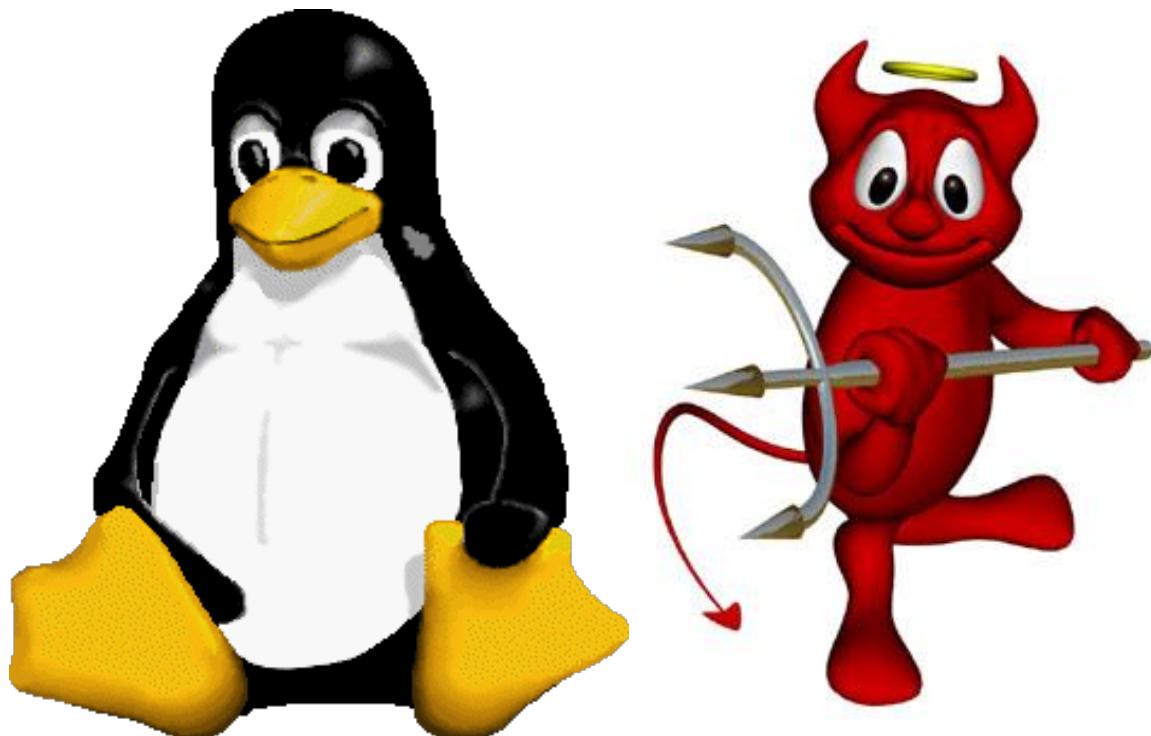


การอบรมยุนิกซ์เชิงปฏิบัติการ

Unix

VSD



โดย

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

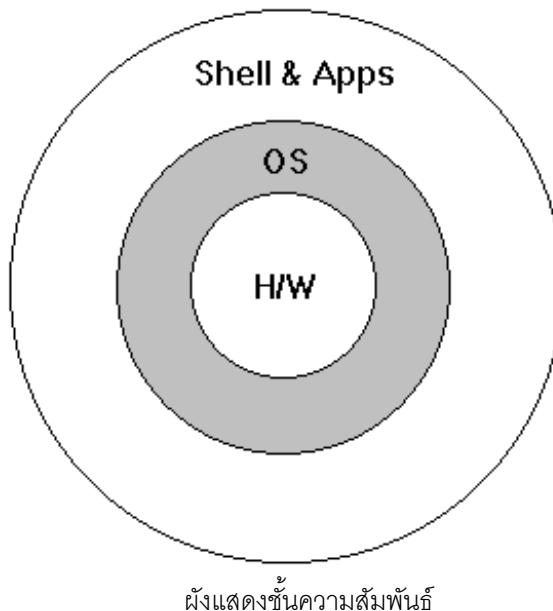
แนะนำระบบปฏิบัติการเบื้องต้น

“A good beginning makes a good ending”

– English proverb

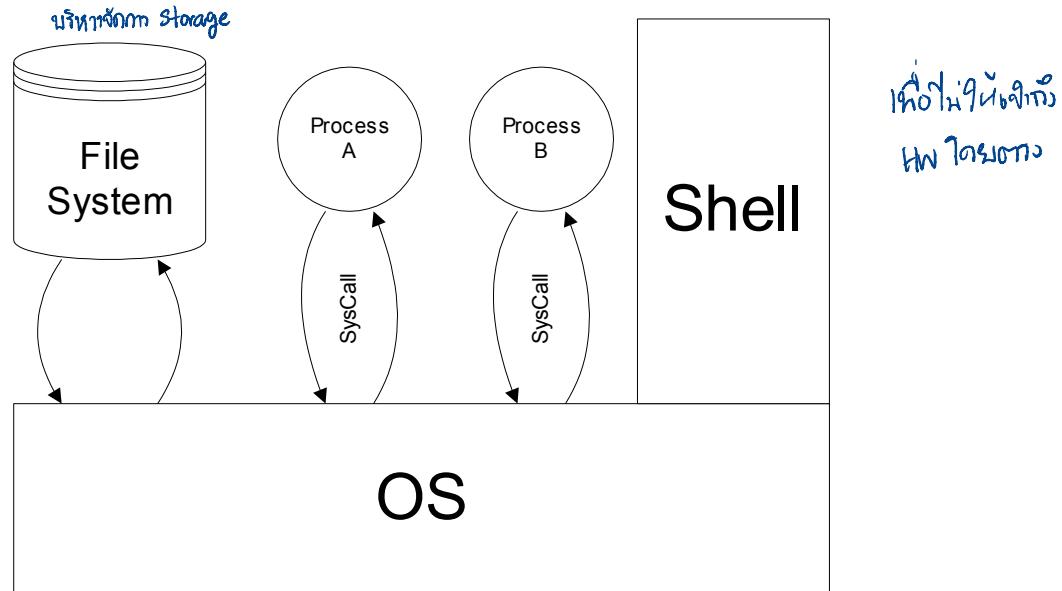
ระบบปฏิบัติการสำหรับคอมพิวเตอร์ถือเป็นส่วนสำคัญยิ่งที่ทำให้มุษย์ผู้ใช้งานใช้คอมพิวเตอร์ได้เต็มประสิทธิภาพอย่างง่ายดาย โดยท่าน้ำที่อยู่ระหว่างชาร์ดแวร์และซอฟฟ์แวร์จะเชื่อมต่อฟร์แวร์ในความเป็นจริงแล้วไม่มีนิยามใดที่สามารถนิยามระบบปฏิบัติการได้อย่างชัดเจนแต่ระบบปฏิบัติการนั้นจำเป็นต้องมีหน้าที่หลักสองประการคือ

1. เพิ่มความสามารถให้แก่เครื่องคอมพิวเตอร์
2. เป็นผู้จัดสรรทรัพยากร



ประวัติของระบบปฏิบัติการ แบ่งตามยุคสมัยของคอมพิวเตอร์

1. ยุคหลอดสูญญากาศและกระดานเชื่อมสาย : ถือได้ว่าไม่มีระบบปฏิบัติการใดเลย หากต้องการใช้งานก็สั่งผ่านกระดานเชื่อมสายหรือสวิตซ์สัญญาณโดยตรงเพื่อบังคับอุปกรณ์อิเล็กทรอนิกส์อีกทีหนึ่ง
2. ยุคทิรานชิสเตอร์และระบบแบบตัวต่อตัว : มีระบบปฏิบัติการเพื่อช่วยเหลือการประมวลผลบัตรเจาะรูแบบแบบต์ชอปผู้บังคับ ไม่มีระบบติดต่อกันผู้ใช้มีแต่ตัวต่อตัวอย่างๆ บนบานพ์ล์ฟล์อ์ร์ด์คือถ่ายข้อมูลไปยังสื่อกลางที่เป็นเทปแม่เหล็กเพื่อเร่งความเร็วด้านไอโอดีของระบบคอมพิวเตอร์



๔ ล ะ เที่ ย บ บ ค ร ี โ ข า ย ค օ մ พ ิ ว ட ი օ ร ্ ว ი ด
อีกทั้งมีการพัฒนาระบบปฏิบัติการให้ใช้งานโพธิ์เซอร์วิสได้มากกว่าหนึ่งตัว ระบบงานควบคุมตามเวลาจริง

