

MAKERERE UNIVERSITY

College Of Computing and Information Sciences (CoCIS)

Bachelor of Science in Software Engineering

BSE 1301: Professional Min SE

Practical Project 1

Group Members

Name	Student Number	Registration Number
Amoko Ivan	217014453	17/U/2680/EVE
Tindyebwa Fortunate Allan	217005376	17/U/10441/EVE
Bwayo Joel	217010218	17/U/3777/EVE
Ssemakula Rajab	217011090	17/U/10165/EVE

Project Procedures to follow in using the Word task server.

The Client enters different commands to be processed via the command line interface, such commands include the following.

Command	Example	Result
Double	double word	wordword
Reverse	rev word	drow
Delete	people 1,3	eple
Replace	replace middle 1-h,2-u,3-s,4-t	hustle
Encryption	encrypt zoo	2y1Hp761nF51nF5
Decryption	decrypt 1oG6E1nF51oG61kC2E	people

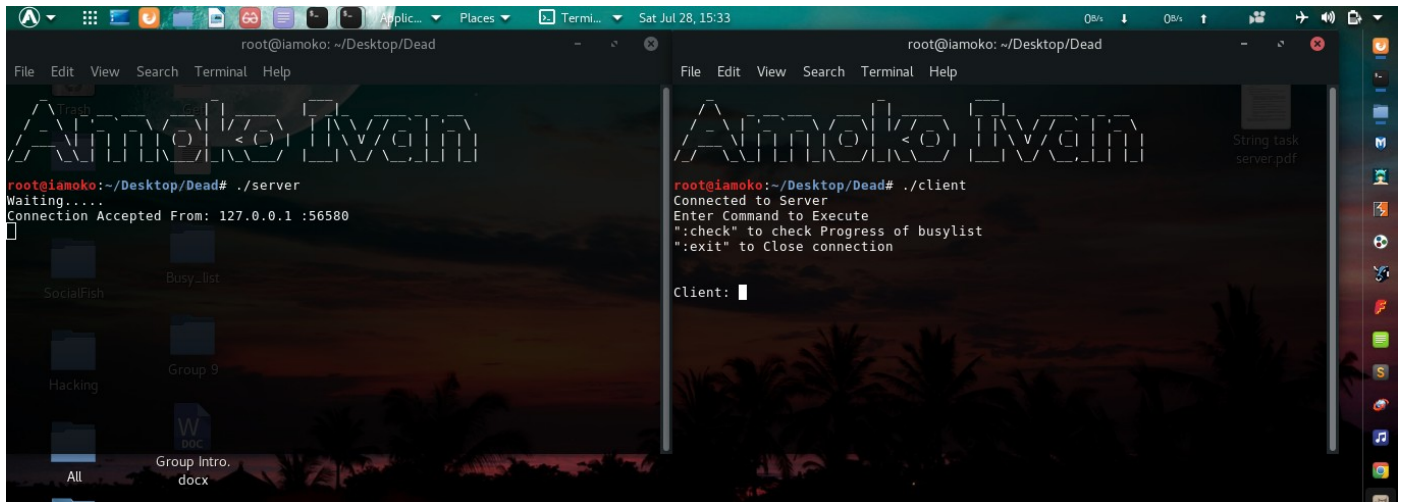
The client program can only run when the server is already running, the client is able to enter the command and string to be processed by the server and at the same time is also able to exit from the server's connection after he/she is done with submitting tasks and has got all the results needed.

To compile server program `gcc -o server $(mysql_config --cflags) server.c $(mysql_config --libs)`

The server and client interfaces are as shown in the images below.

Server Program

Client Program



A client can send as many tasks as possible, these tasks are processed and for any task, the logs are stored directly into the database and the processed results are stored into the ready_job file.

