

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Báo cáo môn
thực hành lập trình mạng
Nhóm

Nhóm sinh viên thực hiện:

1. Nguyễn Ngọc Anh	20120035	Việt Nhật IS K57
2. Đặng Mỹ Linh	20121974	Việt Nhật IS K57

Giảng viên hướng dẫn: Thầy Bùi Trọng Tùng
Hà Nội, tháng 12 năm 2015

Mục lục

A. Giới thiệu chung

B. Lí thuyết

C. Phân tích cách thực hiện

1. Server:

2. Client

3. Cách truyền tin

D. Kết quả chạy thử:

A. Giới thiệu chung:

1. Đề bài:

(17) Xây dựng dịch vụ từ điển trực tuyến: Dịch vụ cho phép người dùng tra và thay đổi nội dung từ điển.

2. Giới thiệu chương trình:

- Chương trình từ điển cho phép người dùng sử dụng những chức năng: Tra từ, thêm từ, xoá từ, đăng nhập, đăng kí, đăng xuất.
- Phía server sử dụng database của mysql, phía client sử dụng gtk để làm giao diện người dùng.
- Server và client sử dụng kĩ thuật **polling server** để kết nối.

3. Tóm tắt nội dung báo cáo:

- Phần giới thiệu mô hình server-client sử dụng: Nêu lên loại hình server, kĩ thuật lập trình mạng.
- Phần phân tích cách thực hiện: Server, client, kịch bản truyền tin.
- Phần kết quả chạy thử: Chạy thử chương trình, test thử cho một server với nhiều client.
- Phần nhận xét: Nhận xét về chương trình và phân công công việc.

B. Lý thuyết mô hình server-client sử dụng:

Chương trình của nhóm em sử dụng Socket API, giao thức TCP và kỹ thuật poll server.

1. Về socket và TCP:

Trong lập trình, Socket là một API (Application Programming Interface) cung cấp các phương thức để giao tiếp thông qua mạng. Nó cho phép chúng ta chọn giao thức ở tầng giao vận cho ứng dụng của mình.

Socket có thể hoạt động theo 2 chế độ:

-*stream sockets* : sử dụng giao thức có nối kết TCP để tạo ra một kênh giao tiếp ảo giữa 2 bên giao tiếp bảo đảm truyền tin đầy đủ, chính xác.

-*datagram socket*: sử dụng giao thức không nối kết UDP do đó cung cấp một dịch vụ truyền tin nhanh nhất.

2. Về lập trình server sử dụng poll():

Có 3 loại server thường được sử dụng đó là

- Iterating server: chỉ một kết nối tại một thời điểm
- Forking server: sau khi kết nối chuyển điều khiển đến chương trình con giống hệt chương trình cha. Cho phép nhiều kết nối tại một thời điểm.
- Concurrent single server: sử dụng để chờ đợi đồng thời trên tất cả các socketID và chỉ đánh thức khi có dữ liệu đến. Cho phép nhiều kết nối tại một thời điểm.

Để server có thể phục vụ nhiều client một lúc có thể sử dụng Forking server hoặc *Concurrent single server*.

So sánh forking server và *Concurrent single server*

	Forking server	<i>Concurrent single server</i>
MÔ TẢ	Nhiều tiến trình (multi-process) chạy đồng thời	Một tiến trình chạy xử lý nhiều kết nối.
ƯU ĐIỂM	Dễ cài đặt Cần dựa vào hệ điều hành đa nhiệm để có thể đồng thời.	Không cần Ra vào database một cách có thứ tự.
NHƯỢC ĐIỂM	Có thể tạo ra quá nhiều tiến trình Dễ khiến máy tính quá tải. Khó trong việc sử dụng một database	Khó cài đặt hơn forking server. Dễ bị chặn khi server quá tải.

Cài đặt server theo chế độ *Concurrent single server*. Thông qua hàm *select()* hoặc hàm *poll()* quy định rõ một danh sách mô tả dùng để kiểm tra các hoạt động cần vào ra trên các socket. Chế độ này sẽ đặt chương trình ở trạng thái chờ cho đến khi có một socket nào đó có tín hiệu vào ra dữ liệu đối với server, sau đó trả về số hiệu của socket có tín hiệu vào ra cho chương trình chính.

Hàm *poll()* và *select()* thực hiện chức năng về cơ bản là giống nhau.

Hàm *select()* sử dụng một array chứa nhiều bits trong đó mỗi bit là một mô tả (descriptor number).

