



OKay Training

The Best solution for seminar training

Basic Linux Admin for iT Staff

Day 1: พื้นฐานการใช้งานและการติดตั้ง Linux

ประวัติความเป็นมาของ Linux

ประวัติของ Linux

- Linux คือระบบปฏิบัติการโอเพนซอร์สที่มีพื้นฐานมาจาก Unix
- เริ่มต้นพัฒนาโดย Linus Torvalds ในปี 1991 โดยใช้หลักการแบ่งปันโค๊ดเพื่อพัฒนาโปรแกรม
- โครงการนี้เกิดจากความต้องการพัฒนาระบบปฏิบัติการที่ใช้งานได้ฟรีและเข้าถึงได้ง่าย
- Timeline สำคัญ:
 - 1991: Linus Torvalds พัฒนา Linux เวอร์ชันแรก (Kernel เวอร์ชัน 0.01)
 - 1994: เปิดตัว Linux Kernel 1.0 ที่มีการพัฒนาเพิ่มเติมและได้รับการสนับสนุนมากขึ้น

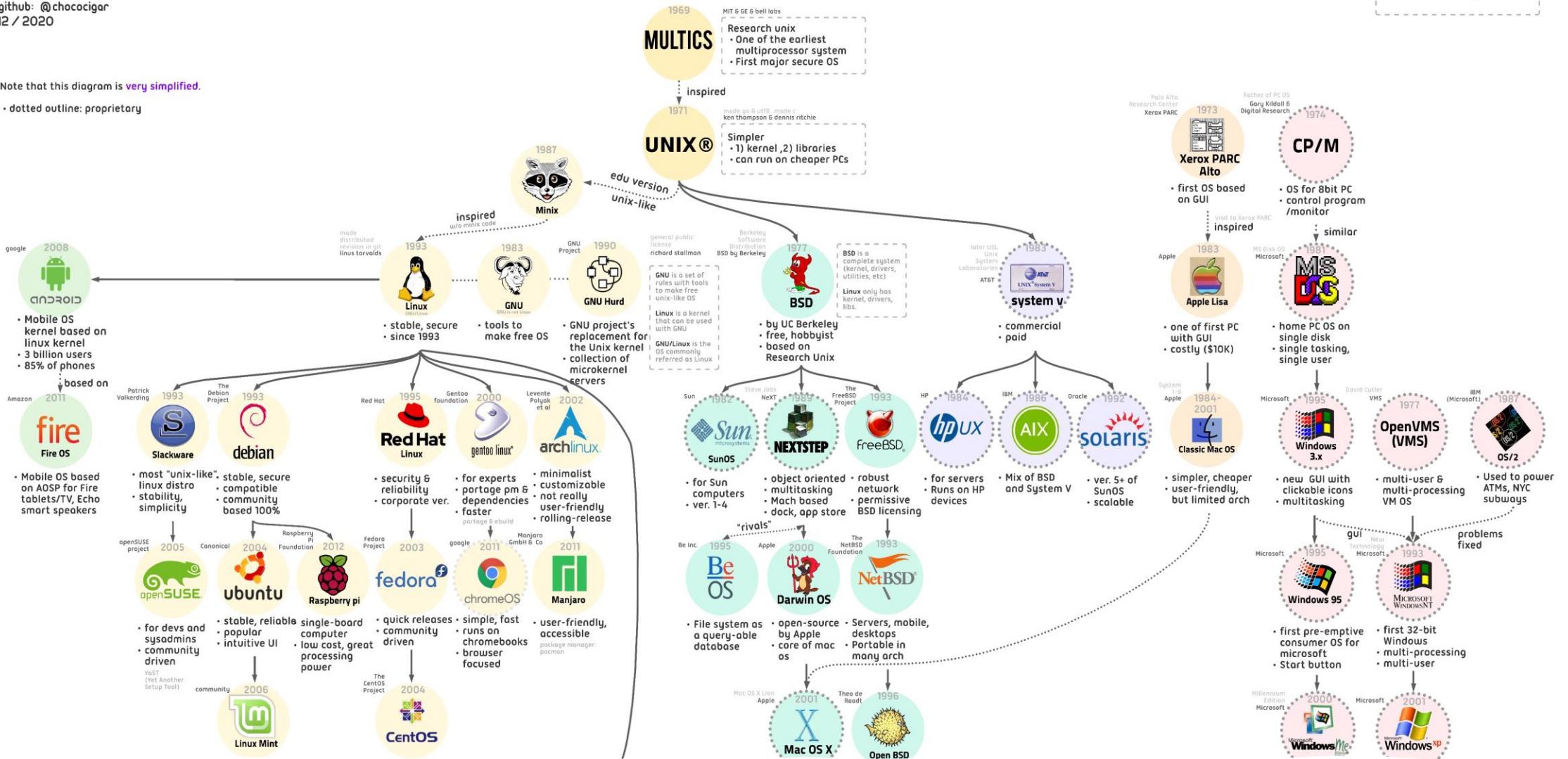


SIMPLE HISTORY OF OS

for any corrections, please contact
github: @chococigar
12 / 2020

Note that this diagram is **very simplified**.

- dotted outline: proprietary



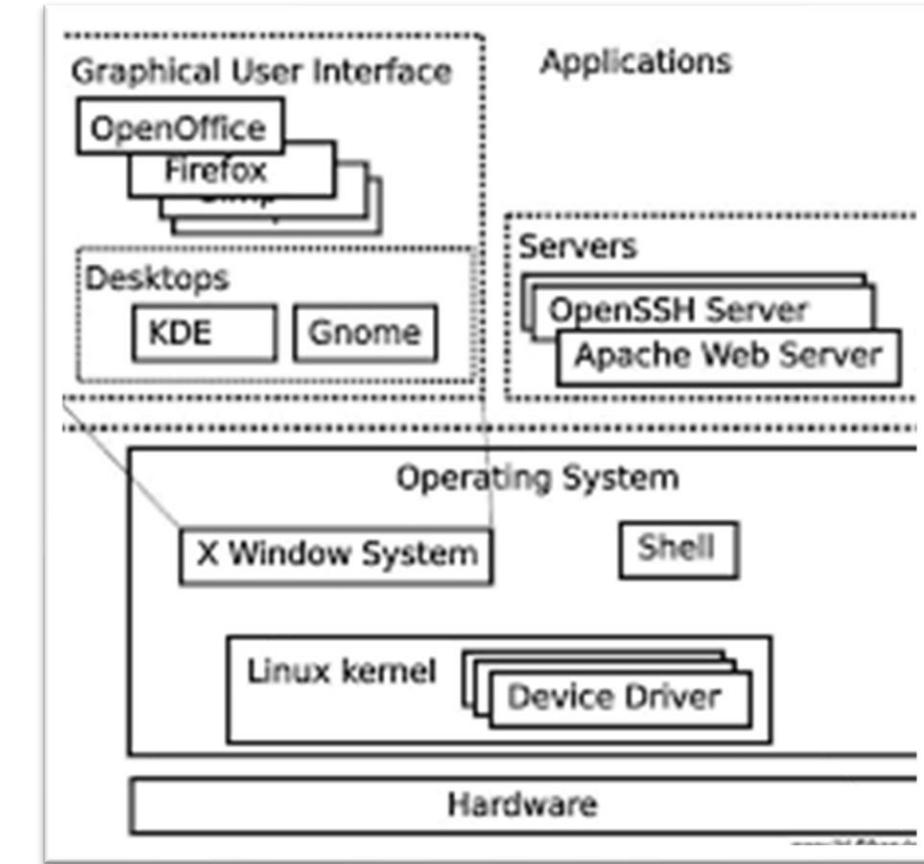
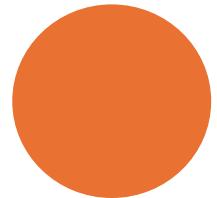
แนวคิดของ Linus Torvalds

- Linus Torvalds พัฒนา Linux โดยใช้แนวคิด Free Software และ Open Source
- จุดเริ่มต้นเกิดจากความต้องการสร้างระบบปฏิบัติการที่เข้าสามารถปรับแต่งเองได้
- Linux ถูกพัฒนาแบบ Community-driven ซึ่งผู้ใช้ก็สามารถร่วมพัฒนาและปรับปรุงโค้ดได้
- วัฒนธรรมการทำงานร่วมกัน: ผู้ใช้สามารถเสนอการแก้ไข, เพิ่มฟีเจอร์ใหม่ ๆ และเรียนรู้การพัฒนาระบบได้จาก Linux



ความสำคัญของ Unix และการแตกต่อมาเป็น Linux

- Unix เป็นระบบปฏิบัติการที่พัฒนาขึ้นในช่วงปี 1970s โดย AT&T Bell Labs
- Unix มีโครงสร้างที่แข็งแกร่งและได้รับการยอมรับในวงการคอมพิวเตอร์ เน้นการทำงานร่วมกับฮาร์ดแวร์ หลากหลายชนิด
- Linus นำโครงสร้างของ Unix มาใช้ใน Linux โดยปรับปรุงให้เหมาะสมและปรับให้เป็นโอเพนซอร์ส
- การนำ Unix มาเป็นพื้นฐานช่วยให้ Linux สามารถใช้กับอุปกรณ์หลากหลายประเภท รวมถึงเซิร์ฟเวอร์, เครื่องคómพิวเตอร์ส่วนบุคคล, และอุปกรณ์พกพา



ประโยชน์ของ Linux และระบบโอเพนซอร์ส

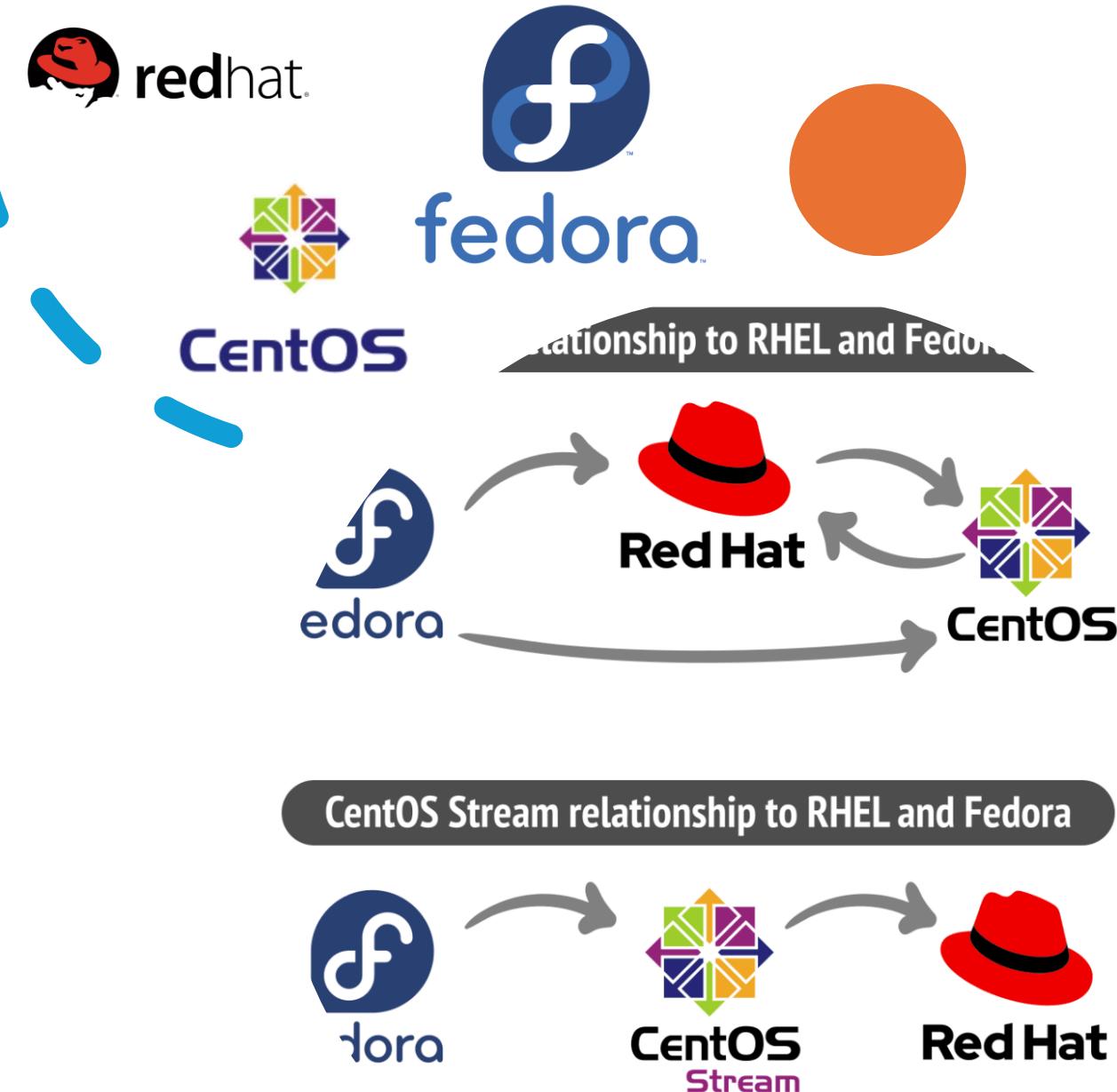
- ค่าใช้จ่ายต่ำ: ไม่มีค่าใช้จ่ายในการใช้งานระบบปฏิบัติการและซอฟต์แวร์โอเพนซอร์ส
- การพัฒนาและปรับแต่ง: ผู้ใช้สามารถปรับแต่งระบบปฏิบัติการได้เอง ซึ่งหมายความว่าการใช้งานที่ต้องการความยืดหยุ่น
- ความปลอดภัยสูง: ระบบโอเพนซอร์สมีความปลอดภัยเนื่องจากมีการพัฒนาตรวจสอบจากชุมชนทั่วโลก
- สนับสนุนการเรียนรู้และพัฒนา: เปิดโอกาสให้ผู้ที่สนใจศึกษาและเข้าร่วมการพัฒนาซอฟต์แวร์ได้
- การใช้งานหลากหลาย: ใช้งานได้กับหลายอุปกรณ์ เช่น คอมพิวเตอร์, เชิร์ฟเวอร์, และอุปกรณ์พกพา



ประวัติของ Red Hat, CentOS และ Fedora

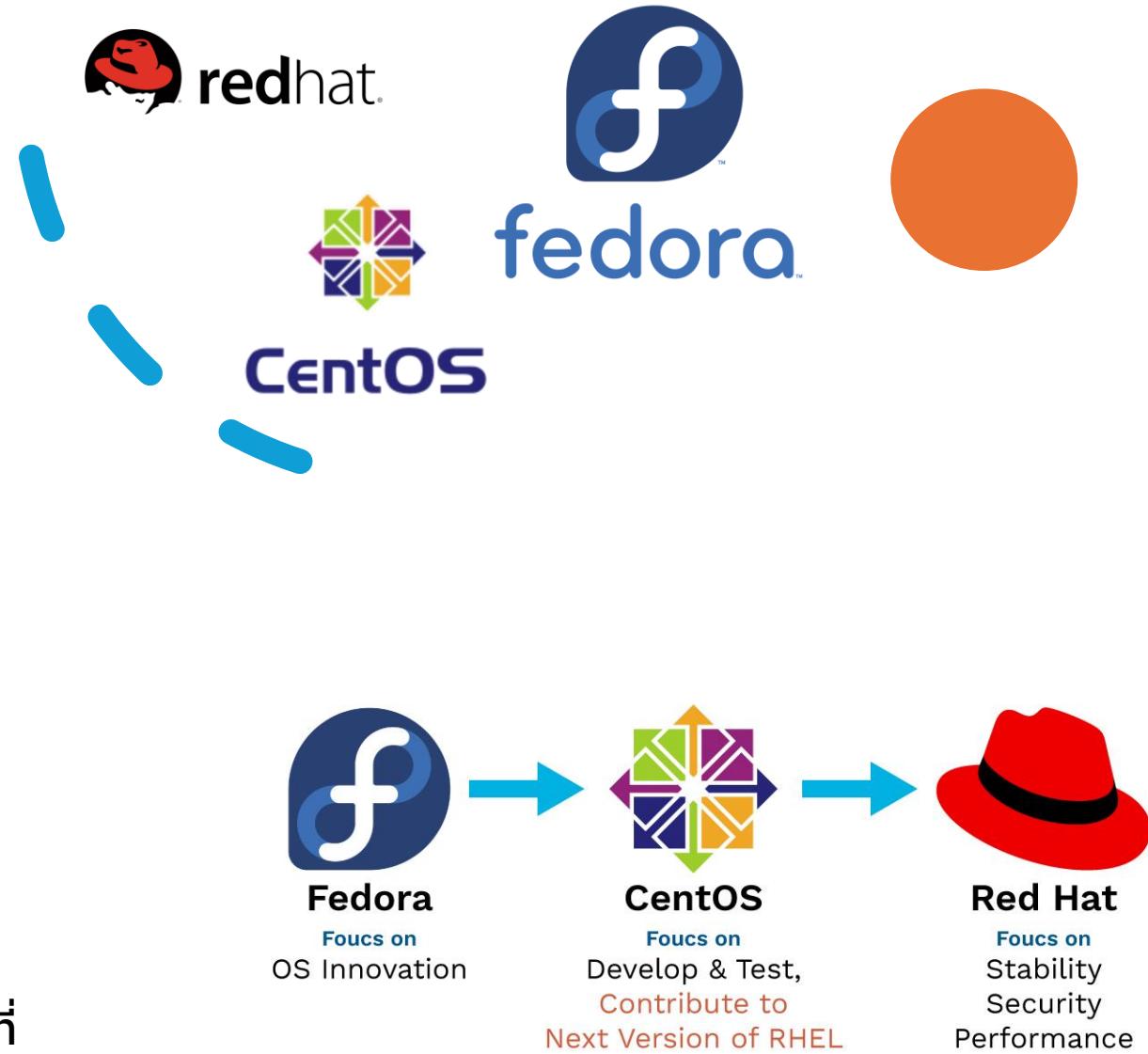
ประวัติและความสัมพันธ์ระหว่าง Red Hat, CentOS, และ Fedora

