

1ª (1.5 puntos): Se tienen 5 procesos que tienen la variable compartida *m* inicializada a 0. Los procesos no utilizan la exclusión mutua, por lo tanto se pueden producir carreras críticas. Y suponer que sobre la variable *m* no se realizan otras operaciones.

Los procesos ejecutan el siguiente bucle

```
for(i=0;i<10;i++) m+=3;
```

donde *i* es una variable local de cada proceso (no se comparte). Di si son posibles los siguientes resultados y razónalo: a) 0 b) 3 c) 6 d) 149 e) 150

2ª (1.5 puntos). Se tienen 5 procesos con peticiones a 3 recursos. Había dos tablas una con recursos asignados a cada proceso y otra con los recursos que le faltan a cada proceso creo recordar. Pedía comentar cómo haría el algoritmo del banquero para pedir de una sola vez varios recursos para un proceso en concreto según los datos. Los recursos que había eran *E* = {10, 5, 7} y pedía era ver que pasaba si el proceso P2 pedía 1 de A y 2 de C en una sola petición.

3ª (1.5 puntos). ¿Qué hace la función `pthread_cond_wait`? ¿Qué hace la función `pthread_cond_signal`? ¿Qué hace la función `pthread_mutex_trylock`?

4ª (2 puntos). Decir qué hace el siguiente código en shell. Explicar línea a línea el código (explicar todo, comillas, \$, etc.). ¿Qué salida muestra el programa?

Código:

```
#!/bin/bash
LISTA=""`ls *.wt`
resultado=""
for i in {LISTA}; do
pri=`cat ${i}`
resultado= ${resultado} ${pri}
echo i; echo $i
done
echo ${resultado} | cat >> $1
```

5ª (1.5 puntos). ¿Qué es un sistema en tiempo real duro y periódico? Para *N* procesos que se debe cumplir para que un sistema sea calendarizable si los cambios de contexto ocurren cada 20 milisegundos y su ejecución dura 200 microsegundos.

6ª (2 puntos).

a) Solución de Peterson. Detectar y corregir los errores del código.

```
int turno, int interesado[N];

void entrar(int proceso){
    int otro;
    otro = 1 – proceso;
    interesado[proceso] = 1;
    turno = proceso;
    while (turno == otro && interesado[otro] == 1);
}

void salir(int proceso){
    interesado[otro] = 1;
}
```

b) Escribe el código de la Solución de Peterson para 3 procesos.