

4. El procesador de un computador recibe una interrupción del reloj hardware cada 10 ms. Cuando recibe la interrupción el sistema operativo salva el contexto del proceso en ejecución, ejecuta la rutina de reconocimiento de interrupciones y ésta invoca a la rutina de tratamiento de la interrupción de reloj. Cuando finaliza de atenderse la interrupción de reloj se restaura el contexto del proceso interrumpido y se continua con su ejecución. Si un proceso finaliza se ejecuta el planificador, que utiliza el algoritmo de planificación SJF, y se realiza un cambio de proceso. En la Tabla 1 se muestran los tiempos promedios que tardan en ejecutarse cada una de estas tareas. Supuesto que en el instante de tiempo  $T_1=100$  s llega una interrupción de reloj mientras se estaba ejecutando un proceso A al que le restaban 0.8 s de ejecución y que la cola de procesos preparados contiene a los procesos B, E y F, cuyos tiempos de servicio son de 2 s, 3.2 s y 1.6 s, respectivamente, se pide:

- a) (1 p) Determinar el instante de finalización del proceso A (expresado en  $\mu$ s) supuesto que durante su ejecución únicamente llegan interrupciones de reloj.
- b) (1 p) Durante el intervalo de tiempo  $T_1=100$  s y  $T_2= 101$  s determinar la sobrecarga (overhead) del sistema, es decir, el porcentaje de tiempo que el procesador no ha podido ser utilizado para la ejecución de procesos sino para la ejecución de tareas del sistema operativo.

Tareas del sistema operativo	Tiempo de ejecución promedio (en $\mu$ s)
Salvar el contexto de un proceso	1,25
Restaurar el contexto de un proceso	1,25
Cambio de proceso	6
Reconocimiento de interrupciones	2
Tratamiento de la interrupción del reloj	2
Planificador	4

Tabla 1

**Solución:**

a) De acuerdo con el enunciado cuando llega una interrupción de reloj (IR) el sistema operativo realiza las siguientes acciones:

- 1) Salva el contexto del proceso en ejecución. Esta tarea requiere un tiempo promedio de ejecución (ver Tabla 1) de 1.25  $\mu$ s
- 2) Ejecuta la rutina de reconocimiento de interrupciones. Esta tarea requiere un tiempo promedio de ejecución de 2  $\mu$ s.
- 3) Invoca a la rutina de tratamiento de la interrupción de reloj. La ejecución de esta rutina requiere un tiempo promedio de 2  $\mu$ s.
- 4) Restaura el contexto del proceso interrumpido. Esta tarea requiere un tiempo promedio de ejecución de 1.25  $\mu$ s.

Luego el tiempo promedio  $T_{IR}$  que emplea el sistema operativo (SO) en tareas asociadas a una IR es:

$$T_{IR} = 1.25 + 2 + 2 + 1.25 = 6.5 \mu s$$

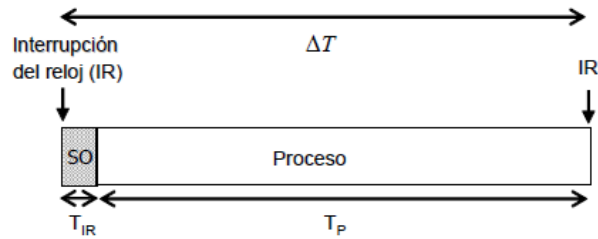


Figura 2

Se sabe, por el enunciado que cada  $\Delta T = 10 \text{ ms} = 10000 \mu s$  llega una IR. Si denotamos como  $T_P$  al tiempo entre dos interrupciones de reloj disponible para ejecutar un proceso (supuesto que no llegan otro tipo de interrupciones) se cumplirá por tanto la siguiente relación (ver Figura 2):

$$\Delta T = T_{IR} + T_P$$

Despejando  $T_P$ , sustituyendo valores y operando se obtiene que

$$T_P = \Delta T - T_{IR} = 10000 - 6.5 = 9993.5 \mu s$$

En  $T_1 = 100 \text{ s}$  llega una IR y al proceso A le restan por ejecutarse  $0.8 \text{ s}$ . El diagrama de Gantt asociado a la ejecución del proceso A desde el instante  $T_1$  hasta que finaliza en  $T_F$  tiene la forma general que se muestra en la Figura 3.

