

ภาคผนวก L

การทดลองที่ 12 การศึกษาอุปกรณ์เก็บรักษาข้อมูลและระบบไฟล์

การทดลองนี้อธิบายและเชื่อมโยงเนื้อหาความรู้ของทุกบทเข้าด้วยกัน แต่จะเน้นบทที่ 6 และบทที่ 7 เพื่อให้ผู้อ่านมองเห็นอุปกรณ์อินพุตและเอาต์พุตเหมือนไฟล์แต่ละไฟล์ โดยมีวัตถุประสงค์ดังนี้

- เพื่อให้เข้าใจการวัดขนาดของไฟล์และไดเรกทอรีในระบบไฟล์
- เพื่อให้รู้จักโครงสร้างและระบบไฟล์ของการกำหนดหน่วยความจำไม่โคร SD ที่ใช้งานในปัจจุบัน
- เพื่อให้เข้าใจระบบไฟล์ (File System) ชนิดต่างๆ บนบอร์ด Pi
- เพื่อให้สามารถเชื่อมโยงอุปกรณ์อินพุต/เอาต์พุตชนิดต่างๆ กับระบบไฟล์

L.1 ขนาดของไฟล์และไดเรกทอรี

ผู้อ่านสามารถตรวจสอบขนาดของไฟล์ใดๆ ชื่อ filename ที่แท้จริง หน่วยเป็นไบต์ ด้วยคำสั่ง `du` (Disk Usage) โดยทำตามขั้นตอนต่อไปนี้

- ย้ายไดเรกทอรีปัจจุบันไปที่ `/home/pi` ซึ่งเป็นไดเรกทอรีหลักของผู้ใช้ชื่อ pi

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

- สร้างไฟล์ข้อความ test.txt ด้วยโปรแกรม nano ด้วยคำสั่งต่อไปนี้

```
$ nano test.txt
```

พิมพ์ข้อความ fdd ลงในไฟล์ ทำการ Write โดยกดปุ่ม Ctrl แห่ตามด้วยปุ่ม o ออกจากโปรแกรมโดยกดปุ่ม Ctrl แห่ตามด้วยปุ่ม x

- คำสั่ง 'du -b filename' จะแสดงผลขนาดเป็นจำนวนไบต์นำหน้าชื่อไฟล์นั้น

```
$ du -b test.txt
4 test.txt
```

ตัวเลข 4 หมายถึง เลขจำนวนไบต์ที่คำสั่ง du แสดงผลมาตามพารามิเตอร์ b ที่ส่งไป เพื่อบอกค่าขนาดของไฟล์ test.txt เป็นจำนวน 4 ไบต์

- คำสั่ง 'du -B1 filename' ผู้อ่านสามารถตรวจสอบขนาดของไฟล์ใดๆ ชื่อ filename ที่จัดเก็บเป็นจำนวนเท่าของ 4096 ไบต์ ในอุปกรณ์เก็บรักษาข้อมูล SD ด้วยคำสั่งต่อไปนี้

```
$ du -B1 test.txt
4096 test.txt
```

ตัวเลข 4096 หมายถึง เลขจำนวนไบต์ที่คำสั่ง du แสดงผลมาตามพารามิเตอร์ B1 ที่ส่งไป โดยผู้อ่านจะสังเกตเห็นความแตกต่าง ถึงแม้ไฟล์มีข้อมูลจำนวนน้อยเพียงไม่กี่ไบต์ แต่การจองพื้นที่ในอุปกรณ์สำรองจะมีขนาดเป็นจำนวนเท่าของ 4096 ไบต์เสมอ เช่น 8192, 16384 เป็นต้น

- คำสั่ง 'du -h' จะแสดงผลขนาดหรือจำนวนไบต์โดยใช้หน่วยเช่น K (Kibi: 1024) M (Mebi: 1048576) G (Gibi: 1073741824) นำหน้าชื่อไดเรกทอรีหรือไฟล์เดอ์ที่อยู่ใต้ไดเรกทอรีปัจจุบัน และจัดบันทึก 5 รายการแรกในตาราง

```
$ du -h
```

Size	Folder Name
4.0k	./gnupg/private-keys-v1.d
8.0k	./gnupg
68k	./asm/Lab8
112k	./asm/Lab7
32k	./asm/Lab6

```
t63010484@Pi432b:~ $ du -h
4.0K    ./gnupg/private-keys-v1.d
8.0K    ./gnupg
68K     ./asm/Lab8
112K    ./asm/Lab7
32K     ./asm/Lab6
216K    ./asm
8.0K    ./local/share/nano
12K     ./local/share
16K     ./local
268K    .
```

