Računarske mreže 2019, Kolokvijum, 4RB

08.04.2019.

1. Selektivno kopiranje fajla **(10p)**

   - Napraviti Java aplikaciju koja koristeći odgovarajuće ulazne i izlazne tokove kopira sadržaj tekstualnog fajla sa imenom koje se unosi preko standardnog ulaza u fajl `hex.txt`. Postarati se da se u slučaju izuzetka prikaže odgovarajuća poruka (različita za različite tipove izuzetaka). (2p)
   - Prekopirati samo one niske koje predstavljaju validne heksadecimalne brojeve (počinju sa `0x` i sastoje se od cifara `0-F`, mogu bili mala ili velika slova) (npr. `0x1Abc2D`). (2p)
   - Koristiti baferisanje ulaznog i izlaznog toka zarad smanjenja broja IO operacija. (2p)
   - Niske ispisati u fajl tako da po jedna niska bude u svakoj liniji. (1p)
   - Podesiti kodnu stranu za izlazni fajl na ASCII. (1p)
   - Postarati se da se u slučaju izuzetka garantuje da su zatvoreni svi korišćeni resursi. (1p)

2. Skalarni proizvod vektora **(10p)**

   Napraviti Java aplikaciju koja koristeći niti računa skalarni proizvod dva vektora.

   - Kao ulaz u program se daje putanja do tekstualnih fajlova u kojima se nalaze vektori ́vciji je skalarni proizvod potrebno izračunati - po jedan u svakom fajlu. Učitati i ispisati vektore na standardni izlaz. (1p)
   - Kreirati posebnu klasu `VectorMultiplicationException` i baciti izuzetak ovog tipa ukoliko vektori nemaju istu dimenziju. (1p)
   - Označimo dimenziju vektora sa n. Pokrenuti n niti i postarati se da svaka računa jedan element rezultujućeg vektora. Ispisati rezultujući vektor na standardni izlaz. (4p)
   - Računati L_1 normu rezultujućeg vektora tokom rada svake niti. Kada svaka nit izračuna svoju vrednost, ažurira globalnu vrednost L_1 norme. Postarati se da nema trke za podacima - obezbediti kritičnu sekciju proizvoljnim mehanizmom. L_1 norma vektora se računa po formuli (gde je x vektor dimenzije n): (3p)

   $$L_1(x) = \sum_{i=1}^{n} x_i$$

   - Voditi računa o obradi izuzetaka - program ili nit ne sme da se zaustavi u slučaju izuzetka (npr. ukoliko se desi izuzetak prilikom računanja elemenata rezultujućeg vektora, smatrati da je rezultat 0 i nastaviti sa radom). (1p)

3. URL Scanner **(10p)**

   - Napraviti Java aplikaciju koja sa standarnog ulaza učitava URL-ove jedan po jedan u svakoj liniji. Formirati URL objekat za svaki učitani URL i ispisati na standardni izlaz poruku ukoliko URL nije validan. (2p)
   - Ispisati protokol, `authority` i putanju iz URL-a koristeći URL klasu. Izlaz formatirati na sledeći način: `<KORIŠĆENI_PROTOKOL> <PODRAZUMEVANI_PORT> <HOSTNAME> <PUTANJA_DO_RESURSA>` npr.
     ulaz: `http://www.matf.bg.ac.rs:3030/dir1/dir2/test.txt`
     izlaz: `http www.matf.bg.ac.rs 80 /dir1/dir2/test.txt` (3p)
   - Ukoliko se unese IP adresa unutar URL-a umesto informacija iznad ispisati kao u primeru ispod (ako port nije unet preskočiti ga), a ako je to IPv4 adresa ispisati i njene bajtove:
     `(v<VERZIJA_IP_ADRESE>) <KORIŠĆENI_PROTOKOL> <PORT> <PUTANJA_DO_RESURSA> [<BAJTOVI_ADRESE>]` npr.
     ulaz: `http:///123.123.123.123:80/dir1/dir2/test.txt`
     izlaz: `(v4) http 80 /dir1/dir2/test.txt [123 123 123 123]`
     ulaz: `sftp://2001:0db8:85a3:::8a2e:0370:7334/dir1/dir2/test.txt`
     izlaz: `(v6) sftp /dir1/dir2/test.txt` (4p)
   - Postarati se da u slučaju izuzetka aplikacija ispravno zatvori korišćene resurse. (1p)