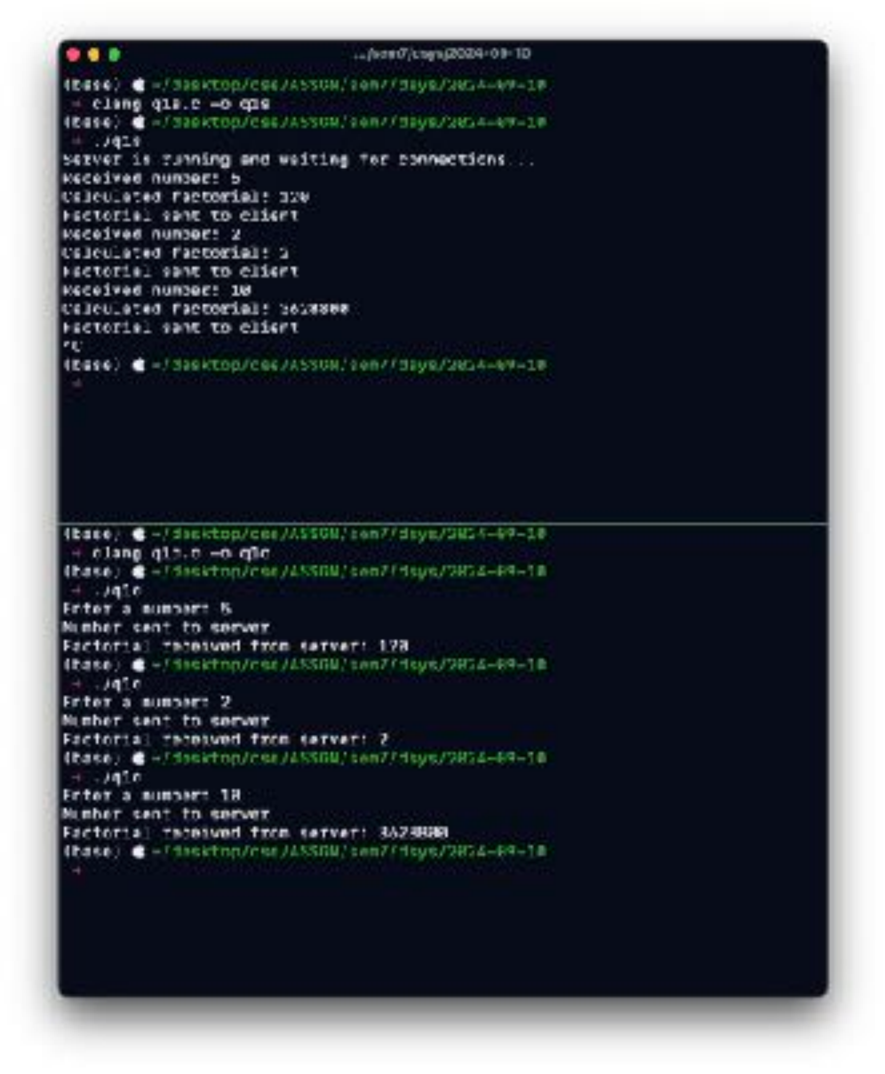
ASSIGNMENT 5
10/09/24

NAME : SHRESTH SONKAR
REGNO : 20214272
GROUP : CS7D
TOPIC : DISTRIBUTED SYSTEM
CODE : CS-17201

```
//Q1 Socket server program for factorial

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>

long long factorial(int n) {
    long long result = 1;
    for (int i = 1; i <= n; i++) {
        result *= i;
    }
    return result;
}

int main() {
    int server_fd, new_socket;
    struct sockaddr_in address;
    int addrlen = sizeof(address);
    int number;
    long long fact;

    if ((server_fd = socket(AF_INET, SOCK_STREAM, 0))
== 0) {
        perror("Socket failed");
        exit(EXIT_FAILURE);
    }

    address.sin_family = AF_INET;
    address.sin_addr.s_addr = INADDR_ANY;
    address.sin_port = htons(8080);

    if (bind(server_fd, (struct sockaddr *) &address,
sizeof(address)) < 0) {
        perror("Bind failed");
        exit(EXIT_FAILURE);
    }
    if (listen(server_fd, 3) < 0) {
        perror("Listen failed");
        exit(EXIT_FAILURE);
    }

    printf("Server is running and waiting for
connections...\n");
```

```
    while (1) {
        if ((new_socket = accept(server_fd, (struct
sockaddr *) &address, (socklen_t * ) & addrlen)) < 0) {
            perror("Accept failed");
            exit(EXIT_FAILURE);
        }

        read(new_socket, &number, sizeof(number));
        printf("Received number: %d\n", number);

        fact = factorial(number);
