

Week 7 Pointers และ Dynamic Memory Allocation

Pointers and Addresses

- Pointer เป็นตัวแปรที่ใช้เก็บ address ในหน่วยความจำที่ต้องการอ้างถึง
- ค่าของ address คือค่าจำนวนเต็มบวกที่เริ่มจาก 0

Pointer Operators

ในการใช้ pointer มี ตัวที่ถุกนิยาม

- & เรียกว่า address operator คือ unary operator ที่ต้องการเพียง 1 operand operand อยู่ในรูปของตัวแปร
`p = &c;`
- * เรียกว่า dereferencing operator หรือ indirection operator คือ unary operator โดยที่ * ถูกใช้ในการเข้าถึงข้อมูลอ้างอิงโดยตำแหน่งในหน่วยความจำที่มีค่าเป็น operand

Scanf()

Scanf() จะตรงข้ามกับ printf() คือ scanf() เป็นการรับข้อมูลเข้ามาในโปรแกรมผ่าน standard input ส่วน printf() คือ การแสดงผลออกไปที่ standard output

```
int input;  
  
scanf("%d", &input);
```

การใช้ const กับ pointers

Const สามารถใช้ร่วมกับ pointer เพื่อแจ้งให้คอมไพเลอร์ทราบว่าค่าของตัวแปรนั้นไม่สามารถแก้ไขได้

นิพจน์ใน pointer และการใช้ตัวดำเนินการทางคณิตศาสตร์กับ pointer

ตัวแปร pointer สามารถเป็น operand ให้กับนิพจน์เหล่านี้ได้

- นิพจน์ทางคณิตศาสตร์
- นิพจน์ในการกำหนดค่า
- นิพจน์ในการเปรียบเทียบ

เครื่องหมายของนิพจน์ที่มี pointer กับตัวดำเนินการทางคณิตศาสตร์

```
Float f[4] = {0.0,1.0,2.0,3.0}
```

```
Float *pf;
```

```
Pf = &f[0];
```

```
Pf = pf + 2;
```

ขนาดของ pf คือ 4 byte เพราะคือ float ดังนั้นจะมีค่าเป็น $2 * 4 = 8$ byte pf = pf + 2 pf จะมีค่าเพิ่มขึ้นไปอีก 8 หน่วย

การใช้ pointer อ้างไปที่ฟังก์ชัน (Pointers to Functions)

Pointer อ้างไปยังตำแหน่งในหน่วยความจำของฟังก์ชันที่ต้องการ ฟังก์ชันที่ถูกเรียกใช้งานจะถูกโหลดเข้ามาในหน่วยความจำ

ชื่อของฟังก์ชันนิพจน์ที่ให้ค่า base address ของฟังก์ชัน ซึ่งมีลักษณะเดียวกับ array ที่ชื่อ array เป็นนิพจน์ที่ให้ค่า base address ของ array

การใช้ free()

```
Free(data);
```

Free() จะเป็นตำแหน่งในหน่วยความจำ (base address) ที่เราต้องการคืนพื้นที่ให้กับระบบ