machine Mach_IPC

refines Mach_IPC_Conds **sees** Ctx_IPC

variables processes processes_of_partition partition_mode process_state periodtype_of_process
process_wait_type
locklevel_of_partition
startcondition_of_partition
basepriority_of_process
period_of_process
timecapacity_of_process
deadline_of_process
currentpriority_of_process
deadlinetime_of_process
releasepoint_of_process
delaytime_of_process
current_partition
current_process
current_partition_flag
current_process_flag
clock_tick

need_reschedule
need_procresch
preempter_of_partition
timeout_trigger
errorhandler_of_partition
process_call_errorhandler
queuing_ports
sampling_ports
RefreshPeriod_of_SamplingPorts
msgspace_of_samplingports
needtrans_of_sourcesamplingport
queue_of_queueingports quediscipline_of_queueingports
processes_waitingfor_queuingports
buffers blackboards semaphores events_ buffers_of_partition blackboards_of_partition
semaphores_of_partition events_of_partition MaxMsgNum_of_Buffers queue_of_buffers
processes_waitingfor_buffers quediscipline_of_buffers msgspace_of_blackboards emptyindicator_of_blackboards
processes_waitingfor_blackboards MaxValue_of_Semaphores value_of_semaphores quediscipline_of_semaphores
processes_waitingfor_semaphores state_of_events processes_waitingfor_events used_messages

invariants

@inv_refreshprd_of_sampports RefreshPeriod_of_SamplingPorts ∈ sampling_ports → ℕ1

@inv_flag_sourcesampport needtrans_of_sourcesamplingport ∈ sampling_ports → BOOL

@inv_flag_means_msg ∀*p*·(*p*∈sampling_ports ∧ needtrans_of_sourcesamplingport(*p*) = TRUE ⇒ *p*∈ dom(msgspace_of_samplingports))

@inv_quediscipline_of_queueingports quediscipline_of_queueingports∈queuing_ports→ **QUEUING_DISCIPLINE**

@inv_quediscipline_of_buffers quediscipline_of_buffers∈buffers→**QUEUING_DISCIPLINE**

@inv_quediscipline_of_semaphores quediscipline_of_semaphores∈semaphores→**QUEUING_DISCIPLINE**

events

event INITIALISATION **extends** INITIALISATION

then

@act400 RefreshPeriod_of_SamplingPorts ≔ ∅

@act401 needtrans_of_sourcesamplingport ≔ ∅

@act402 quediscipline_of_queueingports ≔ ∅

@act407 quediscipline_of_buffers ≔ ∅

@act408 quediscipline_of_semaphores ≔ ∅

end

event create_sampling_port **refines** create_sampling_port

any *port refresh*

where

@grd01 current_partition_flag = TRUE ∧ (partition_mode(current_partition)=**PM_COLD_START** ∨ partition_mode(current_partition)=**PM_WARM_START**)

@grd02 *port*∈**PORTS**∖(sampling_ports ∪ queuing_ports)

@grd03 *port*∈**SamplingPorts**

@grd04 **Ports_of_Partition**(*port*) = current_partition

@grd05 *refresh*∈ℕ1

@grd06 partition_mode(current_partition)≠**PM_NORMAL**

then

@act01 RefreshPeriod_of_SamplingPorts(*port*)≔*refresh*

@act02 sampling_ports ≔ sampling_ports∪{*port*}

@act03 needtrans_of_sourcesamplingport(*port*) ≔ FALSE

end

event write_sampling_message **refines** write_sampling_message

any *port msg t*

where

@grd01 *port*∈sampling_ports

@grd00 current_partition_flag = TRUE ∧ **Ports_of_Partition**(*port*) = current_partition

@grd02 *port*∈**SamplingPorts**

@grd03 **Direction_of_Ports**(*port*)=**PORT_SOURCE**

@grd04 *msg*∈**MESSAGES**∖used_messages

@grd05 t = clock_tick ∗ **ONE_TICK_TIME**

then

@act02 msgspace_of_samplingports($port$) ≔ $msg \mapsto t$

@act04 needtrans_of_sourcesamplingport($port$) ≔ TRUE

@act05 used_messages ≔ used_messages ∪ {msg}

end

event transfer_sampling_msg **refines** transfer_sampling_msg

any p m t

where

@grd02 $p \in$ sampling_ports

@grd03 $m \in$ **MESSAGES** ∧ $p \in$ dom(msgspace_of_samplingports) ∧ $m \mapsto t$=msgspace_of_samplingports(p)

@grd05 needtrans_of_sourcesamplingport(p) = TRUE

@grd06 **Sampling_Channels**~[{p}] ⊆ sampling_ports

@grd07 t = clock_tick ∗ **ONE_TICK_TIME**

then

@act01 needtrans_of_sourcesamplingport(p) ≔ FALSE

@act02 msgspace_of_samplingports ≔ msgspace_of_samplingports (**Sampling_Channels**~[{p}] × {$m \mapsto t$})

end

event read_sampling_message **refines** read_sampling_message
  any *port m t*
  where
    @grd01 *port*∈sampling_ports
    @grd00 current_partition_flag = TRUE ∧ **Ports_of_Partition**(*port*) = current_partition
    @grd03 **Direction_of_Ports**(*port*)=**PORT_DESTINATION**
    @grd04 *port*∈dom(msgspace_of_samplingports) ∧ (*m*↦*t*)=msgspace_of_samplingports(*port*)
    @grd05 *t*+RefreshPeriod_of_SamplingPorts(*port*)≥clock_tick ∗ **ONE_TICK_TIME**
end

event get_sampling_port_id
  any *port*
  where
    @grd01 *port*∈sampling_ports
    @grd00 current_partition_flag = TRUE ∧ **Ports_of_Partition**(*port*) = current_partition
end

event get_sampling_port_status
  any *port*
  where

@grd01 $port \in$ sampling_ports

@grd00 current_partition_flag = TRUE $\wedge$ **Ports_of_Partition**($port$) = current_partition

end

event create_queuing_port **refines** create_queuing_port

any *port discipline*

where

@grd02 $port \in$ **PORTS** $\setminus$ (sampling_ports $\cup$ queuing_ports)

@grd03 $port \in$ **QueuingPorts**

@grd01 current_partition_flag = TRUE $\wedge$ (partition_mode(current_partition)=**PM_COLD_START** $\vee$ partition_mode(current_partition)=**PM_WARM_START**)

@grd04 **Ports_of_Partition**($port$) = current_partition

@grd05 $discipline \in$ **QUEUING_DISCIPLINE**

@grd06 partition_mode(current_partition) $\neq$ **PM_NORMAL**

then

@act01 quediscipline_of_queueingports($port$) $\coloneqq$ $discipline$

@act02 queuing_ports $\coloneqq$ queuing_ports $\cup$ {$port$}

@act03 queue_of_queueingports($port$) $\coloneqq$ $\varnothing$

@act04 processes_waitingfor_queuingports($port$) $\coloneqq$ $\varnothing$

end

event send_queuing_message **refines** send_queuing_message

any *port msg t*

where

@grd01 *port*∈queuing_ports

@grd00 current_partition_flag = TRUE ∧ **Ports_of_Partition**(*port*) = current_partition

