1. การนิยามฟังชัน

Return-value-type function-name(parameter){

Statement

Statement

.

.

.

}

1. Function prototype

Return-value-type function-name (parameter);

(parameter ไม่ต้องใส่ชื่อก็ได้)

1. Standard library กับ math library

Printf scanf getchar อยู่ใน standard library

Sqrt, exp, log อยู่ใน math library

ในการคอมไพล์จำเป็นต้องคอมไพล์รวมกับโค้ดของฟังชันพวกนี้ (gcc มีทำให้อยู่แล้ว)

ปกติ gcc จะโยงฟังชันใน standard library เองอัตโนมัติ โดยไม่ต้องระบุ option ข้างใน gcc แต่ฟังชัน math library ต้องโยงโดย –lm ตอนเรียกใช้ gcc

และหากต้องใช้ math library ก็ควร #include<math.h> ไว้ด้วย

1. Return value

ฟังชันจะส่งค่ากลับมา หรือไม่ส่งกลับมาหากประเภทข้อมูลที่ส่งเป็น void

หากโปรแกรมเจอ return ในฟังชันจะกลับไปทำงานต่อจากจุดที่เรียกใช้ฟังชันทันที

1. ใช้ฟังชันด้วย parameter

ค่าที่ได้จากฟังชันสามารถนำไปใส่เป็นพารามิเตอร์อีกฟังชันได้

1. Recursion

ฟังชันสามารถเรียกใช้ตนเองได้ โดยสามารถอยู่ในรูป direct หรือ indirect

เมื่อฟังชันเรียกใช้ตนเอง จะสร้างสำเนาของตัวมันเองมาอีกชุด แล้วตัวแปรในนั้นจะเป็นอิสระจากฟังชันที่สร้างมันขึ้นมา (ไม่ใช่ตัวแปรเดียวกัน)

1. Scope rule

ในบางบริเวณ สามารถเข้าถึงตัวแปรที่ถูกกำหนดขึ้นมาได้ และพอออกจากบริเวณก็ไม่สามารถเข้าถึงได้อีก

1. Local variable

ตัวแปรที่ถูกประกาศในฟังชันและ parameter ในฟังชัน ล้วนเป็นตัวแปร local ซึ่งสามารถใช้ได้ในฟังชันที่ประกาศตัวแปรนั้นขึ้นมาเท่านั้น

1. Global variable

ตัวแปรที่ประกาศไว้นอกฟังชัน สามารถใช้ได้ในทุกตัวแปร

หากตัวแปร local มีชื่อซ้ำกับตัวแปร global การแก้ไขในตัวแปร local จะไม่มีผลอะไรกับตัวแปร global

1. Storage classes

ผู้ใช้สามารถกำหนดชีวิตของตัวแปรด้วย 4 ลักษณะผ่าน keyword คือ auto, register, extern, static โดยสามารถแบ่งได้เป็น 2 ประเภทคือ automatic storage duration และ static storage duration

ตัวแปร auto กับ register เป็นประเภท automatic storage duration เป็นตัวแปรที่เมื่อโปรแกรมเข้าไปทำงาน block ที่มีตัวแปรเหล่านี้ถูกประกาศ เมื่อตัวแปรออกจาก block ตัวแปรพวกนี้จะไม่มีอยู่ใน memory อีกต่อไป

ทว่าตัวแปร local ก็เป็น automatic storage duration โดย default อยู่แล้ว keyword auto จึงไม่จำเป็น

ส่วน register เป็นการแจ้งคอมไพล์เลอร์ว่าต้องการเป็นข้อมูลไว้ใน register เพื่อความรวดเร็ว ดังนั้นหากระบบมี register เหลือก็จะจัดสรรพื้นที่ในนั้นให้

ส่วน extern กับ global จะเป็น static storage duration จะเป็นตัวแปรที่เมื่อถูกประกาศไปแล้วจะอยู่ในโปรแกรมถาวรไปจนกว่าโปรแกรมจะหยุดการทำงาน

ตัวแปร global และชื่อฟังชันจะเป็น extern โดย default

หากตัวแปร local ถูกประกาศเป็น static ก็จะยังเข้าถึงได้เพียงในฟังชันนั้นๆเท่านั้น แต่ข้อมูลก็จะยังคงอยู่ใน memory ไปจนกว่าจบการทำงาน

1. Dynamic storage duration
2. พื้นที่เก็บข้อมูลสำหรับ Dynamic storage duration สามารถูก allocate ขึ้นมาได้ตามความต้องการของโปรแกรมในขณะที่กำลังทำงานอยู่
3. การคอมไพล์แยกส่วน

ส่วนของการนิยามและเรียกใช้ฟังชันไม่จำเป็นต้องคอมไพล์พร้อมกัน และสามาถอยู่แยกกันคนละไฟล์ รวมถึงตัวแปร global และการเข้าถึงตัวแปรด้วย

การคอมไพล์

gcc –c main.c //จะสร้างไฟล์ main.o ขึ้น

gcc main.o func.o –o main.exe //จะลิงค์ไฟล์ main.o กับ func.o เข้าด้วยกัน

1. การใช้ make

ไฟล์ makefile จะบรรจุกฎต่างๆที่ใช้ในการสร้างอาธิเช่น object file หรือ executable file

การสร้างกฎ

Target: dependency

🡪tab 🡪 คำสั่งที่ใช้ในการสร้าง target

1. Random number generator

Rand() เป็นฟังชันใน standard library ใช้ในการสุ่มตัวเลขระหว่าง 0 และ RAND\_MAX ซึ่งเป็นค่าที่ถูกกำหนดไว้แล้วใน <stdlib.h>

ในโปรแกรมการเรียกใช้ฟังชัน rand() จะได้เลขจำนวนเต็มที่แตกต่างกันออกไป

โดยปกติ การรันในแต่ละครั้งจะสุ่มได้ชุดเลขสุ่มที่เหมือนกัน หากต้องการชุดเลขสุ่มที่แตกต่างออกไปให้ใช้ srand() เพื่อสร้าง seed ให้กับ random number generator ขึ้นมาไหม่ ก่อนที่จะใช้ rand() ต่อไป