

ການພະນັກ L

ການທດລອງທີ 12 ກາຮສຶກຫາອຸປະກຣນເກົບຮັກຫາ ຂໍ້ມູລແລະຮະບບໄຟລ໌

ການທດລອງນີ້ອືບາຍແລະເຂື່ອມໂຢີງເນື້ອຫາຄວາມຮູ້ຂອງທຸກບທເຂົາດ້ວຍກັນ ແຕ່ຈະເນັນບທທີ 7 ແລະບທທີ 6 ເພື່ອໃຫ້ຜູ້ອ່ານມອງເຫັນອຸປະກຣນອິນພຸຖແລະເອົາທິພຸຖເມື່ອນໄຟລ໌ແຕ່ລະໄຟລ໌ ໂດຍມີວັດຖຸປະສົງຄົດັ່ງນີ້

- ເພື່ອໃຫ້ເຂົາໃຈກາຮວດຂາດຂອງໄຟລ໌ແລະໄຟຣີໂດເຮັດທອຣີໃນຮະບບໄຟລ໌
- ເພື່ອໃຫ້ຈັກໂຄຮງສ້າງແລະຮະບບໄຟລ໌ຂອງໜ່ວຍຄວາມຈໍາກາຣີ microSD ທີ່ໃຊ້ງານໃນປ່າຈຸບັນ
- ເພື່ອໃຫ້ເຂົາໃຈຮະບບໄຟລ໌ (File System) ຊື້ນິດຕ່າງໆ ບນບອຮົດ Pi3
- ເພື່ອໃຫ້ສາມາຮາດເຂື່ອມໂຢີງອຸປະກຣນອິນພຸຖເອົາທິພຸຖໜິດຕ່າງໆ ກັບຮະບບໄຟລ໌

L.1 ຂາດຂອງໄຟລ໌ແລະໄຟຣີ

ຜູ້ອ່ານສາມາຮັດຕຽບຂາດຂອງໄຟລ໌ໄດ້ ຊ່ອ filename ທີ່ແທ້ຈິງ ໜ່ວຍເປັນໄປທ໌ ດ້ວຍຄໍາສັ່ງ du (Disk Usage)
ໂດຍທຳມານີ້

- ຍ້າຍໄຟຣີປ່າຈຸບັນໄປທ໌ /home/pi ທີ່ເປັນໄຟຣີຫລັກຂອງຜູ້ໃຊ້ຂໍ້ອ່ານ

```
$ cd /home/pi
```

- ສ້າງໄຟລ໌ຂ້ອຄວາມ test.txt ດ້ວຍໂປຣແກຣມ nano ດ້ວຍຄໍາສັ່ງຕ່ອໄປນີ້

```
$ nano test.txt
```

ພິມພື້ນຂ້ອຄວາມ fdd ລົງໃນໄຟລ໌ ທໍາກາຣ Write ໂດຍກົດປຸ່ມ Ctrl ແຫ່ມດ້ວຍປຸ່ມ O ອອກຈາກໂປຣແກຣມໂດຍກົດປຸ່ມ Ctrl ແຫ່ມດ້ວຍປຸ່ມ X
Ctrl + x

- คำสั่ง 'du -b filename' จะแสดงผลขนาดเป็นจำนวนไปที่หน้าชื่อไฟล์นั้น

\$ du -b test.txt

4 test.txt

4 คืออะไร? กันไฟล์ test.txt

$$4 \text{ byte} = 1 \text{ word} = 1 \text{ กบ}$$

ตัวเลข 4 หมายถึง เลขจำนวนไปที่คำสั่ง du แสดงผลมาตามพารามิเตอร์ b ที่ส่งไป เพื่อบอกค่าขนาดของไฟล์ test.txt เป็นจำนวน 4 ไปที่

- คำสั่ง 'du -B1 filename' ผู้อ่านสามารถตรวจสอบขนาดของไฟล์ได้ฯ ซึ่ง filename ที่จัดเก็บเป็นจำนวนเท่าของ 4096 ไปที่ในอุปกรณ์เก็บรักษาข้อมูล SD ด้วยคำสั่งต่อไปนี้

