Univerzitet Singidunum

Soketi

www.singidunum.ac.rs

MSc Miloš Mravik

Soketi

- Klijenti – programi koji otvaraju soket do servera koji osluškuje konekcije
- Klijentski soketi nisu dovoljni
- Klijenti nemaju mnogo svrhe ako ne komuniciraju sa serverima
- Klasa Socket nije dovoljna za pisanje servera
- Za kreiranje Socket-a mora se znati host sa kojim želimo da se povežemo
- Kada pišemo server, ne znamo unapred ko će nas kontaktirati, a čak i da znamo, ne znamo kada će to biti

Soketi

- Za servere, koji prihvataju konekcije, Java obezbeđuje klasu ServerSocket koja predstavlja serverske sokete
- Serverski soket se izvršava na serveru i osluškuje dolazeće TCP konekcije
- Svaki serverski soket osluškuje na određenom portu serverske mašine
- Kada klijent sa udaljenog hosta pokuša da se konektuje na taj port, server se budi, pregovara o konekciji između klijenta i servera i vraća regularni Socket objekat koji predstavlja soket između dva hosta
- Drugim rečima, serverski soket čeka na konekcije, dok klijentski soketi iniciraju konekcije

Klasa ServerSocket

- Sadrži sve što je potrebno za pisanje servera u Javi
- Ima konstruktore, metode koji osluškuju konekcije na zadatom portu, metode za konfigurisanje raznih opcija serverskog soketa, i uobičajene razne metode kao što je toString()
- ServerSocket osluškuje dolazeće pokušaje konekcija na tom portu koristeći svoj accept() metod. Metod accept() blokira dok klijent ne pokuša da napravi konekciju, kada accept() vraća Socket objekat koji povezuje klijenta i servera
- U zavisnosti od tipa servera, getInputStream(), getOutputStream() ili oba ova metoda se pozivaju za Socket objekat kako bi se komuniciralo sa klijentom

Životni ciklus programa

- Server i klijent interaguju u skladu sa dogovorenim protokolom do zatvaranja konekcije
- Server, klijent, ili obojica zatvaraju konekciju
- Server se vraća nazad i čeka narednu konekciju

Životni ciklus programa

- Ukoliko konekcija između klijenta i servera traje neograničeno u tom slučaju, java programi kreiraju niti za interakciju sa klijentima, tako da server može da procesira narednu konekciju
- Ako je protokol jednostavan i brz i dopušta da server zatvori konekciju kada završi, efikasnije je da server neposredno procesira zahtev klijenta, bez kreiranja niti

Životni ciklus programa

- I generisanje prevelikog broja niti može predstavljati problem. Za sistem sa oko 1GB RAM-a sve od približno 1000 niti će dramatično usporiti i izazvati da CPU često swap-uje podatke u i iz RAM-a.
- Generisanje prevelikog broja niti je jedan od nekoliko načina da se pouzdano sruši java VM
- ServerSocketChannel klasa obezbeđuje neblokirajući I/O zasnovan na kanalima, ne na tokovima. Sa kanalima, jedna nit može procesirati veći broj konekcija.

Životni ciklus programa

- O.S. smešta dolazeće konekcije za određeni port u FIFO red.
- Podrazumevana veličina je 50, ali može da varira od O.S. do O.S.
- Nakon što se red napuni, hostovi odbijaju dodatne konekcije za taj port dok se ne oslobodi mesto u redu
- Mnogi (ali ne svi) klijenti pokušavaju nekoliko puta da naprave konekciju ako inicijalni pokušaj bude odbijen
- OS rukuje dolazećim konekcijama i redom, mi o tome ne moramo da brinemo
- Nekoliko ServerSocket konstruktora omogućuje promenu veličine ovog reda, ali nije moguće povećati red iznad maksimuma koji podržava O.S.

Konstruktori

- public ServerSocket(int port) throws BindException, IOException
- public ServerSocket(int port, int queueLength) throws BindException, IOException
- public ServerSocket(int port, int queueLength, InetAddress bindAddress) throws IOException
- public ServerSocket() throws IOException
- Zadaje se port, dužina reda, local network interface za koji treba da se veže

Hvala na pažnji!

Pitanja?