

Tipo Test Unidad 1:

1.1 ¿Cuál de los siguientes sistemas operativos no es multitarea?

- a) Unix.
- b) Linux.
- c) Windows 10.
- d) MS-DOS.

1.2 En computación, se entiende por multitarea:

- a) La capacidad que tiene un ordenador de realizar tareas complejas.
- b) La que tiene un ordenador de ejecutar un programa detrás de otro.
- c) La capacidad que tiene un ordenador de realizar tareas sincronizadas entre sí.
- d) La capacidad que tiene un ordenador de ejecutar varios programas al mismo tiempo.

1.3 ¿Qué elemento del ordenador se encarga de gestionar la multitarea?

- a) El lenguaje de programación elegido para crear los programas.
- b) El sistema operativo.
- c) El sistema de archivos.
- d) Ninguna de las respuestas anteriores es correcta.

1.4 ¿Qué afirmación referente a los lenguajes de programación compilados no es correcta?

- a) En condiciones normales son más rápidos que los interpretados.
- b) Requieren una compilación específica para cada sistema operativo.
- c) Si el lenguaje es estándar, la misma compilación sirve para todos los sistemas operativos.
- d) El código fuente debe compilarse para obtener el código binario para ser ejecutado. Tengo muchas dudas.

1.5 El identificador de proceso se suele identificar por las siglas:

- a) CPU.
- b) PID.
- c) IPD.
- d) TTY

1.6 El procesamiento que se ejecuta en diferentes ordenadores independientes, pero conectados y sincronizados se denomina:

- a) Distribuido.
- b) Concurrente.
- c) Paralelo.
- d) Monohilo.

1.7 ¿Cuál de los siguientes no es un objetivo del planificador de procesos del sistema operativo?

- a) Maximizar el rendimiento del sistema,
- b) Maximizar los tiempos de respuesta.
- c) Maximizar la equidad en el reparto de los recursos.
- d) Minimizar los tiempos de espera.

- 1.8. ¿Cuál es el estado al que puede pasar un proceso que está en estado Listo?
- a) Nuevo.
 - b) Bloqueado.
 - c) En ejecución.
 - d) Finalizado.
- 1.9. ¿Qué afirmación referente a la clase Runtime de Java es errónea?
- a) Permite lanzar un proceso indicando los parámetros de ejecución.
 - b) Permite quedar a la espera de la terminación del proceso lanzado.
 - c) Permite conocer el valor de las variables internas del proceso lanzado.
 - d) Permite obtener el estado de la finalización el proceso lanzado.
- 1.10. ¿Cómo se llama el método de la clase Process de Java que hace esperar a que termine el proceso lanzado?
- a) sleep0.
 - b) finish().
 - c) waitFor().
 - d) waitEnd().