grd004: $partition\_mode(part) = PM\_COLD\_START \lor partition\_mode(part) = PM\_WARM\_START$

        grd005: $finished\_core(core) = TRUE$

        grd006: $service = Set\_Normal$

        grd201: $part \in dom(current\_partition\_flag) \land current\_partition = part \land current\_partition\_flag(part) = TRUE$

    then

        act001: $location\_of\_service(core) := service \mapsto loc\_i$

        act002: $finished\_core(core) := FALSE$

        act201: $location\_of\_service2(core) := service \mapsto loc\_i$

    end

Event set_partition_mode_to_normal_mode' ⟨ordinary⟩ ≙
extends set_partition_mode_to_normal_mode'
    any
        part
        newm
        core
    where
        grd001: $part \in PARTITIONS$
        grd002: $newm \in PARTITION\_MODES$
        grd101: $core \in CORES \cap dom(location\_of\_service)$
        grd102: $newm = PM\_NORMAL$
        grd103: $finite(processes\_of\_partition^{-1}[\{part\}]) \land card(processes\_of\_partition^{-1}[\{part\}]) > 0$
        grd104: $partition\_mode(part) = PM\_COLD\_START \lor partition\_mode(part) = PM\_WARM\_START$

        grd105: $location\_of\_service(core) = Set\_Normal \mapsto loc\_i$
        grd106: $finished\_core(core) = FALSE$
        grd107: $\neg(location\_of\_service(core) = Set\_Normal \mapsto loc\_i \land finished\_core(core) = FALSE)$
        grd201: $location\_of\_service2(core) = Set\_Normal \mapsto loc\_i$
        grd202: $\neg(location\_of\_service2(core) = Set\_Normal \mapsto loc\_i \land finished\_core(core) = FALSE)$
        grd203: $current\_partition = part \land current\_partition\_flag(part) = TRUE$
    then
        act001: $location\_of\_service(core) := Set\_Normal \mapsto loc\_1$
        act002: $partition\_mode(part) := newm$
        act201: $location\_of\_service2(core) := Set\_Normal \mapsto loc\_1$
    end

Event set_partition_mode_to_normal_ready'_and_fst_point ⟨ordinary⟩ ≙
extends set_partition_mode_to_normal_ready'_and_fst_point
    any
        part
        procs
        procs2
        procsstate
        core
        nrlt
        stperprocs
        dstperprocs
        staperprocs
        dstaperprocs
    where
        grd001: $part \in PARTITIONS$
        grd002: $partition\_mode(part) = PM\_NORMAL$
        grd003: $procs = processes\_of\_partition^{-1}[\{part\}] \cap process\_state^{-1}[\{PS\_Waiting\}]$
        grd004: $procs2 = processes\_of\_partition^{-1}[\{part\}] \cap process\_state^{-1}[\{PS\_WaitandSuspend\}]$
        grd005: $procsstate \in procs \rightarrow \{PS\_Waiting, PS\_Ready\}$
        grd006: $core \in CORES \cap dom(location\_of\_service)$
        grd007: $location\_of\_service(core) = Set\_Normal \mapsto loc\_1$
        grd008: $finished\_core(core) = FALSE$
        grd009: $\neg(location\_of\_service(core) = Set\_Normal \mapsto loc\_1 \land finished\_core(core) = FALSE)$
        grd201: $current\_partition = part \land current\_partition\_flag(part) = TRUE$
        grd202: $part \in ran(processes\_of\_partition)$
        grd203: $stperprocs = (procs \backslash period\_of\_process^{-1}[\{INFINITE\_TIME\_VALUE\}]) \cap process\_wait\_type^{-1}[\{PROC$

        grd204: $dstperprocs = (procs \backslash period\_of\_process^{-1}[\{INFINITE\_TIME\_VALUE\}]) \cap process\_wait\_type^{-1}[\{PROC$

        grd205: $staperprocs = procs \cap period\_of\_process^{-1}[\{INFINITE\_TIME\_VALUE\}] \cap process\_wait\_type^{-1}[\{PROC$

        grd206: $dstaperprocs = procs \cap period\_of\_process^{-1}[\{INFINITE\_TIME\_VALUE\}] \cap process\_wait\_type^{-1}[\{PROC$

        grd207: $nrlt \in stperprocs \rightarrow \mathbb{N}$

grd208: $\forall p, x, y, b \cdot (p \in stperprocs \wedge ((x \mapsto y) \mapsto b) = firstperiodicprocstart\_timeWindow\_of\_Partition(part) \Rightarrow$
$\qquad nrlt(p) = ((clock\_tick * ONE\_TICK\_TIME)/majorFrame + 1) * majorFrame + x)$

grd209: $procsstate = (staperprocs \times \{PS\_Ready\}) \cup ((dstaperprocs \cup stperprocs \cup dstperprocs) \times$
$\qquad \{PS\_Waiting\})$

grd210: $location\_of\_service2(core) = Set\_Normal \mapsto loc\_1$

grd211: $\neg(location\_of\_service2(core) = Set\_Normal \mapsto loc\_1 \wedge finished\_core(core) = FALSE)$

then

act001: $location\_of\_service(core) := Set\_Normal \mapsto loc\_2$

act002: $process\_state := (process\_state \Leftarrow procsstate) \Leftarrow (procs2 \times \{PS\_Suspend\})$

act201: $location\_of\_service2(core) := Set\_Normal \mapsto loc\_2$

act202: $setnorm\_wait\_procs(core) := procs$

act203: $setnorm\_susp\_procs(core) := procs2$

act204: $releasepoint\_of\_process := releasepoint\_of\_process \Leftarrow nrlt$

end

Event set_partition_mode_to_normal_release_point_and_frstpoint2 ⟨ordinary⟩ $\widehat{=}$

extends set_partition_mode_to_normal_release_point_and_frstpoint2

any

> *part*
> *core*
> *procs*
> *rlt*
> *nrlt*
> *dstperprocs*
> *dstaperprocs*

where

grd001: $part \in PARTITIONS$

grd002: $partition\_mode(part) = PM\_NORMAL$

grd003: $core \in CORES$

grd004: $core \in dom(setnorm\_wait\_procs) \wedge procs = setnorm\_wait\_procs(core)$

grd006: $core \in dom(location\_of\_service2) \wedge location\_of\_service2(core) = Set\_Normal \mapsto loc\_2$

grd007: $finished\_core(core) = FALSE$

grd008: $\neg(location\_of\_service2(core) = Set\_Normal \mapsto loc\_2 \wedge finished\_core(core) = FALSE)$

grd009: $current\_partition = part \wedge current\_partition\_flag(part) = TRUE$

grd010: $dstperprocs = (procs \backslash period\_of\_process^{-1}[\{INFINITE\_TIME\_VALUE\}]) \cap process\_wait\_type^{-1}[\{PROC$

grd011: $dstaperprocs = procs \cap period\_of\_process^{-1}[\{INFINITE\_TIME\_VALUE\}] \cap process\_wait\_type^{-1}[\{PROC$

grd012: $rlt \in dstaperprocs \rightarrow \mathbb{N}$

grd013: $\forall p \cdot (p \in dstaperprocs \Rightarrow rlt(p) = clock\_tick * ONE\_TICK\_TIME + delaytime\_of\_process(p))$

grd014: $nrlt \in dstperprocs \rightarrow \mathbb{N}$

grd015: $\forall p, x, y, b \cdot (p \in dstperprocs \wedge ((x \mapsto y) \mapsto b) = firstperiodicprocstart\_timeWindow\_of\_Partition(part) \Rightarrow$
$\qquad nrlt(p) = ((clock\_tick * ONE\_TICK\_TIME)/majorFrame + 1) * majorFrame + x + delaytime\_of\_process(p))$

then

act001: $location\_of\_service2(core) := Set\_Normal \mapsto loc\_3$

act002: $releasepoint\_of\_process := releasepoint\_of\_process \Leftarrow rlt \Leftarrow nrlt$

end

Event set_partition_mode_to_normal_deadlinetime ⟨ordinary⟩ $\widehat{=}$

extends set_partition_mode_to_normal_deadlinetime

any

> *part*
> *core*
> *procs*
> *staperprocs*
> *dstaperprocs*
> *suspaperprocs*
> *stperprocs*
> *dstperprocs*

         dl1
         dl2
         dl3
         dl4

where

grd001: $part \in PARTITIONS$

grd002: $partition\_mode(part) = PM\_NORMAL$

grd003: $core \in CORES$

grd004: $core \in dom(setnorm\_wait\_procs) \land procs = setnorm\_wait\_procs(core)$

grd005: $core \in dom(setnorm\_susp\_procs) \land suspaperprocs = setnorm\_susp\_procs(core)$

grd006: $staperprocs = procs \cap period\_of\_process^{-1}[\{INFINITE\_TIME\_VALUE\}] \cap process\_wait\_type^{-1}[\{PROC$

grd007: $dstaperprocs = procs \cap period\_of\_process^{-1}[\{INFINITE\_TIME\_VALUE\}] \cap process\_wait\_type^{-1}[\{PROC$

grd008: $stperprocs = (procs \backslash period\_of\_process^{-1}[\{INFINITE\_TIME\_VALUE\}]) \cap process\_wait\_type^{-1}[\{PROC$

grd009: $dstperprocs = (procs \backslash period\_of\_process^{-1}[\{INFINITE\_TIME\_VALUE\}]) \cap process\_wait\_type^{-1}[\{PROC$

grd010: $dl1 \in staperprocs \cup suspaperprocs \rightarrow \mathbb{N}$

grd011: $\forall p \cdot (p \in staperprocs \cup suspaperprocs \land p \in dom(timecapacity\_of\_process) \Rightarrow dl1(p) = clock\_tick * ONE\_TICK\_TIME + timecapacity\_of\_process(p))$

grd012: $dl2 \in dstaperprocs \rightarrow \mathbb{N}$

grd013: $\forall p \cdot (p \in dstaperprocs \land p \in dom(delaytime\_of\_process) \land p \in dom(timecapacity\_of\_process) \Rightarrow dl2(p) = clock\_tick * ONE\_TICK\_TIME + delaytime\_of\_process(p) + timecapacity\_of\_process(p))$

grd014: $dl3 \in stperprocs \rightarrow \mathbb{N}$

grd015: $\forall p \cdot (p \in stperprocs \land p \in dom(timecapacity\_of\_process) \Rightarrow dl3(p) = clock\_tick * ONE\_TICK\_TIME + timecapacity\_of\_process(p))$

grd016: $dl4 \in dstperprocs \rightarrow \mathbb{N}$

grd017: $\forall p \cdot (p \in dstperprocs \land p \in dom(delaytime\_of\_process) \land p \in dom(timecapacity\_of\_process) \Rightarrow dl4(p) = clock\_tick * ONE\_TICK\_TIME + delaytime\_of\_process(p) + timecapacity\_of\_process(p))$

grd018: $core \in dom(location\_of\_service2) \land location\_of\_service2(core) = Set\_Normal \mapsto loc\_3$

grd019: $finished\_core(core) = FALSE$

grd020: $\neg(location\_of\_service2(core) = Set\_Normal \mapsto loc\_3 \land finished\_core(core) = FALSE)$

then

act001: $location\_of\_service2(core) := Set\_Normal \mapsto loc\_4$

act002: $deadlinetime\_of\_process := deadlinetime\_of\_process \Leftarrow dl1 \Leftarrow dl2 \Leftarrow dl3 \Leftarrow dl4$

end

Event set_partition_mode_to_normal_locklevel ⟨ordinary⟩ $\widehat{=}$

extends set_partition_mode_to_normal_locklevel

any

         part
         core

where

grd001: $part \in PARTITIONS$

grd002: $partition\_mode(part) = PM\_NORMAL$

grd003: $core \in CORES$

grd004: $core \in dom(location\_of\_service2) \land location\_of\_service2(core) = Set\_Normal \mapsto loc\_4$

grd005: $finished\_core(core) = FALSE$

grd006: $\neg(location\_of\_service2(core) = Set\_Normal \mapsto loc\_4 \land finished\_core(core) = FALSE)$

then

act001: $location\_of\_service2(core) := Set\_Normal \mapsto loc\_5$

act002: $locklevel\_of\_partition(part) := 0$

act003: $preempter\_of\_partition := \{part\} \vartriangleleft preempter\_of\_partition$

act004: $timeout\_trigger := (processes\_of\_partition^{-1}[\{part\}]) \vartriangleleft timeout\_trigger$

end

Event set_partition_mode_to_normal_return' ⟨ordinary⟩ $\widehat{=}$

extends set_partition_mode_to_normal_return'

    any
        part
        core
    where
        grd001: $part \in PARTITIONS$
        grd002: $partition\_mode(part) = PM\_NORMAL$
        grd003: $core \in CORES \cap dom(location\_of\_service)$
        grd004: $location\_of\_service(core) = Set\_Normal \mapsto loc\_2$
        grd005: $finished\_core(core) = FALSE$
        grd006: $\neg(location\_of\_service(core) = Set\_Normal \mapsto loc\_2 \wedge finished\_core(core) = FALSE)$
    then
        act001: $location\_of\_service(core) := Set\_Normal \mapsto loc\_r$
        act002: $finished\_core(core) := TRUE$
    end

Event get_process_id ⟨ordinary⟩ $\widehat{=}$

extends get_process_id

    any
        proc
        core
    where
        grd001: $proc \in processes$
        grd002: $proc \in dom(processes\_of\_partition) \wedge processes\_of\_partition(proc) = current\_partition$
        grd003: $current\_partition \in dom(current\_partition\_flag) \wedge current\_partition\_flag(current\_partition) = TRUE$
        grd004: $core \in CORES$
        grd005: $finished\_core(core) = TRUE$
    then
        skip
    end

Event get_process_status ⟨ordinary⟩ $\widehat{=}$

extends get_process_status

    any
        proc
        core
    where
        grd001: $proc \in processes$
        grd002: $proc \in dom(processes\_of\_partition) \wedge processes\_of\_partition(proc) = current\_partition$
        grd003: $current\_partition \in dom(current\_partition\_flag) \wedge current\_partition\_flag(current\_partition) = TRUE$
        grd004: $core \in CORES$
        grd005: $finished\_core(core) = TRUE$
    then
        skip
    end

Event create_process_init ⟨ordinary⟩ $\widehat{=}$

extends create_process_init

    any
        part
        proc
        core
        service
        ptype
        period
        timecapacity
        basepriority
        dl
    where
        grd001: $part \in PARTITIONS$

        grd002: $proc \in (PROCESSES \setminus processes)$
        grd003: $core \in CORES$
        grd004: $service \in Services$
        grd005: $partition\_mode(part) = PM\_COLD\_START \lor partition\_mode(part) = PM\_WARM\_START$

        grd006: $finished\_core(core) = TRUE$
        grd007: $service = Create\_Process$
        grd101: $ptype \in PROC\_PERIOD\_TYPE$
        grd201: $current\_partition = part$
        grd202: $part \in dom(current\_partition\_flag) \land current\_partition\_flag(part) = TRUE$
        grd203: $period \in \mathbb{N}$
        grd204: $timecapacity \in \mathbb{N}$
        grd205: $basepriority \in MIN\_PRIORITY .. MAX\_PRIORITY$
        grd206: $dl \in DEADLINE\_TYPE$
        grd207: $part \in dom(Period\_of\_Partition) \land period \neq INFINITE\_TIME\_VALUE \Rightarrow (\exists n \cdot (n \in \mathbb{N} \land period = n * Period\_of\_Partition(part)))$
        grd208: $period \neq INFINITE\_TIME\_VALUE \Rightarrow (timecapacity \leq period)$
        grd209: $(ptype = APERIOD\_PROC \Leftrightarrow period = INFINITE\_TIME\_VALUE)$
        grd210: $(ptype = PERIOD\_PROC \Leftrightarrow period > 0)$

then
        act001: $location\_of\_service(core) := service \mapsto loc\_i$
        act002: $finished\_core(core) := FALSE$
        act003: $processes := processes \cup \{proc\}$
        act004: $processes\_of\_partition(proc) := part$
        act005: $create\_process\_parm(core) := proc$
        act101: $periodtype\_of\_process(proc) := ptype$
        act201: $period\_of\_process(proc) := period$
        act202: $timecapacity\_of\_process(proc) := timecapacity$
        act203: $basepriority\_of\_process(proc) := basepriority$
        act204: $deadline\_of\_process(proc) := dl$
        act205: $currentpriority\_of\_process(proc) := basepriority$
        act206: $retainedpriority\_of\_process(proc) := basepriority$
        act207: $preemption\_lock\_mutex(proc) := FALSE$
end

Event create_process_dormant $\langle$ordinary$\rangle$ $\widehat{=}$
extends create_process_dormant
    any
        part
        proc
        core
    where
        grd001: $part \in PARTITIONS$
        grd002: $proc \in processes$
        grd003: $core \in CORES \cap dom(location\_of\_service)$
        grd004: $location\_of\_service(core) = Create\_Process \mapsto loc\_i$
        grd005: $finished\_core(core) = FALSE$
        grd006: $\neg(location\_of\_service(core) = Create\_Process \mapsto loc\_i \land finished\_core(core) = FALSE)$
        grd007: $proc = create\_process\_parm(core)$
        grd008: $processes\_of\_partition(proc) = part$
        grd009: $partition\_mode(part) = PM\_COLD\_START \lor partition\_mode(part) = PM\_WARM\_START$

        grd201: $current\_partition = part$
        grd202: $current\_partition\_flag(part) = TRUE$
    then
        act001: $location\_of\_service(core) := Create\_Process \mapsto loc\_1$
        act002: $process\_state(proc) := PS\_Dormant$
    end

Event create_process_core $\langle$ordinary$\rangle$ $\widehat{=}$
extends create_process_core

any

    part

    proc

    core

where

    grd001: $part \in PARTITIONS$

    grd002: $proc \in processes$

    grd003: $core \in CORES \cap dom(location\_of\_service)$

    grd004: $location\_of\_service(core) = Create\_Process \mapsto loc\_1$

    grd005: $finished\_core(core) = FALSE$

    grd006: $\neg(location\_of\_service(core) = Create\_Process \mapsto loc\_1 \wedge finished\_core(core) = FALSE)$

    grd007: $processes\_of\_partition(proc) = part$

    grd008: $process\_state(proc) = PS\_Dormant$

    grd009: $create\_process\_parm(core) = proc$

    grd010: $partition\_mode(part) = PM\_COLD\_START \vee partition\_mode(part) = PM\_WARM\_START$

    grd201: $current\_partition = part$

    grd202: $current\_partition\_flag(part) = TRUE$

then

    act001: $location\_of\_service(core) := Create\_Process \mapsto loc\_2$

    act002: $processes\_of\_cores(proc) := core$

end

Event create_process_return $\langle ordinary \rangle \,\widehat{=}$

extends create_process_return

any

    part

    proc

    core

where

    grd001: $part \in PARTITIONS$

    grd002: $proc \in processes$

    grd003: $core \in CORES \cap dom(location\_of\_service)$

    grd004: $location\_of\_service(core) = Create\_Process \mapsto loc\_2$

    grd005: $finished\_core(core) = FALSE$

    grd006: $\neg(location\_of\_service(core) = Create\_Process \mapsto loc\_2 \wedge finished\_core(core) = FALSE)$

    grd007: $processes\_of\_partition(proc) = part$

    grd008: $process\_state(proc) = PS\_Dormant$

    grd009: $create\_process\_parm(core) = proc$

    grd010: $partition\_mode(part) = PM\_COLD\_START \vee partition\_mode(part) = PM\_WARM\_START$

    grd201: $current\_partition = part$

    grd202: $current\_partition\_flag(part) = TRUE$

then

    act001: $location\_of\_service(core) := Create\_Process \mapsto loc\_r$

    act002: $finished\_core(core) := TRUE$

    act003: $create\_process\_parm := \{core\} \lhd create\_process\_parm$

end

Event set_priority_init $\langle ordinary \rangle \,\widehat{=}$

extends set_priority_init

any

    part

    proc

    core

    pri

where

    grd001: $part \in PARTITIONS$

    grd002: $current\_partition = part$

    grd003: $part \in dom(current\_partition\_flag) \wedge current\_partition\_flag(part) = TRUE$

    grd004: $proc \in processes$

$\qquad$ grd005: $\quad core \in CORES$
$\qquad$ grd006: $\quad finished\_core2(core) = TRUE$
$\qquad$ grd007: $\quad proc \in dom(process\_state) \land process\_state(proc) \neq PS\_Dormant$
$\qquad$ grd008: $\quad proc \in processes\_of\_partition^{-1}[\{part\}]$
$\qquad$ grd009: $\quad pri \in MIN\_PRIORITY \mathinner{..} MAX\_PRIORITY$

then
$\qquad$ act001: $location\_of\_service2(core) := Set\_Priority \mapsto loc\_i$
$\qquad$ act002: $finished\_core2(core) := FALSE$
$\qquad$ act003: $set\_priority\_parm(core) := pri$

end

Event set_priority_owned_preemption $\langle ordinary \rangle \;\widehat{=}$

extends set_priority_owned_preemption

$\quad$ **any**
$\qquad$ *part*
$\qquad$ *proc*
$\qquad$ *core*

$\quad$ **where**
$\qquad$ grd001: $\quad part \in PARTITIONS$
$\qquad$ grd002: $\quad current\_partition = part$
$\qquad$ grd003: $\quad part \in dom(current\_partition\_flag) \land current\_partition\_flag(part) = TRUE$
$\qquad$ grd004: $\quad proc \in processes$
$\qquad$ grd005: $\quad core \in CORES \cap dom(set\_priority\_parm)$
$\qquad$ grd006: $\quad finished\_core2(core) = FALSE$
$\qquad$ grd007: $\quad core \in dom(location\_of\_service2) \land location\_of\_service2(core) = Set\_Priority \mapsto loc\_i$
$\qquad$ grd008: $\quad \neg(location\_of\_service2(core) = Set\_Priority \mapsto loc\_i \land finished\_core2(core) = FALSE)$
$\qquad$ grd009: $\quad process\_state(proc) \neq PS\_Dormant$
$\qquad$ grd010: $\quad preemption\_lock\_mutex(proc) = TRUE$
$\qquad\qquad$ owned a mutex

$\quad$ **then**
$\qquad$ act001: $location\_of\_service2(core) := Set\_Priority \mapsto loc\_1$
$\qquad$ act002: $retainedpriority\_of\_process(proc) := set\_priority\_parm(core)$

$\quad$ **end**

Event set_priority_notowned_preemption $\langle ordinary \rangle \;\widehat{=}$

extends set_priority_notowned_preemption

$\quad$ **any**
$\qquad$ *part*
$\qquad$ *proc*
$\qquad$ *core*

$\quad$ **where**
$\qquad$ grd001: $\quad part \in PARTITIONS$
$\qquad$ grd002: $\quad current\_partition = part$
$\qquad$ grd003: $\quad part \in dom(current\_partition\_flag) \land current\_partition\_flag(part) = TRUE$
$\qquad$ grd004: $\quad proc \in processes$
$\qquad$ grd005: $\quad core \in CORES \cap dom(set\_priority\_parm)$
$\qquad$ grd006: $\quad finished\_core2(core) = FALSE$
$\qquad$ grd007: $\quad core \in dom(location\_of\_service2) \land location\_of\_service2(core) = Set\_Priority \mapsto loc\_i$
$\qquad$ grd008: $\quad \neg(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Set\_Priority \mapsto loc\_i)$
$\qquad$ grd009: $\quad process\_state(proc) \neq PS\_Dormant$
$\qquad$ grd010: $\quad preemption\_lock\_mutex(proc) = FALSE$
$\qquad\qquad$ not owned a mutex

$\quad$ **then**
$\qquad$ act001: $location\_of\_service2(core) := Set\_Priority \mapsto loc\_1$
$\qquad$ act002: $currentpriority\_of\_process(proc) := set\_priority\_parm(core)$

