 **ТЕХНИЧЕСКИ УНИВЕРИТЕТ – СОФИЯ**

**ФАКУЛТЕТ ПО КОМПЮТЪРНИ**

**СИСТЕМИ И ТЕХНОЛОГИИ**

**КУРСОВ РАБОТА**

**по**

**СИСТЕМНО ПРОГРАМИРАНЕ**

**Изготвили**: **Ръководител:**

Валентин Георгиев ас. Десислав Андреев

ф.н. 121216025 гр. 46

София

2019

# Анализ на изготвеното приложение

## Условие на приложението

Животът е несправедлив за хората, които се опитват да влязат с чужди акаунти, на даден сървър. А сървърите са лоши и помнят! Реализирайте контейнер от поне 10 потребителя, които имат име и парола. Всеки, който направи връзка със сървъра носи своя IP адрес и сървърът го помни в логовете си.

Обаче не всеки потребител е добронамерен – направете така, че ако даден IP адрес се опита да се свърже с невалидни данни, към който и да е потребител поне 20 пъти, то този адрес да бъде записан в „черния списък“ – не му се дава повече възможност дори да въвежда име или парола (т.нар. credentials). Ако даден IP адрес се опита да достъпи конкретен профил с невалидни данни 10 последователни пъти, то се изпраща съобщение до администратора. Реализирайте задачата с две програми, като едната е клиент, а другата сървър.

**Допълнително:** нови потребители могат да се регистрират, като правят заявка към администратора. При одобрение, той създава акаунт и съобщава за това. Когато на сис-админа mu e lo6o, може да изтрива акаунти!

## Анализ

Трябва да се създаде приложение, което предоставя на обикновените потребителите регистрация и вход. Системните администратори може да изтриват акаунти. Приложението трябва да е разделено на две програми, едната да е сървър, а другата клиент.

**В програмата клиент:**

След като потребителят се свърже с програмата сървър, се изпраща текущия му IP адрес. Този адрес се проверява във файл наречен „черен списък“, ако адресът не съществува в този списък, на потребителя се дава възможност да продължи към приложението. Ако адресът съществува, на потребителя не се предоставя възможност да продължи.

Когато потребителят се регистрира попълва потребителско име и парола, след което данните му се записват в контейнер от програмата сървър. Когато потребител извършва вход в приложението, от потребителя се взима текущия му IP адрес. IP адресът се проверява във файл наречен „черен списък“, ако адресът не съществува в този списък, на потребителя се дава възможност да въведе потребителско име и парола. Ако адресът съществува, на потребителя не му се дава възможност да въвежда потребителско име и парола. Ако потребител от даден IP адрес се опита да достъпи конкретен профил с невалидни данни 10 последователни пъти се извежда съобщение в програмата сървър. Потребител се добавя към „черният списък“, когато въведе грешно поне 20 пъти входни данни от един IP адрес.

**В програмата сървър:**

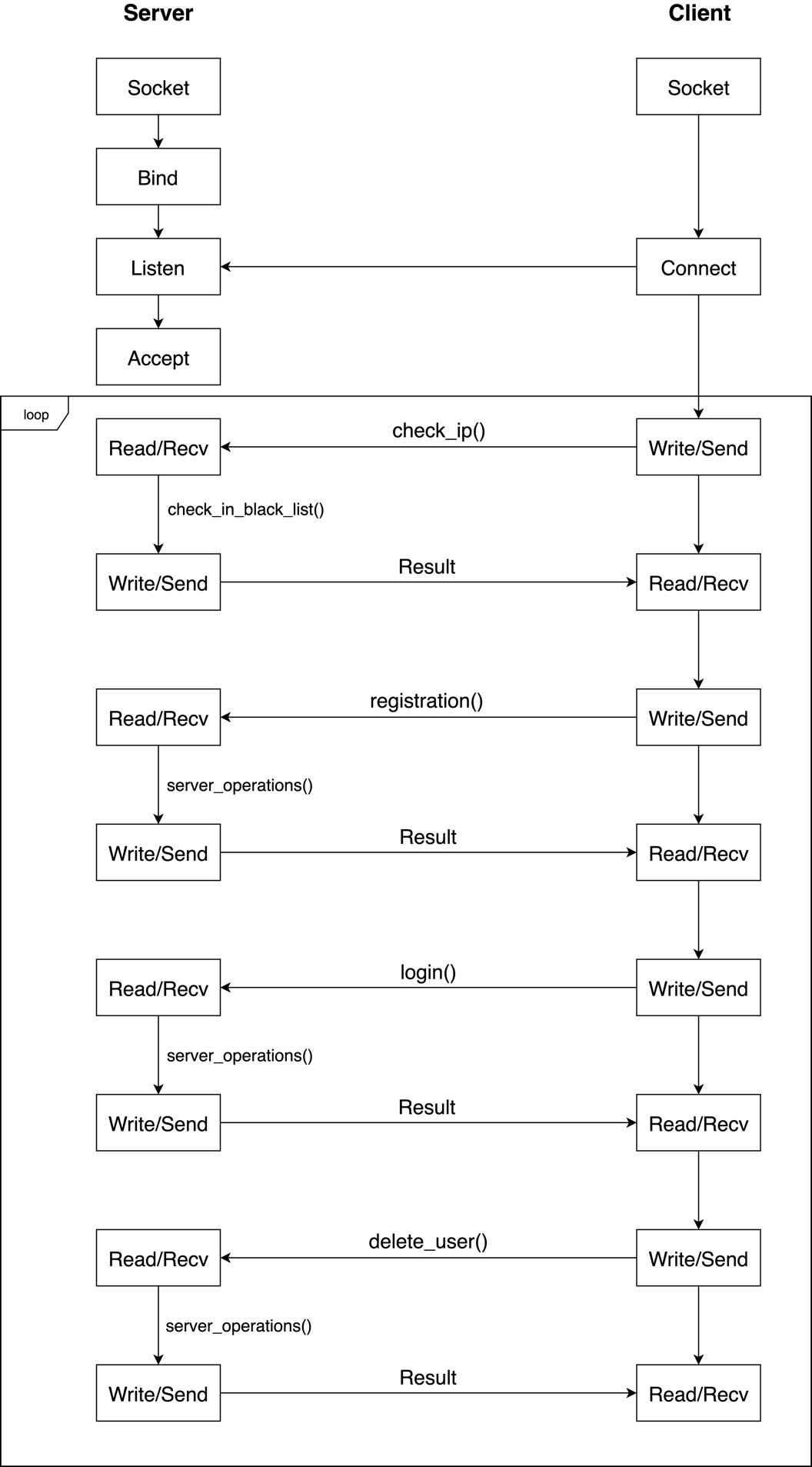
Когато потребител извършва вход в приложението данните от клиентската програма се получават, обработват се и се търсят в контейнера с потребители. Ако получените данни съществуват в контейнера данните се записват в лог файл със статус: „успешен вход“. Ако данните не са намерени, данните се записват в лог файла със статус: „неуспешен вход“. Лог файлът се обхожда и ако в логовете със статус: „неуспешен вход“ се намери един и същ IP адрес 20 пъти, този адрес се записва във файл наречен „черен списък“

# Функционално описание на приложението

От направения анализ произтичат следните изисквания към приложението:

* Потребителят да може да се регистрира в приложението;
* Потребителят да може да извършва вход в приложението;
* Системният администратор да може да изтрива потребители;
* Когато 20 пъти се въведат невалидни данни за който и да е потребител от един и същ IP адрес, този IP адрес да се добавя към „черен списък“ – не му се дава повече възможност дори да въвежда име или парола;
* Когато даден IP адрес се опита да достъпи конкретен профил с невалидни данни 10 последователни пъти, то да се изпраща съобщение до администратора.

**UML диаграма на приложението**

****

# Изпълнение на функционалностите

**В програмата сървър:**

* **void server\_operations(int sockfd) –** извършва комуникацията между клиента и сървъра. Тя осигурява главната функционалност на приложението: регистрация, вход, изтриване на потребител, добавяне към лог файл.

**параметри:**

* + - **sockfd** – файлов дискриптор, отнасящ се до слушания сокет
* **int get\_tries (char \*ip)** – брои колко пъти се среща даден IP адрес в лог файла със статус: „неуспешен вход“. Връща броя на срещанията на IP адреса.

**параметри**:

* **ip** – IP адресът, който трябва да се преброи колко пъти се среща в лог файла
* **does\_it\_exist\_in\_black\_list (char \*ip)** – проверява дали даден IP адрес се среща в „черния списък“. Връща *true*, когато се среща и *false*, ако не се среща.

**параметри:**

* + - **ip** – IP адресът, който трябва да се провери
* **void add\_to\_black\_list(char \*ip)** – добавя IP адрес към „черния списък“

**параметри:**

* + - **ip** – IP адресът, който трябва да се добави
* **bool check\_in\_black\_list()** – проверява дали в лог файла се среща 20 пъти IP адрес със статус „неуспешен вход“. Връща *true* ако се среща и *false*, ако не се среща.

**В програмата клиент:**

* **void registration(int sockfd)** – взима въведените регистрационни данни от потребителя и ги изпраща към програмата сървър.

**параметри:**

* + - **sockfd** – файлов дискриптор
* **void login(int sockfd)** – взима въведените входни данни от потребителя и ги изпраща към програмата сървър.

**параметри:**

* + - **sockfd** – файлов дискриптор
* **void delete\_user(int sockfd)** – взима въведеното потребителско име от системния администратор и го изпраща към програмата сървър.

**параметри:**

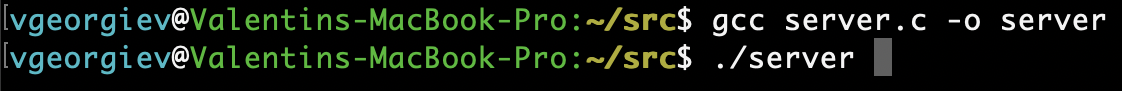
* + - **sockfd –** файлов дискриптор
* **void check\_ip(int sockfd)** – изпраща текущият IP адрес на потребителя към програмата сървър.

**параметри:**

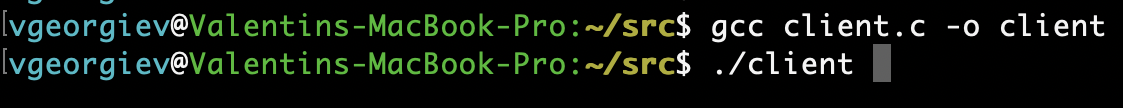
* + - **sockfd –** файлов дискриптор
* **void all\_users()** – извежда всички регистрирани потребители.
* **void read\_log\_file()** – извежда всички логове от лог файла.
* **void read\_black\_list()** – извежда всички IP адреси записани в черния списък

# Експериментални данни

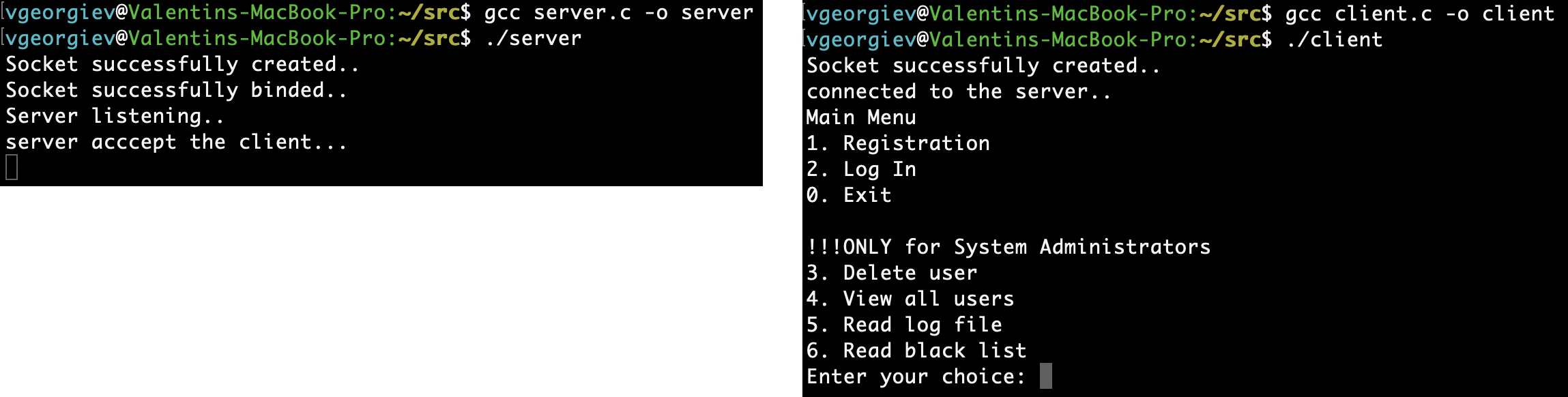
Компилиране и стартиране на програмата сървър:



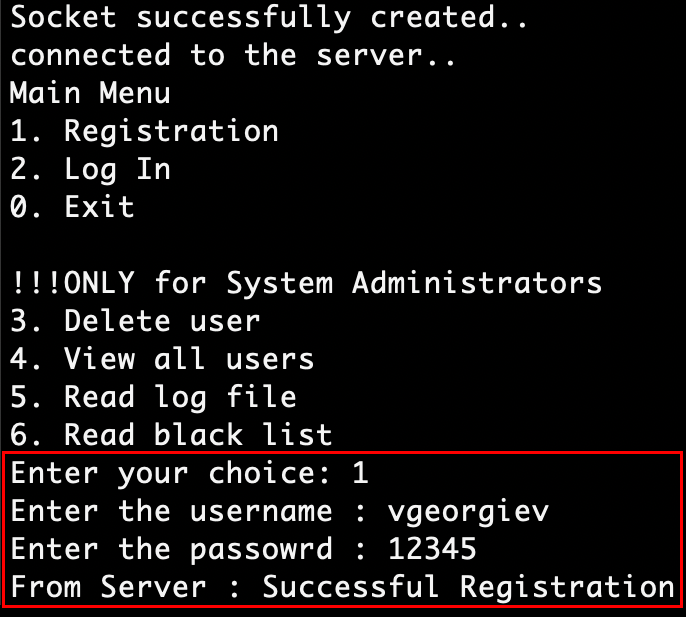
Компилиране и стартиране на програмата клиент:



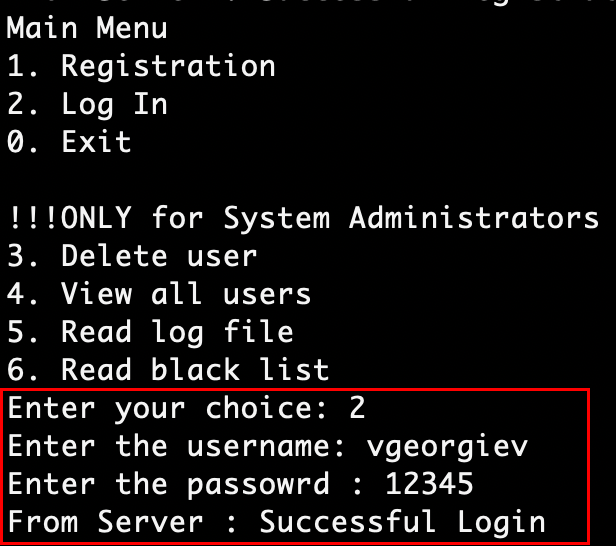
Когато се компилират и стартират програмата сървър и програмата клиент:



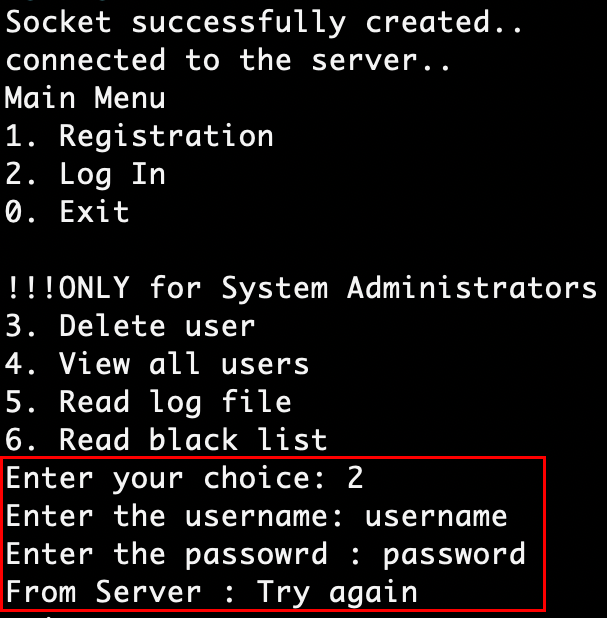
Когато се избере от менюто 1. Registration и се въведат регистрационни данни



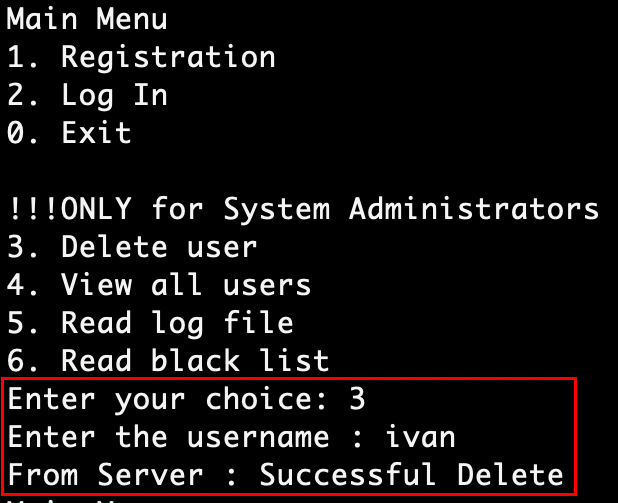
Когато се избере от менюто 2. Log In и се въведат правилни входни данни



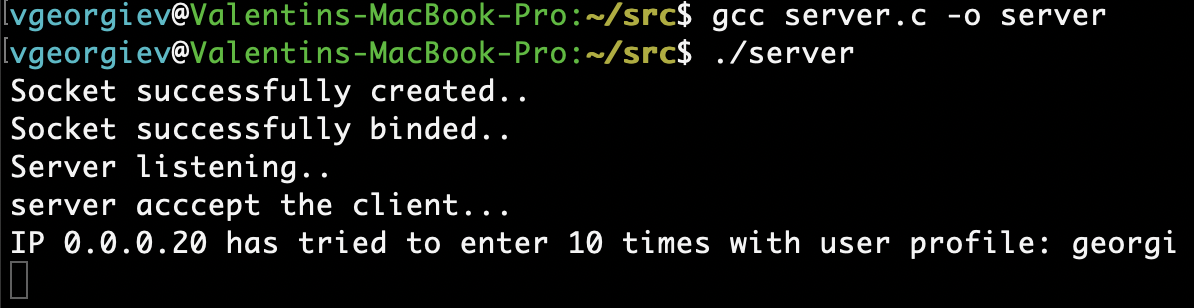
Когато се избере от менюто 2. Log In и се въведат неправилни входни данни



Когато системен администратор изтрие потребител



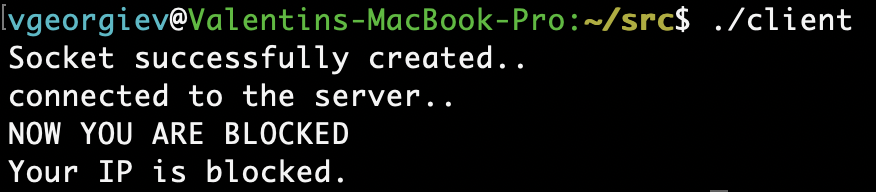
Когато даден IP адрес се опита да достъпи конкретен профил с невалидни данни 10 последователни пъти



Когато 20 пъти се въведат невалидни данни за който и да е потребител от един и същ IP адрес



# Когато потребител се опита да стартира програмата клиент, но IP адресът му е в „черния списък“



# Приложение

Кодът на разработеното приложение:

<https://github.com/valentingeorgiev/tu-system-programming>