หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2

ระบบยูนิกซ์

ระบบยูนิกซ์

ระบบยูนิกซ์เป็นระบบปฏิบัติการคอมพิวเตอร์ (operating system) ซึ่งประกอบด้วยกลุ่มของชุดคำสั่งเพื่อใช้ในการควบคุมระบบคอมพิวเตอร์ ใช้ในการติดต่อกันระหว่างผู้ใช้กับคอมพิวเตอร์ จัดเตรียมเครื่องมือต่าง ๆ ไว้สำหรับการทำงาน ระบบปฏิบัติการยูนิกซ์ถูกพัฒนาขึ้นมาในปี ค.ศ.1970 โดยนักวิจัยที่ห้องวิจัยเบลล์ (Bell Laboratorie) ของบริษัทเอทีแอนด์ที (AT&T) ในปัจจุบันมีระบบยูนิกซ์หลักอยู่ 2 ระบบคือ ระบบยูนิกซ์ SYSTEM V ซึ่งเป็นระบบยูนิกซ์ของบริษัทเอทีแอนด์ที และระบบยูนิกซ์ BSD 4.3 (Berkeley Software Distribution) ซึ่งเป็นระบบยูนิกซ์ของมหาวิทยาลัยแคลิฟอเนียที่เบอร์กเลย์ ระบบยูนิกซ์ได้ถูกออกแบบไว้ใช้สำหรับงานทั่ว ๆ ไปโดยมีสภาพแวดล้อมของระบบเป็นแบบเชิงโต้ตอบ (interactive) แบบหลายผู้ใช้ (multiuser) และแบบหลายภารกิจ (multitask) (AT&T, 1987 a)

โครงสร้างของระบบยูนิกซ์ (structure of UNIX system)

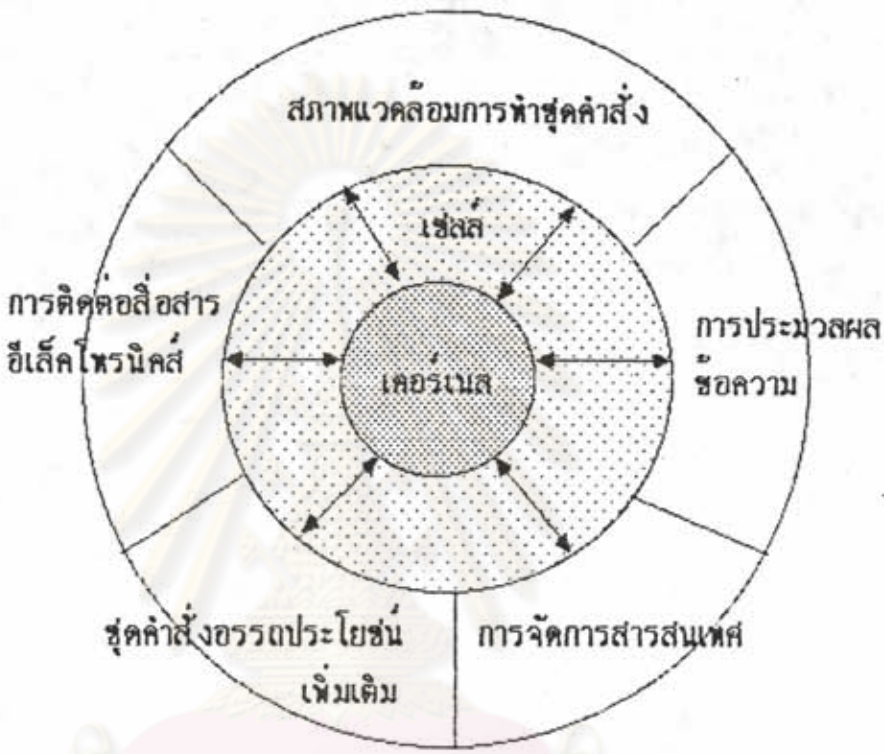
โครงสร้างของระบบยูนิกซ์ดังรูปที่ 2.1 ประกอบด้วยส่วนใหญ่ ๆ 3 ส่วนคือ

