1) Navigating through directory structure

- In the main directory of the project:"*https://gitlab.ethz.ch/cribin/asl-fall17-project/tree/master*" we have 3 subfolders:
    - The "*src*" subfolder contains the java source code.
    - The "*scripts*" subfolder contains all the bash scripts used to conduct the experiment and all the matlab scripts used to analyze and plot the results.
    - The "*experimental-results*" subfolder contains all the results gathered for each experiment.

2) Overview of the Java classes

- The entire source code of the project can be found under the following link:
  "*https://gitlab.ethz.ch/cribin/asl-fall17-project/tree/master/src*"

- All classes mentioned below are explained in detail in the project in section 1.

- In the source code directory there are 4 relevant subfolders, namely "*connection*", "*message*", "*requestHandlers*" and "*utils*".
- The "*connection*" subfolder contains only the **SynConnection.java** class, responsible for synchronous connection between worker and server.
- The "*message*" subfolder contains wrapper classes for requests (**MemcachedRequest.java**) and responses (**MemcachedResponse.java**). Additionally the folder contains a class responsible for parsing incoming requests and responses (**MemcachedMessageParser.java**).
- The "*requestHandlers*" folder contains the classes responsible for handling set (**SetRequestHandler.java**), get (**GetRequestHandler.java**) and multi-get(**MultiGetRequestHandler.java**) requests. These request handler classes are used by the **RequestWorker.java** class, which represent the logic of the worker-thread.

- The "utils" folder contains utility classes for storing commonly used methods, constants, the hasher class (**Hasher.java**) and logging classes.
- Other important classes are directly placed in the source directory:
    - **MyMiddleware.java** (Initializes the workers and runs the net-thread)
    - **RequestWorker.java**(Implements the logic of the worker-thread)
    - **ClientConnectionHandler.java**(Implements the logic of the net-thread)
    - **ClientHandler.java**(Main class for communicating with clients and storing requests)
    - **RunMW.java** (Main class called at the startup of the Middleware)

3) Overview of experiment file naming conventions

- We will use the following variables to define the file names:

w = workload(read or write)
vc = number of VCs per client machine
cm = indicates which client machine produced the file(either 0, 1 or 2)
rep = current repetition of the experiment (either 1, 2 or 3)
sm = indicates server machine(either 1 or 2)
wt = number of worker threads per mw (either 8, 16, 32 or 64)
mwm = indicates which mw machine produced the file (either 1 or 2)
ks = multi-get key size (either 1, 3, 6 or 9)

(For 2k only:)
numServ = number of servers used (2 or 3)
numMW = number of mws used (1 or 2)

- All folder locations have the following prefix:
prefix = *https://gitlab.ethz.ch/cribin/asl-fall17-project/tree/master/experimental-results*

- The log files produced by the middleware have the following syntax:
  - For each worker thread we print multiple lines of data representing the measurements taken over a 2 second window.
  - Each line contains the following information:
    “Throughput, QueueLength, QueueWaitingTime, ServerWaitingTime, ResponseTime”
  - After the data of each worker has been printed, we further log information gathered when the middleware is killed: “Histogram”, “Cache Miss Ratio”, “Num. of Sets/Gets/Multi-Gets” (in later log files we further included the “Server Hash Count”, “Average number of keys” for multi-gets and “Number of unknown request types found”).

3.1) Baseline without MW Files:

- One server setting: File location= "*prefix/1.baseline_no_mw/baseline/one_server*"
  Filename ="baseline_w_vc-cm-rep.txt"

- Two server setting: File Location="*prefix/1.baseline_no_mw/baseline/two_servers*"
  Filename="baseline_w_vc-cm-rep-sm.txt"

3.2) Baseline with MW Files:

- One MW Setting/Client Measurements: File location="*prefix/2.baseline_mw/one_mw/client_log/client_read(write)*"
  Filename="client_one_mw_w_vc-rep-wt.txt"

- One MW Setting/MW Measurements: File location="*prefix/2.baseline_mw/one_mw/mw_log/mw_read(write)*"
  Filename="one_mw_w_vc-rep-wt.log"

- Two MW Settings/Client Measurements: File location="*prefix/2.baseline_mw/two_mws/client_log/client_read(write)*"
  Filename="client_two_mw_w_vc-rep-wt-mwm.txt"