@grd03 **Direction_of_Ports**(*port*)=**PORT_SOURCE**

@grd04 *msg*∈**MESSAGES**∖used_messages

@grd05 card(queue_of_queueingports(*port*))<**MaxMsgNum_of_QueuingPorts**(*port*)

@grd06 processes_waitingfor_queuingports(*port*) = ∅

@grd07 *t* = clock_tick ∗ **ONE_TICK_TIME**

then

@act01 queue_of_queueingports(*port*) ≔ queue_of_queueingports(*port*) {*msg*↦*t*}

@act05 used_messages ≔ used_messages ∪ {*msg*}

end

event send_queuing_message_needwait

refines send_queuing_message_needwait

any *part proc newstate wt timeout tmout_trig port msg t*

where

@grd40 current_partition_flag = TRUE ∧ current_process_flag = TRUE

@grd41 *part* = current_partition ∧ *proc* = current_process ∧ *newstate* = **PS_Waiting**

@grd43 $wt \in$ **PROCESS_WAIT_TYPES** $\wedge$ ($wt =$ **PROC_WAIT_OBJ** $\vee$ $wt =$ **PROC_WAIT_TIMEOUT**)

@grd06 $timeout \in \mathbb{N}$

@grd07 current_process = $proc$

@grd45 $tmout\_trig \in$ processes $\pfun$ (**PROCESS_STATES** $\times \mathbb{N}1$)

@grd46 ($timeout =$ **INFINITE_TIME_VALUE** $\Rightarrow$ $tmout\_trig = \varnothing$)

$\wedge$ ($timeout > 0 \Rightarrow tmout\_trig = \{proc \mapsto ($**PS_Ready**$ \mapsto (timeout +$ clock_tick $*$ **ONE_TICK_TIME**$))\}$)

@grd47 $timeout > 0 \Rightarrow wt =$ **PROC_WAIT_TIMEOUT**

@grd48 $timeout =$ **INFINITE_TIME_VALUE** $\Rightarrow wt =$ **PROC_WAIT_OBJ**

@grd51 $port \in$ queuing_ports

@grd50 **Ports_of_Partition**($port$) = current_partition

@grd53 **Direction_of_Ports**($port$)=**PORT_SOURCE**

@grd54 $msg \in$ **MESSAGES** $\setminus$ used_messages

@grd55 card(queue_of_queueingports($port$))=**MaxMsgNum_of_QueuingPorts**($port$) $\vee$ processes_waitingfor_queuingports($port$) $\neq \varnothing$

@grd57 locklevel_of_partition(current_partition)=0 $\wedge$ (current_partition$\in$dom(errorhandler_of_partition) $\Rightarrow$ current_process$\neq$errorhandler_of_partition(current_partition))

@grd58 $t =$ clock_tick $*$ **ONE_TICK_TIME**

then

@act41 need_reschedule $\bcmeq$ TRUE

@act42 current_process_flag $\bcmeq$ FALSE

@act43 process_wait_type(current_process) $\bcmeq$ wt

@act45 timeout_trigger ≔ timeout_trigger *tmout_trig*

@act52 processes_waitingfor_queuingports(*port*) ≔ processes_waitingfor_queuingports(*port*) {*proc*↦(*msg*↦*t*)}

@act55 used_messages ≔ used_messages ∪ {*msg*}

@act56 process_state(*proc*) ≔ *newstate*

end

event transfer_queuing_msg **extends** transfer_queuing_msg

when

@grd20 (∀*m1*,*t1*·(*m1*↦*t1*∈queue_of_queueingports(p) ⇒ t≤*t1*))

end

event wakeup_waitproc_on_srcqueports

refines wakeup_waitproc_on_srcqueports

any *part proc newstate resch port msg t*

where

@grd500 current_partition_flag = TRUE ∧ partition_mode(current_partition)=**PM_NORMAL**

@grd509 *part* = current_partition

@grd02 *proc* ∈ processes

@grd03 *newstate* ∈ **PROCESS_STATES**

@grd06 processes_of_partition(*proc*) = *part*

@grd31 partition_mode(*part*) = **PM_NORMAL**

@grd32 process_state(*proc*) = **PS_Waiting** ∨ process_state(*proc*) = **PS_WaitandSuspend**

@grd33 process_state(*proc*) = **PS_Waiting** ⇒ *newstate* = **PS_Ready**

@grd34 process_state(*proc*) = **PS_WaitandSuspend** ⇒ *newstate* = **PS_Suspend**

@grd40 process_wait_type(*proc*)= **PROC_WAIT_OBJ**

@grd510 *resch* = TRUE

@grd502 *port* ∈ **Source_QueuingPorts** ∧ *port*∈queuing_ports

@grd504 card(queue_of_queueingports(*port*)) < **MaxMsgNum_of_QueuingPorts**(*port*)

@grd506 (*proc*↦(*msg* ↦ *t*))∈processes_waitingfor_queuingports(*port*)

@grd507 quediscipline_of_queueingports(*port*)=**QUEUE_FIFO** ⇒ (∀p1,t1,m·((p1↦(m↦t1))∈ processes_waitingfor_queuingports(*port*) ⇒ *t*≤t1))

@grd508 quediscipline_of_queueingports(*port*)=**QUEUE_PRIORITY** ⇒ (∀p1,t1,m·((p1↦(m↦t1))∈ processes_waitingfor_queuingports(*port*) ⇒ currentpriority_of_process(*proc*)≥currentpriority_of_process(p1)))

then

@act41 process_wait_type ≔ {*proc*}⩤process_wait_type

@act42 timeout_trigger ≔ {*proc*}⩤timeout_trigger

@act43 need_reschedule ≔ *resch*

@act501 processes_waitingfor_queuingports(*port*) ≔ {*proc*} ⩤ processes_waitingfor_queuingports(*port*)

@act506 queue_of_queueingports(*port*) ≔ queue_of_queueingports(*port*) {*msg*↦*t*}

@act11 process_state(*proc*) ≔ *newstate*

end

event wakeup_waitproc_on_destqueports
refines wakeup_waitproc_on_destqueports
  any *part proc newstate resch port msg t msg1 t1*
  where
    @grd500 current_partition_flag = TRUE ∧ partition_mode(current_partition)=**PM_NORMAL**
    @grd503 *part* = current_partition
    @grd02 *proc* ∈ processes
    @grd03 *newstate* ∈ **PROCESS_STATES**
    @grd06 processes_of_partition(*proc*) = *part*
    @grd31 partition_mode(*part*) = **PM_NORMAL**
    @grd32 process_state(*proc*) = **PS_Waiting** ∨ process_state(*proc*) = **PS_WaitandSuspend**
    @grd33 process_state(*proc*) = **PS_Waiting** ⇒ *newstate* = **PS_Ready**
    @grd34 process_state(*proc*) = **PS_WaitandSuspend** ⇒ *newstate* = **PS_Suspend**
    @grd40 process_wait_type(*proc*)= **PROC_WAIT_OBJ**
    @grd501 *resch* = TRUE
    @grd502 *port* ∈ **Dest_QueuingPorts** ∧ *port*∈queuing_ports
    @grd504 queue_of_queueingports(*port*)≠ ∅
    @grd506 (*proc*↦(*msg* ↦ *t*))∈processes_waitingfor_queuingports(*port*)
    @grd507 quediscipline_of_queueingports(*port*)=**QUEUE_FIFO** ⇒ (∀p1,tt,m·(p1↦(m↦tt)∈

processes_waitingfor_queuingports(*port*) ⇒ *t*≤tt))

