CSE 232: Programming Assignment 2
Socket programming with performance analysis

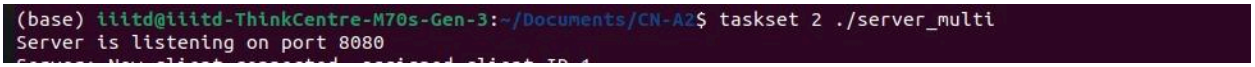
Subham Maurya (2022510), Sujal Soni (2022513)

Q1.

Report for Client-Server Socket Program in C

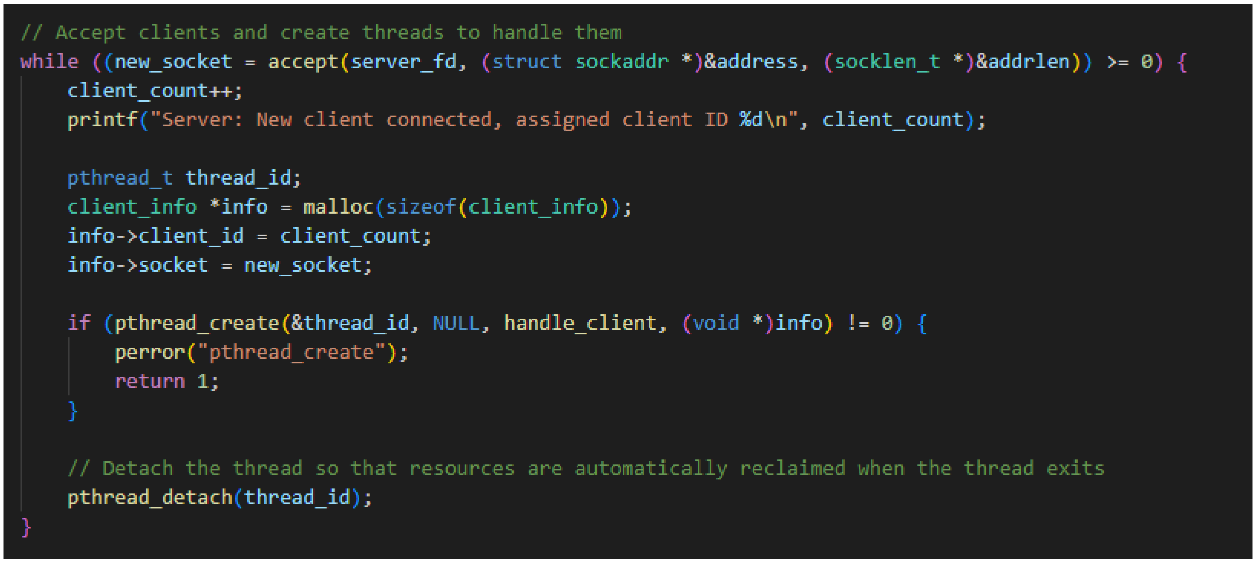
Objective:
To create a client-server socket communication program in C where the client and server processes run on separate virtual machines (VMs) or containers. The communication includes using “taskset” to pin the processes to specific CPUs for performance measurement. The server retrieves the top two CPU-consuming processes on request and sends this information to the client.

Features Implemented:
1. TCP Socket Setup by Server: The server sets up a TCP socket on a specific port (8080) and listens for incoming client connections.



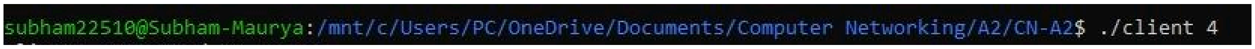
```
(base) iiitd@iiitd-ThinkCentre-M70s-Gen-3:~/Documents/CN-A2$ taskset 2 ./server_multi
Server is listening on port 8080
```

2. Handling Multiple Clients (Multithreaded Server):
The server can handle multiple concurrent clients by creating separate threads for each client connection.The main server socket continues to listen for new clients while individual client requests are handled by separate threads (using the pthread library).



```c
// Accept clients and create threads to handle them
while ((new_socket = accept(server_fd, (struct sockaddr *)&address, (socklen_t *)&addrlen)) >= 0) {
    client_count++;
    printf("Server: New client connected, assigned client ID %d\n", client_count);

    pthread_t thread_id;
    client_info *info = malloc(sizeof(client_info));
    info->client_id = client_count;
    info->socket = new_socket;

    if (pthread_create(&thread_id, NULL, handle_client, (void *)info) != 0) {
        perror("pthread_create");
        return 1;
    }

    // Detach the thread so that resources are automatically reclaimed when the thread exits
    pthread_detach(thread_id);
}
```

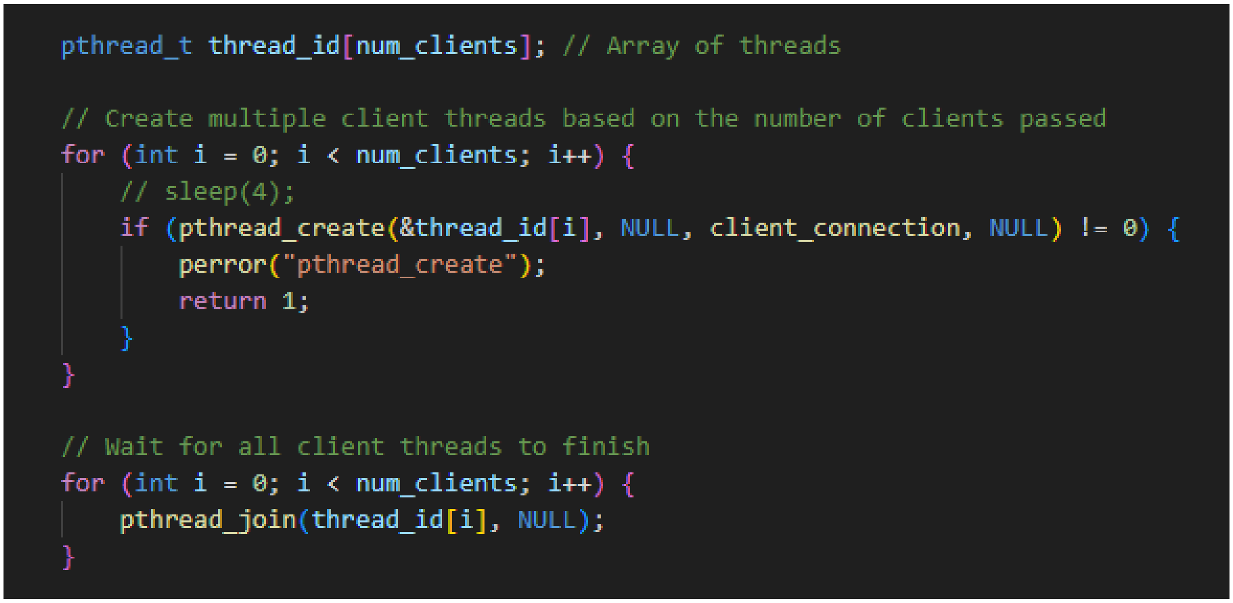
3. Client Socket Connection:
The client program creates a socket to connect to the server using a specific IP address. It supports initiating multiple concurrent client connections (controlled by the argument n).



```
subham22510@Subham-Maurya:/mnt/c/Users/PC/OneDrive/Documents/Computer Networking/A2/CN-A2$ ./client 4
```

