บทที่ 4

ชุดคำสั่งอรรถประโยชน์

ชุดคำสั่งอรรถประโยชน์

ในระบบธุนิกซ์ผู้ใช้สามารถสร้างชุดคำสั่งอรรถประโยชน์ต่าง ๆ เพิ่มเติม เพื่อนำไปใช้งานให้ตรงกับความต้องการของผู้ใช้ ชุดคำสั่งอรรถประโยชน์ที่สร้างขึ้นมาจะเพิ่มเติมนี้สามารถเรียกใช้ได้เหมือนกับชุดคำสั่งอรรถประโยชน์อื่น ๆ ของระบบธุนิกซ์ ชุดคำสั่งอรรถประโยชน์ที่สร้างขึ้นมาเพื่อใช้งานกับระบบการขายและแสดงสินค้าคงคลัง เป็นชุดคำสั่งที่สร้างด้วยสภาพแวดล้อมของระบบธุนิกซ์ ซึ่งประกอบด้วยคำสั่งงานต่าง ๆ ของระบบธุนิกซ์และคำสั่งงานของชุดคำสั่งเชลล์ ยกเว้นชุดคำสั่งอรรถประโยชน์ชื่อ keyret เขียนด้วยภาษาซีซึ่งเป็นชุดคำสั่งรับข้อมูลจากแผงแป้นอักขระ

ชุดคำสั่งอรรถประโยชน์ที่สร้างขึ้นมามีดังนี้

- addcomma : พิมพ์ข้อมูลตัวเลขในลักษณะ comma (,)
- cursor : ย้ายเคอร์เซอร์ไปยังตำแหน่งที่กำหนด
- d2j : เปลี่ยนวันที่จากรูปแบบ YYYYMMDD เป็นตัวเลขวันที่จุเลียน
- d2jlawk : เรียกใช้โดยชุดคำสั่ง d2j
- j2d : เปลี่ยนตัวเลขวันที่จุเลียนเป็นวันที่ในรูปแบบ YYYYMMDD
- j2dlawk : เรียกใช้โดยชุดคำสั่ง j2d
- fdate : คำนวณวันที่ในอดีตและอนาคต
- verildat : ตรวจสอบความถูกต้องของวันที่
- cd2sldat : เปลี่ยนรูปแบบของวันที่สำหรับแสดงผล (DD/MM/YYYY)
- cs2dldat : เปลี่ยนรูปแบบของวันที่สำหรับเก็บลงแฟ้มข้อมูล (YYYYMMDD)
- length : หาความยาวของสายอักขระ
- substr : แยกสายอักขระ
- verilnum : ตรวจสอบความถูกต้องของตัวเลข
- keyret : รับข้อมูลจากแผงแป้นอักขระ

- termput : ดึงข้อมูลรายละเอียดของเครื่องปลายทางจากแฟ้มข้อมูลซึ่งมีชื่อแฟ้มข้อมูลเก็บอยู่ในตัวแปรชื่อ TERMCAP
- c2n : เปลี่ยนคำสั่งงาน echo \c เป็น echo -n
- n2c : เปลี่ยนคำสั่งงาน echo -n เป็น echo \c

<u>รายละเอียดและตัวอย่างการใช้งาน</u>

รายละเอียดและตัวอย่างการใช้งานของชุดคำสั่งอรรถประโยชน์ มีดังนี้

1. <u>addcomma</u>

ใช้ใส่เครื่องหมาย comma (,) ในข้อมูลตัวเลข โดยมีข้อมูลเข้าเป็นตัวเลขที่มีจุดทศนิยมหรือไม่มีก็ได้ ข้อมูลออกเป็นตัวเลขมีเครื่องหมาย comma และจุดทศนิยม 2 หลัก ดังตัวอย่าง

```
$addcomma 123456789
1,234,456,789.00
$addcomma -123456.5
-123,456.50
```

