

Tarea 3. Procesos e IPC

Orientaciones sobre la entrega de la tarea

Realice los ejercicios que aparecen a continuación, cada uno como un proyecto diferente, y suba la respuesta a GitHub o Bitbucket. Comparta en Canvas la URL del repositorio donde aparecen las respuestas a los ejercicios.

Para evaluar la tarea, el profesor compilará y ejecutará el proyecto correspondiente a cada ejercicio de cada estudiante para comprobar que no contienen errores de compilación ni de ejecución, y además, se evaluará el uso de las técnicas correctas de programación vistas en las clases.

La tarea estará activa en Canvas hasta el 02 de octubre de 2015 a las 23:55 horas.

No se aceptan trabajos fuera de fecha ni por correo, en ambos casos la calificación de la tarea será 0 puntos.

Ejercicio 1.

Programe una aplicación en C (para Linux) que se ejecute como un demonio (*daemon*) y que cada un número de minutos (especificados como parámetro del programa, por ejemplo: `daemon -m 3` significaría cada 3 minutos) visualice por pantalla los procesos activos en el sistema.

Ejercicio 2.

Escriba un programa en C que reciba por teclado números y cada vez que recibe un número, se lo envíe a otro proceso utilizando el mecanismo de IPC “Segmentos de memoria compartida” para que este calcule e imprima el factorial de ese número. Solo debe existir un proceso que calcula.

Ejercicio 3.

Escriba un programa en C que reciba por teclado números y cada vez que recibe un número, se lo envíe a otro proceso utilizando el mecanismo de IPC “Semáforos” para que este calcule e imprima el factorial de ese número. Solo debe existir un proceso que calcula.

Ejercicio 4.

Escriba un programa en C que reciba por teclado números y cada vez que recibe un número, se lo envíe a otro proceso utilizando el mecanismo de IPC “Colas de mensajes” para que este calcule e imprima el factorial de ese número. Solo debe existir un proceso que calcula.

Ejercicio 5.

Modifique el ejercicio de sockets realizado en clases para para que tanto el cliente como el servidor se ejecuten en la misma computadora y utilicen sockets UNIX para su comunicación.