- Two MW Settings/MW Measurements: File location="*prefix/2.baseline_mw/two_mws/mw_log/mw_read(write)*"
  Filename="two_mw_w_vc-rep-wt-mwm.log"

- Two MW-Two Clients/Client Measure.: File location="*prefix/2.baseline_mw/two_clients_2_mws/client_log/client_read(write)*"

Filename="client_two_mw_w_vc-cm-rep-wt-mwm.txt"

- Two MW-Two Clients/MW Measure.: File location="*prefix/2.baseline_mw/two_clients_2_mws/mw_log/mw_read(write)*"
  Filename="two_mw_w_vc-rep-wt-mwm.log"

3.3) Throughput Writes Files:

- Client Measurements: File location="*prefix/3.thrp_writes/client_log*"
  Filename="client_thrp_w_vc-cm-rep-wt-mwm.txt"

- MW Measurements: File location=*"prefix/3.thrp_writes/mw_log*"
  Filename="mw_thrp_w_vc-rep-wt-mwm.log"

3.4) Multi-Get Files:

- Sharded Setting/Client Measurements: File location="*prefix/4.multi_get/sharded/client_log*"
  Filename="client_multi_get_shard_vc-cm-rep-wt-ks-mwm.txt"

- Sharded Setting/MW Measurements: File location="*prefix/4.multi_get/sharded/mw_log*"
  Filename="mw_multi_get_shard_vc-rep-wt-ks-mwm.log"

- NonSharded Setting/Client Measure.: File location="*prefix/4.multi_get/nonsharded/client_log*"
  Filename="client_multi_get_non_shard_vc-cm-rep-wt-ks-mwm.txt"

- NonSharded Setting/MW Measure.: File location="*prefix/4.multi_get/nonsharded/mw_log*"
  Filename="mw_multi_get_non_shard_vc-rep-wt-ks-mwm.log"

3.5) 2k Files:

- Client Measurements:	File location="*prefix/5.2k/client_log/write(read/read-write)*" Filename="client_2k_numServ-numMW-wt-rep-cm-mwm.txt"
- MW Measurements:	File location="*prefix/5.2k/mw_log/write(read/read-write)*" Filename="mw_2k_numServ-numMW-wt-rep-mwm.log"

3.6) Server Hash Count Files:

- In order to prove that all servers are under equal load using hashing, we repeated the multi-get experiments with 32 VC per client and a maximum of 1 and 9 key-sizes. We only used the middleware measurements, as the middleware log file contains how many time each server has been hashed.

- MW Measurements:	File location="*prefix/6.server_hash_count/*" Filename="mw_multi_get_non_shard_vc-rep-wt-ks-mwm.log"

4) Overview of the bash scripts

- All bash scripts have the following prefix for the location:
  bashPrefix= “https://gitlab.ethz.ch/cribin/asl-fall17-project/tree/master/scripts/bash-scripts”
- The bash scripts are used to conduct the experiments and gather the results for each section. The experiments are executed according to the experiment settings specified in the report.
- Before the experiments were run, we executed the bash script “fill_cache.sh”(located in the bashPrefix folder) in order to fill the cache of the memcached servers beforehand.

4.1) Baseline without MW:

- One server experiment:	Script location= "*bashPrefix/1.baseline_no_mw/*" Scriptname ="run_baseline_one_server.sh"
- Two server experiment:	Script Location="*prefix/1.baseline_no_mw/*" Scriptname ="run_baseline_two_servers.sh"

4.2) Baseline with MW:

- One mw experiment:	Script location= "*bashPrefix/2.baseline_with_mw/*" Scriptname ="baseline_one_mw.sh"
- Two mw experiment:	Script Location="*bashPrefix/2.baseline_with_mw/*" Scriptname ="baseline_two_mws.sh"
- Two mw/Two clients experiment:	Script Location="*bashPrefix/2.baseline_with_mw/*" Scriptname ="baseline_two_clients_two_mws.sh"

4.3)Throughput Writes:

- Thrp. Writes experiment:	Script location= "*bashPrefix/3.thrp_writes/*" Scriptname ="thrp_writes_exp.sh"

4.4) Multi-Get Experiments:

- Sharded experiment:	Script location= "*bashPrefix/4.multi_get/*"
	Scriptname ="multi_get_sharded.sh"
- NonSharded experiment:	Script Location="*bashPrefix/4.multi_get/*"
	Scriptname ="multi_get_non_shard.sh"

4.5) 2k Experiments:

- 2k experiment:	Script location= "*bashPrefix/5.2k/*"
	Scriptname ="2k_exp.sh"

5) Overview of the matlab scripts

- Analyzing and plotting the data gathered the experiments was done using matlab.
- For most of the sections we have one script to parse the client measurements and another to parse the middleware log files. Finally we have scripts to plot the results of the client and mw measurements. The plotter scripts have to be called after the parser scripts, as the former is dependant on the data of the latter. The same plotter scripts have been used for multiple experiments with slight modifications for each experiment.
- All matlab script locations have the following prefix:

  matlabPrefix=”https://gitlab.ethz.ch/cribin/asl-fall17-project/tree/master/scripts/matlab-scripts”

5.1) Baseline without MW:

- The script location for all scripts are:	Script location= "*matlabPrefix/1.baseline_no_mw/*"
- One Server/Client Parser:	Scriptname ="baseline_parser_one_server.m"
- Two Servers/Client Parser:	Scriptname ="baseline_parser_two_servers.m"

- Client Plotter:	Scriptname ="baselinePlotter.m"

5.2) Baseline with MW:

- One MW/Client Parser:	Script location= "*matlabPrefix/2.baseline_with_mw/one_mw/*" Scriptname =”client_one_mw_parser.m"
- One MW/MW Parser:	Script location= "*matlabPrefix/2.baseline_with_mw/one_mw/*" Scriptname ="mwLog_one_mw_parser.m"
- Two MWs/Client Parser:	Script Location="*matlabPrefix/2.baseline_with_mw/two_mw/*" Scriptname ="client_two_mws_parser.m"
- Two MWs/MW Parser:	Script Location="*matlabPrefix/2.baseline_with_mw/two_mw/*" Scriptname ="mwLog_two_mws_parser.m"
- Client Plotter:	Script Location="*matlabPrefix/2.baseline_with_mw/*" Scriptname ="clientLogPlotter.m"
- MW Plotter:	Script Location="*matlabPrefix/2.baseline_with_mw/*" Scriptname ="mw_log_plotter.m"

5.3) Throughput Writes:

- The script location for all scripts are:	Script location= "*matlabPrefix/3.thrp_writes/*"
- Client Parser:	Scriptname ="client_log_thrp_writes_parser.m"
- MW Parser:	Scriptname ="mw_log_thrp_writes_parser.m"

- The plotter scripts are the same as in section 5.2.

5.4) Multi-Get:

- The script location for all scripts are:	Script location= "*matlabPrefix/4.multi_get/*"

- Sharded/Client Parser:	Scriptname ="client_multi_get_sharded_parser.m"
- NonSharded/Client Parser:	Scriptname ="client_multi_get_non_sharded_parser.m"
- Sharded and NonSharded /MW Parser:	Scriptname ="mw_multi_get_parser_new.m"

(For this parser it is important to first parse the sharded and then the nonsharded files)

- MW Plotter	Scriptname ="mw_multi_get_plotter.m"
- Client Plotter:	Scriptname ="client_multi_get_plotter.m"

5.5) 2k

- Since we mainly used the measured by the clients for the 2k analysis, we only implemented a parser for the client measurements.

- The script location for all scripts are:	Script location= "*matlabPrefix/5.2k/*"
- Client Parser:	Scriptname="parser_2k.m"
- 2k Analysis:	Scriptname="Analysis_2k.m"

(The 2k analysis script has to be called after the client parser script. The analysis script mainly uses the methods described in the book in order to perform a 2k analysis and computes all necessary output parameters.

5.6) Queue Models:

- The script location for all scripts are:	Script location= "*matlabPrefix/6.queue_models/*"
- M/M/1 analysis:	Scriptname ="M_M_1.m"
- M/M/m analysis:	Scriptname ="M_M_m.m"
- Network of queues:	Scriptname ="NetworkOfQueues.m"
- Network of queues plotter:	Scriptname="network_queues_plotter.m"

(All analysis scripts can be called completely independent of any other scripts, as all the input parameters needed are already stored in the scripts.)