- Red Hat: เปิดตัวครั้งแรกในปี 1993 โดย Red Hat, Inc. เป็นระบบปฏิบัติการ Linux เชิงพาณิชย์ที่เน้นการสนับสนุนองค์กร
- Fedora: พัฒนาขึ้นในปี 2003 โดย Fedora Project และ Red Hat สนับสนุนให้ใช้เทคโนโลยีใหม่ ๆ เพื่อเป็น “ห้องทดลอง” สำหรับการพัฒนาและทดสอบเฟิร์มแวร์ใหม่ ก่อนที่จะนำไปใช้ใน Red Hat Enterprise Linux (RHEL)
- CentOS: ก่อตั้งในปี 2004 เป็นเวอร์ชันฟรีที่ใช้โค้ดของ RHEL โดยเน้นความเสถียรในระดับองค์กรและมีเป้าหมายคล้ายกับ RHEL แต่ไม่มีการสนับสนุนจากการค้า
- Red Hat ได้ซื้อ CentOS Project ในปี 2014 เพื่อรวมการพัฒนาและการจัดการ ทำให้ CentOS กลายเป็นส่วนสำคัญของชุมชนผู้ใช้ Red Hat



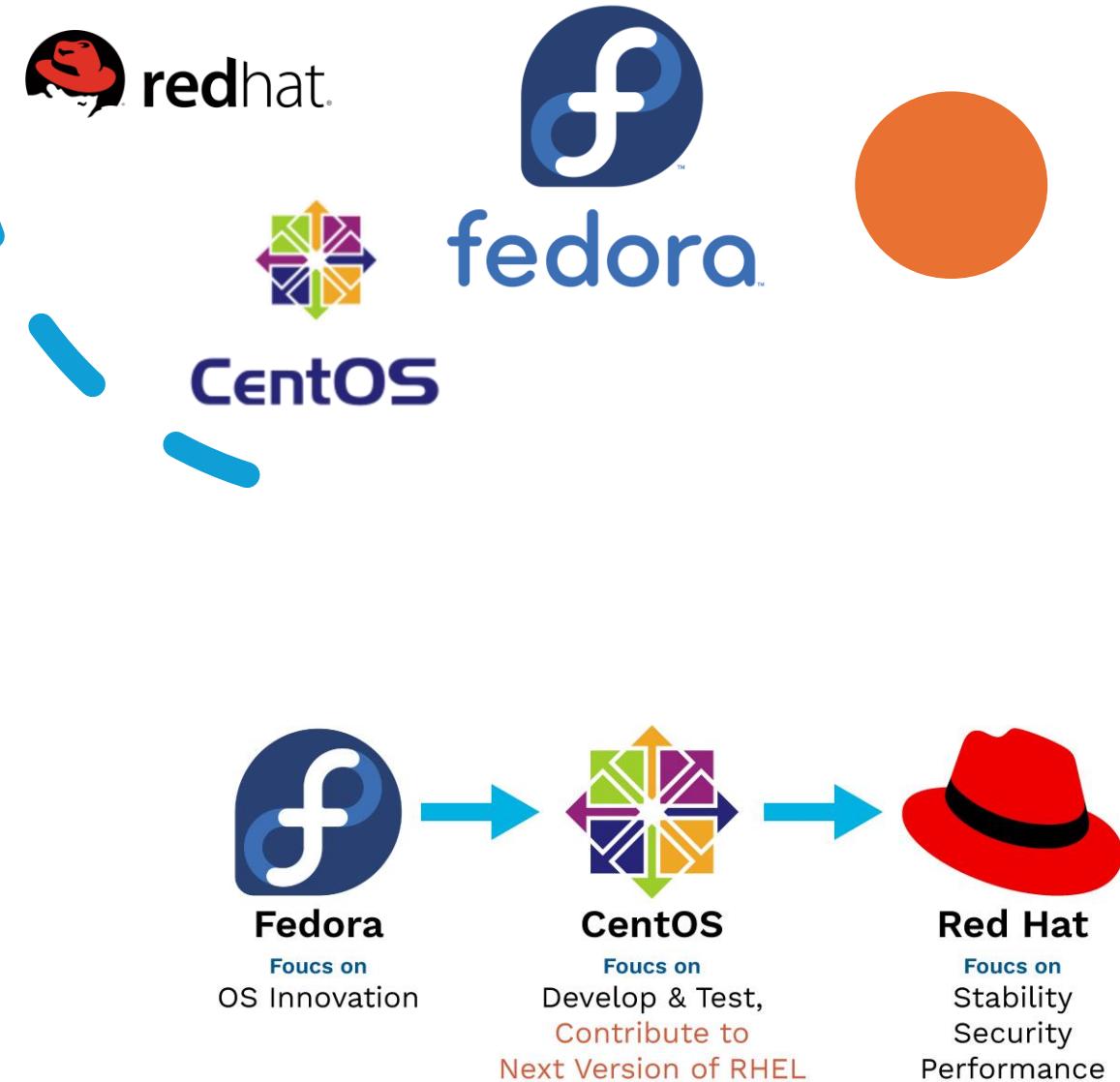
ความแตกต่างในด้านการใช้งานและเป้าหมายของแต่ละ OS

- Red Hat Enterprise Linux (RHEL):
 - เน้นการใช้งานระดับองค์กรและองค์กรธุรกิจที่ต้องการระบบเสถียรและการสนับสนุนการทำงานค้า
 - ใช้ในองค์กรขนาดใหญ่ที่ต้องการการสนับสนุนระยะยาว (LTS) และการรักษาความปลอดภัย
- CentOS:
 - มีโครงสร้างเหมือนกับ RHEL และสามารถใช้งานฟรี
 - เมฆะกับการใช้งานในองค์กรที่ไม่ต้องการการสนับสนุนจากการค้า แต่ต้องการความเสถียรในระดับองค์กร
 - CentOS Stream: ปัจจุบัน CentOS เปลี่ยนเป็น CentOS Stream ที่เน้นการพัฒนาและทดสอบฟีเจอร์ใหม่ๆ ให้แก่ RHEL ในอนาคต
- Fedora:
 - เน้นการนำเทคโนโลยีใหม่ล่าสุดมาใช้ เช่น การสนับสนุนไลบรารีและเครื่องมือใหม่ ๆ
 - เมฆะกับนักพัฒนาและผู้ที่ต้องการระบบปฏิบัติการที่กันสมัยและอัพเดตบ่อยครั้ง



การสนับสนุนและความเป็นชุมชนโอเพนซอร์ส

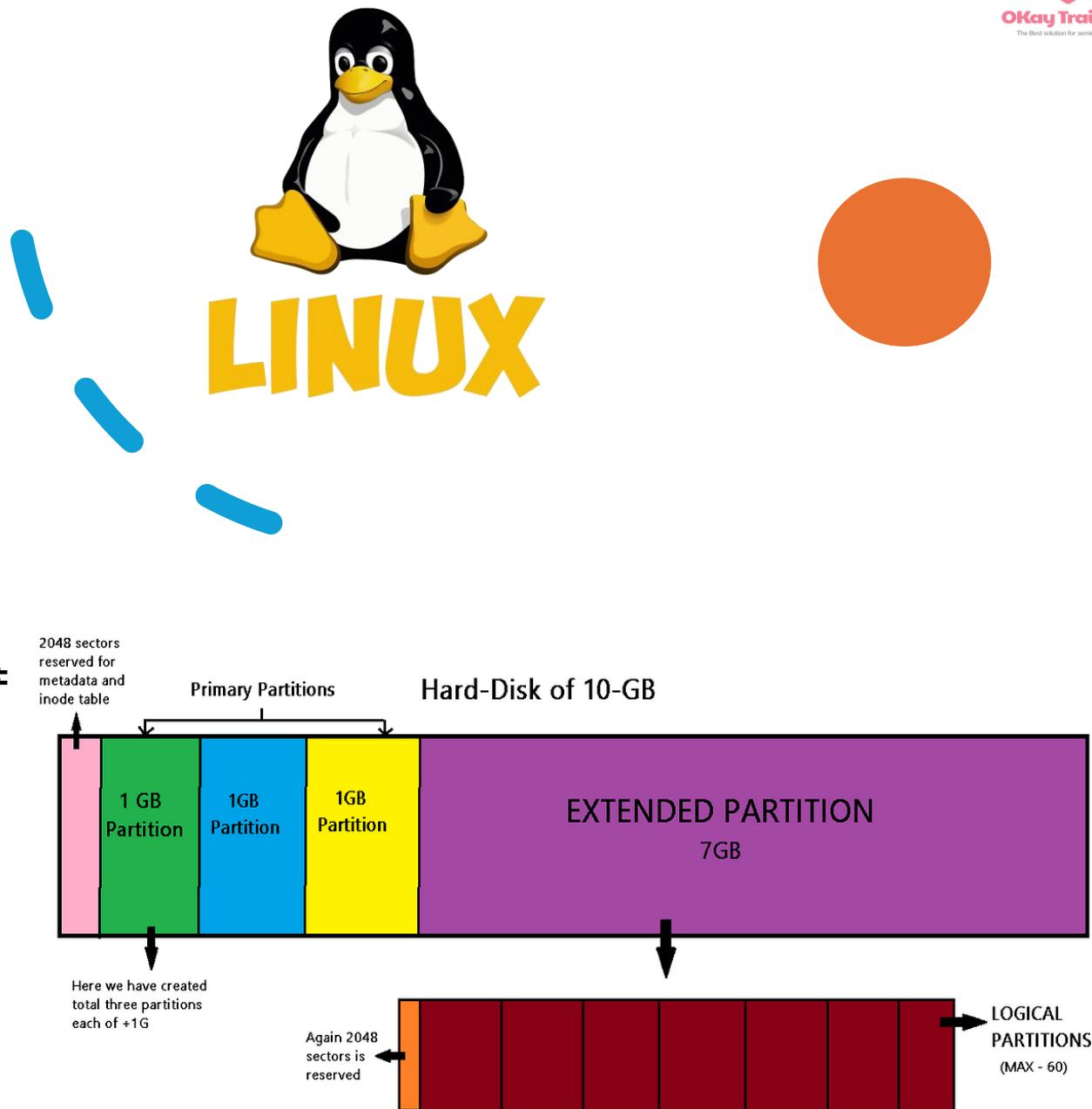
- Red Hat Enterprise Linux (RHEL):
 - ให้การสนับสนุนอย่างเป็นทางการผ่านสัญญาบริการที่มีค่าใช้จ่ายจาก Red Hat ซึ่งรวมถึงการอัปเดตด้านความปลอดภัย, การให้คำปรึกษา, และการสนับสนุนในระยะยาว
- CentOS:
 - ขับเคลื่อนโดยชุมชนโอเพนซอร์สและมีการสนับสนุนจากผู้ใช้ทั่วไปในชุมชน
 - ไม่มีการสนับสนุนเชิงพาณิชย์โดยตรง แต่มีการพัฒนาร่วมกับ RHEL
- Fedora:
 - สนับสนุนโดยชุมชน Fedora Project และ Red Hat อย่างไม่เป็นทางการ ซึ่งเป็นที่รู้จักว่าเป็นการสนับสนุนที่รวดเร็วสำหรับฟีเจอร์ใหม่ ๆ และการแก้ไขปัญหา
- Red Hat ใช้ Fedora และ CentOS เป็นแหล่งข้อมูลสำหรับการทดสอบและพัฒนาฟีเจอร์ใหม่ที่จะนำเข้าสู่ RHEL ในอนาคต



พาร์ติชันใน Linux

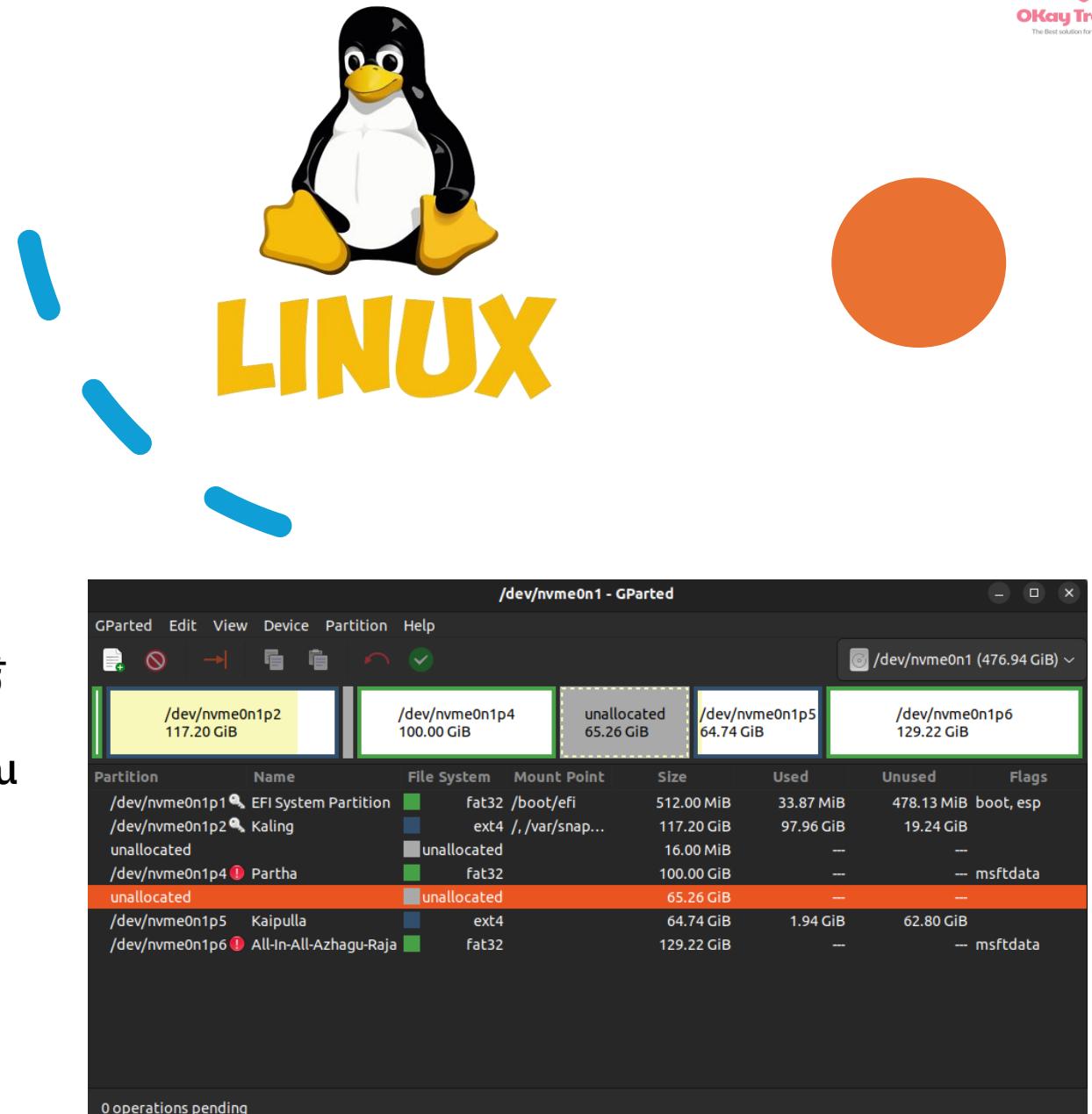
พาร์ติชันใน Linux คืออะไร

- พาร์ติชัน (Partition) คือการแบ่งพื้นที่เก็บข้อมูลบนดิสก์ออกเป็นส่วน ๆ เพื่อให้สามารถจัดการและใช้งานได้อย่างเป็นระเบียบ
- พาร์ติชันใน Linux ทำให้เราสามารถจัดการข้อมูลได้ดีขึ้น แยกการใช้งานของระบบออกจากข้อมูลของผู้ใช้ และเพิ่มประสิทธิภาพการทำงาน
- การจัดพาร์ติชันที่ดีช่วยให้ระบบสามารถทำงานได้เสถียร และง่ายต่อการบำรุงรักษา
- ประเภทหลักของพาร์ติชันใน Linux: มีพาร์ติชันหลายประเภทที่มีบทบาทเฉพาะ เช่น /boot, /home, /swap, และ / (root)