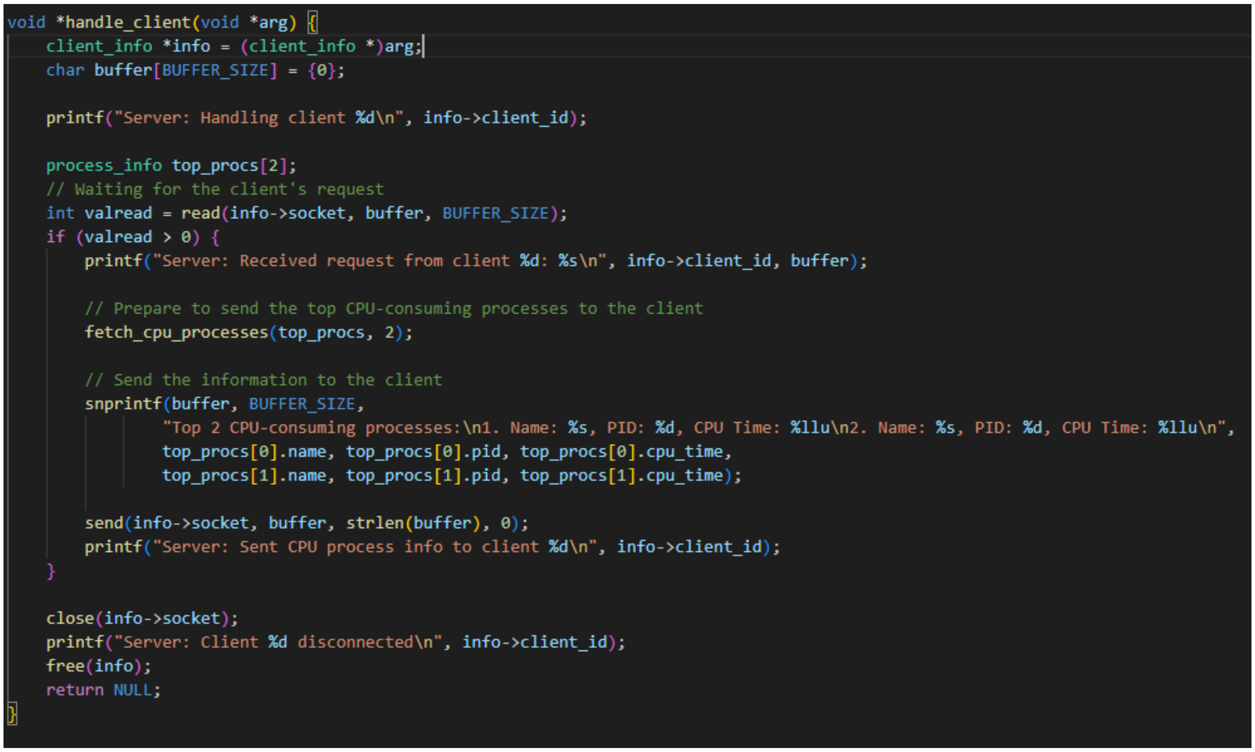
4. Client Request for Top CPU-consuming Processes:
After establishing a connection, the client sends a request to the server for information on the top two CPU-consuming processes. The server reads the **`/proc/[pid]/stat`** files to gather data on each process, including the process name, PID, and CPU usage time in user and kernel mode.

Client Requesting for top CPU processes



```c
pthread_t thread_id[num_clients]; // Array of threads

// Create multiple client threads based on the number of clients passed
for (int i = 0; i < num_clients; i++) {
    // sleep(4);
    if (pthread_create(&thread_id[i], NULL, client_connection, NULL) != 0) {
        perror("pthread_create");
        return 1;
    }
}

// Wait for all client threads to finish
for (int i = 0; i < num_clients; i++) {
    pthread_join(thread_id[i], NULL);
}
```

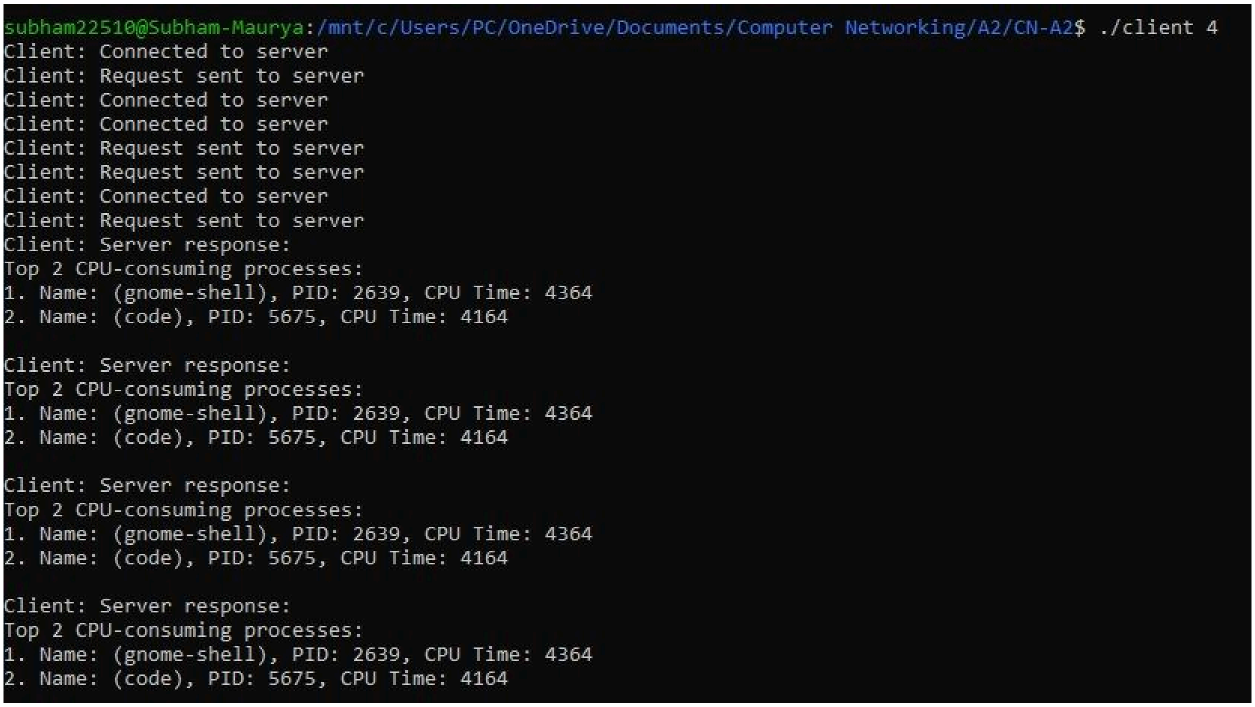
5. Server Response to Client:
After fetching the top two CPU-consuming processes, the server sends the details (including process name, PID, and CPU time) to the client. The server uses sorting to find the two processes with the highest CPU time.



```
void *handle_client(void *arg) {
    client_info *info = (client_info *)arg;
    char buffer[BUFFER_SIZE] = {0};

    printf("Server: Handling client %d\n", info->client_id);

    process_info top_procs[2];
    // Waiting for the client's request
    int valread = read(info->socket, buffer, BUFFER_SIZE);
    if (valread > 0) {
        printf("Server: Received request from client %d: %s\n", info->client_id, buffer);

        // Prepare to send the top CPU-consuming processes to the client
        fetch_cpu_processes(top_procs, 2);

        // Send the information to the client
        snprintf(buffer, BUFFER_SIZE,
                "Top 2 CPU-consuming processes:\n1. Name: %s, PID: %d, CPU Time: %llu\n2. Name: %s, PID: %d, CPU Time: %llu\n",
                top_procs[0].name, top_procs[0].pid, top_procs[0].cpu_time,
                top_procs[1].name, top_procs[1].pid, top_procs[1].cpu_time);

        send(info->socket, buffer, strlen(buffer), 0);
        printf("Server: Sent CPU process info to client %d\n", info->client_id);
    }

    close(info->socket);
    printf("Server: Client %d disconnected\n", info->client_id);
    free(info);
    return NULL;
}
```

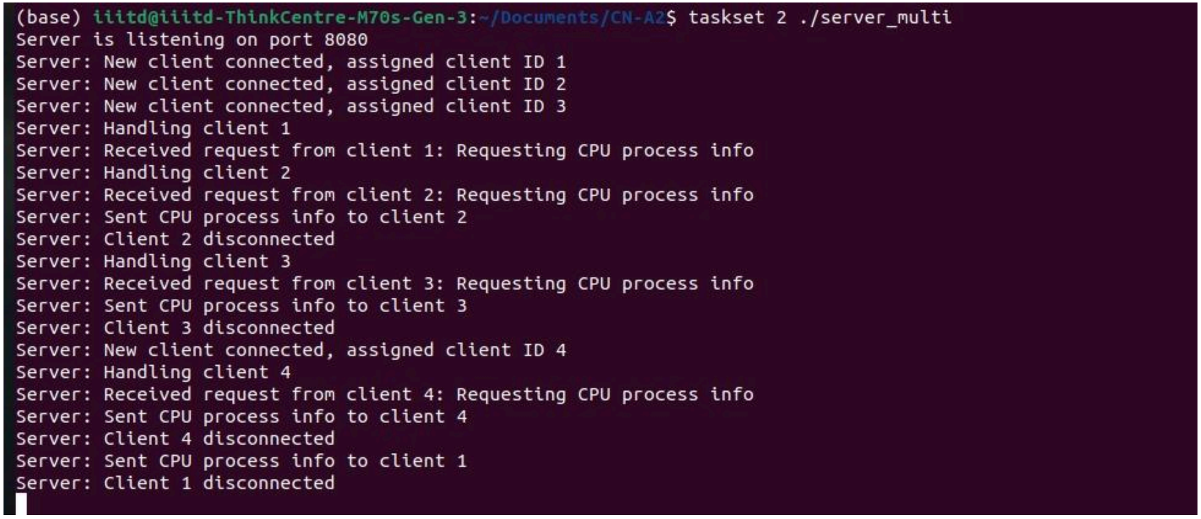
6. Client Output:

The client receives the server’s response and prints the information about the top two CPU-consuming processes.The client then closes the connection.

