CSE 344 Systems Programming Final Report

1801042609

Yakup Talha Yolcu

Client

```
./client -r requestFile -q PORT -s I
```

in client size, I first check input validity then set the signal stuff, then take arguments and assign them to needed numbers and strings. After that, I read the file to get number of lines in the file. Then I close file and reopen it. Then I take characters byte by byte. After that, I initialize my condition variable and mutex to get sync barrier. I create threads and join on them.

After joining, I destroy mutex and cond var and free resources and exit.
My client threads have these sync barrier

```
pthread_mutex_lock(&mutex);
if(terminate_flag==1) {
    pthread_mutex_unlock(&mutex);
}
++arrived;
//synchronization barrier
while(arrived<number_of_threads) {
    if(terminate_flag==1) {
        break;
    }
    pthread_cond_wait(&condition,&mutex);
    if(terminate_flag==1) {
        break;
    }
}
pthread_cond_broadcast(&condition);
pthread_mutex_unlock(&mutex);
//after barrier
```

After threads passed this barrier, It connects to socket to communicate server
Then it sends a request and waits for response
After taking response it simple returns NULL.

Server Side

```
./server -p 33000 -t 11 &
```

Server side firstly checks input validity and initialized the condition variables and mutexes.
I have 1 cond var and 2 mutex.
Cond var and first mutex is to wake up threads, other mutex is needed for servant counter integer which keeps the number of servants server knows.
I have queue which keeps the return value of accepts.

```
typedef struct Queue {
   int front, rear, size;
   unsigned capacity;
   int* array;
}Queue;
```

After socket settings, I have a while loop that accepts the incoming connections.

```
   while(1) {
        if(terminate_flag==1) {
            pthread_cond_broadcast(&cond);
            break;
        }
        client_fd=accept(server_fd,(struct
sockaddr*)&client_addr,(socklen_t*)&client_addr_size);
        //printf("NEW ACCEPT\n");
        if(terminate_flag==1) {
            pthread_cond_broadcast(&cond);
            break;
        }
        if(client_fd==-1) {
            perror("Error while accepting");
            break;
        }
        if(terminate_flag==1) {
            pthread_cond_broadcast(&cond);
            break;
        }
        enqueue(my_q,client_fd);
        client_fd=0;
        pthread_cond_signal(&cond);
   }
```

After this while loop, I join on threads and free resources.

Threads are in the infinite loop to handle incoming connections. They also wait here:

```
pthread_mutex_lock(&mutex);
while(terminate_flag==0 && my_q->size<1) {
    pthread_cond_wait(&cond,&mutex);
}
```

If something is added to queue, then They continue.
My queue adding function solves the race condition problem like that:

```
pthread_mutex_lock(&mutex);
queue->rear = (queue->rear + 1)
                % queue->capacity;
queue->array[queue->rear] = item;
queue->size = queue->size + 1;
//printf("%d enqueued to queue\n", item);
pthread_mutex_unlock(&mutex);
```

```
        int request_fd=dequeue(my_q);
```
My thread gets the socket fd from the queue.
After that it reads the request and decide it is from client or servant.
If it is client request, it forwards it to appropriate servant if city is given, otherwise it contacts all the servants. After getting response from servant(s), it forwards it to the client.

If it is servant lets know, then it saves them to an array, takes ip address, pid, port number, start city, end city.

After thread is done with sending response or taking client fd, then it closes the corresponding socket and continues while loop

Servant side :
```
./servant -d directoryPath -c 10-19 -r IP -p PORT
```
It firstly checks input validity and sets signal stuff. Then it reads stat file to get pid.
```
    int stat_fd=open("/proc/self/stat",(O_RDONLY),0777);
    if(stat_fd==-1) {
        perror("Error on open statfd");
        return 0;
    }
    int pid=0;

    char buffer[MAX_BLKSIZE];
    memset(buffer,0,MAX_BLKSIZE);

    if(read(stat_fd,buffer,MAX_BLKSIZE)==-1) {
        perror("Error on read");
        return 0;
    }
    char*buf=buffer;
    char* pid_char=strtok_r(buf," ",&buf);
    pid=atoi(pid_char);

    close(stat_fd);
```
After that it scans the directories and saves them into Binary Search tree.
After saving them, it closes directories and start connection to server.
First it sends needed info such as ip,port,pid,start city and end city.
Then it closes that connection and listens from the given port.

It waits on that infinite loop:

```
while(1) {
    int new_socket=accept(server_fd,(struct sockaddr*)&address,(socklen_t*)&addrlen);
    if(terminate_flag==1) {
        break;
    }
    if(new_socket<0) {
        perror("Error on accept");
        break;
    }
    pthread_t thread;
    int*index=(int*)malloc(sizeof(int));
    *index=new_socket;
    if(pthread_create(&thread,NULL,run_thread,(void*)index)!=0) {
        perror("Error on pthread create");
        break;
    }
    pthread_detach(thread);
    if(terminate_flag==1) {
        break;
    }
}
```

Servant threads are waiting on receive, it take a request and starts to search on BST tree. After search is done, it sends the response.

BST three:

```
struct node {
   file_entry *key;
   struct node *left, *right;
};
```

Why I used BST tree?
Because we search for a given date interval, by looking intervals I can eliminate the entries with wrong dates.

```
typedef struct file_entry {
   int transaction_id;
   char* type_real_estate;
   char* name_of_street;
   int surface;
   int price;
   char* date;
   char* city_name;
}file_entry;
```

Simple file entry is like that.

```
void take_date(char*date1,int*d,int*m,int*y)
```

Takes date and assigns day,month and year values.

```
int compare_dates(char*date1,char*date2)
```

Compares the given dates, if date1>date2 returns 1, it they are equal returns 0, otherwise -1
It calls take_date function.

```
struct node* newNode(file_entry* item)
```

Adds a new node to BST tree

```c
struct node* insert(struct node* node, file_entry* key)
{
    /* If the tree is empty, return a new node */
    if (node == NULL)
        return newNode(key);

    /* Otherwise, recur down the tree */
    if (compare_dates(node->key->date,key->date)<=0)
        node->left = insert(node->left, key);
    else if (compare_dates(node->key->date,key->date)>0)
        node->right = insert(node->right, key);

    /* return the (unchanged) node pointer */
    return node;
}
```

Inserts an item. It decides where to put by looking the right and left dates.

```c
int find_with_city(struct node* node,char*t_type,char*
date1,char*date2,char*city)
```

Searchs with dates and city name

```c
int find_without_city(struct node* node,char*t_type,char*
date1,char*date2)
```

Searchs with dates.

Client output:

```
==83996== Memcheck, a memory error detector
==83996== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==83996== Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==83996== Command: ./client -r ./requestFile -q 4756 -s 127.0.0.1
==83996==
Client:I have loaded 10 requests and I'm creating 10 threads
Client-Thread-5: Thread-5 has been created
Client-Thread-1: Thread-1 has been created
Client-Thread-6: Thread-6 has been created
Client-Thread-0: Thread-0 has been created
Client-Thread-3: Thread-3 has been created
Client-Thread-7: Thread-7 has been created
Client-Thread-8: Thread-8 has been created
Client-Thread-4: Thread-4 has been created
Client-Thread-2: Thread-2 has been created
Client-Thread-9: Thread-9 has been created
Client-Thread-9: I am requesting "transactionCount IMALATHANE 04-06-2004 11-11-2011 ISPARTA"
Client-Thread-6: I am requesting "transactionCount FABRIKA 22-07-2004 11-05-2072 ANKARA"
Client-Thread-7: I am requesting "transactionCount AMBAR 28-01-2044 13-09-2050 AKSARAY"
Client-Thread-5: I am requesting "transactionCount BAHCE 02-03-2005 17-01-2084"
Client-Thread-4: I am requesting "transactionCount FIDANLIK 02-09-2016 12-09-2081 BALIKESIR"
Client-Thread-8: I am requesting "transactionCount VILLA 22-04-2049 20-03-2061"
Client-Thread-1: I am requesting "transactionCount MERA 03-02-2018 09-11-2050"
Client-Thread-2: I am requesting "transactionCount DUKKAN 20-04-2000 23-01-2031 KILIS"
Client-Thread-3: I am requesting "transactionCount BAG 01-12-2004 27-09-2089 ADIYAMAN"
Client-Thread-0: I am requesting "transactionCount TARLA 01-01-2073 30-12-2074 ADANA"
Client-Thread-9: The server's response to "transactionCount IMALATHANE 04-06-2004 11-11-2011 ISPARTA" is 4
Client-Thread-9: Terminating
Client-Thread-6: The server's response to "transactionCount FABRIKA 22-07-2004 11-05-2072 ANKARA" is 4
Client-Thread-6: Terminating
Client-Thread-4: The server's response to "transactionCount FIDANLIK 02-09-2016 12-09-2081 BALIKESIR" is 3
Client-Thread-4: Terminating
Client-Thread-2: The server's response to "transactionCount DUKKAN 20-04-2000 23-01-2031 KILIS" is 1
Client-Thread-2: Terminating
Client-Thread-3: The server's response to "transactionCount BAG 01-12-2004 27-09-2089 ADIYAMAN" is 5
Client-Thread-3: Terminating
Client-Thread-7: The server's response to "transactionCount AMBAR 28-01-2044 13-09-2050 AKSARAY" is 0
Client-Thread-7: Terminating
Client-Thread-0: The server's response to "transactionCount TARLA 01-01-2073 30-12-2074 ADANA" is 3
Client-Thread-0: Terminating
Client-Thread-8: The server's response to "transactionCount VILLA 22-04-2049 20-03-2061" is 29
Client-Thread-8: Terminating
Client-Thread-5: The server's response to "transactionCount BAHCE 02-03-2005 17-01-2084" is 343
Client-Thread-5: Terminating
Client-Thread-1: The server's response to "transactionCount MERA 03-02-2018 09-11-2050" is 158
Client-Thread-1: Terminating
Client: All threads have terminated, goodbye.
```

```
==83996==
==83996== HEAP SUMMARY:
==83996==     in use at exit: 0 bytes in 0 blocks
==83996==   total heap usage: 43 allocs, 43 frees, 23,328 bytes allocated
==83996==
==83996== All heap blocks were freed -- no leaks are possible
==83996==
==83996== For lists of detected and suppressed errors, rerun with: -s
==83996== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

Server output:

```
==83906== Memcheck, a memory error detector
==83906== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==83906== Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==83906== Command: ./server -p 4756 -t 11
==83906==
Incoming:SERVANT 83934 16076 127.0.0.1 TEKIRDAG ZONGULDAK
Incoming:SERVANT 83929 16031 127.0.0.1 EDIRNE HAKKARI
Incoming:SERVANT 83928 16022 127.0.0.1 BOLU DUZCE
Incoming:SERVANT 83930 16040 127.0.0.1 HATAY KARS
Incoming:SERVANT 83927 16013 127.0.0.1 ARTVIN BITLIS
Incoming:SERVANT 83932 16058 127.0.0.1 MALATYA ORDU
Incoming:SERVANT 83926 16004 127.0.0.1 ADANA ARDAHAN
Incoming:SERVANT 83933 16067 127.0.0.1 OSMANIYE SIVAS
Incoming:SERVANT 83931 16049 127.0.0.1 KASTAMONU KUTAHYA
Incoming:CLIENT transactionCount IMALATHANE 04-06-2004 11-11-2011 ISPARTA
Incoming:CLIENT transactionCount VILLA 22-04-2049 20-03-2061
Incoming:CLIENT transactionCount FABRIKA 22-07-2004 11-05-2072 ANKARA
Incoming:CLIENT transactionCount BAG 01-12-2004 27-09-2089 ADIYAMAN
Incoming:CLIENT transactionCount AMBAR 28-01-2044 13-09-2050 AKSARAY
Incoming:CLIENT transactionCount TARLA 01-01-2073 30-12-2074 ADANA
Incoming:CLIENT transactionCount MERA 03-02-2018 09-11-2050
Incoming:CLIENT transactionCount DUKKAN 20-04-2000 23-01-2031 KILIS
Incoming:CLIENT transactionCount BAHCE 02-03-2005 17-01-2084
Incoming:CLIENT transactionCount FIDANLIK 02-09-2016 12-09-2081 BALIKESIR
Thu Jun 16 04:19:45 2022 => Servant 83929 present at port 16031 handling cities EDIRNE-HAKKARI
Thu Jun 16 04:19:45 2022 => Servant 83934 present at port 16076 handling cities TEKIRDAG-ZONGULDAK
Thu Jun 16 04:19:45 2022 => Servant 83928 present at port 16022 handling cities BOLU-DUZCE
Thu Jun 16 04:19:45 2022 => Servant 83930 present at port 16040 handling cities HATAY-KARS
Thu Jun 16 04:19:45 2022 => Servant 83927 present at port 16013 handling cities ARTVIN-BITLIS
Thu Jun 16 04:19:45 2022 => Servant 83932 present at port 16058 handling cities MALATYA-ORDU
Thu Jun 16 04:19:45 2022 => Servant 83926 present at port 16004 handling cities ADANA-ARDAHAN
Thu Jun 16 04:19:45 2022 => Servant 83933 present at port 16067 handling cities OSMANIYE-SIVAS
Thu Jun 16 04:19:45 2022 => Servant 83931 present at port 16049 handling cities KASTAMONU-KUTAHYA
```

