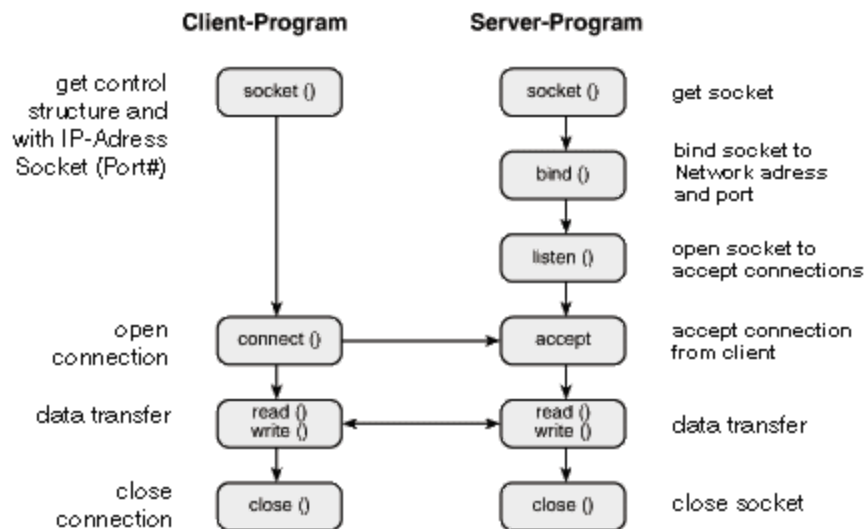


Tugas I IF 2220 Teori Bahasa Formal dan Otomata

Tanggal : 13 September 2017

Topik/Judul : Implementasi DFA untuk pemodelan UNIX Socket

Penjelasan Socket :



Gambar1. Cara Kerja Socket

Sumber: <https://www.wut.de/e-58www-16-apus-000.php>

Glosarium Domain Socket:

1. Socket : communication points on the same or different computers to exchange data. A socket is bound to a port number so that the TCP layer can identify the application that data is destined to be sent to.
2. Bind : assigning an address and port number to the socket
3. Listen : announce willingness to accept connections
4. Accept : block caller until a connection request arrives
5. Connect : actively attempt to establish a connection
6. Wait : wait the server to relpy for the request
7. Close : release a connection
8. Blocking : the default mode of socket calls is blocking. A socket is in blocking mode when an I/O call waits for an event to complete. If the blocking mode is set for a socket, the calling program is suspended until the expected event completes
9. Request : request some data over the connection (usually client to server)
10. Receive : receive some data over the connection (server reply request to client)

- 11. Read : read data from the connection
- 12. Write : write data to the connection
- 13. Abort : return error

Deskripsi Tugas :

Buatlah aplikasi yang memodelkan cara kerja UNIX socket sederhana. Model yang dibuat harus melingkupi semua state socket yang mungkin serta method yang mungkin dipanggil dalam komunikasi client-server sederhana yaitu:

1. Socket dapat melakukan aksi *bind* ke alamat tertentu dan kemudian melakukan aksi *listen* untuk mendeteksi koneksi baru yang datang
2. Socket dapat digunakan untuk membuka koneksi ke socket lain, dimana nantinya akan dapat mengirimkan pesan kepada socket tersebut
3. Socket juga dapat digunakan untuk membaca pesan yang telah dikirimkan oleh socket lain yang telah terhubung (harap juga modelkan state blocking I/O)
4. Socket dapat diatur untuk melakukan blocking / unblocking read
5. Socket dapat menutup koneksi ke socket lain atau terhadap binding

Aplikasi akan membuka file yang berisi informasi mengenai daftar state, daftar simbol, state awal, state akhir dan transition function. Informasi dari file tersebut akan digunakan untuk mengecek masukan dari pengguna. Harap diperhatikan bahwa program diwajibkan untuk membaca konfigurasi dari file eksternal, dan logika state machine tidak boleh di-*hardcode* ke program secara langsung.

Batasan masalah:

Ditekankan bahwa mahasiswa hanya diwajibkan untuk memodelkan perilaku socket yang didaftarkan di deskripsi masalah; semua fitur tambahan serta ketidaksesuaian perilaku model dengan socket sebenarnya akan digunakan sebagai bahan pertimbangan nilai akhir,

Implementasi :

- Program dibuat dalam bahasa pemrograman prosedural (C atau PASCAL).
- Program akan membaca file text yang berisi deskripsi state machine
- Program akan menerima masukan string aksi terhadap state machine; ex:
 - bind, listen, read, wait, return, close, connect, read, wait, abort, read, wait, wait, return, write, read, return, read, return, close
- Keluaran berupa lintasan state yang ditempuh berdasar masukan serta apakah masukan ditolak atau diterima.

Deliverables : 1. Dokumen laporan berisi:

deskripsi persoalan, DFA (notasi lengkap), penjelasan *state* dan aksi, *source code*, contoh masukan dan keluaran yang diterima oleh program

2. *source code* dan *file executable*

Semua file dalam *softcopy* dan dikirim kepada asisten kelas masing-masing.

Keterangan :

1. Tugas dibuat per kelompok dengan anggota maksimal 2 orang.
Bagi kelas yang memiliki jumlah peserta ganjil, satu kelompok boleh memiliki anggota 3 orang sisanya tetap diwajibkan memiliki anggota maksimal 2 orang.
2. Tugas dikumpulkan 27 September 2017 pukul 23.59.
3. Demo program akan ditentukan oleh asisten.

Referensi:

TutorialsPoint. (2017). *Unix Socket Tutorial*. Retrieved September,2017, from https://www.tutorialspoint.com/unix_sockets/index.htm

Oracle. (2016). *What Is a Socket*. Retrieved September,2017, from <https://docs.oracle.com/javase/tutorial/networking/sockets/definition.html>

Kosmas, Eleftherios. (2012). *Introduction to Sockets Programming in C using TCP/IP*. Retrieved September,2017, from <http://www.csd.uoc.gr/~hy556/material/tutorials/cs556-3rd-tutorial.pdf>

IBM Corporation. (2014). *Client/server socket programs: Blocking, nonblocking, and asynchronous socket calls*. Retrieved September, 2017, from https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.1.0/com.ibm.zos.v2r1.hala001/rgbblockasyn.htm

Ingalls, Robert. *Socket Tutorial - The Client Server Model*. Retrieved September, 2017, from <http://www.cs.rpi.edu/~moorthy/Courses/os98/Pgms/socket.html>