1. เคอร์เนล (kernel) เป็นหัวใจของระบบยูนิกซ์ ทำหน้าที่ควบคุมการทำงานของส่วนอุปกรณ์ (hardware) ของระบบคอมพิวเตอร์ จัดการหน่วยความจำและแบ่งปันทรัพยากรของระบบคอมพิวเตอร์ให้แก่ผู้ใช้

2. เชลล์ (shell) เป็นชุดคำสั่งที่ใช้ติดต่อกันระหว่างผู้ใช้กับเครื่องคอมพิวเตอร์ โดยชุดคำสั่งเชลล์ (shell program) จะรับคำสั่งงานจากผู้ใช้ ทำการแปลคำสั่งงานส่งไปให้เคอร์เนลกระทำการและรับข้อมูลผลลัพธ์จากเคอร์เนลส่งกลับไปให้ผู้ใช้

3. ชุดคำสั่งอรรถประโยชน์ (utility program) ประกอบด้วยชุดคำสั่งที่ใช้งานในด้านต่าง ๆ เช่น การประมวลผลข้อความ (text processing) การจัดการสารสนเทศ (information management) การสื่อสารอีเล็คโทรนิกส์ (electronic communication)

และการพัฒนาส่วนชุดคำสั่ง (software development) ชุดคำสั่งอรรถประโยชน์นี้ผู้ใช้สามารถสร้างขึ้นมาใหม่ตามความต้องการของผู้ใช้ได้ (AT&T, 1987 a)



รูปที่ 2.1 โครงสร้างของระบบยูนิกซ์

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ระบบแฟ้มข้อมูลบนระบบยูนิกซ์ (UNIX file system)

ระบบแฟ้มข้อมูลบนระบบยูนิกซ์มีลักษณะโครงสร้างเป็นระบบแฟ้มข้อมูลแบบลำดับชั้น (hirachical file system) ซึ่งแบ่งออกเป็น 3 ชนิดคือ

1. แฟ้มข้อมูลแบบธรรมดา (ordinary file) แฟ้มข้อมูลชนิดนี้ใช้เก็บข้อมูลในลักษณะข้อความ หรือชุดคำสั่งที่สามารถกระทำการได้ (execute program) โดยผู้ใช้สามารถที่จะสร้าง ลบและแก้ไขแฟ้มข้อมูลชนิดนี้ได้

2. แฟ้มข้อมูลแบบสารบบ (directory file) เป็นแฟ้มข้อมูลที่เก็บรายการของแฟ้มข้อมูลต่าง ๆ รายการเหล่านี้ประกอบด้วยชื่อแฟ้มข้อมูลและตัวชี้ไปยังตำแหน่งจริง ๆ ของแฟ้มข้อมูลบนจานบันทึก (disk) ผู้ใช้สามารถที่จะสร้างหรือลบแฟ้มข้อมูลแบบนี้ได้

3. แฟ้มข้อมูลแบบพิเศษ (special file) เป็นแฟ้มข้อมูลที่ใช้อ้างอิงกับส่วนอุปกรณ์ทางกายภาพ เช่น เครื่องปลายทาง (terminal) เครื่องพิมพ์ (printer) ตัวขับแถบบันทึก (tape driver) ตัวขับจานบันทึก (disk driver) เป็นต้น ผู้ใช้สามารถที่จะเขียนหรืออ่านข้อมูลจากแฟ้มข้อมูลแบบนี้ได้เหมือนกับแฟ้มข้อมูลแบบธรรมดา แต่จะมีการกระทำกับเครื่องอุปกรณ์ทางกายภาพจริง ๆ (AT&T, 1987 a)

คำสั่งงานบนระบบยูนิกซ์ (UNIX command)

