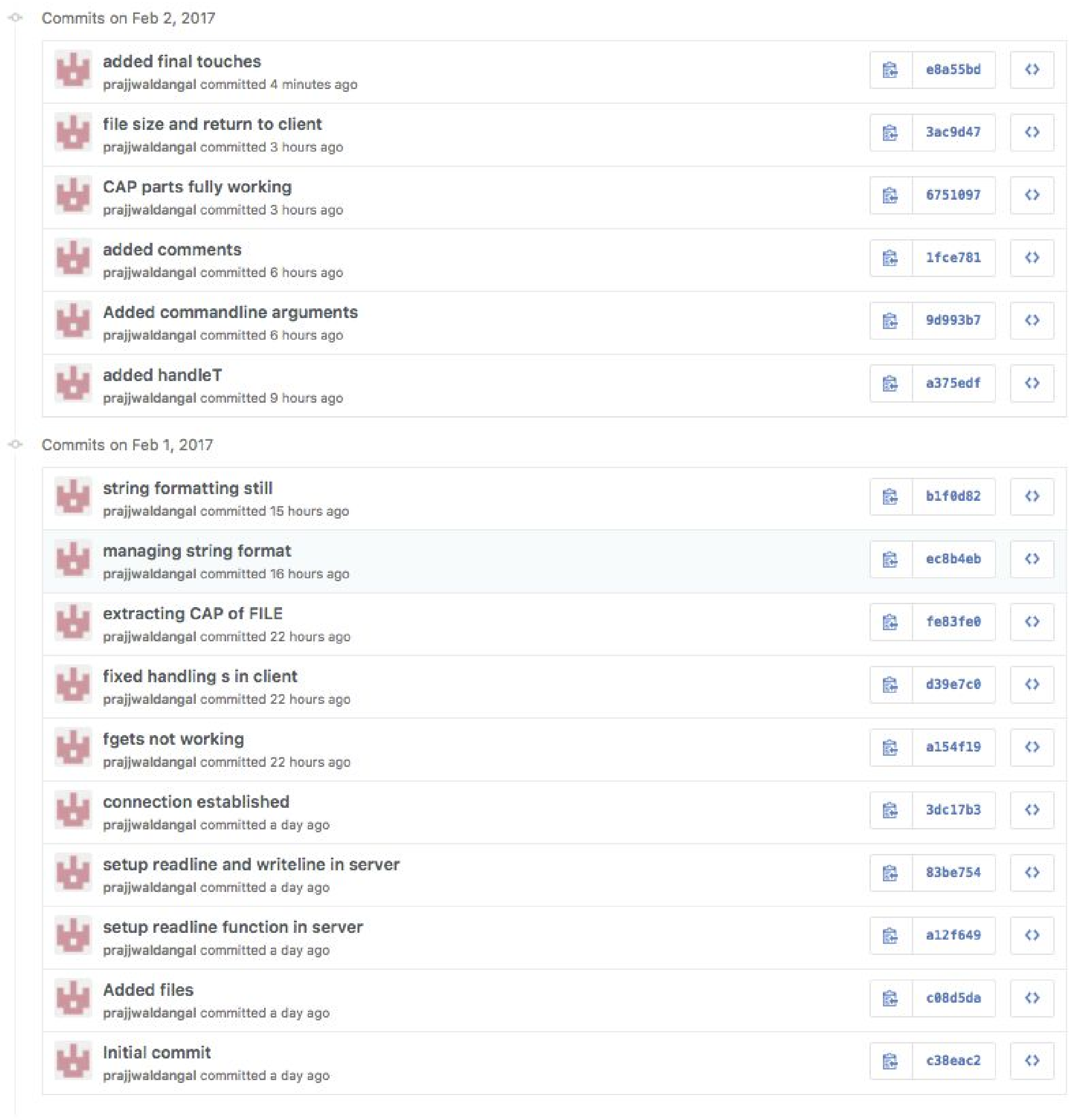
@02721735 Prajjwal Dangal

Github link: https://github.com/prajjwaldangal/SocketProgramming.git

All commits:



Commits on Feb 2, 2017

added final touches
prajjwaldangal committed 4 minutes ago e8a55bd

file size and return to client
prajjwaldangal committed 3 hours ago 3ac9d47

CAP parts fully working
prajjwaldangal committed 3 hours ago 6751097

added comments
prajjwaldangal committed 6 hours ago 1fce781

Added commandline arguments
prajjwaldangal committed 6 hours ago 9d993b7

added handleT
prajjwaldangal committed 9 hours ago a375edf

Commits on Feb 1, 2017

string formatting still
prajjwaldangal committed 15 hours ago b1f0d82

managing string format
prajjwaldangal committed 16 hours ago ec8b4eb

extracting CAP of FILE
prajjwaldangal committed 22 hours ago fe83fe0

fixed handling s in client
prajjwaldangal committed 22 hours ago d39e7c0

fgets not working
prajjwaldangal committed 22 hours ago a154f19

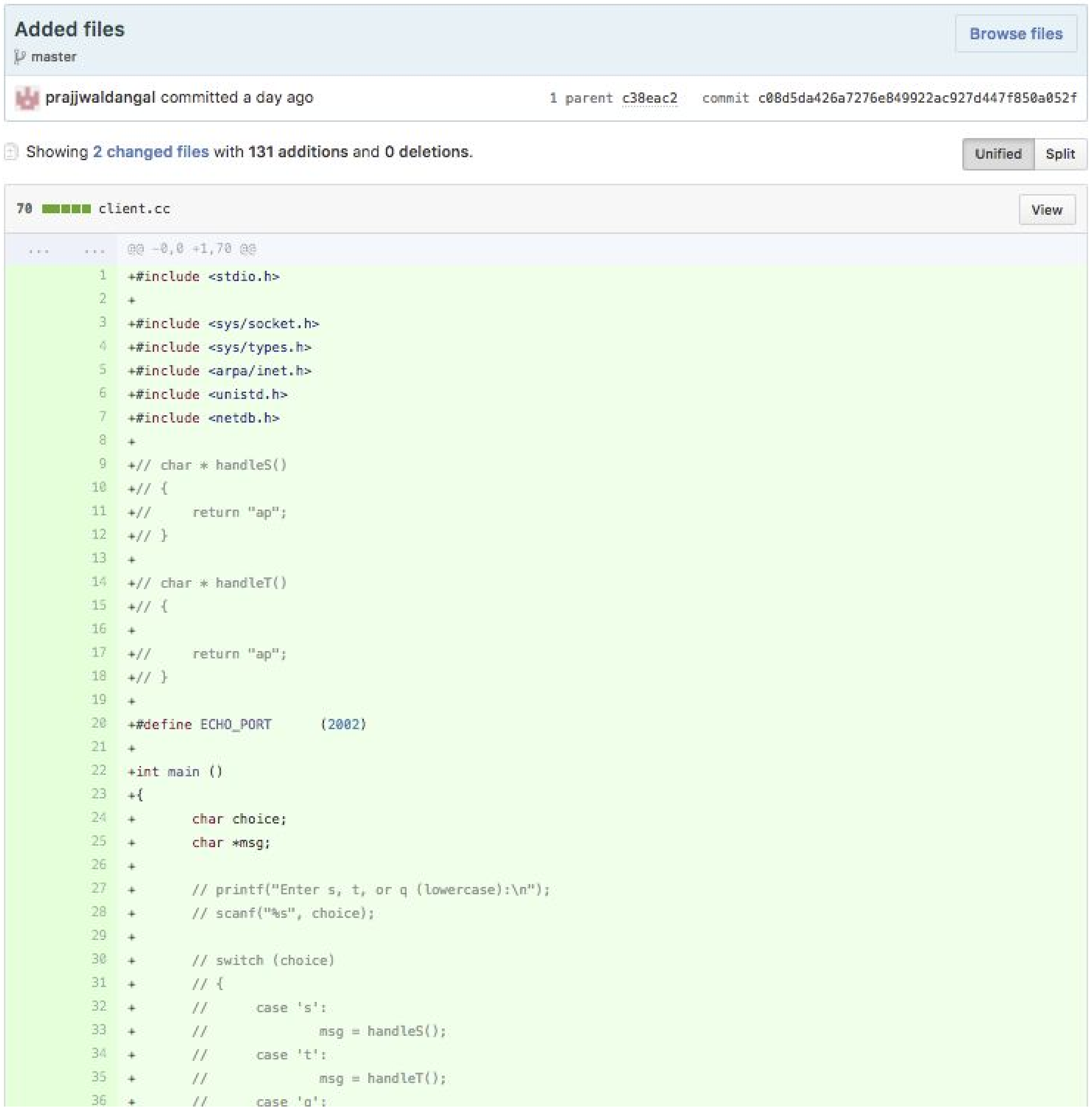
connection established
prajjwaldangal committed a day ago 3dc17b3

