

TEORIA. TEMAS 1 Y 2

Tipo 1

APELLIDOS: _____

NOMBRE: _____

Firma: _____

Responda en la hoja de respuestas asignando **A=VERDADERO; B=FALSO**.

1)	V - Cuando tenemos varios procesos que son ejecuciones simultáneas de un mismo programa, es imprescindible que cada uno de ellos tenga su propia pila.
2)	F - En el PCB o Bloque de Control de Proceso se salva el valor de las variables usadas en el código de cada proceso.
3)	F - Cuando el S.O. detiene un proceso ejecutándose porque le va a asignar la CPU a otro, no es necesario salvar los valores actuales de los registros de la CPU porque ya se han ido salvando adecuadamente durante la ejecución de las instrucciones máquina por las que ha ido pasando la ejecución del proceso.
4)	F - El algoritmo de asignación de CPU FIFO o FCFS (First Come First Served) es válido para entornos interactivos.
5)	F - Los sistemas de tiempo compartido realizan muy pocas veces por segundo la operación de cambio de contexto.
6)	V - El planificador a corto plazo se ejecuta mas veces por unidad de tiempo que el planificador a largo plazo.
7)	F - En el algoritmo de Barrido Cíclico con quantum Q: si el proceso actual se bloquea antes de finalizar el quantum entonces es imposible que en este momento se asigne la CPU a otro proceso en este momento, debiendo quedar ociosa hasta que acabe el quantum para que pueda pasar a ejecutarse otro proceso.
8)	F - Solo puede haber 1 planificador a medio plazo a lo sumo.
9)	V - Los procesos “limitados por E/S” dedican más tiempo a realizar E/S que a cómputo.
10)	V - Siempre que el proceso actual no desea seguir ejecutándose ha de llamarse al planificador a corto plazo.
11)	F - En la política de planificación FCFS (First Come First Served) o FIFO, el tiempo de espera de una ráfaga es mayor para ráfagas largas que para ráfagas cortas.
12)	V - En la política de planificación de Barrido Cíclico con quantum Q, el tiempo de espera de una ráfaga es mayor para ráfagas largas que para ráfagas cortas.
13)	F - En la política de planificación FCFS (First Come First Served) o FIFO, la penalización de una ráfaga es mayor para ráfagas largas que para ráfagas cortas.
14)	F - En la política de planificación de Barrido Cíclico con quantum Q, la penalización de una ráfaga es mayor para ráfagas largas que para ráfagas cortas.
15)	V - Modificar las posiciones de memoria donde se almacena la Tabla de Vectores de Interrupción solo se puede realizar en modo supervisor.
16)	V - Multiproceso implica multiprogramación.

Suponga que tenemos política de planificación Round-Robin **no apropiativa** (sin desplazamiento o sin derecho preferente) y tenemos los siguientes procesos:

- A: En estado **Ejecutándose** y le queda cierto de tiempo para finalizar el *quantum*;
- B: **Preparado** para ejecutarse;
- C: **Bloqueado** esperando una entrada/salida de disco.

Cuando se produce una interrupción de reloj señalando fin de quantum..... (Responda Verdadero/Falso cada uno de los ítems siguientes)

- 17) F - El proceso C pasa a **Preparado**
- 18) F - El proceso A sigue en estado **Ejecutándose**
- 19) V - El proceso B pasa a **Ejecutándose**.

Si en lugar de producirse fin de quantum se produce el fin de la operación de E/S que esperaba C..... (partimos de la situación inicial que describe el enunciado).....

(Responda Verdadero/Falso cada uno de los ítems siguientes)

- 20) F - El proceso C pasa a **Ejecutándose**
- 21) F - El proceso B pasa a **Ejecutándose**.
- 22) V - El proceso A no cambia su estado (es el que está **Ejecutándose**).

Suponga que tenemos política de planificación Round-Robin **no apropiativa** (sin desplazamiento o sin derecho preferente) y tenemos los siguientes procesos:

- A: En estado **Ejecutándose** y le queda cierto de tiempo para finalizar el *quantum*;
- B: **Preparado** para ejecutarse;
- C: **Bloqueado** esperando una entrada/salida de disco.

