1. ¿Qué es un sistema operativo?

-Es el conjunto de programas que gestiona los recursos de hardware y provee servicios a los programas de aplicación de software. Además, actúa como intermediario entre los usuarios y el hardware de una computadora.

2. ¿Qué es un sistema operativo en tiempo real?

-Es un sistema operativo especializado que procesa datos y realiza operaciones dentro de límites de tiempo específicamente definidos.

3. Mencione algunos sistemas en tiempo real

-Free RTOS, VxWorks, QNX.

4. Mencione característica de un sistema normal, respecto a un sistema en tiempo real.

Los sistemas normales están diseñados para admitir múltiples tareas y procesos en ejecución simultanea, mientras que en los SOTR se prioriza la consistencia en los tiempos de respuesta.

En los sistemas normales, la gestión de los recursos es general, se comparten entre procesos y aplicaciones de una manera más flexible, mientras que en los SOTR, la gestión de los recursos se asigna a tareas críticas para garantizar su finalización oportuna.

5. ¿Qué es el contexto de un proceso y para qué sirve?

El contexto de un proceso es la unión de su contexto del nivel de usuario, su contexto de registro y su contexto del nivel de sistema. Un proceso se ejecuta en su contexto, o más exactamente, en su capa de contexto actual. El número de capas de contexto está limitado por el número de niveles de interrupción que soporte la máquina.

6. ¿Qué es concurrencia?

La concurrencia es el número máximo de hilos que podría estar ejecutándose simultáneamente, si dispusiera de un número ilimitado de procesadores.

7. ¿Qué es paralelismo?

Es el número real de hilos que se están ejecutando simultáneamente en un momento determinado y por lo tanto está limitado por el número de procesadores.

8. ¿Qué es y para qué sirve?

-El contador de programa, puntero de pila y registrado de estado del proceso (PS).

-Contador de programa:

-Puntero de pila:

-Estado del proceso

9. ¿Qué es un proceso y cuáles son sus estados? (puede realizar un esquema).

Un proceso en unix es una entidad creada tras la llamada fork.

Diagrama, Esquemático

Descripción generada automáticamente

10. ¿Qué es un bloque irreversible y como se resuelve?

Se refiere a una situación en la que un sistema o proceso queda atrapado en un estado del cual no puede salir por sí mismo. Se resuelve, matando el proceso que lo originó.

11. ¿Qué es el Hilo del Núcleo y que es un hilo de usuario?

Los hilos del núcleo no están asociados con ningún proceso de usuario, son creados y destruidos según las necesidades del sistema, mientras que los hilos del usuario son gestionados por la aplicación sin la intervención del núcleo.

12. ¿Qué es un hilo?

Son objetos dinámicos, cada uno de los cuales ejecutan una secuencia de instrucciones y que comparten entre sí los recursos del proceso.

13. Mencione 3 ventajas de usar un hilo o un proceso.

Eficiencia de recursos.

Concurrencia

Cooperación

14. Escriba las líneas de código necesarias para crear un proceso que apague una computadora en 3min.

#include <unistd.h>

#include <stdlib.h>

int main() {

sleep(180);

system("shutdown -h now");

return 0;

}

15. Escriba las líneas correspondientes para crear un hilo

int pthread\_create (pthread\_t ∗thread, const pthread\_attr\_t ∗attr, void ∗(∗start\_routine)(void∗), void ∗arg);

16. ¿Por qué se le denomina región crítica (sección crítica)?

Es una parte del código que accede y modifica recursos compartidos por varios hilos o procesos.

17. ¿Cuál es la funcionalidad de mutex y explique con un ejemplo -código- ?

Es una estructura de datos que se utiliza para lograr la exclusión mutua en programación concurrente.

#include <stdio.h>

#include <pthread.h>

pthread\_mutex\_t mutex;

void\* funcionHilo(void\* arg) {

pthread\_mutex\_lock(&mutex);

printf("Hilo en la sección crítica**\n**");

pthread\_mutex\_unlock(&mutex);

return NULL;

}

int main() {

pthread\_mutex\_init(&mutex, NULL);

pthread\_t hilo;

pthread\_create(&hilo, NULL, funcionHilo, NULL);

pthread\_mutex\_lock(&mutex);

printf("Hilo principal en la sección crítica**\n**");

pthread\_mutex\_unlock(&mutex);

pthread\_join(hilo, NULL);

pthread\_mutex\_destroy(&mutex);

return 0;

}

18. ¿Cuál es la problemática que resuelve el algoritmo de Peterson?

La problemática que resuelve el algoritmo de Peterson garantiza la exclusión mutua entre dos procesos (o hilos) que comparten recursos comunes.

19. ¿Cuál es el mecanismo que implementa el algoritmo del semáforo? y explíquelo con algún ejemplo

El semáforo es un mecanismo de sincronización utilizado en programación concurrente para controlar el acceso a recursos compartidos.

#include <stdio.h>

#include <pthread.h>

#include <semaphore.h>

#define NUM\_THREADS 5

sem\_t semaforo;

void\* funcionHilo(void\* arg) {

int id = \*(int\*)arg;

printf("Hilo %d esperando**\n**", id);

sem\_wait(&semaforo);

printf("Hilo %d en la sección crítica**\n**", id);

sleep(1);

sem\_post(&semaforo);

printf("Hilo %d ha salido de la sección crítica**\n**", id);

return NULL;

}

int main() {

pthread\_t hilos[NUM\_THREADS];

int ids[NUM\_THREADS];

sem\_init(&semaforo, 0, 1);

// Crear y ejecutar hilos

for (int i = 0; i < NUM\_THREADS; ++i) {

ids[i] = i + 1;

pthread\_create(&hilos[i], NULL, funcionHilo, &ids[i]);

}

for (int i = 0; i < NUM\_THREADS; ++i) {

pthread\_join(hilos[i], NULL);

}

sem\_destroy(&semaforo);

return 0;

}