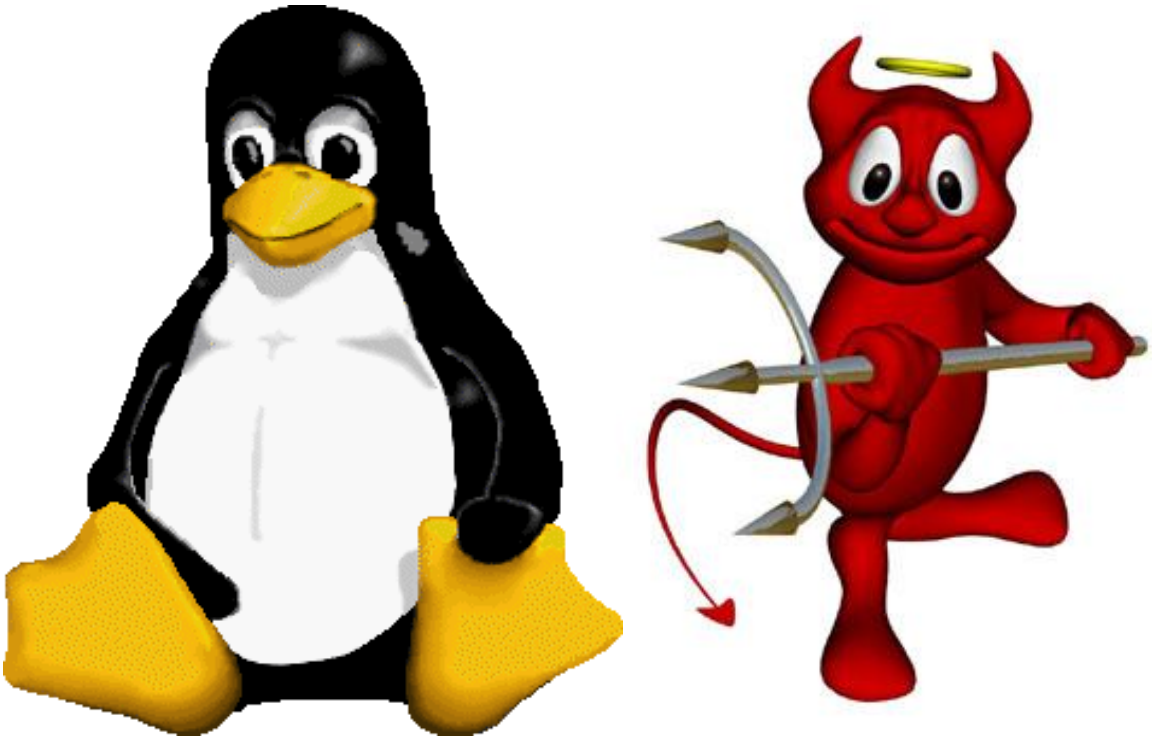
การอบรมยูนิกซ์เชิงปฏิบัติการ



โดย

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

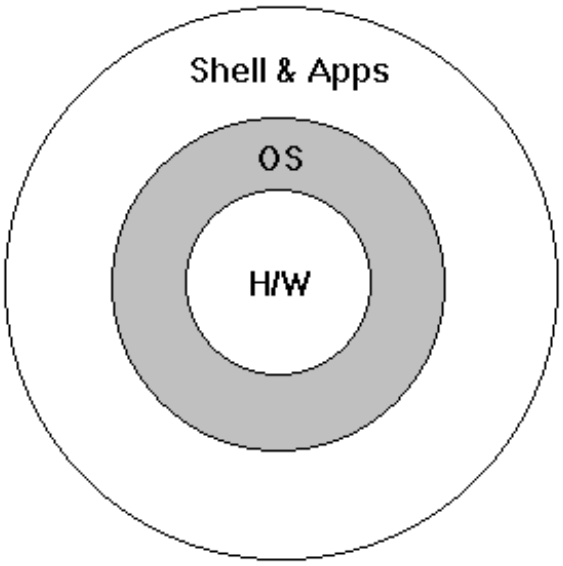
แนะนำระบบปฏิบัติการเบื้องต้น

"A good beginning makes a good ending"

– English proverb

ระบบปฏิบัติการสำหรับคอมพิวเตอร์ถือเป็นส่วนสำคัญยิ่งที่ทำให้มนุษย์ผู้ใช้งานใช้คอมพิวเตอร์ได้เต็มประสิทธิภาพอย่างง่ายดาย โดยทำหน้าที่อยู่ระหว่างฮาร์ดแวร์และแอพพลิเคชันซอฟต์แวร์ ในความเป็นจริงแล้วไม่มีนิยามใดที่สามารถนิยามระบบปฏิบัติการได้อย่างชัดเจน แต่ระบบปฏิบัติการนั้นจำเป็นต้องมีหน้าที่หลักสองประการคือ

1. เพิ่มความสามารถให้แก่เครื่องคอมพิวเตอร์
2. เป็นผู้จัดสรรทรัพยากร

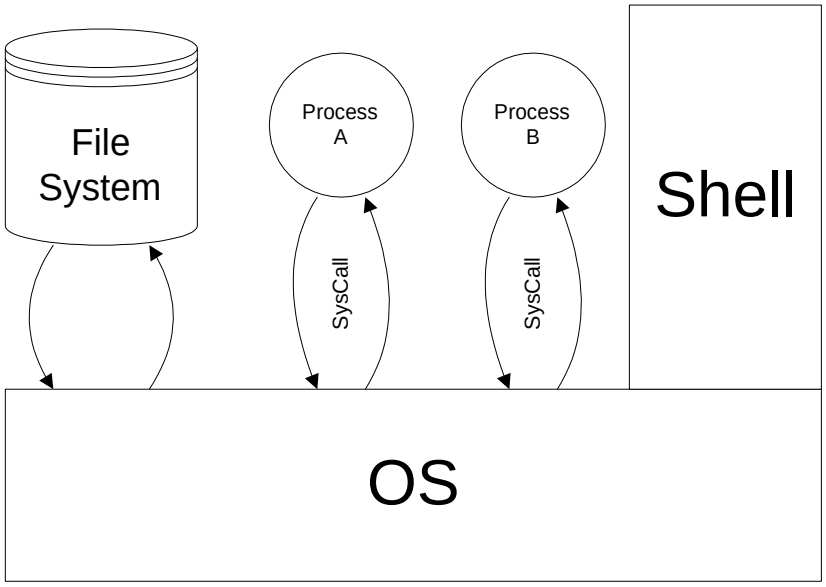


ผังแสดงชั้นความสัมพันธ์

ประวัติของระบบปฏิบัติการ แบ่งตามยุคสมัยของคอมพิวเตอร์

1. ยุคหลอดสูญญากาศและกระดานเชื่อมสาย : ถือได้ว่าไม่มีระบบปฏิบัติการใดเลย หากต้องการใช้งานก็สั่งผ่านกระดานเชื่อมสายหรือสวิตช์สัญญาณโดยตรง เพื่อบังคับอุปกรณ์อิเล็กทรอนิกส์อีกทีหนึ่ง
2. ยุคทรานซิสเตอร์และระบบแบตช์ : มีระบบปฏิบัติการเพื่อช่วยเหลือการประมวลผลบัตรเจาะรูแบบแบตช์อยู่บ้าง ไม่มีระบบโต้ตอบผู้ใช้ มีแต่รายงานผลลัพธ์ ต่อมาพัฒนาเป็นแบบสพูลเลอร์คือถ่ายข้อมูลไปยังสื่อกลางที่เป็นเทปแม่เหล็กเพื่อเร่งความเร็วด้านไอโอของระบบคอมพิวเตอร์
3. ยุคไอซีและมัลติโปรแกรมมิง : ระบบปฏิบัติการมีความสลับซับซ้อนขึ้นมาก เนื่องจากต้องรับภาระงานในแบบหลายงาน (จากมัลติโปรแกรมมิงเป็นมัลติทาสกิง) และหลายผู้ใช้ (จากไทม์แชริงเป็นมัลติยูสเซอร์) อีกทั้งมีระบบโต้ตอบผู้ใช้ซึ่งโดยมากเป็นแบบเท็กซ์

4. ยุคพีซี : ระบบปฏิบัติการมีความคล่องตัวสูง สวยงาม และใช้งานง่าย เหมาะสำหรับผู้ใช้ทั่วไปที่อาจไม่มีความรู้ทางคอมพิวเตอร์มากนัก ส่วนใหญ่มีระบบโต้ตอบผู้ใช้แบบกราฟิก และเชื่อมต่อระบบเครือข่ายคอมพิวเตอร์ได้ อีกทั้งมีการพัฒนาระบบปฏิบัติการให้ใช้งานโพรเซสเซอร์ได้มากกว่าหนึ่งตัว ระบบงาน



ควบคุมตามเวลาจริง และระบบประมวลผลแบบกระจายผ่านเครือข่ายคอมพิวเตอร์ เป็นต้น

ผังแสดงความสัมพันธ์ระหว่างแนวคิดหลักในระบบปฏิบัติการ

แนวคิดหลักในการทำความเข้าใจระบบปฏิบัติการ

1. โพรเซส : ส่วนปฏิบัติงานของโปรแกรมที่ได้โหลดลงในเมโมรีหลักและตีความโดยหน่วยประมวลผลกลาง โดยมากในระบบหนึ่งมักมีงานหรือโพรเซสแอ็กทีฟอยู่มากกว่าหนึ่งโพรเซส
2. ระบบแฟ้มข้อมูล : แนวทางการจัดเก็บข้อมูลลงดิสก์หรือระบบจัดเก็บใดๆ ที่คงสภาพได้โดยไม่ต้องใช้กระแสไฟเลี้ยง
3. การเรียกใช้งานบริการของระบบ : ส่วนการให้บริการของระบบปฏิบัติการแก่โพรเซสต่างๆ เพื่ออำนวยความสะดวกในการปฏิบัติงานและจัดสรรทรัพยากรให้เกิดประโยชน์สูงสุด
4. เชลล์ : ส่วนโต้ตอบผู้ใช้งานจริง เพื่อควบคุมระบบคอมพิวเตอร์ได้อย่างมีประสิทธิภาพและเรียบง่ายที่สุด

ประเภทโครงสร้างภายในของระบบปฏิบัติการ

1. ระบบแบบโมโนลิทิก : ส่วนระบบปฏิบัติการทุกอย่างรวมอยู่ในแอดเดรสสเปซเดียวกันหมด การเรียกใช้ระบบภายในคือการกระโดดไปยังแอดเดรสของรูทีนนั้นๆ เช่น ดอส เป็นต้น
2. ระบบแบบเลเยอร์ : ส่วนระบบปฏิบัติการแบ่งออกเป็นชั้นๆ โดยแต่ละชั้นส่งงานผ่านต่อเป็นทอด (แบ่งตามแนวตั้ง)
3. ระบบแบบเวอร์ชวลแมชีน : มีระบบปฏิบัติการทำงานพร้อมกันอยู่บนเครื่องคอมพิวเตอร์มากกว่าหนึ่งระบบ มักใช้ในระบบปฏิบัติการของเมนเฟรม เพื่อความทนทานต่อข้อผิดพลาด เช่น OS/390 เป็นต้น

4. ระบบแบบไคลเอ็นต์/เซิร์ฟเวอร์ : ส่วนระบบปฏิบัติการแบ่งออกเป็นรูทีนย่อยที่อยู่แยกจากกัน (แบ่งตามแนวนอน) การเรียกใช้ระบบภายในคือการเรียกรูทีนย่อยพร้อมผ่านค่าพารามิเตอร์ประกอบ แล้วรอผลการทำงานคืนกลับ

แนะนำระบบปฏิบัติการยูนิกซ์เบื้องต้นและคำสั่งพื้นฐาน

"The mind of the beginner is empty, free of the habits
of the expert, ready to accept, to doubt, and open to
all the possibilities."
– Shunryu Suzuki

ยูนิกซ์เป็นระบบปฏิบัติการเก่าแก่ที่ได้รับความนิยมมายาวนาน ด้วยการยอมรับแนวคิดที่ครบถ้วนทั้งศาสตร์และศิลป์ทำให้ยูนิกซ์เป็นระบบปฏิบัติการต้นแบบของระบบปฏิบัติการปัจจุบัน แม้ว่าดูเหมือนว่าเป็นของเก่าล้าสมัยและน่าจะใช้งานยาก แต่เมื่อได้ใช้งานอย่างจริงจังแล้ว ผู้ใช้แทบทุกคนต่างพึงพอใจ เนื่องจากยูนิกซ์ตอบสนองความต้องการต่างๆ ได้เป็นอย่างดี โดยเฉพาะความเรียบง่ายแต่แฝงไว้ซึ่งพลังในการก่อกำเนิดผลงาน

<u>การเชื่อมต่อและเข้าสู่ระบบ</u> (Connecting and Logging in)

- การเชื่อมต่อจากระยะไกล : ถ้าคุณใช้คอมพิวเตอร์บนอินเทอร์เน็ตไม่ว่าจากบ้าน มหาวิทยาลัย หรือบริษัท โดยปรกติแล้วมักใช้โปรแกรม telnet เพื่อเชื่อมต่อเข้ากับโฮสยูนิกซ์ หรืออาจใช้โปรแกรม rlogin หรือ ssh (Secure shell) เช่น โปรแกรม SecureCRT และ F-Secure SSH สำหรับวินโดวส์ เป็นต้นก็ได้ แต่ควรใช้ Secure telnet หรือ ssh มากกว่า เพราะการรับส่งข้อความได้รับการเข้ารหัส ต่างจาก telnet และ rlogin ธรรมดาที่การรับส่งข้อความไม่ได้รับการเข้ารหัส อันอาจทำให้มีผู้อื่นดักจับข้อความระหว่างทางได้ง่าย
- การเข้าสู่ระบบ : เมื่อเชื่อมต่อระบบได้แล้วไม่ว่าโดยวิธีใดก็ตาม แรกสุดระบบยูนิกซ์ต้องถามล็อกอินเนม โดยมากเป็นตัวอักขระภาษาอังกฤษพิมพ์เล็กและหรือตัวเลข หลังจากนั้นเป็นการถามพาสเวิร์ดเพื่อยืนยันตัวตนและสิทธิ์ที่มีในระบบ ซึ่งไม่แสดงสิ่งที่ป้อนบนหน้าจอเพื่อป้องกันผู้อื่นเห็นบนหน้าจอ พาสเวิร์ดที่ดีตัวอักขระควรยาวอย่างน้อย 6 ตัวอักษร ใช้ตัวพิมพ์ใหญ่ พิมพ์เล็ก ตัวเลขผสมกัน และยากต่อการคาดเดา พาสเวิร์ดที่ดีนับเป็นปราการด่านแรกอันแข็งแกร่งสำหรับป้องกันผู้แอบอ้างหรือผู้บุกรุก ข้อสำคัญคืออย่าจดบันทึก อย่าเผยแพร่ และอย่าลืมพาสเวิร์ด หากมีปัญหาเรื่องล็อกอินเนมหรือพาสเวิร์ด ควรปรึกษากับผู้ดูแลระบบหรือฝ่ายช่วยเหลือโดยตรงเท่านั้น เมื่อเข้าสู่ระบบเสร็จสมบูรณ์ ผู้ใช้มักพบกับข่าวประกาศและกฎการใช้งานจากผู้ดูแลระบบ จากนั้นระบบยูนิกซ์ก็พร้อมรับคำสั่งโดยขึ้นต้นบรรทัดด้วยเครื่องหมาย $ หรือ % (ถ้าใช้ซีเชลล์) ที่เรียกว่าเชลล์พรอมต์
- การออกจากระบบ : อย่าปิดเครื่อง ปล่อยทิ้งไว้ หรือ ตัดสายโทรศัพท์ที่ใช้ระบบผ่านโมเด็ม โดยยังไม่ได้สั่งออกจากระบบหรือที่เรียกว่า ล็อกเอาต์ เป็นสิ่งจำเป็นที่ยูสเซอร์ต้องไม่ลืม ออกคำสั่งง่ายๆ คือ exit หรือ logout หรือกดปุ่ม ^D ขึ้นกับระบบที่ใช้

<u>ความรู้พื้นฐานที่จำเป็น</u> (Essential basis)

ปุ่มควบคุม : เมื่อพิมพ์ผิดและต้องการแก้ไข ก่อนที่กดปุ่ม Enter/Return ทั้งนี้หน้าที่ของปุ่มต่างๆ ขึ้นกับการตั้งค่าเทอร์มินอลของระบบนั้นด้วย ปรับแต่งได้ด้วยโปรแกรม stty มีดังนี้

- กดปุ่ม Delete หรือ BackSpace เพื่อลบตัวอักษร (erase)
- กดปุ่ม ^W เพื่อลบทั้งคำ (werase)

- กดปุ่ม ^U เพื่อลบทั้งบรรทัด (kill)

ปุ่มพิเศษควบคุมการทำงานโปรแกรม มีดังนี้

^D ออกจากระบบหรือสิ้นสุดแฟ้มข้อมูลที่รับจากคีย์บอร์ด (eof)

^S หยุดการแสดงผลหน้าจอชั่วคราว (stop) มักใช้ในกรณีแสดงผลออกหน้าจอหลายบรรทัดจนดูไม่ทัน

^Q ยกเลิกการหยุดการแสดงผลชั่วคราวที่สั่งจากปุ่ม ^S (start)

^C ขัดจังหวะและยุติการทำงานของโพรเซสที่ทำงานอยู่ปัจจุบัน (intr)

^Z ทำให้โพรเซสที่ทำงานอยู่ปัจจุบันทำงานแบบแบ็กกราวด์ (susp)

อักขระที่มีความหมายพิเศษ : โดยปกติเชลล์จะประมวลผลอักขระพิเศษเหล่านี้ก่อนส่งต่อการทำงานให้แก่โปรแกรมที่เรียกใช้ มีดังนี้

< อ่านจากแฟ้มข้อมูลที่อยู่ทางขวาของสัญลักษณ์นี้แทนการอ่านจากอินพุตมาตรฐาน

> เขียนทับลงบนแฟ้มข้อมูลที่อยู่ทางขวาของสัญลักษณ์นี้แทนการเขียนไปที่เอาต์พุตมาตรฐาน

>> เขียนต่อท้ายแฟ้มข้อมูลที่อยู่ทางขวาของสัญลักษณ์นี้แทนการเขียนไปที่เอาต์พุตมาตรฐาน

| สร้างไปป์ คือการถ่ายทอดเอาต์พุตจากโปรแกรมด้านซ้ายไปเป็นอินพุตของโปรแกรมด้านขวา

* เป็นอักขระเมตา ใช้แทนตัวอักขระใดๆ ยาวตั้งแต่ศูนย์ตัวขึ้นไป เช่น หากอ้าง A* อาจเป็น A1, AA, AbC, ARGUE, African ก็ได้ เป็นต้น

? เป็นอักขระเมตา ใช้แทนตัวอักษรใดๆ ยาวเพียงตัวเดียว เช่น A? อาจเป็น A1, AA ก็ได้ เป็นต้น

\ เป็นอักขระเมตา เพื่อบ่งบอกว่าตัวอักษรที่ตามมานั้นไม่มีการตีความเป็นอักขระความหมายพิเศษ และเสมือนเป็นตัวอักขระปกติตัวหนึ่ง เช่น * ทำให้ * เป็นตัวอักษรไม่ใช่อักขระเมตา หากต้องการอักขระ \ ก็พิมพ์เป็น \\ เป็นต้น

ระบบความช่วยเหลือประกอบไปด้วยคู่มือการใช้คำสั่งที่ถือได้ว่าครบถ้วนและละเอียด เรียกอ่านได้โดยใช้คำสั่ง man เช่น ต้องการอ่านคู่มือของคำสั่ง rm (remove)

```
$ man rm
```

คู่มือแสดงบนหน้าจอแล้วหยุดเป็นหน้าๆ กดปุ่มเว้นวรรคเพื่อไปหน้าต่อไป กดปุ่ม q เพื่อสิ้นสุดการอ่านคู่มือ บางระบบอาจกดปุ่ม ^B และ ^F เพื่อเลื่อนหน้ากลับไปยังหน้าก่อนหน้าและหน้าถัดไปตามลำดับ

<u>ระบบแฟ้มข้อมูลและไดเรกทอรี</u> (File and directory system)

หลักการทั่วไปของระบบแฟ้มข้อมูลและไดเรกทอรีของยูนิกซ์เป็นเช่นเดียวกับของดอสและวินโดวส์ เนื่องจากทั้งดอสและวินโดวส์ได้นำหลักการของยูนิกซ์ไปประยุกต์ แต่ได้ตัดทอนเรื่องสิทธิ์ออกไปและมีอักษรไดรฟ์ขึ้นมาประกอบ ในระบบแฟ้มข้อมูลและไดเรกทอรีของยูนิกซ์มีเรื่องสิทธิ์และต้องมีเจ้าของเสมอ และไม่มีการคำนึงถึงอักษรไดรฟ์เลย ดิสก์ทุกตัวรวมเข้ากับระบบไดเรกทอรีโดยเริ่มจาก / หรือรูตเสมอ ที่เป็นจุดเด่นอีกอย่างของระบบยูนิกซ์คือ ฮาร์ดแวร์ต่อพ่วงทุกอย่าง ระบบยูนิกซ์มองเป็น'แฟ้มข้อมูล'เสมอ เช่น /dev/tty /dev/mouse และ /dev/hda เป็นต้น

ชื่อแฟ้มข้อมูลหรือชื่อไดเรกทอรีบนยูนิกซ์มีความยาวได้มากถึง 255 ตัวอักษร (บางระบบอาจมากกว่านี้) อีกทั้งสามารถใช้ ตัวเลข ตัวเว้นวรรค อักขระพิเศษในการตั้งชื่อได้ ซึ่งมีข้อจำกัดน้อยกว่าดอสและวินโดวส์มาก แต่เพราะเหตุนี้

เช่นกันทำให้การอ้างถึงชื่อแฟ้มข้อมูลหรือชื่อไดเรกทอรีบนยูนิกซ์มีความซับซ้อนยิ่งขึ้น โดยเฉพาะการตั้งชื่อที่ประกอบด้วยอักขระเมตา

คำสั่งเบื้องต้นที่เกี่ยวข้อง คือ

ls : แสดงรายชื่อแฟ้มข้อมูลและไดเรกทอรีย่อยที่มี สั่งเพียงเท่านี้จะไม่แสดงแฟ้มข้อมูลที่มีชื่อขึ้นต้นด้วยเครื่องหมายจุด " . " ซึ่งมักเป็นแฟ้มข้อมูลที่ต้องการซ่อนไว้ สามารถระบุอักขระเมตาเพื่อค้นหาชื่อที่ต้องการและออปชันเพิ่มเติมได้ เช่น

ls -a แสดงรายชื่อทั้งหมด แม้ว่าจะซ่อนไว้ก็ตาม

ls –l แสดงรายชื่อแบบละเอียด ประกอบด้วยสิทธิ์ จำนวนบล็อกขนาด 512 ไบต์ที่ใช้จัดเก็บ ชื่อเจ้าของ ชื่อกลุ่มเจ้าของ ขนาดแฟ้มข้อมูล วันเวลาที่ได้รับการแก้ไขล่าสุด และชื่อ

ls –d แสดงรายชื่อเฉพาะไดเรกทอรี ไม่แสดงรายชื่อแฟ้มข้อมูลที่อยู่ในไดเรกทอรีนั้นๆ

ls –t แสดงรายชื่อเรียงตามลำดับวันเวลาที่ได้รับการแก้ไขล่าสุด

ออปชันต่างๆ ใช้ผสมกันได้ การใช้งานนั้นคล้ายคำสั่ง "DIR" ของดอสมาก ตัวอย่างผลลัพธ์เช่น

```
$ ls
Mail    News    mail    mycode  mydoc   tmp     www
$ ls -alF
total 70
drwx--x--x  10 adek       lect          2048 Apr 17 18:53 ./
drwxr-xr-x  32 root       sys           1024 Nov  3 10:50 ../
-rw-------   1 adek       lect           107 Jul 21  1998 .Xauthority
-rw-------   1 adek       lect           731 Mar 13 15:41 .addressbook
-rw-------   1 adek       lect          2399 Mar 13 15:41 .addressbook.lu
drwx------   2 adek       lect          1024 Mar 30 11:12 .pgp/
-rw-------   1 adek       lect         14097 Apr  3 20:59 .pinerc
-rw-------   1 adek       lect           595 Sep 14  1999 .profile
-rw-------   1 adek       lect          2712 Apr 17 18:59 .sh_history
-rw-------   1 adek       lect           113 Apr  5 09:27 .signature
drwx------   2 adek       lect            24 Aug 28  1998 Mail/
drwx------   2 adek       lect            24 Aug 28  1998 News/
drwx------   2 adek       lect          1024 Apr 17 15:02 mail/
drwx------   3 adek       lect          1024 Apr  3 20:58 mycode/
drwx------   2 adek       lect          1024 Oct 24  1998 mydoc/
drwx------   2 adek       lect          1024 Apr  9 21:49 tmp/
drwx--x--x   6 adek       lect          1024 Apr 17 18:48 www/
```

cp : คัดลอกแฟ้มข้อมูล การใช้งานนั้นคล้ายคำสั่ง "COPY" ของดอสมาก

mv : เปลี่ยนชื่อหรือโอนย้ายแฟ้มข้อมูล การใช้งานนั้นคล้ายคำสั่ง "MOVE" และ "RENAME" ของดอสมาก

rm : ลบแฟ้มข้อมูล การใช้งานนั้นคล้ายคำสั่ง "DELETE" ของดอสมาก เป็นคำสั่งที่อันตรายเนื่องจากในระบบยูนิกซ์มักไม่สามารถกู้คืนแฟ้มที่ลบไปแล้วได้

cat : แสดงรายละเอียดในแฟ้มข้อมูลออกเอาต์พุตมาตรฐาน

pwd : แสดงพาธของไดเรกทอรีปัจจุบัน

cd : ย้ายไดเรกทอรีปัจจุบันไปเป็นตามที่ระบุ หากไม่ระบุจะกลับไปยังพาธที่กำหนดในตัวแปร $HOME

mkdir : สร้างไดเรกทอรีว่างตามที่ระบุ

rmdir : ลบไดเรกทอรีว่างตามที่ระบุ

การจัดการแฟ้มข้อมูลและสิทธิ์ (File and permission management)

แฟ้มข้อมูลและไดเรกทอรีต้องมีเจ้าของ กลุ่มเจ้าของ และสิทธิ์กำกับเสมอ ดังตัวอย่างต่อไปนี้

```
drwx--x--x  10 adek       lect          2048 Apr 17 18:53 ./
drwxr-xr-x  32 root       sys           1024 Nov  3 10:50 ../
-rw-------   1 adek       lect           595 Sep 14  1999 .profile
-rw-------   1 adek       lect          2712 Apr 17 18:59 .sh_history
drwx------   2 adek       lect            24 Aug 28  1998 Mail/
drwx------   2 adek       lect            24 Aug 28  1998 News/
-rwxr-xr-x   2 adek       lect          1024 Apr 17 15:02 test.txt
```

คอลัมน์ที่ 1 เป็นแฟล็กระบุสิทธิ์และอื่นๆ

- อักขระแรกเป็นการระบุชนิด ถ้าเป็น d คือไดเรกทอรี l คือแฟ้มแบบซอฟต์ลิงก์ c คือแฟ้มดีไวซ์แบบคาแรกเตอร์ และ b คือแฟ้มดีไวซ์แบบบล็อก หากไม่ระบุคือเป็นขีดถือเป็นแฟ้มข้อมูลธรรมดา แต่ไม่สามารถบ่งบอกได้ว่าเป็นแฟ้มเท็กซ์หรือแฟ้มไบนารี
- อักขระสามตัวกลุ่มแรกเป็นการระบุสิทธิ์ของเจ้าของ (ตามที่ระบุไว้ในคอลัมน์ที่ 3) ถ้าเป็น r คืออ่านได้ w คือเขียนได้ และ x คือสั่งรันแฟ้มนั้นได้ (อาจเป็นแฟ้มโค้ดไบนารีหรือแฟ้มเท็กซ์เชลล์สคริปต์ก็ได้) หากไม่ระบุคือเป็นขีดถือว่าไม่มีสิทธิ์ในการกระทำนั้นๆ เช่น rwx คือมีสิทธิ์อ่าน เขียน และสั่งรัน เป็นต้น
- อักขระสามตัวกลุ่มกลางเป็นการระบุสิทธิ์ของกลุ่มเจ้าของ (ตามที่ระบุไว้ในคอลัมน์ที่ 4) เช่น r-x คือมีสิทธิ์อ่านและสั่งรันเท่านั้น เขียนแฟ้มนั้นไม่ได้ เป็นต้น
- อักขระสามตัวกลุ่มสุดท้ายเป็นการระบุสิทธิ์ของผู้อื่น (other) ที่ไม่ใช่เจ้าของและไม่อยู่ในกลุ่มเจ้าของ (ตามที่ระบุไว้ในคอลัมน์ที่ 3 และ 4) เช่น --- คือไม่มีสิทธิ์ใดๆ เลย

คอลัมน์ที่ 3 เป็นชื่อเจ้าของ (owner id)

คอลัมน์ที่ 4 เป็นชื่อกลุ่มเจ้าของ (group id)

คำสั่งเบื้องต้นที่เกี่ยวข้อง คือ

chmod : เปลี่ยนสิทธิ์ประจำแฟ้มข้อมูลหรือไดเรกทอรี

chown : เปลี่ยนเจ้าของประจำแฟ้มข้อมูลหรือไดเรกทอรี

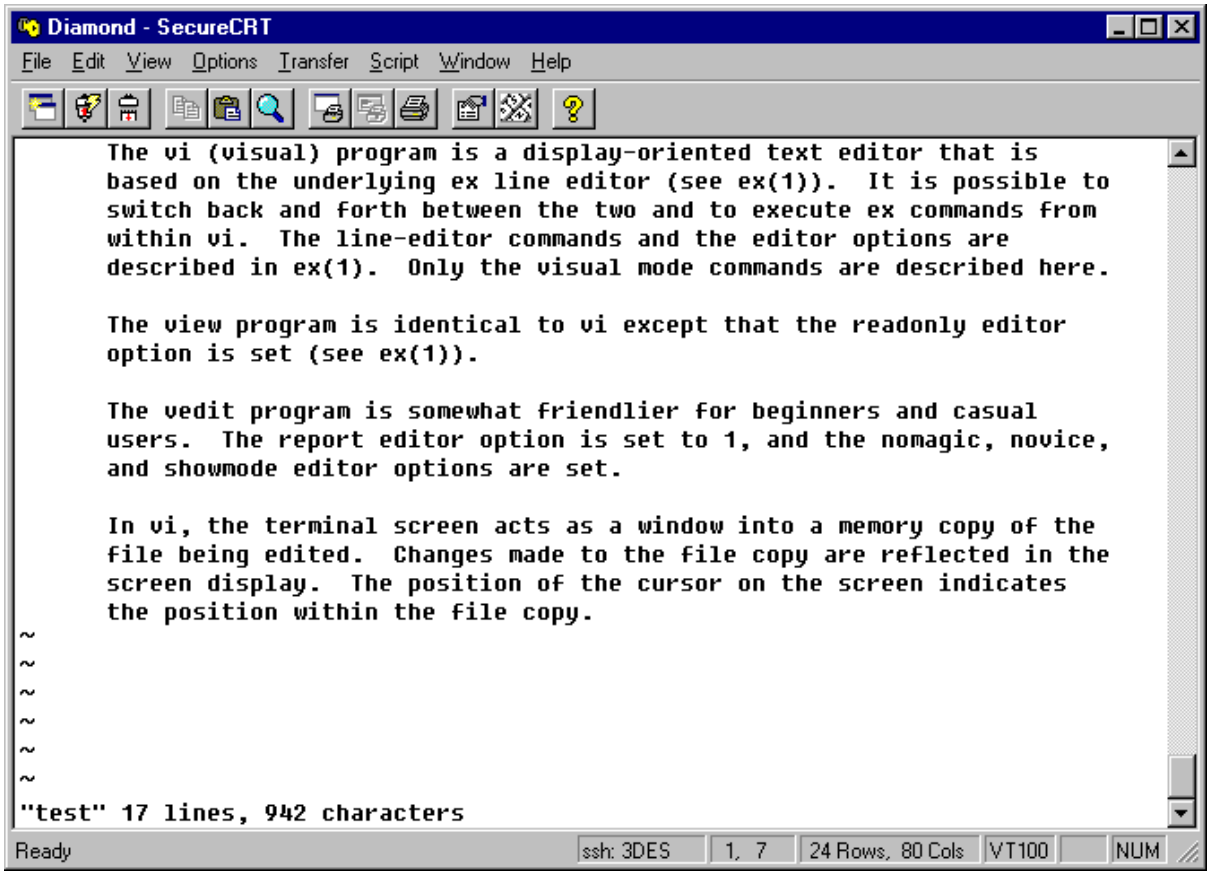
chgrp : เปลี่ยนกลุ่มเจ้าของประจำแฟ้มข้อมูลหรือไดเรกทอรี

ตัวอย่างเช่น

```
$ ls -l test.txt
-rw-rwxrwx   1 adek       lect           942 Apr  9 21:15 test.txt
$ chmod go-rwx test.txt
$ chown root test.txt
$ ls -l test.txt
-rw-------   1 root       lect           942 Apr  9 21:15 test.txt
```

การใช้งานเอดิเตอร์ vi

vi เป็นเอดิเตอร์แบบสกรีนที่ใช้แพร่หลายที่สุด เนื่องจากมีให้ใช้บนยูนิกซ์ทุกรุ่นทุกยี่ห้อ และมีความคล่องตัวในการทำงานสูง แต่สำหรับผู้ใช้ทั่วไปและผู้ใช้มือใหม่แล้วมักหลีกเลี่ยงการใช้งาน vi เนื่องจากไม่เข้าใจการใช้งานที่ค่อนข้างสลับซับซ้อน ทว่าถ้าเข้าใจกลไกการทำงานของ vi แล้วจะพบว่าเป็นเอดิเตอร์แบบสกรีนที่น่าใช้และมีความคล่องตัวสูงมาก



หน้าจอโปรแกรม vi

โดยทั่วไปแล้วพื้นที่เอกสารใน vi เริ่มตั้งแต่บรรทัดแรกเป็นต้นไปถึงก่อนบรรทัดสุดท้ายซึ่งเป็นส่วนแสดงผลและโต้ตอบกับผู้ใช้ของ vi เช่น หากขนาดเทอร์มินอลเป็น 80x24 บรรทัดที่ 24 เป็นส่วนแสดงผลและโต้ตอบกับผู้ใช้ เป็นต้น

โหมดการทำงานของ vi แบ่งเป็นสองโหมด ซึ่งในการใช้งาน vi ทั่วไป ผู้ใช้ต้องสลับไปมาระหว่างสองโหมดนี้บ่อยครั้ง คือ

