

# Sistemas Operativos

Departamento de Computación  
Facultad de Ciencias Exactas y Naturales  
Universidad de Buenos Aires

## Trabajo Práctico 1 - Cachalotes

Integrante	LU	Correo electrónico
Ortiz de Zarate, Juan Manuel	403/10	jmanuoz@gmail.com
Kujawski, Kevin	459/10	kevinkuja@gmail.com
Carreiro, Martin	45/10	carreiromartin@gmail.com

Reservado para la cátedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

# Índice

1. Ejercicio 2	3
2. Ejercicio 4	4
3. Ejercicio 5	5
4. Ejercicio 7	6
5. Ejercicio 8	8

## 1. Ejercicio 2

## 2. Ejercicio 4

### 3. Ejercicio 5

Se indican las simplificaciones tomadas:

Primero queremos aclarar que se decidió que a cada tarea se le asigna 1 ticket y este número irá incrementándose a medida que gane compensaciones por bloqueo.

Decidimos omitir el hecho de que una tarea pueda subdividirse en threads, junto con el manejo de tickets del usuario. Con esto queremos decir que al no existir usuarios (solo uno) no existe la posibilidad del cambio de cantidad de tickets bajo un mismo usuario.

# 4. Ejercicio 7

Lote simulado con Round Robin:

- TaskBatch 10 1
- TaskBatch 20 2
- TaskBatch 13 3
- TaskBatch 17 3
- TaskBatch 27 2
- TaskBatch 11 1

Diagrama con 2 ciclos de quantum:

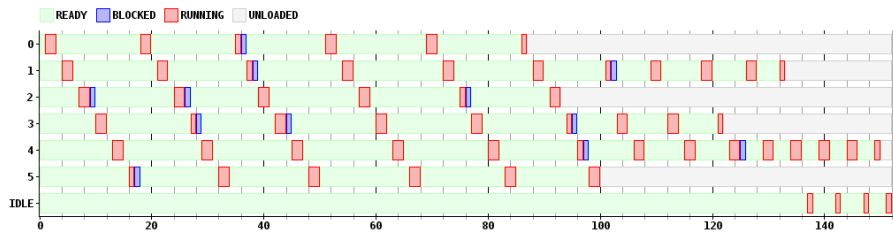


Diagrama con 5 ciclos de quantum:

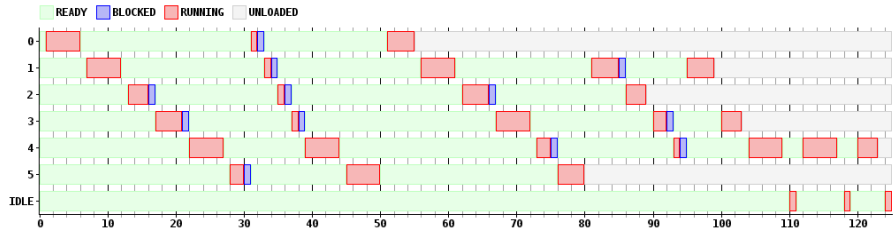


Diagrama con 7 ciclos de quantum:

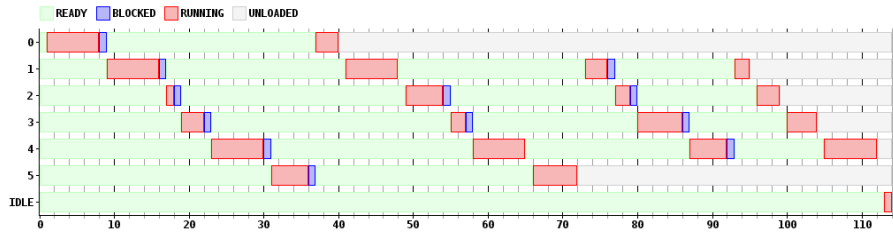


Diagrama con 9 ciclos de quantum:

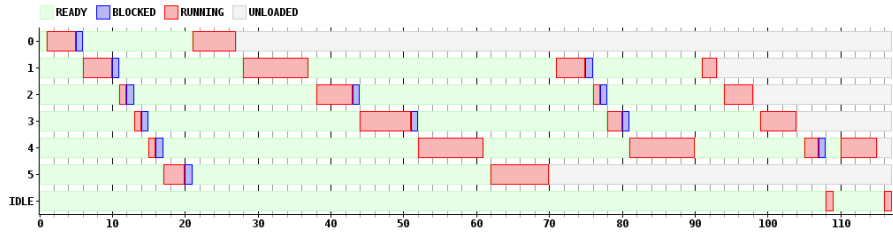


Diagrama con 12 ciclos de quantum:

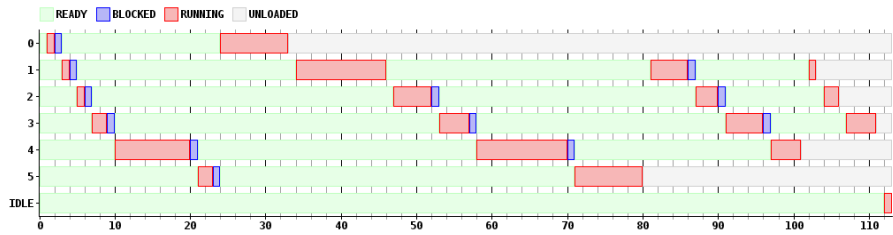


Diagrama con 17 ciclos de quantum:

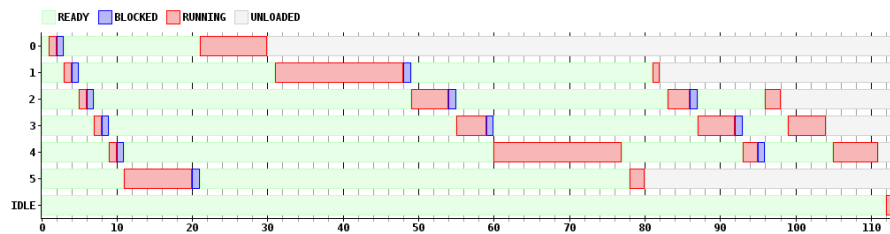
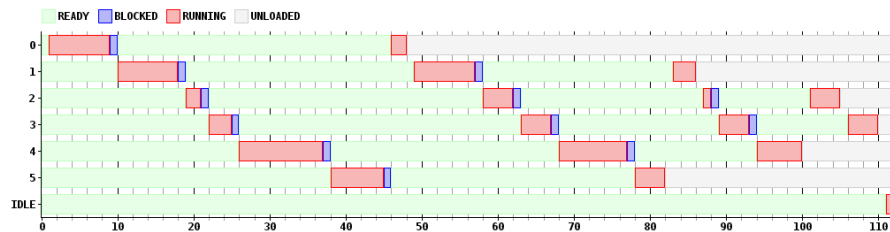


Diagrama con 21 ciclos de quantum:



Se puede observar que a medida que los ciclos del quantum aumentan, las tareas van finalizando más rápido, debido a que se le da mas tiempo de ejecución, hay una menor cambio de contexto entre los procesos y ocurren menos bloqueos por intervalo de ejecución del proceso.

Para este lote de tareas en general, suele ser mas eficiente asignarle una cantidad grande de ciclos por quantum, ya que se realizarán menos bloqueos y cada proceso podrá finalizar en la menor cantidad de "cortes" de ejecución, pero no excesiva, porque, como se aprecia en las imágenes, entre 12 y 21 ciclos por quantum todos tardan lo mismo en finalizar y no mejora la eficiencia.

Con un quantum de 21 ciclos, se puede notar que lo que mas influye en cada proceso son los bloqueos que se producen y no tanto el corte porque no le quedan mas ciclos, produciendo cambios de contexto obligatorios y empeorando bastante la optimización del Round Robin frente a otros algoritmos de scheduling.

## 5. Ejercicio 8

Presentamos el siguiente lote corrido con distintos semillas de aleatoridad y tamaño de quantum.

