Primer Parcial Teórico – Repaso

THREADS

1. Un thread no tiene program counter (PC) propio. Falso

   Cada thread tiene su propia TCB, esta misma contiene propio PC y registros.

2. Un context switch entre threads, no requiere un context switch de registros. Falso

   Cada thread tiene su propia TCB, esta misma contiene propio PC y registros.

3. Un hilo creado por un proceso tendrá su propio contexto. Verdadero

   Cada hilo creado por un proceso tendrá el mismo contexto pesado (ej. espacio de direcciones y recursos) del proceso y a su vez tendrán su propio contexto a nivel de hilo (ej. estado, stacks de usuario y kernel, variables propias)

4. Un hilo creado por un proceso se ejecutará en el espacio de direcciones de este último. Verdadero

   Los hilos de un mismo proceso comparten el espacio de direcciones del proceso.

5. Un hilo es la unidad básica de uso de la CPU. Verdadero

   Unidad básica de utilización de CPU: Hilos

6. Un hilo es la unidad de propiedad de recursos. Falso

   Unidad básica de asignación de RECURSOS: Proceso

7. Dentro de un proceso, un hilo cuenta con un estado de ejecución. Verdadero

8. Existe una PCB por proceso y por cada hilo que él cree. Falso

   Existe una PCB por proceso y una TCB por hilo.

9. Cuando un proceso se swapea, los hilos quedan en memoria en estado de espera. Falso

   Cuando un proceso se swapea, los hilos se quedan suspendidos con su propio estado cada uno. El SO los guarda en el área de swap y, cuando el proceso se vuelve a intercambiar en la memoria, el SO restaura el estado de cada hilo desde el swap, lo que les permite continuar la ejecución desde donde lo dejaron.

10. En la administración de los hilos a nivel de usuario, interviene el kernel. Falso

    Interviene la librería del hilo.

11. En los ULT, cada proceso se encarga de administrar sus hilos. Falso

    La aplicación se encarga de administrar los hilos.

12. La suspensión de un ULT provoca la suspensión del proceso. Falso

    Se pueden reemplazar llamadas al sistema bloqueantes por otras que no bloquean,

13. En los KLT, el context switch entre hilos, no provoca un cambio de modo. Falso

    Se necesitan cambios de modo de ejecución para la gestión.

MULTIPROCESADORES

14. Los KLT, en un ambiente multiprocesador, pueden ejecutarse en distintos procesadores. Verdadero

15. En multiprocesadores, en la organización maestro esclavo, una syscall puede ser atendida en cualquiera de los procesadores. Falso

    Todas las syscalls van a una sola CPU.

16. En multiprocesadores, si cada CPU tiene su SO es posible que una CPU este saturada y otras sin trabajo productivo. Verdadero

    Hay un desbalance de carga.

17. En multiprocesadores, la técnica de SMP no requiere de exclusión mutua para el acceso a las estructuras del kernel. Falso

    Todas las CPUs comparten las estructuras del SO por lo que se debe garantizar su acceso exclusivo.

18. No existen diferencias en la planificación de procesos entre SO monoprocesadores y multiprocesadores. Falso

    Como hay varias CPU la complejidad aumenta en la distribución de tareas, el pasaje de mensajes y el acceso a memoria.

19. En un sistema distribuido, todos los SO de las diferentes computadoras que participan deben ser iguales. Falso

    Pueden ser diferentes SO ya que luego hay un middleware que se encarga de la homogenización.

20. En las multicomputadoras, cada CPU tiene su memoria. Verdadero

21. En multicomputadoras, la comunicación entre procesos se realiza por:
    a) Memoria Compartida
    b) Pasajes de Mensajes
    c) RPC
    d) Ninguna

22. En multicomputadoras, cada nodo puede correr un SO diferente. Verdadero

23. Las computadoras que forman una Grid deben ser todas iguales. Falso

    Pueden tener distinto Hardware.

24. El middleware es una capa de software entre el Hardware y el Sistema Operativo. Falso

    Es una capa de software por encima del SO.

DEADLOCKS

25. Basta que una de las 4 condiciones de deadlock se cumpla, para que haya deadlock. Falso

    Se deben cumplir las 4.

26. La desventaja de usar algoritmos de prevención del deadlock, es que baja el grado de multiprogramación. Verdadero

27. En un esquema de una instancia por tipo de recurso, cuando se encuentra un ciclo en un grafo de alocación de recursos, la asignación de los recursos solicitados:
    a) puede poner al sistema en estado inseguro
    b) pone al sistema en estado inseguro.

28. Todos los estados inseguros son deadlock. Falso

    En estado inseguro hay posibilidad que haya deadlock. No todo estado inseguro es deadlock. Si estoy en estado inseguro y a la larga no se hace nada hay deadlock.

29. El algoritmo del Banquero sirve para sistemas con múltiples instancias de cada recurso. Verdadero

30. Siempre que el grafo de recursos tiene ciclos, hay deadlock. Falso

    Si hay varias instancias por tipo de recurso hay posibilidad de deadlock pero no quiere decir que necesariamente haya.