



1. Los tres procesos (padre y dos hijos) involucrados en el apartado A de la práctica de Sistemas Operativos:
 - a. No están sincronizados y, simplemente, se ejecutan concurrentemente.
 - b. Se sincronizan utilizando como mecanismo de sincronización el paso de mensajes.
 - c. Se sincronizan utilizando el envío de señales y las llamadas al sistema `wait()` y `pause()`.
 - d. Se sincronizan utilizando las llamadas al sistema, `fork()` y `execvp()`.
2. Suponiendo correcta la ejecución de la llamada al sistema `fork()`, ésta:
 - a. Devuelve 0 al proceso padre y un valor distinto de 0 al proceso hijo, que coincide con el PPID (id. del padre).
 - b. Devuelve 0 al proceso hijo y un 1 al proceso padre, ya que esto es suficiente para distinguir los dos procesos.
 - c. Devuelve 0 al proceso hijo y un valor distinto de 0 al proceso padre, que coincide con el PID (id. del hijo).
 - d. No devuelve ningún valor.
3. La versión de la llamada al sistema `execvp()` permite:
 - a. Crear un nuevo proceso, que ejecuta el código que se indica en la invocación.
 - b. Ejecutar, en el contexto de un proceso ya existente, el código de un programa ejecutable y, al mismo tiempo, pasar los argumentos de la línea de órdenes en una lista de cadenas (terminada por `NULL`).
 - c. Ejecutar, en el contexto de un proceso ya existente, el código de un programa ejecutable y, al mismo tiempo, pasar los argumentos de la línea de órdenes en un vector, que se pasa como parámetro en la invocación.
 - d. Crear un nuevo proceso y, al mismo tiempo, pasar los argumentos de la línea de órdenes a través de una lista de cadenas terminada por `NULL`, que se pasan como parámetros en la invocación de la función.
4. La llamada al sistema `kill()` permite:
 - a. Establecer el manejador para una señal.
 - b. Enviar la señal especificada en la invocación, al proceso cuyo PID se indica.
 - c. Bloquear un proceso hasta la recepción de una señal.
 - d. Recibir una señal.
5. La macro `WIFEXITED`(estado) se utiliza para
 - a. Evaluar el valor devuelto por `wait()` y determinar si el proceso hijo que termina, lo hizo normalmente.
 - b. Evaluar el valor devuelto por `fork()` y determinar si el proceso hijo se ha creado correctamente.
 - c. Evaluar el valor devuelto por `wait()` y determinar el valor devuelto por el hijo cuando finaliza con `exit()`.
 - d. Evaluar el valor devuelto por `wait()` y determinar si el hijo terminó por una señal no capturada.
6. En linux, la forma de generar un programa multihilo es:
 - a. Incluir sólo el archivo de cabecera `<pthread.h>` en el programa fuente.
 - b. Incluir sólo el archivo de cabecera `<thread.h>` en el programa fuente.
 - c. Incluir el archivo de cabecera `<pthread.h>` en el programa fuente y enlazar con la biblioteca `pthread`.
 - d. Incluir el archivo de cabecera `<thread.h>` en el programa fuente y enlazar con la biblioteca `thread`.
7. La forma correcta de utilizar semáforos POSIX en un programa multihilo para conseguir exclusión mutua es:
 - a. El hilo principal deberá inicializar a 0 el semáforo, y posteriormente cada hilo debe hacer una operación `sem_wait(&sem)` antes de entrar en su sección crítica (SC) y una operación `sem_post(&sem)` después de salir de la SC.
 - b. El hilo principal deberá inicializar a 1 el semáforo, y posteriormente cada hilo debe hacer una operación `sem_wait(&sem)` antes de entrar en su SC y una operación `sem_post(&sem)` después de salir de la SC.
 - c. El hilo principal deberá inicializar a 0 el semáforo, y posteriormente cada hilo debe hacer una operación `sem_post(&sem)` antes de entrar en su SC y una operación `sem_wait(&sem)` después de salir de la SC.
 - d. El hilo principal deberá inicializar a 1 el semáforo, y posteriormente cada hilo debe hacer una operación `sem_post(&sem)` antes de entrar en su SC y una operación `sem_wait(&sem)` después de salir de la SC.
8. La función de biblioteca de POSIX Thread disponible para que un hilo espere por otro hilo del que se conoce su identificador de hilo (TID) es:
 - a. `pthread_create()`.
 - b. `pthread_self()`.
 - c. `pthread_join()`.
 - d. `pthread_exit()`.

	1	2	3	4	5	6	7	8
Respuestas								

Correcta: +1.25 pts No Contestada: +0 pts Incorrecta: -0,5 pts.