

Spesifikasi Tugas Besar II
IF3130 Jaringan Komputer
Aplikasi *Messaging* Sederhana

A. Deskripsi Umum Tugas

Pada tugas kali ini, Anda diminta untuk membuat sebuah aplikasi *messaging* sederhana berbasis server-klien (seperti WhatsApp atau LINE) dengan karakteristik sebagai berikut:

1. Berbasis server-klien. Aplikasi klien akan mengirimkan pesan ke server, lalu meneruskannya ke aplikasi klien tujuan.
2. Dapat mengirim pesan perorangan maupun grup. Pesan bisa dikirim hanya ke satu orang tertentu, maupun ke seluruh anggota sebuah grup tertentu.
3. Pesan yang dikirimkan tidak disimpan di server, **kecuali** klien yang dituju sedang *offline*. Misalkan klien A mengirim pesan ke klien B. Jika B *online*, maka server akan mengirimkan pesan tersebut ke B lalu menghapusnya. Jika B *offline*, server akan tetap menyimpan pesan tersebut sampai B *online* (disebut pesan *pending*).
4. Pesan berikut disimpan secara permanen (di dalam file):
 - Pesan yang telah diterima oleh klien
 - Pesan yang disimpan sementara oleh server karena klien tujuan *offline* (pesan *pending*)Pesan yang langsung diteruskan oleh server ke klien tujuan **tidak** disimpan pada file. Pesan *pending* yang dikirimkan setelah klien kembali *online* akan dihapus dari file.

Anda diminta untuk membuat aplikasi server dan klien dalam bahasa C/C++ pada lingkungan sistem operasi Linux. Socket yang digunakan adalah TCP, dengan menggunakan *library* socket bawaan Linux. Jika diperlukan, *library* thread yang dapat digunakan adalah pthreads (C/C++ standard) atau std::thread (C++11).

B. Deskripsi Fitur Wajib

Pada bagian berikut akan dijelaskan fitur-fitur yang wajib diimplementasikan, terutama pada aplikasi klien. Diberikan pula contoh interaksi fitur untuk mempermudah penjelasan, tetapi Anda boleh membuat interaksi sendiri yang menurut Anda lebih baik.

1. Signup dan login

Pada awal dijalankan, user dapat memilih signup atau login pada aplikasi klien. Contoh (teks bergaris bawah adalah input user):

```
> signup
Name      : seylum
Password  : versisgood

Signup success!
```

<pre>> <u>login</u> Name : <u>seylum</u> Password : <u>versisgood</u> Login success!</pre>
<pre>> <u>login</u> Name : <u>seylum</u> Password : <u>abcd1234</u> Invalid username or password</pre>

2. Mengirim pesan

User dapat mengirim pesan ke user lain atau grup. Misal terdapat user lain bernama “inaho” dan grup “red-dot”.

<pre>> <u>message inaho</u> Message : <u>yes, I'd love to!</u> Message sent.</pre>
<pre>> <u>message red-dot</u> Message : <u>hello everyone!</u> Message sent.</pre>
<pre>> <u>message madoka</u> madoka doesn't exist.</pre>

3. Create, join, dan leave group

User dapat membuat, mengikuti, dan keluar grup. Grup yang tidak memiliki anggota (semua anggotanya sudah keluar) secara otomatis dihapus oleh server.

<pre>> <u>create red-dot</u> Group red-dot created.</pre>
<pre>> <u>join red-dot</u> Joining group red-dot.</pre>
<pre>> <u>leave red-dot</u> Leaving group red-dot.</pre>
<pre>> <u>create red-dot</u> red-dot already exists.</pre>
<pre>> <u>join red-dot</u> red-dot doesn't exist.</pre>

```
> leave red-dot  
You aren't member of red-dot.
```

4. Melihat percakapan

User dapat melihat percakapan antara user dengan user lain atau grup. Pesan yang belum pernah dibaca sebelumnya harus ditandai.

```
> show slaine  
[2014-11-01 19:12] slaine : Princess, are you free tomorrow?  
[2014-11-01 23:34] seylum : uhm...  
[2014-11-01 23:37] seylum : sorry slaine, but I'm with inaho  
now.  
----- New Message(s) -----  
[2014-11-02 07:43] slaine : oh  
[2014-11-02 07:43] slaine : HOW DARE YOU!! HOW DARE YOU!!!  
[2014-11-02 07:44] slaine : *gunshot*  
  
> show red-dot  
[2014-11-01 22:11] seylum joined group  
[2014-11-01 22:12] seylum : hello everyone!  
[2014-11-01 22:13] jessy76 : welcome seylum!  
----- New Message(s) -----  
[2014-11-02 07:21] walpurgisnacht left group  
[2014-11-02 09:17] ayase : so what is your all-time fav?  
[2014-11-02 09:24] qwerty367 : hagaren ofc  
[2014-11-02 09:24] mike_kun : all haremu ftw lol  
[2014-11-02 09:25] gerrard23 : mine will be gineiden then  
[2014-11-02 09:29] jack0 : free!! haruka <3 <3  
[2014-11-02 09:31] ayase : wha..  
[2014-11-02 09:31] ayase : jack-senpai, I... I really didn't  
expect... bit.ly/10yg9k8  
[2014-11-02 09:40] jack0 : no! my sister hijacks me!!  
[2014-11-02 09:40] jack0 : NOOOOOOOOOOOOO
```

