Week 5

ขอบเขต และ ช่วงชีวิตของตัวแปร

- ในบางบริเวณของโปรแกรม ตัวแปรที่ถูกกำหนดขึ้นในบริเวณอาจถูกเข้าถึงได้ เมื่อออกบริเวณดังกล่าว ตัวแปรนั้นอาจไม่สามารถถูกเข้าถึงได้อีก

Local Variables

ตัวแปรที่ถูกประกาศไว้อยู่ภายใน block, ภายในตัวฟังก์ชัน รวมถึงพารามิเตอร์ของฟังก์ชัน จะเป็นตัวแปร local สามารถถูกใช้งานได้เฉพาะภายในฟังก์ชันที่ถูกประกาศไว้

Global variables

- ตัวแปรที่ถูกประกาศไว้นอกฟังก์ชันเป็นตัวแปร global ซึ่งสามารถถูกเข้าถึงได้จากฟังก์ชันต่างๆ รวมถึงฟังก์ชัน main ได้
- หากตัวแปร local ชื่อซ้ำกับตัวแปร global การแก้ไขตัวแปร local จะไม่มีผลกระทบถึงค่าตัวแปร global

Storage classes มี 4 ลักษณะ

auto, register, extern และ static เพื่อใช้งานกำหนดช่วงชีวิตในการเก็บข้อมูลของตัวแปรนั้นใน memory ได้

แบ่งออกเป็น 2 storage duration

1. automatic storage duration
2. static storage duration

auto และ register เป็น automatic storage duration

register = เป็นการแจ้งให้คอมไพเลอร์ได้ทราบว่า (เพิ่มความเร็ว) ต้องการเก็บตัวแปรไว้ใน register

extern และ static ใช้กำหนดตัวแปรที่เป็น static storage duration ซึ่งก็คือ ตัวแปรเหล่านี้จะถูกกำหนดขึ้นมาใน memory ตั้งแต่โปรแกรมเริ่มทำงาน และจะอยู่ใน memory จนกว่าโปรแกรมจะสิ้นสุดการทำงาน

ตัวแปรจะเริ่มเก็บค่าข้อมูล และเก็บต่อไปจนกว่าโปรแกรมจะทำงานสิ้นสุด

ตัวแปร global จะถูกกำหนดให้เป็น extern

ถ้าตัวแปร local ใช้ static ซึ่งอยู่ภายในฟังก์ชัน จะถูกเข้าถึงได้เฉพาะภายในฟังก์ชันเท่านั้น แต่จะคงอยู่ใน memory แม้ว่าจะออกจากฟังก์ชันนี้ก็ตาม

Dynamic Storage duration

พื้นที่เก็บข้อมูลสามารถถูก allocate ขึ้นมาได้ตามความต้องการของโปรแกรมที่กำลังทำงานอยู่

การคอมไพล์แยกส่วน

วิธีการคอมไพล์

```
gcc -c main.c # สร้าง main.o
gcc -c func.c # สร้าง func.o
gcc main.o func.o -o main.exe # ลิงก์ main.o และ func.o เข้าด้วยกัน
```

การใช้ make

การสร้างกฎ target: dependency

--> tab --> คำสั่งที่ใช้ในการสร้าง target

การสร้าง Static library

```
gcc -c func.c
or cr libfunc.a func.o
```

random number generator

rand() เป็นฟังก์ชันที่ใช้ในการสร้างเลขสุ่มที่เป็นจำนวนเต็ม ระหว่าง 0 กับ RAND_MAX

ใช้ srand() เพื่อให้ชุดเลขสุ่มต่างกันออกไป