

SISTEMAS LEGADOS

Práctica 1

Cajero automático COBOL

Hayk Kocharyan
757715@unizar.es

Enrique Ruiz Flores
766685@unizar.es

José Félix Yagüe Royo
755416@unizar.es

11 de octubre de 2020

Resumen

Durante esta práctica se ha llevado a cabo la implementación, compilación y ejecución de un proyecto en lenguaje COBOL [2]. Para poder resolver todos los problemas ha sido necesario documentarse previamente sobre este lenguaje, tras esto se ha procedido a la corrección de errores de código, y el desarrollo de nuevas funcionalidades para el cajero como son listar transferencias, cambiar el pin de la tarjeta, o programar una transferencia como se comentará a lo largo de esta memoria.

Índice

1. Tareas Realizadas	2
1.1. Búsqueda e instalación del compilador	2
1.2. Corrección de errores	2
1.3. Implementación de funcionalidades	2
1.4. Pruebas realizadas	3
2. Problemas encontrados	4
3. Conclusiones	4
4. Reparto de tareas	4

1. Tareas Realizadas

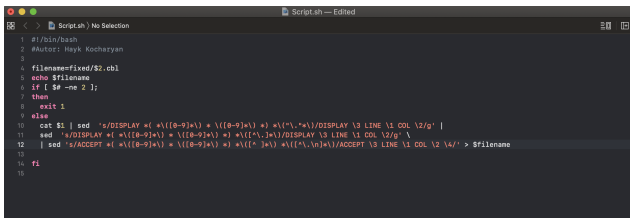
1.1. Búsqueda e instalación del compilador

Como primera tarea los tres miembros del grupo tuvimos que instalar un compilador de COBOL en máquinas Ubuntu para poder compilar y ejecutar los código fuente. El compilador por el que hemos optado es **GnuCOBOL** o también conocido como **Open-COBOL**[3] en su versión 1.1.0.

1.2. Corrección de errores

Tras encontrar un compilador, se realizó una primera compilación y se encontraron varios fallos en los ficheros, principalmente en la sintaxis de las sentencias *DISPLAY* y *ACCEPT*. Por ello como estas sentencias aparecían numerables veces en todos los ficheros fuente, se procedió a la elaboración de dos *shell scripts* que se encargarían de aplicar a cada fichero *.cbl* dos comandos *sed* para modificar la sintaxis de dichas instrucciones y cambiarla por la aceptada por el compilador.

Una vez compilaban todos los ficheros, creamos el ejecutable del cajero a través del comando `cobc -x BANK1.cbl` y realizamos pruebas de ejecución de todas las opciones del cajero.



alguna transferencia fijada para ese día, en tal caso las efectúa y borra los registros correspondientes del fichero de programadas.

En cuanto a las transferencias mensuales, añadimos una opción de *Realizar mensualmente durante X*, si el usuario deja el dato por defecto, en este caso 0, solamente se realiza una vez, en cualquier otro caso se realizará las veces indicadas y como será una transferencia futura se programará.

Como mejora de funcionalidad, se ha modificado la opción de ingresar billetes permitiendo al usuario seleccionar los billetes de cada tipo que desea introducir como se muestra en la siguiente captura.

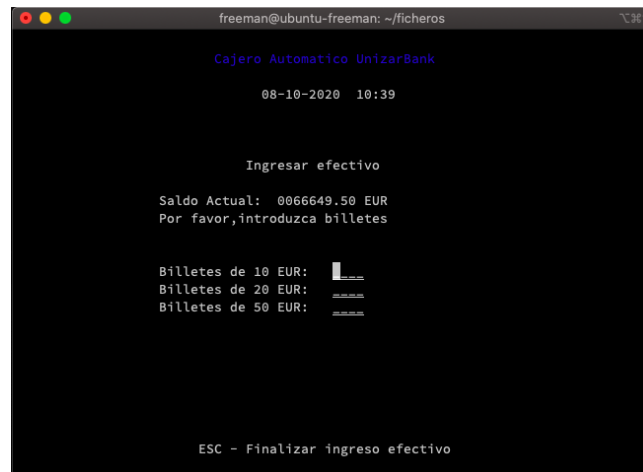


Figura 3: Operación de ingresar billetes

Por último, se ha implementado la funcionalidad de cambio de contraseña, en la que se pide introducir la nueva contraseña dos veces, con el objetivo de evitar fallos a la hora del cambio. Una vez comprobado que las dos contraseñas introducidas coinciden se modifica el fichero correspondiente.

1.4. Pruebas realizadas

Para verificar el correcto funcionamiento tanto de las opciones preexistentes como de las nuevas funcionalidades incorporadas se realizaron pruebas manuales para todas ellas, probando todos los escenarios posibles como se describirá a continuación.

