### **Napomena: U nastavku je preliminarna postavka zadataka. Moguće su manje izmene u narednim nedeljama.**

### **Logistika**

* Predaja se vrši preko github classroom sistema: <https://classroom.github.com/a/WPRRD-_H>
* Sav kod treba da bude u jednom .py fajlu.
* Potrebno je uključiti najmanje jedan primer json fajla i jednu sliku u repozitorijum.
* Rok za predaju je 14.11.2024 do kraja dana.

### **Paralelni Sistem za Obradu Slika**

**Opis Projekta:** Razviti sistem koji omogućava paralelnu obradu velikog broja slika. Sistem bi trebalo da primenjuje različite filtere i transformacije na slike, čuva obrađene verzije i održava evidenciju o statusu obrade.

### **Komponente Sistema**

#### **1. Registar slika [3 boda]**

* **Opis:** Sadrži spisak originalnih i obrađenih slika, njihove jedinstvene identifikatore, id zadatka koji je kreirao sliku (None ako je slika uploadovana u sistem), listu zadataka u kojima se slika koristi, i flag koji se postavlja ako je slika predviđena za brisanje.
* **Dodatni podaci:** Vrste primenjenih filtera, vreme obrade, veličina slike pre i posle obrade.

#### **2. Registar zadataka [5 bodva]**

* **Opis:** Čuva listu zadataka za obradu slika, uključujući identifikatore slika, vrste transformacija koje treba primeniti, status zadatka (npr. "čekanje", "u obradi", "završeno"). Sadrži uslov (Condition) koji se aktivira (Notify All) kada je zadatak završen.

#### **3. Glavna nit za prijem komandi**

* **Opis: Glavna nit koja pokreće sve niti, prima komanda, i prosleđuje ih na izvršavanje**
* **Funkcionalnost**
  + Pri pokretanju kreira registar slika, registar zadataka, kao i sve potrebne niti i procese
  + U petlji čeka na komande sa ulaza
  + Periodično proverava da li je postavljen signal za gašenje (postavlja ga nit koja zadaje exit komandu)
  + Periodično proverava da li postoji tekst koji je potrebno ispisati na komandnoj liniji, i po potrebi da ispisuje (stiže od komandi list i describe kroz poseban red za poruke (Queue)).
  + Po prijemu komande kreira i pokreće nit za njenu obradu

#### **4. Niti za obradu komandi [5 bodova]**

* **Opis:** Niti koje izvršavaju komande korisnika kao što su add, process, delete, list,describe, exit.
* **Funkcionalnost:**
  + Svaka primljena komanda se obrađuje u posebnoj niti kako bi se omogućila paralelna obrada više komandi istovremeno.
  + Komanda add kopira sliku u direktorijum *./slike*.
  + Komanda exit zaustavlja sve aktivne niti i procese na siguran način.
  + Komanda koja kreira sliku (add, process) joj dodeljuje jedinstveni identifikator (na osnovu brojača koji se inkrementira).
  + Komanda koja vrši obradu slike (process) prima svoje parametre kao putanju ka JSON fajlu, koji sadrži parametre komande.
  + Obrada slike se izvršava kao zadatak u posebnom procesu.
  + Komanda koje kreira zadatak mu dodeljuje jedinstevni identifikator (na osnovu brojača koji se inkrementira).
  + Komanda koja zavisi od rezultata nekog drugog zadatka, pre nego što pošalje svoj zadatak na obradu u registru zadataka pronalazi zadatak od kojeg zavisii čeka na njegovom uslovu.
  + Komanda *delete* briše sliku. Prilikom brisanja komanda prvo postavlja flag koji ozanačva da je slika ozančena za brisanje, pa proverava sve zadatke koji koriste tu sliku i čeka da se završe pre brisanja.
  + Komanda koji želi da koristi sliku koja je označena za brisanje se odbija (dovoljno je da je uradi ništa)
  + Komanda list ispisuje sve slike koje se nalaze u registru slika (kao par id slike, putanja do slike)
  + Komanda describe <id slike> ispisuje sve slike i zadatke od kojih je slika nastala

#### **5. Procesi za Obradu Slika [5 bodova]**

* **Opis:** Procesi koje primaju zadatke iz registara i izvršavaju potrebne transformacije na slikama.
* **Implementacija:**
  + Korišćenje **multiprocessing.Pool** za efikasnu paralelnu obradu.
  + Parametri funkcije koja izvršava obradu su najmanje *id zadatka,* *id ulazne slike, putanja do izlazne slike.* Funkcija može primati i dodatne parametre koji uvek imaju podrazumevane vrednosti.
  + Svaki proces dobija zadatak iz registra zadataka, izvršava transformaciju i vraća id zadatka kao rezultat.
  + Koristimo callback funkciju koja id zadatka stavlja u red za završetak obrade

#### **6. Red i nit za završetak obrade [2 boda]**

* **Opis:** Mehanizam kojim se zadaci označavaju kao gotovi i kojim se obaveštavaju komande koje čekaju da na završetak nekog zadatka
* **Implementacija:**
  + Red (Queue) u koje se smaštaju identifikatori zadataka koji su završeni
  + Nit koja prazni red, označava da su zadaci završeni i obaveštava njihove uslove.

### **Transformacije**

1. **Sivljenje (Grayscale)**:  
   Pretvaranje kolor slike u sivu zahteva izračunavanje luminantnosti svake tačke slike (piksela) na osnovu vrednosti crvene (R), zelene (G), i plave (B) komponente. Svaka boja različito doprinosi percepciji svetlosti, pa se koristi ponderisana suma RGB kanala (tipično 0.299 \* R + 0.587 \* G + 0.114 \* B). Ova kombinacija se računa za svaki piksel u slici i rezultat je nova slika u sivim tonovima.
2. **Gaussian Blur (Gaussovo Zamagljivanje)**:  
   Gaussian Blur je tehnika koja se koristi za zamagljivanje slike, a postiže se primenom Gaussian filtera na piksele. Gaussian filter je matrica (kernel) koja sadrži težine raspoređene prema Gaussovoj distribuciji. Za svaku tačku na slici, novi vrednost piksela dobija se množenjem susednih piksela vrednostima u filteru, a zatim sumiranjem rezultata. Ovo omekšava oštre prelaze u slici i stvara efekat zamagljenja.
3. **Podešavanje Svetline (Brightness Adjustment)**:  
   Svetlina slike može se menjati skaliranjem vrednosti piksela za konstantu ili funkcijom. Ako želite da povećate svetlinu, možete dodati ili pomnožiti sve vrednosti piksela određenim faktorom, ali treba paziti da se ne pređe maksimalna vrednost piksela (npr. 255 za slike u formatu 8-bit). Smanjenje svetline se postiže oduzimanjem ili smanjivanjem vrednosti piksela. Takođe, kontrast se može podešavati tako što se prvo normalizuju vrednosti piksela prema njihovom srednjem intenzitetu, a zatim skaliraju.