



อานันท์ สิริพิทักษ์เกียรติ

ภาควิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเชียงใหม่

โปรดอ่าน

เอกสารนี้จัดทำขึ้นโดยมุ่งหวังให้เป็นประโยชน์แก่ผู้คนที่ไปศึกษาหาความรู้ในสาขาวิชานี้ ท่านสามารถนำเอกสารนี้ไปศึกษาได้โดยไม่ต้องเสียค่าใช้จ่ายใดๆ รวมทั้งการนำไปตัดแปลงเพื่อให้เหมาะสมกับความต้องการของท่าน ขอเพียงรักษาชื่อผู้แต่งและหน่วยงานที่ได้จัดทำต้นฉบับนี้ไว้ ผู้จัดทำได้พยายามเขียนเอกสารนี้ให้มีความถูกต้องมากที่สุด อย่างไรก็ตามขอความหรือเนื้อหาในเอกสารนี้ผู้จัดทำไม่สามารถรับประกันได้ว่าจะมีความถูกต้องสมบูรณ์ทุกอย่าง ผู้จัดทำไม่ขอรับผิดชอบต่อความเสียหายไม่ว่ากรณีใดๆ ที่อาจเกิดขึ้นจากการศึกษาหรือใช้เอกสารนี้

ถ้าท่านมีคำแนะนำเชิญส่งมาได้ที่

อ.อานันท์ สีสพิทักษ์เกียรติ

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

มหาวิทยาลัยเชียงใหม่

e-mail : aman@ns.eng.cmu.ac.th

คำนำ

เอกสารประกอบการอบรมเรื่องการใช้งาน UNIX พื้นฐานนี้ ได้จัดทำขึ้นมาเพื่อเป็นแนวทางเบื้องต้นให้กับผู้ที่สนใจศึกษาระบบปฏิบัติการตัวนี้ เนื่องจากการใช้งาน UNIX นั้นค่อนข้างที่จะมีเนื้อหา มาก ปกติแล้วผู้ที่ฝึกฝนจะต้องใช้เวลาหลายเดือนกว่าที่จะทำความเข้าใจกับตัวระบบและเข้าใจวิธีการ ใช้คำสั่งพื้นฐานของ UNIX ได้ทั้งหมด ผู้ที่ไม่ค่อยมีโอกาสได้ใช้งาน UNIX ในชีวิตการทำงานประจำวัน อาจประสบปัญหาในการจดจำและพัฒนาทักษะการใช้งานได้ ผู้เรียน UNIX นั้นต้องอาศัยการฝึกฝน ค่อนข้างมาก อย่างไรก็ตามคู่มือเล่มนี้ก็ได้พยายามที่จะสรุปเนื้อหาเบื้องต้นออกมาให้ดูง่ายที่สุด และหวังว่าจะเป็นประโยชน์กับผู้สนใจ หากว่ามีข้อผิดพลาดประการใดในเอกสารนี้ผู้เขียนต้องขออภัยมา ณ ที่นี้ และยินดีรับข้อเสนอแนะ เพื่อจะได้แก้ไขให้ถูกต้องต่อไป

อานันท์ สิริพิทักษ์เกียรติ

ภาควิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเชียงใหม่
มกราคม 2540

สารบัญ

บทที่ 1 ระบบปฏิบัติการ UNIX

1.1 ระบบปฏิบัติการ	1-1
1.2 ประวัติความเป็นมาของ UNIX	1-1
1.3 คุณสมบัติของระบบ UNIX	1-2
1.4 โครงสร้างของระบบ UNIX	1-2
1.5 โครงสร้าง Directory ใน UNIX	1-3

บทที่ 2 คำสั่ง UNIX พื้นฐาน 1

2.1 การ Login และ Logout	2-1
2.2 กลุ่มคำสั่งที่ใช้จัดการเกี่ยวกับไฟล์	2-1
- ls	
- file	
- cp, mv, rm	
- cat	
2.3 คำสั่งที่ใช้จัดการ directory	2-4
- pwd	
- cd	
- mkdir , rmdir	
2.4 คำสั่งที่ใช้จัดการเกี่ยวกับ Terminal และ user	2-4
- stty	
- who	
- passwd	
2.5 คำสั่งติดต่อสื่อสารระหว่าง user	2-5
- write	
- mesg	
- talk	
- mail	
2.6 คำสั่งเบื้องต้นอื่นๆ	2-7
- clear	
- cal	
- alias	

บทที่ 3 คำสั่ง UNIX พื้นฐาน 2

3.1 การค้นหาไฟล์ และระบบรักษาความปลอดภัยของไฟล์	3-1
- find	
- which	
- chmod	
- chown	
- chgrp	
3.2 การจัดการ Process	3-3
- เครื่องหมาย &	
- ps	
- kill	

3.3 Piping	3-4
- more	
- grep	
- wc	
- sort	
3.4 Shell กับตัวแปร Environment	3-6
- การตั้งค่าตัวแปร Environment	
- Start-up files	

บทที่ 4 UNIX System Administration เบื้องต้น

4.1 Super User	4-1
4.2 การ Shutdown และ Reboot ระบบ	4-1
4.3 การจัดการเกี่ยวกับ user และ group	4-1
- useradd	
- userdel	
- การเปลี่ยนรายละเอียดของ user และการ disable user	4-3
4.4 การ backup ข้อมูล	4-4
- tar	
- การย่อขนาดไฟล์	
4.5 การจัดการ file system และ disk	4-5
- การสร้าง file system	
- การ mount และ unmount file system	

บทที่ 5 การใช้ vi

5.1 โหมดการทำงานของ vi	5-1
5.2 เริ่มต้นใช้งาน vi	5-1
5.3 คำสั่งต่างๆ ใน Command Mode	5-2
- คำสั่งในการเลื่อน cursor	5-2
- คำสั่งที่ใช้ในการลบ	5-3
- คำสั่งที่ใช้ในการแทรก	5-4
- คำสั่งที่ใช้ในการค้นหา	5-4
- การ Copy, cut และ Paste ข้อความ	5-4
- คำสั่งอื่นๆ	5-5

บทที่ 6 บริการพื้นฐานของเครือข่าย INTERNET

6.1 บริการตรวจสอบเครื่องบนระบบเครือข่าย	6-1
- ping	
- nslookup	
- finger	
6.2 Telnet	6-2
6.3 pine	6-3
6.4 FTP	6-4

บทที่ 1

ระบบปฏิบัติการ UNIX

1.1 ระบบปฏิบัติการ

คอมพิวเตอร์ทุกเครื่องไม่ว่าจะเป็นเครื่องแบบใดจะต้องมีระบบปฏิบัติการทำงานอยู่เสมอ สาเหตุที่คอมพิวเตอร์จำเป็นต้องใช้ระบบปฏิบัติการคือ

ตัวเครื่องคอมพิวเตอร์มีรายละเอียดการใช้งานที่ยุ่งยากซับซ้อน และผู้ใช้ทั่วไปย่อมไม่ต้องการทำความเข้าใจกับรายละเอียดเหล่านี้ ดังนั้นนักออกแบบจึงได้นำระบบปฏิบัติการเข้ามาแทรกไว้ตรงกลางระหว่างผู้ใช้กับเครื่องคอมพิวเตอร์ เพื่อ

1. คอยอำนวยความสะดวกในการใช้งานตัวเครื่อง
2. ควบคุมการทำงานของเครื่องให้มีประสิทธิภาพมากที่สุด
3. ป้องกันความผิดพลาดต่างๆ ที่อาจเกิดขึ้น ทั้งจากตัวเครื่อง และจากผู้ใช้

1.2 ประวัติความเป็นมาของ UNIX

- บริษัท AT&T ร่วมมือกับ AIT พัฒนาระบบ MULTICS ขึ้นมาในช่วงปี 1960
- MULTICS ทำท่าว่าจะไปไม่รอด AT&T จึงได้ถอนตัวออกจากโครงการ
- นักพัฒนาของ AT&T จึงได้นำความรู้และปัญหาจากโครงการ MULTICS มาพัฒนาระบบใหม่แล้วสร้างเป็นระบบปฏิบัติการขึ้นมา และใช้ชื่อว่า UNIX
- ในช่วงแรก UNIX ยังถูกใช้งานกับเครื่อง PDP-7 และ PDP-11 ของ AT&T เท่านั้น
- ในปี 1973 UNIX ถูกเขียนขึ้นใหม่ทั้งหมด โดยใช้ภาษา C ซึ่งเป็นจุดเปลี่ยนแปลงที่สำคัญมากที่สุดอย่างหนึ่งของ UNIX เพราะ
 - ทำให้ UNIX สามารถนำไปใช้ได้กับเครื่องหลายชนิด
 - แกไขตัวระบบได้ง่ายกว่าระบบปฏิบัติการอื่นที่เขียนโดยใช้ภาษา Assembly
 - ราคาถูกลง
- AT&T แจก Code ของ UNIX ไปให้กับมหาวิทยาลัยต่างๆ เพื่อหวังให้เป็นที่ยอมรับมากขึ้น แต่กลับทำให้เกิด UNIX ตัวใหม่ขึ้นมา เรียกว่า BSD UNIX (Berkeley Software Distribution UNIX) ซึ่งเขียนโดย University of California ที่ Berkeley ระบบ UNIX ตัวนี้แจกฟรี และกลายเป็น UNIX ที่ใช้กันอย่างกว้างขวางตัวหนึ่ง โดยเฉพาะในมหาวิทยาลัย
- เมื่อเครื่อง PC มีความสามารถสูงขึ้นและราคาถูกลงทำให้เกิด UNIX ที่ใช้บน PC ขึ้นมาชื่อว่า XENIX
- ในช่วงกลางทศวรรษ 1980 ได้มีการพัฒนา X-window ขึ้นมาทำให้การใช้งาน UNIX เริ่มมี Graphic User Interface

- AT&T ได้ทำการพัฒนา UNIX ของตนขึ้นมาเรื่อยๆ จนกระทั่งถึงรุ่น System V Release 4 (SVR4) AT&T ได้รวมข้อกำหนดและมาตรฐานต่างๆ ของ BSD UNIX และ XENIX เข้าไปด้วย ทำให้โปรแกรมที่ออกแบบมาสำหรับ BSD UNIX หรือ XENIX สามารถนำไปใช้บน SVR4 ได้ ซึ่งส่งผลให้กลุ่มนักพัฒนาและบริษัทอื่นวิตกว่า AT&T จะผูกขาดการกำหนดมาตรฐานของระบบ UNIX จึงได้รวมตัวกันจัดตั้ง Open Software Foundation (OSF) ขึ้นมาเพื่อวิจัยและกำหนดมาตรฐานต่างๆ ของระบบ UNIX ขึ้นมาป้องกันการผูกขาดของ AT&T
- ในปัจจุบันมีระบบ UNIX เกิดขึ้นมากมาย เช่น SCO, Linux , SUN-OS, DG-UX, ฯลฯ และ UNIX ยังคงเป็นที่นิยมใช้กันอยู่ ถึงแม้ว่าระบบปฏิบัติการใหม่ ๆ เช่น Windows NT เริ่มจะเข้ามามีบทบาทก็ตาม แต่ UNIX ก็ยังคงความนิยมอยู่ได้และจะยังถูกใช้ต่อไปอีกนาน

1.3 คุณสมบัติของระบบ UNIX

Software Tools	- โปรแกรมบน UNIX จะแบ่งตัวเองออกเป็นส่วนย่อยๆ และสามารถใช้งานส่วนย่อยเหล่านั้นร่วมกันระหว่างหลายๆ โปรแกรมได้
Portability	- เนื่องจาก UNIX สามารถนำไปใช้กับเครื่องแบบต่างๆ ได้มากมาย โปรแกรมที่ใช้งานบน UNIX จะสามารถนำไปใช้ได้ด้วยเช่นเดียวกัน
Flexibility	- UNIX มีความยืดหยุ่นในการใช้งานสูง สามารถนำไปใช้กับงานเล็กๆ หรืองานใหญ่ๆ ก็ได้
Power	- สามารถทำงานที่ซับซ้อนได้ดี และมีความสามารถในด้านต่างๆ มากกว่าระบบปฏิบัติการอื่นๆ
Multi-user & Multitasking	- สามารถมีผู้ใช้งานได้ทีละหลายๆ คน และทำงานหลายๆ อย่างได้พร้อมกัน
Elegance	- หลักการทำงานของส่วนต่างๆ จะเหมือนกัน ดังนั้นเมื่อผู้ใช้งานส่วนหนึ่งได้ ก็จะเรียนรู้และใช้งานส่วนอื่นได้ง่าย
Network Orientation	- UNIX เป็นระบบปฏิบัติการที่สนับสนุนการใช้งานเครือข่าย โดยเฉพาะเครือข่าย TCP/IP ซึ่งใช้ในระบบ INTERNET