พาร์ติชันใน Linux คืออะไร

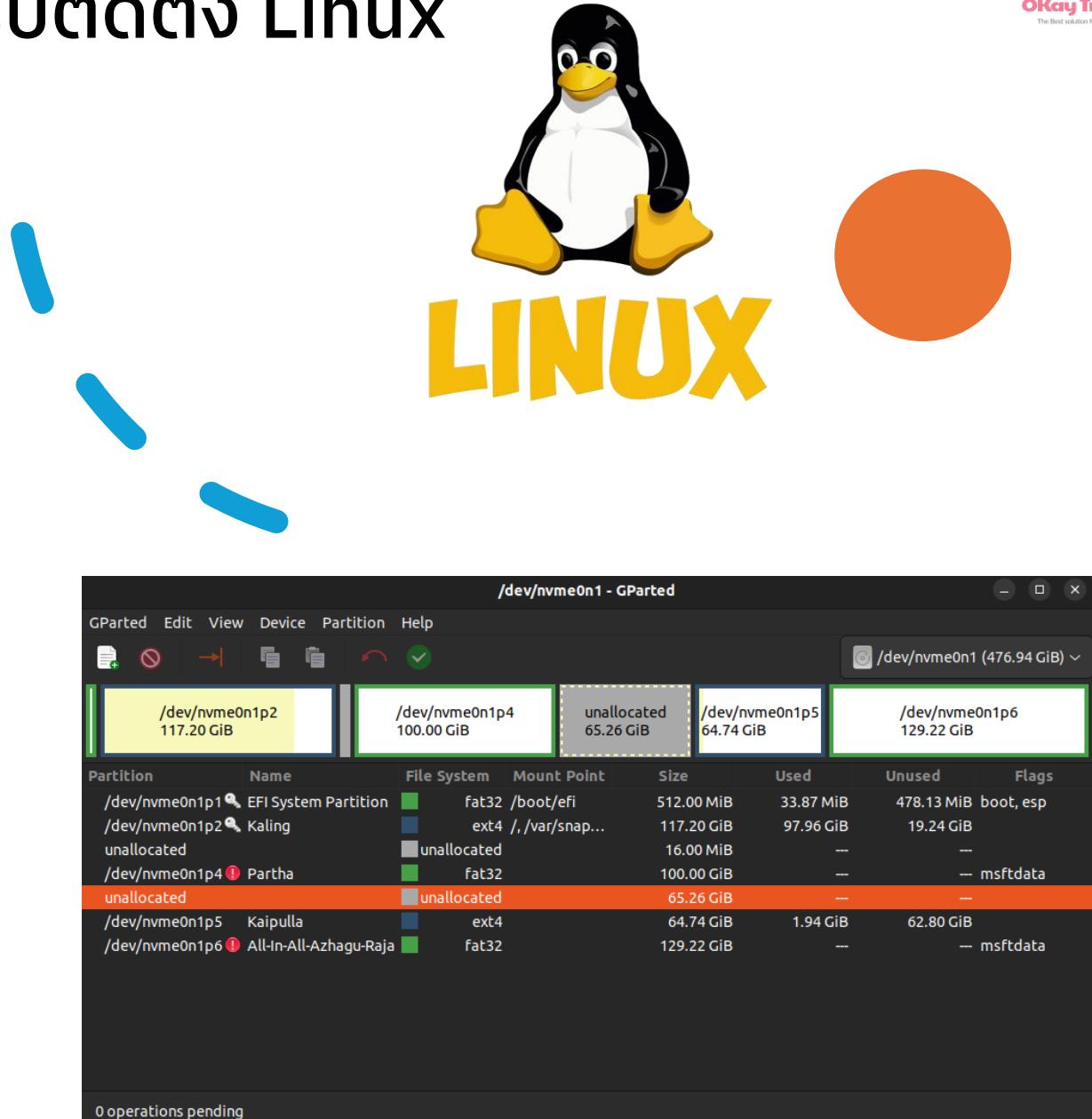
- / (root): เป็นพาร์ติชันหลักที่เก็บไฟล์ระบบกั้งหมด เป็นจุดเริ่มต้นของระบบไฟล์ใน Linux
- /boot: เก็บไฟล์ที่จำเป็นสำหรับการบูตระบบ เช่น ไฟล์ Kernel และ GRUB (bootloader) ขนาดกัววไปของพาร์ติชันนี้คือประมาณ 512MB
- /home: ใช้เก็บข้อมูลส่วนตัวของผู้ใช้กั้งหมด เช่น ไฟล์เอกสาร รูปภาพ และการตั้งค่าส่วนตัว แยกจากพาร์ติชัน root เพื่อป้องกันการเสียหายของข้อมูลหากเกิดข้อผิดพลาดกับระบบ
- /swap: กำหนดที่เป็นหน่วยความจำเสมือนสำหรับระบบ ใช้เมื่อ RAM ไม่เพียงพอในการประมวลผล ขนาดของ swap ขึ้นอยู่กับปริมาณ RAM ที่ใช้งาน (โดยกัววไปมีขนาดประมาณ 1-2 เท่าของ RAM)



ขั้นตอนการเตรียมพาร์ติชันสำหรับติดตั้ง Linux



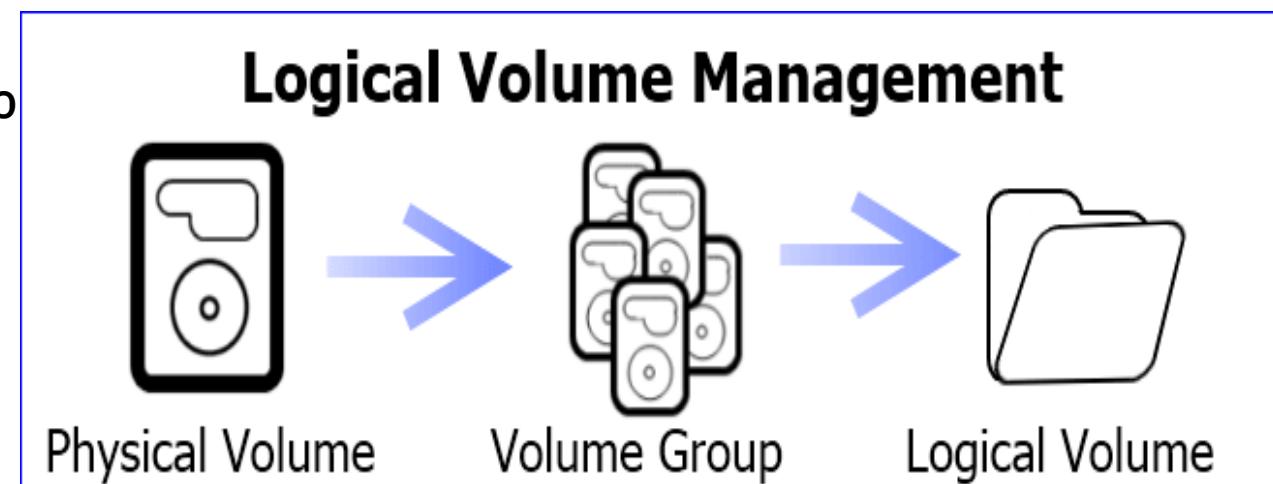
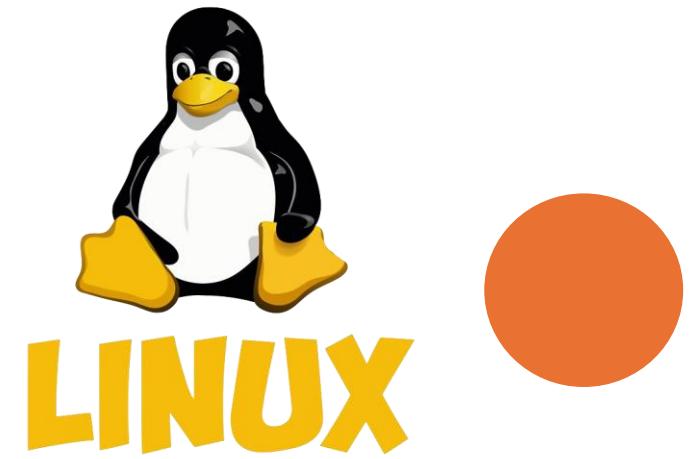
- วางแผนพาร์ติชัน: គຽມกำหนดขนาดของพาร์ติชัน ແຕ່ລະປະເກດຕາມການໃຊ້ງານ ເຊັ່ນ ຂາດຂອງ /boot, /home, ແລະ /swap
 - ການສ່ວນພາບ: ໃຫ້ໂປຣແກຣມທີ່ມາພຣົວມກັບ Linux Installer ເຊັ່ນ GParted ມີ Disk Utility ເພື່ອສ່ວນ ພາບທີ່ຈຳເປັນ
 - ການຝອດແມຸຕພາບ: ເລືອກຮູບແບບໄຟລະບບ (File System) ທີ່ເໜີມສຳເນົາ ເຊັ່ນ ext4 ສໍາຮັບ / ແລະ /home, swap ສໍາຮັບ /swap
 - ຕັ້ງຄ່າການເມາກົງ: ຮະບູພາບທີ່ຕ່າງໆ ແລະ ກຳນົດຈຸດເມາກົງທີ່ເໜີມສຳເນົາ ເຊັ່ນ /home, /boot ແລະ / ໃຫ້ຕຽນກັບການໃຊ້ງານ



LVM (Logical Volume Manager) และ เครื่องมือที่เกี่ยวข้อง

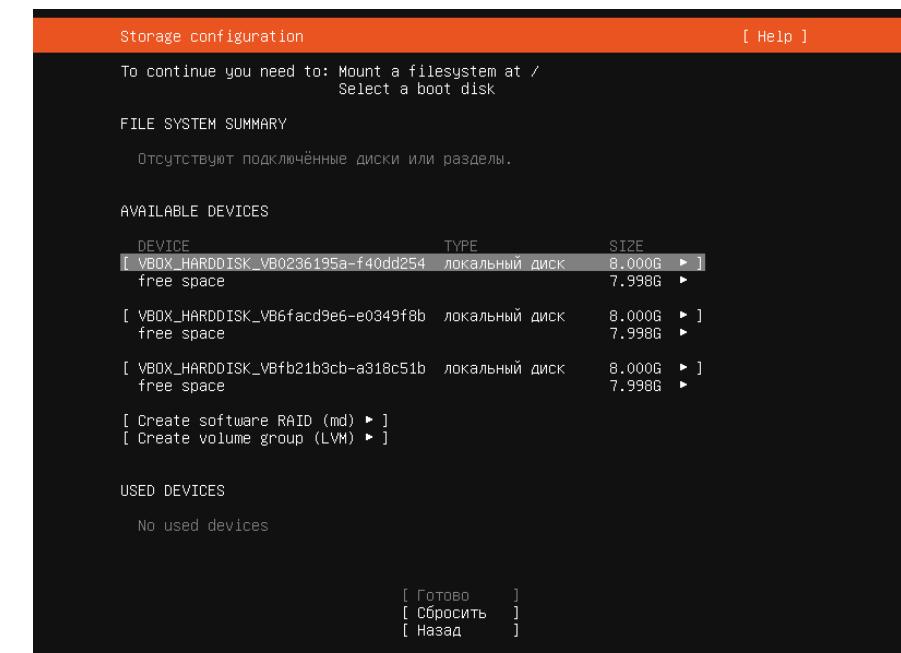
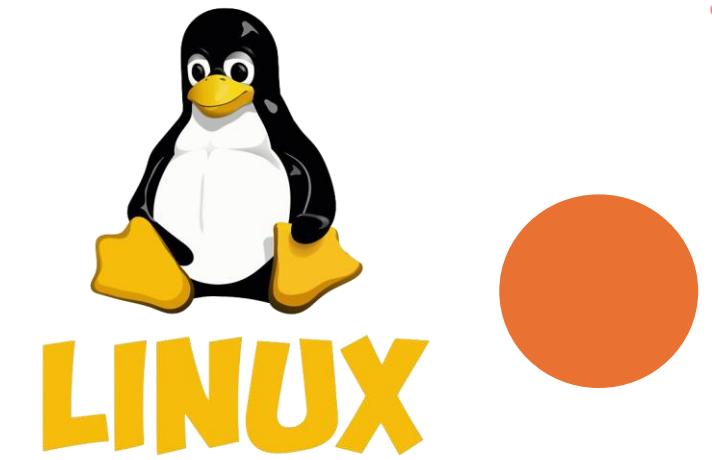
LVM (Logical Volume Manager) คืออะไร

- LVM (Logical Volume Manager) คือเทคโนโลยีสำหรับการจัดการดิสก์ใน Linux ที่ช่วยให้การเพิ่ม, ลด, และจัดการพื้นที่ดิสก์ทำได้อย่างยืดหยุ่น
- LVM ช่วยให้สามารถรวมดิสก์หลายลูกเข้าด้วยกันและแบ่งพื้นที่จัดเก็บข้อมูลได้ตามความต้องการของระบบ
- ประโยชน์ของ LVM:
 - การขยายพื้นที่โดยไม่ต้องหยุดระบบ: LVM ช่วยให้สามารถเพิ่มพื้นที่ได้โดยไม่ต้องรีบูต
 - ความยืดหยุ่นในการจัดสรรพื้นที่เก็บข้อมูล: สามารถปรับขนาดได้ตามการใช้งาน
 - การสำรองข้อมูล: การสร้าง Snapshot เพื่อการสำรองข้อมูลและการกู้คืนข้อมูลได้สะดวกและรวดเร็ว



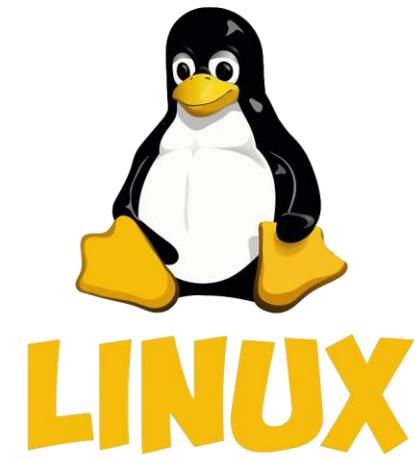
คำสั่งพื้นฐานสำหรับการจัดการ LVM

- คำสั่งในการสร้างและจัดการ Physical Volume (PV), Volume Group (VG), และ Logical Volume (LV):
 - `pvcreate`: ใช้ในการสร้าง Physical Volume (PV) บนดิสก์ หรือパーティชันดิสก์ เพื่อเตรียมให้ LVM ใช้งานได้
 - ตัวอย่าง: `pvcreate /dev/sdb1`
 - `vgcreate`: ใช้ในการสร้าง Volume Group (VG) โดยรวม Physical Volumes หลายตัวเข้าด้วยกัน
 - ตัวอย่าง: `vgcreate myvg /dev/sdb1 /dev/sdc1`
 - `lvcreate`: ใช้ในการสร้าง Logical Volume (LV) บน Volume Group ที่สร้างไว้ สามารถบุധนาดและซื้อได้
 - ตัวอย่าง: `lvcreate -L 10G -n mylv myvg`
- คำสั่งเพิ่มเติมที่สำคัญ:
 - `lvextend`: ใช้ในการขยาย Logical Volume
 - `lvreduce`: ใช้ในการลดขนาด Logical Volume (แนะนำให้สำรองข้อมูลก่อน)
 - `lvremove`: ใช้ในการลบ Logical Volume



โครงสร้างและการทำงานของ LVM

- ขั้นตอนการสร้าง LVM:
 - สร้าง Physical Volume (PV) บันเดิส์ก
 - รวม PV หลายตัวเป็น Volume Group (VG)
 - สร้าง Logical Volume (LV) ภายใน VG เพื่อใช้งานจัดเก็บข้อมูล
- การจัดการ LVM ช่วยให้สามารถเพิ่มหรือลดขนาดพื้นที่เก็บข้อมูลได้ตามต้องการ โดยไม่ต้องหยุดระบบ
- การใช้งาน LVM เมฆาสามารถรับเซิร์ฟเวอร์และระบบที่ต้องการความยืดหยุ่นสูงในการจัดการพื้นที่เก็บข้อมูล



```
ansible APP 11:32 AM
Playbook initiated (548b45)
nginx.yaml

Starting play (548b45)
Install and Start Nginx

Playbook Complete (548b45)

Failed!
+-----+-----+-----+-----+-----+
| Host | Ok | Changed | Unreachable | Failures | Rescued | Ignored |
+-----+-----+-----+-----+-----+-----+
| 10.0.0.10 | 0 | 0 | 0 | 1 | 0 | 0 |
| 192.168.56.108 | 3 | 0 | 0 | 0 | 0 | 0 |
+-----+-----+-----+-----+-----+
```

การใช้คำสั่งจัดการ LVM ผ่านเทอร์มินัล

- สร้าง Physical Volume (PV):

- เปิดเทอร์มินัลและรันคำสั่ง:

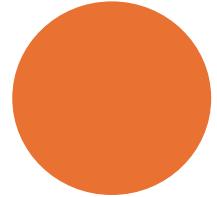
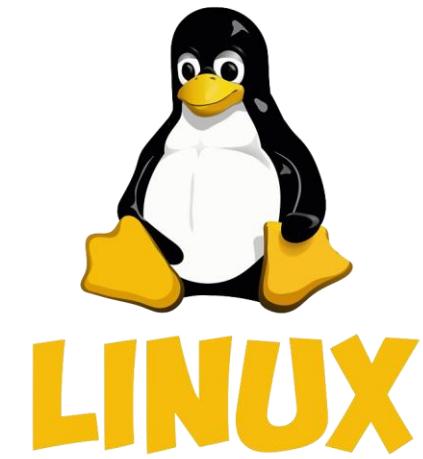
Bash

```
pvcreate /dev/sdb1 /dev/sdc1
```

- ตรวจสอบ PV ที่สร้างด้วย:

Bash

```
pvs
```



```
A ansible APP 11:32 AM
Playbook initiated (548b45)
nginx.yaml

Starting play (548b45)
Install and Start Nginx

Playbook Complete (548b45)

Failed!
+-----+-----+-----+-----+-----+
|     Host      | Ok | Changed | Unreachable | Failures | Rescued | Ignored |
+-----+-----+-----+-----+-----+
| 10.0.0.10   | 0  | 0     | 0           | 0         | 1       | 0       |
| 192.168.56.108 | 3  | 0     | 0           | 0         | 0       | 0       |
+-----+-----+-----+-----+-----+
```

การใช้คำสั่งจัดการ LVM ผ่านเทอร์มินัล

- สร้าง Volume Group (VG):

- รันคำสั่ง:

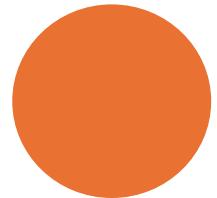
Bash

```
vgcreate myvg /dev/sdb1 /dev/sdc1
```

- ตรวจสอบ VG ด้วย:

Bash

```
vgs
```



```
A ansible APP 11:32 AM
Playbook initiated (548b45)
nginx.yaml

Starting play (548b45)
Install and Start Nginx

Playbook Complete (548b45)

Failed!
+-----+-----+-----+-----+-----+
| Host | Ok | Changed | Unreachable | Failures | Rescued | Ignored |
+-----+-----+-----+-----+-----+
| 10.0.0.10 | 0 | 0 | 0 | 1 | 0 | 0 |
| 192.168.56.108 | 3 | 0 | 0 | 0 | 0 | 0 |
+-----+-----+-----+-----+-----+
```

การใช้คำสั่งจัดการ LVM ผ่านเทอร์มินัล

- สร้าง Logical Volume (LV):

- รันคำสั่ง:

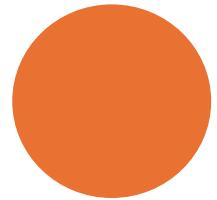
Bash

```
lvcreate -L 10G -n mylv myvg
```

- ตรวจสอบ LV ด้วย:

Bash

```
lvs
```



```
A ansible [APP] 11:32 AM
Playbook initiated (548b45)
nginx.yaml

Starting play (548b45)
Install and Start Nginx

Playbook Complete (548b45)

Failed!
+-----+-----+-----+-----+-----+
|     Host      | Ok | Changed | Unreachable | Failures | Rescued | Ignored |
+-----+-----+-----+-----+-----+
| 10.0.0.10   | 0  | 0     | 0           | 0         | 1       | 0       | 0       |
| 192.168.56.108 | 3  | 0     | 0           | 0         | 0       | 0       | 0       |
+-----+-----+-----+-----+-----+
```

การใช้คำสั่งจัดการ LVM ผ่านเทอร์มินัล

- เพิ่มขนาดของ Logical Volume:
 - ตัวอย่างคำสั่งขยาย Logical Volume:

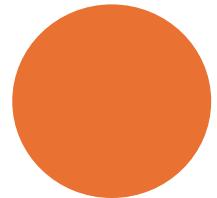
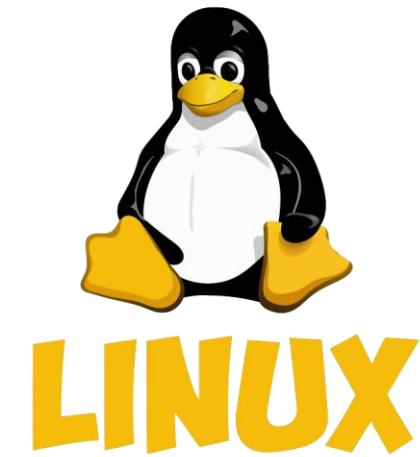
Bash

```
lvextend -L +5G /dev/myvg/mylv
```

- ลบ Logical Volume:
 - ตัวอย่างการลบ Logical Volume:

Bash

```
lvremove /dev/myvg/mylv
```



```
A ansible APP 11:32 AM
Playbook initiated (548b45)
nginx.yaml

Starting play (548b45)
Install and Start Nginx

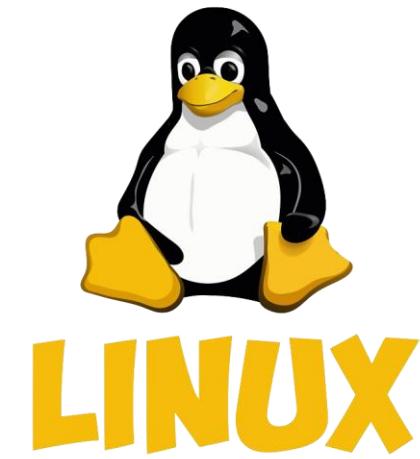
Playbook Complete (548b45)
Failed!

+-----+-----+-----+-----+-----+
| Host | Ok | Changed | Unreachable | Failures | Rescued | Ignored |
+-----+-----+-----+-----+-----+
| 10.0.0.10 | 0 | 0 | 0 | 1 | 0 | 0 |
| 192.168.56.108 | 3 | 0 | 0 | 0 | 0 | 0 |
+-----+-----+-----+-----+-----+
```

โครงสร้างของระบบไฟล์ (File System Structure)

ระบบไฟล์ใน Linux และ FHS

- Filesystem Hierarchy Standard (FHS): เป็นมาตรฐานที่กำหนดโครงสร้างของระบบไฟล์ใน Linux เพื่อให้มีการจัดเก็บไฟล์อย่างเป็นระเบียบและสะดวกต่อการใช้งานและบำรุงรักษา
- โครงสร้างนี้แบ่งได้เริ่กๆตามหน้าที่และการใช้งาน เช่น ไฟล์ระบบ ไฟล์ตั้งค่า และไฟล์ของผู้ใช้งาน
- ข้อดีของ FHS: ช่วยให้ผู้ใช้และผู้ดูแลระบบสามารถค้นหาไฟล์ต่าง ๆ ได้ง่ายยิ่งขึ้นและช่วยให้การพัฒนาระบบและซอฟต์แวร์กำลังได้รับความนิยมและเป็นระเบียบ



```
[john@server ~]$ ls -l /var/log
total 228424
drwx----- 2 root    root        4096 Jun 28 06:33 audit
-rw----- 1 root    root          0 Feb  7 03:46 boot.log
-rw----- 1 root    root      19720 Dec 15 2023 boot.log-20231215
-rw----- 1 root    root      10908 Feb  7 03:46 boot.log-20240207
-rw----- 1 root    utmp     81690624 Jun 28 18:01 btmp
-rw----- 1 root    utmp     66148992 Jun  1 03:05 btmp-20240601
drwxr-x--- 2 chrony  chrony     4096 Jun 24 2021 chrony
-rw----- 1 root    root      38098 Jun 28 18:01 cron
-rw----- 1 root    root      47835 Jun  2 03:09 cron-20240602
-rw----- 1 root    root      47664 Jun  9 03:01 cron-20240609
-rw----- 1 root    root      48080 Jun 16 03:01 cron-20240616
-rw----- 1 root    root      47861 Jun 23 03:01 cron-20240623
-rw-r--r-- 1 root    root     672911 Jun 28 16:27 dnf.librepo.log
-rw-r--r-- 1 root    root    1048552 Apr 19 21:26 dnf.librepo.log.1
-rw-r--r-- 1 root    root     599276 Jun 28 16:27 dnf.log
-rw-r--r-- 1 root    root    1048569 May 31 03:14 dnf.log.1
```

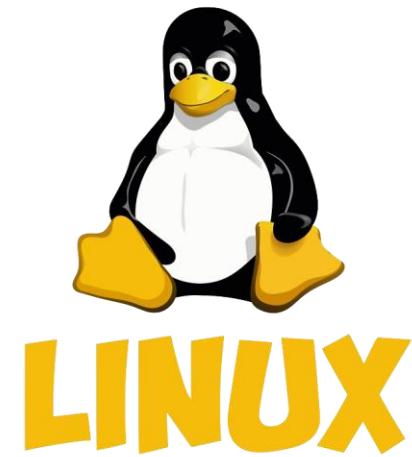
ໄດ້ເຮັດວຽກສໍາຄັນໃນໂຄງສ່າງຮະບບໄຟລ໌ຂອງ Linux

- / (Root Directory): ໄດ້ເຮັດວຽກຫລັກທີ່ເປັນຈຸດເຮີ່ມຕົ້ນຂອງຮະບບໄຟລ໌ກັ່ງໜົດໃນ Linux
- /bin: ເກີບຄຳສັ່ງພື້ນຈານທີ່ຈໍາເປັນສໍາຮັບຜູ້ໃຊ້ແລະຮະບບ ເຊັ່ນ ls, cp, mv
- /etc: ເກີບໄຟລ໌ການຕັ້ງຄ່າແລະໄຟລ໌ຄອນພິກາເຮັດວຽກຂອງຮະບບ ເຊັ່ນ passwd (ໄຟລ໌ຂ້ອມູລຜູ້ໃຊ້), fstab (ໄຟລ໌ຕັ້ງຄ່າຮະບບໄຟລ໌)
- /usr: ໃຊ້ສໍາຮັບເກີບຂອຝົວຝາກແວຣ໌ແລະໄລບຣາຣີທີ່ໄຟລ໌ໃຊ້ຮະບບພື້ນຈານ ເຊັ່ນ /usr/bin ເກີບຄຳສັ່ງທີ່ຜູ້ໃຊ້ກ່ວ່າໄປສາມາດໃຊ້ງານໄດ້
- /var: ເກີບໄຟລ໌ທີ່ມີການເປົ້າປັນແປງລົງບ່ວຍ ເຊັ່ນໄຟລ໌ log, cache, ແລະ spool



สำรวจไดเร็กทอรีด้วยคำสั่งพื้นฐาน

- คำสั่งสำรวจโครงสร้างระบบไฟล์:
 - `ls /`: แสดงไดเร็กทอรี跟หน้าใน root
 - `ls /bin`: แสดงคำสั่งและโปรแกรมในไดเร็กทอรี `/bin`
 - `ls /etc`: แสดงไฟล์คอนฟิกต่าง ๆ ใน `/etc`
 - `ls /usr/bin`: สำรวจคำสั่งเพิ่มเติมที่ติดตั้งในระบบ
- คำสั่งอื่น ๆ ที่เป็นประโยชน์:
 - `tree` (ถ้ามี): แสดงโครงสร้างไดเร็กทอรีเป็นรูปแบบต้นไม้
 - `du -h /var`: ตรวจสอบการใช้พื้นที่ใน `/var`



```
[john@server ~]$ ls -l /var/log
total 228424
drwx----- 2 root    root        4096 Jun 28 06:33 audit
-rw----- 1 root    root         0 Feb  7 03:46 boot.log
-rw----- 1 root    root     19720 Dec 15 2023 boot.log-20231215
-rw----- 1 root    root     10908 Feb  7 03:46 boot.log-20240207
-rw----- 1 root    utmp   81690624 Jun 28 18:01 btmp
-rw----- 1 root    utmp   66148992 Jun  1 03:05 btmp-20240601
drwxr-x--- 2 chrony  chrony     4096 Jun 24 2021 chrony
-rw----- 1 root    root     38098 Jun 28 18:01 cron
-rw----- 1 root    root     47835 Jun  2 03:09 cron-20240602
-rw----- 1 root    root     47664 Jun  9 03:01 cron-20240609
-rw----- 1 root    root     48080 Jun 16 03:01 cron-20240616
-rw----- 1 root    root     47861 Jun 23 03:01 cron-20240623
-rw-r--r-- 1 root    root   672911 Jun 28 16:27 dnf.librepo.log
-rw-r--r-- 1 root    root  1048552 Apr 19 21:26 dnf.librepo.log.1
-rw-r--r-- 1 root    root   599276 Jun 28 16:27 dnf.log
-rw-r--r-- 1 root    root  1048569 May 31 03:14 dnf.log.1
```

hostatom.com

กิจกรรม: สำรวจโครงสร้างไฟล์ระบบผ่านเทอร์มินัล

- สำรวจโครงสร้างหลักของระบบไฟล์ :
 - เปิดเทอร์มินัลและรันคำสั่ง:

Bash

`ls /`

- สำรวจไดเร็กทอรีเฉพาะเพื่อกำความเข้าใจการใช้งาน :
 - คำสั่ง `ls` กับไดเร็กทอรีต่าง ๆ เช่น `/bin`, `/etc`, `/var`, และ `/usr`

Bash

`ls /bin`

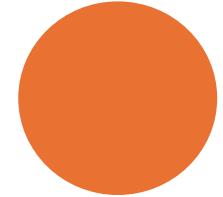
`ls /etc`

`ls /var`

`ls /usr`



LINUX



```
[john@server ~]$ ls -l /var/log
total 228424
drwx----- 2 root    root        4096 Jun 28 06:33 audit
-rw----- 1 root    root         0 Feb  7 03:46 boot.log
-rw----- 1 root    root     19720 Dec 15 2023 boot.log-20231215
-rw----- 1 root    root     10908 Feb  7 03:46 boot.log-20240207
-rw----- 1 root    utmp    81690624 Jun 28 18:01 btmp
-rw----- 1 root    utmp    66148992 Jun  1 03:05 btmp-20240601
drwxr-x--- 2 chrony  chrony     4096 Jun 24 2021 chrony
-rw----- 1 root    root     38098 Jun 28 18:01 cron
-rw----- 1 root    root     47835 Jun  2 03:09 cron-20240602
-rw----- 1 root    root     47664 Jun  9 03:01 cron-20240609
-rw----- 1 root    root     48080 Jun 16 03:01 cron-20240616
-rw----- 1 root    root     47861 Jun 23 03:01 cron-20240623
-rw-r--r-- 1 root    root    672911 Jun 28 16:27 dnf.librepo.log
-rw-r--r-- 1 root    root   1048552 Apr 19 21:26 dnf.librepo.log.1
-rw-r--r-- 1 root    root    599276 Jun 28 16:27 dnf.log
-rw-r--r-- 1 root    root   1048569 May 31 03:14 dnf.log.1          hostatom.com
```

กิจกรรม: สำรวจโครงสร้างไฟล์ระบบผ่านเทอร์มินัล

- ใช้คำสั่ง tree เพื่อดูโครงสร้างไฟล์แบบแผนผัง (ถ้ามี) :
 - หากมีการติดตั้งคำสั่ง tree แล้ว ให้รันคำสั่งนี้เพื่อดูโครงสร้างทั้งหมด:

Bash

```
tree /
```

- กิจกรรมเพิ่มเติม: สังเกตไฟล์ log ใน /var/log :
 - เปิดดูไฟล์ log เช่น syslog หรือ messages (ถ้ามี) ด้วยคำสั่ง cat หรือ less

Bash

```
cat /var/log/syslog
```



LINUX

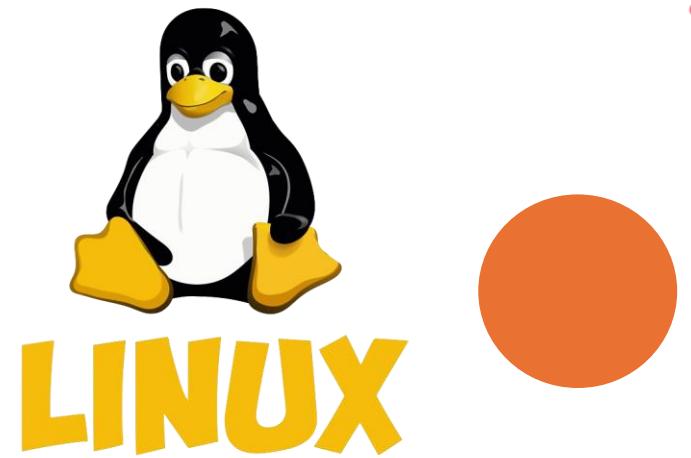
```
[john@server ~]$ ls -l /var/log
total 228424
drwx----- 2 root    root        4096 Jun 28 06:33 audit
-rw----- 1 root    root         0 Feb   7 03:46 boot.log
-rw----- 1 root    root     19720 Dec 15 2023 boot.log-20231215
-rw----- 1 root    root     10908 Feb   7 03:46 boot.log-20240207
-rw----- 1 root    utmp    81690624 Jun 28 18:01 btmp
-rw----- 1 root    utmp    66148992 Jun  1 03:05 btmp-20240601
drwxr-x--- 2 chrony  chrony     4096 Jun 24 2021 chrony
-rw----- 1 root    root     38098 Jun 28 18:01 cron
-rw----- 1 root    root     47835 Jun  2 03:09 cron-20240602
-rw----- 1 root    root     47664 Jun  9 03:01 cron-20240609
-rw----- 1 root    root     48080 Jun 16 03:01 cron-20240616
-rw----- 1 root    root     47861 Jun 23 03:01 cron-20240623
-rw-r--r-- 1 root    root    672911 Jun 28 16:27 dnf.librepo.log
-rw-r--r-- 1 root    root   1048552 Apr 19 21:26 dnf.librepo.log.1
-rw-r--r-- 1 root    root    599276 Jun 28 16:27 dnf.log
-rw-r--r-- 1 root    root   1048569 May 31 03:14 dnf.log.1
```

hostatom.com

พื้นฐานการตั้งค่าเครือข่าย TCP/IP

พื้นฐานของ TCP/IP และโครงสร้างเครือข่าย

- TCP/IP (Transmission Control Protocol/Internet Protocol) คือชุดโปรโตคอลพื้นฐานสำหรับการสื่อสารในเครือข่าย โดยใช้ IP Address เป็นตัวระบุที่อยู่ของอุปกรณ์ในเครือข่าย
- โครงสร้าง TCP/IP: แบ่งเป็นชั้นต่าง ๆ เช่น
 - ชั้น Network (IP) ที่ใช้ในการระบุที่อยู่และการกำหนดเส้นทางข้อมูล
 - ชั้น Transport (TCP) ที่ใช้ในการตรวจสอบและควบคุมการส่งข้อมูลอย่างมีประสิทธิภาพ
- IP Address: หมายเลขประจำอุปกรณ์ในเครือข่ายที่ทำหน้าที่เป็นที่อยู่ส่งข้อมูล
- Subnet Mask: ช่วยระบุขอบเขตของเครือข่ายย่อย (Subnet) ที่อุปกรณ์สามารถเชื่อมต่อได้โดยตรง
- Gateway: เป็นเส้นทางที่ข้อมูลออกจากเครือข่ายไปยังเครือข่ายอื่น



```
how to check ip address in Linux

[root@howtouselinux ~]# ifconfig -a
ens192: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.252.9.61  netmask 255.255.254.0  broadcast 10.252.9.255
          ether 00:50:56:81:68:27  txqueuelen 1000  (Ethernet)
            RX packets 32625850  bytes 116785252243 (108.7 GiB)
            RX errors 0  dropped 0  overruns 0  frame 0
            TX packets 27998185  bytes 30777718929 (28.6 GiB)
            TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
          loop  txqueuelen 1000  (Local Loopback)
            RX packets 24014984  bytes 8058758450 (7.5 GiB)
            RX errors 0  dropped 0  overruns 0  frame 0
            TX packets 24014984  bytes 8058758450 (7.5 GiB)
            TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
```