1. โหมดคำสั่ง (Command Mode) เป็นโหมดที่ผู้ใช้สามารถออกคำสั่งให้ vi ทำงานได้ตามความต้องการ
2. โหมดแก้ไขข้อความ (Editing Mode) เป็นโหมดที่ผู้ใช้ป้อนหรือแก้ไขข้อความในเอกสาร

เมื่อเรียกใช้ vi จะเริ่มต้นที่โหมดคำสั่งเสมอ การสลับจากโหมดคำสั่งเป็นโหมดแก้ไขข้อความทำได้โดยออกคำสั่งแก้ไขข้อความ เช่น i หรือ a เป็นต้น การสลับจากโหมดแก้ไขข้อความเป็นโหมดคำสั่งทำได้โดยกดปุ่ม ESC หรือ ^[(ข้อแนะนำ หากไม่มั่นใจว่า vi อยู่ในโหมดคำสั่งหรือไม่สามารถกด ESC ซ้ำได้ ถ้าอยู่ในโหมดคำสั่งอยู่แล้วจะมีเสียงบี๊บ)

ในส่วนของโหมดคำสั่งยังแบ่งย่อยเป็นสองโหมด เนื่องจากอันที่จริงแล้ว vi เป็นส่วนเสริมจากเอดิเตอร์แบบบรรทัด ex คือ โหมดคำสั่ง vi และโหมดคำสั่ง ex ซึ่งเป็นเอดิเตอร์แบบบรรทัดซึ่งมักขึ้นต้นด้วยเครื่องหมายทวิภาค “ : “

การเรียกใช้ ทำได้โดย

- เรียกโปรแกรม vi โดยตรงจากเชลล์พรอมต์

```
$ vi
```

- หากต้องการแก้ไขแฟ้มข้อมูลเท็กซ์ที่มีอยู่แล้วก็เรียกโปรแกรม vi แล้วตามด้วยชื่อแฟ้มนั้นๆ

```
$ vi filename.txt
```

- หากต้องการสร้างแฟ้มข้อมูลเท็กซ์ใหม่ เรียกโปรแกรม vi แล้วตามด้วยชื่อแฟ้มใหม่นั้นๆ

```
$ vi newfilename.txt
```

คำสั่งพื้นฐานในโหมดคำสั่ง vi

ออกจากโปรแกรม

ZZ ออกจากโปรแกรมและจัดเก็บแฟ้มหากมีการแก้ไข

Q สลับไปยังโหมดคำสั่ง ex (กลับไปยังโหมดคำสั่ง vi โดยพิมพ์ " vi ")

: ออกคำสั่งในโหมดคำสั่ง ex เพียงคำสั่งเดียวแล้วกลับไปยังโหมดคำสั่ง vi โดยอัตโนมัติ

ในโหมดคำสั่ง vi นี้รูปแบบพื้นฐานคือ [count] command [where]

คำประกอบ where พื้นฐาน

w ไปจนสิ้นสุดคำ

^ ไปจนต้นบรรทัด

$ ไปจนท้ายบรรทัด

d ทั้งบรรทัด

ตัวอย่างเช่น

d^ ลบอักขระตั้งแต่เคอร์เซอร์ไปจนต้นบรรทัด

d$ ลบอักขระตั้งแต่เคอร์เซอร์ไปจนท้ายบรรทัด

dw ลบอักขระตั้งแต่เคอร์เซอร์ไปจนสิ้นสุดคำ

3dd ลบบรรทัดทั้งบรรทัดไปสามครั้ง (ลบบรรทัดปัจจุบันและอีกสองบรรทัดถัดลงไป)

การเพิ่มข้อความใหม่

A เข้าสู่โหมดแก้ไขข้อความแบบต่อท้ายบรรทัด

a เข้าสู่โหมดแก้ไขข้อความแบบต่อท้ายเคอร์เซอร์

I เข้าสู่โหมดแก้ไขข้อความแบบแทรกต้นบรรทัด (ตัวไอใหญ่)

i เข้าสู่โหมดแก้ไขข้อความแบบแทรกที่เคอร์เซอร์

O เข้าสู่โหมดแก้ไขข้อความแบบแทรกบรรทัดก่อนบรรทัดปัจจุบัน (ตัวโอใหญ่)

o เข้าสู่โหมดแก้ไขข้อความแบบต่อบรรทัดจากบรรทัดปัจจุบัน (ตัวโอเล็ก)

การเลื่อนเคอร์เซอร์

h เคลื่อนเคอร์เซอร์ไปทางซ้ายมือ หรืออาจใช้ปุ่มลูกศรซ้ายก็ได้

l เคลื่อนเคอร์เซอร์ไปทางขาวมือ หรืออาจใช้ปุ่มลูกศรขวาก็ได้ (ตัวแอลเล็ก)

k เคลื่อนเคอร์เซอร์ไปบรรทัดบน หรืออาจใช้ปุ่มลูกศรชี้บนก็ได้

j เคลื่อนเคอร์เซอร์ไปบรรทัดล่าง หรืออาจใช้ปุ่มลูกศรชี้ล่างก็ได้

$ เคลื่อนเคอร์เซอร์ไปยังท้ายบรรทัด

^ เคลื่อนเคอร์เซอร์ไปยังต้นบรรทัดที่ไม่ใช่อักขระเว้นวรรค

0 เคลื่อนเคอร์เซอร์ไปยังคอลัมน์แรกของบรรทัด (เลขศูนย์)

B ถอยหลังหนึ่งคำ โดยข้ามเครื่องหมายขั้นวรรค

b ถอยหลังหนึ่งคำ โดยหากเคอร์เซอร์อยู่ระหว่างคำจะไปยังต้นคำคำนั้น

W เคลื่อนไปยังต้นคำ โดยข้ามเครื่องหมายขั้นวรรค

w เคลื่อนไปยังคำถัดไป

E เคลื่อนไปยังท้ายคำ โดยข้ามเครื่องหมายขั้นวรรค

e เคลื่อนไปยังท้ายคำ โดยหากเคอร์เซอร์อยู่ระหว่างคำจะไปยังท้ายคำคำนั้น

G เคลื่อนไปยังบรรทัดที่กำหนดโดย count หากไม่ระบุจะไปยังบรรทัดสุดท้ายของแฟ้มข้อมูล

^B เปลี่ยนไปยังหน้าก่อนหน้า

^U เปลี่ยนไปยังครึ่งหน้าก่อนหน้า

^F เปลี่ยนไปยังหน้าถัดไป

^D เปลี่ยนไปยังครึ่งหน้าถัดไป

^P เคลื่อนเคอร์เซอร์ไปบรรทัดบนในคอลัมน์เดียวกัน

^N เคลื่อนเคอร์เซอร์ไปบรรทัดล่างในคอลัมน์เดียวกัน

การเปลี่ยนข้อความ

C เปลี่ยนข้อความตั้งแต่เคอร์เซอร์ไปจนสุดบรรทัด

R เปลี่ยนอักขระไปเรื่อยๆ จนกว่ากดปุ่ม ESC

r เปลี่ยนอักขระหนึ่งตัวที่เคอร์เซอร์

J เชื่อมบรรทัดปัจจุบันกับบรรทัดล่างให้เป็นบรรทัดเดียวกัน

การลบ/ตัด คัดลอก และแปะข้อความ

D ลบ/ตัดข้อความตั้งแต่เคอร์เซอร์ไปจนสุดบรรทัด

d ลบ/ตัดข้อความ (ต้องระบุขอบเขตประกอบด้วย) หากสั่ง “ dd “ คือลบ/ตัดบรรทัดปัจจุบันทั้งบรรทัด

Y คัดลอกบรรทัดปัจจุบันทั้งบรรทัด

y คัดลอกข้อความ (ต้องระบุขอบเขตประกอบด้วย) หากสั่ง “ yy “ คือคัดลอกบรรทัดปัจจุบันทั้งบรรทัด

X ลบอักขระที่ก่อนหน้าเคอร์เซอร์

x ลบอักขระที่เคอร์เซอร์

P แปะข้อความที่ได้ตัดหรือคัดลอกล่าสุดไว้ก่อนหน้าบรรทัดปัจจุบัน

p แปะข้อความที่ได้ตัดหรือคัดลอกล่าสุดไว้ต่อจากบรรทัดปัจจุบัน

การค้นหา

/ ค้นหาคำจากเคอร์เซอร์ลงไปจนถึงบรรทัดท้ายแฟ้มข้อมูล

? ค้นหาคำจากเคอร์เซอร์ขึ้นไปจนถึงบรรทัดแรกของแฟ้มข้อมูล

n ค้นหาต่อไป ในทิศทางคงเดิม

N ค้นหาต่อไป ในทิศทางที่ตรงข้ามจากทิศทางเดิม

คำสั่งอื่นๆ

^G แสดงตำแหน่งเคอร์เซอร์ปัจจุบันและสถานะของแฟ้มข้อมูล

^L วาดหน้าจอเทอร์มินอลใหม่ มักใช้ในกรณีการแสดงผลผิดเพี้ยนหรือแสดงอักขระขยะขึ้นมา

! เรียกเชลล์เพื่อปฏิบัติงาน

u ยกเลิกการกระทำก่อนหน้า (undo) หากสั่ง u อีกครั้งติดกันหมายถึงไม่ยกเลิกการกระทำนั้น (redo)

