UNIX

ความรู้พื้นฐาน

1. Unix เป็นระบบปฏิบัติการแบบ Multi User และ Multi Tasking ซึ่ง กล่าวคือ ณ เวลาหนึ่งๆ บนระบบ Unix จะมีผู้ใช้งานเครื่องคอมพิวเตอร์ได้มากกว่า 1 คนพร้อมกัน ทำให้ Unix มีระบบการจัดการ Permission และระบบรักษาความปลอดภัยของข้อมูล
2. ระบบ File System ของ Unix นั้นจะเป็นระบบ Single Root คือจะมี Logical Driver เพียง Drive เดียวเท่านั้น และกรณีมี Harddisk หลายตัวหรือหลาย Partition แต่ละ Partition จะถูกกำหนดให้เป็นเพียง Directory ย่อยของระบบ ซึ่งจะต่างกับ DOS/Window ที่เป็นระบบ Multiple Root ที่จะแยก Drive / Parition ตามตัวอักษร เช่น A: , C: เป็นต้น
3. เนื่องจาก Unix เป็นระบบปฏิบัติการที่พัฒนาด้วยภาษา C ดังนั้นชื่อต่างๆ บน Unix จึงมีลักษณะเป็น Case-sensitive เช่น กรณีแฟ้มข้อมูลชื่อ MyFile กับ myfile จะเป็นแฟ้มข้อมูลคนละชื่อกัน
4. ระบบ Permission ของ Unix จะแบ่งเป็น 3 ระดับคือ ระดับเจ้าของ (User หรือ Owner) ระดับกลุ่ม (Group) และ ระดับบุคคลอื่น (Other) โดยในแต่ละระดับจะแบ่งออกเป็นสิทธิในการประมวลผล (execute) การอ่าน (read) และ การเขียน (write) ทั้งรายละเอียดเพิ่มเติมให้ดูจากคำสั่ง chmod
5. กรณีที่ผู้ใช้กระทำคำสั่งใดผิดพลาดนั้น บน Unix เราสามารถที่จะ Interrupt เพื่อยกเลิกการทำงานของคำสั่งหรือโปรแกรมนั้นๆ ได้โดยการกด CTRL + C
6. มาตรฐานของระบบ Keyboard บนเครื่อง Unix บางเครื่องอาจจะแตกต่างกับมาตรฐาน Keyboard บนเครื่องที่เราใช้อยู่ ดังนั้นในบางกรณี เช่น การ telnet จากเครื่องอื่นเข้าสู่ระบบ Unix เราจึงไม่อาจใช้ Key บางอันตามปกติได้ เช่น backspace ดังนั้นเพื่ออำนวยความสะดวกให้เราสามารถใช้ backspace ได้ตามปกติจึงต้องมีการ map key ใหม่ด้วยการเรียกคำสั่ง stty erase [backspace]

คำสั่งเกี่ยวกับการจัดการแฟ้มข้อมูล

ls เป็นคำสั่งที่ใช้สำหรับแสดงแฟ้มข้อมูลของไดเร็กทอรี่ปัจจุบัน มากจากคำว่า list

โครงสร้างคำสั่ง ls [option]... [file]...

โดย option ที่มักใช้กันใน ls คือ

-l จะแสดงผลลัพธ์แบบ Long Format ซึ่งจะแสดง Permission ของแฟ้มด้วย ตัวอย่าง ls -l

-a จะแสดงแฟ้มข้อมูลทั้งหมด ตัวอย่าง ls -a

-F จะแสดง / หลัง Directory และ * หลังแฟ้มข้อมูลที่ execute ได้ ตัวอย่าง ls-F

cp (copy) เป็นคำสั่งที่ใช้สำหรับสำเนาแฟ้มข้อมูล

โครงสร้างคำสั่ง cp file1 file2

ตัวอย่าง cp library.txt library.bak

mv (move) เป็นคำสั่งที่ใช้สำหรับการย้ายแฟ้มข้อมูลและ Directory รวมถึงการเปลี่ยนชื่อด้วย

โครงสร้างคำสั่ง mv source target

ตัวอย่าง mv *.txt mv library.txt old.txt

rm (remove) เป็นคำสั่งที่ใช้สำหรับลบแฟ้มข้อมูล

โครงสร้างคำสั่ง rm [option]... [file]...