ลักษณะคำสั่งงานของระบบยูนิกซ์มีรูปแบบดังนี้

command option(s) argument(s) <CR>

- command เป็นชื่อของชุดคำสั่ง
- option เป็นทางเลือกการทำงานในหน้าที่ต่าง ๆ ของชุดคำสั่ง
- argument เป็นข้อมูลหรือแฟ้มข้อมูลที่จะให้ชุดคำสั่งนำไปประมวลผล

การกระทำการของคำสั่งงาน (command execution)

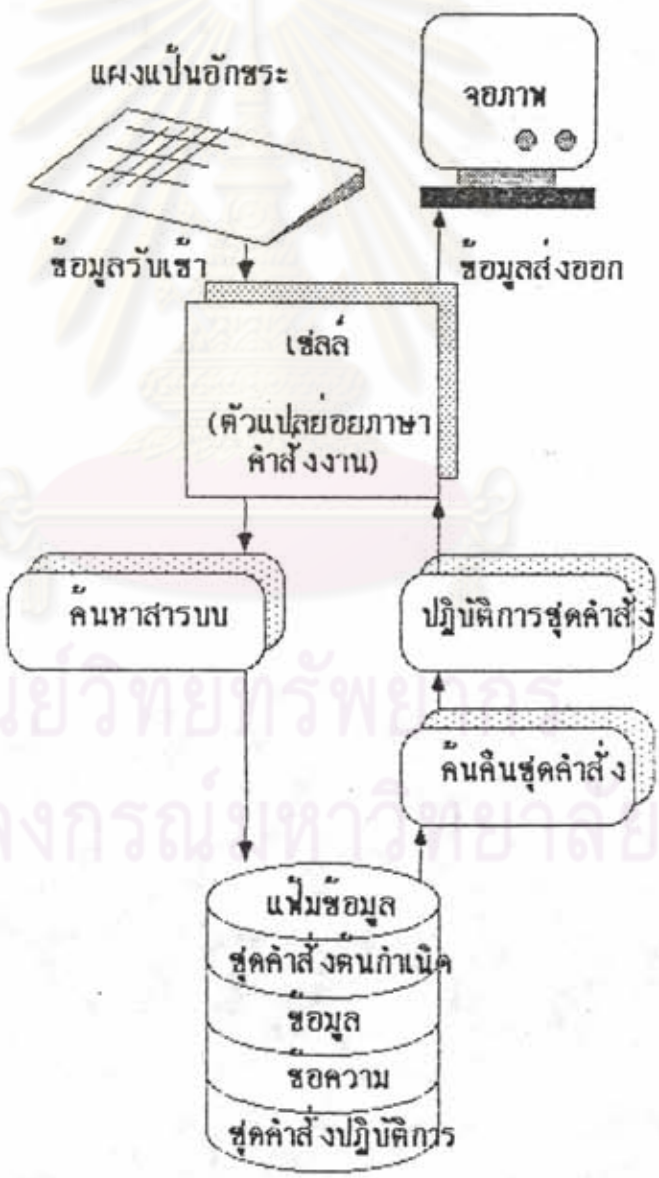
เมื่อมีการป้อนคำสั่งงานเข้ามาทางบรรทัดคำสั่งงาน ชุดคำสั่งเชลล์จะรับคำสั่งงานไปค้นหาในสารบบต่าง ๆ เพื่อจะดึงชุดคำสั่งที่ผู้ใช้ระบุขึ้นมา จากนั้นจะส่งไปให้เคอร์เนลเพื่อกระทำการคำสั่งงานต่าง ๆ ในชุดคำสั่ง เมื่อชุดคำสั่งมีการทำงานเสร็จเชลล์ก็จะส่งสัญญาณบอกให้ผู้ใช้ป้อนคำสั่งงานถัดไป โดยจะแสดงเครื่องหมายนำร้อม (prompt) ออกมา การกระทำการของคำสั่งงานในระบบยูนิกซ์แสดงได้ดังรูปที่ 2.2 (AT&T 1987 a)

