

I OBJETIVOS

Se pretende que después de realizar esta práctica el alumno sea capaz de:

- Implementar los algoritmos de planificación
- Medir el rendimiento de los algoritmos con procesos que usan CPU y/o Entrada y Salida.

II BIBLIOGRAFÍA

William Stallings, “SISTEMAS OPERATIVOS”, Prentice Hall, 4ª Ed.

Silberschatz, Galvin, Gagne, “SISTEMAS OPERATIVOS”, McGraw Hill, 7ª Ed.

Neil Matthew & Richard Stones, “BEGINNING LINUX PROGRAMMING”, Wrox, 2ª Ed.

III RECURSOS

- Una estación de trabajo con Linux y **procesador Intel**¹ con su ambiente gráfico.
- Un editor de texto.
- El compilador de C de UNIX.
- Los archivos Makefile, proceso.c, virtual_processor.h, schedulerFCFS.c, virtual_processor.c. Estos están compactados en el archivo virtual_processor.zip que está disponible en la página del curso

IV DESCRIPCIÓN

Se trata de un procesador virtual que es un proceso que simula la ejecución de procesos usando un subconjunto de funciones del kernel, los procesos que ahí se ejecutan pueden hacer llamadas al sistema (Ejemplo: E/S) que bloquean al proceso. El procesador virtual está preparado para atender eventos como:

- Un tick de reloj
- La llegada de un proceso nuevo en un tick de reloj determinado
- Un proceso realiza una petición de Entrada y Salida; y se bloquea esperando a que termine de atenderse la petición.
- Un proceso notifica que ha terminado de procesar una petición de Entrada y Salida; y está listo para volver a ser ejecutado.
- Un proceso termina

En el momento que ocurre cualquiera de estos eventos, el procesador virtual ejecuta el planificador el cual tiene que decidir cuál es el próximo proceso a ejecutar.

¹ La ejecución del ambiente virtual en procesadores AMD puede entregar resultados diferentes a los esperados durante la ejecución de procesos en el uso del CPU.

Los procesos que se ejecutan son procesos que hacen trabajo de CPU y/o Entrada y Salida, como se vio en la práctica 1.

V ACTIVIDADES

1 *Instalar los programas que crean el ambiente*

En los programas que se le entregan se tiene implementada la planificación FCFS, el algoritmo se encuentra implementado en el archivo `schedulerFCFS.c`. Estos pueden ser compilados utilizando el comando `make`, el cual realiza las acciones especificadas en el `Makefile`.

\$ **make**

Una vez compilados ejecute el programa `fcfs`:

\$ **./fcfs**

Al ejecutarse solicita que se introduzcan los 6 procesos, y de cada uno de los cuales se pide: tiempo de llegada del proceso al sistema, tiempo del CPU y tiempo de Entrada y Salida. Este tiempo está dado en segundos pero considere que el tiempo real de ejecución puede variar ya que este es un procesador virtual que finalmente está siendo planificado por el sistema operativo. Aun así es importante que su práctica sea ejecutada sin ningún otro proceso corriendo.

El conjunto de procesos que va a ejecutar es como se muestra en la Tabla 1.

Proceso	Tiempo de llegada	Tiempo de ejecución CPU	Tiempo de Entrada y Salida	Prioridad
0	0	3	0	0
1	2	0	6	0
2	4	3	2	0
3	6	4	0	0
4	8	0	2	0
5	10	2	2	0

Tabla 1. Conjunto de procesos con sus tiempos de ejecución y entrada y salida

2 Implementación del algoritmo de planificación “Turno Rotatorio $q=1$ ”

2.1 Implementación del algoritmo

Modifique el algoritmo de planificación de manera que ahora la planificación de procesos sea por turno rotatorio (round robin) utilizando un **quantum=1**.

Recuerde que este algoritmo es expropiativo y cuando un proceso solicita E/S éste se bloquea de manera que el planificador escoja solo los procesos que están en la cola de listos.

2.2 Análisis del rendimiento

Ejecute el procesador virtual introduciendo los procesos que se muestran en la tabla 1. Utilice los resultados obtenidos para asegurarse que su algoritmo funciona como lo espera.

Obtenga el tiempo promedio de respuesta y analice en qué afecta que los procesos hagan uso del CPU o usen E/S.

