

Sistemas operativos de tiempo real - 2025

## Práctica N°2

## **Delays y Software timers**

**Valentino Parma**

1)

La diferencia entre la versión de la práctica 1 y este es que al aplicar el vTaskDelay la tarea de mayor prioridad (en este caso la 2) pasa de estado running a estado blocked y mientras pasa eso se ejecuta la tarea 1 para volverse a blocked y así van alternándose

2)

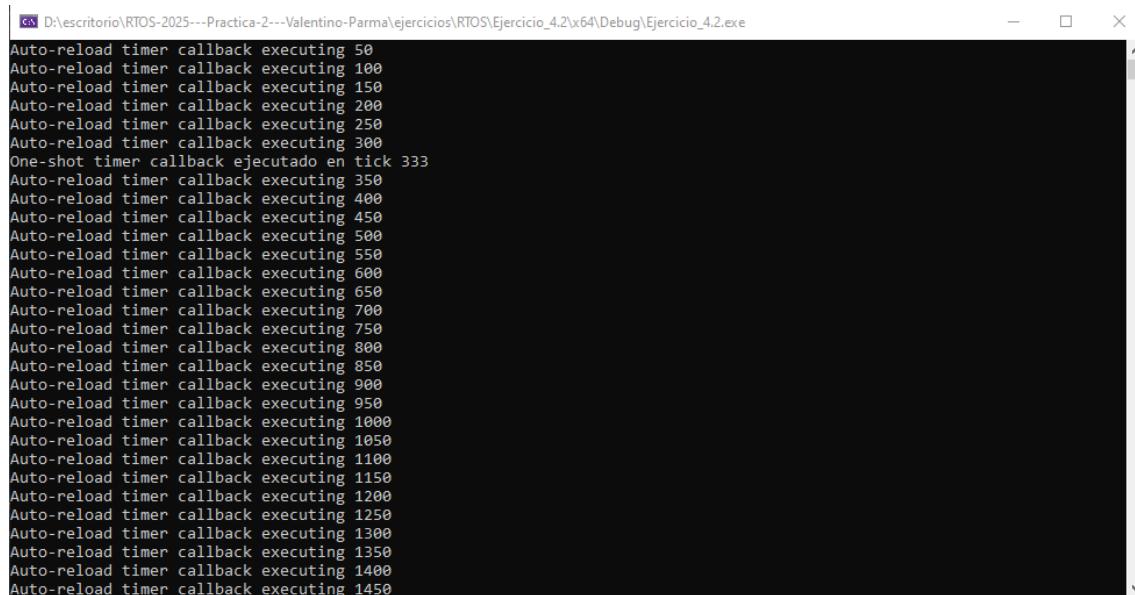
Al usar vTaskDelayUntil tiene una cadencia fija de demora lo que es mejor para tareas periodicas

3)

El sistema esta trabajando en modo time slice para las tareas de igual prioridad (1 y 2 ) porque se reparten el CPU en rondas la tarea periodica funciona en

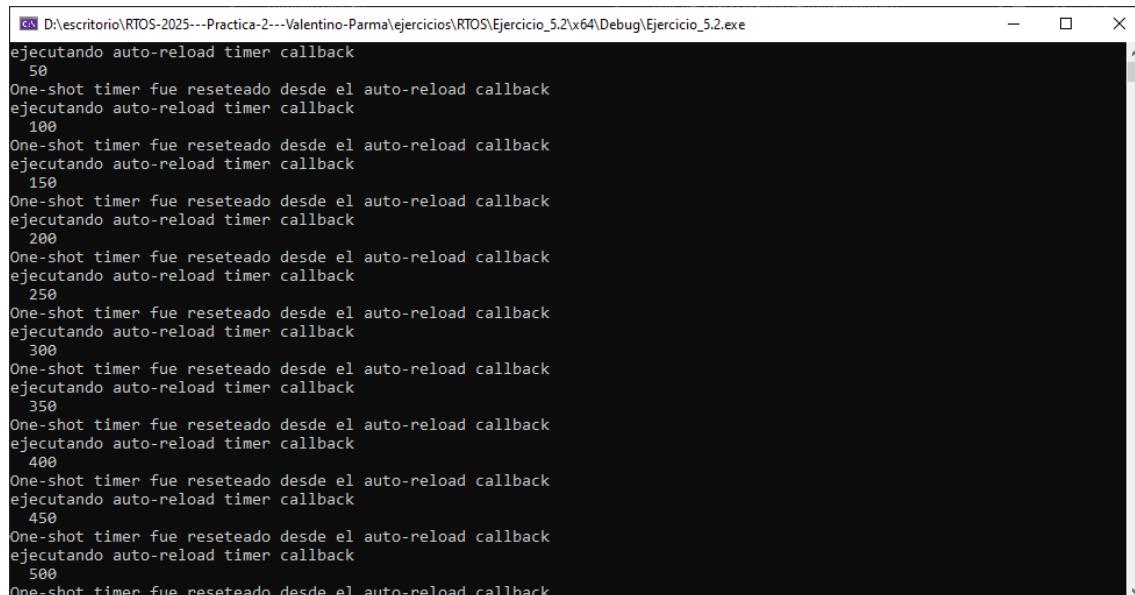
modo run to completion dentro de su periodo: cuando se despierta corre hasta que se bloquea y no se interrumpe por A/B porque tiene mayor prioridad

