

## Lecture 3

### เอกสารประกอบการบรรยาย Data Structure

### เรื่อง Set and String

## Set and String

### เนื้อหา

- โครงสร้างข้อมูลแบบเซต
- โครงสร้างข้อมูลแบบสตริง

## โครงสร้างข้อมูลแบบเซต

เป็นโครงสร้างข้อมูลที่ข้อมูลแต่ละตัวไม่มี  
ความสัมพันธ์กัน ในภาษาซี  
จะไม่มีประเภทข้อมูลแบบเซตนี้เหมือนกับในภาษา  
ปาสคาล แต่สามารถใช้หลักการของการดำเนินงาน  
แบบเซตมาใช้ได้

ตัวดำเนินการของเซต (Set operators)

ประกอบด้วย

- set intersection
- set union
- set difference เป็นต้น

## โครงสร้างข้อมูลแบบเซต (ต่อ)

สมมติว่า ต้องการจัดตารางเรียน 4 วิชา ได้แก่ Math, English,  
Physics และ Chemistry ให้กับผู้ลงทะเบียนเรียน

วิธีการแก้ปัญหาเบื้องต้น

- จะต้องกำหนดเซตของผู้เรียนที่ลงทะเบียนเรียนในแต่ละวิชา
  - นำเซตดังกล่าวที่ได้มาทำการ intersection กัน หากมีเซตใดที่  
ทำการ intersect กันแล้ว มีข้อมูลสมาชิกในเซตที่ซ้ำกันอยู่ จะ  
ไม่สามารถจัดให้วิชาดังกล่าวอยู่ในวันเวลาเดียวกันได้
- ตัวอย่างดังกล่าว เป็นการนำแนวความคิดเรื่องการจัดการแบบ  
เซตมาประยุกต์ใช้งาน

## โครงสร้างข้อมูลแบบสตริง

**สตริง (String)** หรือ **สตริงของอักขระ (Character String)** เป็นข้อมูลที่ประกอบไปด้วย ตัวอักษร ตัวเลขหรือ เครื่องหมายเรียงติดต่อกันไป รวมทั้งช่องว่าง

การประยุกต์ใช้คอมพิวเตอร์ที่เกี่ยวกับข้อมูลที่เป็นสตริง มีการนำไปใช้สร้างโปรแกรมประเภทบรรณาธิการข้อความ (text editor) หรือโปรแกรมประเภทประมวลผลคำ (word processing) ซึ่งมีการทำงานที่อำนวยความสะดวกหลายอย่าง เช่น การตรวจสอบข้อความ การจัดแนวข้อความ ในแต่ละย่อหน้า และการค้นหาคำ เป็นต้น

## โครงสร้างข้อมูลแบบสตริง (ต่อ)

### สตริงกับอะเรย์

สตริง คือ อะเรย์ของอักขระ  
เช่น `char a[6]`  
อาจจะเป็นอะเรย์ขนาด 6 ช่องอักขระ หรือ เป็นสตริงขนาด 5 อักขระก็ได้ โดยจุดสิ้นสุดของ string จะจบด้วย `\0` หรือ null character เช่น  
`char a[]={'H', 'E', 'L', 'L', 'O', '\0'};`  
`char a[]="HELLO";`

## โครงสร้างข้อมูลแบบสตริง (ต่อ)

ความยาวของสตริง จะถูกกำหนดโดย ขนาดของสตริง การกำหนดขนาดของสตริง นั้นต้องจองเนื้อที่ในหน่วยความจำให้กับ `\0` ด้วย

เช่น

`"This is String !"`

จะเป็นข้อมูลแบบสตริงยาว 16 อักขระ

## โครงสร้างข้อมูลแบบสตริง (ต่อ)

### การกำหนดสตริง

การกำหนดสตริงทำได้หลายแบบ คือ

1. กำหนดเป็นสตริงที่มีค่าคงตัว (String Constants)
2. กำหนดโดยใช้ตัวแปรอะเรย์หรือพอยเตอร์

## โครงสร้างข้อมูลแบบสตริง (ต่อ)

### การกำหนดค่าคงตัวสตริง

สามารถกำหนดได้ทั้งนอกและในฟังก์ชัน เมื่อกำหนดไว้นอกฟังก์ชัน ชื่อค่าคงตัวจะเป็นพอยเตอร์ชี้ไปยังหน่วยความจำที่เก็บสตริงนั้น เมื่อกำหนดไว้ในฟังก์ชัน จะเป็นพอยเตอร์ไปยังหน่วยความจำที่เก็บตัวมันเอง

การกำหนดค่าให้กับสตริงนั้น เราจะใช้เครื่องหมาย double quote (“ ”) เช่น “abc” คือ ชุดของอักขระที่มีขนาด 4 (รวม \0 ด้วย)

### ข้อสังเกต

string constant are different from character constant.

