



ใบงานที่ 1

เรื่อง Basic commands

เสนอ

อาจารย์ปิยพล ยืนยงสถาวร

จัดทำโดย

นาย กวีวัฒน์ กาญจน์สุพัฒนากุล 65543206003-7

ใบงานนี้เป็นส่วนหนึ่งของรายวิชา ระบบปฏิบัติการ
หลักสูตรวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา
ประจำภาคที่ 2 ปีการศึกษา 2566

ใบงานที่ 2

Basic commands

ขั้นตอนการทดลอง

1. ให้สมัครสมาชิกเข้าใช้งานระบบใน <https://www.webminal.org/>
2. Login เข้าระบบ และเปิดเมนู Terminal
3. ปฏิบัติการทดลองคำสั่งใน Lessons 1 – 11
4. บันทึกการทดลอง และผลการทดลอง
5. อธิบายการปฏิบัติแต่ละขั้นตอนอย่างละเอียด พร้อมบันทึกภาพประกอบการปฏิบัติ
6. สรุปผลการทดลอง
7. จัดรูปแบบเอกสารใบงานให้เรียบร้อย
8. บันทึกไฟล์ในรูปแบบ PDF ส่งใน Assignment LAB2 : Basic commands

บันทึกการทดลอง และผลการทดลอง

Lesson1 - Basic commands to navigate directories

Lesson2 - Create files, display contents and stats

Lesson3 - Copy, rename, delete files

Lesson4 - Basic process commands

Lesson5 - Manipulate or parse file contents

Lesson6 - Changing file attributes

Lesson7 - Locate file and its type

Lesson8 - System and user details

Lesson9 - Linux Process Basic commands 1

Lesson10 - Linux Process Basic commands 2

Lesson11 - Linux Process states

ผลการทดลอง

Lesson1 - Basic commands to navigate directories

```
-sh-4.2$ pwd
/home/iiuuuu
-sh-4.2$ mkdir -v dir1
mkdir: created directory 'dir1'
-sh-4.2$ mkdir -v dir2
mkdir: created directory 'dir2'
-sh-4.2$ mkdir -v dir2/dir3
mkdir: created directory 'dir2/dir3'
-sh-4.2$ mkdir -v dir2/dir3/dir4
mkdir: created directory 'dir2/dir3/dir4'
-sh-4.2$ ls
dir1 dir2
-sh-4.2$ ls -R
.:
dir1 dir2

./dir1:

./dir2:
dir3

./dir2/dir3:
dir4

./dir2/dir3/dir4:
-sh-4.2$ cd dir2
-sh-4.2$ cd dir3
-sh-4.2$ cd ..
-sh-4.2$ cd -
/home/iiuuuu/dir2/dir3
-sh-4.2$ _cd
```

pwd = แสดง Directory ที่อยู่ปัจจุบัน

mkdir dir1 = สร้าง Directory ชื่อ dir1

ls = ระบุ dir1 dir2 เป็นเนื้อหา Directory ไข้ไหม นั่นคือสิ่งที่เราต้องการ

ls -r = แสดงรายการไฟล์และ Directory ในลำดับย้อนกลับ

cd dir2 = cd คือการเข้าถึง Directory ตามชื่อที่ตั้ง

cd .. = ถอยออกจาก Directory

cd = จะย้ายไปยังโฮม Directory ของคุณ

Lesson2 - Create files, display contents and stats

```
139-162-5-218 login: iiuuuu
Password:
Last login: Fri Nov 24 03:24:50 from localhost, 1.47.82.240
```

```
-sh-4.2$ touch file1.txt
-sh-4.2$ touch file2.txt
-sh-4.2$ dir
dir1 dir2 file1.txt file2.txt
-sh-4.2$ echo "hello"
-sh: echohello: command not found
-sh-4.2$ echo "hello" > hello.txt
-sh-4.2$
-sh-4.2$ echo "Linux" >> hello.txt
-sh-4.2$ echo "world" >> hello.txt
-sh-4.2$ cat hello.txt
hello
Linux
world
-sh-4.2$ head -2 hello.txt
hello
Linux
-sh-4.2$ tail -2 hello.txt
Linux
world
-sh-4.2$ tail hello.txt
hello
Linux
world
```

```
-sh-4.2$ stat hello.txt
  File: 'hello.txt'
  Size: 18          Blocks: 8          IO Block: 4096   regular file
Device: 810h/2064d Inode: 12411220    Links: 1
Access: (0664/-rw-rw-r--)  Uid: (290846/  iiuuuu)   Gid: (290905/  iiuuuu)
Context: guest_u:object_r:user_home_t:s0
Access: 2023-11-24 03:37:43.077386279 +0000
Modify: 2023-11-24 03:37:37.174231617 +0000
Change: 2023-11-24 03:37:37.174231617 +0000
 Birth: -
-sh-4.2$ stat dir1
  File: 'dir1'
  Size: 6          Blocks: 0          IO Block: 4096   directory
Device: 810h/2064d Inode: 271342903   Links: 2
Access: (0775/drwxrwxr-x)  Uid: (290846/  iiuuuu)   Gid: (290905/  iiuuuu)
Context: guest_u:object_r:user_home_t:s0
Access: 2023-11-24 03:28:41.266190888 +0000
Modify: 2023-11-24 03:27:24.779186937 +0000
Change: 2023-11-24 03:27:24.779186937 +0000
 Birth: -
```

touch file1.txt = สร้างไฟล์ใหม่ที่ชื่อ "file1.txt" ถ้าไฟล์นี้ยังไม่มีอยู่ คำสั่ง touch จะใช้สร้างไฟล์ใหม่ และหากไฟล์นี้มีอยู่แล้ว คำสั่ง touch จะใช้ในการอัปเดต timestamp (เวลาที่สร้างและแก้ไขล่าสุดของไฟล์) โดยไม่เปลี่ยนเนื้อหาของไฟล์.

dir = ใช้เพื่อแสดงรายการเนื้อหาไดเรกทอรี ใช้แล้ว ตามที่คุณเคยถูกต้อง dir เทียบเท่ากับ **ls -C -b** (ฉันรู้ว่าคุณไม่ได้เดาว่า

นั่นคือ โดยค่าเริ่มต้น ไฟล์จะแสดงรายการในคอลัมน์ จัดเรียงในแนวตั้ง และอักขระพิเศษจะแสดงด้วยลำดับหลักของแบ็กสแลช

Clear = เคลียร์หน้าจอ

echo "hello" > hello.txt = ใช้ในการสร้างไฟล์ชื่อ "hello.txt" และเขียนข้อความ "hello" เข้าไปในไฟล์นั้น โดยถ้าไฟล์ "hello.txt" มีอยู่แล้ว, คำสั่งนี้จะเขียนทับข้อมูลที่มีอยู่ในไฟล์ด้วยข้อความ "hello".

cat hello.txt = ใช้ในการแสดงเนื้อหาของไฟล์ "hello.txt" บนหน้าจอ.

head -2 hello.txt = ใช้ในการแสดงบางส่วนหรือ "หัว" (head) ของไฟล์ "hello.txt" บนหน้าจอ, โดยมีตัวเลข -2 ระบุว่าเราต้องการแสดงเพียง 2 บรรทัดแรกของไฟล์.

tail hello.txt = ใช้ในการแสดงบางส่วนหรือ "ท้าย" (tail) ของไฟล์ "hello.txt" บนหน้าจอ. คำสั่งนี้จะแสดงทั้งหมดของไฟล์ "hello.txt" หากไฟล์นี้มีขนาดไม่เกินจำนวนบรรทัดที่ระบุไว้ตามค่าตัวเลือก (default คือ 10 บรรทัด).

stat hello.txt = การแสดงข้อมูลสถิติ (statistical information) เกี่ยวกับไฟล์หรือไดเรกทอรีที่ระบุ. เช่น ประเภทไฟล์ , วันวันเลา

stat dir1 = จะแสดงข้อมูลที่เกี่ยวข้องกับไดเรกทอรีนั้น ๆ รวมถึงข้อมูลเกี่ยวกับการสิทธิ์, inode, วันที่และเวลา, และอื่น ๆ.

