

# Uslužilac za brzo indeksiranje datoteka

Bogdan Satarić, Fakultet Tehničkih Nauka, Novi Sad

Dejan Stefanović, Dragan Simić, MicronasNIT, Novosadski Institut za Informacione Tehnologije, Novi Sad

**Sadržaj** – U radu je prikazana realizacija aplikacije za indeksiranje i praćenje liste datoteka u nekoj putanji. Cilj realizacije ovakve aplikacije jeste stvaranje nezavisnog programskog rešenja koje će omogućiti pozadinsko indeksiranje liste datoteka. Na ovaj način rasteretiće se i ubrzati izvršavanje glavne aplikacije koja koristi usluge indeksiranja i praćenja liste datoteka. Date su osnovne karakteristike, kao i kratak opis realizacije date aplikacije.

**Ključne reči** – IPR, JNI, DAEMON

## I. UVOD

Realizacija uslužioca za brzo indeksiranje datoteka je deo šireg projekta Micronas IPR (Intellectual Property Rights) Editora [2], koji se koristi za evidenciju patenata firme Micronas GmbH. U okviru Micronas IPR Editora [2], postoji mogućnost povezivanja delova teksta sa datotekama preko hiperveza. Na taj način omogućuje se brz pristup datotekama koje su povezane sa datim tekstom. Funkcionalnost povezivanja datoteka sa tekstom u Micronas IPR Editoru [2], zahteva uslugu koja će omogućiti pristup listi datoteka sadržanih u okviru izabranog direktorijuma, ali i onih datoteka koje se nalaze u pod-direktorijumima, nezavisno od broja i dubine propagacije pod-direktorijuma. Imajući u vidu prethodni zahtev i uvažavajući činjenicu da koreni direktorijum C:\ (koji je najopštiji primer indeksiranog direktorijuma), može da sadrži i po nekoliko stotina hiljada datoteka u svojim pod-direktorijumima, potrebno je realizovati pozadinsku aplikaciju koja bi indeksirala i pratila listu datoteka u željenim direktorijumima, i na taj način skratila vreme pronalaženja željene datoteke.

Pojam indeksiranja i praćenja liste datoteka uveden je kako bi se eliminisala potreba za ponovnim čitanjem kompletne liste datoteka u datom direktorijumu i njegovim pod-direktorijumima - svaki put kada bi se zahtevala data lista. Ponovno učitavanje bi bilo neophodno zbog mogućnosti da je od zadnjeg učitavanja liste došlo do brisanja pojedinih datoteka ili dodavanja novih, te bi se lista morala ažurirati.

Umesto toga, obavlja se početno indeksiranje liste datoteka, dok se sve promene u datoj listi registruju kao promene na pojedinačnim datotekama u listi. Na taj način se omogućuje ubrzano ažuriranje promena na elementima liste, a ne na celoj listi. Ovo značajno rasterećuje samu aplikaciju i omogućava bržu dostupnost željenih datoteka.

Uslužilac za brzo indeksiranje datoteka je realizovan kao zasebna aplikacija koja obavlja prethodno opisane funkcije indeksiranja za Micronas IPR Editor [2].

Za proveru validnosti rada uslužioca za brzo indeksiranje datoteka kreirana je posebna aplikacija za brzo i automatsko kreiranje, brisanje i preimenovanje datoteka. Korišćenjem ove aplikacije, jednostavno se može utvrditi tačnost ažuriranja liste datoteka, tj. korektnosti rada uslužioca za brzo indeksiranje datoteka.

Uslužilac za brzo indeksiranje datoteka kao i osnovni program Micronas IPR Editor [2], realizovani su u programskom jeziku JAVA.

## II. REALIZACIJA KLASA

Realizacija uslužioca za brzo indeksiranje datoteka u programskom jeziku JAVA omogućila je podelu funkcionalnosti uslužioca na specifične klase i njihove objekte, i na taj način se povećala apstrakcija i lakoća razumevanja nadležnosti pojedinih klasa u okviru samog uslužioca.