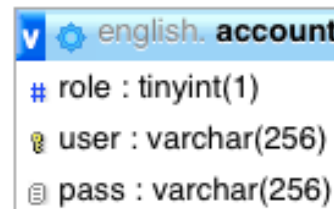
Còn hàm *poll()* cho phép ứng dụng gửi một array chứa nhiều cấu trúc *pollfd*.

```
struct pollfd {
    int    fd;           /* file descriptor */
    short  events;       /* requested events */
    short  revents;      /* returned events */
};
```

C. Phân tích cách thực hiện:

1. Phía server:

-Thiết kế database:



- Các bảng:

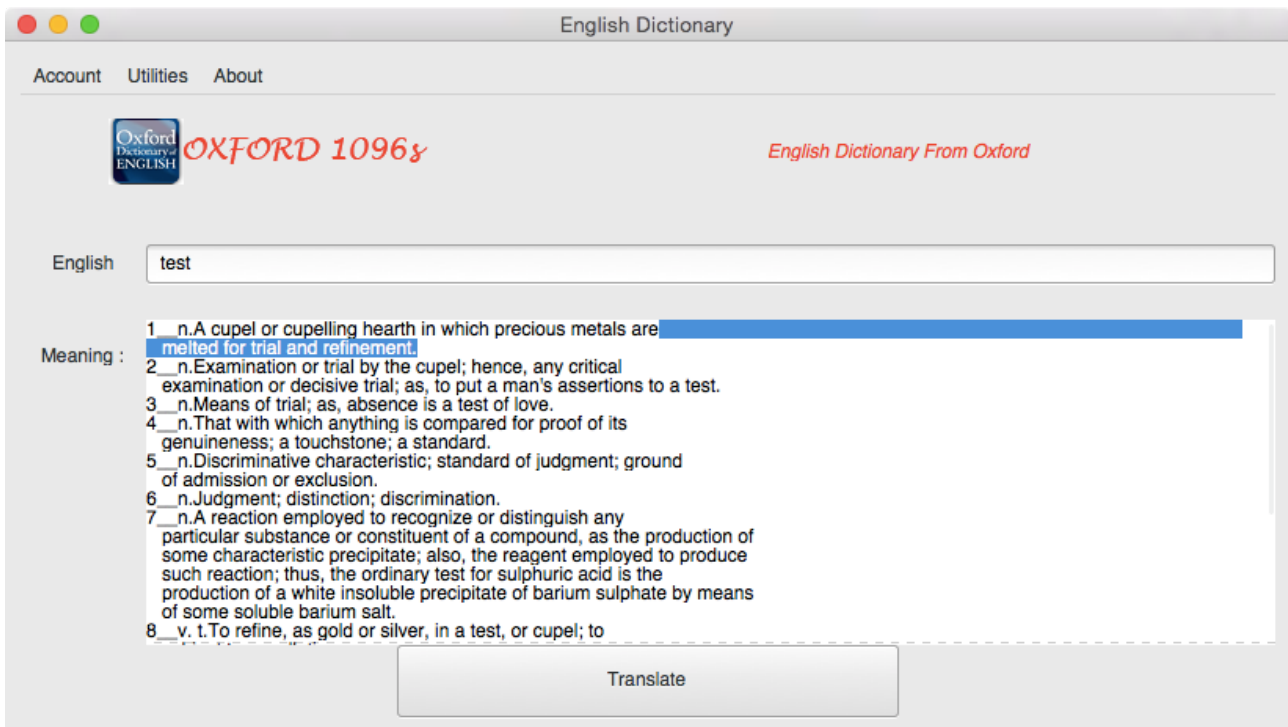
- Bảng **entries**: chứa các từ tiếng Anh để tra cứu.
- Bảng **account**: chứa thông tin người dùng.
- Bảng **modify**: chứa thông tin thêm/xoá của người dùng.

-Code một số chương trình con:

- Hàm **adds(), removes(), login(),logout(),search()...** : thực hiện các chức năng add/ remove/ login/ logout/ register...
- Hàm **AddSql(), RemoveSql ()...** : tạo các câu lệnh SQL xoá, thêm, select...
- Hàm **resultEV(), resultEC()** : gửi kết quả các câu lệnh Select về cho Client.
- Hàm **connectSQL()** : kết nối tới database MySQL.
- Hàm **finish_with_error(), err_sys()**: hàm báo lỗi sql và lỗi kết nối.

2. Phía Client:

-Xây dựng giao diện sử dụng GTK +-3.0:



-Code một số chương trình con:

- Hàm **XX_window()** : chứa code về giao diện GTK. Trong đó XX là add/ remove/ login/ logout/ register ứng với các cửa sổ tương ứng.
- Hàm **XX_function()** : thực hiện các chức năng xóa, thêm, đăng nhập...
- Hàm **transfer_()** : tạo các string có dạng 1%*****.
- Hàm **makegtk()** : tạo giao diện chung cho chương trình.
- Hàm **dialog_()** : tạo cửa sổ thông báo.
- Hàm **checkSQL()** : kiểm tra string gửi đi có chứa các kí tự đặc biệt hay không.

