

INFORME BOLSA DE VALORES

INTEGRANTES: Carlos Orlando Barón León -
Andrés Felipe Cocunubo

ESTRUCTURAS

Nombre: Orden

Variables:

- **Nombre Bróker:** Nombre del bróker que realiza la orden
- **Tipo:** Tipo de orden: Venta (V) o Compra (C).
- **Acciones:** Acciones a vender.
- **Precio:** Precio de cada acción.
- **Nombre Empresa:** Empresa de las acciones.

Uso: Estructura para almacenar una orden.

Nombre: Ordenes

Variables:

- **Orden compras:** Tabla de órdenes de compra.
- **Orden de ventas:** Tabla de órdenes de venta.
- **Tamaño compras:** Tamaño de la tabla de compras.
- **Tamaño ventas:** Tamaño de la tabla de ventas.

Uso: Estructura para almacenar las ordenes en una empresa.

Nombre: libroOrdenes

Variables:

- **Nombre empresa:** Nombre de la empresa.
- **Libro de órdenes:** Libro de órdenes de la empresa.

Uso: Estructura para almacenar las empresas y su respectivo libro en el sistema.

Nombre: Usuario

Variables:

- **ID pipe:** Identificación del proceso del usuario.
- **ID usuario:** Numero identificador del usuario.

- **Nombre usuario:** Nombre identificador del usuario.
- **Nombre pipe:** Nombre del pipe único del usuario.
- **Identificador pipe único:** identificador del pipe único del usuario.

Uso: Estructura para almacenar los usuarios en el sistema.

Nombre: Empresa

Variables:

- **Nombre empresa:** Nombre de la empresa.
- **Acciones:** Acciones que posee de la empresa.
- **Precio:** Precio de una acción.

Uso: Estructura para almacenar una empresa que posee el bróker, su número de acciones y precio acción.

Nombre: Solicitud

Variables:

- **Nombre bróker:** Nombre del bróker que envía la solicitud.
- **Tipo:** Tipo de solicitud: Transacción (T), Consulta (Q), Registro (R), Desconexión (D).
- **ID proceso:** pid del proceso que envía la solicitud, aplica para registro (R).
- **Empresas:** Arreglo de empresas que maneja del usuario, aplica para registro (R).
- **Tamaño de empresas:** Tamaño del arreglo anterior, aplica para registro (R).
- **Orden:** Orden de compra o venta a procesar, aplica para transacción (T).
- **Nombre de la empresa:** Nombre de la empresa a consultar, aplica para consulta (Q).

Uso: Estructura para almacenar las solicitudes que se le envían al StockMarket.

PIPES

Nombre: Pipe general

Descripción: Por este canal de comunicación los brókeres enviarán las diferentes solicitudes para que el stockMarket realice las acciones correspondientes. Este pipe es creado StockMarket y lo abre en modo lectura(O_RDONLY), mientras que el Bróker lo abre en modo escritura(O_WRONLY).

Permisos creación: S_IRUSR | S_IWUSR

Permisos de uso: O_RDONLY | O_WRONLY

Nombre: Pipe Bróker

Descripción: Por este canal de comunicación el stockMarket, enviará las respuestas a las diferentes solicitudes que pida el Bróker. Este pipe es creado Bróker y lo abre en modo lectura(O_RDONLY), mientras que el stockMarket lo abre en modo escritura(O_WRONLY). Cabe resaltar que este pipe es único tanto para un bróker como para el stockMarket, esto para evitar pérdida de información. Se utiliza el permiso O_NONBLOCK, para que este se mantenga abierto en ambos sentidos y no espere a que alguien use el pipe para poder continuar con la ejecución del programa.

Permisos creación: S_IRUSR | S_IWUSR

Permisos de uso: O_RDONLY | O_WRONLY | O_NONBLOCK

SEÑALES

Modo de activación: El stockMarket envía una señal y dependiendo del tipo orden que se reciba se realiza alguna de las funciones auxiliares. El comando utilizado es kill, el cual recibe por parámetro el id del proceso(bróker), y el identificador de la señal.

Tipo: SIGUSR1->

SINCRONIZACIÓN: Como la comunicación es asíncrona, se maneja un ciclo infinito para enviar las solicitudes por parte de bróker y no importa en que momento el stockMarket envía la señal, el bróker siempre realizará el

signal_handler especificado y continuará con la ejecución.

