

# Matrix Solver

## Domaći 1

### 1 Pregled zadatka

Matrix Solver sistem treba da podrži množenje unapred zadatih matrica proizvoljnih veličina. Ova operacija treba da se obavlja konkurentno, uz mogućnost dodavanja novih matrica kao i pregleda rezultata pojedinačnih matrica kao i prikazivanje svih.

Parametri sistema:

- Sistem treba da radi konkurentno.
- Sistem treba implementirati u Java programskom jeziku.
- Sistem je podeljen na nekoliko komponenti, s time da treba da bude moguće dodavati nove komponente u budućnosti relativno lako. Zahtevi za komponente i njihova međusobna komunikacija su dati u **odeljku 2**.
- Bitno je da sistem informiše korisnika kada dođe do problema, kao i da nikada ne kolabira u potpunosti. Zahtevi vezani za razrešavanje grešaka u sistemu su dati u **odeljku 3**.
- Korisnik interaguje sa sistemom preko komandne linije (CLI), unošenjem komandi. Pored toga, rad sistema će se podešavati preko konfiguracione datoteke. Opis ove datoteke, kao i spisak komandi, njihovih parametara, rezultata i mogućih grešaka, uz nekoliko primera za korišćenje sistema biće prikazani u **odeljku 4**.
- Bodovanje zadatka, kao i instrukcije za predaju zadatka su dati u **odeljku 5**.

## 2 Opis sistema

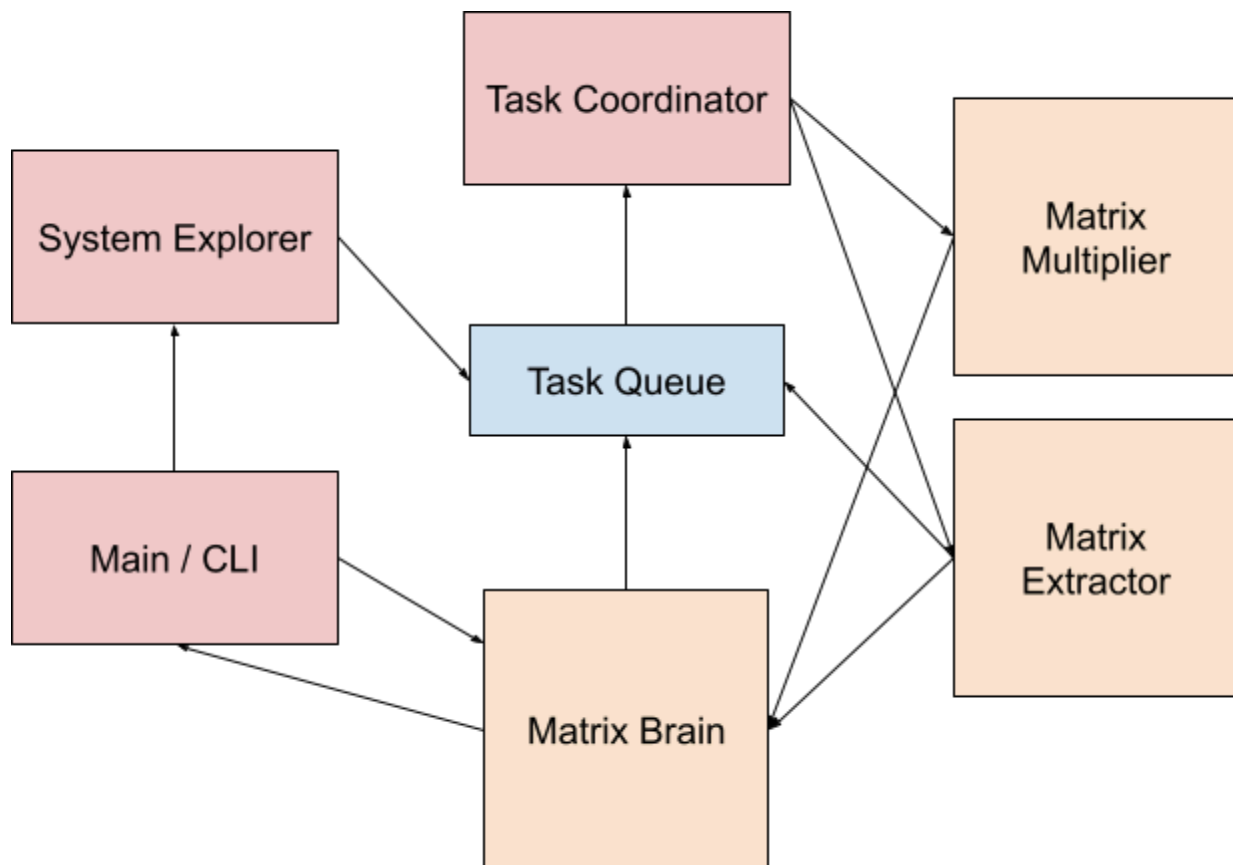
Sistem se sastoji iz tri komponente koje su zasnovane na thread pool-u, i nekoliko pomoćnih komponenti koje se izvršavaju svaka u svojoj niti. Svoj thread pool imaju:

- Matrix Multiplier
- Matrix Extractor
- Matrix Brain

Dok se u pojedinačnim nitima izvršavaju:

- Main / CLI
- Task Coordinator
- System Explorer

Pored ovih aktivnih komponenti, postoji i jedan deljeni red (TaskQueue) koji se koristi za zadavanje novih poslova i njihovo pokretanje. Sistem se grafički može predstaviti na sledeći način:



## 2.1 System Explorer thread

Komponenta System Explorer služi kao ključni deo sistema za dinamičko pretraživanje i identifikaciju tekstualnih fajlova unutar zadatog skupa direktorijuma i poddirektorijuma, sa ciljem pronalaska fajlova koji sadrže matrice. Funkcionalnosti ove komponente su:

**Pretraživanje Direktorijuma:** System Explorer aktivno pretražuje sve navedene direktorijume i njihove poddirektorijume na osnovu podataka dobijenih iz konfiguracione datoteke. Main/CLI komponenta može dodatno da navodi direktorijume za pretragu kroz posebne komande, koje System Explorer potom obilazi.

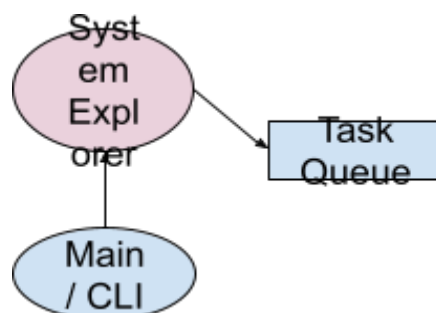
**Identifikacija Tekstualnih Fajlova:** Fokus pretrage je na pronalaženju tekstualnih fajlova, specifično onih sa ekstenzijom *.rix*. Pretpostavka je da će ovi fajlovi sadržati matrice koje su relevantne za dalju obradu.

**Kreiranje i Raspodela Zadataka:** Kada naiđe na *.rix* fajl, System Explorer kreira novi zadatak (Task) za detaljnu analizu sadržaja tog fajla. Ovaj zadatak se potom dodaje u Task Queue, gde čeka na obradu od strane drugih komponenti sistema.

**Detekcija Novih i Izmenjenih Fajlova:** Osim inicijalnog pretraživanja, System Explorer kontinuirano prati pojavu novih fajlova ili izmene na postojećim fajlovima unutar obuhvaćenih direktorijuma. Ova funkcionalnost omogućava ažuriranje baze potencijalnih matrica u realnom vremenu.

**Upravljanje Duplikatima i Izmene:** Za svaki pronađeni fajl, pamti se "last modified" vrednost. Ako fajl nije izmenjen od poslednjeg čitanja (tj. "last modified" vrednost se nije promenila), ne treba kreirati novi zadatak, čime se izbegava nepotrebna obrada. Međutim, ako je došlo do promene u fajlu, System Explorer treba da osigura da se učitaju samo izmenjene matrice, sprečavajući dodavanje duplikata. Potrebno je ukloniti stare verzije izmenjenih matrica i dodati nove (ili izmeniti stare vrednosti novim). Svaka matrica je interno "povezana" sa fajlom iz kog se pročitala.

**Pauza i Ponovno Pretraživanje:** Nakon završetka jednog ciklusa pretraživanja, System Explorer pauzira na određeno vreme, kako je dato u konfiguracionoj datoteci. Nakon pauze, proces se ponavlja, omogućavajući redovno ažuriranje i detekciju novih ili izmenjenih fajlova.



## 2.2 Task Queue

Task Queue je strukturirani blokirajući red koji služi kao centralno mesto za upravljanje zadacima u sistemu. Ovaj red organizuje i raspoređuje poslove koji treba da budu izvršeni, pri čemu svaki posao može biti definisan preko interfejsa Task (u vašoj implementaciji ne mora biti interfejs). Ispod je predlog kako Task može da izgleda.

```
public interface Task {  
  
    TaskType getType();  
  
    // M_Matrix matrixA, matrixB;  
  
    // File potentialMatrixFile;  
  
    Future<M_Matrix> initiate();  
  
}
```

