

SISTEMAS OPERATIVOS

Solución Primera Prueba de Evaluación a Distancia (PED1)

Solución Ejercicio 1.

- I) Se denomina sección crítica o región crítica de un proceso a una instrucción o conjunto de instrucciones secuenciales de su código que requieren manipular un recurso compartido (variable, dato, fichero,...) con otros procesos. Dentro del código de un proceso pueden existir varias secciones críticas. Por tanto, la afirmación del enunciado es **FALSA**.
- II) La creación de un hilo nuevo dentro de un proceso ya existente requiere de menor tiempo que la creación de un nuevo proceso. Asimismo el tiempo necesario para finalizar un hilo es menor que el tiempo necesario para finalizar un proceso. Además, un cambio de proceso requiere más tiempo que un cambio de hilo dentro de un mismo proceso. Por tanto, la afirmación del enunciado es **VERDADERA**.
- III) Un sistema operativo multitarea es aquel en cuya memoria principal se almacenan, aparte del sistema operativo, varios programas que deben compartir durante su ejecución los recursos del computador. Mac OS es un sistema operativo multitarea. Por otro lado, un sistema operativo multiusuario es aquel que puede atender simultáneamente a múltiples usuarios. Mac OS no es un sistema operativo multiusuario sino monousuario. Por tanto, la afirmación del enunciado es **FALSA**.
- IV) El tiempo de espera se define como la suma de los tiempos que un proceso pasa esperando en las diferentes colas del sistema por la obtención de recursos. Por tanto, la afirmación del enunciado es **FALSA**.

Solución Ejercicio 2.

Datos del enunciado:

Utilizando el algoritmo SJF para la planificación de trabajos.

| PROCESO | T_{LL} (Tiempo Llegada) | T_s (Tiempo de servicio) |
|---------|---------------------------|----------------------------|
| A | 0 | 9 |
| B | 0 | 6 |
| C | 0 | 3 |
| D | 0 | 5 |
| E | 0 | X |

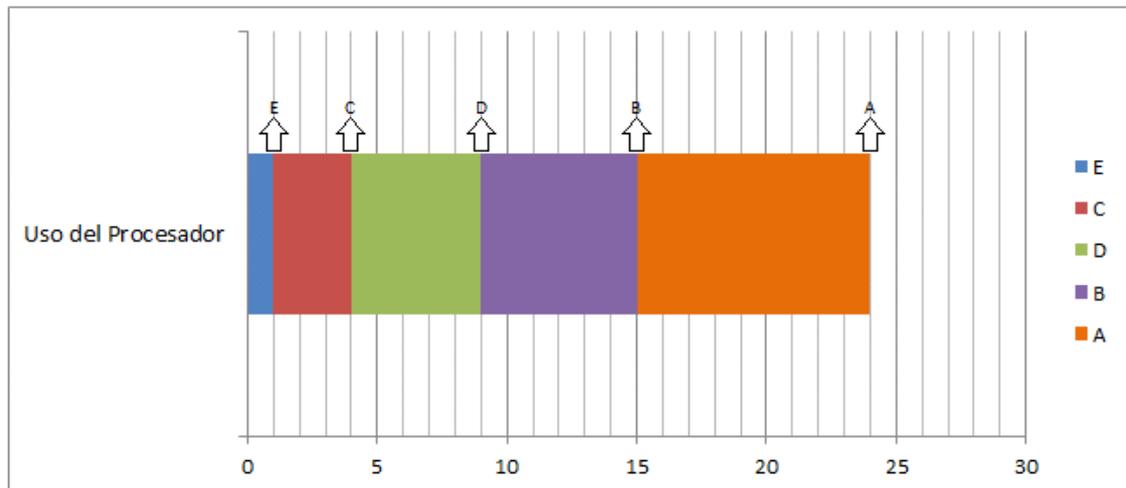
Se pide representar el diagrama de uso del procesador y determinar el tiempo de espera medio en los siguientes casos:

a) $0 < X < 3$

Diagrama de uso del procesador:

Si X debe valer cualquier valor entre 0 y 3 (excluyendo estos), al emplear el algoritmo SJF y viendo la tabla inicial del enunciado podemos garantizar que el proceso E siempre se ejecutará en primer lugar ya que posee el menor tiempo de ejecución.

Para representar el diagrama de uso del procesador se tomará un valor de $X=1$.



Tiempo de espera medio:

El tiempo de espera de un proceso se puede definir mediante la siguiente expresión:

$$T_E = T_F - T_S \quad (\text{Fórmula 1})$$

Donde:

T_E : Tiempo de espera del proceso.

T_F : Tiempo de finalización del proceso.