L.2 ระบบไฟล์

ผู้ใช้หรือผู้ดูแลระบบลินุกซ์ สามารถ**ตรวจสอบ**การใช้งานอุปกรณ์เก็บรักษาข้อมูล เช่น ฮาร์ดดิสก์ไดรฟ์ โซลิดสเตทไดรฟ์ การ์ดหน่วยความจำ SD ได้โดยคำสั่ง

- คำสั่ง **df** (Disk File System) สามารถแสดงรายละเอียดของอุปกรณ์เก็บรักษาข้อมูลในเครื่อง
- คำสั่ง '**df -h**' จะแสดงรายการ ดังต่อไปนี้ จดบันทึก 5 รายการแรกลงในตารางเพื่อเปรียบเทียบกับตารางที่แล้ว

\$ df -h *unkon.byte*

mouse of directory \sub

Filesystem	Size	Used	Available	Use%	Mounted on
<i>/dev/root</i>	<i>29G</i>	<i>6.9G</i>	<i>20G</i>	<i>28%</i>	<i>/</i>
<i>devtmpfs</i>	<i>3.7G</i>	<i>0</i>	<i>3.7G</i>	<i>0%</i>	<i>/dev</i>
<i>tmpfs</i>	<i>3.9G</i>	<i>0</i>	<i>3.9G</i>	<i>0%</i>	<i>/dev/shm</i>
<i>tmpfs</i>	<i>3.9G</i>	<i>25M</i>	<i>3.9G</i>	<i>1%</i>	<i>/run</i>
<i>tmpfs</i>	<i>5.0M</i>	<i>4.0K</i>	<i>5.0M</i>	<i>1%</i>	<i>/run/lock</i>

โดย Size จะแสดงผลขนาดหรือจำนวนไบต์โดยใช้ตัวคูณที่แตกต่างกัน เช่น K (Kibi: 1024) M (Mebi: 1048576) G (Gibi: 1073741824)

```
t63010484@Pi432b:~ $ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root        29G   7.4G   20G   28% /
devtmpfs         3.7G     0   3.7G    0% /dev
tmpfs            3.9G     0   3.9G    0% /dev/shm
tmpfs            3.9G   25M   3.9G    1% /run
tmpfs            5.0M   4.0K   5.0M    1% /run/lock
tmpfs            3.9G     0   3.9G    0% /sys/fs/cgroup
/dev/mmcblk0p1  253M   49M   204M   20% /boot
tmpfs            790M   4.0K   790M    1% /run/user/1000
tmpfs            790M     0   790M    0% /run/user/1069
tmpfs            790M     0   790M    0% /run/user/1060
tmpfs            790M     0   790M    0% /run/user/1031
tmpfs            790M     0   790M    0% /run/user/1014
tmpfs            790M     0   790M    0% /run/user/1036
tmpfs            790M     0   790M    0% /run/user/1008
tmpfs            790M     0   790M    0% /run/user/1041
tmpfs            790M     0   790M    0% /run/user/1024
```

- คำสั่ง 'df -T' จะเพิ่มคอลัมน์ชนิด (Type) ของแต่ละรายการในการแสดงผล และขนาดเป็นจำนวนเท่าของ 1 KiB (KibiByte) (1K) แทน จัดบันทึก 5 รายการที่ตรงกับตารางที่แล้ว