setup readline and writeline in server
prajjwaldangal committed a day ago 83be754

setup readline function in server
prajjwaldangal committed a day ago a12f649

Added files
prajjwaldangal committed a day ago c08d5da

Initial commit
prajjwaldangal committed a day ago c38eac2



Added files

master

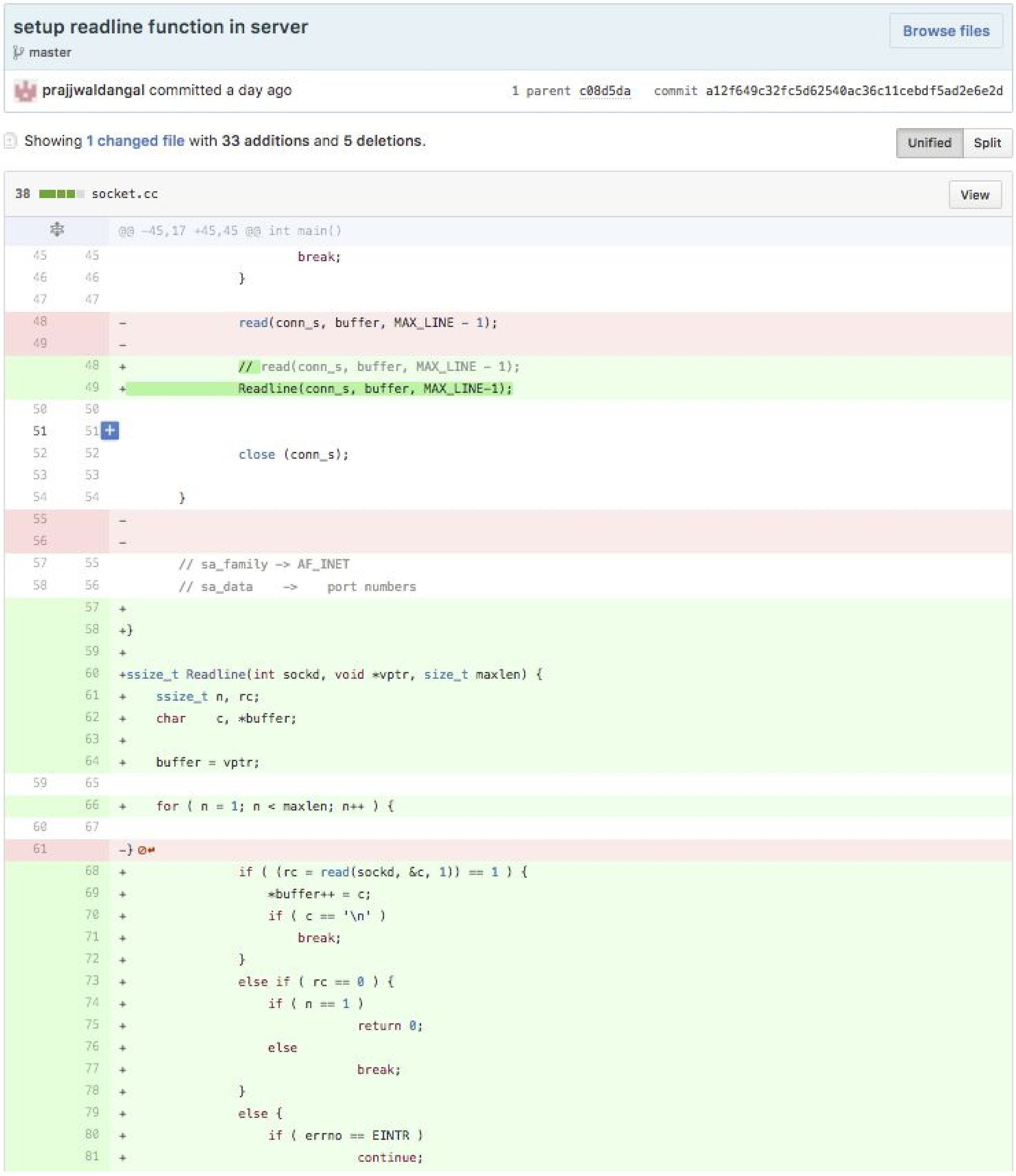
prajjwaldangal committed a day ago   1 parent c38eac2   commit c08d5da426a7276e849922ac927d447f850a052f

Browse files

Showing 2 changed files with 131 additions and 0 deletions.

70 client.cc

```
@@ -0,0 +1,70 @@
+#include <stdio.h>
+
+#include <sys/socket.h>
+#include <sys/types.h>
+#include <arpa/inet.h>
+#include <unistd.h>
+#include <netdb.h>
+
+// char * handleS()
+// {
+//      return "ap";
+// }
+
+// char * handleT()
+// {
+
+//      return "ap";
+// }
+
+#define ECHO_PORT      (2002)
+
+int main ()
+{
+       char choice;
+       char *msg;
+
+       // printf("Enter s, t, or q (lowercase):\n");
+       // scanf("%s", choice);
+
+       // switch (choice)
+       // {
+       //      case 's':
+       //              msg = handleS();
+       //      case 't':
+       //              msg = handleT();
+       //      case 'q':
```