การตั้งค่าเครือข่ายใน Linux

- การตั้งค่า IP Address, Subnet Mask, และ Gateway ทำให้ระบบสามารถสื่อสารในเครือข่ายได้ :
- การตั้งค่าแบบชั่วคราว โดยใช้คำสั่ง ifconfig หรือ ip :
 - ตั้งค่า IP Address และ Subnet Mask:

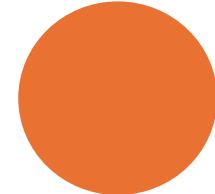
Bash

```
sudo ifconfig eth0 192.168.1.100 netmask  
255.255.255.0
```

- หรือ

Bash

```
sudo ip addr add 192.168.1.100/24 dev eth0
```



```
how to check ip address in Linux  
  
[root@howtouselinux ~]# ifconfig -a  
ens192: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
inet 10.252.9.61 netmask 255.255.254.0 broadcast 10.252.9.255  
ether 00:50:56:81:68:27 txqueuelen 1000 (Ethernet)  
RX packets 32625850 bytes 116785252243 (108.7 GiB)  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 27998185 bytes 30777718929 (28.6 GiB)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
inet 127.0.0.1 netmask 255.0.0.0  
Loop txqueuelen 1000 (Local Loopback)  
RX packets 24014984 bytes 8058758450 (7.5 GiB)  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 24014984 bytes 8058758450 (7.5 GiB)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

การตั้งค่าเครือข่ายใน Linux

- ตั้งค่า Gateway :

Bash

```
sudo route add default gw 192.168.1.1
```

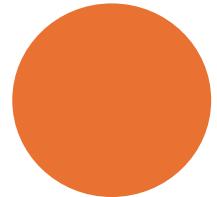
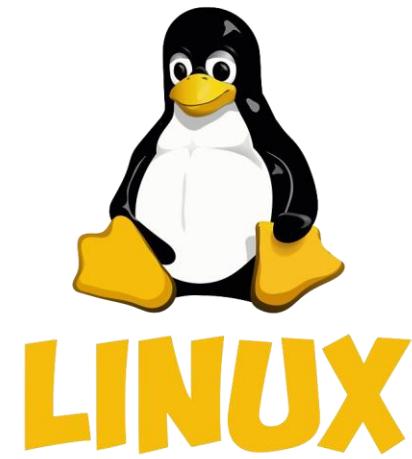
- หรือ

Bash

```
sudo ip route add default via  
192.168.1.1
```

- การตั้งค่าแบบดาวร แก้ไขในไฟล์คอนฟิก :

- แก้ไขไฟล์ `/etc/network/interfaces` หรือ `/etc/netplan` (ขึ้นอยู่กับเดสโกรที่ใช้) เพื่อให้ IP Address, Subnet Mask, และ Gateway คงอยู่หลังจากการบูตระบบใหม่



```
how to check ip address in Linux

[root@howtouselinux ~]# ifconfig -a
ens192: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 10.252.9.61 netmask 255.255.254.0 broadcast 10.252.9.255
          ether 00:50:56:81:68:27 txqueuelen 1000 (Ethernet)
        RX packets 32625850 bytes 116785252243 (108.7 GiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 27998185 bytes 30777718929 (28.6 GiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.1 netmask 255.0.0.0
        Loop txqueuelen 1000 (Local Loopback)
        RX packets 24014984 bytes 8058758450 (7.5 GiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 24014984 bytes 8058758450 (7.5 GiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

เครื่องมือการตรวจสอบเครือข่าย

- เครื่องมือสำคัญในการตรวจสอบเครือข่าย :

- ping: ตรวจสอบการเชื่อมต่อระหว่างเครื่อง โดยส่งแพ็คเก็ตเพื่อดูว่า ปลายทางตอบสนองหรือไม่

Bash

```
ping google.com
```

- ifconfig: ใช้ตรวจสอบและตั้งค่าข้อมูลเครือข่าย เช่น IP Address ของอินเทอร์เฟซเครือข่าย

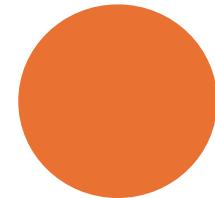
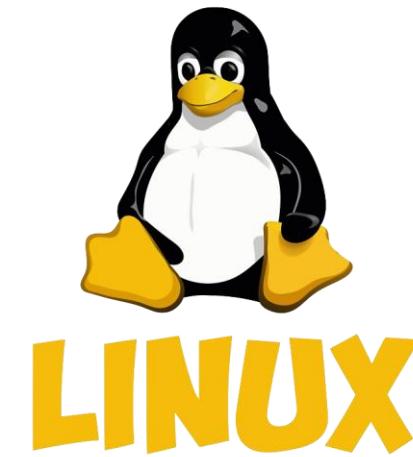
Bash

```
ifconfig
```

- ip: ใช้ตรวจสอบและจัดการเครือข่ายได้เหมือน ifconfig แต่ให้รายละเอียดมากกว่า :

Bash

```
ip addr show
```



```
[john@server ~]$ ls -l /var/log
total 228424
drwx----- 2 root    root        4096 Jun 28 06:33 audit
-rw----- 1 root    root         0 Feb  7 03:46 boot.log
-rw----- 1 root    root      19720 Dec 15 2023 boot.log-20231215
-rw----- 1 root    root      10908 Feb  7 03:46 boot.log-20240207
-rw----- 1 root    utmp     81690624 Jun 28 18:01 btmp
-rw----- 1 root    utmp     66148992 Jun  1 03:05 btmp-20240601
drwxr-x--- 2 chrony  chrony     4096 Jun 24 2021 chrony
-rw----- 1 root    root      38098 Jun 28 18:01 cron
-rw----- 1 root    root      47835 Jun  2 03:09 cron-20240602
-rw----- 1 root    root      47664 Jun  9 03:01 cron-20240609
-rw----- 1 root    root      48080 Jun 16 03:01 cron-20240616
-rw----- 1 root    root      47861 Jun 23 03:01 cron-20240623
-rw-r--r-- 1 root    root      672911 Jun 28 16:27 dnf.librepo.log
-rw-r--r-- 1 root    root     1048552 Apr 19 21:26 dnf.librepo.log.1
-rw-r--r-- 1 root    root      599276 Jun 28 16:27 dnf.log
-rw-r--r-- 1 root    root     1048569 May 31 03:14 dnf.log.1
```

hostatom.com

เครื่องมือการตรวจสอบเครือข่าย

- netstat หรือ ss: ตรวจสอบการเชื่อมต่อ TCP, การฟังของพอร์ต และการใช้งานเครือข่ายอื่น ๆ

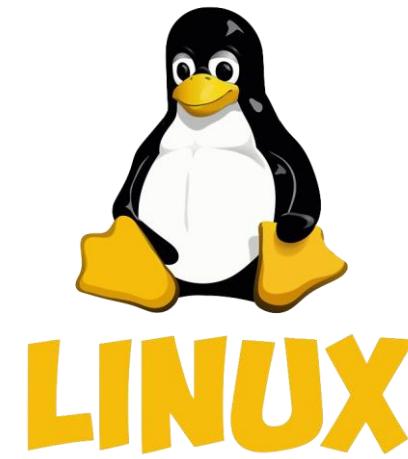
Bash

```
netstat -tuln
```

- traceroute: ตรวจสอบเส้นทางการส่งข้อมูลไปยังปลายทางแต่ละขั้นตอน

Bash

```
traceroute google.com
```



```
[john@server ~]$ ls -l /var/log
total 228424
drwx----- 2 root    root        4096 Jun 28 06:33 audit
-rw----- 1 root    root          0 Feb   7 03:46 boot.log
-rw----- 1 root    root       19720 Dec 15 2023 boot.log-20231215
-rw----- 1 root    root      10908 Feb   7 03:46 boot.log-20240207
-rw----- 1 root    utmp     81690624 Jun 28 18:01 btmp
-rw----- 1 root    utmp     66148992 Jun   1 03:05 btmp-20240601
drwxr-x--- 2 chrony  chrony     4096 Jun 24 2021 chrony
-rw----- 1 root    root      38098 Jun 28 18:01 cron
-rw----- 1 root    root      47835 Jun   2 03:09 cron-20240602
-rw----- 1 root    root      47664 Jun   9 03:01 cron-20240609
-rw----- 1 root    root      48080 Jun 16 03:01 cron-20240616
-rw----- 1 root    root      47861 Jun 23 03:01 cron-20240623
-rw-r--r-- 1 root    root     672911 Jun 28 16:27 dnf.librepo.log
-rw-r--r-- 1 root    root    1048552 Apr 19 21:26 dnf.librepo.log.1
-rw-r--r-- 1 root    root     599276 Jun 28 16:27 dnf.log
-rw-r--r-- 1 root    root    1048569 May 31 03:14 dnf.log.1
```

hostatom.com

การตั้งค่าเครือข่ายในระบบปฏิบัติการ Linux

- ตรวจสอบสถานะการเชื่อมต่อและ IP Address ปัจจุบัน :

- เปิดเทอร์มินัลและรันคำสั่ง :

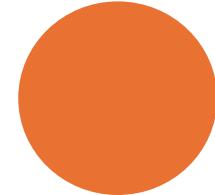
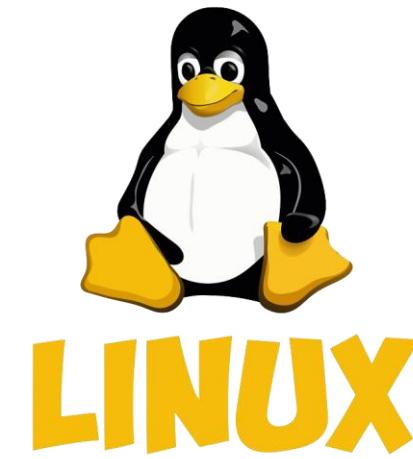
Bash

ifconfig

- หรือ

Bash

ip addr show



```
[john@server ~]$ ls -l /var/log
total 228424
drwx----- 2 root    root        4096 Jun 28 06:33 audit
-rw----- 1 root    root          0 Feb  7 03:46 boot.log
-rw----- 1 root    root      19720 Dec 15 2023 boot.log-20231215
-rw----- 1 root    root      10908 Feb  7 03:46 boot.log-20240207
-rw----- 1 root    utmp     81690624 Jun 28 18:01 btmp
-rw----- 1 root    utmp     66148992 Jun  1 03:05 btmp-20240601
drwxr-x--- 2 chrony  chrony      4096 Jun 24 2021 chrony
-rw----- 1 root    root      38098 Jun 28 18:01 cron
-rw----- 1 root    root      47835 Jun  2 03:09 cron-20240602
-rw----- 1 root    root      47664 Jun  9 03:01 cron-20240609
-rw----- 1 root    root      48080 Jun 16 03:01 cron-20240616
-rw----- 1 root    root      47861 Jun 23 03:01 cron-20240623
-rw-r--r-- 1 root    root     672911 Jun 28 16:27 dnf.librepo.log
-rw-r--r-- 1 root    root    1048552 Apr 19 21:26 dnf.librepo.log.1
-rw-r--r-- 1 root    root     599276 Jun 28 16:27 dnf.log
-rw-r--r-- 1 root    root    1048569 May 31 03:14 dnf.log.1      hostatom.com
```

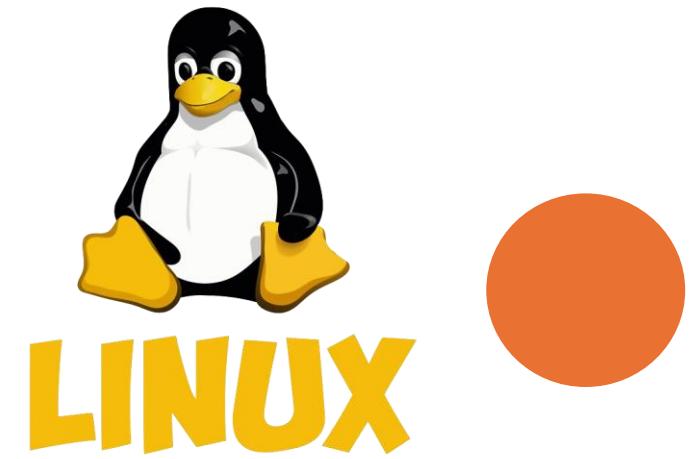
การตั้งค่าเครือข่ายในระบบปฏิบัติการ Linux

- รันคำสั่ง :

- เปิดเทอร์มินัลและรันคำสั่ง :

Bash

```
sudo ifconfig eth0 192.168.1.100 netmask  
255.255.255.0
```



- ตั้งค่า Gateway :

Bash

```
sudo route add default gw 192.168.1.1
```

```
[john@server ~]$ ls -l /var/log  
total 228424  
drwx----- 2 root    root        4096 Jun 28 06:33 audit  
-rw----- 1 root    root         0 Feb  7 03:46 boot.log  
-rw----- 1 root    root      19720 Dec 15 2023 boot.log-20231215  
-rw----- 1 root    root      10908 Feb  7 03:46 boot.log-20240207  
-rw----- 1 root    utmp     81690624 Jun 28 18:01 btmp  
-rw----- 1 root    utmp     66148992 Jun  1 03:05 btmp-20240601  
drwxr-x--- 2 chrony  chrony     4096 Jun 24 2021 chrony  
-rw----- 1 root    root      38098 Jun 28 18:01 cron  
-rw----- 1 root    root      47835 Jun  2 03:09 cron-20240602  
-rw----- 1 root    root      47664 Jun  9 03:01 cron-20240609  
-rw----- 1 root    root      48080 Jun 16 03:01 cron-20240616  
-rw----- 1 root    root      47861 Jun 23 03:01 cron-20240623  
-rw-r--r-- 1 root    root     672911 Jun 28 16:27 dnf.librepo.log  
-rw-r--r-- 1 root    root    1048552 Apr 19 21:26 dnf.librepo.log.1  
-rw-r--r-- 1 root    root     599276 Jun 28 16:27 dnf.log  
-rw-r--r-- 1 root    root    1048569 May 31 03:14 dnf.log.1          hostatom.com
```

การตั้งค่าเครือข่ายในระบบปฏิบัติการ Linux

- ทดสอบการเชื่อมต่อเครือข่าย :
 - ใช้คำสั่ง ping เพื่อตรวจสอบการเชื่อมต่อ :

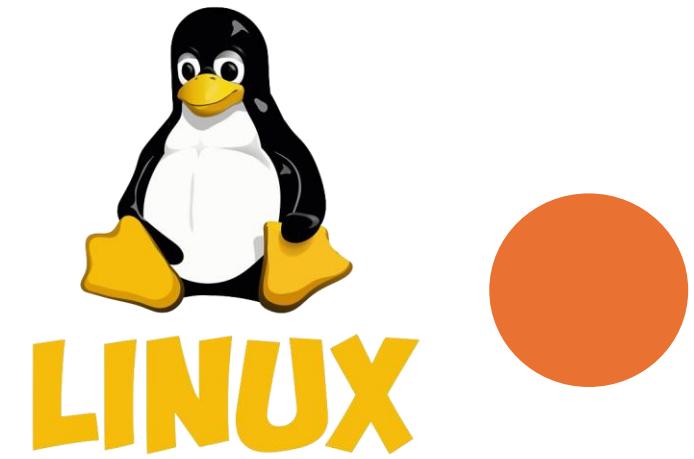
Bash

```
ping 8.8.8.8
```

- สำรวจและตรวจสอบพอร์ตการเชื่อมต่อที่กำลังใช้งาน:
 - รันคำสั่ง :