การเก็บข้อมูลในระบบยูนิกซ์

การเก็บข้อมูลในระบบยูนิกซ์ จะเก็บเป็นลักษณะกระแสของไบต์ (stream of byte) โดยไม่มีโครงสร้างของข้อมูล ชุดคำสั่งจะใช้รหัสบรรทัดใหม่ (newline) เป็นตัวแยกระเบียน (record) และใช้รหัสตั้งระยะ (tab) หรืออักขระอื่น ๆ เป็นตัวแยกเขตข้อมูล (field) ทั้งระเบียนและเขตข้อมูลจะมีลักษณะแบบความยาวแปรได้ (variable length) ซึ่งจะทำให้ประหยัดเนื้อที่ในจานบันทึกมากกว่าการเก็บข้อมูลแบบความยาวตายตัว (fixed length)

การเบี่ยงเบนรับเข้า/ส่งออกมาตรฐาน (standard I/O and redirection)

หลักการของการรับเข้า/ส่งออกมาตรฐาน (standard I/O) คือชุดคำสั่งจะรับข้อมูลจากหน่วยรับเข้ามาตรฐาน (standard input) และส่งข้อมูลไปยังหน่วยส่งออกมาตรฐาน (standard output) โดยชุดคำสั่งไม่ต้องสนใจว่าข้อมูลที่รับเข้ามานั้นจะมาจากอุปกรณ์ชนิดใด เช่น แผงแป้นอักขระ (keyboard) จานบันทึกแข็ง (hard disk) แผ่นบันทึก (floppy disk) หน่วยขับแถบบันทึก (tape driver) หรืออุปกรณ์อื่น ๆ ในการใช้งานจะมีเครื่อง



รูปที่ 2.2 การกระทำการของคำสั่งงานในระบบยูนิกซ์

หมายนิเศษของชุดคำสั่งเชลล์ซึ่งเรียกว่าอักขระการเบี่ยงเบนแฟ้มข้อมูล (file redirection character) เป็นตัวบอกชุดคำสั่งว่าจะรับข้อมูลจากที่ใดและส่งข้อมูลไปยังที่ใด ค่าโดยปริยาย (default) ของหน่วยรับเข้ามาตรฐานคือแผงแป้นอักขระ และค่าโดยปริยายของหน่วยส่งออกมาตรฐานคือจอภาพของเครื่องปลายทาง (terminal screen) อักขระเบี่ยงเบนแฟ้มข้อมูลมีดังต่อไปนี้ (Manis and Meyer, 1986: 44-50)

เครื่องหมาย ">" เป็นการกำหนดการเบี่ยงเบนส่งออกว่าจะส่งข้อมูลผลลัพธ์ไปยังหน่วยส่งออกมาตรฐานชนิดใด

เครื่องหมาย "<" เป็นการกำหนดการเบี่ยงเบนรับเข้าว่าจะรับข้อมูลมาจากหน่วยรับเข้ามาตรฐานชนิดใด

เครื่องหมาย ">>" เป็นการกำหนดการเบี่ยงเบนส่งออกไปยังแฟ้มข้อมูล โดยถ้าแฟ้มข้อมูลที่ระบุยังไม่มีการสร้างไว้ ระบบก็จะสร้างขึ้นมาใหม่ แต่ถ้าแฟ้มข้อมูลมีอยู่แล้วก็จะเบี่ยงเบนผลลัพธ์ไปต่อท้ายข้อมูลของแฟ้มข้อมูลนั้น

เครื่องหมาย "<<" เป็นการกำหนดการเบี่ยงเบนรับเข้า โดยเป็นการป้อนข้อมูลให้กับชุดคำสั่งเชลล์และมีการระบุอักขระหรือคำเป็นตัวบอกจุดจบของข้อมูล