3 Implementación del algoritmo de planificación “El siguiente más corto”

3.1 Implementación del algoritmo

Modifique el algoritmo de planificación de manera que ahora la planificación de procesos sea utilizando el algoritmo “El siguiente más corto”.

Recuerde que este algoritmo es expropiativo y cuando un proceso solicita E/S éste se bloquea de manera que el planificador escoja solo los procesos que están en la cola de listos.

3.2 Análisis del rendimiento

Ejecute el procesador virtual introduciendo los procesos que se muestran en la tabla 1. Utilice los resultados obtenidos para asegurarse que su algoritmo funciona como lo espera.

Obtenga el tiempo promedio de respuesta y analice en qué afecta que los procesos hagan uso del CPU o usen E/S.

4 *Implementación de un algoritmo de planificación expropiativo con prioridades basadas en bandas de prioridad*

Un sistema operativo maneja 6 niveles de prioridad para sus procesos, éstas prioridades son las que se muestran en la Tabla 2.

Nivel	Prioridad
0	Extra-Alta
1	Alta
2	Normal
3	Baja

Tabla 2.- Prioridades soportadas por el sistema operativo

4.1 Implementación del algoritmo de planificación

Modifique el algoritmo de planificación de manera que ahora la planificación de procesos sea por prioridades **implementando 10 colas de procesos listos**, de manera que la cola 0 será la que tenga mayor prioridad y la cola 9 será la de prioridad más baja.

Cada nivel de prioridad de los que se muestra en la Tabla 2, tiene asignadas un conjunto de colas. Un proceso que durante el inicio de su ejecución es establecido en un nivel de prioridad, comienza su ejecución en la cola más alta que le corresponde a su nivel y cada que se ejecuta un cuanto de tiempo desciende un nivel de prioridad hasta llegar a la última cola que le corresponde a su nivel. En la Tabla 3 se muestran las colas asignadas a cada uno de los niveles de prioridad.

Prioridad	Colas
Extra-Alta	De la 0 hasta la 2
Alta	De la 3 hasta la 5
Normal	De la 5 hasta la 7
Baja	De la 7 hasta la 9

Tabla 3.- Colas de prioridad asignadas a los niveles

Recuerde que este algoritmo es expropiativo y cuando un proceso solicita E/S éste se bloquea de manera que el planificador escoja solo los procesos que están en las colas de listos. Cuando un proceso termina E/S regresa a la cola de prioridad más alta que le corresponde a su nivel.

4.2 Análisis del rendimiento

Ejecute el procesador virtual introduciendo los procesos que se muestran en la tabla 1. Utilice los resultados obtenidos para asegurarse que su algoritmo funciona como lo espera.

Obtenga el tiempo promedio de respuesta y analice en qué afecta que los procesos hagan uso del CPU o usen E/S.

VI 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.

1 *Entrega y Revisión*

Entregar en el apartado correspondiente de Moodle en un archivo ZIP los programas fuentes y su respectivo `Makefile` que los compile todos a más tardar el miércoles 6 de Marzo a las 11:59 PM..

2 *Equipos*

Esta práctica se hará en equipos (máximo 3 integrantes), es necesario que en la revisión esté el equipo completo ya que el integrante que no se presente no tendrá 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.

3 Evaluación

Puntualidad en la entrega	El producto fue entregado a tiempo y de acuerdo a las especificaciones/formato de entrega. Si son dos o más sesiones de revisión muestra los avances solicitados en todas las sesiones		El producto que se entregó a tiempo tiene alguna falla y solicitan que se les revise una versión más nueva o no fue entregado de acuerdo a las especificaciones/formato de entrega o en alguna de las sesiones de revisión no tenían los avances esperados.
	+10		0
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
Funcionamiento	El producto cumple con todas las especificaciones indicadas en el documento y no tiene fallas	El producto muestra una falla no esperada o el producto está casi completo, puede funcionar excepto la parte no completada	El producto muestra más de una falla inesperada o no funciona, esto puede ser debido a que no esté completo.
	+25	+12.5	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	+2.5	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	+2.5	0
Defensa del producto	Todos los integrantes son capaces de explicar cualquier parte del producto presentado	Alguno de los integrantes muestra dudas sobre alguna parte del desarrollo del producto presentado	Más de un integrante, o si el trabajo fue individual, el desarrollador duda sobre cómo está desarrollado producto
	+50	+25	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