@grd508 quediscipline_of_queueingports(*port*)=**QUEUE_PRIORITY** ⇒ (∀p1,tt,m·(p1↦(m↦tt)∈ processes_waitingfor_queuingports(*port*) ⇒ currentpriority_of_process(*proc*)≥currentpriority_of_process(p1)))

@grd509 *msg1*↦*t1*∈queue_of_queueingports(*port*)

@grd510 (∀*tt*,*mm*·(*mm*↦*tt*∈queue_of_queueingports(*port*) ⇒*t1*≤*tt*))

then

@act41 process_wait_type ≔ {*proc*}⩤process_wait_type

@act42 timeout_trigger ≔ {*proc*}⩤timeout_trigger

@act43 need_reschedule ≔ *resch*

@act501 processes_waitingfor_queuingports(*port*) ≔ {*proc*} ⩤ processes_waitingfor_queuingports(*port*)

@act506 queue_of_queueingports(*port*) ≔ queue_of_queueingports(*port*)∖{*msg* ↦ *t*}

@act11 process_state(*proc*) ≔ *newstate*

end

event receive_queuing_message **refines** receive_queuing_message

any *port msg t*

where

@grd01 *port*∈queuing_ports

@grd00 current_partition_flag = TRUE ∧ current_process_flag=TRUE ∧ **Ports_of_Partition**(*port*) = current_partition

@grd03 **Direction_of_Ports**(*port*)=**PORT_DESTINATION**

@grd04 *msg*∈**MESSAGES**

@grd06 queue_of_queueingports(*port*)≠∅

@grd05 (*msg* ↦*t*) ∈ queue_of_queueingports(*port*) ∧ (∀m,t1·(m↦t1∈queue_of_queueingports(*port*) ⇒ *t* ≤t1)) *// FIFO queue, read the first msg*

then

@act01 queue_of_queueingports(*port*) ≔ queue_of_queueingports(*port*) ∖ {*msg*↦*t*}

end

event receive_queuing_message_needwait

refines receive_queuing_message_needwait

any *part proc newstate port msg wt timeout tmout_trig t*

where

@grd40 current_partition_flag = TRUE ∧ current_process_flag = TRUE

@grd41 *part* = current_partition ∧ *proc* = current_process

@grd42 *newstate* = **PS_Waiting**

@grd43 *wt*∈**PROCESS_WAIT_TYPES** ∧ (*wt*= **PROC_WAIT_OBJ** ∨ *wt*=**PROC_WAIT_TIMEOUT**)

@grd44 *timeout* ∈ℕ

@grd45 *tmout_trig* ∈ processes ⇸ (**PROCESS_STATES** × ℕ1)

@grd46 (*timeout* = **INFINITE_TIME_VALUE** ⇒ *tmout_trig* = ∅)

∧ (*timeout* >0 ⇒ *tmout_trig* = {*proc*↦(**PS_Ready**↦ (*timeout* +clock_tick ∗ **ONE_TICK_TIME**))})

@grd47 *timeout* >0 ⇒ *wt* = **PROC_WAIT_TIMEOUT**

@grd48 *timeout* = **INFINITE_TIME_VALUE** ⇒ *wt* = **PROC_WAIT_OBJ**

@grd502 *port*∈queuing_ports

@grd503 *port*∈**QueuingPorts**

@grd500 **Ports_of_Partition**(*port*) = current_partition

@grd504 **Direction_of_Ports**(*port*)=**PORT_DESTINATION**

@grd505 queue_of_queueingports(*port*)=∅

@grd506 *msg*∈**MESSAGES**

@grd510 (*msg* ↦ *t*) ∈ queue_of_queueingports(*port*)

@grd507 locklevel_of_partition(current_partition)=0

@grd508 current_partition∈dom(errorhandler_of_partition) ⇒ current_process ≠ errorhandler_of_partition(current_partition)

then

@act41 need_reschedule ≔ TRUE

@act42 current_process_flag ≔ FALSE

@act43 process_wait_type(*proc*) ≔ *wt*

@act05 timeout_trigger ≔ timeout_trigger *tmout_trig*

@act52 processes_waitingfor_queuingports(*port*) ≔ processes_waitingfor_queuingports(*port*) {*proc*↦(*msg*↦*t*)}

@act11 process_state(*proc*) ≔ *newstate*

end

```
event get_queuing_port_id
  any port
  where
    @grd01 port∈QueuingPorts ∧port∈queuing_ports
    @grd00 current_partition_flag = TRUE ∧ Ports_of_Partition(port) = current_partition
end

event get_queuing_port_status
  any port
  where
    @grd01 port∈QueuingPorts ∧ port ∈queuing_ports
    @grd00 current_partition_flag = TRUE ∧ Ports_of_Partition(port) = current_partition
end

event clear_queuing_port refines clear_queuing_port
  any port
  where
    @grd01 port∈QueuingPorts ∧ port∈queuing_ports
    @grd00 current_partition_flag = TRUE ∧ Ports_of_Partition(port) = current_partition
    @grd02 Direction_of_Ports(port)=PORT_DESTINATION
  then
```

@act01 queue_of_queueingports(*port*) ≔ ∅

end

event create_buffer **refines** create_buffer

any *buf max_msg_size quediscip*

where

@grd00 current_partition_flag = TRUE ∧ (partition_mode(current_partition)=**PM_COLD_START** ∨ partition_mode(current_partition)=**PM_WARM_START**)

@grd01 *buf*∈**BUFFERS**∖buffers

@grd03 *max_msg_size*∈ℕ1

@grd04 *quediscip*∈**QUEUING_DISCIPLINE**

@grd06 partition_mode(current_partition)≠**PM_NORMAL**

then

@act01 MaxMsgNum_of_Buffers(*buf*)≔*max_msg_size*

@act02 buffers ≔ buffers∪{*buf*}

@act03 quediscipline_of_buffers(*buf*) ≔ *quediscip*

@act04 buffers_of_partition(*buf*) ≔ current_partition

@act05 queue_of_buffers(*buf*) ≔ ∅

@act06 processes_waitingfor_buffers(*buf*) ≔ ∅

end

```
event send_buffer refines send_buffer
  any buf msg t
  where
    @grd01 buf ∈ buffers
    @grd00 current_partition_flag = TRUE ∧ current_process_flag=TRUE ∧ buffers_of_partition(buf) =
current_partition
    @grd02 msg∈MESSAGES∖used_messages
    @grd05 card(queue_of_buffers(buf))<MaxMsgNum_of_Buffers(buf)
    @grd06 processes_waitingfor_buffers(buf) = ∅
    @grd07 t = clock_tick ∗ ONE_TICK_TIME
  then
 @act01 queue_of_buffers(buf) ≔ queue_of_buffers(buf)  {msg↦t}
    @act05 used_messages ≔ used_messages ∪ {msg}
end

event send_buffer_needwakeuprecvproc
refines send_buffer_needwakeuprecvproc
  any part proc newstate resch buf msg t m
  where
    @grd01 buf ∈ buffers
    @grd02 current_partition_flag = TRUE ∧ current_process_flag=TRUE ∧ buffers_of_partition(buf) =
```

current_partition

@grd03 *part* = current_partition

@grd04 *proc* ∈ processes

@grd05 process_state(*proc*) = **PS_Waiting** ∨ process_state(*proc*) = **PS_WaitandSuspend**