Bash

```
netstat -tuln
```

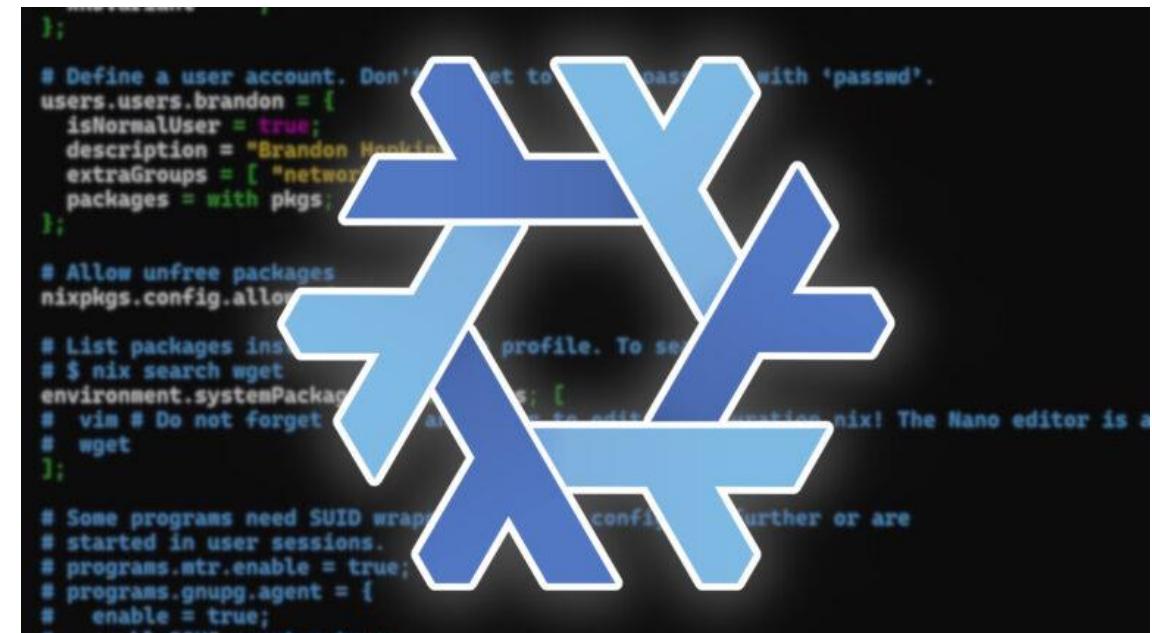
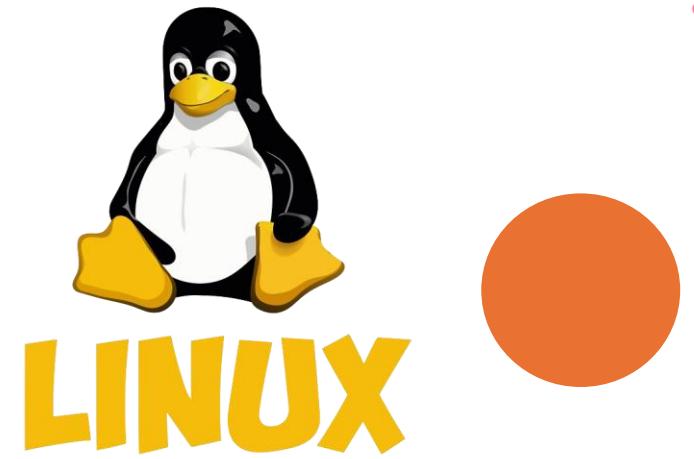


```
[john@server ~]$ ls -l /var/log
total 228424
drwx----- 2 root    root        4096 Jun 28 06:33 audit
-rw----- 1 root    root         0 Feb  7 03:46 boot.log
-rw----- 1 root    root      19720 Dec 15 2023 boot.log-20231215
-rw----- 1 root    root      10908 Feb  7 03:46 boot.log-20240207
-rw----- 1 root    utmp     81690624 Jun 28 18:01 btmp
-rw----- 1 root    utmp     66148992 Jun  1 03:05 btmp-20240601
drwxr-x--- 2 chrony  chrony      4096 Jun 24 2021 chrony
-rw----- 1 root    root      38098 Jun 28 18:01 cron
-rw----- 1 root    root      47835 Jun  2 03:09 cron-20240602
-rw----- 1 root    root      47664 Jun  9 03:01 cron-20240609
-rw----- 1 root    root      48080 Jun 16 03:01 cron-20240616
-rw----- 1 root    root      47861 Jun 23 03:01 cron-20240623
-rw-r--r-- 1 root    root     672911 Jun 28 16:27 dnf.librepo.log
-rw-r--r-- 1 root    root    1048552 Apr 19 21:26 dnf.librepo.log.1
-rw-r--r-- 1 root    root     599276 Jun 28 16:27 dnf.log
-rw-r--r-- 1 root    root    1048569 May 31 03:14 dnf.log.1          hostatom.com
```

เตรียมข้อมูลสำหรับ การติดตั้ง Linux ครั้งแรก

ขั้นตอนการเตรียมฮาร์ดแวร์และซอฟต์แวร์

- ความต้องการของระบบขั้นต่ำ: แรม, พื้นที่ฮาร์ดดิสก์, ชิปปี้ที่รองรับการติดตั้ง
 - ตัวอย่าง: สำหรับ Linux Distribution กว่าไป:
 - RAM: อย่างน้อย 1GB (แนะนำ 2GB ขึ้นไป)
 - พื้นที่ดิสก์: อย่างน้อย 20GB
 - ชิปปี้: x86_64 หรือ ARM ที่รองรับ
- ประเภทการติดตั้งที่ต้องการ: Server, Desktop, Minimal Installation
- การเตรียมไฟล์ติดตั้ง Linux (ISO):
 - ดาวน์โหลดไฟล์ ISO ของ Linux Distribution ที่ต้องการ เช่น Ubuntu Server, CentOS, หรือ Debian
 - ตรวจสอบความสมบูรณ์ของไฟล์ ISO ด้วย checksum เช่น SHA-256
- อุปกรณ์สำหรับการติดตั้ง: USB Flash Drive หรือ DVD



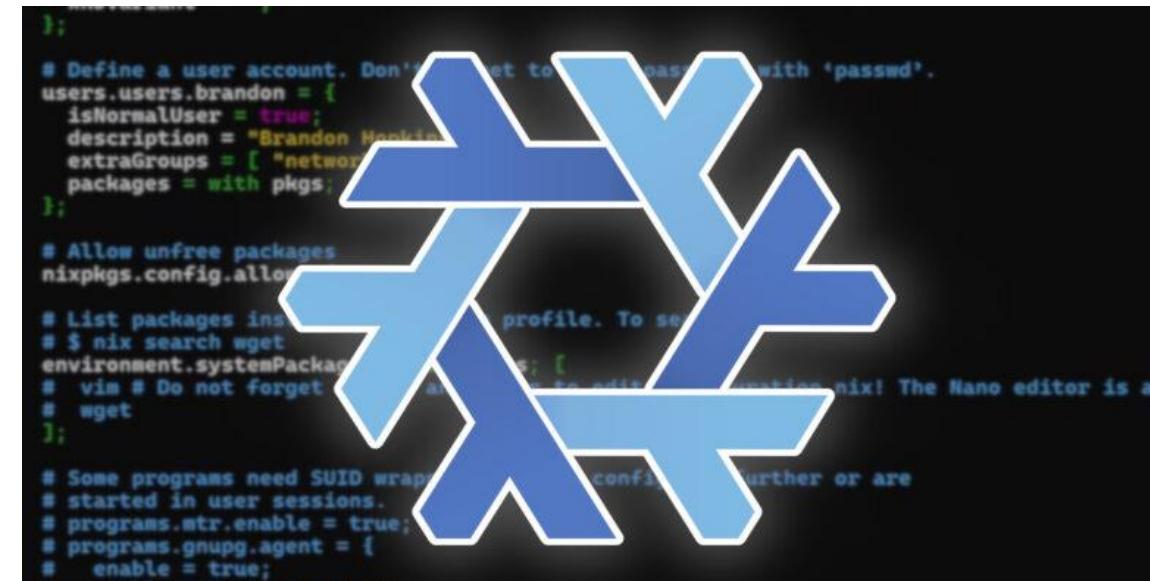
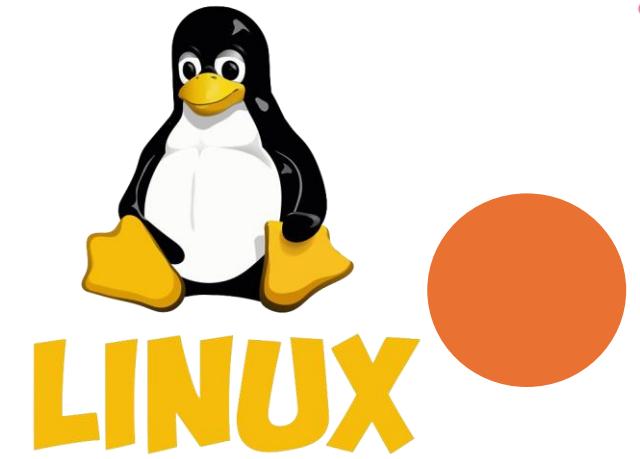
การเตรียมข้อมูลพาร์ติชันและการตั้งค่าเครือข่าย

- **การวางแผนพาร์ติชัน:** เพื่อการใช้งานระบบที่เสถียรและจัดการได้ง่าย
 - ตัวอย่างโครงสร้างพาร์ติชัน:
 - / (root): พื้นที่สำหรับระบบหลัก ประมาณ 15-20GB
 - /home: พื้นที่สำหรับข้อมูลของผู้ใช้ (ถ้ามี) ขึ้นอยู่กับขนาดที่คาดว่าจะใช้
 - /swap: ขึ้นอยู่กับขนาด RAM (2GB ขึ้นไป)
 - /var (ถ้าต้องการ): สำหรับไฟล์ log ข้อมูลระบบ (ถ้าใช้ในระบบ server)
 - **การตั้งค่าเครือข่าย:** เตรียมข้อมูลสำหรับตั้งค่า IP Address, Subnet Mask, Gateway, และ DNS Server
 - หากต้องการใช้ DHCP ให้ยืนยันกับผู้ดูแลเครือข่ายว่าสามารถเชื่อมต่อได้
 - หากต้องการใช้ IP แบบกำหนดเอง (Static IP) ควรเตรียมข้อมูล IP Address, Subnet Mask, Gateway และ DNS Server



เช็กลิสต์การเตรียมข้อมูล

- ข้อมูลฮาร์ดแวร์และซอฟต์แวร์ที่ต้องตรวจสอบ:
 - ตรวจสอบแรม, ซีพียู, ขนาดดิสก์, และการเชื่อมต่ออินเทอร์เน็ต (ถ้าต้องการติดตั้งเพิ่มเติมจากอินเทอร์เน็ต)
- รายการเช็กลิสต์:
 - ไฟล์ ISO ที่ถูกต้อง (ตรวจสอบ SHA-256)
 - พาร์ติชันที่วางแผนไว้ เช่น /, /home, /swap
 - ข้อมูลการตั้งค่าเครือข่าย เช่น IP Address หรือเปิดใช้งาน DHCP
 - สำรองข้อมูลสำคัญในดิสก์ก่อนเริ่มการติดตั้ง (หากติดตั้งในเครื่องที่มีข้อมูลอยู่)



ติดตั้ง Linux ครึ่งแรก (Step-by-Step)

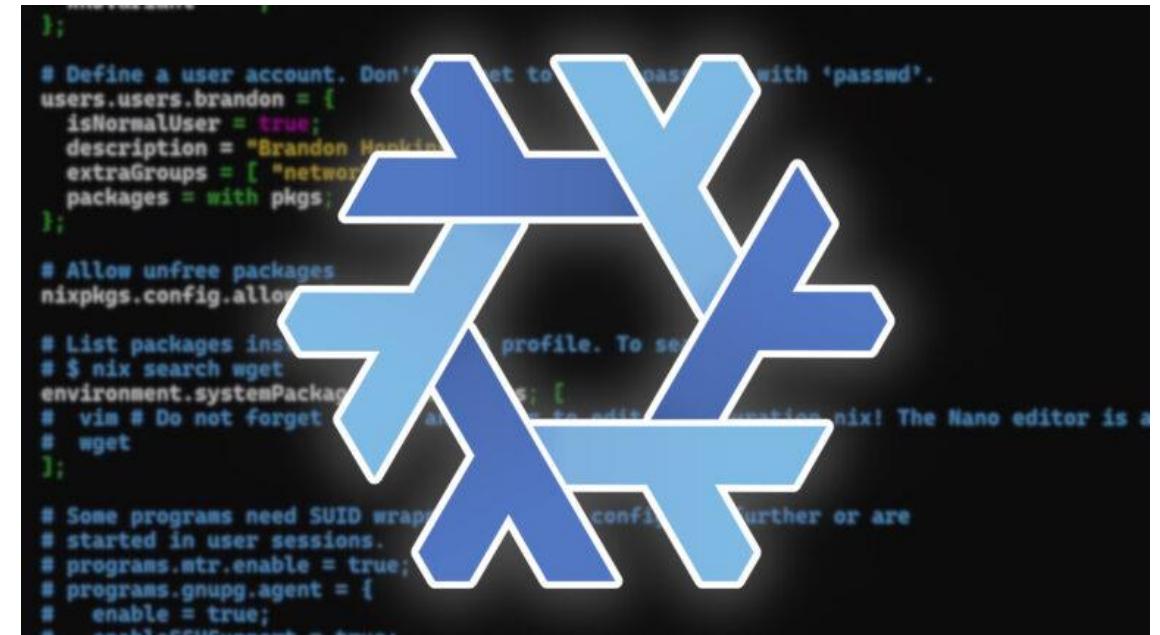
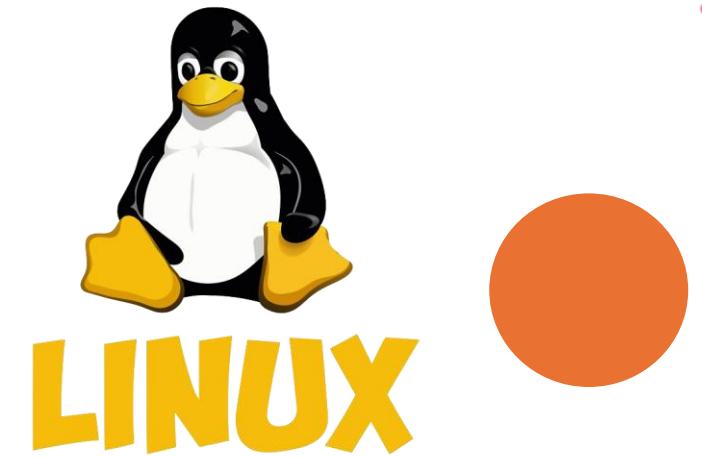
การบูตจากไฟล์ ISO เพื่อเริ่มการติดตั้ง

- ขั้นตอนที่ 1: ใส่ ISO Image
 - ติดตั้งบนเครื่องจริง ให้ใส่ USB หรือ DVD ที่มีไฟล์ ISO
 - ติดตั้งบน VirtualBox:
 - เข้าไปที่ Settings > Storage > เลือกไดร์ฟ Empty > เลือกไฟล์ ISO
- ขั้นตอนที่ 2: ตั้งค่าให้เครื่องบูตจาก USB หรือ DVD หรือ ISO
 - ตั้งค่า BIOS ให้บูตจาก USB หรือ DVD
 - VirtualBox ให้ตรวจสอบว่า ISO ถูกเลือกในตัวเลือก Storage
- ขั้นตอนที่ 3: บูตเข้าสู่การติดตั้ง Linux



การเริ่มติดตั้งจากหน้าจอต่อไปนี้รับของ Linux

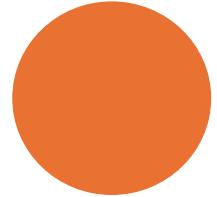
- เลือกภาษาและภูมิภาค: เลือกภาษาและโซนเวลาที่ต้องการ
- เลือกโหมดการติดตั้ง: โดยปกติจะเลือกโหมด Install Linux (ขึ้นอยู่กับดีสโตร อาจมีตัวเลือกให้กดลงในชั้นระบบ)
- ตั้งค่าแป้นพิมพ์: เลือกภาษาของแป้นพิมพ์ที่เหมาะสมกับผู้ใช้
- เลือกประเภทการติดตั้ง: Minimal Install (การติดตั้งแบบพื้นฐาน) หรือ Full Install (การติดตั้งแบบครบชุด)



การจัดการพาร์ติชันและการตั้งค่าดิสก์

- ขั้นตอนการแบ่งพาร์ติชัน:

- Automatic Partitioning: ให้ระบบจัดการพาร์ติชันให้โดยอัตโนมัติ (แนะนำสำหรับผู้เริ่มต้น)
- Manual Partitioning: แบ่งพาร์ติชันเอง (แนะนำสำหรับผู้ที่ต้องการควบคุมขนาดพาร์ติชัน)
- ตัวอย่างโครงสร้างพาร์ติชันแนะนำ:
 - / (root): 15-20GB
 - /home: สำหรับข้อมูลของผู้ใช้
 - /swap: สำหรับการจัดการหน่วยความจำเสมือน
- การตั้งค่าพื้นฐานสำหรับเครือข่าย:
 - เลือก DHCP หรือตั้งค่า IP แบบ Static หากเป็นเซิร์ฟเวอร์



การติดตั้งแพ็คเกจพื้นฐานและการสร้างบัญชีผู้ใช้

- ขั้นตอนการติดตั้งแพ็คเกจพื้นฐาน:
 - เลือกแพ็คเกจที่จำเป็นสำหรับเซิร์ฟเวอร์หรือเดสก์ท็อป เช่น เครื่องมือพื้นฐาน, Web Server, หรือเครื่องมือจัดการเครือข่าย
 - การตั้งค่าบัญชีผู้ใช้:
 - สร้างผู้ใช้หลัก และตั้งรหัสผ่าน
 - ตั้งรหัสผ่านสำหรับผู้ดูแลระบบ (root)
 - เริ่มการติดตั้ง: กดยืนยันและเริ่มขั้นตอนการติดตั้ง ตรวจสอบว่าการติดตั้งเสร็จสมบูรณ์



LINUX



การตั้งค่าพื้นฐานหลังการติดตั้ง

- ตรวจสอบการติดตั้ง: ตรวจสอบว่าระบบบูตเข้าสู่ Linux โดยสมบูรณ์
- อัปเดตระบบ: อัปเดตแพ็กเกจทั้งหมดให้เป็นเวอร์ชันล่าสุด

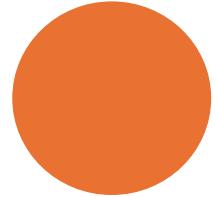
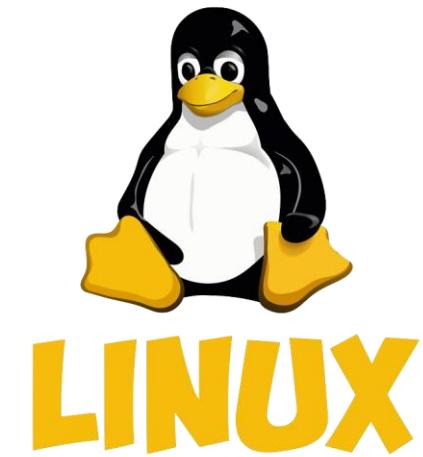
Bash

```
sudo apt update && sudo apt upgrade -y
```

- ตั้งค่าเครือข่ายเพิ่มเติม: หากยังไม่ได้ตั้งค่า Static IP, DNS, หรือ Gateway, ทำได้ในขั้นตอนนี้
- ติดตั้งเครื่องมือพื้นฐาน (ถ้าจำเป็น) :

Bash

```
sudo apt install net-tools curl wget
```



ตรวจสอบและตั้งค่าหลังการติดตั้ง (Post-installation Check)

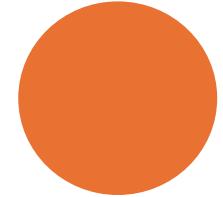
การตรวจสอบการติดตั้งเบื้องต้น

- การตรวจสอบการเชื่อมต่อเครือข่าย:

- ตรวจสอบสถานะของการเชื่อมต่อเครือข่ายด้วยคำสั่ง ping และ ifconfig หรือ ip addr

Bash

```
ping -c 4 google.com
```



- ตรวจสอบการตั้งค่า IP Address, Gateway, และ DNS Server เพื่อให้แน่ใจว่า เครือข่ายสามารถเชื่อมต่อได้อย่างสมบูรณ์

- การตรวจสอบการทำงานเดิสก์:

- ตรวจสอบการติดตั้งพาร์ติชันและขนาดพื้นที่เดิสก์ที่ใช้ด้วยคำสั่ง df -h
- ยืนยันว่าพาร์ติชันที่ตั้งค่าไว้ในขั้นตอนการติดตั้งปราฏและใช้งานได้ตามปกติ

Bash

```
df -h
```



การสร้างผู้ใช้และจัดการสิทธิ์

- การสร้างผู้ใช้ใหม่: สร้างผู้ใช้ใหม่สำหรับการใช้งานก่อไป เพื่อหลีกเลี่ยงการใช้ root ตลอดเวลา

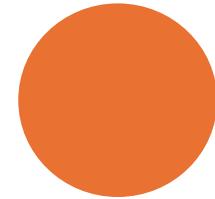
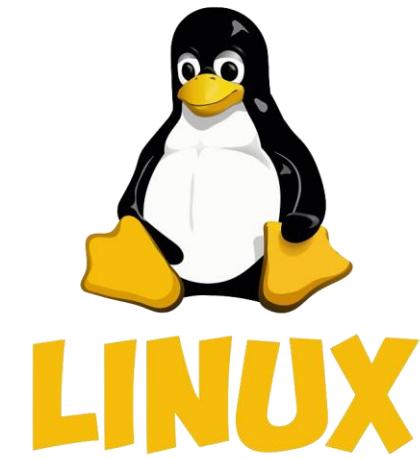
Bash

```
sudo adduser newuser  
sudo passwd newuser
```

- การมอบสิทธิ์ให้ผู้ใช้เป็น sudo:
 - เพิ่มผู้ใช้ไปที่กลุ่ม sudo เพื่อให้สามารถใช้คำสั่งที่ต้องใช้สิทธิ์ระดับผู้ดูแลระบบได้

Bash

```
sudo usermod -aG sudo newuser
```



การติดตั้งแพ็คเกจที่จำเป็นหลังการติดตั้ง

- การอัปเดตระบบ: อัปเดตแพ็คเกจทั้งหมดให้เป็นเวอร์ชันล่าสุด

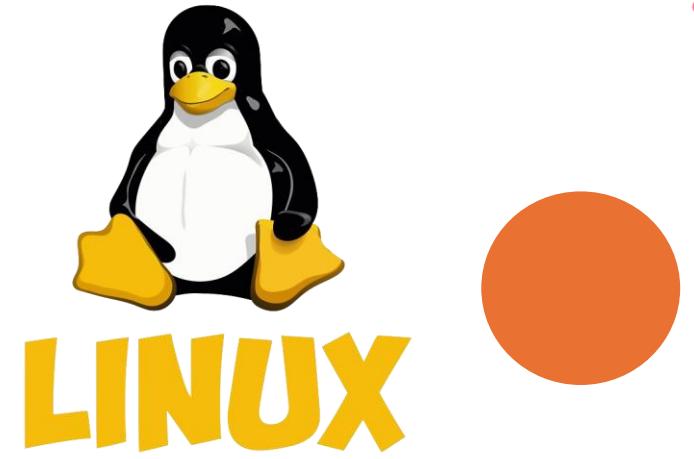
Bash

```
sudo apt update && sudo apt upgrade -y
```

- การติดตั้งเครื่องมือพื้นฐาน: ติดตั้งเครื่องมือที่จำเป็นสำหรับการจัดการระบบ
 - เช่น net-tools, curl, wget, และเครื่องมืออื่น ๆ ที่อาจจำเป็นสำหรับการจัดการระบบ

Bash

```
sudo apt install net-tools curl wget
```



ตรวจสอบสถานะบริการและตั้งค่าไฟร์วอลล์พื้นฐาน

- ตรวจสอบสถานะของบริการที่สำคัญ: ใช้คำสั่ง systemctl เพื่อตรวจสอบบริการหลัก ๆ ที่ควรทำงาน เช่น ssh, cron

Bash

```
sudo systemctl status ssh  
sudo systemctl status cron
```



- การตั้งค่าไฟร์วอลล์พื้นฐาน: ตั้งค่าไฟร์วอลล์เบื้องต้นด้วย ufw (ถ้ามีการติดตั้ง)
 - อนุญาตเฉพาะพอร์ตที่จำเป็น เช่น SSH

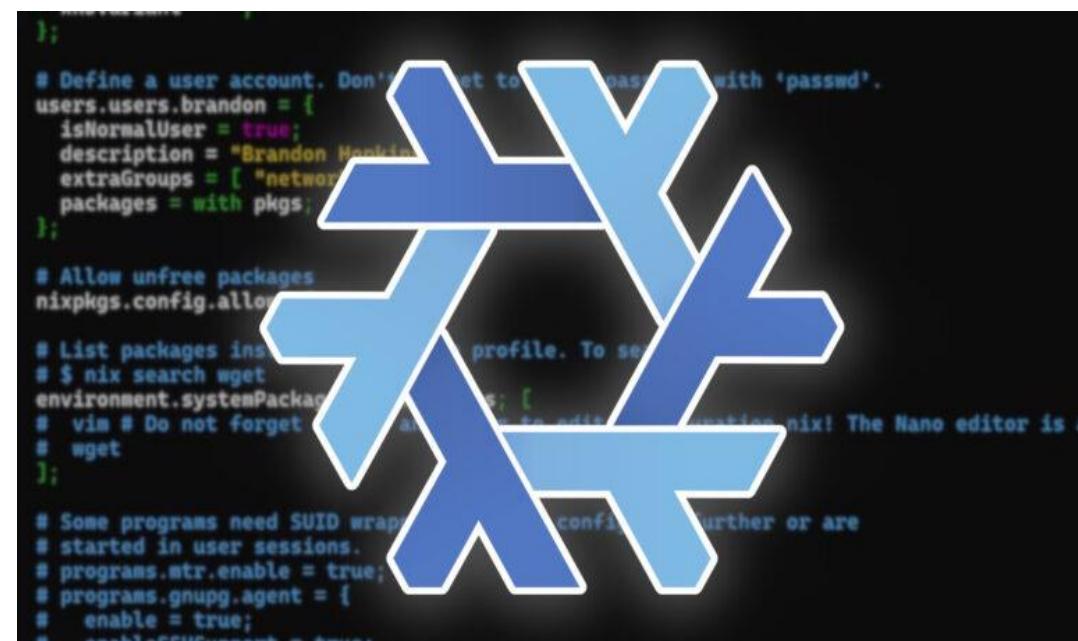
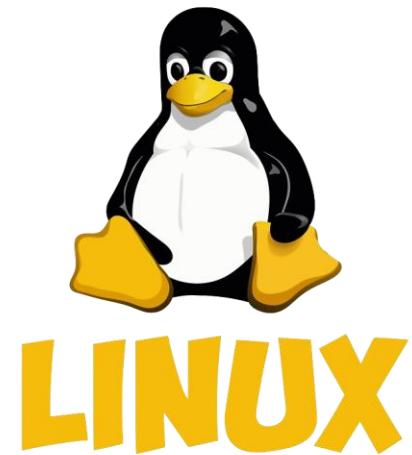
Bash

```
sudo ufw allow OpenSSH  
sudo ufw enable
```



ເຊື້ອລິສຕົງການຕຽບແຈ້ງການຕິດຕັ້ງ

- ตรวจสอบการเชื่อมต่อเครือข่าย: ตรวจสอบ IP Address, การเชื่อมต่อ DNS, และ Gateway
 - ตรวจสอบการติดตั้งพาร์ติชัน: ยืนยันพื้นที่ดิสก์และพาร์ติชันที่ใช้งาน
 - สร้างผู้ใช้ใหม่และมอบสิทธิ์ sudo
 - อัปเดตระบบและติดตั้งแพ็กเกจเก็บพื้นฐาน
 - ตรวจสอบสถานะบริการและตั้งค่าไฟร์วอลล์



คำสั่งพื้นฐานใน Linux (Basic Linux Command)

คำสั่งที่ใช้ป่วยสำหรับการจัดการไฟล์และไดเร็กทอรี

- ls: แสดงรายการไฟล์และไดเร็กทอรี

Bash

```
ls      # แสดงรายการไฟล์และไดเร็กทอรีใน
        # ไดเร็กทอรีปัจจุบัน
```

```
ls -l   # แสดงรายการไฟล์พร้อมรายละเอียด
```

เพิ่มเติม

- cd: เปลี่ยนไดเร็กทอรี

Bash

```
cd /path/to/directory # ไปยังไดเร็กทอรีที่
        # ต้องการ
```

```
cd ..       # ย้อนกลับไปยังไดเร็ก
```

ทอยู่ก่อนหน้า

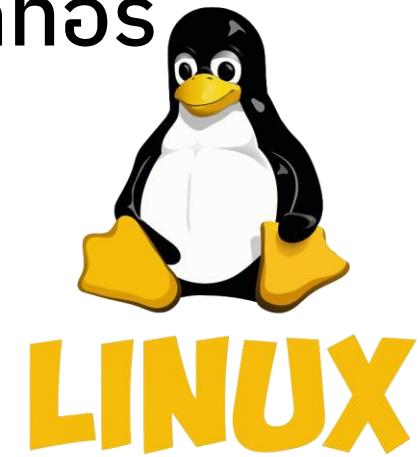
- cp: คัดลอกไฟล์หรือไดเร็กทอรี

Bash

```
cp source.txt destination.txt # คัดลอกไฟล์
```

```
cp -r source_dir destination_dir # คัดลอกไดเร็ก
```

ทอรีทั้งหมด



คำสั่งที่ใช้ป่วยสำหรับการจัดการไฟล์และไดเร็กทอรี

- mv: ย้ายหรือเปลี่ยนชื่อไฟล์และไดเร็กทอรี

Bash

```
mv oldname.txt newname.txt # เปลี่ยนชื่อไฟล์
```

```
mv /source /destination # ย้ายไฟล์  
หรือไดเร็กทอรี
```

- rm: ลบไฟล์หรือไดเร็กทอรี

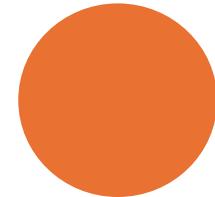
Bash

```
rm filename.txt # ลบไฟล์
```

```
rm -r directory_name # ลบไดเร็กทอรีและไฟล์ภายใน
```



LINUX




```
};  
# Define a user account. Don't forget to pass 'password'.  
users.users.brandon = {  
    isNormalUser = true;  
    description = "Brandon Hodge";  
    extraGroups = [ "netwo...  
    packages = with pkgs;  
};  
# Allow unfree packages  
nixpkgs.config.allowUnfree = true;  
# List packages in the system  
# $ nix search wget  
environment.systemPackages = [ vim # Do not forget  
# wget  
];  
# Some programs need SUID wrappers  
# started in user sessions.  
# programs.mtr.enable = true;  
# programs.grpck.agent = {  
#     enable = true;
```

คำสั่งจัดการผู้ใช้ เช่น การเพิ่ม ลบ และปรับแต่งสิทธิ์ของผู้ใช้

- useradd: สร้างผู้ใช้ใหม่

Bash

```
sudo useradd newuser      # สร้างผู้ใช้ใหม่ชื่อ
                           newuser
sudo passwd newuser       # ตั้งรหัสผ่านให้ผู้ใช้
```

- usermod: ปรับแต่งผู้ใช้ที่มีอยู่ เช่น การเพิ่มกลุ่ม

Bash

```
sudo usermod -aG sudo newuser  # เพิ่ม newuser เข้าไปใน
                               # กลุ่ม sudo
```

- userdel: ลบผู้ใช้

Bash

```
sudo userdel newuser        # ลบผู้ใช้ newuser
                           # ลบผู้ใช้และโฉนด
```

เร็กทอรีของผู้ใช้



คำสั่งเพิ่มเติมที่มีประโยชน์ในการจัดการไฟล์และไดเรกทอรี

- mkdir: สร้างไดเรกทอรีใหม่

Bash

```
mkdir directory_name    # สร้างไดเรกทอรีชื่อ  
directory_name
```

- touch: สร้างไฟล์เปล่าใหม่หรืออัปเดตเวลาเข้าถึงของไฟล์

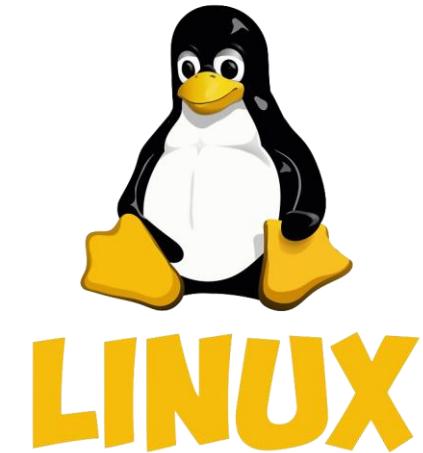
Bash

```
touch filename.txt      # สร้างไฟล์เปล่าชื่อ  
filename.txt
```

- cat: แสดงเนื้อหาไฟล์

Bash

```
cat filename.txt        # แสดงเนื้อหาของ  
filename.txt
```

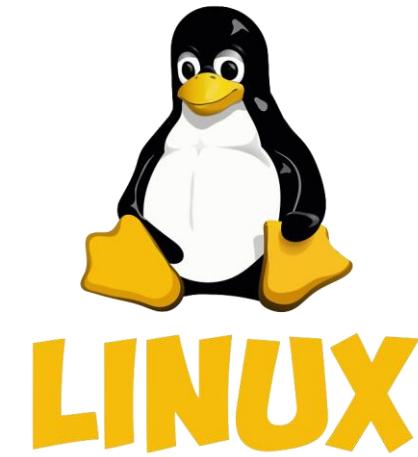


คำสั่งเพิ่มเติมที่มีประโยชน์ในการจัดการไฟล์และไดเร็กทอรี

- nano / vi: เปิดและแก้ไขไฟล์ด้วยตัวแก้ไขข้อความ

Bash

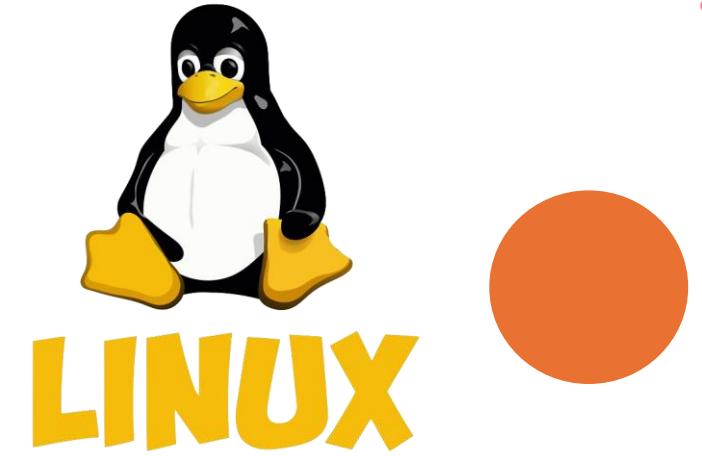
```
nano filename.txt      # แก้ไขไฟล์ด้วย nano  
vi filename.txt       # แก้ไขไฟล์ด้วย vi
```



Runlevels และการปิดระบบ (Shutdown)

Runlevel และ Systemd Targets

- Runlevel: เป็นระดับสถานะการทำงานที่ระบบปฏิบัติการ Linux สามารถบูตหรือเปลี่ยนแปลงได้โดยก้าวไป Runlevel มีการตั้งค่าดังนี้:
 - 0: ปิดเครื่อง
 - 1: โหมดผู้ใช้คนเดียว (Single-user mode)
 - 3: โหมดผู้ใช้หลายคน (Multi-user mode) ไม่มี GUI
 - 5: โหมดผู้ใช้หลายคน พร้อม GUI
 - 6: รีบูตเครื่อง
- Systemd Targets: ในระบบที่ใช้ Systemd (แทน Runlevel) จะมี "Targets" ที่กำหนดให้คล้ายกัน แต่มีความยืดหยุ่นมากขึ้น เช่น:
 - poweroff.target: ปิดเครื่อง
 - rescue.target: โหมดภัยคัน
 - multi-user.target: โหมดหลายคนใช้
 - graphical.target: โหมด GUI พร้อมการทำงานของผู้ใช้หลายคน
 - reboot.target: รีบูตเครื่อง



```
;;
# Define a user account. Don't forget to set a password with 'passwd'.
users.users.brandon = {
    isNormalUser = true;
    description = "Brandon Manic";
    extraGroups = [ "network" ];
    packages = with pkgs;
};

# Allow unfree packages
nixpkgs.config.allowUnfree = true;

# List packages installed by default
environment.systemPackages = [
    # vim # Do not forget to add it to your .vimrc
    # wget
];

# Some programs need SUID wrapper
# started in user sessions.
# programs.xtr.enable = true;
# programs.gnupg.agent = {
#     enable = true;
#     startAtLogin = true;
# };

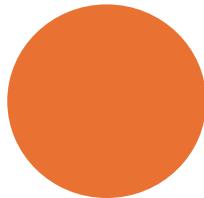
# Set up the root password
rootPassword = "password";
```

