C/ Literat Azorín, 1 46702 – GANDIA
Tfno. 962829430 – Fax. 962829431
Codi: 46004221 – CIF: Q-9655627-I
46004221@edu.gva.es
http://iesmariaenriquez.com

COMPUTER SYSTEMS
UNIT7: OPERATING SYSTEM MANAGEMENT Activity 2

CFGS
DAM

PROCESS MANAGEMENT

Para comprender la planificación de procesos en la CPU es mejor empezar experimentando con un sistema que tenga una sola CPU

Paso 1:

Empezaremos las pruebas con prog1-1. Se trata de un programa que ejecuta dos bucles for anidados. El bucle interior no ejecuta ninguna sentencia. El bucle exterior sólo ejecuta el bucle interior. Cada bucle incrementa una variable de tipo entero desde 0 hasta 200000000. Esto asegura que el programa tarda mucho tiempo en ejecutarse.

H Abre prog1-1.c asegúrate de que comprendes su código.

Observarás que es un programa que no hace E/S y que por tanto utiliza la CPU de forma intensiva.

H Ahora vas a observar la ejecución de este programa con el *Comando Top*.

Observa que en este momento el uso de la CPU. Ejecuta prog1-1.exe.

Observarás que se abre una ventana en la que no hay ningún tipo de actividad. Esto es debido a que el proceso no realiza ninguna operación de E/S. ¿Qué ocurre con el uso de la CPU? Indica a continuación el porcentaje de CPU que utilizan en este momento prog1-1.exe.

–Pregunta 5–

% de CPU proceso prog1-1

H Conmuta a la ficha *Rendimiento* del Monitor del systema y observa que el uso de la CPU se encuentra cercano al 100%.

H Retorna a la ficha *Procesos*. Entonces pulsando con el botón derecho del ratón sobre el proceso prog1-1.exe, termina su ejecución, comprobando ahora el uso de la CPU.

Ahora vamos a comprobar cómo la CPU se reparte entre los procesos en ejecución.

H Ahora vas a poner varias instancias del programa prog1-1.exe en ejecución. A continuación, anotarás el porcentaje de CPU que se asigna a cada proceso correspondiente a este programa cuando hay una, dos, tres y cuatro imágenes de él en ejecución. Es posible que tras lanzar cada instancia tengas que esperar un poco de tiempo hasta que se estabilice la planificación.

–Pregunta 6–

Una instancia en ejecución. % de CPU asociado a la instancia: Dos instancias en ejecución. % de CPU asociado a cada instancia: Tres instancias en ejecución. % de CPU asociado a cada instancia: Cuatro instancias en ejecución. % de CPU asociado a cada instancia:

H Termina la ejecución de todos estos procesos, puedes utilizar el comando kill.

H ¿Qué mecanismo utiliza el sistema operativo para que un programa como prog1-1 no monopolice totalmente la CPU cuando se ejecuta?

–Pregunta 7–

C/ Literat Azorín, 1 46702 – GANDIA
Tfno. 962829430 – Fax. 962829431
Codi: 46004221 – CIF: Q-9655627-I
46004221@edu.gva.es
http://iesmariaenriquez.com

COMPUTER SYSTEMS
UNIT7: OPERATING SYSTEM MANAGEMENT Activity 2

CFGS
DAM

Paso 2:

El programa prog1-1 no es un programa de estructura computacional típica, ya que no hace operaciones de E/S. Ahora vamos a continuar las pruebas con el programa prog2-2, que tiene una estructura computacional más común, es decir, computa y hace E/S alternativamente.

H Abre prog1-2.c y analiza su código.

La parte de este programa que usa intensivamente la CPU está formada por dos bucles for anidados. El bucle interior incrementa la variable j de 0 a 200000000.

El bucle exterior incrementa la variable i hasta que ésta iguale a la variable iter.

Este programa hace también E/S, ya que el usuario debe introducir por consola el valor de la variable **iter**. La E/S es realizada por las funciones *printf()* y *scanf_s()*, que pertenecen a la librería estándar del lenguaje C. *printf()* envía a la consola la cadena "Numero de iteraciones: " y *scanf_s()* recoge el número que el usuario introduce por teclado, almacenándolo en la variable iter. La clave está precisamente en *scanf_s()*. Cuando esta función se ejecuta, el proceso queda bloqueado esperando la entrada de datos del usuario. Mientras esto ocurre el proceso no consume CPU.

Tanto los bucles for anidados como las funciones de E/S se encuentran dentro de un bucle while, que se ejecuta infinitamente, por lo que podremos repetir el proceso de E/S y computación indefinidamente. Mientras se produce la espera en la función *scanf_s()* se puede pulsar Ctrl-C, lo que rompe la ejecución del programa.

H Ejecuta prog1-2.exe.

En este momento observarás en el Monitor del Sistema que el proceso no consume CPU, ¿por qué?

–Pregunta 8–

H Introduce 10 iteraciones. ¿Qué ocurre con el consumo de CPU y por qué?

–Pregunta 9–

H Elige la ficha *Rendimiento* Monitor del Sistema y observa el historial de uso de la CPU mientras el programa ejecuta 5, 10, 15 y 20 iteraciones.

Paso 3:

Terminaremos las pruebas de planificación utilizando el programa prog1-3.

H Abre prog1-3.c y analiza su código.

Este programa también genera un uso intensivo de la CPU mediante dos bucles for anidados. Para observar la evolución de la computación, en el bucle exterior se envía un mensaje a la pantalla indicando el número de la iteración.

Antes y después de la ejecución de los bucles anidados se toman tiempos, utilizando la función *time()*,que pertenece a la librería estándar del lenguaje C. El programa finalmente

C/ Literat Azorín, 1 46702 – GANDIA
Tfno. 962829430 – Fax. 962829431
Codi: 46004221 – CIF: Q-9655627-I
46004221@edu.gva.es
http://iesmariaenriquez.com

imprime la diferencia de los tiempos capturados. Dicha diferencia es el tiempo que tardan los bucles anidados en ejecutarse.

H Ejecuta prog1-3 y anota a continuación el tiempo que tarda en ejecutarse.

–Pregunta 10–

H Lanza ahora dos ejecuciones simultáneas de prog1-3 y anota a continuación el tiempo que tardan en ejecutarse.

–Pregunta 11–

H Finalmente lanza cuatro ejecuciones simultáneas de prog1-3 y anota el tiempo que tardan en ejecutarse.

–Pregunta 12–

H ¿Son coherentes los resultados que has obtenido en los experimentos anteriores?. Explica por qué.

–Pregunta 13–