



Intro to Bash [Linux command line]



เรียนรู้ Linux Command line เบื้องต้น ที่จำเป็น



Linux ต่างกับ Unix ยังไง

UNIX[®]
An Open Group Standard

หรือ *nix เป็น ระบบปฏิบัติการที่มีรากฐานมาจาก Unix



Bash คือ อะไร



Bash มาจากคำว่า Bourne-again shell ซึ่งเป็น Unix shell ประเภทหนึ่ง ที่นิยมใช้กันทั่วไป สำหรับ Unix และ Linux

Bash เป็นภาษา command language ที่ใช้ในการสั่งการระบบปฏิบัติการ ผ่านโปรแกรม terminal ซึ่งภาษาในตระกูลนี้จะเรียกรวม ๆ ได้ว่า **Shell** ซึ่งมีลักษณะเป็น command line Shell อื่น ๆ ที่นิยมใช้ นอกจาก Bash เช่น Zsh (ที่มากับ MacOS ในปัจจุบัน)



การใช้ Bash command line



- Bash command line จะใช้งานผ่านโปรแกรม Terminal ที่มากับเครื่อง สำหรับ Linux หรือ Mac อยู่แล้ว
- ถ้าใน Windows สามารถใช้ WSL (Windows Subsystem for Linux) หรือ Git Bash เพื่อใช้งาน Terminal ได้
- ถ้าใช้งานผ่านเว็บ Google Cloud Console จะสามารถใช้งาน Terminal ผ่าน Cloud Shell (>_) ได้
- สัญลักษณ์ประจำ Bash คือ \$ _ เพราะว่า ทุกคำสั่ง จะมี \$ ขึ้นต้นมาให้เสมอ แทนว่าบรรทัดนั้นคือ command ตัวอย่าง

```
$ ls  
README-cloudshell.txt
```



\$ command

Output

พิมพ์ command หลัง \$

output ของ command นั้น

การใช้งานแบบ command-line มีอีกชื่อหนึ่งเรียกว่า command-line interface หรือ CLI นั่นเอง

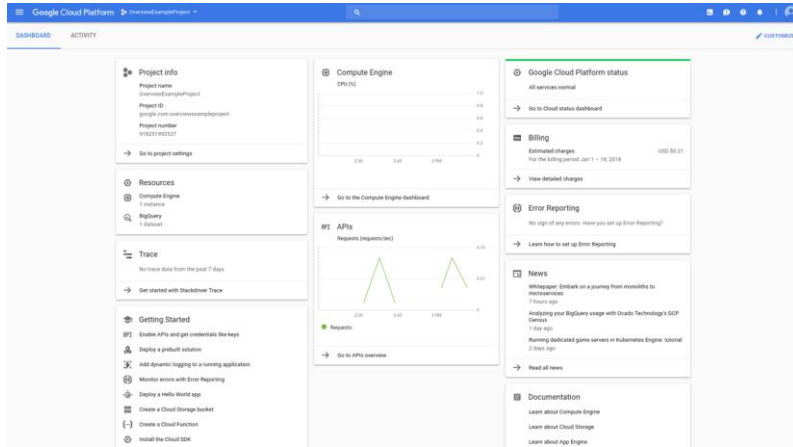


DATATH

มาเริ่มต้นใช้งาน Cloud Shell Terminal กัน

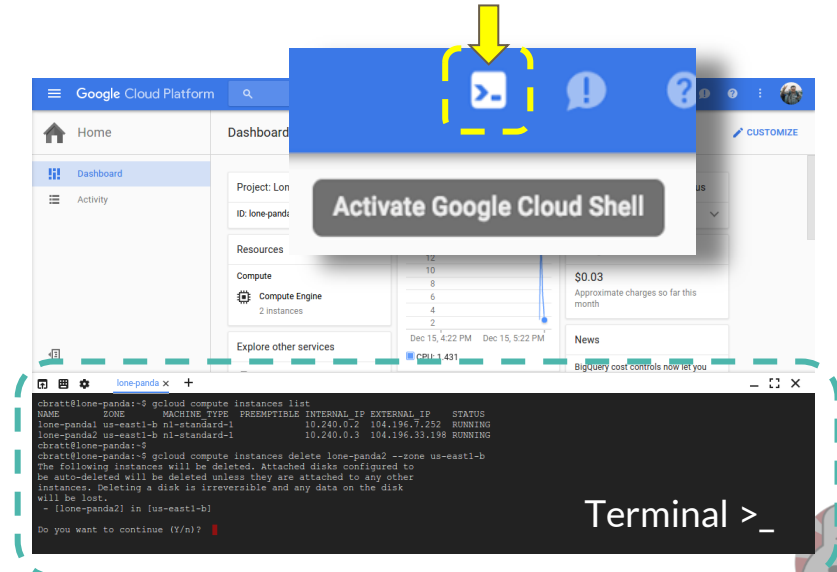
Cloud Console

หน้าเว็บ UI ของ Google Cloud
<https://console.cloud.google.com>



Cloud Shell

เครื่อง VM ชั่วคราว เพื่อใช้รัน command line



DATATH

คำสั่ง Bash ที่เราจะเรียนกันวันนี้



1. คำสั่งพื้นฐานสำหรับท่องโลก

- ls
- pwd
- cd

2. Print, สร้าง/อ่านไฟล์

- echo
- cat
- more / less
- touch

3. การสร้าง directory & การจัดการกับไฟล์

- mkdir
- cp
- mv
- rm

4. คำสั่งอื่น ๆ แถมเพิ่มเติม

- wget
- unzip / zip
- wc
- Bonus: man
- Bonus: วิธีออกจาก vim



1. คำสั่งพื้นฐาน: ท่องโลก terminal



BASH
THE BOURNE-AGAIN SHELL

- ls
- pwd
- cd

ls : list ดูว่ามีไฟล์อะไรบ้าง

คำสั่ง ls สำหรับ list file ใน directory ปัจจุบัน จากในตัวอย่างจะเห็นว่ามี 1 ไฟล์

```
$ ls
README-cloudshell.txt
```

คำสั่ง ls -l สำหรับ list file แบบละเอียดมากขึ้น ที่บอกถึง mode ของไฟล์ เจ้าของไฟล์ เวลาแก้ไขล่าสุด

```
$ ls -l
-rw-r--r-- 1 user user 913 Sep 1 16:08 README-cloudshell.txt
```

Mode ของไฟล์
เจ้าของ/กรุ๊ป/คนอื่นๆ

Owner ของไฟล์
Group ของไฟล์

ขนาดไฟล์
(byte)

วันที่ที่
แก้ไข

ชื่อไฟล์

-l = "long format"

d: directory
r: read
w: write
x: execute

ls -lah

เกร็ดเสริม -h หรือ human readable จะอ่านขนาดของไฟล์เป็น KB, MB, GB เพื่อความเข้าใจง่าย
-a หรือ all จะโชว์ไฟล์ที่ซ่อนอยู่ได้ โดยไฟล์ที่ซ่อน จะเป็นไฟล์ที่ขึ้นต้นด้วยจุด (.)



DATA TH

pwd : print working directory

คำสั่ง pwd สำหรับ print full path ว่าตอนนี้อยู่ที่ directory อะไร

```
$ pwd  
/home/username
```

โดยปกติแล้ว เมื่อเริ่มต้น จะเริ่มที่ **HOME (/home/username)** มักแทนด้วยสัญลักษณ์ ~ (tilde)

Full path ของระบบ Unix จะเริ่มที่ **/ (root)** เสมอ โฟลเดอร์ที่อยู่ภายใต้ root มักเป็น folder ที่เกี่ยวข้องกับระบบ และไม่สามารถแก้ไขได้ด้วยสิทธิ์ของ user ทั่วไป

Relative path คือ path ที่อ้างอิงจาก path ปัจจุบัน หรือ สามารถอ้างอิงจาก HOME (~) ได้เช่นกัน



cd : change directory

คำสั่ง cd สำหรับการเปลี่ยน directory ไปยัง path อื่นๆ คำสั่งนี้ มักใช้คู่กับ ls และ pwd

```
$ cd [PATH]
$ cd ~
`cd ~` or `cd` for back to home/mewteetawat
$ pwd
/home/username
```

สัญลักษณ์พิเศษที่ใช้บ่อย

~	HOME	(~ Tilde)
/	ROOT	(/ Slash)
.	Current Directory	
..	Parent Directory	

"cd -" is for back to previous

Home (~) คือ directory หลักของ user ใช้ในการอ้างอิง path

Root (/) คือ จุดเริ่มต้นของ path โดย full path จะเริ่มจาก / เสมอ

. แทน directory ปัจจุบันที่ใช้งานอยู่

.. แทน directory ก่อนหน้าขึ้นไปหนึ่งชั้น (เหมือนถอยออกจากโฟลเดอร์)



2. คำสั่งพื้นฐาน: การ print, สร้าง/อ่าน ไฟล์



BASH
THE BOURNE-AGAIN SHELL

- echo
- cat
- more / less
- touch

echo : print ออกมาบน Terminal

คำสั่ง echo สำหรับการแสดงข้อความหรือตัวแปรออกมาบน Terminal

```
$ echo "Hello World!"  
Hello World!
```

```
$ echo $USER  
username
```

username of system = mewteetawat

\$ to tell that it's string

เกร็ดเสริม:

\$Variable หรือ \${var}

\$ ใช้หน้าชื่อตัวแปร

คำสั่ง echo ยังใช้สำหรับการ print ข้อความ ลงไปต่อท้ายไฟล์ โดยใช้เครื่องหมาย **>>** ได้อีกด้วย

```
$ echo "Hello World! in a file" >> hello.txt  
$ ls  
README-cloudshell.txt      hello.txt
```

cat : concatenate อ่านไฟล์ออกมาบน Terminal

คำสั่ง cat สำหรับ print เนื้อหาในไฟล์ออกมา

CL in cmd is better for large scale data(VScode can't open big file, so >> is useful)

```
$ cat [FILE]...  
$ cat hello.txt  
Hello World! in a file
```

ที่เรียกว่า concatenate เพราะ ถ้าใส่ชื่อไฟล์หลายไฟล์ใน command เดียวกัน จะนำชื่อไฟล์ทั้งหมดมาต่อกัน และยังใช้ร่วมกับเครื่องหมาย >> เพื่อนำ output ไปใส่ไว้ในไฟล์(ต่อท้ายถ้าเป็นไฟล์ที่มีอยู่แล้ว/สร้างใหม่ถ้ายังไม่มี)

```
$ cat hello.txt hello.txt >> new_file.txt  
$ cat new_file.txt  
Hello World! in a file  
Hello World! in a file
```

```
cat hello.txt hello.txt README-cloudshell.txt >> new_file.txt  
cat new_file.txt
```

more / less : เปิดไฟล์ เป็นหน้า ๆ

บางครั้ง ถ้าไฟล์มีขนาดใหญ่ การใช้คำสั่ง cat สำหรับ print เนื้อหาไฟล์ จะอ่านยาก เพราะจะแสดงผลเนื้อหาของทั้งไฟล์ออกมา ในกรณีนี้ สามารถใช้คำสั่ง more หรือ less แทนได้

```
$ more [FILENAME]
```

```
$ less [FILENAME]
```

more new_file.txt

ทั้งสองคำสั่ง มีความคล้ายคลึงกัน คือ จะแสดงผลเนื้อหาทีละหน้า

- กด space bar เพื่อที่จะดูหน้าถัดไป
- กด q เพื่อออก

ข้อแตกต่างคือ more จะ print เนื้อหาลงใน terminal ทันที เมื่อจบ command จะเห็นเนื้อหาไฟล์นั้นอยู่ ส่วน less จะแยกการแสดงผลออกเป็นอีกหน้า ทำให้ย้อนดูเนื้อหาหน้าก่อนหน้าได้ด้วย และมีฟีเจอร์เยอะกว่า

less is able to arrow back and "q" to get out of terminal, no print in terminal(read only)



touch : สร้างไฟล์เปล่า

32:20

คำสั่ง touch สำหรับการสร้างไฟล์เปล่า ขึ้นมาหนึ่งไฟล์

```
$ touch [FILENAME]  
$ touch blank.txt  
$ cat blank.txt
```

เมื่อลอง cat คุณก็จะไม่เห็นอะไร เพราะว่าเป็นไฟล์เปล่านั่นเอง

คำสั่งนี้อาจจะไม่ค่อยได้ใช้มาก แต่ก็มีประโยชน์ในการสร้างไฟล์อย่างรวดเร็ว



คำสั่ง Bash ที่เราเรียนไปแล้ว



1. คำสั่งพื้นฐานสำหรับห้องโลก

- ls
- pwd
- cd

2. Print, สร้าง/อ่านไฟล์

- echo
- cat
- more / less
- touch

3. การสร้าง directory & การจัดการกับไฟล์

- mkdir
- cp
- mv
- rm

4. คำสั่งอื่น ๆ แถมเพิ่มเติม

- wget
- unzip / zip
- wc
- Bonus: man
- Bonus: วิธีออกจาก vim



3. คำสั่งพื้นฐาน: การสร้าง directory & การจัดการกับไฟล์



BASH
THE BOURNE-AGAIN SHELL

- mkdir
- cp
- mv
- rm

mkdir : make directory สร้างโฟลเดอร์ใหม่

คำสั่ง mkdir หรือ make directory สำหรับการสร้างโฟลเดอร์ใหม่

```
$ mkdir [DIRECTORY_NAME]
$ mkdir new_folder
$ ls
blank.txt  hello.txt  new_file.txt  new_folder  README-cloudshell.txt
```

เมื่อลอง ls ดู จะเห็น new_folder ที่เป็นตัวหนา และสีที่แตกต่าง บ่งบอกว่าเป็น directory

mkdir ยังสามารถสร้าง folder ที่ซ้อนกันได้หลายชั้น โดยใช้ -p (p: parent คือสร้าง parent ด้วยถ้าไม่มี)

```
$ mkdir -p new_folder/in/folder
```

d = directory

```
ls -l
drwxr-xr-x 2 mewteetawat mewteetawat 4096 Nov  6 21:54 new_folder
```

cp : copy ก๊อปปี้ไฟล์

คำสั่ง cp สำหรับ copy ไฟล์ จากต้นทางไปปลายทาง

ถ้าปลายทางเป็น directory สามารถ copy หลายไฟล์เข้าไปใน directory นั้นได้

```
$ cp [SOURCE]... [DESTINATION]
$ cp hello.txt new_folder/
$ ls new_folder
in hello.txt
```

มีหลายsourceได้

ถ้าต้องการ **copy โฟลเดอร์** ต้องใส่ -r (r: recursive) เพื่อให้ copy ไปยังทุก ๆ ไฟล์ในโฟลเดอร์ต้นทาง

```
$ cp -r new_folder/ new_folder2
$ ls new_folder2
in hello.txt
```

mv : move ย้าย / เปลี่ยนชื่อไฟล์ หรือ directory

คำสั่ง mv หรือ move ใช้สำหรับการย้ายไฟล์ หรือ directory ไปอีกที่หนึ่ง เราใช้ mv ในการเปลี่ยนชื่อได้ด้วย

```
$ mv [SOURCE] [DESTINATION]
$ mv new_file.txt new_folder/           # ย้ายไฟล์เข้าไปใน folder new_folder
$ mv new_folder2 good_name              # เปลี่ยนชื่อ folder เป็น good_name
$ ls
blank.txt  good_name  hello.txt  new_folder  README-cloudshell.txt
$ ls new_folder
hello.txt  in  new_file.txt
```

rm : remove ลบไฟล์ หายถาวร

คำสั่ง rm สำหรับ remove เพื่อลบไฟล์ออก

```
$ rm [FILENAME]
$ rm blank.txt
$ ls
$ good_name hello.txt new_folder README-cloudshell.txt
```

ถ้าต้องการลบ directory ต้องใช้ **-r** (r: recursive) เพื่อให้ไปลบไฟล์ย่อยข้างในด้วย

```
$ rm -r new_folder
$ ls
$ good_name hello.txt README-cloudshell.txt
```

4. คำสั่งอื่น ๆ เพิ่มเติม



BASH
THE BOURNE-AGAIN SHELL

- `wget`
- `unzip / zip`
- `wc`
- Bonus: `man`
- Bonus: วิธีออกจาก vim

wget : download ไฟล์

ctrl + R = reverse search
or just click up and down arrow button to
find used command line

คำสั่ง wget สำหรับ การ download ไฟล์จาก URL
-O ใช้สำหรับการตั้งชื่อไฟล์ที่จะโหลดมา

```
$ wget [-O output_name] [URL]
$ wget -O data.zip https://file.designil.com/bhXYol+
data.zip 100%[=====>] 6.39M 1.62MB/s in 3.7s
$ ls *.zip
data.zip
```

ls *.zip ตัว * คือ wildcard ที่ใช้แทน string อะไรก็ได้ ในที่นี้คือ หาไฟล์อะไรก็ได้.zip แล้วได้ผลลัพธ์
ออกมาเป็น data.zip

นอกจากคำสั่ง wget ที่ใช้ download file แล้ว

ยังมีคำสั่ง curl ที่ใช้ในการ download file และยิง API ได้เช่นกัน



DATA TH

curl ที่ใช้ในการ download file และยัง API ได้

whether download file or call API

wtrr.in is weather forecast API

```
mewteetawat@cloudshell:~ (my-project-bash-404007) $ curl wtrr.in
```

Weather report: not found

```
\ /      Partly cloudy
- /""-.  +13(12) °C
 \_( ) .  ↑ 9 km/h
 / ( ( )  16 km
          0.0 mm
```

Mon 06 Nov

Morning	Noon	Evening	Night
<pre>- /""-. Thundery outbr... _() . +7(4) °C / (() ↗ 13-15 km/h ⚡ ⚡ ⚡ 9 km \ \ \ \ 0.0 mm 0%</pre>	<pre>- /""-. Thundery outbr... _() . +8(5) °C / (() ↗ 15-17 km/h ⚡ ⚡ ⚡ 10 km \ \ \ \ 0.0 mm 0%</pre>	<pre>- /""-. Thundery outbr... _() . -1(-4) °C / (() → 6-14 km/h ⚡ ⚡ ⚡ 9 km \ \ \ \ 0.0 mm 0%</pre>	<pre>- .-. Light freezing () . -3(-6) °C (() ↗ 5-11 km/h * \ * 10 km * \ * 0.0 mm 0%</pre>

Tue 07 Nov

Morning	Noon	Evening	Night
<pre>\ / Sunny</pre>	<pre>\ / Sunny</pre>	<pre>- .-. Light freezing</pre>	<pre>- .-. Mist</pre>

unzip : แยกไฟล์ zip

คำสั่ง unzip สำหรับการแตกไฟล์ zip ด้วย command-line

```
$ unzip [ZIP_FILE]
$ unzip data.zip
Archive:  data.zip
  inflating:  ws2-output.csv

mewteetawat@cloudshell:~ (my-project-bash-404007)$ zip new_data.zip ws2-output.csv new_file.tx
t
  adding: ws2-output.csv (deflated 76%)
  adding: new_file.txt (deflated 74%)
mewteetawat@cloudshell:~ (my-project-bash-404007)$ ls
data.zip  new_data.zip  new_folder  ws2-output.csv
hello.txt new_file.txt  README-cloudshell.txt
```

zip : สร้างไฟล์ zip

deflated: how much it's got compressed in %

คำสั่ง zip สำหรับการสร้างไฟล์ zip ด้วย command-line

```
$ zip [ZIP_FILE.zip] [FILE]...
$ zip new_data.zip  ws2_data.csv
adding: ws2_data.csv (deflated 76%)
```

wc : word count นับจำนวนบรรทัด และจำนวนคำ

คำสั่ง wc สำหรับนับจำนวนบรรทัด, จำนวนคำ และ จำนวน byte สามารถใส่ชื่อไฟล์ได้หลายไฟล์

```
$ wc [FILE]...
```

```
$ wc ws2-output.csv
```

```
500001 1126750 29570221 ws2-output.csv
```

จำนวน line
(บรรทัด)

จำนวน word
(คำ)

จำนวน byte

การใช้ wc ในการนับจำนวนบรรทัดเท่านั้น

```
$ wc -l ws2-output.csv
```

```
500001 ws2-output.csv
```

-l คือ แสดงผลแค่ line count ว่าไฟล์มีกี่บรรทัด



DATA TH

Bonus : man อยากรู้คำสั่งไหน เปิดดูคู่มือได้เลย

man มาจาก manual ใช้เพื่อดูคู่มือของคำสั่ง นั้น ๆ ได้ เมื่อดูเสร็จ กด q เพื่อ ออก

```
$ man [COMMAND]
```

ตัวอย่าง

```
$ man wc
```

```
WC(1) User Commands

NAME
    wc - print newline, word, and byte counts for each file

SYNOPSIS
    wc [OPTION]... [FILE]...
    wc [OPTION]... --files0-from=F

DESCRIPTION
    Print newline, word, and byte counts for each FILE, and a total line if more than one FILE is specified.
    acters delimited by white space.

    With no FILE, or when FILE is -, read standard input.

    The options below may be used to select which counts are printed, always in the following order: newline,

    -c, --bytes
        print the byte counts

    -m, --chars
        print the character counts
```



DATA TH

Bonus : ออกจาก vim ยังไง

TIPS: click Tab for predict full text or command

vim คือ โปรแกรม text editor ใน terminal ที่มีชื่อเสียงโปรแกรมหนึ่ง พัฒนามาจาก vi (vim - Vi IMproved) หนึ่งในชื่อเสียงของ vim คือ เข้าแล้วออกไม่ได้

วิธีการเข้า

```
$ vim hello.txt
```

วิธีการออก

```
<ESC> :q
```

ถ้าต้องการแก้ไขไฟล์ จากใน terminal ด้วยโปรแกรมที่ใช้งานง่ายกว่า vim

แนะนำโปรแกรมชื่อ nano (เวลาออก กด ctrl +x)



คำสั่ง Bash ที่เราเรียนกันไปแล้ว



1. คำสั่งพื้นฐาน สำหรับท่องโลก

- ls
- pwd
- cd

2. Print, สร้าง/อ่าน ไฟล์

- echo
- cat
- more / less
- touch

3. การสร้าง directory & การจัดการกับไฟล์

- mkdir
- cp
- mv
- rm

4. คำสั่งอื่น ๆ แถมเพิ่มเติม

- wget
- unzip / zip
- wc
- Bonus: man
- Bonus: วิธีออกจาก vim

