ПРОЈЕКТНИ ЗАДАТАК

ЛОГИЧКИ

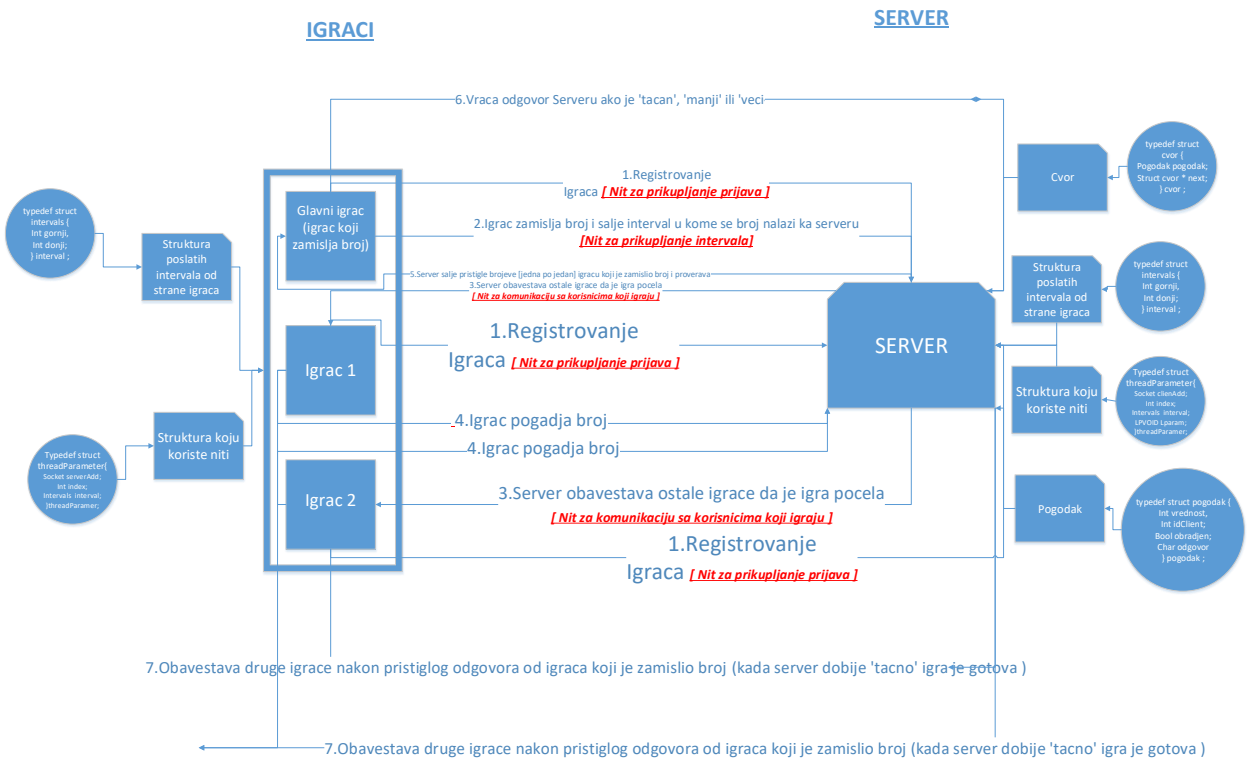
НИКОЛА МИЉКОВИЋ ПР 54-2017

СТЕФАН ШТРБАЦ ПР 87-2017

УВОД

Развијена је игрица погађања бројева. Игра се састоји из две компоненте : сервера и неодређеног броја играча. Игра је замшљена тако да постоји главни играч који замишља број, а остали играчи треба да преко сервера погађају који број је тај играч замислио. Како би се број могућности које је главни играч замислио смањио, главни играч шаље интервал у ком се његов замишљени број налази. Када неки споредни играч погоди тачан број који је главни играч замислио, сервер обавештава остале играче да је неки од играча погодио тачан број и игра се завршава. Уколико број који је споредни играч послао није онај који је главни играч замислио, сервер враћа одговор играчу који је послао тај број (да ли је замишљени број већи или мањи од послатог) и играч наставља са погађањем.

ДИЗАЈН ПРОЈЕКТА



ИМПЛЕМЕНТАЦИЈА

Задатак је имплементиран коришћењем неблокирајуће ТЦП конекције. У пројекту постоје две основне компоненте : сервер и неодређени број играча.