### A. Klasa *IndexMaster*

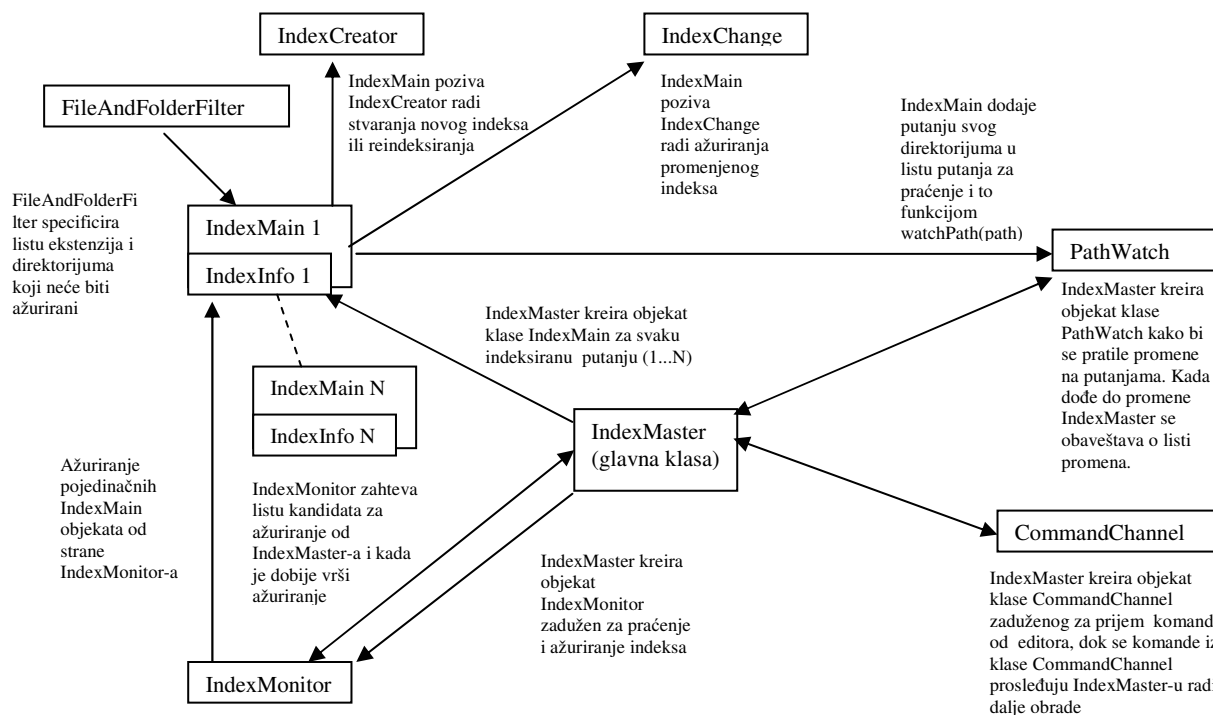
Klasa *IndexMaster* predstavlja osnovnu klasu aplikacije uslužioca. Ova klasa ima sledeće uloge:

- povezuje komandi, lista i zadataka svih njenih pod-klasa
- pokreće nit klase *PathWatch* koja je zadužena za praćenje promena na putanjama koje se dodaju u listu
- pokreće nit klase *CommandChannel*, koja predstavlja apstrakciju komandnog kanala za primanje komandi od strane Micronas IPR Editora [2], i slanja odgovora editoru
- pokreće nit klase *IndexMonitor*, koja predstavlja raspoređivač zadataka dobijenih na osnovu komandi od osnovne aplikacije (Micronas IPR Editora [2]), kao i zadataka koji se ponavljaju u određenim vremenskim intervalima
- kreira liste objekata klasa *IndexMain* i *IndexInfo* koje predstavljaju apstrakciju samih indeksiranih putanja, njihovih stanja kao i informacije o njima

Organizacija klasa i odnosi klasa dati su na slici 1:

---

Rad je delimično podržan u okviru projekta TR-6163B Ministarstva za nauku i zaštitu životne sredine Republike Srbije.  
 Satarić Bogdan, Fakultet tehničkih nauka u Novom Sadu, Srbija (e-mail: [bogdan.satavic@micronasnit.com](mailto:bogdan.satavic@micronasnit.com)).  
 Dejan Stefanović, MicronasNIT, Novosadski Institut za Informacione Tehnologije, Novi Sad, Srbija (e-mail: [dejan.stefanovic@micronas.com](mailto:dejan.stefanovic@micronas.com)).  
 Dragan Simić, MicronasNIT, Novosadski Institut za Informacione Tehnologije, Novi Sad, Srbija (e-mail: [dragan.simic@micronas.com](mailto:dragan.simic@micronas.com)).



