CN Assignment -2

Vedang Rajendrakumar Patel (2022565)
Vedika Agarwal (2022566)

The Assignment involves:

- Developing a client-server application using socket programming in C
- Running client and server on separate Virtual Machines.
- Handling multiple clients concurrently by the server using multithreading
- Processing client requests, retrieving the top two processes consuming most of the CPU
- Analyzing performance using *perf* tools

GitHub: https://github.com/kamperemu/CN-Assignments/tree/main/Assignment2

Question 1:

We need to create two separate files: server.c and client.c.
These files are compiled using make all.
To run the files you use:
taskset -c <cpu>./server and taskset -c <cpu> ./client <number of client connections>

1. **The server sets up a TCP socket and listens for client connections**

   socket(): System call to create a TCP socket on *server side*
   bind(): To bind the socket to the server's IP address and port.
   listen(): Wait for incoming connect requests

2. **The server should be able to handle multiple concurrent clients**

   Multithreading is implemented using pthread library to create a new thread every time the server receives a new connection. This enables us to handle multiple client requests concurrently.

   Upon receiving a connection request, a new socket is created with server IP, server listening port, client IP, client port

3. **The client creates a socket and initiates the TCP connection**

   socket(): System call to create a TCP socket on *client side*
   connect(): Establish connection with the server
   The client is running on multiple connections with the use of pthread. The next 3 steps are run on each of these client connections.

4. **The client sends a request and the server finds top CPU-consuming process**

   The client requests for top CPU processes by sending a message to the server.
   The information about CPU-consuming process is available in /proc directory.
   The specific information is available in **/proc/[pid]/stat**.

We do string manipulation the find the relevant information about the CPU consuming processes on the server.
 We then sort the processes based on total CPU time.

5. **The server sends the information collected to the client**

   After the server gets the information on the top processes it sends a response to the client containing the information of the top 2 processes.
   ```
   send(sock, response, strlen(response), 0);
   ```

6. **The client prints this information and closes the connection**

   The client reads the information sent by the server and prints it. Then closes the connection.
   ```
   int valread = read(sock, buffer, BUFFER_SIZE);
   printf("Client Message received:\n%s\n\n", buffer);
   close(sock);
   ```

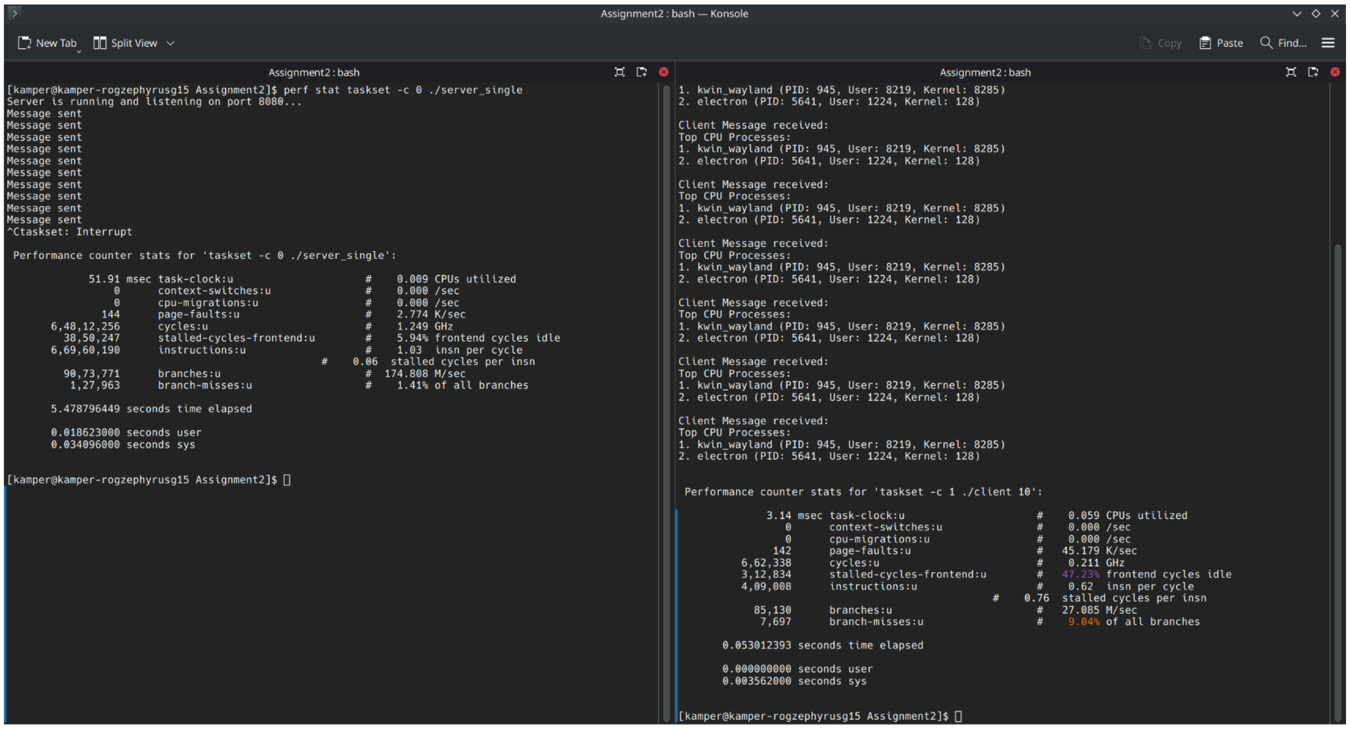
Question 2:

