

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN



21120161 - TIÊU ÂN TUẤN

BÁO CÁO BÀI TẬP THỰC HÀNH 2
SOCKET

| Giảng viên hướng dẫn |

Thầy Lê Quốc Hòa

Thành phố Hồ Chí Minh – 2024

MỤC LỤC

MỤC LỤC	1
1. Ý tưởng thực hiện	2
1.1. Client.....	2
1.1.1. Định nghĩa các hằng số và biến:	2
1.1.2. Tạo socket và kết nối đến Server:	2
1.1.3. Vòng lặp để nhập và gửi biểu thức:	2
1.1.4. Đóng kết nối và kết thúc chương trình:	3
1.2. Server	3
1.2.1. Định nghĩa các hằng số.....	3
1.2.2. Khai báo biến và hàm hỗ trợ:	3
1.2.3. Hàm kiểm tra biểu thức <i>checkExpression()</i> :.....	3
1.2.4. Hàm tính toán biểu thức <i>calculateExpression()</i> :.....	4
1.2.5. Hàm xử lý cho Client <i>handle_client()</i> :	4
1.2.6. Hàm <i>main()</i> :.....	4
2. Kết quả chạy chương trình.....	5
2.1. Test case 1 (1 Server 1 Client).....	5
2.2. Test case 2 (1 Server n Client).....	8
3. Đánh giá mức độ hoàn thành	11

1. Ý tưởng thực hiện

Ý tưởng chung: Để Server có thể kết nối được nhiều Client cùng lúc em sử dụng kết hợp giữa Socket và Thread để tạo nhiều tiểu trình mỗi tiểu trình là 1 Client. Ngoài ra để tính toán biểu thức từ chuỗi được gửi đến client sử dụng một hàm để tính toán các phép toán sử dụng ngăn xếp (Stack) để chứa các toán tử và số của phép toán phục vụ cho việc tính toán theo độ ưu tiên.

Thư viện: Sử dụng các thư viện chuẩn như như

- **stdio.h** cho các thao tác đầu vào/đầu ra tiêu chuẩn.
- **stdlib.h** cho các hàm cấp phát bộ nhớ và thoát, **string.h** cho các hàm xử lý chuỗi.
- **ctype.h** cho các hàm xử lý ký tự.
- **arpa/inet.h, unistd.h, pthread.h** để cung cấp các chức năng liên quan đến lập trình socket và đa luồng.

1.1. Client

Chương trình Client cho phép người dùng nhập một biểu thức toán học, gửi nó đến Server để tính toán, và sau đó hiển thị kết quả nhận được.

1.1.1. Định nghĩa các hằng số và biến:

- **PORT** là hằng số với giá trị 8000, chỉ định số cổng mà Server đang lắng nghe.
- **MAX_BUFFER_SIZE** là hằng số với giá trị 256, quy định kích thước tối đa của bộ đệm được sử dụng để gửi và nhận dữ liệu.
- **clientSocket** là biến để lưu trữ mô tả socket.

1.1.2. Tạo socket và kết nối đến Server:

- Sử dụng hàm *socket* để tạo ra một socket mới.
- Đặt địa chỉ Server trong biến **ServerAddr** và kết nối đến Server bằng hàm *connect*.

1.1.3. Vòng lặp để nhập và gửi biểu thức:

- Sử dụng vòng lặp *while* để client có thể liên tục gửi biểu thức cần tính đến cho Server mà không cần kết nối lại
- Dùng hàm *fgets* để đọc đầu vào và loại bỏ ký tự xuống hàng.

- Khi người dùng muốn ngắt kết nối với Server và kết thúc chương trình thì người dùng có thể nhập “exit”, chương trình sẽ ghi nhận gửi thông báo thoát đến cho Server và chạy lệnh break thoát khỏi vòng lặp để kết thúc.

- Sử dụng hàm *send* để gửi biểu thức đến Server và *recv* để nhận phản hồi từ Server.

1.1.4. Đóng kết nối và kết thúc chương trình:

- Sau khi kết thúc vòng lặp, sử dụng hàm *close* để đóng socket.

1.2. Server

Chương trình Server cho phép nhiều client kết nối vào cùng 1 lúc và xử lý các biểu thức toán học được gửi từ các client, trả về kết quả là giá trị nếu biểu thức hợp lệ và báo lỗi vị trí có ký tự không hợp lệ trong biểu thức nếu biểu thức không hợp lệ.

1.2.1. Định nghĩa các hằng số

- **PORT** là hằng số với giá trị 8000, chỉ định số cổng mà Server đang lắng nghe.

- **MAX_BUFFER_SIZE** là hằng số với giá trị 256, quy định kích thước tối đa của bộ đệm được sử dụng để gửi và nhận dữ liệu.

1.2.2. Khai báo biến và hàm hỗ trợ:

- Định nghĩa hai hàm hỗ trợ *precedence()* nhận một toán tử làm đầu vào (+ - * /) và trả về mức độ ưu tiên tương ứng (1 cho + và -, 2 cho * và /, và 0 cho những ký tự khác).

- Định nghĩa hàm hỗ trợ *calculate()* nhận hai số và một toán tử làm đầu vào và thực hiện phép toán tương ứng.

1.2.3. Hàm kiểm tra biểu thức *checkExpression()*:

- Hàm này kiểm tra tính hợp lệ của biểu thức toán. Nó lặp qua từng ký tự của biểu thức và kiểm tra xem đó có phải là một ký tự hợp lệ không (một chữ số, một toán tử hoặc dấu ngoặc). Nếu một ký tự không hợp lệ được tìm thấy, hàm trả về chỉ số của ký tự đó trong biểu thức. Nếu biểu thức hợp lệ, hàm trả về giá trị 0.