การระบุ "ตัวเลข < แฟ้มข้อมูล" ใช้ในการเปิดแฟ้มข้อมูลเป็นแฟ้มข้อมูลแบบส่งข้อมูลออก (input file) และกำหนดหมายเลขตัวพรรณนาแฟ้มข้อมูล (file descriptor number) ตามตัวเลขที่ระบุ

การระบุ "ตัวเลข > แฟ้มข้อมูล" ใช้ในการเปิดแฟ้มข้อมูลเป็นแฟ้มข้อมูลแบบรับข้อมูลเข้า (output file) และกำหนดหมายเลขตัวพรรณนาแฟ้มข้อมูลตามตัวเลขที่ระบุ

การระบุ "<& ตัวเลข" เป็นการเบี่ยงเบนรับเข้าจากแฟ้มข้อมูลโดยการระบุหมายเลขตัวพรรณนาแฟ้มข้อมูล

การระบุ ">& ตัวเลข" เป็นการเบี่ยงเบนส่งออกไปยังแฟ้มข้อมูลโดยการระบุหมายเลขตัวพรรณนาแฟ้มข้อมูล

เครื่องหมาย "<&-" เป็นการปิดหน่วยรับเข้ามาตรฐาน

เครื่องหมาย ">&-" เป็นการปิดหน่วยส่งออกมาตรฐาน

<u>ไพป์</u> (pipe)

การไพป์ เป็นการส่งผลลัพธ์ของชุดคำสั่งหนึ่งไปเป็นข้อมูลรับเข้าของอีกชุดคำสั่งหนึ่ง โดยใช้เครื่องหมาย "|" การทำงานของไพป์คือ เมื่อชุดคำสั่งเชลล์นับเครื่องหมายไพป์ก็จะกำหนดบัฟเฟอร์ภายใน (internal buffer) เพื่อใช้ในการเก็บข้อมูลผลลัพธ์ของคำสั่งทางซ้าย

ไว้ให้คำสั่งทางขวาอ่านข้อมูลนำไปประมวลผลต่อไป การใช้หลักการของไพป์จะช่วยให้ทำงานได้สะดวกและรวดเร็วขึ้น

ชุดคำสั่งเชลล์ (shell)

ชุดคำสั่งเชลล์ในปัจจุบันมีหลายตัว แต่ที่มีการใช้งานกันอย่างกว้างขวางมี 3 ตัวดังนี้คือ

1. บอร์นเชลล์ (Bourne Shell) พัฒนาโดย สตีเวน บอร์น (Steven Bourne) ที่ห้องวิจัยเบลล์ โดยมีลักษณะเป็นตัวแปลย่อยคำสั่งงาน (command interpreter) และภาษาทำชุดคำสั่ง (programming language) บอร์นเชลล์มีการติดตั้งบนเครื่องคอมพิวเตอร์ทุกเครื่องที่ใช้ระบบปฏิบัติการยูนิกซ์ ดังนั้นจึงทำให้มีการใช้งานบอร์นเชลล์กันอย่างกว้างขวาง

2. ซีเชลล์ (C Shell) พัฒนาโดย บิล จอย (Bill Joy) ที่มหาวิทยาลัยแคลิฟอเนียเบอร์กเลย์ มีลักษณะเป็นตัวแปลย่อยคำสั่งงานและภาษาทำชุดคำสั่งเช่นเดียวกับบอร์นเชลล์ แต่มีการเพิ่มคำสั่งงานเชิงโต้ตอบขึ้นมาอีกมายมาย ทำให้การใช้งานในลักษณะโต้ตอบมีความสะดวกรวดเร็วยิ่งขึ้น รูปแบบของภาษาทำชุดคำสั่งมีลักษณะคล้ายกับรูปแบบของชุดคำสั่งภาษาซี

