Taller de Sistemas Operativos Avanzados
Práctica #3: Llamadas a Procedimientos Remotos (RPC)
"Resguardos del Cliente y del Servidor"

Objetivo

Implementar una simulación del funcionamiento de los resguardos del cliente y del servidor para las llamadas a procedimientos remotos sobre un sistema operativo distribuido con MicroNúcleo basado en el modelo Cliente/Servidor.

Descripción

La función de los resguardos del cliente y del servidor en un sistema distribuido consiste en ocultar al usuario (programador/programa) las operaciones de E/S empleadas para el modelo Cliente-Servidor. La RPC se lleva a cabo de tal forma que el cliente (el procedimiento implementado por el programador) llame al resguardo del cliente, el cual tiene la apariencia de ser una subrutina ordinaria de biblioteca, sólo que se encarga de empacar los parámetros del cliente, enviarlos al servidor, esperar la respuesta y regresar el resultado al cliente; el resguardo del servidor se encarga de esperar las solicitudes de los clientes y al momento de recibir una solicitud revisa la operación solicitada, desempaca el mensaje, llama al servidor adecuado, obtiene la respuesta del servidor, la empaca y envía el resultado al emisor de la solicitud; el servidor no es más que la subrutina ordinaria de biblioteca, como read(fd,buffer,nbytes), la cual el cliente creyó haber llamado en un principio; dicho servidor al momento de ejecutarse cree haber sido llamado por el cliente, siendo que realmente fue llamado por el resguardo del servidor. Como resultado de lo anteriormente expuesto, los resguardos del cliente y del servidor se encargan de lograr un cierto nivel de transparencia ante el programador.

Requerimientos Generales

La simulación de los resguardos del cliente y del servidor para llamadas a procedimientos remotos deberá cumplir con lo siguiente:

1. Todos los requerimientos (generales, visuales, funcionales, no funcionales, complementarios) así como restricciones especificados para la práctica #2.
2. Mediante el MicroNúcleo, se efectúe la comunicación entre dos procesos ejecutándose en dos máquinas distintas.
3. Uso del modelo Cliente-Servidor, es decir, las primitivas send(dest,messagePtr) y receive(addr,messagePtr) de la práctica 2 como parte para efectuar la RPC.

Requerimientos Visuales y Funcionales

4. Desplegar en el área de texto respectiva (proceso cliente ó proceso servidor) todos los eventos significativos (la ruta crítica) ocurridos en el cliente, el resguardo del cliente, el resguardo del servidor y el servidor.
5. Asignación desde la IG del Cliente de todos los datos a ser utilizados en el programa Cliente para todas las subrutinas a llamar, según el punto 8 siguiente.

Requerimientos No Funcionales

6. Especificación de la interfaz del servidor, con soporte para cuatro operaciones distintas las cuales en su conjunto cuenten con lo siguiente:
   a) Todas reciban al menos un parámetro
   b) Alguna reciba dos o más parámetros
   c) Los parámetros sean únicamente tipos de datos primitivos o arreglos de los mismos
   d) Algún parámetro declarado en una operación sea de tipo primitivo y de al menos 2 bytes de longitud (short, int, long); para dicho parámetro la semántica de la operación debe contemplar valores en todo el rango del tipo de dato declarado desde el menor valor (negativo) hasta el mayor valor (positivo)
   e) Alguna sea implementada como función y el valor izquierdo cumpla con lo mismo del inciso “d” anterior (las restricciones descritas para el parámetro aplican al valor izquierdo)
   f) Al menos una declare parámetro de entrada
   g) Al menos una declare parámetro de salida
   h) La especificación de la interfaz se incluya en el código fuente del resguardo del servidor como comentario (/* ... */).
7. Implementación de la interfaz del servidor en la construcción de los servidores respectivos.
8. Implementación del programa cliente el cual efectúe la llamada a cada uno de los cuatro procedimientos remotos disponibles; el programa cliente ejecute en serie (en orden lógico: primero crear archivo, luego escribir, leer; nótese que eliminar, podría ser en cualquier momento) todas las subrutinas, mostrando los resultados tras ejecutar cada una
9. Manejo de los parámetros en pila por parte del cliente, el resguardo del cliente, el resguardo del servidor y el servidor; para este rubro se requiere la existencia de una operación de al menos dos parámetros.

