

SISTEMAS OPERATIVOS

TRABAJO PRÁCTICO DE IMPLEMENTACION N° 1

TPI-01-PP – PLANIFICACION DEL PROCESADOR

Objetivo:

Se trata de programar un sistema que simule distintas estrategias de planificación del procesador (dispatcher), y calcule un conjunto de indicadores que serán utilizados para discutir las ventajas y desventajas de cada estrategia.

Características del sistema a simular:

Asuma que se trata de un sistema multiprogramado y monoprocesador.

El simulador debe leer un archivo en el que cada registro tiene los siguientes datos:

- nombre del proceso
- tiempo de arribo
- cantidad de ráfagas de CPU a emplear para terminar
- duración de la ráfaga de CPU
- duración de la ráfaga de entrada-salida entre ráfagas de CPU
- prioridad externa

Completada la lectura del archivo aceptará una entrada por teclado que indicará la política de planificación a aplicar a la tanda. Como mínimo se deben permitir las siguientes opciones:

- a) FCFS (First Come First Served)
- b) Prioridad Externa
- c) Round-Robin
- d) SPN (Shortest Process Next)
- e) SRTN (Shortest Remaining Time Next)

Finalmente permitirá introducir los siguientes datos:

- Tiempo que utiliza el sistema operativo para aceptar los nuevos procesos (TIP)
- Tiempo que utiliza el sistema operativo para terminar los procesos (TFP)
- Tiempo de conmutación entre procesos (TCP)
- Quantum (si fuera necesario)

El simulador ejecutará la tanda hasta que se hayan completado la totalidad de los trabajos produciendo las siguientes salidas:

Un archivo en el que se indiquen todos los eventos que se producen en el sistema a lo largo de la simulación y el tiempo en el que ocurren los mismos. Ejemplos de eventos: arriba un trabajo, se incorpora un trabajo al sistema, se completa la ráfaga del proceso que se está ejecutando, se agota el quantum, termina una operación de entrada-salida, se atiende una interrupción de entrada-salida, termina un proceso.

Al finalizar la simulación imprimirá y mostrará por pantalla –como mínimo– los siguientes indicadores:

- a) Para cada proceso: Tiempo de Retorno, Tiempo de Retorno Normalizado, Tiempo en Estado de Listo
- b) Para la tanda de procesos: Tiempo de Retorno y Tiempo Medio de Retorno

- c) Para el uso de la CPU: Tiempos de CPU desocupada, CPU utilizada por el SO, CPU utilizada por los procesos (en tiempos absolutos y porcentuales)

Otras condiciones:

- Deberá probarlo con al menos cuatro tandas de trabajos que tengan características distintas cada una y comentar los resultados obtenidos con cada estrategia de planificación en función de las características de las tandas.
- Para resolverlo, utilice java. ~~puede elegir cualquier lenguaje de programación que conozca.~~
- El trabajo es unipersonal.
- Además de probar el simulador en la Universidad, deberá presentar el ejecutable y el código fuente en soporte digital.
- El trabajo correctamente resuelto y presentado antes de rendir el parcial pertinente al tema, exime al alumno de rendir el/los punto/s del mismo que tengan que ver con planificación de procesos, otorgándosele en el examen el máximo puntaje previsto para esos puntos.

Acuerdos para su realización:

a. Orden de procesamiento de eventos:

1. Corriendo a Terminado.
2. Corriendo a Bloqueado.
3. Corriendo a Listo.
4. Bloqueado a Listo.
5. Nuevo a Listo.
6. Finalmente el despacho de Listo a Corriendo.

b. En Round Robin si tenemos un único proceso y su q termina, lo pasamos a listo y luego le volvemos a asignar la cpu (usamos un TCP).

- Para despachar el primer proceso también usamos un TCP.

c. Un proceso pasa de bloqueado a listo instantáneamente (aunque se esté ejecutando otro) y consume 0 unidades de tiempo (este tiempo lo consideramos dentro del TCP posterior).

d. En RR al producirse el cambio de bloqueado a listo de un proceso mientras otro se estaba ejecutando no nos afecta y debemos terminar el tiempo de quantum.

e. Las prioridades las definimos de 1 a 100 siendo los valores mas grandes de mayor prioridad.

f. En Prioridades, y SRT debo expropiarle la CPU a un proceso si, apareció uno con mayor prioridad o con menor tiempo restante y por lo tanto guardo lo que me resta de la ráfaga del proceso que se estaba ejecutando para terminarla cuando le vuelva a tocar.

g. La tanda de trabajos a procesar se cargará en un archivo que el simulador debe leer y será un txt donde cada línea (registro) define un proceso, y cada uno de los campos a saber, se separan por comas:

Campos:

1. Nombre del proceso.
2. Tiempo de arribo.
3. Ráfagas de CPU para completarse.
4. Duración de ráfagas de cpu.
5. Duración de rafagas de I/O.