Ahora tenemos prioridades:

B es el menos prioritario, después está A, y C es el más prioritario.

Cuando se produce una interrupción de reloj señalando fin de quantum..... (Responda Verdadero/Falso cada uno de los ítems siguientes)

- 23) F - El proceso C pasa a **Preparado**
- 24) V - El proceso A sigue en estado **Ejecutándose**
- 25) F - El proceso B pasa a **Ejecutándose**.

Si en lugar de producirse fin de quantum se produce el fin de la operación de E/S que esperaba C..... (partimos de la situación inicial que describe el enunciado).....

(Responda Verdadero/Falso cada uno de los ítems siguientes)

- 26) F - El proceso C pasa a **Ejecutándose**
- 27) F - El proceso B pasa a **Ejecutándose**.
- 28) V - El proceso A no cambia su estado (es el que está **Ejecutándose**)

29)	F - En un algoritmo de “Colas múltiples con realimentación” si se decide que cuando un proceso se desbloquea prosiga en la misma cola en que estaba antes de bloquearse, es porque suponemos que en nuestro sistema los procesos tienen sus ráfagas de duración parecida.
30)	V - La política de planificación “El más corto primero” trata mejor a las ráfagas cortas si hay apropiatividad (derecho preferente o desplazamiento).que si no la hay.
31)	F - Un compilador puede ejecutar todo el repertorio de instrucciones de la CPU incluyendo las instrucciones privilegiadas.
32)	V - En ausencia total del S.O. cada programa deberá incluir todo el código de uso de los dispositivos que utilice.
33)	V - La estructura de S.O. llamada “Cliente-Servidor” tiene más <i>tolerancia a fallos</i> que la de “Capas”.
34)	F - La estructura de S.O. llamada “Cliente-Servidor” presenta en menor medida la característica de <i>extensibilidad</i> que la de “Capas”.
35)	F - En la estructura de S.O. “Cliente-Servidor” un proceso Servidor no puede hacer peticiones convirtiéndose en un cliente de otro proceso Servidor.
36)	F - Cuando un controlador produce una interrupción se produce necesariamente un cambio de contexto.
37)	F - Si tenemos una política de planificación no apropiativa (sin desplazamiento), no se puede manejar una interrupción mientras se está ejecutando un proceso.
38)	V - Las distintas hebras de un proceso han de tener cada una su propia pila.
39)	V - Las distintas hebras de un proceso se pueden comunicar entre sí mediante memoria compartida.
40)	F - La operación de cambiar la asignación de CPU de una hebra a otra (dentro de un mismo proceso) es más corta si se trata de “hebras Nucleo” que si se trata de “hebras de usuario”.
41)	V - En el caso de “hebras de usuario”: La operación de cambiar la asignación de CPU de una hebra a otra es más corta que la de cambiar la asignación de CPU de un proceso a otro.
42)	V - Una ventaja de las “hebras de usuario” es que se pueden ejecutar en cualquier S.O. mejorando la transportabilidad del software.
43)	F - En el caso de “hebras Núcleo”, si una hebra queda bloqueada ello implica que todas las demás del mismo proceso también queden bloqueadas.
44)	V - En el caso de “hebras de usuario”: La existencia de hebras dentro de un proceso no es visible desde fuera del proceso.
45)	V - Un sistema que permita el acceso interactivo simultáneo de varios usuarios es siempre multiprogramado.
46)	F - Cada Rutina de Tratamiento de Interrupcion está representada por un descriptor de proceso (PCB).
47)	F - Cuando se dice que el S.O. gestiona los recursos entre los procesos....con el término “recurso” nos referimos únicamente a elementos hardware.
48)	V - Leer de disco a nivel de pista/sector... debe ser siempre una instrucción privilegiada.

NOMBRE: _____

PRUEBA SOBRE IMPLEMENTACION DE PROCESOS EN UNIX.

