

Računarske mreže, Vanredni kolokvijum

Pročitati sve zadatke **pazljivo** pre rada - sve što nije navedeno ne mora da se implementira!

Vreme za rad: **2.5h**. Srećno!

1. Selektivno kopiranje fajla (**9p**)

- Napraviti Java aplikaciju koja koristeći odgovarajuće ulazne i izlazne tokove kopira sadržaj tekstualnog fajla sa imenom koje se unosi preko standardnog ulaza u fajl `phonenumbers.txt`. Postarati se da se u slučaju izuzetka prikaže odgovarajuća poruka (različita za različite tipove izuzetaka). (2p)
- Prekopirati samo one niske koje predstavljaju validne telefonske brojeve u formatu `+CCC/CCCC-CCC` (gde je C proizvoljna cifra dekadnog zapisa). (2p)
- Koristiti baferisanje ulaznog i izlaznog toka zarad smanjenja broja IO operacija. (2p)
- Niske ispisati u fajl tako da po jedan broj bude u svakoj liniji. (1p)
- Podesiti kodne strane za oba fajla na UTF-8. (1p)
- Postarati se da se u slučaju izuzetka garantuje da su zatvoreni svi korišćeni resursi. (1p)

2. Višenitna pretraga (**12p**)

Napraviti Java aplikaciju koja koristeći niti pretražuje listu fajlova i ispisuje sva pojavljivanja zadate ključne reči.

- Kao ulaz u program se daje putanja do tekstualnog fajla u kom se nalaze putanje do svih fajlova koje je potrebno pretražiti - po jedna u svakoj liniji. Učitati putanje i ispisati ih na standardni izlaz. (1p)
- Učitane putanje staviti u proizvoljnu kolekciju. Zatim učitati od korisnika ključnu reč za pretragu i broj n . Pokrenuti n niti i omogućiti da svaka nit uzima putanju iz kolekcije i pretražuje fajl na toj putanji. Kada završi sa radom, nit uzima novu putanju iz kolekcije. Obezbediti da više niti ne obradjuje isti fajl kao i da se svi fajlovi eventualno obrade. Svaka nit bi trebalo da ima jednaku verovatnoću pristupa elementima kolekcije (drugim rečima, nije u redu da se pokrene 10 niti a samo jedna da uzima putanje i obradjuje ih). Ukoliko se pronadje tražena ključna reč, ispisati tekst u sledećem formatu:
`nit.<ID_NITI>:<APSOLUTNA_PUTANJA_DO_FAJLA>:<BROJ_LINIJE>:<POZICIJA_U_LINIJI>` npr.
`nit.1:/home/user/Desktop/temp/2.txt:33:5` (10p)
- Voditi računa o obradi izuzetaka - program ili nit ne sme da se zaustavi u slučaju izuzetka (npr. ukoliko fajl na datoj putanji ne postoji ignorisati grešku i nastaviti sa radom). (1p)

3. Parser log fajla (**9p**)

- Napraviti Java aplikaciju koja prima putanju do regularnog fajla koji predstavlja log fajl i čita ga koristeći URL klasu i FILE protokol. Ispisati sadržaj fajla na standardni izlaz. (3p)
- U fajlu se nalaze linije u sledećem formatu:
`[<DATUM_VREME>] | <IP_ADRESA> | <URL_DO_RESURSA_NA_SERVERU>` npr.
`[12.12.2010] | 123.123.123.123 | http://poincare.matf.bg.ac.rs/~ivan_ristovic/secret.txt`
Filtrirati pročitani sadržaj tako da se na standardni izlaz ispišu samo one linije u kojima je zahtev za resurs stigao preko HTTP ili SFTP protokola. (3p)
- Format ispisa linije na standardni izlaz promeniti na:
`v<VERZIJA_IP_ADRESE> | <KORIŠĆENI_PROTOKOL> | <PUTANJA_DO_RESURSA>` (zameniti tagove odgovarajućim informacijama) npr.
`v4|http|/courses/rm/secret.txt` (2p)
- Postarati se da u slučaju izuzetka aplikacija ispravno zatvori korišćene resurse. (1p)