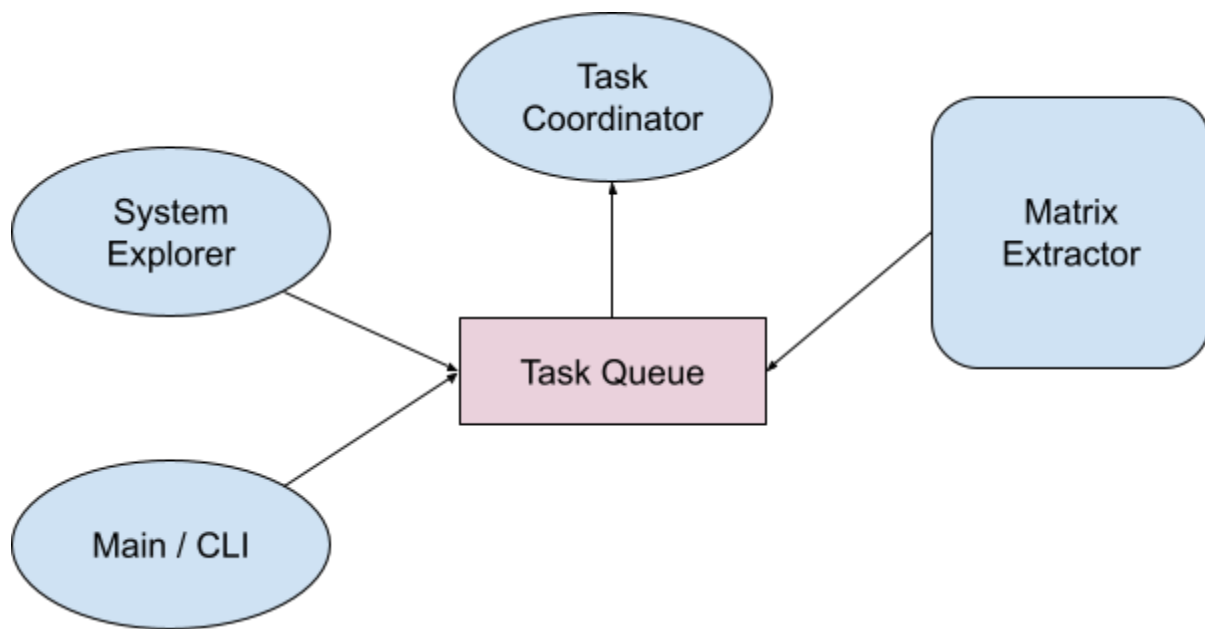
**TaskType:** Ova enumeracija (CREATE / MULTIPLY) određuje tip zadatka koji treba obaviti. CREATE se odnosi na kreiranje matrice iz tekstualnog fajla, dok MULTIPLY implicira množenje dve prethodno definisane matrice.

**initiate():** Metoda initiate odgovorna je za pokretanje zadatka unutar odgovarajućeg skupa niti (thread pool). Izvršavanje ove metode vraća Future<M\_Matrix> objekat, koji predstavlja asinhroni rezultat operacije - bilo to učitavanje matrice iz fajla ili rezultat množenja dve matrice. M\_Matrix je objekat koji predstavlja konačnu matricu nakon operacije.

Integracija sa Komponentama Sistema:

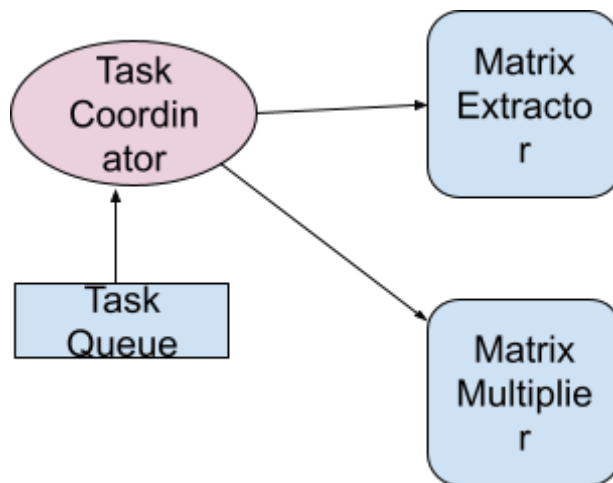
**Pisanje u Red:** Različite komponente sistema imaju mogućnost da dodaju zadatke u Task Queue. Trenutno, to obuhvata Main/CLI komponentu za posao množenja matrica, System Explorer za iniciranje obrade novootkrivenih tekstualnih fajlova, i Matrix Extractor za dodavanje zadatka nakon uspešnog kreiranja matrice.

**Čitanje iz Reda:** Jedina komponenta koja može da čita zadatke iz ovog reda je Task Coordinator komponenta. Ona preuzima i raspoređuje zadatke iz reda na odgovarajuće resurse (druge komponente) za njihovo izvršavanje.



### 2.3 Task Coordinator Thread

Task Coordinator predstavlja nit koja čeka pojavljivanje nekog Task-a, i onda ga delegira odgovarajućem thread pool sistemu. Ova nit treba da se blokira ako nema stavki u redu, a ne da opterećuje procesor konstantnim propitivanjem reda.