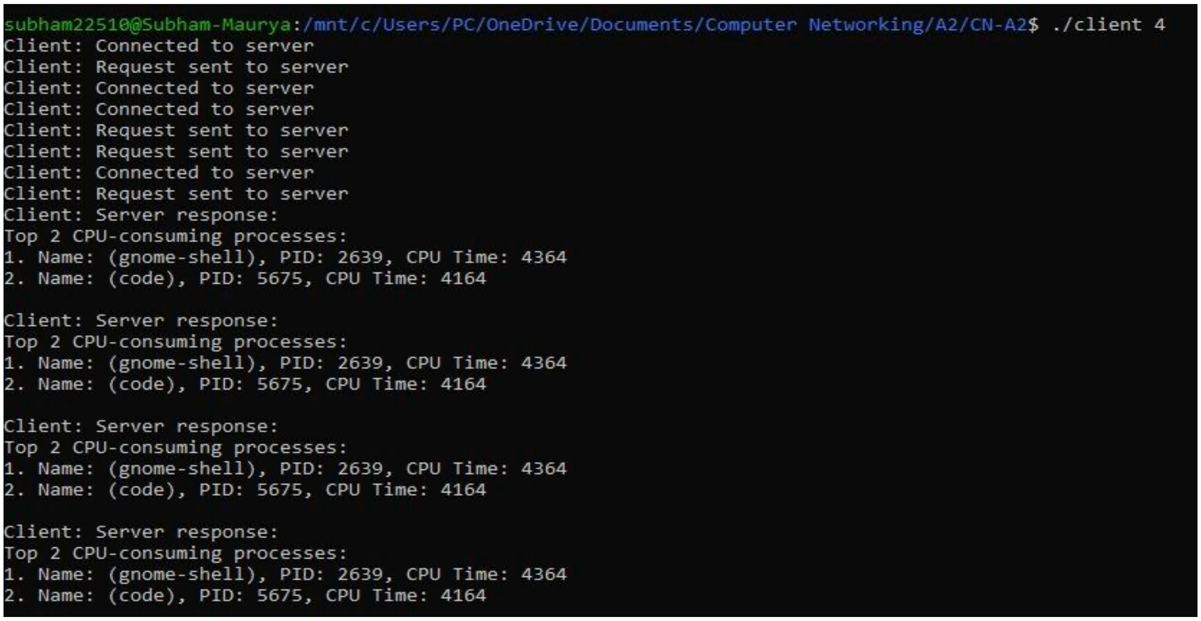


```
subham22510@Subham-Maurya:/mnt/c/Users/PC/OneDrive/Documents/Computer Networking/A2/CN-A2$ ./client 4
Client: Connected to server
Client: Request sent to server
Client: Connected to server
Client: Connected to server
Client: Request sent to server
Client: Request sent to server
Client: Connected to server
Client: Request sent to server
Client: Server response:
Top 2 CPU-consuming processes:
1. Name: (gnome-shell), PID: 2639, CPU Time: 4364
2. Name: (code), PID: 5675, CPU Time: 4164

Client: Server response:
Top 2 CPU-consuming processes:
1. Name: (gnome-shell), PID: 2639, CPU Time: 4364
2. Name: (code), PID: 5675, CPU Time: 4164

Client: Server response:
Top 2 CPU-consuming processes:
1. Name: (gnome-shell), PID: 2639, CPU Time: 4364
2. Name: (code), PID: 5675, CPU Time: 4164

Client: Server response:
Top 2 CPU-consuming processes:
1. Name: (gnome-shell), PID: 2639, CPU Time: 4364
2. Name: (code), PID: 5675, CPU Time: 4164
```

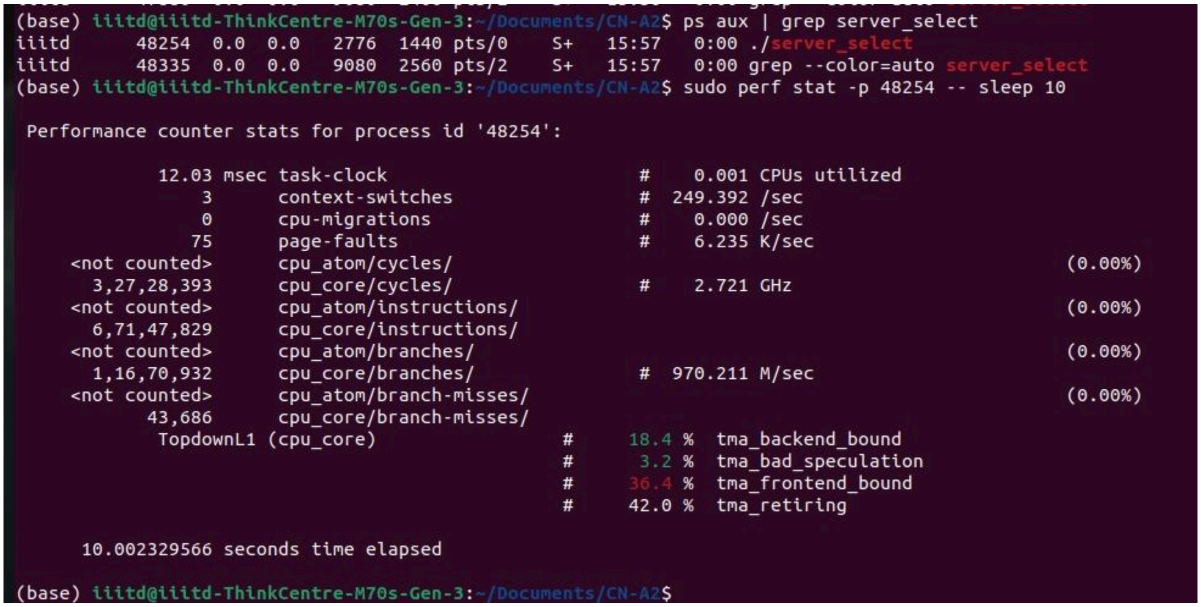
Testing, Observations, and Performance Measurement:



```
(base) iiitd@iiitd-ThinkCentre-M70s-Gen-3:~/Documents/CN-A2$ taskset 2 ./server_multi
Server is listening on port 8080
Server: New client connected, assigned client ID 1
Server: New client connected, assigned client ID 2
Server: New client connected, assigned client ID 3
Server: Handling client 1
Server: Received request from client 1: Requesting CPU process info
Server: Handling client 2
Server: Received request from client 2: Requesting CPU process info
Server: Sent CPU process info to client 2
Server: Client 2 disconnected
Server: Handling client 3
Server: Received request from client 3: Requesting CPU process info
Server: Sent CPU process info to client 3
Server: Client 3 disconnected
Server: New client connected, assigned client ID 4
Server: Handling client 4
Server: Received request from client 4: Requesting CPU process info
Server: Sent CPU process info to client 4
Server: Client 4 disconnected
Server: Sent CPU process info to client 1
Server: Client 1 disconnected
```



```
subham22510@Subham-Maurya:/mnt/c/Users/PC/OneDrive/Documents/Computer Networking/A2/CN-A2$ ./client 4
Client: Connected to server
Client: Request sent to server
Client: Connected to server
Client: Connected to server
Client: Request sent to server
Client: Request sent to server
Client: Connected to server
Client: Request sent to server
Client: Server response:
Top 2 CPU-consuming processes:
1. Name: (gnome-shell), PID: 2639, CPU Time: 4364
2. Name: (code), PID: 5675, CPU Time: 4164

Client: Server response:
Top 2 CPU-consuming processes:
1. Name: (gnome-shell), PID: 2639, CPU Time: 4364
2. Name: (code), PID: 5675, CPU Time: 4164

Client: Server response:
Top 2 CPU-consuming processes:
1. Name: (gnome-shell), PID: 2639, CPU Time: 4364
2. Name: (code), PID: 5675, CPU Time: 4164

Client: Server response:
Top 2 CPU-consuming processes:
1. Name: (gnome-shell), PID: 2639, CPU Time: 4364
2. Name: (code), PID: 5675, CPU Time: 4164
```