T_S : Tiempo de servicio del proceso.

Se considerará que el $T_{SE} = X$ y el tiempo medio de espera se dará en función de X también.

Del diagrama podemos obtener los tiempos de finalización de cada proceso, estos son:

$$T_{FE} = X$$

$$T_{FC} = X + T_{SC}$$

$$T_{FD} = X + T_{SC} + T_{SD}$$

$$T_{FB} = X + T_{SC} + T_{SD} + T_{SB}$$

$$T_{FA} = X + T_{SC} + T_{SD} + T_{SB} + T_{SA}$$

Aplicando los datos obtenidos anteriormente en la fórmula 1 obtenemos el tiempo de espera de cada uno de los procesos:

$$T_{EE} = X - X = 0$$

$$T_{EC} = X + T_{SC} - T_{SC} = X$$

$$T_{ED} = X + T_{SC} + T_{SD} - T_{SD} = X + T_{SC}$$

$$T_{EB} = X + T_{SC} + T_{SD} + T_{SB} - T_{SB} = X + T_{SC} + T_{SD}$$

$$T_{EA} = X + T_{SC} + T_{SD} + T_{SB} + T_{SA} - T_{SA} = X + T_{SC} + T_{SD} + T_{SB}$$

Ahora hacemos el sumatorio de todos los tiempos de espera:

$$\sum T_E = T_{EE} + T_{EC} + T_{ED} + T_{EB} + T_{EA}$$

$$\sum T_E = 0 + X + X + T_{SC} + X + T_{SC} + T_{SD} + X + T_{SC} + T_{SD} + T_{SB}$$

$$\sum T_E = 4X + 3T_{SC} + 2T_{SD} + T_{SB}$$

Para calcular el tiempo de espera medio dividimos el $\sum T_E$ entre el número total de procesos:

$$T_{E\text{medio}} = \sum T_E / 5$$

Sustituyendo los valores en la formula obtenemos:

$$T_{E\text{medio}} = (4X + 3 \cdot 3 + 2 \cdot 5 + 6) / 5$$

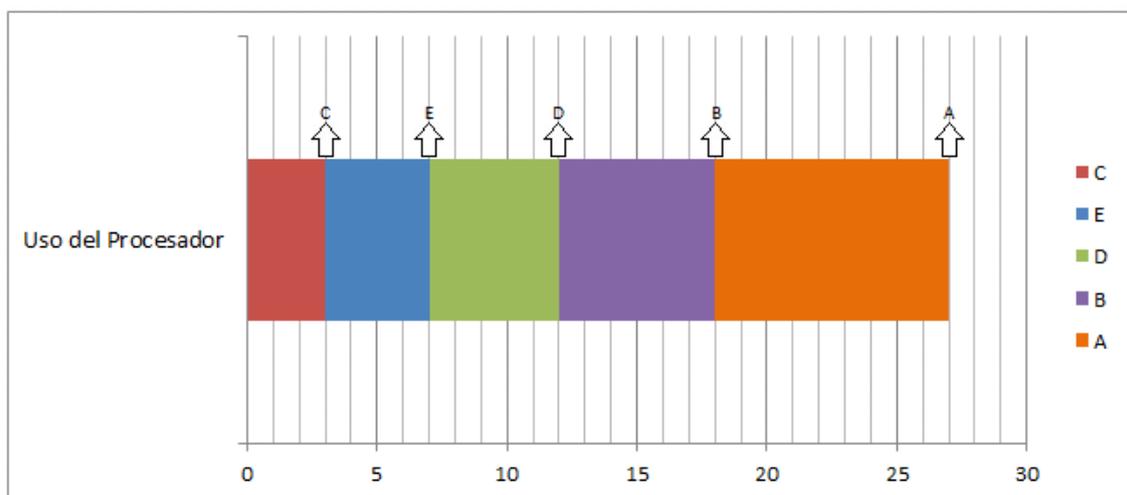
$$T_{E\text{medio}} = \frac{4}{5}X + 5 \text{ ut}$$

b) $3 < X < 5$

Diagrama de uso del procesador:

Si X debe valer cualquier valor entre 3 y 5 (excluyendo estos), al emplear el algoritmo SJF y viendo la tabla inicial del enunciado podemos garantizar que el proceso E siempre se ejecutará en segundo lugar ya que posee el segundo menor tiempo de ejecución.

Para representar el diagrama de uso del procesador se tomará un valor de $X=4$.