## 2.4 Matrix Extractor Thread Pool / Matrix Multiplier Thread Pool

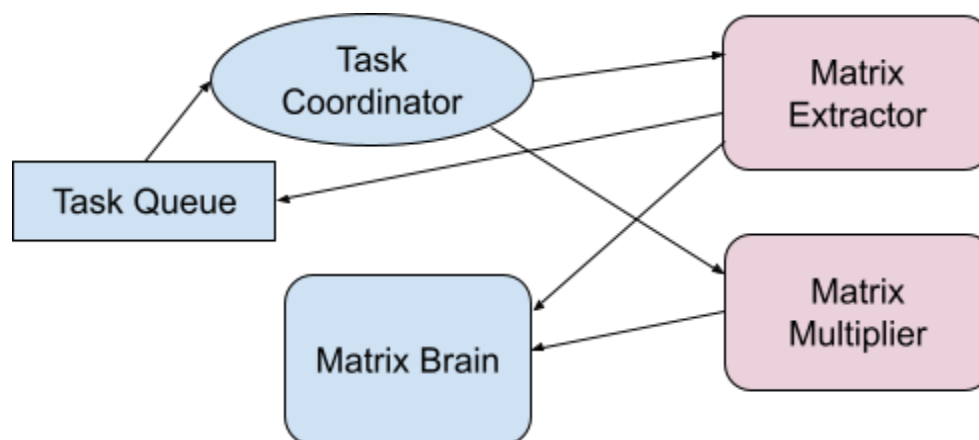
Ovaj deo sistema obuhvata dve ključne funkcionalnosti Matrix Extractor i Matrix Multiplier. Obe funkcionalnosti koriste skupove niti (thread pools) za obradu zadataka.

Kada se pokrene zadatak za množenje matrica, potrebno je navesti dve matrice, A i B, koje će biti pomnožene. Matrix Multiplier koristi thread pool da deli zadatak množenja na manje segmente dok segment obrade ne bude dovoljno mali da ne premašuje definisani limit. U konfiguracionoj datoteci se navodi minimalan (preporučan) broj redova odnosno kolona koji je prihvatljiv za množenje od strane jedne niti.

Svaka nit u pool-u zadužena je za izračunavanje određenog segmenta konačne matrice, bazirano na podelama matrica A i B.

Nakon što su svi segmenti obrađeni, rezultati se kombinuju u novu matricu, koja se potom prosleđuje Matrix Brain komponenti za dalje korišćenje ili skladištenje.

Za Matrix Extractor zadatak, daje mu se datoteka koja sadrži matricu. Zadatak je da se ova datoteka podeli na manje delove, gde svaka nit obrađuje segment datoteke. Segmenti treba da budu dovoljno mali da ne premašuju definisani limit veličine (u bajtovima), koji je zadat u konfiguracionoj datoteci. Deljenje posla treba biti optimizovano tako da se maksimizira broj niti koje istovremeno rade na datoteci, iako nije nužno bitno postići apsolutni maksimum niti. Deljenje se može vršiti na osnovu "pohlepnog" pristupa, uz pretpostavku da datoteke imaju slične veličine. Kao rezultat obrade, vraća se pronađena matrica unutar datoteke. Ova matrica se, takođe, prosleđuje Matrix Brain komponenti. Dodatno, za svaku učitane matricu treba odmah izračunati i njenu kvadratnu matricu, što implicira dodavanje novog zadatka u Task Queue za izračunavanje kvadrata svake pronađene matrice.



Svaka matrica unutar fajla biće predstavljena kroz niz linija koje prate određeni format. Prva linija sadrži osnovne informacije o matrici:

- Naziv Matrice: Jedinstveni identifikator matrice.
- Broj Redova ('rows'): Ukupan broj redova u matrici.
- Broj Kolona ('cols'): Ukupan broj kolona u matrici

Format prve linije izgleda ovako:

`matrix_name, rows=N, cols=M`

Gde je `matrix_name` naziv matrice, `N` je broj redova, a `M` je broj kolona matrice.

Nakon prve linije, svaki element matrice definisan je kroz posebne linije koje prate format:

`i, j = value`

Gde su:

- 'i' indeks reda (počevši od 0),
- 'j' indeks kolone (počevši od 0),
- 'value' numerička vrednost (BigInteger) koja se nalazi na poziciji definisanoj sa `i` i `j`.

`matrix_name=Matrica1, rows=3, cols=3`

`0,0 = 1`

`0,2 = 3`

`1,0 = 4`

`1,1 = 5`

`1,2 = 6`

`2,0 = 7`

`2,1 = 8`

`2,2 = 9`

Data matrica iz fajla ne mora imati popunjene vrednosti za svaku kombinaciju [red,kolona] u tom slucaju je 0.

Na drajvu su okačene matrice za testiranje kao i matrice koje trebati dobiti kao rezultate množenja.

## 2.5 Matrix Brain Thread Pool

Komponenta za dohvaćanje rezultata - Matrix Brain Thread Pool, odgovorna je za upravljanje i pristupanje rezultatima operacija nad matricama, uključujući njihovo čuvanje na disk i izvođenje jednostavnih operacija nad tim rezultatima. Korisnici (kroz Main/CLI komponentu) mogu postavljati upite za dohvaćanje informacija o matricama ili za izračunavanje operacija nad matricama. Postoje tri osnovne vrste upita:

### 2.5.1 Dohvaćanje Osnovnih Informacija o Matrici

Korisnik može zatražiti osnovne informacije o matrici, kao što su njeno ime, dimenzije, i lokacija fajla u kojem se matrica nalazi. Primer upita:

‘info matrix\_name’ koji vraća informacije poput:

```
‘A | rows = 1000, cols = 1000 | matrix_file.rix’
```

**Dodatni Parametri:** Upit može alternativno sadržati dodatne parametre za detaljniji prikaz kada se navede samo komanda `info`, kao što su prikaz svih matrica (‘-all’), sortiranje matrica rastuće (‘-asc’) ili opadajuće (‘-desc’), i ograničavanje broja prikazanih matrica (‘-s N’ za prvih N matrica, ‘-e N’ za poslednjih N matrica).