```
Incoming:CLIENT transactionCount FIDANLIK 02-09-2016 12-09-2081 BALIKESIR
Thu Jun 16 04:19:45 2022 => Servant 83929 present at port 16031 handling cities EDIRNE-HAKKARI
Thu Jun 16 04:19:45 2022 => Servant 83934 present at port 16076 handling cities TEKIRDAG-ZONGULDAK
Thu Jun 16 04:19:45 2022 => Servant 83928 present at port 16022 handling cities BOLU-DUZCE
Thu Jun 16 04:19:45 2022 => Servant 83930 present at port 16040 handling cities HATAY-KARS
Thu Jun 16 04:19:45 2022 => Servant 83927 present at port 16013 handling cities ARTVIN-BITLIS
Thu Jun 16 04:19:45 2022 => Servant 83932 present at port 16058 handling cities MALATYA-ORDU
Thu Jun 16 04:19:45 2022 => Servant 83926 present at port 16004 handling cities ADANA-ARDAHAN
Thu Jun 16 04:19:45 2022 => Servant 83933 present at port 16067 handling cities OSMANIYE-SIVAS
Thu Jun 16 04:19:45 2022 => Servant 83931 present at port 16049 handling cities KASTAMONU-KUTAHYA
Thu Jun 16 04:19:46 2022 => Contacting servant 83930
Thu Jun 16 04:19:46 2022 => Request arrived "transactionCount IMALATHANE 04-06-2004 11-11-2011 ISPARTA"
Thu Jun 16 04:19:46 2022 => Contacting ALL servants
Thu Jun 16 04:19:46 2022 => Request arrived "transactionCount VILLA 22-04-2049 20-03-2061"
Thu Jun 16 04:19:46 2022 => Contacting servant 83926
Thu Jun 16 04:19:46 2022 => Request arrived "transactionCount FABRIKA 22-07-2004 11-05-2072 ANKARA"
Thu Jun 16 04:19:46 2022 => Contacting servant 83926
Thu Jun 16 04:19:46 2022 => Request arrived "transactionCount BAG 01-12-2004 27-09-2089 ADIYAMAN"
Thu Jun 16 04:19:46 2022 => Contacting servant 83926
Thu Jun 16 04:19:46 2022 => Request arrived "transactionCount AMBAR 28-01-2044 13-09-2050 AKSARAY"
Thu Jun 16 04:19:46 2022 => Contacting servant 83926
Thu Jun 16 04:19:46 2022 => Request arrived "transactionCount TARLA 01-01-2073 30-12-2074 ADANA"
Thu Jun 16 04:19:46 2022 => Contacting servant 83931
Thu Jun 16 04:19:46 2022 => Request arrived "transactionCount DUKKAN 20-04-2000 23-01-2031 KILIS"
Thu Jun 16 04:19:46 2022 => Contacting ALL servants
Thu Jun 16 04:19:46 2022 => Request arrived "transactionCount MERA 03-02-2018 09-11-2050"
Thu Jun 16 04:19:46 2022 => Contacting ALL servants
Thu Jun 16 04:19:46 2022 => Request arrived "transactionCount BAHCE 02-03-2005 17-01-2084"
Thu Jun 16 04:19:46 2022 => Contacting servant 83927
Thu Jun 16 04:19:46 2022 => Request arrived "transactionCount FIDANLIK 02-09-2016 12-09-2081 BALIKESIR"
Thu Jun 16 04:19:46 2022 => Response received: 4, forwarded to client
Thu Jun 16 04:19:46 2022 => Response received: 4, forwarded to client
Thu Jun 16 04:19:46 2022 => Response received: 3, forwarded to client
Thu Jun 16 04:19:47 2022 => Response received: 1, forwarded to client
Thu Jun 16 04:19:47 2022 => Request arrived "transactionCount VILLA 22-04-2049 20-03-2061"
Thu Jun 16 04:19:47 2022 => Response received: 5, forwarded to client
Thu Jun 16 04:19:47 2022 => Response received: 0, forwarded to client
Thu Jun 16 04:19:47 2022 => Response received: 3, forwarded to client
Thu Jun 16 04:19:47 2022 => Request arrived "transactionCount MERA 03-02-2018 09-11-2050"
Thu Jun 16 04:19:47 2022 => Request arrived "transactionCount VILLA 22-04-2049 20-03-2061"
Thu Jun 16 04:19:47 2022 => Request arrived "transactionCount BAHCE 02-03-2005 17-01-2084"
Thu Jun 16 04:19:47 2022 => Request arrived "transactionCount MERA 03-02-2018 09-11-2050"
Thu Jun 16 04:19:47 2022 => Request arrived "transactionCount VILLA 22-04-2049 20-03-2061"
Thu Jun 16 04:19:47 2022 => Request arrived "transactionCount BAHCE 02-03-2005 17-01-2084"
Thu Jun 16 04:19:47 2022 => Request arrived "transactionCount VILLA 22-04-2049 20-03-2061"
Thu Jun 16 04:19:47 2022 => Request arrived "transactionCount MERA 03-02-2018 09-11-2050"
Thu Jun 16 04:19:47 2022 => Request arrived "transactionCount MERA 03-02-2018 09-11-2050"
Thu Jun 16 04:19:47 2022 => Request arrived "transactionCount VILLA 22-04-2049 20-03-2061"
Thu Jun 16 04:19:47 2022 => Request arrived "transactionCount BAHCE 02-03-2005 17-01-2084"
```