\$ df -T จำนวน block

Filesystem	Type	1K-blocks Used	Available	Use%	Mounted on
/dev/root	ext4	29733356	20719112	28%	/
devtmpfs	devtmpfs	3879284	3879284	0%	/dev
tmpfs	tmpfs	4044148	4044148	0%	/dev/shm
tmpfs	tmpfs	4044148	4018692	1%	/run
tmpfs	tmpfs	5120	5116	1%	/run/lock

```
t63010484@Pi432b:~ $ df -T
Filesystem      Type      1K-blocks    Used Available Use% Mounted on
/dev/root       ext4      29733356 7752916  20719112  28% /
devtmpfs        devtmpfs   3879284      0    3879284   0% /dev
tmpfs           tmpfs     4044148      0    4044148   0% /dev/shm
tmpfs           tmpfs     4044148    25456  4018692   1% /run
tmpfs           tmpfs      5120         4      5116     1% /run/lock
tmpfs           tmpfs     4044148      0    4044148   0% /sys/fs/cgroup
/dev/mmcblk0p1  vfat     258095    49281   208814   20% /boot
tmpfs           tmpfs     808828      4    808824   1% /run/user/1000
tmpfs           tmpfs     808828      0    808828   0% /run/user/1069
tmpfs           tmpfs     808828      0    808828   0% /run/user/1060
tmpfs           tmpfs     808828      0    808828   0% /run/user/1031
tmpfs           tmpfs     808828      0    808828   0% /run/user/1014
tmpfs           tmpfs     808828      0    808828   0% /run/user/1036
tmpfs           tmpfs     808828      0    808828   0% /run/user/1008
tmpfs           tmpfs     808828      0    808828   0% /run/user/1041
tmpfs           tmpfs     808828      0    808828   0% /run/user/1024
tmpfs           tmpfs     808828      0    808828   0% /run/user/1070
```

คำสั่ง

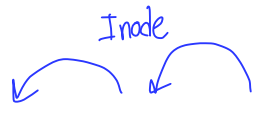
"df -i"

```
t63010484@Pi432b:~ $ df -i
Filesystem      Inodes    IUsed   IFree IUse% Mounted on
/dev/root       1870176 170118 1700058  10% /
devtmpfs        74939    439    74500   1% /dev
tmpfs           157371      1  157370   1% /dev/shm
tmpfs           157371    756  156615   1% /run
tmpfs           157371      3  157368   1% /run/lock
tmpfs           157371    15  157356   1% /sys/fs/cgroup
/dev/mmcblk0p1    0         0        0    - /boot
tmpfs           157371    20  157351   1% /run/user/1000
tmpfs           157371    16  157355   1% /run/user/1069
tmpfs           157371    13  157358   1% /run/user/1060
tmpfs           157371    13  157358   1% /run/user/1031
tmpfs           157371    13  157358   1% /run/user/1014
tmpfs           157371    13  157358   1% /run/user/1036
tmpfs           157371    13  157358   1% /run/user/1008
tmpfs           157371    13  157358   1% /run/user/1041
tmpfs           157371    13  157358   1% /run/user/1024
tmpfs           157371    13  157358   1% /run/user/1070
```


- คำสั่ง 'df -i' จะแสดงรายการต่างๆ ดังนี้ จดบันทึก 5 รายการที่ตรงกับตารางที่แล้ว

\$ df -i

Inode



Filesystem	Inodes	IUsed	IFree	IUse%	Mounted on
/dev/root	1870176	170118	170058	10%	/
devtmpfs	74939	439	74500	1%	/dev
tmpfs	157371	1	157370	1%	/dev/shm
tmpfs	157371	756	156615	1%	/run
tmpfs	157371	3	157368	1%	/run/lock

โดยคอลัมน์ที่ 2 จากทางซ้ายจะแสดงผลเป็นจำนวน **ไอโนด** แทน รายละเอียดเรื่องไอโนด ผู้อ่านสามารถค้นคว้าเพิ่มเติมได้ในบทที่ 7 และทาง [wikipedia](https://www.wikipedia.org)

- คำสั่ง **stat** แสดงรายละเอียดของไฟล์หรือไดเรกทอรี การทดลองนี้จะใช้ไดเรกทอรี asm ที่มีอยู่ และเดิมตัวเลขในช่องว่าง

\$ cd /home/pi
\$ stat asm

```
File: asm
Size: 4096      Blocks: 8      IO Block: 4096  directory
Device: b302h/4582cd Inode: 519088      Links: 3
Access: (0755/drwxr-xr-x)  Uid: (1024/t63010484) Gid: (1024/t63010484)
Access: 2022-02-15 17:04:18.926891186 +0700
Modify: 2022-02-21 14:43:37.663976941 +0700
Change: 2022-02-21 14:43:37.663976941 +0700
Birth: -
```

ผู้อ่านจะต้องกรอกผลลัพธ์ในช่องว่าง ดังต่อไปนี้

```
t63010484@Pi432b:~$ stat asm
File: asm
Size: 4096      Blocks: 8      IO Block: 4096  directory
Device: b302h/4582cd Inode: 519088      Links: 5
Access: (0755/drwxr-xr-x)  Uid: ( 1024/t63010484)  Gid: ( 1024/t63010484)
Access: 2022-02-15 17:04:18.926891186 +0700
Modify: 2022-02-21 14:43:37.663976941 +0700
Change: 2022-02-21 14:43:37.663976941 +0700
Birth: -
```