ผังแสดงความสัมพันธ์ระหว่างแนวคิดหลักในระบบ | กิจกรรมภาษา

แนวคิดหลักในการทำความเข้าใจระบบปฏิบัติการ

1. โพรเชส : ส่วนปฏิบัติงานของโปรแกรมที่ได้โหลดลงในเมโมรีหลักและตีความโดยหน่วยประมวลผลกลาง โดยมากในระบบหนึ่งมักมีงานหรือโพรเชสแอ็อกที่ฟอยู่จำนวนมากกว่าหนึ่งโพรเชส
 2. ระบบแฟ้มข้อมูล : แนวทางการจัดเก็บข้อมูลลงดิสก์หรือระบบจัดเก็บไดๆ ที่คงสภาพได้โดยไม่ต้องใช้กระแสไฟล์เสียง
 3. การเรียกใช้งานบริการของระบบ : ส่วนการให้บริการของระบบปฏิบัติการแก่โพรเชสต่างๆ เพื่อกำหนดความสะดวกในการปฏิบัติงานและจัดสรรทรัพยากรให้เกิดประโยชน์สูงสุด
 4. เชลล์ : ส่วนเดียวของผู้ใช้งานจริง เพื่อควบคุมระบบคอมพิวเตอร์ได้อย่างมีประสิทธิภาพและเรียบง่ายที่สุด

ประเภทของสร้างภัยในของระบบปฏิบัติการ

- ระบบแบบโมโนลิทิก : ส่วนระบบปฏิบัติการทุกอย่างรวมอยู่ในแอ็ดเดรสสเปซเดียวกันหมด การเรียกใช้ระบบภายในคือการกระโดดไปยังแอดเดรสของรูทินนั่นๆ เช่น คอม เป็นต้น
 - ระบบแบบเลเยอร์ : ส่วนระบบปฏิบัติการแบ่งออกเป็นชั้นๆ โดยแต่ละชั้นส่งงานผ่านต่อเป็นทอดๆ (แบ่งตามแนวตั้ง)

3. ระบบแบบเวอร์ชวลแมชีน : มีระบบปฏิบัติการทำงานพร้อมกันอยู่บนเครื่องคอมพิวเตอร์มากกว่าหนึ่งระบบ
มักใช้ในระบบปฏิบัติการของเมนเนเฟร์ม เพื่อความทนทานต่อข้อผิดพลาด เช่น OS/390 เป็นต้น
4. ระบบแบบไฮคลอินเตอร์/เซิร์ฟเวอร์ : ส่วนระบบปฏิบัติการแบ่งออกเป็นรุ่นย่อยที่อยู่แยกจากกัน¹
(แบ่งตามจำนวนอน) การเรียกใช้ระบบภายในคือการเรียกรุ่นย่อยพร้อมผ่านค่าพารามิเตอร์ประกอบ
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

ມູນົງກໍ່ເປັນຮະບບປະລິບຕິກາຣເກົ່າແກ່ທີ່ໄດ້ຮັບຄວາມນິຍມມາຍວານ
ດ້ວຍກາຣຍຄວັບແນວຄົດທີ່ຄວບດ້ານທັງສາສຕ່ວລະສຶລືປໍໃຫ້ມູນົງກໍ່ເປັນຮະບບປະລິບຕິກາຣຕໍ່ນັ້ນແບບຂອງຮະບບປະລິບຕິກາຣປັຈຈຸບັນ
ແນວໜີ່ດູແໜ້ນວ່າເປັນຂອງເກົ່າລ້າສັນຍະແລະນ່າຈະໃຊ້ງານຍາກ ແຕ່ເນື່ອໄດ້ໃຊ້ງານອຍ່າງຈົງຈັງແລ້ວ ຜູ້ໃຊ້ແກບທຸກຄົນຕ່າງໆພົກໃຈ
ເນື່ອງຈາກມູນົງກໍ່ຕອບສົນອັນຄວາມຕໍ່ອັກກາຣຕໍ່າງໆ ໄດ້ເປັນອໍາຍ່າງດີ
ໂດຍເຂົພາະຄວາມເຮົຍບ່າຍແຕ່ແຜ່ງໄວ້ຮັງພລັງໃນກາຣກອກການເປົ້າພລັງ

การเชื่อมต่อและเข้าสู่ระบบ (Connecting and Logging in)

ความรู้พื้นฐานที่จำเป็น (Essential basis)

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

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

$\wedge = \text{ctrl}$

ปุ่มพิเศษควบคุมการทำงานโปรแกรม มีดังนี้

- $\wedge D$ ออกจากระบบหรือสิ้นสุดแฟ้มข้อมูลที่รับจากคีย์บอร์ด (eof)
- $\wedge S$ หยุดการแสดงผลหน้าจอชั่วคราว (stop) แม้ใช้ในกรณีแสดงผลออกหน้าจอหลายบรรทัดจนดูไม่ทัน
- $\wedge Q$ ยกเลิกการหยุดการแสดงผลชั่วคราวที่สั่งจากปุ่ม $\wedge S$ (start)
- $\wedge C$ ขัดจังหวะและยุติการทำงานของโปรแกรมที่ทำงานอยู่ปัจจุบัน (intr)
- $\wedge Z$ ทำให้โปรแกรมที่ทำงานอยู่ปัจจุบันทำงานแบบเบรกกราวด์ (susp)

เมฆรังกิ ram

อ ก ข ร ะ ท ี ม ี ค ว า น ห น า ย พ ิ ศ ษ :

โดยปกติเชลล์จะประมวลผลขักขระพิเศษเหล่านี้ก่อนส่งต่อการทำงานให้แก่โปรแกรมที่เรียกใช้ มีดังนี้

เพื่อกำ redirection

เก็บกันอย่างไรก็ได้

ก็ได้

input แบบ Text file

- | | |
|----------------------------------|--|
| $<$
$>$
$>>$
$ $
$*$ | อ่านจากแฟ้มข้อมูลที่อยู่ทางขวาของสัญลักษณ์นี้แทนการอ่านจากอินพุตมาตรฐาน
เขียนทับลงบนแฟ้มข้อมูลที่อยู่ทางขวาของสัญลักษณ์นี้แทนการเขียนไปที่เอกสารพุตมาตรฐาน
เขียนต่อท้ายแฟ้มข้อมูลที่อยู่ทางขวาของสัญลักษณ์นี้แทนการเขียนไปที่เอกสารพุตมาตรฐาน
สร้างไปปี คือการถ่ายทอดเอกสารพุตจากโปรแกรมด้านซ้ายไปเป็นอินพุตของโปรแกรมด้านขวา
เป็นอักขระเมตา ใช้แทนตัวอักษรใดๆ ยาวตั้งแต่ศูนย์ตัวนี้ไป เช่น หากอ้าง A* อาจเป็น A1, AA, |
|----------------------------------|--|

$\wedge = \text{ctrl}$

Abc, ARGUE, African ก็ได้ เป็นต้น

อักษรmeta (meta character) ตัวอักษรที่ไม่ป้อนมาในส่วนต่อมาเป็นตัวอักษร เช่น

* เหลือ $\wedge A$ \leftarrow ?

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

Ex rm $\wedge A$ \rightarrow rm ก็ได้ ไม่ต้องพิมพ์ตัว A แล้ว

เป็นอักขระเมตา เพื่อบ่งบอกว่าตัวอักษรที่ตามมาต่อไปไม่มีการตีความเป็นอักขระความหมายพิเศษ

$\wedge g = \text{ctrl}$ และ $\wedge d = \text{ctrl}$

และเสริมอีกเป็นตัวอักษรประกอบตัวหนึ่ง เช่น $\wedge *$ ทำให้ * เป็นตัวอักษรไม่ใช้อักขระเมตา หากต้องการอักขระ \ ก็พิมพ์เป็น

$\wedge \wedge$ เป็นต้น

direction pipe ... 100% 

ระบบความช่วยเหลือประกอบไปด้วยคู่มือการใช้คำสั่งที่ถือได้ว่าครบถ้วนและละเอียด เรียกอ่านได้โดยใช้คำสั่ง man เช่น ต้องการอ่านคู่มือของคำสั่ง rm (remove)

\$ man man \rightarrow คู่มือ manual
\$ man rm \rightarrow คู่มือ rm option opt
↳ คู่มือ manual