<u>คำสั่งพื้นฐานในโหมดคำสั่ง ex</u>

:q ออกจากโปรแกรม vi หากแฟ้มข้อมูลได้รับการแก้ไขแต่ยังไม่ได้จัดเก็บ vi จะแจ้งเตือนและยังไม่ให้ออกจากโปรแกรม

:q! ออกจากโปรแกรม vi โดยไม่สนใจจัดเก็บแฟ้มข้อมูลที่อาจได้รับการแก้ไข

:w จัดเก็บแฟ้มข้อมูลทับชื่อเดิม

:w file จัดเก็บแฟ้มข้อมูลในชื่อใหม่ เช่น :w file-a.txt คือจัดเก็บลงแฟ้มชื่อ file-a.txt เป็นต้น

:wq จัดเก็บแฟ้มข้อมูลทับชื่อเดิม แล้วออกจากโปรแกรม

:x จัดเก็บแฟ้มข้อมูลทับชื่อเดิม แล้วออกจากโปรแกรม

<u>ข้อสังเกต</u>

- หากเป็นการสร้างแฟ้มใหม่หรือแก้ไขแฟ้มข้อมูล บรรทัดหลังจากบรรทัดสุดท้ายของแฟ้มข้อมูล vi แสดงอักขระ ~ ขึ้นต้นเพื่อแสดงบรรทัด ”ว่าง” คือไม่มีบรรทัดจริง แสดงเพื่อมิให้ผู้ใช้กังวลใจเมื่อพบหน้าจอว่างเปล่า
- โปรดระวังเรื่องการจัดเก็บข้อมูลทับลงในแฟ้มข้อมูลโดยไม่ได้ตั้งใจ
- ผู้ดูแลระบบนิยมใช้ vi มากกว่า pico
- ในโหมดคำสั่ง vi ตัวพิมพ์เล็กและตัวพิมพ์ใหญ่ถือว่าแตกต่างกัน และหาก ^ นำหน้าอักขระใดถือว่าเป็นการกดปุ่ม Control ร่วมกับปุ่มอักขระนั้น เช่น ^B หมายถึงกดปุ่ม Ctrl แล้วกดปุ่ม B เว้นแต่ ^ ที่ตามหลังอักขระอื่นนั้นหมายถึงอักขระ “ ^ “ จริงๆ ทั้งนี้ต้องพิจารณาให้ดีเองด้วย
- ในโหมดคำสั่ง ex นิยมเขียน “ : “ นำหน้าไว้เพื่อแยกแยะว่าเป็นคำสั่งในโหมดนี้ เมื่อใช้งานจริงในโหมดคำสั่ง ex ไม่ต้องพิมพ์ เช่น “ :q! “ ก็พิมพ์ “ : “ เพื่อเข้าสู่โหมดคำสั่ง ex แล้วตามด้วย “ q! “ เป็นต้น
- หากต้องการพิมพ์อักขระหรือข้อความภาษาไทยซึ่งต้องใช้ ASCII แบบ 8 บิต vi สนับสนุนอยู่แล้ว แต่ต้องตั้งค่าให้เทอร์มินอลรองรับการใช้งาน ASCII แบบ 8 บิตด้วย (ปรกติรับแค่ 7 บิต) โดยสั่งยกเลิกการละเลยบิตที่แปด

```
$ stty -istrip
```

- อาจมีบางกรณีที่เมื่อเรียก vi แล้วโปรแกรมแจ้งเตือนว่า “Visual needs addressible cursor or upline capability” แล้วเข้าสู่โหมดคำสั่ง ex ทันที (ออกจาก vi นี้โดยคำสั่ง “ q! “) หมายถึงตัวแปรสภาพแวดล้อม

TERM ไม่ได้ตั้งค่า แก้ไขโดยตั้งค่าตัวแปรสภาพแวดล้อม TERM ให้เหมาะสมกับเทอร์มินอลนั้น เช่น สำหรับ sh ksh และ bash

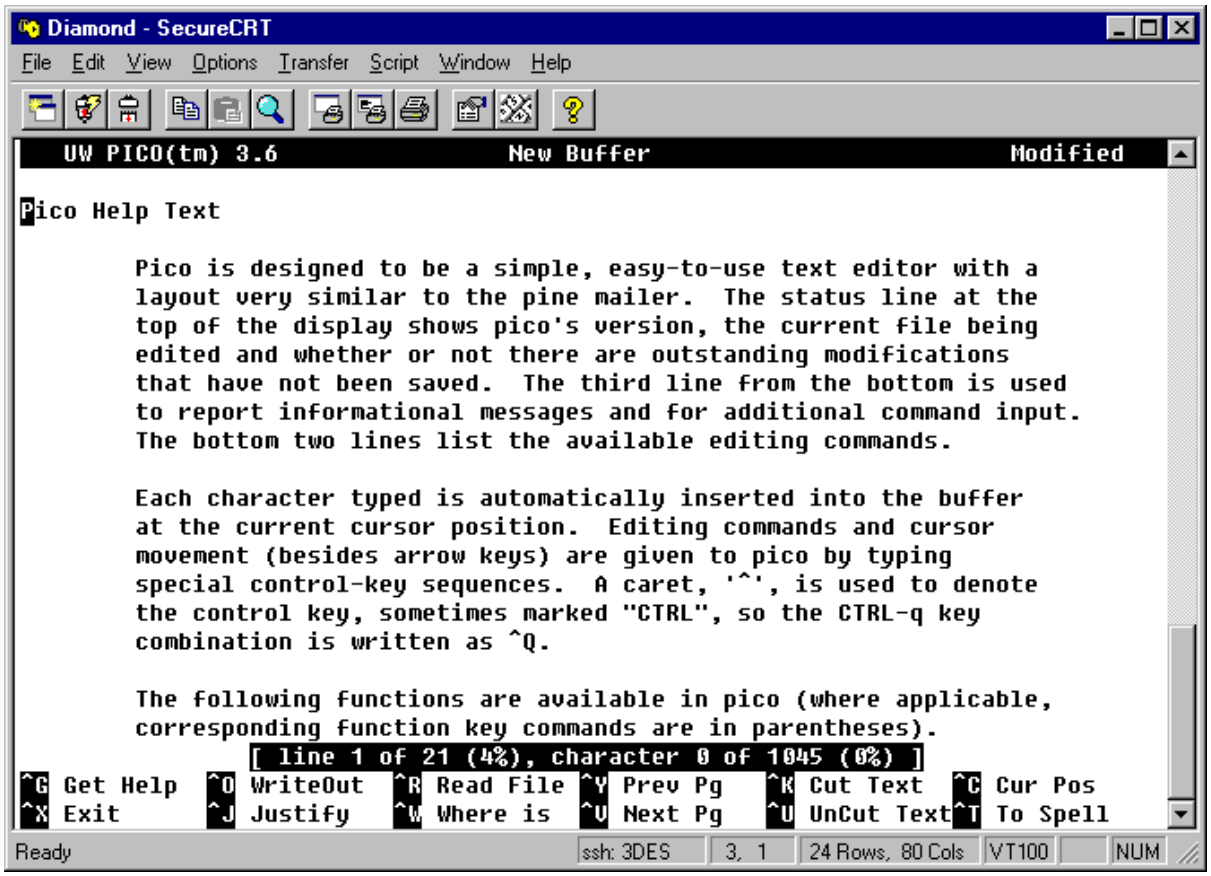
```
$ TERM=vt100; export TERM
```

หรือ สำหรับ csh

```
$ setenv TERM vt100
```

การใช้งานเอดิเตอร์ pico

pico เป็นเอดิเตอร์แบบสกรีนซึ่งมาพร้อมกับโปรแกรมรับส่งอีเมล์ pine จัดทำขึ้นโดย University of Washington แม้ว่าความสามารถจะเป็นรอง vi หรือเอดิเตอร์ตัวอื่นๆ แต่ด้วยความเรียบง่ายจึงทำให้เป็นที่นิยมสำหรับผู้ใช้มือใหม่และผู้ใช้ทั่วไป



หน้าจอโปรแกรม pico

โดยทั่วไปแล้วพื้นที่เอกสารใน pico เริ่มตั้งแต่บรรทัดที่ 3 เป็นต้นไปจนถึงก่อนสามบรรทัดสุดท้ายซึ่งเป็นส่วนแสดงผลและโต้ตอบกับผู้ใช้ของ pico เช่น หากขนาดเทอร์มินอลเป็น 80x24 บรรทัดแรกบ่งบอกเวอร์ชันของ pico ชื่อแฟ้มข้อมูลที่กำลังแก้ไขอยู่ และสถานะของบัฟเฟอร์เอกสารว่ามีการแก้ไขหรือไม่ ต่อมาเป็นขอบเขตของเอกสารเริ่มที่บรรทัดที่ 3 และสิ้นสุดที่บรรทัดที่ 21 ในบรรทัดที่ 23-24 เป็นส่วนแสดงปุ่มด่วนต่างๆ ที่สามารถใช้งานได้กับสถานการณ์ปัจจุบัน และบรรทัดที่ 22 เป็นส่วนแสดงผลและโต้ตอบกับผู้ใช้ เป็นต้น