- ชื่อ asm
- ขนาด 4096 ไบต์ ใช้พื้นที่จำนวน 8 Blocks ซึ่งหมายถึง 8 เซกเตอร์ๆ ละ 512 ไบต์ เป็น directory
- มีหมายเลข Device = b302h/ 4583d หรือเท่ากับ b302₁₆/ 4583₁₀
- มีหมายเลข Inode = 519088₁₀ จำนวน 3 Links
- สิทธิ์เข้าถึง (Access) ด้วยรหัส 0755₁₆ หรือ 111₂:101₂:101₂ โดยผู้ใช้หมายเลข Uid (User ID)= 1023 ชื่อผู้ใช้ (Username)= pi ในรูปหมายเลข Groupid= 1023 ชื่อกรุป t63010484

Meta
Data
ของ asm

*

- เข้าถึง (Access) ... วันที่ 15 กุมภาพันธ์ 2022 เวลา 17:04:18
- เปลี่ยนแปลง (Modify) ... วันที่ 21 กุมภาพันธ์ 2022 เวลา 14:43:37
- เวลาที่ Inode เปลี่ยนแปลง (Change) ... วันที่ 21 กุมภาพันธ์ 2022 เวลา 14:43:37

เบื้องต้นผู้เขียนขอให้ผู้อ่านสร้างไฟล์ผลลัพธ์จากคำสั่ง stat ไปเก็บในไฟล์ เพื่อมาใช้ในการประกอบการทดลองต่อไป

โดย *เขียนลงใน file*
pipe
~~*~~ \$ stat asm > stat_asm.txt

หลังจากนั้น เราสามารถตรวจสอบสถานะของไฟล์ stat_asm.txt ได้ดังนี้

\$ stat stat_asm.txt *เรียก stat ของ file อีกรอบ*

```
t63010484@Pi432b:~$ stat stat_asm.txt
File: stat_asm.txt
Size: 343          Blocks: 8          IO Block: 4096   regular file
Device: b302h/45826d Inode: 399108      Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1024/t63010484)   Gid: ( 1024/t63010484)
Access: 2022-04-05 01:01:27.779625294 +0700
Modify: 2022-04-05 01:01:27.789625070 +0700
Change: 2022-04-05 01:01:27.789625070 +0700
Birth: -
```

File: stat_asm.txt
 Size: 343 Blocks: 8 IO Block: 4096 regular file (คอมมอน)
 Device: b302h/45826d Inode: 399108 Links: 1
 Access: (0644/-rw-r--r--) Uid: (1024/t63010484) Gid: (1024/t63010484)
 Access: ..2022-04-05 01:01:27.779625294 +0700
 Modify: ..2022-04-05 01:01:27.789625070 +0700
 Change: ..2022-04-05 01:01:27.789625070 +0700
 Birth: -

ผู้อ่านจะต้องกรอกผลลัพธ์ในช่องว่าง ดังต่อไปนี้

- ชื่อ stat_asm.txt
- ขนาด 343 ไบต์ ใช้พื้นที่จำนวน 8 Blocks ซึ่งหมายถึง 8 เซ็กเตอร์ๆ ละ 512 ไบต์ เป็น regular file
- มีหมายเลข Device = b302h/45826d หรือเท่ากับ b302₁₆/45826₁₀
- มีหมายเลข Inode = 399108₁₀ จำนวน 1 Links
- สิทธิ์เข้าถึง (Access Permission) ด้วยรหัส 0644₁₆ หรือ 011₂:100₂:100₂ โดยผู้ใช้หมายเลข Uid (User ID)= 1024 ชื่อผู้ใช้ (Username)= t63010484 ในกรุปหมายเลข Groupid=1024 ชื่อกรุป t63010484
- เข้าถึง (Access) ... วันที่ 5 เมษายน 2022 เวลา 01:01:27
- เปลี่ยนแปลง (Modify) ... วันที่ 5 เมษายน 2022 เวลา 01:01:27
- เวลาที่ Inode เปลี่ยนแปลง (Change) ... วันที่ 5 เมษายน 2022 เวลา 01:01:27
- หมายเลข Inode ของ asm กับ หมายเลข Inode ของ stat_asm.txt ตรงกันหรือไม่ เพราะเหตุใด
ไม่ตรงกัน เพราะ disk address มาจากตำแหน่งต่างกัน
- asm เป็น ไดเรกทอรี ในขณะที่ stat_asm.txt เป็น regular file

