

# Linux Basics

อ.ดร.วัชรินทร์ สาระไชย




# การวัดผล

- >- วัดผลความเข้าใจเกี่ยวกับความรู้พื้นฐานคำสั่ง Linux เบื้องต้นของนักศึกษา ด้วยการถามตอบ
- >- วัดผลความเข้าใจด้วยการปฏิบัติของนักศึกษาโดยให้แบบฝึกปฏิบัติการติดตั้ง และการใช้งาน Git Repository และ Docker ได้
- >- วัดผลความเข้าใจพื้นฐานคำสั่ง Linux command โดยภาพรวมด้วยการให้นักศึกษาฝึกปฏิบัติ

# Topic

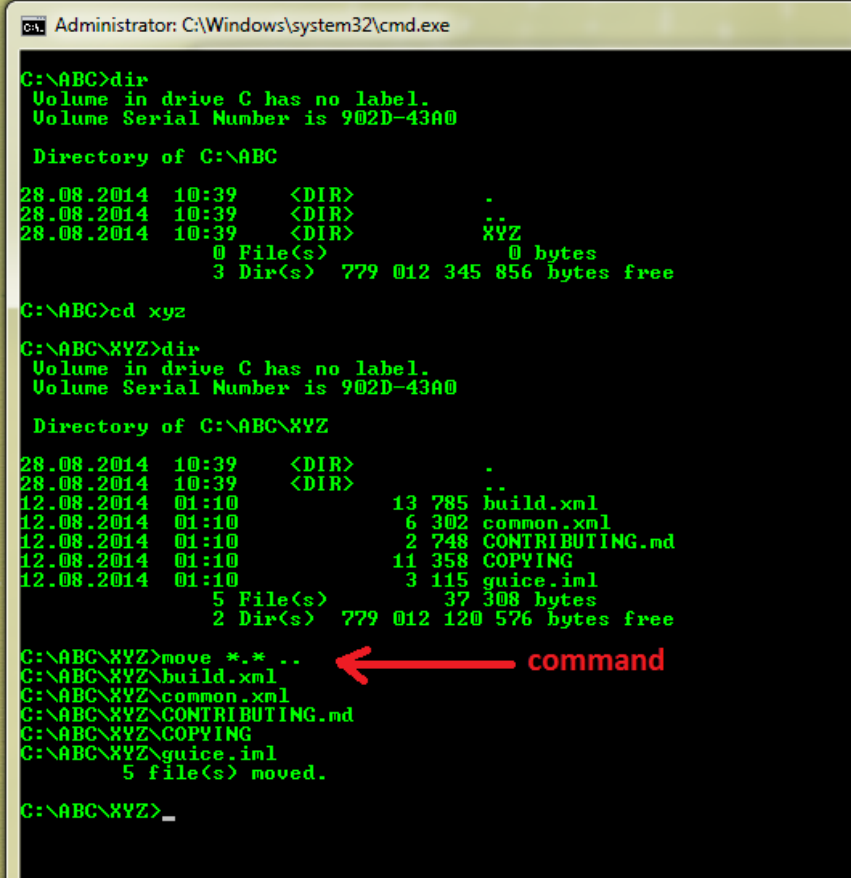
- >- คำสั่ง Linux command line สำหรับ beginners
- >- การใช้ Terminal
- >- Command Input และ Output

# Command Line Overview

-  Linux command line คือ text interface บนคอมพิวเตอร์ของเรา  
 เรา รู้จักในชื่อต่าง ๆ กัน เช่น shell, terminal, console, prompt เป็นต้น
-  ระบบปฏิบัติการที่ใช้ Linux command line เช่น Android, Arch Linux, Centos, Debian, Elementary OS, Fedora Linux, Gentoo Linux, หรือ Kali Linux เป็นต้น

