

Material permitido:
Calculadora NO programable.
Tiempo: **2 horas.**
N

Aviso 1: Todas las respuestas deben estar razonadas.
Aviso 2: Escriba sus respuestas con una letra **lo más clara posible.**
Aviso 3: No use *Tipp-ex* o similares (atasca el escáner).

ESTE EXAMEN CONSTA DE 5 PREGUNTAS

1. (1p) Describa el resultado de la orden `sem=semget(1lave, 3, IPC_CREAT | 0600);`
2. (2p) Explique brevemente para qué usa el núcleo la llamada al sistema `wait` y escriba un pseudocódigo asociado con las principales acciones.
3. (2p) Cuando se produce una interrupción, ¿qué algoritmo invoca el núcleo para el tratamiento de las interrupciones? Describa las principales acciones que realiza dicho algoritmo.
- 4 (2p) Conteste razonadamente a los siguientes apartados:
 - a) (1p) ¿Qué concepto se implementa a partir de las siguientes líneas de código? Explica para qué sirve, su funcionamiento y las principales ventajas.

```
void spin_lock (spinlock_t *s)
{
    while (test_and_set(s) !=0)
    }
void spin_unlock (spinlock_t *s)
{
    *s=0;
}
```

- b) (1p) Describe el comportamiento de las funciones que son invocadas en el código anterior.

(PREGUNTA 5 EN LA SIGUIENTE PÁGINA)

5. (3p) Conteste razonadamente a los siguientes apartados:

a) (2p) Explicar el significado de las sentencias enumeradas ([1]) de este programa.

b) (1p) Si el programa es compilado bajo el nombre DyASO y describir su salida cuando se invoca de la siguiente forma “./DyASO hola”. Suponga que los PIDs de los procesos se asignan de forma consecutiva comenzando desde 1000.

```
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/shm.h>
#include <string.h>
#include <stdio.h>
#include <stdlib.h>

[1] void main(int n, char* arg[])
{
[2] if (n!=2) printf("Argumentos incorrectos");
else {
    int pid, shmid, par, estado;
    char *cadena;
    key_t llave;
[3] llave=ftok(arg[0], 'M');
[4] shmid=shmget(llave,50*sizeof(char),IPC_CREAT | 0600);
    if (shmid==-1)
    {
[5] perror("fallo en shmget");
        exit(1)
    };
[6] cadena=shmat(shmid,0,0);
[7] if ((pid=fork())==-1) exit(2);
    if (pid==0)
    {
        sleep(2);
[8] strcpy(cadena,arg[1]);
[9] shmdt(cadena);
        exit(3);
    }
    else
    {
[10] par=wait(estado);
        printf("El proceso %d dice: %s DyASO\n",pid,cadena);
    }
}
}
```