@grd06 process_state(*proc*) = **PS_Waiting** ⇒ *newstate* = **PS_Ready**

@grd07 process_state(*proc*) = **PS_WaitandSuspend** ⇒ *newstate* = **PS_Suspend**

@grd08 process_wait_type(*proc*)= **PROC_WAIT_OBJ**

@grd09 *resch* ∈BOOL

@grd10 (locklevel_of_partition(current_partition)=0 ⇒ *resch*=TRUE) ∧ (locklevel_of_partition(current_partition)>0 ⇒ *resch*=need_reschedule)

@grd11 *msg*∈**MESSAGES**∖used_messages

@grd12 *t*∈ℕ ∧ *m*∈**MESSAGES**

@grd13 card(queue_of_buffers(*buf*))<MaxMsgNum_of_Buffers(*buf*)

@grd14 processes_waitingfor_buffers(*buf*)≠∅ ∧ (*proc*↦(*m*↦**WAITING_R**↦*t*)) ∈ processes_waitingfor_buffers(*buf*)

@grd15 quediscipline_of_buffers(*buf*)=**QUEUE_FIFO** ⇒ (∀p1,m1,t1·(p1↦(m1↦**WAITING_R**↦t1)∈ processes_waitingfor_buffers(*buf*) ⇒ *t*≤t1))

@grd16 quediscipline_of_buffers(*buf*)=**QUEUE_PRIORITY** ⇒ (∀p1,m1,t1·(p1↦(m1↦**WAITING_R**↦t1)∈ processes_waitingfor_buffers(*buf*) ⇒ currentpriority_of_process(*proc*)≥currentpriority_of_process(p1)))

then

@act41 process_wait_type ≔ {*proc*}⩤process_wait_type

@act42 timeout_trigger ≔ {*proc*}⩤timeout_trigger
@act43 need_reschedule ≔ *resch*
@act501 used_messages ≔ used_messages ∪ {*msg*}
@act502 processes_waitingfor_buffers(*buf*) ≔ {*proc*} ⩤ processes_waitingfor_buffers(*buf*)
@act11 process_state(*proc*) ≔ *newstate*

end

event send_buffer_withfull
refines send_buffer_withfull
any *part proc newstate wt timeout tmout_trig buf msg t*
where
@grd40 current_partition_flag = TRUE ∧ current_process_flag = TRUE
@grd41 *part* = current_partition
@grd42 *proc* = current_process
@grd34 *newstate* = **PS_Waiting**
@grd43 *wt*∈**PROCESS_WAIT_TYPES** ∧ (*wt*= **PROC_WAIT_OBJ** ∨ *wt*=**PROC_WAIT_TIMEOUT**)
@grd44 *timeout* ∈ℕ
@grd45 *tmout_trig* ∈ processes ⇸ (**PROCESS_STATES** × ℕ1)
@grd46 (*timeout* = **INFINITE_TIME_VALUE** ⇒ *tmout_trig* = ∅)
∧ (*timeout* >0 ⇒ *tmout_trig* = {*proc*↦(**PS_Ready**↦ (*timeout* +clock_tick ∗ **ONE_TICK_TIME**))})
@grd47 *timeout*>0 ⇒ *wt* = **PROC_WAIT_TIMEOUT**

@grd48 *timeout* = **INFINITE_TIME_VALUE** ⇒ *wt* = **PROC_WAIT_OBJ**

@grd503 *buf* ∈ buffers

@grd500 buffers_of_partition(*buf*) = current_partition

@grd502 *msg*∈**MESSAGES**∖used_messages

@grd504 buffers_of_partition(*buf*) = current_partition

@grd505 card(queue_of_buffers(*buf*))=MaxMsgNum_of_Buffers(*buf*)

@grd509 locklevel_of_partition(current_partition) = 0

@grd510 current_partition∈dom(errorhandler_of_partition) ⇒ current_process ≠ errorhandler_of_partition(current_partition)

@grd511 *t* = clock_tick ∗ **ONE_TICK_TIME**

then

@act41 need_reschedule ≔ TRUE

@act42 current_process_flag ≔ FALSE

@act43 process_wait_type(*proc*) ≔ *wt*

@act05 timeout_trigger ≔ timeout_trigger *tmout_trig*

@act501 processes_waitingfor_buffers(*buf*) ≔ processes_waitingfor_buffers(*buf*) {*proc* ↦(*msg*↦ **WAITING_W**↦*t*)}

@act502 used_messages ≔ used_messages ∪ {*msg*}

@act11 process_state(*proc*) ≔ *newstate*

end

event receive_buffer **refines** receive_buffer

any *buf msg t*

where

@grd01 *buf* ∈ buffers

@grd00 current_partition_flag = TRUE ∧ current_process_flag=TRUE ∧ buffers_of_partition(*buf*) = current_partition

@grd02 *msg*∈**MESSAGES**

@grd03 queue_of_buffers(*buf*)≠ ∅

@grd04 *msg*↦*t*∈queue_of_buffers(*buf*) ∧ (∀m1,t1·(m1↦t1∈queue_of_buffers(*buf*)⇒*t*≤t1))

@grd05 processes_waitingfor_buffers(*buf*)=∅

then

@act01 queue_of_buffers(*buf*) ≔ queue_of_buffers(*buf*) ∖ {*msg*↦*t*}

end

event receive_buffer_needwakeupsendproc

refines receive_buffer_needwakeupsendproc

any *part proc newstate resch buf msg t m t_*

where

@grd500 current_partition_flag = TRUE ∧ current_process_flag=TRUE

@grd501 *part* = current_partition

@grd02 *proc* ∈ processes

@grd03 *newstate* ∈ **PROCESS_STATES**

@grd06 processes_of_partition(*proc*) = *part*

@grd31 partition_mode(*part*) = **PM_NORMAL**

@grd32 process_state(*proc*) = **PS_Waiting** ∨ process_state(*proc*) = **PS_WaitandSuspend**

@grd33 process_state(*proc*) = **PS_Waiting** ⇒ *newstate* = **PS_Ready**

@grd34 process_state(*proc*) = **PS_WaitandSuspend** ⇒ *newstate* = **PS_Suspend**

@grd40 process_wait_type(*proc*)= **PROC_WAIT_OBJ**

@grd41 *resch*∈BOOL

@grd509 (locklevel_of_partition(current_partition)=0 ⇒ *resch*=TRUE) ∧ (locklevel_of_partition(current_partition)>0 ⇒ *resch*=need_reschedule)

@grd506 *buf* ∈ buffers

@grd05 buffers_of_partition(*buf*) = current_partition

@grd502 *msg*∈**MESSAGES**

@grd508 *m*∈**MESSAGES** ∧ *t*∈ℕ ∧ *t_*∈ℕ

@grd503 queue_of_buffers(*buf*)≠∅

@grd504 *msg*↦*t*∈queue_of_buffers(*buf*) ∧ (∀m1,t1·(m1↦t1∈queue_of_buffers(*buf*)⇒*t*≤t1))

@grd505 processes_waitingfor_buffers(*buf*)≠∅ ∧ (*proc*↦(*m*↦**WAITING_W** ↦*t_*)) ∈ processes_waitingfor_buffers(*buf*)