option ที่นิยมใช้ใน rm คือ -r ทำการลบข้อมูลใน directory ย่อยทั้งหมด

-i โปรแกรมจะถามยืนยันก่อนทำการลบ

-f โปรแกรมจะลบข้อมูลทันที โดยไม่ถามยืนยันก่อน

ตัวอย่าง rm –tempfile.txt

คำสั่งเกี่ยวกับการจัดการ Directory / Folder

mkdir (make directory) เป็นคำสั่งที่ใช้สำหรับการสร้าง directory

โครงสร้างคำสั่ง mkdir [option]... [file]...

option ที่ใช้ใน mkdir คือ

-m จะทำการกำหนด Permissioin (ให้ดูคำสั่ง chmod เพิ่มเติม)

-p จะทำการสร้าง Parent Directory ให้ด้วยกรณีที่ยังไม่มีการระบุ

directory ในที่นี้อาจเป็น relative หรือ absolute path ก็ได้

ตัวอย่าง mddir Unixstuff

rmdir (remove directory) เป็นคำสั่งที่ใช้สำหรับการลบ directory

โครงสร้างคำสั่ง rmdir [option]... [file]...

option ที่ใช้ใน mkdir คือ -p จะทำการลบ Child และ Parent Directory ตามลำดับ

directory ในที่นี้อาจเป็น relative หรือ absolute path ก็ได้

ตัวอย่าง rmdir /home

cd (change directory) เป็นคำสั่งที่ใช้สำหรับเปลี่ยน directory ปัจจุบัน

โครงสร้างคำสั่ง cd directory

ตัวอย่าง cd unixstuff

cd ~ (เป็นการเข้าสู่ home directory)

cd - (เป็นการยกเลิกคำสั่ง cd ครั้งก่อน)

cd .. (เป็นการออกจาก directory 1 ชั้น)

ข้อควรระวัง : คำสั่ง cd บน UNIX จะต้องมีเว้นวรรคเสมอ

pwd (print work directory) เป็นคำสั่งที่ใช้สำหรับแสดง Directory ปัจจุบัน

โครงสร้างคำสั่ง pwd

file บนระบบ DOS/Windows นั้น ประเภทของแฟ้มข้อมูลจะถูกระบุด้วยนามสกุล แต่ใน UNIX จะไม่มีนามสกุลเพื่อใช้ระบุประเภทของแฟ้มข้อมูล ดังนั้นการหาประเภทของแฟ้มข้อมูลจะดูจาก Context ภายในของแฟ้ม ซึ่งคำสั่ง file จะทำการอ่าน Content และบอกประเภทของแฟ้มข้อมูลนั้นๆ

โครงสร้างคำสั่ง file [option]... file

ตัวอย่าง file library.doc

find เป็นคำสั่งที่ใช้สำหรับค้นหาแฟ้มข้อมูล

โครงสร้างคำสั่ง find [path].. expression

ลักษณะของ expression เช่น

-name [pattern] เพื่อใช้หาชื่อ file ตาม pattern ที่ระบุ

-perm [+-] mode เพื่อใช้หา file ตาม mode ที่ต้องการ

-user NAME หา file ที่เป็นของ user ชื่อ NAME

-group NAME หา file ที่เป็นของ group ชื่อ NAME

ตัวอย่าง find -name *.doc

คำสั่งเกี่ยวกับการดู และ แก้ไขข้อมูลในแฟ้มข้อมูล

cat (concatinate**)** ใช้สำหรับดูข้อมูลภายในแฟ้มข้อมูล หรือ Standard Input และแสดงผลออกมาทาง Standard Output

โครงสร้างคำสั่ง cat [optioin]... [file]

option ที่นิยมใช้ใน cat คือ

-n เพื่อทำการแสดงเลขบรรทัด

ตัวอย่าง cat data.txt

cat file1.txt file2.txt > file3.txt (นำข้อมูลใน file1.txt และ file2.txt มาต่อกัน แล้วเก็บไว้ใน file3.txt)

more คำสั่ง cat ไม่เหมาะกับการดูข้อมูลที่มีความยาวมากๆ ถ้าต้องการดูข้อมูลในหน้าต่อไปต้องใช้คำสั่ง more เพื่อช่วยให้สามารถดูข้อมูลที่มีขนาดยาวได้เป็นช่วงๆ

