517 101 Introduction to Computers

Introduction to C Programming (Lab 3)

*เอกสารอ้างอิง : 1. C Programming Language (2nd Edition), Brian W. Kernighan, Dennis M. Ritchie

ความรู้เกี่ยวกับภาษา C

ความเป็นมาของภาษา C

- ภาษา C ถูกพัฒนาขึ้นในปี 1970 ภายใน Bell Labs โดยการนำทีมของ Dennis Ritchie
- สาเหตุของการเกิดภาษา C คือ Bell Labs ต้องพัฒนาระบบปฏิบัติการ UNIX เพื่อใช้กับเครื่อง DEC
  - มี Assembly อยู่แต่ช้า และเป็นภาษาที่ต้องที่ทำความเข้าใจกับอุปกรณ์ชุดนั้นๆ
  - พัฒนาใหม่ เป็น ภาษา C โดยได้พัฒนามาจากภาษา **B** ที่ Ken Thomson เป็นผู้พัฒนาขึ้น

ภาษาซี

ประวัติความเป็นมา

ภาษา BCPL → **ภาษา B** → **ภาษา C**

Basic Combined Programming Language

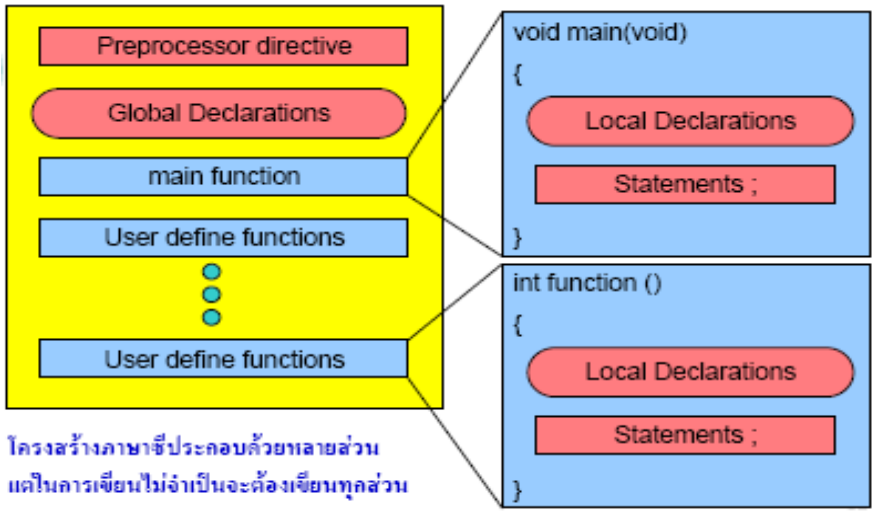
บนเครื่อง PDP-7 (UNIX) พ.ศ. 2513

พ.ศ. 2515 โดย เดนนิช ริทชี่

คุณลักษณะพิเศษของภาษา C

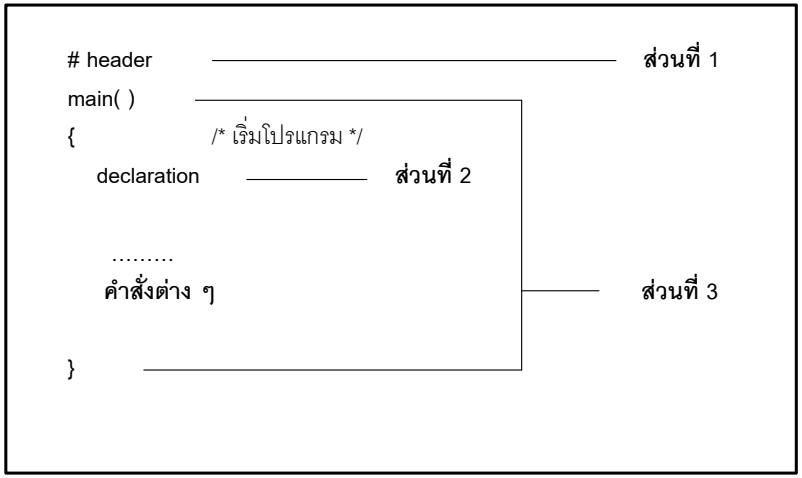
- เป็นภาษาที่มีลักษณะเป็นโครงสร้าง (Structured)
- การโปรแกรมแบบ **Procedural** (สั่งงานได้เร็วกว่าภาษาอื่น ยกเว้น ภาษาเครื่อง และ Assembly
- การติดต่อกับส่วนต่างๆ ของ Computer ทำได้ดีกว่าภาษาอื่นๆ (ยกเว้น ภาษาเครื่อง และ Assembly)
- การนำไปใช้กับระบบที่แตกต่างกันทำได้ง่าย (Portability)
- สามารถนำไปใช้ในการเขียนโปรแกรมประยุกต์ได้หลายระดับ เช่น OS, Compiler, Communication.
- สามารถกำหนดรูปแบบของข้อมูลได้อย่างกว้างขวาง