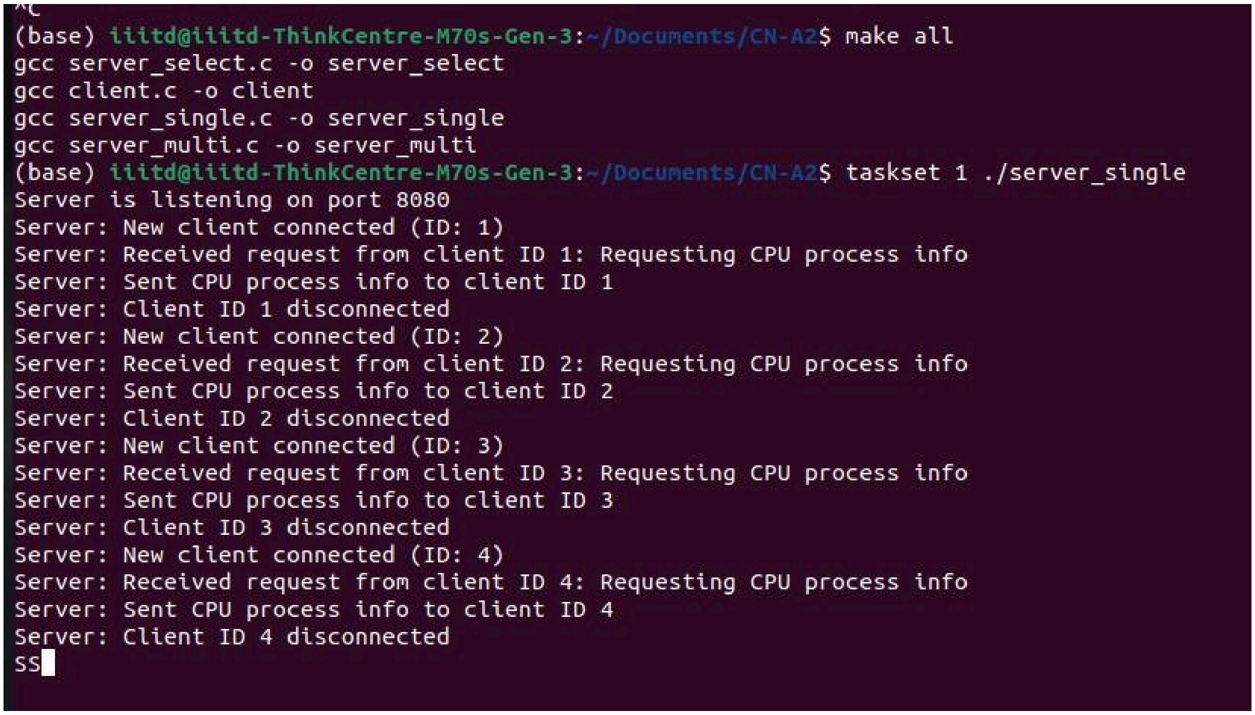
```
(base) iiitd@iiitd-ThinkCentre-M70s-Gen-3:~/Documents/CN-A2$ ps aux | grep server_select
iiitd      48254  0.0  0.0   2776  1440 pts/0    S+   15:57   0:00 ./server_select
iiitd      48335  0.0  0.0   9080  2560 pts/2    S+   15:57   0:00 grep --color=auto server_select
(base) iiitd@iiitd-ThinkCentre-M70s-Gen-3:~/Documents/CN-A2$ sudo perf stat -p 48254 -- sleep 10

 Performance counter stats for process id '48254':

            12.03 msec task-clock                       #    0.001 CPUs utilized
                3      context-switches                 #  249.392 /sec
                0      cpu-migrations                   #    0.000 /sec
               75      page-faults                      #    6.235 K/sec
     <not counted>      cpu_atom/cycles/                                                    (0.00%)
       3,27,28,393      cpu_core/cycles/                #    2.721 GHz
     <not counted>      cpu_atom/instructions/                                              (0.00%)
       6,71,47,829      cpu_core/instructions/
     <not counted>      cpu_atom/branches/                                                  (0.00%)
       1,16,70,932      cpu_core/branches/              #  970.211 M/sec
     <not counted>      cpu_atom/branch-misses/                                             (0.00%)
           43,686      cpu_core/branch-misses/
            TopdownL1 (cpu_core)                 #     18.4 %  tma_backend_bound
                                                 #      3.2 %  tma_bad_speculation
                                                 #     36.4 %  tma_frontend_bound
                                                 #     42.0 %  tma_retiring

      10.002329566 seconds time elapsed

(base) iiitd@iiitd-ThinkCentre-M70s-Gen-3:~/Documents/CN-A2$
```

Q2.

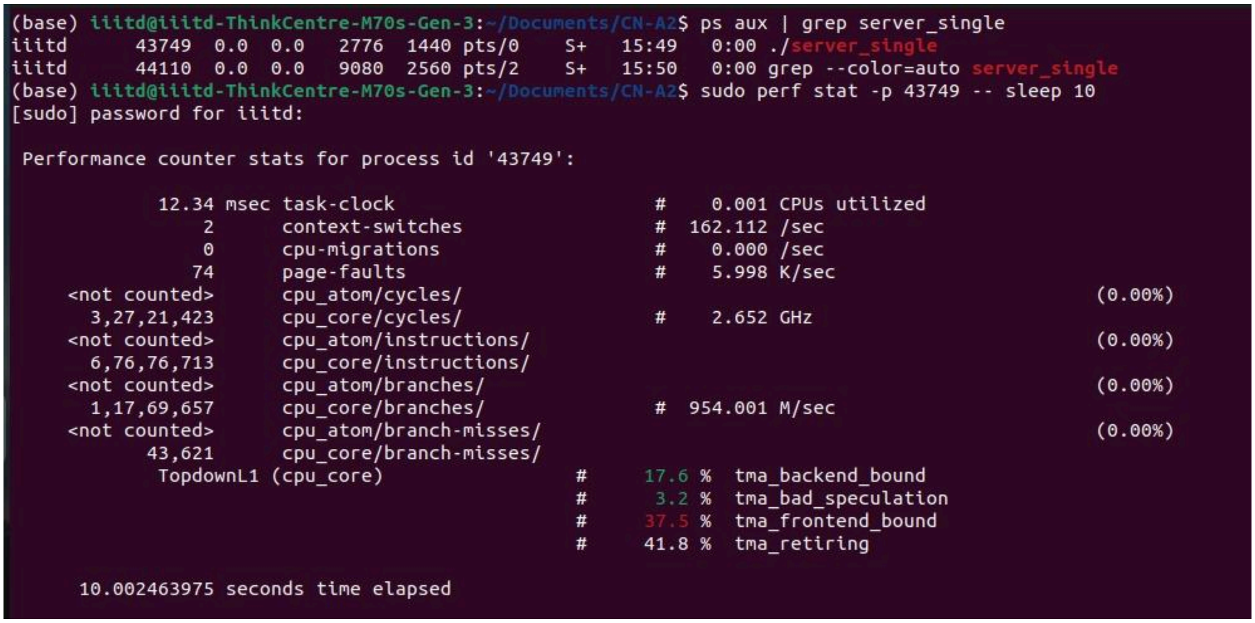
The performance measurement of the (a), (b), and (c) are listed below:

(a). Single-threaded TCP client-server:

Server Side:

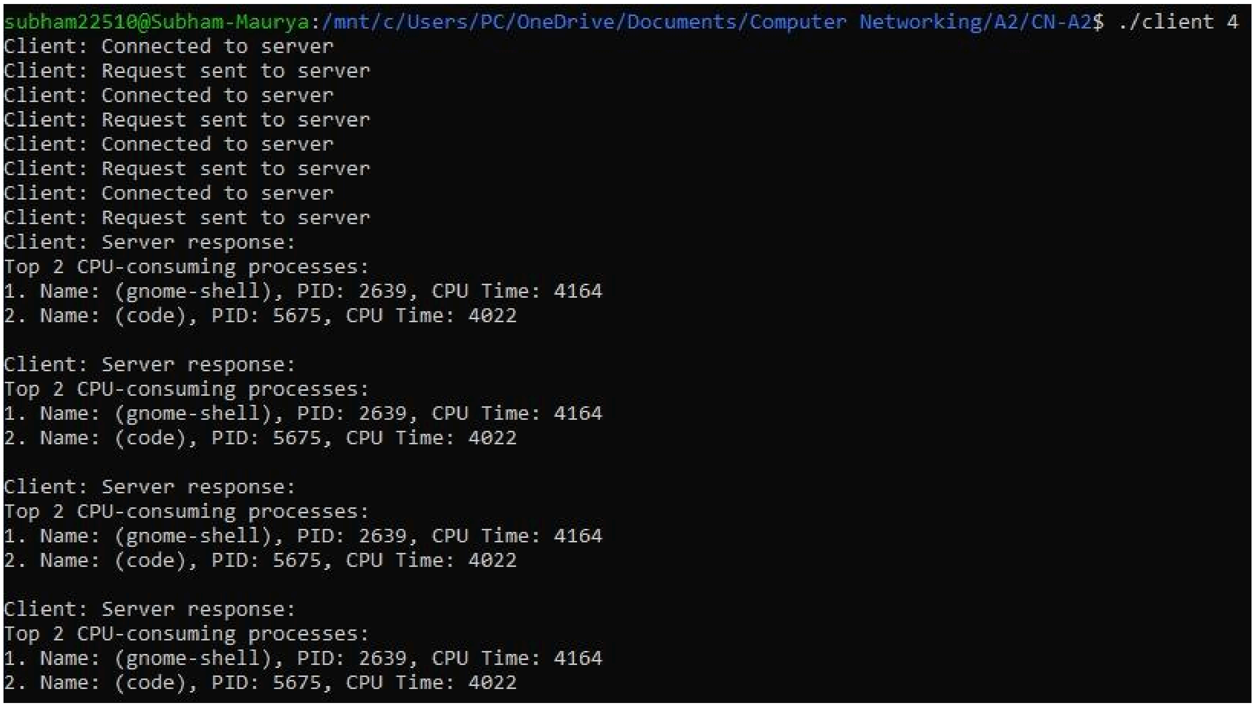


```
^C
(base) iiitd@iiitd-ThinkCentre-M70s-Gen-3:~/Documents/CN-A2$ make all
gcc server_select.c -o server_select
gcc client.c -o client
gcc server_single.c -o server_single
gcc server_multi.c -o server_multi
(base) iiitd@iiitd-ThinkCentre-M70s-Gen-3:~/Documents/CN-A2$ taskset 1 ./server_single
Server is listening on port 8080
Server: New client connected (ID: 1)
Server: Received request from client ID 1: Requesting CPU process info
Server: Sent CPU process info to client ID 1
Server: Client ID 1 disconnected
Server: New client connected (ID: 2)
Server: Received request from client ID 2: Requesting CPU process info
Server: Sent CPU process info to client ID 2
Server: Client ID 2 disconnected
Server: New client connected (ID: 3)
Server: Received request from client ID 3: Requesting CPU process info
Server: Sent CPU process info to client ID 3
Server: Client ID 3 disconnected
Server: New client connected (ID: 4)
Server: Received request from client ID 4: Requesting CPU process info
Server: Sent CPU process info to client ID 4
Server: Client ID 4 disconnected
ss
```



```
(base) iiitd@iiitd-ThinkCentre-M70s-Gen-3:~/Documents/CN-A2$ ps aux | grep server_single
iiitd      43749  0.0  0.0   2776  1440 pts/0    S+   15:49   0:00 ./server_single
iiitd      44110  0.0  0.0   9080  2560 pts/2    S+   15:50   0:00 grep --color=auto server_single
(base) iiitd@iiitd-ThinkCentre-M70s-Gen-3:~/Documents/CN-A2$ sudo perf stat -p 43749 -- sleep 10
[sudo] password for iiitd:

 Performance counter stats for process id '43749':

             12.34 msec task-clock                       #    0.001 CPUs utilized
                 2      context-switches                 #  162.112 /sec
                 0      cpu-migrations                   #    0.000 /sec
                74      page-faults                      #    5.998 K/sec
     <not counted>      cpu_atom/cycles/                                                      (0.00%)
       3,27,21,423      cpu_core/cycles/                 #    2.652 GHz
     <not counted>      cpu_atom/instructions/                                                (0.00%)
       6,76,76,713      cpu_core/instructions/
     <not counted>      cpu_atom/branches/                                                    (0.00%)
       1,17,69,657      cpu_core/branches/               #  954.001 M/sec
     <not counted>      cpu_atom/branch-misses/                                               (0.00%)
            43,621      cpu_core/branch-misses/
             TopdownL1 (cpu_core)                 #     17.6 %  tma_backend_bound
                                                  #      3.2 %  tma_bad_speculation
                                                  #     37.5 %  tma_frontend_bound
                                                  #     41.8 %  tma_retiring

      10.002463975 seconds time elapsed
```

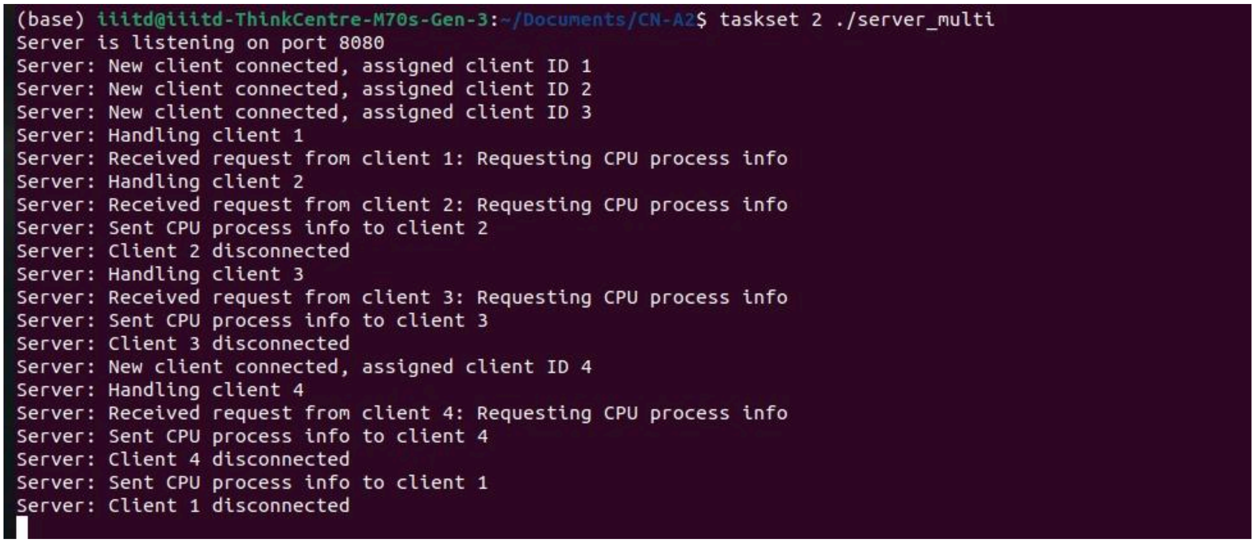
Client Side:



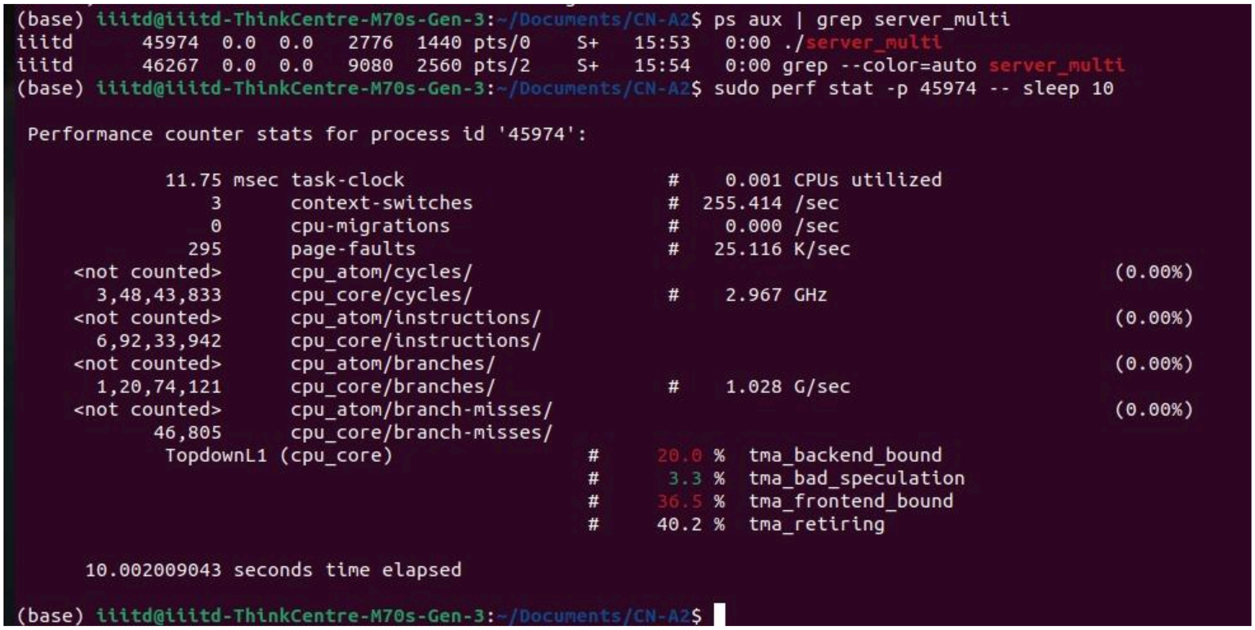
```
subham22510@Subham-Maurya:/mnt/c/Users/PC/OneDrive/Documents/Computer Networking/A2/CN-A2$ ./client 4
Client: Connected to server
Client: Request sent to server
Client: Connected to server
Client: Request sent to server
Client: Connected to server
Client: Request sent to server
Client: Connected to server
Client: Request sent to server
Client: Server response:
Top 2 CPU-consuming processes:
1. Name: (gnome-shell), PID: 2639, CPU Time: 4164
2. Name: (code), PID: 5675, CPU Time: 4022

Client: Server response:
Top 2 CPU-consuming processes:
1. Name: (gnome-shell), PID: 2639, CPU Time: 4164
2. Name: (code), PID: 5675, CPU Time: 4022

Client: Server response:
Top 2 CPU-consuming processes:
1. Name: (gnome-shell), PID: 2639, CPU Time: 4164
2. Name: (code), PID: 5675, CPU Time: 4022

Client: Server response:
Top 2 CPU-consuming processes:
1. Name: (gnome-shell), PID: 2639, CPU Time: 4164
2. Name: (code), PID: 5675, CPU Time: 4022
```

(b). Concurrent TCP client-server:

Server Side:

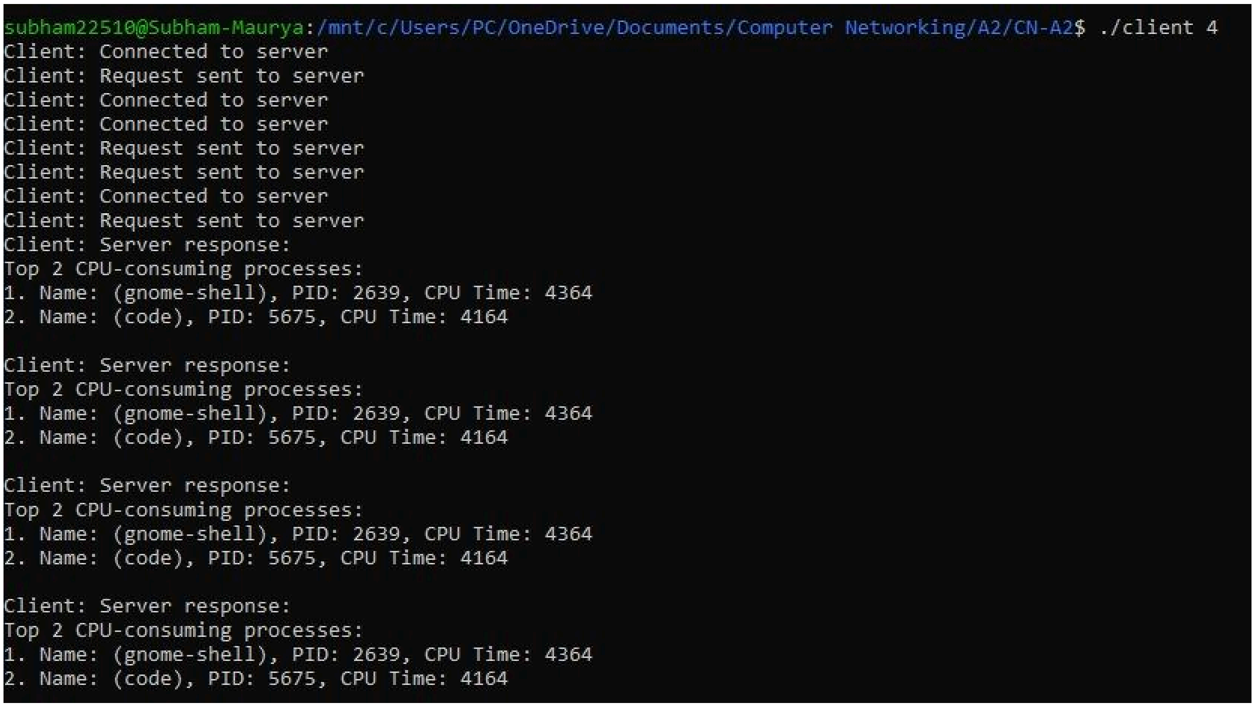


```
(base) iiitd@iiitd-ThinkCentre-M70s-Gen-3:~/Documents/CN-A2$ taskset 2 ./server_multi
Server is listening on port 8080
Server: New client connected, assigned client ID 1
Server: New client connected, assigned client ID 2
Server: New client connected, assigned client ID 3
Server: Handling client 1
Server: Received request from client 1: Requesting CPU process info
Server: Handling client 2
Server: Received request from client 2: Requesting CPU process info
Server: Sent CPU process info to client 2
Server: Client 2 disconnected
Server: Handling client 3
Server: Received request from client 3: Requesting CPU process info
Server: Sent CPU process info to client 3
Server: Client 3 disconnected
Server: New client connected, assigned client ID 4
Server: Handling client 4
Server: Received request from client 4: Requesting CPU process info
Server: Sent CPU process info to client 4
Server: Client 4 disconnected
Server: Sent CPU process info to client 1
Server: Client 1 disconnected
```



```
(base) iiitd@iiitd-ThinkCentre-M70s-Gen-3:~/Documents/CN-A2$ ps aux | grep server_multi
iiitd      45974  0.0  0.0   2776  1440 pts/0    S+   15:53   0:00 ./server_multi
iiitd      46267  0.0  0.0   9080  2560 pts/2    S+   15:54   0:00 grep --color=auto server_multi
(base) iiitd@iiitd-ThinkCentre-M70s-Gen-3:~/Documents/CN-A2$ sudo perf stat -p 45974 -- sleep 10

 Performance counter stats for process id '45974':

            11.75 msec task-clock                       #    0.001 CPUs utilized
                3      context-switches                 #  255.414 /sec
                0      cpu-migrations                   #    0.000 /sec
              295      page-faults                      #   25.116 K/sec
     <not counted>      cpu_atom/cycles/                                                        (0.00%)
       3,48,43,833      cpu_core/cycles/                #    2.967 GHz
     <not counted>      cpu_atom/instructions/                                                  (0.00%)
       6,92,33,942      cpu_core/instructions/
     <not counted>      cpu_atom/branches/                                                      (0.00%)
       1,20,74,121      cpu_core/branches/              #    1.028 G/sec
     <not counted>      cpu_atom/branch-misses/                                                 (0.00%)
            46,805      cpu_core/branch-misses/
            TopdownL1 (cpu_core)                 #     20.0 %  tma_backend_bound
                                                 #      3.3 %  tma_bad_speculation
                                                 #     36.5 %  tma_frontend_bound
                                                 #     40.2 %  tma_retiring

      10.002009043 seconds time elapsed

(base) iiitd@iiitd-ThinkCentre-M70s-Gen-3:~/Documents/CN-A2$
```