(a) A slightly modified version of server.c was created called server_single.c. This file is identical to server.c except that it removes pthread functionality. (Assumption: clients still use pthread to emulate multiple connections that is multiple machines trying to connect to the server)
(b) The code is same as Q1.
(c) The server.c was modified to use select in a new file called server_select.c

To analyse the data between 3 we use **perf stat** before running the server and the client codes.

The performance metrics that matter in this case are:
- CPU utilization
- Page faults
- Cycles
- Instructions
- Time elapsed on the client

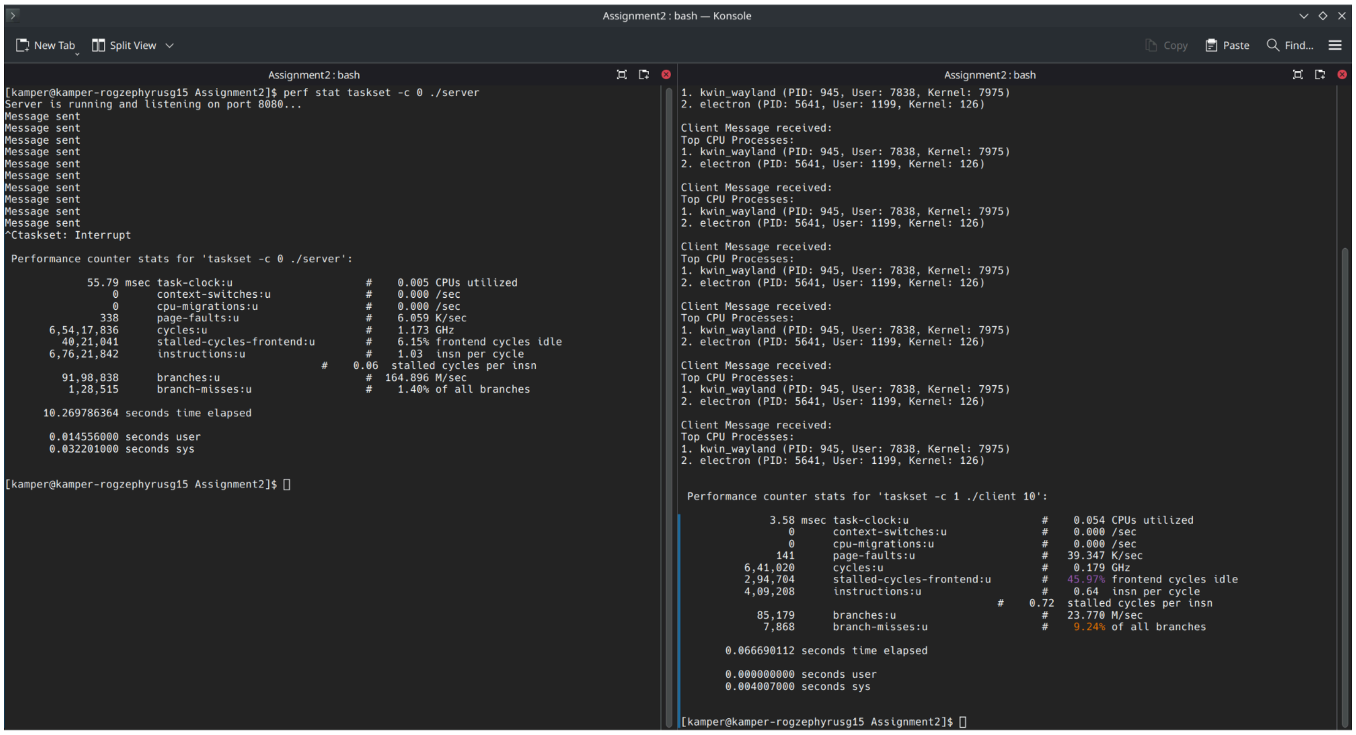
a) Single Threaded TCP Client-Server



```
[kamper@kamper-rogzephyrusg15 Assignment2]$ perf stat taskset -c 0 ./server_single
Server is running and listening on port 8080...
Message sent
Message sent
Message sent
Message sent
Message sent
Message sent
Message sent
Message sent
Message sent
Message sent
^Ctaskset: Interrupt

 Performance counter stats for 'taskset -c 0 ./server_single':

            51.91 msec task-clock:u                     #    0.009 CPUs utilized
                0      context-switches:u               #    0.000 /sec
                0      cpu-migrations:u                 #    0.000 /sec
              144      page-faults:u                    #    2.774 K/sec
      6,48,12,256      cycles:u                         #    1.249 GHz
        38,50,247      stalled-cycles-frontend:u        #    5.94% frontend cycles idle
      6,69,60,190      instructions:u                   #    1.03  insn per cycle
                                                 #    0.06  stalled cycles per insn
        90,73,771      branches:u                       #  174.808 M/sec
         1,27,963      branch-misses:u                  #    1.41% of all branches

      5.478796449 seconds time elapsed

      0.018623000 seconds user
      0.034096000 seconds sys


[kamper@kamper-rogzephyrusg15 Assignment2]$ 
```

```
1. kwin_wayland (PID: 945, User: 8219, Kernel: 8285)
2. electron (PID: 5641, User: 1224, Kernel: 128)

Client Message received:
Top CPU Processes:
1. kwin_wayland (PID: 945, User: 8219, Kernel: 8285)
2. electron (PID: 5641, User: 1224, Kernel: 128)

Client Message received:
Top CPU Processes:
1. kwin_wayland (PID: 945, User: 8219, Kernel: 8285)
2. electron (PID: 5641, User: 1224, Kernel: 128)

Client Message received:
Top CPU Processes:
1. kwin_wayland (PID: 945, User: 8219, Kernel: 8285)
2. electron (PID: 5641, User: 1224, Kernel: 128)

Client Message received:
Top CPU Processes:
1. kwin_wayland (PID: 945, User: 8219, Kernel: 8285)
2. electron (PID: 5641, User: 1224, Kernel: 128)

Client Message received:
Top CPU Processes:
1. kwin_wayland (PID: 945, User: 8219, Kernel: 8285)
2. electron (PID: 5641, User: 1224, Kernel: 128)

Client Message received:
Top CPU Processes:
1. kwin_wayland (PID: 945, User: 8219, Kernel: 8285)
2. electron (PID: 5641, User: 1224, Kernel: 128)


 Performance counter stats for 'taskset -c 1 ./client 10':

             3.14 msec task-clock:u                     #    0.059 CPUs utilized
                0      context-switches:u               #    0.000 /sec
                0      cpu-migrations:u                 #    0.000 /sec
              142      page-faults:u                    #   45.179 K/sec
         6,62,338      cycles:u                         #    0.211 GHz
         3,12,834      stalled-cycles-frontend:u        #   47.23% frontend cycles idle
         4,09,008      instructions:u                   #    0.62  insn per cycle
                                                 #    0.76  stalled cycles per insn
           85,130      branches:u                       #   27.085 M/sec
            7,697      branch-misses:u                  #    9.04% of all branches

      0.053012393 seconds time elapsed

      0.000000000 seconds user
      0.003562000 seconds sys


[kamper@kamper-rogzephyrusg15 Assignment2]$ 
```

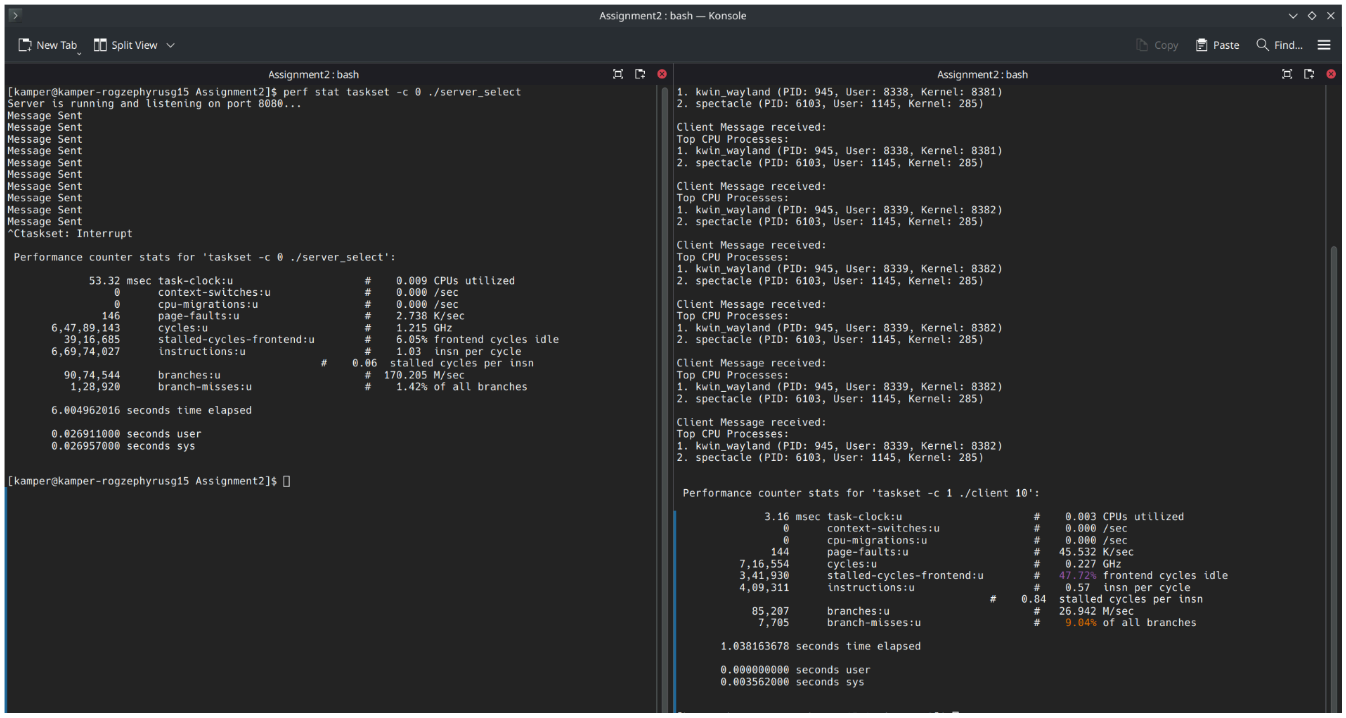
b) Multi Threaded TCP Client-Server



```
[kamper@kamper-rogzephyrusg15 Assignment2]$ perf stat taskset -c 0 ./server
Server is running and listening on port 8080...
Message sent
Message sent
Message sent
Message sent
Message sent
Message sent
Message sent
Message sent
Message sent
Message sent
^Ctaskset: Interrupt

 Performance counter stats for 'taskset -c 0 ./server':

            55.79 msec task-clock:u                     #    0.005 CPUs utilized
                0      context-switches:u               #    0.000 /sec
                0      cpu-migrations:u                 #    0.000 /sec
              338      page-faults:u                    #    6.059 K/sec
      6,54,17,836      cycles:u                         #    1.173 GHz
        40,21,041      stalled-cycles-frontend:u        #    6.15% frontend cycles idle
      6,76,21,842      instructions:u                   #    1.03  insn per cycle
                                                 #    0.06  stalled cycles per insn
        91,98,838      branches:u                       #  164.896 M/sec
         1,28,515      branch-misses:u                  #    1.40% of all branches

     10.269786364 seconds time elapsed

      0.014556000 seconds user
      0.032201000 seconds sys


[kamper@kamper-rogzephyrusg15 Assignment2]$ 
```

```
1. kwin_wayland (PID: 945, User: 7838, Kernel: 7975)
2. electron (PID: 5641, User: 1199, Kernel: 126)

Client Message received:
Top CPU Processes:
1. kwin_wayland (PID: 945, User: 7838, Kernel: 7975)
2. electron (PID: 5641, User: 1199, Kernel: 126)

Client Message received:
Top CPU Processes:
1. kwin_wayland (PID: 945, User: 7838, Kernel: 7975)
2. electron (PID: 5641, User: 1199, Kernel: 126)

Client Message received:
Top CPU Processes:
1. kwin_wayland (PID: 945, User: 7838, Kernel: 7975)
2. electron (PID: 5641, User: 1199, Kernel: 126)

Client Message received:
Top CPU Processes:
1. kwin_wayland (PID: 945, User: 7838, Kernel: 7975)
2. electron (PID: 5641, User: 1199, Kernel: 126)

Client Message received:
Top CPU Processes:
1. kwin_wayland (PID: 945, User: 7838, Kernel: 7975)
2. electron (PID: 5641, User: 1199, Kernel: 126)

Client Message received:
Top CPU Processes:
1. kwin_wayland (PID: 945, User: 7838, Kernel: 7975)
2. electron (PID: 5641, User: 1199, Kernel: 126)


 Performance counter stats for 'taskset -c 1 ./client 10':

             3.58 msec task-clock:u                     #    0.054 CPUs utilized
                0      context-switches:u               #    0.000 /sec
                0      cpu-migrations:u                 #    0.000 /sec
              141      page-faults:u                    #   39.347 K/sec
         6,41,020      cycles:u                         #    0.179 GHz
         2,94,704      stalled-cycles-frontend:u        #   45.97% frontend cycles idle
         4,09,208      instructions:u                   #    0.64  insn per cycle
                                                 #    0.72  stalled cycles per insn
           85,179      branches:u                       #   23.770 M/sec
            7,868      branch-misses:u                  #    9.24% of all branches

      0.066690112 seconds time elapsed

      0.000000000 seconds user
      0.004007000 seconds sys


[kamper@kamper-rogzephyrusg15 Assignment2]$ 
```

