



Facultad de Ciencias Exactas, Ingeniería y Agrimensura
UNIVERSIDAD NACIONAL DE ROSARIO

Ingeniería de Software

Trabajo Práctico

Verificación de Software

Cipullo, Inés

2025

1 Requerimientos

Se describen los requerimientos de un planificador de procesos a corto plazo, encargado de planificar los procesos que están listos para ejecución, también llamado *dispatcher*. Este planificador implementará el algoritmo de planificación “Ronda” (*Round Robin* en inglés), y este sistema tendrá un único procesador.

Un proceso puede estar en alguno de los siguientes estados: *nuevo*, *listo*, *ejecutando*, *bloqueado*, *terminado*. El dispatcher decide entre los procesos que están listos para ejecutarse y determina a cuál de ellos *activar*, y detiene a aquellos que *exceden su tiempo* de procesador, es decir, se encarga de las transiciones entre los estados **listo** y **ejecutando**.

Siguiendo Round Robin, los procesos listos se almacenan en forma de cola, cada proceso listo se ejecuta por un sólo *quantum* y si un proceso no ha terminado de ejecutarse al final de su período, será interrumpido y puesto al final de la cola de procesos listos. Se deben tener en cuenta, también, aquellas transiciones que involucran otros estados de procesos pero inciden sobre alguno de los dos estados que se controlan desde el dispatcher. Los procesos que sean agregados a la cola de listos por estas otras transiciones, se ubicarán al final de la misma.

2 Especificación

Para empezar, se dan las siguientes designaciones.

p es un proceso $\approx p \in PROCESS$

t es un contador de ticks del sistema $\approx t \in TICK$

proceso nulo $\approx nullp$

cantidad de tiempo durante el cual un proceso tiene permiso para ejecutarse en el procesador antes de ser interrumpido $\approx quantum$

cola de procesos listos para ser ejecutados $\approx procQueue$

proceso en ejecución $\approx current$

contador de ticks restantes de ejecución que le corresponden a $current \approx remTicks$

Luego, se introducen los tipos que se utilizan en la especificación.

$[PROCESS]$

$TICK ::= \mathbb{N}$

Además, se presentan las siguientes definiciones axiomáticas, donde se define la existencia del proceso $nullp$, que representa el proceso nulo, y la constante $quantum$, que equivale a 5 ticks.

$| \quad nullp : PROCESS$

$| \quad quantum : TICK$

$| \quad \hline quantum = 5$

Se define entonces el espacio de estados del planificador y su estado inicial.

$Dispatcher$

$procQueue : seq\langle PROCESS \rangle$

$current : PROCESS$

$remTicks : TICK$

<i>InitDispatcher</i>	
<i>Dispatcher</i>	
$procQueue = \langle \rangle$	
$current = nullp$	
$remTicks = 0$	

Como un proceso no puede estar *en ejecución* y *listo* al mismo tiempo, se plantea el invariate de que *current* no puede pertenecer a *procQueue*.

<i>InvDispatcher</i>	
<i>Dispatcher</i>	
$procQueue \triangleright \{current\} = \emptyset$	

Procedemos con la especificación de las operaciones requeridas. Estas son:

- *NewProcess*: para pasar un proceso de estado *nuevo* (o *bloqueado*) a *listo*.
- *Dispatch*: modela el funcionamiento del planificador, se encarga de las transiciones entre los estados *listo* y *en ejecución*.
- *TerminateProcess*: para pasar un proceso de estado *en ejecución* a *terminado*.

<i>NewProcessOk</i>	
$\delta Dispatcher$	
$p? : PROCESS$	
$procQueue \triangleright \{p?\} = \emptyset$	
$current \neq p?$	
$procQueue' = procQueue \frown \langle p? \rangle$	
$current' = current$	
$remTicks' = remTicks$	

NewProcessError

$\exists \text{Dispatcher?} : \text{PROCESS}$

$\text{procQueue} \triangleright \{p?\} \neq \emptyset \vee \text{current} = p?$

$\text{NewProcess} == \text{NewProcessOk} \vee \text{NewProcessError}$

Tick

$\Delta \text{Dispatcher}$

$\text{remTicks} > 0$

$\text{current} \neq \text{nullp}$

$\text{procQueue}' = \text{procQueue}$

$\text{current}' = \text{current}$

$\text{remTicks}' = \text{remTicks} - 1$

Timeout

$\Delta \text{Dispatcher}$

$\text{remTicks} = 0$

$\text{current} \neq \text{nullp}$

$\text{procQueue}' = \text{procQueue} \frown \langle \text{current} \rangle$

$\text{current}' = \text{nullp}$

$\text{remTicks}' = \text{remTicks}$

DispatchProcess

$\Delta \text{Dispatcher}$

$\text{current} = \text{nullp}$

$\text{procQueue} \neq \langle \rangle$

$\text{procQueue}' = \text{procQueue} \frown \langle \text{current} \rangle$

$\text{current}' = \text{nullp}$

$\text{remTicks}' = \text{remTicks}$

<i>Idle</i>	
$\Xi Dispatcher$	
$current = nullp$	
$procQueue = \langle \rangle$	

$$Dispatch == Tick \vee Timeout \vee DispatchProcess \vee Idle$$

<i>TerminateProcessOk</i>	
$\Delta Dispatcher$	
$p? : PROCESS$	
$current = p?$	
$procQueue' = procQueue$	
$current' = nullp$	
$remTicks' = remTicks$	

<i>TerminateProcessError</i>	
$\Xi Dispatcher$	
$p? : PROCESS$	
$current \neq p?$	

$$TerminateProcess == TerminateProcessOk \vee TerminateProcessError$$