Lesson3 - Copy, rename, delete files

```
-sh-4.2$ du --max-depth 3 ~
0      /home/iiuuuu/dir1
0      /home/iiuuuu/dir2/dir3/dir4
0      /home/iiuuuu/dir2/dir3
0      /home/iiuuuu/dir2
24     /home/iiuuuu
-sh-4.2$ cp -v hello.txt dir2
'hello.txt' -> 'dir2/hello.txt'
-sh-4.2$ cp -v hello.txt dir2/file2.txt
'hello.txt' -> 'dir2/file2.txt'
-sh-4.2$ cp -vr dir2/*.txt dir2/dir3
'dir2/file2.txt' -> 'dir2/dir3/file2.txt'
'dir2/hello.txt' -> 'dir2/dir3/hello.txt'
-sh-4.2$ cp -vr dir2/dir3 .
'dir2/dir3' -> './dir3'
'dir2/dir3/dir4' -> './dir3/dir4'
'dir2/dir3/file2.txt' -> './dir3/file2.txt'
'dir2/dir3/hello.txt' -> './dir3/hello.txt'
-sh-4.2$ md5sum hello.txt
6023d9728c6e1170cdf4b3547933d68  hello.txt
-sh-4.2$ md5sum dir2/hello.txt
6023d9728c6e1170cdf4b3547933d68  dir2/hello.txt
-sh-4.2$ du
0      ./dir1
0      ./dir2/dir3/dir4
-sh-4.2$ md5sum hello.txt
6023d9728c6e1170cdf4b3547933d68  hello.txt
-sh-4.2$ mv hello.txt dir2/dir3/dir4/hi.txt
-sh-4.2$ mkdir dir5
-sh-4.2$ mv dir2/*.txt dir5
-sh-4.2$ mv dir5 dir50
-sh-4.2$ ln dir2/dir3/dir4/hi.txt hello
-sh-4.2$ stat hello
  File: 'hello'
  Size: 18             Blocks: 8          IO Block: 4096   regular file
Device: 810h/2064d    Inode: 12411220   Links: 2
Access: (0664/-rw-rw-r--)  Uid: (290846/  iiuuuu)   Gid: (290905/  iiuuuu)
Context: guest_u:object_r:user_home_t:s0
Access: 2023-11-24 03:37:43.077386279 +0000
Modify: 2023-11-24 03:37:37.174231617 +0000
Change: 2023-11-24 03:58:02.601337671 +0000
Birth: -
-sh-4.2$ stat dir2/dir3/dir4/hi.txt
  File: 'dir2/dir3/dir4/hi.txt'
  Size: 18             Blocks: 8          IO Block: 4096   regular file
Device: 810h/2064d    Inode: 12411220   Links: 2
Access: (0664/-rw-rw-r--)  Uid: (290846/  iiuuuu)   Gid: (290905/  iiuuuu)
Context: guest_u:object_r:user_home_t:s0
Access: 2023-11-24 03:37:43.077386279 +0000
Modify: 2023-11-24 03:37:37.174231617 +0000
Change: 2023-11-24 03:58:02.601337671 +0000
Birth: -
-sh-4.2$ ln -s dir2/dir3/dir4/hi.txt softlink
```



```

-sh-4.2$ stat softlink
  File: 'softlink' -> 'dir2/dir3/dir4/hi.txt'
  Size: 21          Blocks: 0          IO Block: 4096   symbolic link
Device: 810h/2064d Inode: 12411258    Links: 1
Access: (0777/lrwxrwxrwx)  Uid: (290846/  iuuuuu)   Gid: (290905/  iuuuuu)
Context: guest_u:object_r:user_home_t:s0
Access: 2023-11-24 03:58:35.706205015 +0000
Modify: 2023-11-24 03:58:35.706205015 +0000
Change: 2023-11-24 03:58:35.706205015 +0000
 Birth: -
-sh-4.2$ rm -i file2.txt
rm: remove regular empty file 'file2.txt'? y
-sh-4.2$ rm -ri dir50/*
rm: remove regular file 'dir50/file2.txt'? y
rm: remove regular file 'dir50/hello.txt'? y
-sh-4.2$ rm -rf junk/*
-sh-4.2$ rmdir dir59
rmdir: failed to remove 'dir59': No such file or directory
-sh-4.2$ rmdir dir50
-sh-4.2$ ls
dir1 dir2 dir3 file1.txt hello softlink

```

du = ใช้ในการแสดงขนาดที่ใช้พื้นที่บนดิสก์ของไดเรกทอรีและไฟล์. คำสั่งนี้สามารถใช้งานร่วมกับตัวเลือกต่าง ๆ เพื่อแสดงข้อมูลเพิ่มเติม เช่น ขนาดทั้งหมดที่ใช้, ขนาดไฟล์แต่ละไฟล์, หรือแสดงในรูปแบบที่เข้าใจง่าย.

du -xh ~ = ใช้ในการแสดงขนาดที่ใช้พื้นที่บนดิสก์ของไดเรกทอรีที่ถูกระบุ (ในที่นี้คือ ~ ซึ่งแทนไดเรกทอรีของผู้ใช้ปัจจุบัน) โดยไม่สนใจการเข้าถึงไฟล์หรือไดเรกทอรีที่ตั้งอยู่ในไดเรกทอรีอื่น.

du: คำสั่งแสดงขนาดที่ใช้พื้นที่บนดิสก์ของไดเรกทอรีและไฟล์.

-x: กำหนดให้คำสั่งไม่ค้นหาขนาดที่ใช้พื้นที่บนดิสก์ของไดเรกทอรีและไฟล์ที่อยู่ในไดเรกทอรีที่ต่างจากไดเรกทอรี

-h: แสดงขนาดในรูปแบบที่อ่านง่าย (human-readable).

cp -v hello.txt dir2 = ใช้ในการคัดลอก (copy) ไฟล์ "hello.txt" ไปยังไดเรกทอรี "dir2" และแสดงข้อความที่ระบุว่าไฟล์ถูกคัดลอก

cp -vr dir2/*.txt dir2/dir3 = ใช้ในการคัดลอกไฟล์ทั้งหมดที่มีนามสกุล ".txt" จากไดเรกทอรี "dir2" ไปยังไดเรกทอรี "dir3" และแสดงข้อความที่ระบุว่าไฟล์ถูกคัดลอก

md5sum hello.txt = ใช้ในการคำนวณและแสดงค่า MD5 checksum (hash) ของไฟล์. MD5 checksum เป็นตัวเลขที่เกิดจากการประมวลผลข้อมูลในไฟล์เพื่อตรวจสอบความถูกต้องของไฟล์นั้น.

mv dir2/*.txt dir5 = ใช้ในการย้าย (หรือเปลี่ยนชื่อ) ไฟล์ทั้งหมดที่มีนามสกุล ".txt" จากไดเรกทอรี "dir2" ไปยังไดเรกทอรี "dir5".

ln dir2/dir3/dir4/hi.txt hello = ใช้ในการสร้าง symbolic link (ที่ระบุโดยตัวชี้ไปยังไฟล์หรือไดเรกทอรีอื่น) ของไฟล์ "dir2/dir3/dir4/hi.txt" ในไดเรกทอรีปัจจุบัน และให้ symbolic link มีชื่อ "hello".

ln -s dir2/dir3/dir4/hi.txt softlink = ใช้สร้าง symbolic link ที่ชื่อ "softlink" และชี้ไปยังไฟล์ "dir2/dir3/dir4/hi.txt".

ln: คำสั่งสร้าง symbolic link หรือ hard link.

-s: เป็นตัวเลือกที่ระบุให้ ln สร้าง symbolic link.

dir2/dir3/dir4/hi.txt: ไฟล์ที่เราต้องการสร้าง symbolic link ชี้ไปยัง.

softlink: ชื่อ symbolic link ที่เราต้องการสร้าง.

rm -i file2.txt = ใช้ในการลบไฟล์ "file2.txt" โดยให้ระบบถามยืนยันจากผู้ใช้ก่อนที่จะทำการลบ (interactive mode).

rm: คำสั่งลบไฟล์หรือไดเรกทอรี.

-i: เปิดโหมด interactive ซึ่งจะระบุให้ระบบถามคำยืนยันจากผู้ใช้ก่อนที่จะทำการลบ.

rm -ri dir50/* = ใช้ในการลบทั้งหมดของไฟล์และไดเรกทอรีที่อยู่ในไดเรกทอรี "dir50" โดยให้ระบบถามคำยืนยันจากผู้ใช้ก่อนที่จะทำการลบแบบแอคชั่นตลอด.

rm -rf junk/* = ใช้ในการลบทั้งหมดของไฟล์และไดเรกทอรีที่อยู่ในไดเรกทอรี "junk" โดยไม่ต้องถามคำยืนยันจากผู้ใช้ (force mode).

rmdir dir50 = ใช้เพียงเท่านั้นสำหรับการลบไดเรกทอรีที่ว่างเปล่าเท่านั้น.

Lesson4 - Basic process commands

```
-sh-4.2$ ps
  PID TTY          TIME CMD
 24901 pts/158    00:00:00 sh
 28947 pts/158    00:00:00 ps
-sh-4.2$ sleep 60 &
[1] 29029
-sh-4.2$ ps
  PID TTY          TIME CMD
 24901 pts/158    00:00:00 sh
 29029 pts/158    00:00:00 sleep
 29035 pts/158    00:00:00 ps
-sh-4.2$ kill 29029
-sh-4.2$ ps
  PID TTY          TIME CMD
 24901 pts/158    00:00:00 sh
 29064 pts/158    00:00:00 ps
[1]+  Terminated                  sleep 60
-sh-4.2$ sleep 30 &
[1] 29426
-sh-4.2$ sleep 30 &
[2] 29428
-sh-4.2$ ps
  PID TTY          TIME CMD
 24901 pts/158    00:00:00 sh
 29426 pts/158    00:00:00 sleep
 29428 pts/158    00:00:00 sleep
 29445 pts/158    00:00:00 ps
-sh-4.2$ killall sleep
[1]-  Terminated                  sleep 30
-sh-4.2$ pidof bash
-sh-4.2$ nice -n 19 sleep 30 &
[1] 30368
-sh-4.2$ renice -n 19 30368
```

4	PID	PPID	UID	USER	RUSER	TTY	TIME+	%CPU	%MEM	S	COMMAND
	24901	24880	2908+	iiuuuu	iiuuuu	pts/158	0:00.14	0.0	0.0	S	sh
	32580	24901	2908+	iiuuuu	iiuuuu	pts/158	0:00.06	0.0	0.0	R	top

```

-sh-4.2$ pstree -p
?(1)└─?(24880)──sh(24901)──top(32580)
    └─?(429)──sh(531)──pstree(567)
-sh-4.2$ time ls -l
total 4
drwxrwxr-x. 2 iuuuuu iuuuuu  6 Nov 24 03:27 dir1
drwxrwxr-x. 3 iuuuuu iuuuuu 18 Nov 24 03:57 dir2
drwxrwxr-x. 3 iuuuuu iuuuuu 52 Nov 24 03:54 dir3
-rw-rw-r--. 1 iuuuuu iuuuuu  0 Nov 24 03:34 file1.txt
-rw-rw-r--. 2 iuuuuu iuuuuu 18 Nov 24 03:37 hello
lrwxrwxrwx. 1 iuuuuu iuuuuu 21 Nov 24 03:58 softlink -> dir2/dir3/dir4/hi.txt

real    0m0.062s
user    0m0.057s
sys     0m0.005s

```

Ps = ใช้ในการแสดงข้อมูลเกี่ยวกับกระบวนการ (processes) ที่กำลังทำงานในระบบ. คำสั่งนี้ไม่แสดงกระบวนการของทุกๆ ผู้ใช้, แต่แสดงกระบวนการของผู้ใช้ปัจจุบัน.

sleep 60 & = ใช้ในการให้ระบบหยุดการทำงาน (sleep) เป็นเวลา 60 วินาที และใช้สัญลักษณ์ & เพื่อให้กระบวนการทำงานที่สร้างขึ้น (ในที่นี้คือ sleep) ทำงานในพื้นหลัง (background).

kill 12345 = ใช้ในการส่งสัญญาณการสิ้นสุด (terminate signal) ไปยังกระบวนการที่มี Process ID (PID) เท่ากับ 12345. สัญญาณการสิ้นสุดทำให้กระบวนการหยุดการทำงาน.

หากคำสั่งทำงานได้สำเร็จ, กระบวนการที่มี PID เท่ากับ 12345 จะถูกสิ้นสุด. หากไม่มีข้อผิดพลาด, kill จะไม่ส่งข้อความยืนยันกลับ.

killall sleep = ใช้ในการส่งสัญญาณการสิ้นสุดไปยังกระบวนการทั้งหมดที่ชื่อ "sleep". คำสั่งนี้มักใช้เพื่อหยุดการทำงานของกระบวนการทั้งหมดที่ชื่อ "sleep" ที่กำลังทำงานในระบบ.

killall -u webminal = ใช้ในการส่งสัญญาณการสิ้นสุดไปยังกระบวนการทั้งหมดที่ถูกเริ่มขึ้นโดยผู้ใช้ที่มีชื่อ "webminal". คำสั่งนี้จะหยุดการทำงานของกระบวนการทั้งหมดที่เป็นของผู้ใช้ "webminal".

killall -w find = ใช้ในการส่งสัญญาณการสิ้นสุดไปยังกระบวนการทั้งหมดที่มีชื่อ "find" และรอให้กระบวนการนั้นจบการทำงานก่อนที่จะสิ้นสุด. การใช้ตัวเลือก -w (wait) ทำให้คำสั่งรอให้กระบวนการ "find" ทุกรายการที่กำลังทำงานเสร็จสิ้นก่อนที่จะส่งสัญญาณการสิ้นสุด.

ในกรณีที่มียหลายกระบวนการ "find" ทำงานพร้อมกัน, -w จะทำให้ killall รอทุกรายการจนกว่ากระบวนการทั้งหมดจะเสร็จสิ้นก่อนที่จะทำการสิ้นสุด. ควรใช้ -w ในกรณีที่คุณต้องการให้การสิ้นสุดกระบวนการเสร็จสมบูรณ์ก่อนที่คำสั่ง killall จะถูกตั้งตาม.

pidof bash = ใช้ในการแสดง Process ID (PID) ของกระบวนการ (process) ที่ชื่อ "bash"

pidof -s bash = ใช้ในการแสดง Process ID (PID) ของกระบวนการ (process) ที่ชื่อ "bash" ที่กำลังทำงานในระบบปฏิบัติการ Linux และให้ผลลัพธ์เป็น PID เดียวที่ตรงกับกระบวนการแรกที่พบ.

nice -n 19 sleep 30 & = ใช้ในการเริ่มกระบวนการ sleep ที่มีความลำบาก (niceness) ที่สูงมาก (19) ในลักษณะ background ในระบบปฏิบัติการ Linux

nice: คำสั่งที่ใช้ในการเปลี่ยนความลำบากของกระบวนการ, ทำให้กระบวนการมีลำบากต่ำลงหรือสูงขึ้น.

-n 19: ตัวเลือก -n ใช้ในการกำหนดความลำบาก (niceness) ของกระบวนการ. ค่า niceness อยู่ในช่วง -20 (สูงสุด) ถึง 19 (ต่ำสุด). ค่า niceness สูงแสดงถึงการทำงานที่น้อยสุดของ CPU.

ในที่นี้, nice -n 19 ทำให้กระบวนการ sleep มีความลำบากสูงมาก (จำนวนความลำบากต่ำ) และ & ทำให้กระบวนการทำงานในพื้นหลัง.

กระบวนการ sleep 30 จะหยุดการทำงานเป็นเวลา 30 วินาที โดยไม่ใช้ทรัพยากร CPU มากนัก, เนื่องจากมีความลำบากสูง.

renice -n 19 12345 = ใช้ในการเปลี่ยนความลำบาก (niceness) ของกระบวนการที่มี Process ID (PID) เท่ากับ 12345 ในระบบ

top = ใช้ในการแสดงข้อมูลเกี่ยวกับประสิทธิภาพของระบบและกระบวนการที่กำลังทำงานในระบบปฏิบัติการ Linux. เมื่อคุณป้อน top ในหน้าต่างคำสั่ง, มันจะแสดงตารางที่ประกอบด้วยข้อมูลต่าง ๆ เกี่ยวกับ CPU, หน่วยความจำ (Memory), และกระบวนการที่กำลังทำงาน.

ps tree -p = ใช้ในการแสดงโครงสร้างของกระบวนการ (process tree) ในระบบปฏิบัติการ Linux พร้อมกับ Process ID (PID) ของแต่ละกระบวนการ.

`time ls -l` = ใช้ในการวัดเวลาที่ใช้ในการทำงานของคำสั่ง `ls -l` ซึ่งในที่นี้เป็นการแสดงรายการไฟล์

และไดเรกทอรีในรูปแบบละเอียด

ผลลัพธ์ที่ได้จะแบ่งเป็นสามส่วน:

real: เวลาที่ใช้ทั้งหมดตั้งแต่เริ่มต้นคำสั่งจนถึงสิ้นสุด.

user: เวลาที่ CPU ใช้ในการประมวลผลคำสั่ง.

sys: เวลาที่ CPU ใช้ในการประมวลผลงานที่เกี่ยวข้องกับระบบ (system-related tasks).

Lesson5 - Manipulate or parse file contents

```
-sh-4.2$ grep "linux" Hello
grep: Hello: No such file or directory
-sh-4.2$ grep "linux" hello
-sh-4.2$ grep -r 'Hello'
.bash_history:grep -r "Hello"
-sh-4.2$ grep -i 'inux' hello
Linux
-sh-4.2$ grep -n 'linux' hello
-sh-4.2$ grep -v 'world' hello
hello
Linux
-sh-4.2$ wc -L hello
5 hello
-sh-4.2$ cut -f1 -d' ' new.txt
cut: the delimiter must be a single character
Try 'cut --help' for more information.
-sh-4.2$ paste hello new.txt
paste: new.txt: No such file or directory
-sh-4.2$ ls
dir1 dir2 dir3 file1.txt hello softlink
-sh-4.2$ mkdir new.txt
-sh-4.2$ paste hello new.txt
hello paste: new.txt: Is a directory

Linux
world
```

grep "linux" hello = ใช้ในการค้นหาข้อความ "linux" ในไฟล์ที่ชื่อ "hello"

grep: เป็นคำสั่งที่ใช้ในการค้นหาข้อความในไฟล์.

"linux": เป็นข้อความที่คุณกำลังค้นหา.

hello: เป็นชื่อของไฟล์ที่คุณต้องการค้นหาใน.

grep -r 'Hello' . = ใช้ในการค้นหาข้อความ "Hello" ในไฟล์ทั้งหมดในไดเรกทอรีปัจจุบัน (รวมถึงไดเรกทอรีย่อยทุกๆ ระดับ) ในระบบปฏิบัติการ Linux.

grep: เป็นคำสั่งที่ใช้ในการค้นหาข้อความในไฟล์.

-r: ตัวเลือกที่ใช้ในการค้นหาไฟล์ทั้งหมดในไดเรกทอรีและไดเรกทอรีย่อย.

'Hello': เป็นข้อความที่คุณกำลังค้นหา.

∴ เป็นพารามิเตอร์ว่าคุณต้องการค้นหาในไดเรกทอรีปัจจุบัน.

grep -i 'linux' hello = ใช้ในการค้นหาข้อความ "linux" (โดยไม่สนใจตัวพิมพ์ใหญ่-เล็ก) ในไฟล์ที่ชื่อ "hello".

grep: เป็นคำสั่งที่ใช้ในการค้นหาข้อความในไฟล์.

-i: ตัวเลือกที่ใช้ในการทำให้ค้นหาเป็น case-insensitive, ซึ่งหมายความว่าไม่สนใจตัวพิมพ์ใหญ่-เล็ก.

grep -n 'linux' hello = ใช้ในการค้นหาข้อความ "linux" ในไฟล์ที่ชื่อ "hello" และแสดงหมายเลขบรรทัดที่พบข้อความนั้นด้วย.

grep: เป็นคำสั่งที่ใช้ในการค้นหาข้อความในไฟล์.

-n: ตัวเลือกที่ใช้ในการแสดงหมายเลขบรรทัดที่พบข้อความ.

grep -v 'world' hello = ใช้ในการแสดงบรรทัดที่ไม่มีข้อความ "world" ในไฟล์ที่ชื่อ "hello".

grep: เป็นคำสั่งที่ใช้ในการค้นหาข้อความในไฟล์.

-v: ตัวเลือกที่ใช้ในการแสดงบรรทัดที่ไม่ตรงกับข้อความที่กำลังค้นหา.

wc -L hello = ใช้ในการแสดงความยาวของบรรทัดที่ยาวที่สุดในไฟล์ที่ชื่อ "hello".

wc: เป็นคำสั่งที่ใช้ในการนับจำนวนบรรทัด, คำ, และตัวอักษรในไฟล์.

-L: ตัวเลือกที่ใช้ในการแสดงความยาวของบรรทัดที่ยาวที่สุด.

cut -f1 -d' ' new.txt = ใช้ในการแสดงเนื้อหาที่อยู่ในคอลัมน์ที่หนึ่ง (-f1) ของไฟล์ที่ชื่อ "new.txt" โดยใช้เครื่องหมายขีดว่างเป็นตัวคั่น (-d' ').

cut: เป็นคำสั่งที่ใช้ในการแสดงหรือตัดข้อมูลจากไฟล์.

-f1: ตัวเลือกที่กำหนดว่าควรแสดงคอลัมน์ที่เท่าไร. ในที่นี้คือคอลัมน์ที่หนึ่ง.

-d' ': ตัวเลือกที่กำหนดตัวคั่น (delimiter) ที่ใช้ในการแบ่งข้อมูล. ในที่นี้คือเครื่องหมายขีดว่าง.

paste hello new.txt = ใช้ในการนำเนื้อหาของไฟล์ "hello" และ "new.txt" มาต่อกันแนวตั้ง (vertically).

sort new.txt = ใช้ในการเรียงลำดับข้อมูลในไฟล์ "new.txt" ตามลำดับที่มีค่าน้อยไปที่มีค่ามาก.

diff hello linux.txt = ใช้ในการเปรียบเทียบเนื้อหาของไฟล์และแสดงการแตกต่างระหว่างไฟล์ทั้งสอง.