        printf("Calculated factorial: %lld\n", fact);

        send(new_socket, &fact, sizeof(fact), 0);
        printf("Factorial sent to client\n");

        close(new_socket);
    }
    close(server_fd);
    return 0;
}

//Q1 Socket client program for factorial

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>

int main() {
    int sock = 0;
    struct sockaddr_in serv_addr;
    int number;
    long long fact;

    if ((sock = socket(AF_INET, SOCK_STREAM, 0)) < 0) {
        printf("\n Socket creation error \n");
        return -1;
    }

    serv_addr.sin_family = AF_INET;
    serv_addr.sin_port = htons(8080);
    if (inet_pton(AF_INET, "127.0.0.1",
&serv_addr.sin_addr) <= 0) {
        printf("\nInvalid address/ Address not
supported \n");
        return -1;
    }
    if (connect(sock, (struct sockaddr *) &serv_addr,
sizeof(serv_addr)) < 0) {
        printf("\nConnection Failed \n");
        return -1;
    }

    printf("Enter a number: ");
    scanf("%d", &number);
    send(sock, &number, sizeof(number), 0);
    printf("Number sent to server\n");
    read(sock, &fact, sizeof(fact));
    printf("Factorial received from server: %lld\n",
fact);

    close(sock);
    return 0;
}
```

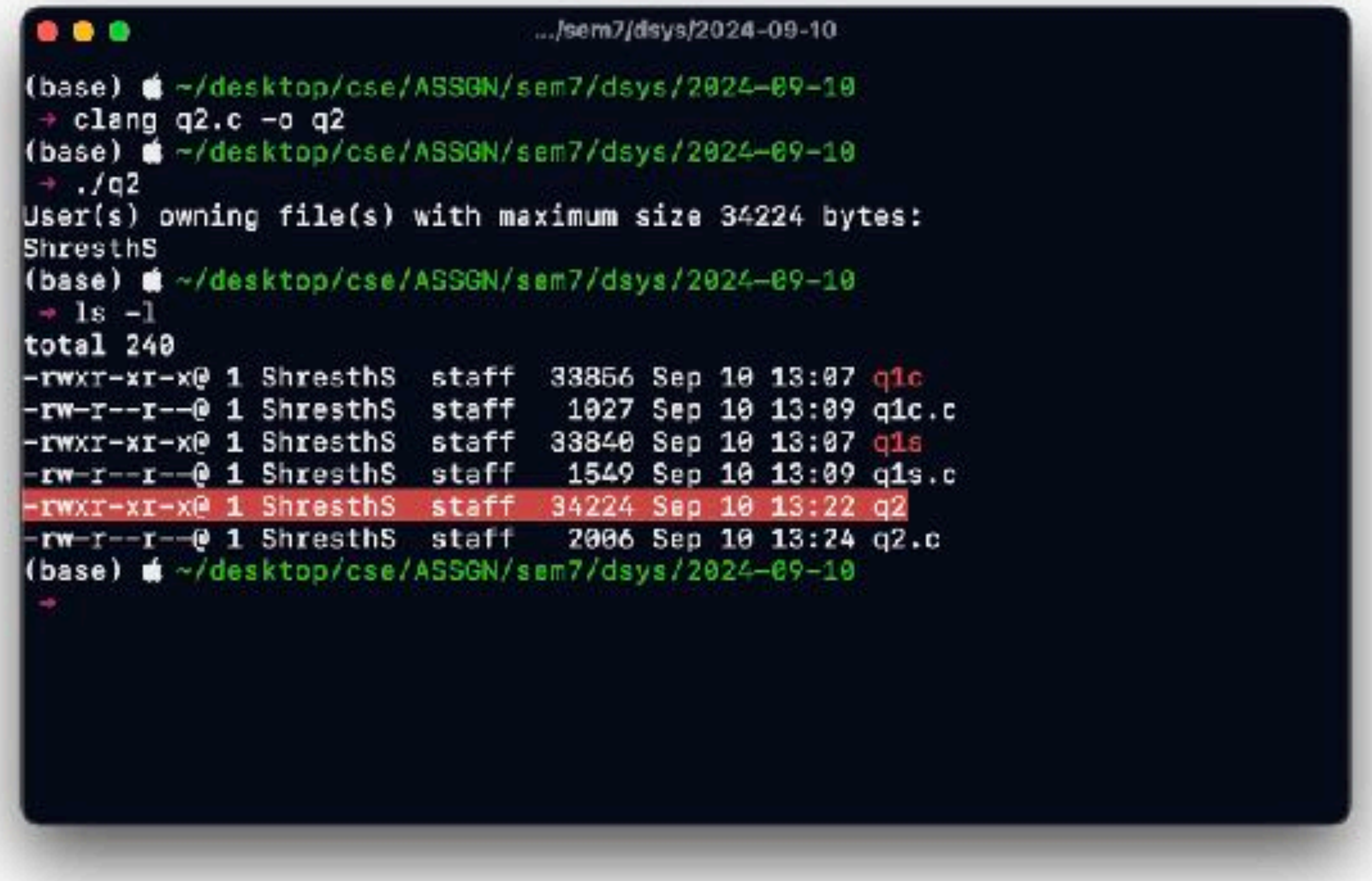


```
(base)  ~/Desktop/cse/ASSGN/sem7/dsys/2024-09-10
→ clang q1s.c -o q1s
(base)  ~/Desktop/cse/ASSGN/sem7/dsys/2024-09-10
→ ./q1s
Server is running and waiting for connections...