setup readline function in server

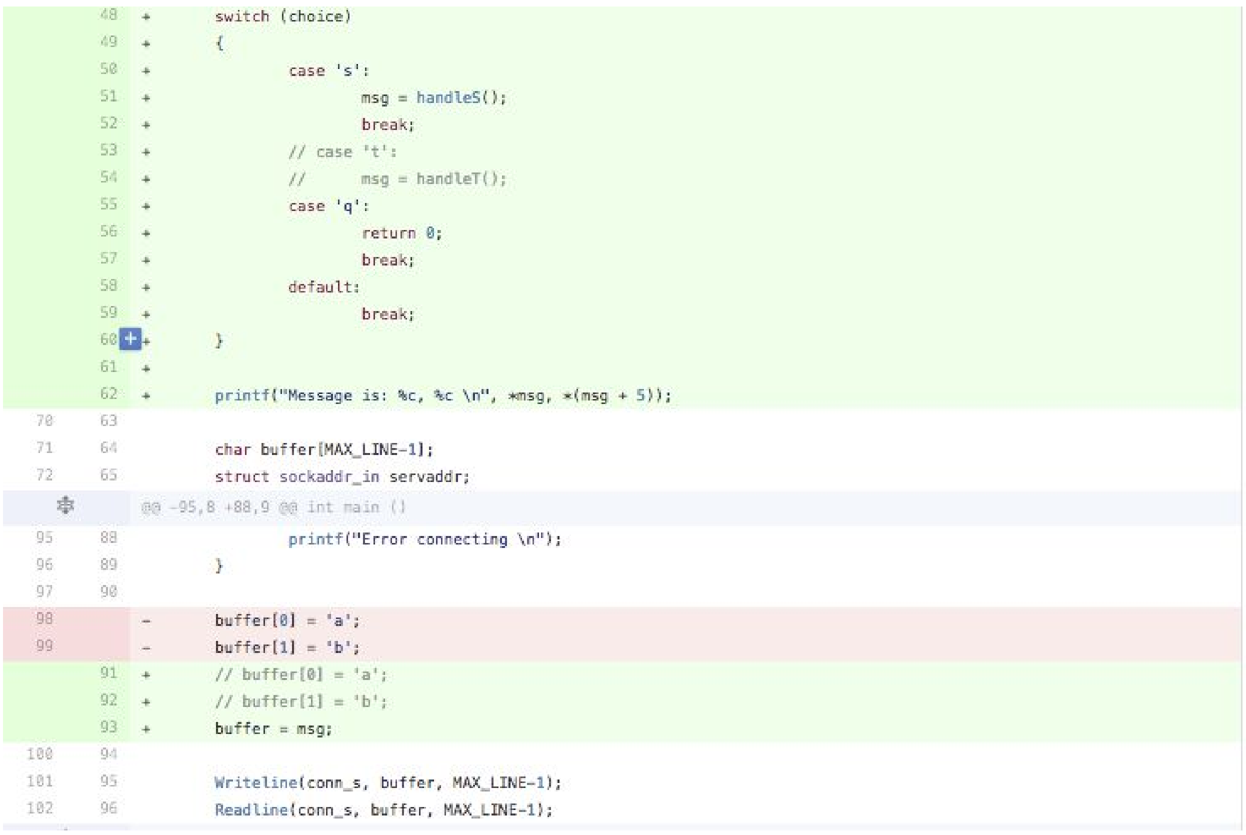
master

prajjwaldangal committed a day ago   1 parent c08d5da   commit a12f649c32fc5d62540ac36c11cebdf5ad2e6e2d

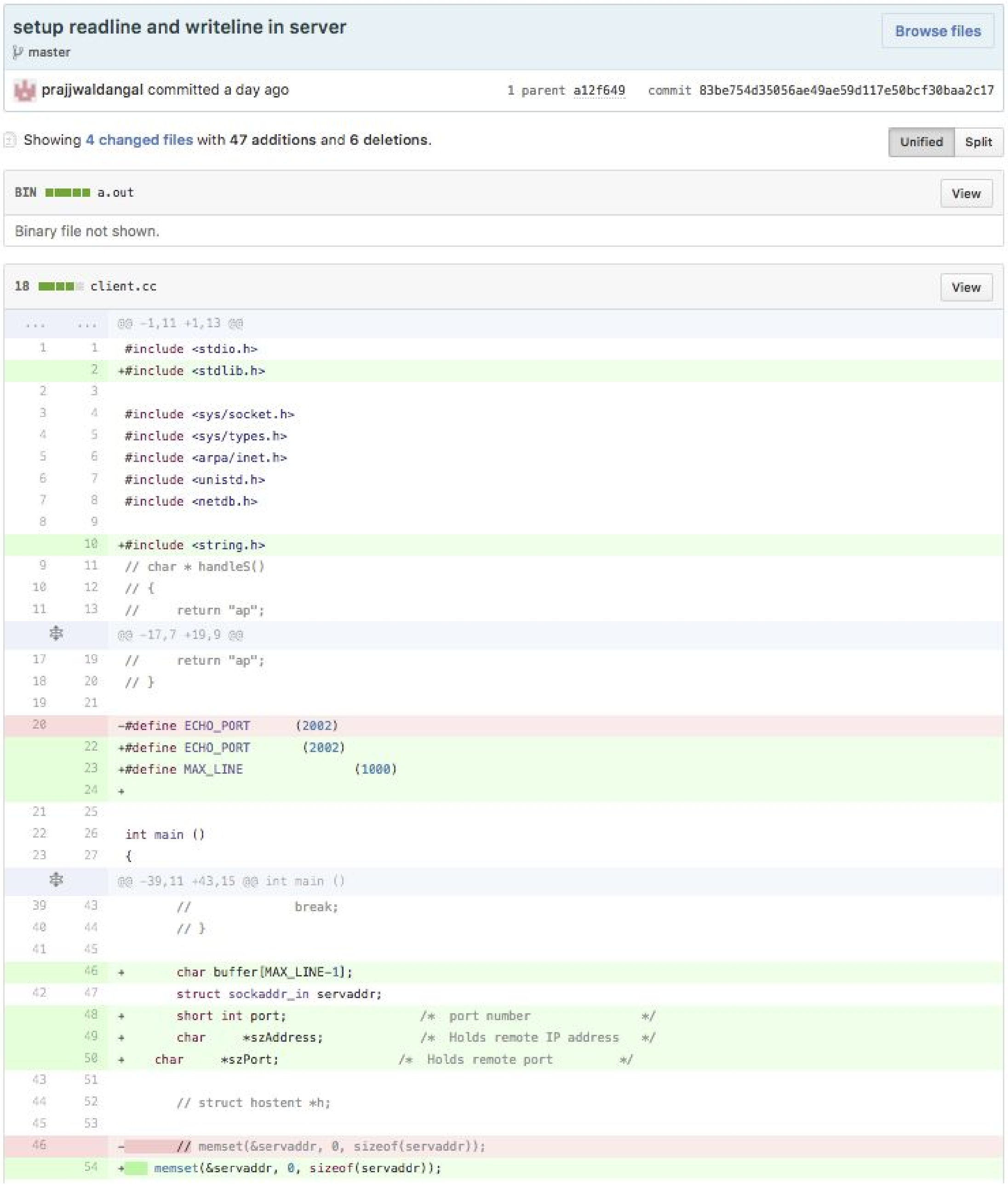
Showing 1 changed file with 33 additions and 5 deletions.

38 socket.cc

```
@@ -45,17 +45,45 @@ int main()
                        break;
                }

-               read(conn_s, buffer, MAX_LINE - 1);
-
+               // read(conn_s, buffer, MAX_LINE - 1);
+               Readline(conn_s, buffer, MAX_LINE-1);


                close (conn_s);

        }
-
-
        // sa_family -> AF_INET
        // sa_data    ->    port numbers
+
+}
+
+ssize_t Readline(int sockd, void *vptr, size_t maxlen) {
+    ssize_t n, rc;
+    char    c, *buffer;
+
+    buffer = vptr;

+    for ( n = 1; n < maxlen; n++ ) {

-}
+               if ( (rc = read(sockd, &c, 1)) == 1 ) {
+                   *buffer++ = c;
+                   if ( c == '\n' )
+                       break;
+               }
+               else if ( rc == 0 ) {
+                   if ( n == 1 )
+                               return 0;
+                   else
+                               break;
+               }
+               else {
+                   if ( errno == EINTR )
+                               continue;
```



```diff
+        switch (choice)
+        {
+                case 's':
+                        msg = handleS();
+                        break;
+                // case 't':
+                //      msg = handleT();
+                case 'q':
+                        return 0;
+                        break;
+                default:
+                        break;
+        }
+
+        printf("Message is: %c, %c \n", *msg, *(msg + 5));

         char buffer[MAX_LINE-1];
         struct sockaddr_in servaddr;
@@ -95,8 +88,9 @@ int main ()
                 printf("Error connecting \n");
         }

-        buffer[0] = 'a';
-        buffer[1] = 'b';
+        // buffer[0] = 'a';
+        // buffer[1] = 'b';
+        buffer = msg;

         Writeline(conn_s, buffer, MAX_LINE-1);
         Readline(conn_s, buffer, MAX_LINE-1);
```



setup readline and writeline in server

master

prajjwaldangal committed a day ago — 1 parent a12f649 — commit 83be754d35056ae49ae59d117e50bcf30baa2c17

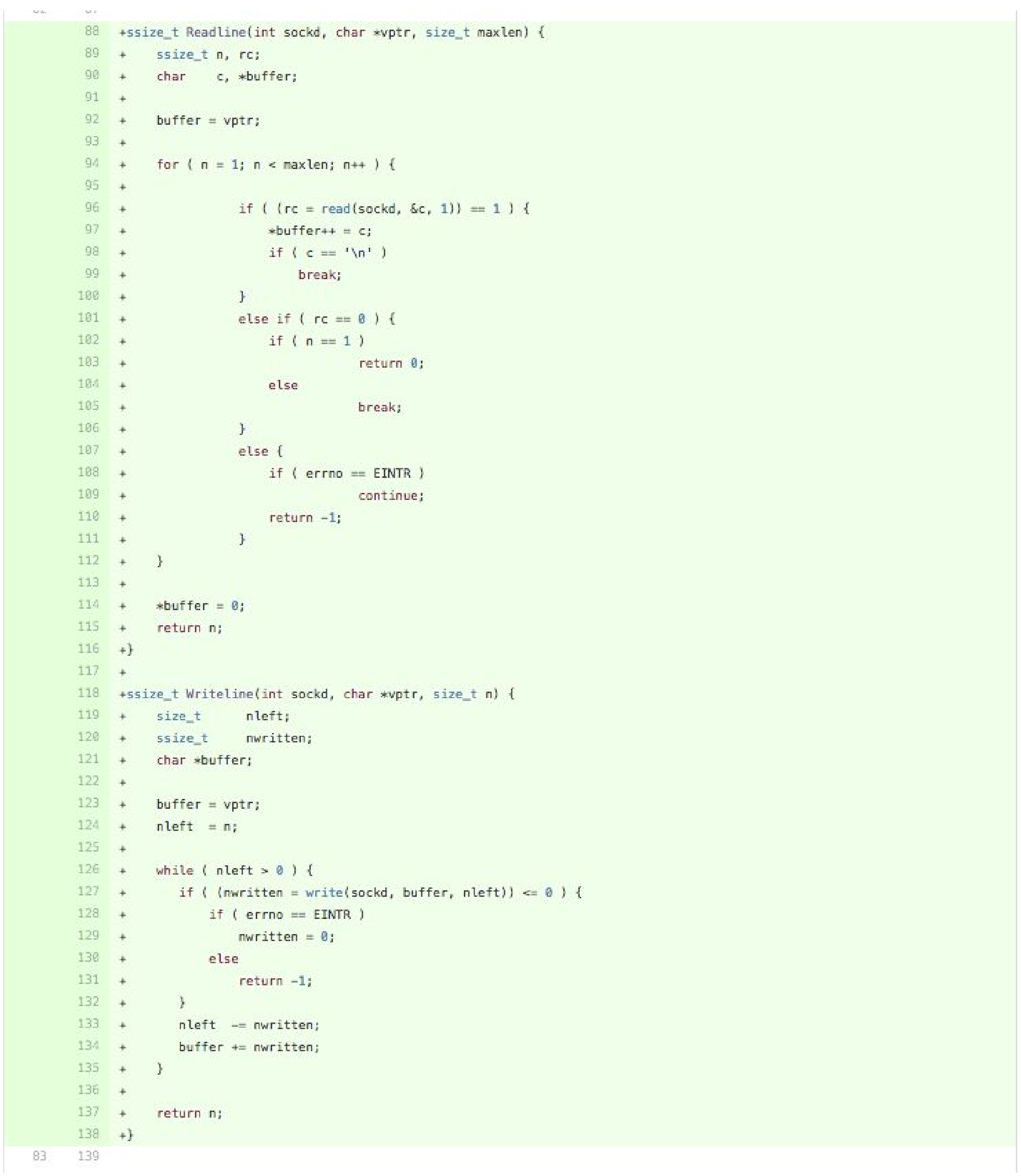
Showing 4 changed files with 47 additions and 6 deletions.

BIN a.out

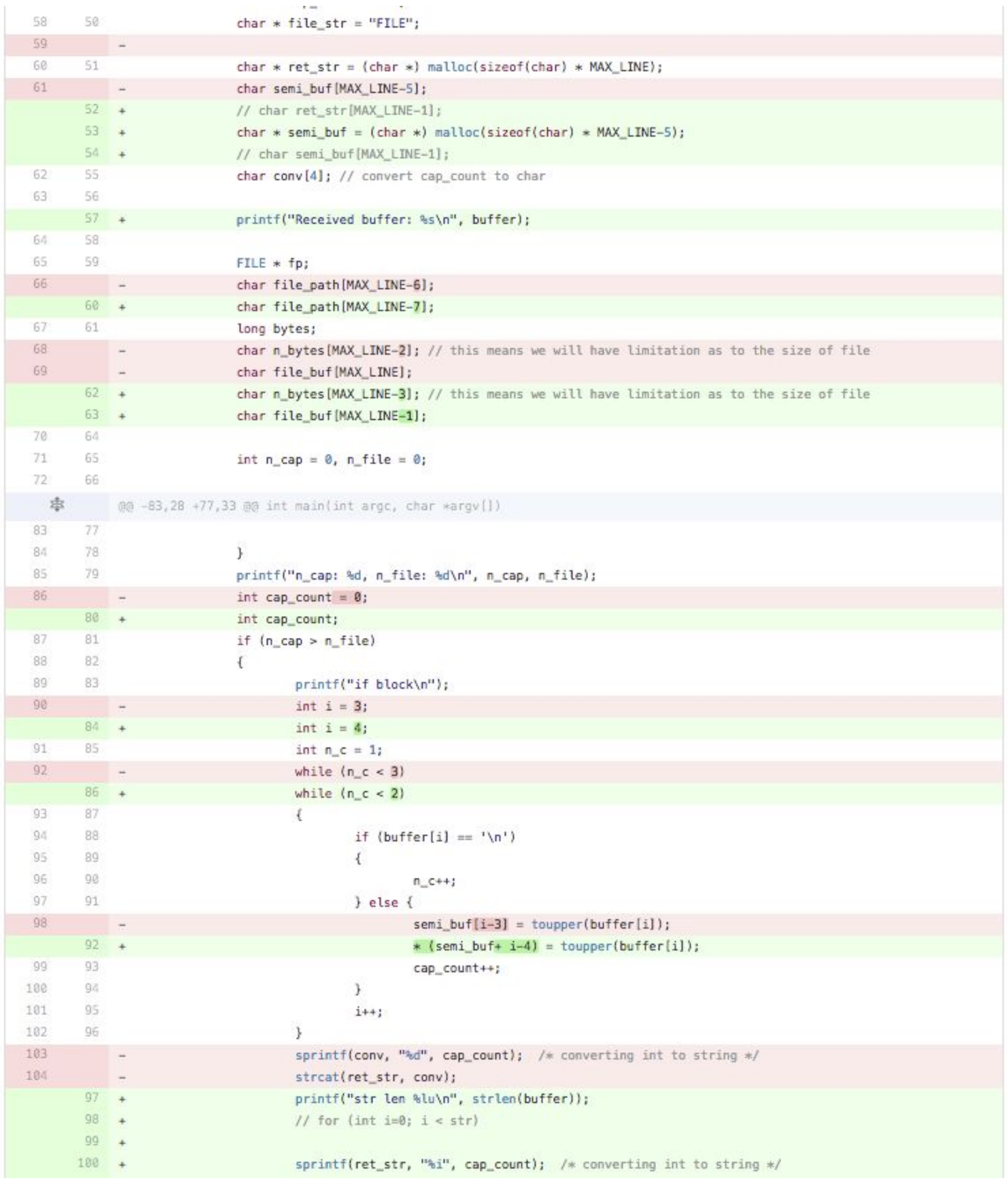
Binary file not shown.

18 client.cc

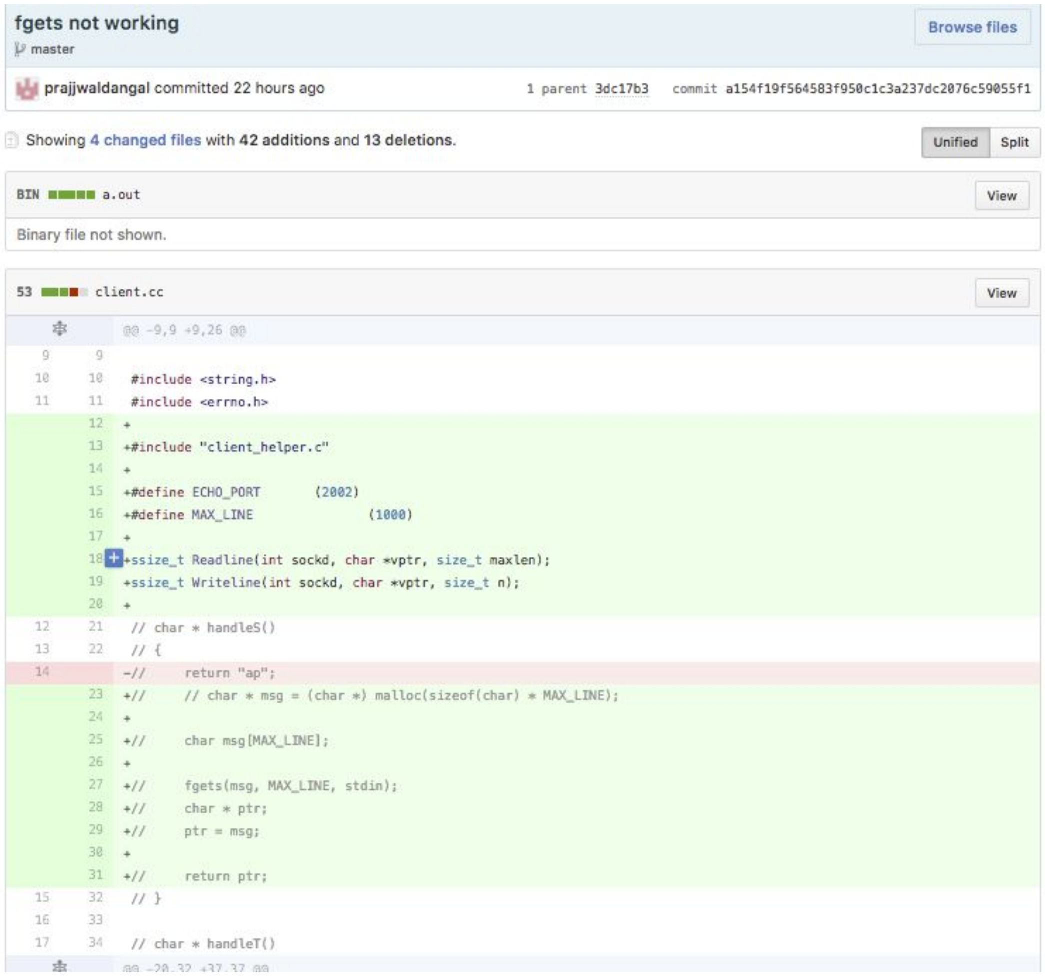
```
@@ -1,11 +1,13 @@
 #include <stdio.h>
+#include <stdlib.h>

 #include <sys/socket.h>
 #include <sys/types.h>
 #include <arpa/inet.h>
 #include <unistd.h>
 #include <netdb.h>

+#include <string.h>
 // char * handleS()
 // {
 //     return "ap";
@@ -17,7 +19,9 @@
 //     return "ap";
 // }

-#define ECHO_PORT      (2002)
+#define ECHO_PORT       (2002)
+#define MAX_LINE               (1000)
+

 int main ()
 {
@@ -39,11 +43,15 @@ int main ()
        //              break;
        // }

+       char buffer[MAX_LINE-1];
        struct sockaddr_in servaddr;
+       short int port;                  /*  port number              */
+       char     *szAddress;             /*  Holds remote IP address   */
+    char     *szPort;                 /*  Holds remote port         */

        // struct hostent *h;

-       // memset(&servaddr, 0, sizeof(servaddr));
+    memset(&servaddr, 0, sizeof(servaddr));
```



```
+ssize_t Readline(int sockd, char *vptr, size_t maxlen) {
+    ssize_t n, rc;
+    char    c, *buffer;
+
+    buffer = vptr;
+
+    for ( n = 1; n < maxlen; n++ ) {
+
+               if ( (rc = read(sockd, &c, 1)) == 1 ) {
+                   *buffer++ = c;
+                   if ( c == '\n' )
+                       break;
+               }
+               else if ( rc == 0 ) {
+                   if ( n == 1 )
+                               return 0;
+                   else
+                               break;
+               }
+               else {
+                   if ( errno == EINTR )
+                               continue;
+                   return -1;
+               }
+    }
+
+    *buffer = 0;
+    return n;
+}
+
+ssize_t Writeline(int sockd, char *vptr, size_t n) {
+    size_t      nleft;
+    ssize_t     nwritten;
+    char *buffer;
+
+    buffer = vptr;
+    nleft  = n;
+
+    while ( nleft > 0 ) {
+       if ( (nwritten = write(sockd, buffer, nleft)) <= 0 ) {
+           if ( errno == EINTR )
+               nwritten = 0;
+           else
+               return -1;
+       }
+       nleft  -= nwritten;
+       buffer += nwritten;
+    }
+
+    return n;
+}
```



```diff
 	char * file_str = "FILE";
-
 	char * ret_str = (char *) malloc(sizeof(char) * MAX_LINE);
-	char semi_buf[MAX_LINE-5];
+	// char ret_str[MAX_LINE-1];
+	char * semi_buf = (char *) malloc(sizeof(char) * MAX_LINE-5);
+	// char semi_buf[MAX_LINE-1];
 	char conv[4]; // convert cap_count to char
 
+	printf("Received buffer: %s\n", buffer);
 
 	FILE * fp;
-	char file_path[MAX_LINE-6];
+	char file_path[MAX_LINE-7];
 	long bytes;
-	char n_bytes[MAX_LINE-2]; // this means we will have limitation as to the size of file
-	char file_buf[MAX_LINE];
+	char n_bytes[MAX_LINE-3]; // this means we will have limitation as to the size of file
+	char file_buf[MAX_LINE-1];
 
 	int n_cap = 0, n_file = 0;
 
@@ -83,28 +77,33 @@ int main(int argc, char *argv[])
 
 	}
 	printf("n_cap: %d, n_file: %d\n", n_cap, n_file);
-	int cap_count = 0;
+	int cap_count;
 	if (n_cap > n_file)
 	{
 		printf("if block\n");
-		int i = 3;
+		int i = 4;
 		int n_c = 1;
-		while (n_c < 3)
+		while (n_c < 2)
 		{
 			if (buffer[i] == '\n')
 			{
 				n_c++;
 			} else {
-				semi_buf[i-3] = toupper(buffer[i]);
+				* (semi_buf+ i-4) = toupper(buffer[i]);
 				cap_count++;
 			}
 			i++;
 		}
-		sprintf(conv, "%d", cap_count);  /* converting int to string */
-		strcat(ret_str, conv);
+		printf("str len %lu\n", strlen(buffer));
+		// for (int i=0; i < str)
+
+		sprintf(ret_str, "%i", cap_count);  /* converting int to string */
```


fgets not working
master
prajjwaldangal committed 22 hours ago   1 parent 3dc17b3   commit a154f19f564583f950c1c3a237dc2076c59055f1

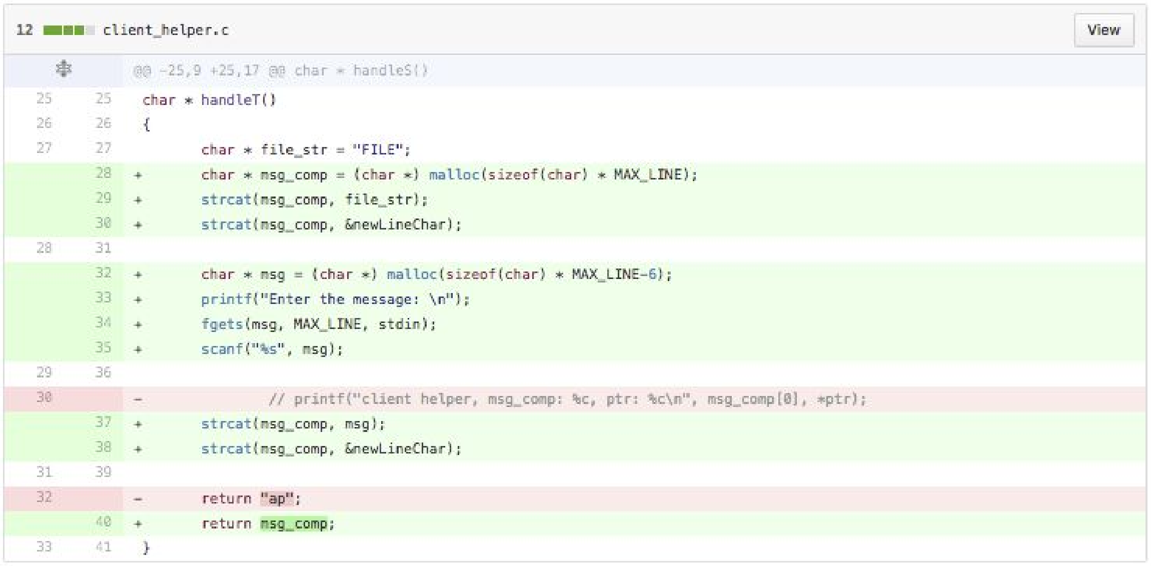
Showing 4 changed files with 42 additions and 13 deletions.

BIN a.out

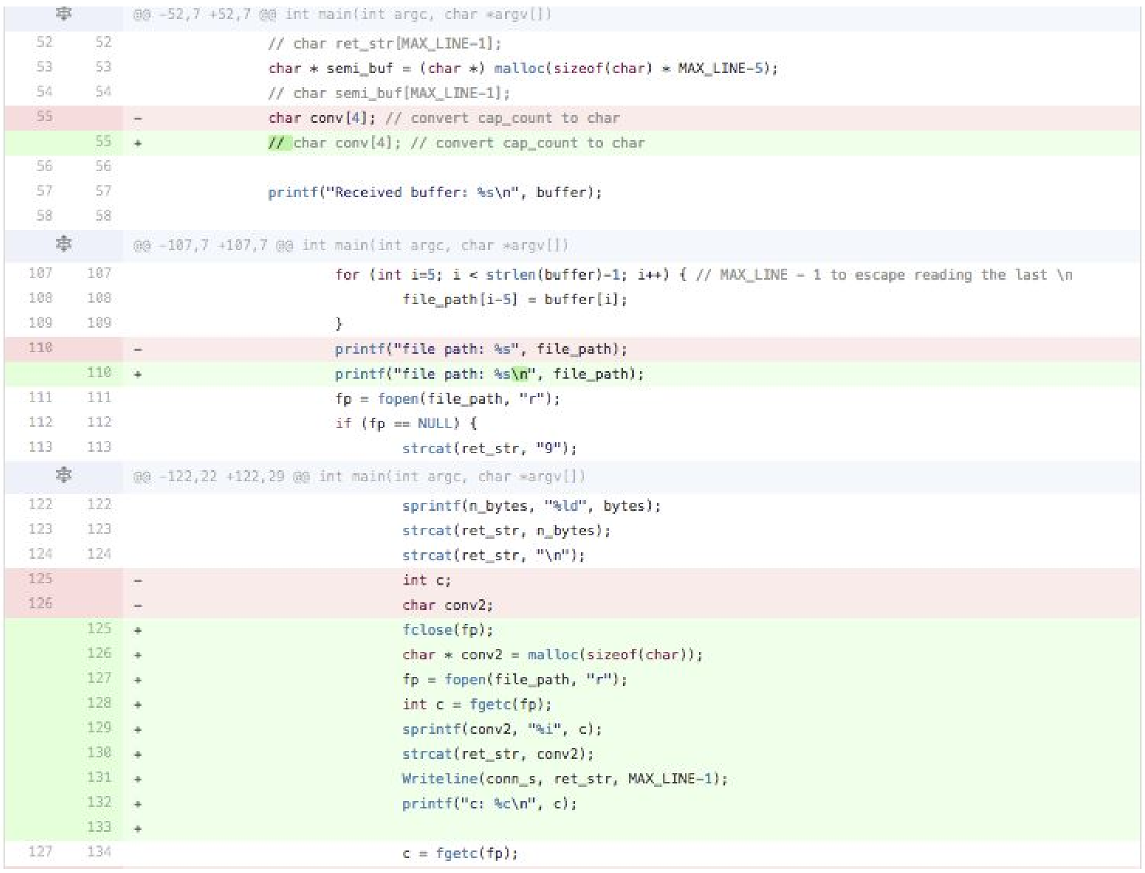
Binary file not shown.