2. <u>cursor</u>

ใช้เคลื่อนย้ายตำแหน่งของเคอร์เซอร์บนจอภาพในยังตำแหน่งแถว (row) และแนวตั้ง (column) ที่กำหนด ดังตัวอย่าง

```
$cursor 5 10
เคอร์เซอร์จะไปปรากฏอยู่ที่ตำแหน่งแถวที่ 5 และแนวตั้งที่ 10
$cursor 5 10 ; echo "hello"
แสดงข้อความว่า "hello" ในตำแหน่งแถวที่ 5 และแนวตั้งที่ 10
```

3. <u>d2j</u>

ใช้เปลี่ยนวันที่จากรูปแบบ YYYYMMDD เป็นตัวเลขวันที่จูเลียน ข้อมูลวันที่เข้ามีรูปแบบ YYYYMMDD ข้อมูลออกเป็นตัวเลขวันที่จูเลียน โดยเริ่มนับจากวันที่ 1 มกราคมปี ค.ศ. 1980 ชุดคำสั่งนี้จะมีการเรียกใช้ชุดคำสั่ง d2jlawk เพื่อตรวจสอบความถูกต้องของวันที่และ

คำนวณค่าตัวเลขวันที่จูเลียน ดังตัวอย่าง

```
$d2j 19800101
1
$d2j 19900101
3654
```

4. j2d

เปลี่ยนตัวเลขวันที่จูเลียน เป็นวันที่ในรูปแบบ YYYYMMDD ข้อมูลที่เข้าเป็นตัวเลขวันที่จูเลียน โดยนับจากวันที่ 1 มกราคม ปี ค.ศ. 1980 ข้อมูลออกจะเป็นวันที่มีรูปแบบ YYYYMMDD ชุดคำสั่งนี้จะมีการเรียกใช้ชุดคำสั่ง j2dlawk เพื่อคำนวณวันที่ ดังตัวอย่าง

```
$j2d 1
19800101
$j2d 3654
19900101
```

5. fdate

หาวันที่ย้อนหลังหรือวันที่ล่วงหน้าจากวันที่และจำนวนวันที่ระบุ ข้อมูลเข้าเป็นวันที่มีรูปแบบ YYYYMMDD จำนวนวันที่มีค่าเป็นบวกหรือลบก็ได้ ถ้าระบุจำนวนวันที่มีค่าเป็นบวกเป็นการหาวันที่ล่วงหน้า และถ้าระบุจำนวนวันที่มีค่าเป็นลบเป็นการหาวันที่ย้อนหลัง ดังตัวอย่าง

```
$fdate 19900101 15
19900116
$fdate 19900101 -15
19891217
```

6. verildat

ตรวจสอบความถูกต้องของวันที่ โดยวันที่เข้ามีรูปแบบ YYYYMMDD และต้องอยู่ในช่วงปี ค.ศ. 1980 ถึง 9999 เท่านั้น ถ้าวันที่ที่ตรวจสอบไม่ถูกต้องจะให้รหัสส่งกลับ (return code) มีค่าเป็น 1 และแสดงข่าวสารความผิดพลาดออกมา ถ้าวันที่ถูกต้องจะแสดงวันที่เดิมออก

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

มา ดังตัวอย่าง

```
$verildat 19880229
19880229
$verildat 19890229
Day > maxfeb
```

7. cd2sldat

เปลี่ยนรูปแบบของวันที่จากรูปแบบ DD/MM/YYYY เป็นรูปแบบ YYYYMMDD เพื่อใช้เก็บลงแฟ้มข้อมูล ซึ่งจะทำให้การตรวจสอบและการคำนวณเกี่ยวกับวันที่กระทำได้ง่ายขึ้น ดังตัวอย่าง

```
$cd2sldat 01/02/1990
19900201
```

