



AMPLIACIÓN SISTEMAS OPERATIVOS/PRÁCTICA – JUN’2006 (A) Nombre \_\_\_\_\_

1. Cuando un servidor, una vez que acepta una conexión, crea un proceso hijo para atender la nueva conexión establecida con el cliente, mientras que el proceso padre permanece iterando y a la espera de nuevas conexiones, se dice que:
  - a. Se trata de un servidor iterativo (interactivo, en las transparencias) basado en procesos.
  - b. Se trata de un servidor concurrente basado en procesos.
  - c. Se trata de un servidor concurrente basado en hilos.
  - d. Se trata de un servidor iterativo (interactivo, en las transparencias) basado en hilos.
2. La llamada al sistema `accept()`,
  - a. Permite que sobre el socket que se pasa como argumento, se establezca una conexión con el otro extremo.
  - b. Permite registrar una dirección ip y un número de puerto ante el sistema operativo.
  - c. Devuelve un nuevo descriptor de socket sobre el que realmente se establece una conexión con el otro extremo.
  - d. Es utilizada por el cliente para hacer una apertura pasiva del socket y aceptar conexiones de los servidores.
3. La llamada al sistema `bind()`:
  - a. La utiliza el cliente para registrar ante el S.O. la dirección completa del socket (dirección IP junto con el número de puerto utilizado) que utiliza el servidor para aceptar conexiones.
  - b. La utiliza el servidor para registrar ante el S.O. la dirección completa del socket que el cliente utiliza para conectarse con él.
  - c. La utiliza el servidor para establecer el tamaño de la cola de conexiones pendientes.
  - d. La utiliza el servidor para registrar ante el S.O. la dirección completa del socket que el servidor utiliza para aceptar conexiones.
4. En el caso de utilizar sockets UDP o no orientados a conexión, cuál de los siguientes enunciados es verdadero:
  - a. Tanto el cliente como el servidor deben registrar su dirección ip junto con el puerto que utilizan mediante sendas llamadas a la función `bind()`.
  - b. El cliente debe conectarse con la primitiva `connect()` al servidor.
  - c. El cliente y el servidor se intercambian información mediante el uso emparejado de llamadas al sistema `recv()` y `send()`.
  - d. El servidor debe aceptar conexiones de los clientes mediante la invocación a la llamada al sistema `accept()`.
5. Las llamadas al sistema `read()` y `write()` requieren la utilización de un buffer, que se supone dispone de un tamaño prefijado `TAM_MAX`. Suponiendo que se quiere leer de un descriptor de archivo (abierto para lectura) para escribirlo en otro (abierto para escritura), cuál de los siguientes es un uso correcto de la `read()` y `write()`:
  - a. Solicitar que el buffer se cargue con `TAM_MAX` bytes en la lectura y ordenar escribir también `TAM_MAX`.
  - b. Solicitar que el buffer se cargue con `TAM_MAX` bytes, guardando el número de bytes leídos efectivamente, pero ordenando escribir los `TAM_MAX` bytes del buffer.
  - c. Solicitar que el buffer se cargue con la cantidad efectiva de bytes que se leen y ordenar escribir esta misma cantidad.
  - d. Solicitar que el buffer se cargue con `TAM_MAX` bytes, guardar el número de bytes leídos efectivamente, y ordenar escribir tan solo esta cantidad.
6. La llamada al sistema `creat()` es equivalente a la llamada al sistema `open()` con la siguiente combinación de flags:
  - a. `O_WRONLY | O_CREAT`.
  - b. `O_WRONLY | O_CREAT | O_EXCL`.
  - c. `O_WRONLY | O_CREAT | O_TRUNC`.
  - d. `O_WRONLY | O_CREAT | O_TRUNC | O_EXCL`.
7. En la estructura de mensaje propuesta en clase de prácticas para soportar los mensajes intercambiados por el cliente y servidor de ficheros, por qué hace falta incluir la longitud de los datos del contenedor:
  - a. No hace falta, pues el tamaño del contenedor de datos está prefijado y siempre es el mismo.
  - b. Para diferenciar entre las operaciones `-get`, `-put` y `-list`, solicitadas en el enunciado de la práctica.
  - c. Hace falta para indicar cuántos de los bytes que vienen en el contenedor son, efectivamente, parte del mensaje enviado por la otra parte.
  - d. Para indicar el estado de realización de una operación solicitada, por ejemplo, el que la longitud sea cero siempre indica que la operación se ha realizado con éxito.
8. La llamada al sistema `opendir()` se utiliza para:
  - a. Para abrir un directorio y obtener, directamente, la lista con los nombres de los archivos del mismo.
  - b. Para abrir un directorio y obtener un apuntador que se utilizará para, posteriormente, ir leyendo una a una las entradas del directorio.
  - c. Para abrir una dirección ip, y por tanto, conectarse con otro proceso que puede estar ejecutándose en otra máquina.
  - d. Para leer una por una las entradas que componen el directorio que se indica como argumento.

	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>
<b>Respuestas</b>								

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

