



BASH
THE BOURNE-AGAIN SHELL

เพื่อ มือใหม่
หัดใช้ LINUX

เข้าใจง่าย รวดเร็ว
ใช้ในชีวิตประจำวัน

 โดย RouEmpire

คำนำ

SAMPLE FOR QA

ปรึกษาให้เรา

SAMPLE FOR QA

ไรด์แมพ

SAMPLE FOR QA

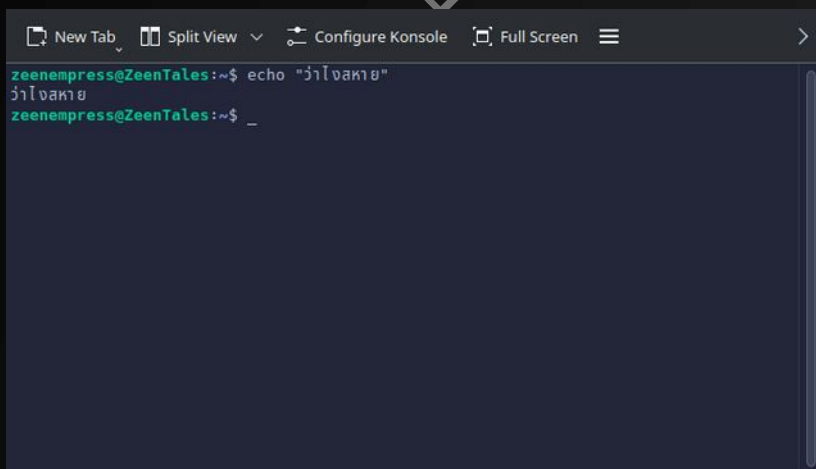
สารบัญ

SAMPLE FOR QA

บทที่ 0

BASH คืออะไร

BASH คือภาษาที่ใช้ใน Terminal ของ Linux ส่วนใหญ่ มีที่มาจาก Shell ซึ่งใช้กับ Linux ยุคก่อน ก่อนที่จะพัฒนามาเรื่อยๆ จนเป็น BASH (Bourne-Again SHell) ที่เรารู้จักกันมาจนถึงทุกวันนี้

A screenshot of a terminal window with a dark blue background. The window has a title bar with buttons for 'New Tab', 'Split View', 'Configure Konsole', and 'Full Screen'. The terminal shows a prompt 'zeenempres@ZeenTales:~\$' followed by the command 'echo "ว่าใจหาย"', the output 'ว่าใจหาย', and another prompt 'zeenempres@ZeenTales:~\$' with a cursor. A vertical scrollbar is visible on the right side of the terminal area.

```
zeenempres@ZeenTales:~$ echo "ว่าใจหาย"
ว่าใจหาย
zeenempres@ZeenTales:~$ _
```

*ภาพของ Terminal ที่กำลัง run คำสั่งง่ายๆ

สำหรับการใช้ภาษา **BASH** นั้น จะต้องเรียกใช้ Terminal Emulator¹ ซึ่งแต่ละ OS ก็จะมี Terminal ที่แตกต่างกัน มีฟีเจอร์ และรูปลักษณ์ต่างกัน แต่ส่วนใหญ่แล้วมักจะใช้ **BASH** เป็นภาษาที่ใช้ ป้อนคำสั่งเป็นหลัก

ในบทต่อไป จะมีการสอนวิธีตรวจสอบภาษาที่ Terminal นั้นใช้อยู่

| ตัวอย่างระบบที่ใช้ ² | | ชื่อ | GNOME Terminal |
|---|---|--------|-------------------------------|
|  |  | ข้อมูล | พบได้ใน linux ที่ใช้ Gnome DE |
|  |  | ชื่อ | Konsole |
|  |  | ข้อมูล | พบได้ใน Linux ที่ใช้ KDE |
|  |  | ชื่อ | Xfce Terminal |
|  |  | ข้อมูล | พบได้ใน linux ที่ใช้ Xfce DE |
|  |  | ชื่อ | QTerminal |
|  |  | ข้อมูล | พบได้ใน linux ที่ใช้ LXQt DE |

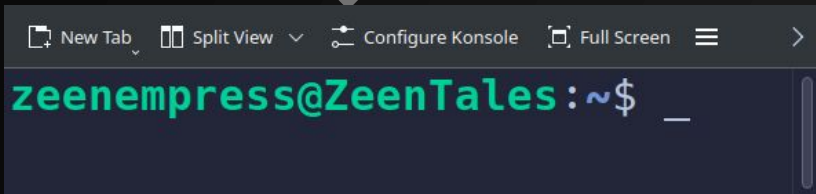
- 1 หมายถึงโปรแกรม Terminal สำหรับคอมพิวเตอร์ยุคใหม่ ซึ่งมีที่มาจาก Terminal แบบเดิมที่ใช้กับจอที่มีแต่ตัวอักษร และควบคุมคอมพิวเตอร์โดยการป้อนคำสั่งเท่านั้น
- 2 Linux เป็นแค่ Kernel ไม่ใช่ระบบปฏิบัติการ จำเป็นต้องมีระบบปฏิบัติการจริงๆ มาคลุมเอาไว้เพื่อให้ทำงานแบบมี GUI ได้

เราเปิดมาเจออะไรใน BASH?

เมื่อเราเปิด Terminal ขึ้นมา เราจะพบกับ Prompt ซึ่งประกอบไปด้วย 4 สิ่งนี้

- ◆ User Name
- ◆ Computer Host
- ◆ Directory
- ◆ User Type

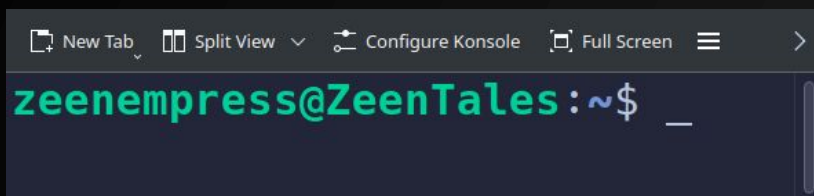
ดังภาพ



Syntax คือ

```
UserName@ComputerHost:~$
```


เราสามารถ Break Down ออกมาได้ดังนี้



The screenshot shows a terminal window with a dark background. The prompt is 'zeenempress@ZeenTales:~\$' in a light blue/green color. Above the terminal, there are window controls: 'New Tab', 'Split View', 'Configure Console', 'Full Screen', and a hamburger menu icon.

| | |
|-----------------|---|
| User@Host:~\$ _ | |
| User | User Name ที่ Login และใช้ Terminal นี้อยู่ |
| @ | ไม่มีความหมายอะไร แค่คั่นระหว่าง User กับ Host |
| Host | ชื่อคอมพิวเตอร์ที่เรากำลังใช้อยู่ |
| :~ | Directory ที่เรากำลังอยู่ ³ ในตอนนี้ |
| \$ | สถานะของผู้ login |
| _ | ไม่มีความหมายอะไร เป็นแค่พื้นที่สำหรับพิมพ์คำสั่ง |