- สิทธิ์เข้าถึง (Access Permission) รหัส 0765_{16} มีความหมายดังต่อไปนี้

$7_6 - 111_2$: เป็นของใคร เจ้าของ file

$6_6 - 110_2$: เป็นของใคร ผู้รักษากลุ่มเกี่ยวกับเจ้าของ file

$5_6 - 101_2$: เป็นของใคร ผู้เฝ้าประตู

L.3 อุปกรณ์อินพุต/เอาต์พุตในระบบไฟล์

การทดลองในหัวข้อนี้จะเชื่อมต่อกับเนื้อหาในบทที่ 3 และ การทดลองที่ 4 ภาคผนวก D หลักการของระบบปฏิบัติการยูนิกซ์ คือ การเมาท์ (Mount) อุปกรณ์กับไดเรกทอรีด้วยระบบไฟล์ (File System) ที่แตกต่างกัน โดยใช้ชื่อไดเรกทอรีที่แตกต่างกัน โดยมีไดเรกทอรีรูท (Root Directory) หรือโฟลเดอร์รูท เป็นตำแหน่งเริ่มต้น ผู้อ่านสามารถพิมพ์คำสั่งใน Terminal

```
$ mount
```

คำสั่งนี้จะแสดงรายชื่อการเมาท์ หรือ ผูกยึด อุปกรณ์อินพุต/เอาต์พุต เข้ากับระบบไฟล์ที่เหมาะสมกับอุปกรณ์นั้นๆ ด้วยชื่อไดเรกทอรีหรือชื่อไฟล์ของระบบปฏิบัติการ ผู้อ่านจะต้องกรอกผลลัพธ์ที่สำคัญในช่องว่าง และศึกษาคำอธิบายต่อไปนี้

- * **`/dev/mmcblk0p2` on `/` type `ext4` (`rw,noatime`) เป็นระบบไฟล์ `ext4`** ซึ่งเป็นระบบไฟล์หลักของลินุกซ์ ย่อมาจากคำว่า Fourth Extended File System เป็นเวอร์ชันที่ 4 พัฒนาจากชนิด `ext3` ซึ่งพัฒนาจากระบบยูนิกซ์ตามรายละเอียดในหัวข้อที่ 7.1 และ [wikipedia](#)
- `devtmpfs` on `/dev` type `devtmpfs` (`rw, relatime, size=3834564k, nr_inodes=958641, mode=755`)
- `proc` on `/proc` type `proc` (`rw, relatime`) เป็นระบบไฟล์เสมือน (Virtual File System) สำหรับระบบสำคัญต่างๆ เช่น CPU, โดยจะสร้างขึ้นเมื่อบูตเครื่อง และลบทิ้งเมื่อชัตดาวน์ระบบ [รายละเอียดเพิ่มเติมที่ wikipedia](#)
- `sysfs` on `/sys` type `sysfs` (`rw,nosuid,nodev,noexec,relatime`) เป็นระบบไฟล์เสมือน (Virtual File System) [รายละเอียดเพิ่มเติมที่ wikipedia](#)
- `securityfs` on `/sys/kernel/security` type `securityfs` (`rw, nosuid, nodev, noexec, relatime`)
- * **`tmpfs` on `/dev/shm` type `tmpfs` (`rw, nosuid, nodev`)** ย่อมาจากคำว่า Temporary File System [รายละเอียดเพิ่มเติมที่ wikipedia](#)
- `devpts` on `/dev/pts` type `devpts` (`rw, nosuid, noexec, relatime, gid=5, mode=620, ptmxmode=000`) เป็นระบบไฟล์เสมือน (Virtual File System) สำหรับอุปกรณ์อินพุตเอาต์พุตต่างๆ [รายละเอียดเพิ่มเติมที่ wikipedia](#)
- ...

- `/dev/mmcblk0p2` on `/boot type vfat` ระบบไฟล์ **vfat** เป็นส่วนต่อขยายของระบบไฟล์ FAT ซึ่งย่อมาจากคำว่า File Allocation Table เพื่อรองรับชื่อไฟล์ที่ยาวกว่า FAT ที่มา: [wikipedia](https://en.wikipedia.org/wiki/VFAT)
- ...