TaskBatch 10 1

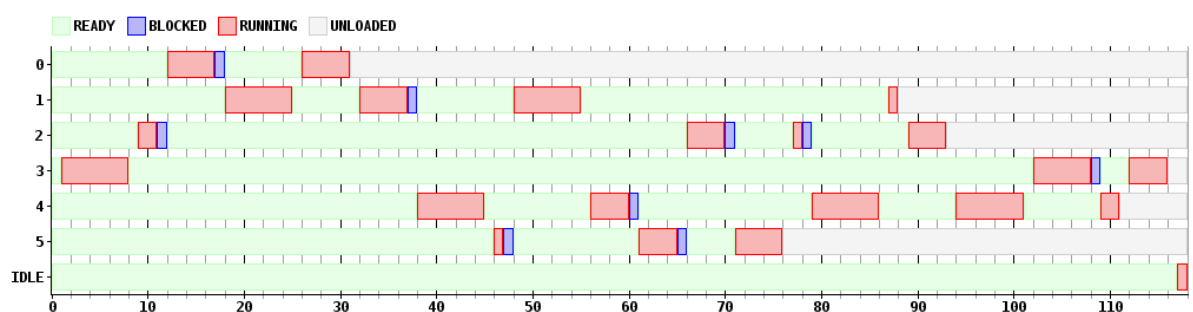
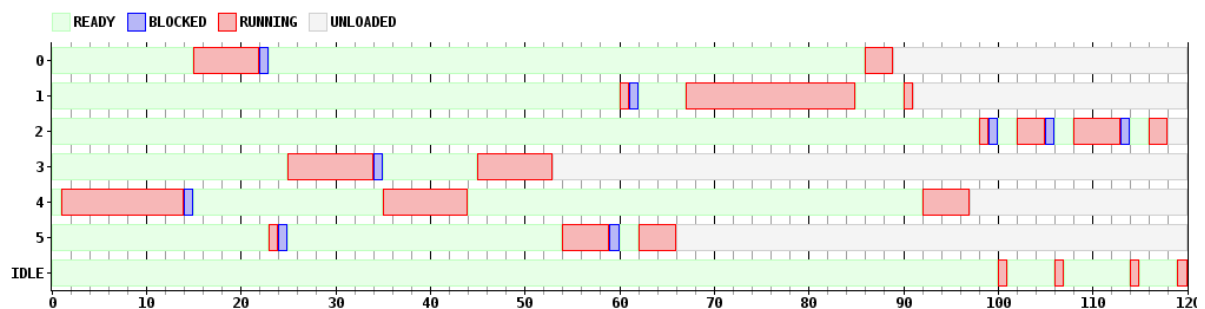
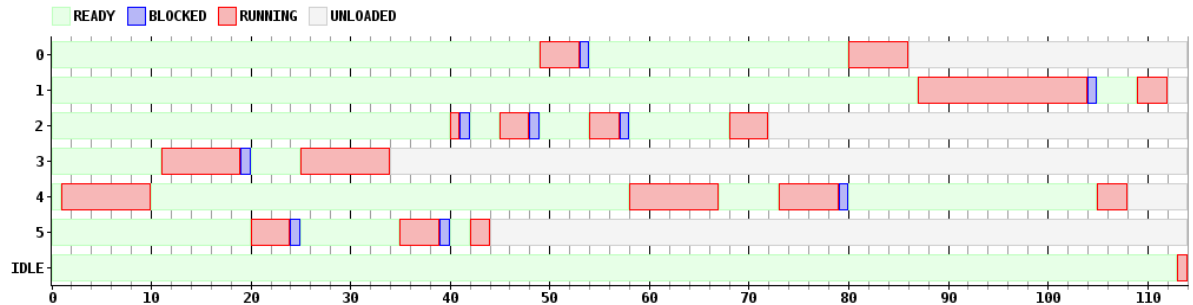
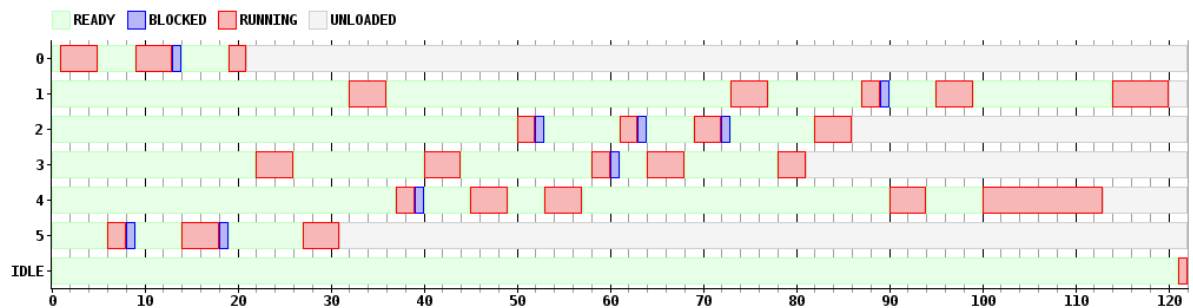
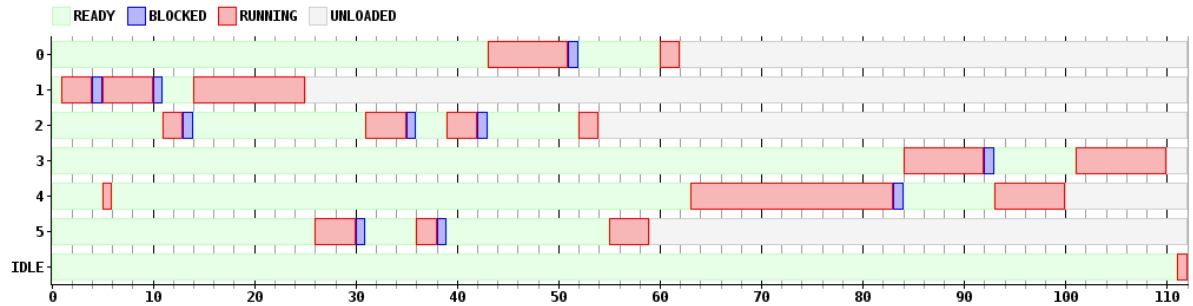
TaskBatch 20 1

TaskBatch 13 3

TaskBatch 17 1

TaskBatch 27 1

TaskBatch 11 2





Se puede notar que aunque se elija el mismo lote, la aleatoriedad del algoritmo muestra como aunque se asignen de distinta forma los recursos, todos los procesos tienden a obtenerlos uniformemente. Utilizamos el mismo ejemplo para notar la "compensación". Notese como en el A el proceso 1 después de haber bloqueado 2 veces obtuvo tantos tickets que casi tuvo exclusividad del procesador. Por el contrario el proceso 4 tuvo que esperar varios sorteos hasta salir ganador.