**Технически университет - София**

**СИСТЕМНО ПРОГРАМИРАНЕ**

КУРСОВА РАБОТА:

“TEMA No 18”

Изпълнена от:

Студент: Шабанали Севтин Иса

Факултет: ФКСТ

Специалност: КСИ

Група: 47

Факултетен №: 121216063

Дата: 30.05.2019 Одобрил: .....................................

Град: София /ас. Десислав Андреев/

1. **Анализ на изготвеното приложение:**

Условие: Трябва да реализирате система за решаване на тестове. Идеята е следната: Клиентът се свързва със сървъра (вашата програма) и подава своето потребителско име. Сървърът (програмата) изпраща на клиента текста а задачата и възможни нейни отговори, от които клиентът трябва да си избере един (броят на отговорите може да е различен за различните задачи). Клиентът въвежда отговор, сървърът проверява дали е верен и препраща клиента към следващата задача. Предишните 2 стъпки се повтарят докато не се изчерпят задачите, предварително зададени във файл с определена структура. В края сървърът изпраща резултата на клиента: брой верни и брой грешни отговори, записва този резултат във файл и затваря връзката. Ако един клиент се опитва да реши теста повторно, трябва да му се позволи, но да се укаже специално, че той вече е попълвал теста и оценката му няма да се промени от повторно решаване.

Анализ: Потребителя трябва да има възможност да се логва в системата с потребителско име и/или парола. След логване, трябва да има възможност да стартира решаването на теста, като въпросите се прочитат от файл. При финализиране на решаването, трябва да се предостави избор на клиента да пререши теста или да напусне приложението. Резултатът, който е постигнал се записва в друг файл.

1. **Функционално описание на приложението**

Работата на приложението се извършва от главен и дъщерен процес. Създаването на дъщерния процес се извършва чрез системната функция fork (). При успешно Създаването се въръща стойност нула на дъщерния процес, а на родителя се връща идентификационния номер на създадения процес. При грешка функцията връща стойност минус едно, процес не се създава и програмата спира своето изпълнение.

За комуникация между процесите ще се използва named pipe. Причината е, че трябва да имаме двупосочна връзка между процесите. Процесът на четене на въпроси, след всеки отговор трябва да върне отговора, сървърът да провери дали той е отговорен правилно и да запише в профила на потребителя, който се е логнал резултата. Той се създава чрез функцията pipe(int fd[2]). При неуспешно създаване функцията връща стойност минус едно и програмата спира своето изпълнение. Каналите като цяло са лесни за употреба и поддръжка.

1. **Изпълнение на функционалностите**

**void choiceAction(int choice)**  - методът служи за управление на избора на потребителя. Спред веведеното чилсло, извиква определен метод.

**int startQuestionnaire (userStruct user)** – Методът служи за решаване на въпросника. Създава named pipe и започва комуникация. След четене и отговаряне на въпрос, резултатът се записва в поле на структурата user дали потребителя е отговорил правилно или грешно.

**int login ()** – методът служи за влизане в системата. Изисква се преди това да е бил създаден такъв профил. Методът отваря файл „users.txt“ и проверява дали там има записан потребител с въведените креденшъли. Ако има – се препраща към следващата стъпка. Ако няма се дава избор да въведе име и парола на ново или да се регистрира.

**int register()** – служи за регистриране на потребител в системата. След въвеждане на потребителско име и парлола, системата проверява дали във файла/базата има потребител с такова име. Ако намери такъв потребител – връща съобщение „this username is already in use. Please choose another one“, след което вика себе си. Ако потребителя не сществува – се създава такъв и се препраща към метода startQuestionnaire();.

**main ()** – главна функция, управляваща останалите и цялостното действие на програмата. При възникване на грешка връща стойност едно, а при нормално приключване – нула.

Линк към repository: **https://github.com/ShabanIsa/Questionnaire.git**