### 2.5.2 Množenje matrica

Blokirajuće Množenje: Korisnik može zahtevati množenje određenih matrica sa blokirajućim izvršavanjem (blokira se Main/CLI), gde se rezultat množenja ispisuje nakon završetka. Ukoliko posao množenja nije prethodno bio započet potrebno je ispisati poruku i započeti posao množenja. Primer upita:

```
‘multiply mat1,mat2’
```

Asinhrono Množenje: Upit može biti asinhron, što znači da Main/CLI nit neće biti blokirana dok se množenje izvršava. Primer upita:

```
‘multiply A,B -async’
```

Rezultati rada ovih upita će se skladištiti (keširati) radi bržeg ponovnog dohvaćanja.

### 2.5.3 Čuvanje rezultata

Rezultati množenja, kao i sve druge matrice mogu biti sačuvani u fajlu komandom `save` uz navedene parametre ‘save -name mat\_name -file file\_name’. Primer upita:

```
‘save -name mat_name -file output.rix’
```



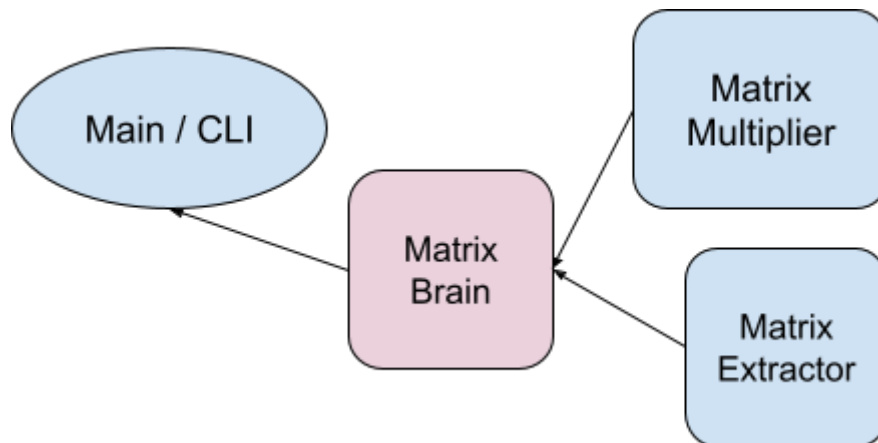
## 2.5.4 Napomene

- Komponenta treba da podržava i blokirajuće (sinhrono) i neblokirajuće (asinhrono) dohvaćanje rezultata modu, prilikom ponavljanja istog zahteva treba pružiti informaciju o statusu obrade (u toku, završeno, nije započeto).

- Radi efikasnosti, rezultati upita se keširaju kako bi se ubrzalo ponovno dohvaćanje istih informacija.

Matrix Brain treba da obezbedi i operacije za brisanje izvesnih skladištenih rezultata, konkretno:

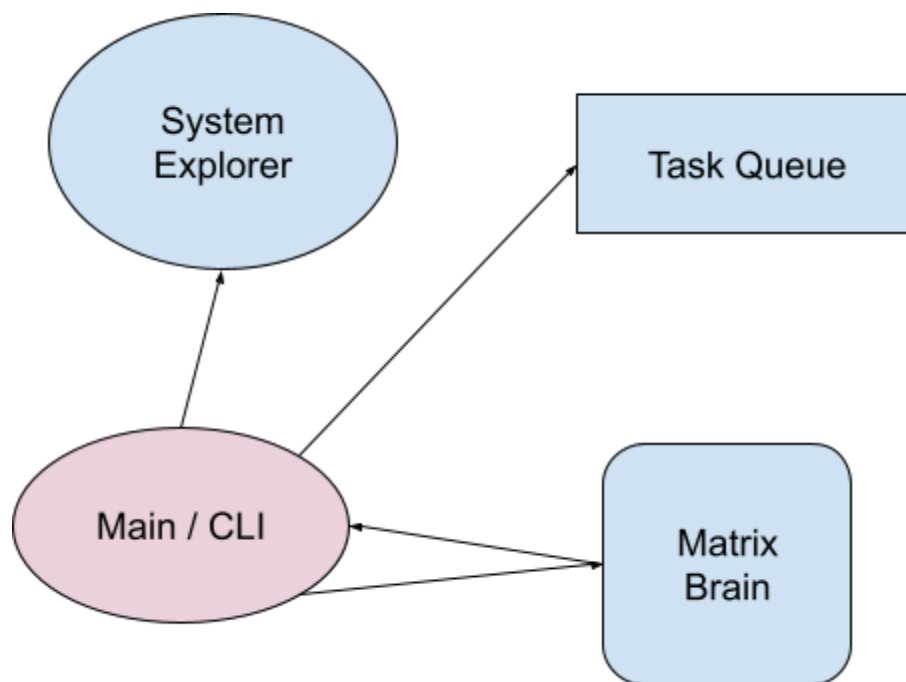
- Brisanje rezultata množenja (biće vršeno preko komande na CLI).
- Prepisivanje postojećih podataka za matrice iz fajla (vrši se automatski kada se neka datoteka na disku izmeni).
- Brisanje podataka o matricama za pojedinačne fajlove. Nakon toga se pri traženju podataka za taj fajl ponovo radi pretraga matrica.



