



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico: Scheduling

5 de septiembre de 2015

Sistemas Operativos

Integrante	LU	Correo electrónico
Arribas, Joaquín	702/13	joacoarribas@hotmail.com
Lebrero, Ignacio	702/13	nachitou@hotmail.com
Vázquez, Jérica	702/13	jesis.93@hotmail.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>

Índice

1. Ejercicio 1	3
2. Ejercicio 2	4

Resumen

En este trabajo se implementaron distintas simulaciones interactivas entre tareas. A su vez se implementaron distintas clases de scheduling para interactuar con las tareas creadas, y dichas interacciones se representaron de manera gráfica.

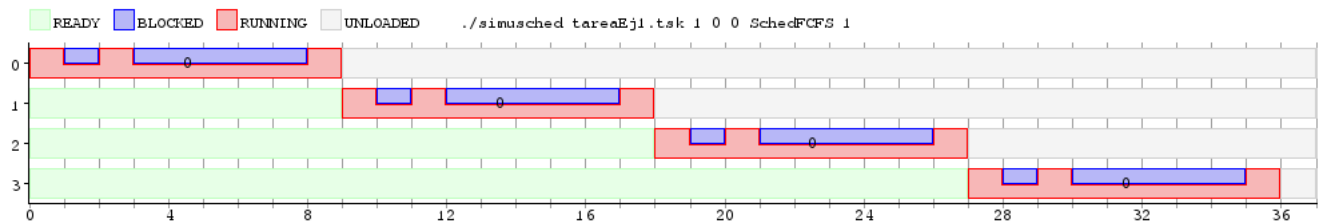
Hare que no se hablar.

1. Ejercicio 1

El ejercicio consiste en implementar una tarea llamada **TaskConsola**, que simule una tarea que realiza llamadas bloqueantes. La tarea recibe por parámetro la cantidad de llamadas bloqueantes que debe realizar, y un intervalo que determina un máximo y un mínimo para la duración de cada una. Dicha duración es generada de manera pseudoaleatoria.

Para resolver el ejercicio creamos una función llamada **generate** que se encarga de realizar la simulación de la tarea. Genera una *semilla* utilizando la función **time** y luego, para cada llamada bloqueante, genera el tiempo usando la función **rand_r**. Para cada valor de la semilla se genera un valor pseudoaleatorio al cual se lo fuerza a caer en el intervalo pasado por parametro, tomándole módulo la distancia entre el máximo y el mínimo, y luego sumándole el mínimo. Una vez calculado el tiempo, se hace la llamada al uso del dispositivo de I/O.

Ejemplo:



El gráfico está corriendo un lote de 4 tareas de tipo **TaskConsola**. El algoritmo de *scheduling* utilizado para representar la interacción entre las tareas es **First Come, First Served**. La cantidad de llamadas bloqueantes son 2 y el intervalo de tiempo para cada llamada es entre 2 y 6. Podemos observar como efectivamente la duración de cada llamada bloqueante pertenece a ese intervalo.

2. Ejercicio 2

El ejercicio consiste en simular la situación que enfrenta nuestro querido amigo Rolando, el cual quiere correr un algoritmo a la vez que escucha música y consume drogas. El algoritmo que corre hace un uso intensivo del cpu por 100 ciclos, mientras que la música y el internet realizan una cantidad determinada de llamadas bloqueantes. La música realiza 20 y el internet 25, cada una de duración variable entre 2 y 4 ciclos. La manera de generar la duración pseudoaleatoria de los ciclos de las llamadas bloqueantes fue la misma que la utilizada en el ejercicio previo, a través de la función **generate**. El algoritmo de *scheduling* utilizado para este ejercicio fue **First Come, First Served**.

Ejemplos: