Student ID: 10135033

Module: 5004CEM Operating Systems and Security

Course: BSc Computer Science

**Table of Contents**

[**Introduction to the Report** 3](#_Toc95739898)

[**Report Tasks** 3](#_Toc95739899)

[**Task 1 - ACL Approaches** 3](#_Toc95739900)

[**Task 2 – Assembly Code Game** 4](#_Toc95739901)

[**Commented Code for the Game** 4](#_Toc95739902)

[**Multiple Examples of Code Running** 4](#_Toc95739903)

[**Task 4 – TCP Festival Booking System** 5](#_Toc95739904)

[**Commented Code** 5](#_Toc95739905)

[**Multiple Examples of Code Running** 5](#_Toc95739906)

[**Conclusion and Reflection** 6](#_Toc95739907)

[**References** 6](#_Toc95739908)

# **Introduction to the Report**

As for an introduction, this report will consist of ACL Approaches, will contain a HI/LO game assembly code and finally C++ code TCP Festival Booking System.

# **Report Tasks**

# **Task 1 - ACL Approaches**

As for access control approaches, there is an option to use passwords or biometrics, both have pros and cons. what is biometrics? you would ask. Biometrics are biological measurements or physical characteristics. (Kaspersky, 2022). Biometrics can be a fingerprint a face picture or its measurements saved for facial recognition Also there are biological biometrics, morphological biometrics and behavioural biometrics I am going to talk about various types of it. Considering nowadays our smartphones have biometric scanners And they allow us to save our face measurements into the phone storage. as if we do that we can unlock the phone with the face only we can also set an option on the phone we can access online banking using facial recognition instead of putting a pin code, the huge proof that we do not need to memorise various pin codes or passwords we can just use our face to identify ourselves. As for computer system security, the face recognition feature is great, we might have many accounts to log in to, etc, a face could replace that. But thinking of cons, if using face recognition and the biometrics get leaked into the web which is unlikely but still can happen (Gargaro, 2021), someone else could take over all possessions of that person who uses their face as login, and a face could not be changed. On side of passwords, passwords stored on the servers are encrypted and they preserve anonymity. Password can be changed anytime as it could be also leaked anytime, it only depends on the user’s actions what websites the user visits or what software they download. Or passwords could be forgotten with no way of recovery, that is the worst that can happen, sometimes websites or software provide the user the recovery phrases but those can be forgotten, written on paper and thrown away into the bin. Face recognition is great at workplaces, it can be used by employees to clock in and clock out, there is no need to make a card with yourself that you have to punch in each time. There is voice recognition which is still used in banks nowadays, but that can be manipulated, if the person received a spam call and started talking, the voice can be recorded and rebuilt for certain phrases (using audacity). Fingerprint scanning is a good alternative to Facial recognition, this thing was very quiet until Apple released their new iPhone with Touch ID technology, (Wikipedia contributors, 2022) which had a fingerprint scanner and it allowed the phone users to unlock their phone with a fingerprint. This thing has evolved to the level of face recognition. Some biometrical scanners have either face recognition or fingerprint scanner together. Touch ID allows to confirm payments using a fingerprint, view saved passwords, and access some phone settings. Well if only using fingerprint scanners everywhere, it might not be a great thing as a fingerprint can be cloned, or if they lose a finger, there will be no more access, and passwords are still used together with fingerprint scanners. If we can’t access it with a fingerprint, we can access it with a password. If there would be a guarantee that biometrics would be fully protected and no one could ever steal any data, we could start using face recognition everywhere. Evolve it and link a bank account to your face, you could come to the train station, pick a destination, and pay with your face, go through the gate with your face, no need to carry papers, even no need to unlock a phone. To summarise right now face recognition is a controversial topic and many people speculate about it, if biometrics would be secure, it can replace all wallets, a face could be accessed to everything, also It would be a great thing against thieves, as no need to carry a wallet. If identification is prompt, face recognition would be the easiest thing to do. Passwords are getting slowly outdated, there are already large databases of leaked passwords, parse all into a bruteforcer and you could maybe get access to an account you need. There are many password-stealing methods, phishing, rat viruses, spyware, etc. Passwords would only preserve anonymity, those can be changed from time to time to feel more secure, but a password leak could happen anytime and someone whose password is stolen would have to react quickly. Face recognition in my opinion is more secure and the only registered face can access a dedicated account.

# **Task 2 – Assembly Code Game**

## **Commented Code for the Game**

section .data

;

;

;Welcome message for game

;Welcome message size

;Message that the guess is correct

;Size of the message

samemsg db "You are correct! ", 10

samelen equ $-samemsg

;Message that the guess is incorrect

;Size of the message

notsamemsg db "You are incorrect! ", 10

notsamelen equ $-notsamemsg

;Same as endl in C++

cr db 10

score db 0 ; no idea

xscore db 0

welcome db "The temperature on the 7th of April in London was 5 degrees", 20

welcomeLen equ $-welcome

question db "Was the temperature in Abui Dhabi Higher or lower ?", 10

questionLen equ $-question

question2 db "Was the temperature in Rome Higher or Lower than Abui Dhabi? "

question2Len equ $-question2

question3 db "Was the temperature in Paris Higher or Lower than Rome ?"

question3Len equ $-question3

question4 db "Was the temperature in Vilnius Higher or Lower than Paris ?"

question4Len equ $-question4

question5 db "Was the temperature in Talinn Higher or Lower than Vilnius ?"

question5Len equ $-question5

scoreans db "Your score is now: "

scoreansLen equ $-scoreans

global listQuestions

listQuestions:

dq question ; the answer letters are stores in 8 bytes to aid the comparison

dq question2

dq question3

dq question4

dq question5

global listQuestionsLen

listQuestionsLen:

dq questionLen ; the answer letters are stores in 8 bytes to aid the comparison

dq question2Len

dq question3Len

dq question4Len

dq question5Len

global listAnswers

listAnswers:

dq 'H' ; the answer letters are stores in 8 bytes to aid the comparison

dq 'L'

dq 'L'

dq 'H'

dq 'H'

letter:

dq 0 ; where we store each of the answers one at a time

segment .bss

guess resb 1 ; store the users guess

;score resb 2 ; store the users score

section .text

global \_start ;must be declared for linker (ld)

\_start:

call displayWelcome

call newLine

mov rax,5 ;number of answers

mov rbx,0 ;RBX will store the letter

mov r8, listQuestions

mov rcx, listAnswers ;RCX will point to the current element array to be guessed

;Main function that calls other functions

top:

mov rbx, [rcx] ; put the current letter being guessed in rbx

mov [letter],rbx ; move rbx into a variable letter that stores the current guess

;mov r8,[letter]

push rax ; push rax on stack

push rcx ; push rax on stack

call displayQuestion ; display the question

call reading ; call reading to get the users guess

call displayletterMessage ; Display the message for the correct letter

call newLine ; new line like endl in C++

call display ; print the letter they should have guessed

call newLine ; new line like endl in C++

pop rcx ; get back from stack

pop rax ; get back from stack

add rcx,8 ;move pointer to next element as 8 bits for each move on by 8

add r8, 8 ; move pointer to next element

dec rax ;decrement counter so going down

jnz top ;if counter not 0, then loop again

call done ; end program

;Display function

display:

mov edx,1 ;message length

mov ecx, letter ;message to write the letter to be predicted

mov ebx, 1 ;file descriptor (stdout)

mov eax, 4 ;system call number (sys\_write)

int 0x80 ;call kernel

ret

;function to read the user guess and do comparison with the answer

reading:

mov eax, 3 ; read from keyboard

mov ebx, 2;

mov ecx, guess ; move guess into ecx

mov edx, 1 ; As single letter using 1 byte

int 80h ; call interrupt

mov rax, [guess] ; move guess by user into rax

cmp rax, [letter] ; compare correct answer with what in rax

je same ; if guess was correct jump to same function

call Notsame ; if the guess is incorrect then go to Notsame function

ret

; function to show message that answer was not correct answer

Notsame:

mov ecx,notsamemsg ; Not same message

mov edx, notsamelen ; length of same message

mov ebx,1 ;file descriptor (stdout)

mov eax,4 ;system call number (sys\_write)

int 80h

mov eax, 3 ; read previous enter key press

mov ebx, 2;

mov ecx, guess ; Deal with previous enter key press so it doesnot messy up loop

mov edx, 1 ; As single letter using 1 byte

int 80h ; call interrupt

ret

; function to show message answer was correct

same:

call increaseScore

mov ecx,samemsg ; same message

mov edx, samelen ; length of same message

mov ebx,1 ;file descriptor (stdout)

mov eax,4 ;system call number (sys\_write)

int 80h

mov eax, 3 ; read previous enter key press from keyboard

mov ebx, 2;

mov ecx, guess ; Deal with previous enter key press so it doesnot messy up loop

mov edx, 1 ; As single letter using 1 byte

int 80h ; call interrupt

ret

; Function to create a New line like endl in C++

newLine:

mov eax,4 ; Put 4 in eax register into which is system

;call for write (sys\_write)

mov ebx,1 ; Put 1 in ebx register which is the standard

; output to the screen

mov ecx, cr ; Put the newline value into ecx register

mov edx, 1 ; Put the length of the newline value into edx

; register

int 80h ; Call the kernel with interrupt to check the

; registers and perform the action of moving to

; the next line like endl in c++

ret ; return to previous position in code

;Function to display welcome to game message

displayWelcome:

mov edx,welcomeLen ;message length

mov ecx, welcome ;message to write

mov ebx, 1 ;file descriptor (stdout)

mov eax, 4 ;system call number (sys\_write)

int 0x80 ;call kernel

ret

;Function to display question for quiz

displayQuestion:

mov edx, 61 ;message length

mov ecx, [r8] ;message to write

mov ebx, 1 ;file descriptor (stdout)

mov eax, 4 ;system call number (sys\_write)

int 0x80 ;call kernel

ret

;function to increase the score

increaseScore:

mov ecx, score

inc byte [score] ; increase score by byte which each time +1

mov ecx, score

mov eax, [score]

add eax, 48

mov [xscore], eax

ret

;printmessagescore:

; mov edx, scoreansLen ;message length

; mov ecx, scoreans ;message to write

; mov ebx, 1 ;file descriptor (stdout)

; mov eax, 4 ;----system call number (sys\_write)

; int 80h ;call kernel

; ret

; Function to display the correct answer sentence

;zeotherway:

displayletterMessage:

mov eax,4

mov ebx, 1

mov ecx, scoreans ;message to write

mov edx, scoreansLen ;message length

int 80h

mov edx, 1

mov ecx, xscore

mov ebx, 1

mov eax, 4

int 80h

call newLine

ret

; Function to end the program

done:

mov eax, 1 ;system call number (sys\_exit)

int 0x80 ;call kernel

## **Multiple Examples of Code Running**

Compiles.

Answerred all correctly:

Text

Description automatically generated

Answerred partly incorrect:

Text

Description automatically generated

Answerred all incorrectly:

Text

Description automatically generated

# **Task 4 – TCP Festival Booking System**

## **Commented Code**

//THIS IS TCP-SERVER

*//tcp-server*

*//TCP Festival booking system*

*//Takes input as a whole sentence then splits it.*

*//*

*//Output is produced on tcp-client*

*//isdigit() https://www.cplusplus.com/reference/cctype/isdigit/*

*//stoi() https://www.cplusplus.com/reference/string/stoi/*

*//push\_back https://www.cplusplus.com/reference/vector/vector/push\_back/*

#include <arpa/inet.h>

#include <netdb.h>

#include <netinet/in.h>

#include <unistd.h>

#include <iostream>

#include <cstring>

#include <stdlib.h>

#include <cmath>

#include <string>

#include <vector>

#define MAX\_MSG 100

#define LINE\_ARRAY\_SIZE (MAX\_MSG+1)

**using** **namespace** std;

bool isnum(string phrase) *//boolean isnum*

{ *//array is built from split up sentence(input)*

**for**(int i=0;i<phrase.length();i++)

{ *//for loop to run for the length of the array*

**if**(isdigit(phrase[i])==false) *//check if the element is not a number*

**return** false; *//if its not return false*

}

**return** true; *//if it is return true*

}

int main()

{

int listenSocket, connectSocket, i;

unsigned short int listenPort;

socklen\_t clientAddressLength;

**struct** sockaddr\_in clientAddress, serverAddress;

char line[LINE\_ARRAY\_SIZE];

cout << "Enter port number to listen on (between 1500 and 65000): ";

cin >> listenPort;