Sl. 1. Organizacija klasa i njihovi odnosi

### B. Klasa PathWatch

Ova klasa predstavlja klasu pratioca putanja određenih direktorijuma i promena na datom direktorijumu. Funkcionalnost klase PathWatch je realizovana kao programska nit u kojoj se proverava da li je došlo do promene na određenoj putanji, i ako je došlo do promene – promena se dodaje u listu promena. Data lista se zatim prosleđuje glavnoj klasi IndexMaster i preko nje klasi IndexMonitor (zaduženom za raspoređivanje zadataka ažuriranja). Promene uključuju brisanje, preimenovanje i dodavanje datoteka i poddirektorijuma u okviru praćenog direktorijuma.

Same funkcije provere promena na putanjama i preuzimanja zadnje promene su realizovane u programskom jeziku C++. Od datih funkcija je formirana biblioteka filesystemwatcher.dll. Funkcije date biblioteke su uključene u JAVA aplikaciju kao nasleđene (native) funkcije preko Java Native interfejsa (JNI) [1]. Ovakav pristup je potreban iz razloga što je JAVA - programski jezik visokog nivoa apstrakcije, i za sada ne obezbeđuje sve funkcionalnosti niskog nivoa. U ovom slučaju funkcionalnost koja se morala obezbediti van JAVA-e jeste rad sa sistemom datoteka PC računara i registrovanje promena u sistemu datoteka. Radi početka praćenja stanja određene putanje, data putanja se dodaje u listu praćenih putanja funkcijom watchFolder(folderName). Radi provere da li je došlo do promene na određenoj putanji poziva se funkcija watchedFoldersChanged. Poslednja funkcija koja je bitna za praćenje promene stanja putanja jeste funkcija getLastChange, koja vraća zadnju izmenu u okviru direktorijuma, bilo da je u pitanje brisanje, pisanje ili preimenovanje datoteka.

### C. Klasa CommandChannel

S obzirom da je uslužioc za brzo indeksiranje datoteka aplikacija čiji rad direktno zavisi od komandi osnovne aplikacije Micronas IPR Editora [2], ukazala se potreba da se na određeni način povežu dve aplikacije, kako bi se obezbedilo pravilno i nesmetano slanje komandi. Rešenje je nađeno u vidu komandne datoteke koja je deljena između dve aplikacije (uslužioca i editora), a u koju se smeštaju komandne poruke i čitaju odgovori uslužioca sa editorske strane, i iz koje se čitaju komandne poruke i šalju odgovori uslužioca sa strane uslužioca. Ovakva implementacija zahteva pažljivo rukovanje pristupom komandnoj datoteci, kako ne bi došlo do istovremenog upisa i čitanja iz date datoteke, tj. kako se ne bi desilo da je datoteka u nekonzistentnom stanju.

Iz tog razloga uvodi se blokirajuća datoteka koja ima zadatak da onemogući istovremen pristup dveju aplikacija (editora i uslužioca) komandnoj datoteci. Princip funkcionisanja je da aplikacija koja prva dobije pristup blokirajućoj datoteci, preuzima pravo pristupa nad komandnom datotekom, i nakon toga upisuje ili čita iz iste. Nakon obavljene operacije čitanja komandi editora ili slanja odgovora uslužioca (u slučaju klase CommandChannel), tj. pisanja komandi editora i čitanja odgovora uslužioca (u slučaju funkcija Editor), blokirajuća datoteka postaje dostupna drugoj strani kako bi mogla da pristupi komandnoj datoteci.

Komande koje Editor može zadati uslužiocu jesu - indeksiranje nove putanje i upit o dostupnosti (postojanju) uslužioca za indeksiranje. Iako se još ne opisuje sama implementacija indeksiranja, potrebno je naglasiti da je indeks predstavljen objektima klase IndexMain i IndexInfo. Indeks se prvi put kreira za svaki direktorijum

na zahtev Micronas IPR Editora [2], pri izlistavanju sadržaja datog direktorijuma.