c) TCP client-server using “select”



```
[kamper@kamper-rogzephyrusg15 Assignment2]$ perf stat taskset -c 0 ./server_select
Server is running and listening on port 8080...
Message Sent
Message Sent
Message Sent
Message Sent
Message Sent
Message Sent
Message Sent
Message Sent
Message Sent
Message Sent
^Ctaskset: Interrupt

 Performance counter stats for 'taskset -c 0 ./server_select':

             53.32 msec task-clock:u                     #    0.009 CPUs utilized
                 0      context-switches:u               #    0.000 /sec
                 0      cpu-migrations:u                 #    0.000 /sec
               146      page-faults:u                    #    2.738 K/sec
       6,47,89,143      cycles:u                         #    1.215 GHz
         39,16,685      stalled-cycles-frontend:u        #    6.05% frontend cycles idle
       6,69,74,027      instructions:u                   #    1.03  insn per cycle
                                                  #    0.06  stalled cycles per insn
         90,74,544      branches:u                       #  170.205 M/sec
          1,28,920      branch-misses:u                  #    1.42% of all branches

       6.004962016 seconds time elapsed

       0.026911000 seconds user
       0.026957000 seconds sys


[kamper@kamper-rogzephyrusg15 Assignment2]$ 
```

```
1. kwin_wayland (PID: 945, User: 8338, Kernel: 8381)
2. spectacle (PID: 6103, User: 1145, Kernel: 285)

Client Message received:
Top CPU Processes:
1. kwin_wayland (PID: 945, User: 8338, Kernel: 8381)
2. spectacle (PID: 6103, User: 1145, Kernel: 285)

Client Message received:
Top CPU Processes:
1. kwin_wayland (PID: 945, User: 8339, Kernel: 8382)
2. spectacle (PID: 6103, User: 1145, Kernel: 285)

Client Message received:
Top CPU Processes:
1. kwin_wayland (PID: 945, User: 8339, Kernel: 8382)
2. spectacle (PID: 6103, User: 1145, Kernel: 285)

Client Message received:
Top CPU Processes:
1. kwin_wayland (PID: 945, User: 8339, Kernel: 8382)
2. spectacle (PID: 6103, User: 1145, Kernel: 285)

Client Message received:
Top CPU Processes:
1. kwin_wayland (PID: 945, User: 8339, Kernel: 8382)
2. spectacle (PID: 6103, User: 1145, Kernel: 285)

Client Message received:
Top CPU Processes:
1. kwin_wayland (PID: 945, User: 8339, Kernel: 8382)
2. spectacle (PID: 6103, User: 1145, Kernel: 285)


 Performance counter stats for 'taskset -c 1 ./client 10':

              3.16 msec task-clock:u                     #    0.003 CPUs utilized
                 0      context-switches:u               #    0.000 /sec
                 0      cpu-migrations:u                 #    0.000 /sec
               144      page-faults:u                    #   45.532 K/sec
         7,16,554      cycles:u                          #    0.227 GHz
         3,41,930      stalled-cycles-frontend:u         #   47.72% frontend cycles idle
         4,09,311      instructions:u                    #    0.57  insn per cycle
                                                  #    0.84  stalled cycles per insn
           85,207      branches:u                        #   26.942 M/sec
            7,705      branch-misses:u                   #    9.04% of all branches

       1.038163678 seconds time elapsed

       0.000000000 seconds user
       0.003562000 seconds sys
```

Performance Observations made:

- Multithreading server requires less CPU utilization
- Multithreading server uses more page faults.
- The time elapsed in the clients in case of using select is by far the longest.
- Not much performance information can be extracted from the information from other statistics as they aren’t very different.