รายชื่อต่อไปนี้ คือ ตัวเลือกคุณสมบัติ (Attribute) ที่สำคัญของระบบไฟล์ เช่น

- `rw` : read/write สามารถอ่านและเขียนได้
- `noatime` และ `atime`: No/ Access Time หมายถึง ไม่มี/มีการบันทึกเวลาในการสร้าง อ่านหรือเขียนไฟล์ทุกครั้ง
- `relatime` หมายถึง มีการบันทึกเวลาในการสร้าง อ่านหรือเขียนไฟล์ เมื่อเกิดการแก้ไขไฟล์ หรือ การอ่านหรือเข้าถึงไฟล์มากกว่าเวลาที่บันทึกไว้ก่อนหน้านี้อย่างน้อย 24 ชั่วโมง
- `nosuid`: No SuperUser ID เป็นการป้องกันไม่ให้ผู้ดูแลระบบ (SuperUser) กระทำการใดๆ ได้ เพื่อความมั่นคงปลอดภัย
- `noexec`: No Execution เพื่อตั้งค่าไม่ให้รันไฟล์ที่อยู่ในไดเรกทอรีนี้ได้ เช่น ไฟล์ที่เป็นไวรัสหรือมัลแวร์ (Malware) ที่แอบแฝงเข้ามา
- `nodev`: No Device หมายถึง การไม่อนุญาตให้สร้างหรืออ่านโหนด (Node) ซึ่งเป็นไฟล์ชนิดพิเศษ
- `mode` หมายถึง สิทธิ์การเข้าถึงไฟล์หรือไดเรกทอรี ประกอบด้วย บิตควบคุม Read Write Execute 3 ชุด รวมทั้งหมด 9 บิต ซึ่งได้อธิบายแล้วในหัวข้อที่ 7.1.4

ผู้อ่านสามารถ แสดง รายชื่อไฟล์หรือไดเรกทอรีหรือชื่ออุปกรณ์ภายใต้ไดเรกทอรี `/dev` โดยพิมพ์คำสั่งบนโปรแกรม Terminal

```
$ ls /dev
```

ผู้อ่านต้องเปรียบเทียบกับชื่ออุปกรณ์ที่ผู้เขียนตัวหนาไว้ว่าตรงกันหรือไม่ อย่างไร เพื่อให้ผู้อ่านมองเห็นชัดว่า **mmcblk0p2** มีอยู่จริงและระบบได้ทำการเม้าท์เข้ากับไดเรกทอรีรูท (Root) นั่นคือ ไดเรกทอรี `/` ด้วยชนิด `ext4` ตามที่ได้แสดงในคำสั่งก่อนหน้านี้แล้ว

```
ashmem autofs block btrfs-control bus cachefiles cec0 cec1 char console cpu_dma_latency
cuse disk dma_heap dri fb0 fd full fuse gpiochip0 gpiochip1 gpiochip2 gpiomem hidraw0
hidraw1 hidraw2 hidraw3 hwrng i2c-20 i2c-21 initctl input kmsg kvm log loop0 loop1 loop2
loop3 loop4 loop5 loop6 loop7 loop-control mapper media0 media1 mem mmcblk0
mmcblk0p1 mmcblk0p2 mqueue net null port ppp ptmx pts ram0 ram1 ram10 ram11
ram12 ram13 ram14 ram15 ram2 ram3 ram4 ram5 ram6 ram7 ram8 ram9 random raw rkill
rpivid-h264mem rpivid-hevcmmem rpivid-initc rpivid-vp9mem serial1 shm snd stderr stdin
stdout tty tty0 ... ttyAMA0 ttyprintk uhid uinput urandom vchiq vcio vc-mem vcs ... watchdog
watchdog0 zero
```