Received number: 5
Calculated factorial: 120
Factorial sent to client
Received number: 2
Calculated factorial: 2
Factorial sent to client
Received number: 10
Calculated factorial: 3628800
Factorial sent to client
^C
(base)  ~/Desktop/cse/ASSGN/sem7/dsys/2024-09-10
→
```

```
(base)  ~/Desktop/cse/ASSGN/sem7/dsys/2024-09-10
→ clang q1c.c -o q1c
(base)  ~/Desktop/cse/ASSGN/sem7/dsys/2024-09-10
→ ./q1c
Enter a number: 5
Number sent to server
Factorial received from server: 120
(base)  ~/Desktop/cse/ASSGN/sem7/dsys/2024-09-10
→ ./q1c
Enter a number: 2
Number sent to server
Factorial received from server: 2
(base)  ~/Desktop/cse/ASSGN/sem7/dsys/2024-09-10
→ ./q1c
Enter a number: 10
Number sent to server
Factorial received from server: 3628800
(base)  ~/Desktop/cse/ASSGN/sem7/dsys/2024-09-10
→
```

```c
//Q2 : Map Reduce for file owner with max size

#include <stdio.h>
#include <stdlib.h>
#include <dirent.h>
#include <sys/stat.h>
#include <unistd.h>
#include <pwd.h>
#include <string.h>

struct file_info {
    char owner[256];
    off_t size;
};

void get_file_owner(uid_t uid, char *owner_name) {
    struct passwd *pwd = getpwuid(uid);
    if (pwd) {
        strcpy(owner_name, pwd->pw_name);
    } else {
        strcpy(owner_name, "Unknown");
    }
}

void map_files(const char *directory, struct file_info
*file_list, int *count) {
    DIR *dir;
    struct dirent *entry;
    struct stat file_stat;
    char filepath[1024];

    if ((dir = opendir(directory)) == NULL) {
        perror("opendir() error");
        exit(EXIT_FAILURE);
    }

    *count = 0;
    while ((entry = readdir(dir)) != NULL) {
        if (strcmp(entry->d_name, ".") == 0 ||
strcmp(entry->d_name, "..") == 0)
            continue;

        snprintf(filepath, sizeof(filepath), "%s/%s",
directory, entry->d_name);
```

```c
        if (stat(filepath, &file_stat) == 0) {
            if (S_ISREG(file_stat.st_mode)) {
                get_file_owner(file_stat.st_uid,
file_list[*count].owner);
                file_list[*count].size =
file_stat.st_size;
                (*count)++;
            }
        }
    }
    closedir(dir);
}

void reduce_files(struct file_info *file_list, int
count) {
    off_t max_size = 0;

    for (int i = 0; i < count; i++) {
        if (file_list[i].size > max_size)
            max_size = file_list[i].size;
    }

    printf("User(s) owning file(s) with maximum size
%lld bytes:\n", (long long) max_size);
    for (int i = 0; i < count; i++) {
        if (file_list[i].size == max_size)
            printf("%s\n", file_list[i].owner);
    }
}

int main() {
    const char *directory = ".";
    struct file_info file_list[1024];
    int file_count = 0;

    map_files(directory, file_list, &file_count);

    if (file_count > 0)
        reduce_files(file_list, file_count);
    else
        printf("No files found in the directory.\n");


    return 0;
}
```



```
(base)  ~/desktop/cse/ASSGN/sem7/dsys/2024-09-10
 → clang q2.c -o q2
(base)  ~/desktop/cse/ASSGN/sem7/dsys/2024-09-10
 → ./q2
User(s) owning file(s) with maximum size 34224 bytes:
ShresthS
(base)  ~/desktop/cse/ASSGN/sem7/dsys/2024-09-10
 → ls -l
total 240
-rwxr-xr-x@ 1 ShresthS  staff  33856 Sep 10 13:07 q1c
-rw-r--r--@ 1 ShresthS  staff   1027 Sep 10 13:09 q1c.c
-rwxr-xr-x@ 1 ShresthS  staff  33840 Sep 10 13:07 q1s
-rw-r--r--@ 1 ShresthS  staff   1549 Sep 10 13:09 q1s.c
-rwxr-xr-x@ 1 ShresthS  staff  34224 Sep 10 13:22 q2
-rw-r--r--@ 1 ShresthS  staff   2006 Sep 10 13:24 q2.c
(base)  ~/desktop/cse/ASSGN/sem7/dsys/2024-09-10
 →
```