\$ du -B1 test.txt

4096 test.txt

พื้นที่หน่วย

ตัวเลข 4096 หมายถึง เลขจำนวนไปที่คำสั่ง du แสดงผลมาตามพารามิเตอร์ B1 ที่ส่งไป โดยผู้อ่านจะสังเกตเห็นความแตกต่าง ถึงแม้ไฟล์มีข้อมูลจำนวนน้อยเพียงไม่ถึงไปที่ แต่การจองพื้นที่ในอุปกรณ์สำรองจะมีขนาดเป็นจำนวนเท่าของ 4096 ไปที่เสมอ ✓ = 1 page. micro sd

- คำสั่ง 'du -h' จะแสดงผลขนาดหรือจำนวนไปที่โดยใช้หน่วยเช่น K (Kilo: 1024 หรือประมาณ 10^3) M (Mega: 1048576 หรือประมาณ 10^6) G (Giga: 1073741824 หรือประมาณ 10^9) หน้าชื่อไดเรกทอรีหรือไฟล์เดอร์ที่อยู่ใต้ไดเรกทอรีปัจจุบัน และจะบันทึก 10 รายการสุดท้ายลงในตาราง

\$ du -h

Size	Folder Name
10^6	208 K ./VSCode/extensions/ms-vscode.cpptools-1.3.0-insiders/bin/message/tr
2.9 M	./VSCode/extensions/ms-vscode.cpptools-1.3.0-insiders/bin/message
29 M	./VSCode/extensions/ms-vscode.cpptools-1.3.0-insiders/bin
56 M	./VSCode/extensions/ms-vscode.cpptools-1.3.0-insiders
122 M	./VSCode/extensions
122 M	./VSCode
4.0 K	./Desktop
4.0 K	./gnupg/private-keys-v1.d
8.0 K	./gnupg
441 M	.

L.2 ระบบไฟล์

micro sd.

ผู้ใช้หรือผู้ดูแลระบบลินุกซ์ สามารถตรวจสอบการใช้งานอุปกรณ์เก็บรักษาข้อมูล เช่น ฮาร์ดดิสก์ โซลิดสเตทไดส์ก การดูนิยความจำ ได้โดยคำสั่ง

- คำสั่ง df (Disk File System) สามารถแสดงรายละเอียดของอุปกรณ์เก็บรักษาข้อมูลในเครื่อง
- คำสั่ง 'df -h' จะแสดงรายการ ดังต่อไปนี้ จดบันทึก 10 รายการแรกลงในตาราง

\$ df -h ?

ห้องน้ำ ห้องน้ำ ห้องน้ำ ห้องน้ำ

human readable

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/root	15G	4.5G	9.3G	33%	/
dertmpfs	430M	0	430M	0%	/dev
tmpfs	463M	0	463M	0%	/dev/shm
tmpfs	463M	47M	416M	11%	/run
tmpfs	5.0M	4.0K	5.0M	1%	/run/lock
tmpfs	463M	0	463M	0%	/sys/fs/cgroup
* /dev/mmcblk0p1	253M	48M	205M	19%	/boot
tmpfs	93M	4.0K	93M	1%	/run/user/1000

* ?

โดย Size จะแสดงผลขนาดหรือจำนวนไบท์โดยใช้ตัวคูณที่แตกต่างกัน เช่น K (Kilo: 1024 หรือประมาณ 10^3) M (Mega: 1048576 หรือประมาณ 10^6) G (Giga: 1073741824 หรือประมาณ 10^9)

- คำสั่ง 'df -T' จะเพิ่มคอลัมน์ชนิด (Type) ของแต่ละรายการในการแสดงผล และขนาดเป็นจำนวนเท่าของ 1 กิโลไบท์ (1K) แทน จดบันทึก 5 รายการแรกลงในตาราง

