



I OBJETIVOS

- A través de procesos cooperantes, optimizar el uso del CPU en la ejecución de procesos que demandan CPU en arquitecturas de múltiples núcleos (Multicore).
- Usar las llamadas al sistema para crear, esperar y terminar procesos.
- Comparar la solución al problema de usar los núcleos través de hilos contra procesos cooperantes.
- Usar un mecanismo para la comunicación entre procesos.

II BIBLIOGRAFÍA

- Silberschatz, Galvin, Gagne, “SISTEMAS OPERATIVOS”, McGraw Hill, 10ª Ed.
- William Stallings, “SISTEMAS OPERATIVOS”, Prentice Hall, 4ª Ed.
- Neil Matthew & Richard Stones, “BEGINNING LINUX PROGRAMMING”, Wrox, 2ª Ed.
- William Stallings, “ARQUITECTURA DE COMPUTADORAS”, Prentice Hallm 5ª Ed.

III RECURSOS

- Una estación de trabajo con Linux con procesador Multi-Core.

IV ACTIVIDADES

1 *Calcular el valor de PI a través de la regla del trapecio.*

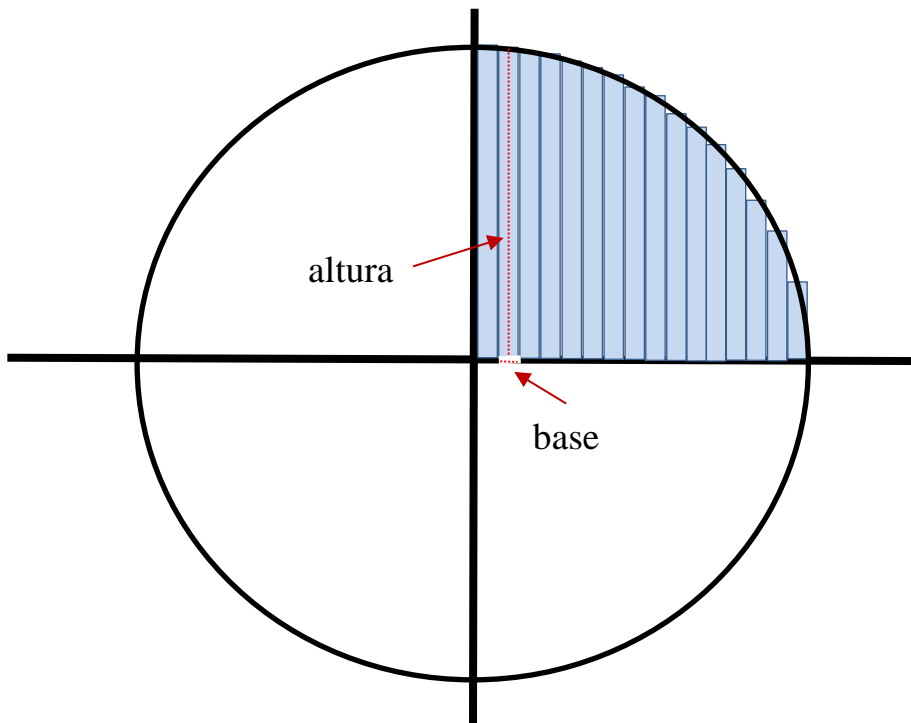
La regla del trapecio es un método de aproximación a la integral definida de una función. Se basa en la idea de aproximar el área bajo una curva mediante una serie de trapecios en lugar de rectángulos, lo que proporciona una estimación más precisa.

En este caso tenemos la función $f(x) = \sqrt{1-y^2}$, que si graficamos esta función de 0 a 1 nos da como resultado un cuarto de círculo. Para hacer el cálculo de PI es



necesario calcular el área que hay debajo del cuarto del círculo y el resultado multiplicarlo por 4.

Para calcular el área que está debajo del cuarto de círculo vamos a usar la regla del trapecio. Esta consiste en dividir el intervalo de 0 a 1 en n intervalos con los que formaremos pequeños rectángulos. Para cada uno de los rectángulos vamos a calcular la altura $f(x)$ tomando x a la mitad del rectángulo, para que con esa altura multiplicada por la base del rectángulo obtenemos el área de cada uno de los rectángulos. La suma del área de todos los rectángulos será $\pi/4$. Mientras más sean los rectángulos y más pequeños, el valor obtenido será más preciso.



1.1 Versión paralela con procesos



Utilizando la llamada al sistema `fork()` que te permite duplicar la imagen de un proceso, desarrolla una versión paralela que haga el cálculo de PI empleando la regla del trapecio de manera que ahora se utilicen todos los núcleos del procesador y se obtenga el mismo resultado en menos tiempo.

Realiza pruebas con tu programa y toma tiempos que se lleva la ejecución, compara con el tiempo obtenido en la versión serial.