6. Prioridad.

h. Un proceso no computará estado de listo hasta que no haya cumplido su TIP (inicialmente no computa tiempo de listo).

i. Recordar:

- 1) Tiempo de Retorno de un proceso ($\mathbf{TR_p}$): es desde que arriba el proceso hasta que termina (después de su \mathbf{TFP} , incluyendo éste).
- 2) Tiempo de retorno normalizado ($\mathbf{TR_n}$)= Es el tiempo de Retorno del proceso dividido el tiempo efectivo de CPU que utilizó.
- 3) Tiempo de retorno de la tanda ($\mathbf{TR_t}$)= desde que arriba el primer proceso hasta que se realiza el último \mathbf{TFP} (incluyendo el tiempo de éste).
- 4) Tiempo Medio de retorno de la tanda ($\mathbf{TMR_t}$)= la suma de los tiempos de retorno de los procesos, dividido la cantidad de procesos.

j. Las ráfagas de I/O pueden ejecutar en //.

TRABAJO PRÁCTICO DE IMPLEMENTACION N° 2

TPI-02-AM – ADMINISTRACION DE MEMORIA: ALOCACION CONTIGUA DE MEMORIA – PARTICIONES DINAMICAS

Objetivo:

Se trata de programar un sistema que simule distintas estrategias de asignación de particiones dinámicas de memoria a una tanda de trabajos y calcule un conjunto de indicadores que serán utilizados para discutir las ventajas y desventajas de cada estrategia.

Características del sistema a simular:

Asuma que se trata de un sistema multiprogramado y monoprocesador.

El simulador debe leer un archivo de texto que define una tanda de trabajos. Cada registro describe uno de los trabajos de la tanda mediante los siguientes datos:

- nombre del proceso
- instante de arribo
- duración total del trabajo (tiempo que debe permanecer en memoria principal)
- cantidad de memoria requerida

Completada la lectura del archivo aceptará el ingreso por teclado de los siguientes datos:

- tamaño de la memoria física disponible para usuarios (excluye la utilizada por el sistema operativo)
- estrategia de asignación de particiones. Contemplará al menos las siguientes: first-fit, best-fit, next-fit y worst-fit.
- tiempo de selección de partición (incluye el recálculo de la tabla de particiones en caso de corresponder).
- tiempo de carga promedio (media del tiempo que toma cargar de memoria secundaria a principal un programa).
- tiempo de liberación de partición.

El simulador simulará la tanda hasta que se hayan completado la totalidad de los trabajos produciendo las siguientes salidas:

- Un archivo en el que se indiquen todos los eventos que se producen en el sistema a lo largo de la simulación y el tiempo en el que ocurren los mismos. Ejemplos de eventos: se selecciona una partición para el trabajo x , se carga el trabajo y , termina el trabajo z , etc. En el mismo archivo (o en uno asociado a éste por el tiempo en el que ocurre un evento), se guardará el estado de la tabla de particiones cada vez que se modifique la misma (al momento de generar nuevas particiones para cargar un trabajo o unificar particiones por terminar otro). Se deberá conservar, como mínimo, identificación de la partición, dirección de comienzo, tamaño y estado (libre/ocupada).

Al finalizar la simulación imprimirá y mostrará por pantalla –como mínimo– los siguientes indicadores:

- a) Para cada proceso: Tiempo de Retorno.
- b) Para la tanda de procesos: Tiempo de Retorno, Tiempo Medio de Retorno e Índice de Fragmentación Externa.

Con el fin de estudiar, comparar estrategias y validar el simulador, Ud. debe proveer una tanda de trabajos para cada estrategia, tal que ésta sea mejor que las otras en términos de Tiempo de Retorno.

Otras condiciones:

- a) Deberá probarlo con al menos cuatro tandas de trabajos que tengan características distintas cada una y comentar los resultados obtenidos con cada estrategia de selección en función de las características de las tandas.
- b) Resuelva utilizando el lenguaje de programación Java.
- c) El trabajo es unipersonal.
- d) Además de probar el simulador en clase, deberá presentar el ejecutable y el código fuente en soporte digital.
- e) El trabajo correctamente resuelto y presentado antes de rendir el parcial pertinente al tema, exime al alumno de rendir el/los puntos del mismo que tengan que ver con Administración de Memoria con especificación dinámica de particiones, otorgándose en el examen el máximo puntaje previsto para esos ítems.