- Para probar la opción de consultar saldo se hicieron diversas consultas antes y después de realizar varios movimientos para comprobar que se actualizaba correctamente. Así mismo se comprobó que dichos movimientos estaban quedando registrados mediante la opción de consultar movimientos, probando también las diferentes opciones de filtrado.
- Para la opción de ingreso de efectivo se hicieron pruebas tras incorporar la opción de seleccionar el número de billetes de cada tipo que se desean introducir, mientras que para la de retirada de efectivo se probó a seleccionar tanto cantidades disponibles como cantidades superiores al saldo disponible para comprobar que en dicho caso se notificaba del error y no permite realizar el movimiento.
- También se hicieron pruebas para comprobar el correcto funcionamiento de la función de ordenar transferencias, probando las 4 opciones posibles, realizar una transferencia instantánea, una transferencia instantánea que se repita mensualmente, una transferencia programada y una programada que se repita mensualmente. Tras la corrección de varios errores se comprobó el correcto funcionamiento en los 4 casos.
- Se verificó que las transferencias ordenadas fueran listadas mediante la nueva funcionalidad incorporada de listar transferencias.
- Se hicieron pruebas de compra de entradas a espectáculos y se verificó que la compra aparecía en la consulta de movimientos.

- Por último se hicieron pruebas de cambio de clave de varias tarjetas y se verificó que el cambio se realizaba correctamente.

2. Problemas encontrados

El primer problema encontrado fue a la hora de buscar un compilador. En primer lugar, instalamos sin darnos cuenta versiones diferentes del compilador seleccionado puesto que los integrantes tenemos diferentes sistemas operativos, por ello al compilar los fuentes cada miembro obtenía un ejecutable con ciertas variaciones respecto a los demás como diferencias en la visualización y comportamiento en ciertas funciones. Tras esto se decidió usar una misma versión estable para los tres, por lo que se optó por la versión 1.1.0 instalado en Ubuntu 18.04 (en versiones superiores de Ubuntu este compilador no existe).

Otro de los problemas con los que nos encontramos fue que el fichero de movimientos no existía, es decir, el programa daba un error debido a que intentaba ejecutar *open* sobre un fichero inexistente. Desconocíamos lo que era un fichero *.ubd* por lo que tras investigar y ver que se trataba de un fichero binario, no sabíamos como crearlo. Tras varias pruebas, llegamos a la solución de crearlo con un comando **touch movimientos.ubd**. Una vez creado, realizamos pruebas de ingresos y retiradas con la cuenta accedida sin ningún problema.

El siguiente problema fue no poder realizar transferencias a ciertas cuentas, esto era debido a que si no se accedía al sistema con estas tarjetas, no aparecía en el fichero de *tarjetas.ubd* por lo que para el sistema no existía y daba error al intentar hacer una transferencia a dicha cuenta. De todas formas, ciertas tarjetas como la *00000000000000000006* y superiores, no se pueden acceder a ellas por error de pin.

En cuanto a la opción de venta de entradas, cuando quisimos realizar pruebas de esta funcionalidad para ver su correcta ejecución, no podíamos comprar entradas, ya que no aparecían, sin embargo el fichero de espectáculos no estaba vacío. Este "fallo" era debido a que el programa que se encarga de mostrar las opciones comprobaba que la fecha del espectáculo fuese superior a la fecha actual, y como eran espectáculos antiguos, no aparecían.

3. Conclusiones

En conclusión, esta práctica si bien ha tenido una curva de aprendizaje un tanto pronunciada al comienzo de la misma debido a la necesidad de aprender COBOL que es un lenguaje bastante diferente a lo que estamos habituados los miembros del equipo debido a su antigüedad, ha sido una buena primera toma de contacto con el lenguaje y nos ha permitido, tras varias horas de pruebas, documentación e investigación, adquirir una cierta soltura con el mismo, resultando en una experiencia bastante enriquecedora puesto que el lenguaje pese a su antigüedad sigue siendo utilizado en multitud de lugares.

4. Reparto de tareas

TAREA	HAYK	ENRIQUE	JOSÉ FÉLIX
INTALACIÓN DE COMPILADOR	1 h	1 h	1 h
CORRECCIÓN DE ERRORES	5 h	6 h	7 h
IMPLEMENTACIÓN DE FUNCIONALIDADES	7 h	8 h	8 h
PRUEBAS	8 h	8 h	7 h
MEMORIA	1.5 h	1.5 h	1.5 h

Referencias

- [1] *Cobol error codes*. URL: <http://www.escobol.com/modules.php?name=Sections&op=viewarticle&artid=21>.
- [2] *Lenguaje Cobol*. URL: <https://en.wikipedia.org/wiki/COBOL>.

[3] *Open Cobol*. URL: <https://en.wikipedia.org/wiki/GnuCOBOL>.