1.2.4. Hàm tính toán biểu thức *calculateExpression()*:

- Hàm này dùng để tính toán biểu thức được đưa vào. Nó sử dụng hai mảng dưới dạng ngăn xếp (Stack), *numbers* và *operators*, để lưu trữ số và toán tử gặp phải trong quá trình phân tích biểu thức. Dùng vòng lặp, lặp qua từng ký tự của biểu thức và thực hiện:

- Nếu ký tự là một chữ số, nó chuyển đổi các chữ số liên tiếp thành một số và đẩy nó vào ngăn xếp *numbers*.

- Nếu ký tự là dấu ngoặc mở, nó đẩy nó vào ngăn xếp *operators*.

- Nếu ký tự là dấu ngoặc đóng, nó lặp đi lặp lại việc lấy các toán tử từ ngăn xếp *operators* và số từ ngăn xếp *numbers*, thực hiện phép toán tương ứng và đẩy kết quả trở lại ngăn xếp *numbers* cho đến khi gặp dấu ngoặc mở.

- Nếu ký tự là toán tử, nó lặp đi lặp lại việc lấy các toán tử từ ngăn xếp *operators* và số từ ngăn xếp *numbers*, thực hiện phép toán tương ứng và đẩy kết quả trở lại ngăn xếp *numbers* cho đến khi gặp toán tử có mức độ ưu tiên thấp hơn hoặc gặp dấu ngoặc mở.

- Nếu ký tự không phải là chữ số, toán tử hoặc dấu ngoặc, hàm trả về 0 để chỉ đánh dấu lỗi.

Sau khi xử lý tất cả các ký tự, hàm thực hiện các phép toán còn lại bằng cách lấy các toán tử từ ngăn xếp *operators* và số từ ngăn xếp *numbers*, thực hiện phép toán tương ứng và đẩy kết quả trở lại ngăn xếp *numbers*. Cuối cùng, nó trả về kết quả ở đầu ngăn xếp *numbers* *numbers[0]*.

1.2.5. Hàm xử lý cho Client *handle_client()*:

- Hàm này được thực hiện trong một luồng riêng biệt cho mỗi Client được kết nối. Nó nhận một socket Client làm đối số. Trong hàm này, nó liên tục nhận dữ liệu từ Client tương ứng sử dụng hàm *recv()*. Nếu dữ liệu nhận được là "exit", nó sẽ đóng socket Client và thoát khỏi luồng. Nếu không, nó kiểm tra tính hợp lệ của biểu thức nhận được bằng cách sử dụng hàm *checkExpression()*. Nếu biểu thức không hợp lệ, nó gửi một thông báo lỗi trở lại cho Client. Nếu biểu thức hợp lệ, nó tính kết quả dựa trên biểu thức nhận được bằng hàm *calculateExpression()* và gửi kết quả trở lại cho Client.

1.2.6. Hàm *main()*:

- Tạo một socket Server bằng cách sử dụng hàm *socket()*.

- Thiết lập cấu trúc địa chỉ Server (*ServerAddr*) và số cổng được đặt là *PORT*.

- Liên kết socket Server với địa chỉ Server bằng cách sử dụng hàm *bind()*.
- Lắng nghe kết nối đến từ các Client sử dụng hàm *listen()*.
- Đi vào vòng lặp *while(1)* để chấp nhận các kết nối đến và xử lý chúng trong các luồng riêng biệt:

- + Chấp nhận một kết nối Client mới bằng cách sử dụng hàm *accept()*, trả về một socket mới cho Client.

- + Tạo một luồng mới để xử lý kết nối Client bằng cách sử dụng hàm *pthread_create()*, truyền socket Client như là đối số cho hàm *handle_client()*.

- + Tiếp tục chấp nhận các kết nối Client mới.

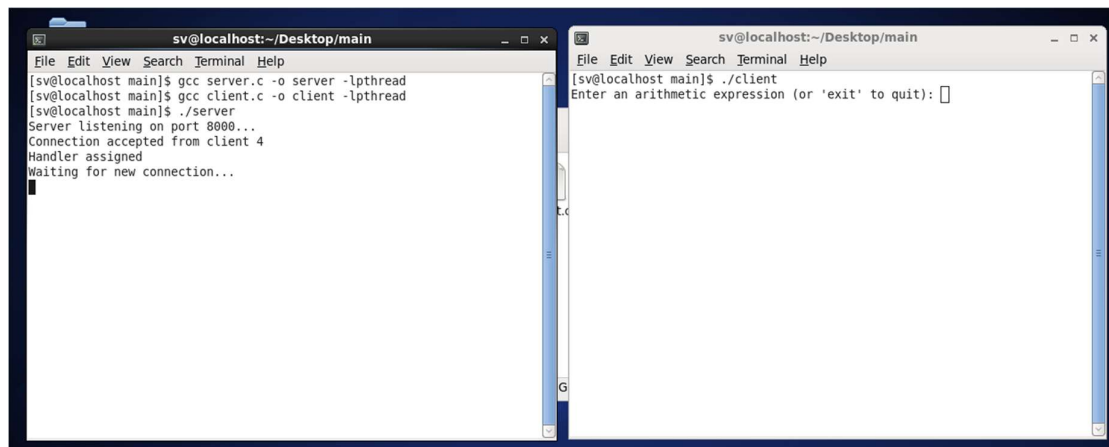
- Cuối cùng nếu người dùng đóng terminal, chương trình sẽ đóng socket Server và thoát khỏi chương trình.