- 3 ใน Linux คำว่า Directory มีความหมายตรงกับคำว่า Folder ในระบบปฏิบัติการอื่น ๆ ซึ่งหมายถึงแฟ้มที่เอาไว้เก็บไฟล์ข้างใน [และสามารถเก็บ Folder อื่นๆ ไว้ได้ด้วย]
- 4 ทุกครั้งที่เราใช้ Terminal เราจะมี **ที่อยู่** เป็นของตัวเอง กิจกรรมที่เราทำ และคำสั่งที่เราใช้ จะถูกทำ/สร้างขึ้นใน **ที่อยู่** ของเราในเวลานั้นๆ

อธิบายความหมายของ :~ เพิ่มเติม

| ตัวอย่าง Folder ที่อยู่ | Prompt ที่ปรากฏ |
|-------------------------|-----------------------------|
| home | User@Host:~\$ |
| home/Downloads | User@Host:~/Downloads\$ |
| home/Empress | User@Host:~/Empress\$ |
| home/Downloads/IDM | User@Host:~/Downloads/IDM\$ |

เช่น

The screenshot shows a terminal window with a dark background. The prompt is 'zeenempress@ZeenTales:~/EducationFlie/Book\$'. The window title bar includes 'New Tab', 'Split View', 'Configure Konsole', 'Full Screen', and 'Window Color Scheme'.

ตอนนี้เราอยู่ที่โฟลเดอร์ Book โดยมี Path คือ

home/EducationFlie/Book

อธิบายความหมายของ \$ เพิ่มเติม

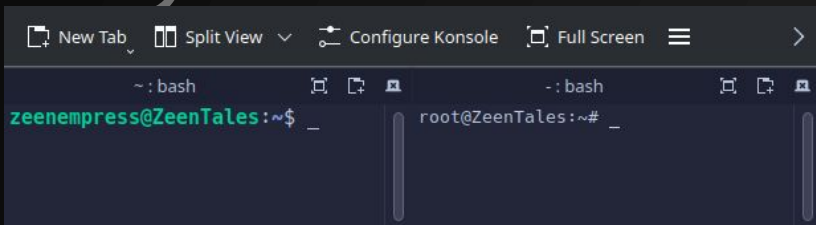
ใน Prompt ของ BASH นั้นจะมีสัญลักษณ์หนึ่ง วางอยู่ด้านหน้าของคำสั่งที่เรากำลังจะพิมพ์เสมอ นั่นคือสัญลักษณ์สำหรับการบอกสถานะของ User ที่กำลังพิมพ์คำสั่งอยู่ มักเรียกว่า *Prompt Character*

```
User@Host:~$
```

โดย Prompt Character นั้น สามารถเป็นได้ 2 ตัวอักษร คือ \$ และ #

\$ คือ สัญลักษณ์ที่บอกว่าเรากำลัง Login และใช้ Terminal นี้ อยู่ในฐานะ **RegularUser (General User)** ที่มีสิทธิ์ในการใช้คำสั่งเฉพาะบางคำสั่งตามที่ระบบกำหนด และเป็นค่า Default ของ Terminal ด้วย

คือ สัญลักษณ์ที่บอกว่าเรากำลัง Login และใช้ Terminal นี้ อยู่ในสิทธิ์ของ Root User หรือในอีกชื่อ **“SuperUser (Administrator)”** โดยเราจะมีสิทธิ์เข้าถึงทุกคำสั่งและทุกไฟล์ของทุก User ใน Host นั้นๆ



ด้านซ้าย คือ **Regular User** ด้านขวา คือ **Super User**

บทที่ 1

การย้ายไปมาด้วยคำสั่ง (CD)

จากภาพด้านขวา หากเราต้องการย้ายที่อยู่ของเราจากภาพด้านบน (home) ไปโฟลเดอร์ในภาพด้านล่าง (home/Downloads) เราจะต้องใช้คำสั่งอะไร?



```
zeenempres@ZeenTales: ~$
```

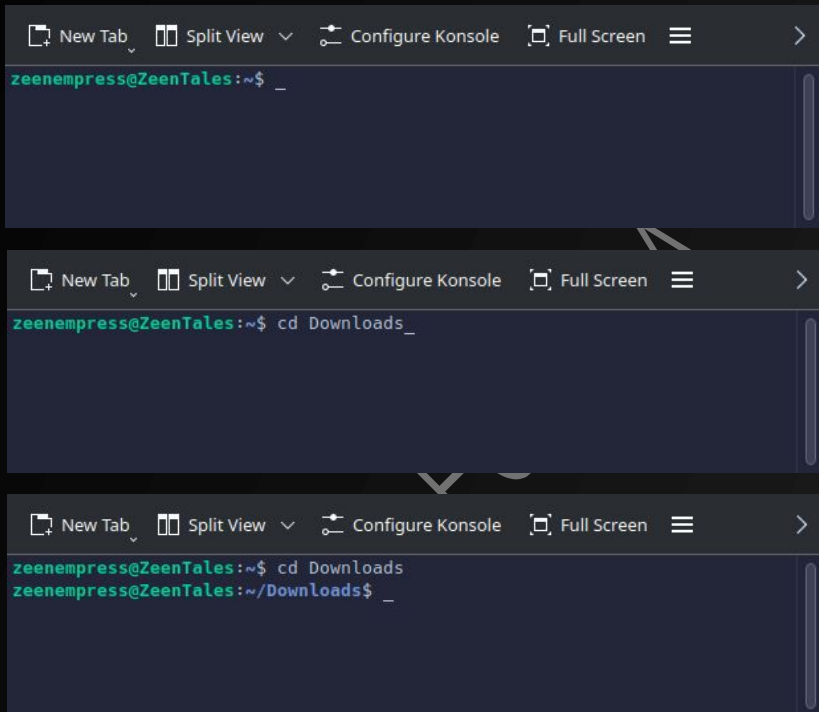


```
zeenempres@ZeenTales: ~/Downloads$
```

คำตอบคือ คำสั่ง “CD” ซึ่งย่อมาจาก Change Directory

```
User@Host: ~$ cd targetDirectory
```

ใส่คำสั่ง `cd` คั่นด้วยเว้นวรรค และใส่ตำแหน่ง หรือ ชื่อของโฟลเดอร์ที่ต้องการที่จะเข้าไปได้เลย จากนั้นให้กด Enter บนคีย์บอร์ด แล้ว Prompt ที่บอกตำแหน่งของเราจะเปลี่ยนตำแหน่งเป็นตำแหน่งใหม่ทันที



คำอธิบาย :

ภาพที่ 1 : เปิด Terminal ขึ้นมา

ภาพที่ 2 : ใส่คำสั่ง

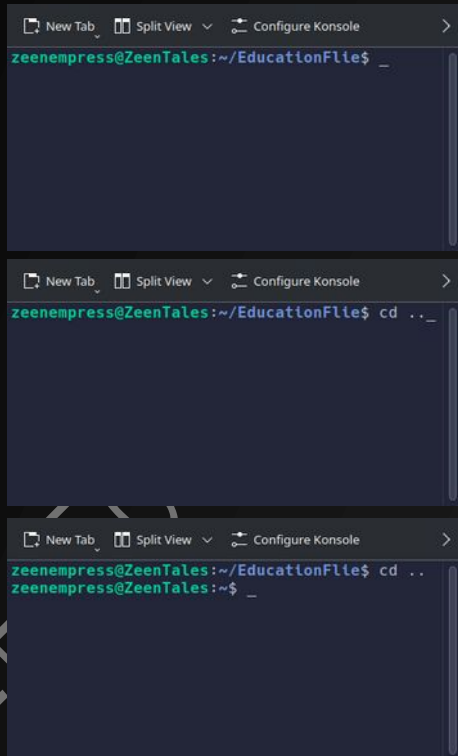
```
cd Downloads
```

ภาพที่ 3 : กด Enter จากนั้น User จะเข้าสู่ Folder ที่มีชื่อว่า home/Downloads และบรรทัดใหม่ที่รอคำสั่งใหม่จะโผล่ขึ้นมา

เราสามารถย้อนกลับไปยังโฟลเดอร์ที่แล้วได้ โดยการใช้คำสั่ง

```
cd ..
```

จะเห็นได้ว่า ภาพทางด้านขวานั้น เดิม ^{ภาพที่ 1} ที่เราจะอยู่ที่ `home/EducationFlies` แต่หลังจากที่เราใส่คำสั่งดังกล่าว ^{ภาพที่ 2} และกด Enter ตำแหน่งของเราจะเปลี่ยนไปเป็น `home` แทน ^{ภาพที่ 3}



เราสามารถใช่วิธีนี้ในการ Navigate (นำทาง) ไปยัง Directory ต่างๆ ตลอดทั้งเครื่อง Linux ของเราได้เลย ตราบได้ที่เราเป็นเจ้าของเครื่อง หรือ มีสิทธิ์ เทียบเท่า

- 5 สิทธิ์ [Permission] หมายถึงการได้รับการอนุญาตจาก Root ให้ทำสิ่งใด/ใช้คำสั่งอะไรได้บ้าง ซึ่งปกติแล้วเจ้าของเครื่องจะมีสิทธิ์สูงสุด รองลงมาจาก Root โดยอัตโนมัติ

เราสามารถย้ายไปที่โฟลเดอร์อื่นๆ ได้ในการใช้คำสั่งเดียว

| | |
|---------|---|
| แบบปกติ | U@H:~\$ <code>cd TargetDirectory</code> |
| แบบลึก | U@H:~\$ <code>cd TargetDirectory1/TargetDirectory2</code> |

The image shows three sequential screenshots of a terminal window. The terminal has a dark background with green text for the prompt and white text for the commands and output. The window title bar includes icons for 'New Tab', 'Split View', 'Configure Konsole', 'Full Screen', and a menu icon.

- First screenshot:** The prompt is `zeenempress@ZeenTales:~$` followed by a cursor.
- Second screenshot:** The command `cd EducationFlie/Book` has been entered. The prompt is `zeenempress@ZeenTales:~$`.
- Third screenshot:** The command `cd EducationFlie/Book` has been entered, and the prompt has changed to `zeenempress@ZeenTales:~/EducationFlie/Book$`.

เราสามารถใช้วิธีเดียวกันนี้กับการถอยกลับได้เช่นกัน

| | |
|-------------------------------|--|
| U@H:~\$ <code>cd ../..</code> | U@H:~\$ <code>cd ../../../../..</code> |
|-------------------------------|--|

ทางลัดของ CD

เพื่อให้ user จะสามารถ navigate ไปมาในคอมพิวเตอร์ได้รวดเร็วยิ่งขึ้น ในคำสั่ง `cd` จึงมี shortcut พิเศษต่างๆ ไว้สำหรับ user ที่ต้องการความรวดเร็วและความเข้าใจในการใช้ Terminal และมักจะมีประโยชน์มาในแทบทุกสถานะการณ์ที่ต้องใช้ Terminal ในการทำงานเป็นหลัก

ทางลัดกลับบ้าน : เป็นการใช้คำสั่ง `cd` เฉยๆ โดยที่ไม่มียอะไรต่อหลังเลย เมื่อกด Enter หลังพิมพ์คำสั่งนี้ Terminal จะพา User มาที่ home⁶ ทันที

```
cd
```

ทางลัด ~ : ให้ผลลัพธ์แบบเดียวกันกับ “ทางลัดกลับบ้าน”

```
cd ~
```

ทางลัด - : สลับไปมาระหว่างตำแหน่งปัจจุบันกับตำแหน่งที่แล้ว

```
cd -
```

ทางลัด / : ไปที่ตำแหน่ง root⁷ ทันที

```
cd /
```

6 home หรือ home directory เป็นตำแหน่งเริ่มต้นเมื่อ user เปิด Terminal ขึ้นมาครั้งแรก

7 มีอีกชื่อหนึ่งว่า root directory เป็นตำแหน่งที่ไกลที่สุดเท่าที่จะถอยไปได้

บทที่ 2

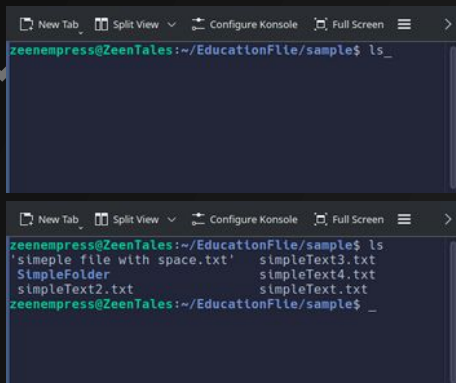
การ List ไฟล์

ถ้าหากว่าเราอยู่ใน Directory ใดๆ และอยากทราบว่าใน Directory นั้นมี File กับ Folder อะไรบ้าง เราจะทำอย่างไร?

การใช้คำสั่ง **ls** (ย่อมาจาก List) คือคำตอบ

```
User@Host:~$ ls
```

เมื่อเราใช้คำสั่ง **ls** จะพบได้ว่า Terminal จะ List รายชื่อ File และ Folder ที่มีอยู่ในที่อยู่ของเรา โดยตามที่ List มา เราสามารถแบ่งสิ่งที่ Terminal List มาให้ได้อยู่ 3 ประเภท



```
zeenempress@ZeenTales:~/EducationFile/sample$ ls_

zeenempress@ZeenTales:~/EducationFile/sample$ ls
'simple file with space.txt'  simpleText3.txt
simpleFolder                  simpleText4.txt
simpleText2.txt               simpleText.txt
zeenempress@ZeenTales:~/EducationFile/sample$ _
```

```

New Tab Split View Configure Konsole Full Screen
zeenempress@ZeenTales:~/EducationFile/sample$ ls
'Folder With Space'      SimpleFolder
'simple file with space.txt'  simpleText.txt
zeenempress@ZeenTales:~/EducationFile/sample$ _

```

จากภาพ เราจะแบ่งสิ่งที่ List ออกมาได้เป็น 3 สิ่ง ดังนี้

| | | |
|--|--------|---------------------------------|
| simpleFolder | ประเภท | Folder หรือ Directory |
| | ข้อมูล | ถูกไฮไลต์ด้วยสีพิเศษ |
| SimpleText.txt | ประเภท | File |
| | ข้อมูล | แสดงผลปกติ |
| 'File with space.txt' 'Folder With Space' | ประเภท | File/Folder ที่มีเว้นวรรคในชื่อ |
| | ข้อมูล | มีสัญลักษณ์ ' ครอบซ้ายขวา |

*ในทางเดียวกัน หากเราต้องการใช้คำสั่ง cd กับ Folder ที่มีเว้นวรรคในชื่อ เราจะต้องใส่เครื่องหมาย ' หรือ " ครอบไว้ด้วย

| | |
|-----|-------------------------------------|
| ถูก | User@Host:~& cd 'Folder With Space' |
| ถูก | User@Host:~& cd "Folder With Space" |
| ผิด | User@Host:~& cd Folder With Space |

เมื่อ User ใช้คำสั่ง `ls` ตัว Terminal จะ list มาให้แค่ไฟล์และโฟลเดอร์ปกติเท่านั้น หากอยากให้ Terminal แสดงไฟล์กับโฟลเดอร์ที่ซ่อนอยู่⁸ เราจะต้องเติม Flag⁹ ที่มีชื่อว่า `-a` (ย่อมาจาก `--all`) ไว้ข้างหลัง

```
User@Host:~$ ls -a
```

```
User@Host:~$ ls --all
```

The first screenshot shows the command `ls -a` being executed in a terminal window. The output lists hidden files and folders: `.`, `..`, `'Folder With Space'`, and `.PRAYUT_CHAN_O_CHA.mp4`. It also shows a file named `'simeple file with space.txt'` and a folder named `SimpleFolder` containing `simpleText.txt`. The second screenshot shows the command `ls --all` being executed, which produces the same output as `ls -a`.

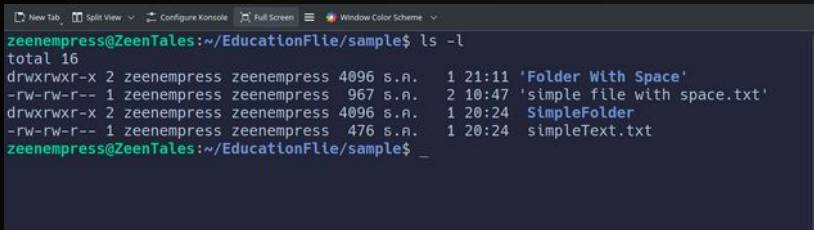
จะเห็นได้ว่า ไฟล์ `.PRAYUT_CHAN_O_CHA.mp4` ที่ถูกซ่อนอยู่ได้ถูกแสดงผลขึ้นมาเมื่อใช้คำสั่งนี้ (การใช้คำสั่ง `ls --all` จะให้ผลแบบเดียวกัน)

8 ไฟล์/โฟลเดอร์ที่ซ่อนอยู่ หมายถึง ไฟล์หรือโฟลเดอร์ใดๆ ที่มีจุดนำหน้าชื่อ เช่น `.calcuator.exe` เป็นต้น ไฟล์เหล่านี้มักจะถูกซอฟต์แวร์ส่วนใหญ่ทำเป็นเหมือนไม่มีอยู่ หรือหาไม่พบ

9 หรืออีกชื่อคือ **Option** เป็นคำสั่งเสริมที่มักเอาไว้กำหนดการทำงานของคำสั่งหลัก

หากผู้ใช้ต้องการดูข้อมูลเพิ่มเติมของไฟล์นั้นๆ ใน Terminal ผู้ใช้สามารถใช้คำสั่ง `ls -l` ซึ่งจะเป็นการสั่งให้ Terminal แสดงรายชื่อของไฟล์ และ ข้อมูลอื่นๆ ของไฟล์นั้นๆ ด้วย

```
User@Host:~$ ls -l
```



```
zeenempres@Zeentales:~/EducationFile/sample$ ls -l
total 16
drwxrwxr-x 2 zeenempres zeenempres 4096 6.ค. 1 21:11 'Folder With Space'
-rw-rw-r-- 1 zeenempres zeenempres 967 6.ค. 2 10:47 'simple file with space.txt'
drwxrwxr-x 2 zeenempres zeenempres 4096 6.ค. 1 20:24 SimpleFolder
-rw-rw-r-- 1 zeenempres zeenempres 476 6.ค. 1 20:24 simpleText.txt
zeenempres@Zeentales:~/EducationFile/sample$
```

ข้อมูลที่แสดงขึ้นมาจะมีส่วน และอาจจะดูยากไปหน่อย แต่มีประโยชน์มากหากดูเป็น โดยจะมีการสอนวิธีการใช้ในบทต่อไป ของหนังสือเล่มนี้

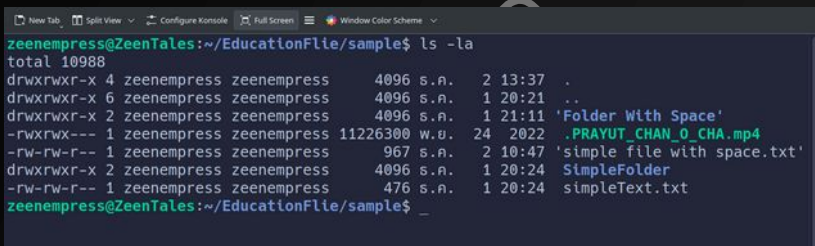
สำหรับตอนนี้ ให้ดูทางด้านขวาสุดของแต่ละแถว นับมาทีละลำดับ

- ลำดับที่ 1 : ชื่อไฟล์/โฟลเดอร์
- ลำดับที่ 2 : เวลาที่แก้ไขล่าสุด เรียงเป็น *เดือน วันที่ เวลา*
- ลำดับที่ 3 : ขนาดไฟล์ (มีหน่วยเป็น bytes) สำหรับโฟลเดอร์จะมีจำนวนเป็น 4096