53 client.cc

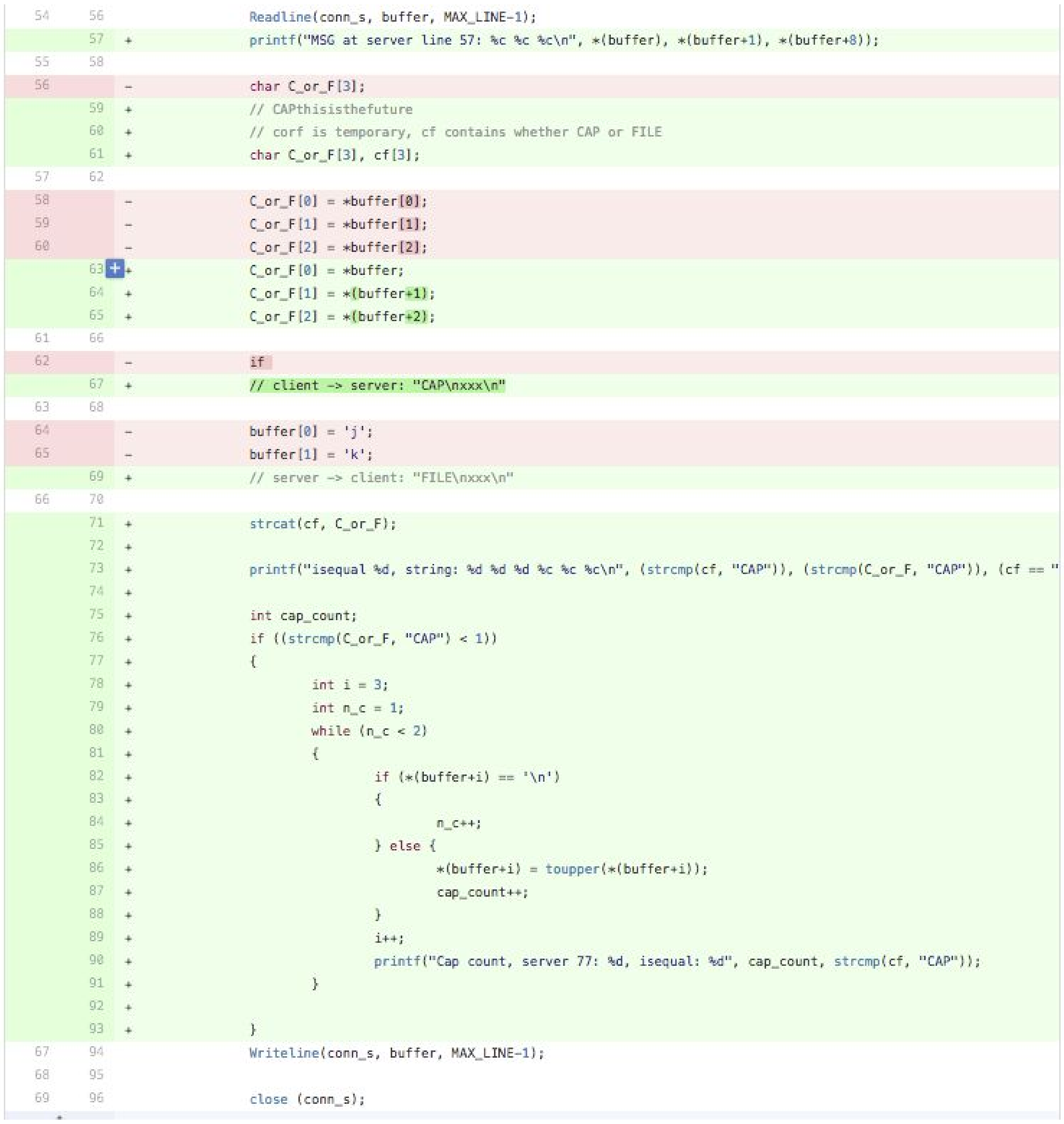
```
@@ -9,9 +9,26 @@

 #include <string.h>
 #include <errno.h>
+
+#include "client_helper.c"
+
+#define ECHO_PORT       (2002)
+#define MAX_LINE                (1000)
+
+ssize_t Readline(int sockd, char *vptr, size_t maxlen);
+ssize_t Writeline(int sockd, char *vptr, size_t n);
+
 // char * handleS()
 // {
-//      return "ap";
+//      // char * msg = (char *) malloc(sizeof(char) * MAX_LINE);
+
+//      char msg[MAX_LINE];
+
+//      fgets(msg, MAX_LINE, stdin);
+//      char * ptr;
+//      ptr = msg;
+
+//      return ptr;
 // }

 // char * handleT()
@@ -20,32 +37,37 @@
```