2. Kết quả chạy chương trình

2.1. Test case 1 (1 Server 1 Client)

Nội dung test case 1: tạo 1 Server và 1 Client để chúng trao đổi với nhau, gửi biểu thức tính toán và trả về kết quả cho Client.

- Kết nối Client và Server thành công:



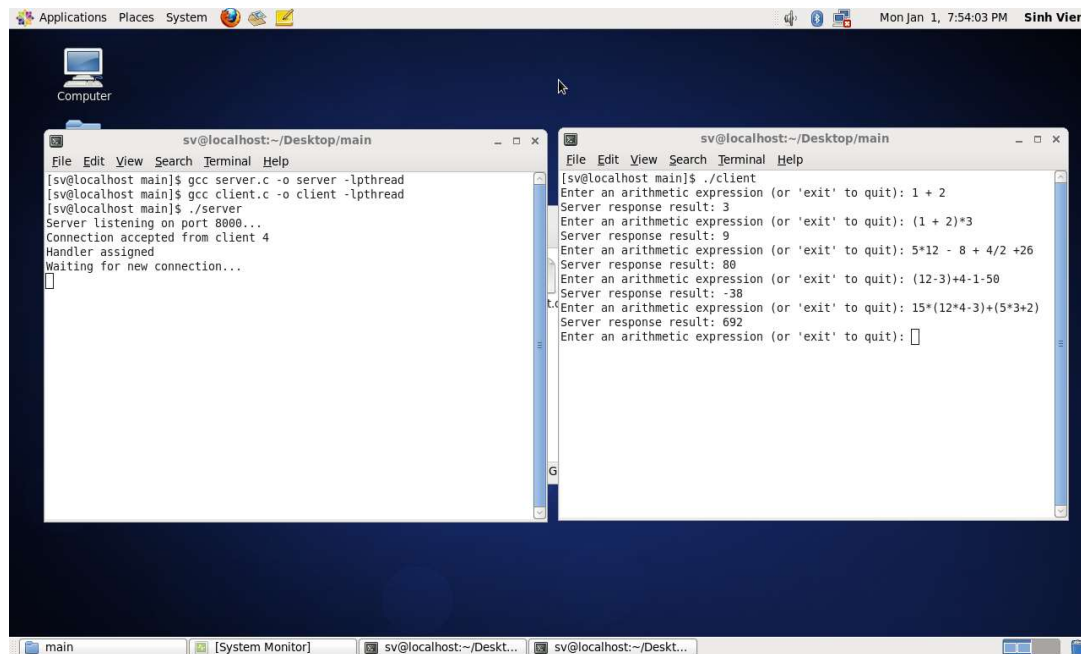
The image shows two terminal windows side-by-side. The left window is titled 'sv@localhost:~/Desktop/main' and contains the following text:

```
[sv@localhost main]$ gcc server.c -o server -lpthread
[sv@localhost main]$ gcc client.c -o client -lpthread
[sv@localhost main]$ ./server
Server listening on port 8000...
Connection accepted from client 4
Handler assigned
Waiting for new connection...
```

 The right window is also titled 'sv@localhost:~/Desktop/main' and contains:

```
[sv@localhost main]$ ./client
Enter an arithmetic expression (or 'exit' to quit):
```

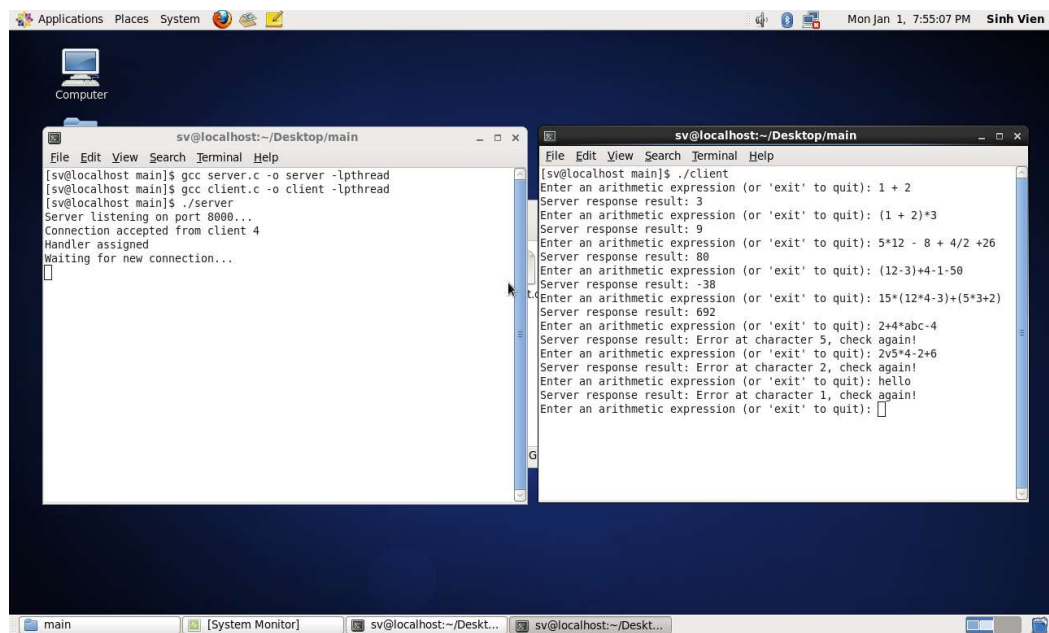
- Gửi nhận kết quả:



```
sv@localhost:~/Desktop/main
File Edit View Search Terminal Help
[sv@localhost main]$ gcc server.c -o server -lpthread
[sv@localhost main]$ gcc client.c -o client -lpthread
[sv@localhost main]$ ./server
Server listening on port 8080...
Connection accepted from client 4
Handler assigned
Waiting for new connection...
[]

sv@localhost:~/Desktop/main
File Edit View Search Terminal Help
[sv@localhost main]$ ./client
Enter an arithmetic expression (or 'exit' to quit): 1 + 2
Server response result: 3
Enter an arithmetic expression (or 'exit' to quit): (1 + 2)*3
Server response result: 9
Enter an arithmetic expression (or 'exit' to quit): 5*12 - 8 + 4/2 +26
Server response result: 80
Enter an arithmetic expression (or 'exit' to quit): (12-3)+4-1-50
Server response result: -38
Enter an arithmetic expression (or 'exit' to quit): 15*(12*4-3)+(5*3+2)
Server response result: 692
Enter an arithmetic expression (or 'exit' to quit): []
```