1.4 โครงสร้างของระบบ UNIX

ในระบบ UNIX นั้นจะประกอบด้วย 4 ส่วนสำคัญด้วยกันคือ

1.4.1 Hardware

คือตัวเครื่องคอมพิวเตอร์นั่นเอง

1.4.2 UNIX Kernel

Kernel เป็นส่วนที่สำคัญที่สุดของ UNIX มันเป็นส่วนหลักที่เก็บตัวโปรแกรมของ UNIX และจะกำหนดความสามารถทั้งหมดของ UNIX นั้นๆ เมื่อเราเริ่มเปิดเครื่องสิ่งแรกที่มีหน้าที่คือ อ่าน Kernel เข้าสู่หน่วยความจำ แล้ว UNIX จึงจะเริ่มทำงานได้

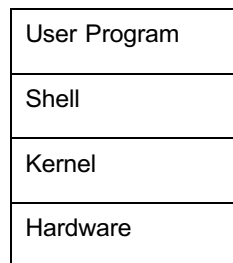
1.4.3 Shell

Shell เป็นตัวกลางที่ติดต่อระหว่างผู้ใช้กับ Kernel และเป็นตัวกำหนดลักษณะหน้าตาของ UNIX ที่จะปรากฏแก่ผู้ใช้ Shell ที่ใช้กันทั่วไปใน UNIX ได้แก่ Bourne Shell, Korn Shell และ C Shell

1.4.4 User Program

คือโปรแกรมที่ผู้ใช้เรียกใช้งานนั่นเอง

ส่วนประกอบทั้ง 4 ข้างต้นจะเรียงกันเป็น Layer ดังแสดงในรูป



โครงสร้างของ UNIX

จากรูป การทำงานของแต่ละชั้นจะเป็นอิสระซึ่งกันและกัน และมันจะติดต่อกับชั้นที่อยู่ติดกับมันเท่านั้น

1.5 โครงสร้าง Directory ใน UNIX

/	Root Directory เป็นจุดเริ่มต้นของโครงสร้างทั้งหมด
/bin	จะเก็บคำสั่งใช้งานทั่วไป
/lib	เก็บ run-time library
/etc	จะเก็บ config และ ตารางที่ใช้กำหนดค่าต่างๆ ของระบบ
/dev	จะเก็บ Device driver
/tmp	เก็บข้อมูลชั่วคราว และจะถูกลบทุกครั้งที่เปิดเครื่อง
/var	จะเก็บบันทึกต่างๆ ของระบบ (system log)ไว้เพื่อใช้ตรวจสอบ
/home	มักจะเก็บ directory เริ่มต้นของผู้ใช้ต่างๆ ในระบบ
/usr	จะเก็บโปรแกรมและคำสั่งต่างๆ ที่มักจะต้องใช้ร่วมกันระหว่างผู้ใช้หลายๆ คน
/usr/local	มักจะใช้เก็บโปรแกรมที่นำมาใส่เพิ่มเติมให้กับระบบ

บทที่ 2

คำสั่ง UNIX พื้นฐาน 1

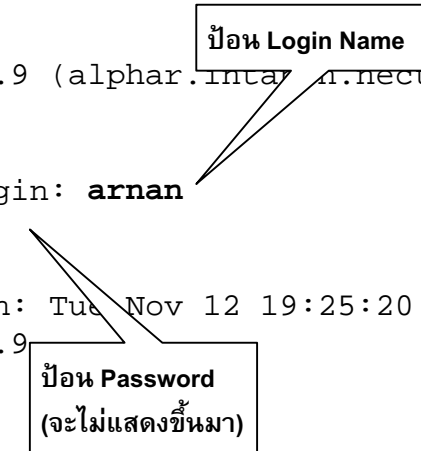
2.1 ms Login และ Logout

การ Login เป็นขั้นตอนแรกสุดที่เราจะต้องทำก่อนที่จะเข้าใช้งานระบบ UNIX ได้ เป็นระบบรักษาความปลอดภัยแบบหนึ่งของ UNIX การที่เราจะเข้าไปใช้งานได้เราจะต้องมี User-name และ Password อยู่บนเครื่องๆ นั้นเสมอ ตัวอย่างของการ Login แสดงไว้ในรูปต่อไปนี้

```
Linux 2.0.9 (alphanec.tec.or.th) (tty0)

alphanec login: arnan
Password:

Last login: Tue Nov 12 19:25:20 from atec.intanon.nec
Linux 2.0.9
alphanec:~$
```



รูปตัวอย่างการ Login

การ Logout นั้นจะทำเมื่อเราใช้งานระบบเสร็จเรียบร้อยแล้ว และไม่ต้องการทำงานอีกต่อไป วิธีการ Logout นั้นให้เราพิมพ์คำว่า exit หรือ logout ที่ prompt

ในกรณีที่เรต้องการจะปิดเครื่อง UNIX เราจะต้องทำการ shutdown ก่อนเสมอ เราจะไม่ปิดเครื่องในขณะที่อยู่ที่ Prompt เหมือนกับ DOS การ logout ก็ไม่ใช้การ shutdown ขอให้ศึกษาเพิ่มเติมได้ในบทที่ 4 หัวข้อ 4.2

2.2 กลุ่มคำสั่งที่ใช้จัดการเกี่ยวกับไฟล์ (File management)

2.2.1 ls – การแสดงรายการของไฟล์ใน directory

คำสั่งนี้จะคล้ายกับคำสั่ง dir ใน DOS ใช้ในการแสดงรายการของไฟล์ต่างๆ ที่มีอยู่ใน directory ดังตัวอย่างการใช้งานต่อไปนี้

- `ls` เฉยๆ จะแสดงรายการชื่อไฟล์ออกมา ดังตัวอย่าง

```
alphar:~$ ls
readme.tx_*    rex.bat*      temp/         unixbig.ppt*  ver.dl_*
readme.wr_*    rex.zip*      unix.ppt*     vbrun300.dl_*
```

- `ls -l` จะแสดงรายการชื่อและรายละเอียดของไฟล์ออกมา ดังตัวอย่าง

```
alphar:~$ ls -l
total 3286
-rwxr-xr-x 1 arnan  users      2752 Jan 11 13:01 readme.tx_*
-rwxr-xr-x 1 arnan  users      3297 Jan 11 13:01 readme.wr_*
-rwxr-xr-x 1 arnan  users         35 Jan 11 12:59 rex.bat*
-rwxr-xr-x 1 arnan  users    2957920 Jan 11 12:59 rex.zip*
drwxr-xr-x 2 arnan  users     1024 Jan 11 13:01 temp/
-rwxr-xr-x 1 arnan  users    45270 Jan 11 13:01 unix.ppt*
-rwxr-xr-x 1 arnan  users   44053 Jan 11 13:01 unixbig.ppt*
-rwxr-xr-x 1 arnan  users  276684 Jan 11 13:01 vbrun300.dl_*
-rwxr-xr-x 1 arnan  users     9696 Jan 11 13:01 ver.dl_*
```

File Owner Group File size date&time File name
 permission

- `ls -a`

ในระบบ UNIX นั้น ไฟล์ที่มีชื่อขึ้นต้นด้วยจุด “.” จะถือว่าเป็น Hidden file ซึ่งเราสามารถดู hidden file เหล่านี้ได้โดยการใช้ option `-a` ดังตัวอย่าง

```
alphar:~$ ls -a
./          .less      readme.wr_*  unix.ppt*
../         .lessrc    rex.bat*     unixbig.ppt*
.bash_history .term/     rex.zip*     vbrun300.dl_*
.kermrc     readme.tx_* temp/         ver.dl_*
```

2.2.2 file – ใช้แสดงชนิดของไฟล์ใน directory

คำสั่ง `file` จะทำการแสดงชนิดของไฟล์ที่มีใน directory นั้นออกมาดังเช่นตัวอย่าง

```
alphar:~$ file *
readme.tx_: data
readme.wr_: data
rex.bat:    ascii text
rex.zip:    zip archive file - version 1.0
temp:       directory
unix.ppt:   data
unixbig.ppt: data
vbrun300.dl_: data
ver.dl_:    DOS executable (EXE)
```

2.2.3 cp – ใช้ในกรณี copy ไฟล์

รูปแบบการใช้คำสั่งนี้จะเหมือน copy ใน DOS คือ

cp [Option] ไฟล์ต้นทาง ปลายทาง

Option ที่น่าสนใจได้แก่

- r ทำการ copy sub-directory ด้วย
- v verbose - คือแสดงผลการทำงานด้วย

ตัวอย่างการใช้งาน เช่น

```
alphar:~/temp$ cp -v c* ..
cloneo_u.ic_ -> ../cloneo_u.ic_
clonex_u.ic_ -> ../clonex_u.ic_
```

จุดสำคัญที่ควรสังเกตคือ ใน DOS เราจะใช้ *.* แทนไฟล์ทุกไฟล์ แต่ *.* ใน UNIX จะหมายถึงไฟล์ทุกไฟล์ที่มีเครื่องหมายจุด ใน UNIX จะใช้ * แทนไฟล์ทุกไฟล์

2.2.4 mv – ย้ายไฟล์หรือเปลี่ยนชื่อไฟล์

คำสั่งนี้มีรูปแบบการใช้งานเหมือนกันกับคำสั่ง cp ทุกอย่างแต่จะไม่มี Option -r เท่านั้น

2.2.5 rm – ลบไฟล์

รูปแบบการใช้งานคือ

rm [Option] ชื่อไฟล์

Option ก็มีเหมือนคำสั่ง cp

2.2.5 cat – ใช้ในกรณี view ไฟล์

คำสั่งนี้เหมือนคำสั่ง type ใน DOS คือใช้ดูข้อมูลข้างในไฟล์รูปแบบการใช้งานเป็นดังนี้

cat ชื่อไฟล์

2.3 คำสั่งที่ใช้จัดการ directory

2.3.1 pwd – ใช้แสดงตำแหน่ง directory ปัจจุบัน

ตัวอย่างการใช้งานได้แก่

```
alphan:~$ pwd
/home/arnan
```

2.3.2 cd – ใช้ในการเปลี่ยน directory

รูปแบบการใช้งานคือ

```
cd directory
```

2.3.3 mkdir – ใช้สร้าง directory

รูปแบบการใช้งานคือ

```
mkdir [Option] ชื่อdirectory1 [ชื่อ directory 2] ...
```

Option ที่น่าสนใจคือ -p ซึ่งอนุญาตให้เราสร้าง directory ได้ที่หลายๆ ชั้น

2.3.4 rmdir ใช้ลบ directory

รูปแบบการใช้งานจะเหมือนกัน mkdir โดย directory ที่จะถูกลบนั้นจะต้องไม่มีไฟล์อยู่ ถ้าเราต้องการลบ directory ที่ไม่ว่างเราสามารถใช้อำนาจ rm -r ได้ ซึ่งจะให้ผลเหมือนคำสั่ง deltree ของ dos

2.4 คำสั่งที่ใช้จัดการเกี่ยวกับ Terminal และ User

2.4.1 stty ใช้กำหนดคุณสมบัติของ Terminal

คำสั่งนี้เป็นคำสั่งเก่าแก่ที่มีมากับ UNIX สมัยแรกๆ ซึ่งสมัยนั้นจะใช้สิ่งที่เราเรียกว่า terminal ในการพวงเข้ากับเครื่อง UNIX เพื่อขอใช้งาน terminal พวกนี้มักมีอยู่หลากหลายรูปแบบ หลายมาตรฐาน ดังนั้นคุณสมบัติต่างๆ ของมันก็จะต่างกันด้วย เช่น จำนวนบรรทัดที่แสดงบนหน้าจอ ความกว้างของจอ ปุ่มพิเศษต่างๆ ที่ใช้ ในปัจจุบันเรามักไม่ได้ใช้คำสั่งนี้แล้ว ยกเว้นใช้ในการกำหนดค่าปุ่มพิเศษบนแป้นพิมพ์เท่านั้น เวลาเราเข้าใช้ UNIX จากที่ต่างๆ กันมักจะมีปุ่มๆ หนึ่งบนแป้นพิมพ์ที่ทำงานไม่ถูกต้องคือปุ่ม back-space เวลากดมันแล้วมันจะไม่ยอมลบ เราสามารถแก้ได้โดยใช้อำนาจต่อไปนี้

```
stty erase <Hit the backspace key>
```