คู่มือแสดงบนหน้าจอแล้วหยุดเป็นหน้าๆ กดปุ่มเว้นวรรคเพื่อไปหน้าต่อไป กดปุ่ม q เพื่อสิ้นสุดการอ่านคู่มือ บางระบบอาจกดปุ่ม $\wedge B$ และ $\wedge F$ เพื่อเลื่อนหน้ากลับไปยังหน้าก่อนหน้าและหน้าถัดไปตามลำดับ

ระบบแฟ้มข้อมูลและไดเรกทอรี (File and directory system)

หลักการทั่วไปของระบบแฟ้มข้อมูลและไดเรกทอรีของยูนิฟอร์มเป็นเช่นเดียวกับของดอสและวินโดว์ เนื้องจากทั้งสองและวินโดว์

แต่ได้ติดตอนเรื่องสิทธิออกໄປ และมีอักษรไวดาร์พีชื่นมาประกอบในระบบเพิ่มข้อมูลและได้เรียกว่าของยูนิกซ์มีเรื่องสิทธิ์และต้องมีเจ้าของเสมอ และไม่มีการทำคำนึงถึงอักษรไวดาร์พีเลย ดิสก์ทุกตัวรวมเข้ากับระบบได้เรียกว่าโดยเริ่มจาก / หรือรูตเสมอ ที่เป็นจุดเด่นอีกอย่างของระบบยูนิกซ์คือ ฮาร์ดแวร์ต่อพ่วงทุกอย่าง ระบบยูนิกซ์มองเป็น 'แฟ้มข้อมูล' เสมอ เช่น /dev/tty /dev/mouse และ /dev/hda เป็นต้น

ชื่อแฟ้มข้อมูลหรือชื่อได้เรียกว่ารีบันยูนิกซ์มีความหมายได้มากถึง 255 ตัวอักษร (บางระบบอาจมากกว่านี้) อีกทั้งสามารถใช้ตัวเลข ตัวอักษร คักหรือพิเศษในการตั้งชื่อได้ ซึ่งมีข้อจำกัดน้อยกว่าดอสและวินโดวส์มาก แต่เพราจะเห็นนี้เข่นกันทำให้การอ้างถึงชื่อแฟ้มข้อมูลหรือชื่อได้เรียกว่ารีบันยูนิกซ์มีความซับซ้อนยิ่งขึ้น โดยเฉพาะการตั้งชื่อที่ประกอบด้วยอักษรและตัวเลข

คำสั่งเบื้องต้นที่เกี่ยวข้อง คือ

การแสดงผล simple

ls : แสดงรายชื่อไฟล์และโฟลเดอร์ที่มีชื่อเริ่มต้นด้วยเครื่องหมายจุด “.” ซึ่งมักเป็นแฟ้มข้อมูลที่ต้องการซ่อนไว้ สามารถระบุอักษรและตัวเลขเพื่อค้นหาชื่อที่ต้องการและขอบเขตเพิ่มเติมได้ เช่น

option

ls -alF แสดงรายชื่อทั้งหมด แม้ว่าจะซ่อนไว้ก็ตาม

ls -l แสดงรายชื่อแบบละเอียด ประกอบด้วยสิทธิ์จำนวนบล็อกขนาด 512 ไปต่ำสุด เก็บชื่อเจ้าของ ชื่อกลุ่มเจ้าของ ขนาดแฟ้มข้อมูล วันเวลาที่ได้รับการแก้ไขล่าสุด และชื่อ

ls -d sort by timestamp แสดงรายชื่อเฉพาะได้เรียกว่าไม่แสดงรายชื่อแฟ้มข้อมูลที่อยู่ในได้เรียกว่า "dot"

ls -t แสดงรายชื่อเรียงตามลำดับวันเวลาที่ได้รับการแก้ไขล่าสุด

อปชันต่างๆ ใช้ผสมกันได้ การใช้งานนั้นคล้ายคำสั่ง "DIR" ของdosมาก ตัวอย่างผลลัพธ์เช่น

Admin ห้องทำงาน → /right permission / ผู้ใช้งาน ตัวอย่าง

Unix file owner	Unix file permission	การกำหนดค่าส่วนตัว
-----------------	----------------------	--------------------

chmod ไฟล์ mode
(read/write ,read only ถ้าไม่ถูก)

r : อ่าน
w : เขียน
- : ลบ

\$ ls
Mail News mail
\$ ls -alF
total 70
drwx--x--x 10 adek
drwxr-xr-x 32 root
-rw----- 1 adek
-rw----- 1 adek
-rw----- 1 adek
drwx----- 2 adek
-rw----- 1 adek
-rw----- 1 adek
-rw----- 1 adek
drwx----- 2 adek
drwx----- 2 adek
drwx----- 2 adek
drwx----- 3 adek
drwx----- 2 adek
drwx----- 2 adek
drwx----- 6 adek

file permission 1 ก้อน
- permissions
- ไม่มี r
- ไม่มี w
- ไม่มี x
1 บันทึกที่ไม่มี permission
ไม่มี r ไม่มี w ไม่มี x

ACL (access control list)
user ห้องทำงานไม่ได้รับ
- ไม่มี r
- ไม่มี w
- ไม่มี x
group ห้องทำงาน
- ไม่มี r
- ไม่มี w
- ไม่มี x
others ห้องทำงาน
- ไม่มี r
- ไม่มี w
- ไม่มี x
A: ห้องทำงานให้ทุกคน

OWNER	dir	file
lect	./ = direct directory	1024 Nov 3 10:50 .. /
sys		107 Jul 21 1998 .Xauthority
lect		731 Mar 13 15:41 .addressbook
lect		2399 Mar 13 15:41 .addressbook.lu
lect		1024 Mar 30 11:12 .pgp/
lect		14097 Apr 3 20:59 .pinerc
lect		595 Sep 14 1999 .profile
lect		2712 Apr 17 18:59 .sh_history
lect		113 Apr 5 09:27 .signature
lect		24 Aug 28 1998 Mail/
lect		24 Aug 28 1998 News/
lect		1024 Apr 17 15:02 mail/
lect		1024 Apr 3 20:58 mycode/
lect		1024 Oct 24 1998 mydoc/
lect		1024 Apr 9 21:49 tmp/
lect		1024 Apr 17 18:48 www/

ln -s ไฟล์ ลิงก์ ไฟล์ ลิงก์
ln -s ไฟล์ ลิงก์ ไฟล์ ลิงก์

cp : คัดลอกแฟ้มข้อมูล การใช้งานนั้นคล้ายคำสั่ง "COPY" ของdosมาก

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

`rm` : ล บ แ ฟ մ ข อ ม ู ล ก า ร ิ ช ั ง า น น ั น ค ล ա ย คำ ส ั ง “DELETE” ของ ด օ ส մ า ก เป็นคำสั่งที่ขันตราณ์ออกจากในระบบญี่ปุ่นิกซ์หากไม่สามารถถูกลบได้แม้มที่ลบไปแล้วได้

`cat` : แสดงรายละเอียดในไฟล์ข้อมูลออกเอกสาร์พุฒมาตรฐาน

`pwd` : แสดงพาธของไดเรกทอรีปัจจุบัน

`cd` : ย้ายไดเรกทอรีปัจจุบันไปเป็นตามที่ระบุ หากไม่ระบุจะกลับไปยังพารที่กำหนดในตัวแปร \$HOME

`mkdir` : สร้างไดเรกทอรีว่างตามที่ระบุ

`rmdir` : ลบไดเรกทอรีว่างตามที่ระบุ

การจัดการไฟล์และสิทธิ์ (File and permission management)

ไฟล์ข้อมูลและไดเรกทอรีต้องมีเจ้าของ 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

- อักษร r 代表เป็นการระบุชนิด ถ้าเป็น d คือไดเรกทอรี | คือไฟล์แบบขอฟ์ต์ลิงก์ c คือ แฟ้ม ดี ไวร์ แบบ ค่า แรก เตอร์ และ b คือ แฟ้ม ดี ไวร์ แบบ บล็อก หากไม่ระบุคือเป็นข้อดีคือเป็นไฟล์ข้อมูลธรรมด้า แต่ไม่สามารถบอกรหัสได้ว่าเป็นไฟล์เทกซ์หรือไฟล์binary
- อักษร s 代表สามารถตัวกลุ่มแรกเป็นการระบุสิทธิ์ของเจ้าของ (ตามที่ระบุไว้ในคอลัมน์ที่ 3) ถ้าเป็น r คืออ่านได้ w คือเขียนได้ และ x คือสั่งรันไฟล์ได้ (อาจเป็นไฟล์โค้ดไบนาเรียหรือไฟล์เทกซ์เชลล์สคริปต์ก็ได้) หากไม่ระบุคือเป็นข้อดีคือว่าไม่มีสิทธิ์ในการกระทำนั้นๆ เช่น rx คือมีสิทธิ์อ่าน เขียน และสั่งรัน เป็นต้น
- อักษร s 代表สามารถตัวกลุ่มกลางเป็นการระบุสิทธิ์ของกลุ่มเจ้าของ (ตามที่ระบุไว้ในคอลัมน์ที่ 4) เช่น r-x คือมีสิทธิ์อ่านและสั่งรันเท่านั้น เขียนไฟล์ไม่ได้ เป็นต้น
- อักษร s 代表สามารถตัวกลุ่มสุดท้ายเป็นการระบุสิทธิ์ของผู้อื่น (other) ที่ไม่ใช่เจ้าของและไม่อยู่ในกลุ่มเจ้าของ (ตามที่ระบุไว้ในคอลัมน์ที่ 3 และ 4) เช่น --- คือไม่มีสิทธิ์ใดๆ เลย