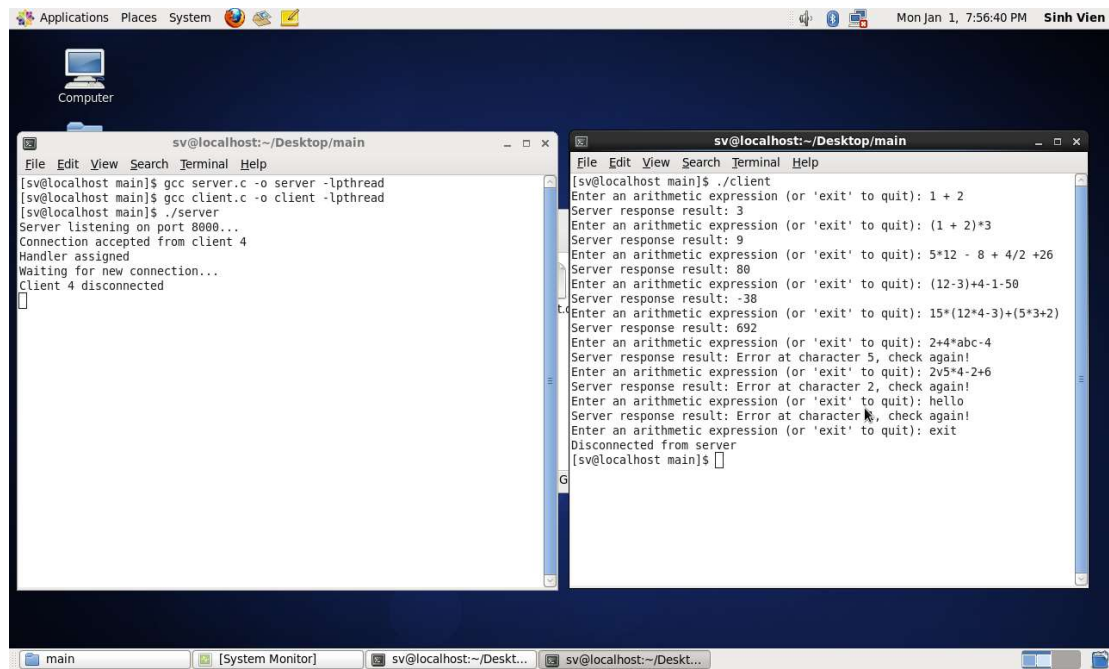
- Báo lỗi khi nhập sai biểu thức:



```
sv@localhost:~/Desktop/main
File Edit View Search Terminal Help
[sv@localhost main]$ gcc server.c -o server -lpthread
[sv@localhost main]$ gcc client.c -o client -lpthread
[sv@localhost main]$ ./server
Server listening on port 8080...
Connection accepted from client 4
Handler assigned
Waiting for new connection...
[]

sv@localhost:~/Desktop/main
File Edit View Search Terminal Help
[sv@localhost main]$ ./client
Enter an arithmetic expression (or 'exit' to quit): 1 + 2
Server response result: 3
Enter an arithmetic expression (or 'exit' to quit): (1 + 2)*3
Server response result: 9
Enter an arithmetic expression (or 'exit' to quit): 5*12 - 8 + 4/2 +26
Server response result: 80
Enter an arithmetic expression (or 'exit' to quit): (12-3)+4-1-50
Server response result: -38
Enter an arithmetic expression (or 'exit' to quit): 15*(12*4-3)+(5*3+2)
Server response result: 692
Enter an arithmetic expression (or 'exit' to quit): 2*4*abc-4
Server response result: Error at character 5, check again!
Enter an arithmetic expression (or 'exit' to quit): 2*5*4-2*6
Server response result: Error at character 2, check again!
Enter an arithmetic expression (or 'exit' to quit): hello
Server response result: Error at character 1, check again!
Enter an arithmetic expression (or 'exit' to quit): []
```

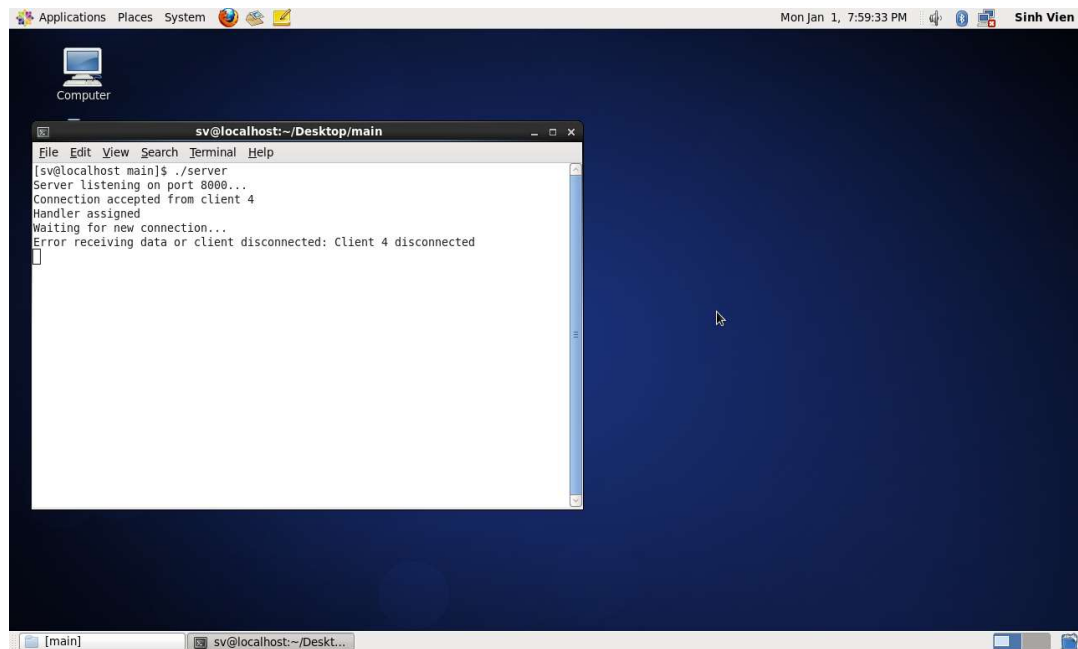
- Kết thúc chương trình bằng exit:



```
sv@localhost:~/Desktop/main
File Edit View Search Terminal Help
[sv@localhost main]$ gcc server.c -o server -lpthread
[sv@localhost main]$ gcc client.c -o client -lpthread
[sv@localhost main]$ ./server
Server listening on port 8000...
Connection accepted from client 4
Handler assigned
Waiting for new connection...
Client 4 disconnected

sv@localhost:~/Desktop/main
File Edit View Search Terminal Help
[sv@localhost main]$ ./client
Enter an arithmetic expression (or 'exit' to quit): 1 + 2
Server response result: 3
Enter an arithmetic expression (or 'exit' to quit): (1 + 2)*3
Server response result: 9
Enter an arithmetic expression (or 'exit' to quit): 5*12 - 8 + 4/2 +26
Server response result: 80
Enter an arithmetic expression (or 'exit' to quit): (12-3)+4-1-50
Server response result: -38
Enter an arithmetic expression (or 'exit' to quit): 15*(12*4-3)+(5*3+2)
Server response result: 692
Enter an arithmetic expression (or 'exit' to quit): 2+4*abc-4
Server response result: Error at character 5, check again!
Enter an arithmetic expression (or 'exit' to quit): 2v5*4-2+6
Server response result: Error at character 2, check again!
Enter an arithmetic expression (or 'exit' to quit): hello
Server response result: Error at character 1, check again!
Enter an arithmetic expression (or 'exit' to quit): exit
Disconnected from server
[sv@localhost main]$
```

- Kết thúc chương trình bằng việc tắt terminal:

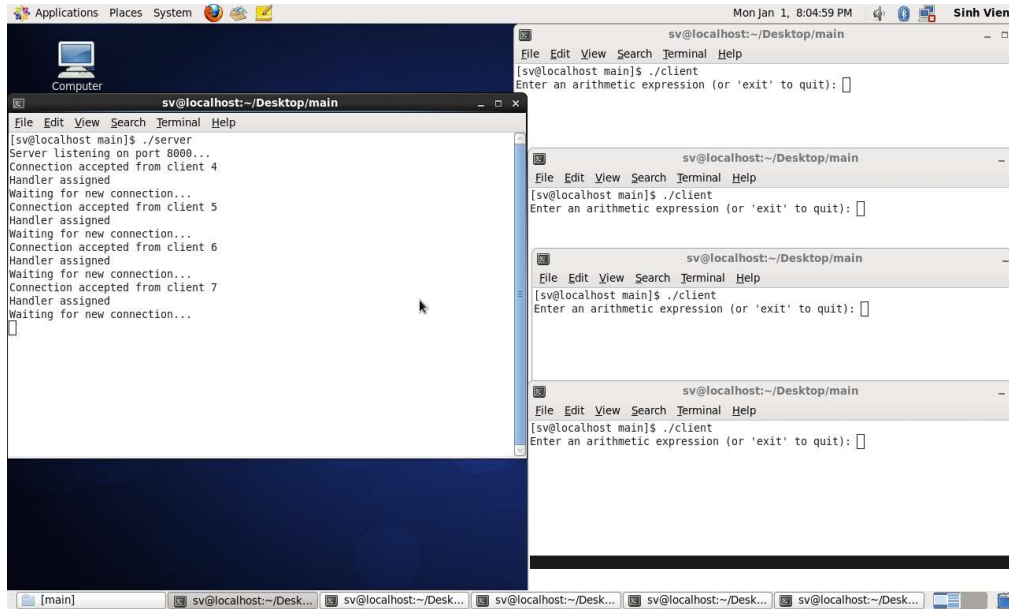


```
sv@localhost:~/Desktop/main
File Edit View Search Terminal Help
[sv@localhost main]$ ./server
Server listening on port 8000...
Connection accepted from client 4
Handler assigned
Waiting for new connection...
Error receiving data or client disconnected: Client 4 disconnected
```

