Sesión 2: Procesos

Objetivo: Crear/destruir/gestionar procesos en Linux

Ejercicio 1:

Escribe un programa, llamado ej1.c, que cree un proceso hijo. Tanto padre como hijo tienen que mostrar su propio PID y el de su padre o hijo según corresponda.

Llamadas al sistema que se tienen que utilizar:

int fork();

int getpid();

int getppid();

Experimenta con los comandos de sistema ps y top para ver qué información puedes obtener de este proceso padre e hijo. También tenemos el comando “kill -9 PID” que nos permite matar un proceso concreto (en caso que se hubiera quedado colgado).

Ejercicio 2:

Escribe un programa, llamado ej2.c, en el cual existe una variable global a, de tipo long. Este programa tiene que crear un proceso hijo. El proceso padre, tiene que decrementar de forma monótona el valor de la variable a, mientras que el hijo tiene que incrementar el valor de esta variable. Después de incrementar/decrementar el valor de la variable, ambos tienen que mostrar por la salida estándar el valor de esta variable.

¿El resultado de las ejecuciones de estos procesos muestran los resultados que esperabas? ¿Por qué la variable a va cogiendo esos valores?

Ejercicio 3:

Escribe un programa, llamado ej3.c, en el cual se creen, de forma secuencial, 10 procesos hijos. Estos procesos hijos tienen que mostrar 1000 veces, cada uno, su número de PID y el número de hijo que es. No pueden existir a la vez más de un hijo en ejecución.

Llamadas al sistema:

int wait(int *status);

Ejercicio 4:

Escribe un programa, llamado ej4.c, en el cual se creen 10 procesos hijos que trabajen todos de forma concurrente. Los hijos tienen que hacer el mismo trabajo que en ejercicio 3. En este caso, el proceso padre tendrá que crear los 10 hijos y después esperar la finalización de todos sus procesos hijos.