คอลัมน์ที่ 3 เป็นชื่อเจ้าของ (owner id)

คอลัมน์ที่ 4 เป็นชื่อกลุ่มเจ้าของ (group id)

คำสั่งเบื้องต้นที่เกี่ยวข้อง คือ

`chmod` : เปลี่ยนสิทธิ์ประจำไฟล์ข้อมูลหรือไดเรกทอรี

`chown` : เปลี่ยนเจ้าของประจำไฟล์ข้อมูลหรือไดเรกทอรี

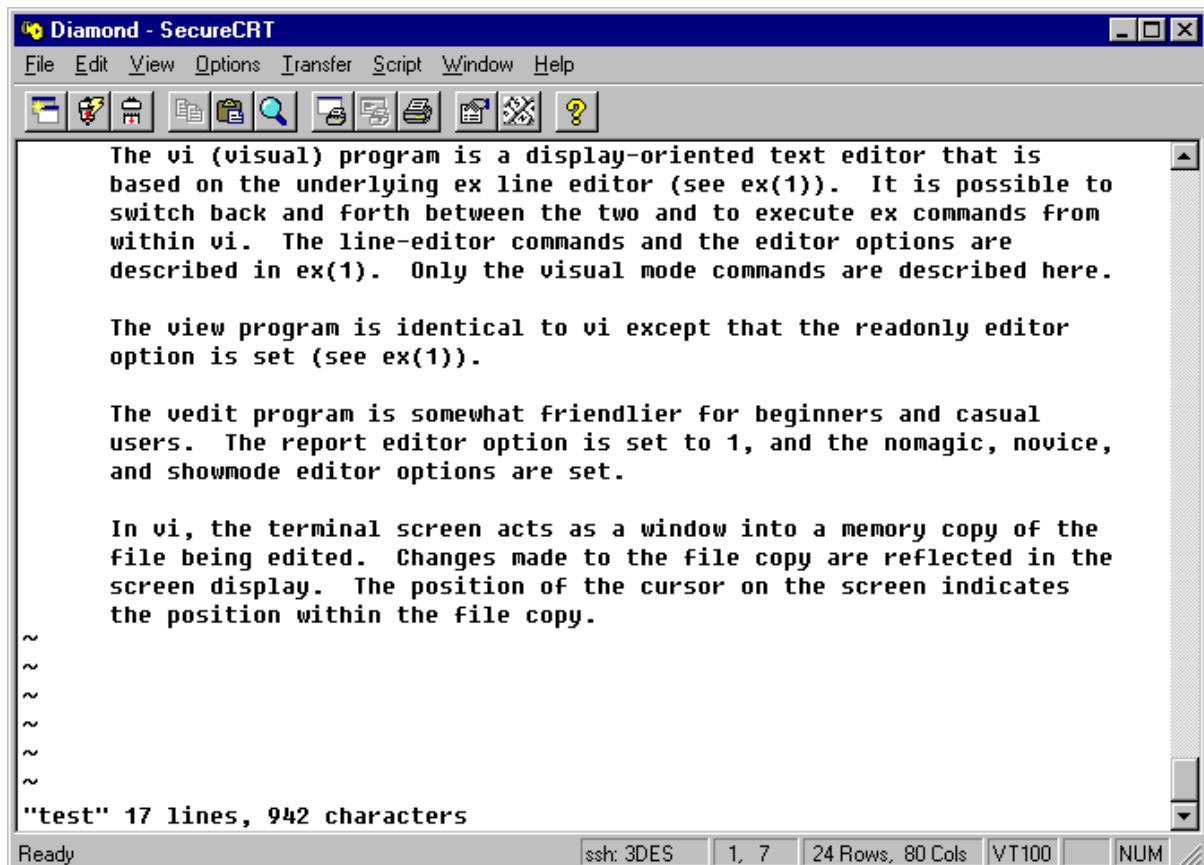
`chgrp` : เปลี่ยนกลุ่มเจ้าของประจำไฟล์ข้อมูลหรือไดเรกทอรี

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 เป็นเอดิเตอร์แบบสกรีนที่ใช้แฟร์ชัลเวย์ที่สุด เนื่องจากมีให้ใช้บนยูนิกซ์ทุกรุ่นทุกยี่ห้อ และมีความคล่องตัวในการทำงานสูง แต่สำหรับผู้ใช้ทั่วไปและผู้ใช้มือใหม่แล้วมักหลีกเลี่ยงการใช้งาน vi เนื่องจากไม่เข้าใจการใช้งานที่ค่อนข้าง слับซับซ้อน ทว่าถ้าเข้าใจกลไกการทำงานของ vi แล้วจะพบว่าเป็นเอดิเตอร์แบบสกรีนที่น่าใช้และมีความคล่องตัวสูงมาก



หน้าจอโปรแกรม vi

เริ่มตั้งแต่บรรทัดแรกเป็นต้นไปถึงก่อนบรรทัดสุดท้ายซึ่งเป็นส่วนแสดงผลและต้องตอบกับผู้ใช้ของ vi เช่น หากขนาดเทอร์มินอลเป็น 80x24 บรรทัดที่ 24 เป็นส่วนแสดงผลและต้องตอบกับผู้ใช้ เป็นต้น

โหมดการทำงานของ vi แบ่งเป็นสองโหมด คือ โหมดต้องการคำสั่ง (Command Mode) และโหมดแก้ไขข้อความ (Editing Mode)

- โหมดคำสั่ง (Command Mode) เป็นโหมดที่ผู้ใช้สามารถออกคำสั่งให้ vi ทำงานได้ตามความต้องการ
- โหมดแก้ไขข้อความ (Editing Mode) เป็นโหมดที่ผู้ใช้ป้อนหรือแก้ไขข้อความในเอกสาร

เมื่อเรียกใช้ vi จะเริ่มต้นที่โหมดคำสั่ง ให้โดยออกคำสั่งแก้ไขข้อความ เช่น i หรือ a เป็นต้น

การสับจากโหมดแก้ไขข้อความเป็นโหมดคำสั่งทำได้โดยกดปุ่ม ESC หรือ ^[(ข้อแนะนำ หากไม่มั่นใจว่า vi อยู่ในโหมดคำสั่งหรือไม่สามารถกด ESC ข้ามได้ถ้าอยู่ในโหมดคำสั่งอยู่แล้วจะมีเสียงบีบ)

ในส่วนของโหมดคำสั่งยังแบ่งอยู่เป็นสองโหมด เนื่องจากอันที่จริงแล้ว vi เป็นส่วนเสริมจากエ迪เตอร์แบบบรรทัดซึ่งมักเขียนด้วยเครื่องหมายทวิภาค “：“ ซึ่งเป็นเอ迪เตอร์แบบบรรทัดซึ่งมักเขียนด้วยเครื่องหมายทวิภาค “：“

การเรียกใช้ ทำได้โดย

- เรียกโปรแกรม vi โดยตรงจากชุดพรมต์

```
$ vi
```

- หากต้องการแก้ไขไฟล์ข้อมูลเท็กซ์ทมีอยู่แล้วก็เรียกโปรแกรม vi และตามด้วยชื่อไฟล์นั้นๆ

```
$ vi filename.txt
```

- หากต้องการสร้างไฟล์ข้อมูลเท็กซ์ใหม่ เรียกโปรแกรม vi และตามด้วยชื่อไฟล์ใหม่นั้นๆ

```
$ vi newfilename.txt
```

คำสั่งเพิ่มฐานในโหมดคำสั่ง vi

ออกจากการโปรแกรม

ZZ ออกจากโปรแกรมและจัดเก็บไฟล์หากมีการแก้ไข

Q สลับไปยังโหมดคำสั่ง ex (กลับไปยังโหมดคำสั่ง vi โดยพิมพ์ “vi”)