Signal_Handler:

- **leer_pipe:** Cuando el stockMarket realiza el llamado de la señal, esta función inicialmente lee del pipe, para obtener la respuesta de la solicitud realizada, y según el tipo ejecuta alguna de las funciones auxiliares.

Funciones auxiliares:

- **procesarCompra:** Recibe por parámetro la estructura de la respuesta con orden tipo compra y en su implementación realiza las operaciones correspondientes para actualizar los datos necesarios dentro del bróker.
- **procesarVenta:** Recibe por parámetro la estructura de la respuesta con orden tipo venta y en su implementación realiza las operaciones correspondientes para actualizar los datos necesarios dentro del bróker.
- **procesarQuery:** Recibe por parámetro la estructura de la respuesta con orden tipo consulta y en su implementación realiza las operaciones correspondientes para mostrar por pantalla el precio de una acción de una empresa específica en el mercado.

FUNCIONES STOCKMARKET

Nombre Función: procesarSolicitud

Descripción: Se encarga de revisar el tipo de cada solicitud y aplicar ciertas operaciones dependiendo de su tipo.

Parámetros:

- **Solicitud:** Solicitud recibida por el bróker.

Nombre Función: registrarUsuario

Descripción: Se usa para las solicitudes de registro. Crea un registro en el sistema del usuario con su nombre y id de pipe único.

Parámetros:

- **Solicitud:** Solicitud recibida por el bróker.

Nombre Función: registrarEmpresa

Descripción: Crea un registro de las empresas en el sistema, de acuerdo, a las empresas que envía un usuario.

Parámetros:

- **Solicitud:** Solicitud recibida por el bróker.

Nombre Función: procesarCompra

Descripción: Se encarga intenta comprar acciones de la empresa dada, si no hay éxito realiza un registro en la compra del libro de órdenes.

Parámetros:

- **Orden:** Subestructura de la estructura solicitud que contiene los datos de la empresa, y la transacción a realizar.
- **Posición Empresa:** Posición de la empresa en el arreglo del libro de ventas.

Nombre Función: procesarVenta

Descripción: Se encarga intenta vender acciones de la empresa dada, si no hay éxito realiza un registro en la venta del libro de órdenes.

Parámetros:

- **Orden:** Subestructura de la estructura solicitud que contiene los datos de la empresa, y la transacción a realizar.
- **Posición Empresa:** Posición de la empresa en el arreglo del libro de ventas.

Nombre Función: consultarPrecio

Descripción: Se encarga de darle al usuario la información del precio vigente de las acciones de una empresa dada, o un mensaje de error en caso de que no exista.

Parámetros:

- **Solicitud:** Solicitud recibida por el Broker.

Nombre Función: comprarAcciones

Descripción: Se encarga de realizar una compra de x acciones de una empresa a un precio y, dadas las reglas de negocio. Muestra un mensaje de error cuando las reglas no se cumplen.

Parámetros:

- **Orden:** Subestructura de la estructura solicitud que contiene los datos de la empresa, y la transacción a realizar.
- **Posición Empresa:** Posición de la empresa en el arreglo del libro de ventas.

Retorno:

- Retorna 0 si no se compraron o no se compraron todas las acciones, 1 si se pudieron comprar todas.

Nombre Función: venderAcciones

Descripción: Se encarga de realizar una venta de x acciones de una empresa a un precio y, dadas las reglas de negocio. Muestra un mensaje de error cuando las reglas no se cumplen.

Parámetros:

- **Orden:** Subestructura de la estructura solicitud que contiene los datos de la empresa, y la transacción a realizar.
- **Posición Empresa:** Posición de la empresa en el arreglo del libro de ventas.

Retorno:

- Retorna 0 si no se compraron o no se compraron todas las acciones, 1 si se pudieron comprar todas.

Nombre Función: eliminarOrden

Descripción: Elimina un registro en el libro de órdenes, cuando una operación de compra o venta se realiza con éxito.

Parámetros:

- **Posición Empresa:** Posición de la empresa en el arreglo del libro de ventas.
- **Tipo:** Tipo de orden.

Nombre Función: desconexion

Descripción: Cierra el pipe de un usuario dado, cuando este salga del mercado.

Parámetros:

- **Solicitud:** Solicitud recibida por el usuario.

FUNCIONES BROKER

Nombre Función: registrarEmpresas

Descripción: Obtiene las empresas que tiene el bróker como recursos iniciales y les asigna una estructura para cargarlas en el sistema.