$\quad$ **end**

Event set_priority_check_reschedule $\langle ordinary \rangle \;\widehat{=}$

extends set_priority_check_reschedule

$\quad$ **any**
$\qquad$ *part*

   core
   needproc

where

   grd001: $part \in PARTITIONS$
   grd002: $current\_partition = part$
   grd003: $part \in dom(current\_partition\_flag) \land current\_partition\_flag(part) = TRUE$
   grd004: $core \in CORES$
   grd005: $needproc \in BOOL$
   grd006: $part \in dom(locklevel\_of\_partition) \land locklevel\_of\_partition(part) = 0 \Rightarrow needproc = TRUE$
   grd007: $part \in dom(locklevel\_of\_partition) \land locklevel\_of\_partition(part) \neq 0 \Rightarrow needproc = need\_reschedule$
   grd008: $finished\_core2(core) = FALSE$
   grd009: $core \in dom(location\_of\_service2) \land location\_of\_service2(core) = Set\_Priority \mapsto loc\_1$
   grd010: $\neg(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Set\_Priority \mapsto loc\_1)$

then

   act001: $location\_of\_service2(core) := Set\_Priority \mapsto loc\_2$
   act002: $need\_reschedule := needproc$

end

Event set_priority_return ⟨ordinary⟩ $\widehat{=}$

extends set_priority_return

any

   part
   core
   proc

where

   grd001: $part \in PARTITIONS$
   grd002: $current\_partition = part$
   grd003: $part \in dom(current\_partition\_flag) \land current\_partition\_flag(part) = TRUE$
   grd004: $core \in CORES$
   grd005: $proc \in processes$
   grd006: $proc \in dom(process\_state) \land process\_state(proc) \neq PS\_Dormant$
   grd007: $finished\_core2(core) = FALSE$
   grd008: $core \in dom(location\_of\_service2) \land location\_of\_service2(core) = Set\_Priority \mapsto loc\_2$
   grd009: $\neg(location\_of\_service2(core) = Set\_Priority \mapsto loc\_2 \land finished\_core(core) = FALSE)$

then

   act001: $location\_of\_service2(core) := Set\_Priority \mapsto loc\_r$
   act002: $finished\_core2(core) := TRUE$
   act003: $set\_priority\_parm := \{core\} \lhd set\_priority\_parm$

end

Event suspend_self_init ⟨ordinary⟩ $\widehat{=}$

extends suspend_self_init

any

   part
   proc
   newstate
   core
   timeout

where

   grd001: $part \in PARTITIONS$
   grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state) \cap dom(periodtype\_of\_process) \land proc \in ran(current\_processes)$
   grd003: $newstate \in PROCESS\_STATES$
   grd004: $core \in CORES$
   grd005: $processes\_of\_partition(proc) = part$
   grd017: $finished\_core2(core) = TRUE$
   grd101: $partition\_mode(part) = PM\_NORMAL$
   grd102: $process\_state(proc) = PS\_Running$
   grd103: $newstate = PS\_Suspend$

     grd104: $periodtype\_of\_process(proc) = APERIOD\_PROC$
     grd201: $timeout \in \mathbb{Z} \wedge timeout \neq 0$
     grd202: $part = current\_partition$
     grd211: $core \in current\_processes^{-1}[\{proc\}] \wedge core \in dom(current\_processes\_flag)$
     grd213: $core \in dom(current\_processes)$
     grd209: $part \in dom(current\_partition\_flag)$
     grd214: $current\_partition\_flag(part) = TRUE$
     grd204: $current\_processes\_flag(core) = TRUE$
     grd203: $proc = current\_processes(core)$
     grd205: $part \in dom(errorhandler\_of\_partition) \Rightarrow proc \neq errorhandler\_of\_partition(part)$
     grd210: $part \in dom(locklevel\_of\_partition)$
     grd206: $locklevel\_of\_partition(part) = 0$
     grd212: $proc \in dom(preemption\_lock\_mutex)$
     grd207: $preemption\_lock\_mutex(proc) = FALSE$

then

     act001: $process\_state(proc) := newstate$
     act101: $location\_of\_service2(core) := Suspend\_self \mapsto loc\_i$
     act102: $finished\_core2(core) := FALSE$
     act103: $suspend\_self\_timeout(core) := timeout$
     act104: $suspend\_self\_waitproc(core) := proc$
     act105: $current\_processes\_flag(core) := FALSE$
     act106: $current\_processes := \{core\} \lhd current\_processes$

end

Event suspend_self_timeout ⟨ordinary⟩ $\widehat{=}$

extends suspend_self_timeout

    any

        part
        proc
        core
        timeout
        timeouttrig
        waittype

    where

     grd001: $part \in PARTITIONS$
     grd002: $proc \in processes$
     grd003: $partition\_mode(part) = PM\_NORMAL$
     grd004: $proc \in dom(processes\_of\_partition) \wedge processes\_of\_partition(proc) = part$
     grd005: $core \in CORES$
     grd006: $timeout \in \mathbb{Z} \wedge timeout \neq 0$
     grd007: $core \in dom(suspend\_self\_timeout) \wedge core \in dom(current\_processes\_flag)$
     grd008: $part = current\_partition$
     grd010: $part \in dom(errorhandler\_of\_partition) \Rightarrow proc \neq errorhandler\_of\_partition(part)$
     grd011: $processes\_of\_partition(proc) \in dom(locklevel\_of\_partition) \wedge locklevel\_of\_partition(part) = 0$
     grd012: $finished\_core2(core) = FALSE$
     grd013: $core \in dom(location\_of\_service2) \wedge location\_of\_service2(core) = Suspend\_self \mapsto loc\_i$
     grd014: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Suspend\_self \mapsto loc\_i)$
     grd015: $timeout = suspend\_self\_timeout(core)$
     grd016: $timeouttrig \in processes \nrightarrow (PROCESS\_STATES \times \mathbb{N}_1)$
     grd020: $proc = suspend\_self\_waitproc(core)$
     grd017: $timeout \neq INFINITE\_TIME\_VALUE \wedge timeout \neq 0 \Rightarrow timeouttrig = \{proc \mapsto (PS\_Ready \mapsto (timeout + clock\_tick * ONE\_TICK\_TIME))\}$
     grd018: $timeout = INFINITE\_TIME\_VALUE \Rightarrow timeouttrig = \varnothing$
     grd019: $waittype \in processes \nrightarrow PROCESS\_WAIT\_TYPES$
     grd021: $timeout > 0 \Rightarrow waittype = \{proc \mapsto PROC\_WAIT\_TIMEOUT\}$
     grd022: $(timeout = INFINITE\_TIME\_VALUE \vee timeout = 0) \Rightarrow waittype = \varnothing$

then

     act001: $location\_of\_service2(core) := Suspend\_self \mapsto loc\_1$
     act002: $timeout\_trigger := timeout\_trigger \lhd\mkern-14mu- timeouttrig$

$\qquad$ act003: $process\_wait\_type := process\_wait\_type \lessdot\!\!- waittype$

$\quad$ **end**

Event suspend_self_ask_schedule $\langle ordinary \rangle \,\widehat{=}$

extends suspend_self_ask_schedule

$\quad$ **any**

$\qquad$ *part*

$\qquad$ *core*

$\qquad$ *timeout*

$\qquad$ *needresch*

$\quad$ **where**

$\qquad$ grd001: $part \in PARTITIONS$

$\qquad$ grd002: $part = current\_partition$

$\qquad$ grd003: $partition\_mode(part) = PM\_NORMAL$

$\qquad$ grd004: $core \in CORES \land core \in dom(location\_of\_service2) \land core \in dom(current\_processes\_flag)$

$\qquad$ grd005: $core \in dom(suspend\_self\_timeout)$

$\qquad$ grd007: $timeout \in \mathbb{Z} \land timeout \neq 0$

$\qquad$ grd008: $timeout = suspend\_self\_timeout(core)$

$\qquad$ grd010: $needresch \in BOOL$

$\qquad$ grd012: $(timeout = 0 \Rightarrow needresch = FALSE) \land (timeout > 0 \Rightarrow needresch = TRUE)$

$\qquad$ grd014: $finished\_core2(core) = FALSE$

$\qquad$ grd015: $location\_of\_service2(core) = Suspend\_self \mapsto loc\_1$

$\qquad$ grd016: $\neg(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Suspend\_self \mapsto loc\_1)$

$\quad$ **then**

$\qquad$ act001: $location\_of\_service2(core) := Suspend\_self \mapsto loc\_2$

$\qquad$ act003: $need\_reschedule := needresch$

$\quad$ **end**

Event suspend_self_return $\langle ordinary \rangle \,\widehat{=}$

extends suspend_self_return

$\quad$ **any**

$\qquad$ *part*

$\qquad$ *core*

$\quad$ **where**

$\qquad$ grd001: $part \in PARTITIONS$

$\qquad$ grd002: $part = current\_partition$

$\qquad$ grd003: $partition\_mode(part) = PM\_NORMAL$

$\qquad$ grd004: $core \in CORES \land core \in dom(location\_of\_service2)$

$\qquad$ grd005: $core \in dom(suspend\_self\_timeout) \land core \in dom(suspend\_self\_waitproc)$

$\qquad$ grd006: $finished\_core2(core) = FALSE$

$\qquad$ grd007: $location\_of\_service2(core) = Suspend\_self \mapsto loc\_2$

$\qquad$ grd008: $\neg(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Suspend\_self \mapsto loc\_2)$

$\quad$ **then**

$\qquad$ act001: $location\_of\_service2(core) := Suspend\_self \mapsto loc\_r$

$\qquad$ act002: $finished\_core2(core) := TRUE$

$\qquad$ act003: $suspend\_self\_timeout := \{core\} \lessdot\!\!- suspend\_self\_timeout$

$\qquad$ act004: $suspend\_self\_waitproc := \{core\} \lessdot\!\!- suspend\_self\_waitproc$

$\quad$ **end**

Event suspend $\langle ordinary \rangle \,\widehat{=}$

extends suspend

$\quad$ **any**

$\qquad$ *part*

$\qquad$ *proc*

$\qquad$ *newstate*

$\qquad$ *core*

$\quad$ **where**

$\qquad$ grd001: $part \in PARTITIONS$

$\qquad$ grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state) \cap dom(periodtype\_of\_process)$

        grd003: $newstate \in PROCESS\_STATES$

        grd004: $core \in CORES \land core \in dom(current\_processes\_flag)$

        grd005: $processes\_of\_partition(proc) = part$

        grd006: $partition\_mode(part) = PM\_COLD\_START \lor partition\_mode(part) = PM\_WARM\_START \lor$
           $partition\_mode(part) = PM\_NORMAL$

        grd017: $finished\_core(core) = TRUE$

        grd101: $partition\_mode(part) = PM\_NORMAL \Rightarrow (process\_state(proc) = PS\_Ready \land newstate =$
           $PS\_Suspend) \lor (process\_state(proc) = PS\_Waiting \land newstate = PS\_WaitandSuspend)$

        grd102: $partition\_mode(part) = PM\_COLD\_START \lor partition\_mode(part) = PM\_WARM\_START \Rightarrow$
           $(process\_state(proc) = PS\_Waiting \land newstate = PS\_WaitandSuspend)$

        grd103: $periodtype\_of\_process(proc) = APERIOD\_PROC$

        grd201: $part = current\_partition$

        grd202: $processes\_of\_partition(proc) \in dom(current\_partition\_flag) \land current\_partition\_flag(part) =$
           $TRUE \land current\_processes\_flag(core) = TRUE$

        grd203: $current\_processes\_flag(core) = TRUE \Rightarrow proc \notin ran(current\_processes)$

        grd204: $processes\_of\_partition(proc) \in dom(locklevel\_of\_partition) \land (locklevel\_of\_partition(part) =$
           $0 \lor proc \notin ran(process\_call\_errorhandler))$

        grd205: $proc \in dom(period\_of\_process) \land period\_of\_process(proc) = INFINITE\_TIME\_VALUE$

        grd206: $process\_state(proc) \neq PS\_Dormant$

        grd207: $process\_state(proc) \neq PS\_Suspend \land process\_state(proc) \neq PS\_WaitandSuspend$

        grd208: $proc \in dom(preemption\_lock\_mutex) \land preemption\_lock\_mutex(proc) = FALSE$

        grd209: $process\_state(proc) \neq PS\_Faulted$

then

        act001: $process\_state(proc) := newstate$

end

Event resume_init ⟨ordinary⟩ $\widehat{=}$

extends resume_init

    any

        part

        proc

        newstate

        core

        trigs

    where

        grd001: $part \in PARTITIONS$

        grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state) \cap dom(periodtype\_of\_process)$

        grd003: $newstate \in PROCESS\_STATES$

        grd004: $core \in CORES \land core \in dom(current\_processes\_flag)$

        grd208: $proc \in dom(timeout\_trigger)$

        grd005: $processes\_of\_partition(proc) = part$

        grd006: $partition\_mode(part) = PM\_COLD\_START \lor partition\_mode(part) = PM\_WARM\_START \lor$
           $partition\_mode(part) = PM\_NORMAL$

        grd017: $finished\_core2(core) = TRUE$

        grd101: $partition\_mode(part) = PM\_NORMAL \Rightarrow (process\_state(proc) = PS\_Suspend \land newstate =$
           $PS\_Ready) \lor (process\_state(proc) = PS\_WaitandSuspend \land newstate = PS\_Waiting)$

        grd102: $partition\_mode(part) = PM\_COLD\_START \lor partition\_mode(part) = PM\_WARM\_START \Rightarrow$
           $(process\_state(proc) = PS\_WaitandSuspend \land newstate = PS\_Waiting)$

        grd103: $periodtype\_of\_process(proc) = APERIOD\_PROC$

        grd201: $current\_partition = part$

        grd202: $processes\_of\_partition(proc) \in dom(current\_partition\_flag) \land current\_partition\_flag(part) =$
           $TRUE$

        grd203: $current\_processes\_flag(core) = TRUE \Rightarrow proc \in ran(current\_processes)$

        grd204: $process\_state(proc) \neq PS\_Dormant$

        grd205: $process\_state(proc) = PS\_Suspend \Rightarrow newstate = PS\_Ready$

        grd206: $process\_state(proc) = PS\_WaitandSuspend \Rightarrow newstate = PS\_Waiting$

        grd207: $process\_state(proc) \neq PS\_Faulted$

        grd209: $newstate = PS\_Ready \Rightarrow trigs = \{proc\}$

$\qquad$ **grd210**: $newstate = PS\_Waiting \Rightarrow trigs = \varnothing$

then

$\qquad$ **act001**: $process\_state(proc) := newstate$

$\qquad$ **act201**: $location\_of\_service2(core) := Resume \mapsto loc\_i$

$\qquad$ **act202**: $finished\_core2(core) := FALSE$

$\qquad$ **act203**: $resume\_proc(core) := proc$

$\qquad$ **act204**: $timeout\_trigger := trigs \lhd timeout\_trigger$

end

Event resume_check_reschedule $\langle ordinary \rangle \;\widehat{=}$

extends resume_check_reschedule

$\qquad$ **any**

$\qquad\qquad$ *part*

$\qquad\qquad$ *proc*

$\qquad\qquad$ *core*

$\qquad\qquad$ *reschedule*

$\qquad$ **where**

$\qquad\qquad$ **grd001**: $part \in PARTITIONS$

$\qquad\qquad$ **grd002**: $proc \in processes \land proc \in ran(resume\_proc) \land proc \in dom(processes\_of\_partition)$

$\qquad\qquad$ **grd003**: $core \in CORES \land core \in dom(resume\_proc) \land core \in dom(current\_processes\_flag) \land core \in dom(location\_of\_service2)$

$\qquad\qquad$ **grd004**: $processes\_of\_partition(proc) = part$

$\qquad\qquad$ **grd005**: $current\_partition = part$

$\qquad\qquad$ **grd006**: $processes\_of\_partition(proc) \in dom(current\_partition\_flag) \land current\_partition\_flag(part) = TRUE$

$\qquad\qquad$ **grd014**: $proc = resume\_proc(core)$

$\qquad\qquad$ **grd007**: $reschedule \in BOOL$

$\qquad\qquad$ **grd015**: $resume\_proc(core) \in dom(process\_state) \land processes\_of\_partition(resume\_proc(core)) \in dom(locklevel\_of\_partition)$

$\qquad\qquad$ **grd008**: $locklevel\_of\_partition(part) = 0 \land process\_state(proc) = PS\_Ready \Rightarrow reschedule = TRUE$

$\qquad\qquad$ **grd009**: $(locklevel\_of\_partition(part) > 0) \land (process\_state(proc) = PS\_Waiting \Rightarrow reschedule = need\_reschedule)$

$\qquad\qquad$ **grd010**: $current\_processes\_flag(core) = TRUE \Rightarrow proc \in ran(current\_processes)$

$\qquad\qquad$ **grd011**: $finished\_core2(core) = FALSE$

$\qquad\qquad$ **grd012**: $location\_of\_service2(core) = Resume \mapsto loc\_i$

$\qquad\qquad$ **grd013**: $\neg(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Resume \mapsto loc\_i)$

$\qquad$ **then**

$\qquad\qquad$ **act001**: $location\_of\_service2(core) := Resume \mapsto loc\_1$

$\qquad\qquad$ **act002**: $need\_reschedule := reschedule$

$\qquad$ **end**

Event resume_return $\langle ordinary \rangle \;\widehat{=}$

extends resume_return

$\qquad$ **any**

$\qquad\qquad$ *part*

$\qquad\qquad$ *proc*

$\qquad\qquad$ *core*

$\qquad$ **where**

$\qquad\qquad$ **grd001**: $part \in PARTITIONS$

$\qquad\qquad$ **grd002**: $proc \in processes \land proc \in ran(resume\_proc)$

$\qquad\qquad$ **grd003**: $core \in CORES \land core \in dom(resume\_proc) \land core \in dom(current\_processes\_flag) \land core \in dom(location\_of\_service2)$

$\qquad\qquad$ **grd004**: $proc = resume\_proc(core)$

$\qquad\qquad$ **grd012**: $resume\_proc(core) \in dom(processes\_of\_partition)$

$\qquad\qquad$ **grd005**: $processes\_of\_partition(proc) = part$

$\qquad\qquad$ **grd006**: $part = current\_partition$

$\qquad\qquad$ **grd007**: $processes\_of\_partition(resume\_proc(core)) \in dom(current\_partition\_flag) \land current\_partition\_flag(part) = TRUE$

$\qquad\qquad$ **grd008**: $current\_processes\_flag(core) = TRUE \Rightarrow proc \notin ran(current\_processes)$

$\qquad\qquad$ **grd009**: $finished\_core2(core) = FALSE$

$\quad$ **grd010:** $\quad location\_of\_service2(core) = Resume \mapsto loc\_1$

$\quad$ **grd011:** $\quad \neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Resume \mapsto loc\_1)$

then

$\quad$ **act001:** $location\_of\_service2(core) := Resume \mapsto loc\_r$

$\quad$ **act002:** $finished\_core2(core) := TRUE$

$\quad$ **act003:** $resume\_proc := \{core\} \lhd resume\_proc$

end

Event stop_self_init $\langle ordinary \rangle \; \widehat{=}$

extends stop_self_init

$\quad$ **any**

$\quad\quad part$

$\quad\quad proc$

$\quad\quad newstate$

$\quad\quad core$

$\quad$ **where**

$\quad$ **grd001:** $\quad part \in PARTITIONS$

$\quad$ **grd002:** $\quad proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state)$

$\quad$ **grd003:** $\quad newstate \in PROCESS\_STATES$

$\quad$ **grd004:** $\quad core \in CORES \wedge core \in dom(current\_processes\_flag)$

$\quad$ **grd005:** $\quad processes\_of\_partition(proc) = part$

$\quad$ **grd017:** $\quad finished\_core2(core) = TRUE$

$\quad$ **grd101:** $\quad partition\_mode(part) = PM\_NORMAL$

$\quad$ **grd102:** $\quad process\_state(proc) = PS\_Running \wedge newstate = PS\_Dormant$

$\quad$ **grd201:** $\quad current\_partition = part$

$\quad$ **grd205:** $\quad processes\_of\_partition(proc) \in dom(current\_partition\_flag)$

$\quad$ **grd202:** $\quad current\_partition\_flag(part) = TRUE$

$\quad$ **grd203:** $\quad current\_processes\_flag(core) = TRUE$

$\quad$ **grd204:** $\quad proc \in ran(current\_processes)$

$\quad$ **then**

$\quad$ **act001:** $process\_state(proc) := newstate$

$\quad$ **act201:** $location\_of\_service2(core) := Stop\_self \mapsto loc\_i$

$\quad$ **act202:** $finished\_core2(core) := FALSE$

$\quad$ **act203:** $stop\_self\_proc(core) := proc$

$\quad$ **act204:** $timeout\_trigger := \{proc\} \lhd timeout\_trigger$

$\quad$ **act205:** $current\_processes\_flag(core) := FALSE$

$\quad$ **act206:** $current\_processes := \{core\} \lhd current\_processes$

$\quad$ **end**

Event stop_self_reschedule $\langle ordinary \rangle \; \widehat{=}$

extends stop_self_reschedule

$\quad$ **any**

$\quad\quad part$

$\quad\quad proc$

$\quad\quad core$

$\quad\quad reschedule$

$\quad$ **where**

$\quad$ **grd001:** $\quad part \in PARTITIONS$

$\quad$ **grd002:** $\quad proc \in processes \wedge proc \in dom(processes\_of\_partition)$

$\quad$ **grd003:** $\quad core \in (CORES \cap dom(stop\_self\_proc)) \wedge core \in dom(location\_of\_service2)$

$\quad$ **grd004:** $\quad processes\_of\_partition(proc) = part$

$\quad$ **grd005:** $\quad part = current\_partition$

$\quad$ **grd006:** $\quad proc = stop\_self\_proc(core)$

$\quad$ **grd014:** $\quad processes\_of\_partition(stop\_self\_proc(core)) \in dom(current\_partition\_flag) \wedge processes\_of\_partition(stop\_$

$\quad\quad\quad dom(locklevel\_of\_partition)$

$\quad$ **grd007:** $\quad current\_partition\_flag(part) = TRUE$

$\quad$ **grd008:** $\quad reschedule \in BOOL$

$\quad$ **grd015:** $\quad stop\_self\_proc(core) \in dom(process\_call\_errorhandler) \wedge process\_call\_errorhandler(stop\_self\_proc(core)) \in$

$\quad\quad\quad dom(process\_state)$

$\quad$ **grd009:**

$\quad\quad part \in dom(errorhandler\_of\_partition) \wedge proc = errorhandler\_of\_partition(part) \wedge locklevel\_of\_partition(part) >$

0

$\wedge process\_state(process\_call\_errorhandler(proc)) \neq PS\_Dormant \Rightarrow reschedule = FALSE$

grd010:

$\neg(part \in dom(errorhandler\_of\_partition) \wedge proc = errorhandler\_of\_partition(part) \wedge locklevel\_of\_partition(part)$

