

# Bitne i korisne informacije prilikom izrade projekta

## Formalne informacije:

- Poenta projekta jeste da sumirate znanje stečeno tokom celog semestra iz predmeta Paralelno Programiranje, te samoj izradi treba pristupiti odgovorno i na vreme (koga ima sasvim dovoljno).
- Odbrana projekta biće organizovana uživo u unapred dogovorenim terminima u prostorijama naučno-tehnološkog parka (učionice NTP 504-508). Za oni koji to žele, biće omogućeno da se projekat brani ranije u takođe dogovorenom terminu (takozvani predrok).
- Prezentacija se sastoji iz pokretanja rešenja (nekog reprezentativnog testnog slučaja), kratke diskusije i analize, davanje odgovora na par pitanja i ocenjivanjem dokumentacije. Celokupna odbrana tipično traje između 15 i 20 minuta.
- Na samu odbranu je potrebno poneti jedan primerak odštampane dokumentacije, poželjno ukoričen ili zaheftan. Nije potrebno štampanje u boji, crno-belo je sasvim dovoljno – samo u tom slučaju obratite pažnju da boje na slikama budu i dalje dovoljno jasne.
- Takođe bi najbolje bilo da ponesete sopstveni ili pozajmljen laptop računar na kome će sve biti podešeno za potrebe projekta. Ukoliko to nije moguće, biće obezbeđen adekvatan računar u samoj laboratoriji samo je potrebno javiti unapred.

## Tehničke informacije:

- Prvi korak u izradi projekta jeste napraviti neki koncept. Taj koncept treba da prati serijska implementacija algoritma. Rešenje dobijeno serijskom implementacijom treba proveriti sa nekim drugim izvorom (npr. MATLAB). Tek kada je ustanovljeno da serijski algoritam radi ispravno treba preći na sledeći korak. **Prvi korak treba da bude gotov zaključno sa prvim terminom nakon zadatka iz OpenCL-a.**
- Drugi korak u izradi projekta jeste paralelno rešenje upotrebom TBB biblioteke. Osim delova koji se tiču TBB-a, ne treba menjati sam algoritam na drugačije načine kako bismo bili što objektivniji prilikom merenja performansi odnosno vremena izvršavanja. Na primer, ako smo koristili metodu pushBack za dodavanje elementa u vektor u serijskom rešenju, istu metodu treba koristiti i u paralelnom. Ako pushBack metodu zamenimo sa direktnim pristupom u paralelnom rešenju, nećemo dobiti realnu sliku i samim tim i u serijskoj implemntaciji treba koristiti onda direktni pristup. **Drugi korak treba da bude gotov nedelju dana nakon prvog koraka.**
- Kada ustanovite da paralelno rešenje radi, prelazi se na 3. korak a to je ispitivanje. Za potrebe ispitivanja možete koristiti unit test-ove ili možete napraviti neke svoje. Testiranje treba da bude detaljno i iscrpno, da pokrije različite veličine ulaznih parametara, kako validnih, tako i nevalidnih. Ukoliko imamo kao zadatak sortiranje niza, treba kao ulazne podatke imati nizove različite dužine. **Treći korak treba da bude gotov nedelju dana nakon drugog koraka.**

- Poslednji korak jeste pisanje dokumentacije. Dokumentacija treba da liči po sadržaju na onu koja je priložena na Canvas-u. Nemojte se opterećivati dužinom dokumentacije, bitno je šta se unutra suštinski nalazi. Nemojte praviti nepotrebno dugačke uvode koji su doslovno kopirani sa Wikipedije ali nemojte biti ni šturi na rečima – neko ko pročita dokumentaciju bi trebalo da zna o čemu se radi u vašem projektu bez uvida u kod. Ne morate za svaku funkciju praviti podnaslov, ali bi bilo lepo da dokumentacija ima određena poglavlja i izdvojene najvažnije celine. **Dokumentacija treba da bude spremna za odbranu.**
- Vodite računa o formatiranju dokumentacije i koristite justifed poravnanje kad god je moguće. Slike treba da se nalaze u sredini, kao i tabele (centrirano poravnanje).
- Kod treba da bude modularan. Sve što ima smisla da bude zasebna funkcija neka bude izdvojeno kao zasebna funkcija. Izbegavajte ponavljanje koda kad god je moguće. Najbolje bi bilo da u okviru jednog solution-a imate odvojene projekte za serijsku i paralelnu implementaciju.
- U kodu treba da postoje komentari koji treba da budu kratki i jasni i da opisuju stvari koje nisu na prvi pogled očigledne. Npr. nije neophodno da komentarišete funkciju za ispis matrice na konzolu, ali jeste dobro komentarisati klasu koja predstavlja srž vašeg rešenja.
- Podaci treba da budu pripremljeni unapred. Drugim rečima, matrice, nizovi i ostale strukture podataka treba da se nalaze u nekim datotekama koje će biti na početku pročitane. Takođe, treba da postoje i izlazne datoteke kako bi rezultat mogao da se proverí. Nije dobro svaki put na početku izvršavanja generisati nasumične strukture.
- Treba voditi računa o rukovanju greškama tamo gde do njih može doći – ako se pojavi nevalidni ulazi na početku, ako dimenzije matrice nisu odgovarajuće i slično.
- U slučaju dinamičkog zauzimanja memorije, potrebno je tu istu memoriju na kraju osloboditi.
- Bilo da dokument pišete na ćirilici ili latinici, potrudite se da pišete gramatički ispravno. Nemojte pisati ošišanom latinicom u kojoj neće biti š, č, đ, ć, ž. Cilj je između ostalog da kad završite budete i pismeni inženjeri.