5. Notifikasi

User diberi pemberitahuan / notifikasi jika terdapat pesan baru. Jika menggunakan CLI, pesan notifikasi muncul setelah input terakhir diproses. Contoh

```
(Kasus: ketika seylum belum selesai menulis pesan, ada pesan baru dari red-dot dan  
slaine)  
> message inaho  
Message :  
yes, I'd love to!  
  
Message sent.  
New message(s) from red-dot and slaine.  
  
(Kasus: ketika seylum login, ada pesan pending dari red-dot)  
> login  
Name : seylum  
Password : versisgood
```

```
Login success!  
New message(s) from red-dot.
```

6. Logout

Klien memberitahu server bahwa user logout, lalu keluar.

7. Server log

Terdapat log aktivitas yang terjadi di server. Log **tidak** boleh memuat pesan user. Contoh

```
[2014-11-01 10:00] Server starts  
[2014-11-01 22:00] seylum logged in  
[2014-11-01 22:11] seylum joined red-dot  
[2014-11-01 22:12] seylum messages red-dot  
[2014-11-01 22:13] jessy76 messages red-dot  
[2014-11-01 22:14] seylum messages inaho  
[2014-11-01 23:20] jessy76 logged out  
[2014-11-01 23:34] seylum messages slaine  
[2014-11-01 23:37] seylum messages slaine  
[2014-11-01 23:40] seylum logged out  
[2014-11-02 07:17] walpurgisnacht logged in  
[2014-11-02 07:21] walpurgisnacht left red-dot
```

8. File pesan, daftar user, daftar grup dan server log.

Pesan disimpan pada file sesuai dengan ketentuan pada bab sebelumnya. Daftar user-password, daftar keanggotaan grup dan server log disimpan pada file di server. Format file dibebaskan.

C. Deskripsi Fitur Bonus

Adapun fitur bonus yang dapat dikerjakan adalah:

1. Pendeteksian klien mati tanpa *logout*. Server dapat mengetahui klien yang tiba-tiba mati tanpa sebelumnya logout (misal: komputer klien mati atau putus dari jaringan). Jika ada klien yang tiba-tiba mati, maka klien dianggap *offline* oleh server.
2. Memungkinkan pengiriman file melalui chat. Definisikan protokol dan interaksi untuk pengiriman file, sebisa mungkin konsisten dengan protokol dan interaksi chat biasa.
3. Membuat fitur “read”. Aplikasi klien dapat melihat apakah pesan yang dikirimkan sudah dibaca oleh klien penerima atau belum. Definisikan protokol dan interaksi (termasuk cara menampilkannya) untuk fitur ini, sebisa mungkin konsisten dengan protokol dan chat biasa.

D. Format Laporan

Untuk tugas kali ini Anda perlu menulis laporan yang berisi:

1. Deskripsi permasalahan, singkat saja maksimal 2-3 paragraf.
2. Desain protokol komunikasi, yaitu menjelaskan aturan komunikasi antara server dengan klien dan format data yang dikirimkan untuk setiap fitur yang dibangun. Visualisasikan langkah-langkah protokol komunikasi dan format data jika memungkinkan.
3. Detail singkat implementasi server dan klien, misalnya menjelaskan cara kerja secara umum, thread yang berjalan dan tugasnya, port-port yang dibuka, dan sebagainya.

4. Bagian paling sulit dan paling menarik.
5. Pembagian kerja.

E. Pengumpulan Tugas

Buatlah kelompok beranggotakan 3 orang dan daftarkan kelompok Anda pada link berikut (<http://bit.ly/jarkomtubes2>). Pengumpulan tugas akan dilakukan di situs milestone pada **Selasa, 2 Desember 2014 pukul 23.59**. Adapun *deliverables* yang perlu dikumpulkan adalah:

1. Folder src: berisi seluruh source code program (*.c, *.cpp, *.h, *.hpp)
2. Folder bin: berisi file *binary* program hasil kompilasi
3. Folder doc: berisi laporan **dalam format pdf**
4. File Makefile untuk kompilasi source code
5. File Readme untuk menjelaskan cara penggunaan program

Deliverables dikumpulkan dalam arsip zip dengan format nama **TB2_XX_YY.zip**, dengan XX nomor kelas dan YY nomor kelompok sesuai pendaftaran.

F. Referensi

Pemrograman socket:

- http://www.linuxhowtos.org/C_C++/socket.htm
- <http://www.yolinux.com/TUTORIALS/Sockets.html>

Pemrograman Thread:

- <http://www.yolinux.com/TUTORIALS/LinuxTutorialPosixThreads.html> (pthread)
- http://www.tutorialspoint.com/cplusplus/cpp_multithreading.htm (pthread)
- <http://baptiste-wicht.com/posts/2012/03/cpp11-concurrency-part1-start-threads.html> (std::thread)
- <https://www.justsoftwaresolutions.co.uk/threading/multithreading-in-c++0x-part-1-starting-threads.html> (std::thread)

Contoh pemrograman socket:

- <https://github.com/faizilham/JarkomSampleSocket> (server-client sederhana)

Contoh pemrograman thread:

- <https://github.com/faizilham/JarkomThreadingSample> (producer-consumer sederhana)