```
Thu Jun 16 04:19:46 2022 => Contacting ALL servants
Thu Jun 16 04:19:46 2022 => Request arrived "transactionCount MERA 03-02-2018 09-11-2050"
Thu Jun 16 04:19:46 2022 => Contacting ALL servants
Thu Jun 16 04:19:46 2022 => Request arrived "transactionCount BAHCE 02-03-2005 17-01-2084"
Thu Jun 16 04:19:46 2022 => Contacting servant 83927
Thu Jun 16 04:19:46 2022 => Request arrived "transactionCount FIDANLIK 02-09-2016 12-09-2081 BALIKESIR"
Thu Jun 16 04:19:46 2022 => Response received: 4, forwarded to client
Thu Jun 16 04:19:46 2022 => Response received: 4, forwarded to client
Thu Jun 16 04:19:46 2022 => Response received: 3, forwarded to client
Thu Jun 16 04:19:47 2022 => Response received: 1, forwarded to client
Thu Jun 16 04:19:47 2022 => Request arrived "transactionCount VILLA 22-04-2049 20-03-2061"
Thu Jun 16 04:19:47 2022 => Response received: 5, forwarded to client
Thu Jun 16 04:19:47 2022 => Response received: 0, forwarded to client
Thu Jun 16 04:19:47 2022 => Response received: 3, forwarded to client
Thu Jun 16 04:19:47 2022 => Request arrived "transactionCount MERA 03-02-2018 09-11-2050"
Thu Jun 16 04:19:47 2022 => Request arrived "transactionCount VILLA 22-04-2049 20-03-2061"
Thu Jun 16 04:19:47 2022 => Request arrived "transactionCount BAHCE 02-03-2005 17-01-2084"
Thu Jun 16 04:19:47 2022 => Request arrived "transactionCount MERA 03-02-2018 09-11-2050"
Thu Jun 16 04:19:47 2022 => Request arrived "transactionCount VILLA 22-04-2049 20-03-2061"
Thu Jun 16 04:19:47 2022 => Request arrived "transactionCount BAHCE 02-03-2005 17-01-2084"
Thu Jun 16 04:19:47 2022 => Request arrived "transactionCount VILLA 22-04-2049 20-03-2061"
Thu Jun 16 04:19:47 2022 => Request arrived "transactionCount MERA 03-02-2018 09-11-2050"
Thu Jun 16 04:19:47 2022 => Request arrived "transactionCount MERA 03-02-2018 09-11-2050"
Thu Jun 16 04:19:47 2022 => Request arrived "transactionCount VILLA 22-04-2049 20-03-2061"
Thu Jun 16 04:19:47 2022 => Request arrived "transactionCount BAHCE 02-03-2005 17-01-2084"
Thu Jun 16 04:19:47 2022 => Request arrived "transactionCount MERA 03-02-2018 09-11-2050"
Thu Jun 16 04:19:47 2022 => Request arrived "transactionCount BAHCE 02-03-2005
```

Before sending SIGINT :