\$ df -T

พื้นที่ฟรี

- คำสั่ง 'df -i' จะแสดงรายการต่างๆ ดังนี้ จบันทึก 10 รายการแรกลงในตาราง

\$ df -i

Filesystem	Inodes	IUsed	IFree	IUse%	Mounted on
/dev/root	941616	124233	817383	14%	/
devtmpfs	109979	388	109591	1%	/dev
tmpfs	118299	1	118298	1%	/dev/shm
tmpfs	118299	550	117749	1%	/run
tmpfs	118299	3	118296	1%	/run/lock
tmpfs	118299	15	118284	1%	/sys/fs/cgroup
/dev/mmcblk0p1	0	0	0	-	/boot
tmpfs	118299	23	118276	1%	/run/user/1000

โดยคอลัมน์จะแสดงผลเป็นจำนวน ไอโนด (inode) แทน รายละเอียดเรื่อง inode ผู้อ่านสามารถค้นคว้าเพิ่มเติมได้ในบทที่ 7 และทาง wikipedia

- คำสั่ง stat แสดงรายละเอียดของไฟล์หรือไดเรกทอรี

\$ stat asm

File: asm	regular file.		
Size: 4096	Blocks: 8	IO Block: 4096	directory
Device: b307h/45831d	Inode: 521754	Links: 3	
Access: (0755/drwxr-xr-x)	Uid: (1000/ pi)	Gid: (1000/ pi)	
Modify: 2019-03-19 19:43:05.449401732 +0700			
Change: 2019-03-19 19:43:05.449401732 +0700			
Birth: -			

ผู้เขียนอธิบายผลลัพธ์ที่ได้ตามลำดับดังนี้

- ชื่อ asm

Device : b307h/45831d.

- ขนาด 4096 ไบท์ ใช้พื้นที่จำนวน 8 Blocks เป็นไดเรกทอรี (directory) ✓

- มีหมายเลข Device = b307h/45831d หรือเท่ากับ b307₁₆/45831₁₀

- มีหมายเลข Inode ที่ 521754 จำนวน 3 Links

- สิทธิ์เข้าถึง (Access) ด้วยรหัส 0644 หรือ 011₂:100₂:100₂ โดยผู้ใช้หมายเลข ID=1000 ชื่อผู้ใช้ (Username)=pi ในกรุํปหมายเลข Groupid=1000 ชื่อกรุํป pi

- เข้าถึง (Access) ณ วันที่ 19 มีนาคม 2019 เวลา 19.43.05

- เปลี่ยนแปลง (Modify) ณ วันที่ 19 มีนาคม 2019 เวลา 19.43.05

- เวลาที่ Inode เปลี่ยนแปลง (Change) ณ วันที่ 19 มีนาคม 2019 เวลา 19.43.05

เบื้องต้นผู้เขียนขอให้ผู้อ่านสร้างไฟล์ผลลัพธ์จากคำสั่ง stat ไปเก็บในไฟล์ เพื่อมาใช้ประกอบการทดลองต่อไป โดย

```
$ stat asm > stat_asm.txt
```

หลังจากนั้น เราสามารถตรวจสอบสถานะของไฟล์ **stat_asm.txt** ได้ดังนี้

```
$ stat stat_asm.txt
```

```
File: stat\asm.txt
Size: 341          Blocks: 8          IO Block: 4096   regular file
Device: b307h/45831d  Inode: 524766      Links: 1
Access: (0644/-rw-r--r--) Uid: ( 1000/      pi)    Gid: ( 1000/      pi)
Access: 2019-03-19 19:45:05.459401732 +0700
Modify: 2019-03-19 19:45:05.459401732 +0700
Change: 2019-03-19 19:45:05.459401732 +0700
Birth: -
```

ผู้เขียนอธิบายผลลัพธ์ที่ได้ตามลำดับ ดังนี้

- ชื่อ stat_asm.txt
- ขนาด 341 ไบท์ ใช้พื้นที่จำนวน 8 Blocks เป็นไฟล์ธรรมดา (regular File)
- หมายเลข Device = b307h/45831d หรือเท่ากับ $b307_{16}/45831_{10}$
- หมายเลข Inode ที่ 524766 จำนวน 1 Links
- สิทธิ์เข้าถึง (Access Permission) ด้วยรหัส 0644 หรือ $011_2:100_2:100_2$ โดยผู้ใช้หมายเลข User ID=1000 ชื่อผู้ใช้ (Username)=pi ในกรุ๊ปหมายเลข Groupid=1000 ชื่อกรุ๊ป pi
- เข้าถึง (Access) ณ วันที่ 19 มีนาคม 2019 เวลา 19.45.05
- เปลี่ยนแปลง (Modify) ณ วันที่ 19 มีนาคม 2019 เวลา 19.45.05
- เวลาที่ Inode เปลี่ยนแปลง (Change) ณ วันที่ 19 มีนาคม 2019 เวลา 19.45.05

