CAB403 Assignment

Statement of Completness:

Tasks Attempted: Task 1

- Server command line parameter – configurable port & default port
- Server exits gracefully upon receiving SIGNAL (ctrl + c)
- Client command line parameters
- Implementation of SUB
- Implementation of UNSUB
- Implementation of SEND
- Implementation of BYE
- Implementation of NEXT <channel_id> and NEXT
- Description of data structures in the report

Deviations from the assignment taken: None

Problems and Deficencies in Solution: Did not complete the LIVEFEED and CHANNEL command. The client exits gracefully with the server. When the server closes down, it does not close down the client. When sending messages to the server, the messages are truncated.

Information about team:

- Harry Newton (n10133810)

Implemented Datastructures:

The enum data type was used to recognise different types of commands being sent to and from the server. The enums each apply to the different commands that can be entered on the command line from the client program.

```
typedef enum
{
    SUB,
    CHANNELS,
    UNSUB,
    NEXT,
    NEXTID,
    LIVEFEEDID,
    LIVEFEED,
    SEND,
    BYE,
} request_t;
```

This structure combines 3 different variables. The first two are optional, providing the channel_id, which some commands use, and the message, which one command uses. The third variable is of type request_t, which allows the server to know what command is being sent and how to process it.

```
typedef struct
{
    int channel_id;         // Optional
    char *message;          // Optional
    request_t request_type; // Required
} command_request_t;
```

This structure is used for storing sent messages to the server. The message can contain up to 100 characters and is marked read or not. This allows the server to know what messages to send to the clients.

```
typedef struct
{
    bool read;
    char message[100];
} message;
```

This struct is used to represent a channel within the server. Each channel can contain up to 100 messages and can be subscribed to, using the subscribed boolean. The three variables messages_sent/read/unread are used for the CHANNELS command, when the server sends over information about each channel the client is subscribed to.

```
typedef struct
{
    bool subscribed;
    int messages_sent;
    int messages_read;
    int messages_unread;
    message messages[100];
} channel;
```

This structure holds the number of current subscriptions the client has, as well as a list containing all of the channels.

```
typedef struct
{
    int subscriptions;
    channel channel_list[MAX_CHANNELS];
} channel_list_t;
```

How To Compile The Program:

In order to compile the program, run the command as shown in the example below

```
➜ make
gcc -Wall    -c -o server.o server.c
gcc -Wall    -c -o utils.o utils.c
gcc -Wall  -o server server.o utils.o
gcc -Wall    -c -o client.o client.c
gcc -Wall  -o client client.o utils.o
rm -f *.o
```

This should successfully compile all of the source files needed. Next run the command below to initalize the server. After you should receive the following output from the terminal.

```
➜ ./server 8081
Successfully created socket
Successfully binded socket to server port
Successfully listening on socket
Successfully connected to client 127.0.0.1:56948
```

This should successfully initalize the server. Once running, the client needs to be run. Run the command as shown below to connect the client to the server.

```
University/CAB441/Assignment took 41s
➜ ./client 127.0.0.1 8081
Successfully created client socket
Successfully connected to the server, please enter commands below:
```