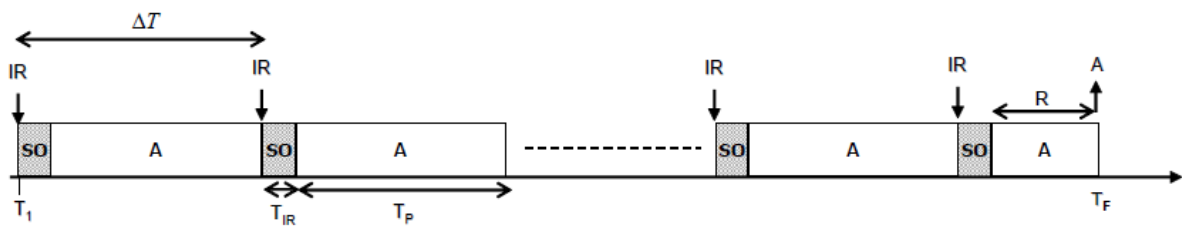


Figura 3

De la observación de este diagrama se deduce que el instante de finalización  $T_F$  del proceso A se puede calcular mediante la siguiente expresión:

$$T_F = T_1 + N \cdot \Delta T + T_{IR} + R = T_1 + N \cdot T_P + (N + 1) \cdot T_{IR} + R$$

N es el número de periodos completos  $T_P$  que requiere el proceso A para poder completar los  $T_S = 0.8 \text{ s} = 800000 \mu\text{s}$  de ejecución que le restan y se calcula de la siguiente forma:

$$N = \frac{T_S}{T_P} = \frac{800000}{9993.5} = 80.052$$

Se observa que no es un número entero eso significa que necesita para completarse de 80 periodos completos  $T_P$  y una parte R de otro periodo. R es el resto de la división entera entre  $T_S$  y  $T_P$ . Se cumple por tanto la siguiente relación:

$$T_S = N \cdot T_P + R$$

Luego

$$R = T_S - N \cdot T_P = 800000 - 80 \cdot 9993.5 = 520 \mu\text{s}$$

Sustituyendo valores en la expresión del tiempo de finalización y operando se obtiene finalmente el siguiente resultado:

$$T_F = T_1 + N \cdot \Delta T + T_{IR} + R = 100 \cdot 10^6 + 80 \cdot 10000 + 6.5 + 520 = 100800526.5 \mu\text{s}$$

**b)** En este apartado piden determinar el porcentaje de tiempo que el procesador ha sido utilizado por el SO durante el intervalo de tiempo  $T_1 = 100 \text{ s}$  y  $T_2 = 101 \text{ s}$ .

De acuerdo con el diagrama de Gantt de la Figura 2 desde  $T_1 = 100 \text{ s}$  hasta  $T = 100.80 \text{ s}$  el tiempo  $T_{S01}$  que el SO ha usado el procesador es igual a

$$T_{S01} = N \cdot T_{IR} = 80 \cdot 6.5 = 520 \mu\text{s}$$

Hasta que llega la próxima interrupción de reloj en  $T = 100.81 \text{ s}$  (ver Figura 4), se atiende una IR, se ejecuta el proceso A hasta su finalización, se ejecuta el planificador del SO durante  $4 \mu\text{s}$  y se realiza un cambio de proceso que tarda  $6 \mu\text{s}$ . Como el planificador utiliza el algoritmo de planificación SJF (primero la tarea más corta) elige al proceso F para ser ejecutado ya que es el que tiene el tiempo de servicio más pequeño ( $1.6 \text{ s}$ ). Luego desde  $T_1 = 100.80 \text{ s}$  hasta  $T = 100.81 \text{ s}$  el tiempo  $T_{S02}$  que el SO ha usado el procesador es igual a