L.3 อุปกรณ์อินพุตและเอาท์พุตในระบบไฟล์

การทดลองในหัวข้อนี้จะเข้มต่อ กับเนื้อหาในบทที่ 3 และ การทดลองที่ 4 ภาคผนวก D หลักการของระบบปฏิบัติการ Unix คือ **การมาท์ (Mount)** อุปกรณ์กับไดเรคทอรีด้วยระบบไฟล์ (File System) ที่แตกต่างกัน โดยใช้ชื่อไดเรคทอรีที่แตกต่างกัน โดยมีรูทไดเรคทอรีหรือโฟลเดอร์ (Root Directory) เป็นตำแหน่งเริ่มต้น ผู้อ่านสามารถพิมพ์คำสั่งใน Terminal

\$ mount

คำสั่งนี้จะแสดงรายชื่อการมาท์ หรือ ผูกยึด อุปกรณ์อินพุตเอาท์พุต เข้ากับไดเรคทอรีของระบบปฏิบัติการ ตัวอย่างผลลัพธ์และคำอธิบายต่อไปนี้

- /dev/mmcblk0p7 on / type ext4 (rw,noatime,data=ordered) x 125.
- devtmpfs on /dev type devtmpfs (rw,relatime,size=470116k,nr_inodes=117529,mode=755)
- sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
- proc on /proc type proc (rw,relatime)
- tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
- devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
- ...

โดยมีชนิด (type) หรือระบบไฟล์ที่แตกต่างกัน เช่น

ext.

- ชนิด **ext4** ซึ่งเป็นระบบไฟล์หลักของลีนุกซ์ ย่อมาจากคำว่า Fourth Extended File System เป็นเวอร์ชันที่ 4 พัฒนาจากชนิด **ext3** รายละเอียดเพิ่มเติมที่ [wikipedia](#)
- ชนิด **sysfs** เป็นระบบไฟล์เสมือน (Virtual File System) รายละเอียดเพิ่มเติมที่ [wikipedia](#)
- ชนิด **devfs** เป็นระบบไฟล์เสมือน (Virtual File System) สำหรับอุปกรณ์อินพุตเอาท์พุตต่างๆ รายละเอียดเพิ่มเติมที่ [wikipedia](#)
- ชนิด **tmpfs** ย่อมาจากคำว่า Temporary File System รายละเอียดเพิ่มเติมที่ [wikipedia](#)
- ชนิด **proc** เป็นระบบไฟล์เสมือน (Virtual File System) สำหรับระบบสำคัญต่างๆ เช่น CPU, โดยจะสร้างขึ้นเมื่อบูตเครื่อง และลบทิ้งเมื่อชั้นดาวน์ระบบ รายละเอียดเพิ่มเติมที่ [wikipedia](#)

รายชื่อต่อไปนี้คือ ตัวเลือกคุณสมบัติ (Attribute) ที่สำคัญของระบบไฟล์ เช่น

- **rw**: read/write สามารถอ่านและเขียนได้ ✓
- **noatime** และ **atime**: No/ Access Time หมายถึง ไม่มี/มีการบันทึกเวลาในการสร้าง อ่านหรือเขียนไฟล์ ไฟล์ทุกครั้ง
- **relatime** หมายถึง มีการบันทึกเวลาในการสร้าง อ่านหรือเขียนไฟล์ เมื่อเกิดการแก้ไขไฟล์ หรือ การอ่านหรือเข้าถึงไฟล์มากกว่าเวลาที่บันทึกไว้ก่อนหน้าอย่างน้อย 24 ชั่วโมง

- nosuid: No SuperUser ID เป็นการป้องกันไม่ให้ผู้ดูแลระบบ (SuperUser) กระทำการใดๆ ได้ เพื่อความมั่นคงปลอดภัย
- noexec: No Execution เพื่อตั้งค่าไม่ให้รันไฟล์ที่อยู่ในไดเรคทอรีได้ เช่น ไฟล์ที่เป็นไวรัสหรือมัลแวร์ (Malware) ที่แอوبແפגห์มา
- nodev: No Device หมายถึง การไม่อนุญาตให้สร้างหรืออ่านโหนด (Node) ซึ่งเป็นไฟล์ชนิดพิเศษ
- mode หมายถึง Group 3 บิต คือ บิทควบคุม Read Write Execute รวมทั้งหมด 9 บิต

ผู้อ่านสามารถแสดงรายชื่อไดเรคทอรีหรือไดเรคทอรีหรือชื่ออุปกรณ์ภายใต้ไดเรคทอรี /dev โดยพิมพ์คำสั่งบนโปรแกรม Terminal

```
$ ls /dev
```

ผู้อ่านจะเห็นผลลัพธ์ที่ได้ทั้งหมดซึ่งมีจำนวนมากพอสมควร แต่ผู้เขียนได้พิมพ์ชื่ออุปกรณ์ที่สำคัญๆ ด้วยตัวหนา เพื่อให้ผู้อ่านมองเห็นชัดว่า **mmcblk0p7** มีอยู่จริงและระบบได้ทำการมาที่เข้ากับไดเรคทอรีรoot (Root) นั้นคือ ไดเรคทอรี / ด้วยชนิด ext4 ตามที่ได้แสดงในคำสั่งก่อนหน้าแล้ว

```
autofs block btrfs-control bus cachefiles char console cpu_dma_latency cuse disk fb0
fd full fuse gpiochip0 gpiochip1 gpiochip2 gpiomem hidraw0 hidraw1 hwrng initctl input kms
log loop0 loop1 loop2 loop3 loop4 loop5 loop6 loop7 loop-control mapper mem memory_bandwidth
mmcblk0 mmcblk0p1 mmcblk0p2 mmcblk0p5 mmcblk0p6 mmcblk0p7
mqueue net network_latency network_throughput null ppp ptmx pts ram0 ram1 ram10
ram11 ram12 ram13 ram14 ram15 ram2 ram3 ram4 ram5 ram6 ram7 ram8 ram9 random raw
rfkill serial1 shm snd stderr stdin stdout tty tty0 tty1 tty10 tty11 tty12 tty13 tty14 tty15
tty16 tty17 tty18 tty19 tty20 tty21 tty22 tty23 tty24 tty25 tty26 tty27 tty28 tty29 tty3
tty30 tty31 tty32 tty33 tty34 tty35 tty36 tty37 tty38 tty39 tty4 tty40 tty41 tty42 tty43 tty44
tty45 tty46 tty47 tty48 tty49 tty50 tty51 tty52 tty53 tty54 tty55 tty56 tty57 tty58 tty59
tty6 tty60 tty61 tty62 tty63 tty7 tty8 tty9 ttyAMA0 ttyprintk uhid uinput urandom vchiq vcio
vc-mem vcs vcs1 vcs2 vcs3 vcs4 vcs5 vcs6 vcs7 vcsa vcsa1 vcsa2 vcsa3 vcsa4 vcsa5 vcsa6
vcsa7 vcsa vhci watchdog watchdog0 zero
```

นอกจากนี้ อุปกรณ์สำคัญอื่นๆ เช่น stdin (standard input) stdout (standard output) และ stderr (standard error) นั้นเกี่ยวข้องกับโปรแกรม Terminal ซึ่งเชื่อมโยงกับภาษา C ในการทดลองที่ 5 ภาคผนวก E

```
#include <stdio.h>
```

มีทั้ง
 mmcblk0
 mmcblk0p1
 mmcblk0p2.

*total size of files: ขนาดของไฟล์ทั้งหมด**size on disk: ขนาดในหน่วยจัดเก็บไฟล์ แบ่งตามจ่ารรม (disk)**ต้องมีที่ size on disk จะจะมีหน่วยเป็น byte ไม่ต่อ page. ขนาดเป็น*

L.4 กิจกรรมท้ายการทดลอง

1. ใช้โปรแกรม File Manager แล้วคลิกขวาบนชื่อไฟล์เพื่อแสดงคุณสมบัติ (Properties) ต่างๆ บนแท็บ General และอธิบายโดยเฉพาะหัวข้อ Total size of files และ Size on disk ว่าเหตุใดถึงแตกต่างกัน 4 kB
โดยทั่วไป
2. สร้างไฟล์ (New) ด้วยโปรแกรม nano คือข้อความด้วยตัวอักษรจำนวน 1 ตัวแล้วบันทึก (Save) ใช้คำสั่ง ls -l เพื่อแสดงรายละเอียดของไดเรคทอรีที่บรรจุไฟล์นั้น เพื่อหาขนาดไฟล์ที่แท้จริง ขนาดไฟล์นั้นคือ 2 Bytes
ขนาดไฟล์นั้นคือ 2 Bytes
3. โปรดสังเกตว่าใน test.txt ที่สร้างด้วยโปรแกรม nano เราได้พิมพ์ประโยค fdisk คิดเป็นจำนวน 3 ตัวอักษรฯ ละ 1 byte เท่านั้น จงหาว่า byte ที่ 4 คือตัวอักษรใดในรูปที่ 2.12 มีน้อยกว่า null ('0') ตัวแรกของตัวอักษร ASCII
4. เพิ่มจำนวนตัวอักษรไปเรื่อยๆ ใน test.txt จนทำให้ไฟล์มีขนาดมากกว่าเท่ากับ 4096 byte และใช้คำสั่ง du -B1 test.txt ตรวจสอบขนาดไฟล์อีกรอบ บันทึกและอธิบายผลที่ได้ ขนาดไฟล์ 744 Byte ขนาดบน磁盘 8192 Byte (ขนาด 4096 Byte)
5. งด

```
งดใช้คำสั่ง type แต่จะต้องดูใน stat ระหว่างไดเรคทอรี และไฟล์
```
6. สิทธิ์การเข้าถึง (Permission) ของไดเรคทอรีหรือของไฟล์ประกอบด้วยบิตจำนวน 9 บิต แบ่งเป็น 3 ชุดๆ ละ 3 บิต จงเรียงลำดับชุดต่างๆ ว่าเป็นของสิทธิ์ของใครบ้าง ? III II I
7. งดใช้คำสั่งต่อไปนี้ เพื่อแสดงรายชื่อไดเรคทอรีและไฟล์ และอธิบายผลว่าหมายเลขอุปกรณ์ที่อยู่ด้านซ้ายสุดคืออะไร และเหตุใดจึงมีค่าซ้ำ ? คืออะไร นะครับ?
8. งดใช้คำสั่งต่อไปนี้ เพื่อแสดงรายละเอียดของชื่อไดเรคทอรีคู่ที่ข้ามจากข้อที่แล้ว และอธิบายผลว่ามีอะไรที่แตกต่างกัน เพราะเหตุใด

\$ stat /proc

\$ stat /sys

\$ stat /dev

\$ stat /run

9. งดใช้คำสั่งต่อไปนี้ เพื่อแสดงรายละเอียดของอุปกรณ์ และอธิบายว่ามีผลลัพธ์ที่แตกต่างกันหรือไม่ เพราะเหตุใด block special file

\$ stat /dev/mmcblk0p7 - คืออะไร

\$ stat / directory .

10. งดอธิบายว่าเหตุใดไดเรคทอรี asound จึงอยู่ใต้ /proc ในหัวข้อที่ 1.2.1 การทดลองที่ 9 ภาคผนวก I

11. งดอธิบายความเชื่อมโยงระหว่าง gpiomem ที่ได้จากคำสั่ง ls /dev กับกิจกรรมท้ายการทดลองที่ 10 ภาคผนวก J spin memory