2.4.2 who ใช้ในการเรียกดูว่ามีใคร login อยู่ในระบบบ้าง

ตัวอย่างการใช้งานเช่น

```
alphar:~$ who
root      tty1      Jan 11 13:55
arnan     tty0      Jan 11 12:52 (atec.intanon.nec)
guest     tty3      Jan 11 13:56
```

ในตัวอย่างมี user 3 คนกำลังใช้งานอยู่คือ root, arnan และ guest

2.4.3 passwd ใช้ในการเปลี่ยน password

ในบางครั้งเราอาจต้องการเปลี่ยนรหัสผ่านของเราใหม่ เราสามารถทำได้โดยการใช้คำสั่ง passwd โดยมันจะให้เราป้อนรหัสผ่านเดิมก่อน แล้วจะถามรหัสใหม่ โดยมันจะถามเราสองครั้ง ให้พิมพ์เข้าไปให้เหมือนกันทั้งสองครั้ง ถ้าพิมพ์ไม่เหมือนกันมันจะไม่ยอมเปลี่ยนให้ ตัวอย่างต่อไปนี้แสดงให้เห็นถึงการใช้คำสั่ง passwd

```
alphar:~$ passwd
Changing password for arnan
Enter old password:
Enter new password:
Re-type new password:
Password changed.
```

The diagram consists of three rectangular boxes with callout lines pointing to specific prompts in the terminal output above:

- A box labeled "ป้อนรหัสเก่า" (Enter old password) points to the "Enter old password:" prompt.
- A box labeled "ป้อนรหัสใหม่" (Enter new password) points to the "Enter new password:" prompt.
- A box labeled "ป้อนรหัสใหม่อีกครั้ง" (Re-type new password) points to the "Re-type new password:" prompt.

2.5 คำสั่งติดต่อสื่อสารระหว่าง user

2.5.1 write ใช้ส่งข้อความไปหา user อื่นที่ใช้งานเครื่องอยู่

รูปแบบการใช้งานคือ

```
write user-name
Message ...
^D
```

เมื่อพิมพ์บรรทัดแรกเสร็จก็จะสามารถพิมพ์ข้อความให้กับ user ปลายทางที่บรรทัดก็ได้ เมื่อเราพอใจแล้วก็จบโดยการกด Control-D

2.5.2 msg ใช้ในการกำหนดว่าต้องการเปิดบริการติดต่อจากผู้อื่นหรือไม่

ในบางครั้งเราอาจไม่ต้องการให้ใครมารบกวนเราในขณะที่กำลังทำงาน เช่น ไม่ต้องการให้ใครส่ง message มาหาเป็นต้น เราสามารถใช้คำสั่ง msg กำหนดได้โดยมีการใช้งาน 2 รูปแบบคือ

```
msg n
msg y
```

รูปแบบแรกนั้นจะทำให้ผู้ใช้คนอื่นไม่สามารถส่งข้อความขึ้นมาบนจอของเราได้ ส่วนรูปแบบที่สองนั้นเป็นการยกเลิกคำสั่งแรก นั่นคืออนุญาตให้ผู้อื่นติดต่อมาได้

2.5.3 talk ใช้ในการติดต่อส่งข้อความคุยกัน

คำสั่งนี้จะคล้ายกับคำสั่ง write แต่การส่งข้อความหากันจะเป็นแบบ online-bidirectional คือสามารถส่งข้อความโต้ตอบกันได้ รูปแบบการใช้งานเป็นดังนี้

```
talk user-name
```

เมื่อผู้ใช้คนแรกใช้คำสั่ง talk ไปหาใครผู้ใช้ปลายทางจะต้องใช้คำสั่ง talk นี้กลับด้วยการติดต่อจึงจะถือว่าสมบูรณ์ และการส่งข้อความจึงจะเริ่มขึ้น

2.5.4 mail ใช้ในการส่งจดหมายอิเล็กทรอนิกส์ (e-mail)

E-mail นั้นเป็นบริการที่ได้รับความนิยมเป็นอย่างมากใน INTERNET แต่ e-mail ที่เราจะกล่าวถึงในตอนนี้เป็นเพียงการส่งภายในเครื่องๆ เดียวเท่านั้น บริการนี้นับได้ว่าเป็นบริการที่เก่าแก่ที่สุดตัวหนึ่งในสมัยแรกเราจะใช้คำสั่ง mail ในการอ่านและส่งจดหมายโดยมีรูปแบบดังนี้

- ถ้าต้องการส่งจดหมาย mail username
- ถ้าต้องการอ่านจดหมาย mail

การใช้คำสั่ง mail ในการส่งจดหมายนั้นจะมีลักษณะดังเช่นในตัวอย่าง

```
alphan:~# mail arnan
Subject: testmail
```

```
Hello,
        This is a test mail for you.
```

```
bye,
root
.
```

ตัวอย่างข้างต้นเป็นการส่งจดหมายไปยังผู้ใช้ที่มีชื่อ login ว่า arnan ส่วน Subject นั้นคือหัวข้อของเนื้อหาในจดหมายฉบับนั้น ผู้รับจะสามารถเห็นหัวข้อเหล่านี้ได้ก่อนที่จะเปิดจดหมายออกมาอ่าน ส่วนที่เหลือคือตัวจดหมาย เราสามารถพิมพ์ไปที่บรรทัดก็ได้ โดยเมื่อเราพิมพ์เสร็จแล้วให้พิมพ์ตัว จุด ที่หลักแรกของแถวใหม่ เพื่อบอกจุดจบของจดหมาย

เมื่อเราใช้คำสั่ง mail เพื่ออ่านจดหมายเราจะเข้ามาสู่ prompt ของโปรแกรม mail (รูปตัว &) เราสามารถพิมพ์คำสั่งต่างๆ ลงไปได้ โดยคำสั่งที่น่าจะทราบได้แก่

h	แสดงรายการจดหมายที่มีอยู่
หมายเลข	ใช้เลือกจดหมายที่ต้องการอ่าน โดยเราจะทราบหมายเลขโดยการใช้คำสั่ง h
d หมายเลข	ใช้ลบจดหมาย
q	ใช้จบการทำงานและกลับไป UNIX prompt

ในปัจจุบันเรามักไม่ใช้คำสั่ง mail แล้วเพราะมีรูปแบบการใช้งานที่ไม่สะดวก ประกอบกับมีโปรแกรมใหม่ๆ เข้ามาแทนที่ โปรแกรมที่ได้รับความนิยมมากที่สุดตัวหนึ่งคือ pine ซึ่งเราจะศึกษาในบทที่แนะนำบริการเบื้องต้นของ INTERNET

2.6 คำสั่งเบื้องต้นอื่นๆ

2.6.1 clear – ใช้ในการลบข้อความบนหน้าจอทิ้ง

2.6.2 cal – ใช้ในการแสดงปฏิทินออกมา เช่น

```
alphar:~# cal
      January 1997
Su Mo Tu We Th Fr Sa
                1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31
```

เราสามารถกำหนด เดือนกับปีที่ต้องการแสดงได้ เช่น

```
alphar:~# cal 4 97
      April 97
Su Mo Tu We Th Fr Sa
                1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30
```

หรืออาจจะพิมพ์ปฏิทินทั้งปีเลยก็ได้ เช่น

alphar:~# cal 97

97

January							February							March						
Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7				1	2	3	4				1	2	3	4
8	9	10	11	12	13	14	5	6	7	8	9	10	11	5	6	7	8	9	10	11
15	16	17	18	19	20	21	12	13	14	15	16	17	18	12	13	14	15	16	17	18
22	23	24	25	26	27	28	19	20	21	22	23	24	25	19	20	21	22	23	24	25
29	30	31					26	27	28					26	27	28	29	30	31	

April							May							June						
Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa
						1		1	2	3	4	5	6					1	2	3
2	3	4	5	6	7	8	7	8	9	10	11	12	13	4	5	6	7	8	9	10
9	10	11	12	13	14	15	14	15	16	17	18	19	20	11	12	13	14	15	16	17
16	17	18	19	20	21	22	21	22	23	24	25	26	27	18	19	20	21	22	23	24
23	24	25	26	27	28	29	28	29	30	31				25	26	27	28	29	30	
30																				

July							August							September						
Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa
						1			1	2	3	4	5						1	2
2	3	4	5	6	7	8	6	7	8	9	10	11	12	3	4	5	6	7	8	9
9	10	11	12	13	14	15	13	14	15	16	17	18	19	10	11	12	13	14	15	16
16	17	18	19	20	21	22	20	21	22	23	24	25	26	17	18	19	20	21	22	23
23	24	25	26	27	28	29	27	28	29	30	31			24	25	26	27	28	29	30
30	31																			

October							November							December						
Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7				1	2	3	4						1	2
8	9	10	11	12	13	14	5	6	7	8	9	10	11	3	4	5	6	7	8	9
15	16	17	18	19	20	21	12	13	14	15	16	17	18	10	11	12	13	14	15	16
22	23	24	25	26	27	28	19	20	21	22	23	24	25	17	18	19	20	21	22	23
29	30	31					26	27	28	29	30			24	25	26	27	28	29	30
														31						

2.6.3 alias ใช้กำหนดชื่อเสมือนให้กับคำสั่ง

คำสั่งนี้มีประโยชน์ค่อนข้างมากในกรณีที่เรามักจะต้องใช้คำสั่งใดคำสั่งหนึ่งบ่อยๆ หรือคำสั่งบางคำสั่ง ยาวมาก และจำได้ยาก เราสามารถทำการตั้งชื่อเสมือนให้กับคำสั่งเหล่านั้นได้ รูปแบบการใช้งาน alias เป็นดังนี้

```
alias ชื่อเสมือน='คำสั่ง'
```

ยกตัวอย่าง เช่น

```
alias dir='ls -l'
```

ตัวอย่างข้างต้นทำให้เราสามารถพิมพ์คำสั่ง dir ได้โดยจะมีความหมายเหมือนกับการพิมพ์ ls -l

บทที่ 3

คำสั่ง UNIX พื้นฐาน 2

3.1 การค้นหาไฟล์ และระบบรักษาความปลอดภัยของไฟล์

3.1.1 find ค้นหาไฟล์

คำสั่งนี้เป็นคำสั่งที่ใช้ช่วยเราค้นหาไฟล์ที่ต้องการ โดยมีรูปแบบการใช้งานพื้นฐานดังนี้

```
find <starting directory> [-name “ชื่อไฟล์”] [-print]
```

ยกตัวอย่างเช่น

```
find / -name “*arnan*” -print
```

คำสั่งข้างต้นเป็นคำสั่งที่ใช้ในการค้นหาไฟล์ทุกไฟล์ที่มีคำว่า arnan อยู่ และเริ่มต้นค้นหาจาก root directory เมื่อพบแล้วก็ให้พิมพ์ออกมาบนจอด้วย

3.1.2 which แสดงตำแหน่งของไฟล์ที่เรียกใช้

คำสั่งนี้ใช้แสดงตำแหน่งของไฟล์ที่เราทำการเรียกใช้ มีรูปแบบการใช้งานดังนี้

```
which file-name
```

ยกตัวอย่างเช่น

```
which ls
```

คำสั่งข้างต้นจะทำการแสดงตำแหน่งของไฟล์ ls ขึ้นมาว่าอยู่ใน directory ไດ

3.1.3 chmod ใช้กำหนดสิทธิการใช้งานไฟล์

คำสั่งนี้เราสามารถใช้ในการกำหนดสิทธิของผู้ใช้กลุ่มต่างๆ ในระบบที่มีต่อไฟล์ๆ หนึ่งได้ โดยปกติกลุ่มของผู้ใช้ในมุมมองของผู้ใช้คนหนึ่งๆ จะมีอยู่ 3 กลุ่มด้วยกันคือ

1. เจ้าของ (Owner)- นั่นคือผู้ที่สร้างไฟล์นั้นขึ้นมา
2. กลุ่ม (Group) - ใน UNIX เราสามารถกำหนดกลุ่มผู้ใช้ขึ้นมาได้ ผู้ใช้ที่อยู่ในกลุ่มเดียวกันมักจะมีสิทธิในการใช้งานไฟล์ของเพื่อนร่วมกลุ่มแตกต่างจากผู้ใช้ที่ไม่ได้อยู่ในกลุ่มนั้นๆ
3. ผู้อื่น (Other) - คือผู้ใช้ที่ไม่ได้อยู่ในข้อ 1 และ 2