<u>การเรียกใช้</u> ทำได้โดย

- เรียกโปรแกรม pico โดยตรงจากเชลล์พรอมต์

```
$ pico
```

- หากต้องการแก้ไขแฟ้มข้อมูลเท็กซ์ที่มีอยู่แล้วก็เรียกโปรแกรม pico แล้วตามด้วยชื่อแฟ้มนั้นๆ

```
$ pico filename.txt
```

- หากต้องการสร้างแฟ้มข้อมูลเท็กซ์ใหม่ เรียกโปรแกรม pico แล้วตามด้วยชื่อแฟ้มใหม่นั้นๆ

```
$ pico newfilename.txt
```

ปุ่มควบคุมสำหรับการเคลื่อนย้ายเคอร์เซอร์และหน้าเอกสาร

^B เคลื่อนเคอร์เซอร์ไปทางซ้ายมือ หรืออาจใช้ปุ่มลูกศรซ้ายก็ได้

^F เคลื่อนเคอร์เซอร์ไปทางขาวมือ หรืออาจใช้ปุ่มลูกศรขวาก็ได้

^P เคลื่อนเคอร์เซอร์ไปบรรทัดบน หรืออาจใช้ปุ่มลูกศรชี้บนก็ได้

^N เคลื่อนเคอร์เซอร์ไปบรรทัดล่าง หรืออาจใช้ปุ่มลูกศรชี้ล่างก็ได้

^A เคลื่อนเคอร์เซอร์ไปยังต้นบรรทัด

^E เคลื่อนเคอร์เซอร์ไปยังท้ายบรรทัด

^Y เปลี่ยนไปยังหน้าก่อนหน้า

^V เปลี่ยนไปยังหน้าถัดไป

ปุ่มควบคุมสำหรับการแก้ไขเอกสาร

^C แสดงตำแหน่งปัจจุบันและข้อมูลเอกสาร ณ ส่วนแสดงผลและโต้ตอบผู้ใช้

^D ลบอักขระที่เคอร์เซอร์ หรืออาจใช้ปุ่ม Delete ก็ได้

^H ลบอักขระด้านซ้ายเคอร์เซอร์ หรืออาจใช้ปุ่ม BackSpace ก็ได้

^I พิมพ์อักขระแทบ (tab) หรืออาจใช้ปุ่ม Tab ก็ได้

^J ปรับแต่งย่อหน้าเอกสารให้สวยงาม หากต้องการยกเลิกการปรับแต่งกดปุ่ม ^U

^K ลบ/ตัดบรรทัดที่เคอร์เซอร์

^^ ทำเครื่องหมายจุดเริ่มต้นเพื่อตัดเป็นบล็อกข้อความด้วย ^K

^U เพิ่มบรรทัดที่ตัดด้วย ^K ครั้งล่าสุด ณ เคอร์เซอร์

^M พิมพ์อักขระขึ้นบรรทัดใหม่ (carriage return) หรืออาจใช้ปุ่ม Enter/Return ก็ได้

^T ตรวจคำสะกดภาษาอังกฤษ

^W หาคำ

ปุ่มควบคุมสำหรับการอ่าน/เขียนแฟ้มข้อมูล

^R สั่งให้ pico อ่านแฟ้มข้อมูลเท็กซ์ขึ้นมาประกอบกับเอกสารปัจจุบัน โดยแทรกที่เคอร์เซอร์ ทั้งนี้อาจป้อนชื่อแฟ้มโดยตรงหรือกด ^T เพื่อบราวซ์รายชื่อแฟ้มข้อมูลก็ได้ สังเกตได้จากส่วนแสดงปุ่มด่วน

^O จัดเก็บแฟ้มข้อมูลทันที โดย pico จะถามชื่อแฟ้มข้อมูลและมักปรากฏชื่อโดยปริยายไว้ให้ ทั้งนี้อาจป้อนชื่อแฟ้มโดยตรงหรือกด ^T เพื่อบราวซ์รายชื่อแฟ้มข้อมูลก็ได้ สังเกตได้จากส่วนแสดงปุ่มด่วน

ปุ่มควบคุมอื่นๆ

^G เข้าสู่ระบบช่วยเหลือที่สอดคล้องกับสถานการณ์ปัจจุบัน

^X ออกจากส่วนช่วยเหลือ หรือ ออกจาก pico โดยหากผู้ใช้ได้มีการแก้ไขแฟ้มข้อมูลและยังไม่ได้จัดเก็บ pico จะถามย้ำการจัดเก็บ ตอบ Y ถ้าต้องการจัดเก็บ และตอบ N เมื่อไม่ต้องการจัดเก็บ

^L วาดหน้าจอเทอร์มินอลใหม่ มักใช้ในกรณีการแสดงผลผิดเพี้ยนหรือแสดงอักขระขยะขึ้นมา

ข้อสังเกต

- แม้ว่าในระบบยูนิกซ์ ตัวพิมพ์เล็กและตัวพิมพ์ใหญ่ถือว่าแตกต่างกัน แต่สำหรับกรณีการกดปุ่มควบคุมแล้วให้ผลเหมือนกันไม่ว่ากดแป้นเป็นตัวพิมพ์เล็กหรือตัวพิมพ์ใหญ่ เช่น ^A หรือ ^a ถือว่าเหมือนกัน เป็นต้น
- ปุ่มควบคุมเดียวกันอาจให้ผลต่างกันเมื่ออยู่ในสถานการณ์ที่ต่างกัน เช่น ^T ในส่วนการแก้ไข เป็นการสั่งตรวจคำสะกด แต่ ^T ในส่วนการอ่าน/เขียนแฟ้มข้อมูล เป็นการบราวซ์รายชื่อแฟ้มข้อมูล เป็นต้น ฉะนั้นจึงต้องใส่ใจในการใช้งานปุ่มควบคุมให้ดี
- โปรดระวังเรื่องการจัดเก็บข้อมูลทับลงในแฟ้มข้อมูลโดยไม่ได้ตั้งใจ
- ผู้ดูแลระบบไม่นิยมใช้ pico มักใช้ vi มากกว่า
- หากต้องการพิมพ์อักขระหรือข้อความภาษาไทยซึ่งต้องใช้ ASCII แบบ 8 บิต pico สนับสนุนอยู่แล้ว แต่ต้องตั้งค่าให้เทอร์มินอลรองรับการใช้งาน ASCII แบบ 8 บิตด้วย (ปรกติรับแค่ 7 บิต) โดยสั่งยกเลิกการละเลยบิตที่แปด

```
$ stty -istrip
```

- อาจมีบางกรณีที่เมื่อเรียก pico แล้วโปรแกรมแจ้งเตือนว่า “Environment variable TERM not defined!” หมายถึงตัวแปรสภาพแวดล้อม TERM ไม่ได้ตั้งค่า แก้ไขโดยตั้งค่าตัวแปรสภาพแวดล้อม TERM ให้เหมาะสมกับเทอร์มินอลนั้น เช่น สำหรับ sh ksh และ bash

```
$ TERM=vt100; export TERM
```

