Електротехнички факултет

Универзитета у Београду

Конкурентно и дистрибуирано програмирање

Задатак: Дистрибуирани систем за синхронизацију докумената

***Садржај****:*

[***Домаћи задатак август 2019 3***](#_Toc18246659)

[***Опис рјешења 5***](#_Toc18246660)

[***UML дијаграм 6***](#_Toc18246661)

[***Упутство*** *7*](#_Toc18246662)

[***За мање искусне кориснике:*** *7*](#_Toc18246663)

[***За искусне кориснике:*** *14*](#_Toc18246664)

# **Домаћи задатак август 2019**

**Дистрибуирани веб сервери**

Пројектовати дистрибуирани рачунарски систем који треба да омогући синхронизацију докумената на већем броју веб сервера. Програм треба да ради у систему који се састоји од више рачунара повезаних у LAN (*Local Area Network*) или WAN (*Wide Area Network*).

У систему постоји три типа програма:

1. Централни сервер који служи за чување оригиналних верзија докумената и информација о распрострањености копија докумената.

2. Подсервери на којима се чувају копије докумената, чији се оригинали налазе на серверу.

3. Клијентски програм преко кога клијент може да приступа документима на подсерверима.

Када клијентски рачунар пошаље захтев за неким документом неком од подсервера, тај подсервер проверава да ли се тражени документ налази код њега. Уколико се документ налази на подсерверу, тада подсервер кориснику враћа тражени документ. Уколико се документ не налази на подсерверу, онда подсервер од централног сервера тражи да му пошаље копију документа. Када на подсервер пристигне копија документа, она се памти на подсерверу, а такође се прослеђује и кориснику који је тражио документ.

Централи сервер периодично мења садржаје оригиналних докумената. Потребно је обезбедити одржање лабаве конзистентности на следећи начин. Када централни сервер утврди да неки документ треба променити он интерно провери на којим се све подсерверима налази тражени документ. Када је сервер утврдио листу подсервера, онда шаље свим подсерверима наредбу за брисање копије документа. Да се на серверу не би стално креирале нити које служе за слање захтева за брисањем докумената, на централном серверу постоји група нити која обавља овај посао. Број нити је параметар који се задаје приликом рада серверске апликације. Централни сервер не шаље захтеве за брисањем директно нитима већ их убацује у јединствени бафер захтева за брисањем докумената. Нити узимају захтев по захтев из бафера и прослеђују га одговарајућем подсерверу у редоследу у коме су и пристизали. Када се са свих подсервера обрише копија докумената информација о успешном брисању се записује у лог.

Уколико је подсервер недоступан у моменту када се упућује захтев за брисањем покушава се поново након *m/3* секунди. Уколико ажурирање свих подсервера на којима се налази копија документа не успе у интервалу од *m* секунди, тај захтев се проглашава необрађеним и то се уписује у лог. Притом се у лог уписују и подсервери на којима није успело ажурирање, али се поступак слања наставља све док се подсервер сматра исправним. Подсервер на коме од тренутка покретања није успело *k* захтева за брисањем проглашава се за неисправан и чека се његово поновно покретање. Уколико се неки подсервер не одазове извесно време (*x* секунди) такође се проглашава за неисправан. Неисправном подсерверу се не шаљу захтеви за брисањем, нити се одговара на захтеве за страницама док се он не јави централном серверу и потврди да је поновно покренут. Јављање да је подсервер покренут треба реализовати на тај начин што сваки од подсервера од сервера тражи неки одређени документ. Приликом покретања, подсервер брише све копије довучених докумената.

Водити рачуна да подсервери и сервер моду да приступају само документима који се налазе почев од специфицираног директоријума, односно да није дозвољено да се приступа документима који су изван специфицираног директоријума. Специфицирани директоријум може да има структуру произвољног нивоа дубине. Уколико тражени документ не постоји на подсерверима и серверу кориснику треба вратити страницу/документ са стандардном поруком да се јавила грешка 404. Ову поруку подсервер треба да врати клијенту и у случају да је централни сервер недоступан, а да не поседује копију траженог документа. Захтев за изменом садашње верзије документа, односно постављањем новог документа, корисник задаје директно централном серверу преко одговарајуће странице.

