Programación de Servicios y Procesos

Rafael Ruiz
gandano@gmail.com

Actividad 1.1

1. El objetivo de este ejercicio es practicar las distintas llamadas a la familia de funciones ***exec()***, para ello, se pide desarrollar un programa de nombre .ejecutar.en el que se invoque el comando ***ls -al /usr/bin*** con cada una de las funciones de esta familia, tal y como se indica: .ejecutar opción", Donde las opciones válidas son:
   - -l. Utiliza la función execl().
   - -v. Utiliza la función execv().
   - -le. Utiliza la función execle().
   - -ve. Utiliza la función execve().
   - -lp. Utiliza la función execlp().
   - -vp. Utiliza la función execvp().
2. Escribir un programa C que muestre el identificador del proceso creado al ejecutarlo, el identificador del proceso padre, y el propietario del mismo.
3. Escribir un programa que cree un proceso hijo, y que tanto hijo como padre, escriban en pantalla 10 mensajes identificativos y numerados a partir de una variable común.
4. ¿A qué se denomina proceso Zombi? Escribir código necesario para crear un proceso en este estado.
5. Escribir un programa en el que el proceso principal y el proceso hijo compartan una variable y cada uno de ellos la modifique e imprima su valor en la salida estándar. Por ejemplo, el padre imprima los valores de 0 a 50, y el hijo de 51 a 100.
6. Escribir un programa en el que el proceso principal cree 5 procesos hijos a los cuales se les pasará un código identificador propio y esperará a la finalización de todos los hijos. Cada proceso hijo deberá mostrar 10 veces por la salida estándar su código identificador ***id***, el valor de una variable local ***i***, y el valor de una varible global ***gi***.
7. Realizar las modificaciones necesarias en el ejercicio anterior para crear 100 hijos.

8. Escribir un programa que cree un proceso hijo, y que tanto padre como hijo muestren su ***PID*** y su ***PPID***, y su identificador de grupo de procesos. Pasados 5 segundos, el proceso padre debe terminar (morir de forma natural), y el hijo mostrará de nuevo sus valores ***PID***, ***PPID***, y su identificador de grupo. Pasados 10 segundos, el proceso hijo debe convertirse en el líder de un nuevo grupo de procesos y volverá a mostrar sus indicadores.

9. Escribir un programa llamado ***spy*** que ejecute un comando cuando un fichero se modifique. La forma del programa será: ***spy fichero [-p período] [orden]***. El programa consultará el estado del fichero con la periodicida especificada en el parámetro ***-p*** y expresada en segundos. Si el fichero especificado sufre alguna modificació, entonces ejecutará el comando especificado en el parámetro orden. Los valores por defecto son 10 segundos para el período, y ***ls -l fichero*** para orden.

10. Escribir un programa que reciba a través de la línea de comandos el número de procesos hijos que debe crear. Cada proceso hijo deberá dormir un tiempo aleatorio entre 0 y 50 segundos. El proceso padre debe esperar la finalización de cada uno de sus hijos, y según vayan terminando, presentará en pantalla el ***PID*** del hijo finalizado y el número de segundos que ha estado durmiendo cada hijo.

11. Desarrollar un programa para comprobar las relaciones entre padre e hijo en relación a los ficheros abiertos entre ambos. ¿Qué ocurre si uno de ambos cierra uno o varios ficheros? ¿Qué ocurre si uno de ellos realiza modificaciones en los ficheros?

12. ¿Cómo funciona la función ***waitpid()***? ¿Cómo podría sustituir a la función ***wait()***? Compara el funcionamiento de ambas funciones.

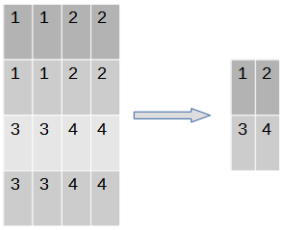
13. Escribir un programa llamado ***mypcat*** que tome como argumento el nombre de un fichero y realice de forma paralela su impresión en pantalla y el conteo del número de caracteres, palabras y líneas, que se mostrará al final de la impresión en pantalla.

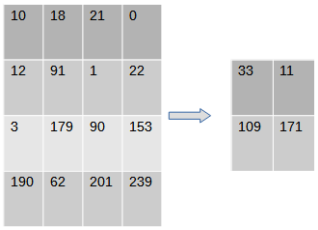
14. Escribir un programa llamado ***mypcp*** que tome como argumento un fichero y cree dos copias exactas denominadas ***nombre_fichero_c1***, y ***nombre_fichero_c2*** de forma paralela.

15. Escribir un programa que cree un hijo y, tanto hijo como padre muestren en valor de sus variables de entorno, y cree una nueva denominada ***MYEDITOR*** con el valor ***joe*** (El cual habrá que instalar previamente).

16. Crear una aplicación para calcular el factorial de un número natural introducido por teclado. El procreso padre se encargará de recoger el número y de crear el primer proceso hijo, que resolverá una parte más pequeña del problema. Este proceso hijo, a su vez, creará otro proceso hijo con el mismo objetivo que el anterior, es decir, resolver un problema menor. Se creará, por tanto una cadena de hijos cada uno de los cuales ayudará en resolver parte del problema, y será el proceso padre el que construya el resultado final.

17. En múltiples algoritmos para el procesamiento de imágenes, se suelen relizar operaciones de medias entre los píxeles adyacentes de una imagen. Así se consigue, por ejemplo, minimizar el pixelado, efectos de difuminado, etc. Se pide realizar una aplicación para que, dada una matriz de valores aleatorios de un tamaño dado (ej. 4 x 4 de bytes), se obtenga una matriz (en este caso de ej. 2 x 2), obtenida como resultado de realizar la media aritmética de cada cuatro bloques de celdas adyacentes.





Se deben tener en cuenta los siguientes requisitos:

a) El numero de procesos a crear sera función del tamaño de la matriz.

b) Se debe implementar una primera solución en el que los procesos hijos sean procesos creados de la forma estándar.

c) Se debe aportar otra solución en la que los procesos hijos sean procesos independientes, es decir, otros procesos que el proceso padre lanzará pasando como argumento los datos que se consideren necesarios, y recogiendo los valores adecuados.


PROGRAMMING
404