3. คอร์นเชลล์ (Korn Shell) พัฒนาโดย เดวิด คอร์น (David Korn) ที่ห้องวิจัยเบลล์ ถูกออกแบบให้มีความเข้ากันได้กับบอร์นเชลล์ โดยการนำข้อดีของบอร์นเชลล์และซีเชลล์มารวมกัน มีการปรับปรุงให้เป็นภาษาทำชุดคำสั่งที่สมบูรณ์และเพิ่มความสามารถใหม่ ๆ ขึ้นมา แก้ไขข้อบกพร่องต่าง ๆ ที่เกิดขึ้นบนบอร์นเชลล์ รวมทั้งพัฒนาให้มีความเร็วเพิ่มขึ้น

สัญญลักษณ์นิเศษของชุดคำสั่งเชลล์

สัญญลักษณ์นิเศษของชุดคำสั่งเชลล์ (Manis and Meyer, 1986: 69-73) มีการใช้งานต่าง ๆ กัน ดังต่อไปนี้

1. เครื่องหมาย "*" แทนอักขระใด ๆ กี่ตัวก็ได้ในชื่อแฟ้มข้อมูล
2. เครื่องหมาย "?" แทนอักขระใด ๆ หนึ่งตัวในชื่อแฟ้มข้อมูล
3. เครื่องหมาย "[]" แทนอักขระใด ๆ หนึ่งตัวที่บรรจุอยู่ภายในเครื่องหมายนี้
4. เครื่องหมาย "[-]" แทนช่วงของข้อมูลที่ระบุภายในเครื่องหมายนี้
5. เครื่องหมาย "\" ใช้ฆ่าความหมายนิเศษของอักขระใด ๆ หนึ่งตัว
6. เครื่องหมาย "'" ใช้ฆ่าความหมายนิเศษของอักขระใด ๆ ที่บรรจุอยู่ภายในเครื่องหมายนี้
7. เครื่องหมาย """ ใช้ฆ่าความหมายนิเศษของอักขระใด ๆ ที่บรรจุอยู่ภายใน

เครื่องหมายนี้ ยกเว้นเครื่องหมายพิเศษ "\" "$" และ "*"

8. เครื่องหมาย "`" ใช้กระทำการคำสั่งงานที่อยู่ภายในเครื่องหมายนี้

ตัวแปรชุดคำสั่งเชลล์ (shell variable)

ตัวแปรของชุดคำสั่งเชลล์มีชนิดเดียวเท่านั้นคือ ชนิดสายอักขระ (string) การกำหนดค่าของตัวแปรใช้เครื่องหมายเท่ากับ ("=") การเรียกใช้ตัวแปรจะใช้เครื่องหมาย "$" นำหน้าชื่อตัวแปรนั้น ตัวแปรชุดคำสั่งเชลล์แบ่งออกเป็น 6 ชนิดดังต่อไปนี้

1. ตัวแปรที่ผู้ใช้กำหนด เป็นตัวแปรที่ผู้ใช้กำหนดขึ้นมาเอง ชื่อตัวแปรจะต้องขึ้นต้นด้วยตัวอักษร ตามด้วยตัวอักษร ตัวเลขหรือเครื่องหมาย "_" ก็ได้

2. ตัวแปรมาตรฐาน เป็นตัวแปรที่ระบบยูนิกซ์กำหนดให้เมื่อมีการเข้ามาในระบบ ตัวแปรเหล่านี้สามารถเปลี่ยนแปลงค่าได้ เช่น

HOME ตัวแปรกำหนดเส้นทางของสารบบเหย้าของผู้ใช้ (home directory)

PATH ตัวแปรกำหนดเส้นทางของชุดคำสั่งเชลล์

PS1 ตัวแปรกำหนดเครื่องหมายพร้อมที่ 1

PS2 ตัวแปรกำหนดเครื่องหมายพร้อมที่ 2

IFS ตัวกำหนดอักขระแยกเขตข้อมูลภายใน (internal field seperate)