8. cs2dldat

เปลี่ยนรูปแบบของวันที่จากรูปแบบ YYYYMMDD เป็นรูปแบบ DD/MM/YYYY เพื่อใช้ในการแสดงผล ดังตัวอย่าง

```
$cs2dldat 19900102
02/01/1990
```

9. length

หาความยาวของสายอักขระ ข้อมูลเข้าสามารถป้อนที่บรรทัดคำสั่งงานหรือจะรับข้อมูลจากแฟ้มข้อมูลโดยใช้การเบี่ยงเบนรับเข้าก็ได้ ดังตัวอย่าง

```
$length hello
5
$echo -n 'hello computer' | length
14
$length < filein
100
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

10. substr

ดึงสายอักขระย่อยจากสายอักขระที่กำหนด โดยระบุตำแหน่งเริ่มต้นและจำนวนของอักขระที่ต้องการ ดังตัวอย่าง

```
$ substr computer 1 4
comp
$substr computer 4 3
put
```

11. verilnum

ตรวจสอบความถูกต้องของตัวเลข ถ้ามีการระบุทางเลือก -d0 ตัวเลขข้อมูลเข้าจะต้องเป็นตัวเลข 0 ถึง 9 เท่านั้นและตัวเลขข้อมูลออกไม่มีจุดทศนิยม แต่ถ้าระบุทางเลือก -d2 ตัวเลขข้อมูลเข้าเป็นตัวเลขไม่มีจุดทศนิยมหรือมีจุดทศนิยม 2 หลักก็ได้ และตัวเลขข้อมูลออกจะมีจุดทศนิยม 2 ตำแหน่ง ดังตัวอย่าง

```
$verilnum -d0 1000
1000
$verilnum -d2 1000
1000.00
$verilnum -d2 1000.5
1000.50
```

12. termput

ดึงข้อมูลรายละเอียดของเครื่องปลายทางจากแฟ้มข้อมูล ซึ่งมีชื่อเก็บอยู่ในตัวแปรชื่อ TERMCAP โดยข้อมูลเข้ามีการระบุชื่อรายละเอียดที่ต้องการ ดังตัวอย่าง

```
$termput cl     ลบจอภาพ
$termput mb     การแสดงผลที่จอภาพเป็นลักษณะกระพริบ
$termput mr     การแสดงผลที่จอภาพเป็นลักษณะการแสดงตรงข้าม
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

13. keyret

รับข้อมูลจากแผงแป้นอักขระ โดยสามารถระบุจำนวนตัวอักขระที่จะรับได้ ชุดคำสั่งอรรถประโยชน์นี้เขียนขึ้นด้วยภาษาซี เพื่อให้มีความสามารถควบคุมการรับข้อมูลจากผู้ใช้ให้มีความถูกต้องและให้มีความสะดวกมากขึ้น ดังตัวอย่าง

```
$keyret        รับข้อมูล 1 ตัวอักษร
$keyret 5      รับข้อมูล 5 ตัวอักษร
```

14. c2n

เปลี่ยนคำสั่งงานจาก echo \c เป็น echo -n เพื่อให้สามารถปฏิบัติการได้ในสภาพแวดล้อมที่แตกต่างกันออกไปเช่น การปฏิบัติการบนระบบยูนิกซ์ SYSTEM V หรือบนระบบยูนิกซ์ BSD ตัวอย่างการเปลี่ยนคำสั่งงานเช่น เปลี่ยนคำสั่งงาน echo "hello\c" เป็น echo -n "hllo" การใช้งานข้อมูลเข้าจะต้องระบุชื่อชุดคำสั่งที่ต้องการจะเปลี่ยนคำสั่งงาน ดังตัวอย่าง

```
$c2n program1
```

15. n2c

เปลี่ยนคำสั่งงานจาก echo -n เป็น echo \c เช่นเปลี่ยนคำสั่งงานจาก echo -n "hello" เป็น echo "hello\c" ดังตัวอย่าง

```
$n2c program1
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย