Calculator program

CS1021 FINAL PROJECT Full 2022

Dr. Mohmmad Nauman

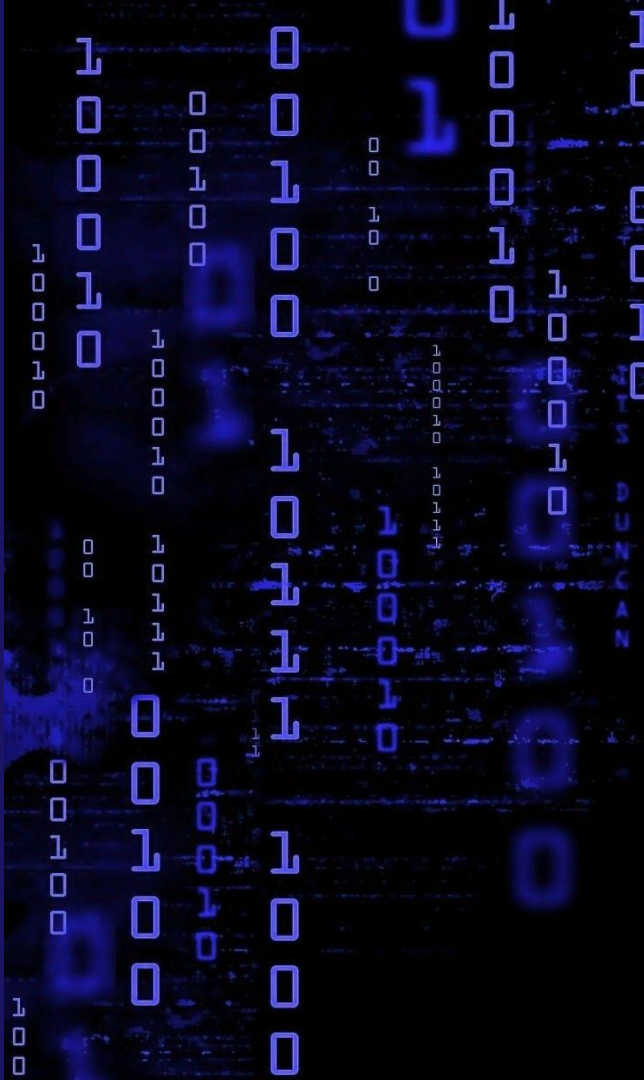
Outlines:

- Introduction
- C language code
- Assembly language code
- Link C with Assembly code
- Execution of the program
- Conclusion

Introduction

Introduction

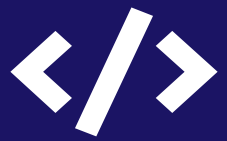
The objective of this project is to write some basic assembly language code and link it to a high-level language through a virtual operating system.



Calculator program

Our program is a simple calculator, was created using the C programming language. It used for four operations (addition, subtraction, multiplication, and division) mathematical operations in this application.

C code



Function(1)

```
int addition(int num1, int num2, int res)
{
    int val;
    res = num1 + num2;
    val = add(res);
    printf("%d",val);
    printf("\n");
}
```

Function(2)

```
int subtraction(int num1, int num2, int res)
{
    int val;
    res = num1 - num2;
    val = sub(res);
    printf("%d",val);
    printf("\n");
}
```

Function(3)

```
int multiplication(int num1, int num2, int res)
{
    int val;
    res = num1 * num2;
    val = mul(res);
    printf("%d",val);
    printf("\n");
}
```

Function(4)

```
int division(int num1, int num2, int res)
{
    int val;
    res = num1 / num2;
    val = div(res);
    printf("%d",val);
    printf("\n");
}
```

Main function

```c
int main ()
{
  char op;//operator
  int num1, num2, res ;
  double result = 0.0;

  printf ("WELCOME TO OUR SIMPLE CALCULATOR\n");
  printf ("---------------------------\n");

    printf("Enter [Enter numbers with operation [+, -, *, /] \n");
    scanf ("%d %c %d", &num1, &op, &num2);

```

Main function

```
    if (op == '+' )
    {
        addition(num1, num2 , res);
    }
    else if (op == '-')
    {
        subtraction(num1, num2 , res);
     }

    else if (op == '*')
    {
        multiplication(num1, num2, res);
    }
        else if (op == '/')
    {
        division(num1, num2, res);
    }
        else
    {
        printf("Invalid operator");
    }

  return 0;
}
```

Assembly code

```
; SECTION .DATA

    msg:    db'Enter an operation: ',10
    msgLen: equ $-msg

    msg2:   db'Result is: ',10
    msg2Len: equ $-msg2


  ; SECTION .TEXT
```

```
GLOBAL add
add:
     mov  rax, rdi
     push rax

     mov eax, 4
     mov ebx, 1
     mov ecx, msg2
     mov edx, msg2Len

     int 80h

     pop  rax
     ret
```

```
GLOBAL sub
sub:
     mov  rax, rdi
     push rax

     mov eax, 4
     mov ebx, 1
     mov ecx, msg2
     mov edx, msg2Len

     int 80h

     pop  rax
     ret
```

```
GLOBAL mul
mul:
     mov  rax, rdi
     push rax

     mov eax, 4
     mov ebx, 1
     mov ecx, msg2
     mov edx, msg2Len

     int 80h

     pop  rax
     ret
```

```
GLOBAL div
div:
     mov  rax, rdi
     push rax

     mov eax, 4
     mov ebx, 1
     mov ecx, msg2
     mov edx, msg2Len

     int 80h

     pop  rax
     ret
```

Link

Kali Linux

To connect the C programming code with the assembly code and execute our program, we used the virtual operating system Kali Linux.

Linking code

```
File  Actions  Edit  View  Help
┌──(kali㉿kali)-[~]
└─$ cd Desktop

┌──(kali㉿kali)-[~/Desktop]
└─$ ls
c1.asm  codec2.c

┌──(kali㉿kali)-[~/Desktop]
└─$ nasm -f elf64 c1.asm  -o codec2.o

┌──(kali㉿kali)-[~/Desktop]
└─$ gcc -no-pie codec2.c codec2.o -o calc
/usr/bin/ld: warning: codec2.o: missing .note.GNU-stack section implies executable stack
/usr/bin/ld: NOTE: This behaviour is deprecated and will be removed in a future version of the linker
```

Execution

```
┌──(kali㉿kali)-[~/Desktop]
└─$ ./calc
WELCOME TO OUR SIMPLE CALCULATOR
──────────────────────────────
Enter numbers with operation [+, -, *, /]
19+30
Result is:
49
```

Addition

```
┌──(kali㉿kali)-[~/Desktop]
└─$ ./calc
WELCOME TO OUR SIMPLE CALCULATOR
──────────────────────────────
Enter numbers with operation [+, -, *, /]
10*3
Result is:
30
```

Multiplication

```
┌──(kali㉿kali)-[~/Desktop]
└─$ ./calc
WELCOME TO OUR SIMPLE CALCULATOR
──────────────────────────────
Enter numbers with operation [+, -, *, /]
50-13
Result is:
37
```

Subtraction

```
┌──(kali㉿kali)-[~/Desktop]
└─$ ./calc
WELCOME TO OUR SIMPLE CALCULATOR
──────────────────────────────
Enter numbers with operation [+, -, *, /]
200/50
Result is:
4
```

Division

Conclusion

We didn't have a complete understanding at first, but we worked hard together and went deep to understand things thoroughly, and then everything became clear. It was a challenging project, but it was also interesting, and we enjoyed it.

THANKS!

Do you have any questions?

Group members

Araa Almarhabi	S20106395
Mawaddah Alagha	S20106707
Leen bajunaid	S21107285