The ready file is arranged line by line with

line 1 – Client's ID

line 2 – Time in which the task was sent

line 3 – Date in which the task was sent

line 4 – Task ID

line 5 – Task Type

line 6 – String

line 7 – Result

line 8 – Processing duration of the task

line 9 – Priority

```

root@iamoko: ~/Desktop/Dead
File Edit View Search Terminal Help
Annoko Ivan
root@iamoko:~/Desktop/Dead# ./server
Waiting.....
Connection Accepted From: 127.0.0.1 :56580
Client: double many
String doubled
Client: rev few
String reversed
Client: people 2,5
Specified characters Deleted
Client: replace joan 1-I,2-v,3-a
String Replaced
Client: encrypt alot
String Encrypted
Client: decrypt ABD
String Decrypted

root@iamoko:~/Desktop/Dead
File Edit View Search Terminal Help
Annoko Ivan
root@iamoko:~/Desktop/Dead# ./client
Connected to Server
Enter Command to Execute
":check" to check Progress of busylist
":exit" to Close connection
Client: double many;rev few;people 2,5;replace joan 1-I,2-v,3-a;encrypt alot;decrypt ABD
Feedback: manymany
Feedback: wef
Feedback: pope
Feedback: Ivan
Feedback: A1kC2lnF52s1Bj10
Feedback: abd
Client:
  
```

The ready list file

Database Table showing Logs

ready_jobs.txt

```

56581
15:33:16
2018-7-28
1
Double
many
manymany
0.000031
0
56581
15:33:16
2018-7-28
2
Reverse
few
wef
0.000002
0
56581
15:33:16
2018-7-28
3
Delete
people
pope
0.000026
0
56581
15:33:16
2018-7-28
4
Replace
joan 1-I,2-v,3-a
Ivan
0.000053
0
56581
  
```

localhost / localhost / recess / Logs | phpMyAdmin 4.6.6deb5 - Mozilla Firefox

localhost/phpmyadmin/sql.php?server=1&d

Most Visited Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng

phpMyAdmin

Recent Favorites

New

- information_schema
- mysql
- performance_schema
- recess
 - New
 - Entry
 - Logs
 - member
- testdb

Server: localhost:3306 Database: recess Table: Logs

Browse Structure SQL More

SELECT * FROM `Logs`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all Number of rows: 25 Filter rows: Search

+ Options

User_ID	Task_ID	Time	Date	Task
56581	1	15:35:55	2018-07-28	Double
56581	2	15:35:57	2018-07-28	Reverse
56581	3	15:35:59	2018-07-28	Delete
56581	4	15:35:59	2018-07-28	Replace
56581	5	15:36:01	2018-07-28	Encrypt
56581	6	15:36:03	2018-07-28	Decrypt

Show all Number of rows: 25 Filter rows: Search

Console

When a PHP Script that reads data from the ready_job file is run, it records the data from the file into the database table and each read data that has been entered into the database tables, is immediately deleted from the ready_job file every 1 minute.

read_jobs file

PHP Scripts that reads data

Database into which data is stored

The screenshot shows a web application interface. On the left, a terminal window displays a list of tasks with columns: ID, Time, Date, Job ID, Task Type, String, Result, Processing Duration, and Priority. The tasks are:

ID	Time	Date	Job ID	Task Type	String	Result	Processing Duration	Priority
56581	15:33:16	2018-7-28	2	Reverse	few	wef	0.000002	0
56581	15:33:16	2018-7-28	2	Reverse	few	wef	0.000002	0

In the center, a message states: **Data Has been Successfully Uploaded to the Database**.

On the right, a phpMyAdmin interface shows the 'Entry' table in the 'recess' database. The table structure is as follows:

ID	Time	Date	Job ID	Task Type	String	Result	Processing Duration	Priority
56581	15:33:16	2018-7-28	1	Double	many	manymany	0.000031	0
56581	15:33:16	2018-7-28	2	Reverse	few	wef	0.000002	0

Task whose strings have more than 50 characters are not processed by the server. Such tasks are saved in the file called "blacklisted.txt".