#### D. Klasa *IndexMonitor*

Ulogu dispečera zadataka indeksiranja, reindexiranja i ažuriranja indeksa preuzima klasa *IndexMonitor*. Ova klasa je realizovana kroz programsku nit koja ciklično proverava zahteve editora za indeksiranjem. Takođe, *IndexMonitor* proverava da li je došlo do izmena u određenim indeksima. Na kraju, zadatak klase *IndexMonitor* je i da sačuva stanje indeksiranih direktorijuma i to upisom parametara prethodno indeksiranih direktorijuma u posebnu datoteku (po jedna datoteka za datoteke i direktorijume). Parametri su na primer: broj datoteka u direktorijumu, broj pod-direktorijuma, dužina vremena indeksiranja itd.

Ukoliko je došlo do zahteva za indeksiranjem novog direktorijuma od strane editora, *IndexMonitor* formira nove objekte klase *IndexMain* i *IndexInfo* koji zajedno opisuju datu indeksiranu putanju. Takođe pri indeksiranju, data putanja se dodaje u listu putanja čije izmene se prate, funkcijom *addPathToWatchingList(path)* klase *IndexMaster*. Pri indeksiranju pamte se zasebno lista datoteka i lista direktorijuma. U listi direktorijuma nalaze se svi direktorijumi i njihovi pod-direktorijumi. Za svaki direktorijum ili pod-direktorijum u listi pamti se datum zadnje izmene.

Prilikom formiranja novog indeksa takođe se postavlja i vrednost perioda reindexiranja indeksa. Ovaj vremenski period ne bi trebalo da bude mali pošto bi velika učestanost reindexiranja opteretila aplikaciju a samim tim i procesor.

Reindexiranje se obavlja prolaskom kroz listu direktorijuma i pod-direktorijuma i upoređivanjem zapamćenog datuma zadnje izmene, sa datumom nove izmene. Ako su ti datumi različiti (došlo je do modifikacije direktorijuma), obavlja se njegovo reindexiranje. Reindexiranje se takođe obavlja periodično, bez obzira na promene u direktorijumima i to u periodu koji je zadat prilikom formiranja indeksa. Ovo je neophodno iz razloga što se na taj način čuvaju podaci o poslednjim izmenama na putanjama, u slučaju da dođe do nepredviđenih događaja tipa - gašenja računara ili gašenja aplikacije.

Objekat *IndexMain* može biti u stanjima *IDLE*, *INDEXING* ili *REINDEXING*. Data stanja predstavljaju u redosledu: inertno stanje, stanje prilikom indeksiranja i stanje prilikom reindexiranja putanje. Ova 3 stanja data su u klasi *IndexStates*.

Sve promene na indeksima registruju se tako što se proveravaju kandidati među postojećim indeksima, i to kandidati za:

- indeksiranje (ako je stanje objekta *IndexInfo* klase neke indeksirane putanje - "not indexed")
- ažuriranje (ako objekat klase *IndexMain* date putanje ima elemenata u vektoru tekućih izmena). Izmene se dodaju u ovaj vektor kao posledica registrovanja izmena od strane C++ funkcija klase *PathWatch*, a predstavljaju brisanje, preimenovanje i dodavanje datoteka.
- reindexiranje (ako je određeni direktorijum promenjen od prethodnog indeksiranja, tj. ako je neka datoteka u njemu promenjena)

Potrebno je naglasiti da su sve prethodno opisane akcije moguće jedino ako je objekat klase *IndexMain* (klase koja opisuje indeksiranu putanju) u stanju *IDLE*. Ovakav zahtev proističe iz činjenice da je jedino neaktivan indeks koji nije u stanjima *INDEXING*, *UPDATING* ili *REINDEXING*, moguće uvesti u neko od ta tri stanja, a ne pre toga. Takođe, u jednom trenutku je moguće izvršavanje dva zadatka simultano, bilo da je u pitanju indeksiranje ili reindexiranje. Ovakvo ograničenje je postavljeno kako ne bi došlo do preopterećenja procesora sa više istovremenih (re)indeksiranja.

Nakon prethodnih operacija moguće je zapisivanje svih prethodno obrađenih indeksa u posebnu datoteku.