Recuerda que cuando creas procesos con la llamada al sistema `fork()`, no se comparten recursos como la memoria, por lo que será necesario que utilices un mecanismo de comunicación entre procesos. Es recomendable usar segmentos de memoria compartida, la cual la puedes solicitar al sistema con la llamada al sistema `shmget()` y conectar con la llamada al sistema `shmat()`.

Realiza pruebas con tu programa y toma tiempos que se lleva la ejecución, compara con el tiempo obtenido en la versión serial.



ENTREGA



No incluya líneas de código en sus programas de las cuales desconozca su funcionamiento. El código no conocido será anulado en el funcionamiento de la práctica.

2 Entrega y Revisión

Todos los programas deben ser desarrollados en C para la plataforma Linux, y deben entregarse en el apartado correspondiente de Canvas un archivo ZIP que contenga los siguientes archivos:

- El programa fuente .c que se pide en la sección: 1.1
- El archivo `Makefile` con el que se puedan compilar el programa tecleando solamente el comando `make`.

La revisión de la práctica se realizará durante de la semana 3.

3 Equipos

Esta práctica se hará en equipos (máximo 2 integrantes), es necesario que en la revisión esté el equipo completo ya que los integrantes que falten a la revisión no tendrán calificación en la práctica.

Importante: Al indicarse que el trabajo debe ser desarrollado por equipos, se entiende que no se permite colaboración entre equipos, cualquier evidencia de esto será considerada plagio.



4 Evaluación

Puntualidad en las revisiones	El equipo estuvo completo y puntual en todas las sesiones de revisión.		Si hubo dos o más sesiones con el equipo, el equipo estuvo completo y puntual en casi todas las sesiones de revisión		Si solo hubo una sesión de revisión, el equipo no estuvo completo o no fue puntual. Si fueron dos o más sesiones de revisión, en más de una sesión el equipo no estuvo completo o fue puntual
	+5		+2.5		0
Especificaciones de entrega	La entrega del producto cumple con todas las especificaciones indicadas en el documento de la práctica, por ejemplo, los archivos se entregan de acuerdo a las formas indicadas en el documento de la práctica.				La entrega del producto no cumple con al menos una de las especificaciones indicadas en el documento de la práctica
	+5				0
Funcionamiento	El producto cumple con todas las especificaciones indicadas en el documento y no tiene fallas	El producto muestra una falla no esperada.	Se da una de las siguientes condiciones: -Está incompleto (falta máximo aprox 50%), pero lo demás puede funcionar bien. -Está completo pero muestra dos fallas	Se da una de las siguientes condiciones: -Está completo pero muestra 3 fallas o más. -Está incompleto (falta máximo aprox 50%) y además muestra fallas -Está incompleto (falta máximo aprox 66%)	El producto no funciona o está incompleto (más del 66%).
	+80	+60	+40	+20	0
Interfaz con el usuario	El producto funciona y pudo ser utilizado sin necesidad de recibir indicaciones por el desarrollador, tiene instrucciones claras para ser utilizado.		El producto funciona, pero hubo necesidad de recibir alguna indicación para su uso por parte del desarrollador del producto		El producto carece de instrucciones claras para ser utilizado y requiere que alguno de los desarrolladores esté presente para su utilización o no puede utilizarse debido a que no está completo
	+5		+3.5		0
Claridad en el código	El código es claro, usa nombres de variables adecuadas, está debidamente comentado e indentado. Puede ser entendido por cualquier otra persona que no intervino en su desarrollo.		El código carece de claridad, puede ser entendido por cualquier persona ajena a su desarrollo pero con cierta dificultad.		El código carece de comentarios, está mal indentado, usa nombres de variables no adecuadas.
	+5		+3.5		0
Defensa del producto	Todos los que presentan la práctica son capaces de explicar cualquier parte del producto presentado		Uno de los que presenta la práctica muestra dudas sobre alguna parte del desarrollo del producto presentado		Más de un integrante no muestra evidencia de que conoce el producto, o si el trabajo fue individual, el desarrollador duda sobre el desarrollo del producto que presenta.
	x 1 (puntos se multiplican por 1)		x 0.5 (puntos se multiplican por 0.5)		x 0 (puntos se multiplican por 0)
Sobresaliente 20 %	Tiene 1 en todos los puntos anteriores. El producto entregado es sobresaliente, muestra tener la calidad para ser expuesto como un producto representativo de la carrera Hay evidencia de que los desarrolladores se documentaron y muestran aprendizajes más allá de lo esperado				No tiene 1 en todos los puntos anteriores, o el producto entregado no es sobresaliente y no muestra tener la calidad para ser expuesto como un producto representativo de la carrera o no hay evidencia de que los desarrolladores se documentaron y muestran aprendizajes más allá de lo esperado
	+20				0