MODULE *cache*

EXTENDS TLC, $Integers$

CONSTANTS
$ResourceCap$,
$MaxConsumerCap$,
$Actors$

ASSUME $ResourceCap > 0$
ASSUME $MaxConsumerCap \in 1 \,..\, ResourceCap$

$Time \triangleq \{\text{"t1"}\}$
$Processes \triangleq Actors \cup Time$

VARIABLES
$resources\_left$,
$resources\_needed$,
$reserved$,
pc

$vars \triangleq \langle resources\_left, resources\_needed, pc, reserved \rangle$

$Init \triangleq$
$\land\ resources\_left = ResourceCap$
$\land\ reserved = 0$
$\land\ resources\_needed \in [Actors \to 1 \,..\, MaxConsumerCap]$
$\land\ pc \in [Processes \to \{\text{"init"}\}]$

$CheckConsume(actor) \triangleq$
Check if there are enought resources to consume. If so reserve them and set up as reday for consumption
$\land\ pc[actor] = \text{"init"}$
$\land\ resources\_left - reserved \geq resources\_needed[actor]$
$\land\ reserved' = reserved + resources\_needed[actor]$
$\land\ pc' = [pc \text{ EXCEPT } ![actor] = \text{"ready"}]$
$\land$ UNCHANGED $\langle resources\_left, resources\_needed \rangle$

$Consume(actor) \triangleq$
Given that there are enough reserved resources, consume them one at a time
$\land\ pc[actor] = \text{"ready"}$
$\land$ IF $resources\_needed[actor] > 0$
THEN $\land\ resources\_left' = resources\_left - 1$
$\land\ resources\_needed' = [resources\_needed \text{ EXCEPT } ![actor] = resources\_needed[actor] - 1]$
$\land\ reserved' = reserved - 1$

$\land$ UNCHANGED $\langle pc \rangle$

ELSE $\land \exists\, x \in 1 \,..\, MaxConsumerCap : resources\_needed' = [resources\_needed$ EXCEPT $![actor] =$

$\land\ pc' = [pc$ EXCEPT $![actor] =$ "init"$]$

$\land$ UNCHANGED $\langle resources\_left,\ reserved \rangle$

$Refill(time) \triangleq$

Refill resources at any time

$\land\ pc[time] =$ "init"

$\land\ resources\_left' = ResourceCap$

$\land$ UNCHANGED $\langle pc,\ resources\_needed,\ reserved \rangle$

$Next \triangleq$

$\lor \exists\, actor \in Actors : Consume(actor) \lor CheckConsume(actor)$

$\lor \exists\, timer \in Time : Refill(timer)$

$Spec \triangleq$

$\land\ Init$

$\land \Box[Next]_{vars}$

$\land \exists\, actor \in Actors : \mathrm{WF}_{vars}(Consume(actor))$

$\land \exists\, actor \in Actors : \mathrm{WF}_{vars}(CheckConsume(actor))$

$NoZeroResources \triangleq resources\_left \geq 0$

$EventuallyRefills \triangleq \exists\, n \in 1 \,..\, ResourceCap : (reserved = n) \leadsto (reserved > n)$