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

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


L.4 กิจกรรมท้ายการทดลอง เล็กท่า 2 ข้อ

1. จงใช้โปรแกรม File Manager แล้วคลิกขวาบนชื่อไฟล์เพื่อแสดงคุณสมบัติ (Properties) ต่างๆ บนแท็บ General และอธิบายโดยเฉพาะหัวข้อ Total size of files และ Size on disk ว่าเหตุใดถึงแตกต่างกัน
2. สร้างไฟล์ (New) ด้วยโปรแกรม nano คีย์ข้อความด้วยตัวอักษรจำนวน 1 ตัวแล้วบันทึก (Save) ใช้คำสั่ง `ls -l` เพื่อแสดงรายละเอียดของไดเรกทอรีที่บรรจุไฟล์นั้น เพื่อหาขนาดไฟล์ที่แท้จริง
3. โปรดสังเกตว่าใน test.txt ที่สร้างด้วยโปรแกรม nano เราได้พิมพ์ประโยค fdd คิดเป็นจำนวน 3 ตัวอักษร ละ 1 ไบต์เท่านั้น จงหาว่าไบต์ที่ 4 คือตัวอักษรใดในรูปที่ 2.12
4. เพิ่มจำนวนตัวอักษรไปเรื่อยๆ ใน test.txt จนทำให้ไฟล์มีขนาดมากกว่าเท่ากับ 4096 ไบต์ แล้วใช้คำสั่ง `du -B1 test.txt` ตรวจสอบขนาดไฟล์อีกรอบ บันทึกและอธิบายผลที่ได้โดยเฉพาะจำนวน Blocks ที่ได้ จากคำสั่งว่าเท่ากับกี่เซกเตอร์
5. จงเปรียบเทียบผลลัพธ์ของคำสั่ง `stat` ระหว่าง ไดเรกทอรี และ ไฟล์
6. สิทธิ์การเข้าถึง (Permission) ของไดเรกทอรีหรือของไฟล์ประกอบด้วยบิตจำนวน 9 บิต แบ่งเป็น 3 ชุดๆ ละ 3 บิต จงเรียงลำดับชุดต่างๆ ว่าเป็นของสิทธิ์ของใครบ้าง
7. จงใช้คำสั่งต่อไปนี้ เพื่อแสดงรายชื่อไดเรกทอรีและไฟล์ และอธิบายผลว่าหมายเลขที่อยู่ด้านซ้ายสุดคืออะไร และเหตุใดจึงมีค่าซ้ำ

```
$ ls -li -l /
```

8. จงใช้คำสั่งต่อไปนี้ เพื่อแสดงรายละเอียดของชื่อไดเรกทอรีคู่ที่ซ้ำจากข้อที่แล้ว และอธิบายผลว่ามีอะไรที่แตกต่างกัน เพราะเหตุใด

```
$ stat /proc
```

```
$ stat /sys
```

```
$ stat /dev
```

```
$ stat /run
```

9. จงใช้คำสั่งต่อไปนี้ เพื่อแสดงรายละเอียดของอุปกรณ์ และอธิบายว่ามีผลลัพธ์ที่แตกต่างกันหรือไม่ เพราะเหตุใด

```
$ stat /dev/mmcb1k0p2
```

```
$ stat /
```

10. จงอธิบายว่าเหตุใดไดเรกทอรี asound จึงอยู่ใต้ /proc ในหัวข้อที่ I.2.3 การทดลองที่ 9 ภาคผนวก I
11. จงอธิบายความเชื่อมโยงระหว่าง gpiomem ที่ได้จากคำสั่ง `ls /dev` กับกิจกรรมท้ายการทดลองที่ 10 ภาคผนวก J

① Total size of files = ขนาดที่แท้จริงของ file นั้น

Size on disk = ขนาดของพื้นที่ที่จัดเก็บ file นั้น ซึ่งประกอบด้วยส่วนที่ว่างที่จัดเก็บ file
ด้วยการแบ่งพื้นที่เป็น Allocate Unit

∴ สรุปได้ว่า Size on disk จะมีขนาดมากกว่าหรือเท่ากับ Total size of files เสมอ

② 2 bytes

```
t63010484@Pi432b:~ $ nano New.txt
t63010484@Pi432b:~ $ ls -l
total 20
drwxr-xr-x 5 t63010484 t63010484 4096 Feb 21 14:43 asm
-rw----- 1 t63010484 t63010484 1 Feb 21 17:39 main.c.save
-rw-r--r-- 1 t63010484 t63010484 2 Apr 5 01:07 New.txt
-rw-r--r-- 1 t63010484 t63010484 343 Apr 5 01:01 stat_asm.txt
-rw-r--r-- 1 t63010484 t63010484 4 Apr 5 00:53 test.txt
t63010484@Pi432b:~ $ du -b New.txt
2 New.txt
```