สิทธิ์ต่างๆ ที่ผู้ใช้สามารถกำหนดให้กับไฟล์ได้มีอยู่ 3 อย่างด้วยกันคือ

1. Read - คือสิทธิ์ในการอ่านข้อมูลในไฟล์
2. Write - คือสิทธิ์ในการแก้ไขข้อมูลในไฟล์ (ไม่ใช่สิทธิ์ในการลบไฟล์)
3. Execute - คือสิทธิ์ในการเรียกใช้งานไฟล์

ผู้ใช้สามารถกำหนดสิทธิ์ทั้งสามอย่างนี้ให้กับผู้ใช้แต่ละกลุ่มได้ว่าต้องการให้ทำอะไรได้และทำอะไรไม่ได้ เราสามารถเรียกดูสิทธิ์ของไฟล์ได้จากคำสั่ง ls -l ดังในตัวอย่างนี้

```
-rwxr-xr-x 1 arnan users 45270 Jan 11 13:01 unix.ppt*
```

ชุดของตัวอักษรทางซ้ายสุดคือส่วนที่แสดงไฟล์ permission หรือสิทธิ์การใช้งานไฟล์ ตัวอักษรเหล่านี้แบบออกเป็น 3 กลุ่มตามชนิดของผู้ใช้ข้างต้น และแต่ละกลุ่มก็มีอักษร 3 ตัวใช้ในการแสดงสิทธิ์ที่ผู้ใช้ในกลุ่มนั้นมี

```

rwx rwx rwx
  }  }  }
owner group other
    
```

จากรูปข้างต้นตัวอักษรแต่ละตัวมีความหมายดังนี้ r=read, w=write และ x=execute

การเปลี่ยนสิทธิ์ของไฟล์นั้นเราสามารถทำได้โดยใช้คำสั่ง chmod ซึ่งจะมีรูปแบบการใช้งานดังนี้

chmod [u/g/o]<+/-><r/w/x> file-name

ยกตัวอย่างเช่น

```

chmod g+w test.txt
chmod o-r test.txt
chmod +x test.txt
    
```

ตัวอย่างแรกจะทำการเพิ่มสิทธิ์ในการแก้ไขข้อมูลในไฟล์ test.txt ให้กับ group ส่วนคำสั่งที่สองจะทำการลบสิทธิ์ในการอ่านของผู้ใช้อื่นๆ (other) และคำสั่งที่สามจะเพิ่มสิทธิ์ในการใช้งานไฟล์ test.txt ให้กับผู้ใช้ทั้ง 3 กลุ่ม

สิ่งที่ได้กล่าวไปข้างต้นว่าสิทธิ์ write ของไฟล์นั้นไม่ใช่สิทธิ์ที่กำหนดว่าผู้ใช้จะสามารถลบไฟล์นั้นทิ้งได้หรือไม่ การจะดูว่าผู้ใช้ใดๆ สามารถลบไฟล์ใดทิ้งได้หรือไม่เราจะต้องดูที่สิทธิ์ของผู้ใช้ใน directory นั้นๆ เช่นถ้าไฟล์ test.txt อยู่ใน directory ที่ชื่อ guest เราก็จะต้องดูสิทธิ์ที่กำหนดไว้ให้กับ directory นั้น เช่น ถ้าสิทธิ์ของ directory guest เป็น rwxr-x-- แสดงว่าเจ้าของไฟล์จะสามารถลบไฟล์ทิ้งได้ ส่วน group จะไม่สามารถลบได้ เช่นเดียวกับผู้ใช้อื่นๆ (other) ซึ่งไม่สามารถแม้แต่จะอ่านข้อมูลใน directory นี้ (ไม่มีสิทธิ์ในการอ่าน)

3.1.4 chown เปลี่ยนเจ้าของไฟล์

ไฟล์แต่ละไฟล์ในระบบ UNIX จะต้องมีเจ้าของและกลุ่ม โดยเราสามารถดูชื่อเจ้าของและกลุ่มของไฟล์ใดๆ ได้โดยใช้คำสั่ง ls -l เช่น

```
-rwxr-xr-x 1 arnan users 45270 Jan 11 13:01 unix.ppt*
```

ในตัวอย่างข้างต้นนี้ ไฟล์ unix.ppt มีชื่อเจ้าของเป็น arnan และอยู่ในกลุ่ม users เราสามารถเปลี่ยนชื่อเจ้าของไฟล์ได้โดยใช้คำสั่ง chown โดยมีรูปแบบดังนี้

chown new-owner file-name

เช่น

```
chown guest unix.ppt
```

ตัวอย่างข้างต้นนี้เป็นการเปลี่ยนเจ้าของไฟล์ unix.ppt ไปเป็นผู้ใช้ที่มี login name ว่า guest เฉพาะเจ้าของไฟล์เท่านั้นที่จะทำแบบนี้ได้

ข้อควรระวังในการใช้คำสั่งนี้คือเมื่อเราเปลี่ยนชื่อเจ้าของไปแล้ว เราจะเสียสิทธิ์ในการเป็นเจ้าของไฟล์ๆ นั้นไปทันทีและเราจะไม่สามารถเปลี่ยนชื่อเจ้าของให้กลับมาเป็นของเราได้ จะต้องให้เจ้าของคนใหม่เปลี่ยนให้เท่านั้น ทำให้บางครั้งเราอาจไม่สามารถใช้งานไฟล์ๆ นั้นได้ เช่น ไฟล์ unix.ppt ข้างต้นเมื่อเราเปลี่ยนเจ้าของไปแล้วจะทำให้เราไม่สามารถแก้ไขข้อมูลในไฟล์ๆ นี้ได้

3.1.5 chgrp เปลี่ยน group ของไฟล์

คำสั่งนี้ใช้ในการเปลี่ยน group ที่ไฟล์หนึ่งๆ เป็นสมาชิกอยู่ รูปแบบการใช้งานก็เหมือนกับคำสั่ง chown เช่น

```
chgrp student unix.ppt
```

คำสั่งข้างต้นเป็นการเปลี่ยนกลุ่มที่ไฟล์ unix.ppt เป็นสมาชิกไปเป็นกลุ่มที่ชื่อว่า student กลุ่มที่เปลี่ยนไปนั้นจะต้องมีอยู่จริง โดยเราสามารถจะดูได้ว่าเครื่องๆ นี้มีกลุ่มอะไรอยู่บ้างโดยดูจากไฟล์ /etc/group

3.2 การจัดการ Process

ในระบบ UNIX เมื่อเราสั่งงานอะไรไป จะทำให้เกิดเป็น process ขึ้นมา แต่ละ process จะมีหมายเลขประจำของมันอยู่ และเราสามารถควบคุม process เหล่านี้ได้ โดยใช้คำสั่งเหล่านี้

3.2.1 & ใช้สั่งให้คำสั่งทำงานเป็น background

ถ้าหากว่าเรามีการใช้งานคำสั่งใดที่ใช้เวลาในการทำงานนาน เราสามารถสั่งให้มันทำงานเป็น background ได้โดยเราจะกลับไปที่ UNIX prompt และสามารถทำงานอื่นต่อไปได้ เช่น คำสั่ง find ถ้าเราสั่งให้มันค้นหาเริ่มตั้งแต่ root directory มันมักจะใช้เวลาคันทนาน เราสามารถใช้คำสั่งในรูปแบบต่อไปนี้ในการสั่งให้มันทำงานเป็น background ได้

```
find / -name "*arnan*" -print &
```

3.2.2 ps ใช้ในการแสดงรายการ process ที่กำลังทำงานอยู่

เราสามารถเรียกดู process ของเรากำลังทำงานอยู่ได้ โดยใช้คำสั่ง ps ซึ่งจะแสดงผลดังตัวอย่าง

```
alphar:~$ ps
  PID TTY STAT  TIME COMMAND
   150 pQb  1 N    0:00 -bash
   155 pQb  1 N    0:00 ps
```

ตัวอย่างข้างต้นแสดงให้เห็นว่ามีงาน 2 งานกำลังทำงานอยู่ คือ -bash ซึ่งหมายถึง shell ซึ่งจะกล่าวถึงในหัวข้อถัดไป ส่วน ps ก็คือคำสั่ง ps ที่เรากำลังใช้งานอยู่นั่นเอง สิ่งสำคัญที่เรามักจะนำมาใช้จากคำสั่งนี้คือ หมายเลข process (PID) เช่นในตัวอย่าง process ID = 150 กับ 155

3.2.3 kill ยกเลิกการทำงานของ process

เราสามารถยกเลิก process ของเรากำลังทำงานอยู่ได้ โดย ขั้นแรกเราจะต้องทราบ PID ของ process นั้นก่อน โดยใช้คำสั่ง ps เมื่อได้เลข PID มาแล้ว ต่อไปก็ใช้คำสั่ง kill ในการหยุดการทำงาน เช่น

```
kill 201
```

จากตัวอย่างข้างต้น คำสั่งหรือ process ที่มี PID เป็น 201 จะหยุดทำงานทันที

3.3 Piping

Piping เป็นวิธีการนำเอาผลลัพธ์ที่ได้จากคำสั่งหนึ่งไปประมวลผลก่อนที่จะแสดงออกมา การ pipe นั้นเราจะใช้เครื่องหมาย Vertical bar “|” ตามท้ายคำสั่งหนึ่งๆ แล้วตามด้วยชื่อโปรแกรมที่ต้องการ ซึ่งโปรแกรมที่น่าสนใจได้แก่

3.3.1 more ใช้ในการแสดงผลทีละหน้า

เราจะใช้มันเมื่อมีคำสั่งหนึ่งทีแสดงผลออกมามากเกินไปจนเกินหนึ่งหน้าจอ ทำให้เรามองไม่เห็นผลลัพธ์ในตอนต้น ๆ ยกตัวอย่างการใช้งาน more เช่น

```
ls -l | more
cat /etc/passwd | more
```

ตัวอย่างแรกจะให้ผลเหมือนกับการใช้ dir /p ใน DOS นั่นคือแสดงรายการไฟล์ออกมาทีละหน้า ส่วนตัวอย่างที่สองเป็นการแสดงข้อความข้างในไฟล์ที่ชื่อ /etc/passwd ออกมาทีละหน้า

3.3.2 grep ใช้ในการค้นหาคำที่ตรงการ

เรามักใช้ grep เมื่อเราต้องการผลเพียงบางบรรทัดจากผลทั้งหมดที่คำสั่งใดๆ สร้างออกมา เช่น

```
ls -l | grep arnan
cat /etc/passwd | grep guest
```

ตัวอย่างแรกเป็นการเลือกแสดงผลเฉพาะบรรทัดที่มีคำว่า aman อยู่เท่านั้น ซึ่งบรรทัดที่แสดงผลออกมา อาจมีคำว่า aman อยู่ในชื่อไฟล์ หรืออาจเป็น ชื่อเจ้าของ หรือ กลุ่ม ก็ได้ ส่วนตัวอย่างที่สองมันจะแสดงผลเฉพาะบรรทัดที่มีคำว่า guest ออกมา

3.3.3 wc ใช้ในการนับจำนวนบรรทัด, คำ และตัวอักษร

เราจะใช้ wc เมื่อเราต้องการนับจำนวน บรรทัด, คำ หรือตัวอักษรของผลลัพธ์ เช่น

```
alphar:~# ls -l |wc
      7      60     422
```

ตัวอย่างข้างต้น ผลลัพธ์ที่ได้ออกมาคือจำนวนบรรทัดที่ ls -l แสดงออกมา (7) ตามด้วย 60 ซึ่งเป็นจำนวนคำ และ 422 คือจำนวนตัวอักษร

3.3.4 sort ใช้ในการเรียงผลลัพธ์

ถ้าเราต้องการให้ผลลัพธ์ที่ได้จากคำสั่งหนึ่งมีการเรียงลำดับก่อนหลัง เราจะใช้ sort เช่น ลองดูผลลัพธ์ที่จะได้จากการใช้คำสั่ง ls ทั่วๆ ไป

```
alphar:~# ls -l
total 4
lrwxrwxrwx  1 root    root          8 Oct 12 05:51 INSTALL
lrwxrwxrwx  1 root    root         14 Oct 12 05:51 linux
drwx-----  2 root    root       1024 Dec  3 15:30 mail/
-rw-r--r--  1 root    root        135 Dec  3 15:34 ppps.dip
-rwxr-xr-x  1 root    root         53 Dec  3 15:47 pppconnect*
-rw-----  1 root    root         51 Jan 11 20:24 dead.letter
```