$$T_{S02} = 6.5 + 4 + 6 = 16.5 \mu\text{s}$$

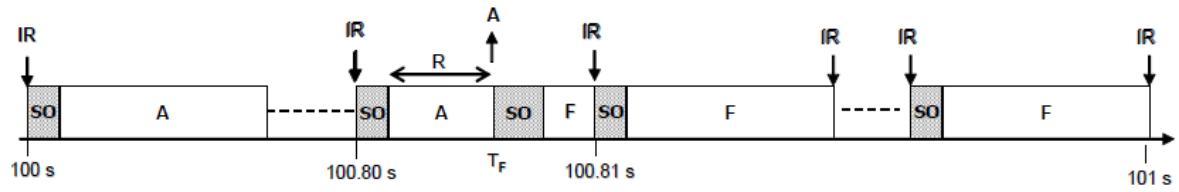


Figura 4

Desde  $T=100.81$  s hasta  $T=101$  s hay que atender un total de  $N_{IR}$  interrupciones de reloj, esta cantidad se determina de la siguiente forma:

$$N_{IR} = \frac{101s - 100.81s}{10 \cdot ms} = \frac{0.19}{0.01} = 19 \text{ interrupciones}$$

Luego desde  $T_1=100.81$  s hasta  $T=101$  s el tiempo  $T_{SO2}$  que el SO ha usado el procesador es igual a

$$T_{SO3} = N_{IR} \cdot T_{IR} = 19 \cdot 6.5 = 123.5 \mu s$$

Finalmente, el porcentaje de tiempo  $U_{SO}$  que el procesador ha sido utilizado por el SO durante el intervalo de tiempo  $T_1=100$  s y  $T_2=101$  s se calcula de la siguiente forma:

$$U_{SO} = \frac{T_{SO1} + T_{SO2} + T_{SO3}}{T_2 - T_1} \cdot 100 = \frac{520 + 16.5 + 123.5}{(101 - 100) \cdot 10^6} \cdot 100 = 0.066 \%$$

**2. (2 p) Explique razonadamente que es un *bloque de control de procesos* (BCP) y que información contiene.**

**Solución:**

Un *bloque de control de procesos* (BCP) es una estructura de datos donde el sistema mantiene toda la información sobre un proceso. La creación de un proceso origina la creación de su BCP que sirve para describirle hasta que se elimina o pasa al estado muerto, donde los procesos no poseen un BCP.

En el BCP se guarda la información que necesita el sistema para controlar al proceso y dar cuenta de sus recursos y toda la información que influye en la ejecución de un programa, como por ejemplo:

- El identificador único del proceso (pid)
- El estado del proceso (activo, preparado, bloqueado,...)
- La prioridad
- El estado hardware (contador de programa, códigos de condición, punteros de pila, etc)
- Información para gestionar la memoria (punteros, tablas, registros)
- Información de estado del sistema de E/S (dispositivos de E/S asignados al proceso, lista de archivos abiertos, etc)
- Información de contabilidad y planificación

3. Sean cuatro procesos A, B, C y D cuyos tiempos de llegada y de ejecución expresados en unidades de tiempo (u.t.) se muestran en la tabla adjunta. Supuesto que el tiempo de cambio de contexto es de 0.5 u.t. y que se utiliza en la planificación el algoritmo SRT se pide:

a) (1 p) Dibujar el diagrama de Gantt.

b) (1 p) Calcular el tiempo de retomo y el tiempo de espera de cada proceso.

Proceso	Tiempo de llegada (u.t.)	Tiempo de ejecución (u.t.)
A	0	5
B	1	3
C	2	4
D	3	2

**Solución:**

a) El algoritmo SRT (Shortest-Remain-Time) elige para ejecutar a continuación aquel proceso al que le queda menos tiempo para terminar su ejecución, incluyendo a los nuevos que lleguen.

Cuando llega el proceso A en  $t=0$  u.t., puesto que no hay procesos en la cola de preparados se planifica inmediatamente.

En  $t= 1$  u.t. llega el proceso B, como su tiempo de servicio es de 3 u.t. menor que las 4 u.t de servicio que le restan al proceso A, se interrumpe el proceso A y se comienza a ejecutar el proceso B en  $t=1.5$  u.t una vez finalizado el cambio de contexto.

En  $t = 2$  u.t. llega el proceso C, como su tiempo de servicio es de 4 u.t mayor que las 2.5 u.t de tiempo de servicio que le restan al proceso B, el proceso B continúa ejecutándose.

