

# SISTEMAS OPERATIVOS

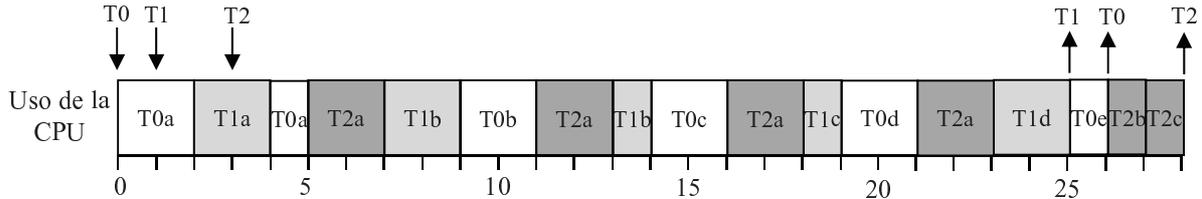
## Solución PED1 (Noviembre 2016)

### Solución Ejercicio 1

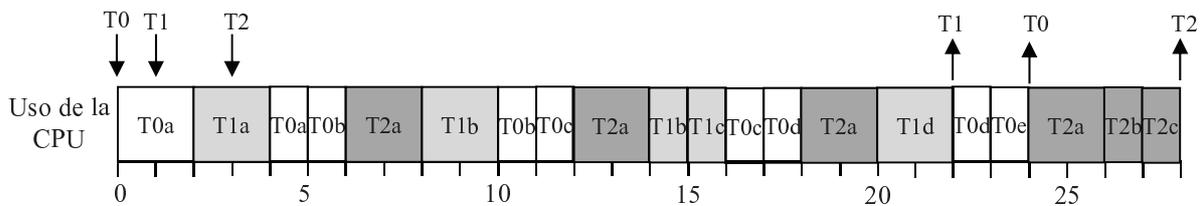
- I) El bloqueo de las interrupciones aunque es la solución más simple para garantizar la exclusión mutua presenta sin embargo serios inconvenientes. En primer lugar el rendimiento del sistema se puede degradar bastante al no permitirle atender las interrupciones más prioritarias en el momento en que llegan. Además se deja en manos del proceso la decisión de ceder el procesador y si éste nunca lo devuelve el sistema se quedará colgado. En conclusión la afirmación es **FALSA**.
  
- II) El *tiempo de conmutación* de un cambio de proceso depende principalmente de factores asociados al hardware del computador como por ejemplo, la velocidad de lectura/escritura de la memoria principal, el número de registros del procesador cuyo contenido hay que salvar o cargar, la velocidad del procesador y la existencia de instrucciones especiales en el repertorio del procesador para salvar o cargar todos los registros. En conclusión la afirmación es **FALSA**.
  
- III) Se denomina *grado de multiprogramación* al número de programas cargados en memoria principal. En conclusión la afirmación es **FALSA**.
  
- IV) El algoritmo de planificación basado en prioridades puede ser expropiativo o no expropiativo. En su implementación expropiativa, si llega a la cola de procesos preparados un proceso B con una prioridad mayor que la del proceso actual A, entonces se interrumpe la ejecución de A y se planifica inmediatamente al proceso B. Un algoritmo de planificación basado en prioridades expropiativo se dice que está *guiado por eventos* (event-driven). Por el contrario en su implementación no expropiativa la planificación del proceso B solo se realiza cuando el proceso A se bloquea o finaliza. En conclusión la afirmación es **FALSA**.

## Solución Ejercicio 2

- a) El final de una ráfaga de CPU de un proceso (trabajo) lo marca el que éste tenga que bloquearse en espera de algún evento: la finalización de una operación de E/S, la finalización de la espera en un semáforo o en una cola de mensajes, etc. Si el proceso se bloquea, independientemente de la circunstancia que lo produzca, debe activarse el planificador. Como ésto no se explica explícitamente ni el libro base ni el enunciado del ejercicio se va a considerar por bueno el haber considerado que la finalización de una ráfaga de CPU no produce la invocación del planificador excepto si finaliza el cuanto. En consecuencia se tienen dos posibles soluciones (ver Figura 1 y Figura 2).