ถ้าหากว่าเราต้องการให้มันเรียงตามชื่อไฟล์เราจะใช้

```
alphar:~# ls -l |sort +8
total 4
lrwxrwxrwx  1 root    root          8 Oct 12 05:51 INSTALL
-rw-----  1 root    root         51 Jan 11 20:24 dead.letter
lrwxrwxrwx  1 root    root         14 Oct 12 05:51 linux
drwx-----  2 root    root       1024 Dec  3 15:30 mail/
-rwxr-xr-x  1 root    root         53 Dec  3 15:47 pppconnect*
-rw-r--r--  1 root    root        135 Dec  3 15:34 ppps.dip
```

ตัวอย่างข้างต้นเราใช้ sort +n โดย n คือหมายเลข column ที่ต้องการจะเรียง โดยเริ่มนับจากซ้ายมาขวา มันจะเริ่มนับจาก 0 และใช้ช่องว่างเป็นตัวแบ่ง column ในตัวอย่างเป็นการเรียงจากน้อยไปมาก ถ้าเราต้องการเรียงจากมากไปน้อยเราสามารถทำได้โดยเพิ่ม option -r เข้าไป

3.4 Shell กับตัวแปร Environment

อย่างที่ได้อธิบายไปในบทที่หนึ่ง Shell เป็นส่วนที่เชื่อมต่อ Kernel เข้ากับ โปรแกรมของผู้ใช้ โดย Shell นั้นจะมีให้เลือกใช้หลายตัวแล้วแต่ความพอใจของผู้ใช้ แต่ shell ที่เป็นที่ยอมรับมีอยู่ 3 ตัวด้วยกันคือ Bourne Shell (sh) , Korn Shell (ksh) และ C Shell (csh) สำหรับ Bourne Shell นั้นเป็น Shell ดั้งเดิมที่ใช้มาตั้งแต่สมัยแรกๆ และยังคงเป็นที่นิยมใช้กันอยู่ ส่วนสอง Shell หลังนั้น C Shell จะเป็นที่นิยมมากกว่า

โดยปกติแล้ว ผู้ดูแลระบบจะเป็นคนกำหนด Shell เริ่มต้นให้กับผู้ใช้ ถ้าเราต้องการเปลี่ยน shell เริ่มต้นเราจะต้องบอกให้ผู้ดูแลระบบเปลี่ยนให้จริงๆ แล้วเราสามารถเรียกใช้ shell ใหม่กับ shell เดิมได้เลย แต่มันจะเป็นการเปลืองหน่วยความจำ เพราะเป็นการเรียกใช้ shell ซ้อนเป็นชั้นๆ ถ้าต้องการเปลี่ยนใช้ shell ใหม่แทน shell เดิมเลย จะต้องใช้คำสั่ง exec ช่วย ดังเช่นตัวอย่าง

```
csh
exec sh
```

คำสั่งแรกเป็นการเรียกใช้ C Shell ซ้อนทับ shell เดิม ส่วนคำสั่งที่สองเป็นการเรียกใช้ Bourne Shell แทนที่ shell เดิม

3.4.1 การตั้งค่าตัวแปร Environment

ตัวแปร Environment นั้นเป็นตัวแปรที่มักถูกใช้โดยโปรแกรมต่างๆ เพื่อให้ทราบถึงสถานะและรายละเอียดของสภาวะแวดล้อมขณะนั้นก่อนทำงาน เช่น ชนิดของจอภาพ (Terminal type), ชื่อผู้ใช้, ตำแหน่ง directory ปัจจุบัน เป็นต้น เราสามารถกำหนดค่าตัวแปรเหล่านี้ได้โดยการใช้คำสั่งในรูปแบบต่อไปนี้

สำหรับ Bourne และ Korn Shell : `export ตัวแปร = ค่า`
 สำหรับ C Shell : `set ตัวแปร=ค่า`

ยกตัวอย่าง เช่น

```
export TERM=vt100
set term=vt100
```

ตัวอย่างข้างต้นเป็นการตั้งค่าชนิดของจอภาพ (terminal) ให้เป็นชนิดที่มีชื่อว่า vt100 โดยคำสั่งแรกใช้กับ Bourne,Korn ส่วนคำสั่งที่สองใช้กับ C shell

3.4.2 Start-up Files

ถ้าเราต้องการให้มีการตั้งค่าตัวแปร Environment หรือทำคำสั่งอะไรบางอย่างทุกครั้งที่เรา Login เข้าสู่ระบบ เราสามารถทำได้โดยเขียนคำสั่งเหล่านั้นไว้ใน Start-up File ซึ่งคล้ายๆ กับไฟล์ Config.sys และ Autoexec.bat ใน DOS โดยไฟล์ที่เราจะใช้คือ

สำหรับ Bourne Shell: `.profile`
 สำหรับ C Shell: `.login` กับ `.cshrc`
 สำหรับ Korn Shell: `.profile` กับไฟล์ที่ตั้งค่าไว้ในตัวแปร ENV

ใน C Shell นั้นมี Start-up file อยู่ 2 ไฟล์ โดยไฟล์ .login จะทำงานทุกครั้งที่เรา login เข้ามา ส่วน .cshrc นั้นจะทำงานทุกครั้งที่มีการเรียก Shell ใหม่ขึ้นมา นั่นคือเมื่อเรา login ครั้งแรก ไฟล์ .login และ .cshrc จะทำงานทั้งคู่ แต่ถ้าเราเรียก C Shell ขึ้นมาใหม่ เฉพาะไฟล์ .cshrc จะทำงาน ใน Korn Shell เราสามารถสร้างไฟล์ที่ทำหน้าที่คล้ายไฟล์ .cshrc ได้โดยกำหนดชื่อไว้ในตัวแปรชื่อ ENV เช่น

```
export ENV=$HOME/.kshrc
```

คำสั่งข้างต้นเป็นการกำหนดให้ไฟล์ที่ชื่อ .kshrc ที่อยู่ใน directory บ้านของผู้ใช้ เป็นไฟล์ที่จะทำงานทุกครั้งที่มีการเรียก Korn Shell

บทที่ 4

UNIX System Administration เบื้องต้น

4.1 Super User

Super User หรือ ผู้ดูแลระบบนั้นคือ user ที่มีอำนาจสูงสุด สามารถทำอะไรก็ได้ตามใจชอบ ระบบรักษาความปลอดภัยต่างๆ ในระบบนั้นไม่สามารถกัน Super User ได้

ในการทำหน้าที่เป็น Super User นั้นเราจะต้องทำการ Login โดยใช้ user name ว่า root หรืออาจ login เป็นผู้ใช้ธรรมดาก็ได้แล้วใช้คำสั่ง su เพื่อเปลี่ยนตนเองเป็น Super User อีกทีหนึ่ง

4.2 การ Shutdown และ Reboot ระบบ

ในระบบ UNIX นั้นเราไม่สามารถที่จะปิดเครื่องหรือ reboot เครื่องในขณะที่อยู่ที่ Shell Prompt หรือที่ Login Prompt ได้ ผู้ดูแลระบบเท่านั้นที่จะทำการ ปิดเครื่องได้ โดยจะต้องใช้คำสั่ง init ซึ่งมีรูปแบบดังนี้

```
init <run level>
```

จากรูปแบบข้างต้น run level จะเป็นตัวเลขที่ใช้ในการระบุว่าเราต้องการทำอะไร ถ้าเราใช้เลข 0 จะเป็นการ shutdown ระบบ ส่วนเลข 6 จะเป็นการ reboot ตัวเลขอื่นก็จะมีควมหมายอย่างอื่น แต่จะไม่กล่าวถึงในที่นี้

ปกติแล้ว UNIX จะมีคำสั่งอื่นที่ทำหน้าที่เหมือน init แต่รูปแบบการใช้งานมักจะแตกต่างกันระหว่าง UNIX ชนิดต่างๆ เช่นใน Linux เราสามารถใช้คำสั่ง Shutdown และ Reboot แทนคำสั่ง init 0 และ init 6 ได้ ดังเช่นตัวอย่าง

```
shutdown -h now  
reboot
```

คำสั่งแรกเป็นการ shutdown ระบบ คำว่า now หมายถึงให้ shutdown ทันทีไม่ต้องรอ ส่วนคำสั่ง reboot นั้นมีความหมายตรงตัวคือจะทำให้เครื่องทำการ reboot ใหม่

4.3 การจัดการ user และ group

ในบทที่ 3 ได้กล่าวถึงคำสั่งที่ใช้ในการจัดการสิทธิและเจ้าของๆ ไฟล์และ directory ไปแล้ว ในส่วนนี้เราจะศึกษาถึงการ เพิ่มและลบ user กับ group ของระบบ

4.3.1 useradd ใช้ในการเพิ่มผู้ใช้เข้าไปในระบบ

เมื่อเราต้องการให้ใครสามารถเข้ามาใช้ระบบของเราได้ เราจะต้องเพิ่มชื่อของเขาเข้าไปในรายการ user ของระบบ โดยเราจะใช้คำสั่ง useradd ซึ่งมีรูปแบบต่างๆ ไปดังนี้

`useradd -u <uid> -g <group> -d <home dir> -s <shell> -m login-name`
 โดย Option แต่ละส่วนมีความหมายดังนี้

-u <uid>	กำหนดหมายเลข User ID ซึ่งเป็นหมายเลขประจำตัวผู้ใช้ทุกคนและจะไม่ซ้ำกัน
-g <group>	กำหนด group ให้กับ user
-d <home dir>	กำหนด directory บ้าน
-s <shell>	กำหนด Shell ให้กับ user
-m	บอกให้ทำการสร้าง directory บ้านขึ้นมา ถ้ายังไม่มี
login-name	คือชื่อ login ของผู้ใช้ใหม่

ใน UNIX บางตัวอาจใช้ Option ที่แตกต่างจากนี้ไปก็ได้ ต่อไปนี้เป็นตัวอย่างการใช้คำสั่ง `useradd`
`useradd -u 120 -g teacher -d /home/arnan -s /bin/sh -m arnan`

คำสั่งข้างต้นเป็นการเพิ่ม user ที่ชื่อ `arnan` เข้าไปในระบบโดยมี `user-id=120` , อยู่ในกลุ่ม `teacher`, Home directory อยู่ที่ `/home/arnan` และ ใช้ `Broune Shell`

เมื่อเราเพิ่ม user เข้าไปในระบบโดยใช้คำสั่ง `useradd` แล้ว user นั้นจะยังไม่สามารถเข้ามาใช้งานได้ เพราะเรายังไม่ได้กำหนด password ให้กับผู้ใช้ใหม่ การกำหนด password นั้นเราจะใช้คำสั่ง `passwd` เช่น

ชื่อ user ที่ต้องการเปลี่ยน

```

alphar:~# passwd arnan
Changing password for arnan
Enter new password:
Re-type new password:
    
```

ป้อน password 2 รอบ

การสร้าง user นี้จริงๆ แล้วเป็นการเพิ่มรายการใหม่เข้าไปในไฟล์ที่ชื่อ `/etc/passwd` ไฟล์ๆ นี้มีลักษณะดังแสดงในรูปต่อไปนี้

```

root:SSwKjrA3vsNLo:0:0:root:/root:/bin/bash
bin:*:1:1:bin:/bin:
daemon:*:2:2:daemon:/sbin:
adm:*:3:4:adm:/var/adm:
lp:*:4:7:lp:/var/spool/lpd:
sync:*:5:0:sync:/sbin:/bin/sync
shutdown:*:6:0:shutdown:/sbin:/sbin/shutdown
postmaster:*:14:12:postmaster:/var/spool/mail:/bin/bash
nobody:*:-1:100:nobody:/dev/null:
ftp:*:404:1::/home/ftp:/bin/bash
    
```

```
guest:z9nQ6gr5DAD3c:405:100:guest:/home/guest:/bin/bash
arnan:ZJyeBGNF56SGU:501:100:Arnan Sipitakiat:/home/arnan:/bin/bash
```

จากตัวอย่างข้างต้นหนึ่งบรรทัดก็จะแทนข้อมูลของผู้ใช้หนึ่งคน ซึ่งในแต่ละบรรทัดก็จะแบ่งออกเป็นหลายๆ ส่วนดังนี้

```
arnan:ZJyeBGNF56SGU:501:100:Arnan Sipitakiat:/home/arnan:/bin/bash
```

The diagram shows the following breakdown of the user entry:

- arnan: user name
- ZJyeBGNF56SGU: encrypted password
- 501: UID
- 100: GID
- Arnan Sipitakiat: comment
- /home/arnan: home dir
- /bin/bash: shell

ถ้าเราไม่ต้องการใช้คำสั่ง `useradd` เราสามารถเข้าไปแก้ไขไฟล์ `/etc/passwd` นี้โดยตรงแทนก็ได้

4.3.2 userdel ใช้งาน user

ถ้าเราต้องการลบ user คนใดที่เราสามารถทำได้โดยการใช้คำสั่ง `userdel` เช่น

```
userdel arnan
```

ตัวอย่างข้างต้นเป็นการลบ user ที่ชื่อ `arnan` ออกไปจากระบบ ถ้าเราไม่ใช้คำสั่ง `userdel` เราสามารถลบ user ได้โดยการลบบรรทัดในไฟล์ `/etc/passwd` ก็ได้

4.3.3 การเปลี่ยนแปลงรายละเอียดของ user และการ disable user

ถ้าเราต้องการเปลี่ยนแปลงรายละเอียดบางอย่างของ user เช่น home directory หรือ shell ที่ใช้ เราสามารถทำได้โดยการใช้คำสั่ง `usermod` แต่ใน UNIX บางระบบจะมีคำสั่งที่ทำหน้าที่นี้โดยเฉพาะให้ได้แก่ คำสั่ง `usermod` เป็นต้น

ส่วนการ disable user นั้นเราจะทำเมื่อเราต้องการยับยั้งการเข้าใช้งานของ user บางคนไว้ชั่วคราว โดยยังไม่ลบ user นั้นทิ้ง วิธีง่ายที่สุดที่เราสามารถทำได้คือ เปลี่ยน password ของ user นั้นใหม่ แต่ก็ทำให้ password เดิมของ user นั้นหายไป อีกวิธีหนึ่งที่ใช้ได้คือการเข้าไปแก้ไขไฟล์ `/etc/passwd` โดยอาจเพิ่มตัวอักษรอะไรสักตัวหนึ่งเข้าไปใน password เดิม เช่น จากรูปใน 4.3.1 password ของ user ที่ชื่อ `arnan` ถูกเก็บไว้เป็น

```
ZJyeBGNF56SGU
```

password นี้เป็น password ที่ถูกเข้ารหัสไว้แล้ว เราสามารถเพิ่มตัวอักษรเข้าไปได้เช่น

```
!ZJyeBGNF56SGU
```

ตัวอย่างข้างต้นเราใส่เครื่องหมาย `!` นำหน้า password เดิม ซึ่งจะทำให้ password ทั้งหมดเปลี่ยนไป เมื่อเราต้องการให้ password เก่ากลับมาเราก็ทำได้โดยการลบเครื่องหมาย `!` นี้ทิ้ง

4.4 ms backup ข้อมูล

โดยปกติในระบบ UNIX ที่มีผู้ใช้มากๆ ผู้ดูแลระบบจะต้องทำการ backup ข้อมูลของระบบไว้ โดยการ backup นั้นสามารถทำได้โดยใช้คำสั่งหลายคำสั่ง แต่คำสั่งที่เป็นที่นิยมคือคำสั่ง tar

4.4.1 tar ใช้ในms backup ข้อมูล

คำสั่งนี้จะทำการรวบรวมไฟล์หลายๆ ไฟล์เข้าด้วยกันเป็นไฟล์เดียว ซึ่งมีรูปแบบการใช้งานดังนี้

```
tar -cvf <tar filename> <files to tar>
```

คำสั่งข้างต้นจะเป็นการสร้างไฟล์ tar ขึ้นมา เช่น

```
tar -cvf test.tar *
```

ตัวอย่างข้างต้นนี้จะทำการ copy ไฟล์ทุกไฟล์ใน directory ปัจจุบัน รวมทั้งไฟล์ใน directory ย่อย มาไว้ในไฟล์ที่ชื่อว่า test.tar

ถ้าเราต้องการขยายไฟล์ออก เราก็สามารถทำได้โดยใช้คำสั่ง tar ในรูปแบบต่อไปนี้

```
tar -xvf <tar file>
```

ยกตัวอย่างเช่น

```
tar test.tar
```

จะเป็นการขยายไฟล์ต่างๆ ใน test.tar ออกมา

4.4.2 การย่อขนาดไฟล์

ในการใช้คำสั่ง tar นั้นจะเป็นการรวมไฟล์หลายๆ ไฟล์เข้าด้วยกันเท่านั้น แต่ขนาดของมันไม่ได้ถูกย่อลงเลย แต่ในการ backup ข้อมูลนั้นเรานิยมที่จะทำการย่อข้อมูลก่อนเก็บเพื่อประหยัดเนื้อที่ ใน UNIX จะมีโปรแกรมที่ใช้ย่อขนาดไฟล์ที่เป็นที่นิยมอยู่สองตัวด้วยกันคือ compress และ gzip ซึ่งมีรูปแบบการใช้งานดังนี้

```
compress <file to compress>
```

```
gzip <file to compress>
```

ไฟล์ที่ได้จากการใช้คำสั่ง compress จะมีนามสกุล .Z ตามท้าย ส่วน gzip จะได้ไฟล์ที่มีนามสกุล .gz ตามท้าย โดยปกติแล้วไฟล์ที่ย่อโดยใช้ gzip จะมีขนาดเล็กกว่าไฟล์ที่ใช้ compress

ในกรณีที่เรต้องการขยายไฟล์กลับคืน เราจะใช้คำสั่งในรูปแบบต่อไปนี้

```
uncompress <.Z file>
```

```
gzip -d <.gz file> or gunzip <.gz file>
```

4.5 การจัดการ file system และ disk

คำสั่งในหัวข้อนี้จะเกี่ยวกับ disk ในระบบ ซึ่งมักจะหมายถึงการเพิ่ม disk ถ้าหากว่าเราต้องการเพิ่มพื้นที่ disk ในระบบของเรา โดยการใส่ disk ตัวใหม่เข้าไปสิ่งต่าง ๆ ที่จะต้องทำคือ

1. ทำการแบ่ง partition ตามต้องการ
2. สร้าง file system ขึ้นมาบน partition
3. ทำการติดตั้ง file system นั้นเข้าสู่ระบบโดยการ mount disk

ในขั้นตอนของการแบ่ง partition นั้นเราอาจข้ามไปได้ถ้าไม่ต้องการแบ่งมัน คำสั่งที่ใช้ในการแบ่ง partition ก็ได้แก่คำสั่ง fdisk

4.5.1 การสร้าง file system

เมื่อเราแบ่ง partition เรียบร้อยแล้ว ต่อไปก็ต้องสร้าง file system ขึ้นมาบนแต่ละ partition โดย file system นั้นมีอยู่หลายแบบแล้วแต่เราจะเลือกใช้แบบไหน เช่น UFS, DOS-FAT, EXT2 เป็นต้น UNIX แต่ละตัวก็จะรู้จักและใช้งาน file system ได้หลายชนิด ซึ่งขึ้นอยู่กับผู้ดูแลระบบว่าจะเลือกใช้แบบใด คำสั่งที่ใช้ในการสร้าง file system นั้น มักจะต่างกันสำหรับ UNIX แต่ละตัว แต่โดยทั่วไปมักจะใช้คำสั่ง mkfs (make file system) ซึ่งมีรูปแบบการใช้งานดังนี้

```
mkfs -t <file system type> <device>
```

โดยที่ file system type คือชนิดของ file system ที่ต้องการจะสร้าง ซึ่งปกติแล้วสามารถหาดูได้จาก /etc/fs ส่วน device นั้นหมายถึงชื่อ device ของ disk ที่ต้องการ ซึ่งนี้จะอยู่ภายใต้ directory /dev และ UNIX แต่ละตัวก็จะใช้ชื่อที่แตกต่างกัน เช่น ใน Linux จะใช้ /dev/hda สำหรับ Harddisk ตัวที่ 1 และ /dev/hdb สำหรับตัวที่สอง และ /dev/hdc สำหรับตัวที่สาม ไปเรื่อยๆ ใน Harddisk แต่ละตัวก็สามารถมีได้หลาย partition จึงแบ่งเป็น /dev/hdax โดย x แทนเลขของ partition เช่น /dev/hda1, /dev/hda2 แทน Harddisk ตัวที่หนึ่ง partition ที่ 1 และ 2 ตามลำดับ เป็นต้น

```
mkfs -t ext2 /dev/hda2
```

ตัวอย่างคำสั่งข้างต้นเป็นการสร้าง file system ชนิดที่ชื่อ EXT2 ขึ้นมาโดยใช้ Harddisk ตัวที่ 1 partition ที่ 2

4.5.2 การ mount และ unmount file system

เมื่อเราทำการสร้าง file system ขึ้นมาแล้วสิ่งสุดท้ายที่จะต้องทำเพื่อให้ระบบสามารถใช้งาน disk นั้นได้คือ การ mount มันเข้ามาในระบบ ใน UNIX นั้นจะไม่มอง disk แต่ละตัวเป็น drive แยกต่างหากเหมือนใน DOS แต่จะมองเป็น directory แทน โดยเราจะใช้คำสั่ง mount ในการกำหนดว่า ต้องการให้ disk ไปปรากฏอยู่ที่ directory ใด โดยมีรูปแบบการใช้งานดังนี้

```
mount -t <file system type> <device name> <mount point>
```

โดย <file system type> ก็คือชนิดของ file system ที่ต้องการ mount ส่วน device name ก็คือชื่อของ disk นั้นๆ และ mount point หมายถึง directory ที่ต้องการให้ disk นั้นไปปรากฏ โดย directory นั้นจะต้องมีอยู่แล้วและว่างอยู่ไม่มี disk อื่น mount อยู่ก่อนหน้า

```
mount -t msdos /dev/hda1 /mnt
```

ตัวอย่างข้างต้นเป็นการ mount Harddisk แรก partition แรก ซึ่งเป็น partition ของ DOS (ในกรณีที่เครื่องนั้นมีทั้ง DOS และ UNIX) มาไว้ที่ directory /mnt เมื่อเราทำคำสั่งนี้เสร็จแล้ว drive C ของ DOS ก็จะมาปรากฏอยู่ใน directory /mnt ของ UNIX

ถ้าเราต้องการดูว่าในปัจจุบันเรา mount อะไรไว้บ้างก็ทำได้โดยการใช้คำสั่ง mount เฉยๆ เช่น

```
alphar:~# mount
/dev/hda3 on / type ext2 (rw)
/dev/hda1 on /dos type msdos (rw)
none on /proc type proc (rw)
```

ในทางตรงกันข้ามถ้าเราต้องการที่จะเอา disk ตัวใดออกจากระบบ เราจะใช้คำสั่ง umount ซึ่งมีรูปแบบการใช้งานดังนี้

umount <device name or mount point>

พารามิเตอร์ที่ต้องใส่ไปคือ ชื่อ device ของ disk หรือ directory ที่ทำการ mount disk นั้นเข้ามา ยกตัวอย่างเช่น

```
umount /mnt
```

ตัวอย่างข้างต้นเป็นการ unmount disk ที่ mount อยู่ที่ directory /mnt

บทที่ 5

การใช้ vi

vi คือ editor ที่ใช้กันมากในระบบ UNIX เนื่องจากเป็น editor ที่เก่าแก่และอยู่คู่กับระบบ UNIX มา นานมาก และถือได้ว่าเป็น Full Screen Editor ตัวแรกๆ ของระบบคอมพิวเตอร์ ถึงแม้ว่าในปัจจุบันจะมี editor ตัวใหม่เกิดขึ้นมากมาย ซึ่งใช้งานได้ง่ายกว่า vi มาก แต่ vi ก็ยังเป็น editor ที่สมควรทำความรู้จักไว้อยู่ เพราะ เป็น editor ที่มั่นใจได้ว่าต้องมีใน UNIX ทุกตัว ในขณะที่ editor ตัวอื่นอาจจะไม่มีใน UNIX บางระบบก็ได้ และเนื่องจากว่า vi นั้นต้องใช้ทักษะและความเคยชินค่อนข้างมากในการใช้งาน ดังนั้นถ้าไม่รู้จักใช้มันเลย เวลา ถึงคราวที่จะต้องใช้จริงๆ อาจจะลำบากมาก