@grd510 quediscipline_of_buffers(*buf*)=**QUEUE_FIFO** ⇒ (∀p1,m1,t1·(p1↦(m1↦**WAITING_R**↦t1)∈ processes_waitingfor_buffers(*buf*) ⇒ *t*≤t1))

@grd507 quediscipline_of_buffers(*buf*)=**QUEUE_PRIORITY** ⇒ (∀p1,m1,t1·(p1↦(m1↦**WAITING_R**↦t1)∈

processes_waitingfor_buffers(*buf*) ⇒ currentpriority_of_process(*proc*)≥currentpriority_of_process(p1)))

then

@act41 process_wait_type ≔ {*proc*}⩤process_wait_type

@act42 timeout_trigger ≔ {*proc*}⩤timeout_trigger

@act43 need_reschedule ≔ *resch*

@act501 queue_of_buffers(*buf*) ≔ (queue_of_buffers(*buf*) ∖ {*msg*↦*t*})

@act502 processes_waitingfor_buffers(*buf*) ≔ {*proc*} ⩤ processes_waitingfor_buffers(*buf*)

@act11 process_state(*proc*) ≔ *newstate*

end

event receive_buffer_whenempty

refines receive_buffer_whenempty

any *part proc newstate wt timeout tmout_trig buf msg t*

where

@grd40 current_partition_flag = TRUE ∧ current_process_flag = TRUE

@grd41 *part* = current_partition

@grd42 *proc* = current_process

@grd34 *newstate* = **PS_Waiting**

@grd43 *wt*∈**PROCESS_WAIT_TYPES** ∧ (*wt*= **PROC_WAIT_OBJ** ∨ *wt*=**PROC_WAIT_TIMEOUT**)

@grd44 *timeout* ∈ℕ

@grd45 *tmout_trig* ∈ processes ⇸ (**PROCESS_STATES** × ℕ1)

@grd46 (*timeout* = **INFINITE_TIME_VALUE** ⇒ *tmout_trig* = ∅)
∧ (*timeout* >0 ⇒ *tmout_trig* = {*proc*↦(**PS_Ready**↦ (*timeout* +clock_tick ∗ **ONE_TICK_TIME**))})
@grd47 *timeout* >0 ⇒ *wt* = **PROC_WAIT_TIMEOUT**
@grd48 *timeout* = **INFINITE_TIME_VALUE** ⇒ *wt* = **PROC_WAIT_OBJ**
@grd504 *buf* ∈ buffers
@grd500 buffers_of_partition(*buf*) = current_partition
@grd502 queue_of_buffers(*buf*)=∅
@grd503 *msg*∈**MESSAGES**
@grd509 locklevel_of_partition(current_partition) = 0
@grd510 current_partition∈dom(errorhandler_of_partition) ⇒ current_process ≠ errorhandler_of_partition(current_partition)
@grd511 *t* = clock_tick ∗ **ONE_TICK_TIME**

then

@act41 need_reschedule ≔ TRUE
@act42 current_process_flag ≔ FALSE
@act43 process_wait_type(*proc*) ≔ *wt*
@act05 timeout_trigger ≔ timeout_trigger *tmout_trig*
@act11 process_state(*proc*) ≔ *newstate*
@act501 processes_waitingfor_buffers(*buf*) ≔ processes_waitingfor_buffers(*buf*) {*proc* ↦(*msg*↦ **WAITING_R**↦*t*)}

end

```
event get_buffer_id
  any buf
  where
    @grd01 buf ∈ buffers
    @grd00 current_partition_flag = TRUE ∧ buffers_of_partition(buf) = current_partition
end

event get_buffer_status
  any buf
  where
    @grd01 buf ∈ buffers
    @grd00 current_partition_flag = TRUE ∧ buffers_of_partition(buf) = current_partition
end

event create_blackboard refines create_blackboard
  any bb
  where
    @grd00 current_partition_flag = TRUE ∧ (partition_mode(current_partition)=PM_COLD_START ∨
partition_mode(current_partition)=PM_WARM_START)
    @grd01 bb∈BLACKBOARDS∖blackboards
```

@grd06 partition_mode(current_partition)≠**PM_NORMAL**

then

@act02 blackboards ≔ blackboards∪{*bb*}

@act03 blackboards_of_partition(*bb*)≔ current_partition

@act04 emptyindicator_of_blackboards(*bb*)≔**BB_EMPTY**

@act05 processes_waitingfor_blackboards(*bb*) ≔ ∅

end

event display_blackboard **refines** display_blackboard

any *bb msg*

where

@grd00 current_partition_flag = TRUE ∧ current_process_flag=TRUE

@grd01 *bb*∈blackboards ∧ blackboards_of_partition(*bb*) = current_partition

@grd02 *msg*∈**MESSAGES**∖used_messages

@grd03 processes_waitingfor_blackboards(*bb*) = ∅

then

@act01 msgspace_of_blackboards(*bb*) ≔ *msg*

@act02 emptyindicator_of_blackboards(*bb*) ≔ **BB_OCCUPIED**

@act03 used_messages ≔ used_messages ∪ {*msg*}

end

event display_blackboard_needwakeuprdprocs

refines display_blackboard_needwakeuprdprocs

any *part procs newstates resch bb msg*

where

@grd01 current_partition_flag = TRUE ∧ current_process_flag=TRUE

@grd02 *part* = current_partition

@grd13 *bb*∈blackboards ∧ blackboards_of_partition(*bb*) = current_partition

@grd03 *procs*⊆**PROCESSES** ∧ *procs* ⊆ processes_of_partition~[{*part*}]

@grd04 *procs* = processes_waitingfor_blackboards(*bb*) ∧ *procs*⊆ dom(process_wait_type)

@grd05 ∀*proc*·(*proc*∈*procs* ⇒ process_wait_type(*proc*)= **PROC_WAIT_OBJ**)

@grd06 *newstates* ∈ *procs* → **PROCESS_STATES**

@grd07 partition_mode(*part*) = **PM_NORMAL**

@grd08 ∀*proc*·(*proc*∈*procs* ⇒ process_state(*proc*) = **PS_Waiting** ∨ process_state(*proc*) = **PS_WaitandSuspend**)

@grd09 ∀*proc*·(*proc*∈*procs* ∧ process_state(*proc*) = **PS_Waiting** ⇒ *newstates*(*proc*) = **PS_Ready**)

@grd10 ∀*proc*·(*proc*∈*procs* ∧ process_state(*proc*) = **PS_WaitandSuspend** ⇒ *newstates*(*proc*) = **PS_Suspend**)

@grd11 *resch*∈BOOL

@grd12 (locklevel_of_partition(current_partition)=0 ⇒ *resch*=TRUE) ∧ (locklevel_of_partition(current_partition)>0 ⇒ *resch*=need_reschedule)

@grd14 *msg*∈**MESSAGES**∖used_messages

@grd15 processes_waitingfor_blackboards(*bb*) ≠ ∅

then

@act41 process_wait_type ≔ *procs*⩤process_wait_type

@act42 timeout_trigger ≔ *procs*⩤timeout_trigger

@act43 need_reschedule ≔ *resch*

@act11 process_state ≔ process_state *newstates*

@act501 msgspace_of_blackboards(*bb*) ≔ *msg*

@act502 emptyindicator_of_blackboards(*bb*) ≔ **BB_OCCUPIED**

@act503 processes_waitingfor_blackboards(*bb*) ≔ processes_waitingfor_blackboards(*bb*)∖*procs*