* Todas las respuestas deben tener su justificación para que sean puntuadas.

1. ¿Un proceso que está bloqueado porque espera el fin de una E/s puede intercambiarse a disco quedando privado de MP? Respuesta SI/NO:_____
2. (Viendo el diagrama de estados) Sea un proceso en estado 3, cuando es elegido para ejecutarse, ¿porqué pasa al estado 2 y no al 1?
3. Un proceso que está bloqueado e intercambiado en disco, ¿puede desbloquearse (si ocurre el evento que esperaba) o ha de obtener memoria previamente?
Respuesta : _____

4. Un proceso que está bloqueado e intercambiado en disco, ¿puede llevarse a MP o ha de desbloquearse previamente (es decir esperar primero a que se produzca el evento que esperaba)?

Respuesta : _____

5. ¿Del diagrama de estados se desprende que no se usa el concepto de Quantum?

Respuesta SI/NO: _____

6. Indique qué transiciones del diagrama de estados suponen añadir una nueva capa de contexto:

7. Idem eliminar una nueva capa de contexto:

8. Al comienzo del tema de procesos se presentaba una descripción simplificada de los estados posibles de un proceso en que se usaban únicamente los estados Bloqueado, Ejecutándose y Preparado. Situándonos en los estados aludidos en el diagrama de estados en Unix, **indique qué estados corresponden al estado Bloqueado** del esquema simplificado:

9. Idem para el **estado Ejecutándose**:

10. Idem para el **estado Preparado**:
11. Ponga un ejemplo en que dos procesos puedan compartir la región de código.
12. Si dos procesos A y B comparten una misma región, ¿pueden tener permisos distintos sobre esa región? Respuesta SI/NO:_____
13. Si dos procesos A y B comparten una misma región, ¿esta región ha de estar ubicada en el mismo rango de direcciones en los dos procesos? Respuesta SI/NO:_____
14. Cuando un proceso termina y se va liberando toda la memoria ocupada por él, ¿puede haber algún caso en que una región no se elimine de memoria? Respuesta SI/NO:_____
- En caso afirmativo explique cuál:

15. Si un proceso A crea mediante fork un proceso B, ¿hereda B como terminal de inicio el que tenía A? Respuesta SI/NO:_____

Justifíquelo en base al tratamiento realizado por fork con las estructuras de datos que Unix guarda sobre los procesos.

16. Si un proceso A crea mediante fork un proceso B, ¿hereda B la Tabla de Archivos Abiertos que tenía A? Respuesta SI/NO:_____

Justifíquelo en base al tratamiento realizado por fork con las estructuras de datos que Unix guarda sobre los procesos.

17. Explique algunos casos en que una región esté en estado "Bloqueada".

Caso 1:

Caso 2:

Caso 3:

18. Uno de los datos de la Tabla de Regiones es un "Contador de procesos que referencian la región", ¿para qué sirve?

APELLIDOS: _____

NOMBRE: _____

SISTEMAS OPERATIVOS. GRUPO D

PRUEBA SOBRE IMPLEMENTACION DE ARCHIVOS EN UNIX.

* Todas las preguntas siguientes se responderán en el contexto del contenido del archivo

SOGrupoDTeoriaTema4-Parte2

* Todas las respuestas deben tener su justificación para que sean puntuadas.

CRITERIOS DE PUNTUACION:

PREGUNTA 1: 2 puntos

PREGUNTA 2: 2 puntos

PREGUNTA 3: 2 puntos

PREGUNTA 4: 5 puntos

PREGUNTA 5: 5 puntos

PREGUNTA 6: ANULADA

PREGUNTA 7. 3 puntos

TOTAL DE PUNTOS: 19 (OJO: 4 puntos son de regalo, he considerado que el 100% son 15 puntos)

1. Cuando se realiza la orden **ls /usr/ast/work/f1** se necesitan varios accesos a disco. Calcule el número de accesos a disco requeridos (como máximo) bajo la suposición de que el i-nodo raíz ya se encuentra en memoria y que todos los directorios necesitan como máximo 1 bloque para almacenar los datos de sus archivos.