## Example



## Example

```
#define NME "Semi"
main ( ){
    char *cpntr;
    cpntr=NME;
    printf("con\n");
    printf("%s, %u, %c\n", "con", "duc", *"tor");
    printf("%s, %u, %c\n", NME, NME, *NME);
    printf("%s, %u, %c\n", cpntr, cpntr, *cpntr);
}
```

## Example

ผลการรันโปรแกรม

con

con, 37, t

Semi, 16, S

Semi, 16, S

## โครงสร้างข้อมูลแบบสตริง (ต่อ)

จากตัวอย่าง กำหนดค่าคงตัว NME ขึ้น โดยให้มีค่าเป็น Semi ทำให้ MNE เป็นพอยเตอร์ชี้ไปที่ข้อมูล Semi จากคำสั่ง `printf("con\n");` คำว่า con จะอยู่ในฐานะข้อมูล แต่จากคำสั่ง `printf("%s, %u, %c\n", "con", "duc", *tor);` ค่าคงตัว con duc และ tor จะอยู่ในฐานะพอยเตอร์ เพราะรูปแบบการพิมพ์ของ `printf( )` ได้กำหนดไว้ว่าถ้าให้พิมพ์สตริงจะต้องผ่านค่าพอยเตอร์

## โครงสร้างข้อมูลแบบสตริง (ต่อ)

### การกำหนดค่าคงตัวสตริงให้แก่ตัวแปรพอยเตอร์และอะเรย์

สามารถกำหนดค่าคงตัวสตริงให้พอยเตอร์หรืออะเรย์ได้ในฐานะค่าเริ่มต้น

```
เช่น main ( ) {  
    char ary[ ] = "This is the house. ";  
    char *cpntr="This is the door.";  
    printf("%s %s",ary,cpntr);  
}
```

## โครงสร้างข้อมูลแบบสตริง (ต่อ)

ผลการรันโปรแกรม

This is the house. This is the door.  
จะเห็นได้ว่าการใช้งานดูไม่แตกต่างกัน แต่ ary เป็นตัวแปรอะเรย์ ค่าที่ให้จะต้องเป็นค่าข้อมูลในอะเรย์ ส่วน cpntr เป็นพอยเตอร์ ค่าที่ให้นั้นไม่ใช่ค่าข้อมูล แต่เป็นค่าแอดเดรสเริ่มต้นของสตริง (ค่าคงตัวสตริงเป็นทั้งข้อมูลสตริงและพอยเตอร์) นอกจากนี้ยังสามารถเพิ่มลดค่าตัวแปรพอยเตอร์ได้ แต่สำหรับอะเรย์ทำไม่ได้

## โครงสร้างข้อมูลแบบสตริง (ต่อ)

เช่น

```
while (*cpntr !='\0')  
    putchar(*cpntr++);
```

เป็นการให้พิมพ์ข้อความ This is the door ทีละอักขระ จนกว่าจะสุดสตริง คือ เมื่อ \*cpntr มีค่าเป็น null character

## โครงสร้างข้อมูลแบบสตริง (ต่อ)

### การกำหนดตัวแปรสตริง

ในการกำหนดตัวแปรของสตริง อาศัยหลักการของอะเรย์ เพราะ สตริงก็คืออะเรย์ของอักขระที่ปิดท้ายด้วย null character (\0) และมีฟังก์ชันพิเศษสำหรับทำงานกับสตริงโดยเฉพาะ

เช่น

ต้องการสตริงสำหรับเก็บชื่อบุคคลยาวไม่เกิน 30 อักขระ ต้องกำหนดเป็นอะเรย์ขนาด 31 ช่อง เพื่อเก็บ null character อีก 1 ช่อง

## Example

```
main ( ) {  
    char name[31];  
    printf("Hi, What's your name?\n");  
    gets(name);  
    printf("Nice to meet you, %s\n");  
}
```

## Example

ผลการรันโปรแกรม

Hi, What's your name?