ลักษณะโครงสร้างของโปรแกรมภาษา C



โครงสร้างภาษาซีประกอบด้วยหลายส่วน
แต่ในการเขียนไม่จำเป็นจะต้องเขียนทุกส่วน

โครงสร้างพื้นฐานของภาษาซี



ส่วนประกอบของภาษา C

- โครงสร้างของภาษา C แบ่งออกได้เป็น 5 ส่วน คือ

1. ปรีโปรเซสเซอร์ (Preprocessor directive)
2. การกำหนดค่า (Global declaration)
3. ฟังก์ชั่นหลัก (Main function)
4. การสร้างและการใช้ฟังก์ชั่น(Uses-defined function)
5. ส่วนอธิบายโปรแกรม (Program comment)

ปรีโปรเซสเซอร์ (Preprocessor directive)

- ส่วนนี้เป็นส่วนต้นของโปรแกรมที่ต้องมี เพื่อเรียกใช้ไฟล์ต่างๆ ที่โปรแกรมต้องการในการทำงานในการกำหนดค่าต่างๆ
- ส่วนนี้จะเริ่มต้นด้วยเครื่องหมาย # ตามด้วยคำสั่งและชื่อไฟล์หรือค่าที่จะต้องกำหนด โดยไม่ต้องปิดท้ายด้วยเครื่องหมาย ; เช่น

  #include <stdio.h> เป็นการกำหนดว่าในโปรแกรมนี้ จะมีการใช้คำสั่งที่เกี่ยวกับการจัดการแฟ้มข้อมูลต่างๆ การพิมพ์ การแสดงผลลัพธ์ต่างๆ

  #define END 10 เป็นการกำหนดว่าในโปรแกรมนี้ จะมีการใช้คำสั่งที่เกี่ยวกับการจัดการจอภาพและการรับข้อมูลจากแป้นพิมพ์
- การที่ต้อง include ไฟล์นั้น เนื่องจากผู้ใช้ต้องการใช้งาน function สำเร็จรูปต่างๆที่ก่อนใช้งานต้องมีการประกาศค่าต่างๆ ก่อน เมื่อมีการ include ไฟล์เหล่านั้น ผู้ใช้ไม่จำเป็นต้องมาประกาศค่าต่างๆ ในโปรแกรมซ้ำอีก ทำให้ลดเวลาและสะดวกในการเขียนโปรแกรมมากขึ้น

การใช้ Preprocessor Directive

- ทุกโปรแกรมต้องมี
- ใช้เรียกไฟล์ที่โปรแกรมใช้ในการทำงานร่วมกัน
- ใช้กำหนดค่าคงที่ให้กับโปรแกรม
- เริ่มต้นด้วยเครื่องหมาย #
- ที่จะใช้มากมี 2 directives คือ
  - #include ใช้สำหรับเรียกไฟล์ที่โปรแกรมใชในการทำงาน
  - #define ใช้สำหรับกำหนดมาโครที่ให้กับโปรแกรม

การใช้ Preprocessor Directive : การใช้ #define

#define ชื่อ ค่าที่ต้องการ

ตัวอย่าง

```
#define START 10        (กำหนดค่า START = 10)
#define A 3*5/4         (กำหนดค่า A=3*5/4)
#define pi 3.14159      (กำหนดค่า pi = 3.14159)
#define sum(a,b) a+b
        (กำหนดค่า sum(ตัวแปรที่1, ตัวแปรที่2) = ตัวแปรที่1+ตัวแปรที่2
```


การกำหนดค่า (Global declaration)

- ส่วนนี้ใช้ในการประกาศตัวแปร หรือชื่อฟังก์ชั่นต่างๆ ที่จะใช้ในโปรแกรม
- ทุกๆ ส่วนของโปรแกรมจะสามารถเรียกใช้ข้อมูลที่ประกาศไว้ในส่วนนี้ได้ เช่น
  - int j,k,l; เป็นการประกาศตัวแปลแบบ global ทำให้ส่วนอื่นๆ ของโปรแกรมสามารถเรียกใช้ตัวแปรเหล่านี้ได้
  - void samplefunc(); เป็นการประกาศเฉพาะชื่อของฟังก์ชั่น เพื่อให้ส่วนอื่นๆของโปรแกรมสามารถเรียกใช้ฟังก์ชั่นนี้ได้
- ส่วนนี้จะมีหรือไม่มีก็ได้

ฟังก์ชั่นหลัก (Main function)

- เป็นส่วนที่ทุกโปรแกรมจะต้องมี เป็นส่วนการทำงานหลักที่จะนำคำสั่งต่างๆ หรือฟังก์ชั่นต่าง ๆ มาประกอบกันเป็นโปรแกรม
- เป็นส่วนของรายละเอียดคำสั่งต่างๆที่เราต้องการให้คอมพิวเตอร์ทำงาน
- โดยการเขียนฟังก์ชันจะเริ่มด้วยการเขียน ชื่อฟังก์ชัน (Function Name) ก่อน จากนั้นจึงเขียนตัวคำสั่งต่างๆไว้ภายในวงเล็บ { } ใต้ชื่อฟังก์ชันนั้น
- ในโปรแกรมหนึ่งๆจะมีกี่ฟังก์ชันก็ได้ แต่อย่างน้อยที่สุดจะต้องมีอยู่ 1 ฟังก์ชันเสมอ คือ ฟังก์ชันที่มีชื่อว่า main ( ) เพราะฟังก์ชัน main ( ) นี้จะทำหน้าที่เป็น ตัวโปรแกรมหลักในการทำงาน
- ทุกๆ โปรแกรม จะมีฟังก์ชั่น main ได้เพียงฟังก์ชั่นเดียวเท่านั้น
- ทุก ๆ คำสั่ง จะปิดด้วยเครื่องหมาย ;

ส่วนประกอบภายในฟังก์ชั่น main

- ในฟังก์ชั่น main และฟังก์ชั่นอื่นๆ จะประกอบด้วยหลายๆ ส่วนเพื่อทำงาน ได้แก่
  - 1. คำสั่งสำหรับการประกาศค่าตัวแปรต่าง ๆ ซึ่งต้องมีการประกาศตั้งแต่ส่วนต้นของโปรแกรม
  - 2. ส่วนของคำสั่ง เช่นคำสั่ง for ,goto ,etc.
  - 3. ส่วนของการเรียกใช้ฟังก์ชั่นต่างๆ
  - 4. การส่งกลับค่า (retern)

ฟังก์ชันหลักของโปรแกรม (Main Function)

- ตัวอย่าง

```
#include <stdio.h>
int feet,inches;
void main()
{
  feet = 6;
  inches = feet * 12;
  printf("Height in inches is %d",inches);
}
```

ผลการทำงาน

```
Height in inches is 72
```


การสร้างและการใช้ฟังก์ชั่น(Uses-defined function)

- ส่วนนี้เป็นการกำหนดฟังก์ชั่นต่างๆ ขึ้นมาเอง โดยแต่ละฟังก์ชั่นจะอยู่ภายในเครื่องหมาย { }
- การเขียนโปรแกรมแบ่งเป็นฟังก์ชั่น ทำให้โปรแกรมเมอร์สามารถเขียนส่วนของโปรแกรมที่มีหน้าที่เฉพาะอย่างเตรียมไว้ เพื่อให้ส่วนของโปรแกรมส่วนอื่นๆ เรียกใช้ได้
- นอกจากฟังก์ชั่นที่โปรแกรมเมอร์เขียนเองแล้ว ยังมีฟังก์ชั่นอีกประเภทเรียกว่าไลบรารี่ฟังชั่น(Library Function) หรือฟังก์ชั่นสำเร็จรูป ซึ่งผู้พัฒนา compiler เตรียมไว้ให้ เช่น printf(), scanf()

ส่วนอธิบายโปรแกรม (Program comment)

- เป็นส่วนของการคอมเมนท์โปรแกรม เพื่อขยายความหรืออธิบายการทำงานต่างๆ เพื่อให้ผู้ศึกษาโปรแกรมนั้น เข้าใจได้ง่ายขึ้น
- ส่วนนี้จะไม่ถูกนำไป compile เป็นโปรแกรม และไม่มีผลใดๆ ต่อการทำงานของโปรแกรม
- การใช้ comment มี 2 แบบ
  - // เป็นการ comment เฉพาะบรรทัด ข้อความที่อยู่หลังเครื่องหมาย // จะถือว่าเป็น comment ทั้งบรรทัด เช่น **// comments**
  - **/* comments */** เป็นการ comment เป็นช่วง ข้อความใด

รูปแบบคำสั่งในภาษาซี

- รูปแบบคำสั่งในภาษาซี มีกฎเกณฑ์ในการเขียนคำสั่ง ดังนี้
  - 1. คำสั่งทุกคำสั่งต้องเขียนด้วยอักษรตัวเล็กเสมอ เช่นคำสั่ง printf , scanf , for
  - 2. ทุกคำสั่งจะใช้เครื่องหมาย ; แสดงการจบของคำสั่ง เช่น printf(“Hello”) ;
  - 3. การเขียนคำสั่ง จะเขียนได้แบบอิสระ (Free Format) คือ สามารถเขียนหลายๆคำสั่งต่อกันได้ เช่น

```
printf(“Hello”);   printf(“Goodbye”);   a = 95;
```


ตัวอย่างโปรแกรมของภาษาซี

```
/* My First C Program */
#include <stdio.h>
main()
{ float x,y,z;
  scanf(“%f %f”,&x,&y);
  z= x+y;
  printf(“The sum is %f\n”,z);
}
```

Comment

Preprocessor Directive

Declaration

Input

Computation

Output

การคอมไพล์และลิงค์โปรแกรมในภาษาซี

- สร้างตัวโปรแกรมที่เป็นตัวอักษร หรือเรียกว่า **Source file** โดยมีนามสกุลเป็น ***.c*** หรือ ***.cpp*** ขึ้นมาก่อน โดยใช้โปรแกรมที่สามารถเขียนไฟล์ที่เก็บอักขระ (Editor) ใดๆ ก็ได้
- คอมไพล์เลอร์ของภาษาซี (C Compiler) จะทำการแปลง **Source file** จากอักขระใดๆ ให้เป็นรหัสที่เครื่องคอมพิวเตอร์สามารถเข้าใจได้เก็บไว้ในอีกไฟล์หนึ่ง เรียกว่า **Object file** ที่มีนามสกุล ***.obj***
- ตัวเชื่อม (Linker) จะทำการตรวจสอบว่าในโปรแกรมที่เขียนขึ้นนั้น มีการเรียกใช้งานฟังก์ชันมาตรฐานใด จากห้องสมุดของภาษาซี (C Library) บ้างหรือไม่ ถ้ามีตัวเชื่อมจะทำการรวมเอาฟังก์ชันเหล่านั้นเข้ากับไฟล์วัตถุประสงค์ แล้วจะได้ไฟล์ที่สามารถทำงานได้ โดยมีนามสกุลเป็น ***.exe*** (ขั้นตอนนี้เรียกว่า การลิงค์ เป็นการรวมฟังก์ชันสำเร็จรูปเข้าไป แล้วสร้างไฟล์ที่ทำงานได้)

ตัวอย่าง

1. สร้างซอร์สไฟล์ที่มีข้อความ ทีเก็บโปรแกรมภาษาซี ดังนี้

```
#include <stdio.h>
void main(void)
{
      printf(“Hello World”);
}
```

แล้วเก็บไว้ในไฟล์ชื่อ test.c

2. ใช้คำสั่งเพื่อเรียกให้คอมไพล์เลอร์ของภาษาซีทำงาน โดยคอมไพล์เลอร์จะทำการแปลงอักขระจากซอร์สไฟล์ ให้เป็นรหัสที่คอมพิวเตอร์สามารถเข้าใจได้ จะได้ไฟล์ชื่อ test.obj

ตัวอย่าง (ต่อ)

3. ใช้คำสั่งเพื่อเรียกให้ตัวเชื่อมของภาษาซีทำงาน ตัวเชื่อมจะทำการรวมไฟล์ห้องสมุดที่ชื่อ *stdio.h* เข้ามา (ตามข้อความ #include …) แล้วสร้างไฟล์ที่สามารถทำงานได้ชื่อ *test.exe*

**หมายเหตุ คำว่า printf ในตัวโปรแกรม คือฟังก์ชันมาตรฐานหนึ่งที่พิมพ์ข้อความออกทางหน้าจอ โดยขั้นตอนและวิธีการทำงานของฟังก์ชัน printf จะอยู่ในไฟล์ห้องสมุดชื่อ *stdio.h*

ดังนั้น ถ้าตัวเชื่อมไม่ทำการรวมไฟล์ห้องสมุดที่ชื่อ *stdio.h* เข้าทำ จะทำให้ไม่สามารถเรียกใช้งานฟังก์ชัน printf ได้เลย

ตัวอย่างโปรแกรม 1

```
# include <stdio.h>
int main (void )
{
    printf("Hello, Good morning. \n");
}
```

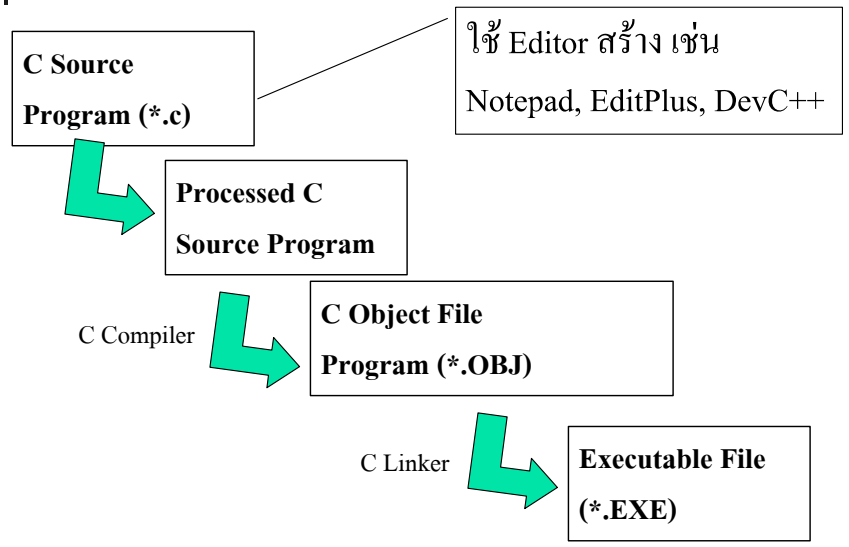
เป็นโปรแกรมสั่งพิมพ์ข้อความ Hello, Good morning.

ตัวอย่างโปรแกรม 2

```
# include <stdio.h>
main ( )
{
    float   point;
    printf("\n\nPut your score in\n");
    scanf("%f", &point);
    printf("Your score is %f point\n\n", point);
}
```

เป็นโปรแกรมรับคะแนนและเก็บค่าที่ตัวแปร **point** หลังจากนั้นสั่งให้มีการพิมพ์คะแนนออกมา

การแปลโปรแกรมภาษา C ให้เป็นภาษาเครื่อง (Compile)



กลุ่มคำในภาษาซี: คำสงวน (Reserve word)

- ในภาษา C มีคำสงวน (reserved word) จำนวนหนึ่งซึ่งเรียกว่าคำหลัก ซึ่งคำหลักนี้จะเป็นมาตรฐานและได้กำหนดความหมายไว้ก่อนแล้ว
- โดยคำหลักเหล่านี้ใช้ได้เฉพาะเพื่อวัตถุประสงค์ตามที่กำหนดไว้เท่านั้น
- ไม่สามารถนำมาใช้เป็นตัวกำหนดชื่อที่นักเขียนโปรแกรมกำหนดขึ้นเองได้

คำสงวน (Reserve word)

dxwr	euhdn	fdvh	fkdu
frqvw	frqwlqxh	ghidxow	gr
grxeoh	hovh	hqxp	h{whuq
iordw	iru	jrwr	li
lqw	orqj	uhjlvwhu	uhwxuq
vkruw	vljqhg	vl}hri	vwdwlf
vwuxfw	vz lwfk	w|shghi	xqlrq
xqvljqhg	yrlg	yrodwloh	

ตัวแปลภาษาที่ใช้ในวิชา 517101

- โปรแกรม Dev C++ version 4.9.9.2 (Dev-C++ 5.0 beta 9.2 (4.9.9.2)
- เป็น Freeware สามารถใช้งานได้โดยไม่มีค่าใช้จ่าย
- เป็นภาษา C ตาม C Standard
- มี IDE (Integrated Development Environment) ในตัว ทำให้เขียนโปรแกรมได้สะดวกโดยไม่ต้องใช้ editor อื่นในการเขียนโปรแกรม
- ใช้ได้ทั้งภาษา C และ C++
- สามารถ download ได้ที่

http://prdownloads.sourceforge.net/dev-cpp/devcpp-4.9.9.2_setup.exe