หรือ สำหรับ csh

```
$ setenv TERM vt100
```

สรุปคำสั่งที่จำเป็น

alias	นิยามคำสั่งแบบย่อ
ar	จัดการไลบรารีและดูแลแฟ้มข้อมูลซอร์สโค้ด
as	ตัวแปลภาษาแอสเซมบลี
awk	เทียบรูปแบบและประมวลผลถ้อยคำ
bash	บอร์นอะเกนเชลล์
bg	นำโพรเซสไปทำงานแบบแบ็กกราวด์
cat	นำข้อมูลในแฟ้มแสดงออกเอาต์พุตมาตรฐาน
cc	คอมไพเลอร์ภาษาซี
CC	คอมไพเลอร์ภาษาซีพลัสพลัส
cd	เปลี่ยนไดเรกทอรีปัจจุบัน
chmod	เปลี่ยนสิทธิ์แฟ้มข้อมูลหรือไดเรกทอรี
chsh	เปลี่ยนเชลล์เริ่มต้นเป็นตัวอื่น
clear	ล้างหน้าจอ
cmp	เปรียบเทียบแฟ้มข้อมูลสองแฟ้ม
compress	บีบอัดแฟ้มข้อมูลอย่างง่าย
cp	สำเนาแฟ้มข้อมูล
crontab	จัดตารางเวลาโปรแกรมให้ทำงานเป็นประจำโดยอัตโนมัติ
crypt	เข้ารหัสหรือถอดรหัสแฟ้มข้อมูลอย่างง่าย
csh	ซีเชลล์
cut	ตัดคอลัมน์จากแฟ้มข้อมูล
date	แสดงวันและเวลา ถ้าเป็นบัญชีผู้ใช้รูตสามารถตั้งวันเวลาใหม่ได้ด้วย
dbx	ดีบักเกอร์ระดับซอร์สโค้ด
df	สรุปเนื้อที่จัดเก็บแฟ้มที่เหลืออยู่
diff	เปรียบเทียบข้อมูลในแฟ้มเท็กซ์
du	สรุปเนื้อที่จัดเก็บที่ใช้ไปในไดเรกทอรีปัจจุบันหรือที่ระบุ
echo	แสดงสายอักษรไปยังเอาต์พุตมาตรฐาน
ed	เท็กซ์เอดิเตอร์แบบบรรทัด
eqn	กำหนดรูปแบบสูตรทางคณิตศาสตร์ใช้ในโปรแกรม troff
ex	เท็กซ์เอดิเตอร์แบบบรรทัด
exit	ออกจากเชลล์ หรือออกจากระบบในกรณีที่ทำงานบนเชลล์ตัวแรกสุด
export	นำค่าตัวแปรสภาพแวดล้อมสำหรับเชลล์นั้นให้กลายเป็นค่าตัวแปรสภาพแวดล้อมกลางของทุกเชลล์
f77	คอมไพเลอร์ภาษาฟอร์แทรน 77
fg	นำโพรเซสที่เป็นแบ็กกราวด์กลับมาทำงานแบบโฟร์กราวด์
file	คาดเดาชนิดแฟ้มข้อมูล

finger ดูข้อมูลของประจำตัวผู้ใช้บนระบบ

ftp โปรแกรมขนถ่ายแฟ้มข้อมูลผ่านเน็ตเวิร์ก

g++ คอมไพเลอร์ภาษาซีพลัสพลัสของ GNU

gcc คอมไพเลอร์ภาษาซีของ GNU

gdb ดีบักเกอร์ของ GNU

grep ค้นหาข้อความในแฟ้มข้อมูล

gunzip คลายบีบอัดแฟ้มข้อมูลของ GNU

gzip บีบอัดแฟ้มข้อมูลของ GNU

head แสดงส่วนหัวของแฟ้มข้อมูล

history แสดงคำสั่งที่ป้อนไปทั้งหมด (ซีเชลล์ บอร์นอะเกนเชลล์ และคอร์นเชลล์)

ispell โปรแกรมตรวจสอบการสะกดคำภาษาอังกฤษ

jobs แสดงโพรเซสทั้งหมดที่ทำงานอยู่

join เชื่อมข้อมูลระหว่างแฟ้มแบบอินเทอร์เซ็กชัน

kill ส่งสัญญาณหรือฆ่าโพรเซสตามหมายเลขที่ระบุ

ksh คอร์นเชลล์

last แสดงบันทึกเวลาและวันที่ที่ล็อกอินของผู้ใช้ในระบบ

lint ปรับแต่งซอร์สโค้ดภาษาซีให้น่าอ่าน

ln สร้างลิงก์แฟ้มข้อมูลหรือไดเรกทอรี

logout ออกจากระบบ

lpq ตรวจสอบคิวเครื่องพิมพ์

lpr พิมพ์แฟ้มข้อมูลออกเครื่องพิมพ์ที่กำหนดไว้

lprm ลบงานออกจากคิวเครื่องพิมพ์

lynx เวบบราวเซอร์แบบเท็กซ์

mail ส่งและรับจดหมายอิเล็กทรอนิกส์อย่างง่าย

make ดูแลส่วนประกอบทั้งหมดของซอร์สโค้ดเพื่อแปลงเป็นโค้ดที่รันได้

man ดูคู่มือคำสั่ง

mesg อนุญาตหรือปฏิเสธเมสเซจจากเทอร์มินอลหรือคำร้องขอจากคำสั่ง talk และ write

mkdir สร้างไดเรกทอรี

more ดูข้อมูลในแฟ้มข้อมูลทีละหน้าจอ

mv ย้ายหรือเปลี่ยนชื่อแฟ้มข้อมูลหรือไดเรกทอรี

neqn กำหนดรูปแบบสูตรทางคณิตศาสตร์ใช้ในโปรแกรม nroff

netscape เวบบราวเซอร์แบบกราฟิก

netstat แสดงสถิติของเครือข่าย

nice เปลี่ยนระดับความสำคัญหรือไพรออริตี้ของโพรเซส

nohup บังคับให้โพรเซสเพิกเฉยต่อสัญญาณ SIGHUP เพื่อทำงานต่อไปแม้ว่าผู้ใช้ล็อกเอาต์จากระบบแล้ว

nroff จัดรูปแบบตัวอักษรเพื่อใช้ในการแสดงผลอย่างสวยงาม

page คล้าย more

passwd เปลี่ยนพาสเวิร์ด

paste แปะบรรทัดจากแฟ้มข้อมูล

perl อินเทอร์พรีเตอร์ภาษาเพิร์ล

pic สร้างรูปอย่างง่ายสำหรับอินพุตแก่ troff

pico เอดิเตอร์แบบสกรีน (ง่ายต่อการใช้)

pine ส่งและรับจดหมายอิเล็กทรอนิกส์ (ง่ายต่อการใช้)

pr พิมพ์แฟ้มข้อมูล

ps แสดงสถานะของโพรเซสบนระบบ

pwd แสดงพาธเต็มของไดเรกทอรีปัจจุบัน

quota แสดงเนื้อที่และขีดจำกัดการใช้เนื้อที่จัดเก็บของผู้ใช้

rcp คัดลอกแฟ้มข้อมูลผ่านเครือข่าย

rlogin เข้าสู่ระบบระยะไกล

rm ลบแฟ้มข้อมูล

rmdir ลบไดเรกทอรี

rsh สั่งให้โปรแกรมหรือเชลล์ทำงานระยะไกล

scp คล้าย rcp แต่เพิ่มการเข้ารหัส

sed เอดิเตอร์เท็กซ์แบบสตรีม

set ตั้งค่าตัวแปรสภาพแวดล้อมสำหรับเชลล์นั้นๆ

setenv ตั้งค่าตัวแปรสภาพแวดล้อมกลาง มีเฉพาะซีเชลล์ หากเชลล์อื่นต้องใช้คำสั่ง set และ export ควบคู่กัน

sh บอร์นเชลล์

sort เรียงลำดับแฟ้มเท็กซ์

spell โปรแกรมหาคำภาษาอังกฤษที่สะกดผิด

ssh ซีเคียวเชลล์ คล้าย rlogin และ rsh

stty ตั้งค่าเทอร์มินอล

tail แสดงส่วนท้ายของแฟ้มข้อมูล

talk คุยกับผู้ใช้ที่อยู่บนระบบผ่านเครือข่าย

tar สำรองข้อมูลลงเทปสำรองข้อมูลหรือแฟ้มข้อมูล

tbl จัดรูปแบบตารางสำหรับโปรแกรม nroff/troff

tee แสดงเอาต์พุตออกไปทางเอาต์พุตมาตรฐานและแฟ้มข้อมูล

telnet เข้าสู่ระบบระยะไกลโดยการจำลองเทอร์มินอล

tr แปลงอักขระตามตารางที่กำหนด

troff จัดรูปแบบตัวอักษรเพื่อใช้ในการแสดงผลอย่างสวยงาม คล้าย nroff

umask เปลี่ยนค่าปริยายของสิทธิ์ในการสร้างแฟ้มข้อมูลใหม่

uncompress ขยายแฟ้มข้อมูลที่บีบอัดไว้อย่างง่าย

uniq รายงานหรือลบบรรทัดที่ซ้ำกันในแฟ้มข้อมูล

unzip คลายบีบอัดแฟ้มข้อมูลแบบ ZIP

uptime แสดงภาระของระบบและระยะเวลาตั้งแต่ระบบเริ่มทำงาน

vi	เอดิเตอร์แบบสกรีน (มีให้ใช้ในทุกๆ ระบบยูนิกซ์)
w	แสดงผู้ใช้ที่อยู่บนระบบขณะนี้ และดูว่ากำลังใช้งานคำสั่งใดอยู่
wc	นับจำนวนบรรทัด คำ หรือ อักขระในแฟ้มข้อมูล
whereis	ค้นหาแฟ้มโปรแกรมที่ต้องการในระบบแฟ้มข้อมูลตามพาธ
who	แสดงผู้ใช้ที่กำลังใช้บนระบบ
write	ส่งเมซเสจด่วนให้ยูสเซอร์ที่อยู่บนระบบ
zcat	แสดงข้อมูลในแฟ้มข้อมูลที่บีบอัดไว้ คล้ายคำสั่ง cat
zip	บีบอัดแฟ้มข้อมูลแบบ ZIP