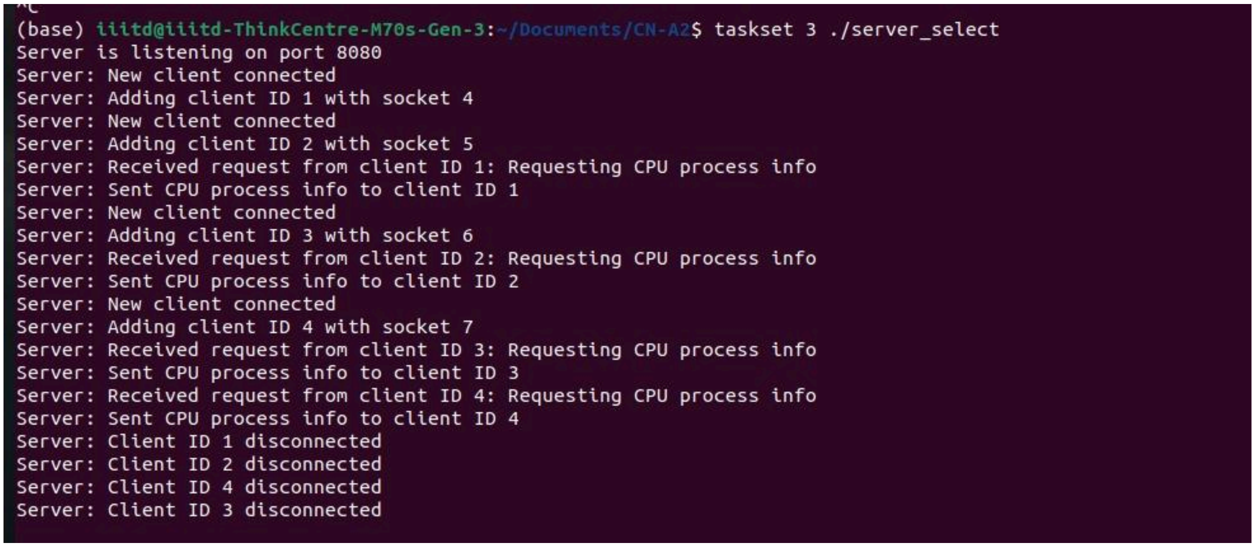
Client Side:



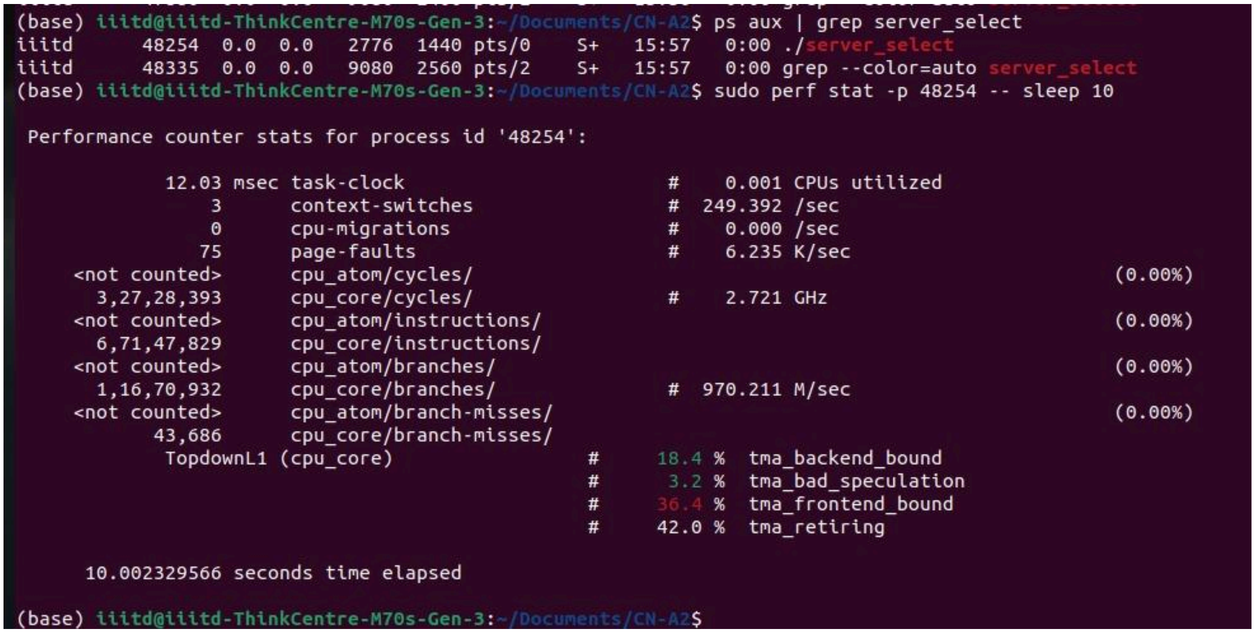
```
subham22510@Subham-Maurya:/mnt/c/Users/PC/OneDrive/Documents/Computer Networking/A2/CN-A2$ ./client 4
Client: Connected to server
Client: Request sent to server
Client: Connected to server
Client: Connected to server
Client: Request sent to server
Client: Request sent to server
Client: Connected to server
Client: Request sent to server
Client: Server response:
Top 2 CPU-consuming processes:
1. Name: (gnome-shell), PID: 2639, CPU Time: 4364
2. Name: (code), PID: 5675, CPU Time: 4164

Client: Server response:
Top 2 CPU-consuming processes:
1. Name: (gnome-shell), PID: 2639, CPU Time: 4364
2. Name: (code), PID: 5675, CPU Time: 4164

Client: Server response:
Top 2 CPU-consuming processes:
1. Name: (gnome-shell), PID: 2639, CPU Time: 4364
2. Name: (code), PID: 5675, CPU Time: 4164

Client: Server response:
Top 2 CPU-consuming processes:
1. Name: (gnome-shell), PID: 2639, CPU Time: 4364
2. Name: (code), PID: 5675, CPU Time: 4164
```

(c). TCP client-server using “select”:

Server Side:

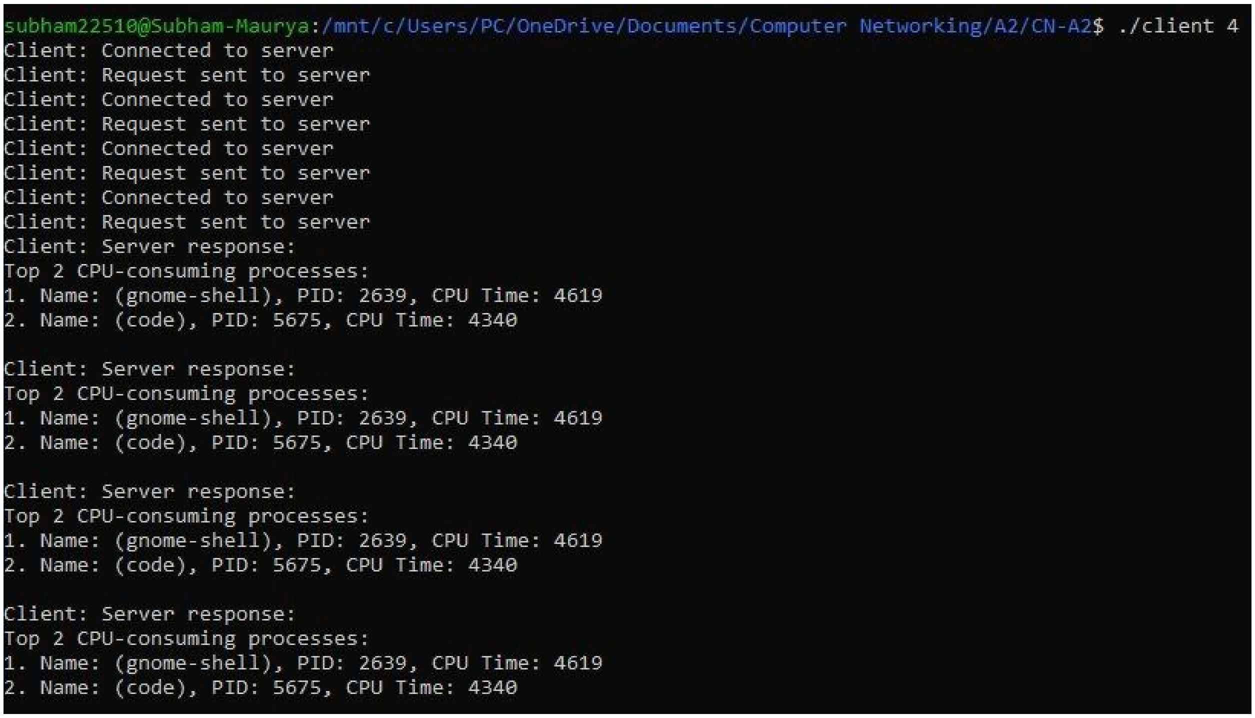


```
^C
(base) iiitd@iiitd-ThinkCentre-M70s-Gen-3:~/Documents/CN-A2$ taskset 3 ./server_select
Server is listening on port 8080
Server: New client connected
Server: Adding client ID 1 with socket 4
Server: New client connected
Server: Adding client ID 2 with socket 5
Server: Received request from client ID 1: Requesting CPU process info
Server: Sent CPU process info to client ID 1
Server: New client connected
Server: Adding client ID 3 with socket 6
Server: Received request from client ID 2: Requesting CPU process info
Server: Sent CPU process info to client ID 2
Server: New client connected
Server: Adding client ID 4 with socket 7
Server: Received request from client ID 3: Requesting CPU process info
Server: Sent CPU process info to client ID 3
Server: Received request from client ID 4: Requesting CPU process info
Server: Sent CPU process info to client ID 4
Server: Client ID 1 disconnected
Server: Client ID 2 disconnected
Server: Client ID 4 disconnected
Server: Client ID 3 disconnected
```



