

Programa de Ingeniería de Sistemas Laboratorio de Sistemas Operativos Proyecto Primer Parcial-2-2025 Grupo A

Fecha: 9 de septiembre de 2025

Planificación de procesos

En un sistema operativo multitarea, los procesos aparentan ser ejecutados en forma simultánea. En realidad, las CPU disponibles deben ser compartidas entre una gran cantidad de procesos, que a su vez ejecutan diversas instrucciones durante un determinado intervalo de tiempo antes de ser desalojados para dar paso a otros.

El problema consiste en **simular** la tarea de planificación de procesos en un sistema operativo. Para ello, deberá implementar las primitivas necesarias para permitir la planificación de procesos utilizando colas de prioridad.

Entrada del programa

Los datos del simulador se leerán por entrada estándar, y consistirán en una serie de comandos (DEFINE, PROCESS, START), que indican la definición de un parámetro de simulación, la llegada de un nuevo proceso o el inicio de la tarea de planificación, respectivamente. Una vez que se lea el comando START (es decir que ha terminado la lectura de los datos), se deberá comenzar la simulación. A continuación, se describe el uso de estos comandos.

Comando	Descripción	Ejemplo
---------	-------------	---------

DEFINE	Define un parámetro para la simulación. Los únicos parámetros que se pueden definir son el número de colas de prioridad a utilizar, la estrategia de planificación (FIFO, RR) para cada cola y el <i>quantum</i> para cada cola.	#Utilizar tres colas de prioridad DEFINE queues 3 #Definir la estrategia para cada cola DEFINE scheduling 1 RR DEFINE scheduling 2 RR DEFINE scheduling 3 FIFO #Definir quantum (prioridad) para cada #cola DEFINE quantum 1 5 DEFINE quantum 2 4 DEFINE quantum 3 2
PROCESS	Anuncia la llegada de un proceso. Sus parámetros son el tiempo de llegada del proceso, el tiempo de ejecución y la prioridad (1 .. n) a la cual pertenece.	PROCESS process1 0 5 1 PROCESS process2 0 3 1 PROCESS process3 0 4 1
START	Anuncia el fin de la lectura de los datos y el inicio de la simulación. No especifica ningún parámetro.	START

Salida del programa

La salida de su programa deberá ser una tabla como la que se muestra a continuación.

Resultados de la simulación

Colas de prioridad: 3

Tiempo total de la simulación: # unidades de tiempo

Tiempo promedio de espera: # unidades de tiempo

# Proceso	T. Llegada	Tamaño	T. Espera	T. Finalización
1	process1	10	30	#
2	process2	0	21	#
3	process3	7	25	#

Secuencia de ejecución

-- presentar la secuencia de ejecución en formato libre, por ejemplo --

Los datos presentados en la tabla son:

- **#:** Numero del proceso, se asigna en forma secuencial de acuerdo con el orden de llegada (PROCESS) de los procesos.
- **Proceso:** Nombre del proceso (nombre del archivo en el cual se encuentra el código del proceso)
- **T. Llegada:** Tiempo de llegada del proceso. Este parámetro es el tiempo especificado en el comando BEGIN dentro del código de alto nivel del proceso.
- **Tamaño:** Tamaño en unidades de tiempo del proceso. Se puede calcular como la suma de todas las instrucciones CPU que se encuentran en el código del programa.
- **T. Espera:** Tiempo total de espera para el proceso.
- **T. Finalización:** Unidad de tiempo en la cual el proceso finaliza la ejecución.

Un ejemplo de secuencia de ejecución para los datos de ejemplo es la siguiente:

2(4) – 2(4) – 3(2) – 1(5) -

El primer proceso en ejecutarse es el proceso 2 (con prioridad 2), ya que en el tiempo 0 la cola de prioridad 1 está vacía. Este proceso alcanza a ejecutar dos tajadas de tiempo de su prioridad (para la prioridad 2, la tajada es 4), Ya que hasta $t=7$ las demás colas de prioridad están vacías. En $t=8$, el proceso 3 ya ha llegado, por lo cual se le da la CPU durante 2 unidades de tiempo. En $t=10$, le

corresponde a la cola de prioridad 1, y se ejecuta el proceso 1 (que coincidentalmente llega en $t=10$), y así sucesivamente.

Observe también que, si una cola de prioridad se planifica con Round Robin, el proceso ingresará al final de la cola. Por el contrario, si la cola de prioridad se planifica con algún algoritmo sin expropiación como FIFO, no se podrá ejecutar ningún otro proceso hasta que el proceso que se haya bloqueado retome su ejecución y termine. Es decir, el proceso volverá al inicio de la cola de prioridad correspondiente.

¿Qué se debe entregar?

En una carpeta denominada **IsoA-proy01-g0X**, (donde X corresponde al número de grupo asignado en este curso), crear 2 subcarpetas denominadas **src/**: En la carpeta **src/** incluir los archivos **.c**, **.h** y el archivo **Makefile**; y crear dentro de **src/** la sub-carpeta **test/** donde se deben almacenar los archivos **.txt**, cuyo contenido configura la prueba a realizar. Antes de enviar los archivos fuente de la solución, limpiar la carpeta **src/** mediante el comando “make clean”. Comprimir la carpeta y subir el archivo comprimido (en formato **.rar** o **.zip**) mediante el enlace que se defina a la plataforma Univirtual.

Fecha de entrega: 24 de septiembre de 2025, hasta las 9:00 am.