ด้านหลังเครื่องหมาย – สามารถใส่หลายๆ คำสั่งเข้าไปพร้อมกันได้
หลายตัวอักษร ขึ้นอยู่กับความต้องการของ User

```
User@Host:~$ ls -al
```

คำสั่งด้านบนเป็นการรวมกันระหว่าง `-l` และ `-a` เข้าด้วยกัน กลายเป็น `-al` (สามารถเขียนสลับกันเป็น `-la` ได้) โดยจะเป็นคำสั่งที่ทำให้ Terminal แสดงไฟล์ทั้งหมด ไม่สนว่าไฟล์นั้นจะถูกซ่อนอยู่หรือไม่ และแสดงข้อมูลของไฟล์นั้นๆ ด้วย



```
zeenempres@ZeenTales:~/EducationFile/sample$ ls -la
total 10988
drwxrwxr-x 4 zeenempres zeenempres 4096 s.ค. 2 13:37 .
drwxrwxr-x 6 zeenempres zeenempres 4096 s.ค. 1 20:21 ..
drwxrwxr-x 2 zeenempres zeenempres 4096 s.ค. 1 21:11 'Folder With Space'
-rwxrwx--- 1 zeenempres zeenempres 11226300 w.ย. 24 2022 '.PRAYUT_CHAN_0_CHA.mp4'
-rw-rw-r-- 1 zeenempres zeenempres 967 s.ค. 2 10:47 'simple file with space.txt'
drwxrwxr-x 2 zeenempres zeenempres 4096 s.ค. 1 20:24 SimpleFolder
-rw-rw-r-- 1 zeenempres zeenempres 476 s.ค. 1 20:24 simpleText.txt
zeenempres@ZeenTales:~/EducationFile/sample$
```

จากภาพจะเห็นได้ว่า ไฟล์ที่ถูกซ่อนไว้จะถูกแสดงขึ้นมาตามแบบ
ของ `-a` และมีการให้ข้อมูลอื่นๆ ของไฟล์นั้นตามรูปแบบของคำสั่ง `-l` ด้วย

ในคำสั่ง `ls` นั้นไม่ได้มีแค่ `-l` กับ `-a` ที่เป็นตัวช่วยในการทำงาน
เพียงอย่างเดียว ยังมีตัวช่วยอีกมากมายที่เป็นประโยชน์ เราสามารถใช้คำ
สั่ง `man` เพื่อเปิด “Official Manual” ของคำสั่งนั้นๆ ได้ และในนั้นจะมี
ลิสต์ของ Flag ที่เราสามารถนำมาใช้ได้

```
User@Host:~$ man ls
```

บทที่ 3

การจัดการไฟล์ใน Terminal

การสร้างไฟล์

ใช้คำสั่ง **Touch** เพื่อสร้างไฟล์ใน Terminal ณ ตำแหน่งที่เราอยู่

```
touch fileName.txt
```

หรือใช้คำสั่งเดียวกัน สร้างไฟล์ในตำแหน่งอื่นๆ ที่ลึกกว่า

```
touch folder/folder/folder/fileName.txt
```

การสร้าง Folder/Directory

ใช้คำสั่ง **mkdir** (ย่อมาจาก **Make Directory**) เพื่อสร้าง Folder ในตำแหน่งที่เราอยู่ หรือสร้างในตำแหน่งที่ลึกกว่า แบบเดียวกันกับคำสั่ง Touch

```
mkdir folderName
```

```
mkdir folder/folder/folder/folderName
```

การย้ายไฟล์ / โฟลเดอร์

ใช้คำสั่ง **mv** (ย่อมาจาก **Move**)

```
mv file.txt destination
```

- file.txt หมายถึง: ไฟล์ที่ต้องการจะย้าย
- destination หมายถึง: ตำแหน่งปลายทางที่ต้องการจะย้ายไป

เราสามารถใช่วิธีเดียวกันนี้กับการย้ายโฟลเดอร์ได้เช่นกัน

```
mv sourceFolder destinationFolder
```

หากต้องการย้ายไฟล์หลายๆ ไฟล์พร้อมกัน ให้เขียนเรียงกันได้เลย

```
mv file1.txt file2.txt file3.txt  
file4.txt file5.txt destinationFolder
```

หากเราต้องการย้ายไฟล์จำนวนทั้ง Folder ให้ใช้เครื่องหมาย ***** ซึ่งหมายถึงการ Mention ทุกไฟล์และโฟลเดอร์ที่เราอยู่ในตำแหน่งนั้นๆ

```
mv * destinationFolder
```

เรายังสามารถดู Official Manual ด้วยคำสั่ง **man** ได้ด้วย แนะนำให้ใช้คำสั่งนี้ครั้งแรกเพื่อศึกษาเพิ่มเติมเกี่ยวกับวิธีใช้ขั้นสูง

```
man mv
```

การลบไฟล์ / โฟลเดอร์

เราใช้คำสั่ง `rm` (ย่อมาจาก `Remove`) ในการลบไฟล์

```
rm file.txt
```

```
rm file1.txt file2.txt file3.txt
```

หากเราต้องการลบโฟลเดอร์ และไฟล์อื่นๆ ทุกไฟล์ที่อยู่ในนั้น ให้เติม Option ที่มีชื่อว่า Recursive ลงไป โดยใช้ Flag `-r` หรือ `--recursive`

```
rm -r folderName
```

เมื่อเราใช้คำสั่ง `rm` คอมพิวเตอร์จะลบไฟล์/โฟลเดอร์ที่เราบอกไปทันที โดยที่ไม่มีการแจ้งเตือนใดๆ หากต้องการให้ Terminal ลิสต์สิ่งที่ลบไป ให้เติม Option ที่มีชื่อว่า Verbose ลงไป โดยใช้ Flag `-v` หรือ `--verbose` (แน่นอนว่าใช้ร่วมกับ `-r` ได้)

```
rm -v file1.txt file2.txt file3.txt
```

```
rm -vr folderName
```

การใช้คำสั่ง `rm` จะเป็นการลบไฟล์จากเครื่องไปอย่างถาวร

ไม่ใช่การย้ายไฟล์ไป "ถังขยะ"

โปรดคิดดีๆ ก่อนกด Enter นะ

การดูเนื้อหาไฟล์ด้วย “แมว”

เราสามารถแสดงเนื้อหาของไฟล์ใดๆ ที่เก็บข้อมูลเป็น Text ใน Terminal ได้ โดยการใช้คำสั่ง **cat** ดังนี้

```
cat fileName.txt
```

เพื่อให้อ่านไฟล์ดังกล่าวได้ง่ายขึ้น เรายังสามารถเพิ่ม Option ที่ช่วยให้อ่านสัญลักษณ์ที่มองไม่เห็นได้ง่ายขึ้น