The screenshot shows a terminal window with the following output:

```

root@iamoko: ~/Desktop/Dead
File Edit View Search Terminal Help
root@iamoko:~/Desktop/Dead# ./server
Waiting.....
Connection Accepted From: 127.0.0.1 :33638
root@iamoko:~/Desktop/Dead# ./client
Connected to Server
Enter Command to Execute
":check" to check Progress of busylist
":exit" to Close connection
Client: double qwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwert
yuio
Feedback: String is lengthy hence can't be Processed
Client:

```

Tasks that are sent by client2 when the server is still handling tasks from client1 are stored into the busy_list file as shown below.


```

root@iamoko: ~/Desktop/Dead
File Edit View Search Terminal Help
Client: rev few
String reversed
Client: people 2,5
Specified characters Deleted
Client: replace joan 1-I,2-v,3-a
String Replaced
Client: encrypt alot
String Encrypted
Client: decrypt ABD
String Decrypted
Connection Accepted From: 127.0.0.1 :56626
Client: encrypt something
String Encrypted
Client: double another
Server busy
file small
Server busy
Server is still busy
file small
String doubled
Client: rev lesser
String reversed

root@iamoko: ~/Desktop/Dead
File Edit View Search Terminal Help
root@iamoko:~/Desktop/Dead# ./client
Connected to Server
Enter Command to Execute
":check" to check Progress of busylist
":exit" to Close connection
Client: double many;rev few;people 2,5;replace joan 1-I,2-
v,3-a;encrypt alot;decrypt ABD
Feedback: manymany
Feedback: wef
Feedback: pope
Feedback: Ivan
Feedback: A1kC2lnF52s1Bj10
Feedback: abd
Client: encrypt something;double another;rev lesser;replac
e laptop 1-c,2-o,3-m
Feedback: 1r1Ai091nF51lD3E2s1Bj10HI1mE4G
Feedback: anotheranother
Feedback: ressel
Feedback: comtop

root@iamoko:~/Desktop/Dead# ./client
Connected to Server
Enter Command to Execute
":check" to check Progress of busylist
":exit" to Close connection
Client: replace mind 1-h;double nothing
Feedback: Server is Still busy
Feedback: Server is Still busy
Client:

```

Busy_list file is as shown.

Where;

Column 1 – Client ID.

Column 2 – Task Type.

Column 3 -String to be processed.

```

root@iamoko: ~/Desktop/Dead
File Edit View Search Terminal Help
Open ready_jobs.txt x busy_list.txt x waiting.txt x
56626 replace mind 1-h
56626 double nothing
root@
Conne
Enter
":che
":exi

Clie
v,3-a
Feedb
Feedb
Feedb
Feedb
Feedb
Feedb
Clie
e lap
Feedb
Feedb

root@iamoko:~/Desktop/Dead# ./client
Connected to Server
Enter Command to Execute
":check" to check Progress of busylist
":exit" to Close connection
Client: replace mind 1-h;double nothing
Feedback: Server is Still busy
Feedback: Server is Still busy
Client:

```

When the server is done processing the other client's tasks, It's going to Process the tasks in the Busy_list file by assigning priorities to them and they are stored into a different file called "waiting" as shown below.

```

Open waitin... /var/fw... Save
ready_jobs.txt x busy_list.txt x waiting.txt x
1 56626 rev none
2 56626 double nothing
3 56626 replace mind 1-h

```

Client 2 Enters the ':check' command to retrieve his\her results that were send when the server was busy. On retrieving the results, the server reads the "waiting" list file line by line and processes the tasks. Tasks are read from the file, processed, cleared from the waiting file and the results are stored into the ready_list file which is again read by the PHP script and stored into the database.

```

root@iamoko: ~/Desktop/Dead
Client: decrypt ABD
String Decrypted
Connection Accepted From: 127.0.0.1:56626

Client: encrypt something
String Encrypted

Client: double another
Server busy
Server is still busy
file small
Server busy
Server is still busy
file small
String doubled

Client: rev lesser
String reversed

Client: replace laptop 1-c,2-o,3-m
String Replaced
Server busy
Server is still busy
Please wait as we Process the busylist
String reversed
1 rev enon
nothingnothing
String doubled
2 double nothing
String Replaced
3 replace mind 1-h
Continue

root@iamoko: ~/Desktop/Dead
Connected to Server
Enter Command to Execute
":check" to check Progress of busylist
":exit" to Close connection

Client: replace mind 1-h;double nothing
Feedback: Server is Still busy
Feedback: Server is Still busy

Client: rev none
Feedback: Server is Still busy

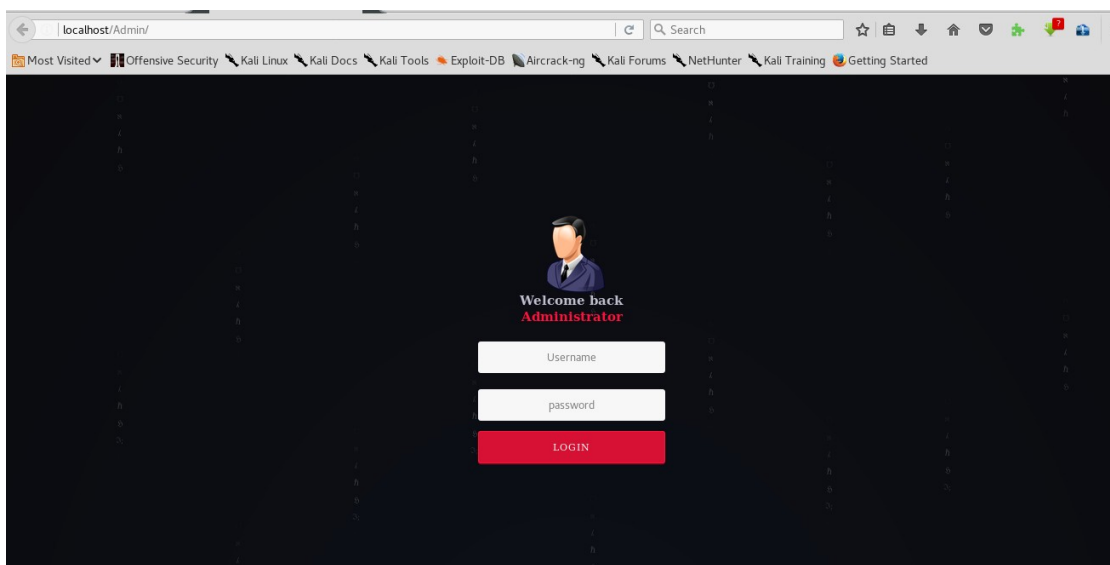
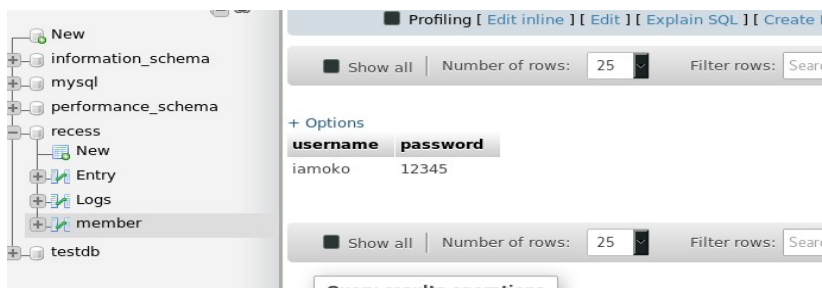
Client: :check
Feedback: enon
Feedback: nothingnothing
Feedback: hind
Feedback: Done

Client:

```

The Administrator is able to monitor the operation of the entire system via the web interface.

The Admin has to login via web interface and the details used to log into the web interface are in the database.



On Logging into the web page, the administrator is able to view;

Average rate at which server Processes Jobs.

Ready jobs, Waiting Jobs, Waiting Jobs with Priority, Jobs with high Success rate and the jobs with High failure rate.