โครงสร้างคำสั่ง **more file** ภายในโปรแกรม more จะมีคำสั่งเพื่อใช้งาน ดังนี้

= แสดงเลขบรรทัด

q ออกจากโปรแกรม

<space> เลื่อนไปยังหน้าถัดไป

<enter> เลื่อนไปยังบรรทัดถัดไป

h แสดง help

ตัวอย่าง more data.txt

less เป็นการพัฒนาคำสั่ง more ให้มีประสิทธิภาพมากขึ้น เนื่องจาก more จะไม่สามารถดูข้อมูลย้อนหลังได้ less จึงเป็นปรับปรุงและเพิ่มเติมเงื่อนไขบางอย่างให้ more

โครงสร้างคำสั่ง less file

ตัวอย่าง less data.txt

head จะแสดงส่วนหัวของแฟ้มข้อมูล ตามจำนวนบรรทัดที่ต้องการ

โครงสร้างคำสั่ง head [option] file

option ที่นิยมใช้ใน head คือ

-n เพื่อทำการระบุบรรทัดที่ต้องการ (หากไม่ระบุจะเป็น 10 บรรทัด)

ตัวอย่าง head data.txt

head -n 10 data.txt

tail แสดงส่วนท้ายของแฟ้มข้อมูล ตามจำนวนบรรทัดที่ต้องการ

โครงสร้างคำสั่ง tail [option] file

option ที่นิยมใช้ใน tail คือ

-n เพื่อทำการระบุบรรทัดที่ต้องการ (หากไม่ระบุจะเป็น 10 บรรทัด)

-c เพื่อระบุจำนวน byte

ตัวอย่าง tail library.txt tail -n 10 data.txt

คำสั่งเกี่ยวกับผู้ใช้ และ การสื่อสาร

whoami ใช้เพื่อแสดงว่าผู้ใช้ซึ่ง login เข้าสู่ระบบนั้น (ตัวเราเอง) login ด้วยชื่ออะไร

โครงสร้างคำสั่ง whoami หรือ who am i (บน SUN OS หรือ UNIX บางตัวเท่านั้น)

who ใช้เพื่อแสดงว่ามีผู้ใช้ใดบ้างที่กำลังทำงานอยู่บนระบบ

โครงสร้างคำสั่ง who

finger ใช้สำหรับแสดงรายละเอียดของผู้ใช้

โครงสร้างคำสั่ง finger [user@host] หรือ finger [@host]

กรณีไม่ระบุชื่อ finger จะแสดงรายละเอียดของ User ที่กำลัง logon อยู่บนเครื่องนั้นๆ ทั้งหมด ซึ่งหากไม่ระบุ host ด้วย โปรแกรมจะถือว่าหมายถึงเครื่องปัจจุบัน

finger @student.netserv.chula.ac.th

แหล่งข้อมูลเพิ่มเติม : man finger

talk ใช้สำหรับการพูดคุยระหว่างผู้ใช้ด้วยกันบนระบบ ซึ่งผู้ใช้ทั้งทั้ง 2 ฝ่ายจะต้องพิมพ์คำสั่ง Talk ถึงกันก่อน จึงจะเริ่มการสนทนาได้

โครงสร้างคำสั่ง talk user[@host] [tty]

กรณีไม่ระบุ host โปรแกรมจะถือว่าหมายถึงเครื่องปัจจุบัน (นอกจากนี้ยังมีคำสั่ง ytalk ซึ่งสามารถพูดคุยได้พร้อมกันมากกว่า 2 คน) ซึงบางกรณีเราอาจจะต้องระบุ tty ด้วยหากมีผู้ใช้ Log in เข้าสู่ระบบด้วยชื่อเดียวกันมากกว่า 1 หน้าจอ

ตัวอย่าง talk vduangna@pioneer.netserv.chula.ac.th

write จะใช้เพื่อการส่งข้อมูลทางเดียวจากผู้เขียนไปถึงผู้รับบนเครื่องเดียวกันเท่านั้น

โครงสร้างคำสั่ง write user [tty]