Jame Smith

Nice to meet you, Jame Smith

## โครงสร้างข้อมูลแบบสตริง (ต่อ)

ในการกำหนดตัวแปร name เท่ากับเป็นการกำหนดอะเรย์ ขนาด 31 ช่อง ซึ่งเครื่องจะจองเนื้อที่ในหน่วยความจำไว้ 31 ไบต์ และให้ตัวแปร name ชี้ไปที่ต้นอะเรย์นี้

ในการกำหนดตัวแปร name กำหนดไว้ว่า ชื่อจะต้องยาวไม่เกิน 30 อักขระ หากผู้ใช้ป้อนไม่ถึง 30 อักขระ เครื่องจะทำการเติม null character ให้จนครบ 31 ช่อง แต่ถ้าผู้ใช้ป้อนเกินจะเกิดข้อผิดพลาดหลุดจากโปรแกรมนี้ เพราะฉะนั้นจึงต้องกำหนดความยาวของสตริงให้เพียงพอ

## โครงสร้างข้อมูลแบบสตริง (ต่อ)

ฟังก์ชัน gets( ) เป็นฟังก์ชันที่อ่านค่าจากแป้นพิมพ์มาเก็บไว้ในหน่วยความจำ ซึ่งก็คืออะเรย์ที่ตัวแปร name ซี่อยู่ รวมทั้งช่องว่าง จนกว่าผู้ใช้จะกด Enter จะเติม null character ให้

หากใช้ scanf( ) จะถือว่า Jame เป็นค่าหนึ่ง และ Smith เป็นอีกค่าหนึ่ง เพราะมีช่องว่างคั่น และไม่เติม null character นั่นคือจะได้อะเรย์ที่มีค่าเป็น Jame ไม่ใช่ String ที่มีค่าเป็น Jame Smith

## โครงสร้างข้อมูลแบบสตริง (ต่อ)

### อะเรย์ของสตริง

ถ้าหากมีสตริงจำนวนมาก ก็ควรจะทำให้เป็นอะเรย์ของสตริง เพื่อที่จะเขียนโปรแกรมได้สะดวก การสร้างอะเรย์ของสตริง สามารถสร้างได้ทั้งแบบที่ให้ค่าเริ่มต้นและแบบที่กำหนดเป็นตัวแปร

## โครงสร้างข้อมูลแบบสตริง (ต่อ)

### อะเรย์ของสตริงที่ยาวไม่เท่ากัน

ทำได้เฉพาะเมื่อมีการกำหนดค่าเริ่มต้นเท่านั้น  
เช่น

```
#define N 4
main () {
char *contry[N]={“Thailand”, “United State of Ameica”,
“England”, “Indonesia”};

int a;
for (a=0; a<N;a++)
puts(contry[a]);
}
```

## โครงสร้างข้อมูลแบบสตริง (ต่อ)

ผลการรันโปรแกรม

Thailand  
United State of America  
England  
Indonesia

## โครงสร้างข้อมูลแบบสตริง (ต่อ)

การกำหนดตัวแปร `country` จะแตกต่างกับการกำหนดตัวแปรอะเรย์ เพราะเป็นการกำหนดตัวแปรพอยเตอร์ขึ้น 4 ตัว โดยให้แต่ละตัวชี้ไปยังค่าคงตัวสตริงทั้ง 4 ตัว โดยที่ `contry[0]` จะชี้ไปที่ข้อมูลแรก `contry[1]` จะชี้ข้อมูลที่สอง `contry[2]` จะชี้ข้อมูลที่สาม และ `contry[3]` จะชี้ข้อมูลตัวสุดท้าย

ในการเขียนค่าเริ่มต้น คือ ค่าคงตัวสตริง เขียนไว้ในเครื่องหมายวงเล็บปีกกา และข้อมูลในเครื่องหมายคำพูด คือ ค่าคงตัวสตริง

## โครงสร้างข้อมูลแบบสตริง (ต่อ)

ฟังก์ชัน `puts ( )` ใช้ในการพิมพ์สตริงออกทางจอภาพ โดยการผ่านค่าแอดเดรสของสตริงไปให้เท่านั้น

### ข้อสังเกต

การกำหนดอะเรย์ของสตริงในลักษณะอย่างนี้ ไม่ใช่อะเรย์ที่แท้จริงตามหลักการของอะเรย์ เนื่องจากขนาดของช่องในอะเรย์ไม่เท่ากัน แต่อนุโลมให้ถือว่าเป็นอะเรย์

