Teoría

1) Enumere al menos 5 funciones básicas que cumple el S.O. Describa los diferentes tipos de interrupciones y de al menos 2 ejemplos de cada uno.

Administración de recursos, ejecución de procesos, interfaz con el usuario, interfaz con el hardware, seguridad, etc.
Las interrupciones de HW y las excepciones son los dos tipos de interrupciones que tenemos. Ejemplos de interrupciones: fin de quantum (se usa en algoritmos de planificación, cuando ejecutan los procesos), fin de entrada-salida (se relaciona con algunos recursos, que informan que una operación finalizó, y son parte de la interfaz con el HW). Ejemplos de excepciones: división por cero (durante la ejecución de un proceso), acceso inválido (por ej, al querer escribir la imagen de otro proceso no podremos hacerlo, por cuestiones de seguridad).

2) El buffet implementa un algoritmo de planificación para servir los cafés. Hasta 3 personas sirven café en paralelo. Si viene un profesor apurado, se lo atiende rápido para que vuelva a clases. Cómo el submarino lleva mucho tiempo, se lo posterga hasta que no haya café para preparar. Si los que esperan un submarino dejaron pasar a más de 10 personas, los atienden sin importar el resto. Indique cómo configuraría la planificación a corto plazo para que esto ocurra correctamente.

Hay diferentes opciones para hacer esto. Una opción es usar colas multinivel, preparadas para trabajar con nivel de multiprocesamiento = 3. La cola de mayor prioridad será para profesores, y atenderá FIFO, sin desalojo (para no interrumpir la preparación de nadie). La de siguiente prioridad será para alumnos, que podrían ser atendidos en FIFO o RR (si una persona puede preparar varios cafés, por partes). Por último, tenemos una cola de menor prioridad, para los submarinos, pero para evitar inanición, se usa un mecanismo de envejecimiento.

3) Haga una comparativa de las estructuras principales que se encuentran en un proceso pesado y un proceso que soporta hilos de manera nativa. Enumere al menos 5 atributos compartidos por las estructuras.

Los procesos pesados y los procesos que soportan hilos de forma nativa, encontramos los siguiente atributos compartidos:

- PID
- PPID
- File open table
- Datos
- Heap
- Código

Comparativa

Proceso Pesado	Proceso que soporta hilos nativos
● Única estructura de control (PCB) ● Único flujo de ejecución. ● Único stack, datos, heap y código	● Una estructura de control por hilo (TCB) ● Cada hilo tiene una estructura de stack ● Heap, datos y código compartidos

4) Defina de forma breve “sección crítica”. Arme un ejemplo simple y concreto.

Conjunto de sentencias donde se accede a un recurso que es compartido y si no se toman recaudos necesarios, se produce condición de carrera. Se hace referencia a un recurso compartido, siendo este accesible por varios procesos/hilos.

Ejemplo:
int var_global=1;

//HiloA	//HiloB

void funcionA()
{
  var_global++;
}
	void funcionB()
	{
	  var_global++;
	}

5) ¿Qué condiciones son necesarias y suficientes para que ocurra un deadlock? ¿En qué caso podemos usar el grafo de asignación de recursos para detectarlo?

Breve descripción de :

- Mutua exclusión
- Retención y Espera
- No desalojo
- Espera Circular

Solo es posible hacer detección de deadlock mediante un grafo de asignación, cuando tenemos recursos con una única instancia.

Práctica

1) amanecer = atardecer = pantano = 0; comer = 20; mut_pasadizo = 1; distraído=1;

Cocodrilo (100 instancias)	Cazador (1 instancia)	Día (1 instancia)

while(1) {
wait(atardecer);
wait(comer);
comer();
signal(comer);
wait(amanecer);
dormir();
wait(pantano);
wait(mut_pasadizo);
escapar();
signal(mut_pasadizo);
	while(1) {
	wait(amanecer);
	ir_al_pantano();
	signal(pantano, 100);
	wait (distraído, 100);
	if(cazar()) {
	hacer_cartera();
	}
	wait(atardecer);
	ir_a_casa();
	}
		while(1) {
		amanecer();
		signal(amanecer, 101);
		atardecer();
		signal(atardecer, 101);
		}

signal(distraído); }		

2)

P1		UTLA1																				
	KLTA	UTLA2																				
		UTLA3																				
	KLTB	UTLB1																				
		UTLB2																				
P2	KLTC	ULTC1																				
	KLTD	ULTD1																				
				CPU			I/O															

3.a) Disponibles (0,0,0) → No puede ejecutar ninguno

3.b) En rojo se ve un deadlock. P3 se ve en inanición. El algoritmo de detección no detecta procesos en inanición, si tienen recursos asignados.