: ออกคำสั่งในโหมดคำสั่ง ex เพียงคำสั่งเดียวแล้วกลับไปยังโหมดคำสั่ง vi โดยอัตโนมัติ

ในโหมดคำสั่ง vi นี้รูปแบบเพิ่มฐานคือ [count] command [where]

คำประกอบ where เพิ่มฐาน

w ไปจนสิ้นสุดคำ

^ ไปจนต้นบรรทัด

\$ ไปจนท้ายบรรทัด

d ทั้งบรรทัด

ตัวอย่างเช่น

d^ ลบอักษรตั้งแต่เครื่องเซอร์ไปจนต้นบรรทัด

d\$ ลบอักษรตั้งแต่เครื่องเซอร์ไปจนท้ายบรรทัด

dw ลบอักษรตั้งแต่เครื่องเซอร์ไปจนสิ้นสุดคำ

3dd ลบบรรทัดทั้งบรรทัดไปสามครั้ง (ลบบรรทัดปัจจุบันและอีกสองบรรทัดลงไป)

การเพิ่มข้อความใหม่

A เข้าสู่โหมดแก้ไขข้อความแบบต่อท้ายบรรทัด

a เข้าสู่โหมดแก้ไขข้อความแบบต่อท้ายเครื่องเซอร์

- | เข้าสู่ใหม่แก้ไขข้อความแบบแทรกต้นบรรทัด (ตัวไอในญี่ปุ่น)
- | เข้าสู่ใหม่แก้ไขข้อความแบบแทรกที่เครอร์เชอร์
- | เข้าสู่ใหม่แก้ไขข้อความแบบแทรกบรรทัดก่อนบรรทัดปัจจุบัน (ตัวไอในญี่ปุ่น)
- | เข้าสู่ใหม่แก้ไขข้อความแบบต่อบรรทัดจากบรรทัดปัจจุบัน (ตัวโอลีก)

การเลื่อนเครอร์เชอร์

- h เคลื่อนเครอร์เชอร์ไปทางซ้ายเมื่อ หรืออาจใช้ปุ่มลูกศรซ้ายก็ได้
- i เคลื่อนเครอร์เชอร์ไปทางขวาเมื่อ หรืออาจใช้ปุ่มลูกศรขวาก็ได้ (ตัวแอลลีก)
- k เคลื่อนเครอร์เชอร์ไปบรรทัดบน หรืออาจใช้ปุ่มลูกศรชี้บนก็ได้
- j เคลื่อนเครอร์เชอร์ไปบรรทัดล่าง หรืออาจใช้ปุ่มลูกศรชี้ลงก็ได้
- \$ เคลื่อนเครอร์เชอร์ไปยังท้ายบรรทัด
- ^ เคลื่อนเครอร์เชอร์ไปยังต้นบรรทัดที่ไม่ใช้อักษรเด่นๆ
- 0 เคลื่อนเครอร์เชอร์ไปยังคอลัมน์แรกของบรรทัด (เลขศูนย์)
- B ดอยหลังหนึ่งคำ โดยข้ามเครื่องหมายขั้นวรรค
- b ดอยหลังหนึ่งคำ โดยหากเครอร์เชอร์อยู่ระหว่างคำจะไปยังต้นคำนั้น
- W เคลื่อนไปยังต้นคำ โดยข้ามเครื่องหมายขั้นวรรค
- w เคลื่อนไปยังคำถัดไป
- E เคลื่อนไปยังท้ายคำ โดยข้ามเครื่องหมายขั้นวรรค
- e เคลื่อนไปยังท้ายคำ โดยหากเครอร์เชอร์อยู่ระหว่างคำจะไปยังท้ายคำนั้น
- G เคลื่อนไปยังบรรทัดที่กำหนดโดย count หากไม่วางจะไปยังบรรทัดสุดท้ายของแฟ้มข้อมูล
- [^]B เปลี่ยนไปยังหน้าก่อนหน้า
- [^]U เปลี่ยนไปยังครึ่งหน้าก่อนหน้า
- [^]F เปลี่ยนไปยังหน้าถัดไป
- [^]D เปลี่ยนไปยังครึ่งหน้าถัดไป
- [^]P เคลื่อนเครอร์เชอร์ไปบรรทัดบนในคอลัมน์เดียวกัน
- [^]N เคลื่อนเครอร์เชอร์ไปบรรทัดล่างในคอลัมน์เดียวกัน

การเปลี่ยนข้อความ

- C เปลี่ยนข้อความตั้งแต่เครอร์เชอร์ไปจนสุดบรรทัด
 - R เปลี่ยนอักษรไว้เปรียบฯ จนกว่ากดปุ่ม ESC
 - r เปลี่ยนอักษรหนึ่งตัวที่เครอร์เชอร์
 - J เชื่อมบรรทัดปัจจุบันกับบรรทัดล่างให้เป็นบรรทัดเดียวกัน
- การลบ/ตัด คัดลอก และแปะข้อความ
- D ลบ/ตัดข้อความตั้งแต่เครอร์เชอร์ไปจนสุดบรรทัด
 - d ลบ/ตัดข้อความ (ต้องระบุขอบเขตประกอบด้วย) หากสั่ง “ dd ” คือลบ/ตัดบรรทัดปัจจุบันทั้งบรรทัด
 - Y คัดลอกบรรทัดปัจจุบันทั้งบรรทัด
 - y คัดลอกข้อความ (ต้องระบุขอบเขตประกอบด้วย) หากสั่ง “ yy ” คือคัดลอกบรรทัดปัจจุบันทั้งบรรทัด
 - X ลบอักษรที่ก่อนหน้าเครอร์เชอร์
 - x ลบอักษรที่เครอร์เชอร์

- P ประชี้อความที่ได้ตัดหรือคัดลอกล่าสุดไว้ก่อนหน้าบรรทัดปัจจุบัน
- P ประชี้อความที่ได้ตัดหรือคัดลอกล่าสุดไว้ต่อจากบรรทัดปัจจุบัน

การค้นหา

- / ค้นหาคำจากเครื่องเซอร์วิสไปจนถึงบรรทัดท้ายแฟ้มข้อมูล
 - ? ค้นหาคำจากเครื่องเซอร์วิสไปจนถึงบรรทัดแรกของแฟ้มข้อมูล
 - n ค้นหาต่อไป ในทิศทางคงเดิม
 - N ค้นหาต่อไป ในทิศทางที่ตรงข้ามจากทิศทางเดิม
- คำสั่งอื่นๆ
- ^{^G} แสดงตำแหน่งเครื่องเซอร์วิสปัจจุบันและสถานะของแฟ้มข้อมูล
 - ^{^L} วาดหน้าจอเทอร์มินอลใหม่ มักใช้ในการแสดงผลผิดเพี้ยนหรือแสดงอักษรขยะขึ้นมา
 - ! เรียกเชลล์เพื่อปฏิบัติงาน
 - u ยกเลิกการกระทำการแก้ไข (undo) หากสั่ง u อีกครั้งติดกันหมายถึงไม่ยกเลิกการกระทำการทำนั้น (redo)

คำสั่งพื้นฐานในโหมดคำสั่ง ex

- :q ออกจากรูปแบบ vi หากแฟ้มข้อมูลได้รับการแก้ไขแต่ยังไม่ได้จัดเก็บ vi จะแจ้งเตือนและยังไม่ให้ออกจากไปรูปแบบ
- :q! ออกจากไปรูปแบบ vi โดยไม่สนใจจัดเก็บแฟ้มข้อมูลที่อาจได้รับการแก้ไข
- :w จัดเก็บแฟ้มข้อมูลทับชื่อเดิม
- :w file จัดเก็บแฟ้มข้อมูลในชื่อใหม่ เช่น :w file-a.txt คือจัดเก็บลงแฟ้มชื่อ file-a.txt เป็นต้น
- :wq จัดเก็บแฟ้มข้อมูลทับชื่อเดิม และออกจากรูปแบบ
- :x จัดเก็บแฟ้มข้อมูลทับชื่อเดิม และออกจากรูปแบบ