12 client_helper.c

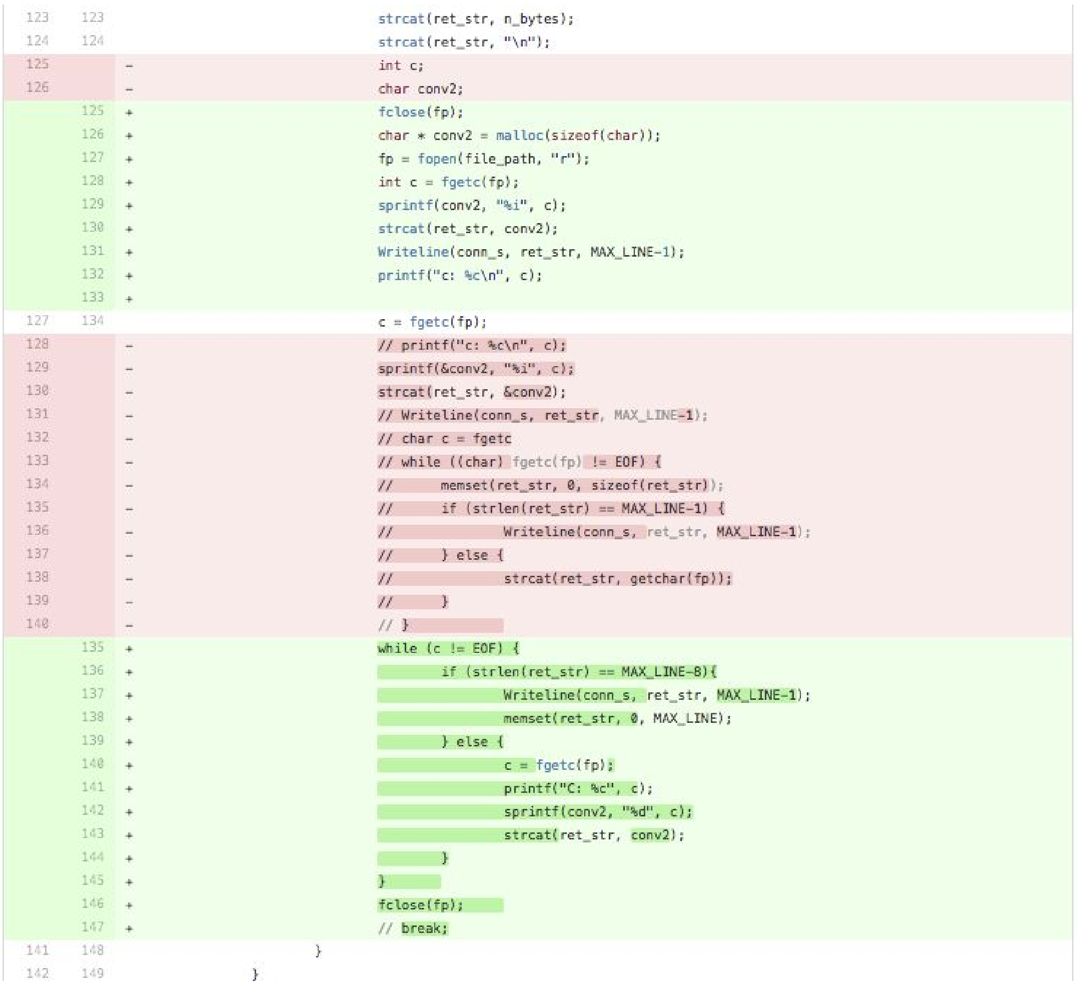
```
@@ -25,9 +25,17 @@ char * handleS()
 char * handleT()
 {
         char * file_str = "FILE";
+        char * msg_comp = (char *) malloc(sizeof(char) * MAX_LINE);
+        strcat(msg_comp, file_str);
+        strcat(msg_comp, &newLineChar);

+        char * msg = (char *) malloc(sizeof(char) * MAX_LINE-6);
+        printf("Enter the message: \n");
+        fgets(msg, MAX_LINE, stdin);
+        scanf("%s", msg);

-                // printf("client helper, msg_comp: %c, ptr: %c\n", msg_comp[0], *ptr);
+        strcat(msg_comp, msg);
+        strcat(msg_comp, &newLineChar);

-        return "ap";
+        return msg_comp;
 }
```



```diff
@@ -52,7 +52,7 @@ int main(int argc, char *argv[])
 			// char ret_str[MAX_LINE-1];
 			char * semi_buf = (char *) malloc(sizeof(char) * MAX_LINE-5);
 			// char semi_buf[MAX_LINE-1];
-			char conv[4]; // convert cap_count to char
+			// char conv[4]; // convert cap_count to char
 
 			printf("Received buffer: %s\n", buffer);
 
@@ -107,7 +107,7 @@ int main(int argc, char *argv[])
 				for (int i=5; i < strlen(buffer)-1; i++) { // MAX_LINE - 1 to escape reading the last \n
 					file_path[i-5] = buffer[i];
 				}
-				printf("file path: %s", file_path);
+				printf("file path: %s\n", file_path);
 				fp = fopen(file_path, "r");
 				if (fp == NULL) {
 					strcat(ret_str, "9");
@@ -122,22 +122,29 @@ int main(int argc, char *argv[])
 					sprintf(n_bytes, "%ld", bytes);
 					strcat(ret_str, n_bytes);
 					strcat(ret_str, "\n");
-					int c;
-					char conv2;
+					fclose(fp);
+					char * conv2 = malloc(sizeof(char));
+					fp = fopen(file_path, "r");
+					int c = fgetc(fp);
+					sprintf(conv2, "%i", c);
+					strcat(ret_str, conv2);
+					Writeline(conn_s, ret_str, MAX_LINE-1);
+					printf("c: %c\n", c);
+
 					c = fgetc(fp);
```



```
        Readline(conn_s, buffer, MAX_LINE-1);
+       printf("MSG at server line 57: %c %c %c\n", *(buffer), *(buffer+1), *(buffer+8));

-       char C_or_F[3];
+       // CAPthisisthefuture
+       // corf is temporary, cf contains whether CAP or FILE
+       char C_or_F[3], cf[3];

-       C_or_F[0] = *buffer[0];
-       C_or_F[1] = *buffer[1];
-       C_or_F[2] = *buffer[2];
+       C_or_F[0] = *buffer;
+       C_or_F[1] = *(buffer+1);
+       C_or_F[2] = *(buffer+2);

-       if
+       // client -> server: "CAP\nxxx\n"

-       buffer[0] = 'j';
-       buffer[1] = 'k';
+       // server -> client: "FILE\nxxx\n"

+       strcat(cf, C_or_F);
+
+       printf("isequal %d, string: %d %d %d %c %c %c\n", (strcmp(cf, "CAP")), (strcmp(C_or_F, "CAP")), (cf == "
+
+       int cap_count;
+       if ((strcmp(C_or_F, "CAP") < 1))
+       {
+               int i = 3;
+               int n_c = 1;
+               while (n_c < 2)
+               {
+                       if (*(buffer+i) == '\n')
+                       {
+                               n_c++;
+                       } else {
+                               *(buffer+i) = toupper(*(buffer+i));
+                               cap_count++;
+                       }
+                       i++;
+                       printf("Cap count, server 77: %d, isequal: %d", cap_count, strcmp(cf, "CAP"));
+               }
+
+       }
        Writeline(conn_s, buffer, MAX_LINE-1);

        close (conn_s);
```



```diff
                          strcat(ret_str, n_bytes);
                          strcat(ret_str, "\n");
-                         int c;
-                         char conv2;
+                         fclose(fp);
+                         char * conv2 = malloc(sizeof(char));
+                         fp = fopen(file_path, "r");
+                         int c = fgetc(fp);
+                         sprintf(conv2, "%i", c);
+                         strcat(ret_str, conv2);
+                         Writeline(conn_s, ret_str, MAX_LINE-1);
+                         printf("c: %c\n", c);
+
                          c = fgetc(fp);
-                         // printf("c: %c\n", c);
-                         sprintf(&conv2, "%i", c);
-                         strcat(ret_str, &conv2);
-                         // Writeline(conn_s, ret_str, MAX_LINE-1);
-                         // char c = fgetc
-                         // while ((char) fgetc(fp) != EOF) {
-                         //      memset(ret_str, 0, sizeof(ret_str));
-                         //      if (strlen(ret_str) == MAX_LINE-1) {
-                         //              Writeline(conn_s, ret_str, MAX_LINE-1);
-                         //      } else {
-                         //              strcat(ret_str, getchar(fp));
-                         //      }
-                         // }
+                         while (c != EOF) {
+                                 if (strlen(ret_str) == MAX_LINE-8){
+                                         Writeline(conn_s, ret_str, MAX_LINE-1);
+                                         memset(ret_str, 0, MAX_LINE);
+                                 } else {
+                                         c = fgetc(fp);
+                                         printf("C: %c", c);
+                                         sprintf(conv2, "%d", c);
+                                         strcat(ret_str, conv2);
+                                 }
+                         }
+                         fclose(fp);
+                         // break;
                  }
          }
```