0

$\wedge process\_state(process\_call\_errorhandler(proc)) \neq PS\_Dormant) \Rightarrow reschedule = TRUE$

grd011: $finished\_core2(core) = FALSE$

grd012: $location\_of\_service2(core) = Stop\_self \mapsto loc\_i$

grd013: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Stop\_self \mapsto loc\_i)$

then

act001: $location\_of\_service2(core) := Stop\_self \mapsto loc\_1$

act002: $need\_reschedule := reschedule$

end

Event stop_self_return_no_mutex ⟨ordinary⟩ ≙

extends stop_self_return_no_mutex

any

 part

 proc

 core

where

grd001: $part \in PARTITIONS$

grd002: $proc \in (processes \cap ran(stop\_self\_proc))$

grd003: $core \in (CORES \cap dom(stop\_self\_proc)) \wedge core \in dom(current\_processes\_flag) \wedge core \in dom(location\_of\_service2)$

grd004: $proc = stop\_self\_proc(core)$

grd013: $stop\_self\_proc(core) \in dom(processes\_of\_partition) \wedge processes\_of\_partition(stop\_self\_proc(core)) \in dom(current\_partition\_flag)$

grd005: $processes\_of\_partition(proc) = part$

grd006: $part = current\_partition$

grd007: $current\_partition\_flag(part) = TRUE$

grd014: $stop\_self\_proc(core) \in dom(preemption\_lock\_mutex)$

grd012: $preemption\_lock\_mutex(proc) = FALSE$

grd009: $finished\_core2(core) = FALSE$

grd010: $location\_of\_service2(core) = Stop\_self \mapsto loc\_1$

grd011: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Stop\_self \mapsto loc\_1)$

then

act001: $location\_of\_service2(core) := Stop\_self \mapsto loc\_r$

act002: $finished\_core2(core) := TRUE$

act003: $stop\_self\_proc := \{core\} \lhd stop\_self\_proc$

end

Event stop_self_mutex_zero ⟨ordinary⟩ ≙

extends stop_self_mutex_zero

any

 part

 proc

 core

where

grd001: $part \in PARTITIONS$

grd002: $proc \in (processes \cap ran(stop\_self\_proc))$

grd003: $core \in (CORES \cap dom(stop\_self\_proc)) \wedge core \in dom(current\_processes\_flag) \wedge core \in dom(location\_of\_service2)$

grd004: $proc = stop\_self\_proc(core)$

grd014: $stop\_self\_proc(core) \in dom(processes\_of\_partition) \wedge processes\_of\_partition(stop\_self\_proc(core)) \in dom(current\_partition\_flag)$

grd005: $processes\_of\_partition(proc) = part$

grd006: $part = current\_partition$

grd013: $proc \notin ran(errorhandler\_of\_partition)$

grd007: $current\_partition\_flag(part) = TRUE$

grd015: $stop\_self\_proc(core) \in dom(preemption\_lock\_mutex)$

grd009: $preemption\_lock\_mutex(proc) = TRUE$

grd010: $finished\_core2(core) = FALSE$

grd011: $location\_of\_service2(core) = Stop\_self \mapsto loc\_1$

grd012: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Stop\_self \mapsto loc\_1)$

then

act001: $location\_of\_service2(core) := Stop\_self \mapsto loc\_2$

act002: $locklevel\_of\_partition(part) := 0$

act003: $preempter\_of\_partition := \{part\} \mathbin{\lhd\mkern-9mu-} preempter\_of\_partition$

end

Event stop_self_mutex_avail ⟨ordinary⟩ $\widehat{=}$

extends stop_self_mutex_avail

    any

        part

        proc

        core

    where

grd001: $part \in PARTITIONS$

grd002: $proc \in (processes \cap ran(stop\_self\_proc))$

grd003: $core \in (CORES \cap dom(stop\_self\_proc)) \wedge core \in dom(current\_processes\_flag) \wedge core \in dom(location\_of\_service2)$

grd004: $proc = stop\_self\_proc(core)$

grd013: $stop\_self\_proc(core) \in dom(processes\_of\_partition) \wedge processes\_of\_partition(stop\_self\_proc(core)) \in dom(current\_partition\_flag)$

grd005: $processes\_of\_partition(proc) = part$

grd014: $stop\_self\_proc(core) \in dom(preemption\_lock\_mutex)$

grd006: $part = current\_partition$

grd007: $current\_partition\_flag(part) = TRUE$

grd009: $preemption\_lock\_mutex(proc) = TRUE$

grd010: $finished\_core2(core) = FALSE$

grd011: $location\_of\_service2(core) = Stop\_self \mapsto loc\_2$

grd012: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Stop\_self \mapsto loc\_2)$

then

act001: $location\_of\_service2(core) := Stop\_self \mapsto loc\_3$

act002: $preemption\_lock\_mutex(proc) := FALSE$

end

Event stop_self_return_mutex ⟨ordinary⟩ $\widehat{=}$

extends stop_self_return_mutex

    any

        part

        proc

        core

    where

grd001: $part \in PARTITIONS$

grd002: $proc \in processes \cap ran(stop\_self\_proc)$

grd003: $core \in (CORES \cap dom(stop\_self\_proc)) \wedge core \in dom(current\_processes\_flag) \wedge core \in dom(location\_of\_service2)$

grd004: $proc = stop\_self\_proc(core)$

grd012: $stop\_self\_proc(core) \in dom(processes\_of\_partition) \wedge processes\_of\_partition(stop\_self\_proc(core)) \in dom(current\_partition\_flag)$

grd005: $processes\_of\_partition(proc) = part$

grd006: $part = current\_partition$

grd007: $current\_partition\_flag(part) = TRUE$

grd009: $finished\_core2(core) = FALSE$

grd010: $location\_of\_service2(core) = Stop\_self \mapsto loc\_3$

grd011: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Stop\_self \mapsto loc\_3)$

then

act001: $location\_of\_service2(core) := Stop\_self \mapsto loc\_r$

act002: $finished\_core(core) := TRUE$

act003: $stop\_self\_proc := \{core\} \mathbin{\lhd\mkern-9mu-} stop\_self\_proc$

end

Event stop_init ⟨ordinary⟩ ≙

extends stop_init

    any

        part

        proc

        newstate

        core

    where

        grd001: $part \in PARTITIONS$

        grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state)$

        grd003: $newstate \in PROCESS\_STATES$

        grd004: $core \in CORES \land core \in dom(current\_processes\_flag)$

        grd005: $processes\_of\_partition(proc) = part$

        grd006: $partition\_mode(part) = PM\_COLD\_START \lor partition\_mode(part) = PM\_WARM\_START \lor$
$partition\_mode(part) = PM\_NORMAL$

        grd017: $finished\_core2(core) = TRUE$

        grd101: $partition\_mode(part) = PM\_COLD\_START \lor partition\_mode(part) = PM\_WARM\_START \Rightarrow$
$((process\_state(proc) = PS\_Waiting \lor process\_state(proc) = PS\_WaitandSuspend) \land newstate =$
$PS\_Dormant)$

        grd102: $partition\_mode(part) = PM\_NORMAL \Rightarrow ((process\_state(proc) = PS\_Ready \lor process\_state(proc) =$
$PS\_Waiting \lor process\_state(proc) = PS\_WaitandSuspend \lor process\_state(proc) = PS\_Suspend \lor$
$process\_state(proc) = PS\_Faulted) \land newstate = PS\_Dormant)$

        grd201: $current\_partition = part$

        grd205: $processes\_of\_partition(proc) \in dom(current\_partition\_flag)$

        grd202: $current\_partition\_flag(part) = TRUE$

        grd203: $current\_processes\_flag(core) = TRUE \Rightarrow proc \notin ran(current\_processes)$

        grd204: $newstate = PS\_Dormant$

        grd301: $\neg(\exists r \cdot r \in queuing\_ports \land proc \in dom(processes\_waiting for\_queuingports(r)))$

        grd302: $\neg(\exists r \cdot r \in buffers \land proc \in dom(processes\_waiting for\_buffers(r)))$

        grd303: $\neg(\exists r \cdot r \in semaphores \land proc \in dom(processes\_waiting for\_semaphores(r)))$

        grd305: $\neg(\exists r \cdot r \in blackboards \land proc \in processes\_waiting for\_blackboards(r))$

        grd304: $\neg(\exists r \cdot r \in events \land proc \in processes\_waiting for\_events(r))$

    then

        act001: $process\_state(proc) := newstate$

        act201: $location\_of\_service2(core) := Stop \mapsto loc\_i$

        act202: $finished\_core2(core) := FALSE$

        act203: $stop\_proc(core) := proc$

        act204: $timeout\_trigger := \{proc\} \lhd timeout\_trigger$

    end

Event stop_reschedule ⟨ordinary⟩ ≙

extends stop_reschedule

    any

        part

        proc

        core

        reschedule

    where

        grd001: $part \in PARTITIONS$

        grd002: $proc \in processes \land proc \in dom(processes\_of\_partition)$

        grd003: $core \in CORES \cap dom(stop\_proc) \land core \in dom(current\_processes\_flag) \land core \in$
$dom(location\_of\_service2)$

        grd004: $processes\_of\_partition(proc) = part$

        grd005: $part = current\_partition$

        grd014: $processes\_of\_partition(proc) \in dom(current\_partition\_flag)$

        grd006: $current\_partition\_flag(part) = TRUE$

        grd007: $proc = stop\_proc(core)$

        grd008: $reschedule \in BOOL$

        grd009: $current\_processes\_flag(core) = TRUE \Rightarrow proc \notin ran(current\_processes)$

grd010: $reschedule = TRUE$

grd011: $finished\_core2(core) = FALSE$

grd012: $location\_of\_service2(core) = Stop \mapsto loc\_i$

grd013: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Stop \mapsto loc\_i)$

grd301: $\neg(\exists r \cdot r \in queuing\_ports \wedge proc \in dom(processes\_waiting for\_queuing ports(r)))$

grd302: $\neg(\exists r \cdot r \in buffers \wedge proc \in dom(processes\_waiting for\_buffers(r)))$

grd303: $\neg(\exists r \cdot r \in semaphores \wedge proc \in dom(processes\_waiting for\_semaphores(r)))$

grd305: $\neg(\exists r \cdot r \in blackboards \wedge proc \in processes\_waiting for\_blackboards(r))$

grd304: $\neg(\exists r \cdot r \in events \wedge proc \in processes\_waiting for\_events(r))$

then

act001: $location\_of\_service2(core) := Stop \mapsto loc\_1$

act002: $need\_reschedule := reschedule$

end

Event stop_return_no_mutex ⟨ordinary⟩ $\widehat{=}$

extends stop_return_no_mutex

any

> part
> proc
> core

where

grd001: $part \in PARTITIONS$

grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition)$

grd003: $core \in CORES \cap dom(stop\_proc) \wedge core \in dom(current\_processes\_flag) \wedge core \in dom(location\_of\_service2)$

grd004: $processes\_of\_partition(proc) = part$

grd005: $proc = stop\_proc(core)$

grd006: $part = current\_partition$

grd013: $processes\_of\_partition(stop\_proc(core)) \in dom(current\_partition\_flag)$

grd012: $current\_partition\_flag(part) = TRUE$

grd007: $current\_processes\_flag(core) = TRUE \Rightarrow proc \notin ran(current\_processes)$

grd014: $stop\_proc(core) \in dom(preemption\_lock\_mutex)$

grd008: $preemption\_lock\_mutex(proc) = FALSE$

grd009: $finished\_core2(core) = FALSE$

grd010: $location\_of\_service2(core) = Stop \mapsto loc\_1$

grd011: $\neg(finished\_core(core) = FALSE \wedge location\_of\_service2(core) = Stop \mapsto loc\_1)$

then

act001: $location\_of\_service2(core) := Stop \mapsto loc\_r$

act002: $finished\_core2(core) := TRUE$

act003: $stop\_proc := \{core\} \lhd stop\_proc$

end

Event stop_mutex_zero ⟨ordinary⟩ $\widehat{=}$

extends stop_mutex_zero

any

> part
> proc
> core

where

grd001: $part \in PARTITIONS$

grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition)$

grd003: $core \in CORES \cap dom(stop\_proc) \wedge core \in dom(current\_processes\_flag) \wedge core \in dom(location\_of\_service2)$

grd004: $processes\_of\_partition(proc) = part$

grd005: $proc = stop\_proc(core)$

grd006: $part = current\_partition$

grd012: $processes\_of\_partition(stop\_proc(core)) \in dom(current\_partition\_flag)$

grd007: $current\_partition\_flag(part) = TRUE$

grd008: $current\_processes\_flag(core) = TRUE \Rightarrow proc \notin ran(current\_processes)$

grd009: $finished\_core2(core) = FALSE$

grd010: $location\_of\_service2(core) = Stop \mapsto loc\_1$

grd011: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Stop \mapsto loc\_1)$
grd301: $\neg(\exists r \cdot r \in queuing\_ports \wedge proc \in dom(processes\_waiting for\_queuing ports(r)))$
grd302: $\neg(\exists r \cdot r \in buffers \wedge proc \in dom(processes\_waiting for\_buffers(r)))$
grd303: $\neg(\exists r \cdot r \in semaphores \wedge proc \in dom(processes\_waiting for\_semaphores(r)))$
grd305: $\neg(\exists r \cdot r \in blackboards \wedge proc \in processes\_waiting for\_blackboards(r))$
grd304: $\neg(\exists r \cdot r \in events \wedge proc \in processes\_waiting for\_events(r))$

then

act001: $location\_of\_service2(core) := Stop \mapsto loc\_2$
act002: $locklevel\_of\_partition(part) := 0$
act003: $preempter\_of\_partition := \{part\} \lhd preempter\_of\_partition$

end

Event stop_mutex_avail $\langle ordinary \rangle \;\widehat{=}$

extends stop_mutex_avail

any

    part
    proc
    core

where

grd001: $part \in PARTITIONS$
grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition) \wedge proc \in dom(preemption\_lock\_mutex)$

grd003: $core \in CORES \cap dom(stop\_proc) \wedge core \in dom(current\_processes\_flag) \wedge core \in dom(location\_of\_service2)$
grd004: $processes\_of\_partition(proc) = part$
grd005: $proc = stop\_proc(core)$
grd006: $part = current\_partition$
grd013: $processes\_of\_partition(stop\_proc(core)) \in dom(current\_partition\_flag)$
grd007: $current\_partition\_flag(part) = TRUE$
grd008: $current\_processes\_flag(core) = TRUE \Rightarrow proc \notin ran(current\_processes)$
grd009: $preemption\_lock\_mutex(proc) = TRUE$
grd010: $finished\_core2(core) = FALSE$
grd011: $location\_of\_service2(core) = Stop \mapsto loc\_2$
grd012: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Stop \mapsto loc\_2)$
grd301: $\neg(\exists r \cdot r \in queuing\_ports \wedge proc \in dom(processes\_waiting for\_queuing ports(r)))$
grd302: $\neg(\exists r \cdot r \in buffers \wedge proc \in dom(processes\_waiting for\_buffers(r)))$
grd303: $\neg(\exists r \cdot r \in semaphores \wedge proc \in dom(processes\_waiting for\_semaphores(r)))$
grd305: $\neg(\exists r \cdot r \in blackboards \wedge proc \in processes\_waiting for\_blackboards(r))$
grd304: $\neg(\exists r \cdot r \in events \wedge proc \in processes\_waiting for\_events(r))$

then

act001: $location\_of\_service2(core) := Stop \mapsto loc\_3$
act002: $preemption\_lock\_mutex(proc) := FALSE$

end

Event stop_return_mutex $\langle ordinary \rangle \;\widehat{=}$

extends stop_return_mutex

any

    part
    proc
    core

where

grd001: $part \in PARTITIONS$
grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition)$
grd003: $core \in CORES \cap dom(stop\_proc) \wedge core \in dom(current\_processes\_flag) \wedge core \in dom(location\_of\_service2)$
grd004: $processes\_of\_partition(proc) = part$
grd005: $part = current\_partition$
grd011: $processes\_of\_partition(proc) \in dom(current\_partition\_flag)$
grd006: $current\_partition\_flag(part) = TRUE$
grd007: $current\_processes\_flag(core) = TRUE \Rightarrow proc \notin ran(current\_processes)$
grd008: $finished\_core2(core) = FALSE$

$grd009$: $location\_of\_service2(core) = Stop \mapsto loc\_3$

$grd010$: $\neg(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Stop \mapsto loc\_3)$

then

$act001$: $location\_of\_service2(core) := Stop \mapsto loc\_r$

$act002$: $finished\_core2(core) := TRUE$

$act003$: $stop\_proc := \{core\} \lessdot stop\_proc$

end

Event stop_wf_qport_init ⟨ordinary⟩ $\widehat{=}$

extends stop_wf_qport_init

any

$part$

$proc$

$newstate$

$core$

r

where

$grd001$: $part \in PARTITIONS$

$grd002$: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state)$

$grd003$: $newstate \in PROCESS\_STATES$

$grd004$: $core \in CORES \land core \in dom(current\_processes\_flag)$

$grd005$: $processes\_of\_partition(proc) = part$

$grd006$: $partition\_mode(part) = PM\_COLD\_START \lor partition\_mode(part) = PM\_WARM\_START \lor$
$partition\_mode(part) = PM\_NORMAL$

$grd017$: $finished\_core2(core) = TRUE$

$grd101$: $partition\_mode(part) = PM\_COLD\_START \lor partition\_mode(part) = PM\_WARM\_START \Rightarrow$
$((process\_state(proc) = PS\_Waiting \lor process\_state(proc) = PS\_WaitandSuspend) \land newstate =$
$PS\_Dormant)$

$grd102$: $partition\_mode(part) = PM\_NORMAL \Rightarrow ((process\_state(proc) = PS\_Ready \lor process\_state(proc) =$
$PS\_Waiting \lor process\_state(proc) = PS\_WaitandSuspend \lor process\_state(proc) = PS\_Suspend \lor$
$process\_state(proc) = PS\_Faulted) \land newstate = PS\_Dormant)$

$grd201$: $current\_partition = part$

$grd205$: $processes\_of\_partition(proc) \in dom(current\_partition\_flag)$

$grd202$: $current\_partition\_flag(part) = TRUE$

$grd203$: $current\_processes\_flag(core) = TRUE \Rightarrow proc \notin ran(current\_processes)$

$grd204$: $newstate = PS\_Dormant$

$grd301$: $r \in queuing\_ports \land proc \in dom(processes\_waitingfor\_queuingports(r))$

then

$act001$: $process\_state(proc) := newstate$

$act201$: $location\_of\_service2(core) := Stop \mapsto loc\_i$

$act202$: $finished\_core2(core) := FALSE$

$act203$: $stop\_proc(core) := proc$

$act204$: $timeout\_trigger := \{proc\} \lessdot timeout\_trigger$

$act301$: $processes\_waitingfor\_queuingports := (processes\_waitingfor\_queuingports \lessdot \{r \mapsto (\{proc\} \lessdot$
$processes\_waitingfor\_queuingports(r))\})$

end

Event stop_wf_qport_reschedule ⟨ordinary⟩ $\widehat{=}$

extends stop_wf_qport_reschedule

any

$part$

$proc$

$core$

$reschedule$

where

$grd001$: $part \in PARTITIONS$

$grd002$: $proc \in processes \land proc \in dom(processes\_of\_partition)$

$grd003$: $core \in CORES \cap dom(stop\_proc) \land core \in dom(current\_processes\_flag) \land core \in$
$dom(location\_of\_service2)$

$grd004$: $processes\_of\_partition(proc) = part$

$grd005$: $part = current\_partition$

        grd014: $processes\_of\_partition(proc) \in dom(current\_partition\_flag)$

        grd006: $current\_partition\_flag(part) = TRUE$

        grd007: $proc = stop\_proc(core)$

        grd008: $reschedule \in BOOL$

        grd009: $current\_processes\_flag(core) = TRUE \Rightarrow proc \notin ran(current\_processes)$

        grd010: $reschedule = TRUE$

        grd011: $finished\_core2(core) = FALSE$

        grd012: $location\_of\_service2(core) = Stop \mapsto loc\_i$

        grd013: $\neg(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Stop \mapsto loc\_i)$

then

        act001: $location\_of\_service2(core) := Stop \mapsto loc\_1$

        act002: $need\_reschedule := reschedule$

end

Event stop_wf_return_no_mutex ⟨ordinary⟩ $\widehat{=}$

extends stop_wf_return_no_mutex

    any

        part

        proc

        core

    where

        grd001: $part \in PARTITIONS$

        grd002: $proc \in processes \land proc \in dom(processes\_of\_partition)$

        grd003: $core \in CORES \cap dom(stop\_proc) \land core \in dom(current\_processes\_flag) \land core \in dom(location\_of\_service2)$

        grd004: $processes\_of\_partition(proc) = part$

        grd005: $proc = stop\_proc(core)$

        grd006: $part = current\_partition$

        grd013: $processes\_of\_partition(stop\_proc(core)) \in dom(current\_partition\_flag)$

        grd012: $current\_partition\_flag(part) = TRUE$

        grd007: $current\_processes\_flag(core) = TRUE \Rightarrow proc \notin ran(current\_processes)$

        grd014: $stop\_proc(core) \in dom(preemption\_lock\_mutex)$

        grd008: $preemption\_lock\_mutex(proc) = FALSE$

        grd009: $finished\_core2(core) = FALSE$

        grd010: $location\_of\_service2(core) = Stop \mapsto loc\_1$

        grd011: $\neg(finished\_core(core) = FALSE \land location\_of\_service2(core) = Stop \mapsto loc\_1)$

    then

        act001: $location\_of\_service2(core) := Stop \mapsto loc\_r$

        act002: $finished\_core2(core) := TRUE$

        act003: $stop\_proc := \{core\} \lhd stop\_proc$

    end

Event stop_wf_mutex_zero ⟨ordinary⟩ $\widehat{=}$

extends stop_wf_mutex_zero

    any

        part

        proc

        core

    where

        grd001: $part \in PARTITIONS$

        grd002: $proc \in processes \land proc \in dom(processes\_of\_partition)$

        grd003: $core \in CORES \cap dom(stop\_proc) \land core \in dom(current\_processes\_flag) \land core \in dom(location\_of\_service2)$

        grd004: $processes\_of\_partition(proc) = part$

        grd005: $proc = stop\_proc(core)$

        grd006: $part = current\_partition$

        grd012: $processes\_of\_partition(stop\_proc(core)) \in dom(current\_partition\_flag)$

        grd007: $current\_partition\_flag(part) = TRUE$

        grd008: $current\_processes\_flag(core) = TRUE \Rightarrow proc \notin ran(current\_processes)$

        grd009: $finished\_core2(core) = FALSE$

        grd010: $location\_of\_service2(core) = Stop \mapsto loc\_1$

grd011: $\neg(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Stop \mapsto loc\_1)$

then

act001: $location\_of\_service2(core) := Stop \mapsto loc\_2$

act002: $locklevel\_of\_partition(part) := 0$

act003: $preempter\_of\_partition := \{part\} \vartriangleleft preempter\_of\_partition$

end

Event stop_wf_mutex_avail ⟨ordinary⟩ $\widehat{=}$

extends stop_wf_mutex_avail

any

    part

    proc

    core

where

grd001: $part \in PARTITIONS$

grd002: $proc \in processes \land proc \in dom(processes\_of\_partition) \land proc \in dom(preemption\_lock\_mutex)$

grd003: $core \in CORES \cap dom(stop\_proc) \land core \in dom(current\_processes\_flag) \land core \in dom(location\_of\_service2)$

grd004: $processes\_of\_partition(proc) = part$

grd005: $proc = stop\_proc(core)$

grd006: $part = current\_partition$

grd013: $processes\_of\_partition(stop\_proc(core)) \in dom(current\_partition\_flag)$

grd007: $current\_partition\_flag(part) = TRUE$

grd008: $current\_processes\_flag(core) = TRUE \Rightarrow proc \notin ran(current\_processes)$

grd009: $preemption\_lock\_mutex(proc) = TRUE$

grd010: $finished\_core2(core) = FALSE$

grd011: $location\_of\_service2(core) = Stop \mapsto loc\_2$

grd012: $\neg(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Stop \mapsto loc\_2)$

then

act001: $location\_of\_service2(core) := Stop \mapsto loc\_3$

act002: $preemption\_lock\_mutex(proc) := FALSE$

end

Event stop_wf_return_mutex ⟨ordinary⟩ $\widehat{=}$

extends stop_wf_return_mutex

any

    part

    proc

    core

where

grd001: $part \in PARTITIONS$

grd002: $proc \in processes \land proc \in dom(processes\_of\_partition)$

grd003: $core \in CORES \cap dom(stop\_proc) \land core \in dom(current\_processes\_flag) \land core \in dom(location\_of\_service2)$

grd004: $processes\_of\_partition(proc) = part$

grd005: $part = current\_partition$

grd011: $processes\_of\_partition(proc) \in dom(current\_partition\_flag)$

grd006: $current\_partition\_flag(part) = TRUE$

grd007: $current\_processes\_flag(core) = TRUE \Rightarrow proc \notin ran(current\_processes)$

