

SISTEMAS OPERATIVOS – PRÁCTICAS MODULO I – SESION 3

REFLEXION 1. Sean estos tres script bash de nombres s1, s2 y s3:

s1	s2	s3
<pre>lim=1000000 for ((C=1;C<lim;C++));do ## se ejecuta el script ## llamado calculo1 que realiza un ## calculo aritmetico de 1 seg calculo1 ## done echo Fin s1 con pid \$\$</pre>	<pre>lim=1000000 for ((C=1;C<lim;C++));do sleep 1 done echo Fin s2 con pid \$\$</pre>	<pre>lim=1000000 sleep \$lim echo Fin s3 con pid \$\$</pre>

Califique los siguientes enunciados como Verdaderos o Falsos: (Soluciones al final)

1.	El proceso resultante de ejecutar s1 es un proceso “limitado por CPU”
2.	El proceso resultante de ejecutar s2 es un proceso “limitado por E/S”
3.	El proceso resultante de ejecutar s3 es un proceso “limitado por E/S”
4.	Si como únicos procesos tenemos muchas ejecuciones simultáneas de s1 , la cola de ejecutables estará vacía la mayor parte del tiempo.
5.	Si como únicos procesos tenemos muchas ejecuciones simultáneas de s3 , la cola de ejecutables estará vacía la mayor parte del tiempo.
6.	Si como únicos procesos tenemos muchas ejecuciones simultáneas de s3 con distintas prioridades , el hecho de que tengan distintas prioridades no va a repercutir apreciablemente en cómo van progresando en su ejecución.
7.	Si como únicos procesos tenemos muchas ejecuciones simultáneas de s1 con distintas prioridades , el hecho de que tengan distintas prioridades no va a repercutir apreciablemente en cómo van progresando en su ejecución.

REFLEXION 2. Observemos estas dos posibilidades de ejecutar la orden `time` sobre el script prueba pasándole como argumento el valor 10000 (supongamos que `.` no está en `$PATH`); explique porqué una da error y otra no:

- `time prueba 10000 &`
- `time bash prueba 10000 &`

REFLEXION 3. Sobre la orden `nice -n <ruta>`

Ejecuta `<ruta>` cambiando la prioridad que por defecto tienen los procesos que crea este usuario: le suma el valor `n`

En el valor de prioridad de un proceso, menor valor implica mayor importancia

Si `n` es positivo (ej: `nice -10 /home/ruz/e1`) se está creando un proceso con **menor** importancia relativa que los que este usuario crea por defecto.

Si `n` es negativo (ej: `nice --10 /home/ruz/e1`) se está creando un proceso con **mayor** importancia relativa que los que este usuario crea por defecto.

REFLEXION 4. Sobre las “marcas de tiempo”: En Linux existen tres tiempos almacenados en un inodo:

- **ctime:** 'change time' indica el tiempo de la última modificación que se realizó a los metadatos del archivo. Es posible ver este tiempo con la opción *c* de la orden *ls*.
- **atime:** 'access time' indica el tiempo en que el contenido del archivo fue por última accedido para mostrar su contenido. Es posible ver este tiempo con la opción *u* de la orden *ls*.
- **mtime:** 'modify time' indica el tiempo en que el contenido del archivo fue por última vez modificado. Es posible ver este tiempo con la opción *l* de la orden *ls*.

REFLEXION 5. Sobre la orden siguiente (el carácter | expresa opcionalidad entre lo que se expresa a la izquierda y a la derecha):

```
find ruta -ctime|-mtime|-atime +|- n
```

Si se ha especificado +:

Selecciona los archivos descendientes de ruta que tienen un valor de tiempo ctime o mtime o atime (según se haya especificado) **inferior** a (fecha actual - n días)

Si se ha especificado -:

Selecciona los archivos descendientes de ruta que tienen un valor de tiempo ctime o mtime o atime (según se haya especificado) **superior** a (fecha actual - n días)

Solución Reflexión 1:

1. V

2. V La orden *sleep* provoca el bloqueo del proceso, por lo que éste tiene ráfagas cortas.

3. V

4. F Un proceso resultante de ejecutar *sl* no se bloquea nunca, por tanto siempre que no se esté ejecutando estará en estado ejecutable

5. V

6. V

7. F