Flag **-E**¹⁰ (ย่อมาจาก **--show-ends**) เมื่อใส่ Option นี้ ทุกๆ บรรทัดสุดท้ายของไฟล์จะมีเครื่องหมาย \$ ปรากฏขึ้น

```
cat -E fileName.txt
```

Flag **-T** (ย่อมาจาก **--show-tabs**) เมื่อใส่ Option นี้ ทุกๆ ที่ในไฟล์ที่มี “tab” จะถูกเขียนแทนด้วย ^I แทน

```
cat -T fileName.txt
```

Flag **-n** (ย่อมาจาก **--number**) เมื่อใส่ Option นี้ จะมีการแสดงตัวเลขหน้าบรรทัดของไฟล์นั้นๆ ว่า บรรทัดนั้นเป็นบรรทัดที่เท่าไร

```
cat -n fileName.txt
```

10 ในการเขียน Flag นั้น จะต้องดูให้ตัวว่าตัวที่ต้องการเขียนนั้นเป็น *ตัวอักษรพิมพ์ใหญ่หรือพิมพ์เล็ก* ทั้งสองแบบนี้มักจะมีฟังก์ชันต่างกันในบางคำสั่ง และเหมือนกันในบางคำสั่ง ให้เช็คที่ **Official Manual** ของคำสั่งนั้นๆ

การ Copy ไฟล์/โฟลเดอร์

ใช้คำสั่ง **cp** (ย่อมาจาก Copy) ดังนี้

```
cp fileName.txt folder
```

```
cp fileName.txt folder1/folder2/folder3
```

หากต้องการ Copy โฟลเดอร์และไฟล์ทั้งหมดในนั้น ให้เพิ่ม Option ที่มีชื่อว่า Recursive ซึ่งใช้ Flag **-r** หรือ **--recursive**

```
cp -r SourceFolder DestinationFolder
```

เมื่อทำการ Copy ไฟล์ไปที่อื่น แล้วมีไฟล์ชื่อซ้ำกัน?

ต้องการเขียนทับ : ให้รันคำสั่งได้เลย ไฟล์เก่าจะหายไปและไฟล์ใหม่จะมาแทนที่

ต้องการหลีกเลี่ยงการเขียนทับ : ใช้ Flag **-n** (ย่อมาจาก **--no-clobber**) คำสั่งที่ใช้ Option นี้จะไม่เขียนทับไฟล์นั้นกรณีที่มีไฟล์ที่ชื่อซ้ำกัน

ต้องการเขียนทับไฟล์เก่า¹¹ : ใช้ Flag **-u** (ย่อมาจาก **--update**) กรณีที่มีไฟล์ที่ชื่อซ้ำกัน เครื่องจะเช็คค่าไฟล์ที่จะเขียนทับนั้นเก่ากว่าหรือไม่ ถ้าเก่ากว่า ก็จะเขียนทับ ถ้าใหม่กว่า ก็จะไม่มีการเขียนทับ

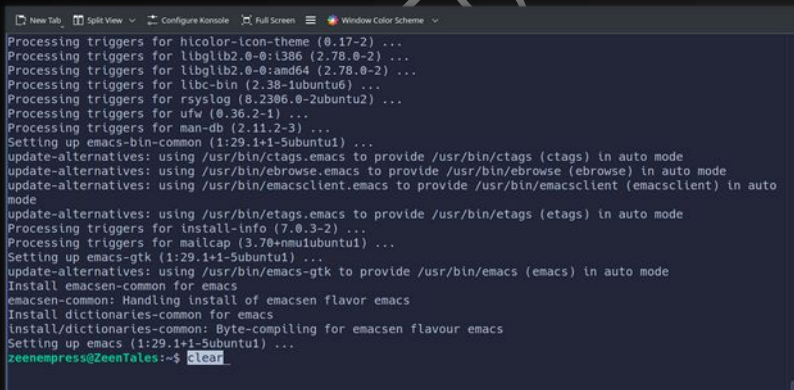
11 ไฟล์ที่มีเวลา/วันที่การแก้ไขครั้งสุดท้ายน้อยกว่า สามารถตรวจสอบได้โดยใช้คำสั่ง
`ls -l`

บทที่ 4

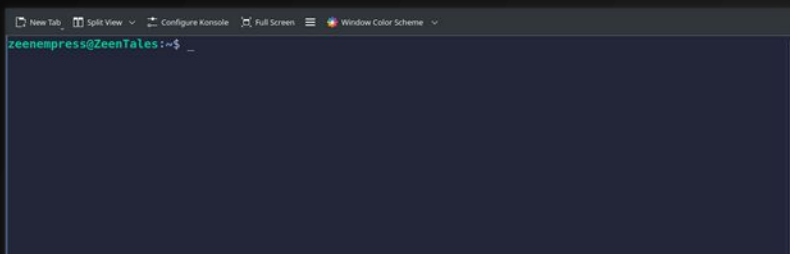
การจัดการ Terminal

ทำความสะอาด Terminal

ใช้คำสั่ง **clear** เพื่อ clear Terminal ให้สะอาดหมดจด



```
New Tab  Split View  Configure Konsole  Full Screen  Window Color Scheme
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for libgl1b2.0-0:amd64 (2.78.0-2) ...
Processing triggers for libgl1b2.0-0:amd64 (2.78.0-2) ...
Processing triggers for libc-bin (2.38-1ubuntu6) ...
Processing triggers for rsyslog (8.2306.0-2ubuntu2) ...
Processing triggers for ufw (0.36.2-1) ...
Processing triggers for man-db (2.11.2-3) ...
Setting up emacs-bin-common (1:29.1+1-5ubuntu1) ...
update-alternatives: using /usr/bin/ctags.emacs to provide /usr/bin/ctags (ctags) in auto mode
update-alternatives: using /usr/bin/ebrowse.emacs to provide /usr/bin/ebrowse (ebrowse) in auto mode
update-alternatives: using /usr/bin/emacscclient.emacs to provide /usr/bin/emacscclient (emacscclient) in auto mode
update-alternatives: using /usr/bin/etags.emacs to provide /usr/bin/etags (etags) in auto mode
Processing triggers for install-info (7.0.3-2) ...
Processing triggers for mailcap (3.70+nmu1ubuntu1) ...
Setting up emacs-gtk (1:29.1+1-5ubuntu1) ...
update-alternatives: using /usr/bin/emacs-gtk to provide /usr/bin/emacs (emacs) in auto mode
Install emacs-common for emacs
emacs-common: Handling install of emacs flavor emacs
Install dictionaries-common for emacs
install/dictionaries-common: Byte-compiling for emacs flavor emacs
Setting up emacs (1:29.1+1-5ubuntu1) ...
zeenempress@Zeentales:~$ clear
```