grd008: $finished\_core2(core) = FALSE$

grd009: $location\_of\_service2(core) = Stop \mapsto loc\_3$

grd010: $\neg(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Stop \mapsto loc\_3)$

then

act001: $location\_of\_service2(core) := Stop \mapsto loc\_r$

act002: $finished\_core2(core) := TRUE$

act003: $stop\_proc := \{core\} \vartriangleleft stop\_proc$

end

Event stop_wf_buf_init ⟨ordinary⟩ $\widehat{=}$

extends stop_wf_buf_init

any
> *part*
> *proc*
> *newstate*
> *core*
> *r*

where
> **grd001:** $part \in PARTITIONS$
> **grd002:** $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state)$
> **grd003:** $newstate \in PROCESS\_STATES$
> **grd004:** $core \in CORES \wedge core \in dom(current\_processes\_flag)$
> **grd005:** $processes\_of\_partition(proc) = part$
> **grd006:** $partition\_mode(part) = PM\_COLD\_START \vee partition\_mode(part) = PM\_WARM\_START \vee$
> $partition\_mode(part) = PM\_NORMAL$
> **grd017:** $finished\_core2(core) = TRUE$
> **grd101:** $partition\_mode(part) = PM\_COLD\_START \vee partition\_mode(part) = PM\_WARM\_START \Rightarrow$
> $((process\_state(proc) = PS\_Waiting \vee process\_state(proc) = PS\_WaitandSuspend) \wedge newstate =$
> $PS\_Dormant)$
> **grd102:** $partition\_mode(part) = PM\_NORMAL \Rightarrow ((process\_state(proc) = PS\_Ready \vee process\_state(proc) =$
> $PS\_Waiting \vee process\_state(proc) = PS\_WaitandSuspend \vee process\_state(proc) = PS\_Suspend \vee$
> $process\_state(proc) = PS\_Faulted) \wedge newstate = PS\_Dormant)$
> **grd201:** $current\_partition = part$
> **grd205:** $processes\_of\_partition(proc) \in dom(current\_partition\_flag)$
> **grd202:** $current\_partition\_flag(part) = TRUE$
> **grd203:** $current\_processes\_flag(core) = TRUE \Rightarrow proc \notin ran(current\_processes)$
> **grd204:** $newstate = PS\_Dormant$
> **grd301:** $r \in buffers \wedge proc \in dom(processes\_waitingfor\_buffers(r))$

then
> **act001:** $process\_state(proc) := newstate$
> **act201:** $location\_of\_service2(core) := Stop \mapsto loc\_i$
> **act202:** $finished\_core2(core) := FALSE$
> **act203:** $stop\_proc(core) := proc$
> **act204:** $timeout\_trigger := \{proc\} \lhd timeout\_trigger$
> **act301:** $processes\_waitingfor\_buffers := (processes\_waitingfor\_buffers \lhd \{r \mapsto (\{proc\} \lhd processes\_waitingfor\_bu$

end

Event stop_wf_buf_reschedule ⟨ordinary⟩ ≙

extends stop_wf_buf_reschedule

any
> *part*
> *proc*
> *core*
> *reschedule*

where
> **grd001:** $part \in PARTITIONS$
> **grd002:** $proc \in processes \wedge proc \in dom(processes\_of\_partition)$
> **grd003:** $core \in CORES \cap dom(stop\_proc) \wedge core \in dom(current\_processes\_flag) \wedge core \in$
> $dom(location\_of\_service2)$
> **grd004:** $processes\_of\_partition(proc) = part$
> **grd005:** $part = current\_partition$
> **grd014:** $processes\_of\_partition(proc) \in dom(current\_partition\_flag)$
> **grd006:** $current\_partition\_flag(part) = TRUE$
> **grd007:** $proc = stop\_proc(core)$
> **grd008:** $reschedule \in BOOL$
> **grd009:** $current\_processes\_flag(core) = TRUE \Rightarrow proc \notin ran(current\_processes)$
> **grd010:** $reschedule = TRUE$
> **grd011:** $finished\_core2(core) = FALSE$
> **grd012:** $location\_of\_service2(core) = Stop \mapsto loc\_i$
> **grd013:** $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Stop \mapsto loc\_i)$

then

    act001: $location\_of\_service2(core) := Stop \mapsto loc\_1$

    act002: $need\_reschedule := reschedule$

end

Event stop_wf_buf_return_no_mutex $\langle ordinary \rangle \; \widehat{=}$

extends stop_wf_buf_return_no_mutex

    any

        part

        proc

        core

    where

        grd001: $part \in PARTITIONS$

        grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition)$

        grd003: $core \in CORES \cap dom(stop\_proc) \wedge core \in dom(current\_processes\_flag) \wedge core \in dom(location\_of\_service2)$

        grd004: $processes\_of\_partition(proc) = part$

        grd005: $proc = stop\_proc(core)$

        grd006: $part = current\_partition$

        grd013: $processes\_of\_partition(stop\_proc(core)) \in dom(current\_partition\_flag)$

        grd012: $current\_partition\_flag(part) = TRUE$

        grd007: $current\_processes\_flag(core) = TRUE \Rightarrow proc \notin ran(current\_processes)$

        grd014: $stop\_proc(core) \in dom(preemption\_lock\_mutex)$

        grd008: $preemption\_lock\_mutex(proc) = FALSE$

        grd009: $finished\_core2(core) = FALSE$

        grd010: $location\_of\_service2(core) = Stop \mapsto loc\_1$

        grd011: $\neg(finished\_core(core) = FALSE \wedge location\_of\_service2(core) = Stop \mapsto loc\_1)$

    then

        act001: $location\_of\_service2(core) := Stop \mapsto loc\_r$

        act002: $finished\_core2(core) := TRUE$

        act003: $stop\_proc := \{core\} \lhd stop\_proc$

    end

Event stop_wf_buf_mutex_zero $\langle ordinary \rangle \; \widehat{=}$

extends stop_wf_buf_mutex_zero

    any

        part

        proc

        core

    where

        grd001: $part \in PARTITIONS$

        grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition)$

        grd003: $core \in CORES \cap dom(stop\_proc) \wedge core \in dom(current\_processes\_flag) \wedge core \in dom(location\_of\_service2)$

        grd004: $processes\_of\_partition(proc) = part$

        grd005: $proc = stop\_proc(core)$

        grd006: $part = current\_partition$

        grd012: $processes\_of\_partition(stop\_proc(core)) \in dom(current\_partition\_flag)$

        grd007: $current\_partition\_flag(part) = TRUE$

        grd008: $current\_processes\_flag(core) = TRUE \Rightarrow proc \notin ran(current\_processes)$

        grd009: $finished\_core2(core) = FALSE$

        grd010: $location\_of\_service2(core) = Stop \mapsto loc\_1$

        grd011: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Stop \mapsto loc\_1)$

    then

        act001: $location\_of\_service2(core) := Stop \mapsto loc\_2$

        act002: $locklevel\_of\_partition(part) := 0$

        act003: $preempter\_of\_partition := \{part\} \lhd preempter\_of\_partition$

    end

Event stop_wf_buf_mutex_avail $\langle ordinary \rangle \; \widehat{=}$

extends stop_wf_buf_mutex_avail

any
> *part*
> *proc*
> *core*

where
> grd001: $part \in PARTITIONS$
> grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition) \wedge proc \in dom(preemption\_lock\_mutex)$
>
> grd003: $core \in CORES \cap dom(stop\_proc) \wedge core \in dom(current\_processes\_flag) \wedge core \in dom(location\_of\_service2)$
> grd004: $processes\_of\_partition(proc) = part$
> grd005: $proc = stop\_proc(core)$
> grd006: $part = current\_partition$
> grd013: $processes\_of\_partition(stop\_proc(core)) \in dom(current\_partition\_flag)$
> grd007: $current\_partition\_flag(part) = TRUE$
> grd008: $current\_processes\_flag(core) = TRUE \Rightarrow proc \notin ran(current\_processes)$
> grd009: $preemption\_lock\_mutex(proc) = TRUE$
> grd010: $finished\_core2(core) = FALSE$
> grd011: $location\_of\_service2(core) = Stop \mapsto loc\_2$
> grd012: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Stop \mapsto loc\_2)$

then
> act001: $location\_of\_service2(core) := Stop \mapsto loc\_3$
> act002: $preemption\_lock\_mutex(proc) := FALSE$

end

Event stop_wf_buf_return_mutex $\langle ordinary \rangle \;\widehat{=}$

extends stop_wf_buf_return_mutex

any
> *part*
> *proc*
> *core*

where
> grd001: $part \in PARTITIONS$
> grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition)$
> grd003: $core \in CORES \cap dom(stop\_proc) \wedge core \in dom(current\_processes\_flag) \wedge core \in dom(location\_of\_service2)$
> grd004: $processes\_of\_partition(proc) = part$
> grd005: $part = current\_partition$
> grd011: $processes\_of\_partition(proc) \in dom(current\_partition\_flag)$
> grd006: $current\_partition\_flag(part) = TRUE$
> grd007: $current\_processes\_flag(core) = TRUE \Rightarrow proc \notin ran(current\_processes)$
> grd008: $finished\_core2(core) = FALSE$
> grd009: $location\_of\_service2(core) = Stop \mapsto loc\_3$
> grd010: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Stop \mapsto loc\_3)$

then
> act001: $location\_of\_service2(core) := Stop \mapsto loc\_r$
> act002: $finished\_core2(core) := TRUE$
> act003: $stop\_proc := \{core\} \vartriangleleft stop\_proc$

end

Event stop_wf_sem_init $\langle ordinary \rangle \;\widehat{=}$

extends stop_wf_sem_init

any
> *part*
> *proc*
> *newstate*
> *core*
> *r*

where
> grd001: $part \in PARTITIONS$
> grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state)$

        grd003: $newstate \in PROCESS\_STATES$

        grd004: $core \in CORES \land core \in dom(current\_processes\_flag)$

        grd005: $processes\_of\_partition(proc) = part$

        grd006: $partition\_mode(part) = PM\_COLD\_START \lor partition\_mode(part) = PM\_WARM\_START \lor$
$partition\_mode(part) = PM\_NORMAL$

        grd017: $finished\_core2(core) = TRUE$

        grd101: $partition\_mode(part) = PM\_COLD\_START \lor partition\_mode(part) = PM\_WARM\_START \Rightarrow$
$((process\_state(proc) = PS\_Waiting \lor process\_state(proc) = PS\_WaitandSuspend) \land newstate =$
$PS\_Dormant)$

        grd102: $partition\_mode(part) = PM\_NORMAL \Rightarrow ((process\_state(proc) = PS\_Ready \lor process\_state(proc) =$
$PS\_Waiting \lor process\_state(proc) = PS\_WaitandSuspend \lor process\_state(proc) = PS\_Suspend \lor$
$process\_state(proc) = PS\_Faulted) \land newstate = PS\_Dormant)$

        grd201: $current\_partition = part$

        grd205: $processes\_of\_partition(proc) \in dom(current\_partition\_flag)$

        grd202: $current\_partition\_flag(part) = TRUE$

        grd203: $current\_processes\_flag(core) = TRUE \Rightarrow proc \notin ran(current\_processes)$

        grd204: $newstate = PS\_Dormant$

        grd301: $r \in semaphores \land proc \in dom(processes\_waiting for\_semaphores(r))$

then

        act001: $process\_state(proc) := newstate$

        act201: $location\_of\_service2(core) := Stop \mapsto loc\_i$

        act202: $finished\_core2(core) := FALSE$

        act203: $stop\_proc(core) := proc$

        act204: $timeout\_trigger := \{proc\} \mathbin{\lhd\mkern-14mu-} timeout\_trigger$

        act301: $processes\_waiting for\_semaphores := (processes\_waiting for\_semaphores \mathbin{\lhd\mkern-14mu-} \{r \mapsto (\{proc\} \mathbin{\lhd\mkern-14mu-}$
$processes\_waiting for\_semaphores(r))\})$

end

Event stop_wf_sem_reschedule ⟨ordinary⟩ $\mathrel{\widehat{=}}$

extends stop_wf_sem_reschedule

    any

        part

        proc

        core

        reschedule

    where

        grd001: $part \in PARTITIONS$

        grd002: $proc \in processes \land proc \in dom(processes\_of\_partition)$

        grd003: $core \in CORES \cap dom(stop\_proc) \land core \in dom(current\_processes\_flag) \land core \in$
$dom(location\_of\_service2)$

        grd004: $processes\_of\_partition(proc) = part$

        grd005: $part = current\_partition$

        grd014: $processes\_of\_partition(proc) \in dom(current\_partition\_flag)$

        grd006: $current\_partition\_flag(part) = TRUE$

        grd007: $proc = stop\_proc(core)$

        grd008: $reschedule \in BOOL$

        grd009: $current\_processes\_flag(core) = TRUE \Rightarrow proc \notin ran(current\_processes)$

        grd010: $reschedule = TRUE$

        grd011: $finished\_core2(core) = FALSE$

        grd012: $location\_of\_service2(core) = Stop \mapsto loc\_i$

        grd013: $\neg(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Stop \mapsto loc\_i)$

    then

        act001: $location\_of\_service2(core) := Stop \mapsto loc\_1$

        act002: $need\_reschedule := reschedule$

    end

Event stop_wf_sem_return_no_mutex ⟨ordinary⟩ $\mathrel{\widehat{=}}$

extends stop_wf_sem_return_no_mutex

    any

        part

        proc

$core$

where

    grd001: $part \in PARTITIONS$

    grd002: $proc \in processes \land proc \in dom(processes\_of\_partition)$

    grd003: $core \in CORES \cap dom(stop\_proc) \land core \in dom(current\_processes\_flag) \land core \in dom(location\_of\_service2)$

    grd004: $processes\_of\_partition(proc) = part$

    grd005: $proc = stop\_proc(core)$

    grd006: $part = current\_partition$

    grd013: $processes\_of\_partition(stop\_proc(core)) \in dom(current\_partition\_flag)$

    grd012: $current\_partition\_flag(part) = TRUE$

    grd007: $current\_processes\_flag(core) = TRUE \Rightarrow proc \notin ran(current\_processes)$

    grd014: $stop\_proc(core) \in dom(preemption\_lock\_mutex)$

    grd008: $preemption\_lock\_mutex(proc) = FALSE$

    grd009: $finished\_core2(core) = FALSE$

    grd010: $location\_of\_service2(core) = Stop \mapsto loc\_1$

    grd011: $\neg(finished\_core(core) = FALSE \land location\_of\_service2(core) = Stop \mapsto loc\_1)$

then

    act001: $location\_of\_service2(core) := Stop \mapsto loc\_r$

    act002: $finished\_core2(core) := TRUE$

    act003: $stop\_proc := \{core\} \lhd stop\_proc$

end

Event stop_wf_sem_mutex_zero $\langle ordinary \rangle \;\hat{=}$

extends stop_wf_sem_mutex_zero

    any

        $part$

        $proc$

        $core$

    where

    grd001: $part \in PARTITIONS$

    grd002: $proc \in processes \land proc \in dom(processes\_of\_partition)$

    grd003: $core \in CORES \cap dom(stop\_proc) \land core \in dom(current\_processes\_flag) \land core \in dom(location\_of\_service2)$

    grd004: $processes\_of\_partition(proc) = part$

    grd005: $proc = stop\_proc(core)$

    grd006: $part = current\_partition$

    grd012: $processes\_of\_partition(stop\_proc(core)) \in dom(current\_partition\_flag)$

    grd007: $current\_partition\_flag(part) = TRUE$

    grd008: $current\_processes\_flag(core) = TRUE \Rightarrow proc \notin ran(current\_processes)$

    grd009: $finished\_core2(core) = FALSE$

    grd010: $location\_of\_service2(core) = Stop \mapsto loc\_1$

    grd011: $\neg(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Stop \mapsto loc\_1)$

    then

    act001: $location\_of\_service2(core) := Stop \mapsto loc\_2$

    act002: $locklevel\_of\_partition(part) := 0$

    act003: $preempter\_of\_partition := \{part\} \lhd preempter\_of\_partition$

    end

Event stop_wf_sem_mutex_avail $\langle ordinary \rangle \;\hat{=}$

extends stop_wf_sem_mutex_avail

    any

        $part$

        $proc$

        $core$

    where

    grd001: $part \in PARTITIONS$

    grd002: $proc \in processes \land proc \in dom(processes\_of\_partition) \land proc \in dom(preemption\_lock\_mutex)$

    grd003: $core \in CORES \cap dom(stop\_proc) \land core \in dom(current\_processes\_flag) \land core \in dom(location\_of\_service2)$

        grd004: $processes\_of\_partition(proc) = part$

        grd005: $proc = stop\_proc(core)$

        grd006: $part = current\_partition$

        grd013: $processes\_of\_partition(stop\_proc(core)) \in dom(current\_partition\_flag)$

        grd007: $current\_partition\_flag(part) = TRUE$

        grd008: $current\_processes\_flag(core) = TRUE \Rightarrow proc \notin ran(current\_processes)$

        grd009: $preemption\_lock\_mutex(proc) = TRUE$

        grd010: $finished\_core2(core) = FALSE$

        grd011: $location\_of\_service2(core) = Stop \mapsto loc\_2$

        grd012: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Stop \mapsto loc\_2)$

then

        act001: $location\_of\_service2(core) := Stop \mapsto loc\_3$

        act002: $preemption\_lock\_mutex(proc) := FALSE$

end

Event stop_wf_sem_return_mutex ⟨ordinary⟩ $\widehat{=}$

extends stop_wf_sem_return_mutex

    any

        part

        proc

        core

    where

        grd001: $part \in PARTITIONS$

        grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition)$

        grd003: $core \in CORES \cap dom(stop\_proc) \wedge core \in dom(current\_processes\_flag) \wedge core \in dom(location\_of\_service2)$

        grd004: $processes\_of\_partition(proc) = part$

        grd005: $part = current\_partition$

        grd011: $processes\_of\_partition(proc) \in dom(current\_partition\_flag)$

        grd006: $current\_partition\_flag(part) = TRUE$

        grd007: $current\_processes\_flag(core) = TRUE \Rightarrow proc \notin ran(current\_processes)$

        grd008: $finished\_core2(core) = FALSE$

        grd009: $location\_of\_service2(core) = Stop \mapsto loc\_3$

        grd010: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Stop \mapsto loc\_3)$

    then

        act001: $location\_of\_service2(core) := Stop \mapsto loc\_r$

        act002: $finished\_core2(core) := TRUE$

        act003: $stop\_proc := \{core\} \lhd stop\_proc$

    end

Event stop_wf_bb_init ⟨ordinary⟩ $\widehat{=}$

extends stop_wf_bb_init

    any

        part

        proc

        newstate

        core

        r

    where

        grd001: $part \in PARTITIONS$

        grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state)$

        grd003: $newstate \in PROCESS\_STATES$

        grd004: $core \in CORES \wedge core \in dom(current\_processes\_flag)$

        grd005: $processes\_of\_partition(proc) = part$

        grd006: $partition\_mode(part) = PM\_COLD\_START \vee partition\_mode(part) = PM\_WARM\_START \vee partition\_mode(part) = PM\_NORMAL$

        grd017: $finished\_core2(core) = TRUE$

        grd101: $partition\_mode(part) = PM\_COLD\_START \vee partition\_mode(part) = PM\_WARM\_START \Rightarrow ((process\_state(proc) = PS\_Waiting \vee process\_state(proc) = PS\_WaitandSuspend) \wedge newstate = PS\_Dormant)$

        grd102: $partition\_mode(part) = PM\_NORMAL \Rightarrow ((process\_state(proc) = PS\_Ready \lor process\_state(proc) = PS\_Waiting \lor process\_state(proc) = PS\_WaitandSuspend \lor process\_state(proc) = PS\_Suspend \lor process\_state(proc) = PS\_Faulted) \land newstate = PS\_Dormant)$

        grd201: $current\_partition = part$

        grd205: $processes\_of\_partition(proc) \in dom(current\_partition\_flag)$

        grd202: $current\_partition\_flag(part) = TRUE$

        grd203: $current\_processes\_flag(core) = TRUE \Rightarrow proc \notin ran(current\_processes)$

        grd204: $newstate = PS\_Dormant$

        grd301: $r \in blackboards \land proc \in processes\_waiting for\_blackboards(r)$

then

        act001: $process\_state(proc) := newstate$

        act201: $location\_of\_service2(core) := Stop \mapsto loc\_i$

        act202: $finished\_core2(core) := FALSE$

        act203: $stop\_proc(core) := proc$

        act204: $timeout\_trigger := \{proc\} \lhd timeout\_trigger$

        act301: $processes\_waiting for\_blackboards := processes\_waiting for\_blackboards \Leftarrow \{r \mapsto (processes\_waiting for\_blac \{proc\})\}$

end

Event stop_wf_bb_reschedule ⟨ordinary⟩ $\widehat{=}$

extends stop_wf_bb_reschedule

    any

        part

        proc

        core

        reschedule

    where

        grd001: $part \in PARTITIONS$

        grd002: $proc \in processes \land proc \in dom(processes\_of\_partition)$

        grd003: $core \in CORES \cap dom(stop\_proc) \land core \in dom(current\_processes\_flag) \land core \in dom(location\_of\_service2)$

        grd004: $processes\_of\_partition(proc) = part$

        grd005: $part = current\_partition$

        grd014: $processes\_of\_partition(proc) \in dom(current\_partition\_flag)$

        grd006: $current\_partition\_flag(part) = TRUE$

        grd007: $proc = stop\_proc(core)$

        grd008: $reschedule \in BOOL$

        grd009: $current\_processes\_flag(core) = TRUE \Rightarrow proc \notin ran(current\_processes)$

        grd010: $reschedule = TRUE$

        grd011: $finished\_core2(core) = FALSE$

        grd012: $location\_of\_service2(core) = Stop \mapsto loc\_i$

        grd013: $\neg(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Stop \mapsto loc\_i)$

    then

        act001: $location\_of\_service2(core) := Stop \mapsto loc\_1$

        act002: $need\_reschedule := reschedule$

    end

Event stop_wf_bb_return_no_mutex ⟨ordinary⟩ $\widehat{=}$

extends stop_wf_bb_return_no_mutex

    any

        part

        proc

        core

    where

        grd001: $part \in PARTITIONS$

        grd002: $proc \in processes \land proc \in dom(processes\_of\_partition)$

        grd003: $core \in CORES \cap dom(stop\_proc) \land core \in dom(current\_processes\_flag) \land core \in dom(location\_of\_service2)$

        grd004: $processes\_of\_partition(proc) = part$

        grd005: $proc = stop\_proc(core)$

        grd006: $part = current\_partition$

grd013: $processes\_of\_partition(stop\_proc(core)) \in dom(current\_partition\_flag)$

grd012: $current\_partition\_flag(part) = TRUE$

grd007: $current\_processes\_flag(core) = TRUE \Rightarrow proc \notin ran(current\_processes)$

grd014: $stop\_proc(core) \in dom(preemption\_lock\_mutex)$

grd008: $preemption\_lock\_mutex(proc) = FALSE$

grd009: $finished\_core2(core) = FALSE$

grd010: $location\_of\_service2(core) = Stop \mapsto loc\_1$

grd011: $\neg(finished\_core(core) = FALSE \wedge location\_of\_service2(core) = Stop \mapsto loc\_1)$

then

act001: $location\_of\_service2(core) := Stop \mapsto loc\_r$

act002: $finished\_core2(core) := TRUE$

act003: $stop\_proc := \{core\} \lhd stop\_proc$

end

Event stop_wf_bb_mutex_zero $\langle ordinary \rangle \,\widehat{=}$

extends stop_wf_bb_mutex_zero

any

   part

   proc

   core

where

grd001: $part \in PARTITIONS$

grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition)$

grd003: $core \in CORES \cap dom(stop\_proc) \wedge core \in dom(current\_processes\_flag) \wedge core \in dom(location\_of\_service2)$

grd004: $processes\_of\_partition(proc) = part$

grd005: $proc = stop\_proc(core)$

grd006: $part = current\_partition$

grd012: $processes\_of\_partition(stop\_proc(core)) \in dom(current\_partition\_flag)$

grd007: $current\_partition\_flag(part) = TRUE$

grd008: $current\_processes\_flag(core) = TRUE \Rightarrow proc \notin ran(current\_processes)$

grd009: $finished\_core2(core) = FALSE$

grd010: $location\_of\_service2(core) = Stop \mapsto loc\_1$

grd011: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Stop \mapsto loc\_1)$

then

act001: $location\_of\_service2(core) := Stop \mapsto loc\_2$

act002: $locklevel\_of\_partition(part) := 0$

act003: $preempter\_of\_partition := \{part\} \lhd preempter\_of\_partition$

end

Event stop_wf_bb_mutex_avail $\langle ordinary \rangle \,\widehat{=}$

extends stop_wf_bb_mutex_avail

any

   part

   proc

   core

where

grd001: $part \in PARTITIONS$

grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition) \wedge proc \in dom(preemption\_lock\_mutex)$

grd003: $core \in CORES \cap dom(stop\_proc) \wedge core \in dom(current\_processes\_flag) \wedge core \in dom(location\_of\_service2)$

grd004: $processes\_of\_partition(proc) = part$

grd005: $proc = stop\_proc(core)$

grd006: $part = current\_partition$

grd013: $processes\_of\_partition(stop\_proc(core)) \in dom(current\_partition\_flag)$

grd007: $current\_partition\_flag(part) = TRUE$

grd008: $current\_processes\_flag(core) = TRUE \Rightarrow proc \notin ran(current\_processes)$

grd009: $preemption\_lock\_mutex(proc) = TRUE$

grd010: $finished\_core2(core) = FALSE$

grd011: $location\_of\_service2(core) = Stop \mapsto loc\_2$

$\qquad$ grd012: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Stop \mapsto loc\_2)$

then

$\qquad$ act001: $location\_of\_service2(core) := Stop \mapsto loc\_3$

$\qquad$ act002: $preemption\_lock\_mutex(proc) := FALSE$

end

Event stop_wf_bb_return_mutex $\langle ordinary \rangle \; \widehat{=}$

extends stop_wf_bb_return_mutex

$\quad$ **any**

$\qquad$ *part*

$\qquad$ *proc*

$\qquad$ *core*

$\quad$ **where**

$\qquad$ grd001: $part \in PARTITIONS$

$\qquad$ grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition)$

$\qquad$ grd003: $core \in CORES \cap dom(stop\_proc) \wedge core \in dom(current\_processes\_flag) \wedge core \in dom(location\_of\_service2)$

$\qquad$ grd004: $processes\_of\_partition(proc) = part$

$\qquad$ grd005: $part = current\_partition$

$\qquad$ grd011: $processes\_of\_partition(proc) \in dom(current\_partition\_flag)$

$\qquad$ grd006: $current\_partition\_flag(part) = TRUE$

$\qquad$ grd007: $current\_processes\_flag(core) = TRUE \Rightarrow proc \notin ran(current\_processes)$

$\qquad$ grd008: $finished\_core2(core) = FALSE$

$\qquad$ grd009: $location\_of\_service2(core) = Stop \mapsto loc\_3$

$\qquad$ grd010: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Stop \mapsto loc\_3)$

then

$\qquad$ act001: $location\_of\_service2(core) := Stop \mapsto loc\_r$

$\qquad$ act002: $finished\_core2(core) := TRUE$

$\qquad$ act003: $stop\_proc := \{core\} \vartriangleleft stop\_proc$

end

Event stop_wf_evt_init $\langle ordinary \rangle \; \widehat{=}$

extends stop_wf_evt_init

$\quad$ **any**

$\qquad$ *part*

$\qquad$ *proc*

$\qquad$ *newstate*

$\qquad$ *core*

$\qquad$ *r*

$\quad$ **where**

$\qquad$ grd001: $part \in PARTITIONS$

$\qquad$ grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state)$

$\qquad$ grd003: $newstate \in PROCESS\_STATES$

$\qquad$ grd004: $core \in CORES \wedge core \in dom(current\_processes\_flag)$

$\qquad$ grd005: $processes\_of\_partition(proc) = part$

$\qquad$ grd006: $partition\_mode(part) = PM\_COLD\_START \vee partition\_mode(part) = PM\_WARM\_START \vee partition\_mode(part) = PM\_NORMAL$

$\qquad$ grd017: $finished\_core2(core) = TRUE$

$\qquad$ grd101: $partition\_mode(part) = PM\_COLD\_START \vee partition\_mode(part) = PM\_WARM\_START \Rightarrow ((process\_state(proc) = PS\_Waiting \vee process\_state(proc) = PS\_WaitandSuspend) \wedge newstate = PS\_Dormant)$

$\qquad$ grd102: $partition\_mode(part) = PM\_NORMAL \Rightarrow ((process\_state(proc) = PS\_Ready \vee process\_state(proc) = PS\_Waiting \vee process\_state(proc) = PS\_WaitandSuspend \vee process\_state(proc) = PS\_Suspend \vee process\_state(proc) = PS\_Faulted) \wedge newstate = PS\_Dormant)$

$\qquad$ grd201: $current\_partition = part$

$\qquad$ grd205: $processes\_of\_partition(proc) \in dom(current\_partition\_flag)$

$\qquad$ grd202: $current\_partition\_flag(part) = TRUE$

$\qquad$ grd203: $current\_processes\_flag(core) = TRUE \Rightarrow proc \notin ran(current\_processes)$

$\qquad$ grd204: $newstate = PS\_Dormant$

$\qquad$ grd301: $r \in events \wedge proc \in processes\_waiting for\_events(r)$

then

act001: $process\_state(proc) := newstate$
act201: $location\_of\_service2(core) := Stop \mapsto loc\_i$
act202: $finished\_core2(core) := FALSE$
act203: $stop\_proc(core) := proc$
act204: $timeout\_trigger := \{proc\} \lhd timeout\_trigger$
act301: $processes\_waiting for\_events := processes\_waiting for\_events \lhd \{r \mapsto (processes\_waiting for\_events(r)\backslash \{proc\})\}$

end

Event stop_wf_evt_reschedule $\langle ordinary \rangle \;\widehat{=}$

extends stop_wf_evt_reschedule

    any

        part
        proc
        core
        reschedule

    where

grd001: $part \in PARTITIONS$
grd002: $proc \in processes \land proc \in dom(processes\_of\_partition)$
grd003: $core \in CORES \cap dom(stop\_proc) \land core \in dom(current\_processes\_flag) \land core \in dom(location\_of\_service2)$
grd004: $processes\_of\_partition(proc) = part$
grd005: $part = current\_partition$
grd014: $processes\_of\_partition(proc) \in dom(current\_partition\_flag)$
grd006: $current\_partition\_flag(part) = TRUE$
grd007: $proc = stop\_proc(core)$
grd008: $reschedule \in BOOL$
grd009: $current\_processes\_flag(core) = TRUE \Rightarrow proc \notin ran(current\_processes)$
grd010: $reschedule = TRUE$
grd011: $finished\_core2(core) = FALSE$
grd012: $location\_of\_service2(core) = Stop \mapsto loc\_i$
grd013: $\neg(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Stop \mapsto loc\_i)$

    then

act001: $location\_of\_service2(core) := Stop \mapsto loc\_1$
act002: $need\_reschedule := reschedule$

    end

Event stop_wf_evt_return_no_mutex $\langle ordinary \rangle \;\widehat{=}$

extends stop_wf_evt_return_no_mutex

    any

        part
        proc
        core

    where

grd001: $part \in PARTITIONS$
grd002: $proc \in processes \land proc \in dom(processes\_of\_partition)$
grd003: $core \in CORES \cap dom(stop\_proc) \land core \in dom(current\_processes\_flag) \land core \in dom(location\_of\_service2)$
grd004: $processes\_of\_partition(proc) = part$
grd005: $proc = stop\_proc(core)$
grd006: $part = current\_partition$
grd013: $processes\_of\_partition(stop\_proc(core)) \in dom(current\_partition\_flag)$
grd012: $current\_partition\_flag(part) = TRUE$
grd007: $current\_processes\_flag(core) = TRUE \Rightarrow proc \notin ran(current\_processes)$
grd014: $stop\_proc(core) \in dom(preemption\_lock\_mutex)$
grd008: $preemption\_lock\_mutex(proc) = FALSE$
grd009: $finished\_core2(core) = FALSE$
grd010: $location\_of\_service2(core) = Stop \mapsto loc\_1$
grd011: $\neg(finished\_core(core) = FALSE \land location\_of\_service2(core) = Stop \mapsto loc\_1)$

    then

act001: $location\_of\_service2(core) := Stop \mapsto loc\_r$

   act002: $finished\_core2(core) := TRUE$
   act003: $stop\_proc := \{core\} \lhd stop\_proc$
  end

Event stop_wf_evt_mutex_zero ⟨ordinary⟩ $\widehat{=}$
extends stop_wf_evt_mutex_zero
  any
   part
   proc
   core
  where
   grd001: $part \in PARTITIONS$
   grd002: $proc \in processes \land proc \in dom(processes\_of\_partition)$
   grd003: $core \in CORES \cap dom(stop\_proc) \land core \in dom(current\_processes\_flag) \land core \in dom(location\_of\_service2)$
   grd004: $processes\_of\_partition(proc) = part$
   grd005: $proc = stop\_proc(core)$
   grd006: $part = current\_partition$
   grd012: $processes\_of\_partition(stop\_proc(core)) \in dom(current\_partition\_flag)$
   grd007: $current\_partition\_flag(part) = TRUE$
   grd008: $current\_processes\_flag(core) = TRUE \Rightarrow proc \notin ran(current\_processes)$
   grd009: $finished\_core2(core) = FALSE$
   grd010: $location\_of\_service2(core) = Stop \mapsto loc\_1$
   grd011: $\neg(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Stop \mapsto loc\_1)$
  then
   act001: $location\_of\_service2(core) := Stop \mapsto loc\_2$
   act002: $locklevel\_of\_partition(part) := 0$
   act003: $preempter\_of\_partition := \{part\} \lhd preempter\_of\_partition$
  end

Event stop_wf_evt_mutex_avail ⟨ordinary⟩ $\widehat{=}$
extends stop_wf_evt_mutex_avail
  any
   part
   proc
   core
  where
   grd001: $part \in PARTITIONS$
   grd002: $proc \in processes \land proc \in dom(processes\_of\_partition) \land proc \in dom(preemption\_lock\_mutex)$

   grd003: $core \in CORES \cap dom(stop\_proc) \land core \in dom(current\_processes\_flag) \land core \in dom(location\_of\_service2)$
   grd004: $processes\_of\_partition(proc) = part$
   grd005: $proc = stop\_proc(core)$
   grd006: $part = current\_partition$
   grd013: $processes\_of\_partition(stop\_proc(core)) \in dom(current\_partition\_flag)$
   grd007: $current\_partition\_flag(part) = TRUE$
   grd008: $current\_processes\_flag(core) = TRUE \Rightarrow proc \notin ran(current\_processes)$
   grd009: $preemption\_lock\_mutex(proc) = TRUE$
   grd010: $finished\_core2(core) = FALSE$
   grd011: $location\_of\_service2(core) = Stop \mapsto loc\_2$
   grd012: $\neg(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Stop \mapsto loc\_2)$
  then
   act001: $location\_of\_service2(core) := Stop \mapsto loc\_3$
   act002: $preemption\_lock\_mutex(proc) := FALSE$
  end

Event stop_wf_evt_return_mutex ⟨ordinary⟩ $\widehat{=}$
extends stop_wf_evt_return_mutex
  any
   part

   $proc$
   $core$

where

  grd001: $part \in PARTITIONS$

  grd002: $proc \in processes \land proc \in dom(processes\_of\_partition)$

  grd003: $core \in CORES \cap dom(stop\_proc) \land core \in dom(current\_processes\_flag) \land core \in dom(location\_of\_service2)$

  grd004: $processes\_of\_partition(proc) = part$

  grd005: $part = current\_partition$

  grd011: $processes\_of\_partition(proc) \in dom(current\_partition\_flag)$

  grd006: $current\_partition\_flag(part) = TRUE$

  grd007: $current\_processes\_flag(core) = TRUE \Rightarrow proc \notin ran(current\_processes)$

  grd008: $finished\_core2(core) = FALSE$

  grd009: $location\_of\_service2(core) = Stop \mapsto loc\_3$

  grd010: $\neg(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Stop \mapsto loc\_3)$

then

  act001: $location\_of\_service2(core) := Stop \mapsto loc\_r$

  act002: $finished\_core2(core) := TRUE$

  act003: $stop\_proc := \{core\} \lhd stop\_proc$

end

Event start_aperiodprocess_instart_init ⟨ordinary⟩ $\widehat{=}$

extends start_aperiodprocess_instart_init

any

   $part$
   $proc$
   $newstate$
   $core$

where

  grd001: $part \in PARTITIONS$

  grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state) \cap dom(periodtype\_of\_process) \land proc \in dom(period\_of\_process)$

  grd003: $newstate \in PROCESS\_STATES$

  grd004: $core \in CORES$

  grd005: $processes\_of\_partition(proc) = part$

  grd017: $finished\_core2(core) = TRUE$

  grd101: $current\_partition = part$

  grd107: $part \in dom(current\_partition\_flag)$

  grd102: $current\_partition\_flag(part) = TRUE$

  grd103: $partition\_mode(part) = PM\_COLD\_START \lor partition\_mode(part) = PM\_WARM\_START$

  grd104: $process\_state(proc) = PS\_Dormant$

  grd105: $newstate = PS\_Waiting$

  grd106: $period\_of\_process(proc) = INFINITE\_TIME\_VALUE$

then

  act001: $process\_state(proc) := newstate$

  act101: $location\_of\_service2(core) := Start\_aperiod\_instart \mapsto loc\_i$

  act102: $process\_wait\_type(proc) := PROC\_WAIT\_PARTITIONNORMAL$

  act103: $finished\_core2(core) := FALSE$

  act104: $start\_aperiod\_proc(core) := proc$

end

Event start_aperiodprocess_instart_currentpri ⟨ordinary⟩ $\widehat{=}$

extends start_aperiodprocess_instart_currentpri

any

   $part$
   $proc$
   $core$

where

  grd001: $part \in PARTITIONS$

  grd002: $proc \in processes \land proc \in dom(processes\_of\_partition) \land proc \in dom(process\_state)$

grd003: $core \in CORES \cap dom(start\_aperiod\_proc) \wedge core \in dom(location\_of\_service2)$

grd004: $processes\_of\_partition(proc) = part$

grd005: $proc = start\_aperiod\_proc(core)$

grd012: $part \in dom(current\_partition\_flag)$

grd006: $current\_partition = part$

grd007: $current\_partition\_flag(part) = TRUE$

grd008: $process\_state(proc) = PS\_Waiting$

grd009: $finished\_core2(core) = FALSE$

grd010: $location\_of\_service2(core) = Start\_aperiod\_instart \mapsto loc\_i$

grd011: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Start\_aperiod\_instart \mapsto loc\_i)$

then

act001: $location\_of\_service2(core) := Start\_aperiod\_instart \mapsto loc\_1$

act002: $currentpriority\_of\_process(proc) := basepriority\_of\_process(proc)$

end

Event start_aperiodprocess_instart_return ⟨ordinary⟩ ≙

extends start_aperiodprocess_instart_return

    any

        part

        proc

        core

    where

grd001: $part \in PARTITIONS$

grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition) \wedge proc \in dom(process\_state)$

grd003: $core \in CORES \cap dom(start\_aperiod\_proc) \wedge core \in dom(location\_of\_service2)$

grd004: $proc = start\_aperiod\_proc(core)$

grd005: $processes\_of\_partition(proc) = part$

grd012: $part \in dom(current\_partition\_flag)$

grd006: $current\_partition = part$

grd007: $current\_partition\_flag(part) = TRUE$

grd008: $process\_state(proc) = PS\_Waiting$

grd009: $finished\_core2(core) = FALSE$

grd010: $location\_of\_service2(core) = Start\_aperiod\_instart \mapsto loc\_1$

grd011: $\neg(finished\_core2(core) = TRUE \wedge location\_of\_service2(core) = Start\_aperiod\_instart \mapsto loc\_1)$

then

act001: $location\_of\_service2(core) := Start\_aperiod\_instart \mapsto loc\_r$

act002: $finished\_core2(core) := TRUE$

act003: $start\_aperiod\_proc := \{core\} \lhd start\_aperiod\_proc$

end

Event start_aperiodprocess_innormal_init ⟨ordinary⟩ ≙

extends start_aperiodprocess_innormal_init

    any

        part

        proc

        newstate

        core

    where

grd001: $part \in PARTITIONS$

grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state) \cap dom(periodtype\_of\_process) \wedge proc \in dom(period\_of\_process)$

grd003: $newstate \in PROCESS\_STATES$

grd004: $core \in CORES \wedge core \in dom(current\_processes\_flag)$

grd005: $processes\_of\_partition(proc) = part$

grd017: $finished\_core2(core) = TRUE$

grd101: $current\_partition = part$

grd108: $part \in dom(current\_partition\_flag)$

grd102: $current\_partition\_flag(part) = TRUE$

grd103: $current\_processes\_flag(core) = TRUE$

      grd104: $partition\_mode(part) = PM\_NORMAL$
      grd105: $process\_state(proc) = PS\_Dormant$
      grd106: $newstate = PS\_Ready$
      grd107: $period\_of\_process(proc) = INFINITE\_TIME\_VALUE$

then

      act001: $process\_state(proc) := newstate$
      act101: $location\_of\_service2(core) := Start\_aperiod\_innormal \mapsto loc\_i$
      act102: $finished\_core2(core) := FALSE$
      act103: $start\_aperiod\_innormal\_proc(core) := proc$

end

Event start_aperiodprocess_innormal_deadline_time ⟨ordinary⟩ $\hat{=}$

extends start_aperiodprocess_innormal_deadline_time

    any

      part
      proc
      core

    where

      grd001: $part \in PARTITIONS$
      grd002: $proc \in processes \land proc \in dom(process\_state) \land proc \in dom(period\_of\_process)$
      grd003: $core \in CORES \cap dom(start\_aperiod\_innormal\_proc) \land core \in dom(current\_processes\_flag) \land$
        $core \in dom(location\_of\_service2)$
      grd004: $proc = start\_aperiod\_innormal\_proc(core)$
      grd014: $start\_aperiod\_innormal\_proc(core) \in dom(processes\_of\_partition)$
      grd005: $processes\_of\_partition(proc) = part$
      grd006: $current\_partition = part$
      grd015: $part \in dom(current\_partition\_flag)$
      grd007: $current\_partition\_flag(part) = TRUE$
      grd008: $current\_processes\_flag(core) = TRUE$
      grd009: $process\_state(proc) = PS\_Ready$
      grd010: $period\_of\_process(proc) = INFINITE\_TIME\_VALUE$
      grd011: $finished\_core2(core) = FALSE$
      grd012: $location\_of\_service2(core) = Start\_aperiod\_innormal \mapsto loc\_i$
      grd013: $\neg(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Start\_aperiod\_innormal \mapsto$
        $loc\_i)$

    then

      act001: $location\_of\_service2(core) := Start\_aperiod\_innormal \mapsto loc\_1$
      act002: $deadlinetime\_of\_process(proc) := clock\_tick * ONE\_TICK\_TIME + timecapacity\_of\_process(proc)$

    end

Event start_aperiodprocess_innormal_reschedule ⟨ordinary⟩ $\hat{=}$

extends start_aperiodprocess_innormal_reschedule

    any

      part
      proc
      core
      reschedule

    where

      grd001: $part \in PARTITIONS$
      grd002: $proc \in processes \land proc \in dom(processes\_of\_partition) \land proc \in dom(process\_state) \land$
        $proc \in dom(period\_of\_process)$
      grd003: $core \in CORES \cap dom(start\_aperiod\_innormal\_proc) \land core \in dom(current\_processes\_flag) \land$
        $core \in dom(location\_of\_service2)$
      grd004: $reschedule \in BOOL$
      grd005: $proc = start\_aperiod\_innormal\_proc(core)$
      grd006: $processes\_of\_partition(proc) = part$
      grd007: $current\_partition = part$
      grd016: $part \in dom(current\_partition\_flag)$
      grd008: $current\_partition\_flag(part) = TRUE$
      grd009: $current\_processes\_flag(core) = TRUE$

grd010: $process\_state(proc) = PS\_Ready$

grd011: $period\_of\_process(proc) = INFINITE\_TIME\_VALUE$

grd017: $processes\_of\_partition(start\_aperiod\_innormal\_proc(core)) \in dom(locklevel\_of\_partition)$

grd015: $(locklevel\_of\_partition(part) = 0 \Rightarrow reschedule = TRUE) \wedge (locklevel\_of\_partition(part) > 0 \Rightarrow reschedule = need\_reschedule)$

grd012: $finished\_core2(core) = FALSE$

grd013: $location\_of\_service2(core) = Start\_aperiod\_innormal \mapsto loc\_1$

grd014: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Start\_aperiod\_innormal \mapsto loc\_1)$

then

act001: $location\_of\_service2(core) := Start\_aperiod\_innormal \mapsto loc\_2$

act002: $need\_reschedule := reschedule$

end

Event start_aperiodprocess_innormal_currentpri ⟨ordinary⟩ ≙

extends start_aperiodprocess_innormal_currentpri

any

    part

    proc

    core

where

grd001: $part \in PARTITIONS$

grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition) \wedge proc \in dom(process\_state) \wedge proc \in dom(period\_of\_process)$

grd003: $core \in CORES \cap dom(start\_aperiod\_innormal\_proc) \wedge core \in dom(current\_processes\_flag) \wedge core \in dom(location\_of\_service2)$

grd004: $proc = start\_aperiod\_innormal\_proc(core)$

grd005: $processes\_of\_partition(proc) = part$

grd006: $part = current\_partition$

grd014: $part \in dom(current\_partition\_flag)$

grd007: $current\_partition\_flag(part) = TRUE$

grd008: $current\_processes\_flag(core) = TRUE$

grd009: $process\_state(proc) = PS\_Ready$

grd010: $period\_of\_process(proc) = INFINITE\_TIME\_VALUE$

grd011: $finished\_core2(core) = FALSE$

grd012: $location\_of\_service2(core) = Start\_aperiod\_innormal \mapsto loc\_2$

grd013: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Start\_aperiod\_innormal \mapsto loc\_2)$

then

act001: $location\_of\_service2(core) := Start\_aperiod\_innormal \mapsto loc\_3$

act002: $currentpriority\_of\_process(proc) := basepriority\_of\_process(proc)$

end

Event start_aperiodprocess_innormal_return ⟨ordinary⟩ ≙

extends start_aperiodprocess_innormal_return

any

    part

    proc

    core

where

grd001: $part \in PARTITIONS$

grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition) \wedge proc \in dom(process\_state) \wedge proc \in dom(period\_of\_process)$

grd003: $core \in CORES \cap dom(start\_aperiod\_innormal\_proc) \wedge core \in dom(current\_processes\_flag) \wedge core \in dom(location\_of\_service2)$

grd004: $proc = start\_aperiod\_innormal\_proc(core)$

grd005: $processes\_of\_partition(proc) = part$

grd006: $part = current\_partition$

grd014: $part \in dom(current\_partition\_flag)$

grd007: $current\_partition\_flag(part) = TRUE$

        grd008: $current\_processes\_flag(core) = TRUE$

        grd009: $process\_state(proc) = PS\_Ready$

        grd010: $period\_of\_process(proc) = INFINITE\_TIME\_VALUE$

        grd011: $finished\_core2(core) = FALSE$

        grd012: $location\_of\_service2(core) = Start\_aperiod\_innormal \mapsto loc\_3$

        grd013: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Start\_aperiod\_innormal \mapsto loc\_3)$

then

        act001: $location\_of\_service2(core) := Start\_aperiod\_innormal \mapsto loc\_r$

