Examen de Sistemas Operativos

Septiembre 2010

(2 puntos por cuestión)

1) Escribir un programa en C que sea capaz de ejecutar el programa cuyo nombre se le pasa como primer argumento desde la línea de comandos. El resto de los argumentos que se le pasan al programa que hay que desarrollar (en número indeterminado) serán utilizados como argumentos del programa que se desea ejecutar. Se debe pintar un aviso cuando el programa haya finalizado. Si el programa se llama `ejecuta` el siguiente ejemplo ejecutaría el programa `prog1` con los argumentos `arg1`, `arg2` y `arg3`:

   ```
   [xxxxx]$ ejecuta prog1 arg1 arg2 arg3
   ```

2) Escribir una función que sirva para copiar la totalidad de los datos de un fichero origen en un fichero destino manteniendo la información que pudiera haber en los primeros 1500 bytes de este fichero destino. La función recibe como parámetro un puntero a un dato del tipo siguiente, que contiene los nombres de los ficheros origen y destino:

   ```
   typedef struct {
      char origen[100];
      char destino[100];
   } nombres;
   ```

   Tratar los errores que se puedan producir. Si se produce un error la función retorna su código, en caso contrario devuelve un 0.

3) Se dispone de las siguientes funciones definidas en el fichero de cabeceras `funciones.h` y presentan los siguientes prototipos:

   ```
   void uno();
   void dos();
   void tres();
   void cuatro();
   ```

   Escribir un programa que ejecute concurrentemente las funciones `uno`, y `dos`. El programa debe ejecurar la función `tres` inmediatamente después de que termine la función `uno`, y debe ejecutar la función `cuatro` después de que hayan terminado las funciones `dos` y `tres`. Utilizar los mecanismos de sincronización que se consideren más adecuados para la ejecución de este sistema.

4) Escribir un fragmento de programa que espere con `nanosleep`() hasta que transcurran 4,165 segundos. En caso de que la función retorne por una señal escribir en la pantalla el tiempo que falta por esperar y después lanzar la señal SIGINT al propio proceso.

5) Crear un *script* para una *shell* de Unix que copie el fichero cuyo nombre se pasa como primer parámetro en el fichero cuyo nombre se pasa como segundo parámetro. El *script* debe comprobar que el fichero destino no existe, y en el caso de que existiera debería preguntar al usuario si desea sobre-escribirlo. Realizar la detección de los posibles errores. Ejemplo de llamada al *script* "`script_copia`" que copia el fichero `nombre_origen` en el fichero `nombre_destino`:

   ```
   $ script_copia nombre_origen nombre_destino
   ```