5.1 โหมดการทำงานของ vi

ใน vi นั้นจะมีโหมดการทำงานอยู่ 2 โหมดด้วยกัน คือ

1. Command Mode - เป็นโหมดที่ใช้ในการสั่งงาน
2. Insert Mode - เป็นโหมดที่ใช้ในการป้อนหรือพิมพ์ข้อความ

เมื่อเราใช้งาน vi มันจะต้องอยู่ในโหมดใดโหมดหนึ่งเสมอ และการทำงานกับโหมดทั้งสองนี้มักเป็นสิ่งที่ สับสนวุ่นวายมากสำหรับคนที่ไม่เคยเล่น vi มาก่อน เนื่องจาก vi ไม่มีอะไรบนจอเป็นเครื่องหมายให้เห็นว่ามัน อยู่ในโหมดใด

ปกติเมื่อเราเริ่มต้นใช้งาน vi โหมดแรกที่เราจะอยู่คือ Command Mode ซึ่งเป็นโหมดที่เราจะใช้ในการ สั่งงาน vi ให้ทำสิ่งต่างๆ ทุกสิ่งยกเว้นการป้อนข้อความ เช่น การเลื่อนตำแหน่ง cursor การลบตัวอักษร, คำ หรือบรรทัด การแทรกบรรทัด ฯลฯ เมื่อใดที่เราพร้อมจะพิมพ์ข้อความ เราจะต้องเปลี่ยนไปใช้ Insert Mode ก่อนเสมอ และเช่นเดียวกันถ้าเราพิมพ์ข้อความอยู่และต้องการจะสั่งงานให้ vi ทำอย่างอื่นที่อยู่นอกเหนือจาก การพิมพ์ข้อความ เราจะต้องเปลี่ยนกลับไปใช้ Command Mode ก่อน เนื่องจาก vi ไม่มี menu ดังนั้นการสั่ง งาน vi นั้นจะใช้การกดตัวอักษรบนแป้นพิมพ์แทน เช่น ปุ่ม x จะลบตัวอักษรหนึ่งตัว ปุ่ม A จะแทรกตัวอักษร ที่ตำแหน่งท้ายบรรทัด เป็นต้น

5.2 เริ่มต้นใช้งาน vi

การเข้าสู่โปรแกรม vi นั้นทำได้ง่าย ๆ โดยการเรียกใช้ดังนี้

vi [file name]

โดย file name คือชื่อไฟล์ที่ต้องการเปิดขึ้นมาแก้ไข ถ้าไฟล์นั้นยังไม่มีอยู่ก็จะเป็นการสร้างไฟล์นั้นขึ้น มา เมื่อเราเข้าสู่ vi เราจะอยู่ใน command mode การเปลี่ยนไปเป็น insert mode นั้นทำได้หลายวิธีแต่วิธีที่เป็น ที่นิยมมากที่สุดคือการกดอักษร I หลักจากที่กดอักษร I แล้ว ทุกอย่างบนหน้าจอจะเหมือนเดิม แต่ตัวอักษรทุก

ตัวที่เรากดหลังจากนั้นจะไปปรากฏอยู่บนจอ เมื่อเราต้องการเปลี่ยนกลับไป `command mode` ให้กดปุ่ม `Escape` ซึ่งจะไม่มียอะไรเกิดขึ้นบนหน้าจอเช่นกัน แต่ตัวอักษรที่เราพิมพ์หลังจากนั้นจะเป็นการส่งคำสั่งให้กับ `vi`

5.3 คำสั่งต่าง ๆ ใน Command Mode

คำสั่งต่าง ๆ ใน `command mode` นั้นมีอยู่มากมาย ซึ่งสามารถแบ่งออกเป็นกลุ่มๆ ได้ดังตารางต่อไปนี้
คือ

5.3.1 คำสั่งในการเลื่อน cursor

l	ไปทางขวา 1 ตัวอักษร
h	ไปทางซ้าย 1 ตัวอักษร
j	ลงล่างหนึ่งบรรทัด
k	ขึ้นบนหนึ่งบรรทัด
w	ไปที่คำถัดไปทางขวา
W	ไปที่คำนับตามจริงตัวถัดไป
b	ไปที่คำก่อนหน้า
B	ไปที่คำนับตามจริงก่อนหน้า
o	ไปต้นแถวปัจจุบัน
\$	ไปท้ายแถวปัจจุบัน
(ไปต้นประโยคปัจจุบัน
)	ไปท้ายประโยคปัจจุบัน
{	ไปต้นย่อหน้าปัจจุบัน
}	ไปท้ายย่อหน้าปัจจุบัน
G	ไปที่บรรทัดสุดท้ายของไฟล์
nG	ไปที่บรรทัดที่ n ในไฟล์

จากตารางข้อต้นจะเห็นได้ว่ามีคำอยู่บางคำที่มีความหมายพิเศษ ได้แก่

คำ หมายถึงตัวอักษร 1 กลุ่มซึ่งถูกแยกจากอักษรกลุ่มอื่นด้วย ตัวอักษรต่อไปนี้

- ตัวเลข
- ตัวอักษรที่ไม่ใช่ตัวอักษรและตัวเลข
- ช่องว่าง และ tab

เช่น

anan	1 คำ
anan?	2 คำ คือ anan กับ ?
My name! Anan	4 คำ คือ My, name, ! และ Anan

ค่านับตามจริง หมายถึงตัวอักษร 1 กลุ่มซึ่งถูกแยกจากอักษรกลุ่มอื่นด้วย ช่องว่าง และ tab เท่านั้น

เช่น

arnan	1 คำ
arnan?	1 คำ
My name! Arnan	3 คำ คือ My, name! และ Arnan

แถว หมายถึงกลุ่มของคำ ซึ่งแถวจะถูกขึ้นด้วยการกดปุ่ม Enter

ประโยค หมายถึงกลุ่มของคำ ซึ่งลงท้ายด้วย จุด(.) ตกใจ (!) หรือ คำถาม (?) และ ตามด้วย ช่องว่างสองตัว

ย่อหน้า คือกลุ่มของคำหรือประโยค ซึ่งย่อหน้าจะถูกขึ้นระหว่างกันด้วยแถวว่างหนึ่งแถว นั่นคือเราต้องกดปุ่ม Enter สองครั้งเพื่อแบ่งย่อหน้าออกจากกัน

5.3.2 คำสั่งที่ใช้ในrsau

x	ลบ 1 ตัวอักษรที่ตำแหน่ง cursor
X	ลบ 1 ตัวอักษรที่อยู่ทางซ้ายของ cursor
dw	ลบ 1 คำทางขวา
dW	ลบ 1 คำนับตามจริงทางขวา
db	ลบ 1 คำทางซ้าย
dB	ลบ 1 คำนับตามจริงทางซ้าย
dd	ลบ 1 แถวที่ cursor อยู่
d0	ลบบย้อนตั้งแต่ตำแหน่ง cursor จนถึงต้นแถว
D	ลบตั้งแต่ตำแหน่ง cursor จนถึงท้ายแถว
d)	ลบตั้งแต่ตำแหน่ง cursor ถึงท้ายประโยค
d(ลบตั้งแต่ตำแหน่ง cursor ถึงต้นประโยค
d}	ลบตั้งแต่ตำแหน่ง cursor ถึงท้ายย่อหน้า
d{	ลบตั้งแต่ตำแหน่ง cursor ถึงต้นย่อหน้า

5.3.3 คำสั่งที่ใช้ในการแทรก และเปลี่ยนจาก Command mode ไปเป็น Insert mode

I	แทรกที่ตำแหน่ง cursor
i	แทรกที่ตำแหน่งแรกของแถวปัจจุบัน
a	แทรกหลังตำแหน่ง cursor
A	แทรกที่ตำแหน่งท้ายสุดของแถวปัจจุบัน
o	แทรกแถวใหม่เข้าไปหนึ่งแถวใต้ตำแหน่ง cursor
O	แทรกแถวใหม่เข้าไปหนึ่งแถวเหนือตำแหน่ง cursor

5.3.4 คำสั่งที่ใช้ในการค้นหา

/คำ	ค้นหา "คำ"
n	ค้นหาต่อไปในทิศทางเดิม
N	ค้นหาในทิศทางตรงข้ามกับทิศเดิม

5.3.4 ms Copy, Cut และ Paste ข้อความ

yy	Copy แถวปัจจุบันเข้าไปใน general buffer
y(move cmd)	Copy ข้อความที่ cursor เคลื่อนผ่าน ตาม move cmd ซึ่งคือคำสั่งที่ใช้ในการเลื่อน cursor ใน 5.3.1 เช่น yw เป็นการ Copy คำที่ตำแหน่ง cursor
p	Paste จาก general buffer ไปที่ตำแหน่ง cursor
"<ch><cmd>	<p><ch> คือตัวอักษรที่จะใช้เป็น buffer ซึ่งเลือกใช้ได้ตั้งแต่ a-z</p> <p><cmd> คือคำสั่งที่ใช้ในการ Copy (เริ่มด้วยตัว y) หรือ คำสั่งที่ใช้ในการ Cut ซึ่งเป็นคำสั่งในกลุ่มของการลบใน 5.3.2 เช่น</p> <p>“ayy เป็นการ copy บรรทัดปัจจุบันไปไว้ใน buffer a</p> <p>“cdd เป็นการ cut บรรทัดปัจจุบันไว้ใน buffer c</p> <p>คำสั่งนี้แสดงให้เห็นว่าเรามีที่เก็บข้อมูลที่ Copy หรือ Cut ถึง 26 ที่ (a-z)</p>
"<ch>p	<p>เป็นการ paste จาก buffer <ch> มาที่ตำแหน่ง cursor เช่น</p> <p>“ap เป็นการ paste ออกจาก buffer a</p>

5.3.4 คำสั่งอื่น ๆ

u	Undo ยกเลิกคำสั่งที่ทำไปก่อนหน้า
:w	save file
:w filename	ตั้งชื่อไฟล์ และ save
:q	ออกจากโปรแกรม vi
ZZ	ออกจากโปรแกรม vi
:set number	ทำให้เกิดมีเลขบรรทัดปรากฏขึ้น ถ้าจะเอาเลขออกให้ใช้ :set nonumber

บทที่ 6

บริการพื้นฐานของเครือข่าย INTERNET

ในบทนี้เราจะทำการศึกษาถึง บริการพื้นฐานต่างๆ ที่มีอยู่บน INTERNET และยังคงเป็นที่นิยมใช้กันอยู่ในปัจจุบัน

6.1 บริการตรวจสอบเครื่องบนระบบเครือข่าย

6.1.1 ping – ใช้ในการตรวจสอบเครื่องปลายทางว่าทำงานอยู่หรือไม่

คำสั่งนี้เราจะใช้เมื่อเราทราบชื่อเครื่อง หรือหมายเลข IP Address ของเครื่องที่ต้องการจะติดต่อ และต้องการตรวจสอบดูว่าเครื่องๆ นั้นทำงานอยู่หรือไม่ รูปแบบการใช้งานคำสั่งเป็นดังนี้

```
ping <Host name or IP Address>
```

ยกตัวอย่างเช่น

```
ping 202.44.145.20
```

คำสั่งข้างต้นเป็นคำสั่งที่ใช้ตรวจสอบว่าระบบของเราสามารถติดต่อกับเครื่อง 202.44.145.20 ได้หรือไม่ ถ้าได้ผลลัพธ์ที่ได้อาจมีลักษณะดังนี้

```
alphar:~# ping 202.44.145.20
PING 202.44.145.20 (202.44.145.20): 56 data bytes
64 bytes from 202.44.145.20: icmp_seq=0 ttl=32 time=2.7 ms
64 bytes from 202.44.145.20: icmp_seq=1 ttl=32 time=1.4 ms
64 bytes from 202.44.145.20: icmp_seq=2 ttl=32 time=1.4 ms
64 bytes from 202.44.145.20: icmp_seq=3 ttl=32 time=1.4 ms
64 bytes from 202.44.145.20: icmp_seq=4 ttl=32 time=1.4 ms
64 bytes from 202.44.145.20: icmp_seq=5 ttl=32 time=1.4 ms

--- 202.44.145.20 ping statistics ---
6 packets transmitted, 6 packets received, 0% packet loss
round-trip min/avg/max = 1.4/1.6/2.7 ms
```

บางครั้งคำสั่ง ping อาจแสดงผลลัพธ์ขึ้นมาสั้น ๆ ก็ได้เช่น

```
202.44.145.20 is alive
```