        act002: $finished\_core2(core) := TRUE$

        act003: $start\_aperiod\_innormal\_proc := \{core\} \lhd start\_aperiod\_innormal\_proc$

end

Event start_periodprocess_instart_init ⟨ordinary⟩ $\widehat{=}$

extends start_periodprocess_instart_init

    any

        part

        proc

        newstate

        core

    where

        grd001: $part \in PARTITIONS$

        grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state) \cap dom(periodtype\_of\_process) \wedge proc \in dom(period\_of\_process)$

        grd003: $newstate \in PROCESS\_STATES$

        grd004: $core \in CORES$

        grd005: $processes\_of\_partition(proc) = part$

        grd017: $finished\_core2(core) = TRUE$

        grd101: $partition\_mode(part) = PM\_COLD\_START \vee partition\_mode(part) = PM\_WARM\_START$

        grd107: $part \in dom(current\_partition\_flag)$

        grd102: $current\_partition = part$

        grd103: $current\_partition\_flag(part) = TRUE$

        grd104: $process\_state(proc) = PS\_Dormant$

        grd105: $newstate = PS\_Waiting$

        grd106: $period\_of\_process(proc) > 0$

    then

        act001: $process\_state(proc) := newstate$

        act101: $location\_of\_service2(core) := Start\_period\_instart \mapsto loc\_i$

        act102: $finished\_core2(core) := FALSE$

        act103: $process\_wait\_type(proc) := PROC\_WAIT\_PARTITIONNORMAL$

        act104: $start\_period\_instart\_proc(core) := proc$

    end

Event start_periodprocess_instart_currentpri ⟨ordinary⟩ $\widehat{=}$

extends start_periodprocess_instart_currentpri

    any

        part

        proc

        core

    where

        grd001: $part \in PARTITIONS$

        grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition) \wedge proc \in dom(process\_state) \wedge proc \in dom(period\_of\_process)$

        grd003: $core \in CORES \cap dom(start\_period\_instart\_proc) \wedge core \in dom(location\_of\_service2)$

        grd004: $proc = start\_period\_instart\_proc(core)$

        grd005: $processes\_of\_partition(proc) = part$

        grd006: $current\_partition = part$

        grd013: $part \in dom(current\_partition\_flag)$

        grd007: $current\_partition\_flag(part) = TRUE$

        grd008: $process\_state(proc) = PS\_Waiting$

grd009: $period\_of\_process(proc) > 0$
grd010: $finished\_core2(core) = FALSE$
grd011: $location\_of\_service2(core) = Start\_period\_instart \mapsto loc\_i$
grd012: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Start\_period\_instart \mapsto loc\_i)$

then

act001: $location\_of\_service2(core) := Start\_period\_instart \mapsto loc\_1$
act002: $currentpriority\_of\_process(proc) := basepriority\_of\_process(proc)$

end

Event start_periodprocess_instart_return ⟨ordinary⟩ $\widehat{=}$

extends start_periodprocess_instart_return

any

$part$
$proc$
$core$

where

grd001: $part \in PARTITIONS$
grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition) \wedge proc \in dom(process\_state) \wedge proc \in dom(period\_of\_process)$
grd003: $core \in CORES \cap dom(start\_period\_instart\_proc) \wedge core \in dom(location\_of\_service2)$
grd004: $proc = start\_period\_instart\_proc(core)$
grd005: $processes\_of\_partition(proc) = part$
grd006: $current\_partition = part$
grd013: $part \in dom(current\_partition\_flag)$
grd007: $current\_partition\_flag(part) = TRUE$
grd008: $process\_state(proc) = PS\_Waiting$
grd009: $period\_of\_process(proc) > 0$
grd010: $finished\_core2(core) = FALSE$
grd011: $location\_of\_service2(core) = Start\_period\_instart \mapsto loc\_1$
grd012: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Start\_period\_instart \mapsto loc\_1)$

then

act001: $location\_of\_service2(core) := Start\_period\_instart \mapsto loc\_r$
act002: $finished\_core2(core) := TRUE$
act003: $start\_period\_instart\_proc := \{core\} \mathbin{\lhd\mkern-9mu-} start\_period\_instart\_proc$

end

Event start_periodprocess_innormal_init ⟨ordinary⟩ $\widehat{=}$

extends start_periodprocess_innormal_init

any

$part$
$proc$
$newstate$
$core$

where

grd001: $part \in PARTITIONS$
grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state) \cap dom(periodtype\_of\_process) \wedge proc \in dom(period\_of\_process)$
grd003: $newstate \in PROCESS\_STATES$
grd004: $core \in CORES \wedge core \in dom(current\_processes\_flag)$
grd005: $processes\_of\_partition(proc) = part$
grd017: $finished\_core2(core) = TRUE$
grd101: $partition\_mode(part) = PM\_NORMAL$
grd102: $current\_partition = part$
grd108: $part \in dom(current\_partition\_flag)$
grd109: $proc \in dom(releasepoint\_of\_process)$
grd103: $current\_partition\_flag(part) = TRUE$
grd104: $current\_processes\_flag(core) = TRUE$
grd105: $process\_state(proc) = PS\_Dormant$
grd106: $newstate = PS\_Waiting$

   grd107: $period\_of\_process(proc) > 0$
   grd110: $proc \notin ran(current\_processes)$
  then
   act001: $process\_state(proc) := newstate$
   act101: $location\_of\_service2(core) := Start\_period\_innormal \mapsto loc\_i$
   act102: $finished\_core2(core) := FALSE$
   act103: $process\_wait\_type(proc) := PROC\_WAIT\_PERIOD$
   act104: $start\_period\_innormal\_proc(core) := proc$
  end

Event start_periodprocess_innormal_releasepoint ⟨ordinary⟩ $\widehat{=}$

extends start_periodprocess_innormal_releasepoint

  any
   part
   proc
   core
   fstrl
  where
   grd001: $part \in PARTITIONS$
   grd002: $proc \in processes \land proc \in dom(processes\_of\_partition) \land proc \in dom(process\_state) \land$
    $proc \in dom(period\_of\_process)$
   grd003: $core \in CORES \cap dom(start\_period\_innormal\_proc) \land core \in dom(current\_processes\_flag) \land$
    $core \in dom(location\_of\_service2)$
   grd015: $fstrl \in \mathbb{N}_1$
   grd004: $proc = start\_period\_innormal\_proc(core)$
   grd005: $processes\_of\_partition(proc) = part$
   grd006: $partition\_mode(part) = PM\_NORMAL$
   grd007: $current\_partition = part$
   grd017: $part \in dom(current\_partition\_flag)$
   grd008: $current\_partition\_flag(part) = TRUE$
   grd009: $current\_processes\_flag(core) = TRUE$
   grd010: $process\_state(proc) = PS\_Waiting$
   grd011: $period\_of\_process(proc) > 0$
   grd016: $\exists x, y, b \cdot (((x \mapsto y) \mapsto b) = firstperiodicprocstart\_timeWindow\_of\_Partition(part) \Rightarrow$
    $fstrl = ((clock\_tick * ONE\_TICK\_TIME)/majorFrame + 1) * majorFrame + x)$
   grd012: $finished\_core2(core) = FALSE$
   grd013: $location\_of\_service2(core) = Start\_period\_innormal \mapsto loc\_i$
   grd014: $\neg(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Start\_period\_innormal \mapsto$
    $loc\_i)$
  then
   act001: $location\_of\_service2(core) := Start\_period\_innormal \mapsto loc\_1$
   act002: $releasepoint\_of\_process(proc) := fstrl$
  end

Event start_periodprocess_innormal_deadlinetime ⟨ordinary⟩ $\widehat{=}$

extends start_periodprocess_innormal_deadlinetime

  any
   part
   proc
   core
   fstrl
  where
   grd001: $part \in PARTITIONS$
   grd002: $proc \in processes \land proc \in dom(processes\_of\_partition) \land proc \in dom(process\_state) \land$
    $proc \in dom(period\_of\_process)$
   grd003: $core \in CORES \cap dom(start\_period\_innormal\_proc) \land core \in dom(current\_processes\_flag) \land$
    $core \in dom(location\_of\_service2)$
   grd004: $fstrl \in \mathbb{N}_1$
   grd005: $proc = start\_period\_innormal\_proc(core)$
   grd006: $processes\_of\_partition(proc) = part$
   grd007: $partition\_mode(part) = PM\_NORMAL$

  grd008: $current\_partition = part$

  grd017: $part \in dom(current\_partition\_flag)$

  grd009: $current\_partition\_flag(part) = TRUE$

  grd010: $current\_processes\_flag(core) = TRUE$

  grd011: $process\_state(proc) = PS\_Waiting$

  grd012: $period\_of\_process(proc) > 0$

  grd013: $\exists x, y, b \cdot (((x \mapsto y) \mapsto b) = firstperiodicprocstart\_timeWindow\_of\_Partition(part) \Rightarrow fstrl = ((clock\_tick * ONE\_TICK\_TIME)/majorFrame + 1) * majorFrame + x)$

  grd014: $finished\_core2(core) = FALSE$

  grd015: $location\_of\_service2(core) = Start\_period\_innormal \mapsto loc\_1$

  grd016: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Start\_period\_innormal \mapsto loc\_1)$

then

  act001: $location\_of\_service2(core) := Start\_period\_innormal \mapsto loc\_2$

  act002: $deadlinetime\_of\_process(proc) := fstrl + timecapacity\_of\_process(proc)$

end

Event start_periodprocess_innormal_currentpri ⟨ordinary⟩ $\widehat{=}$

extends start_periodprocess_innormal_currentpri

 any

  $part$

  $proc$

  $core$

 where

  grd001: $part \in PARTITIONS$

  grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition) \wedge proc \in dom(process\_state) \wedge proc \in dom(period\_of\_process)$

  grd003: $core \in CORES \cap dom(start\_period\_innormal\_proc) \wedge core \in dom(current\_processes\_flag) \wedge core \in dom(location\_of\_service2)$

  grd004: $proc = start\_period\_innormal\_proc(core)$

  grd005: $processes\_of\_partition(proc) = part$

  grd006: $partition\_mode(part) = PM\_NORMAL$

  grd007: $current\_partition = part$

  grd015: $part \in dom(current\_partition\_flag)$

  grd008: $current\_partition\_flag(part) = TRUE$

  grd009: $current\_processes\_flag(core) = TRUE$

  grd010: $process\_state(proc) = PS\_Waiting$

  grd011: $period\_of\_process(proc) > 0$

  grd012: $finished\_core2(core) = FALSE$

  grd013: $location\_of\_service2(core) = Start\_period\_innormal \mapsto loc\_2$

  grd014: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Start\_period\_innormal \mapsto loc\_2)$

 then

  act001: $location\_of\_service2(core) := Start\_period\_innormal \mapsto loc\_3$

  act002: $currentpriority\_of\_process(proc) := basepriority\_of\_process(proc)$

 end

Event start_periodprocess_innormal_return ⟨ordinary⟩ $\widehat{=}$

extends start_periodprocess_innormal_return

 any

  $part$

  $proc$

  $core$

 where

  grd001: $part \in PARTITIONS$

  grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition) \wedge proc \in dom(process\_state) \wedge proc \in dom(period\_of\_process)$

  grd003: $core \in CORES \cap dom(start\_period\_innormal\_proc) \wedge core \in dom(current\_processes\_flag) \wedge core \in dom(location\_of\_service2)$

  grd004: $proc = start\_period\_innormal\_proc(core)$

  grd005: $processes\_of\_partition(proc) = part$

grd006: $partition\_mode(part) = PM\_NORMAL$

grd007: $current\_partition = part$

grd015: $part \in dom(current\_partition\_flag)$

grd008: $current\_partition\_flag(part) = TRUE$

grd009: $current\_processes\_flag(core) = TRUE$

grd010: $process\_state(proc) = PS\_Waiting$

grd011: $period\_of\_process(proc) > 0$

grd012: $finished\_core2(core) = FALSE$

grd013: $location\_of\_service2(core) = Start\_period\_innormal \mapsto loc\_3$

grd014: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Start\_period\_innormal \mapsto loc\_3)$

then

act001: $location\_of\_service2(core) := Start\_period\_innormal \mapsto loc\_r$

act002: $finished\_core2(core) := TRUE$

act003: $start\_period\_innormal\_proc := \{core\} \lhd start\_period\_innormal\_proc$

end

Event delay_start_aperiodprocess_instart_init ⟨ordinary⟩ $\widehat{=}$

extends delay_start_aperiodprocess_instart_init

any

> *part*
>
> *proc*
>
> *newstate*
>
> *core*
>
> *delaytime*

where

grd001: $part \in PARTITIONS$

grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state) \wedge proc \in dom(period\_of\_process)$

grd003: $newstate \in PROCESS\_STATES$

grd004: $core \in CORES$

grd005: $processes\_of\_partition(proc) = part$

grd017: $finished\_core2(core) = TRUE$

grd101: $current\_partition = part$

grd108: $part \in dom(current\_partition\_flag)$

grd102: $current\_partition\_flag(part) = TRUE$

grd103: $partition\_mode(part) = PM\_COLD\_START \vee partition\_mode(part) = PM\_WARM\_START$

grd104: $process\_state(proc) = PS\_Dormant$

grd105: $newstate = PS\_Waiting$

grd106: $period\_of\_process(proc) = INFINITE\_TIME\_VALUE$

grd107: $delaytime \in \mathbb{N} \wedge delaytime \neq INFINITE\_TIME\_VALUE$

then

act001: $process\_state(proc) := newstate$

act101: $location\_of\_service2(core) := Delay\_start\_aperiod\_instart \mapsto loc\_i$

act102: $process\_wait\_type(proc) := PROC\_WAIT\_DELAY$

act103: $finished\_core2(core) := FALSE$

act104: $delay\_start\_ainstart\_proc(core) := proc$

act105: $delaytime\_of\_process(proc) := delaytime$

end

Event delay_start_aperiodprocess_instart_currentpri ⟨ordinary⟩ $\widehat{=}$

extends delay_start_aperiodprocess_instart_currentpri

any

> *part*
>
> *proc*
>
> *core*

where

grd001: $part \in PARTITIONS$

grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition) \wedge proc \in dom(process\_state) \wedge proc \in dom(period\_of\_process)$

grd003: $core \in CORES \cap dom(delay\_start\_ainstart\_proc) \wedge core \in dom(location\_of\_service2)$

grd004: $processes\_of\_partition(proc) = part$

grd005: $proc = delay\_start\_ainstart\_proc(core)$

grd006: $current\_partition = part$

grd013: $part \in dom(current\_partition\_flag)$

grd007: $current\_partition\_flag(part) = TRUE$

grd008: $process\_state(proc) = PS\_Waiting$

grd009: $period\_of\_process(proc) = INFINITE\_TIME\_VALUE$

grd010: $finished\_core2(core) = FALSE$

grd011: $location\_of\_service2(core) = Delay\_start\_aperiod\_instart \mapsto loc\_i$

grd012: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Delay\_start\_aperiod\_instart \mapsto loc\_i)$

then

act001: $location\_of\_service2(core) := Delay\_start\_aperiod\_instart \mapsto loc\_1$

act002: $currentpriority\_of\_process(proc) := basepriority\_of\_process(proc)$

end

Event delay_start_aperiodprocess_instart_return $\langle ordinary \rangle \; \widehat{=}$

extends delay_start_aperiodprocess_instart_return

any

  part

  proc

  core

where

grd001: $part \in PARTITIONS$

grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition) \wedge proc \in dom(process\_state) \wedge proc \in dom(period\_of\_process)$

grd003: $core \in CORES \cap dom(delay\_start\_ainstart\_proc) \wedge core \in dom(location\_of\_service2)$

grd004: $processes\_of\_partition(proc) = part$

grd005: $proc = delay\_start\_ainstart\_proc(core)$

grd006: $current\_partition = part$

grd013: $part \in dom(current\_partition\_flag)$

grd007: $current\_partition\_flag(part) = TRUE$

grd008: $process\_state(proc) = PS\_Waiting$

grd009: $period\_of\_process(proc) = INFINITE\_TIME\_VALUE$

grd010: $finished\_core2(core) = FALSE$

grd011: $location\_of\_service2(core) = Delay\_start\_aperiod\_instart \mapsto loc\_1$

grd012: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Delay\_start\_aperiod\_instart \mapsto loc\_1)$

then

act001: $location\_of\_service2(core) := Delay\_start\_aperiod\_instart \mapsto loc\_r$

act002: $finished\_core2(core) := TRUE$

act003: $delay\_start\_ainstart\_proc := \{core\} \lhd delay\_start\_ainstart\_proc$

end

Event delay_start_aperiodprocess_innormal_init $\langle ordinary \rangle \; \widehat{=}$

extends delay_start_aperiodprocess_innormal_init

any

  part

  proc

  newstate

  core

  delaytime

where

grd001: $part \in PARTITIONS$

grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state) \wedge proc \in dom(period\_of\_process)$

grd003: $newstate \in PROCESS\_STATES$

grd004: $core \in CORES \wedge core \in dom(current\_processes\_flag)$

grd005: $processes\_of\_partition(proc) = part$

grd102: $newstate = PS\_Waiting$

grd017: $finished\_core2(core) = TRUE$
grd201: $current\_partition = part$
grd209: $part \in dom(current\_partition\_flag)$
grd210: $proc \in dom(delaytime\_of\_process) \wedge proc \in dom(process\_wait\_type)$
grd202: $current\_partition\_flag(part) = TRUE$
grd203: $current\_processes\_flag(core) = TRUE$
grd204: $partition\_mode(part) = PM\_NORMAL$
grd205: $process\_state(proc) = PS\_Dormant$
grd206: $delaytime > 0 \wedge delaytime \neq INFINITE\_TIME\_VALUE$
grd207: $newstate = PS\_Waiting$
grd208: $period\_of\_process(proc) = INFINITE\_TIME\_VALUE$
grd211: $proc \notin ran(current\_processes)$

then

act001: $process\_state(proc) := newstate$
act201: $location\_of\_service2(core) := Delay\_start\_aperiod\_innormal \mapsto loc\_i$
act202: $finished\_core2(core) := FALSE$
act203: $delay\_start\_ainnormal\_proc(core) := proc$
act204: $delay\_start\_ainnormal\_delaytime(core) := delaytime$
act205: $process\_wait\_type(proc) := PROC\_WAIT\_DELAY$

end

Event delay_start_aperiodprocess_innormal_deadline_time ⟨ordinary⟩ $\widehat{=}$
extends delay_start_aperiodprocess_innormal_deadline_time

any

    part
    proc
    core
    delaytime

where

grd001: $part \in PARTITIONS$
grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition) \wedge proc \in dom(process\_state) \wedge$ $proc \in dom(period\_of\_process)$
grd003: $core \in CORES \cap dom(delay\_start\_ainnormal\_proc) \cap dom(delay\_start\_ainnormal\_delaytime) \wedge$ $core \in dom(current\_processes\_flag) \wedge core \in dom(location\_of\_service2)$
grd014: $delaytime \in \mathbb{N}$
grd004: $proc = delay\_start\_ainnormal\_proc(core)$
grd005: $processes\_of\_partition(proc) = part$
grd006: $current\_partition = part$
grd016: $part \in dom(current\_partition\_flag)$
grd007: $current\_partition\_flag(part) = TRUE$
grd008: $current\_processes\_flag(core) = TRUE$
grd009: $process\_state(proc) = PS\_Waiting$
grd010: $period\_of\_process(proc) = INFINITE\_TIME\_VALUE$
grd015: $delaytime = delay\_start\_ainnormal\_delaytime(core)$
grd011: $finished\_core2(core) = FALSE$
grd012: $location\_of\_service2(core) = Delay\_start\_aperiod\_innormal \mapsto loc\_i$
grd013: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Delay\_start\_aperiod\_innormal \mapsto$ $loc\_i)$

then

act001: $location\_of\_service2(core) := Delay\_start\_aperiod\_innormal \mapsto loc\_1$
act002: $deadlinetime\_of\_process(proc) := clock\_tick * ONE\_TICK\_TIME + timecapacity\_of\_process(proc) +$ $delaytime$

end

Event delay_start_aperiodprocess_innormal_trigger ⟨ordinary⟩ $\widehat{=}$
extends delay_start_aperiodprocess_innormal_trigger

any

    part
    proc
    core
    delaytime

where

      grd001: $part \in PARTITIONS$

      grd002: $proc \in processes \land proc \in dom(processes\_of\_partition) \land proc \in dom(process\_state) \land$
      $proc \in dom(period\_of\_process)$

      grd003: $core \in CORES \cap dom(delay\_start\_ainnormal\_delaytime) \cap dom(delay\_start\_ainnormal\_proc) \land$
      $core \in dom(current\_processes\_flag) \land core \in dom(location\_of\_service2)$

      grd004: $delaytime \in \mathbb{N}$

      grd005: $proc = delay\_start\_ainnormal\_proc(core)$

      grd006: $delaytime = delay\_start\_ainnormal\_delaytime(core)$

      grd007: $processes\_of\_partition(proc) = part$

      grd008: $current\_partition = part$

      grd016: $part \in dom(current\_partition\_flag)$

      grd009: $current\_partition\_flag(part) = TRUE$

      grd010: $current\_processes\_flag(core) = TRUE$

      grd011: $process\_state(proc) = PS\_Waiting$

      grd012: $period\_of\_process(proc) = INFINITE\_TIME\_VALUE$

      grd013: $finished\_core2(core) = FALSE$

      grd014: $location\_of\_service2(core) = Delay\_start\_aperiod\_innormal \mapsto loc\_1$

      grd015: $\neg(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Delay\_start\_aperiod\_innormal \mapsto$
      $loc\_1)$

then

      act001: $location\_of\_service2(core) := Delay\_start\_aperiod\_innormal \mapsto loc\_2$

      act002: $timeout\_trigger := timeout\_trigger \Leftarrow \{proc \mapsto (PS\_Ready \mapsto (delaytime + clock\_tick *$
      $ONE\_TICK\_TIME))\}$

end

Event delay_start_aperiodprocess_innormal_reschedule ⟨ordinary⟩ $\widehat{=}$

extends delay_start_aperiodprocess_innormal_reschedule

any

      part

      proc

      core

      reschedule

where

      grd001: $part \in PARTITIONS$

      grd002: $proc \in processes \land proc \in dom(processes\_of\_partition) \land proc \in dom(process\_state) \land$
      $proc \in dom(period\_of\_process)$

      grd003: $core \in CORES \cap dom(delay\_start\_ainnormal\_proc) \land core \in dom(current\_processes\_flag) \land$
      $core \in dom(location\_of\_service2)$

      grd014: $reschedule \in BOOL$

      grd004: $proc = delay\_start\_ainnormal\_proc(core)$

      grd005: $processes\_of\_partition(proc) = part$

      grd006: $current\_partition = part$

      grd016: $part \in dom(current\_partition\_flag)$

      grd007: $current\_partition\_flag(part) = TRUE$

      grd008: $current\_processes\_flag(core) = TRUE$

      grd009: $process\_state(proc) = PS\_Waiting$

      grd010: $period\_of\_process(proc) = INFINITE\_TIME\_VALUE$

      grd017: $processes\_of\_partition(delay\_start\_ainnormal\_proc(core)) \in dom(locklevel\_of\_partition)$

      grd015: $(locklevel\_of\_partition(part) = 0 \Rightarrow reschedule = TRUE) \land (locklevel\_of\_partition(part) >$
      $0 \Rightarrow reschedule = need\_reschedule)$

      grd011: $finished\_core2(core) = FALSE$

      grd012: $location\_of\_service2(core) = Delay\_start\_aperiod\_innormal \mapsto loc\_2$

      grd013: $\neg(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Delay\_start\_aperiod\_innormal \mapsto$
      $loc\_2)$

then

      act001: $location\_of\_service2(core) := Delay\_start\_aperiod\_innormal \mapsto loc\_3$