```
talha@talha-X580VD:~/Masaüstü/CSE344 Systems Programming/final/1801042609$ ps -aux | grep server
talha       2100  0.0  0.1 581500 20240 ?        Sl   Haz15   0:00 /usr/libexec/gnome-shell-calendar-server
talha       2198  0.0  0.3 643748 57480 ?        Sl   Haz15   0:00 /usr/libexec/evolution-data-server/evolution-alarm-not
ify
talha      22942  0.2  0.3 825620 59696 ?        Ssl  Haz15   0:50 /usr/libexec/gnome-terminal-server
talha      83906  0.5  1.0 312284 177304 pts/0   Sl   04:19   0:01 /usr/bin/valgrind.bin --leak-check=full --leak-check=y
es --show-leak-kinds=all --track-origins=yes --track-fds=yes ./server -p 4756 -t 11
talha      84707  0.0  0.0  11792  2604 pts/1    S+   04:24   0:00 grep --color=auto server
talha@talha-X580VD:~/Masaüstü/CSE344 Systems Programming/final/1801042609$ ps -aux | grep servant
talha      83926  0.8  0.9 222428 155548 pts/0   S    04:19   0:02 /usr/bin/valgrind.bin --leak-check=full --leak-check=y
es --show-leak-kinds=all --track-origins=yes --track-fds=yes ./servant -d ./dataset -c 1-9 -r 127.0.0.1 -p 4756
talha      83927  0.8  0.9 211144 152372 pts/0   S    04:19   0:02 /usr/bin/valgrind.bin --leak-check=full --leak-check=y
es --show-leak-kinds=all --track-origins=yes --track-fds=yes ./servant -d ./dataset -c 10-18 -r 127.0.0.1 -p 4756
talha      83928  0.8  0.9 211144 152360 pts/0   S    04:19   0:02 /usr/bin/valgrind.bin --leak-check=full --leak-check=y
es --show-leak-kinds=all --track-origins=yes --track-fds=yes ./servant -d ./dataset -c 19-27 -r 127.0.0.1 -p 4756
talha      83929  0.8  0.9 211144 152348 pts/0   S    04:19   0:02 /usr/bin/valgrind.bin --leak-check=full --leak-check=y
es --show-leak-kinds=all --track-origins=yes --track-fds=yes ./servant -d ./dataset -c 28-36 -r 127.0.0.1 -p 4756
talha      83930  0.8  0.9 199860 149204 pts/0   S    04:19   0:02 /usr/bin/valgrind.bin --leak-check=full --leak-check=y
es --show-leak-kinds=all --track-origins=yes --track-fds=yes ./servant -d ./dataset -c 37-45 -r 127.0.0.1 -p 4756
talha      83931  0.8  0.9 199860 149212 pts/0   S    04:19   0:02 /usr/bin/valgrind.bin --leak-check=full --leak-check=y
es --show-leak-kinds=all --track-origins=yes --track-fds=yes ./servant -d ./dataset -c 46-54 -r 127.0.0.1 -p 4756
talha      83932  0.9  0.9 211144 152352 pts/0   S    04:19   0:02 /usr/bin/valgrind.bin --leak-check=full --leak-check=y
es --show-leak-kinds=all --track-origins=yes --track-fds=yes ./servant -d ./dataset -c 55-63 -r 127.0.0.1 -p 4756
talha      83933  0.8  0.9 211144 152348 pts/0   S    04:19   0:02 /usr/bin/valgrind.bin --leak-check=full --leak-check=y
es --show-leak-kinds=all --track-origins=yes --track-fds=yes ./servant -d ./dataset -c 64-72 -r 127.0.0.1 -p 4756
talha      83934  0.8  0.9 211144 152344 pts/0   S    04:19   0:02 /usr/bin/valgrind.bin --leak-check=full --leak-check=y
es --show-leak-kinds=all --track-origins=yes --track-fds=yes ./servant -d ./dataset -c 73-81 -r 127.0.0.1 -p 4756
talha      84711  0.0  0.0  11792  2548 pts/1    S+   04:24   0:00 grep --color=auto servant
talha@talha-X580VD:~/Masaüstü/CSE344 Systems Programming/final/1801042609$
```

After sending SIGINT:

```
talha@talha-X580VD:~/Masaüstü/CSE344 Systems Programming/final/1801042609$ ps -aux | grep servant
talha      83926  0.8  0.9 222428 155548 pts/0   S    04:19   0:02 /usr/bin/valgrind.bin --leak-check=full --leak-check=y
es --show-leak-kinds=all --track-origins=yes --track-fds=yes ./servant -d ./dataset -c 1-9 -r 127.0.0.1 -p 4756
talha      83927  0.8  0.9 211144 152372 pts/0   S    04:19   0:02 /usr/bin/valgrind.bin --leak-check=full --leak-check=y
es --show-leak-kinds=all --track-origins=yes --track-fds=yes ./servant -d ./dataset -c 10-18 -r 127.0.0.1 -p 4756
talha      83928  0.8  0.9 211144 152360 pts/0   S    04:19   0:02 /usr/bin/valgrind.bin --leak-check=full --leak-check=y
es --show-leak-kinds=all --track-origins=yes --track-fds=yes ./servant -d ./dataset -c 19-27 -r 127.0.0.1 -p 4756
talha      83929  0.8  0.9 211144 152348 pts/0   S    04:19   0:02 /usr/bin/valgrind.bin --leak-check=full --leak-check=y
es --show-leak-kinds=all --track-origins=yes --track-fds=yes ./servant -d ./dataset -c 28-36 -r 127.0.0.1 -p 4756
talha      83930  0.8  0.9 199860 149204 pts/0   S    04:19   0:02 /usr/bin/valgrind.bin --leak-check=full --leak-check=y
es --show-leak-kinds=all --track-origins=yes --track-fds=yes ./servant -d ./dataset -c 37-45 -r 127.0.0.1 -p 4756
talha      83931  0.8  0.9 199860 149212 pts/0   S    04:19   0:02 /usr/bin/valgrind.bin --leak-check=full --leak-check=y
es --show-leak-kinds=all --track-origins=yes --track-fds=yes ./servant -d ./dataset -c 46-54 -r 127.0.0.1 -p 4756
talha      83932  0.9  0.9 211144 152352 pts/0   S    04:19   0:02 /usr/bin/valgrind.bin --leak-check=full --leak-check=y
es --show-leak-kinds=all --track-origins=yes --track-fds=yes ./servant -d ./dataset -c 55-63 -r 127.0.0.1 -p 4756
talha      83933  0.8  0.9 211144 152348 pts/0   S    04:19   0:02 /usr/bin/valgrind.bin --leak-check=full --leak-check=y
es --show-leak-kinds=all --track-origins=yes --track-fds=yes ./servant -d ./dataset -c 64-72 -r 127.0.0.1 -p 4756
talha      83934  0.8  0.9 211144 152344 pts/0   S    04:19   0:02 /usr/bin/valgrind.bin --leak-check=full --leak-check=y
es --show-leak-kinds=all --track-origins=yes --track-fds=yes ./servant -d ./dataset -c 73-81 -r 127.0.0.1 -p 4756
talha      84711  0.0  0.0  11792  2548 pts/1    S+   04:24   0:00 grep --color=auto servant
talha@talha-X580VD:~/Masaüstü/CSE344 Systems Programming/final/1801042609$ kill -2 83906
talha@talha-X580VD:~/Masaüstü/CSE344 Systems Programming/final/1801042609$ ps -aux | grep servant
talha      84830  0.0  0.0  11660  2548 pts/1    S+   04:24   0:00 grep --color=auto servant
talha@talha-X580VD:~/Masaüstü/CSE344 Systems Programming/final/1801042609$ ps -aux | grep server
talha       2100  0.0  0.1 581500 20240 ?        Sl   Haz15   0:00 /usr/libexec/gnome-shell-calendar-server
talha       2198  0.0  0.3 643748 57480 ?        Sl   Haz15   0:00 /usr/libexec/evolution-data-server/evolution-alarm-not
ify
talha      22942  0.2  0.3 825788 60016 ?        Ssl  Haz15   0:51 /usr/libexec/gnome-terminal-server
talha      84832  0.0  0.0  11792  2556 pts/1    S+   04:24   0:00 grep --color=auto server
talha@talha-X580VD:~/Masaüstü/CSE344 Systems Programming/final/1801042609$
```

Servant output example :