3. Cách truyền tin:

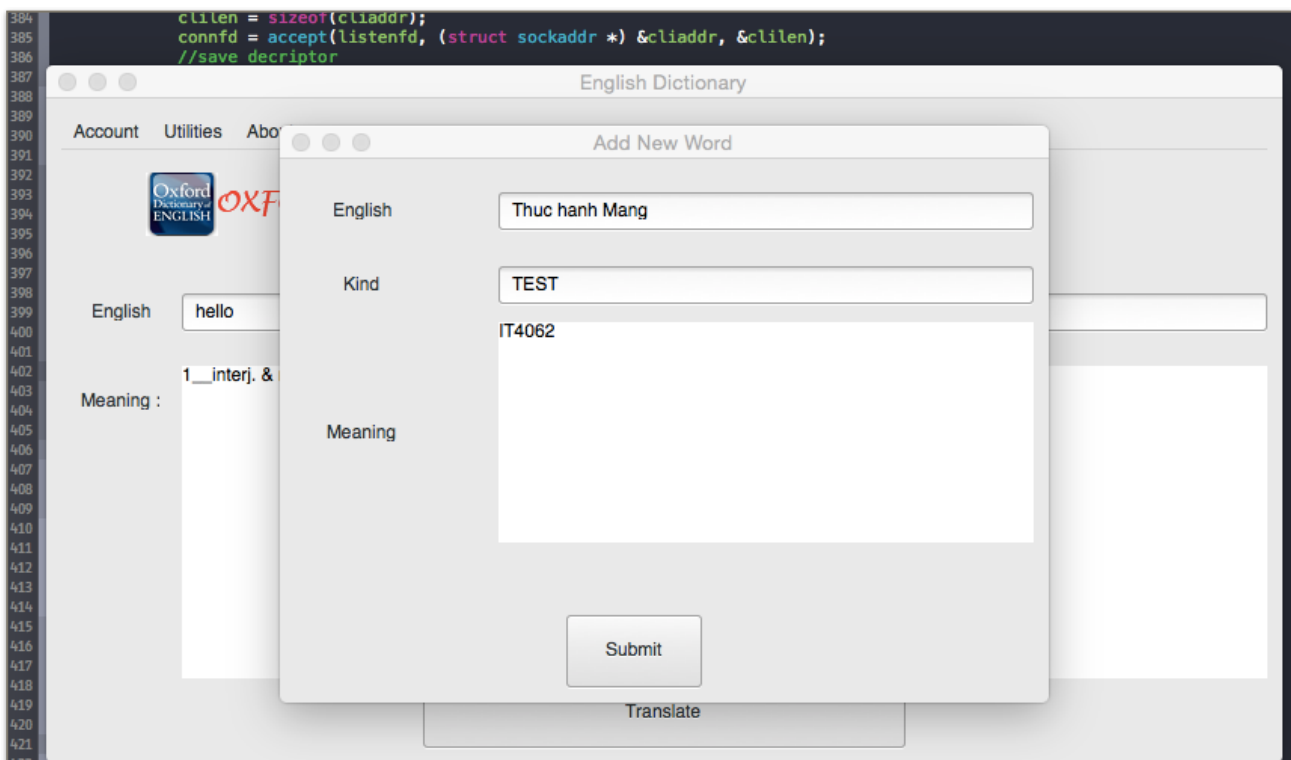
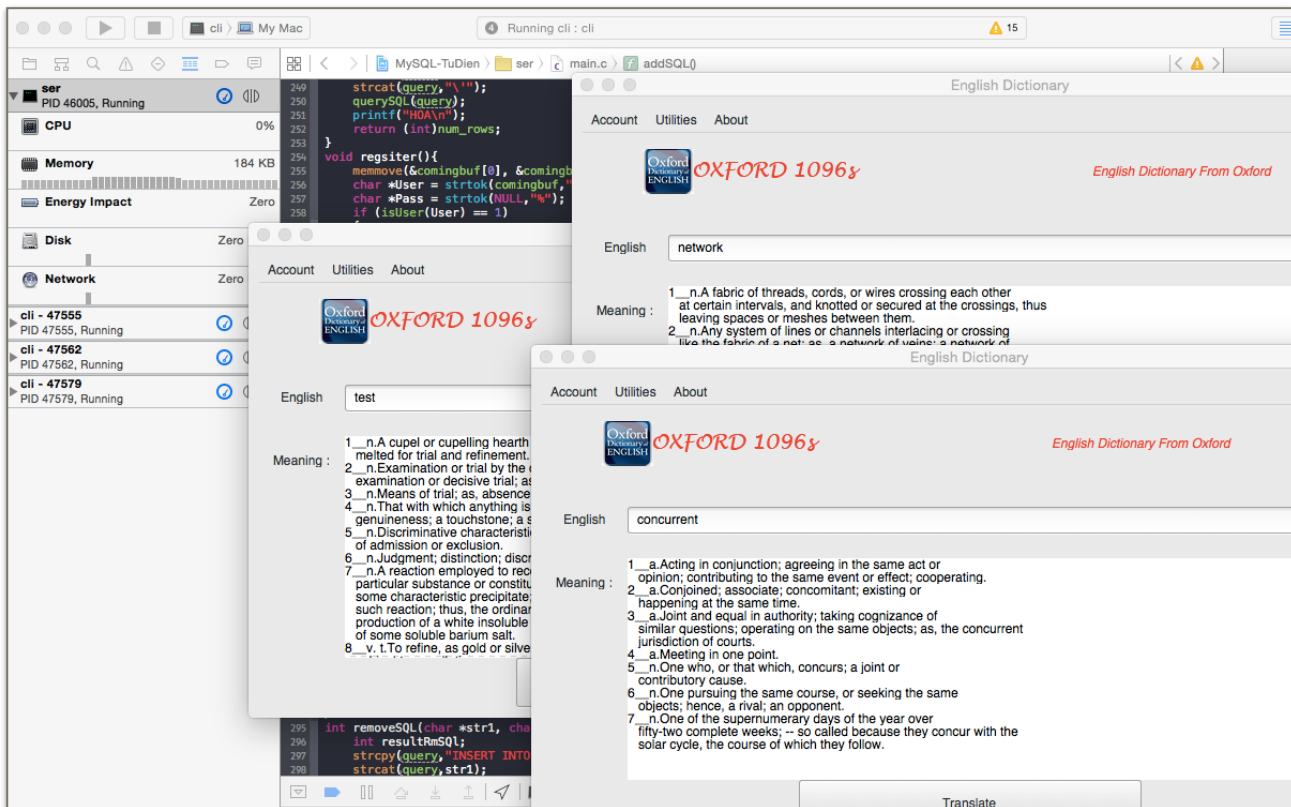
-Khuôn dạng mẫu tin:

Client gửi	CHÚ THÍCH	CHỨC NĂNG	Server trả	CHÚ THÍCH
0%XX	XX là từ để tra	tra từ có trong từ điển	Kết quả truy vấn SQL	Nghĩa của từ trong bảng entries
1%XX%YY	XX là tên người dùng YY là từ	tra từ trong bảng modify	Kết quả truy vấn SQL	Nghĩa của từ trong bảng modify hoặc entries
2%XX%YY	XX là tên người dùng YY là password	kiểm tra người dùng khi đăng nhập	1 hoặc 0	1 - login thành công 0 - login thất bại
3%XX%YY	XX là tên người dùng YY là password	Đăng ký tài khoản mới	1 hoặc 0	như trên
4%XX%YY %ZZ	XX là tên người dùng YY là từ ZZ là nghĩa	Thêm từ mới vào modify	1 hoặc 0	như trên
5%XX%YY	XX là tên người dùng YY là từ	Xoá từ	1 hoặc 0	như trên
6%XX	XX là tên người dùng	Đăng xuất		không trả về

-Cách truyền tin:

- Ở *client*: Mỗi khi thực hiện một chức năng thì gọi hàm **connectSK()** để tạo kết nối mới. Sau khi gửi/nhận tin với server thì gọi hàm **close(sockfd)** để đóng kết nối.
- Ở *server*: Chỉ thực hiện kết nối với CSDL một lần thông qua hàm **connectSQL()**. Sau đó gọi hàm **poll()** đợi kết nối mới. Trong khi thực hiện kết nối này thì kết nối khác sẽ bị chặn tạm thời.

C. Kết quả chạy thử:



D. Nhận xét và phân công công việc:

1. Nhận xét:

- Ưu điểm:
 - +Đã kết nối được giữa Client Và Server , không gây ra lỗi khi Connect
 - +Thực hiện được việc đăng nhập ,đăng ký, đăng xuất.
 - +Thực hiện tra từ, thêm từ, xoá từ.
- Nhược điểm:
 - + Còn có nhiều lỗi.
 - + Giao diện đơn giản chưa thực sự đầy đủ thành phần. Các chức năng có thể phải bổ sung thêm.

2. Phân công công việc:

Nguyễn Ngọc Anh:

- *Tìm hiểu IDE và tool phát triển cho việc lập trình*
- *Xây dựng giao diện trên GTK*
- *Xây dựng giao tiếp Client-Server*

Đặng Mỹ Linh:

- *Tạo cơ sở dữ liệu*
- *Viết báo cáo*
- *Kiểm thử chương trình*
- *Xây dựng lý thuyết chương trình*