## 2.6 Main / CLI thread

Main komponenta je odgovorna za iniciranje i upravljanje interakcijom korisnika sa sistemom. Na početku, učitavaju se podešavanja iz konfiguracione datoteke, nakon čega se korisniku omogućava unos komandi. Detaljan opis ove datoteke, kao i komandi su dati u **odeljku 4**. Za sada samo taksativno navodimo kako je ova komponenta povezana sa ostatkom sistema:

- System Explorer
  - Komanda 'dir dir\_name' se koristi za dodavanje novih direktorijuma na listu koju System Explorer treba da pretražuje.
- Matrix Brain
  - Postoji nekoliko komandi za dohvaćanje rezultata rada sistema. Sve ove komande će se zapravo obraćati Matrix Brain komponenti i sinhrono ili asinhrono dohvatati rezultate.



### 3 Razrešavanje grešaka

Sistem treba da bude otporan na greške, omogućavajući nastavak rada čak i kada se susretne sa problemima. Probleme koji nisu navedeni u tabeli nema potrebe rešavati.

Svi navedeni problemi treba da se reše graciozno, tj. da sistem nastavi normalno da funkcioniše. Main nit, Task Coordinator i System Explorer, kao i svi pool-ovi treba da nastave da funkcionišu u slučaju da se javi neki od navedenih problema.

- **Nepostojeća ili Nevalidna Komanda:** Ako korisnik unese komandu koja nije prepoznata ili unese pogrešan broj argumenata, sistem treba jasno da obavesti korisnika o grešci i da omogućiti nastavak normalnog rada.
- **Nepostojeći Direktorijum:** Ukoliko je direktorijum koji korisnik želi da doda nevalidan ili ne postoji, sistem treba da informiše korisnika o problemu, bez prekidanja rada.
- **Nedostupna Datoteka:** Ako Matrix Extractor naiđe na datoteku koju ne može da pročita (npr. zbog dozvola ili korupcije), trebalo bi ispisati upozorenje i preskočiti tu datoteku, omogućavajući obradu ostalih datoteka.
- **Nepostojeća Matrica:** Traženje informacija ili bilo kakav pristup matrici koja ne postoji u sistemu treba ispisati informativnu poruku koja ukazuje na to da matrica nije pronađena.
- **Nedovršen Zahtev za Množenje:** Ako korisnik traži rezultat množenja koji još nije završen, sistem treba obavestiti korisnika da rezultat još nije spreman.
- **Nekompatibilne matrice za množenje:** Ukoliko korisnik zatraži množenje matrice koje nije moguće pomnožiti potrebno je ispisati odgovarajuću poruku i preskočiti to množenje.

## 4 Podešavanje sistema i komande

Korisnik interaguje sa sistemom tako što unosi komande na komandnoj liniji. Pored toga, rad sistema se dodatno podešava pomoću konfiguracione datoteke.

### 4.1 Konfiguraciona datoteka

Sistem se podešava pomoću konfiguracione datoteke `app.properties`, koja ima sledeće parametre:

`#period pauziranja za system explorer, dat u ms`

`sys_explorer_sleep_time=1000`

`#ograničenje za matrix extractor komponentu, dato u bajtovima`

`maximum_file_chunk_size=1024`

`#ograničenje za matrix multiplier komponentu`

`maximum_rows_size=3`

`// pocetni direktorijum za pretragu`

`start_dir=/`

Svi ovi parametri se čitaju jednom pri startovanju aplikacije, i neće se menjati tokom rada. Jedini način da se vrednosti ovih parametara izmene je da se aplikacija potpuno zaustavi i ponovo pokrene.

## 4.2 Komande

Sistem podržava sledeće komande:

---

Naziv komande: `dir dir_name`

Dodatni parametri: /

Opis: Add directory. Dodaje novi direktorijum za skeniranje, koji se prosleđuje System Explorer komponenti. Direktorijum se nalazi unutar projekta i zadaje se kao relativna putanja. Unutar ovog direktorijuma može da bude proizvoljno mnogo poddirektorijuma.

---

Naziv komande: `info matrix_name`

Alternativna verzija, parametri koji se koriste kada se ne da naziv matrice:

- all: Prikazuje sve matrice (koristi se bez arguments za naziv matrice).
- asc: Sortira matrice rastuće po broju redova, a zatim po broju kolona.
- desc: Sortira matrice opadajuće po broju redova, a zatim po broju kolona.
- s N: Prikazuje prvih N matrica. Na primer, -s 10 prikazuje prvih 10 matrica.
- e N: Prikazuje poslednjih N matrica. Na primer, -e 5 prikazuje poslednjih 5 matrica.