เมื่อมีการพิมพ์คำสั่ง write ผู้ใช้จะเห็นข้อความซึ่งจะแสดงว่าข้อความดังกล่าวถูกส่งมาโดยใคร ซึ่งหากผู้รับต้องการตอบกลับ ก็จะต้องใช้คำสั่ง write เช่นกัน เมื่อพิมพ์เสร็จแล้วให้พิมพ์ตัวอักษร EOF หรือ กด CTRL+C เพื่อเป็นการ interrupt ทั้งนี้ข้อความที่พิมพ์หลังจาก write จะถูกส่งหลังจากการกด Enter เท่านั้น

ตัวอย่าง write duangnate

mesg จะใช้เพื่อควบคุมว่าผู้อื่นมีสิทธิที่จะส่งข้อความ write ถึงเราหรือไม่

โครงสร้างคำสั่ง mesg [y | n]

โดย option มีความหมายคือ

y - หมายถึงผู้อื่นมีสิทธิที่จะส่งข้อความถึงเรา

n - หมายถึงผู้อื่นมีไม่สิทธิที่จะส่งข้อความถึงเรา

ตัวอย่าง mesg y mesg n

คำสั่งทั่วไป / อื่นๆ

man(manual) เพื่อใช้แสดงรายละเอียดข้อมูลของคำสั่ง หรือ วิธีการใช้แฟ้มข้อมูลต่างๆ

โครงสร้างคำสั่ง man [section]... manpage

โดย section ต่างๆ ของ manpage คือ

1 จะเป็น User Command

2 จะเป็น System Calls

3 จะเป็น Sub Routines

4 จะเป็น Devices

5 จะเป็น File Format

ตัวอย่าง man printf man 1 ls

tar ใช้เพื่อการ backup และ restore file ทั้งนี้การ tar จะเก็บทั้งโครงสร้าง directory และ file permission ด้วย (เหมาะสำหรับการเคลื่อนย้าย หรือแจกจ่ายโปรแกรมบนระบบ UNIX) มาจากคำว่า tape archive

โครงสร้างคำสั่ง

tar [option]... [file]...

option ที่นิยมใช้ใน tar คือ

-c ทำการสร้างใหม่ (backup)

-t แสดงรายชื่อแฟ้มข้อมูลในแฟ้มที่ backup ไว้

-v ตรวจสอบความถูกต้องของการประมวลผล

-f ผลลัพธ์ของมาที่ file

-x ทำการ restore

ตัวอย่าง tar -cvf mybackup.tar /home/*

tar -tf mybackup.tar

tar -xvf mybackup.tar

alias เพื่อกำหนด macro ให้ใช้คำสั่งได้สะดวกมากขึ้น (แบบเดียวกันกับการกำหนด macro ด้วย doskey)

โครงสร้างคำสั่ง alias macroname='command'

ตัวอย่าง alias ll='ls -F -l'

echo แสดงข้อความออกทาง standard output

โครงสร้างคำสั่ง echo [option]... msg

option ที่นิยมใช้ใน echo คือ -n ไม่ต้องขึ้นบรรทัดใหม่

ตัวอย่าง echo -n "Hello" echo "Hi.."

free แสดงหน่วยความจำที่เหลืออยู่บนระบบ

โครงสร้างคำสั่ง free [-b|-k|-m]

option ที่ใช้ใน free คือ

-b แสดงผลลัพธ์เป็นหน่วย byte

-k แสดงผลลัพธ์เป็นหน่วย kilobyte

-m แสดงผลลัพธ์เป็นหน่วย megabyte

ตัวอย่าง free free –b free –k

sort ใช้เพื่อทำการจัดเรียงข้อมูลในแฟ้มตามลำดับ (ทั้งนี้จะถือว่าข้อมูลแต่ละบรรทัดเป็น 1 record และจะใช้ field แรกเป็น key)

โครงสร้างคำสั่ง sort [option] file

ตัวอย่าง sort data.txt

วัน เวลา และปฏิทิน

date

cal cal 2004 cal 2004 | more

การใช้ basic calculator

bc ใส่ตัวเลขที่ต้องการคำนวณ ถ้าต้องการออกจากระบบให้ใช้คำสั่ง quit