Administrator's Home page.

Welcome iamoko

Home

Ready Jobs

Waiting Jobs

Waiting Jobs with Priority

Jobs With High Success Rate

Jobs With High Failure Rate

Logout

Average Rate at which Server Processes Jobs:
0.00012533333333333334
Seconds

Processed Jobs
9
Jobs

Unprocessed Jobs
5
Jobs

Welcome
Administrator

Ready Jobs;

These are jobs are read from the database who processing was successful and the clients got results.

Welcome iamoko

READY JOBS

Home

Ready Jobs

Waiting Jobs

Waiting Jobs with Priority

Jobs With High Success Rate

Jobs With High Failure Rate

Logout

CLIENT ID	TIME	DATE	JOB ID	PROCESSING DURATION	TASK TYPE	STRING	RESULT	PRIOR
56581	15:33:16	2018-7-28	1	0.000031	Double	many	manymany	0
56581	15:33:16	2018-7-28	2	0.000002	Reverse	few	wef	0
56625	15:58:6	2018-7-28	5	0.00096	Encrypt	something	1r1Ai091nF51ID3E2s1Bj10HI1mE4G	0
56625	15:58:6	2018-7-28	1	0.000025	Double	another	anotheranother	0
56625	15:58:6	2018-7-28	2	0.000007	Reverse	lesser	ressel	0
56625	15:58:6	2018-7-28	4	0.000033	Replace	laptop 1-c,2-0,3-m	comtop	0
56626	15:58:6	2018-7-28	2	0.000013	Reverse	none	enon	1
56626	15:58:6	2018-7-28	1	0.000024	Double	nothing	nothingnothing	2
56626	15:58:6	2018-7-28	4	0.000033	Replace	mind	hind	3

Waiting Jobs;

These are the jobs which haven't been assigned priorities yet, and so the chances of their success would be low.

Welcome iamoko

Home

Ready Jobs

Waiting Jobs

Waiting Jobs with Priority

Jobs With High Success Rate

Jobs With High Failure Rate

Logout

WAITING JOBS

CLIENT ID	TASK TYPE	STRING
56626	replace	mind 1-h
56626	double	nothing

Waiting Jobs with Priority;

Jobs that have been assigned priorities meaning that they are going to be processed by the server and hence their success rate is high.

Welcome iamoko

Home

Ready Jobs

Waiting Jobs

Waiting Jobs with Priority

Jobs With High Success Rate

Jobs With High Failure Rate

Logout

WAITING JOBS WITH PRIORITY

PRIORITY	CLIENT ID	TASK TYPE	STRING
1	56626	rev	none
2	56626	double	nothing
3	56626	replace	mind 1-h

Jobs With High Success Rate;

This is calculated from
$$= \frac{(\text{Number of Jobs for a student in ready list}) * 100 \%}{(\text{Sum of jobs in ready list}) + (\text{Sum of jobs in busy list})}$$

Welcome iamoko		JOBS WITH HIGH SUCCESS RATE	
<div>Home</div> <div>Ready Jobs</div> <div>Waiting Jobs</div> <div>Waiting Jobs with Priority</div> <div>Jobs With High Success Rate</div> <div>Jobs With High Failure Rate</div> <div>Logout</div>	CLIENT ID		PERCENTAGE (%)
	38242		13.333333333333
	38318		33.333333333333

Jobs With High Failure Rate;

This is calculated from
$$= \frac{(\text{Number of Jobs for a student in busy list}) * 100 \%}{(\text{Sum of jobs in ready list}) + (\text{Sum of jobs in busy list})}$$

Welcome iamoko		JOBS WITH HIGH FAILURE RATE	
<div>Home</div> <div>Ready Jobs</div> <div>Waiting Jobs</div> <div>Waiting Jobs with Priority</div> <div>Jobs With High Success Rate</div> <div>Jobs With High Failure Rate</div> <div>Logout</div>	CLIENT ID		PERCENTAGE (%)
	56626		25