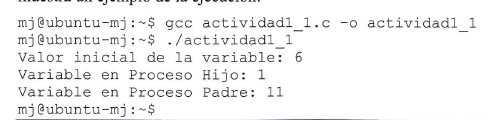
C

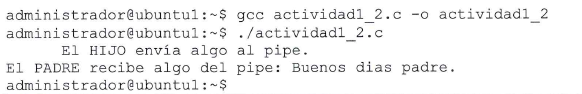
Actividad 1.1

Realiza un programa en C que cree un proceso (tendremos 2 procesos, uno padre y otro hijo). El programa definirá una variable entera y le dará un valor de 6. El proceso padre incrementará dicho valor en 5, y el hijo restará 5. Se deben mostrar los valores en pantalla. A continuación, un ejemplo de la ejecución:



Actividad 1.2

Realizar un programa en C que cree un pipe en el que el hijo envíe un mensaje al padre. Es decir, que la información fluya del hijo al padre. La ejecución debe mostrar la siguiente salida:



**Actividad 1.3**

Realiza un programa en C donde un hijo envíe 3 señales SIGUSR1 a su padre y después envíe una señal SIGKILL para que el proceso padre termine.

Actividad 1.4

Pag 36

Crea un programa Java llamado LeerNombre.java que reciba desde los argumentos de main() un nombre y lo visualice en pantalla. Utiliza System.exit(1) para una finalización correcta del programa, y System.exit(-1) para el caso que no se hayan introducido los argumentos correctos en el main().

A continuación, haz un programa parecido a Ejemplo3.java para ejecutar LeerNombre.java. Utiliza el método waitFor() para comprobar el valor de salida del proceso que se ejecuta. Prueba la ejecución del programa dando valor a los argumentos de main() y sin darle valor. ¿Qué valor devuelve waitFor() en un caso y en otro?

Actividad 1.5

Pag 37

Partiendo del Ejemplo3.java, muestra los errores que se producen al ejecutar un programa Java que no exista.

Actividad 1.6

Pag 38

Escribe un programa Java que lea dos números desde la entrada estándar y visualice su suma. Controlar que lo introducido por teclado sean dos números. Haz otro programa Java para ejecutar el anterior.

Actividad 1.7

Pag 41

Modifica el Ejemplo5.java para que la salida del proceso y la salida de error se almacenen en un fichero de texto, y la entrada la tome desde otro fichero de texto.

Actividad 1.8

Pag 42

Usando ProcessBuilder.Redirect, modifica el Ejemplo5.java para que la salida del proceso se muestre en la consola, la entrada la tome desde un fichero de texto, y la salida la lleve a un fichero de texto.

Actividad 1.12

Pag 66

Para enviar enteros se usa la función pvm\_pkint(). Por ejemplo, para enviar un número

Actividad 2.8

Pag104

Crea una clase saldo

Actividad 3.1

Pag 122

Realiza un programa Java que admita desde la línea de comandos un nombre de máquina o una dirección IP y visualice información sobre ella.

Actividad 3.2

Pag 135

Realiza un programa servidor TCP que acepte dos clientes. Muestra por cada cliente conectado sus puertos local y remoto.

Crea también el programa cliente que se conecte a ese servidor. Muestra los puertos locales y remotos a los que está conectado su socket, y la dirección IP de la máquina remota a la que se conecta.

Actividad 3.3

Pag 140

Crea un programa servidor que envíe un mensaje a otro programa cliente y el programa cliente le devuelva el mensaje en minúscula.

Actividad 3.4

Pag 140

Crea un programa cliente que introduzca por teclado un número entero y se lo envíe al servidor. El servidor le devolverá el cuadrado del número.

Actividad 3.5

Pag 140

Crea un programa servidor que pueda atender hasta 3 clientes. Debe enviar a cada cliente un mensaje indicando el número de cliente que es. Este número será 1, 2 o 3. El cliente mostrará el mensaje recibido. Cambia el programa para que lo haga para N clientes, siendo N un parámetro que tendrás que definir en el programa.

Actividad 3.6

Pag 149

Crea un programa cliente usando sockets UDP que envíe el texto escrito desde la entrada estándar al servidor. El servidor le devolverá la cadena en mayúsculas. El proceso de entrada de datos finalizará cuando el cliente introduzca un asterisco. Crea un programa servidor que reciba cadenas de caracteres, las muestre en pantalla y se las envíe al emisor en mayúscula. El proceso finalizará cuando reciba un asterisco.

Establece un tiempo de espera de 5000 milisegundos con el método setSoTimeout(5000) par hacer que el método receive() del programa cliente se bloquee. Pasado ese tiempo controlar si no se reciben datos lanzando la excepción InterruptedIOException, en cuyo caso visualiza un mensaje indicado que el paquete se ha perdido. Para probarlo ejecuta el programa cliente sin ejecutar el servidor. Puedes ejecutar varios programas cliente a la vez.