Lesson6 - Changing file attributes

```
-sh-4.2$ dirname dir2/dir3/dir4/hi.txt
dir2/dir3/dir4
-sh-4.2$ basename dir2/dir3/dir4/hi.txt
hi.txt
-sh-4.2$ chmod -v 666 file1.txt
mode of 'file1.txt' changed from 0664 (rw-rw-r--) to 0666 (rw-rw-rw-)
-sh-4.2$ chmod a+rw file1.txt
-sh-4.2$ chown root -R ~/dir2
chown: changing ownership of '/home/iiuuuu/dir2/dir3/dir4/hi.txt': Operation not permitted
chown: changing ownership of '/home/iiuuuu/dir2/dir3/dir4': Operation not permitted
chown: changing ownership of '/home/iiuuuu/dir2/dir3/file2.txt': Operation not permitted
chown: changing ownership of '/home/iiuuuu/dir2/dir3/hello.txt': Operation not permitted
chown: changing ownership of '/home/iiuuuu/dir2/dir3': Operation not permitted
chown: changing ownership of '/home/iiuuuu/dir2': Operation not permitted
:webminal root:staff -R ~/dir2
-sh-4.2$ chgrp -hR root dir2
chgrp: changing group of 'dir2/dir3/dir4/hi.txt': Operation not permitted
chgrp: changing group of 'dir2/dir3/dir4': Operation not permitted
chgrp: changing group of 'dir2/dir3/file2.txt': Operation not permitted
chgrp: changing group of 'dir2/dir3/hello.txt': Operation not permitted
chgrp: changing group of 'dir2/dir3': Operation not permitted
chgrp: changing group of 'dir2': Operation not permitted
```

`dirname dir2/dir3/dir4/hi.txt` = ใช้ในการแสดง directory ที่บรรจุ path ของไฟล์หรือ directory ที่ระบุ.

`basename dir2/dir3/dir4/hi.txt` = ใช้ในการแสดงชื่อของไฟล์หรือ directory ที่ถูกระบุใน path.

`chmod -v 666 file1.txt` = ใช้ในการเปลี่ยนสิทธิ์การเข้าถึงของไฟล์ "file1.txt" เป็น 666 ซึ่งหมายถึงทุกคน (owner, group, others) สามารถอ่านและเขียนไฟล์นี้ได้.

`chmod`: เป็นคำสั่งที่ใช้ในการเปลี่ยนแปลงสิทธิ์การเข้าถึงของไฟล์หรือไดเรกทอรี.

`-v`: ตัวเลือกที่ใช้ในการแสดงผลการทำงานของคำสั่ง (verbose mode).

`666`: ค่าสิทธิ์ที่กำหนด 666 ให้ทุกคนมีสิทธิ์อ่านและเขียนไฟล์.

file1.txt: ชื่อของไฟล์ที่คุณต้องการเปลี่ยนแปลงสิทธิ์.

chmod a+rw file1.txt = ใช้ในการเปลี่ยนสิทธิ์การเข้าถึงของไฟล์ "file1.txt" ให้ทุกคน (owner, group, others) มีสิทธิ์อ่านและเขียนไฟล์.

chmod: เป็นคำสั่งที่ใช้ในการเปลี่ยนแปลงสิทธิ์การเข้าถึงของไฟล์หรือไดเรกทอรี.

a: คือ "all" หรือทุกคน (owner, group, others).

+rw: เพิ่มสิทธิ์ให้ทุกคนอ่าน (read) และเขียน (write).

file1.txt: ชื่อของไฟล์ที่คุณต้องการเปลี่ยนแปลงสิทธิ์

chmod -R 644 ~/chmod_dir = ใช้ในการเปลี่ยนสิทธิ์การเข้าถึงของไดเรกทอรี "chmod_dir" และไฟล์ทั้งหมดที่อยู่ในไดเรกทอรีนี้ให้มีสิทธิ์อ่าน (read) เฉพาะสำหรับเจ้าของไฟล์และสิทธิ์อ่าน (read) เฉพาะสำหรับกลุ่มและสิทธิ์ไม่มีสิทธิ์ในการเขียน (write) สำหรับทุกคนอื่น ๆ.

chmod: เป็นคำสั่งที่ใช้ในการเปลี่ยนแปลงสิทธิ์การเข้าถึงของไฟล์หรือไดเรกทอรี.

-R: ตัวเลือกที่ใช้ในการทำการเปลี่ยนแปลงทั้งหมดในโครงสร้างไดเรกทอรี (recursive).

644: ค่าสิทธิ์ที่กำหนด 644 ให้เจ้าของไฟล์มีสิทธิ์อ่าน (read) และเขียน (write) และกลุ่มและทุกคนมีสิทธิ์อ่าน (read) เท่านั้น.

~/chmod_dir: ที่อยู่ของไดเรกทอรีที่คุณต้องการเปลี่ยนแปลงสิทธิ์.

chown root:staff -R ~/dir2 = ใช้ในการเปลี่ยนเจ้าของและกลุ่มของไดเรกทอรี "dir2" และไฟล์ทั้งหมดที่อยู่ใน "dir2" เป็น root เป็นเจ้าของ และ staff เป็นกลุ่ม.

chown: เป็นคำสั่งที่ใช้ในการเปลี่ยนเจ้าของและกลุ่มของไฟล์หรือไดเรกทอรี.

root:staff: ระบุเจ้าของและกลุ่มที่คุณต้องการกำหนด (root เป็นเจ้าของ, staff เป็นกลุ่ม).

-R: ตัวเลือกที่ใช้ในการทำการเปลี่ยนแปลงทั้งหมดในโครงสร้างไดเรกทอรี (recursive).

~/dir2: ที่อยู่ของไดเรกทอรีที่คุณต้องการเปลี่ยนแปลงเจ้าของและกลุ่ม.

chgrp root file1.txt = ใช้ในการเปลี่ยนกลุ่มของไฟล์ "file1.txt" เป็นกลุ่ม "root".

chgrp: เป็นคำสั่งที่ใช้ในการเปลี่ยนกลุ่มของไฟล์หรือไดเรกทอรี.

root: ระบุกลุ่มที่คุณต้องการกำหนดให้กับไฟล์ (ในที่นี้คือ "root").

file1.txt: ชื่อของไฟล์ที่คุณต้องการเปลี่ยนกลุ่ม.

chgrp -hR root dir2 = ใช้ในการเปลี่ยนกลุ่มของไดเรกทอรี "dir2" และไฟล์ทั้งหมดที่อยู่ใน "dir2" และซับไดเรกทอรีทั้งหมด เป็นกลุ่ม "root" โดยรวมกับตัวเลือก -R ทำให้มีการเปลี่ยนแปลงทั้งหมดในโครงสร้างไดเรกทอรี

chgrp: เป็นคำสั่งที่ใช้ในการเปลี่ยนกลุ่มของไฟล์หรือไดเรกทอรี.

-h: ตัวเลือกที่ใช้ในกรณีที่กำหนดกลุ่มบน symbolic link ไฟล์หรือไดเรกทอรี จะเปลี่ยนกลุ่มของ symbolic link ไม่ได้ แต่จะเปลี่ยนกลุ่มของไฟล์หรือไดเรกทอรีที่ symbolic link ชี้ไป.

-R: ตัวเลือกที่ใช้ในการทำการเปลี่ยนแปลงทั้งหมดในโครงสร้างไดเรกทอรี (recursive).

root: ระบุกลุ่มที่คุณต้องการกำหนดให้กับไฟล์และไดเรกทอรี (ในที่นี้คือ "root").

dir2: ที่อยู่ของไดเรกทอรีที่คุณต้องการเปลี่ยนกลุ่ม.

Lesson7 - Locate file and its type

```
-sh-4.2$ file linux.txt
linux.txt: directory
-sh-4.2$ whereis ls
ls: /usr/bin/ls /usr/share/man/man1/ls.1.gz
-sh-4.2$ whereis stdio.h
stdio: /usr/include/stdio.h
-sh-4.2$ which php
/usr/bin/which: no php in (/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin)
-sh-4.2$ find ~ -name "linux.txt"
/home/iuuuuu/linux.txt
```

file linux.txt = ใช้ในการแสดงประเภทของไฟล์ "linux.txt" โดยตรวจสอบความหลากหลายของไฟล์ตามลายลักษณ์บนไฟล์.

file /dev/null = ใช้ในการแสดงประเภทของไฟล์ที่ตำแหน่ง "/dev/null".

whereis ls = ใช้ในการค้นหาตำแหน่งของไบนารี (binary), ไฟล์ความช่วยเหลือ (man page), และไฟล์ที่ทำให้โปรแกรมรันได้ (source code) ของคำสั่ง "ls" ในระบบของคุณ.

whereis stdio.h = ใช้ในการค้นหาตำแหน่งของไฟล์ "stdio.h" ในระบบของคุณ.

php = ใช้เรียกใช้ PHP interpreter ในโปรแกรมคอมพิวเตอร์หรือ command line interface (CLI). การเรียกใช้คำสั่ง php โดยอยู่ในโหมดแบบโต้ตอบ (interactive mode) จะเปิดหน้าต่างสำหรับป้อนคำสั่ง PHP ต่อมา.

which php = ใช้ในการแสดงตำแหน่งของไฟล์ทำงาน (executable) สำหรับโปรแกรม PHP ในระบบของคุณ.

find ~ -name "linux.txt" = ใช้ในการค้นหาไฟล์ที่มีชื่อ "linux.txt" ในโฮมไดเรกทอรีของผู้ใช้ปัจจุบันและตำแหน่งย่อยทั้งหมด.

~: แทนโฮมไดเรกทอรีของผู้ใช้ปัจจุบัน.

-name "linux.txt": กำหนดเงื่อนไขในการค้นหา ในที่นี้คือ การค้นหาไฟล์ที่มีชื่อเป็น "linux.txt".

Lesson8 - System and user details

```
-sh-4.2$ uptime
 16:22:08 up 556 days, 11:32,  0 users,  load av
erage: 0.34, 0.21, 0.19
-sh-4.2$ date
Wed Nov 29 16:22:11 UTC 2023
-sh-4.2$ who -a
-sh-4.2$ mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec
,relatime,seclabel)
proc on /proc type proc (rw,nosuid,nodev,noexec,
relatime,hidepid=2)
devtmpfs on /dev type devtmpfs (rw,nosuid,seclab
el,size=8188224k,nr_inodes=2047056,mode=755)
securityfs on /sys/kernel/security type security
fs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev,se
clabel)
devpts on /dev/pts type devpts (rw,nosuid,noexec
,relatime,seclabel,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,nodev,seclab
el,mode=755)
tmpfs on /sys/fs/cgroup type tmpfs (ro,nosuid,no
dev,noexec,seclabel,mode=755)
cgroup on /sys/fs/cgroup/systemd type cgroup (rw
,nosuid,nodev,noexec,relatime,seclabel,xattr,rel
ease_agent=/usr/lib/systemd/systemd-cgroups-agen
t,name=systemd)
pstore on /sys/fs/pstore type pstore (rw,nosuid,
nodev,noexec,relatime,seclabel)
cgroup on /sys/fs/cgroup/pids type cgroup (rw,no
suid,nodev,noexec,relatime,seclabel,pids)
cgroup on /sys/fs/cgroup/perf_event type cgroup
(rw,nosuid,nodev,noexec,relatime,seclabel,perf_e
vent)
cgroup on /sys/fs/cgroup/net_cls,net_prio type c
group (rw,nosuid,nodev,noexec,relatime,seclabel,
net_cls,net_prio)
cgroup on /sys/fs/cgroup/blkio type cgroup (rw,n
osuid,nodev,noexec,relatime,seclabel,blkio)
cgroup on /sys/fs/cgroup/cpu,cpuacct type cgroup
(rw,nosuid,nodev,noexec,relatime,seclabel,cpu,c
puacct)
cgroup on /sys/fs/cgroup/memory type cgroup (rw,
nosuid,nodev,noexec,relatime,seclabel,memory)
cgroup on /sys/fs/cgroup/rdma type cgroup (rw,no
suid,nodev,noexec,relatime,seclabel,rdma)
cgroup on /sys/fs/cgroup/devices type cgroup (rw
,nosuid,nodev,noexec,relatime,seclabel,devices)
cgroup on /sys/fs/cgroup/hugetlb type cgroup (rw
,nosuid,nodev,noexec,relatime,seclabel,hugetlb)
cgroup on /sys/fs/cgroup/freezer type cgroup (rw
```

```

-sh-4.2$ mount -t ext4
/dev/sda on / type ext4 (rw,relatime,seclabel)
-sh-4.2$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        7.9G   0  7.9G   0% /dev
tmpfs           7.9G 201M  7.7G   3% /dev/shm
tmpfs           7.9G 810M  7.1G  11% /run
tmpfs           7.9G   0  7.9G   0% /sys/fs/cg
roup
/dev/sda        116G  31G   79G  29% /
/dev/sdb        118G  51G   67G  43% /home
tmpfs           1.6G   0  1.6G   0% /run/user/
0
tmpfs           1.6G   0  1.6G   0% /run/user/
230165
tmpfs           1.6G   0  1.6G   0% /run/user/
230291
tmpfs           1.6G   0  1.6G   0% /run/user/
230347
tmpfs           1.6G   0  1.6G   0% /run/user/
212599
tmpfs           1.6G   0  1.6G   0% /run/user/
199548
tmpfs           1.6G   0  1.6G   0% /run/user/
219058
tmpfs           1.6G   0  1.6G   0% /run/user/
220923
tmpfs           1.6G   0  1.6G   0% /run/user/
223554
tmpfs           1.6G   0  1.6G   0% /run/user/
231909
tmpfs           1.6G   0  1.6G   0% /run/user/
231900
tmpfs           1.6G   0  1.6G   0% /run/user/
232285
tmpfs           1.6G   0  1.6G   0% /run/user/
214148
tmpfs           1.6G   0  1.6G   0% /run/user/
232491
tmpfs           1.6G   0  1.6G   0% /run/user/
233083
tmpfs           1.6G   0  1.6G   0% /run/user/
233137
tmpfs           1.6G   0  1.6G   0% /run/user/
234183
tmpfs           1.6G   0  1.6G   0% /run/user/
234240
/dev/sdc        79G  2.1G   75G   3% /common_po
ol
-sh-4.2$ free -m
              total          used          free
shared buff/cache available
Mem:        16015           3317          2602
   1010      10094          11353
Swap:         0              0              0

```

Uptime = ใช้เพื่อแสดงข้อมูลเกี่ยวกับระยะเวลาที่ระบบได้ทำงาน, จำนวนผู้ใช้ที่เข้าสู่ระบบ, และ โหลดของระบบ (system load).

date = ใช้เพื่อแสดงเวลาและวันที่ปัจจุบันของระบบ.

who -a = ใช้ในการแสดงข้อมูลเกี่ยวกับผู้ใช้ที่เข้าสู่ระบบ รวมถึงข้อมูลเพิ่มเติมเกี่ยวกับการเชื่อมต่อ (login sessions).

w = ใช้ในการแสดงข้อมูลเกี่ยวกับผู้ใช้ที่เข้าสู่ระบบและกำลังทำงานในระบบ รวมถึงข้อมูลที่เกี่ยวข้องกับโหลดของระบบ (system load).

mount = ใช้ในการแสดงรายการของพาร์ติชันที่มีการติดตั้งและใช้งานในระบบไฟล์ของ Linux. ตำแหน่งการติดตั้ง, พาร์ติชัน, และประเภทของระบบไฟล์ที่กำลังใช้งานจะถูกแสดงในผลลัพธ์.

mount -t ext4 = ใช้ในการแสดงรายการของพาร์ติชันที่ใช้ระบบไฟล์ประเภท ext4 ที่ถูกติดตั้งและใช้งานในระบบ Linux.

df -h = ใช้ในการแสดงข้อมูลเกี่ยวกับการใช้พื้นที่ดิสก์ (disk space) ในระบบไฟล์. ตัวอักษร -h ในที่นี้หมายถึง "human-readable," ทำให้ข้อมูลแสดงในรูปแบบที่อ่านง่ายสำหรับมนุษย์.

free -m = ใช้ในการแสดงข้อมูลเกี่ยวกับการใช้งานหน่วยความจำ (RAM) ในระบบ. ตัวอักษร -m หมายถึง "megabytes," ทำให้ข้อมูลแสดงในหน่วยเป็นเมกะไบต์.

Lesson9 - Linux Process Basic commands 1

```
-sh-4.2$ hostname
139-162-5-218
-sh-4.2$ file /bin/hostname
/bin/hostname: ELF 64-bit LSB executable, x86-64
, version 1 (SYSV), dynamically linked (uses sha
red libs), for GNU/Linux 2.6.32, BuildID[sha1]=9
3633698bd11eeb4bee21a388c191a5656990d8e, strippe
d
-sh-4.2$ cat /bin/hostname
ELF...>@@@h6@8      @@@@@@@@@@@@@8@8@8@8@8@8@8@
@4-4- @@..`~t@@ @@..`~@@@@T@T@T@DD@Ptd@**@*@t
t@Qtd@Rtd@..`~@@@/lib64/ld-linux-x86-64.so.2@@
@,@@@@@@@,-.)@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@D@@@@@@@@@aa@u@@@@
@@p@@@@2@@@@@@\@@g@@@@2@I@P@@@[@@@@@w@@V@@@@e@@@@
@@1/@@@@n@@1`@(@@@1libns1.so.1__gmon_start__yp_g
et_default_domainyperr_stringlibc.so.6__printf_c
hkexitfopengai_strerroroptind__strdupgetdomainna
meputs__stack_chk_failputcharreallocmemchrstrchr
nulfgetsstrlengetaddrinfo__errno_locationgetname
infosethostnamestdoutfclosemalloc__ctype_b_locop
targetstderrgethostbynamegetopt_longfreeifaddrsget
ifaddrs__fxstatfilenogethostnameefwritestrchrrend
exhstrerror__h_errno_locationerrxstrcmp__libc_st
art_mainsetdomainnamefreeGLIBC_2.2.5GLIBC_2.3.4G
LIBC_2.3GLIBC_2.4@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@ui    @@/~@@1`@,1`@.1`@/1`@-0`@ 0`@ (0`@00`@8
0`@@0`@H0`@P0X0` `0`
h0`
p0`
0`◆@@0`@0`@0`@0`@0`@0`@0`@0`@0`@0`@0`@0`@0`@1`1`@
@1` 1`! 1`"(1`#01`$81`%@1`&H1`'P1`(X1`)`1`*h1`H@
5! Ht@@5"! %$! @@"! h%! h@%!! h@%
! h@%!! h@% h@% h@% hp% `% h P% h
@% h
0% h
@% h% % h@% h@% h@% h@% h@%z h@%r h@%j
h@p%b h`%Z h@P%R h@%J h%B h@ %: h@@%2 h
@%* h@%" h % h!%@ h"%
h#%@ h$%@ h%%@ h&%@ h'p%@ h(`%@ h)P%@ h*@AVAU
ATUSL&H/LcLh@H?@MDA@L-@ L8t@LA@E"E1E1E19 (@
```


hostname = ใช้ในการแสดงชื่อของเครื่อง (hostname) ที่กำลังใช้งาน

file /bin/hostname = ใช้ในการแสดงประเภทของไฟล์ที่ตำแหน่ง "/bin/hostname"

cat /bin/hostname = เรามีอักขระที่อ่านได้และอ่านไม่ได้รวมกัน ที่ใดที่หนึ่งภายในเอาต์พุตเหล่านั้น เรากำลังบอกให้ Kernel อ่านชื่อไฟล์ '/etc/hosts' และค้นหารายการ '127.0.0.1' (ซึ่งโดยทั่วไปจะชี้ไปที่ชื่อเครื่องโฮสต์) และแสดงเนื้อหาขึ้น

ps = ใช้ในการแสดงข้อมูลเกี่ยวกับกระบวนการ (processes) ที่กำลังทำงานในระบบ.

Lesson10 - Linux Process Basic commands 2

```
-sh-4.2$ ps
  PID TTY          TIME CMD
  710 pts/85    00:00:00 sh
 3152 pts/85    00:00:00 ps
-sh-4.2$ sleep 5
-sh-4.2$ sleep 5 &
[1] 3176
-sh-4.2$ ps
  PID TTY          TIME CMD
  710 pts/85    00:00:00 sh
 3257 pts/85    00:00:00 ps
[1]+  Done                  sleep 5
-sh-4.2$ pstree -p 3176
-sh-4.2$ sleep 45 &
[1] 3471
-sh-4.2$ sleep 45 &
[2] 3475
-sh-4.2$ sleep 45 &
[3] 3477
-sh-4.2$ sleep 45 &
[4] 3478
-sh-4.2$ sleep 3000 &
[5] 3487
-sh-4.2$ pstree -p 3257
-sh-4.2$ ps
  PID TTY          TIME CMD
  710 pts/85    00:00:00 sh
 3487 pts/85    00:00:00 sleep
 3561 pts/85    00:00:00 ps
[1]  Done                  sleep 45
[2]  Done                  sleep 45
[3]  Done                  sleep 45
[4]- Done                  sleep 45
-sh-4.2$ fg5
-sh: fg5: command not found
-sh-4.2$ fg 5
sleep 3000
jobs
```

sleep 5 จะมันค้างเป็นเวลา 5 วินาทีแล้วแจ้ง bash prompt ให้คุณอีกครั้ง

ps tree ซึ่งจะบอกคุณถึงการแม่ประหว่างกระบวนการลูกและกระบวนการหลัก มันบอกว่าเรามี Bash และโพรเซสลูกหนึ่งตัวชื่อ 'ps tree' คอยดูแลลูกของเรา ตอนนี้มันบอกว่า 'Bash' มีลูกสองคน ชื่อ ps tree และกระบวนการเบื้องหลัง 'sleep' ps tree มีตัวเลือกให้พุดถึง -p ซึ่งแสดง pid ถัดจากชื่อกระบวนการ pid ที่เราได้รับขณะเริ่มกระบวนการพื้นหลังเหมือนกับที่เราได้รับจากเอาต์พุด ps tree สมมติว่าเราได้เริ่มต้นงานเบื้องหลังที่ทำงานระยะยาว 4 งาน แต่ละงานใช้เวลา 145 วินาที และ 1 กระบวนการที่ ยาวมากเป็นเวลา 3000 วินาที

List background jobs ps tree ให้ข้อมูลเกี่ยวกับงานทั้งหมด เราไม่ต้องการ ps tree(6410) เพราะสนใจเฉพาะงานพื้นหลังเท่านั้น วิธีดู เฉพาะงานเหล่านั้น เพื่อจุดประสงค์นี้ เรามีคำสั่ง 'jobs' ที่จะให้ผลลัพธ์เช่น