Respuesta: Número de accesos a disco requeridos (como máximo):

Justificación:

ESTÁ RESPONDIDA EN LAS DIAPOSITIVAS CUANDO SE HABLA DEL ALGORITMO NAMEI

2. Estamos situados en un determinado directorio como directorio de trabajo actual, ya tenemos su inodo en memoria debido a operaciones anteriores.

a) ahora hacemos **ls**. ¿Cuántos accesos a disco se necesitan para realizar esta orden?

Respuesta:

DEPENDE DEL NUMERO DE HIJOS DEL DIRECTORIO ACTUAL. ALMENOS UNO,

Justificación:

UN ACCESO PARA LEER EL BLOQUE ASOCIADO AL DIRECTORIO, AHÍ ESTÁN LOS NOMBRES DE SUS HIJOS, SI HAY MÁS BLOQUES PARA ALBERGAR MAS HIJOS SERÁN NECESARIAS MÁS LECTURAS DE DISCO.

b) ahora hacemos **ls -l**. ¿Cuántos accesos a disco se necesitan para realizar esta orden?

Respuesta:

LOS QUE HAGAN FALTA PARA EL APARTADO (A) MAS UNO POR CADA HIJO

Justificación:

ADEMAS, AHORA POR CADA HIJO HAY QUE LEER SU INODO PARA VER SUS METADATOS Y MOSTRARLOS.

3. Nos situamos en el programa de la Figura 7.4. del libro de Bach. El inodo asociado a argv[1], ¿está bloqueado durante toda la ejecución del programa? Respuesta : (Si/No): NO

Justificación:

LA OPERACION OPEN LO DESBLOQUEA AL FINAL

4. Nos situamos en el programa de la Figura 7.4. del libro de Bach.

a) Tras la ejecución de fork, represente el contenido de las Tabla de Descriptores de Archivos de padre y del hijo, y el contenido de la Tabla de Archivos.

ANTES DEL FORK EL PADRE TIENE.....

* UNA FILA EN LA TABLA DE DESCRIPTORES DE ARCHIVOS (TAMBIEN LLAMADA TABLA DE ARCHIVOS ABIERTOS) PARA ARGV[1] QUE APUNTA A LA FILA EN CUESTION DE LA TABLA DE ARCHIVOS (ALLI SE GUARDA EL OFFSET).

* LO MISMO PARA ARGV[2]

TRASEL FORK EL HIJO HEREDA ESOS VALORES, ES DECIR SU TABLA DE DESCRIPTORES DE ARCHIVOS CONTIENE LOS MISMOS VALORES QUE LOS DEL PADRE, DESCRITOS MAS ARRIBA.

ES DECIR, EN LA TABLA DE ARCHIVOS LA FILA CORRESPONDIENTE A LA APERTURA DE ARGV[1] ES APUNTADA DESDE LAS DOS TABLAS DE DESCRIPTORES DEL PADRE Y DEL HIJO, Y LO MISMO OCURRE CON LA APERTURA DE ARGV[2]

HABRIA QUE REPRESENTAR GRAFICAMENTE LAS TRES TABLAS ALUDIDAS, REPRESENTANDO LO DESCRITO ANTES

b)¿Se genera una copia del archivo original o dos copias? Justifique su respuesta

ES IMPRESCINDIBLE VER QUE SE EJECUTA DOS VECES EL CÓDIGO DE LA FUNCION rdwrt.

Una vez lo ejecuta el padre, y otra vez lo ejecut el hijo. Pero tras la ejecución del programa en el archivo argv[2] no hay dos copias del contenido de argv[1], sino una sola copia. Ello es debido a que ambas aperturas para leer del original comparten el offset (pues ambas aperturas comparten la misma fila de la Tabla de Archivos que es donde se guarda el offset).

c)¿El resultado es siempre el mismo o depende de la velocidad relativa de ejecución de los procesos existentes? Justifique su respuesta.

