Ejercicios sobre handlers en *Lower_Layer_UDP*

Departamento de Teoría de la Señal y Comunicaciones y Sistemas Telemáticos y Computación (GSyC)

Noviembre de 2017

Ejercicio 1

- Descarga los programas cliente (`client.adb`) y servidor (`server.adb`) de la página de la asignatura en el Aula Virtual, que son los mismos que se utilizaron en los ejercicios de introducción al paquete LLU.
- Compila y ejecuta ambos programas. Para ello abre 2 ventanas de terminal diferentes, y colócalas de forma que el contenido completo de ambas pueda verse a la vez. Cámbiate en las dos ventanas a la carpeta donde tengas los programas, y después haz, en este orden, lo siguiente:
  1. En una de las ventanas de terminal, ejecuta el programa servidor
  2. En la otra ventana de terminal, ejecuta el programa cliente
  3. En la ventana del cliente, introduce una cadena de caracteres
  4. Comprueba en las dos ventanas el resultado de la ejecución de cliente y servidor.

Ejercicio 2

Modifica el servidor de forma que:

- La recepción de mensajes se realice dentro del procedimiento `Server_Handler` del paquete `Handlers`, cuya especificación (`handlers.ads`) es la siguiente:

```
with Lower_Layer_UDP;

package Handlers is
   package LLU renames Lower_Layer_UDP;

   -- Handler para utilizar como parámetro en LLU.Bind en el servidor
   -- Muestra en pantalla la cadena de texto recibida y responde enviando
   --   la cadena "¡Bienvenido!"
   -- Este procedimiento NO debe llamarse explícitamente
   procedure Server_Handler (From    : in     LLU.End_Point_Type;
                             To      : in     LLU.End_Point_Type;
                             P_Buffer: access LLU.Buffer_Type);


end Handlers;
```

Para ello debes trasladar el código del bucle que recibía y respondía a los mensajes del programa `server.adb` original a `handlers.adb`. Así mismo, debes registrar y activar el handler cuando enlazas el servidor, utilizando código como el siguiente:

```
-- Se enlaza al End_Point para poder recibir en él con un handler.
-- Tras llamar a Bind ya se pueden estar recibiendo mensajes automáticamente
-- en el manejador
LLU.Bind (Server_EP, Handlers.Server_Handler'Access);
```

- El servidor termine su ejecución cuando el usuario pulse la tecla `T` o `t`. Puedes utilizar código como el siguiente:

```
-- A la vez que se están recibiendo mensajes en el manejador se
-- siguen ejecutando las siguientes sentencias en el programa principal
Ada.Text_IO.Put_Line ("Servidor arrancado");
Ada.Text_IO.Put_Line ("Para terminar este servidor pulse 'T' o 't'");

-- Hasta que no pulsen 'T' o 't' en el teclado no termina el servidor
Finish := False;
while not Finish loop
   Ada.Text_IO.Get_Immediate (C);
   if C = 'T' or C = 't' then
      Finish := True;
      LLU.Finalize;
   else
      Ada.Text_IO.Put_Line ("Para terminar este servidor pulse 'T' o 't'");
   end if;
end loop;
```

IMPORTANTE: para que este código funcione debes definir las variables necesarias como el `Boolean` `Finish` o el `Character` `C`.

IMPORTANTE: Observa que este bucle se ejecuta continuamente al mismo tiempo que el handler es capaz de recibir los mensajes que llegan de los clientes.

Ejercicio 3

Modifica el cliente de forma que:

- Utilice un handler para recibir las respuestas del servidor. Para ello debes añadir el siguiente procedimiento a `handlers.ads` e implementarlo dentro de `handlers.adb`:

```
-- Handler para utilizar como parámetro en LLU.Bind en el cliente
-- Muestra en pantalla la cadena de texto recibida
-- Este procedimiento NO debe llamarse explícitamente
procedure Client_Handler (From    : in     LLU.End_Point_Type;
                          To      : in     LLU.End_Point_Type;
                          P_Buffer: access LLU.Buffer_Type);
```

- Conste de un bucle infinito que lea cadenas de caracteres y las mande al servidor.
- Termine cuando la cadena introducida por el usuario esté vacía (es decir, su longitud sea 0).

Observa que el código principal del cliente únicamente está encargado de leer y enviar las cadenas que el usuario escribe en el teclado. La recepción de las respuestas tiene lugar en el código del handler.