```
==83926== Memcheck, a memory error detector
==83926== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==83926== Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==83926== Command: ./servant -d ./dataset -c 1-9 -r 127.0.0.1 -p 4756
==83926==
Servant 83926: loaded dataset, cities ADANA-ARDAHAN
Servant sent:SERVANT 83926 16004 127.0.0.1 ADANA ARDAHAN
Servant 83926: listening at port 16004
Servant 83926: termination message received, handled 7 requests in total.
==83926==
```

Server output continues :

```
Thu Jun 16 04:19:47 2022 => Request arrived "transactionCount VILLA 22-04-2049 20-03-2061"
Thu Jun 16 04:19:47 2022 => Request arrived "transactionCount MERA 03-02-2018 09-11-2050"
Response received: 29, forwarded to client
Thu Jun 16 04:19:47 2022 => Request arrived "transactionCount BAHCE 02-03-2005 17-01-2084"
Response received: 343, forwarded to client
Thu Jun 16 04:19:47 2022 => Request arrived "transactionCount MERA 03-02-2018 09-11-2050"
Response received: 158, forwarded to client
Thu Jun 16 04:24:42 2022 => SIGINT has been received. I handled a total of 10 requests. Goodbye.
==83906==
```

```
==83906==
==83906== HEAP SUMMARY:
==83906==     in use at exit: 0 bytes in 0 blocks
==83906==   total heap usage: 290 allocs, 290 frees, 115,750 bytes allocated
==83906==
==83906== All heap blocks were freed -- no leaks are possible
==83906==
==83906== For lists of detected and suppressed errors, rerun with: -s
==83906== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

I achieved these requierements:
Server-client-servants can connect each other which can be seen in outputs. But after each run, you need to wait at least 2 minutes to cleaning of the port.