ข้อสังเกต

- หากเป็นการสร้างแฟ้มใหม่หรือแก้ไขแฟ้มข้อมูล บรรทัดหลังจากบรรทัดสุดท้ายของแฟ้มข้อมูล vi แสดงเพื่อมให้ผู้ใช้งานว่าไม่มีพบท้ายจากว่างเปล่า
- โปรดระวังเรื่องการจัดเก็บข้อมูลทับลงในแฟ้มข้อมูลโดยไม่ได้ตั้งใจ
- ผู้ดูแลระบบบินิมใช้ vi มากกว่า pico
- ในโหมดคำสั่ง vi ตัวพิมพ์เล็กและตัวพิมพ์ใหญ่ถือว่าแตกต่างกัน และหาก ^ นำหน้าอักษรใดอีกว่าเป็นการกดปุ่ม Control ร่วมกับปุ่มอักษรนั้น เช่น ^B หมายถึงกดปุ่ม Ctrl และกดปุ่ม B เวลาเดียวกัน ที่ตามหลังอักษรอื่นๆหมายถึงอักษร ^ จริงๆ ทั้งนี้ต้องพิจารณาให้ดีเมื่อต้องใช้
- ในโหมดคำสั่ง ex นิยมเขียน “：“ นำหน้าไว้เพื่อแยกแยะว่าเป็นคำสั่งในโหมดนี้ เมื่อใช้งานจริงในโหมดคำสั่ง ex ไม่ต้องพิมพ์ เช่น “:q!” ก็พิมพ์ “：“ เพื่อเข้าสู่โหมดคำสั่ง ex และตามด้วย “q!” เป็นต้น
- หากต้องการพิมพ์อักษรหรือข้อความภาษาไทยซึ่งต้องใช้ ASCII แบบ 8 บิต vi สนับสนุนอยู่แล้ว แต่ต้องตั้งค่าให้เทอร์มินอลรองรับการใช้งาน ASCII แบบ 8 บิตด้วย (ปรกติรับแค่ 7 บิต) โดยสั่งยกเลิกการลงทะเบียนบิตที่แปด

```
$ stty -istrip
```

- อาจมีบางกรณีที่เมื่อเรียก vi แล้วไปร่างrogram แจ้งเตือนว่า “Visual needs addressible cursor or upline capability” และเข้าสู่โหมดคำสั่ง ex ทันที (ออกจาก vi นี้โดยคำสั่ง “q! ”) หมายถึงตัวแปรสภาพแผลล้อม TERM ไม่ได้ตั้งค่า แก้ไขโดยตั้งค่าตัวแปรสภาพแผลล้อม TERM ให้เหมาะสมกับเทอร์มินอลนั้น เช่น สำหรับ sh ksh และ bash

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

หรือ สำหรับ csh

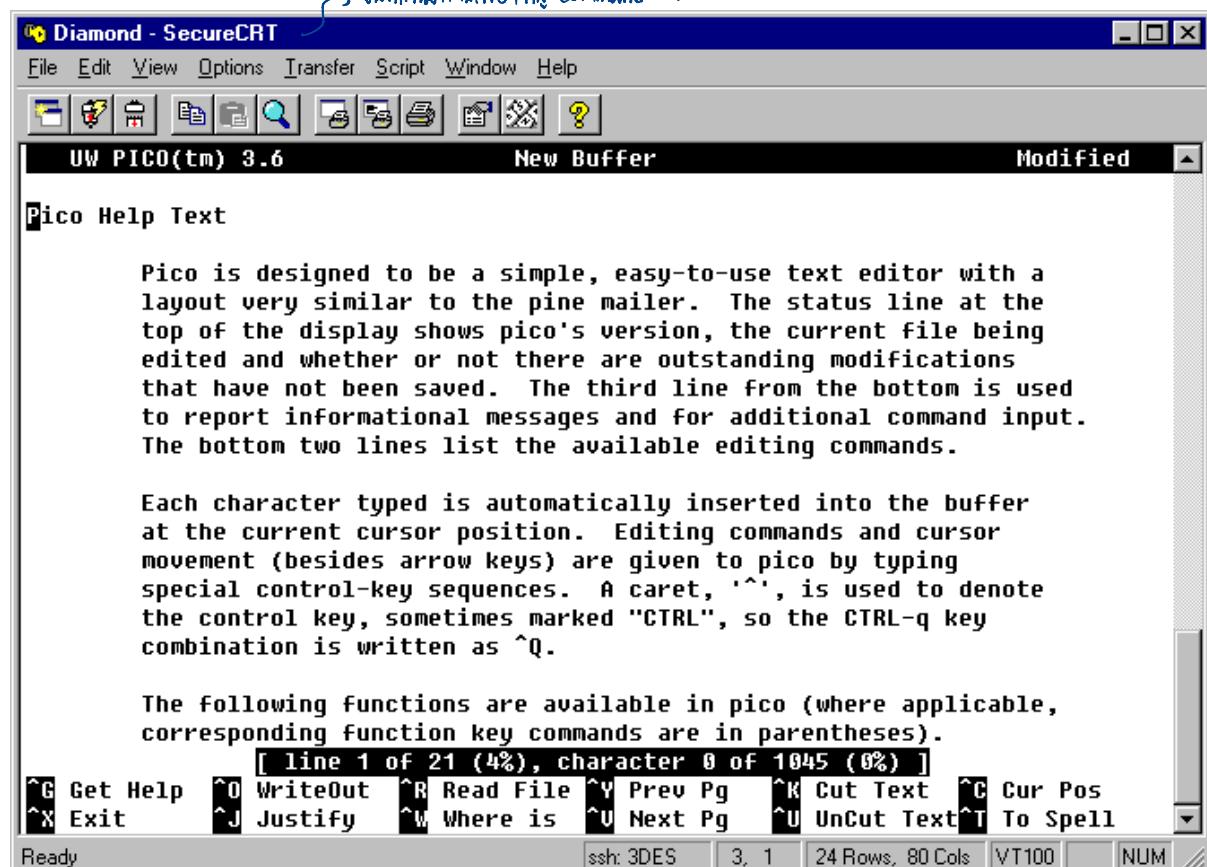
```
$ setenv TERM vt100
```

การใช้งานエディเตอร์ pico မြန်မြန်

မြန်မြန် Nano

pico เป็นเอดิเตอร์แบบสก्रีนซึ่งมาพร้อมกับโปรแกรมรับส่งอีเมล pine จัดทำขึ้นโดย University of Washington และว่าかもสามารถใช้งานได้ตั้งแต่ปี 1990 เดือนตุลาคม แต่ด้วยความเรียบง่ายจึงทำให้เป็นที่นิยมสำหรับผู้ใช้มือใหม่และผู้ใช้ทั่วไป

→ โปรแกรมที่ใช้เพื่อ执行 command ของ pico



หน้าจอโปรแกรม pico

โดยทั่วไปแล้วพื้นที่เอกสารภายใน pico จะมีตั้งแต่บรรทัดที่ 3 เป็นต้นไปจนถึงก่อนสามบรรทัดสุดท้ายซึ่งเป็นส่วนแสดงผลและต้องตอบกับผู้ใช้ของ pico เช่น หากขนาดเทอร์มินอลเป็น 80x24 บรรทัด 24 และบังคับเอกสารชั้นของ pico ซึ่งมีฟังก์ชันที่กำหนดไว้ เช่น สำเนา粘性 แก้ไข ลบ และสถานะของบันทึกเอกสารว่ามีการแก้ไขหรือไม่ ต่อมาเป็นขอบเขตของเอกสารเริ่มที่บรรทัดที่ 3 และสิ้นสุดที่บรรทัดที่ 21 ในบรรทัดที่ 23-24 เป็นส่วนแสดงปุ่มด่วนต่างๆ ที่สามารถใช้งานได้กับสถานการณ์ปัจจุบัน และบรรทัดที่ 22 เป็นส่วนแสดงผลและต้องตอบกับผู้ใช้เป็นต้น

การเรียกใช้ ทำได้โดย

- เรียกโปรแกรม pico โดยตรงจากเซลล์พรมต์

\$ pico

- หากต้องการแก้ไขเพิ่มข้อมูลเท็กซ์ที่มีอยู่แล้วก็เรียกโปรแกรม pico แล้วตามด้วยชื่อไฟล์นั้นๆ

```
$ pico filename.txt
```

- หากต้องการสร้างไฟล์ข้อมูลเท็กซ์ใหม่ เรียกโปรแกรม pico และตามด้วยชื่อไฟล์ใหม่นั่นๆ

```
$ pico newfilename.txt
```

