

## Домашна работа 2 по Мрежово програмиране

**Име:** Анна Георгиева Ангелова  
**Специалност:** Компютърни науки  
**Факултетен номер:** 80306  
**Административна група:** 1  
**Курс:** 3

**Задача:** Напишете TCP мулти сървър и клиент за обмяна на файлове, като:

А) Използвате прозоречен интерфейс, за да зададете параметрите на заявката.

Б) Напишете Java сървлет, който ще играе роля на socket клиент и в края на операцията ще изпрати на браузера HTML страница със статуса на операцията.

### Решение:

За socket сървър използвам програмата от първото домашно с име server.c. Решението работи под Tomcat в средата на Linux. То съдържа всички компоненти на системата (изходните текстове на програмите, HTML файл) и текстов файл, който инструктира, как да се инсталира и използва системата и описанието на протокола за копиране.

### Стъпки:

- 1. Трябва да си стартираме tomcat от терминала.** Как става това? Влизаме в директорията, в която сме запазили tomcat и от там в директорията bin на tomcat-a, след това стартираме tomcat чрез ./startup.sh.
- 2. Трябва да си стартираме и server-a от терминала.** Как става това? Влизаме в директорията, в която сме запазили файла server.c. След това си правим exe файл на сървъра, това става с командата `cc -o server.exe server.c`. Стартираме си сървъра чрез ./server.exe.
- 3. Трябва да си направим WAR файл на Java проекта, който всъщност е нашият клиент.** Как става това? От текст менюто влизаме във File, след това избираме Export... Отваря ни се нов прозорец, от който си избираме дестинация: Web-WAR file. След това натискаме бутона Next>. Отваря ни се нов прозорец, в които в полето Web project - пишем името на нашия проект (напр.: homework2), а в полето Destination – пишем пътя, на който ще се запази нашия WAR файл (напр.: home/user/Desktop/homework2). След това натискаме бутона Finish. И така вече си имаме WAR файл.
- 4. Трябва да си deploy-нем WAR файл.** Как става това? Влизаме в browser-a и пишем localhost:8080. Отваря ни се страницата на Apache Tomcat. Отляво (под картинката на котенцето) в Administration избираме Tomcat Manager. Отваря ни се страницата на Tomcat Web Application Manager. Сега отиваме най-долу на страницата и в прозорчето Deploy, където пише War file to deploy, избираме кой WAR файл да бъде зареден, след това натискаме бутона Deploy. Като погледнем по-горе в прозорчето Applications ще видим че нашия файл е deploy-нат (напр.: с Path: /homework2 и DisplayName: homework2). По-горе в прозорчето Message: трябва да пише ОК, ако всичко е наред.
- 5. Трябва да пробваме дали работи всичко.** Как става това? Както сме в browser-a пишем localhost:8080/името на нашия проект"/името на нашия клас" (напр.:

localhost:8080/homework2/ClientServlet). Така ни се отваря прозорчен интерфейс, в който ще задаваме параметрите на заявката. В полето **Local file** – пишем пълното име на файла, който ще бъде копиран (напр.: home/user/Desktop/fileName), в полето **Remode file** – под какво име ще бъде записан (напр.:

home/user/Desktop/newFileName), можем да изберем една от двете опции: **Download** или **Upload**, като ако сме отбелязали Download – означава че сървъра ще изпрати файла fileName на клиента, който ще го запише под името newFileName, а ако сме избрали опцията Upload, тогава клиента ще изпрати файла filename на сървъра, който ще го запише под името newFileName. Натискайки бутона Copy – се изпълнява операцията, а с бутона Reset се изчистват полета Local file и Remode file.

**6. Трябва да разберем, ако е възникнала каква грешка или ако операцията е завършила успешно.** Как става това? След като сме натиснали бутона Copy се отваря нов прозорчен интерфейс, в който ни излиза съобщение, което показва как е завършил процеса:

а) Ако е успешно – излиза съобщението **EVERYTING IS OK! CREAT NEW FILE;**

б) Ако е накое от полета Local file или Remode file е празно – излиза съобщението **ONE OF FIELD IS EMPTY!!!;**

в) Ако има проблем с файл – излиза съобщение на това от какъв тип е грешката (напр.: **IOExseption**);

Във всеки един от случаите под съобщението има връзка “**back**”, кликайки върху нея ще се зареди прозорчения интерфейс за задаване на нова заявка.

**7.Трябва да разберем какво става със сървъра.** Как става това? Отиваме в терминала, в който е стартиран сървъра и там ще имаме съобщения от вида:

а) **Server waiting ...** – означава че сървъра чака клиента да се свърже с него;

б) **File fileName can't be created!** – означава че сървъра не може да създаде файл с такова име;

в) **File fileName can't be opened!** – означава че сървъра не може да отвори файл с такова име;

г) **Client can't create file!** – означава че клиента не може да създаде файла;

д) **Client can't open file!** – означава че клиента не може да отвори файла;

е) **File fileName is open!** – означава че файла fileName е отворен за четене или писане в зависимост от това каква опция сме избрали – Upload или Download;

ж) **File fileName is closed!** – означава че файла fileName е затворен;

з) **Operation completed!** – означава че операцията е завършила;

**П.С.:** в Java кода в метода doGet(...) в реда FileReader fileReader = **new** FileReader("/home/anki4/Desktop/HomeWork2.html"); трябва да се промени пълното име на html файла (който служи за прозорчения интерфейс) в зависимост от това къде е запаметен;