

Part 2

Generated by Doxygen 1.9.8

1 DOCS-DB	1
2 File Index	3
2.1 File List	3
3 File Documentation	5
3.1 tcp_client.cpp File Reference	5
3.2 tcp_server.cpp File Reference	5
3.2.1 Detailed Description	6
3.2.2 Function Documentation	6
3.2.2.1 get_cpu_usage()	6
3.2.2.2 get_memory_usage()	7
3.2.2.3 main()	7
3.2.2.4 monitor_resource_usage()	7
Index	9

Chapter 1

DOCS-DB

DOCS DB is a key-value based high performance database.

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

tcp_client.cpp	This is a simple TCP client that sends random data to the server and receives a response . . .	5
tcp_server.cpp	This is a simple TCP server that listens for client connections and responds with the number of bytes received	5

Chapter 3

File Documentation

3.1 tcp_client.cpp File Reference

This is a simple TCP client that sends random data to the server and receives a response.

```
#include <iostream>
#include <string>
#include <cstring>
#include <cstdlib>
#include <ctime>
#include <unistd.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#include <chrono>
```

Include dependency graph for tcp_client.cpp:

3.2 tcp_server.cpp File Reference

This is a simple TCP server that listens for client connections and responds with the number of bytes received.

```
#include <iostream>
#include <string>
#include <cstring>
#include <cstdlib>
#include <unistd.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#include <thread>
#include <chrono>
#include <fstream>
#include <sstream>
#include <ctime>
```

Include dependency graph for tcp_server.cpp:

Macros

- `#define PORT 12345`
Port number for the connection.
- `#define BUFFER_SIZE 2048`
Size of the buffer used for communication.
- `#define LOG_FILE "1M.log"`

Functions

- float `get_cpu_usage` (int pid)
Get the CPU usage of a specific process.
- long `get_memory_usage` (int pid)
Get the memory usage of a specific process.
- void `monitor_resource_usage` (int pid)
Monitor the CPU and memory usage of a process every minute.
- int `main` ()
Main function of the TCP server.

3.2.1 Detailed Description

This is a simple TCP server that listens for client connections and responds with the number of bytes received.

The server listens on a specified port, accepts incoming client connections, receives data, and sends an acknowledgment with the number of bytes received.

3.2.2 Function Documentation

3.2.2.1 `get_cpu_usage()`

```
float get_cpu_usage (  
    int pid )
```

Get the CPU usage of a specific process.

This function calculates the CPU usage percentage of a given process by reading its `/proc/<pid>/stat` file and the system-wide CPU usage from `/proc/stat`. It compares the CPU times over a 1-second interval to compute the usage.

Parameters

<code>pid</code>	The process ID of the target process.
------------------	---------------------------------------

Returns

The CPU usage percentage of the process.

3.2.2.2 get_memory_usage()

```
long get_memory_usage (
    int pid )
```

Get the memory usage of a specific process.

This function retrieves the memory usage of a process by reading the `/proc/<pid>/status` file. The memory usage is returned in kilobytes (KB).

Parameters

<i>pid</i>	The process ID of the target process.
------------	---------------------------------------

Returns

The memory usage of the process in kilobytes (KB).

3.2.2.3 main()

```
int main ( )
```

Main function of the TCP server.

This function creates a server socket, binds it to a specified port, listens for incoming connections, and sends the received byte count back to the client.

Returns

Exit status (0 for success)

3.2.2.4 monitor_resource_usage()

```
void monitor_resource_usage (
    int pid )
```

Monitor the CPU and memory usage of a process every minute.

This function continuously monitors the CPU and memory usage of the specified process every minute. The data is logged to a CSV file, and printed to the console.

Parameters

<i>pid</i>	The process ID of the target process.
------------	---------------------------------------

Index

DOCS-DB, [1](#)

get_cpu_usage
tcp_server.cpp, [6](#)

get_memory_usage
tcp_server.cpp, [6](#)

main
tcp_server.cpp, [7](#)
monitor_resource_usage
tcp_server.cpp, [7](#)

tcp_client.cpp, [5](#)
tcp_server.cpp, [5](#)
get_cpu_usage, [6](#)
get_memory_usage, [6](#)
main, [7](#)
monitor_resource_usage, [7](#)