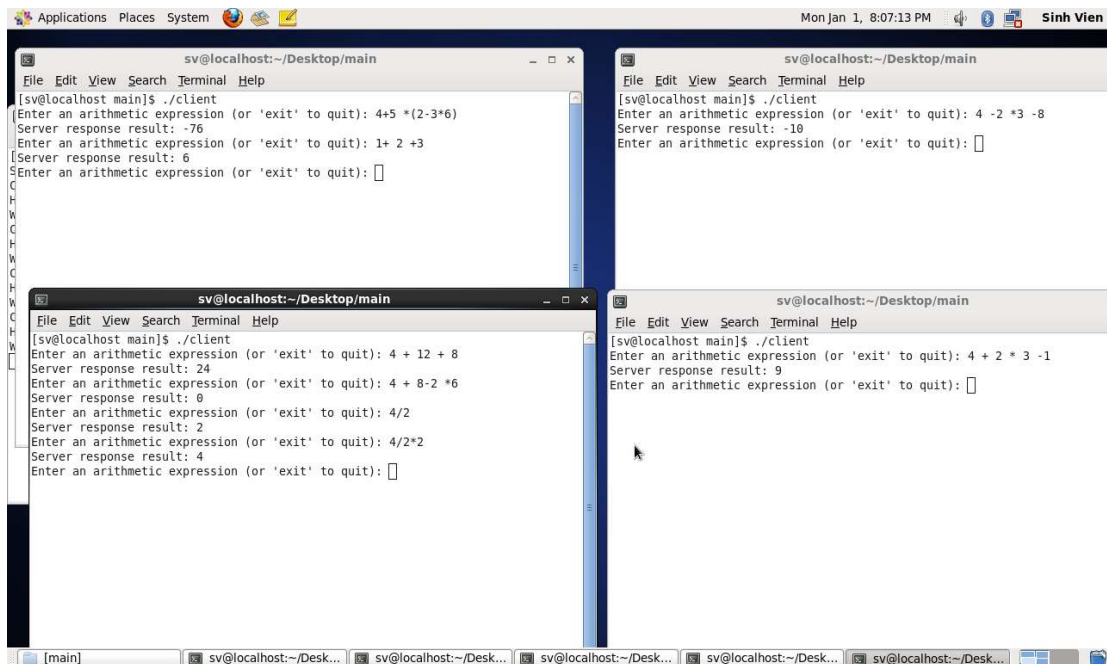

2.2. Test case 2 (1 Server n Client)

Nội dung test case 1: tạo 1 Server và nhiều Client để chúng trao đổi với nhau, gửi biểu thức tính toán và trả về kết quả cho Client. Ở đây em sẽ tạo 4 Client để kết nối với Server cùng 1 lúc.

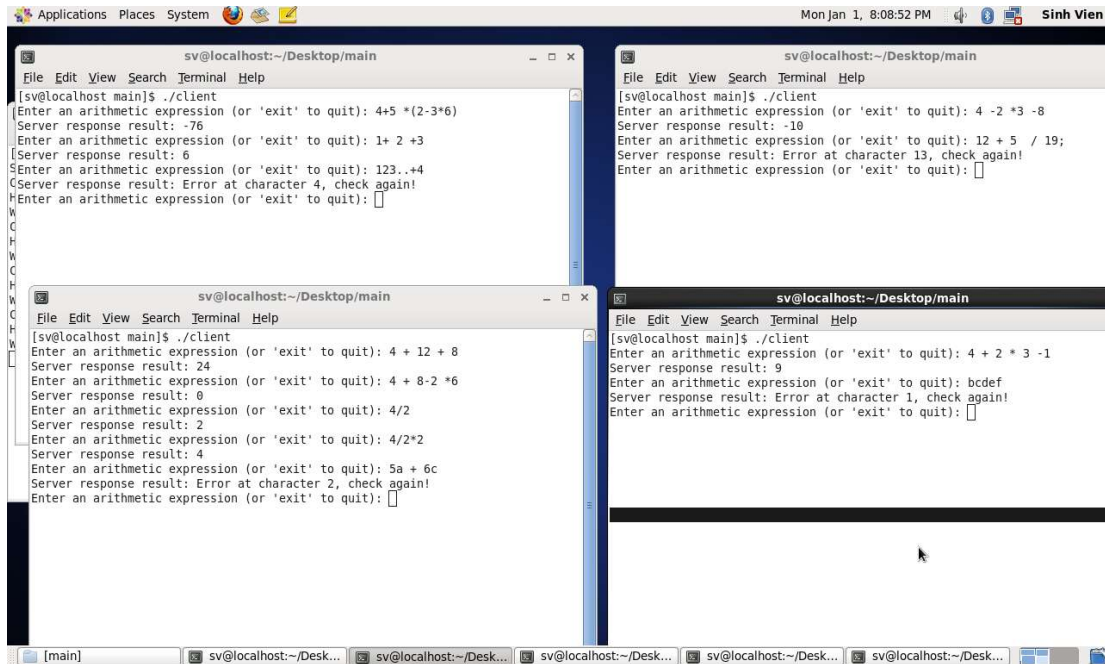
- Kết nối 1 Server và 4 Client cùng lúc thành công:



- Các Client được tính toán riêng biệt không cần tuần tự:



- Các Client nhận báo lỗi:



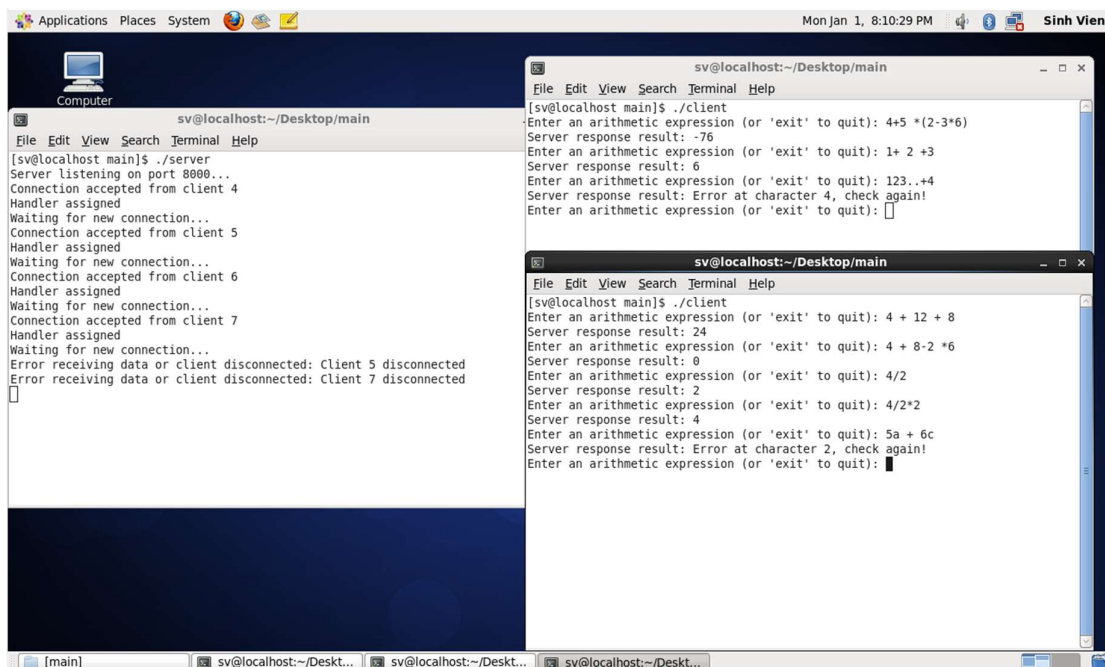
```
sv@localhost:~/Desktop/main
[sv@localhost main]$ ./client
Enter an arithmetic expression (or 'exit' to quit): 4+5 *(2-3*6)
Server response result: -76
Enter an arithmetic expression (or 'exit' to quit): 1+ 2 +3
Server response result: 6
Enter an arithmetic expression (or 'exit' to quit): 123..+4
Server response result: Error at character 4, check again!
Enter an arithmetic expression (or 'exit' to quit):

sv@localhost:~/Desktop/main
[sv@localhost main]$ ./client
Enter an arithmetic expression (or 'exit' to quit): 4 -2 *3 -8
Server response result: -10
Enter an arithmetic expression (or 'exit' to quit): 12 + 5 / 19;
Server response result: Error at character 13, check again!
Enter an arithmetic expression (or 'exit' to quit):

sv@localhost:~/Desktop/main
[sv@localhost main]$ ./client
Enter an arithmetic expression (or 'exit' to quit): 4 + 12 + 8
Server response result: 24
Enter an arithmetic expression (or 'exit' to quit): 4 + 8-2 *6
Server response result: 0
Enter an arithmetic expression (or 'exit' to quit): 4/2
Server response result: 2
Enter an arithmetic expression (or 'exit' to quit): 4/2*2
Server response result: 4
Enter an arithmetic expression (or 'exit' to quit): 5a + 6c
Server response result: Error at character 2, check again!
Enter an arithmetic expression (or 'exit' to quit):

sv@localhost:~/Desktop/main
[sv@localhost main]$ ./client
Enter an arithmetic expression (or 'exit' to quit): 4 + 2 * 3 -1
Server response result: 9
Enter an arithmetic expression (or 'exit' to quit): bcdef
Server response result: Error at character 1, check again!
Enter an arithmetic expression (or 'exit' to quit):
```

- Thoát 2 Client bằng các tắt Terminal:

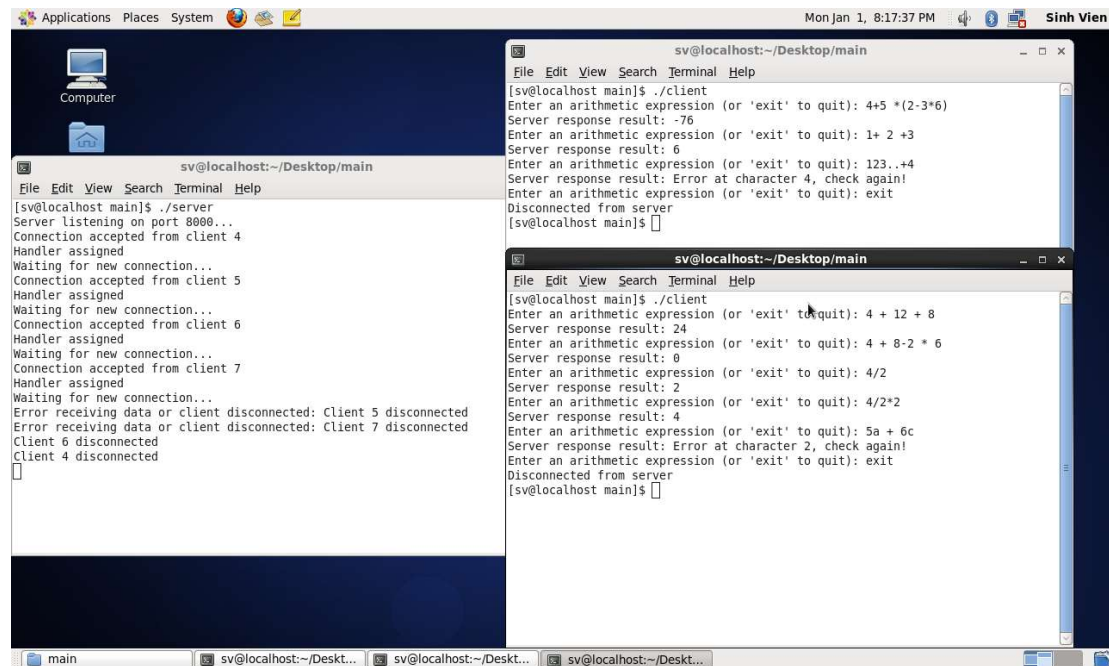


```
sv@localhost:~/Desktop/main
[sv@localhost main]$ ./server
Server listening on port 8000...
Connection accepted from client 4
Handler assigned
Waiting for new connection...
Connection accepted from client 5
Handler assigned
Waiting for new connection...
Connection accepted from client 6
Handler assigned
Waiting for new connection...
Connection accepted from client 7
Handler assigned
Waiting for new connection...
Error receiving data or client disconnected: Client 5 disconnected
Error receiving data or client disconnected: Client 7 disconnected

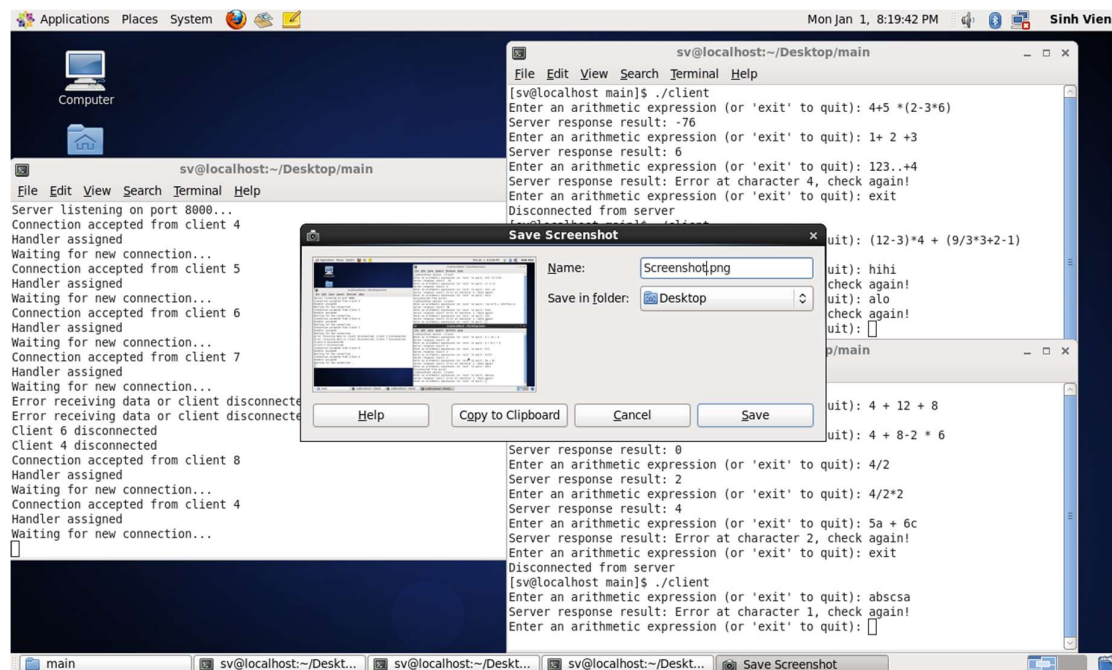
sv@localhost:~/Desktop/main
[sv@localhost main]$ ./client
Enter an arithmetic expression (or 'exit' to quit): 4+5 *(2-3*6)
Server response result: -76
Enter an arithmetic expression (or 'exit' to quit): 1+ 2 +3
Server response result: 6
Enter an arithmetic expression (or 'exit' to quit): 123..+4
Server response result: Error at character 4, check again!
Enter an arithmetic expression (or 'exit' to quit):

sv@localhost:~/Desktop/main
[sv@localhost main]$ ./client
Enter an arithmetic expression (or 'exit' to quit): 4 + 12 + 8
Server response result: 24
Enter an arithmetic expression (or 'exit' to quit): 4 + 8-2 *6
Server response result: 0
Enter an arithmetic expression (or 'exit' to quit): 4/2
Server response result: 2
Enter an arithmetic expression (or 'exit' to quit): 4/2*2
Server response result: 4
Enter an arithmetic expression (or 'exit' to quit): 5a + 6c
Server response result: Error at character 2, check again!
Enter an arithmetic expression (or 'exit' to quit):
```

- Thoát 2 Client bằng cách exit:



- Kết nối Client mới bằng 2 terminal đã thoát (kết quả này có hiện save screenshot là tại vì em chụp hình bằng máy ảo xong cái nó bị đứng nên em sử dụng chụp màn hình của máy để chụp lại, do bị đứng nên em không thể tắt màn hình đó được):



3. Đánh giá mức độ hoàn thành

Nội dung	Mức độ hoàn thành
Tạo được chương trình Socket 1 server – n client	100%
Client gửi yêu cầu để Server tính toán số nguyên (các phép cơ bản +, -, *, /) và trả kết quả về cho client.	100%
Client nhập sai cú pháp thì server gửi thông báo không hợp lệ về cho client.	100%
Kết nối cùng lúc được nhiều client trên cùng 1 server	100%
Các Client kết nối cùng lúc có thể được xử lý riêng lẻ mà không cần phải đợi xử lý tuần tự từng Client	100%
Tổng mức độ hoàn thành:	100%

---END---