TRAS LA EJECUCIÓN DE PADRE E HIJO, el orden de los caracteres en argv[1] puede variar según la secuencia de ejecución que haya tenido el programa

5. Nos situamos en el mismo programa anterior pero ahora la operación de apertura del archivo argv[1] la cambiamos de lugar llevándola justo tras fork.

a) Tras la ejecución de fork y de la operación open que acabamos de traer aquí, represente el contenido de las Tabla de Descriptores de Archivos de padre y del hijo, y el contenido de la Tabla de Archivos.

AHORA LA APERTURA DEL ARCHIVO ORIGINAL SE HACE TRAS FORK, ES DECIR CADA PROCESO HACE LA SUYA, ES DECIR SON APERTURAS DIFERENTES QUE CONLLEVAN FILAS DIFERENTES DE LA TABLA DE ARCHIVOS.

RESPECTO AL ARCHIVO ARGV[1] LOCURRE LO MISMO QUE EN LA PREGUNTA 4.

HABRIA QUE REPRESENTAR LAS MISMAS TABLAS QUE EN 4.A CON ESTA CIRCUNSTANCIA DISTINTA

b)¿Se genera una copia del archivo original o dos copias? Justifique su respuesta

AHORA SE GENERAN DOS COPIAS DEL ARCHIVO ORIGINAL EN ARGV[1], ELLO ES DEBIDO A QUE NO COMPARTEN EL OFFSET Y POR TANTO LAS SUCEVAS LECTURAS DE UNO Y OTRO PROCESO RECORREN DOS VECES EL ARCHIVO ORIGINAL

c)¿El resultado es siempre el mismo o depende de la velocidad relativa de ejecución de los procesos existentes? Justifique su respuesta.

IGUAL QUE EN 4.C EL ORDEN DE LOS CARACTERES EN ARGV[2] DEPENDE DE LA SECUENCIA DE ENTREMEZCLAMIENTO DE LOS PROCESOS.

6. Suponiendo una ejecución correcta de las siguientes órdenes en el sistema operativo Unix:

```
/home/jgarcia/prog > ls -li      (* lista los archivos y sus números de i-nodos del directorio prog*)
18020      fich1.c
18071      fich2.c
18001      pract1.c
```

```
/home/jgarcia/prog > cd ../tmp
/home/jgarcia/tmp > ln -s ../prog/pract1.c p1.c      (* crea un enlace simbólico *)
/home/jgarcia/tmp > ln ../prog/pract1.c p2.c        (* crea un enlace absoluto o duro*)
```

represente gráficamente cómo y dónde quedaría reflejada y almacenada **toda** la información referente a la creación anterior de un enlace simbólico y absoluto ("hard") a un mismo archivo, **pract1.c**.

REPRESENTACIÓN DEL ENLACE DURO. HABRIA QUE REPRESENTAR EL CONTENIDO DEL DIRECTORIO /home/jgarcia/prog, y el contenido del directorio /home/jgarcia/tmp, donde verse que hay una entrada para p2.c cuyo numero de inodo es el mismo que para pract1.c, es decir, 18001.

REPRESENTACION DEL ENLACE SIMBOLICO. LO MISMO, PERO AHORA LA ENTRADA DE PRACT1 EN SU DIRECTORIO CONTIENE LA RUTA DE P1.C

7. Supongamos que los procesos A, B y C realizan operaciones open sobre el mismo archivo F1, ¿comparten el modo de apertura del archivo, es decir, han de tener todas esas operaciones open el mismo modo de apertura? Respuesta : (Si/No): NO

Justificación en base a las estructuras de datos utilizadas por el núcleo:

DISTINTAS APERTURAS SE REPRESENTAN MEDIANTE DISTINTAS FILAS DE LA TABLA DE ARCHIVOS, CADA FILA CONTIENE EL MODO DE APERTURA (POR TANTO PUEDE SER DISTINTO PARA DISTINTAS APERTURAS, CADA UNA TIENE ALMACENADO EL SUYO)