Opis: Dohvata osnovne informacije o specifičnoj matrici ili skupu matrica. Ako se koristi bez dodatnih parametara, uz naziv matrice `matrix_name`, vraća osnovne informacije kao što su ime matrice, broj redova i kolona, lokacija fajla u kojem se matrica nalazi. Primena dodatnih parametara omogućava korisniku da prilagodi prikaz matrica prema svojim potrebama, uključujući sortiranje i ograničavanje broja prikazanih matrica.

---

Naziv komande: `multiply mat1,mat2`

Dodatni parametri:

- async: Omogućava asinhrono izvršavanje množenja, bez blokiranja Matrix Brain niti.
- name `matrix_name`: Omogućava imenovanje matrice, ukoliko se ne navede kao parametar ime koje matrica koristi je konkatencija naziva prve i druge matrice, primer: `mat1mat2`

Opis: Komanda `multiply` omogućava korisniku da zatraži množenje dve matrice. Bez dodatnih parametara, množenje je blokirajuće, što znači da će se Main/CLI niti pauzirati dok se množenje ne završi, a rezultat množenja će ispisati poruku o uspehu.

---

Naziv komande: `save -name mat_name -file file_name`

Dodatni parametri: /

Opis: Komanda `save` omogućava korisniku da sačuva matricu na disk. Ova operacija nije blokirajuća i za nju je zadužen Matrix Brain.

---

Naziv komande: `clear mat_name / clear file_name`

Dodatni parametri: /

Opis: Ove dve komande se navode sa argumentom koji može biti naziv matrice ili naziv fajla. Naziv fajla se od naziva matrice razlikuje po ekstenziji. Treba javiti Matrix Brain komponenti da obriše odgovarajuće rezultate vezane za navedenu matricu ili sve matrice koje su pročitane iz navedenog fajla i da se njihove vrednosti ponovo prikupe uz pomoć System Explorer-a.

---

Naziv komande: `stop`

Parametar: -

Opis: Gasi aplikaciju. Zaustavlja sve thread pool-ove i javlja svim nitima da uredno završe sa radom. Ne koristiti nasilna prekidanja / `interrupt` / `daemon` za potrebe ove komande. Task Coordinator, Matrix Extractor i Matrix Multiplier se gase korišćenjem `poison pill` metode.

---

### 4.3 Primeri korišćenja sistema

Primeri pretpostavljaju da se koristi konfiguraciona datoteka navedena u odeljku 4.1. Boldovan tekst predstavlja upisane komande, dok je običan tekst ispis programa.

---

Primer 1: dodavanje direktorijuma za skeniranje i dohvaćanje rezultata. Direktorijum **data** unutar projekta sadrži poddirektorijume **folder\_a** i **folder\_a2** negde unutar svoje strukture ti direktorijumi sadrže tekstualne datoteke matrica mat1.rix i mat2.rix.

**dir data**

Adding dir /home/mladen/Apps/intellij/workspace/KiDSDomaci1/data

Exploring file|folder\_a2

Exploring file|folder\_a

Found matrix file|mat1.rix

Found matrix file|mat2.rix

Matrix mat1 created

Matrix mat2 created

Calculating mat2 x mat2

Calculating mat1 x mat1

**multiply mat1,mat2 -name mat12**

[mat12] mat1 x mat2 calculation completed

**stop**

Stopping...

---

---

Primer 2: dodavanje direktorijuma za skeniranje i dohvaćanje rezultata. Direktorijum **data** unutar projekta sadrži poddirektorijume **folder\_a** i **folder\_a2** negde unutar svoje strukture ti direktorijumi sadrže tekstualne datoteke matrica mat1.rix i mat2.rix.

**dir data**

Adding dir /home/mladen/Apps/intellij/workspace/KiDSDomaci1/data

Exploring file|folder\_a2

Exploring file|folder\_a

Found matrix file|mat1.txt

Found matrix file|mat2.txt

Matrix mat1 created

Matrix mat2 created

Calculating mat2 x mat2

Calculating mat1 x mat1

**multiply mat1,mat2 -async**

Calculating mat1 x mat2

**multiply mat1,mat2 -async**

Task not finished yet.

**multiply mat1,mat2 -async**

[mat1mat2] mat1 x mat2 calculation completed

**save -name mat1mat2 -file result.rix**

[mat1mat2]: mat1 x mat2 successfully saved in file

**stop**

Stopping...

---



## 5 Predaja zadatka

### 5.1 Način predaje zadatka

#### Github Classroom

Zadatak se predaje slanjem putem Git Classroom-a.

Prijavite se na Github classroom u zavisnosti od dana kada po rasporedu imate vežbe iz KiDS-a.