Tiempo de espera medio:

Del desarrollo del apartado anterior podemos fijarnos en que el $\sum TE$ de los procesos se puede definir como:

$$\sum TE = 4*(T_{S1^{\text{º}} \text{ proceso ejecutado}}) + 3*(T_{S2^{\text{º}} \text{ proceso ejecutado}}) + 2*(T_{S3^{\text{º}} \text{ proceso ejecutado}}) + T_{S4^{\text{º}} \text{ proceso ejecutado}}$$

Sustituyendo en la expresión anterior obtenemos:

$$\sum TE = 4T_{SC} + 3X + 2T_{SD} + T_{SB}$$

Para calcular el tiempo de espera medio dividimos el $\sum T_E$ entre el número total de procesos:

$$T_{E_{\text{medio}}} = \sum T_E / 5$$

Sustituyendo los valores en la formula obtenemos:

$$T_{E_{\text{medio}}} = (4*3 + 3X + 2*5 + 6)/5$$

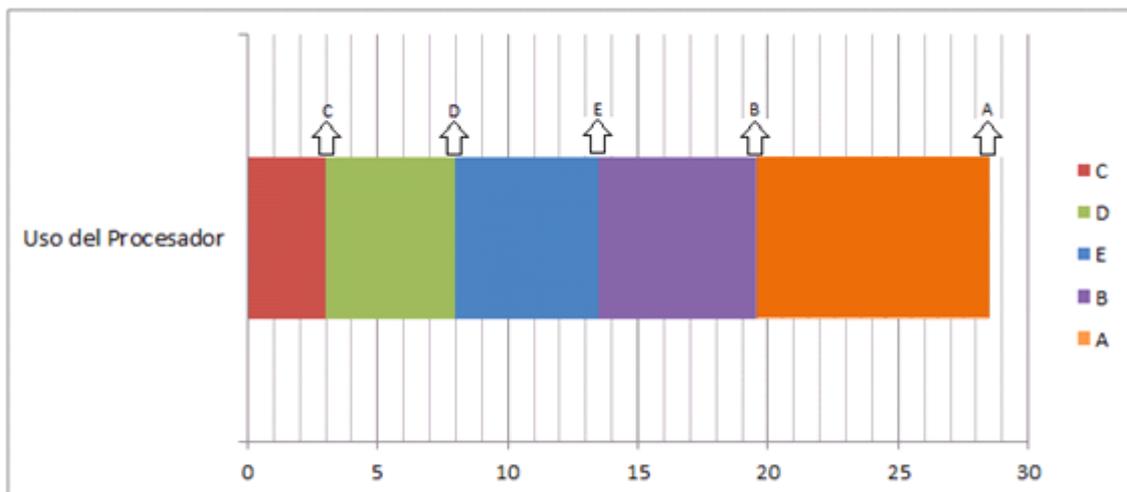
$$T_{E_{\text{medio}}} = \frac{28+3X}{5} \text{ ut}$$

$$c) \quad 5 < X < 6$$

Diagrama de uso del procesador:

Si X debe valer cualquier valor entre 5 y 6 (excluyendo estos), al emplear el algoritmo SJF y viendo la tabla inicial del enunciado podemos garantizar que el proceso E siempre se ejecutará en tercer lugar ya que posee el tercer menor tiempo de ejecución.

Para representar el diagrama de uso del procesador se tomará un valor de X=5,5.



Tiempo de espera medio:

Tal y como se vio en el apartado anterior, el ΣTE de los procesos se define como:

$$\Sigma TE = 4*(T_{S1^{\text{º}} \text{ proceso ejecutado}}) + 3*(T_{S2^{\text{º}} \text{ proceso ejecutado}}) + 2*(T_{S3^{\text{º}} \text{ proceso ejecutado}}) + T_{S4^{\text{º}} \text{ proceso ejecutado}}$$

Sustituyendo en la expresión anterior obtenemos:

$$\Sigma TE = 4T_{SC} + 3T_{SD} + 2X + T_{SB}$$

Para calcular el tiempo de espera medio dividimos el ΣT_E entre el número total de procesos:

$$T_{E_{\text{medio}}} = \Sigma T_E / 5$$

Sustituyendo los valores en la formula obtenemos:

$$T_{E_{\text{medio}}} = (4*3 + 3*5 + 2X + 6)/5$$

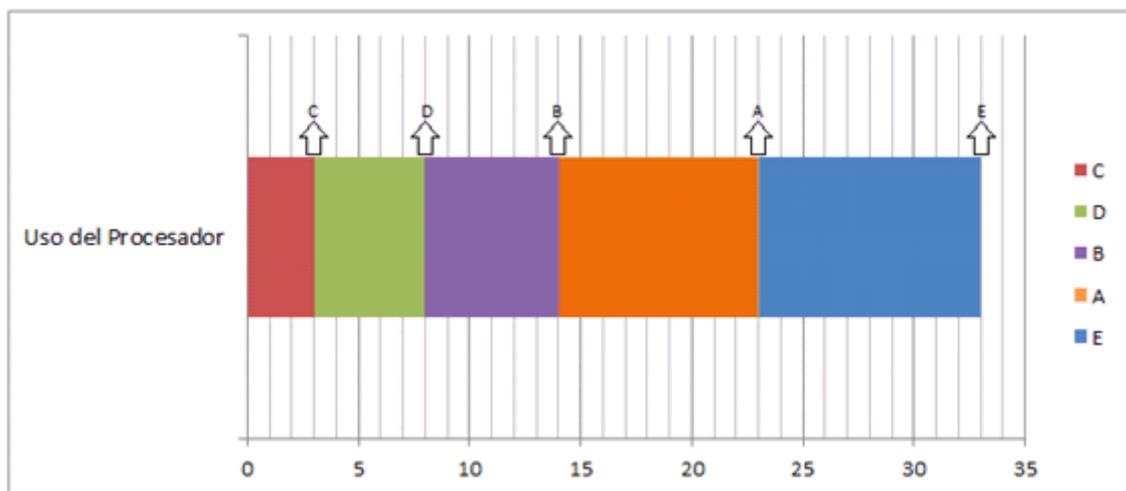
$$T_{E_{\text{medio}}} = \frac{33+2X}{5} \text{ ut}$$

d) $9 < X$

Diagrama de uso del procesador:

Si X debe valer cualquier valor mayor que 9 (excluyendo este), al emplear el algoritmo SJF y viendo la tabla inicial del enunciado podemos garantizar que el proceso E siempre se ejecutará en último lugar ya que posee el mayor tiempo de ejecución.

Para representar el diagrama de uso del procesador se tomará un valor de $X=10$.



Tiempo de espera medio:

El ΣTE de los procesos se define como:

$$\Sigma TE = 4*(T_{S1^{\text{º}} \text{ proceso ejecutado}}) + 3*(T_{S2^{\text{º}} \text{ proceso ejecutado}}) + 2*(T_{S3^{\text{º}} \text{ proceso ejecutado}}) + T_{S4^{\text{º}} \text{ proceso ejecutado}}$$

Sustituyendo en la expresión anterior obtenemos:

$$\Sigma TE = 4T_{SC} + 3T_{SD} + 2T_{SB} + T_{SA}$$

Para calcular el tiempo de espera medio dividimos el ΣT_E entre el número total de procesos:

$$T_{E\text{medio}} = \Sigma T_E / 5$$

Sustituyendo los valores en la formula obtenemos:

$$T_{E\text{medio}} = (4*3 + 3*5 + 2*6 + 9)/5$$

$$T_{E\text{medio}} = 9,6 \text{ ut}$$

Solución Ejercicio 3.

La solución que se propone a continuación para este problema utiliza las siguientes variables globales y semáforos binarios.

- *contador*: Variable global de tipo entero para llevar la cuenta del número de aviones que desean aterrizar.
- *S1*: Semáforo binario que se utiliza para garantizar la exclusión mutua en el uso de la variable global *contador*.
- *S2*: Semáforo binario que se utiliza para garantizar la exclusión mutua en el uso de la pista del aeropuerto.
- *S3*: Semáforo binario que se utiliza para sincronizar los procesos de aterrizaje y despegue dando mayor prioridad a los aterrizajes que a los despegues.

Pseudocódigo:

```
/*Definición variables y semáforos*/
int contador=0;
semáforo_binario S1, S2, S3;

void avion_aterriza()
{
    wait_sem(S3); /*Bloquea los despegues*/

    wait_sem(S1);
    contador = contador + 1;
    signal_sem(S1);

    wait_sem(S2); /*Ocupa la pista para aterrizar*/

    aterriza();

    signal_sem(S2); /*Libera la pista*/

    wait_sem(S1);
    contador = contador - 1;
    if (contador == 0)
    {
        signal_sem(S3); /*Ya no quedan más aterrizajes, se puede
despegar*/
    }
    signal_sem(S1);
}

void avion_despega()
{
    wait_sem(S3); /*Comprueba si se puede despegar*/

    wait_sem(S2); /*Ocupa la pista para despegar*/

    despega();

    signal_sem(S2); /*Libera la pista*/
}

void main() /*Inizialización de semáforos y ejecución concuerrente*/
{
    init_sem(S1,1);
    init_sem(S2,1);
    init_sem(S3,1);
    ejecución_concurrente(avion_aterriza, avion_aterriza, ...,
avion_despega, avion_despega,...);
}
```