## โครงสร้างข้อมูลแบบสตริง (ต่อ)

### อะเรย์ของสตริงที่ยาวเท่ากัน

อะเรย์ในลักษณะนี้จะถือว่าเป็นอะเรย์ที่แท้จริง และสามารถกำหนดได้ทั้งเมื่อมีการให้ค่าเริ่มต้น และเมื่อกำหนดเป็นตัวแปร โดยดำเนินการตามแบบการกำหนดอะเรย์ 2 มิติ

เช่น

```
char fruit [3][7]={"Apple", "Orange", "Mango"};
```

กำหนดตัวแปร `fruit` เป็นแบบ 3 แถว 7 คอลัมน์ ในแต่ละช่องจะเก็บข้อมูลแบบอักขระ

## โครงสร้างข้อมูลแบบสตริง (ต่อ)

### อะเรย์ของสตริงที่ยาวเท่ากัน

อะเรย์ในลักษณะนี้จะถือว่าเป็นอะเรย์ที่แท้จริง และสามารถกำหนดได้ทั้งเมื่อมีการให้ค่าเริ่มต้น และเมื่อกำหนดเป็นตัวแปร โดยดำเนินการตามแบบการกำหนดอะเรย์ 2 มิติ

## โครงสร้างข้อมูลแบบสตริง (ต่อ)

การกำหนดตัวแปรในลักษณะนี้ จะแตกต่างจากการกำหนดตัวแปรแบบความยาวไม่เท่ากัน คือ ในแบบความยาวไม่เท่ากัน ท้ายของสตริงจะเครื่องหมายเติม null character ให้เพียงตัวเดียว แต่ในแบบความยาวเท่ากัน จะเติม null character ให้จนครบทุกช่อง

## Example

```
char fruit[3]= {"Apple", "Orange", "Mango"};
```

A	p	p	l	e	\0	
O	r	a	n	g	e	\0
M	a	n	g	o	\0	

## Example

```
char fruit[3][7]= {"Apple", "Orange", "Mango"};
```

A	p	p	l	e	\0	\0
O	r	a	n	g	e	\0
M	a	n	g	o	\0	\0

## โครงสร้างข้อมูลแบบสตริง (ต่อ)

### การดำเนินการเกี่ยวกับสตริง

ในการดำเนินการเกี่ยวกับสตริง จะมีฟังก์ชันที่อยู่ในแฟ้ม ข้อมูล stdio.h เก็บอยู่ใน C Library อยู่แล้ว สามารถนำมาใช้ได้ โดยการใส่คำสั่ง #include ในการเรียกใช้ เช่น

- ฟังก์ชัน strlen(str) ใช้หาความยาวของสตริง
- ฟังก์ชัน strcpy (str1, str2) ใช้คัดลอกข้อมูลจาก string หนึ่งไปยังอีก string หนึ่ง



## โครงสร้างข้อมูลแบบสตริง (ต่อ)

- ฟังก์ชัน `strcat(str1, str2)` ใช้เชื่อมต่อข้อความ 2 ข้อความเข้าด้วยกัน
- ฟังก์ชัน `strcmp(str1, str2)` ใช้เปรียบเทียบข้อความ 2 ข้อความว่ามีค่าเท่ากันหรือไม่ ถ้อยหลักการเปรียบเทียบแบบพจนานุกรม เช่น `abcda` จะมีค่าน้อยกว่า `abcde` และ `abcdf` จะมีค่ามากกว่า `abcde` ค่าที่เท่ากัน คือ ค่าที่เหมือนกัน เช่น `abcd` กับ `abcd` สำหรับอักษรตัวเล็กตัวใหญ่ จะถือว่าอักษรตัวใหญ่มีค่าน้อยกว่าอักษรตัวเล็ก ตามลำดับรหัส ASCII

## แบบฝึกหัด

1. การกำหนดค่าตัวแปรแบบ Set มีประโยชน์อย่างไร ให้นักศึกษาอธิบายการทำงานที่เกี่ยวข้อง
2. การกำหนดค่าตัวแปรสตริงสามารถกำหนดอย่างไรได้บ้าง ให้นักศึกษากำหนดตัวแปรชนิดสตริงเพื่อจัดเก็บข้อมูลตามประเภทที่เป็นรายละเอียด เช่น ที่อยู่ ชื่อ
3. ให้นักศึกษาลองเขียนโปรแกรม ภาษาซี เพื่อกำหนด การใช้ฟังก์ชัน `strlen()`, `strcmp()`, `strcat()`, `strcpy()`