#### E. Klasa *FileAndFolderFilter*

Uloga ove klase jeste filtriranje liste datoteka i pod-direktorijuma u okviru indeksiranog direktorijuma. Filtrira se navođenjem liste proširenja datoteka (u konfiguracionoj datoteci) koja će biti izostavljena prilikom indeksiranja direktorijuma.

#### F. Klasa *IndexMain*

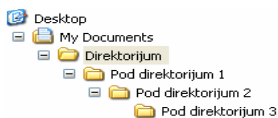
*IndexMain* predstavlja apstrakciju same indeksirane putanje i kao takva ova klasa je najbitniji deo aplikacije uslužioaca. Klasa *IndexMain* sadrži informacije o putanji koju predstavlja, a informacija se nalazi u objektu druge klase – *IndexInfo*, iz čega sledi da je objekat klase *IndexInfo* deo objekta klase *IndexMain*.

*IndexInfo* klasa sadrži informacije o putanji, stanju indeksa (indeksiran ili neindeksiran), broju datoteka u indeksu, broju direktorijuma u indeksu, vreme indeksiranja, dužini indeksiranja, vremenu reindexiranja i dužini reindexiranja.

Sledeći element klase *IndexMain* jeste vektor tekućih izmena spomenut ranije u radu. On sadrži listu promena na datotekama posmatranog direktorijuma i bitan je za funkciju ažuriranja indeksa. Za ažuriranje indeksa je uvedena nova klasa *IndexChange*. Metoda *doUpdate* ove klase omogućuje brisanje datoteka iz liste indeksiranih datoteka (ako je datoteka zaista izbrisana), zatim dodavanja datoteke u listu datoteka (ako je datoteka zaista i dodata) i sekvence brisanja stare datoteke i dodavanja nove preimenovane datoteke u slučaju da je datoteka u stvarnosti preimenovana (ovo eliminiše potrebu za uvođenjem specijalne operacije preimenovanja datoteke u listi).

Bitan korak u ažuriranju indeksa jeste provera da li je zapravo došlo do dodavanja pod-direktorijuma u okviru posmatranog direktorijuma. U ovom slučaju nije dovoljno jednostavno dodavanje samog pod-direktorijuma u listu, pošto je ceo proces indeksiranja vezan za direktno listanje datoteka, a ne direktorijuma (oni se kao takvi ne vide u listi indeksiranih datoteka). Stoga kada je dodat novi pod direktorijum – mora se izvršiti početna operacija indeksiranja tog pod-direktorijuma uz pomoć metode – *doIndexing*.

Funkcija indeksiranja datoteka je realizovana rekurzivno. Ovakva realizacija je optimalna s obzirom na dubinsku propagaciju pod-direktorijuma u okviru jednog direktorijuma, kao što je prikazano na slici 2:



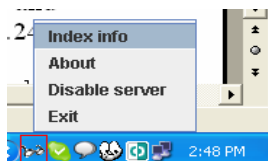
Sl. 2. Dubinska propagacija pod-direktorijuma

U praksi to znači da će metoda `doIndexing` dodavati datoteke u listu indeksiranih datoteka, dok će, u slučaju da je u pitanju direktorijum, pozvati samu sebe i istu funkciju ponavljati za datoteke i pod-direktorijume u okviru (pod)direktorijuma pozivaoca funkcije. U toku ažuriranja indeksa objekat `IndexMain` prelazi iz stanja `IDLE` u stanje `REINDEXING`.

Pored ažuriranja, dve moguće operacije jednog indeksa (predstavljenog objektom `IndexMain`) jesu indeksiranje i reindexiranje. Indeksiranje se događa samo prilikom prvog stvaranja indeksa direktorijuma na zahtev Micronas IPR Editora [2]. Reindexiranje se obavlja ako je došlo do modifikacije datoteke i periodično (određeno periodom reindexiranja). I indeksiranje i reindexiranje su opisani klasom `IndexCreator`.

U slučaju indeksiranja indeks prelazi iz stanja `IDLE` u stanje `INDEXING`, a u slučaju reindexiranja u stanje `REINDEXING`. Suština procesa indeksiranja i reindexiranja je ista a opisana je prethodno - rekurzivnom funkcijom `doIndexing`. Indeksirane putanje datoteka se u obliku bajta upisuju u datoteku indeksa. Ovo se čini kako bi se aplikacija učinila nezavisnom od platforme i operativnog sistema, s obzirom da je uslužioc za indeksiranje aplikacija koja će se koristiti ili se već koristi u većem broju računara kompanije Micronas GmbH. Klasa `IndexCreator` takođe izračunava razliku vremena između početka (re)indeksiranja i kraja (re)indeksiranja.