Parámetros:

- **Archivo:** Archivo con los recursos iniciales.
- **Nombre Empresa:** Nombre de la empresa que se quiere cargar.
- **Cantidad de acciones:** Cantidad de acciones de la empresa que se quiere cargar.

Retorno:

- **Cantidad de empresas:** Cantidad de empresas que tiene el usuario hasta el momento.

Nombre Función: registroStock

Descripción: El bróker se registra en el mercado para hacer solicitudes y se crea el pipe único.

Parámetros:

- **Pipe:** Código del pipe
- **Cantidad Empresas:** Cantidad de empresas que maneja el bróker.

- **Nombre del bróker:** Nombre del bróker en el sistema.

Nombre Función: comprarAccion

Descripción: Se crea el paquete de solicitud con los datos necesarios para realizar una compra en el mercado.

Parámetros:

- **Pipe:** Código del pipe
- **Nombre empresa.**
- **Cantidad de acciones.**
- **Precio:** Precio de una sola acción.
- **Saldo bróker:** Monto del bróker que tiene disponible.
- **Cantidad de empresas.**
- **Nombre del bróker.**

Nombre Función: venderAccion

Descripción: Se crea el paquete de solicitud con los datos necesarios para realizar una venta en el mercado.

Parámetros:

- **Pipe:** Código del pipe
- **Nombre empresa.**
- **Cantidad de acciones.**
- **Precio:** Precio de una sola acción.
- **Cantidad de empresas.**
- **Nombre del bróker.**

Nombre Función: venderTodo

Descripción: Se crea el paquete de solicitud con los datos necesarios para realizar una compra en el mercado de todas las acciones de la empresa al precio del mercado.

Parámetros:

- **Pipe:** Código del pipe
- **Nombre empresa.**
- **Cantidad de acciones.**
- **Nombre del bróker.**

Nombre Función: consultarEmpresa

Descripción: Se arma el paquete solicitud para conocer el precio del mercado de cierta empresa.

Parámetros:

- **Pipe:** Código del pipe
- **Nombre empresa.**
- **Nombre del bróker.**

Nombre Función: monto

Descripción: Muestra por pantalla el saldo actual del bróker y además el estado de cada una de las empresas que tiene a su cargo.

Nombre Función: desconexion

Descripción: Se arma un paquete de solicitud, para desconectar el bróker del mercado.

Parámetros:

- **Pipe:** Código del pipe
- **Nombre empresa.**
- **Nombre del bróker.**

FUNCIONES UTILS

Nombre Función: crearPipe

Descripción: Crea un nuevo pipe con el nombre que se le pasa por parámetro.

Parámetros:

- **Nombre del pipe.**

Retorno:

- **Id pipe:** Identificador único del pipe.

Nombre Función: abrirPipe

Descripción: Abre el canal de comunicación(pipe) con el nombre que se le pasa por parámetro.

Parámetros:

- **Nombre del pipe.**
- **Banderas:** Permisos que tiene el pipe al momento de abrir.

Retorno:

- **Id pipe:** Identificador único del pipe.

Nombre Función: cerrarPipe

Descripción: Cierra el pipe con el nombre que se le pasa por parámetro y el código.

Parámetros:

- **Nombre del pipe.**
- **Código del pipe.**

Nombre Función: abrirArchivo

Descripción: Abre el archivo de recursos iniciales del bróker, retorna el archivo abierto.

Parámetros:

- **Nombre del archivo.**

PROBLEMAS ENCONTRADOS:

- **Pipe bloqueante:** Al momento de ejecutar el programa tanto el programa del bróker como el programa StockMarket se quedaban esperando que alguno de los enviara información por el pipe.
 - **Solución:** Se colocó el permiso de uso O_NONBLOCK en el pipe único para bróker y stockMarket, para que la ejecución de los programas continuara sin necesidad de enviar información a través del pipe.
- **Obtención errónea del identificador del proceso del bróker:** Inicialmente se obtenía el identificador con el comando getppid(), lo que provocaba que después de enviar una señal la consola se cerraba.
 - **Solución:** La manera correcta de enviar un señal es con getpid().
- **Comunicación asíncrona:** Al momento de obtener el precio de una acción específica y guardarla en el sistema, el programa continuaba su ejecución, sin haber esperado la respuesta por parte del stockMarket, lo que provocaba que la información estuviera mal asignada.
 - **Solución:** Se utilizó la llamada al sistema "pause ()", para que el programa esperara la respuesta y pudiera asignar el precio correspondiente.