4)



```
D:\escritorio\RTOS-2025---Practica-2---Valentino-Parma\ejercicios\RTOS\Ejercicio_4.2\x64\Debug\Ejercicio_4.2.exe
Auto-reload timer callback executing 50
Auto-reload timer callback executing 100
Auto-reload timer callback executing 150
Auto-reload timer callback executing 200
Auto-reload timer callback executing 250
Auto-reload timer callback executing 300
One-shot timer callback ejecutado en tick 333
Auto-reload timer callback executing 350
Auto-reload timer callback executing 400
Auto-reload timer callback executing 450
Auto-reload timer callback executing 500
Auto-reload timer callback executing 550
Auto-reload timer callback executing 600
Auto-reload timer callback executing 650
Auto-reload timer callback executing 700
Auto-reload timer callback executing 750
Auto-reload timer callback executing 800
Auto-reload timer callback executing 850
Auto-reload timer callback executing 900
Auto-reload timer callback executing 950
Auto-reload timer callback executing 1000
Auto-reload timer callback executing 1050
Auto-reload timer callback executing 1100
Auto-reload timer callback executing 1150
Auto-reload timer callback executing 1200
Auto-reload timer callback executing 1250
Auto-reload timer callback executing 1300
Auto-reload timer callback executing 1350
Auto-reload timer callback executing 1400
Auto-reload timer callback executing 1450
```

5)



```
D:\escritorio\RTOS-2025---Practica-2---Valentino-Parma\ejercicios\RTOS\Ejercicio_5.2\x64\Debug\Ejercicio_5.2.exe
ejecutando auto-reload timer callback
 50
One-shot timer fue reseteado desde el auto-reload callback
ejecutando auto-reload timer callback
 100
One-shot timer fue reseteado desde el auto-reload callback
ejecutando auto-reload timer callback
 150
One-shot timer fue reseteado desde el auto-reload callback
ejecutando auto-reload timer callback
 200
One-shot timer fue reseteado desde el auto-reload callback
ejecutando auto-reload timer callback
 250
One-shot timer fue reseteado desde el auto-reload callback
ejecutando auto-reload timer callback
 300
One-shot timer fue reseteado desde el auto-reload callback
ejecutando auto-reload timer callback
 350
One-shot timer fue reseteado desde el auto-reload callback
ejecutando auto-reload timer callback
 400
One-shot timer fue reseteado desde el auto-reload callback
ejecutando auto-reload timer callback
 450
One-shot timer fue reseteado desde el auto-reload callback
ejecutando auto-reload timer callback
 500
One-shot timer fue reseteado desde el auto-reload callback
```

Al agregarse xTimerReset al timer auto reloaded hace que el timer one shot vuelva a comenzar su cuenta 333 ticks desde cero cada vez que el auto reload dispara por eso antes de que transcurra su periodo completo el auto reload lo reinicia

5)

D:\escritorio\RTOS-2025---Practica-2---Valentino-Parma\ejercicios\RTOS\Ejercicio\_6.2\x64\Debug\Ejercicio\_6.2.exe

```
100
Watchdog reseteado por heartbeat
Heartbeat enviado en tick
200
Watchdog reseteado por heartbeat
Heartbeat enviado en tick
300
Watchdog reseteado por heartbeat
Heartbeat enviado en tick
400
Watchdog reseteado por heartbeat
Heartbeat enviado en tick
500
Watchdog reseteado por heartbeat
Heartbeat enviado en tick
600
Watchdog reseteado por heartbeat
Heartbeat enviado en tick
700
Watchdog reseteado por heartbeat
Heartbeat enviado en tick
800
Watchdog reseteado por heartbeat
Heartbeat enviado en tick
900
Watchdog reseteado por heartbeat
Heartbeat enviado en tick
1000
Watchdog reseteado por heartbeat
```

Y en caso de que agregemos vTaskDelay(pdMS\_TO\_TICKS(10000)) (Esperar 10 segundos) y xTimerStop(xHeartbeatTimer, 0) (detener el heartbeat) el watchdog se activa

D:\escritorio\RTOS-2025---Practica-2---Valentino-Parma\ejercicios\RTOS\Ejercicio\_6.2\x64\Debug\Ejercicio\_6.2.exe

```
Heartbeat enviado en tick
100
Watchdog reseteado por heartbeat
WATCHDOG ACTIVADO en tick
1100
```

Si simulamos una falla en el heartbeat despues de 10 ciclos quedaria asi:

D:\escritorio\RTOS-2025---Practica-2---Valentino-Parma\ejercicios\RTOS\Ejercicio\_6.2\x64\Debug\Ejercicio\_6.2.exe

```
Heartbeat enviado en tick
100
Watchdog reseteado por heartbeat
Heartbeat enviado en tick
200
Watchdog reseteado por heartbeat
Heartbeat enviado en tick
300
Watchdog reseteado por heartbeat
Heartbeat enviado en tick
400
Watchdog reseteado por heartbeat
Heartbeat enviado en tick
500
Watchdog reseteado por heartbeat
Heartbeat enviado en tick
600
Watchdog reseteado por heartbeat
Heartbeat enviado en tick
700
Watchdog reseteado por heartbeat
Heartbeat enviado en tick
800
Watchdog reseteado por heartbeat
Heartbeat enviado en tick
900
Watchdog reseteado por heartbeat
Heartbeat enviado en tick
1000
Watchdog reseteado por heartbeat
Heartbeat enviado en tick
1100
FALLA simulada: no se envia senial de vida
Heartbeat enviado en tick
1200
FALLA simulada: no se envia senial de vida
```