# A brief history lesson

- >- ระบบปฏิบัติการ Unix เป็นต้นกำเนิดของระบบปฏิบัติการ Linux
- >- Unix ออกแบบมาให้ทำงานแบบ multi-user สำหรับ mainframe computers โดยผู้ใช้เชื่อมผู้ใช้สามารถเชื่อมต่อระยะไกลผ่าน Terminal
- >- การใช้ Terminals คือการติดต่อผ่านระบบข้อความ (ไม่มีรูปภาพ ไม่มีเมาส์ ไม่มีสี) โดยพิมพ์ข้อความด้วยคีย์บอร์ด
- >- การสื่อสารระหว่างผู้ใช้และคอมพิวเตอร์เป็นการส่งข้อความแบบ Text และรับข้อมูลกลับมาแบบ Text
- >- ทุก ๆ โปรแกรมที่ทำงานบนคอมพิวเตอร์จะรับ input เป็น Text และแสดงผลลัพธ์เป็น Text เท่านั้น



```
Administrator: C:\Windows\system32\cmd.exe
C:\ABC>dir
Volume in drive C has no label.
Volume Serial Number is 902D-43A0

Directory of C:\ABC
28.08.2014 10:39 <DIR>      .
28.08.2014 10:39 <DIR>      ..
28.08.2014 10:39 <DIR>      XYZ
               0 File(s)          0 bytes
               3 Dir(s)      779 012 345 856 bytes free

C:\ABC>cd xyz

C:\ABC\XYZ>dir
Volume in drive C has no label.
Volume Serial Number is 902D-43A0

Directory of C:\ABC\XYZ
28.08.2014 10:39 <DIR>      .
28.08.2014 10:39 <DIR>      ..
12.08.2014 01:10          13 785 build.xml
12.08.2014 01:10           6 302 common.xml
12.08.2014 01:10          2 748 CONTRIBUTING.md
12.08.2014 01:10          11 358 COPYING
12.08.2014 01:10           3 115 guice.iml
               5 File(s)          37 308 bytes
               2 Dir(s)      779 012 120 576 bytes free

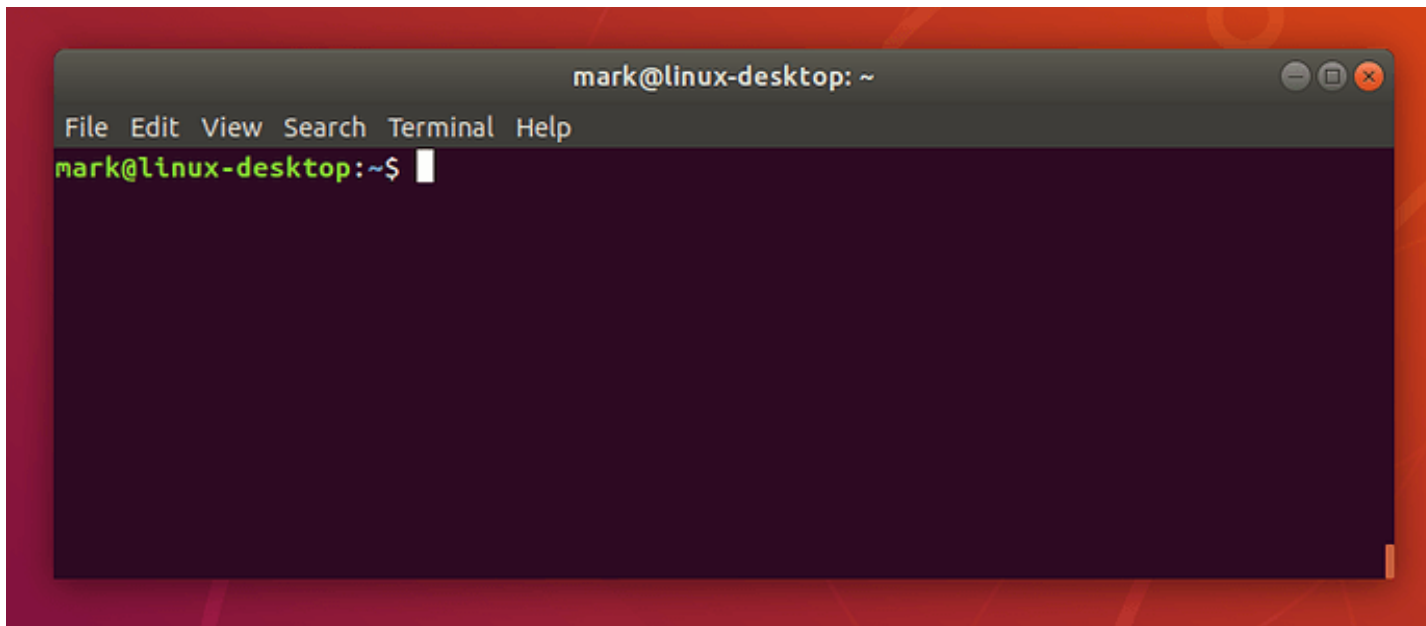
C:\ABC\XYZ>move *.* ..
C:\ABC\XYZ\build.xml
C:\ABC\XYZ\common.xml
C:\ABC\XYZ\CONTRIBUTING.md
C:\ABC\XYZ\COPYING
C:\ABC\XYZ\guice.iml
                    5 file(s) moved.

C:\ABC\XYZ>
```

A red arrow points to the command `move *.* ..` in the screenshot, with the word "command" written next to it.

# Opening a terminal

-  ในบทเรียนของเราจะอ้างอิงการใช้ Command line บนระบบปฏิบัติการ Ubuntu เป็นหลัก



# Run Our First Command

- >- เมื่อเริ่มต้นระบบจะอยู่ในสถานะพร้อมรับคำสั่งเราเรียกว่า “Prompt”

```
$ _
```

- >- พิมพ์คำสั่งแสดงที่อยู่ของ directory path ปัจจุบัน

```
$ pwd
```

```
.
└─ home
   └─ itsci ←
```








 เราจะเห็น directory path แสดงออกมา เช่น

```
$ pwd
/home/itsci
```

ข้อควรระวัง: คำสั่ง PWD  
และ pwd ไม่เหมือนกัน

หมายเหตุ: ในกรณีนี้ชื่อผู้ใช้ปัจจุบันก็คือ itsci

# Run Our First Command

-  คำสั่ง `pwd` เป็นคำย่อของ ‘**print working directory**’
  -  ‘**print working directory**’ คือพิมพ์ directory การทำงานปัจจุบัน
-  อะไรคือ working directory?
  -  แนวคิดสำคัญคือ กระบวนการเกี่ยวกับตำแหน่งเริ่มต้นที่จะดำเนินการกับไฟล์ใด ๆ ที่จะเกิดขึ้น
  -  หากเราสร้าง file หรือ directory ใหม่ การดู file ที่มีอยู่ หรือแม้แต่ลบออก ระบบปฏิบัติการจะถือว่าเรากำลังค้นหา file หรือ directory เหล่านี้ใน directory การทำงานปัจจุบัน เว้นแต่เราจะระบุให้เป็นอย่างอื่น
  -  ดังนั้นเป็นเรื่องสำคัญที่จะต้องทราบว่า directory ปัจจุบันอยู่ที่ไหน ไม่ว่าในช่วงเวลาใดก็ตาม การลบไฟล์ออกจาก directory ที่ไม่ถูกต้องอาจเป็นผลเสียหายได้
  -  หากต้องการทราบว่า directory ปัจจุบันอยู่ที่ไหนให้ใช้คำสั่ง `pwd` ซึ่งจะบอกเราว่า directory การทำงานปัจจุบันคืออะไร



# Navigate the File System

- >- เราสามารถเปลี่ยน directory ปัจจุบันโดยใช้คำสั่ง `cd` ย่อมาจาก ‘**C**hange **D**irectory’
- >- พิมพ์คำสั่งต่อไปนี้:

```
$ cd /  
$ pwd
```



- 🐧 หลังจากคำสั่ง working directory จะเป็น “/” บนระบบปฏิบัติการ Window จะใช้ “C:\”
- 🐧 “/” ปกติแล้วจะเป็น root directory หรือเป็น directory นอกสุดของระบบแฟ้มข้อมูล

หมายเหตุ: เครื่องหมาย (“/”) คือ directory separator เราเรียกว่า forward slash ส่วนในระบบปฏิบัติการ Window จะใช้ (“\”) เรียกว่า backslash ซึ่งจะไม่เหมือนกัน

# Navigate the File System

- >- จากคำสั่งก่อนหน้านี้เราอยู่ที่ root directory
- >- พิมพ์คำสั่งต่อไปนี้:

```
$ cd home  
$ pwd
```



-  เป็นคำสั่งย้ายไป parent directory (/home)
-  จากนั้นหากเราพิมพ์คำสั่ง `cd ..` ก็จะเป็นคำสั่งย้าย directory กลับขึ้นไปอีกระดับ ในกรณีนี้ก็คือ / (root directory) นั่นเอง

หมายเหตุ: เครื่องหมาย (“..”) คือ directory ก่อนหน้าที่อยู่ขึ้นไปอีก 1 ระดับ

# Navigate the File System

- พิมพ์คำสั่ง `cd` ณ. ตำแหน่งใด ๆ เป็นคำสั่งทางลัดให้กลับไปยัง home directory:
- พิมพ์คำสั่งต่อไปนี้:

```
$ cd  
$ pwd
```

```
.  
└─ home  
   └─ itsci ←
```

 เป็นคำสั่งย้ายไป parent directory (`/home/your_username`) จากที่ไหนก็ได้

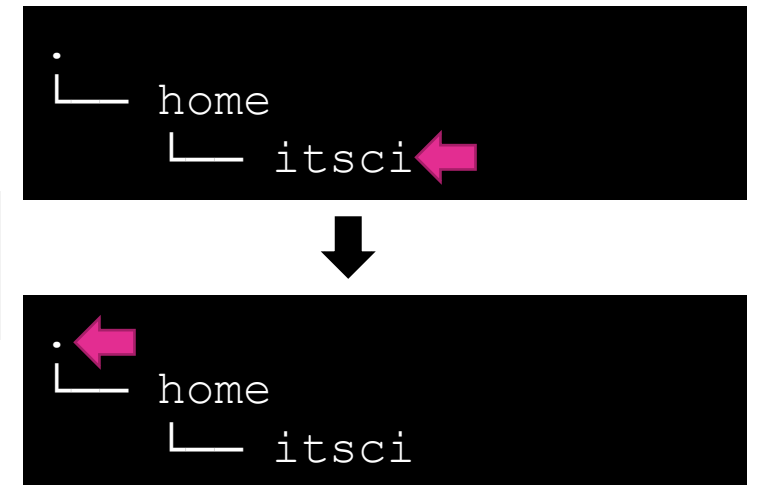
# Navigate the File System

>- เราสามารถใช้ .. หลายครั้งเมื่อต้องการย้ายไปยัง directory หลายระดับจาก directory ปัจจุบัน:

>- พิมพ์คำสั่งต่อไปนี้:

```
$ cd ../../  
$ pwd
```

 เป็นคำสั่งย้ายไป parent directory (/)



หมายเหตุ: เครื่องหมาย . คือ directory ปัจจุบันส่วน .. คือ directory ก่อนหน้า เช่น cd . หรือ cd ..

# Relative and absolute paths

>- จากหลายตัวอย่างที่ผ่านมาแสดงการย้าย directory โดยใช้ relative paths

 นั่นคือการย้ายไป directory จะอ้างอิงจากจุดที่อยู่ของ directory ปัจจุบัน

>- เมื่อต้องการใช้คำสั่ง cd ไปที่ “etc” folder ในกรณีที่เรายู่ที่ root directory นั้นไม่ยากเพียงใช้คำสั่ง:

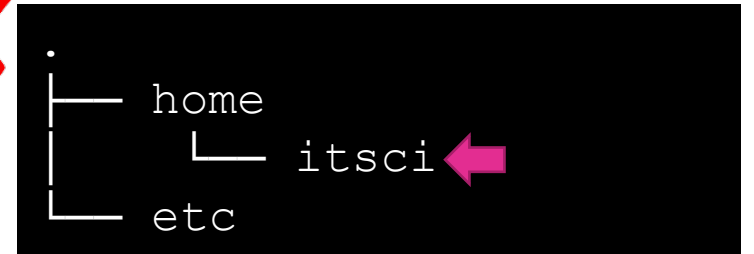
```
$ cd /  
$ pwd  
$ cd etc  
$ pwd
```



# Relative and absolute paths

 เมื่อต้องการใช้คำสั่ง `cd` ไปที่ “etc” folder ในกรณีที่เรายู่ที่ home directory แล้วใช้คำสั่ง:

```
$ cd  
$ pwd  
$ cd etc  
$ pwd
```



 เราจะเห็นข้อความ “No such file or directory”

 การเปลี่ยน directory โดยระบุชื่อหรือใช้ `..` อาจมีผลลัพธ์ที่ต่างกันขึ้นอยู่กับตำแหน่งเริ่มต้นของ directory ปัจจุบันของเรา.

# Relative and absolute paths

>- การเปลี่ยน directory โดยใช้ absolute path จะมีผลลัพธ์แบบเดียวกันไม่ว่าเราจะอยู่ใน directory ที่ไหนก็ตาม.

>- จากตัวอย่างต่อไปนี้

🐧 เมื่อเราเปลี่ยนไปที่ home directory.

🐧 จากนั้นใช้คำสั่ง `cd /` ตามด้วยชื่อของ directory ที่ต้องการ เช่น

```
$ cd
$ pwd
$ cd /etc
$ pwd
```



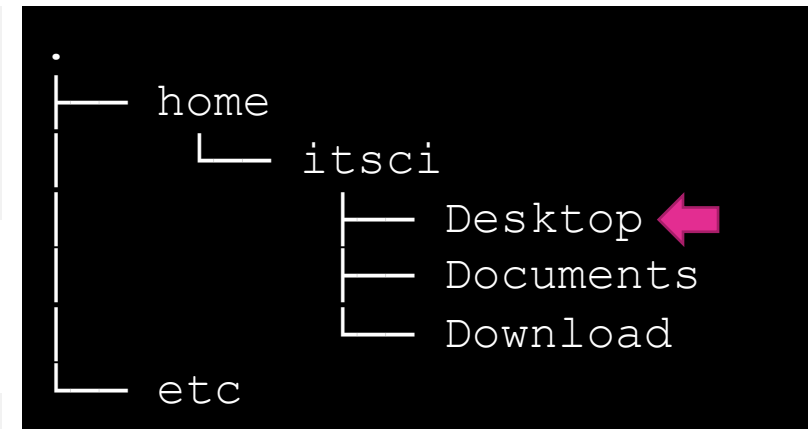
# Relative and absolute paths

- >- หากต้องการเปลี่ยนไปยัง `/home/your_username/Desktop` (การหาชื่อผู้จะใช้คำสั่ง `whoami`)

```
$ whoami  
$ cd /home/itsci/Desktop  
$ pwd
```

- >- ยังมีอีกวิธีในการเปลี่ยนไปยัง home directory

```
$ cd ~  
$ pwd  
$ cd ~/Desktop  
$ pwd
```





# Using the Terminal

>- คำสั่ง echo ใช้แสดงข้อความไปยัง Terminal

```
$ echo Hello  
Hello
```

>- คำสั่ง cal ใช้แสดงปฏิทิน

```
$ cal  
      April 2023  
Su Mo Tu We Th Fr Sa  
      1  
  2  3  4  5  6  7  8  
  9 10 11 12 13 14 15  
16 17 18 19 20 21 22  
23 24 25 26 27 28 29  
30
```

# Using the Terminal

- `>-` คำสั่ง `cal` จะแสดงค่า default ออกมาเป็นเดือนปัจจุบัน หากต้องการแสดงปฏิทินทั้งปีให้ระบุปีที่ต้องการ เช่น `cal 2023` หรือ `cal -y`

```
$ cal 2023
```

```
                2023
      January          February          March
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
  1  2  3  4  5  6  7          1  2  3  4          1  2  3  4
  8  9 10 11 12 13 14    5  6  7  8  9 10 11    5  6  7  8  9 10 11
15 16 17 18 19 20 21    12 13 14 15 16 17 18    12 13 14 15 16 17 18
22 23 24 25 26 27 28    19 20 21 22 23 24 25    19 20 21 22 23 24 25
29 30 31                26 27 28                26 27 28 29 30 31
                ...
```

# Using the Terminal

>- คำสั่ง date ใช้แสดงวันที่ปัจจุบัน

```
$ date
```

```
Tue Apr 25 14:35:38 +07 2023
```

>- คำสั่ง clear ใช้ล้างหน้าจอ หรือกดคีย์ ctrl+l

```
$ clear
```

# Using the Terminal

- >- การกดคีย์ลูกศรขึ้นและลงจะเป็นการกลับไปใช้คำสั่งก่อนหน้า
- >- คำสั่ง `history` จะแสดงรายการประวัติคำสั่งที่เคยใช้

```
$ history
1010 whoami
1011 cd /Users/watcharinsarachai/Desktop
1012 pwd
1016 cal
1017 cal 2019
1019 cal 2023
1020 cal -y
1021 date
```

เราสามารถนำคำสั่งในรายการนี้ได้ โดยใช้เครื่องหมาย ! ตามด้วยหมายเลขรายการ เช่น

```
$ !1010
whoami
watcharinsarachai
```

# Command Line Syntax

 Command = `commandName` `options` `inputs`

 `commandName` จะต้องอยู่ใน search path ของระบบ

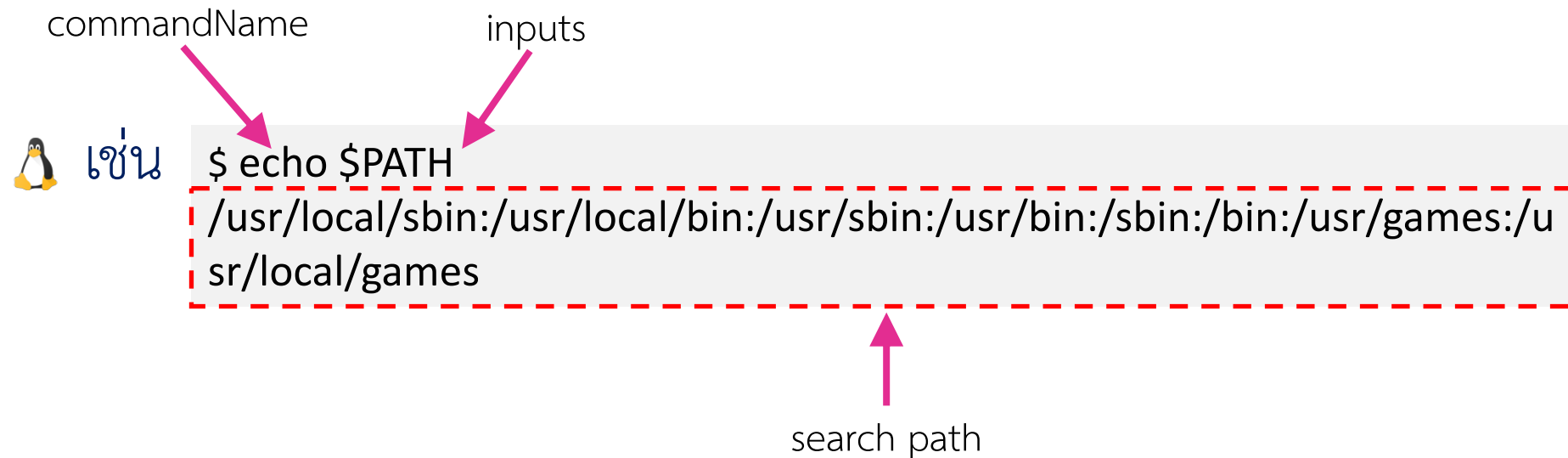
 `command` จะทำงานตาม `inputs` ที่ระบุ

 `options` จะกำหนดพฤติกรรมการทำงานของ `command`

# Command Line Syntax

>- รูปแบบของคำสั่งประกอบไปด้วย

```
command = commandName options inputs
```



หมายเหตุ: `$PATH` หรือ `search path` คือรายการ `directory` ของคำสั่งต่าง ๆ ที่ใช้ในระบบ เราสามารถพิมพ์คำสั่งต่าง ๆ เหล่านั้น ณ ตำแหน่งไหนก็ได้

# Command Line Syntax

 เมื่อเราพิมพ์คำสั่งระบบจะค้นหาคำสั่งจากรายการ directory ใน \$PATH โดยเรียงลำดับจากหน้าไปหลัง



เช่น

```
$ which cal  
/usr/bin/cal
```



จะเห็นว่าคำสั่ง cal อยู่ใน /usr/bin/cal ระบบจะเริ่มค้นหาจาก

```
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games
```

↑  
1

↑  
2

↑  
3


↑  
4


# Command Line Syntax

 จากรูปแบบของคำสั่งเราสามารถใส่ inputs ให้กับ command ได้หลายตัว

```
command = commandName options inputs
```

commandName      inputs

 เช่น `$ cal 12 2023`

 การใส่ options เช่น `$ cal -y`

options

*หมายเหตุ: option จะมีเครื่องหมาย - นำหน้าเสมอต่างกับ inputs จะไม่มี*



# Command Line Syntax

>- Options มีสองรูปแบบคือ

 แบบสั้น เช่น `$ date -u`

 แบบยาว เช่น `$ date --universal`

>- ทั้งสองรูปแบบมีการทำงานและความหมายเหมือนกัน

>- Options เป็น case sensitive นั้นหมายความว่า `--universal` และ `--Universal` ไม่เหมือนกัน (เช่นเดียวกัน `-u` และ `-U` ไม่เหมือนกัน)

*หมายเหตุ: option จะมีเครื่องหมาย - นำหน้าเสมอต่างกับ inputs จะไม่มี*

# Command Line Syntax

 ตัวอย่างคำสั่ง

```
$ cal -A 1 12 2023
```

```
$ cal -B 12 2023
```

```
$ cal -A 1 -B 1 12 2023
```

*หมายเหตุ: -A 1 จะแสดงเดือนหลังจากเดือนปัจจุบันอีก 1 เดือนและ -B 1 จะแสดงเดือนก่อนเดือนปัจจุบันอีก 1 เดือน*

# Man command

## Manual Structure

Section	Contains	Description
1	User Commands	Commands that can be run from the shell by a normal user (typically no administrative privileges are needed)
2	System Calls	Programming functions used to make calls to the Linux kernel
3	C Library Functions	Programming functions that provide interfaces to specific programming libraries.
4	Devices and Special Files	File system nodes that represent hardware devices or software devices.
5	File Formats and Conventions	The structure and format of file types or specific configuration files.
6	Games	Games available on the system
7	Miscellaneous	Overviews of miscellaneous topics such as protocols, filesystems and so on.
8	System administration tools and Daemons	Commands that require root or other administrative privileges to use

# Man command

`>-` เมื่อเราพิมพ์คำสั่ง `man` จะเป็นการแสดงคู่มือการใช้คำสั่งต่าง ๆ (Manual)

เช่น `$ man cal`

```
CAL(1) BSD General Commands Manual CAL(1)
NAME
  cal, ncal - displays a calendar and the date of Easter
SYNOPSIS
  cal [-3h]jy [-A number] [-B number] [[month] year]
  cal [-3hj] [-A number] [-B number] -m month [year]
  ncal [-3bhj]pwySM [-A number] [-B number] [-W number] [-s country_code] [[month] year]
  ncal [-Jeo] [-A number] [-B number] [year]
  ncal [-CN] [-H yyyy-mm-dd] [-d yyyy-mm]
DESCRIPTION
  The cal utility displays a simple calendar in traditional format and ncal offers an alternative layout, more options and the date of Easter. The new format is a little cramped but it makes a year fit on a 25x80 terminal. If arguments are not specified, the current month is displayed.
  The options are as follows:
  -h      Turns off highlighting of today.
  -J      Display Julian Calendar, if combined with the -o option, display date of Orthodox Easter according to the Julian Calendar.
  -e      Display date of Easter (for western churches).
  -j      Display Julian days (days one-based, numbered from January 1).
  -m month
Manual page cal(1) line 1 (press h for help or q to quit)
```

# Man command

 คำสั่ง man สามารถค้นหาคำสั่งตามต้องการได้โดยการระบุ options “-k”

เช่น `$ man -k “list directory contents”`

 เป็นการค้นหาคำสั่งต่าง ๆ ที่เกี่ยวข้องกับ “list directory contents” ซึ่งผลลัพธ์ดังแสดงต่อไปนี้

```
$ man -k “list directory contents”
```

```
dir (1)           - list directory contents
ls (1)            - list directory contents
vdir (1)          - list directory contents
ntfsls (8)        - list directory contents on an NTFS filesystem
```

 เมื่อต้องการดูรายละเอียด เช่น คำสั่ง ntfsls ซึ่งอยู่ในบทที่ 8 เราจะพิมพ์คำสั่ง

```
$ man 8 ntfsls
```

# Man command

`>-` เมื่อพิมพ์คำสั่ง `$ man 1 ls`

```
LS(1)                                User Commands                                LS(1)
NAME
  ls - list directory contents
SYNOPSIS
  ls [OPTION]... [FILE]...
DESCRIPTION
  List information about the FILES (the current directory by default). Sort entries alphabetically if none of
  -cftuvSUX nor --sort is specified.

  Mandatory arguments to long options are mandatory for short options too.

  -a, --all
      do not ignore entries starting with .

  -A, --almost-all
      do not list implied . and ..

  --author
      with -l, print the author of each file

  -b, --escape
      print C-style escapes for nongraphic characters

  --block-size=SIZE
      with -l, scale sizes by SIZE when printing them; e.g., '--block-size=M'; see SIZE format below

Manual page ls(1) line 1 (press h for help or q to quit)
```

# Command Synopsis Cheat Sheet

SYNOPSIS คือรูปแบบของคำสั่ง เช่น

```
ls [OPTION]... [FILE]...
```

Section	Meaning
[THING]	THING is optional.
<THING>	THING is mandatory (required)
THING ...	THING can be repeated (limitlessly)
THING1   THING2	Use THING1 OR THING2. Not Both.
<i>THING</i>	[Notice the Italics] Replace THING with whatever is appropriate.

อ่านเพิ่มเติมได้ที่: <https://medium.com/@jaewei.j.w/how-to-read-man-page-synopsis-3408e7fd0e42>

# Command Input and Output

- >- เรามี 2 ช่องทางในการส่งข้อมูลให้ command และ 2 วิธีที่ command ส่งผลลัพธ์ออกมา



หมายเหตุ: Standard Input (0) ก็คือ keyboard

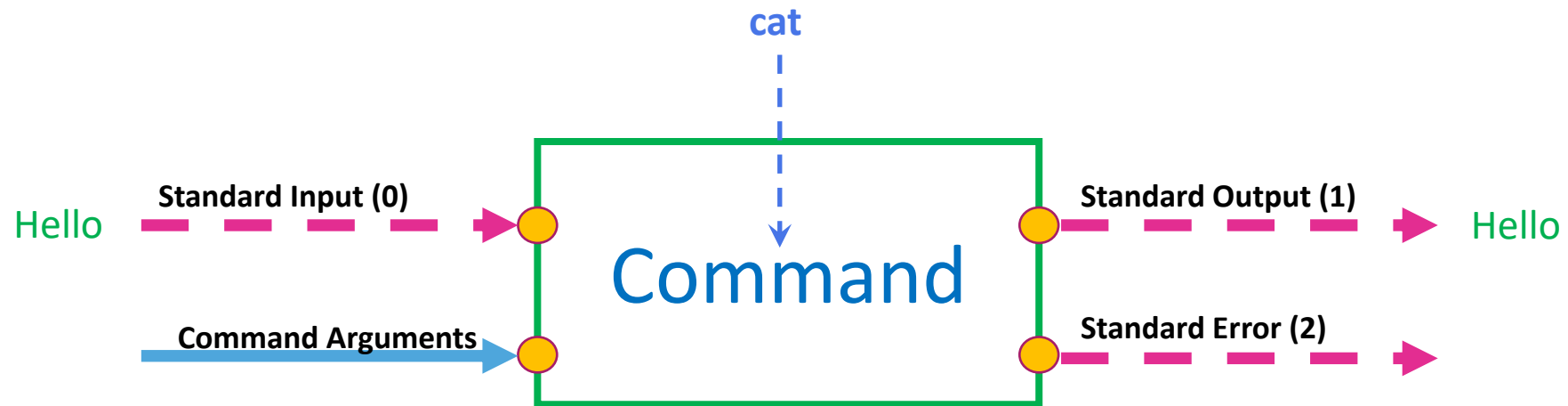
Key:  
- - - - -> Standard Data Stream  
—————> Not a Data Stream



# Command Input and Output

>- เมื่อพิมพ์คำสั่ง

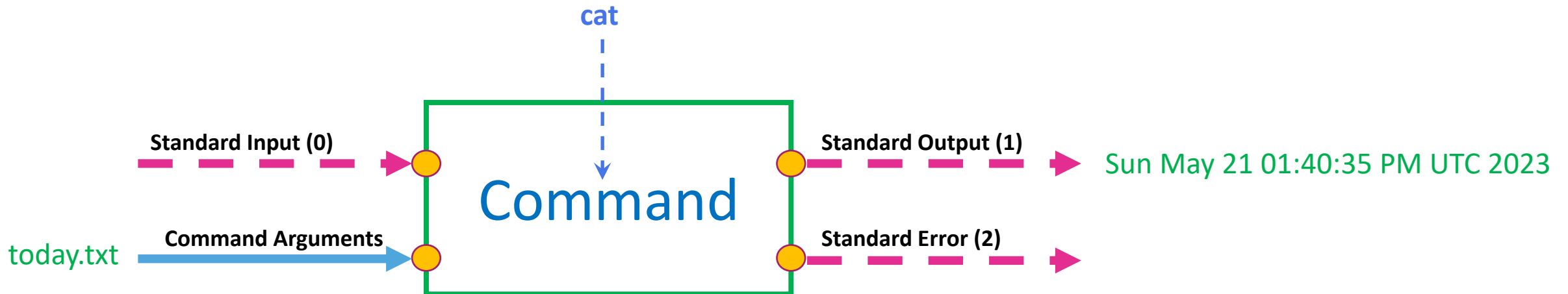
```
$ cat  
Hello  
Hello
```



# Command Input and Output

> เมื่อพิมพ์คำสั่ง

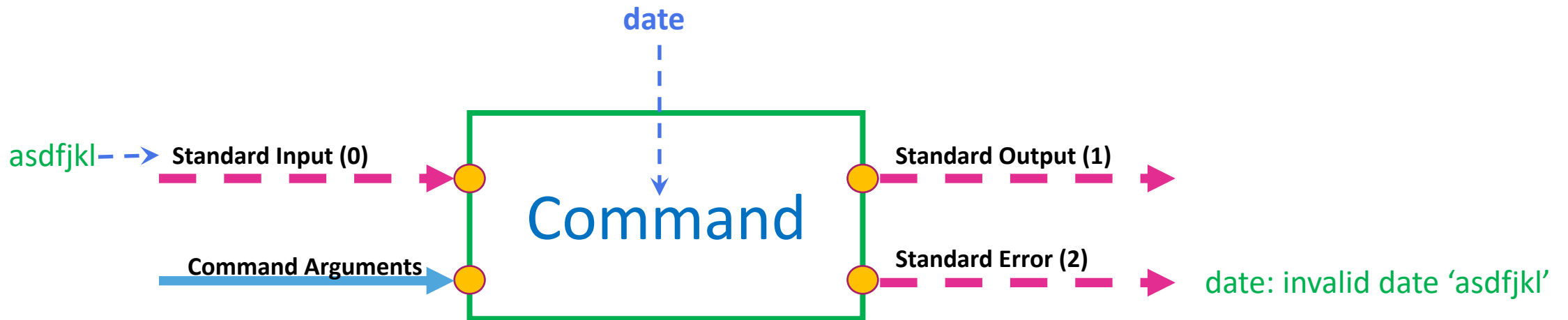
```
$ date > today.txt  
$ cat today.txt  
Sun May 21 01:40:35 PM UTC 2023
```



# Command Input and Output

> เมื่อพิมพ์คำสั่ง

```
$ date asdfjkl  
date: invalid date 'asdfjkl'
```



# Command Input and Output

`>-` เมื่อพิมพ์คำสั่ง `$ cat`

รันคำสั่ง `man cat` เพื่อดูรายละเอียดคู่มือเพิ่มเติม

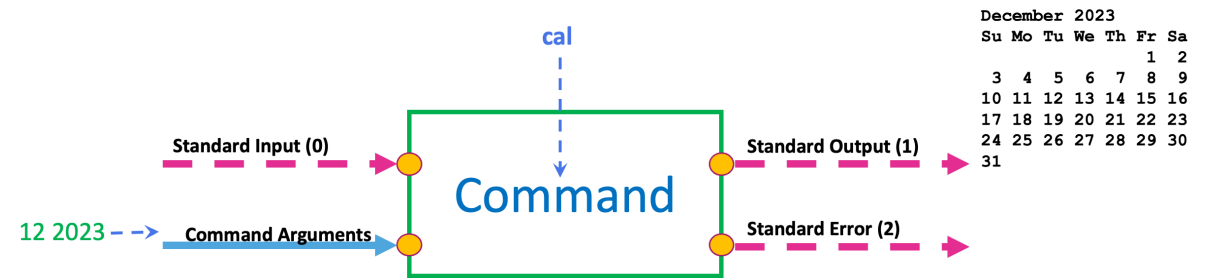
คำสั่งนี้จะใช้ Standard Input (0) เช่น

```
Input --> $ cat
Output --> hello
```

คำสั่ง `cal`

```
$ cal 12 2023
December 2023
Su Mo Tu We Th Fr Sa
          1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
```

Command line arguments



# Command Input and Output

> เมื่อพิมพ์คำสั่ง

```
$ cal -A 1 -B 1 12 2023
```

```
November 2023          December 2023          January 2024
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
                1  2  3  4                1  2                1  2  3  4  5  6
 5  6  7  8  9 10 11    3  4  5  6  7  8  9    7  8  9 10 11 12 13
12 13 14 15 16 17 18   10 11 12 13 14 15 16   14 15 16 17 18 19 20
19 20 21 22 23 24 25   17 18 19 20 21 22 23   21 22 23 24 25 26 27
26 27 28 29 30                24 25 26 27 28 29 30   28 29 30 31
                               31
```

```
$ cal -A 1 -B 1 12 2023
```

Command line arguments

Options

-A มีค่าเท่ากับ 1

-B มีค่าเท่ากับ 1

# Command Input and Output

 Command line สามารถส่งต่อได้ (Flow)

 Standard Input, Standard Output และ Standard Error เป็น “Standard Data Streams”

 Data streams สามารถส่งต่อได้และสามารถส่งต่อ (redirected) จาก default ไปยังที่อื่นได้

 ค่า default ของ output stream คือ Terminal

 เราสามารถส่งต่อ (redirect) standard output ของคำสั่งหนึ่งไปยัง standard input ของอีกคำสั่งหนึ่งได้ เรียกว่า **“piping”**



# Redirection

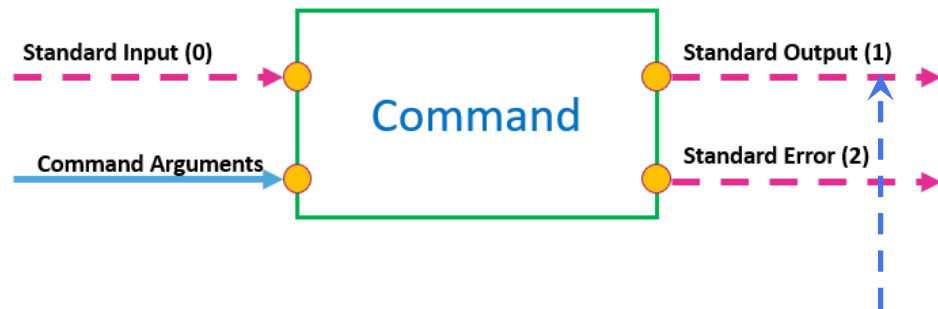
>- เปลี่ยน output stream (1) จาก Terminal ไปยังไฟล์

```
$ cat 1> output.txt
```

```
Hello, ITSCI ← - - - - - พิมพ์จากคีย์บอร์ด
```

```
^C ←
```

กดปุ่ม ctrl+c เพื่อสิ้นสุดการพิมพ์



หมายเลข 1

output.txt

```
Hello, ITSCI
```





# Error Redirection

>- ปกติเมื่อมี error ในคำสั่งเช่น

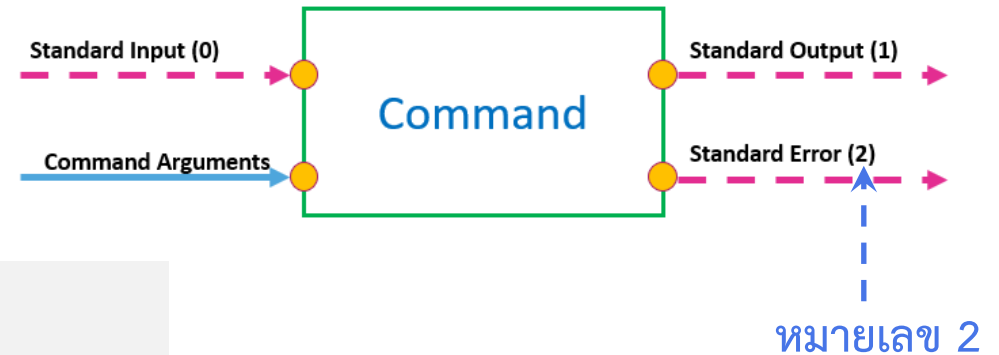
```
$ cat -k bla  
cat: invalid option -- 'k'  
Try 'cat --help' for more information. }
```

ข้อความ error

>- เปลี่ยนข้อความ error ไปเก็บที่ไฟล์ใช้หมายเลข 2

```
$ cat -k bla 2>> error.txt  
$
```

Stream number    ข้อความ error มักจะต่อท้ายไฟล์เดิมเสมอ



error.txt

```
cat: invalid option -- 'k'  
Try 'cat --help' for more information.
```

# Redirection

 การอ่านข้อมูลจากไฟล์แทนการพิมพ์จากคีย์บอร์ด

```
$ cat 1> input.txt  
Hello World!  
^C  
$ cat 0< input.txt  
Hello World!
```

input.txt

Hello World!

# Redirection

 พิจารณาตัวอย่างคำสั่งต่าง ๆ ดังต่อไปนี้

```
$ cat 1> output.txt 2>> error.txt
```

```
$ cat > output.txt 2>> error.txt
```

```
$ cat > input.txt
```

```
Hello World!
```

```
^C
```

```
$ cat < input.txt > hello.txt
```

```
$ cat 0< input.txt 1>> hello.txt
```

# Redirection

 การส่งข้อมูลจาก Terminal หนึ่งไปยังอีก Terminal

 คำสั่ง `tty` แสดงที่อยู่ของ Terminal ปัจจุบันในรูปแบบ File System เช่น

```
$ tty  
/dev/pts/0
```

หมายเลขประจำตัวของแต่ละ Terminal อาจเป็น 1,2,3,...

# Redirection

>- การส่งข้อมูลจาก Terminal หนึ่งไปยังอีก Terminal

Terminal-1:/dev/pts/0

```
$ cat > input.txt  
Hello World!  
^C  
$ cat < input.txt > /dev/pts/1
```

Terminal-2:/dev/pts/1

```
$ tty  
/dev/pts/1  
$ Hello World!
```



# Reference

- `>-` <https://ubuntu.com/tutorials/command-line-for-beginners#1-overview>
- `>-` <https://www.youtube.com>
- `>-` <https://itsci.mju.ac.th/~watcharin/wordpress>