@act504 used_messages ≔ used_messages ∪ {*msg*}

end

event read_blackboard **refines** read_blackboard

any *bb msg*

where

@grd00 current_partition_flag = TRUE ∧ current_process_flag=TRUE

@grd01 *bb*∈blackboards ∧ blackboards_of_partition(*bb*) = current_partition

@grd02 *msg*∈**MESSAGES**

@grd03 *bb*∈dom(msgspace_of_blackboards) ∧ *msg* = msgspace_of_blackboards(*bb*)

@grd04 emptyindicator_of_blackboards(*bb*) = **BB_OCCUPIED**

end

event read_blackboard_whenempty

refines read_blackboard_whenempty

any *part proc newstate wt timeout tmout_trig bb*

where

@grd40 current_partition_flag = TRUE ∧ current_process_flag = TRUE

@grd41 *part* = current_partition

@grd42 *proc* = current_process

@grd43 *wt*∈**PROCESS_WAIT_TYPES** ∧ (*wt*= **PROC_WAIT_OBJ** ∨ *wt*=**PROC_WAIT_TIMEOUT**)

@grd34 *newstate* = **PS_Waiting**

@grd44 *timeout* ∈ℕ

@grd45 *tmout_trig* ∈ processes ⇸ (**PROCESS_STATES** × ℕ1)

@grd46 (*timeout* = **INFINITE_TIME_VALUE** ⇒ *tmout_trig* = ∅)

∧ (*timeout* >0 ⇒ *tmout_trig* = {*proc*↦(**PS_Ready**↦ (*timeout* +clock_tick ∗ **ONE_TICK_TIME**))})

@grd47 *timeout* >0 ⇒ *wt* = **PROC_WAIT_TIMEOUT**

@grd48 *timeout* = **INFINITE_TIME_VALUE** ⇒ *wt* = **PROC_WAIT_OBJ**

@grd501 *bb*∈blackboards ∧ blackboards_of_partition(*bb*) = current_partition

@grd503 emptyindicator_of_blackboards(*bb*) = **BB_EMPTY**

@grd504 locklevel_of_partition(current_partition) = 0

@grd515 current_partition∈dom(errorhandler_of_partition) ⇒ current_process ≠ errorhandler_of_partition(current_partition)

then

@act41 need_reschedule ≔ TRUE

@act42 current_process_flag ≔ FALSE

@act43 process_wait_type(*proc*) ≔ *wt*

@act05 timeout_trigger ≔ timeout_trigger *tmout_trig*

@act501 processes_waitingfor_blackboards(*bb*) ≔ processes_waitingfor_blackboards(*bb*) ∪ {*proc*}

@act11 process_state(*proc*) ≔ *newstate*

end

event clear_blackboard **refines** clear_blackboard

any *bb*

where

@grd00 current_partition_flag = TRUE ∧ current_process_flag=TRUE

@grd01 *bb*∈blackboards ∧ blackboards_of_partition(*bb*) = current_partition

then

@act01 emptyindicator_of_blackboards(*bb*) ≔ **BB_EMPTY**

@act02 msgspace_of_blackboards ≔ {*bb*} ⩤ msgspace_of_blackboards

end

event get_blackboard_id

any *bb*

where

@grd01 $bb \in$ blackboards

@grd00 current_partition_flag = TRUE $\wedge$ blackboards_of_partition(bb) = current_partition

end

event get_blackboard_status

any *bb*

where

@grd01 $bb \in$ blackboards

@grd00 current_partition_flag = TRUE $\wedge$ blackboards_of_partition(bb) = current_partition

end

event create_semaphore **refines** create_semaphore

any *sem maxval currentval quediscip*

where

@grd01 current_partition_flag = TRUE $\wedge$ (partition_mode(current_partition)=**PM_COLD_START** $\vee$ partition_mode(current_partition)=**PM_WARM_START**)

@grd02 $sem \in$ **SEMAPHORES** $\setminus$ semaphores

@grd05 $quediscip \in$ **QUEUING_DISCIPLINE**

@grd06 partition_mode(current_partition) $\neq$ **PM_NORMAL**

@grd07 $maxval \in \mathbb{N}1$

@grd08 $currentval \in \mathbb{N} \land currentval \leq maxval$

then

@act01 quediscipline_of_semaphores(sem) $\coloneqq$ $quediscip$

@act02 semaphores $\coloneqq$ semaphores $\cup \{sem\}$

@act03 value_of_semaphores(sem) $\coloneqq$ $currentval$

@act04 MaxValue_of_Semaphores(sem) $\coloneqq$ $maxval$

@act05 semaphores_of_partition(sem) $\coloneqq$ current_partition

@act06 processes_waitingfor_semaphores(sem) $\coloneqq \varnothing$

end

event wait_semaphore **refines** wait_semaphore

any sem

where

@grd00 current_partition_flag = TRUE $\land$ current_process_flag=TRUE

@grd01 $sem \in$ semaphores $\land$ semaphores_of_partition(sem) = current_partition

@grd02 value_of_semaphores(sem) > 0

then

@act01 value_of_semaphores(sem) $\coloneqq$ value_of_semaphores(sem) -1

end

event wait_semahpore_whenzero

refines wait_semahpore_whenzero

any *part proc newstate wt timeout tmout_trig sem t*

where

@grd40 current_partition_flag = TRUE ∧ current_process_flag = TRUE

@grd41 *part* = current_partition

@grd42 *proc* = current_process

@grd34 *newstate* = **PS_Waiting**

@grd43 *wt*∈**PROCESS_WAIT_TYPES** ∧ (*wt*= **PROC_WAIT_OBJ** ∨ *wt*=**PROC_WAIT_TIMEOUT**)

@grd44 *timeout* ∈ℕ

@grd45 *tmout_trig* ∈ processes ⇸ (**PROCESS_STATES** × ℕ1)

@grd46 (*timeout* = **INFINITE_TIME_VALUE** ⇒ *tmout_trig* = ∅)

∧ (*timeout* > 0 ⇒ *tmout_trig* = {*proc*↦(**PS_Ready**↦ (*timeout* +clock_tick ∗ **ONE_TICK_TIME**))})

@grd47 *timeout* > 0 ⇒ *wt* = **PROC_WAIT_TIMEOUT**

@grd48 *timeout* = **INFINITE_TIME_VALUE** ⇒ *wt* = **PROC_WAIT_OBJ**

@grd502 *sem*∈semaphores ∧ semaphores_of_partition(*sem*) = current_partition

@grd504 value_of_semaphores(*sem*) = 0

@grd505 locklevel_of_partition(current_partition) = 0

@grd506 current_partition∈dom(errorhandler_of_partition) ⇒ current_process ≠ errorhandler_of_partition(current_partition)

@grd507 *t* = clock_tick ∗ **ONE_TICK_TIME**

then

@act41 need_reschedule ≔ TRUE
@act42 current_process_flag ≔ FALSE
@act43 process_wait_type(*proc*) ≔ *wt*
@act05 timeout_trigger ≔ timeout_trigger *tmout_trig*
@act501 processes_waitingfor_semaphores(*sem*) ≔ processes_waitingfor_semaphores(*sem*) {*proc* ↦ *t*}
@act11 process_state(*proc*) ≔ *newstate*

end

event signal_semaphore **refines** signal_semaphore
any *sem*
where
@grd00 current_partition_flag = TRUE ∧ current_process_flag=TRUE
@grd01 *sem*∈semaphores ∧ semaphores_of_partition(*sem*) = current_partition
@grd02 value_of_semaphores(*sem*) ≠ MaxValue_of_Semaphores(*sem*)
@grd03 processes_waitingfor_semaphores(*sem*) = ∅
then
@act01 value_of_semaphores(*sem*) ≔ value_of_semaphores(*sem*) + 1

end

event signal_semaphore_needwakeupproc
refines signal_semaphore_needwakeupproc

any *part proc newstate resch sem t*

where

@grd02 *proc* ∈ processes

@grd32 process_state(*proc*) = **PS_Waiting** ∨ process_state(*proc*) = **PS_WaitandSuspend**

@grd33 process_state(*proc*) = **PS_Waiting** ⇒ *newstate* = **PS_Ready**

@grd34 process_state(*proc*) = **PS_WaitandSuspend** ⇒ *newstate* = **PS_Suspend**

@grd35 *proc*∈ dom(process_wait_type)

@grd40 process_wait_type(*proc*)= **PROC_WAIT_OBJ**

@grd41 *resch*∈BOOL

@grd42 *t*∈ℕ

@grd500 current_partition_flag = TRUE ∧ current_process_flag=TRUE

@grd501 *part* = current_partition

@grd509 (locklevel_of_partition(current_partition)=0 ⇒ *resch*=TRUE) ∧ (locklevel_of_partition(current_partition)>0 ⇒ *resch*=need_reschedule)

@grd502 *sem*∈semaphores ∧ semaphores_of_partition(*sem*) = current_partition

@grd503 value_of_semaphores(*sem*) < MaxValue_of_Semaphores(*sem*)

@grd506 processes_waitingfor_semaphores(*sem*)≠∅ ∧ (*proc*↦*t*) ∈processes_waitingfor_semaphores(*sem*)

@grd507 quediscipline_of_semaphores(*sem*)=**QUEUE_FIFO** ⇒ (∀p1,t1·(p1↦t1∈ processes_waitingfor_semaphores(*sem*) ⇒ *t*≤t1))

@grd508 quediscipline_of_semaphores(*sem*)=**QUEUE_PRIORITY** ⇒ (∀p1,t1·(p1↦t1∈ processes_waitingfor_semaphores(*sem*) ⇒ currentpriority_of_process(*proc*)≥currentpriority_of_process(p1)))

```
  then
    @act41 process_wait_type ≔ {proc}⩤process_wait_type
    @act42 timeout_trigger ≔ {proc}⩤timeout_trigger
    @act43 need_reschedule ≔ resch
    @act501 processes_waitingfor_semaphores(sem) ≔ {proc} ⩤ processes_waitingfor_semaphores(sem)
    @act11 process_state(proc) ≔ newstate
end

event get_semaphore_id
  any sem
  where
    @grd01 sem∈semaphores
    @grd00 current_partition_flag = TRUE ∧ semaphores_of_partition(sem) = current_partition
end

event get_semaphore_status
  any sem
  where
    @grd01 sem∈semaphores
    @grd00 current_partition_flag = TRUE ∧ semaphores_of_partition(sem) = current_partition
end
```

event create_event **refines** create_event

any *ev*

where

@grd01 current_partition_flag = TRUE ∧ (partition_mode(current_partition)=**PM_COLD_START** ∨ partition_mode(current_partition)=**PM_WARM_START**)

@grd02 *ev*∈**EVENTS**∖events_

@grd06 partition_mode(current_partition)≠**PM_NORMAL**

then

@act01 events_ ≔ events_∪{*ev*}

@act02 state_of_events(*ev*) ≔ **EVENT_DOWN**

@act03 events_of_partition(*ev*) ≔ current_partition

@act04 processes_waitingfor_events(*ev*) ≔ ∅

end

event set_event **refines** set_event

any *ev*

where

@grd00 current_partition_flag = TRUE ∧ current_process_flag=TRUE

@grd01 *ev*∈events_ ∧ events_of_partition(*ev*) = current_partition

@grd03 processes_waitingfor_events(*ev*) = ∅

```
    then
      @act01 state_of_events(ev)≔EVENT_UP
  end

  event set_event_needwakeupprocs
  refines set_event_needwakeupprocs
    any part procs newstates resch ev
    where
      @grd500 current_partition_flag = TRUE ∧ current_process_flag=TRUE
      @grd501 part = current_partition
      @grd502 ev∈events_ ∧ events_of_partition(ev) = current_partition
      @grd02 procs ⊆ processes
      @grd06 procs ⊆ processes_of_partition~[{part}]
      @grd504 procs = processes_waitingfor_events(ev) ∧ procs⊆ dom(process_wait_type)
      @grd40 ∀proc·(proc∈procs ⇒ process_wait_type(proc)= PROC_WAIT_OBJ)
      @grd03 newstates ∈ procs → PROCESS_STATES
      @grd07 partition_mode(part) = PM_NORMAL
      @grd32 ∀proc·(proc∈procs ⇒ process_state(proc) = PS_Waiting ∨ process_state(proc) =
PS_WaitandSuspend)
      @grd33 ∀proc·(proc∈procs ∧ process_state(proc) = PS_Waiting ⇒ newstates(proc) = PS_Ready)
      @grd34 ∀proc·(proc∈procs ∧ process_state(proc) = PS_WaitandSuspend ⇒ newstates(proc) =
```

PS_Suspend)

@grd41 *resch*∈BOOL

@grd507 (locklevel_of_partition(current_partition)=0 ⇒ *resch*=TRUE) ∧ (locklevel_of_partition(current_partition)>0 ⇒ *resch*=need_reschedule)

@grd503 processes_waitingfor_events(*ev*) ≠ ∅

then

@act41 process_wait_type ≔ *procs*⩤process_wait_type

@act42 timeout_trigger ≔ *procs*⩤timeout_trigger

@act43 need_reschedule ≔ *resch*

@act11 process_state ≔ process_state *newstates*

@act501 state_of_events(*ev*)≔**EVENT_UP**

@act503 processes_waitingfor_events(*ev*) ≔ processes_waitingfor_events(*ev*)∖*procs*

end

event reset_event **refines** reset_event

any *ev*

where

@grd00 current_partition_flag = TRUE ∧ current_process_flag=TRUE

@grd01 *ev*∈events_ ∧ events_of_partition(*ev*) = current_partition

then

@act01 state_of_events(*ev*)≔**EVENT_DOWN**

end

event wait_event **refines** wait_event
  any *ev*
  where
    @grd00 current_partition_flag = TRUE ∧ current_process_flag=TRUE
    @grd01 *ev*∈events_ ∧ events_of_partition(*ev*) = current_partition
    @grd02 state_of_events(*ev*)=**EVENT_UP**
end

event wait_event_whendown
refines wait_event_whendown
  any *part proc newstate wt timeout tmout_trig ev*
  where
    @grd40 current_partition_flag = TRUE ∧ current_process_flag = TRUE
    @grd41 *part* = current_partition
    @grd42 *proc* = current_process
    @grd34 *newstate* = **PS_Waiting**
    @grd43 *wt*∈**PROCESS_WAIT_TYPES** ∧ (*wt*= **PROC_WAIT_OBJ** ∨ *wt*=**PROC_WAIT_TIMEOUT**)
    @grd44 *timeout* ∈ℕ
    @grd45 *tmout_trig* ∈ processes ⇸ (**PROCESS_STATES** × ℕ1)