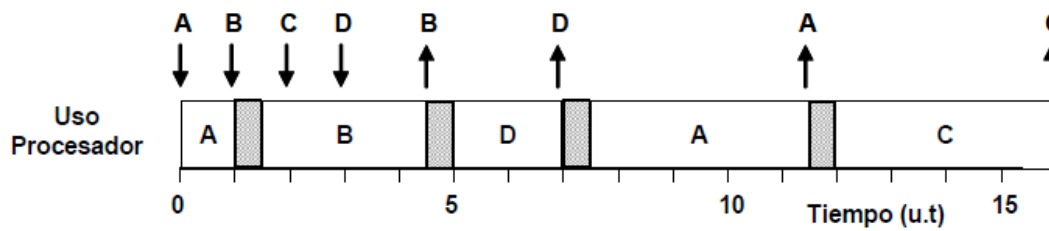
En  $t= 3$  u.t. llega el proceso D, como su tiempo de servicio (2 u.t.) es mayor que el tiempo de servicio que le resta (1.5 u.t.) al proceso B continúa ejecutándose éste.

En  $t=4.5$  u.t finaliza el proceso B, en la cola de listos se encuentran los procesos A, C y D cuyos tiempo de servicio restantes son 4, 4 y 2 u.t. En consecuencia se planifica el proceso D por tener el menor tiempo de ejecución restante y comienza a ejecutarse en  $t= 5$  u.t.

En  $t=7$  u.t. finaliza el proceso D y se planifica el proceso A, ya que está antes en la cola que el proceso C. Conviene recordar que en igualdad de tiempo de servicio restante se aplica la planificación FCFS. El proceso A comienza a ejecutarse en  $t= 7.5$  u.t.

En  $t=11.5$  u.t finaliza el proceso A y se planifica el proceso C que comienza a ejecutarse en  $t= 12$  u.t. , cuya ejecución finaliza en  $t= 16$  u.t.

En la siguiente figura se representa el diagrama de Gantt, se han añadido al mismo unas flechas para resaltar los instantes de llegada y finalización de cada proceso. Las zonas sombreadas representan los cambios de contexto.



b) En la siguiente tabla se muestran los valores del tiempo de llegada  $T_{LL}$ , tiempo de finalización  $T_F$ , tiempo de retorno  $T_R$ , tiempo de ejecución o servicio  $T_S$  y el tiempo de espera  $T_E$  para los procesos A, B, C y D. Recuerdese que  $T_R = T_F - T_{LL}$  y que  $T_E = T_R - T_S$

Procesos	$T_{LL}(u.t)$	$T_F(u.t)$	$T_R(u.t)$	$T_S(u.t)$	$T_E(u.t)$
A	0	11.5	11.5	5	6.5
B	1	4.5	3.5	3	0.5
C	2	16	14	4	10
D	3	7	4	2	2

1. Explique **razonadamente** si las siguientes afirmaciones son verdaderas o falsas:

- I) (1 p) El planificador a corto plazo (PCP) determina que trabajos se admiten en el sistema para su procesamiento y son, por lo tanto, cargados en la memoria disponible.
- I) (1 p) El DMA por ráfagas interfiere más la actividad de la CPU que el DMA por robo de ciclos..

**Solución:**

I) El planificador a corto plazo [PCP] se encarga de seleccionar al proceso que pasará al estado activo de entre todos los procesos residentes en memoria que se encuentran en el estado preparado. Es el planificador a largo plazo [PLP] el que determina que trabajos se admiten en el sistema para su procesamiento y son, por lo tanto, cargados en la memoria disponible. Por lo tanto la afirmación es **FALSA**.

II) En el DMA por ráfagas, cuando el controlador de DMA toma el control del bus no lo libera hasta haber transmitido el bloque de datos solicitado. Obviamente de esta forma se consigue la mayor velocidad de transferencia posible pero se tiene a la CPU inactiva durante periodos de tiempo relativamente grandes, por lo que es la estrategia de DMA que más interfiere en la actividad normal de la CPU. Por lo tanto la afirmación es **VERDADERA**.