**Figura 1** – Solución 1: Se considera que el final de una ráfaga de CPU (se nombran con letras) produce la activación del planificador



**Figura 2** – Solución 2: Se considera que el final de una ráfaga de CPU (se nombran con letras) NO produce la activación del planificador salvo que finalice el cuanto

- b) **Solución 1:** En la Tabla 1 se muestran los los valores de tiempos de llegada  $T_{LL}$ , tiempos de finalización  $T_F$ , tiempos de retorno  $T_R$ , tiempos de servicio  $T_S$  y tiempos de espera  $T_E$  para los procesos T0, T1 y T2 de acuerdo con la Figura 1. Recuérdese que el tiempo de retorno se calcula como la diferencia entre el tiempo de finalización  $T_F$  y el tiempo de llegada  $T_{LL}$ :

$$T_R = T_F - T_{LL} \quad (1)$$

Mientras que el tiempo de espera  $T_E$  se calcula como la diferencia entre el tiempo de retorno  $T_R$  y el tiempo de servicio o ejecución  $T_S$ :

$$T_E = T_R - T_S \quad (2)$$

Trabajos	$T_{LL}$ (ut)	$T_F$ (ut)	$T_R$ (ut)	$T_S$ (ut)	$T_E$ (ut)
T0	0	26	26	10	16
T1	1	25	24	8	16
T2	3	28	25	10	15

**Tabla 1**

**Solución 2:** En la Tabla 2 se muestran los los valores de tiempos de llegada  $T_{LL}$ , tiempos de finalización  $T_F$ , tiempos de retorno  $T_R$ , tiempos de servicio  $T_S$  y tiempos de espera  $T_E$  para los procesos T0, T1 y T2 de acuerdo con la Figura 2.

Trabajos	$T_{LL}$ (ut)	$T_F$ (ut)	$T_R$ (ut)	$T_S$ (ut)	$T_E$ (ut)
T0	0	24	24	10	14
T1	1	22	21	8	13
T2	3	28	25	10	15

**Tabla 2**

c) El tiempo transcurrido en el sistema para completarse una ráfaga se define como el tiempo transcurrido desde el momento en que podría comenzarse a ejecutarse la ráfaga, lo cual no puede suceder hasta que se complete la ráfaga anterior, hasta que finaliza dicha ráfaga. Esta definición no aparece en el libro base ni tampoco se indica en el enunciado por lo que se va a considerar por bueno el haber considerado que el origen desde el que se empieza a medir el tiempo transcurrido es el tiempo de llegada del trabajo al que pertenece la ráfaga.

Teniendo en cuenta lo anterior y las dos soluciones del apartado a) se pueden encontrar los siguientes casos a la hora de calcular el tiempo de respuesta promedio de cada uno de los trabajos:

- Caso 1: Se toma la Figura 1 y se considera que el tiempo transcurrido para completarse una ráfaga comienza desde el instante en que termina la ráfaga anterior.

$$T_{respuesta}(T0) = \frac{(5 - 0) + (11 - 5) + (16 - 11) + (21 - 16) + (26 - 21)}{5} = \frac{26}{5} = 5,2 \text{ ut}$$

$$T_{respuesta}(T1) = \frac{(4 - 1) + (14 - 4) + (19 - 14) + (25 - 19)}{4} = \frac{24}{4} = 6 \text{ ut}$$

$$T_{respuesta}(T2) = \frac{(23 - 3) + (27 - 23) + (28 - 27)}{3} = \frac{25}{3} = 8,3 \text{ ut}$$

- Caso 2: Se toma la Figura 1 y se considera que el tiempo transcurrido para completarse una ráfaga comienza desde el instante de llegada del trabajo.

$$T_{respuesta}(T0) = \frac{(5 - 0) + (11 - 0) + (16 - 0) + (21 - 0) + (26 - 0)}{5} = \frac{79}{5} = 15,8 \text{ ut}$$

$$T_{respuesta}(T1) = \frac{(4 - 1) + (14 - 1) + (19 - 1) + (25 - 1)}{4} = \frac{58}{4} = 14,5 \text{ ut}$$

$$T_{respuesta}(T2) = \frac{(23 - 3) + (27 - 3) + (28 - 3)}{3} = \frac{69}{3} = 23 \text{ ut}$$