      act002: $need\_reschedule := reschedule$

end

Event delay_start_aperiodprocess_innormal_currentpri ⟨ordinary⟩ ≙

extends delay_start_aperiodprocess_innormal_currentpri

    any

        part

        proc

        core

    where

        grd001: $part \in PARTITIONS$

        grd002: $proc \in processes \land proc \in dom(processes\_of\_partition) \land proc \in dom(process\_state) \land proc \in dom(period\_of\_process)$

        grd003: $core \in CORES \cap dom(delay\_start\_ainnormal\_proc) \land core \in dom(current\_processes\_flag) \land core \in dom(location\_of\_service2)$

        grd004: $proc = delay\_start\_ainnormal\_proc(core)$

        grd005: $processes\_of\_partition(proc) = part$

        grd006: $current\_partition = part$

        grd014: $part \in dom(current\_partition\_flag)$

        grd007: $current\_partition\_flag(part) = TRUE$

        grd008: $current\_processes\_flag(core) = TRUE$

        grd009: $process\_state(proc) = PS\_Waiting$

        grd010: $period\_of\_process(proc) = INFINITE\_TIME\_VALUE$

        grd011: $finished\_core2(core) = FALSE$

        grd012: $location\_of\_service2(core) = Delay\_start\_aperiod\_innormal \mapsto loc\_3$

        grd013: $\neg(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Delay\_start\_aperiod\_innormal \mapsto loc\_3)$

    then

        act001: $location\_of\_service2(core) := Delay\_start\_aperiod\_innormal \mapsto loc\_4$

        act002: $currentpriority\_of\_process(proc) := basepriority\_of\_process(proc)$

    end

Event delay_start_aperiodprocess_innormal_return ⟨ordinary⟩ ≙

extends delay_start_aperiodprocess_innormal_return

    any

        part

        proc

        core

    where

        grd001: $part \in PARTITIONS$

        grd002: $proc \in processes \land proc \in dom(processes\_of\_partition) \land proc \in dom(process\_state) \land proc \in dom(period\_of\_process)$

        grd003: $core \in CORES \cap dom(delay\_start\_ainnormal\_proc) \cap dom(delay\_start\_ainnormal\_delaytime) \land core \in dom(current\_processes\_flag) \land core \in dom(location\_of\_service2)$

        grd004: $proc = delay\_start\_ainnormal\_proc(core)$

        grd005: $processes\_of\_partition(proc) = part$

        grd006: $current\_partition = part$

        grd014: $part \in dom(current\_partition\_flag)$

        grd007: $current\_partition\_flag(part) = TRUE$

        grd008: $current\_processes\_flag(core) = TRUE$

        grd009: $process\_state(proc) = PS\_Waiting$

        grd010: $period\_of\_process(proc) = INFINITE\_TIME\_VALUE$

        grd011: $finished\_core2(core) = FALSE$

        grd012: $location\_of\_service2(core) = Delay\_start\_aperiod\_innormal \mapsto loc\_4$

        grd013: $\neg(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Delay\_start\_aperiod\_innormal \mapsto loc\_4)$

    then

        act001: $location\_of\_service2(core) := Delay\_start\_aperiod\_innormal \mapsto loc\_r$

        act002: $finished\_core2(core) := TRUE$

        act003: $delay\_start\_ainnormal\_proc := \{core\} \lhd delay\_start\_ainnormal\_proc$

        act004: $delay\_start\_ainnormal\_delaytime := \{core\} \lhd delay\_start\_ainnormal\_delaytime$

    end

Event delay_start_periodprocess_instart_init ⟨ordinary⟩ ≙

extends delay_start_periodprocess_instart_init

    any

        part

        proc

        newstate

        core

        delaytime

    where

        grd001: $part \in PARTITIONS$

        grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state) \land proc \in dom(period\_of\_process)$

        grd003: $newstate \in PROCESS\_STATES$

        grd004: $core \in CORES$

        grd005: $processes\_of\_partition(proc) = part$

        grd017: $finished\_core2(core) = TRUE$

        grd201: $current\_partition = part$

        grd208: $part \in dom(current\_partition\_flag)$

        grd202: $current\_partition\_flag(part) = TRUE$

        grd203: $partition\_mode(part) = PM\_COLD\_START \lor partition\_mode(part) = PM\_WARM\_START$

        grd204: $process\_state(proc) = PS\_Dormant$

        grd205: $newstate = PS\_Waiting$

        grd206: $period\_of\_process(proc) > 0$

        grd207: $delaytime \in \mathbb{N} \land delaytime \neq INFINITE\_TIME\_VALUE \land delaytime < period\_of\_process(proc)$

    then

        act001: $process\_state(proc) := newstate$

        act201: $location\_of\_service2(core) := Delay\_start\_period\_instart \mapsto loc\_i$

        act202: $process\_wait\_type(proc) := PROC\_WAIT\_DELAY$

        act203: $finished\_core2(core) := FALSE$

        act204: $delaytime\_of\_process(proc) := delaytime$

        act205: $delay\_start\_instart\_proc(core) := proc$

    end

Event delay_start_periodprocess_instart_currentpri ⟨ordinary⟩ $\hat{=}$

extends delay_start_periodprocess_instart_currentpri

    any

        part

        proc

        core

    where

        grd001: $part \in PARTITIONS$

        grd002: $proc \in processes \land proc \in dom(processes\_of\_partition) \land proc \in dom(process\_state) \land proc \in dom(period\_of\_process)$

        grd003: $core \in CORES \cap dom(delay\_start\_instart\_proc) \land core \in dom(location\_of\_service2)$

        grd004: $processes\_of\_partition(proc) = part$

        grd005: $proc = delay\_start\_instart\_proc(core)$

        grd006: $current\_partition = part$

        grd013: $part \in dom(current\_partition\_flag)$

        grd007: $current\_partition\_flag(part) = TRUE$

        grd008: $process\_state(proc) = PS\_Waiting$

        grd009: $period\_of\_process(proc) > 0$

        grd010: $finished\_core2(core) = FALSE$

        grd011: $location\_of\_service2(core) = Delay\_start\_period\_instart \mapsto loc\_i$

        grd012: $\neg(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Delay\_start\_period\_instart \mapsto loc\_i)$

    then

        act001: $location\_of\_service2(core) := Delay\_start\_period\_instart \mapsto loc\_1$

        act002: $currentpriority\_of\_process(proc) := basepriority\_of\_process(proc)$

    end

Event delay_start_periodprocess_instart_return ⟨ordinary⟩ ≙
extends delay_start_periodprocess_instart_return
    any
        part
        proc
        core
    where
        grd001: $part \in PARTITIONS$
        grd002: $proc \in processes \land proc \in dom(processes\_of\_partition) \land proc \in dom(process\_state) \land$
          $proc \in dom(period\_of\_process)$
        grd003: $core \in CORES \cap dom(delay\_start\_instart\_proc) \land core \in dom(location\_of\_service2)$
        grd004: $processes\_of\_partition(proc) = part$
        grd005: $proc = delay\_start\_instart\_proc(core)$
        grd006: $current\_partition = part$
        grd013: $part \in dom(current\_partition\_flag)$
        grd007: $current\_partition\_flag(part) = TRUE$
        grd008: $process\_state(proc) = PS\_Waiting$
        grd009: $period\_of\_process(proc) > 0$
        grd010: $finished\_core2(core) = FALSE$
        grd011: $location\_of\_service2(core) = Delay\_start\_period\_instart \mapsto loc\_1$
        grd012: $\lnot(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Delay\_start\_period\_instart \mapsto$
          $loc\_1)$
    then
        act001: $location\_of\_service2(core) := Delay\_start\_period\_instart \mapsto loc\_r$
        act002: $finished\_core2(core) := TRUE$
        act003: $delay\_start\_instart\_proc := \{core\} \lhd delay\_start\_instart\_proc$
    end
Event delay_start_periodprocess_innormal_init ⟨ordinary⟩ ≙
extends delay_start_periodprocess_innormal_init
    any
        part
        proc
        newstate
        core
        delaytime
    where
        grd001: $part \in PARTITIONS$
        grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state) \land proc \in dom(period\_of\_process)$

        grd003: $newstate \in PROCESS\_STATES$
        grd004: $core \in CORES \land core \in dom(current\_processes\_flag)$
        grd005: $processes\_of\_partition(proc) = part$
        grd017: $finished\_core2(core) = TRUE$
        grd102: $newstate = PS\_Waiting$
        grd201: $partition\_mode(part) = PM\_NORMAL$
        grd202: $current\_partition = part$
        grd208: $part \in dom(current\_partition\_flag)$
        grd209: $proc \in dom(releasepoint\_of\_process)$
        grd203: $current\_partition\_flag(part) = TRUE$
        grd204: $current\_processes\_flag(core) = TRUE$
        grd205: $process\_state(proc) = PS\_Dormant$
        grd206: $period\_of\_process(proc) > 0$
        grd207: $delaytime \in \mathbb{N} \land delaytime > 0 \land delaytime < period\_of\_process(proc)$
        grd210: $proc \notin ran(current\_processes)$
    then
        act001: $process\_state(proc) := newstate$
        act201: $location\_of\_service2(core) := Delay\_start\_period\_innormal \mapsto loc\_i$
        act202: $finished\_core2(core) := FALSE$
        act203: $process\_wait\_type(proc) := PROC\_WAIT\_DELAY$

          act204: $delaytime\_of\_process(proc) := delaytime$

          act205: $delay\_start\_innormal\_proc(core) := proc$

          act206: $delay\_start\_innormal\_delaytime(core) := delaytime$

    end

Event delay_start_periodprocess_innormal_releasepoint $\langle \text{ordinary} \rangle \ \widehat{=}$

extends delay_start_periodprocess_innormal_releasepoint

    any

        part

        proc

        core

        fstrl

        delaytime

    where

        grd001: $part \in PARTITIONS$

        grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition) \wedge proc \in dom(process\_state) \wedge$
$proc \in dom(period\_of\_process)$

        grd003: $core \in CORES \cap dom(delay\_start\_innormal\_proc) \cap dom(delay\_start\_ainnormal\_delaytime) \wedge$
$core \in dom(current\_processes\_flag) \wedge core \in dom(location\_of\_service2)$

        grd006: $fstrl \in \mathbb{N}_1$

        grd017: $delaytime = delay\_start\_ainnormal\_delaytime(core)$

        grd004: $processes\_of\_partition(proc) = part$

        grd005: $proc = delay\_start\_innormal\_proc(core)$

        grd007: $partition\_mode(part) = PM\_NORMAL$

        grd008: $current\_partition = part$

        grd018: $part \in dom(current\_partition\_flag)$

        grd009: $current\_partition\_flag(part) = TRUE$

        grd010: $current\_processes\_flag(core) = TRUE$

        grd011: $process\_state(proc) = PS\_Waiting$

        grd012: $period\_of\_process(proc) > 0$

        grd013: $\exists x, y, b \cdot (((x \mapsto y) \mapsto b) = firstperiodicprocstart\_timeWindow\_of\_Partition(part) \Rightarrow$
$fstrl = ((clock\_tick * ONE\_TICK\_TIME)/majorFrame + 1) * majorFrame + x)$

        grd014: $finished\_core2(core) = FALSE$

        grd015: $location\_of\_service2(core) = Delay\_start\_period\_innormal \mapsto loc\_i$

        grd016: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Delay\_start\_period\_innormal \mapsto$
$loc\_i)$

    then

        act001: $location\_of\_service2(core) := Delay\_start\_period\_innormal \mapsto loc\_1$

        act002: $releasepoint\_of\_process(proc) := fstrl + delaytime$

    end

Event delay_start_periodprocess_innormal_deadlinetime $\langle \text{ordinary} \rangle \ \widehat{=}$

extends delay_start_periodprocess_innormal_deadlinetime

    any

        part

        proc

        core

        fstrl

        delaytime

    where

        grd001: $part \in PARTITIONS$

        grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition) \wedge proc \in dom(process\_state) \wedge$
$proc \in dom(period\_of\_process)$

        grd003: $core \in CORES \cap dom(delay\_start\_innormal\_delaytime) \cap dom(delay\_start\_innormal\_proc) \wedge$
$core \in dom(current\_processes\_flag) \wedge core \in dom(location\_of\_service2)$

        grd004: $delaytime = delay\_start\_innormal\_delaytime(core)$

        grd005: $proc = delay\_start\_innormal\_proc(core)$

        grd006: $\exists x, y, b \cdot (((x \mapsto y) \mapsto b) = firstperiodicprocstart\_timeWindow\_of\_Partition(part) \Rightarrow$
$fstrl = ((clock\_tick * ONE\_TICK\_TIME)/majorFrame + 1) * majorFrame + x)$

        grd007: $processes\_of\_partition(proc) = part$

        grd008: $partition\_mode(part) = PM\_NORMAL$

grd009: $current\_partition = part$

grd017: $part \in dom(current\_partition\_flag)$

grd010: $current\_partition\_flag(part) = TRUE$

grd011: $current\_processes\_flag(core) = TRUE$

grd012: $process\_state(proc) = PS\_Waiting$

grd013: $period\_of\_process(proc) > 0$

grd014: $finished\_core2(core) = FALSE$

grd015: $location\_of\_service2(core) = Delay\_start\_period\_innormal \mapsto loc\_1$

grd016: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Delay\_start\_period\_innormal \mapsto loc\_1)$

then

act001: $location\_of\_service2(core) := Delay\_start\_period\_innormal \mapsto loc\_2$

act002: $deadlinetime\_of\_process(proc) := fstrl + delaytime + timecapacity\_of\_process(proc)$

end

Event delay_start_periodprocess_innormal_currentpri ⟨ordinary⟩ $\widehat{=}$

extends delay_start_periodprocess_innormal_currentpri

any

    part

    proc

    core

where

grd001: $part \in PARTITIONS$

grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition) \wedge proc \in dom(process\_state) \wedge proc \in dom(period\_of\_process)$

grd003: $core \in CORES \cap dom(delay\_start\_innormal\_proc) \wedge core \in dom(current\_processes\_flag) \wedge core \in dom(location\_of\_service2)$

grd004: $proc = delay\_start\_innormal\_proc(core)$

grd005: $processes\_of\_partition(proc) = part$

grd006: $part = current\_partition$

grd014: $part \in dom(current\_partition\_flag)$

grd007: $current\_partition\_flag(part) = TRUE$

grd008: $current\_processes\_flag(core) = TRUE$

grd009: $process\_state(proc) = PS\_Waiting$

grd010: $period\_of\_process(proc) > 0$

grd011: $finished\_core2(core) = FALSE$

grd012: $location\_of\_service2(core) = Delay\_start\_period\_innormal \mapsto loc\_2$

grd013: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Delay\_start\_period\_innormal \mapsto loc\_2)$

then

act001: $location\_of\_service2(core) := Delay\_start\_period\_innormal \mapsto loc\_3$

act002: $currentpriority\_of\_process(proc) := basepriority\_of\_process(proc)$

end

Event delay_start_periodprocess_innormal_return ⟨ordinary⟩ $\widehat{=}$

extends delay_start_periodprocess_innormal_return

any

    part

    proc

    core

where

grd001: $part \in PARTITIONS$

grd002: $proc \in processes \wedge proc \in dom(processes\_of\_partition) \wedge proc \in dom(process\_state) \wedge proc \in dom(period\_of\_process)$

grd003: $core \in CORES \cap dom(delay\_start\_innormal\_proc) \cap dom(delay\_start\_innormal\_delaytime) \wedge core \in dom(current\_processes\_flag) \wedge core \in dom(location\_of\_service2)$

grd004: $proc = delay\_start\_innormal\_proc(core)$

grd005: $processes\_of\_partition(proc) = part$

grd006: $current\_partition = part$

grd014: $part \in dom(current\_partition\_flag)$

grd007: $current\_partition\_flag(part) = TRUE$

grd008: $current\_processes\_flag(core) = TRUE$
grd009: $process\_state(proc) = PS\_Waiting$
grd010: $period\_of\_process(proc) > 0$
grd011: $finished\_core2(core) = FALSE$
grd012: $location\_of\_service2(core) = Delay\_start\_period\_innormal \mapsto loc\_3$
grd013: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Delay\_start\_period\_innormal \mapsto loc\_3)$

then

act001: $location\_of\_service2(core) := Delay\_start\_period\_innormal \mapsto loc\_r$
act002: $finished\_core2(core) := TRUE$
act003: $delay\_start\_innormal\_proc := \{core\} \lhd delay\_start\_innormal\_proc$
act004: $delay\_start\_innormal\_delaytime := \{core\} \lhd delay\_start\_innormal\_delaytime$

end

Event get_my_id ⟨ordinary⟩ $\widehat{=}$
extends get_my_id

any

    part
    proc
    core

where

grd001: $part \in PARTITIONS \cap dom(current\_partition\_flag)$
grd002: $core \in CORES \cap dom(current\_processes\_flag)$
grd007: $proc \in processes$
grd003: $current\_partition\_flag(part) = TRUE$
grd004: $current\_processes\_flag(core) = TRUE$
grd008: $proc = current\_processes(core)$
grd005: $current\_partition = part$
grd006: $part \in dom(errorhandler\_of\_partition) \Rightarrow proc \neq errorhandler\_of\_partition(part)$
grd009: $finished\_core(core) = TRUE$

then

    skip

end

Event initialize_process_core_affinity ⟨ordinary⟩ $\widehat{=}$
extends initialize_process_core_affinity

any

    part
    proc
    core

where

grd001: $part \in PARTITIONS$
grd002: $proc \in processes$
grd003: $core \in CORES$
grd004: $partition\_mode(part) = PM\_COLD\_START \vee partition\_mode(part) = PM\_WARM\_START$
grd005: $finished\_core(core) = TRUE$

then

    skip

end

Event get_my_processor_core_id ⟨ordinary⟩ $\widehat{=}$
extends get_my_processor_core_id

any

    part
    proc
    core

where

grd001: $part \in PARTITIONS$
grd002: $proc \in processes$
grd003: $core \in CORES \wedge core \in dom(current\_processes\_flag)$

        grd004: $partition\_mode(part) = PM\_NORMAL$

        grd005: $part = current\_partition \wedge current\_partition \in dom(current\_partition\_flag)$

        grd006: $current\_partition\_flag(part) = TRUE$

        grd007: $current\_processes\_flag(core) = TRUE$

        grd008: $proc = current\_processes(core)$

        grd009: $finished\_core(core) = TRUE$

then

        skip

end

Event process_faulted ⟨ordinary⟩ ≙

    new!! running –> faulted

extends process_faulted

    any

        part

        proc

        newstate

        core

    where

        grd001: $part \in PARTITIONS$

        grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state)$

        grd003: $newstate \in PROCESS\_STATES$

        grd004: $core \in CORES$

        grd005: $processes\_of\_partition(proc) = part$

        grd101: $partition\_mode(part) = PM\_NORMAL$

        grd102: $process\_state(proc) = PS\_Running \wedge newstate = PS\_Faulted$

        grd305: $part \in dom(current\_partition\_flag)$

        grd301: $part = current\_partition$

        grd304: $core \in dom(current\_processes)$

        grd307: $current\_processes\_flag(core) = TRUE$

        grd302: $proc = current\_processes(core)$

        grd303: $current\_partition\_flag(part) = TRUE$

        grd306: $current\_processes\_flag(core) = TRUE$

    then

        act001: $process\_state(proc) := newstate$

        act301: $need\_reschedule := TRUE$

        act302: $current\_processes\_flag(core) := FALSE$

        act303: $current\_processes := \{core\} \lhd current\_processes$

    end

Event time_wait_init ⟨ordinary⟩ ≙

extends time_wait_init

    any

        part

        proc

        newstate

        core

    where

        grd001: $part \in PARTITIONS \wedge part \in dom(locklevel\_of\_partition) \wedge part \in dom(current\_partition\_flag)$

        grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state) \cap dom(periodtype\_of\_process)$

        grd003: $newstate \in PROCESS\_STATES$

        grd004: $core \in CORES \wedge core \in dom(current\_processes)$

        grd005: $processes\_of\_partition(proc) = part$

        grd101: $partition\_mode(part) = PM\_NORMAL$

        grd102: $process\_state(proc) = PS\_Running \wedge (newstate = PS\_Ready \vee newstate = PS\_Waiting)$

        grd209: $proc \in dom(delaytime\_of\_process) \wedge proc \in dom(process\_wait\_type)$

        grd207: $current\_partition\_flag(part) = TRUE$

        grd206: $current\_processes\_flag(core) = TRUE$

        grd201: $proc = current\_processes(core)$

        grd202: $part = current\_partition$

        grd203: $part \in dom(errorhandler\_of\_partition) \Rightarrow proc \neq errorhandler\_of\_partition(part)$

        grd208: $periodtype\_of\_process(proc) = APERIOD\_PROC \lor periodtype\_of\_process(proc) = PERIOD\_PROC$

        grd204: $locklevel\_of\_partition(part) = 0$

        grd205: $finished\_core2(core) = TRUE$

then

        act001: $process\_state(proc) := newstate$

        act201: $location\_of\_service2(core) := Time\_Wait \mapsto loc\_i$

        act202: $finished\_core2(core) := FALSE$

        act203: $time\_wait\_proc(core) := proc$

        act204: $current\_processes\_flag(core) := FALSE$

        act205: $current\_processes := \{core\} \lhd current\_processes$

end

Event time_wait_delay_time ⟨ordinary⟩ ≙

extends time_wait_delay_time

    any

        part

        proc

        core

        delaytime

    where

        grd001: $part \in PARTITIONS$

        grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state)$

        grd003: $core \in CORES \cap dom(time\_wait\_proc) \land core \in dom(location\_of\_service2)$

        grd004: $processes\_of\_partition(proc) = part$

        grd005: $partition\_mode(part) = PM\_NORMAL$

        grd006: $proc = time\_wait\_proc(core)$

        grd012: $part \in dom(locklevel\_of\_partition)$

        grd007: $locklevel\_of\_partition(part) = 0$

        grd008: $delaytime \in \mathbb{N}_1$

        grd009: $finished\_core2(core) = FALSE$

        grd010: $location\_of\_service2(core) = Time\_Wait \mapsto loc\_i$

        grd011: $\neg(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Time\_Wait \mapsto loc\_i)$

    then

        act001: $location\_of\_service2(core) := Time\_Wait \mapsto loc\_1$

        act002: $timeout\_trigger := timeout\_trigger \lhd\!\!\!- \{proc \mapsto (PS\_Ready \mapsto (delaytime + clock\_tick * ONE\_TICK\_TIME))\}$

        act003: $process\_wait\_type(proc) := PROC\_WAIT\_TIMEOUT$

        act004: $delaytime\_of\_process(proc) := delaytime$

    end

Event time_wait_reschedule ⟨ordinary⟩ ≙

extends time_wait_reschedule

    any

        part

        proc

        core

    where

        grd001: $part \in PARTITIONS$

        grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state)$

        grd003: $core \in CORES \cap dom(time\_wait\_proc) \land core \in dom(location\_of\_service2)$

        grd004: $processes\_of\_partition(proc) = part$

        grd005: $partition\_mode(part) = PM\_NORMAL$

        grd006: $proc = time\_wait\_proc(core)$

        grd011: $part \in dom(locklevel\_of\_partition)$

        grd007: $locklevel\_of\_partition(part) = 0$

        grd008: $finished\_core2(core) = FALSE$

        grd009: $location\_of\_service2(core) = Time\_Wait \mapsto loc\_1$

        grd010: $\neg(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Time\_Wait \mapsto loc\_1)$

then

      act001: $location\_of\_service2(core) := Time\_Wait \mapsto loc\_2$

      act002: $need\_reschedule := TRUE$

end

Event time_wait_return ⟨ordinary⟩ $\widehat{=}$

extends time_wait_return

    any

        part

        proc

        core

    where

      grd001: $part \in PARTITIONS$

      grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state)$

      grd003: $core \in CORES \cap dom(time\_wait\_proc) \wedge core \in dom(location\_of\_service2)$