TERM ตัวแปรกำหนดชื่อของเครื่องปลายทาง

TERMCAP ตัวแปรกำหนดรายละเอียดต่าง ๆ ของเครื่องปลายทาง

3. ตัวแปรที่กำหนดค่าโดยชุดคำสั่งเชลล์ (Manis and Meyer, 1986: 84-85) มีดังต่อไปนี้

$* เป็นตัวแปรที่ใช้เก็บค่าตัวแปรพารามิเตอร์ทั้งหมด

$@ เป็นตัวแปรที่ใช้เก็บค่าตัวแปรพารามิเตอร์ทั้งหมด

$# เป็นตัวแปรที่ใช้เก็บจำนวนของอาร์กิวเมนต์ที่ส่งให้กับชุดคำสั่งเชลล์

$? เป็นตัวแปรที่ใช้เก็บสถานะการทำงานของคำสั่งการสุดท้าย

$$ เป็นตัวแปรที่ใช้เก็บหมายเลขโปรเซสของชุดคำสั่งเชลล์

$! เป็นตัวแปรที่ใช้เก็บหมายเลขโปรเซสของชุดคำสั่งเชลล์ที่กระทำการอยู่เบื้องหลัง

4. ตัวแปรที่ถูกกำหนดโดยปริยาย (Manis and Meyer, 1986: 85-87) มีลักษณะดังต่อไปนี้

${พารามิเตอร์} ตัวแปรจะมีค่าเท่ากับค่าพารามิเตอร์ที่กำหนด

${พารามิเตอร์:-คำ} ถ้ามีการกำหนดค่าพารามิเตอร์ ตัวแปรจะมีค่าเท่ากับค่าพารามิเตอร์นั้น แต่ถ้าไม่มีการกำหนดค่าพารามิเตอร์ ตัวแปรจะมีค่าเท่ากับคำ

${พารามิเตอร์:=คำ} ถ้ามีการกำหนดค่าพารามิเตอร์ไว้ ตัวแปรจะมีค่าเท่ากับค่าพารามิเตอร์นั้น ถ้าไม่มีการกำหนดค่าพารามิเตอร์ตัวแปรจะมีค่าเท่ากับคำและมีการกำหนดค่าพารามิเตอร์ให้มีค่าเท่ากับคำด้วย

${พารามิเตอร์:?คำ} ถ้ามีการกำหนดค่าพารามิเตอร์ไว้ ตัวแปรจะมีค่าเท่ากับค่าพารามิเตอร์นั้น ถ้าไม่มีการกำหนดค่าพารามิเตอร์ ตัวแปรจะมีค่าเป็นคำนั้น

${พารามิเตอร์:+คำ} ถ้ามีการกำหนดค่าพารามิเตอร์ไว้ ตัวแปรจะมีค่าเท่ากับคำ ถ้าไม่มีการกำหนดค่าพารามิเตอร์ จะไม่มีการแทนค่าตัวแปร

5. ตัวแปรตามตำแหน่ง เป็นตัวแปรที่รับค่ามาจากอาร์กิวเมนต์ที่ส่งให้กับชุดคำสั่งเชลล์ โดยส่งค่าอาร์กิวเมนต์ให้กับตัวแปรในตำแหน่งนั้น ๆ เช่น

$1 แทนอาร์กิวเมนต์ตัวแรกของบรรทัดคำสั่งงาน

$2 แทนอาร์กิวเมนต์ตัวที่สองของบรรทัดคำสั่งงาน

ทั้งนี้มีได้ 9 ตัว คือ $1 ถึง $9

6. ตัวแปรคำสั่ง เป็นตัวแปรที่ถูกกำหนดค่าด้วยคำสั่งงาน และจะแสดงผลลัพธ์ที่ได้จากการทำงานของคำสั่งงานนั้นเมื่อมีการเรียกใช้

การควบคุมสายงาน (control flow)

