



UTM
UNIVERSITI TEKNOLOGI MALAYSIA

SECR2033 SEC09
COMPUTER ORGANIZATION & ARCHITECTURE
GROUP PROJECT

LECTURER: MR AHMAD FARIZ BIN ALI

NO.	GROUP MEMBER	MATRIC NO.
1	NG JING ER (LEADER)	A19EC0115
2	SHAFIA CARINA SAPTOMO	A19EC0286
3	VIMALRAJ A/L SIVARAJOO	A19EC0213

TABLE OF CONTENTS

TOPIC	PAGE
INTRODUCTION	1
MEMBER RESPONSIBILITIES	2
EXAMPLE OF INPUTS & OUTPUTS	3-5
DISCUSSION	6-7
WEAKNESS OF THE PROGRAM	8
CONCLUSION	9
REFERENCE	10
APPENDIX - CODE	11-15

INTRODUCTION

In this project, we have tried to apply the knowledge of assembly language that we learnt in the subject computer organization and architecture to complete all the tasks required. The main task of this project is required students to develop a simulation program of a vending machine with different items. We have designed a doughnut vending machine with different flavours and different prices. We assigned prices according to their flavours as shown in the table below:

Item #	Flavour	Price / cents
1	Original Glazed	250
2	Chocolate Kreme Cake With Oreo	300
3	Caramel Kreme Crunch	200
4	Chocolate Iced Glazed With Sprinkles	200

The details about the assembly language such as the assembly code or instruction that was used in the program we designed will be explained more at the discussion part. And the example of input and output will be performed in the form of a screenshot to show the results of our program with different inputs.

MEMBER RESPONSIBILITIES

All members participated in the discussion and gave their own opinion but were also willing to listen to others' suggestions.

No.	Members	Tasks
1	Ng Jing Er (Leader)	Program: ● Design a program according to the requirements. Report: ● Discussion ● Introduction
2	Shafia Carina Saptomo	Program: ● Design a program according to the requirements Report: ● Discussion ● Conclusion
3	Vimalraj a/l Sivarajoo	Video: ● Record and edit a demo video. Report: ● Example of input and output ● Discussion part

EXAMPLE OF INPUTS & OUTPUTS

Here is the print screen when we build project and debug:

1. Example of total money inserted is enough

```
*****
*      1. ORIGINAL GLAZED                250 cents      *
*      2. CHOCOLATE KREME CAKE WITH OREO  300 cents      *
*      3. CARAMEL KREME CRUNCH            200 cents      *
*      4. CHOCOLATE ICED GLAZED WITH SPRINKLES 200 cents  *
*****
Please input money :
10 cents: 10

20 cents: 10

50 cents: 2

Total Money Inserted: 400 cents
Please choose item : 2
Here is your      : CHOCOLATE KREME CAKE WITH OREO FLAVOUR DOUGHNUT
This is your change : 100 cents

Do you want to add item? (Yes:0 / No: 1) : 0

*****
*      1. ORIGINAL GLAZED                250 cents      *
*      2. CHOCOLATE KREME CAKE WITH OREO  300 cents      *
*      3. CARAMEL KREME CRUNCH            200 cents      *
*      4. CHOCOLATE ICED GLAZED WITH SPRINKLES 200 cents  *
*****
Please input money :
10 cents:
```

2. Example of total money inserted is not enough

```
*****
*      1. ORIGINAL GLAZED                250 cents      *
*      2. CHOCOLATE KREME CAKE WITH OREO  300 cents      *
*      3. CARAMEL KREME CRUNCH            200 cents      *
*      4. CHOCOLATE ICED GLAZED WITH SPRINKLES 200 cents  *
*****
Please input money :
10 cents: 1

20 cents: 2

50 cents: 3

Total Money Inserted: 200 cents
Please choose item : 1
MONEY YOU ENTERED IS NOT ENOUGH !!! PLEASE TRY AGAIN !

Please input money :
10 cents: 1

20 cents: 2

50 cents: 3

Total Money Inserted: 200 cents
Please choose item : 3
Here is your      : CARAMEL KREME CRUNCH FLAVOUR DOUGHNUT
This is your change : 0 cents

Do you want to add item? (Yes:0 / No: 1) : 0
```

3. Example to continue/ stop the program

```
*****
*      1. ORIGINAL GLAZED                250 cents      *
*      2. CHOCOLATE KREME CAKE WITH OREO  300 cents      *
*      3. CARAMEL KREME CRUNCH            200 cents      *
*      4. CHOCOLATE ICED GLAZED WITH SPRINKLES 200 cents  *
*****
Please input money :
10 cents: 10

20 cents: 5

50 cents: 2

Total Money Inserted: 300 cents
Please choose item : 4
Here is your      : CHOCOLATE ICED GLAZED WITH SPRINKLES FLAVOUR DOUGHNUT
This is your change : 100 cents

Do you want to add item? (Yes:0 / No: 1) : 0

*****
*      1. ORIGINAL GLAZED                250 cents      *
*      2. CHOCOLATE KREME CAKE WITH OREO  300 cents      *
*      3. CARAMEL KREME CRUNCH            200 cents      *
*      4. CHOCOLATE ICED GLAZED WITH SPRINKLES 200 cents  *
*****
Please input money :
10 cents: 0

20 cents: 5

50 cents: 4

Total Money Inserted: 300 cents
Please choose item : 2
Here is your      : CHOCOLATE KREME CAKE WITH OREO FLAVOUR DOUGHNUT
This is your change : 0 cents

Do you want to add item? (Yes:0 / No: 1) : 1

Thank you! Press any key to continue . . .
```

DISCUSSION

Visual Studio requires assembly language source files to belong to a project, which is a kind of a box container. A project holds configuration information such as the locations of the assembler, linker, and required libraries. A project has its folder, and it holds the names and locations of all files belonging to it. This project uses the project template in the form *.asm* given by our lecturer from the beginning of learning assembly language. We are required to use the Irvine Link Library. The Irvine link library contains several useful routines to input data, output data, and perform several tasks that one would normally have to use many operating systems calls to complete. This library is called Irvine. Lib and should have been installed when you installed the CD-ROM from the Irvine book. We only use a few basic things here. Here are list some procedures in the Irvine32 and Irvine16 libraries that we used in making this project:

<i>Clrscr</i>	Clears the screen, moves the cursor to the upper-left corner.
<i>Crlf</i>	Writes a carriage return / linefeed to the display.
<i>Mov</i>	Copies a byte or word from a source operand to a destination operand.
<i>Writestring</i>	Write a null-terminated string. Input: EDI points to the strings offset
<i>Mul</i>	Multiplies AL, AX, or EAX by a source operand.
<i>ReadDec</i>	Reads an unsigned 32-bit decimal integer from the keyboard, terminated by the Enter key.
<i>WriteDec</i>	Writes an unsigned 32-bit integer to the console window in decimal format.
<i>Cmp</i>	Compares the destination to the source by performing an implied subtraction of the source from the destination.
<i>Jmp</i>	<i>(Jump Unconditionally)</i> Jump to a code label.
<i>Jz</i>	<i>(Conditional Jump)</i> Jump Equal or Jump Zero.
<i>Jge</i>	<i>(Conditional Jump)</i> Jump Greater/Equal or Jump Not Less.
<i>Jl</i>	<i>(Conditional Jump)</i> Jump Less or Jump Not Greater/Equal.

The loop instruction is frequently used in our project such as L1, L2, FLAV, F1-F4 , Cal1-Cal4 and so on. The functions of each loop used in the program are shown in the table below:

Loop	Function
L1	Display the menu
L2	Input the total number of money to be inserted according to the value of money
FLAV	Input an item that choose from the menu
F1-F4	F1 to F4 denoted as the flavours of doughnut as shown in the item banner. Compare the total of money inserted, the instruction will jump to the loop SRY if the money inserted is not enough. Compare the item chosen so that the instruction will jump to the corresponding loop for calculation (Cal1-Cal4) .
Cal1-Cal4	Calculate the change after the input of the item chosen.
ASK	Ask the user whether to continue the program or quit.
CONTINUE	Continue the program by restart from the loop L1.
SRY	Display a string showing the money inserted is not enough. The instruction will jump back to L2 and ask the user to input again for the money.
STOP	Stop the program.

WEAKNESS OF THE PROGRAM

The weakness of the coding we made is that it is too long and simple. We are in our first year degree program for the software course and still learning. Much efficient ways or shortcuts for a much simpler coding may be learnt over time through experience. But even so, this code still runs according to the instructions following this project and uses assembly language that we have learned during class sessions.

In addition, one of the problems that we faced while designing the code is that we hope to enable the user to add as many items as possible if the changes are enough for the next items. Example if the user inserted a total of 500 cents. The first item chosen is original glazed flavour doughnut (250 cents). And the change is 250 cents, which should be possible to add another item that is below 250 cents. So this is the part that we failed to add into our program. And we think that the code would be more perfect if this part is included.

CONCLUSION

All the group members show the spirit of teamwork throughout the whole process of discussion until the project successfully completed. There were few problems we had to face throughout the long-distance discussion. It is a period of pandemic and we did face a crisis. Everyone stayed home as per order and went through online classes. Communication was not as fluent and without face to face interactions it was hard to understand each other sometimes. But all the members were able to get their work done on time.

In this project, we have applied our knowledge that we learnt from this course Computer Organization and Architecture to design a simple vending machine program. However, although we completed the project on time, there were still a few weaknesses in our project. Such as the program may be relatively simple without much techniques or specific assembly language we learnt. Even though the program is able to display the output that we desired, we still found that the program is too simple. Due to the limited time left, we are only able to come out with a very simple program. And most of us are still not comfortable with the assembly language but we have tried our best to finish the project on time.

Last but not least, we have a basic understanding about the terminologies of Computer Programming and the concept of Assembly programming. And it is very important to have a strong basic knowledge on assembly programming as a programmer. Since computer programming languages are often used for development of software programs and also applications, we should not underestimate the importance of assembly language.

REFERENCES

1. https://www.tutorialspoint.com/assembly_programming/assembly_conditions.htm
2. <http://math.uaa.alaska.edu/~afkjm/cs221/handouts/procedures.pdf>

APPENDIX-CODE

```
TITLE MASM Template                                (main.asm)

; Description: PROJECT GROUP: THE3

INCLUDE Irvine32.inc
.data

D1 BYTE "*****" ,0
D2 BYTE "*"      1. ORIGINAL GLAZED                250 cents    "*" ,0
D3 BYTE "*"      2. CHOCOLATE KREME CAKE WITH OREO    300 cents    "*" ,0
D4 BYTE "*"      3. CARAMEL KREME CRUNCH              200 cents    "*" ,0
D5 BYTE "*"      4. CHOCOLATE ICED GLAZED WITH SPRINKLES 200 cents    "*" ,0

item1 BYTE "Here is your          : ORIGINAL GLAZED FLAVOUR DOUGHNUT",0
item2 BYTE "Here is your          : CHOCOLATE KREME CAKE WITH OREO FLAVOUR
DOUGHNUT",0
item3 BYTE "Here is your          : CARAMEL KREME CRUNCH FLAVOUR DOUGHNUT",0
item4 BYTE "Here is your          : CHOCOLATE ICED GLAZED WITH SPRINKLES FLAVOUR
DOUGHNUT",0

str1 BYTE "Please input money : ",0
str2 BYTE "10 cents: ",0
str3 BYTE "20 cents: ",0
str4 BYTE "50 cents: ",0
str5 BYTE "Total Money Inserted: ",0
str6 BYTE "Please choose item : ",0
str7 BYTE "This is your change : ",0
str8 BYTE "Thank you! ", 0
q1 BYTE "Do you want to add item? (Yes:0 / No: 1) : ",0
sry BYTE "MONEY YOU ENTERED IS NOT ENOUGH !!! PLEASE TRY AGAIN !",0
cent BYTE " cents",0

flavour DWORD 0
input1 DWORD 0
input2 DWORD 0
input3 DWORD 0

total DWORD 0
change DWORD 0
choice DWORD 0

ten DWORD 10
twenty DWORD 20
fifty DWORD 50

cost1 DWORD 250
cost2 DWORD 300
cost3 DWORD 200
cost4 DWORD 200

.code
main PROC

    call Clrscr

L1:
    mov edx, OFFSET D1
    call WriteString
    call crlf

    mov edx, OFFSET D2
    call WriteString
    call crlf
```

```

mov edx, OFFSET D3
call WriteString
call crlf
mov edx, OFFSET D4
call WriteString
call crlf
mov edx, OFFSET D5
call WriteString
call crlf

```

```

mov edx, OFFSET D1
call WriteString
call crlf
jmp L2

```

L2:

```

mov edx, OFFSET str1
call WriteString
call crlf

```

```

mov edx, OFFSET str2
call WriteString
call ReadDec
mov input1, eax

```

```

mov eax, input1
mul ten
mov total, eax
call crlf

```

```

mov edx, OFFSET str3
call WriteString
call ReadDec
mov input2, eax

```

```

mov eax, input2
mul twenty
add total, eax
call crlf

```

```

mov edx, OFFSET str4
call WriteString
call ReadDec
mov input3, eax

```

```

mov eax, input3
mul fifty
add total, eax
call crlf

```

```

mov edx, OFFSET str5
call WriteString
mov eax, total
call WriteDec
mov edx, OFFSET cent
call WriteString
call crlf
jmp FLAV

```

FLAV:

```

mov edx, OFFSET str6
call WriteString
call ReadDec
mov flavour, eax

cmp flavour, 1

```

```

        jz F1
        cmp flavour,2
        jz F2
        cmp flavour,3
        jz F3
        cmp flavour,4
        jz F4

        jmp FLAV

F1:
        cmp total, 250
        jge Cal1
        jl SR

Cal1:
        mov edx, OFFSET item1
        call WriteString
        call crlf
        mov eax, total
        sub eax, cost1
        mov change, eax

        mov edx, OFFSET str7
        call WriteString
        mov eax, change
        call WriteDec
        mov edx, OFFSET cent
        call WriteString
        call crlf

        jmp ASK

F2:
        cmp total, 300
        jge Cal2
        jl SR

Cal2:
        mov edx, OFFSET item2
        call WriteString
        call crlf
        mov eax, total
        sub eax, cost2
        mov change, eax

        mov edx, OFFSET str7
        call WriteString
        mov eax, change
        call WriteDec
        mov edx, OFFSET cent
        call WriteString
        call crlf

        jmp ASK

F3:
        cmp total, 200
        jge Cal3
        jl SR

Cal3:
        mov edx, OFFSET item3
        call WriteString
        call crlf

```

```

    mov eax, total
    sub eax, cost3
    mov change, eax

    mov edx, OFFSET str7
    call WriteString
    mov eax, change
    call WriteDec
    mov edx, OFFSET cent
    call WriteString
    call crlf

    jmp ASK

F4:
    cmp total, 200
    jge Cal4
    jl SR

Cal4:
    mov edx, OFFSET item4
    call WriteString
    call crlf
    mov eax, total
    sub eax, cost4
    mov change, eax

    mov edx, OFFSET str7
    call WriteString
    mov eax, change
    call WriteDec
    mov edx, OFFSET cent
    call WriteString
    call crlf

    jmp ASK

ASK:
    call crlf
    mov edx, OFFSET q1
    call WriteString
    call ReadDec
    mov choice, eax

    call crlf
    call crlf

    cmp choice, 0
    jz Continue
    jmp Stop

Continue:
    jmp L1

SR:
    mov edx, OFFSET sry
    call WriteString
    call crlf
    call crlf
    JMP L2

Stop:
    mov edx, OFFSET str8
    call WriteString
    call crlf

```



```
exit  
main ENDP  
  
END main
```