Проблем решити користећи мрежну комуникацију у програмском језику Јава. За сваки од ова три типа рачунара треба да постоји одговарајући графички кориснички интерфејс (GUI треба да буде развијен користећи Java **SWING** компоненте).

**Напомена**: Целокупна комуникација требало би да иде разменом текстуалних порука. Поруке треба да буду стандардне HTTP GET, PUT и DELETE поруке (http://www.w3.org/Protocols/rfc2616/rfc2616-sec5.html). Ово значи да подсервери и сервер треба да одговарају на захтеве које може да упути и програм за преглед страница (Chrome, Internet Explorer, Firefox, Opera, …)

# **Опис рјешења**

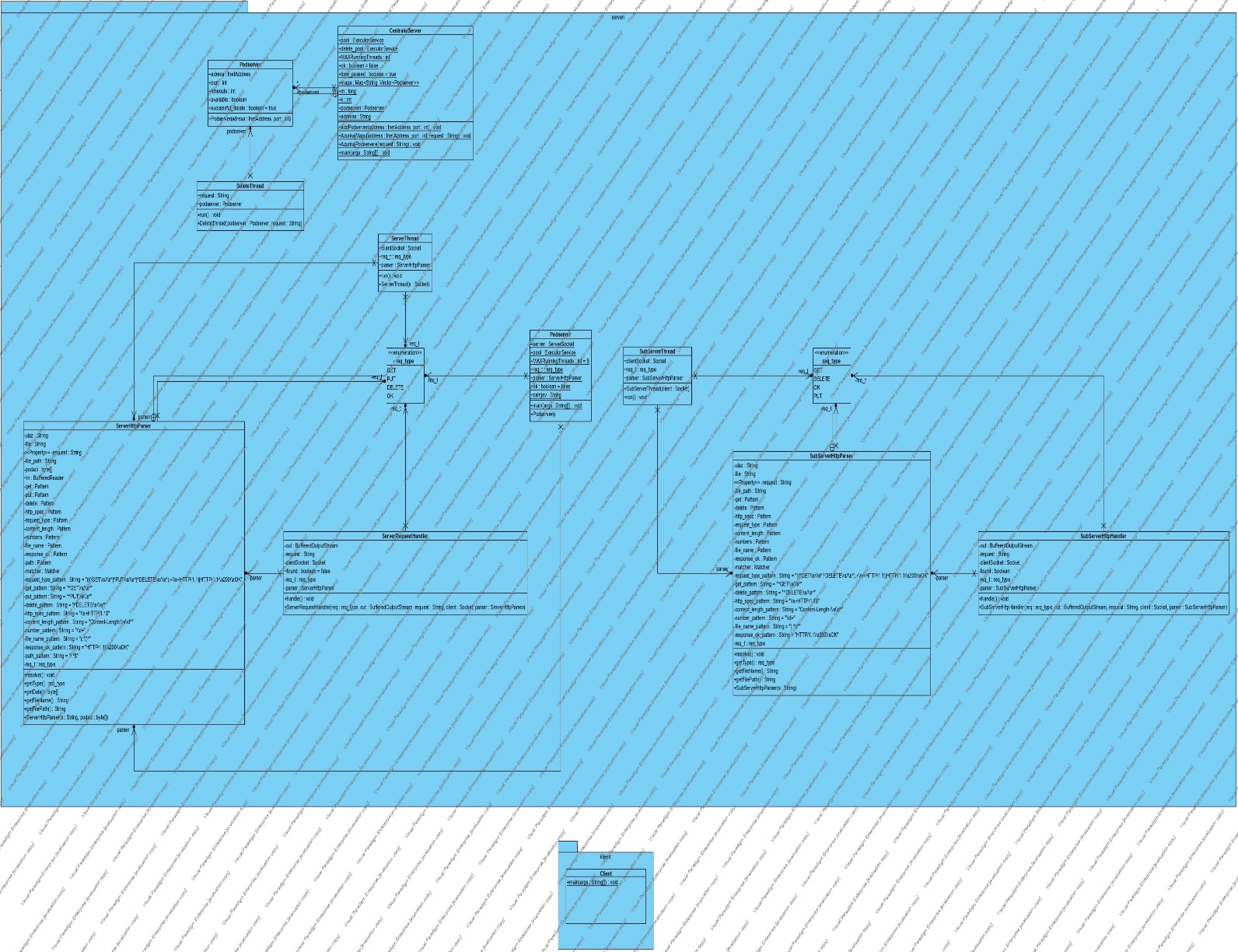
Рјешење се састоји из 2 пакета и 10 класа, од којих 3 садрже главну (main) методу која се позива приликом покретања програма. Основа рјешења је комуникација путем прикључница и помоћних Java класа које служе за креирање HTTP захтјева. Клијент за захтјев који му је прослијеђен преко графичког интерфејса, ствара HTTP GET или HTTP PUT захтјев који се шаље подсерверу (HTTP GET) или централном серверу (HTTP PUT) који је специфициран адресом. У наставку је описан HTTP GET захтјев:

Подсервер прима тај захтјев и провјерава да ли се дати документ налази у његовом фајл систему. У случају да пронађе такав документ одговара одмах на захтјев клијенту и шаље му тражени документ, у супротном шаље захтјев централном серверу који такође провјерава да ли има тражени документ код себе. У случају да централни сервер има тај документ, враћа га подсерверу, који га затим просљеђује клијенту, а у случају да не посједује тај документ, шаље „404 Not Found“ код о грешци који се затим просљеђује клијенту. Сервер таокђе памти сваки подсервер који му је упутио HTTP GET захтјев.

Клијент такође има могућност да шаље нове документе или новије верзије постојећих докумената директно централном серверу коришћењем HTTP PUT захтјева. Сервер шаље „200 OK“ одговор и упише послати фајл код себе притом бришући стару верзију фајла. Сервер затим свим подсерверима које је детектовао (примивши HTTP GET захтјев), а који такође посједују дати фајл (те информације се чувају у једној хеш табели), шаље HTTP DELETE захтјеве, сигналирајући им да обришу постојеће верзије докумената.

Сервер и подсервере је потребно конфигурисати тако да отварају прикључницу која ослушкује на IP адреси рачунара на којем су покренути како би систем функционисао преко LAN и WAN мреже.

# **UML дијаграм**



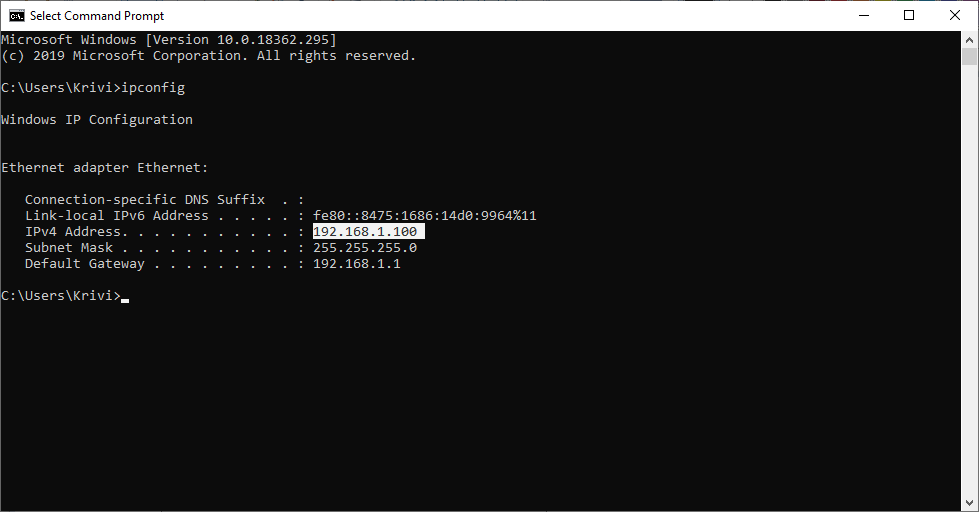
# **Упутство**

## **За мање искусне кориснике:**

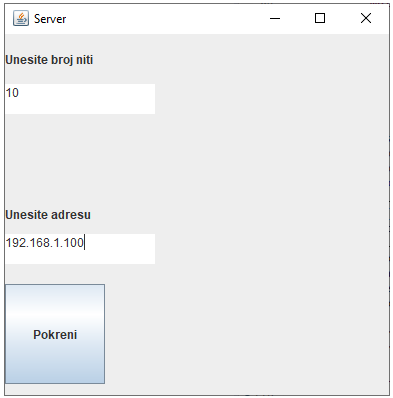
Отворити пројекат у Java Eclipse IDE (или креирати нови пројекат користећи изворне .java фајлове). Покренути неку од 3 апликације.

* Централни сервер:

У првом простору за текст изнад којег пише „Unesite broj niti“ унијети максимални број нити које ће централни сервер користити. Затим покренути на Windows систему командну линију и у њој команду ipconfig, ископирати IPv4 адресу у простор за текст у графичком прозору централног сервера изнад којег пише „Unesite adresu“. На крају, притиснути дугме и сервер ће бити покренут на порту 8080 и ослушкиваће захтјеве.

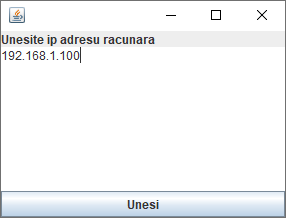
 Слика 1: Командна линија и команда за

налажење адресе

 Слика 2: Сервер са примјером коришћења

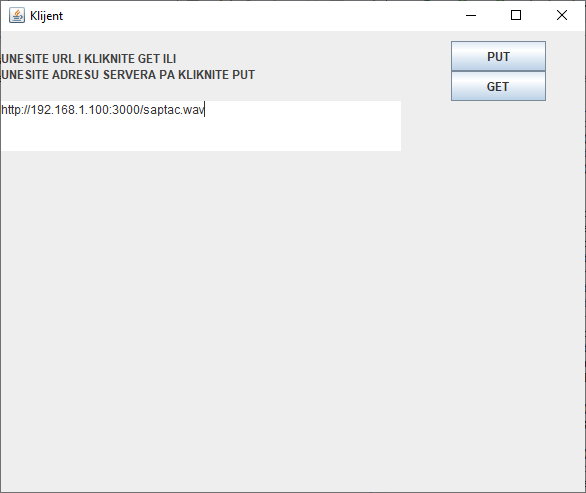
* Подсервер:

Слично као код централног сервера са разликом у томе што се не уноси број нити за захтјеве. Портови на којима могу подсервери да ослушкују су у опсегу од 3000 до 4500 са кораком 30.

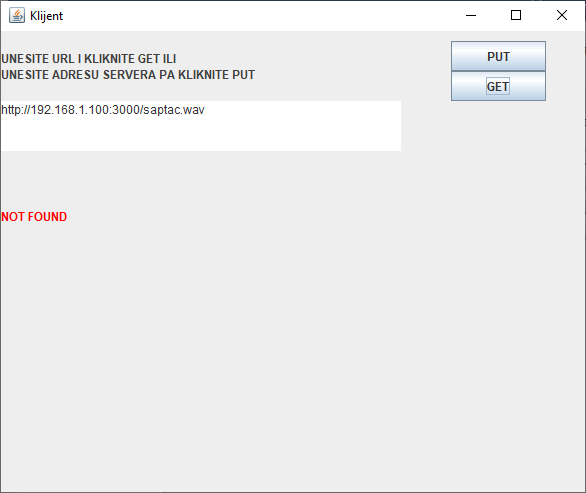
 Слика 3: Подсервер са примјером

* Клијент:

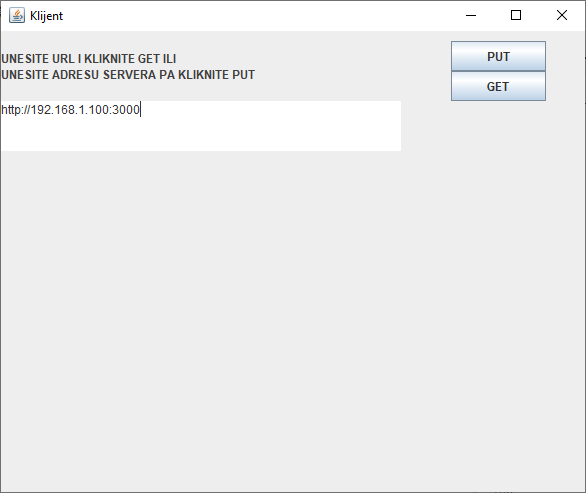
У простору за текст унијети или адресу и потпуну путању до фајла на серверу или адресу централног сервера на којем се жели поставити фајл. Ако се захтјева довлачење фајла, након уноса адресе сервера + путање до фајла притиснути дугме „GET“, а ако се поставља фајл до сервера, након уноса адресе сервера, притиснути дугме „GET“ и затим ће се појавити прозор за селекцију фајла који се жели послати на сервер. Након извршења захтјева појављује се порука о успјешности.

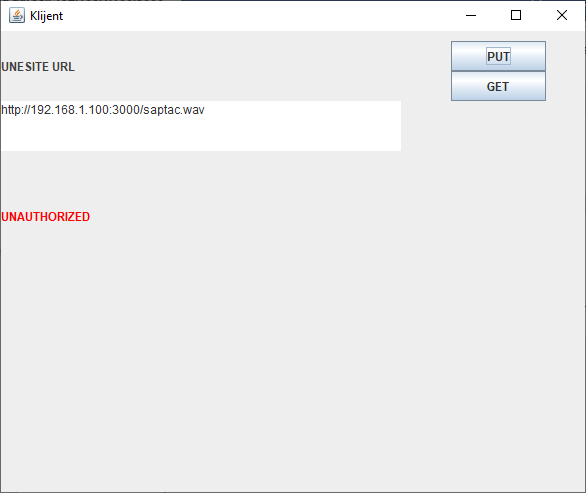
 Слика 4: Примјер коришћења клијента са путањом за GET захтјев

Слика 5: Одговор на претходни захтјев (у случају да сервер има тражени фајл)

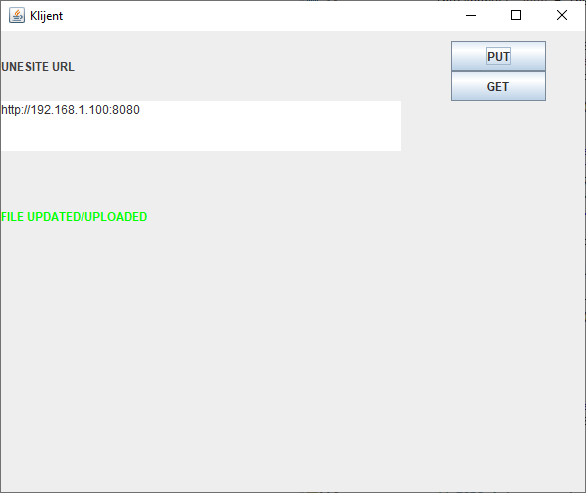


Слика 6: Одговор на претходни захтјев (у случају да сервер нема тражени фајл)

 Слика 7: Примјер коришћења клијента са путањом за PUT захтјев



Слика 8: Одговор на претходни захтјев (Није одобрен захтјев због покушаја уписа на подсервер)

Слика 9: Одговор на претходни захтјев у случају да се пормијени порт на 8080 који је серверски. У овом случају и сервер и подсервер се налазе на истом рачунару

## **За искусне кориснике:**

Исто важи као и за мање искусне кориснике, са разликом да се код централног сервера могу мијењати параметри м и к који су везани за timeout подсервера, такође могуће је мијењати ручно број нити за обраду код подсервера.