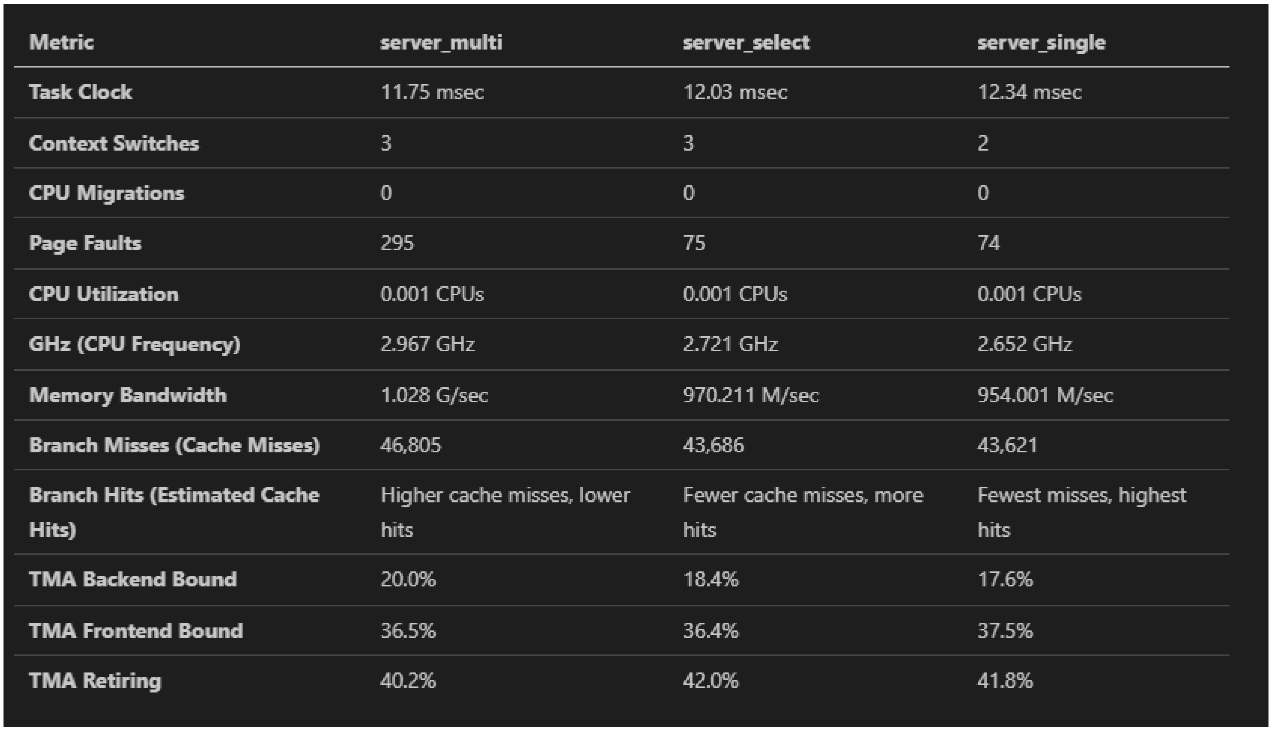
```
(base) iiitd@iiitd-ThinkCentre-M70s-Gen-3:~/Documents/CN-A2$ ps aux | grep server_select
iiitd      48254  0.0  0.0   2776  1440 pts/0    S+   15:57   0:00 ./server_select
iiitd      48335  0.0  0.0   9080  2560 pts/2    S+   15:57   0:00 grep --color=auto server_select
(base) iiitd@iiitd-ThinkCentre-M70s-Gen-3:~/Documents/CN-A2$ sudo perf stat -p 48254 -- sleep 10

 Performance counter stats for process id '48254':

            12.03 msec task-clock                       #    0.001 CPUs utilized
                3      context-switches                 #  249.392 /sec
                0      cpu-migrations                   #    0.000 /sec
               75      page-faults                      #    6.235 K/sec
    <not counted>      cpu_atom/cycles/                                                        (0.00%)
      3,27,28,393      cpu_core/cycles/                 #    2.721 GHz
    <not counted>      cpu_atom/instructions/                                                  (0.00%)
      6,71,47,829      cpu_core/instructions/
    <not counted>      cpu_atom/branches/                                                      (0.00%)
      1,16,70,932      cpu_core/branches/               #  970.211 M/sec
    <not counted>      cpu_atom/branch-misses/                                                 (0.00%)
           43,686      cpu_core/branch-misses/
            TopdownL1 (cpu_core)                 #     18.4 %  tma_backend_bound
                                                 #      3.2 %  tma_bad_speculation
                                                 #     36.4 %  tma_frontend_bound
                                                 #     42.0 %  tma_retiring

      10.002329566 seconds time elapsed

(base) iiitd@iiitd-ThinkCentre-M70s-Gen-3:~/Documents/CN-A2$
```

Client Side:



```
subham22510@Subham-Maurya:/mnt/c/Users/PC/OneDrive/Documents/Computer Networking/A2/CN-A2$ ./client 4
Client: Connected to server
Client: Request sent to server
Client: Connected to server
Client: Request sent to server
Client: Connected to server
Client: Request sent to server
Client: Connected to server
Client: Request sent to server
Client: Server response:
Top 2 CPU-consuming processes:
1. Name: (gnome-shell), PID: 2639, CPU Time: 4619
2. Name: (code), PID: 5675, CPU Time: 4340

Client: Server response:
Top 2 CPU-consuming processes:
1. Name: (gnome-shell), PID: 2639, CPU Time: 4619
2. Name: (code), PID: 5675, CPU Time: 4340

Client: Server response:
Top 2 CPU-consuming processes:
1. Name: (gnome-shell), PID: 2639, CPU Time: 4619
2. Name: (code), PID: 5675, CPU Time: 4340

Client: Server response:
Top 2 CPU-consuming processes:
1. Name: (gnome-shell), PID: 2639, CPU Time: 4619
2. Name: (code), PID: 5675, CPU Time: 4340
```

Comparison Table:



Metric	server_multi	server_select	server_single
Task Clock	11.75 msec	12.03 msec	12.34 msec
Context Switches	3	3	2
CPU Migrations	0	0	0
Page Faults	295	75	74
CPU Utilization	0.001 CPUs	0.001 CPUs	0.001 CPUs
GHz (CPU Frequency)	2.967 GHz	2.721 GHz	2.652 GHz
Memory Bandwidth	1.028 G/sec	970.211 M/sec	954.001 M/sec
Branch Misses (Cache Misses)	46,805	43,686	43,621
Branch Hits (Estimated Cache Hits)	Higher cache misses, lower hits	Fewer cache misses, more hits	Fewest misses, highest hits
TMA Backend Bound	20.0%	18.4%	17.6%
TMA Frontend Bound	36.5%	36.4%	37.5%
TMA Retiring	40.2%	42.0%	41.8%

Detailed Analysis:

1. **Multithreaded Server (server_multi)**: The most resource-heavy server, has higher memory consumption, more page faults, and more branch misses, resulting in higher cache miss rates. It's also more backend-bound, indicating memory or I/O access delays, likely leading to higher task clock and CPU cycles. This server may be better suited for complex, multi-threaded tasks but struggles with memory efficiency.

2. **Select Server (server_select)**: This server strikes a balance between memory usage and CPU performance. With fewer page faults, a lower memory bandwidth, and better cache utilization, it’s more efficient in terms of retiring instructions. This makes ‘server_select’ a good choice for moderate load situations, balancing performance and resource efficiency.

3. **Single-threaded Server (server_single)**: The most lightweight of the three, ‘server_single’ demonstrates efficient memory and cache use with the fewest cache misses and the lowest backend-bound percentage. However, its slightly higher frontend-bound percentage suggests that its single-threaded nature or fewer CPU cycles may cause instruction-fetching delays. It’s ideal for less complex, single-threaded operations but could struggle with more demanding, multi-threaded workloads.

Conclusion:-

- **server_multi** is best for CPU and memory-intensive workloads but suffers from cache misses and backend-bound stalls.
- **server_select** offers a balanced approach with fewer resource demands, good cache hit rates, and better instruction retiring.
- **server_single** is optimal for lightweight, single-threaded tasks, with the best cache performance but slightly more instruction-fetching bottlenecks.