Ток комуникације са серверске стране изгледа овако:

- Иницијализије се више *acceptedSockets* преко којих се играчи конектују.
- Направи се *listenSocket*, уради се *Bind* и стави се у неблокирајући режим и он чека нову конекцију.
- Иницијализује се *READ_FD* структура, сокети се ставе у ту структуру и позове се select.
- Ако је select прошао, први слободан acceptedSocket прихвата конекцију новог клијента која је стигла.
- За сваки acceptedSocket се креира одговарајућа нит (сваки играч има своју нит на серверској страни).
- Пројекат је тако замишљен да први играч који се конектује („са нултим индексом“) увек буде главни и он замишља број, а остали играчи погађају.
- Када се конектује први играч (онај који је замислио број) креира се нит *threadParametarMain*, а уколико је у питању неки од играча који погађају креира се *threadParametarSide*.
- *ThreadParametarMain* је стркутура у којој се чувају све потребне информације које ће бити неопходне за извршавање нити (сокет за комуникацију, који је индекс, адреса клијента и број чворова (број чворова у листи коју треба обрадити)) и она се прослеђује као параметар приликом креирања нити.
- Свакој нити је придружен и један семафор. На почетку семафор је сигнализарн и прва се покреће нит која извршава функцију *acceptThreadMainPlayer* која је задужена за комуникацију сервера и главног играча.
- Када дође до следећег захтева за конекцију (то је сигурно захтев играча који погађа) позива се функција *acceptThreaSidePlayer* која исто има свој семафор (пошто има више играча креирана је структура од Н нити и Н семафора за те споредне играче).

Ток комуникације са клијентске стране :

- Сваки играч имат *connectSocket,* након постављених података одради се *connect*.
- На почетку имамо позив *recive-a* који чека од сервера свој индекс (сервер након конекције сваком пошаље индекс) и на основу њега се одређује да ли је у питању главни или споредни играч.
- Ако је у питању главну играч (индекс 0) он уноси интервал у коме се налази његов замишњени број (доња и горња граница) и примењује се слична логика као са серверске стране. Креира се нит *threadParametarMain* која се везује за функцију *acceptThreadMainPlayer*.
- Ако није у питању главни играч (индекс који је различит од 0), њему кажемо да се успешно регистровао и да треба да сачека да игра почне. Слична логика као и код главног играча, креира се нит и везује се за функцију *acceptThreaSidePlayer*.

Ток комуникације током трајања игрице:

- Након успостављене иницијалне конекције и након што је главни играч унео и послао интервал ка серверу (сервер га је прихватио), сервер обавештава редом нити које комуницирају са играчима да пошаљу интервал ка њима како би они знали да игра почиње (позива се *RelaseSemaphore* и она обавештава прву нит која чека да пошаље интервал). Након тога његова нит одлази на чекање (позивањем *WaitForSingloObject)* док јој не стигне неки од предлога који је споредни играч послао.
- Након што се семафор споредног играча сигнализирао, сервер сачува интервал и шаље га свом играчу (свака нит пошаље свом). Након тога улази у петљу и чека предлог тачног броја од играча.
- Приликом слања предлога споредни играч користи једну од 2 могуће опције за погађање: погађање случајним путем или применом алгоритма средње вредности. Након што се његов предлог израчуна, он га шаље серверу.
- Када тај предлог стигне на сервер, сервер прихвата предлог и поставља га у листу предлога које треба обрадити. Улази у критичну секцију и додаје тај чвор у листу чворова које треба обрадити.
- Проверава да ли је крај игре, уколико није сигнализира главној нити да постоји нови предлог који је потребно обрадити. Ова нит затим чека на свом семафору да захтев буде обрађен.
- Сервер узима први необрађени предлог из листе и шаље га свом клијенту који је замислио број.
- Са клијентске стране, нит за главног играча прима поруку коју је сервер послао. Смешта број у *receivedNumber* и врши проверу.

- Уколико је тај број онај који је он замислио у *databuffer* ставља *„тачно“, односно ако није ставља „веће“ или „мање“* и шаље серверу одговор. Затим се понавља чекање за следећи број – петља која траје док број не буде погођен.
- Сервер прихвата одговор, уписује га и означава да је предлог обрађен. Уколико је пристигао одговор „тачно“ игра се завршава. Уколико није обавештава клијента који чека да тај захтев буде обрађен, да је обрађен (позива *RelaseSemaphore* за тог клијента).
- Затим се одговор прослеђује клијенту који је послао предлог и ако није погодио број границе интервала се мењају зависно од одговора. Након тога он поново покушава да погоди број (по претходно описаном поступку).
- Игра се понавља све док један од играча не пошаље тачан број. Када се број погоди играч који је погодио број се информише да је баш тај број био тачан (односно да је он победник), док се остали играчи обавештавају да је дошло до краја игрице.