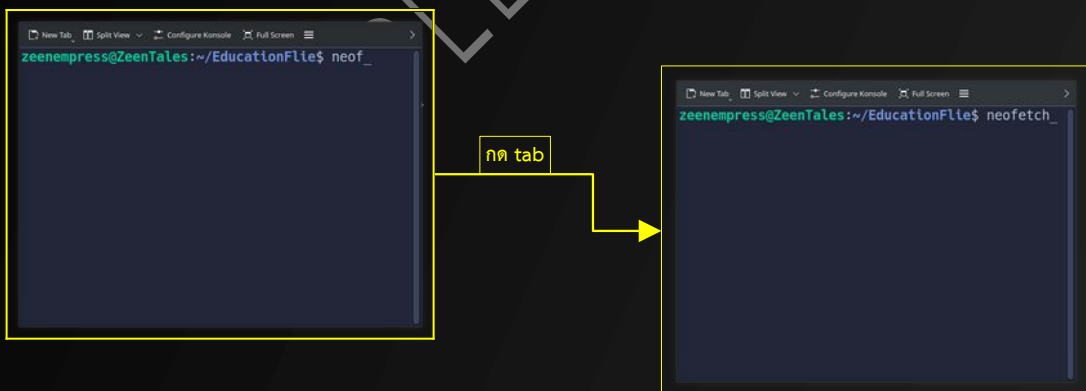
```
New Tab  Split View  Configure Konsole  Full Screen  Window Color Scheme
zeenempress@Zeentales:~$ _
```


การใช้ tab เพื่อ auto-fill

เมื่อเราต้องการพิมพ์คำสั่งยาวๆ ใน Terminal บางทีการพิมพ์คำสั่งให้ครบทุกตัวอักษรอาจจะทำให้การใช้คำสั่งของเราช้าโดยไม่จำเป็น การใช้ auto-fill จึงเป็นคำตอบ

ลองกดปุ่ม **tab** บนแป้นพิมพ์ขณะที่เราพิมพ์คำว่า “cle” เราจะได้พบว่า คำว่า “cle” จะถูกเปลี่ยนเป็น “clear” ทันที ซึ่งจะทำให้การพิมพ์คำสั่งของเราใน Terminal นั้นเร็วขึ้นมาก

แน่นอนว่าเราสามารถใช่วิธีเดียวกันนี้กับทุกคำสั่งที่มีใน Terminal ได้เลย เช่นการกด tab ขณะพิมพ์คำว่า “tou” จะทำให้คำสั้่งนั้นกลายเป็น “touch” ทันที



การใช้คำสั่ง “ที่แล้ว”

ให้ใช้ปุ่มลูกศร **ขึ้น** และ **ลง** บนคีย์บอร์ดเพื่อดูคำสั่งที่เพิ่งใช้ไป

สมมติว่าเราใช้คำสั่งไปแล้ว 3 คำสั่ง ตามลิสต์นี้

```
mkdir MyCode  
touch MyCode/index.html  
touch MyCode/style.css
```

ในการพิมพ์คำสั่งครั้งที่ 4 หากเรากดปุ่มลูกศร **ขึ้น** บนคีย์บอร์ด จะปรากฏ code ที่เขียนว่า `touch MyCode/style.css` ขึ้นมาทันที และหากเรากดปุ่มลูกศร **ขึ้น** อีกครั้ง code ที่เราเขียนไว้ก่อนหน้านี้ก็จะขึ้นมาอีก เป็น `touch MyCode/index.html` ซึ่งวิธีนี้ถือว่าเป็นวิธีที่ดีมากในการเขียน code ที่ต้องเขียนซ้ำๆ กันหลายๆ ครั้ง หรือต้องการย้อนกลับไปใช้คำสั่งเก่าและแก้ไขนิดหน่อยเพื่อที่จะส่งคำซ้ำคล้ายๆ กันกับแบบคำสั่งที่แล้ว โดยไม่จำเป็นต้องพิมพ์ใหม่ทั้งหมด

การกดปุ่มลูกศร **ลง** บนคีย์บอร์ด จะเป็นการทำให้ Terminal พาเรากลับมาที่ code เดิมของเรา (ที่เขียนล่าสุด)

Shortcut ใน Terminal

ใน Terminal นั้นมี Shortcut ที่ช่วยให้การพิมพ์คำสั่งของเราเร็วขึ้นอยู่อีกมาก เช่น

| | |
|-------------------------|---|
| Ctrl + A | : เลื่อน Cursor มาที่ด้านหน้าสุดของบรรทัด |
| Ctrl + E | : เลื่อน Cursor มาที่ด้านหลังสุดของบรรทัด |
| Ctrl + U | : ลบคำสั่งที่เขียนไป ตั้งแต่ตัวอักษรแรกจนถึง Cursor |
| Ctrl + K | : ลบคำสั่งที่เขียนไป ตั้งแต่ตัวอักษรสุดท้ายจนถึง Cursor |
| Ctrl + Shift + V | : วาง Text ที่อยู่ใน Clipboard ¹³ ลง Terminal |
| Ctrl + Shift + C | : คัดลอก Text ใน Terminal ลง Clipboard |
| Ctrl + + | : ขยายขนาดตัวอักษรใน Terminal |
| Ctrl + - | : ลดขนาดตัวอักษรใน Terminal |
| Ctrl + Shift + - | : Undo Shortcut ล่าสุดที่กดไป |
| Ctrl + W | : ลบตัวอักษรทางซ้ายของ Cursor ไปเรื่อยๆ จนกว่าจะเจอเว้นวรรค |
| Ctrl + D | : ออกจาก Terminal |

13 หมายถึงข้อความที่อยู่ในพื้นที่จัดเก็บเล็กๆ ที่ User ได้มาจากการกด Copy เช่น การใช้ **Ctrl + C** เป็นต้น

บทที่ 5

การแก้ไขไฟล์ Text

การแก้ไขไฟล์ด้วย nano

Nano (**Nano's ANOther Editor** หรือ **GNU nano**) คือซอฟต์แวร์ที่ทำให้เราสามารถแก้ไขไฟล์จำพวก Text ใน Terminal ได้เลยโดยไม่ต้องจำเป็นต้องเปิดแอปพลิเคชันอื่นๆ แยก

ก่อนที่จะเราจะแก้ไขไฟล์ด้วย **Nano** ได้ เราจะต้องมั่นใจก่อนว่าคอมพิวเตอร์ของเรามีซอฟต์แวร์ตัวนี้ติดตั้งอยู่ในเครื่อง

ให้พิมพ์คำสั่ง

```
nano --version
```

```
zeenempresse@ZeenTales:~/EducationFile/sample$ nano --version
GNU nano, version 7.2
(C) 2023 the Free Software Foundation and various contributors
Compiled options: --disable-libmagic --enable-utf8
zeenempresse@ZeenTales:~/EducationFile/sample$ _
```