**Radovi studenata koji ne udju na Github classroom u terminu u kom slušaju predmet boduju se sa 0 poena.**

Rok za predaju i Github Classroom link je:

- Ponedjeljak, 1. april 23:59:59 za grupu koja sluša KiDS ponedjeljkom.  
Github classroom link: <https://classroom.github.com/a/fEsCpfvY>
- Utorak, 2. april 23:59:59 za grupu koja sluša KiDS utorkom.  
Github classroom link: <https://classroom.github.com/a/wQzQvb8v>
- Četvrtak, 4. april 23:59:59 za grupe koje slušaju KiDS četvrtkom.  
Github classroom link: <https://classroom.github.com/a/Su2a0LCz>
- Petak, 5. april 23:59:59 za grupu koja sluša KiDS petkom.  
Github classroom link: <https://classroom.github.com/a/nl-jDHta>

Ako nemate nalog na GitHub-u, morate ga napraviti. Ako već imate nalog, prijavite se na GitHub sa Vašim postojećim nalogom.

Nakon što prihvatite zadatak, GitHub će automatski kreirati repozitorijum za Vas sa odgovarajućim imenom i postavkama za zadatak.

Da biste radili na zadatku, morate klonirati repozitorijum na svoj lokalni računar. To možete uraditi koristeći komandu `git clone` u terminalu sa linkom do Vašeg repozitorijuma.

Nakon što klonirate repozitorijum, možete početi da radite na zadatku lokalno na svom računaru.

Kada završite sa izmenama, odradite komandu `git add .` da dodate sve izmene, a potom `git commit -m "Opis izmena"` da ih komitujete na lokalni repozitorijum.

Nakon dodavanja izmena komandom `git add .`, korisno je proveriti status repozitorijuma komandom `git status` kako biste bili sigurni da su sve izmene uspešno dodate.

Kada ste spremni da otpremite izmene na GitHub, koristite komandu `git push` da ih otpremite na GitHub.

Nakon što otpremite izmene, možete ih pregledati na GitHub-u da biste se uverili da su ispravne.

Važno je da pratite uputstva i da se javite mejlom ako imate pitanja ili probleme, kako bismo mogli na vreme da ih uklonimo.

## Ponovci

Za studente koji predmet ne slušaju prvi put, domaći se predaje na mail: [ml\\_jovanovic@raf.rs](mailto:ml_jovanovic@raf.rs).

Projekat imenovati na sledeći način: "kids\_d1\_ime\_prezime\_ind".

Npr. "kids\_d1\_student\_studentic\_rn0101".

Uraditi clean projekta. Potom arhivirati ovaj direktorijum (.zip) i arhivu poslati kao attachment uz mail ili kao link ka drive na kojem se nalazi arhiva.

U tekstu mail-a obavezno navesti:

- Ime i prezime
- Broj indeksa
- Grupa, po zvaničnom spisku (Ponovac)
- Razvojno okruženje (Eclipse/Intelij)

Subject mail-a mora da bude u obliku: "[KiDS 2024] D1 ime\_prezime\_ind".

Npr. "[KiDS 2024] D1 student\_studentic\_rn0101"

Naziv arhive mora da bude u obliku: "kids\_d1\_ime\_prezime\_ind.zip"

Npr. "kids\_d1\_student\_studentic\_rn0101.zip"

Rok za studente koji su preneli predmet je poslednji rok projekta.

Rok je definisan po grupi kojoj student zvanično pripada, tj. na čijem spisku se nalazi.

Neće se pregledati zadaci (tj. biće dodeljeno 0 poena) koji nisu predati u odgovarajućem formatu.

## 5.2 Odbrana i bodovanje

Odbrana domaćih zadataka je obavezna. Termin za odbranu prvog domaćeg zadatka će biti u terminu rezervisanom za KiDS u kolokvijumskoj nedelji. Odbrane će se vršiti individualno na fakultetu. Raspored odbrana će biti formiran i objavljen ubrzo nakon isteka roka za predaju domaćeg zadatka. Ako ste iz bilo kog razloga sprečeni da prisustvujete odbrani, obavezno to najavite što pre, kako bismo mogli da zakažemo vanredni termin za odbranu.

Svrha odbrane je da se pokaže autentičnost zadatka. Ovo podrazumeva odgovaranje na pitanja u vezi načina izrade zadatka, ili izvršavanje neke izmene nad zadatkom na licu mesta. U slučaju da odbrana nije uspešna, dodeljuje se 0 poena.

Zadatak se boduje na sledeći način:

- |                                      |   |          |
|--------------------------------------|---|----------|
| ● Matrix Multiplier komponenta       | - | 4 poena  |
| ● Matrix Extractor komponenta        | - | 3 poena  |
| ● Matrix Brain komponenta            | - | 4 poena  |
| ● System Explorer i Task Coordinator | - | 2 poena  |
| ● Main / CLI i konfiguracija         | - | 2 poena  |
| ● Greške nisu pravilno razrešene     | - | -2 poena |
| ● Sistem nije lepo ugašen            | - | -2 poena |

Zadatak je moguće raditi parcijalno, ali obavezno je da se kompajluje i da može da se pokrene, kao i da je moguće pokazati da implementirana komponenta radi po zahtevima iz odeljka 2 ovog dokumenta. Stavka “Main / CLI i konfiguracija” se prihvata samo uz implementiranu još makar jednu stavku.