ปุ่มควบคุมสำหรับการเคลื่อนย้ายเคอร์เซอร์และหน้าเอกสาร

- ^B เคลื่อนเคอร์เซอร์ไปทางซ้ายเมื่อ หรืออาจใช้ปุ่มลูกศรซ้ายก็ได้
- ^F เคลื่อนเคอร์เซอร์ไปทางขวาเมื่อ หรืออาจใช้ปุ่มลูกศรขวาก็ได้
- ^P เคลื่อนเคอร์เซอร์ไปบรรทัดบน หรืออาจใช้ปุ่มลูกศรซึ่งบนก็ได้
- ^N เคลื่อนเคอร์เซอร์ไปบรรทัดล่าง หรืออาจใช้ปุ่มลูกศรซึ่งล่างก็ได้
- ^A เคลื่อนเคอร์เซอร์ไปยังต้นบรรทัด
- ^E เคลื่อนเคอร์เซอร์ไปยังท้ายบรรทัด
- ^Y เปลี่ยนไปยังหน้าก่อนหน้า
- ^V เปลี่ยนไปยังหน้าถัดไป

ปุ่มควบคุมสำหรับการแก้ไขเอกสาร

- ^C แสดงตำแหน่งปัจจุบันและข้อมูลเอกสาร ณ ส่วนแสดงผลและให้ต่อไปนี้ใช้
- ^D ลบอักษรที่เคอร์เซอร์ หรืออาจใช้ปุ่ม Delete ก็ได้
- ^H ลบอักษรด้านซ้ายเคอร์เซอร์ หรืออาจใช้ปุ่ม BackSpace ก็ได้
- ^I พิมพ์อักษรแทบ (tab) หรืออาจใช้ปุ่ม Tab ก็ได้
- ^J ปรับแต่งอย่างหน้าเอกสารให้สวยงาม หากต้องการยกเลิกการปรับแต่งกดปุ่ม ^P
- ^K ลบ/ตัดบรรทัดที่เคอร์เซอร์
- ^^ ทำเครื่องหมายจุดเริ่มต้นเพื่อตัดเป็นบล็อกข้อความด้วย ^K
- ^U เพิ่มบรรทัดที่ตัดด้วย ^K ครั้งล่าสุด ณ เคอร์เซอร์
- ^M พิมพ์อักษรขึ้นบรรทัดใหม่ (carriage return) หรืออาจใช้ปุ่ม Enter/Return ก็ได้
- ^T ตรวจคำสะกดภาษาอังกฤษ
- ^W หาคำ

ปุ่มควบคุมสำหรับการอ่าน/เขียนไฟล์ข้อมูล

- ^R สั่งให้ pico อ่านไฟล์ข้อมูลเท็กซ์ขึ้นมาประกอบกับเอกสารปัจจุบัน โดยแทรกที่เคอร์เซอร์ทั้งนี้อาจป้อนชื่อไฟล์โดยตรงหรือกด ^T เพื่อกราฟฟิกชื่อไฟล์ข้อมูลก็ได้ สำเกตได้จากส่วนแสดงปุ่มด้าน
- ^O จัดเก็บไฟล์ข้อมูลทันที โดย pico จะถามชื่อไฟล์ข้อมูลและมักปรากฏชื่อโดยปริยายไว้ให้ทั้งนี้อาจป้อนชื่อไฟล์โดยตรงหรือกด ^T เพื่อกราฟฟิกชื่อไฟล์ข้อมูลก็ได้ สำเกตได้จากส่วนแสดงปุ่มด้าน

ปุ่มควบคุมอื่นๆ

- ^G เข้าสู่ระบบช่วยเหลือที่สอดคล้องกับสถานการณ์ปัจจุบัน
- ^X ออกจากส่วนช่วยเหลือ หรือ ออกจาก pico โดยหากผู้ใช้ได้มีการแก้ไขไฟล์ข้อมูลและยังไม่ได้จัดเก็บ pico จะถามย้ำการจัดเก็บ ตอบ Y ถ้าต้องการจัดเก็บ และตอบ N เมื่อไม่ต้องการจัดเก็บ
- ^L วาดหน้าจอเทอร์มินอลใหม่ มักใช้ในการนีการแสดงผลผิดเพี้ยนหรือแสดงอักษรขยะขึ้นมา

ข้อสังเกต

- แม้ว่าในระบบ Unix นิกซ์ ตัวพิมพ์เล็กและตัวพิมพ์ใหญ่ถือว่าแตกต่างกันแต่สำหรับกรณีการกดปุ่มควบคุมแล้วให้ผลเหมือนกันไม่ว่ากดแป้นเป็นตัวพิมพ์เล็กหรือตัวพิมพ์ใหญ่ เช่น ^A หรือ ^a ถือว่าเหมือนกัน เป็นต้น
- ปุ่มควบคุมเดียวที่ยกเว้นอาจให้ผลต่างกันเมื่ออยู่ในสถานการณ์ที่ต่างกัน เช่น ^T ในส่วนการแก้ไข เป็นการสั่งตรวจสอบคำสะกด แต่ ^T ในส่วนการอ่าน/เขียนแฟ้มข้อมูล เป็นการบรรยายชื่อแฟ้มข้อมูล เป็นต้น ฉะนั้นจึงต้องใส่ใจในการใช้งานปุ่มควบคุมให้ดี
- โปรดระวังเรื่องการจัดเก็บข้อมูลทั่วไปในแฟ้มข้อมูลโดยไม่ได้ตั้งใจ
- ผู้ดูแลระบบไม่นิยมใช้ pico มักใช้ vi มากกว่า
- หากต้องการพิมพ์อักษรระหว่างห้องข้อความภาษาไทยซึ่งต้องใช้ ASCII แบบ 8 บิต pico สนับสนุนอยู่แล้ว แต่ต้องตั้งค่าให้เทอร์มินอลรองรับการใช้งาน ASCII แบบ 8 บิตด้วย (pragtic รับแค่ 7 บิต) โดยสั่งยกเลิกการละเลยบิตที่แปด

```
$ stty -istrip
```

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

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

หรือ สำหรับ csh

```
$ setenv TERM vt100
```

สรุปคำสั่งที่จำเป็น

alias	นิยามคำสั่งแบบย่อ
ar	จัดการไฟล์บาร์และดูแลแฟ้มข้อมูลซอร์สโค้ด
as	ตัวแปลภาษาแอสเซมบลี
awk	เทียบชูปแบบและประมวลผลถ้อยคำ
bash	บอร์นอะเกนเซลล์
bg	นำโปรแกรมไปทำงานแบบเบื้องกราวด์
cat	นำข้อมูลในแฟ้มแสดงออกເອົາຕຸພູມມາດວິສັນ
cc	คอมໄຟເລອຮກພາສະກື້າ
CC	คอมໄຟເລອຮກພາສະກື້າພັບສປລັສ
cd	เปลี่ยนໄດ້ເຮັດວຽບຈຸບັນ
chmod	เปลี่ยนສิทธີ່ແພີມຂໍ້ມູນທີ່ໄດ້ເຮັດວຽບ
chsh	เปลี่ยนເຊີລ໌ເວັ້ນເປັນຕົວຊື່ນ
clear	ล้างหน้าจอ
cmp	เปรียบเทียบແພີມຂໍ້ມູນສອງແພີມ
compress	ປັບອັດແພີມຂໍ້ມູນໂຍ່ງ່ງ່າຍ
cp	ສໍາເນາແພີມຂໍ້ມູນ
crontab	ຈັດຕາຮາງເວລາໂປຣແກຣມໃຫ້ທຳກຳເປັນປະຈຳໂດຍອັດໃນມັດ
crypt	ເຂົ້າຮັສທີ່ອັດດ້ວຍຮັສແພີມຂໍ້ມູນໂຍ່ງ່າຍ
csh	ຈີເຊີລ໌
cut	ຕັດຄອລັນຈຳກາແພີມຂໍ້ມູນ
date	แสดงວັນແລະເວລາ ລັ້າເປັນບັນຫຼຸງທີ່ຜູ້ໃຫ້ຮູດສາມາຮັດຕັ້ງວັນເລາໄໝມ້າດ້ວຍ
dbx	ດີບັກເກອຮະດັບຊອບສີໂຄ້ດ
df	ສູບເນື້ອທີ່ຈັດເກີບແພີມທີ່ເໜືອດູ້
diff	ເປົ້າຍົງເທິຍບໍ່ຂໍ້ມູນໃນແພີມເທິງ
du	ສູບເນື້ອທີ່ຈັດເກີບທີ່ເໜີໄປໃນໄດ້ເຮັດວຽບຈຸບັນທີ່ໄວະນຸ
echo	แสดงສາຍັກຂ່າວໄປຢັງເອົາຕຸພູມມາດວິສັນ
ed	ເທິກ້ອົງໂຄດີເຕອຮົງແບບປວກທັດ
eqn	ກຳທັນດຽນແບບສູດທາງຄນິຕສາສຕ່ຣີໃໝ່ໃນໂປຣແກຣມ troff
ex	ເທິກ້ອົງໂຄດີເຕອຮົງແບບປວກທັດ
exit	ອອກຈາກເຊີລ໌ ບໍ່ຮູ້ອອກຈາກຮະບບໃນກວນທີ່ທຳກຳນັ້ນ
export	ນຳຄ່າຕົວແປຣສາພແວດລ້ອມສໍານັກລົບເຊີລ໌ນັ້ນໃໝ່ກາລຍເປັນຄ່າຕົວແປຣສາພແວດລ້ອມກາລັງຂອງທຸກເຊີລ໌
f77	คอมໄຟເລອຮກພາສະກື້າຟອຣ໌ແກຣນ 77
fg	นำໂປຣແກຣມທີ່ເປັນແບັກກາງດົກລັບມາທຳກຳນັ້ນ
file	ຄາດເຕາຫິນດແພີມຂໍ້ມູນ