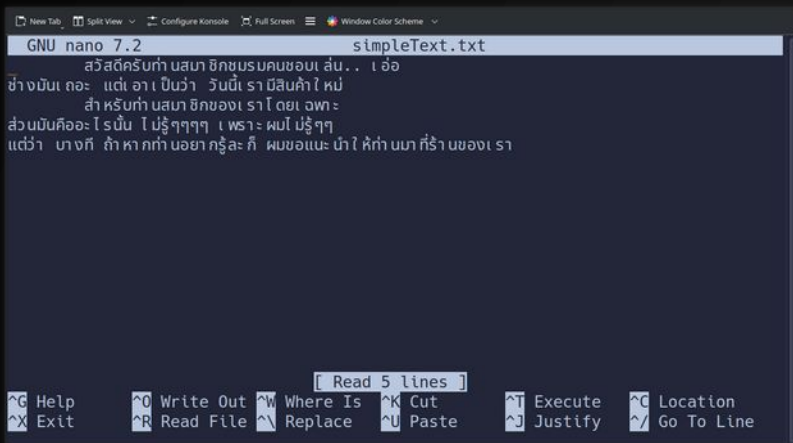
หากผลที่ได้ ไม่ใช่ Error แต่เป็นการบอกเวอร์ชันของ **nano** ในเครื่อง ก็หมายความว่าเราสามารถ
ใช้ **nano** ได้แล้ว

กรณีที่ยังไม่ได้ติดตั้ง ให้ดูบทที่ 7

วิธีการเปิดไฟล์ด้วย nano คือ

```
nano fileName.txt
```

โดยหน้าแรกหลังจากเปิดไฟล์มาคือหน้านี้



การใช้คำสั่ง **cat** จะเป็นการแสดง Text ในไฟล์บน Terminal แต่ **nano** จะแยกไปแสดงใน Terminal พิเศษ (แต่ยังคงอยู่ Console เดิม) โดยใน Terminal พิเศษนี้จะมีหน้าต่างดังนี้

- ด้านล่าง : Shortcut ที่ใช้ได้
- บนซ้าย : เวอร์ชันของ nano ที่ใช้อยู่
- บนกลาง : ชื่อไฟล์ที่กำลังดูอยู่

เมื่อผู้ใช้อยู่ในหน้านี้ ผู้ใช้สามารถใช้คำสั่ง **Ctrl + X** เพื่อออกจากหน้านี้ไปยัง Terminal ได้ หรือหากต้องการแก้ไข ผู้ใช้สามารถใช้ปุ่มลูกศรบนคีย์บอร์ดเพื่อ Navigate ไปมาในไฟล์ และอักขระต่างๆ บนแป้นพิมพ์เพื่อแก้ไขข้อความได้เลย

```

GNU nano 7.2 simpleText.txt *
สวัสดีครับทำสมาธิชมรมคนชอบเล่น... เอ้อ
ข้างมันละ แต่เอาเป็นว่าวันนี้เรามีสินค้าใหม่
สำหรับทำสมาธิของเราโดยเฉพาะ
ส่วนมันคืออะไรนั้น ไม่รู้จริงๆ เพราะผมไม่รู้ๆ
แต่ว่า บางที ถ้าหากทำนอยากรู้ละก็ ผมขอแนะนำไปทำสมาธิของเรา
เพราะว่าร้านของเรา นั้น มีแก้วที่อร่อยที่สุดในโลกรอทำนอย...

```

[^]G Help [^]O Write Out [^]W Where Is [^]K Cut [^]T Execute
[^]X Exit [^]R Read File [^]P Replace [^]U Paste [^]J Justify

ถ้าเกิดว่าเราต้องการจะออกจากหน้านี้โดยการกด **Ctrl + X** ตัว Terminal จะถามเราว่า “ต้องการที่จะ save ไหม” ถ้าหากต้องการ ก็ให้กด **Y (Yes)** บนแป้นพิมพ์ หากไม่ต้องการ ก็ให้กด **N (No)** จากนั้นเราจะเข้าโหมดตั้งชื่อไฟล์ (ในกรณีที่เรตอบ **Y (Yes)**) หากต้องการกลับไปแก้ไขอีกครั้ง ให้กด **Ctrl + C** แต่หากต้องการออกจากหน้านั้น (หน้าตั้งชื่อไฟล์) ก็ให้กด **Enter** แล้วเราจะกลับมาที่ Terminal ทันที

คำอธิบายของ 2 Mode

Normal Mode : โหมดนี้จะทำให้ user สามารถ navigate ได้ทั่วไฟล์ จัดการรูปร่างของไฟล์ ลบเนื้อหาบางอย่าง ใช้คีย์ลัด ใช้คำสั่งอื่นๆ

Insert Mode : โหมดนี้จะเน้นไปที่การเขียนเนื้อหาในไฟล์ล้วนๆ คีย์ลัดบางส่วนใช้ไม่ได้ในโหมดนี้

SAMPLE FOR QA

บทที่ 6

การติดตั้งโปรแกรม

การติดตั้งโปรแกรม/ซอฟต์แวร์ของ Linux แต่ละตัวนั้นจะแตกต่างกันเล็กน้อย ขึ้นอยู่กับว่า Distro ที่คุณใช้อยู่ นั้น Base on อะไร

บทที่ 7

เครื่องมืออรรถประโยชน์

ใน Terminal นั้น

SAMPLE FOR QA

บทพิเศษ

ลิสต์คำสั่งในหนังสือนี้

`echo "Hello World"`

แสดง คำ ว่า Hello World ใน Terminal

`ls`

ลิสต์ไฟล์/โฟลเดอร์ในตำแหน่งปัจจุบัน

`man`

ดู Official Manual ของคำสั่งอื่นๆ

`cd`

ย้ายตำแหน่งของ User

`touch`

สร้างไฟล์

`mkdir`

สร้างโฟลเดอร์

`cat`

แสดงเนื้อหาของไฟล์ใน Terminal

`mv`

ย้ายไฟล์/โฟลเดอร์

`rm`

ลบไฟล์/โฟลเดอร์

`cp`

Copy ไฟล์/โฟลเดอร์

```
clear
```

ลบข้อความใน Terminal ทิ้ง

```
neofetch
```

ดูข้อมูลของคอมพิวเตอร์ที่ใช้อยู่

```
nano
```

เปิด Nano Text Editor

```
vi
```

เปิด Vim Text Editor

```
sudo
```

ใช้นำหน้าคำสั่งที่ต้องการสิทธิ์ root

```
apt
```

ใช้จัดการแอปพลิเคชันในเครื่อง

```
poweroff
```

ใช้เพื่อปิดเครื่อง

```
systemctl reboot
```

ใช้เพื่อปิดเครื่อง และเปิดขึ้นมาใหม่

```
w3m
```

เรียกดูเว็บไซต์