**Assignment 2**

**COMPUTER ORGANIZATION AND ARCHITECTURE**

Group 14

1. 65015101 นายพนธกร พวงพะยอม 2. 65015137 นายวัฒชัย เตชะลือ

จากโจทย์

ให้แต่ละกลุ่มเขียนโปรแกรมเปรียบเทียบความเร็วในการทำงานระหว่างภาษา C กับ Assembly โดยใช้ฟังก์ชัน time จับเวลา โดยสร้างฟังก์ชันอะไรก็ได้ เช่น shi9 เพืOอคูณ 4 หรือ บวกเลข หรืออืOนๆ

#include <stdio.h> #include <time.h>

int main ()

}

time t seconds;

seconds = time(NULL);

printf("Seconds since January 1, 1970 = %ld\n", seconds);

return(0);

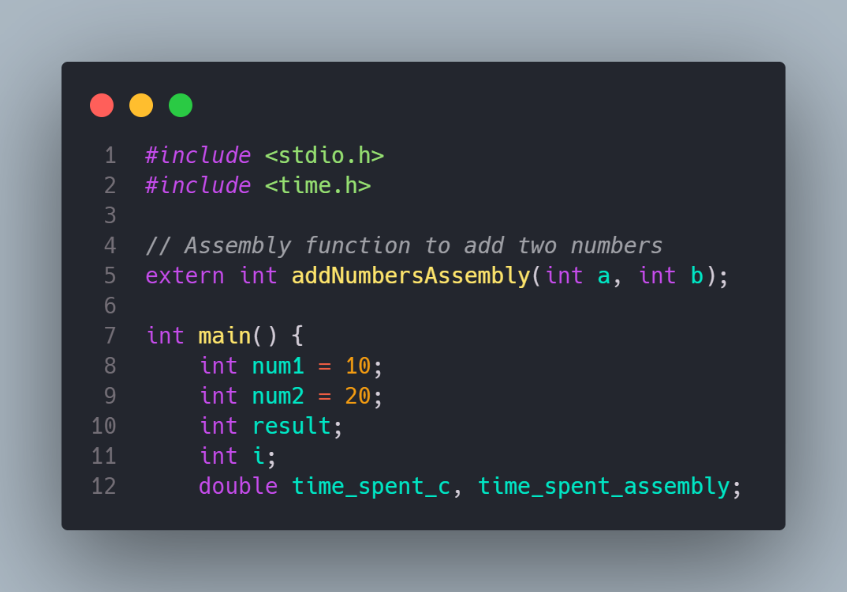
}

เวลาที่ได้จะเป็นจำนวนวินาทีตั้งแต่ January 1, 1970 ดังนั้นต้องอ่านค่า 2 ครั้งมาลบกัน ครั้งแรกก่อนทำ และครั้งที่ 2 หลัง ทำ และเพื่อให้เกิดจำนวนเวลามากๆ ให้วนลูปทำหลาย ๆ ครั้งเช่น 1 ล้านรอบ การแสดงผลให้แสดงผล 2 ค่า คือ เวลาที่ใช้ เมื่อใช้ภาษา C และเวลาที่ใช้เมื่อใช้ภาษา Assembly

ตัวอย่างโคด



การทำงานของโคด



#include <stdio.h>

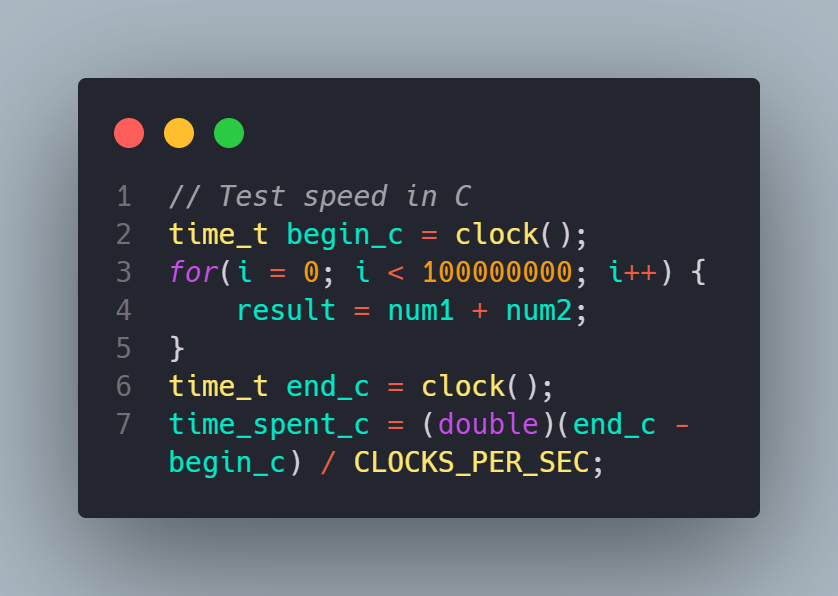
จะเก็บคำสั่งเกี่ยวกับการรับค่า แสดงค่าของโปรแกรม