การควบคุมสายงานในระบบยูนิกซ์ (AT&T, 1987 b) ประกอบด้วยประโยคต่าง ๆ ดังต่อไปนี้

1. ประโยคเงื่อนไข if

```
if เงื่อนไข ; then กลุ่มชุดคำสั่ง ; else กลุ่มชุดคำสั่ง ; fi
```

2. การวนลูป while

```
while เงื่อนไข ; do กลุ่มชุดคำสั่ง ; done
```

3. การวนลูป until

```
until เงื่อนไข ; do กลุ่มชุดคำสั่ง ; done
```

4. ประโยค for

```
for ชื่อตัวแปร in คำที่ 1  คำที่ 2 ... ; do กลุ่มชุดคำสั่ง ; done
```

5. ประโยค case

```
case คำ in  รูปแบบที่ 1 ) กลุ่มชุดคำสั่ง ;; ... ; esac
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

คำสั่งงานภายในชุดคำสั่งเชลล์ (shell built-in command)

คำสั่งงานภายในชุดคำสั่งเชลล์เป็นคำสั่งที่สามารถเรียกใช้ได้โดยที่ชุดคำสั่งเชลล์ไม่ต้องไปค้นหาจากสารบบ ดังตัวอย่างต่อไปนี้

break	ใช้ในการบังคับให้การทำงานหลุดออกจากลูป
continue	ใช้ในการบังคับให้การทำงานกลับไปยังจุดเริ่มต้นของลูป
cd	ใช้เปลี่ยนสารบบปัจจุบัน
echo	ใช้เขียนอาร์กิวเมนต์ไปยังหน่วยส่งออกมาตรฐาน
read	อ่านข้อมูลจากเครื่องปลายทางให้แก่ตัวแปร
exit	ใช้หยุดการกระทำการของชุดคำสั่งเชลล์
test	ใช้ทดสอบแฟ้มข้อมูล เปรียบเทียบค่าของตัวเลขหรือเปรียบเทียบข้อความ
trap	ใช้ดักจับสัญญาณต่าง ๆ ที่เกิดขึ้นจากชุดคำสั่งหรือเกิดจากระบบยูนิกซ์

คำสั่งงานของระบบยูนิกซ์ (UNIX command)

ในระบบยูนิกซ์มีคำสั่งงานเพื่อใช้งานในด้านต่าง ๆ มากมายเพียงพอสำหรับการพัฒนาระบบงานและสร้างชุดคำสั่งอรรถประโยชน์ขึ้นมาใช้งานได้ ดังตัวอย่างต่อไปนี้

1. คำสั่งงานจัดการแฟ้มข้อมูล เช่น cat cp mkdir find
2. คำสั่งงานปฏิบัติการชุดคำสั่ง เช่น test expr read echo kill at
3. คำสั่งงานติดต่อสื่อสาร เช่น mail write wall uucp
4. คำสั่งงานประมวลผลข้อความ เช่น head tail sed awk grep sort wc diff
5. คำสั่งงานคำนวณตัวเลข เช่น bc dc
6. คำสั่งงานพัฒนาชุดคำสั่งงาน เช่น make

ชุดคำสั่งเชลล์สคริป (shell script)

ชุดคำสั่งเชลล์สคริป เป็นการนำคำสั่งงานของระบบยูนิกซ์มาเขียนเป็นลักษณะชุดคำสั่งงาน มีการควบคุมสายงานให้เป็นไปตามที่ผู้ใช้ต้องการ จากนั้นมีการบันทึกชุดคำสั่งเหล่านี้ลงแฟ้มข้อมูลและเปลี่ยนชนิดของแฟ้มข้อมูลนี้ให้สามารถปฏิบัติการ (execute) ได้ การเรียกใช้ชุดคำสั่งการก็จะสามารถเรียกใช้ได้จากเครื่องหมายพร้อมของชุดคำสั่งเชลล์ได้เลย