*// Create socket for listening for client connection*

*// requests.*

listenSocket = socket(AF\_INET, SOCK\_STREAM, 0);

**if** (listenSocket < 0) {

cerr << "cannot create listen socket";

exit(1);

}

*// Bind listen socket to listen port. First set various*

*// fields in the serverAddress structure, then call*

*// bind().*

*// htonl() and htons() convert long integers and short*

*// integers (respectively) from host byte order (on x86*

*// this is Least Significant Byte first) to network byte*

*// order (Most Significant Byte first).*

serverAddress.sin\_family = AF\_INET;

serverAddress.sin\_addr.s\_addr = htonl(INADDR\_ANY);

serverAddress.sin\_port = htons(listenPort);

**if** (bind(listenSocket,

(**struct** sockaddr \*) &serverAddress,

**sizeof**(serverAddress)) < 0) {

cerr << "cannot bind socket";

exit(1);

}

*// Wait for connections from clients. This is a*

*// non-blocking call; i.e., it registers this program with*

*// the system as expecting connections on this socket, and*

*// then this thread of execution continues on.*

listen(listenSocket, 5);

**while** (1) {

string phrase="";

string NaN="";

string name="";

string Type="";

vector<float> QTY;

vector<string> det;

vector<string> savednames;

*//vector<string> types = {"saturday", "VIP", "sunday", "weekend"};*

*//vector<float> aprice = {25.0, 50.0, 10.0, 30.0};*

*//vector<float> cprice = {20.0, 25.0, 7.5, 22.0};*

float e\_cost = 0.0;

float a\_cost = 0.0;

float total = 0.0;

int number=0;

int adults=0;

int childs=0;

bool isanum=false;

cout << "Waiting for TCP connection on port " << listenPort << " ...**\n**";

*// Accept a connection with a client that is requesting*

*// one. The accept() call is a blocking call; i.e., this*

*// thread of execution stops until a connection comes*

*// in. connectSocket is a new socket that the system*

*// provides, separate from listenSocket. We \*could\**

*// accept more connections on listenSocket, before*

*// connectSocket is closed, but this program doesn't do*

*// that.*

clientAddressLength = **sizeof**(clientAddress);

connectSocket = accept(listenSocket,

(**struct** sockaddr \*) &clientAddress,

&clientAddressLength);

**if** (connectSocket < 0) {

cerr << "cannot accept connection ";

exit(1);

}

*// Show the IP address of the client.*

*// inet\_ntoa() converts an IP address from binary form to the*

*// standard "numbers and dots" notation.*

cout << " connected to " << inet\_ntoa(clientAddress.sin\_addr);

*// Show the client's port number.*

*// ntohs() converts a short int from network byte order (which is*

*// Most Significant Byte first) to host byte order (which on x86,*

*// for example, is Least Significant Byte first).*

cout << ":" << ntohs(clientAddress.sin\_port) << "**\n**";

*// Read lines from socket, using recv(), storing them in the line*

*// array. If no messages are currently available, recv() blocks*

*// until one arrives.*

*// First set line to all zeroes, so we'll know where the end of*

*// the string is.*

memset(line, 0x0, LINE\_ARRAY\_SIZE);

**while** (recv(connectSocket, line, MAX\_MSG, 0) > 0) {

cout << " -- " << line << "**\n**";

*// Convert line to upper case.*

*// Split sentence into phrases*

**for** (i = 0; line[i] != '\0'; i++)

{

**if** ((line[i] != '.') && (line[i] != ' '))

{

phrase=phrase+line[i];

}

**if** ((line[i] == ' ')|| (line[i]=='.'))

{

isanum=isnum(phrase);

**if** (isanum)

{

number = stoi(phrase);

QTY.push\_back(number);

}

**else**

{

NaN=phrase;

det.push\_back(NaN);

}

phrase="";

}

}

*/\*for(int i=0; i<4; i++)*

*{*

*if (det[1] == types[i])*

*{*

*float aprice = aprice[i]; //related price variables*

*float cprice = cprice[i];*

*float acost = QTY[0] \* aprice; //create cost based on QTY and type*

*float ccost = QTY[1] \* cprice;*

*}*

*else*

*{*

*cout << "Wrong type." << endl;*

*}*

*}\*/*

**if** (det[1] == "saturday") *//check what type ticket user is booking*

{

float aprice = 25; *//related price variables as in brief*

float cprice = 20;

float acost=QTY[0] \* aprice; *//create cost based on QTY and type*

float ccost=QTY[1] \* cprice;

e\_cost = acost+ccost;

}

**else** **if** (det[1] == "vip")

{

float aprice = 50;

float cprice = 25;

float acost=QTY[0] \* aprice;

float ccost=QTY[1] \* cprice;

e\_cost = acost+ccost;

}

**else** **if** (det[1] == "sunday")

{

float aprice = 10;

float cprice = 7.5;

float acost=QTY[0] \* aprice;

float ccost=QTY[1] \* cprice;

e\_cost = acost+ccost;

}

**else** **if** (det[1] == "weekend")

{

float aprice = 30;

float cprice = 22;

float acost=QTY[0] \* aprice;

float ccost=QTY[1] \* cprice;

e\_cost = acost+ccost;

}

*//same principle is here but with activities*

**if** (det[2] == "baking")

{

float aprice = 8;

float cprice = 5;

float acost=QTY[2] \* aprice;

float ccost=QTY[3] \* cprice;

a\_cost = acost+ccost;

}

**else** **if** (det[2] == "dancing")

{

float aprice = 15;

float cprice = 10;

float acost=QTY[2] \* aprice;

float ccost=QTY[3] \* cprice;

a\_cost = acost+ccost;

}

**else** **if** (det[2] == "craft")

{

float aprice = 10;

float cprice = 7;

float acost=QTY[2] \* aprice;

float ccost=QTY[3] \* cprice;

a\_cost = acost+ccost;

}

**else** **if** (det[2] == "disco")

{

float aprice = 15;

float cprice = 11;

float acost=QTY[2] \* aprice;

float ccost=QTY[3] \* cprice;

a\_cost = acost+ccost;

}

*//else if (det[2] == "N/A")*

*//{*

*// a\_cost = 0.0;*

*//}*

savednames.push\_back(det[0]); *//saves last user into det*

total = e\_cost + a\_cost; *//get total cost*

**for**(i=0;i<savednames.size()-1;i++){ *//iterate over users to check if there is a booking*

**if** (det[0]==savednames[i]){

total=0.0; *//set total to be 0 to prevent making any more bookings*

}

};

det.clear(); *//clear vectors in order to prepare for next user*

QTY.clear();

a\_cost = 0.0; *//clear prices for the next user*

e\_cost = 0.0;

sprintf(line," The cost of the booking is: %.2f £", total);

**if** (send(connectSocket, line, strlen(line) + 1, 0) < 0) *//send info to client*

cerr << "Error: cannot send modified data";

memset(line, 0x0, LINE\_ARRAY\_SIZE); *// set line to all zeros*

total= 0.0;

}

}

}

## 

**//**This is TCP-CLIENT

#include <netdb.h>

#include <netinet/in.h>

#include <unistd.h>

#include <iostream>

#include <cstring>

#include <stdlib.h>

#define MAX\_LINE 100

#define LINE\_ARRAY\_SIZE (MAX\_LINE+1)

**using** **namespace** std;

int main()

{

int socketDescriptor;

unsigned short int serverPort;

**struct** sockaddr\_in serverAddress;

**struct** hostent \*hostInfo;

char buf[LINE\_ARRAY\_SIZE], c;

cout << "Enter server host name or IP address: ";

cin.get(buf, MAX\_LINE, '\n');

*// gethostbyname() takes a host name or ip address in "numbers and*

*// dots" notation, and returns a pointer to a hostent structure,*

*// which we'll need later. It's not important for us what this*

*// structure is actually composed of.*

hostInfo = gethostbyname(buf);

**if** (hostInfo == NULL) {

cout << "problem interpreting host: " << buf << "**\n**";

exit(1);

}

cout << "Enter server port number: ";

cin >> serverPort;

cin.get(c); *// dispose of the newline*

*// Create a socket. "AF\_INET" means it will use the IPv4 protocol.*

*// "SOCK\_STREAM" means it will be a reliable connection (i.e., TCP;*

*// for UDP use SOCK\_DGRAM), and I'm not sure what the 0 for the last*

*// parameter means, but it seems to work.*

socketDescriptor = socket(AF\_INET, SOCK\_STREAM, 0);

**if** (socketDescriptor < 0) {

cerr << "cannot create socket**\n**";

exit(1);

}

*// Connect to server. First we have to set some fields in the*

*// serverAddress structure. The system will assign me an arbitrary*

*// local port that is not in use.*

serverAddress.sin\_family = hostInfo->h\_addrtype;

memcpy((char \*) &serverAddress.sin\_addr.s\_addr,

hostInfo->h\_addr\_list[0], hostInfo->h\_length);

serverAddress.sin\_port = htons(serverPort);

**if** (connect(socketDescriptor,

(**struct** sockaddr \*) &serverAddress,

**sizeof**(serverAddress)) < 0) {

cerr << "cannot connect**\n**";

exit(1);

}

*//cout statements main screen*

cout << "**\n**Ticket booking system";

cout << "**\n**Available types: **\n** saturday **\n** VIP **\n** sunday **\n** weekend.**\n**Please type in lowercase.";

cout << "**\n**Available activities: **\n** baking **\n** dancing **\n** craft **\n** disco**\n**Please type in lowercase.";

cout << "**\n**Please type it in the format [Name] [Type of Ticket] [Quantity of Adult’s tickets] [Quantity of Children’s Tickets] [Name] [Type of Activity] [Quantity of Adult’s tickets] [Quantity of Children’s Tickets]";

*//cout << "\nIf you do not wish to book any activities then please put N/A";*

cout << "**\n**only the first " << MAX\_LINE << " characters will be used.**\n\n**";

cout << "**\n**Please put the dot at the end. **\n**";

*// Prompt the user for input, then read in the input, up to MAX\_LINE*

*// charactars, and then dispose of the rest of the line, including*

*// the newline character.*

cout << "Input: ";

cin.get(buf, MAX\_LINE, '\n');

**while** (cin.get(c) && c != '\n')

; *//Loop does nothing except consume the spare bytes*

*// Stop when the user inputs a line with just a dot.*

**while** (strcmp(buf, ".")) { *//strcmp returns 0 when the two strings*

*//are the same, so this continues when*

*//they are differenta*

*// Send the line to the server.*

**if** (send(socketDescriptor, buf, strlen(buf) + 1, 0) < 0) {

cerr << "cannot send data ";

close(socketDescriptor); *//Note this is just like using files...*

exit(1);

}

*// Zero out the buffer.*

memset(buf, 0x0, LINE\_ARRAY\_SIZE);

*// Read the modified line back from the server.*

**if** (recv(socketDescriptor, buf, MAX\_LINE, 0) < 0) {

cerr << "didn't get response from server?";

close(socketDescriptor);

exit(1);

}

*// Inform the user of successful booking*

cout << "You have successfully booked tickets, total cost: " << buf << "**\n**If your total is 0.0, tickets might not be available**\n**";

*// Prompt the user for input, then read in the input, up to MAX\_LINE*

*// charactars, and then dispose of the rest of the line, including*

*// the newline character. As above.*

cout << "Input: ";

cin.get(buf, MAX\_LINE, '\n');

**while** (cin.get(c) && c != '\n')

; *//Chomp chomp chomp*

}

close(socketDescriptor);

**return** 0;

}

## **Multiple Examples of Code Running**

Text

Description automatically generated

That would be enough to produce end output

# **Conclusion and Reflection**

To conclude, there were ACL approaches listed, assembly game created and TCP booking system made with C++, assembly game lacks prize feature, TCP system may produce wrong outputs to the user, overall there is functionality and demonstrated knowledge.

# **References**

Gargaro, D. (2021, July 20). *The pros and cons of facial recognition technology*. IT PRO. Retrieved 7 April 2022, from https://www.itpro.co.uk/security/privacy/356882/the-pros-and-cons-of-facial-recognition-technology#:%7E:text=As%20with%20any%20technology%2C%20there,to%20flaws%20in%20the%20technology.

Kaspersky. (2022, February 9). *What is Biometrics? How is it used in security?* Www.Kaspersky.Com. Retrieved 7 April 2022, from https://www.kaspersky.com/resource-center/definitions/biometrics

Wikipedia contributors. (2022, April 5). *Touch ID*. Wikipedia. Retrieved 7 April 2022, from https://en.wikipedia.org/wiki/Touch\_ID#References