#include <time.h>

รวมตัวแปรและฟังก์ชันที่ใช้สำหรับทำงานและ

จัดการเกี่ยวกับเรื่องวันที่และเวลา (date & time)

จากนั้นสร้างตัวแปรต่างๆ ที่เราจะใช้งานในฟังก์ชัน

การทำงานภาษา C

เริ่มต้นตัวแปร begin\_c ประเภท time\_t และ

กำหนดค่าที่ส่งคืนโดยclock()ฟังก์ชัน ฟังก์ชัน นี้clock()

ใช้เพื่อวัดเวลาตัวประมวลผลที่ใช้โดยโปรแกรม

บรรทัด for loop ที่ดำเนินการเพิ่ม num1 + num2 เป็น

100,000,000 ครั้ง ผลลัพธ์ถูกกำหนดให้กับตัวแปร result

กำหนดค่าที่ส่งคืนที่ ฟังก์ชันclock()

CLOCKS\_PER\_SEC ซึ่งเป็นค่าคงที่ที่แสดงจำนวนสัญญาณนาฬิกาต่อวินาที รหัสจะคำนวณเวลาที่ใช้ในหน่วยวินาที ผลลัพธ์จะถูกกำหนดให้ time\_spent\_c

โดยสรุป โค้ดนี้วัดเวลาดำเนินการของลูปที่ดำเนินการเพิ่ม num1 + num2เป็น 100,000,000 ครั้ง และเก็บเวลาที่ใช้เป็นวินาทีในฟังก์ชัน time\_spent\_c

บรรทัดแรก คือให้ begin\_assembly ของ time\_t

และกำหนดฟังก์ชัน clock()

เช่นเดียวกับข้อมูลโค้ดก่อนหน้า การวนซ้ำนี้ดำเนินการเพิ่ม

addNumbersAssembly(num1, num2)

จำนวน 100,000,000 ครั้ง

หลังจากที่ทำการลูปเสร็จส่งค่าคืนที่ฟังก์ชัน clock()

ให้กับตัวแปร end\_assembly โดยจับเวลาตัวประมวลผล

จากนั้นในบรรทัดที่ time\_spent\_assembly = (double)(end\_assembly - begin\_assembly) / CLOCKS\_PER\_SEC; ก็คือ time\_spent\_assembly เป็นตัวแปรประเภท double และ end\_assembly, begin\_assembly แสดงถึงเวลาตัวประมวลผลที่ใช้โดยการใช้แอสเซมบลีหารความแตกต่างนี้ด้วย CLOCKS\_PER\_SEC หลังจากนั้นจะคำนวณเวลาที่ใช้ในหน่วยวินาทีและผลลัพธ์จะถูกกำหนดให้กับ time\_spent\_assembly และรองสองบรรทัดสุดท้ายก็ปริ้นค่าออกมา

**และเราจะหาผลลัพธ์ได้ยังไงล่ะ?** โดยที่กลุ่มผู้เรียนได้เขียนไฟล์แยก โดยการให้คำสั่ง extern int addNumbersAssembly(int a, int b); ในบรรทัดต้นๆของภาษา C โดยคีย์เวิร์ด extern ใช้เพื่อประกาศไฟล์ โดยมีการกำหนดการใช้งานไฟล์แอสเซมบลีของฟังก์ชัน addNumbersAssembly ในไฟล์ต้นฉบับแอสเซมบลีที่แยกต่างหากครับ

ผลการทำงานจะได้ดังนี้



สรุปผลการทำงานในการเขียนโคด การใช้ภาษา C มีความเร็ว  
 ในการทำงานมากว่า ภาษา Assembly