



# **Banksystem i C**

**Leslie Dahlberg** (*ldg14001*), **Mattias Johansson** (*mjn14021*)

Mälardalens Högskola

Akademien för Innovation, Design och Teknik

Programmering i C, HT14

DVA117

3 november 2014

# 1 Allmän beskrivning

## 1.1 Problemformulering

Syftet med detta projekt är att skapa ett program som simulerar ett banksystem som kan användas av både bankanställda och kunder.

Kunder ska kunna logga in i systemet och se sina konton och transaktioner samt göra överföringar mellan olika konton. De ska kunna begära ett nytt konto från banken samt kunna begära att ett konto stängs ner.

De bankanställda ska kunna lista information om sina kunder och deras konton, lägga till nya användare, ta bort användare och redigera användarnas personuppgifter. De ska kunna lista kundernas begäranden och skapa nya konton åt dem samt stänga ner konton. De ska också kunna se en log som förs över användarnas inloggningar.

All information ska sparas i en fil när programmet stängs ner och läsas in när det startas.

## 1.2 Antaganden och begränsningar

Programmet kommer att skrivas i programmeringsspråket C för plattformen Windows och kommer att köras i kommandotolken. Programmet kommer bara att använda sig av funktioner och bibliotek som är inbyggda i C.

# 2 Användargränssnitt

Programmet kräver inga externa bibliotek och kan byggas med valfri C-kompilator med stöd för språkets C99 standard. Programmet kan kompileras med GNUs gcc kompilator med hjälp av detta kommando:

```
gcc main.c bank.c xml.c
```

För att köra programmet krävs det att det finns en mapp kallad data i samma mapp som exe-file som innehåller en XML fil kallad data.xml. Innan programmet kan köras för den första gången måste denna fil skapas och den ska innehålla följande kod:

```
<FILE>
  <FILE_NAME>INFO
  <RECORD>
    <BANK_NAME>BANKEN
    <LAST_USER_ID>0
    <LAST_ACCOUNT_ID>0
  </RECORD>
</FILE>
<FILE>
  <FILE_NAME>USER
  <RECORD>
    <USER_ID>0
    <PERSONAL_NUMBER>0
    <USERNAME>admin
    <FIRST_NAME>
    <LAST_NAME>
    <ADDRESS>
    <USER_TYPE>admin
    <PASSWORD>admin
    <ACTIVE>1
  </RECORD>
</FILE>
```

För att använda programmet krävs en klaviatur. Programmet listar möjliga kommandon tillsammans med siffror och användaren kan sedan välja ett kommando genom att slå in respektive siffra och trycka på enter. Om användaren inte vill genomföra ett kommando kan han eller hon slå in # för att avbryta.

Första gången programmet körs finns bara en användare, den har användarnamnet och lösenordet admin och är av typen "bankanställd". Nya kunder kan registrera sig direkt i programmet genom att välja alternativ 2 i grundmenyn. Admin kan skapa nya användare och även bestämma om de ska vara av typen kund eller bankanställd.

### 3 Genomförande

Programmets källkod är uppdelad i tre c-filer (och respektive h-filer):

1. main.c
2. bank.c
3. xml.c

Det finns också en h-fil kallad shared.h som innehåller definitionerna för alla globala variabler. De globala variablerna innehåller data gällande användare, transaktioner, begäranden osv. samt information om vilken användare som är inloggad.

Main innehåller en loop som ger användarna tillgång till programmets grundläggande funktioner (inloggning, utloggning, registrering, avslutning av programmet, användarkommandon för kunder och bankanställda) och anrop till funktioner som läser från och skriver till datafilen data.xml.

Bank innehåller strukturer för all bankens data och funktioner för att manipulera datan (detta inkluderar även de grundläggande funktionerna som anropas i main.c).

Xml innehåller två funktioner, en för att läsa informationen från data.xml och en för att skriva information till data.xml. Programmet lagras datan från XML-filen i *structer* (info, user, account, transaction, request).

Se bilagan för hela källkoden.

### 4 Diskussion/slutsats

Under utvecklingen av programmet uppstod inga större problem. Ett få enskilda fel uppstod i samband med buffeöverskridning av strängar och skillnader mellan C kompilatorer på OS X och Windows.

I framtiden skulle ett grafiskt användargränssnitt kunna skapas för banksystemet för underlätta systemets användning och funktioner för att sätta in och ta ut pengar från konton skulle kunna läggas till. För att på ett bra sätt implementera detta skulle även ett hårdvarugränssnitt för intag och uttag av kontanter samt en server för att sköta bankdatabasen behöva skapas.