- Caso 3: Se toma la Figura 2 y se considera que el tiempo transcurrido para completarse una r faga comienza desde el instante en que termina la r faga anterior.

$$T_{respuesta}(T0) = \frac{(5 - 0) + (11 - 5) + (17 - 11) + (23 - 17) + (24 - 23)}{5} = \frac{24}{5} = 4,8 \text{ ut}$$

$$T_{respuesta}(T1) = \frac{(4 - 1) + (15 - 4) + (16 - 15) + (22 - 16)}{4} = \frac{21}{4} = 5,25 \text{ ut}$$

$$T_{respuesta}(T2) = \frac{(26 - 3) + (27 - 26) + (28 - 27)}{3} = \frac{25}{3} = 8,3 \text{ ut}$$

- Caso 4: Se toma la Figura 2 y se considera que el tiempo transcurrido para completarse una r faga comienza desde el instante de llegada del trabajo.

$$T_{respuesta}(T0) = \frac{(5 - 0) + (11 - 0) + (17 - 0) + (23 - 0) + (24 - 0)}{5} = \frac{80}{5} = 16 \text{ ut}$$

$$T_{respuesta}(T1) = \frac{(4 - 1) + (15 - 1) + (16 - 1) + (22 - 1)}{4} = \frac{53}{4} = 13,25 \text{ ut}$$

$$T_{respuesta}(T2) = \frac{(26 - 3) + (27 - 3) + (28 - 3)}{3} = \frac{72}{3} = 24 \text{ ut}$$

### Soluci n Ejercicio 3

En las Figuras 3 y 4 se muestra una posible soluci n para este ejercicio. Esta soluci n utiliza los siguientes sem foros binarios:

- S13. Se utiliza para que el robot R1 pueda avisar al robot R3 de que ya ha colocado dos piezas tipo T1 en la bandeja compartida.
- S23. Se utiliza para que el robot R2 pueda avisar al robot R3 de que ya ha colocado dos piezas tipo T2 en la bandeja compartida.
- S31. Se utiliza para que el robot R3 pueda avisar al robot R1 de que ya puede reanudar su actividad.
- S32. Se utiliza para que el robot R3 pueda avisar al robot R2 de que ya puede reanudar su actividad.

```

#define TRUE 1;
semáforo_binario S13, S23, S31, S32; /* Definición semáforos binarios */

void R1() /* Robot R1 */
{
    int h;
    while(TRUE)
    {
        for(h=1;h<=2;h++) /* Fabricar y colocar dos piezas T1 */
        {
            fabricar_pieza_T1();
            colocar_pieza_T1_en_bandeja();
        }
        signal_sem(S13); /* Avisar a R3 */
        wait_sem(S31); /* Esperar a que le avise R3 */
    }
}

void R2() /* Robot R2 */
{
    int h;
    while(TRUE)
    {
        for(h=1;h<=2;h++) /* Fabricar y colocar dos piezas T2 */
        {
            fabricar_pieza_T2();
            colocar_pieza_T2_en_bandeja();
        }
        signal_sem(S23); /* Avisar a R3 */
        wait_sem(S32); /* Esperar a que le avise R3 */
    }
}

```

**Figura 3** – Solución Ejercicio 3 (1ª parte)

```

void R3() /* Robot R3 */
{
    while(TRUE)
    {
        wait_sem(S13); /* Esperar a que le avise R1 */
        wait_sem(S12); /* Esperar a que le avise R2 */
        recoger_piezas_bandeja();
        signal_sem(S31); /* Avisar a R1 */
        signal_sem(S32); /* Avisar a R2 */
        fabricar_pieza_T3();
    }
}

main() /* Inicialización semáforos y ejecución concurrente */
{
    init_sem(S13,0);
    init_sem(S23,0);
    init_sem(S31,0);
    init_sem(S32,0);
    vaciar_bandeja_compartida(); /* Para que los robots operen correctamente */
    ejecución_concurrente(R1,R2,R3);
}

```

**Figura 4** – Solución Ejercicio 3 (2ª parte)