finger	ดูข้อมูลของประจำตัวผู้ใช้บนระบบ
ftp	โปรแกรมขนาดถ่ายแฟ้มข้อมูลผ่านเน็ตเวิร์ก
g++	คอมpile เครื่องภาษา C/C++ ของ GNU
gcc	คอมpile เครื่องภาษา C ของ GNU
gdb	ดีบักเกอร์ของ GNU
grep	ค้นหาข้อความในแฟ้มข้อมูล
gunzip	คลายบีบอัดแฟ้มข้อมูลของ GNU
gzip	บีบอัดแฟ้มข้อมูลของ GNU
head	แสดงส่วนหัวของแฟ้มข้อมูล
history	แสดงคำสั่งที่ป้อนไปทั้งหมด (ชีเซลล์ บอร์นอะเกนเซลล์ และคอร์นเซลล์)
ispell	โปรแกรมตรวจสอบการสะกดคำภาษาอังกฤษ
jobs	แสดงโปรแกรมทั้งหมดที่ทำงานอยู่
join	เชื่อมข้อมูลระหว่างแฟ้มแบบอินเทอร์เซกชัน
kill	ส่งสัญญาณหรือ指令 ให้โปรแกรมตามหมายเลขอรุ่น
ksh	คอร์นเซลล์
last	แสดงบันทึกเวลาและวันที่ล็อกอินของผู้ใช้ในระบบ
lint	ปรับแต่งซอฟต์แวร์ให้ดีกว่าเดิม
ln	สร้างลิงก์แฟ้มข้อมูลหรือไดเรกทอรี
logout	ออกจากระบบ
lpq	ตรวจสอบคิวเครื่องพิมพ์
lpr	พิมพ์แฟ้มข้อมูลออกเครื่องพิมพ์ที่กำหนดไว้
lprm	ลบงานออกจากคิวเครื่องพิมพ์
lynx	เว็บбраузอร์แบบเท็กซ์
mail	ส่งและรับจดหมายอิเล็กทรอนิกส์อย่างง่าย
make	ดูแลส่วนประกอบทั้งหมดของซอฟต์แวร์เพื่อแปลงเป็นโค้ดที่รันได้
man	ดูคู่มือคำสั่ง
mesg	อนุญาตหรือปฏิเสธเมสเซจจากเทอร์มินอลหรือคำร้องขอจากคำสั่ง talk และ write
mkdir	สร้างไดเรกทอรี
more	ดูข้อมูลในแฟ้มข้อมูลที่ละหน้าจอ
mv	ย้ายหรือเปลี่ยนชื่อแฟ้มข้อมูลหรือไดเรกทอรี
neqn	กำหนดรูปแบบสูตรทางคณิตศาสตร์ให้ในโปรแกรม nroff
netscape	เว็บбраузอร์แบบกราฟิก
netstat	แสดงสถิติของเครือข่าย
nice	เปลี่ยนระดับความสำคัญหรือไพรอริตี้ของโปรแกรม
nohup	บังคับให้โปรแกรม执行ต่อสัญญาณ SIGHUP เพื่อทำงานต่อไปแม้ว่าผู้ใช้ล็อกเอาต์จากระบบแล้ว
nroff	จัดรูปแบบตัวอักษรเพื่อใช้ในการแสดงผลอย่างสวยงาม
page	คล้าย more

passwd	เปลี่ยนพาสเวิร์ด
paste	แบ่งบรรทัดจากแฟ้มข้อมูล
perl	อินเทอร์พรีเตอร์ภาษาเพิร์ล
pic	สร้างรูปอย่างง่ายสำหรับอินพุตแก้ troff
pico	เอดิเตอร์แบบสก्रีน (ง่ายต่อการใช้)
pine	ส่งและรับจดหมายอิเล็กทรอนิกส์ (ง่ายต่อการใช้)
pr	พิมพ์แฟ้มข้อมูล
ps	แสดงสถานะของโปรแกรมบนระบบ
pwd	แสดงพาธที่อยู่ในเครื่องคอมพิวเตอร์ปัจจุบัน
quota	แสดงเนื้อที่และขีดจำกัดการใช้เนื้อที่จัดเก็บของผู้ใช้
rcp	คัดลอกแฟ้มข้อมูลผ่านเครือข่าย
rlogin	เข้าสู่ระบบระยะไกล
rm	ลบแฟ้มข้อมูล
rmdir	ลบไดเรกทอรี
rsh	สั่งให้โปรแกรมหรือชีล์ดทำงานระยะไกล
scp	คล้าย rcp แต่เพิ่มการเข้ารหัส
sed	เอดิเตอร์เทกซ์แบบสตรีม
set	ตั้งค่าตัวแปรสภาพแวดล้อมสำหรับชีล์ดนั้นๆ
setenv	ตั้งค่าตัวแปรสภาพแวดล้อมคงทาง มีเฉพาะชีล์ด หากชีล์ดอื่นต้องใช้คำสั่ง set และ export ควบคู่กัน
sh	บอร์นชีล์ด
sort	เรียงลำดับแฟ้มเทกซ์
spell	โปรแกรมหาคำภาษาอังกฤษที่สะกดผิด
ssh	ชีล์ดเชล์ด คล้าย rlogin และ rsh
stty	ตั้งค่าเทอร์มินอล
tail	แสดงส่วนท้ายของแฟ้มข้อมูล
talk	คุยกับผู้ใช้ที่อยู่บนระบบผ่านเครือข่าย
tar	สำรองข้อมูลลงไฟล์สำรองข้อมูลหรือแฟ้มข้อมูล
tbl	จัดรูปแบบตารางสำหรับโปรแกรม troff/troff
tee	แสดงเอกสารออกไปทางเอกสารพิมพ์และแฟ้มข้อมูล
telnet	เข้าสู่ระบบระยะไกลโดยการจำลองเทอร์มินอล
tr	แปลงอักษรตามตารางที่กำหนด
troff	จัดรูปแบบตัวอักษรเพื่อใช้ในการแสดงผลอย่างสวยงาม คล้าย nroff
umask	เปลี่ยนค่าบิริยาของสิทธิในการสร้างแฟ้มข้อมูลใหม่
uncompress	ขยายแฟ้มข้อมูลที่บีบอัดไว้อีกครั้ง
uniq	รายงานห้องลับบริหารที่ซ้ำกันในแฟ้มข้อมูล
unzip	คลายบีบอัดแฟ้มข้อมูลแบบ ZIP
uptime	แสดงภาวะของระบบและระยะเวลาตั้งแต่ระบบเริ่มทำงาน

vi	เอดิเตอร์แบบสก्रีน (มีให้ใช้ในทุกๆ ระบบยูนิฟอร์ม)
w	แสดงผู้ใช้ที่อยู่บนระบบขณะนี้ และดูว่ากำลังใช้งานคำสั่งใดอยู่
wc	นับจำนวนบรรทัด คำ หรือ อักขระในแฟ้มข้อมูล
whereis	ค้นหาแฟ้มโปรแกรมที่ต้องการในระบบแฟ้มข้อมูลตามพาร์ช
who	แสดงผู้ใช้ที่กำลังใช้บนระบบ
write	ส่งเมล์เสียด่วนให้ยูสเซอร์ที่อยู่บนระบบ
zcat	แสดงข้อมูลในแฟ้มข้อมูลที่บีบอัดไว้ คล้ายคำสั่ง cat
zip	บีบอัดแฟ้มข้อมูลแบบ ZIP