Foreground process กระบวนการของเรา-[5] ทำงานเป็นเวลา 3,000 วินาที ซึ่งใช้เวลานานจึงจะเสร็จสมบูรณ์ พื้นหลังจะใช้เวลา CPU น้อยกว่าเมื่อเทียบกับกระบวนการที่ไม่ใช่พื้นหลัง กระบวนการเบื้องหน้า เราเข้ามาสู่กระบวนการเบื้องหน้ากัน ดีกว่า พิมพ์ fg ในกรณีของเรา เราต้องนำงานพื้นหลัง-5 มาด้วย

Switch between foreground to background ไม่ต้องกังวล Linux มีความยืดหยุ่นมาก เราก็สามารถทำได้เช่นกัน เพียงกด 'ctrl+z' คุณจะได้รับเอาต์พุดแจ้งว่า งานหยุดแล้ว

Lesson11 - Linux Process states

```
-sh-4.2$ ps -S
  PID TTY          STAT       TIME COMMAND
 4290 pts/107    Ss          0:00 -sh
 4436 pts/107    T           0:00 sleep 100
 4473 pts/107    S+          0:00 seq 1 500000
 4528 pts/124    Ss          0:00 -sh
 4572 pts/124    R+          0:00 ps -S
-sh-4.2$ sleep 100
^Z
[1]+  Stopped(SIGTSTP)      sleep 100
-sh-4.2$ ps S
  PID TTY          STAT       TIME COMMAND
 4290 pts/107    Ss          0:00 -sh
 4436 pts/107    T           0:00 sleep 100
 4473 pts/107    S+          0:00 seq 1 500000
 4528 pts/124    Ss          0:00 -sh
 4636 pts/124    T           0:00 sleep 100
 4641 pts/124    R+          0:00 ps S
-sh-4.2$ seq 1 5
1
2
3
4
5
-sh-4.2$ bash
```

D การนอนหลับอย่างต่อเนื่อง (โดยปกติคือ IO)

R กำลังรันหรือรันได้ (บนคิรวรัน)

S การนอนหลับแบบขัดจังหวะ (รอให้เหตุการณ์เสร็จสิ้น)

T หยุดทำงานโดยสัญญาณควบคุมงาน

X ตายแล้ว (ไม่ควรเห็น)

กระบวนการ Z Defunct ('ซอมบี้') ถูกยกเลิกแต่ไม่ได้เก็บเกี่ยวโดยผู้ปกครอง

พิมพ์ `ps -S` = S ย่อมาจาก Interruptible sleep ที่นี้ bash/shell ของเรากำลังรอให้ลูก 28682 ดำเนินการให้เสร็จสิ้น 28682 อยู่ในสถานะกำลังทำงาน s - ย่อมาจาก session leader และ + บอกว่ามันจะทำงานในเบื้องหน้า

`sleep 100` แล้ว `ctrl + z`

`ps S`

PID (Process ID): เลขระบุกระบวนการ ซึ่งเป็นตัวเลขที่ไม่ซ้ำกันที่ใช้ระบุแต่ละกระบวนการ.

TTY (Terminal): บอกถึงที่สัญญาณเอาต์พุตของกระบวนการ หรือ terminal ที่กระบวนการนั้นๆ กำลังทำงานอยู่.

STAT (Status): แสดงสถานะปัจจุบันของกระบวนการ เช่น R (Running), S (Sleeping), Z (Zombie) ฯลฯ

TIME: เวลาที่ CPU ใช้ในการประมวลผลกระบวนการนั้นๆ.

COMMAND: คำสั่งที่ใช้เริ่มต้นกระบวนการนั้นๆ.


`seq 1 50` คือ ใช้ในการสร้างลำดับตัวเลขที่เริ่มต้นที่ 1 และสิ้นสุดที่ 50 แล้วแสดงผลลัพธ์บนหน้าจอ. โดยปกติ, ผลลัพธ์จะมีจำนวนมากเพื่อแสดงลำดับตัวเลขทั้งหมด.

Zombie process

Zombie เป็นกระบวนการที่ถูกยกเลิกแต่ไม่ได้ถูกเก็บเกี่ยวโดยผู้ปกครอง เมื่อเราบอกว่าพ่อแม่ไม่ได้เก็บเกี่ยว เราหมายถึง 'ผู้ปกครองยังไม่ได้รับสถานะการออกจากลูก' ลูกดำเนินการเสร็จสิ้นพร้อมทั้งสถานะออกและรอให้ผู้ปกครองขอ นี่เป็นกรณีที่ยุงยากในการทำซ้ำ ให้ลอง: เรามีผู้นำเซชันของเรา 2249

bash คือ ชื่อของ Shell ซึ่งเป็นโปรแกรมที่ให้เราสามารถใช้ในการประมวลผลคำสั่งที่คุณป้อนในระบบ Unix-like. ถ้าคุณพิมพ์ "bash" และกด Enter, นั่นจะทำให้คุณเข้าสู่โหมด Bash Shell ที่ให้คุณป้อนคำสั่งโดยตรง. คุณจะเห็นพรอมpts ประสิทธิภาพคำสั่งที่สามารถใช้งานได้ Bash

พิมพ์ `ps`

พิมพ์ (`kill -STOP 5621`) ตัวเลข 5621  ดูตรงแถว PID ของใครของมัน

พิมพ์ `ps` จะเห็นได้ว่า PID 5621 ถูก `kill` ไปแล้ว

แสดงว่ากระบวนการย่อยของเราหยุดทำงานโดยไม่รวบรวมสถานะทางออกจากรายการย่อย เห็นได้ชัดว่าลูกของมัน 2547 อยู่ในสถานะ Zombie Z

Orphaned process

เด็กกำพร้ากำลังดำเนินการโดยไม่มีผู้ปกครอง บิดามารดาเสียชีวิตโดยไม่ได้เก็บเกี่ยวบุตร

(ได้รับสถานะออกจากบุตร) กระบวนการลูกที่ไม่มีพารেন্টเหล่านี้เรียกว่ากระบวนการกำพร้า กระบวนการเหล่านี้ถูกนำมาใช้โดย init ซึ่งมี pid 1 มาลองสร้างกระบวนการที่ถูกปล่อยซึ่งคล้ายกับขั้นตอนข้างต้น:

```
$ bash
```

```
$ ( sleep 100 & ( kill -9 3329 ) )
```

ให้เราแยกแยะที่ละส่วน:

sleep 100 &: คำสั่ง sleep นี้จะทำให้กระบวนการหยุดการทำงาน (sleep) นาน 100 วินาที และ & ทำให้กระบวนการนี้ทำงานใน background.

(kill -9 3329): คำสั่ง kill นี้จะใช้ส่งการจบการทำงานของกระบวนการที่มี PID คือ 3329 โดยใช้สัญลักษณ์ -9 ซึ่งหมายถึงใช้ส่งการของการทำลาย (forceful kill). การใช้ () รวมคำสั่ง kill ใน subshell นี้ทำให้ทั้งคำสั่งทำงานในกระบวนการย่อยแยกต่างหาก.

การรวมทั้งหมดเข้าด้วยกัน, กระบวนการ sleep 100 จะทำงานใน background นาน 100 วินาที และพร้อมกับนั้น คำสั่ง kill -9 3329 จะถูกเรียกใช้เพื่อทำลายกระบวนการที่มี PID เท่ากับ 3329 โดยใช้การทำลายแบบแรงดัน

```
$ ps S จะเห็นได้ว่า PID 3329 ถูก kill แล้ว
```

สรุปผลการทดลอง

เรียนรู้คำสั่งพื้นฐานตั้งแต่การเรียกไฟล์สร้างไฟล์จนถึงการตั้งค่าเวลาเปิดและปิดของเครื่องการย้ายไฟล์การลบไฟล์พวก text หรือ directory การcopy การเขียนไฟล์ การค้นหาข้อความในไฟล์โดยใช้คำสั่งแทนการเปิดค้นเข้าไปดู