

Examen de Sistemas Operativos

Junio 2007

(2 puntos por cuestión)

- 1) Escribir un programa que sea capaz de ejecutar tres programas cuyos nombres se le pasan como argumentos, y que pinte un mensaje de aviso cuando los tres programas hayan finalizado. Ejemplo de llamada al programa `ejecuta3` para que ejecute `a.out`, `ordena` y `calcula`:

```
[xxxxx]$ ejecuta3 a.out ordena calcula
```

- 2) Escribir una función que borre todos los ficheros del directorio que se pasa como parámetro, y cuyo primer byte almacenado corresponda al carácter 'A' (mayúscula). La especificación de la función es la siguiente:

```
int borra (const char *nombredir);
```

La función retorna el número de ficheros que han sido borrados.

- 3) Implementar una estructura de datos de tipo pila (stack) a la que pueden acceder varios threads concurrentemente para tomar o depositar datos. Los datos serán números reales y el tamaño máximo de la pila será 100. Implementar además las operaciones de la pila, que tienen la siguiente interfaz:

```
int pon (const float *num);  
float quita ()
```

`pon` devuelve un 0 si la operación ha tenido éxito y un -1 si ha fallado porque la pila está llena, y `quita` sólo puede devolver un valor correcto o esperar si no hay nada en la pila.

- 4) Escribir un fragmento de programa que espere en un `sleep` de alta resolución hasta que transcurran 4,650 segundos, haciendo uso del reloj de tiempo real. En caso de que la función retorne por una señal escribir en la pantalla los segundos que se han esperado y volver a esperar con el tiempo que no se había consumido de la espera anterior.

- 5) Crear un *script* para una *shell* de Unix que permita compilar sin enlazar todos los ficheros acabados en ".c" (que contienen código en lenguaje C) del directorio que se pasa como primer parámetro, y mover al directorio que se pasa como segundo parámetro todos aquellos que hayan compilado sin errores. Realizar el tratamiento de errores. Ejemplo:

```
[xxxxx]$ script dir_fuentes dir_correctos
```