      grd004: $processes\_of\_partition(proc) = part$

      grd005: $partition\_mode(part) = PM\_NORMAL$

      grd006: $proc = time\_wait\_proc(core)$

      grd011: $part \in dom(locklevel\_of\_partition)$

      grd007: $locklevel\_of\_partition(part) = 0$

      grd008: $finished\_core2(core) = FALSE$

      grd009: $location\_of\_service2(core) = Time\_Wait \mapsto loc\_2$

      grd010: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Time\_Wait \mapsto loc\_2)$

    then

      act001: $location\_of\_service2(core) := Time\_Wait \mapsto loc\_r$

      act002: $time\_wait\_proc := \{core\} \lessdot time\_wait\_proc$

      act003: $finished\_core2(core) := TRUE$

    end

Event period_wait_init ⟨ordinary⟩ $\widehat{=}$

extends period_wait_init

    any

        part

        proc

        newstate

        core

    where

      grd001: $part \in PARTITIONS$

      grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state) \cap dom(period\_of\_process)$

      grd003: $newstate \in PROCESS\_STATES$

      grd004: $core \in CORES$

      grd005: $processes\_of\_partition(proc) = part$

      grd101: $partition\_mode(part) = PM\_NORMAL$

      grd102: $process\_state(proc) = PS\_Running \wedge newstate = PS\_Waiting$

      grd210: $proc \in dom(delaytime\_of\_process) \wedge proc \in dom(process\_wait\_type)$

      grd201: $current\_processes\_flag(core) = TRUE$

      grd209: $part \in dom(current\_partition\_flag) \wedge part \in dom(locklevel\_of\_partition)$

      grd202: $current\_partition\_flag(part) = TRUE$

      grd203: $part = current\_partition$

      grd204: $proc = current\_processes(core)$

      grd205: $part \in dom(errorhandler\_of\_partition) \Rightarrow proc \neq errorhandler\_of\_partition(part)$

      grd206: $locklevel\_of\_partition(part) = 0$

      grd207: $period\_of\_process(proc) > 0$

      grd208: $finished\_core2(core) = TRUE$

    then

      act001: $process\_state(proc) := newstate$

      act201: $location\_of\_service2(core) := Period\_Wait \mapsto loc\_i$

      act202: $finished\_core2(core) := FALSE$

      act203: $period\_wait\_proc(core) := proc$

      act204: $current\_processes\_flag(core) := FALSE$

    act205: $current\_processes := \{core\} \lhd current\_processes$
    end
Event period_wait_deadline_time $\langle$ordinary$\rangle$ $\widehat{=}$
extends period_wait_deadline_time
    any
        part
        proc
        core
    where
        grd001: $part \in PARTITIONS \wedge part \in dom(current\_partition\_flag) \wedge part \in dom(locklevel\_of\_partition)$

        grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state)$
        grd014: $proc \in dom(period\_of\_process)$
        grd003: $core \in CORES \wedge core \in dom(location\_of\_service2) \wedge core \in dom(period\_wait\_proc)$
        grd004: $processes\_of\_partition(proc) = part$
        grd005: $partition\_mode(part) = PM\_NORMAL$
        grd006: $current\_processes\_flag(core) = TRUE$
        grd007: $current\_partition\_flag(part) = TRUE$
        grd008: $proc = period\_wait\_proc(core)$
        grd009: $locklevel\_of\_partition(part) = 0$
        grd010: $period\_of\_process(proc) > 0$
        grd011: $finished\_core2(core) = FALSE$
        grd012: $location\_of\_service2(core) = Period\_Wait \mapsto loc\_i$
        grd013: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Period\_Wait \mapsto loc\_i)$
    then
        act001: $location\_of\_service2(core) := Period\_Wait \mapsto loc\_1$
        act002: $releasepoint\_of\_process(proc) := releasepoint\_of\_process(proc) + period\_of\_process(proc)$
        act003: $deadlinetime\_of\_process(proc) := releasepoint\_of\_process(proc) + timecapacity\_of\_process(proc)$

        act004: $process\_wait\_type(proc) := PROC\_WAIT\_PERIOD$
    end
Event period_wait_schedule $\langle$ordinary$\rangle$ $\widehat{=}$
extends period_wait_schedule
    any
        part
        proc
        core
    where
        grd001: $part \in PARTITIONS \wedge part \in dom(current\_partition\_flag) \wedge part \in dom(locklevel\_of\_partition)$

        grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state)$
        grd003: $core \in CORES \wedge core \in dom(location\_of\_service2) \wedge core \in dom(period\_wait\_proc)$
        grd004: $processes\_of\_partition(proc) = part$
        grd005: $partition\_mode(part) = PM\_NORMAL$
        grd006: $current\_processes\_flag(core) = TRUE$
        grd007: $current\_partition\_flag(part) = TRUE$
        grd008: $proc = period\_wait\_proc(core)$
        grd009: $locklevel\_of\_partition(part) = 0$
        grd010: $finished\_core2(core) = FALSE$
        grd011: $location\_of\_service2(core) = Period\_Wait \mapsto loc\_1$
        grd012: $\neg(finished\_core2(core) = FALSE \wedge location\_of\_service2(core) = Period\_Wait \mapsto loc\_1)$
    then
        act001: $location\_of\_service2(core) := Period\_Wait \mapsto loc\_2$
        act002: $need\_reschedule := TRUE$
    end
Event period_wait_return $\langle$ordinary$\rangle$ $\widehat{=}$
extends period_wait_return
    any

    part

    proc

    core

where

    grd001: $part \in PARTITIONS \land part \in dom(current\_partition\_flag)$

    grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state)$

    grd003: $core \in CORES \land core \in dom(location\_of\_service2)$

    grd004: $processes\_of\_partition(proc) = part$

    grd005: $partition\_mode(part) = PM\_NORMAL$

    grd006: $current\_processes\_flag(core) = TRUE$

    grd007: $current\_partition\_flag(part) = TRUE$

    grd008: $finished\_core2(core) = FALSE$

    grd009: $location\_of\_service2(core) = Period\_Wait \mapsto loc\_2$

    grd010: $\neg(finished\_core2(core) = FALSE \land location\_of\_service2(core) = Period\_Wait \mapsto loc\_2)$

then

    act001: $location\_of\_service2(core) := Period\_Wait \mapsto loc\_r$

    act002: $period\_wait\_proc := \{core\} \lhd period\_wait\_proc$

    act003: $finished\_core2(core) := TRUE$

end

Event get_time $\langle ordinary \rangle \;\widehat{=}$

extends get_time

    any

        part

        core

    where

        grd001: $part \in PARTITIONS \land part \in dom(current\_partition\_flag)$

        grd002: $core \in CORES \land core \in dom(current\_processes\_flag)$

        grd003: $part = current\_partition$

        grd004: $current\_processes\_flag(core) = TRUE \land current\_partition\_flag(part) = TRUE$

        grd005: $partition\_mode(part) = PM\_NORMAL$

    then

        skip

    end

Event replenish $\langle ordinary \rangle \;\widehat{=}$

extends replenish

    any

        part

        proc

        core

        budget_time

        ddtm

    where

        grd001: $part \in PARTITIONS \land part \in dom(current\_partition\_flag)$

        grd002: $core \in CORES \land core \in dom(current\_processes) \land core \in dom(current\_processes\_flag)$

        grd012: $proc \in processes \land proc \in dom(period\_of\_process) \land proc \in dom(releasepoint\_of\_process) \land$
             $proc \in dom(timecapacity\_of\_process)$

        grd003: $part = current\_partition$

        grd013: $current\_processes\_flag(core) = TRUE$

        grd004: $proc = current\_processes(core)$

        grd005: $current\_partition\_flag(part) = TRUE$

        grd006: $partition\_mode(part) = PM\_NORMAL$

        grd007: $budget\_time \in \mathbb{N}$

        grd008: $ddtm \in \mathbb{N}$

        grd009:

             $period\_of\_process(proc) > 0$
             $\land clock\_tick * ONE\_TICK\_TIME + budget\_time \leq releasepoint\_of\_process(proc) + timecapacity\_of\_process(proc)$

        grd010: $budget\_time > 0 \Rightarrow ddtm = clock\_tick * ONE\_TICK\_TIME + budget\_time$

    grd011: $(budget\_time = INFINITE\_TIME\_VALUE \lor timecapacity\_of\_process(proc) = INFINITE\_TIME\_VAl$
      $ddtm = INFINITE\_TIME\_VALUE$

  then

    act001: $deadlinetime\_of\_process(proc) := ddtm$

  end

Event aperiodicprocess_finished ⟨ordinary⟩ ≙

extends aperiodicprocess_finished

  any

    $part$
    $proc$
    $newstate$
    $core$

  where

    grd001: $part \in PARTITIONS$
    grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state)$
    grd003: $newstate \in PROCESS\_STATES$
    grd004: $core \in CORES$
    grd005: $processes\_of\_partition(proc) = part$
    grd101: $partition\_mode(part) = PM\_NORMAL$
    grd102: $process\_state(proc) = PS\_Running \land (newstate = PS\_Waiting \lor newstate = PS\_Dormant)$

    grd201: $proc \in dom(process\_wait\_type) \land proc \in dom(period\_of\_process)$
    grd307: $core \in dom(current\_processes\_flag)$
    grd308: $part \in dom(current\_partition\_flag)$
    grd301: $part = current\_partition$
    grd306: $current\_processes\_flag(core) = TRUE$
    grd302: $proc = current\_processes(core)$
    grd303: $current\_partition\_flag(part) = TRUE$
    grd304: $newstate = PS\_Dormant$
    grd305: $period\_of\_process(proc) = INFINITE\_TIME\_VALUE$

  then

    act001: $process\_state(proc) := newstate$
    act301: $need\_reschedule := TRUE$
    act302: $current\_processes\_flag(core) := FALSE$
    act303: $current\_processes := \{core\} \lhd current\_processes$

  end

Event periodicprocess_finished ⟨ordinary⟩ ≙

extends periodicprocess_finished

  any

    $part$
    $proc$
    $newstate$
    $core$

  where

    grd001: $part \in PARTITIONS$
    grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state)$
    grd003: $newstate \in PROCESS\_STATES$
    grd004: $core \in CORES$
    grd005: $processes\_of\_partition(proc) = part$
    grd101: $partition\_mode(part) = PM\_NORMAL$
    grd102: $process\_state(proc) = PS\_Running \land (newstate = PS\_Waiting \lor newstate = PS\_Dormant)$

    grd201: $proc \in dom(process\_wait\_type) \land proc \in dom(period\_of\_process)$
    grd307: $core \in dom(current\_processes\_flag)$
    grd308: $part \in dom(current\_partition\_flag)$
    grd301: $part = current\_partition$
    grd306: $current\_processes\_flag(core) = TRUE$
    grd302: $proc = current\_processes(core)$
    grd303: $current\_partition\_flag(part) = TRUE$

```
        grd304:   newstate = PS_Waiting
        grd305:   period_of_process(proc) ≠ INFINITE_TIME_VALUE
    then
        act001: process_state(proc) := newstate
        act301: need_reschedule := TRUE
        act302: process_wait_type(proc) := PROC_WAIT_PERIOD
        act303: current_processes_flag(core) := FALSE
        act304: current_processes := {core} ⩤ current_processes
    end
```

Event time_out ⟨ordinary⟩ ≙

extends time_out

    any

        part

        proc

        newstate

        core

        time

    where

        grd001: $part \in PARTITIONS$

        grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state)$

        grd003: $newstate \in PROCESS\_STATES$

        grd004: $core \in CORES$

        grd005: $processes\_of\_partition(proc) = part$

        grd101: $partition\_mode(part) = PM\_NORMAL$

        grd102: $process\_state(proc) = PS\_Waiting \lor process\_state(proc) = PS\_Suspend \lor process\_state(proc) = PS\_WaitandSuspend$

        grd103: $process\_state(proc) = PS\_Waiting \lor process\_state(proc) = PS\_Suspend \Rightarrow newstate = PS\_Ready$

        grd104: $process\_state(proc) = PS\_WaitandSuspend \Rightarrow newstate = PS\_Suspend$

        grd201: $time \in \mathbb{N}$

        grd202: $proc \in dom(timeout\_trigger)$

        grd203: $newstate \mapsto time = timeout\_trigger(proc)$

        grd204: $time \geq (clock\_tick - 1) * ONE\_TICK\_TIME \land time \leq clock\_tick * ONE\_TICK\_TIME$

        grd205: $process\_state(proc) = PS\_Waiting$

        grd301: $\neg(\exists r \cdot r \in queuing\_ports \land proc \in dom(processes\_waitingfor\_queuingports(r)))$

        grd302: $\neg(\exists r \cdot r \in buffers \land proc \in dom(processes\_waitingfor\_buffers(r)))$

        grd303: $\neg(\exists r \cdot r \in semaphores \land proc \in dom(processes\_waitingfor\_semaphores(r)))$

        grd304: $\neg(\exists r \cdot r \in blackboards \land proc \in processes\_waitingfor\_blackboards(r))$

        grd305: $\neg(\exists r \cdot r \in blackboards \land proc \in processes\_waitingfor\_blackboards(r))$

    then

        act001: $process\_state(proc) := newstate$

        act201: $timeout\_trigger := timeout\_trigger \setminus \{proc \mapsto (newstate \mapsto time)\}$

        act202: $process\_wait\_type := \{proc\} \lhd process\_wait\_type$

    end

Event time_out_wf_qport ⟨ordinary⟩ ≙

extends time_out_wf_qport

    any

        part

        proc

        newstate

        core

        time

        r

    where

        grd001: $part \in PARTITIONS$

        grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state)$

        grd003: $newstate \in PROCESS\_STATES$

        grd004: $core \in CORES$

        grd005: $processes\_of\_partition(proc) = part$

      grd101: $partition\_mode(part) = PM\_NORMAL$

      grd102: $process\_state(proc) = PS\_Waiting \lor process\_state(proc) = PS\_Suspend \lor process\_state(proc) = PS\_WaitandSuspend$

      grd103: $process\_state(proc) = PS\_Waiting \lor process\_state(proc) = PS\_Suspend \Rightarrow newstate = PS\_Ready$

      grd104: $process\_state(proc) = PS\_WaitandSuspend \Rightarrow newstate = PS\_Suspend$

      grd201: $time \in \mathbb{N}$

      grd202: $proc \in dom(timeout\_trigger)$

      grd203: $newstate \mapsto time = timeout\_trigger(proc)$

      grd204: $time \geq (clock\_tick - 1) * ONE\_TICK\_TIME \land time \leq clock\_tick * ONE\_TICK\_TIME$

      grd205: $process\_state(proc) = PS\_Waiting$

      grd301: $r \in queuing\_ports \land proc \in dom(processes\_waitingfor\_queuingports(r))$

then

      act001: $process\_state(proc) := newstate$

      act201: $timeout\_trigger := timeout\_trigger \setminus \{proc \mapsto (newstate \mapsto time)\}$

      act202: $process\_wait\_type := \{proc\} \lhd process\_wait\_type$

      act301: $processes\_waitingfor\_queuingports := (processes\_waitingfor\_queuingports \lhd \{r \mapsto \{proc\} \lhd processes\_waitingfor\_queuingports(r)\})$

end

Event time_out_wf_buf ⟨ordinary⟩ $\widehat{=}$

extends time_out_wf_buf

    any

      part

      proc

      newstate

      core

      time

      r

    where

      grd001: $part \in PARTITIONS$

      grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state)$

      grd003: $newstate \in PROCESS\_STATES$

      grd004: $core \in CORES$

      grd005: $processes\_of\_partition(proc) = part$

      grd101: $partition\_mode(part) = PM\_NORMAL$

      grd102: $process\_state(proc) = PS\_Waiting \lor process\_state(proc) = PS\_Suspend \lor process\_state(proc) = PS\_WaitandSuspend$

      grd103: $process\_state(proc) = PS\_Waiting \lor process\_state(proc) = PS\_Suspend \Rightarrow newstate = PS\_Ready$

      grd104: $process\_state(proc) = PS\_WaitandSuspend \Rightarrow newstate = PS\_Suspend$

      grd201: $time \in \mathbb{N}$

      grd202: $proc \in dom(timeout\_trigger)$

      grd203: $newstate \mapsto time = timeout\_trigger(proc)$

      grd204: $time \geq (clock\_tick - 1) * ONE\_TICK\_TIME \land time \leq clock\_tick * ONE\_TICK\_TIME$

      grd205: $process\_state(proc) = PS\_Waiting$

      grd301: $r \in buffers \land proc \in dom(processes\_waitingfor\_buffers(r))$

    then

      act001: $process\_state(proc) := newstate$

      act201: $timeout\_trigger := timeout\_trigger \setminus \{proc \mapsto (newstate \mapsto time)\}$

      act202: $process\_wait\_type := \{proc\} \lhd process\_wait\_type$

      act301: $processes\_waitingfor\_buffers := (processes\_waitingfor\_buffers \lhd \{r \mapsto \{proc\} \lhd processes\_waitingfor\_bu_\ldots$

    end

Event time_out_wf_sem ⟨ordinary⟩ $\widehat{=}$

extends time_out_wf_sem

    any

      part

      proc

      newstate

        core
        time
        r

where

    grd001: $part \in PARTITIONS$

    grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state)$

    grd003: $newstate \in PROCESS\_STATES$

    grd004: $core \in CORES$

    grd005: $processes\_of\_partition(proc) = part$

    grd101: $partition\_mode(part) = PM\_NORMAL$

    grd102: $process\_state(proc) = PS\_Waiting \lor process\_state(proc) = PS\_Suspend \lor process\_state(proc) = PS\_WaitandSuspend$

    grd103: $process\_state(proc) = PS\_Waiting \lor process\_state(proc) = PS\_Suspend \Rightarrow newstate = PS\_Ready$

    grd104: $process\_state(proc) = PS\_WaitandSuspend \Rightarrow newstate = PS\_Suspend$

    grd201: $time \in \mathbb{N}$

    grd202: $proc \in dom(timeout\_trigger)$

    grd203: $newstate \mapsto time = timeout\_trigger(proc)$

    grd204: $time \geq (clock\_tick - 1) * ONE\_TICK\_TIME \land time \leq clock\_tick * ONE\_TICK\_TIME$

    grd205: $process\_state(proc) = PS\_Waiting$

    grd301: $r \in semaphores \land proc \in dom(processes\_waitingfor\_semaphores(r))$

then

    act001: $process\_state(proc) := newstate$

    act201: $timeout\_trigger := timeout\_trigger \setminus \{proc \mapsto (newstate \mapsto time)\}$

    act202: $process\_wait\_type := \{proc\} \vartriangleleft process\_wait\_type$

    act301: $processes\_waitingfor\_semaphores := (processes\_waitingfor\_semaphores \vartriangleleft \{r \mapsto \{proc\} \vartriangleleft processes\_waitingfor\_semaphores(r)\})$

end

Event time_out_wf_bb ⟨ordinary⟩ $\hat{=}$

extends time_out_wf_bb

any

        part
        proc
        newstate
        core
        time
        r

where

    grd001: $part \in PARTITIONS$

    grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state)$

    grd003: $newstate \in PROCESS\_STATES$

    grd004: $core \in CORES$

    grd005: $processes\_of\_partition(proc) = part$

    grd101: $partition\_mode(part) = PM\_NORMAL$

    grd102: $process\_state(proc) = PS\_Waiting \lor process\_state(proc) = PS\_Suspend \lor process\_state(proc) = PS\_WaitandSuspend$

    grd103: $process\_state(proc) = PS\_Waiting \lor process\_state(proc) = PS\_Suspend \Rightarrow newstate = PS\_Ready$

    grd104: $process\_state(proc) = PS\_WaitandSuspend \Rightarrow newstate = PS\_Suspend$

    grd201: $time \in \mathbb{N}$

    grd202: $proc \in dom(timeout\_trigger)$

    grd203: $newstate \mapsto time = timeout\_trigger(proc)$

    grd204: $time \geq (clock\_tick - 1) * ONE\_TICK\_TIME \land time \leq clock\_tick * ONE\_TICK\_TIME$

    grd205: $process\_state(proc) = PS\_Waiting$

    grd301: $r \in blackboards \land proc \in processes\_waitingfor\_blackboards(r)$

then

    act001: $process\_state(proc) := newstate$

    act201: $timeout\_trigger := timeout\_trigger \setminus \{proc \mapsto (newstate \mapsto time)\}$

    act202: $process\_wait\_type := \{proc\} \vartriangleleft process\_wait\_type$

act301: $processes\_waiting for\_blackboards := processes\_waiting for\_blackboards \lhd\mkern-14mu- \{r \mapsto (processes\_waiting for\_blac$
$\{proc\})\}$

end

Event time_out_wf_evt ⟨ordinary⟩ $\widehat{=}$

extends time_out_wf_evt

    any

        part
        proc
        newstate
        core
        time
        r

    where

        grd001: $part \in PARTITIONS$
        grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state)$
        grd003: $newstate \in PROCESS\_STATES$
        grd004: $core \in CORES$
        grd005: $processes\_of\_partition(proc) = part$
        grd101: $partition\_mode(part) = PM\_NORMAL$
        grd102: $process\_state(proc) = PS\_Waiting \lor process\_state(proc) = PS\_Suspend \lor process\_state(proc) =$
           $PS\_WaitandSuspend$
        grd103: $process\_state(proc) = PS\_Waiting \lor process\_state(proc) = PS\_Suspend \Rightarrow newstate =$
           PS_Ready
        grd104: $process\_state(proc) = PS\_WaitandSuspend \Rightarrow newstate = PS\_Suspend$
        grd201: $time \in \mathbb{N}$
        grd202: $proc \in dom(timeout\_trigger)$
        grd203: $newstate \mapsto time = timeout\_trigger(proc)$
        grd204: $time \geq (clock\_tick - 1) * ONE\_TICK\_TIME \land time \leq clock\_tick * ONE\_TICK\_TIME$
        grd205: $process\_state(proc) = PS\_Waiting$
        grd301: $r \in events \land proc \in processes\_waiting for\_events(r)$

    then

        act001: $process\_state(proc) := newstate$
        act201: $timeout\_trigger := timeout\_trigger \setminus \{proc \mapsto (newstate \mapsto time)\}$
        act202: $process\_wait\_type := \{proc\} \lhd\mkern-14mu- process\_wait\_type$
        act301: $processes\_waiting for\_events := processes\_waiting for\_events \lhd\mkern-14mu- \{r \mapsto (processes\_waiting for\_events(r) \setminus$
           $\{proc\})\}$

    end

Event periodicproc_reach_releasepoint ⟨ordinary⟩ $\widehat{=}$

extends periodicproc_reach_releasepoint

    any

        part
        proc
        newstate
        core

    where

        grd001: $part \in PARTITIONS$
        grd002: $proc \in processes \cap dom(processes\_of\_partition) \cap dom(process\_state) \cap dom(periodtype\_of\_process)$
        grd003: $newstate \in PROCESS\_STATES$
        grd004: $core \in CORES$
        grd005: $processes\_of\_partition(proc) = part$
        grd101: $partition\_mode(part) = PM\_NORMAL$
        grd102: $periodtype\_of\_process(proc) = PERIOD\_PROC$
        grd103: $process\_state(proc) = PS\_Waiting$
        grd104: $newstate = PS\_Ready$
        grd204: $proc \in dom(period\_of\_process) \land proc \in dom(releasepoint\_of\_process) \land proc \in dom(process\_wait\_type)$
        grd205: $proc \in dom(timecapacity\_of\_process) \land proc \in dom(deadlinetime\_of\_process)$
        grd201: $period\_of\_process(proc) \neq INFINITE\_TIME\_VALUE$

        grd202: $clock\_tick * ONE\_TICK\_TIME \geq releasepoint\_of\_process(proc)$

        grd203: $process\_wait\_type(proc) = PROC\_WAIT\_PERIOD$

then

        act001: $process\_state(proc) := newstate$

        act201: $timeout\_trigger := \{proc\} \lhd timeout\_trigger$

        act202: $releasepoint\_of\_process(proc) := releasepoint\_of\_process(proc) + period\_of\_process(proc)$

        act203: $deadlinetime\_of\_process(proc) := releasepoint\_of\_process(proc) + timecapacity\_of\_process(proc)$

end

END