a) El cliente meta los parámetros del procedimiento en una pila accesible a todo el proceso; para esto, programar una subrutina que simule el código ensamblador generado por un compilador tras encontrar la llamada a una subrutina, a modo de insertar en pila los parámetros y sacar el resultado a consecuencia de la llamada al resguardo del cliente de tal forma que la llamada a dicha subrutina se vea en código fuente del cliente, con la transparencia buscada para RPC.
b) La implementación del resguardo del cliente carezca de parámetros y que saque de la pila los parámetros proporcionados por el cliente según el inciso anterior
c) El resguardo del servidor meta en la pila del proceso los parámetros obtenidos del desordenamiento de parámetros

d) El servidor saque de la pila los parámetros insertados por el resguardo del servidor, trabaje con ellos y coloque la respuesta (el valor izquierdo) en la pila cuando sea el caso
10. Cliente: la parte cliente no debe incluir código relacionado a la interfaz gráfica.
11. Llamada al resguardo del cliente.
12. Manejo de buffers por los resguardos: los buffers deberán ser creados del tamaño exacto que se necesiten para ser completamente llenados, sin bytes de sobra.
13. Ordenamiento y desordenamiento de parámetros: los parámetros que representen los tipos primitivos de datos (short, int, long) al ser ordenados, no deberán ocupar más espacio en el buffer del que necesitan (ej. un tipo de dato short de Java requiere de solo 2 bytes y puede representar números en el rango –32768 a +32767); lo anterior aplica así mismo para el CODOP; debe no ser incluido en el mensaje tipo alguno de separador entre los parámetros ordenados.
14. El ordenamiento y desordenamiento de parámetros, así como el empaque y desempaque de la respuesta debe realizarse de acuerdo al código de operación, de tal forma que el código fuente correspondiente a cada operación sea independiente.
15. Señalamiento al núcleo de la forma ordinaria abordada en el modelo cliente/servidor, obviando lo necesario para efectuar la conexión dinámica, tema que se aborda en la siguiente práctica.
16. Llamada al servidor por el resguardo del servidor.
17. Envío de solicitud y respuesta entre los procesos cliente y servidor a través de los resguardos y los núcleos respectivos.
18. La comunicación entre el cliente y el servidor deberá estar exenta de fallos a nivel de procesos, esto es, los procedimientos remotos deberán estar correctamente programados y validados para cualquier solicitud posible del cliente según lo especificado para la interfaz del servidor; para lo anterior es aplicable la modificación y recompilación de código en la parte cliente.

Requerimientos y Restricciones Complementarias

19. La inserción en código fuente de instrucciones para retardo en ejecución de hilos (Thread.sleep(long milisegundos)) es aplicable para la validación de esta práctica y subsecuentes durante la revisión.
20. Invariablemente se debe cumplir con los requerimientos citados en los incisos c y d del punto 6.

Pseudocódigo

```
Cliente{
  inicializar parámetros
  int leidos=read(fd,buffer,nbytes);
  imprimir "leidos y buffer"
}
CodigoEnsamble{
  meter parámetros en pila
  JMP a subrutina  //ejem: read
  recibir valor izquierdo
}
```

```
Servidor{
  sacar parámetros de pila
  realizar servicio
  regresar valor izquierdo
}
ResguardoCliente{
  sacar parámetros de pila
  preparar buffer de mensajes
  ordenamiento de parámetros
  llenar campos de encabezado
  importación de interfaz
  send(dest,solicitud)
  receive(addr,respuesta)
  desempacar respuesta
  restauración de parámetros out ó in-out
  regresar valor izquierdo
}
ResguardoServidor{
  exportación de interfaz
  while(...){
    receive(addr,solicitud)
    segun codop{
      desordenamiento de parámetros
      meter parámetros en pila
      llamar al servidor  //ejem: read
      recibir valor izquierdo
      empacar respuesta
    }
    send(dest,respuesta)
  }
  deregistro del servicio
}
```

Criterios de evaluación

- Revisión sólo si en la práctica #2 se logra que el proceso Cliente reciba su respuesta.
- Los establecidos en las "Reglas de Operación y Evaluación" del Taller de Sistemas Operativos Avanzados y los correspondientes "Periodos de Entrega" de la práctica.
- Código fuente indicando en la parte superior de cada archivo: nombre del(la) alumn@, sección y no. de práctica; entregado vía e-mail en formato zip.
- Esta práctica se evaluará en dos fases, calificando en la primera los 14 requerimientos de la práctica, verificados en una RPC completa al nivel de proceso. Las otras tres operaciones quedan como tarea a calificarse la semana siguiente al primer periodo de entrega, como parte de la práctica.
- Fecha de asignación de la práctica: 01 de Abril de 2009
- Fecha límite para entrega del primer avance de la práctica: 24 de Abril de 2009
- Fecha límite para entrega de la práctica completa: 11 de Mayo de 2009
- Observación: Se posterga la práctica completa por reanudar clases el 11/Mayo
- Evaluación por cobertura de requerimientos y fecha de entrega.