



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico I

Scheduling

Sistemas Operativos
Primer Cuatrimestre 2015

Integrante	LU	Correo electrónico
Aldasoro Agustina	86/13	agusaldasoro@gmail.com
More Ángel	XXX/XX	mail
Zimenspitz Ezequiel	155/13	ezeqzim@gmail.com



Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2160 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (54 11) 4576-3359

<http://www.fcen.uba.ar>

Resumen

En el presente trabajo se describe la problemática de ...

Índice

1. Parte I	3
1.1. Ejercicio 1: TaskConsola	3
1.2. Ejercicio 2: Ejecución de tres tareas	3
2. Parte II	3
2.1. Ejercicio 3: Scheduler Round-Robin	3
2.2. Ejercicio 4: Ejecución de lotes de tareas	3
2.3. Ejercicio 5: Scheduling algorithms for multiprogramming in a hard-real-time environment	3
3. Parte III	3
3.1. Ejercicio 6: TaskBatch	3
3.2. Ejercicio 7: Ejecución lote de tareas	4
3.3. Ejercicio 8: Scheduler Round-Robin modificado	4
3.4. Ejercicio 9: Ejecución lote de tareas	4
3.5. Ejercicio 10: Ejecución lote de tareas	4

1. Parte I

1.1. Ejercicio 1: TaskConsola

Programar un tipo de tarea TaskConsola, que simulará una tarea interactiva.

La tarea debe realizar n llamadas bloqueantes, cada una de una duración al azar¹ entre $bmin$ y $bmax$ (inclusive). La tarea debe recibir tres parámetros: n , $bmin$ y $bmax$ (en ese orden) que serán interpretados como los tres elementos del vector de enteros que recibe la función.

1.2. Ejercicio 2: Ejecución de tres tareas

Ejercicio 2 Escribir un lote de 3 tareas distintas: una intensiva en CPU y las otras dos de tipo interactivo (TaskConsola). Ejecutar y graficar la simulación usando el algoritmo FCFS para 1, 2 y 3 núcleos.

2. Parte II

2.1. Ejercicio 3: Scheduler Round-Robin

Completar la implementación del scheduler Round-Robin implementando los métodos de la clase SchedRR en los archivos sched_rr.cpp y sched_rr.h. La implementación recibe como primer parámetro la cantidad de núcleos y a continuación los valores de sus respectivos quantums. Debe utilizar una única cola global, permitiendo así la migración de procesos entre núcleos.

2.2. Ejercicio 4: Ejecución de lotes de tareas

Diseñar uno o más lotes de tareas para ejecutar con el algoritmo del ejercicio anterior. Graficar las simulaciones y comentarlas, justificando brevemente por qué el comportamiento observado es efectivamente el esperable de un algoritmo Round-Robin.

2.3. Ejercicio 5: Scheduling algorithms for multiprogramming in a hard-real-time environment

A partir del artículo Liu, Chung Laung, and James W. Layland. Scheduling algorithms for multiprogramming in a hard-real-time environment. Journal of the ACM (JACM) 20.1 (1973): 46-61.

1. Responda:

- ¿Qué problema están intentando resolver los autores?
- ¿Por qué introducen el algoritmo de la sección 7? ¿Qué problema buscan resolver con esto?
- Explicar coloquialmente el significado del teorema 7.

2. Diseñar e implementar un scheduler basado en prioridades fijas y otro en prioridades dinámicas. Para eso complete las clases SchedFixed y SchedDynamic que se encuentran en los archivos sched_fixed.[h—cpp] y sched_dynamic.[h—cpp] respectivamente.

3. Parte III

3.1. Ejercicio 6: TaskBatch

Programar un tipo de tarea TaskBatch que reciba dos parámetros: total cpu y cant bloqueos. Una tarea de este tipo deberá realizar cant bloqueos llamadas bloqueantes, en momentos elegidos pseudoaleatoriamente. En cada tal ocasión, la tarea deberá permanecer bloqueada durante exactamente un (1) ciclo de reloj. El tiempo de CPU total que utilice una tarea TaskBatch deberá ser de total cpu ciclos de reloj (incluyendo el tiempo utilizado para lanzar las llamadas bloqueantes; no así el tiempo en que la tarea permanezca bloqueada).

3.2. Ejercicio 7: Ejecución lote de tareas

Elegir al menos dos métricas diferentes, definirlas y explicar la semántica de su definición. Diseñar un lote de tareas TaskBatch, todas ellas con igual uso de CPU, pero con diversas cantidades de bloqueos. Simular este lote utilizando el algoritmo SchedRR y una variedad apropiada de valores de quantum. Mantener fijo en un (1) ciclo de reloj el costo de cambio de contexto y dos (2) ciclos el de migración. Deben variar la cantidad de núcleos de procesamiento. Para cada una de las métricas elegidas, concluir cuál es el valor óptimo de quantum a los efectos de dicha métrica.

3.3. Ejercicio 8: Scheduler Round-Robin modificado

Implemente un scheduler Round-Robin que no permita la migración de procesos entre núcleos (SchedRR2). La asignación de CPU se debe realizar en el momento en que se produce la carga de un proceso (load). El núcleo correspondiente a un nuevo proceso será aquel con menor cantidad de procesos activos totales (RUNNING + BLOCKED + READY). Diseñe y realice un conjunto de experimentos que permita evaluar comparativamente las dos implementaciones de Round-Robin.

3.4. Ejercicio 9: Ejecución lote de tareas

Diseñar un lote de tareas cuyo scheduling no sea factible para el algoritmo de prioridades fijas pero sí para el algoritmo de prioridades dinámicas.

3.5. Ejercicio 10: Ejecución lote de tareas

Diseñar un lote de tareas, cuyo scheduling sí sea factible con el algoritmo de prioridades fijas, donde se observe un mejor uso del CPU por parte del algoritmo de prioridades dinámicas.