

Универзитет у Београду
Електротехнички факултет

Катедра за рачунарску технику и информатику

Списак могућих тема за завршне и мастер радове на предмету за студенте
на Одсеку за РТИ и Одсеку за СИ

Интердисциплинарне, мултидисциплинарне, и трансдисциплинарне теме

(ажурирано 23.03.2026.)

Контакт:

Проф. др Марко Мишић (marko.misic@etf.bg.ac.rs)

Асистент Матија Додовић (mdodovic@etf.bg.ac.rs)

Финална тема се дефинише у разговору са предметним наставником на основу оквирних тема које су наведене у прилогу.

Студенти могу сами да предложе тему из сродних области.

Велики језички модели (Large Language Models - LLM)

- RAG sistem za ETF dokumenta sa citatima
- Fine-tuning LLM-a za pomoć pri učenju programiranja
- LLM kao *code reviewer* za Python/C/C++ zadatke i programski kod
- LLM kao *reviewer* i autocomment generator za Python/C/C++
- LLM kao *rewriter* za pseudokod
- LLM evaluacija kvaliteta studentskih odgovora (rubrike + feedback)
- Web aplikacija za “human-in-the-loop” korekciju LLM izlaza (*dataset builder*) – na bilo koju željenu temu
- Detekcija plagijata i “code similarity” u sekvencijalno i paralelnom programskom kodu
- LLM asistirana optimizacija koda: predlog paralelizacije + validacija performansi

Мултипроцесорски системи и паралелно програмирање (MUPS)

- Simulacija održavanja koherencije keš memorije (svi protokoli) i analiza performansi
- CUDA *kernel fusion* na standardnim *banchmark* skupovima (*Rodinia*, *Parboil*, *NASA Parallel Benchmark*, *PolyBench*, *Stream* i sl.)

- CUDA 2D *kernel fusion*
- OpenMP/Pthreads za grafovske algoritme
- GPU ubrzanje pretrage sličnosti (kNN / cosine) za RAG sisteme
- Paralelni *map-reduce* na GPU - atomics vs. block-level redukcija (performanse)
- *Tasking vs parallel-for* na rekurzivnim problemima
- Određivanje optimalnog broja niti i veličine bloka u paralelnim problemima – *grid search* pristup
- *Adaptive parallelism*: dinamičko menjanje broja niti tokom rada
- Analiza uticaja kompajlerskih direktiva (npr. *-O2/-O3/-march=native*)
- *Atomic vs reduction vs critical* – analiza uticaja, prednosti i nedostataka
- Analiza uticaja sinhronizacije na barijerei - meriti cenu *#pragma omp barrier* u različitim tačkama; uporediti sa pristupom bez barijera
- Analiza *thread oversubscription* efekata za više niti nego jezgara; meriti degradaciju i preporučiti pravilo izbora OMP_NUM_THREADS
- Analiza efekata *spinlock vs mutex vs atomic-fence* u kratkim kritičnim sekcijama
- Poređenje *#pragma omp simd, auto-vectorization* i čistog OpenMP koda

LLM + MUPS (agentski zasnovano ili na bazi *prompt engineering-a*)

- Predlaganje optimalne politike raspoređivanja petlji kod paralelizacije direktivama korišćenjem LLM
 - LLM predlaže *schedule* politiku za petlje (*static/dynamic/guided*)
- Predlaganje optimalne politike sjedinjavanja petlji kod paralelizacije direktivama korišćenjem LLM
 - LLM predlaže *collapse(n)* i preuređenje ugnježenih petlji
- Korišćenje LLM za automatizovano sprovođenje paralelne redukcije kod paralelizacije direktivama
 - LLM detektuje redukcije i automatski dodaje *reduction(...)*
- Korišćenje LLM za automatizovano dodeljivanje opsega važenja promenljivih u paralelnom programiranju
 - LLM prepoznaje deljene i privatne promenljive (*shared/private/firstprivate*)

- LLM generiše micro-benchmark za datu petlju i bira optimalnu paralelizaciju
- Optimizacija paralelnog programskog koda za iskorišćavanje organizacije keš memorija korišćenjem LLM
 - LLM detektuje *cache-friendly* paralelizaciju i predlaže adekvatan *tiling*
- Analiza sprovođenja anti-optimizacija korišćenjem LLM
 - LLM predlaže gde NE treba paralelizovati (*anti-optimization*) i poređenje opiranja *syncophancy* efektu (generalno slaganje sa *user* odgovorima)

Mašinsko učenje (Machine Learning - ML)

- Poređenje različitih *tokenizer* rešenja za *pipeline* treniranja
 - BPE tokenizer i sl.
- Analiza treniranja modela na različitim paralelnim platformama
 - CPU (OpenMP) + GPU (CUDA) podela posla po *batch*-evima
 - Mali *batch* na CPU, veliki na GPU, odrediti optimum za raspoređivanje
- Implementacija NER (*Named Entity Recognition*) za srpski jezik

Algoritmi i strukture podataka (ASP)

- Poređenje reprezentacija retko posednutih matrica i implementacija algoritama
- Poređenje implementacija heš tabela
 - Cuckoo hashing vs Robin Hood
 - Python implementacija rečnika
 - Adaptivne heš tabele - *dynamic resizing* politike u *hash map*: kada i kako proširivati (*threshold*, incremental rehashing); meriti *pause time* i *throughput*.
- *HyperLogLog* struktura podataka za procenu broja različitih elemenata
 - Implementacija i poređenje tačnosti i memorije sa tačnim brojanjem.
- Intervalna stabla
 - Preklapanje intervala (kalendar/rezervacije)
- B /B+ stablo kao indeks za mini bazu na disku
 - Implementacija i integracija
- Upiti za intervale – *Mo* algoritam vs segmentno stablo

- *Union-Find* optimizacije
- Implementacija *Dijkstra* algoritma korišćenjem binarnog *heap*-a vs binomnog *heap*-a vs Fibonačijevog *heap*-a
- Oređivanje jako povezanih komponenti
 - *Kosaraju* vs *Tarjan* algoritmi
 - Mogućnost vizuelizacije
- Aho–Corasick algoritam
 - *Multipattern* pretraga
- Algoritmi za pretragu stringova
 - KMP/Z algoritam vs *Boyer–Moore* u praksi
- *MinHash* + LSH za detekciju sličnih dokumenata
 - *Near-duplicate* identifikacija
- Analiza benefita dinamičkog programiranja
 - Na nekoliko primera

Softverski alati i aplikacije (SAA)

- Genetski algoritmi za rasporede
 - Ispita, časova, dežurstava, itd.
- Implementacija IDE dodataka
 - VS Code i sl.
- Implementacija *Chrome browser* ekstenzija