@grd46 (*timeout* = **INFINITE_TIME_VALUE** ⇒ *tmout_trig* = ∅)
∧ (*timeout* >0 ⇒ *tmout_trig* = {*proc*↦(**PS_Ready**↦ (*timeout* +clock_tick ∗ **ONE_TICK_TIME**))})
@grd47 *timeout*>0 ⇒ *wt* = **PROC_WAIT_TIMEOUT**
@grd48 *timeout* = **INFINITE_TIME_VALUE** ⇒ *wt* = **PROC_WAIT_OBJ**
@grd503 *ev*∈events_ ∧ events_of_partition(*ev*) = current_partition
@grd504 state_of_events(*ev*)=**EVENT_DOWN**
@grd509 locklevel_of_partition(current_partition) = 0
@grd510 current_partition∈dom(errorhandler_of_partition) ⇒ current_process ≠ errorhandler_of_partition(current_partition)

then

@act41 need_reschedule ≔ TRUE
@act42 current_process_flag ≔ FALSE
@act43 process_wait_type(*proc*) ≔ *wt*
@act05 timeout_trigger ≔ timeout_trigger *tmout_trig*
@act501 processes_waitingfor_events(*ev*) ≔ processes_waitingfor_events(*ev*) ∪ {*proc*}
@act11 process_state(*proc*) ≔ *newstate*

end

event get_event_id

any *ev*

where

```
    @grd01 ev∈events_
    @grd00 current_partition_flag = TRUE ∧ events_of_partition(ev) = current_partition
end

event get_event_status
  any ev
  where
    @grd01 ev∈events_
    @grd00 current_partition_flag = TRUE ∧ events_of_partition(ev) = current_partition
end

event ticktock
extends ticktock
end

event partition_schedule extends partition_schedule
end

event process_schedule
extends process_schedule
end
```

```
  event run_errorhandler_preempter
  extends run_errorhandler_preempter
  end

  event get_partition_status extends get_partition_status
  end

  event set_partition_mode_to_idle
  extends set_partition_mode_to_idle
    then
      @act601 RefreshPeriod_of_SamplingPorts ≔ Ports_of_Partition~[{part}] ⩤
RefreshPeriod_of_SamplingPorts
      @act602 needtrans_of_sourcesamplingport ≔  Ports_of_Partition~[{part}] ⩤
needtrans_of_sourcesamplingport
      @act603 quediscipline_of_queueingports ≔ Ports_of_Partition~[{part}] ⩤ quediscipline_of_queueingports
      @act604 quediscipline_of_buffers ≔ buffers_of_partition~[{part}]⩤quediscipline_of_buffers
      @act605 quediscipline_of_semaphores ≔ semaphores_of_partition~[{part}] ⩤
quediscipline_of_semaphores
  end
```

```
  event set_partition_mode_to_normal extends set_partition_mode_to_normal
  end

  event set_partition_mode_to_coldstart extends set_partition_mode_to_coldstart
    then
      @act601 RefreshPeriod_of_SamplingPorts ≔ Ports_of_Partition~[{part}] ⩤
RefreshPeriod_of_SamplingPorts
      @act602 needtrans_of_sourcesamplingport ≔  Ports_of_Partition~[{part}] ⩤
needtrans_of_sourcesamplingport
      @act603 quediscipline_of_queueingports ≔ Ports_of_Partition~[{part}] ⩤ quediscipline_of_queueingports
      @act604 quediscipline_of_buffers ≔ buffers_of_partition~[{part}]⩤quediscipline_of_buffers
      @act605 quediscipline_of_semaphores ≔ semaphores_of_partition~[{part}] ⩤
quediscipline_of_semaphores
  end

  event set_partition_mode_to_warmstart extends set_partition_mode_to_warmstart
    then
      @act601 RefreshPeriod_of_SamplingPorts ≔ Ports_of_Partition~[{part}] ⩤
RefreshPeriod_of_SamplingPorts
      @act602 needtrans_of_sourcesamplingport ≔  Ports_of_Partition~[{part}] ⩤
needtrans_of_sourcesamplingport
```

```
        @act603 quediscipline_of_queueingports ≔ Ports_of_Partition~[{part}] ⩤ quediscipline_of_queueingports
        @act604 quediscipline_of_buffers ≔ buffers_of_partition~[{part}]⩤quediscipline_of_buffers
        @act605 quediscipline_of_semaphores ≔ semaphores_of_partition~[{part}] ⩤
quediscipline_of_semaphores
    end

    event get_process_id extends get_process_id
    end

    event get_process_status extends get_process_status
    end

    event create_process extends create_process
    end

    event set_priority extends set_priority
    end

    event suspend_self
    extends suspend_self
    end
```

```
event suspend
extends suspend
end

event resume
extends resume
end

event stop_self extends stop_self
end

event stop extends stop
end

event stop_wf_qport extends stop_wf_qport
end

event stop_wf_buf extends stop_wf_buf
end
```

```
event stop_wf_sem extends stop_wf_sem
end

event stop_wf_bb extends stop_wf_bb
end

event stop_wf_evt extends stop_wf_evt
end

event start_aperiodprocess_instart
extends start_aperiodprocess_instart
end

event start_aperiodprocess_innormal
extends start_aperiodprocess_innormal
end

event start_periodprocess_instart
extends start_periodprocess_instart
end
```

```
event start_periodprocess_innormal
extends start_periodprocess_innormal
end

event delaystart_aperiodprocess_instart
extends delaystart_aperiodprocess_instart
end

event delaystart_aperiodprocess_innormal
extends delaystart_aperiodprocess_innormal
end

event delaystart_periodprocess_instart
extends delaystart_periodprocess_instart
end

event delaystart_periodprocess_innormal
extends delaystart_periodprocess_innormal
end

event lock_preemption extends lock_preemption
```

```
end

event unlock_preemption extends unlock_preemption
end

event get_my_id extends get_my_id
end

event timed_wait extends timed_wait
end

event period_wait extends period_wait
end

event get_time extends get_time
end

event replenish extends replenish
end

event aperiodicprocess_finished extends aperiodicprocess_finished
```

```
end

event periodicprocess_finished extends periodicprocess_finished
end

event time_out // should refined to support remove process on waiting queue of comm resources
extends time_out
end


event time_out_wf_qport extends time_out_wf_qport
end

event time_out_wf_buf extends time_out_wf_buf
end

event time_out_wf_sem extends time_out_wf_sem
end

event time_out_wf_bb extends time_out_wf_bb
end
```

```
  event time_out_wf_evt extends time_out_wf_evt
  end

  event periodicproc_reach_releasepoint extends periodicproc_reach_releasepoint
  end

  event coldstart_partition_fromidle extends coldstart_partition_fromidle
  end

  event warmstart_partition_fromidle extends warmstart_partition_fromidle
  end
end
```