Po završetku indeksiranja, reindexiranja ili ažuriranja, obeštava se glavna klasa `IndexMaster`, kako bi mogla da se sačuva informacije o indeksu u datoteci. Ove informacije se mogu videti, pošto se uslužioc za indeksiranje pokreće kao pozadinska aplikacija operativnog sistema Windows. Ovo je moguće s obzirom da je JAVA aplikacija uslužioca pretvorena u izvršnu aplikaciju korišćenjem alata `Excelsior JET` [3]. Pristup uslužiocu za indeksiranje moguć je desnim klikom miša na ikonu u desnom donjem uglu ekrana. Ikona uslužioca je prikazana na slici 3, a zaokružena je crvenom bojom. U okviru menija moguće je izabrati informacije o indeksima, informaciju o verziji uslužioca, a takođe postoji mogućnost isključivanja uslužioca. Meni uslužioca za brzo indeksiranje je prikazan na slici 3:



Sl. 3. Meni uslužioca

### III. TESTIRANJE USLUŽIOCA

Testiranje uslužioca za brzo indeksiranje datoteka podrazumeva proveru korektnog indeksiranja, reindexiranja kao i ažuriranja specificiranih direktorijuma. Radi dobijanja rezultata tačnosti ažuriranja indeksa napravljena je zasebna testna aplikacija `RandomFileCreator`. Ova aplikacija služi za automatsko generisanje, preimenovanje i brisanje datoteka. Sekvenca prethodnih operacija se zadaje u konfiguracionoj datoteci a ponavlja se ciklično sa novim parametrima preimenovanja kada se završi jedan ciklus operacija. Operacije se obavljaju na svakih 300 milisekundi. Ovo je dovoljna učestanost da bi se mogla proveriti tačnost ažuriranih podataka.

Svako dodavanje, brisanje i preimenovanje datoteka je propraćeno promenom u samoj listi datoteka koja se vidi u okviru editora. Promene se ažuriraju gotovo u realnom vremenu, dovoljno brzo da korisnik editora ima na raspolaganju izmenjenu listu datoteka u svakom trenutku, odmah nakon izmena. Ovo je veoma bitno znajući da direktorijumi koji se indeksiraju za Micronas IPR Editor mogu biti deljeni [2], a samim tim i njihov sadržaj može biti podložan stalnoj promeni i to od strane više korisnika sa različitih računara.

### IV. ZAKLJUČAK

Opisana aplikacija predstavlja jednu izvedbu pozadinske aplikacije za praćenje liste datoteka u okviru proizvoljnog broja direktorijuma i pod-direktorijuma. Kao takva ova aplikacija oslobađa glavnu aplikaciju funkcije praćenja indeksa, i na taj način smanjuje opterećenje glavne aplikacije. S obzirom na tu osobinu ovakva aplikacija spada u stalne pozadinske procese, koji se obavljaju s vremena na vreme a služe za ažuriranje određenih informacija. Takve aplikacije se u operativnim sistemima nazivaju i demoni (daemons). Server je preko klase `CommandChannel` definisao spregu preko koje bi sa njim mogle da komuniciraju i druge aplikacije osim Micronas IPR Editora [2]. Takođe, uslužilac bi mogao biti smešten na poseban računar, kako bi više klijenata sa različitim aplikacijama mogli da koriste njegove usluge.

### LITERATURA

- [1] *JNI Technology online training* - <http://java.sun.com/>
- [2] *IPR Editor documentation, Micronas NIT, Novi Sad 2007*
- [3] *Excelsior JET Tutorial*, <http://www.xlsoft.com/en/index.html>

### ABSTRACT

This paper presents the application for fast file indexing. The application runs as background application, although it works in collaboration with the editor. As such, this application reduces the workload of editor and holds the latest list of indexed files. The list of indexed files can be passed on to editor (main application) on demand.

### FAST FILE INDEXING SERVER

Bogdan Satarić, Dejan Stefanović, Dragan Simić