TRABAJO PRÁCTICO DE IMPLEMENTACION N° 3

TPI-03-SD – SCHEDULING DE DISCO (MAGNETICO)

Objetivo:

Se trata de programar un sistema que simule distintos algoritmos de scheduling de peticiones a un disco magnético y calcule un conjunto de indicadores que serán utilizados para discutir las ventajas y desventajas de cada uno de ellos.

Características del sistema a simular:

Asuma que se trata de un sistema multiprogramado y monoprocesador.

El simulador debe permitir armar y mantener hasta dos cadenas de peticiones a disco determinadas por un **nro. de petición** más el **número de pista** donde se encuentra el bloque requerido. Cada cadena de peticiones a disco debe aceptar por lo menos 15 requerimientos de **bloques** de disco y la simulación ocurre solamente para una de las cadenas, la que se seleccione antes de correr la simulación.

Además, el programa debe requerir el ingreso de los siguientes parámetros del disco rígido:

ST_M: seek time medio.

VR: Velocidad de rotación del disco dada en rpm.

TT_{1S}: Tiempo de transferencia de 1 sector.

TB: Tamaño de bloque dado en sectores.

TP: Total de pistas del disco.

PC: Posición de las cabezas de lectura/escritura dada en nro. de pista (desde 0 a PT).

SC: Sentido en el que está avanzando las cabezas (C = creciente, D = Decreciente), sólo en el caso que $PC \neq 0$, $PC \neq TP$ y la estrategia lo requiera.

Algoritmo a utilizar:

- FIFO (First In First Out)
- SSTF (Shortest Seek Time First)
- SCAN
- C-SCAN
- LOOK
- C-LOOK
- F-SCAN

Al elegir esta estrategia, el simulador debe requerir además, que se indique cómo se divide la lista.

- N-STEP-SCAN

Al elegir esta estrategia, el simulador debe requerir además, que se indique el parámetro adicional de longitud de lista **N**.

Una vez ingresados todos los datos y parámetros, un botón debe permitir iniciar la simulación. El simulador ejecutará la estrategia de scheduling sobre la cadena de peticiones de bloques hasta que se haya completado la totalidad de los requerimientos, produciendo como salida el siguiente detalle:

- La propia cadena de peticiones que se simuló.
- La estrategia utilizada por el scheduler de disco para simular.
- Si la estrategia elegida para simular fuera F-SCAN o N-STEP-SCAN se debe detallar cómo quedaron armadas las sublistas.
- Detalle del orden en el que se atendieron las peticiones, en el que se especifique el orden, nro. de petición, pista o cilindro requerido y distancia parcial recorrida para satisfacer ese requerimiento.
- La **d_t** (distancia total) que arroja el algoritmo.

- El **rr** (retardo rotacional) calculado en base a VR.
- El **tt_{IB}** (tiempo de transferencia de 1 bloque) calculado en base al **TT_{IS}** y el TB.
- El tiempo de acceso total dado por la fórmula general: **ta_T** = st + rr + tt. No solo se debe indicar el resultado de **ta_T**, sino que se debe poder ver cuántos **ST_M** y rr se han considerado en este cálculo.

Con el fin de estudiar, comparar estrategias y validar el simulador, Ud. debe proveer una lista de peticiones de bloques de disco para cada estrategia, tal que ésta sea mejor que las otras en términos de **ta_T**, **elaborando una conclusión sobre el tema.**

Asuma que:

- El disco magnético en cuestión posee un mecanismo de cabezas solidarias (todas se mueven juntas) o bien hay una única cara de un único plato. Esto además implica que no hay tiempo de selección de cabezas a considerar.
- El dispositivo está siempre listo; no hay que considerar este delay antes de procesar una lista de requerimientos.

Otras condiciones:

- a) Resuelva utilizando el lenguaje de programación Java.
- b) El trabajo es unipersonal.
- c) Además de probar el simulador en clase, deberá presentar el ejecutable y el código fuente en soporte digital.
- d) El trabajo correctamente resuelto y presentado antes de rendir el parcial pertinente al tema, exime al alumno de rendir el/los puntos del mismo que tengan que ver con Scheduling de Disco, otorgándose en el examen el máximo puntaje previsto para ese/os ítems.