6.1.2 nslookup – ใช้ในการขอข้อมูลเกี่ยวกับชื่อเครื่องหรือเลข IP

เราจะใช้คำสั่งนี้ในกรณีที่เรต้องการแปลงเลข IP Address ให้เป็นชื่อเครื่อง หรือในทางตรงกันข้ามก็ได้ เช่นถ้าเราทราบเลข IP Address มาตัวหนึ่งและอยากจะทราบว่าเครื่องๆ นั้นชื่อว่าอะไร ก็สามารถใช้คำสั่งนี้ได้ โดย nslookup มีรูปแบบการใช้งานดังนี้

nslookup <Host name or IP Address>

ยกตัวอย่างเช่น

```
nslookup 202.44.145.20
nslookup roger.mptc.eng.cmu.ac.th
```

ผลที่จะได้จากตัวอย่างแรกคือ ชื่อของเครื่องที่มีเลข IP Address เป็น 202.44.145.20 ซึ่งในที่นี้คือ เครื่องที่มีชื่อว่า ds90.intanon.nectec.or.th ส่วนคำสั่งที่สองผลลัพธ์ที่ได้คือหมายเลข IP Address ของเครื่อง ที่ชื่อ roger.mptc.eng.cmu.ac.th ซึ่งในที่นี้คือ 202.44.248.85

6.1.3 finger – ใช้ในการขอรายชื่อ user ที่กำลังใช้งานเครื่อง ๆ หนึ่งอยู่

คำสั่งนี้จะคล้ายๆ กับคำสั่ง who ในบทที่ 2 คือจะแสดงรายการ user ที่กำลังใช้งานเครื่องอยู่ แต่รายละเอียดที่ได้จากคำสั่ง finger จะมากกว่า รูปแบบการใช้งานคำสั่ง finger เป็นดังนี้

finger <@hostname>

ยกตัวอย่างเช่น

```
alphar:~$ finger @ds90.intanon.nectec.or.th
Login      Name                Tty  Idle  Login Time   Office
arnan      Arnan Sipitakiat    p0   Jan 13 19:21 (atec.intanon.nec)
guest      guest                *p1   Jan 13 19:28 (atec.intanon.nec)
```

ในตัวอย่างข้างต้นผลที่ได้แสดงว่ามี user ที่ใช้งานเครื่อง ds90.intanon.nectec.or.th อยู่ 2 คนคือ arnan และ guest ข้อมูลต่างๆ ที่ปรากฏมีความหมายดังนี้

Login	- ชื่อ login ของ user ที่กำลังใช้งานเครื่องอยู่
Name	- ชื่อจริงของ user
Tty	- ถ้า user ใดที่มีเครื่องหมาย * อยู่แสดงว่า user นั้นใช้คำสั่ง mesg n ไว้
Idle	- เป็นระยะเวลาที่ user นั้นไม่ได้กดปุ่มใดๆ บนแป้นพิมพ์
Login Time	- เวลาที่ user นั้น login เข้ามาใช้ระบบ
Office	- user คนนั้น login เข้ามาจากที่ใด

6.2 Telnet – เป็นบริการขอใช้เครื่องจากระยะไกล

บริการนี้อ่อนุญาตให้เราสามารถเข้าใช้เครื่องคอมพิวเตอร์จากระยะไกลได้ โดยโปรแกรม telnet นี้จะทำการจำลองตัวเครื่องที่เราใช้อยู่เป็น dumb terminal และคอยรับส่งข้อมูลไปมาระหว่าง terminal และ server รูปแบบการใช้คำสั่งนี้คือ

telnet <Host name or IP Address>

ยกตัวอย่างเช่น

```
telnet alphar.intanon.nectec.or.th
```

คำสั่งข้างต้นเป็นการขอใช้เครื่อง alphas.intanon.nectec.or.th เมื่อโปรแกรม Telnet ติดต่อกได้แล้วสิ่งที่ปรากฏบนจอจะเหมือนกับหน้าจอของการ Login ใน UNIX ทั่วๆ ไป เช่น

```

atec:~$ telnet alphas.intanon.nectec.or.th

Trying 10.0.2.20...
Connected to alphas.intanon.nectec.or.th.
Escape character is '^]'.

Linux 2.0.9 (alphas.intanon.nectec.or.th) (tty2)

alphas login: arnan
Password:
Last login: Mon Jan 13 19:21:32 from atec.intanon.nec
Linux 2.0.9.
alphas:~$
    
```

การที่เราจะเข้าใช้งานเครื่องใดได้ เราจะต้องมี user name และ password บนเครื่องนั้นเสมอ

6.3 pine – ใช้ในการส่งจดหมายอิเล็กทรอนิกส์ (e-mail)

บริการส่ง e-mail นั้นเป็นบริการที่ได้รับความนิยมมากนับตั้งแต่เริ่มต้นคิดค้นขึ้นมา ในบทที่ 2 เราได้ทำการศึกษาถึงการใช้งานโปรแกรม mail ไปแล้ว แต่โปรแกรม mail นั้นใช้งานได้ไม่สะดวกเท่ากับโปรแกรม pine นี้ เพราะโปรแกรม pine มีลักษณะการติดต่อกับผู้ใช้แบบ Menu การเรียกใช้โปรแกรม pine สามารถทำได้โดยการพิมพ์คำว่า pine ที่ UNIX prompt ซึ่งหน้าต่างของโปรแกรม pine จะเป็นดังในรูป

```

PINE 3.91  MAIN MENU                               Folder: INBOX  0 Messages

?  HELP                -  Get help using Pine
C  COMPOSE MESSAGE     -  Compose and send a message
I  FOLDER INDEX        -  View messages in current folder
I  FOLDER LIST        -  Select a folder to view
A  ADDRESS BOOK        -  Update address book
S  SETUP               -  Configure or update Pine
Q  QUIT                -  Exit the Pine program

Copyright 1989-1994.  PINE is a trademark of the University of Washington.
[Folder "INBOX" opened with 0 messages]
? Help                F PrevCmd                R RelNotes
C OTHER CMDS  L [ListFldrs]  N NextCmd                K KBLock
    
```

เมนูที่จะใช้กันบ่อยๆ คือ COMPOSE MESSAGE ซึ่งใช้ในการสร้างและส่งจดหมาย กับ FOLDER LIST ซึ่งใช้ในการอ่านจดหมาย

เมื่อเราเลือกเมนู COMPOSE MESSAGE โปรแกรมก็จะเข้าสู่หน้าต่างการสร้างจดหมาย ดังแสดงในรูปต่อไปนี้

```

PINE 3.91  COMPOSE MESSAGE                               Folder: INBOX  0 Messages
To      : arnan@ds90.intanon.nectec.or.th
Cc      :
Attchmnt:
Subject : Test mail
----- Message Text -----
Hello,
      This is a test mail. I hope you receive it. I used pine to write this
mail. It's a very easy program to use. I like it very much. You should
try to use it too.

bye,
Roger.
█

^G Get Help   ^X Send       ^R Read File  ^Y Prev Pg   ^K Cut Text   ^C Postpone
^C Cancel     ^J Justify    ^W Where is   ^V Next Pg   ^U UnCut Text ^T To Spell
    
```

แต่ส่วนของหน้าต่างมีความหมายดังนี้

- To** - จะส่งถึงใคร ให้ป้อน E-Mail Address ของผู้รับที่นี่
- Cc** - จะให้ส่งสำเนาจดหมายไปหาใครบ้าง ใส่เป็น E-Mail Address
- Attachment** - ไฟล์ที่ต้องการส่งไปด้วย
- Subject** - หัวข้อของจดหมายฉบับนี้
- Message Text** - พื้นที่ส่วนนี้ใช้ในการพิมพ์เนื้อความของจดหมาย

เมื่อเราพิมพ์จดหมายเสร็จแล้วและต้องการส่งให้กดปุ่ม Ctrl-X แต่ถ้าต้องการจะยกเลิกให้กดปุ่ม Ctrl-C

6.4 ftp – ใช้ในการ download และ upload ไฟล์ระหว่างเครื่อง

ftp (File Transfer Protocol) เป็นอีกบริการหนึ่งที่ได้รับคามนิยมเป็นอย่างมาก ซึ่งงานที่นำ ftp ไปใช้มากที่สุดก็คือการทำธุรกิจเกี่ยวกับ Software โดย user มักจะทำการ download โปรแกรมจากที่ต่างๆ ใน INTERNET มาใช้ โปรแกรมเหล่านี้บางโปรแกรมก็แจกฟรี บางโปรแกรมก็เป็น Shareware หรือโปรแกรมที่ให้ทดลองนำมาใช้ดูก่อนถ้าพอใจจริงจะซื้อ รูปแบบการใช้งานโปรแกรม ftp มีดังนี้

ftp hostname

ยกตัวอย่างเช่น

```
ftp alphar.intanon.nectec.or.th
```

เมื่อเราเชื่อมต่อกับเครื่องปลายทางได้แล้วก็จะมี Login prompt ปรากฏขึ้นมา โดยทั่วไป เราจะสามารถ login เข้าได้โดยใช้ user name ที่ชื่อว่า anonymous ซึ่งจะหมายถึง “ผู้มาเยือนทั่วไป” เมื่อเราป้อน

user name เป็น anonymous แล้ว ส่วนใหญ่เครื่องปลายทางจะให้เราป้อน e-mail address ของเราเข้าไปเป็น password โดยเราไม่จำเป็นต้องเขียนจนครบ ให้พิมพ์ไปจนถึงตัวอักษร @ ก็เพียงพอแล้ว

```
alphar:~$ ftp alphar.intanon.nectec.or.th
Connected to alphar.intanon.nectec.or.th.
220 alphar FTP server (Tue Aug 8 15:50:43 CDT 1995) ready.
Name (alphar:arnan): anonymous
331 Guest login ok, send your complete e-mail address as password.
Password: <type username@ here>
```

```
230-Welcome, archive user! This is an experimental FTP server. If have any
230-unusual problems, please report them via e-mail to root@alphar
230-If you do have problems, please try using a dash as the first character
230-of your password - this will turn off the continuation messages that may
230-be confusing your ftp client.
230-
230 Guest login ok, access restrictions apply.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

เมื่อเรา login เข้าเรียบร้อยแล้ว เราก็จะอยู่ใน ftp prompt ซึ่งคำสั่งต่างๆ ที่เราสามารถป้อนเข้าไปใช้งานได้แก่

pwd	- แสดง directory ที่เราอยู่ในปัจจุบัน
cd	- ใช้เปลี่ยน directory
ls -l หรือ dir	- ใช้แสดงไฟล์ใน directory
bin	- ใช้กำหนดลักษณะการส่งข้อมูลเป็นแบบ binary
ascii	- ใช้กำหนดลักษณะการส่งข้อมูลเป็นแบบ ASCII (สำหรับ Text file)
get <file>	- ใช้ download file
put <file>	- ใช้ upload file
mget <file1><file2> ...	- ใช้ download file หลายๆ ไฟล์
mput <file1><file2> ...	- ใช้ upload file หลายๆ ไฟล์
prompt	- ใช้กำหนด prompt โหมดว่าจะให้ถามก่อน download/upload หรือไม่ (ใช้กับ mput และ mget)
bye	- ใช้ออกจากโปรแกรม ftp

ยกตัวอย่างเช่น

```
ftp> cd pub
250 CWD command successful.
ftp> ls -l
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
total 2
drwxrwxr-x  2 root    wheel      1024 Oct 11 22:58 .
drwxrwxr-x  8 root    wheel      1024 Oct 11 22:58 ..
-rwxrwxr-x  1 root    wheel      125 Jul 10 1993 test_file
226 Transfer complete.
ftp> bin
```

```
200 Type set to I.  
ftp> get test_file  
200 PORT command successful.  
150 Opening BINARY mode data connection for test_file (125 bytes).  
226 Transfer complete.  
ftp> bye  
221 Goodbye.
```

ในตัวอย่างข้างต้นเราเริ่มต้นด้วยการ **cd** เข้าไปใน directory ชื่อว่า **pub** แล้วใช้คำสั่ง **ls -l** เพื่อดูรายการของไฟล์ใน directory นั้น จะเห็นได้ว่ามีไฟล์ที่ชื่อ **test_file** อยู่ สมมุติว่าเราต้องการ download ไฟล์นี้ เราก็เริ่มต้นด้วยการกำหนดว่าต้องการจะ download แบบ **binary** หรือ **ascii** ในตัวอย่างนี้สมมุติว่าไฟล์นี้เป็น **binary** ดังนั้นเราจะใช้คำสั่ง **bin** แล้วตามด้วย **get test_file** เมื่อ download มาเสร็จแล้วเราก็ออกจากโปรแกรม ftp โดยใช้คำสั่ง **bye**