

Курсова работа по дисциплината

„Системно програмиране“

Вариант №8

Изготвил: Проверил:

Йордан Начев гр. 41 ас. В. Поляков

Ф.Н. 121213049

29.05.2016г.

**Задание**

8. Когато искате да изтеглите пари от банкомат, се прави трансакция от този банкомат към Вашата банкова сметка. Реализирайте две програми, като едната е самият банкомат - изпълнява всички операции: избор на език, сума, дали да има касов бон и PIN-код (проверката за него става в банкомата). Скрито за потребителя остава вземането на номера на картата (но той все пак се взима предвид). След като операцията е разрешена, се изпраща заявка до банковата сметка (сума и номер на сметката), където се прави проверка. Ако проверката върне грешка, се извежда съобщение на екрана (с цел улеснение - на банката). Информацията за клиентите се пази във файл и се зарежда в контейнер при всяко стартиране на програмата. Имайте предвид, че няколко банкомата могат да теглят пари от една банка едновременно (но за различни клиенти).

Подзадачи

Клиент-сървър моделът е най-подходящия метод за имплементация на заданието. Той позволява поддръжката на много клиенти (банкомати) едновременно чрез работата само на един централизиран сървър – банката. Тя предава актуални данни на всеки новосвързан банкомат, чрез прочитането им от файл и „складирането“ им в структура. Последва прочитането на данните от банкомата, респективно клиента и обработката им. Първоначално се избира език от стандартното входно устройство, последвано от избор на разписка с информация за транзакцията и въвеждането на IBAN-а на клиента. Последва обработка на входните данни, проверка в данните, пратени от сървъра – банката и обновяване на файла, съдържащ информация за банковите сметки.

**Планиране на отделните компоненти**

1. Сокет – ползва стандартните на своята същност библиотеки, а именно sys/socket, netinet/in, string, netdb, както и стандартната за I/O операции – stdio.h. Всички необходими параметри за инициализация на клиента/сървъра се дават при стартирането на всеки компонент – през командния ред. Клиента приема IP на сървъра, към когото ще се опитва да осъщестява връзка, последван от номера на порта, на който сървъра го очаква, разделени с интервал. Сървърът пък, от своя страна, очаква само един параметър, а именно порта, на който ще „слуша“ за клиенти. Функции като socket(), bind(), gethostbyname(), listen(), accept() са неразделна част от имплементацията на клиент-сървърната комунация.
2. Главно меню – реализира функционалността по настройка на език, въвеждането на данни за разписка и IBAN. Използва стандартни библиотеки като stdio и string. Функциите за преобразуване на входните параметри в такива, удобни за пресмятане, а именно в цели десетични стойности. Ограничението идва от добре пазното ни такова, че ATM машините не позволяват да изтегляме стотинки. ☺
3. Взимане на данните – чрез въвеждане на кода на дадена държава или езика, изписан на латиница програмата се превежда на съответния. Поддържат се 2 езика – Английски и Български. Разписката ще се/няма да се визуализира след транзакцията, ако потребителят е въвел Y/N. IBAN-ът на клиента се взима, чрез функция от библиотеката unistd – getpass() ни позволява да въвеждаме символи, които не се визуализират на екрана. Това позволява скриването на данните, които въвежда клиента по време на изтеглянето с цел подсигуряването на личните му данни и средства в съответната банкова сметка.
4. Преобразуване на входните данни в десетични – при въвеждането на сумата, която клиента възнемерява да изтегли се прави преобразуване на входните данни към десетични числа от тип int (максималната дневна сума, която даден клиент може да изтегли от банкомат, според Българското законодателство, е 400 лева), за да се улесни пресмятането. След отчитането на транзакцията се прави актуализация на записа в структурата от данни, която пази информация за банковите сметки, прочетени от банката.
5. Текущо време за разписка – ползва стандартна библиотека time.h. В нея взимаме информация за текущи дата и време.

**Имплементация на отделните компоненти**

1. Main() – в нея е имплементиран сокета за връзка със сървъра. Референции като sockaddr\_in, hostent, portno, sockfd и функции като gethostbyname, bind, listen, accept са неразделна част от неговата имплементация. Порта се приема като параметър от командния ред.
2. Transaction() – в нея първо се избира езика, а след това програмата се визуализира на съответния такъв. Има помощна променлива, която се ползва в случаите, когато се изпълняват функции, свързани с уведомяването на клиента в процеса на работа на програмата. От нея се извикват всички функции по взимане, преобразуване и визуализиране на данните. Функцията също така визуализира и разписката, ако е избрана такава. Извиква се функция getCurrentTime().
3. ConvertToInt() – приема входен параметър – число във вид на string. Взима се големината на низа и се преобразува всяко число, срещнато в него, чрез проверката за това дали е между ASCII кода на числата 0 и 9. Ако е, текущото число се събира с множителя, отговарящ за това, дали числото е в позицията на единица, десетица, стотица и т.н., умножен със символа (числото), извадено от ASCII кода на числото 0.
4. TreasuryNote() – взима написаното от потребителя и проверява, дали първия символ отговаря на символите Y/y/N/n и ако това е така, вдига флаг за разписка, ако не – съобщение за грешка.
5. getIBAN() – взима IBAN-а, въведен от потребителя и го съхранява в глобалната променлива iban.
6. calculateWithdraw() – записва последните 4 символа от всеки IBAN, предаван от сървъра – банката в буфериращ указател от тип char. Добавя терминиращата нула, преобразува го в цяло число, чрез функцията convertToInt() и след това проверява, дали има наличност в сметката за осъществяването на транзакцията или не. При достатъчна наличност транзакцията е успешна и сумата в банковата сметка се актуализира, в противен случай се извежда съобщение за грешка с помощта на флага за език, упоменат по-горе.
7. convertToString() – преобразува цяло число в низ от символи – цифри. Първо, преобразува десетичното число в низ от последователни символи, представени в ASCII код. В последствие променя съдържанието на стурктурата от данни, а след това актуализира списъка, където се пази информацията за клиентите на банката.
8. getCurrentTime() – прави се инстанция на структура, член на библиотеката time.h – tm. Взима се текущото системно време и в буфериращ низ от символи се записва информацията, съхранена в структурата.

**User Manual**

При стартиране на програмата тя очаква от клиента да въведе езика, на който иска тя да подава информацията. След това ще очаква от клиента да избере, дали иска разписка или не на съответния избран език. След избора му, програмата очаква от потребителя да въведе своя PIN код. След това програмата приключва изпълнението си, като обновява наличността по банковата сметка - в случай на успешна транзакция, а в случай на неуспешна, или друга грешка се извежда съобщение за грешка.

**Заключение**

В заключение, мога да кажа, че курсовата работа покрива основните изисквания по заданието, а именно симулирането на транзакция през банкомат. Старал съм се да използвам минимален брой библиотечни функциии и съм реализирал свои такива, които работят в контекста на програмата. Забележките по кода ще оставя на Вас.