Runlevel และ Systemd Targets

- การตรวจสอบ Target ปัจจุบัน

Bash

```
systemctl get-default
```



```
};  
# Define a user account. Don't forget to pass 'password' with 'passed'.  
users.users.brandon = {  
    isNormalUser = true;  
    description = "Brandon Mull";  
    extraGroups = [ "netwo...  
    packages = with pkgs;  
};  
# Allow unfree packages  
nixpkgs.config.allowingUnfree = true;  
# List packages in the system  
# $ nix search wget  
environment.systemPackages = [  
    vim # Do not forget to add 'allowUnfree = true;' to your  
    # wget  
];  
# Some programs need SUID wrappers.  
# These are usually located in /etc/sudoers.d/  
# started in user sessions.  
# programs.mtr.enable = true;  
# programs.grpck.agent = {  
#     enable = true;
```

คำสั่งสำหรับปิดเครื่องและรีบูตระบบ

- shutdown: ปิดเครื่องหรือรีบูตหลังเวลาที่กำหนด

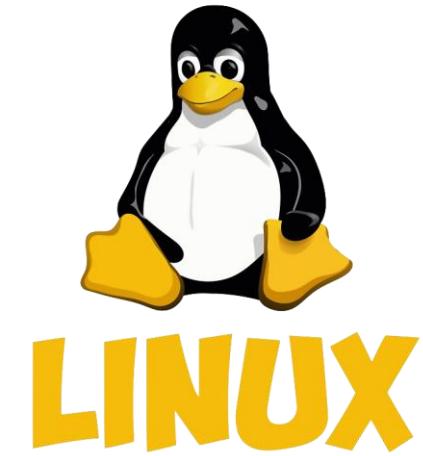
Bash

```
sudo shutdown now          # ปิดเครื่องทันที
sudo shutdown -r now       # รีบูตทันที
sudo shutdown +10          # ปิดเครื่องใน 10 นาที
sudo shutdown -c           # ยกเลิกการปิดเครื่อง
                           # ที่ตั้งไว้
```

- reboot: รีบูตระบบกันที

Bash

```
sudo reboot
```



คำสั่งสำหรับปิดเครื่องและรีบูตระบบ

- halt: หยุดระบบโดยไม่ปิดเครื่อง

Bash

```
sudo halt
```

- poweroff: ปิดเครื่องกันที

Bash

```
sudo poweroff
```



```
};  
# Define a user account. Don't forget to pass 'password' with 'passwd'.  
users.users.brandon = {  
    isNormalUser = true;  
    description = "Brandon Hart";  
    extraGroups = [ "network" ];  
    packages = with pkgs;  
};  
# Allow unfree packages  
nixpkgs.config.allowUnfree = true;  
# List packages in /etc/nixos/default.nix  
# nix search wget  
environment.systemPackages = [  
    vim # Do not forget to add 'syntax=nano' to your  
    # .vimrc  
    wget  
];  
# Some programs need SUID wrappers.  
# They are started in user sessions.  
# programs.mtr.enable = true;  
# programs.grpck.agent = {  
#     enable = true;  
# }
```

A watermark logo for Nix, featuring a stylized blue 'X' shape composed of multiple intersecting lines, set against a dark background with some green text from a configuration file visible at the bottom.

การเปลี่ยน Default Systemd Target

- เปลี่ยน Default Target: กำหนด Systemd Target ที่ระบบจะbootเข้าโดยอัตโนมัติ

Bash

```
sudo systemctl set-default multi-user.target # ตั้งค่าให้บูตเข้า
โหมดหลายผู้ใช้
```

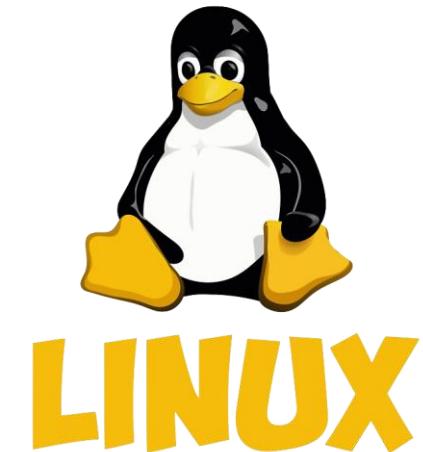
```
sudo systemctl set-default graphical.target # ตั้งค่าให้บูตเข้า
โหมด GUI
```

- เปลี่ยน Target ขณะใช้งาน: เปลี่ยนสถานะการทำงานในขณะระบบเปิดใช้งานอยู่

Bash

```
sudo systemctl isolate multi-user.target # เปลี่ยนไปโหมด
หลายผู้ใช้
```

```
sudo systemctl isolate graphical.target # เปลี่ยนไปโหมด GUI
```



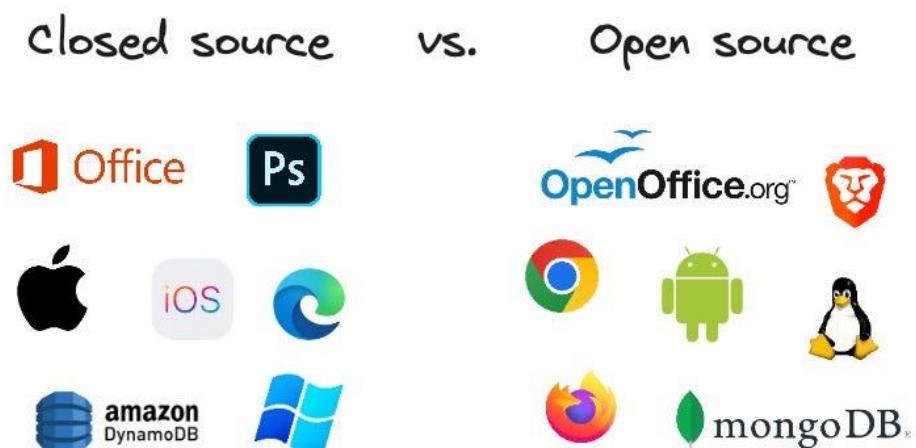
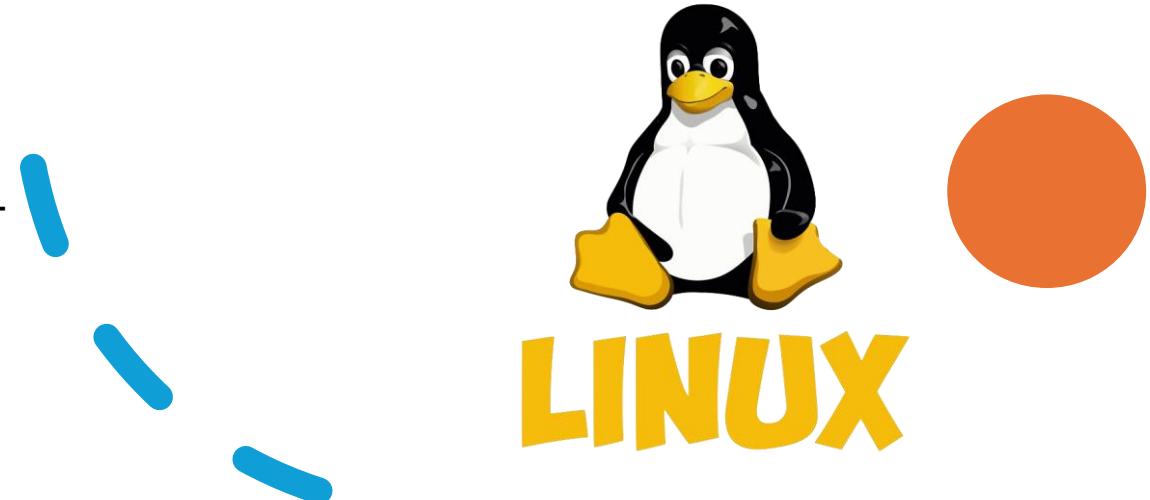
สรุปและ Q&A

Day 2: การจัดการระบบและคำสั่งขั้นสูง ใน Linux

ประวัติของ Free Software และ Open Source

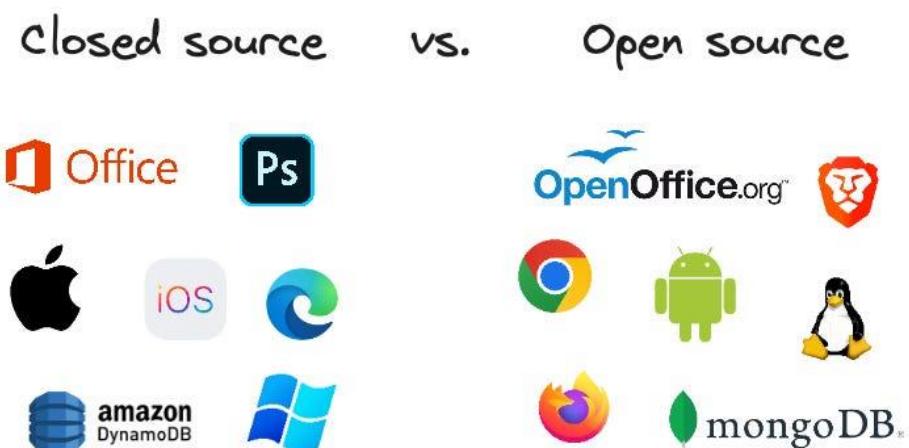
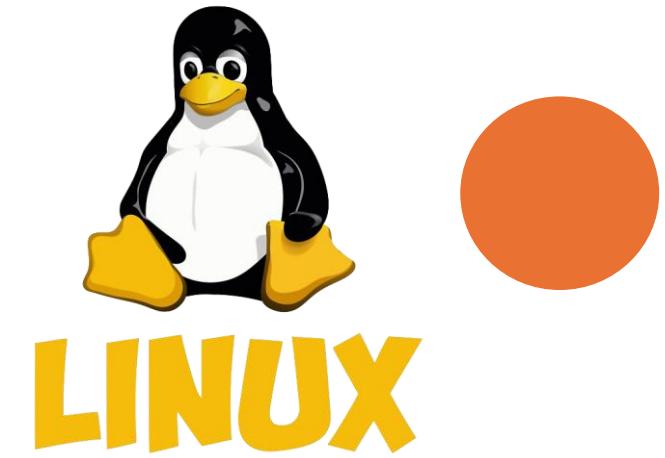
ความหมายและความสำคัญของ Free Software และ Open Source

- Free Software: แนวคิดที่พัฒนาขึ้นโดย Richard Stallman ในช่วงปี 1980 ที่เน้นเสรีภาพในการใช้งาน ซอฟต์แวร์ต้องมีการเข้าถึงและแก้ไขโค้ดได้ วิสัยในการใช้งาน แก้ไข แบ่งปัน และกระจายซอฟต์แวร์
 - หลัก 4 ประการของ Free Software:
 - การใช้งานซอฟต์แวร์
 - การศึกษาและปรับแต่งซอฟต์แวร์
 - การแจกจ่ายซอฟต์แวร์
 - การแจกจ่ายเวอร์ชันที่ปรับแต่งแล้ว
- Open Source Software (OSS): แนวคิดที่เน้นการพัฒนา ซอฟต์แวร์แบบเปิดให้กับคนเข้ามา มีส่วนร่วม ใช้ License ที่ เปิดโอกาสให้กับคนสามารถดู แก้ไข และพัฒนาซอฟต์แวร์ ต่อได้ ฟรี กสที่คุณภาพและความปลอดภัยของซอฟต์แวร์



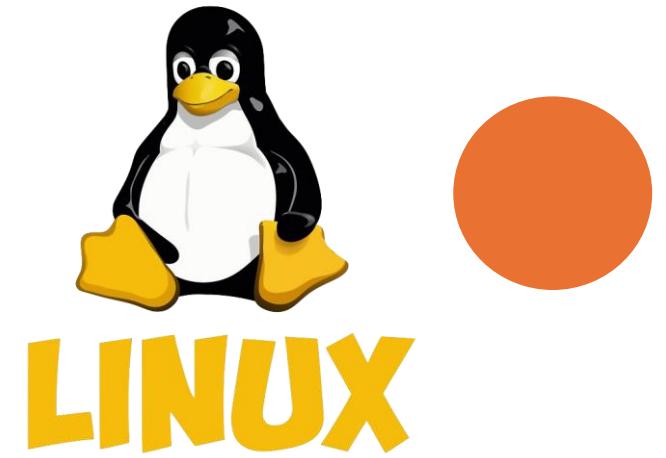
ความสำคัญของใบอนุญาตโอเพนซอร์ส เช่น GPL และ MIT License

- GNU General Public License (GPL): ใบอนุญาตที่กำหนดให้ซอฟต์แวร์ใด ๆ ที่ใช้ GPL ต้องเปิดเผย源代码 และอนุญาตให้ผู้ใช้สามารถปรับปรุงหรือแจกจ่าย ซอฟต์แวร์นั้นได้ภายใต้เงื่อนไขเดียวกัน
 - ข้อกำหนดที่โดดเด่นของ GPL คือ Copyleft ซึ่งป้องกันไม่ให้ซอฟต์แวร์ที่ใช้ GPL ถูกเปลี่ยนเป็นซอฟต์แวร์ที่ไม่ได้เปิดเผย源代码
- MIT License: ใบอนุญาตแบบเสรีที่อนุญาตให้ผู้ใช้สามารถทำอะไรก็ได้กับซอฟต์แวร์ รวมถึงการนำไปพัฒนาและแจกจ่ายใหม่โดยไม่ต้องเปิดเผย源代码



ความแตกต่างระหว่าง Free Software กับ Open Source

- Free Software เน้น "เสรีภาพ" ของผู้ใช้ (Freedom) โดยเน้นกีเสรีภาพในการใช้งานและปรับปรุงซอฟต์แวร์
- Open Source Software เน้นการพัฒนาที่มีประสิทธิภาพคุณภาพ ความปลอดภัย ความร่วมมือ และความโปร่งใส
- แม้ว่าทั้งสองแนวคิดจะสนับสนุนการเข้าถึงซอฟต์แวร์โดยสาธารณะและการแบ่งปัน แต่เป้าหมายและคุณค่าอาจแตกต่างกันเล็กน้อย

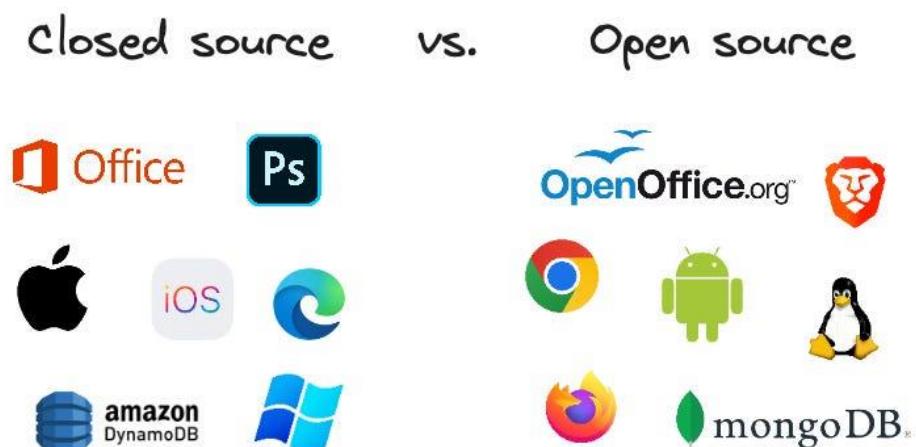
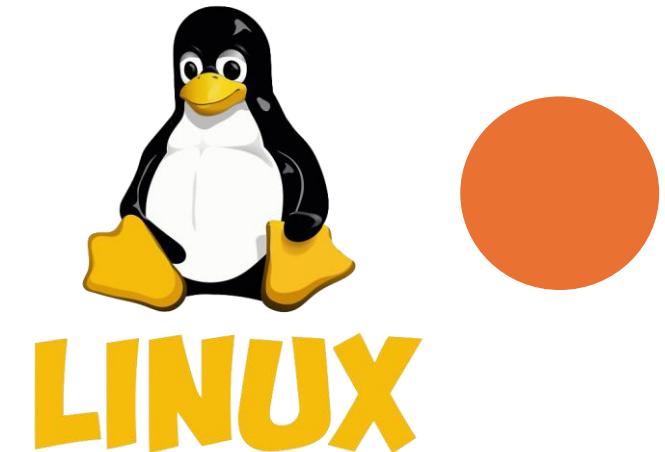


closed source vs. Open source



ตัวอย่างโครงการ Open Source ที่สำคัญ

- Linux Kernel: ระบบปฏิบัติการโอเพนซอร์สที่พัฒนาโดย Linus Torvalds และชุมชนซอฟต์แวร์เสรี ซึ่งเป็นรากฐานของระบบปฏิบัติการมากมาย
- Apache HTTP Server: เว็บเซิร์ฟเวอร์โอเพนซอร์สที่ใช้กันอย่างแพร่หลายในระบบเซิร์ฟเวอร์เว็บทั่วโลก
- MySQL: ระบบจัดการฐานข้อมูลที่เป็นที่นิยม ใช้สำหรับการจัดเก็บและจัดการข้อมูลในรูปแบบฐานข้อมูล SQL

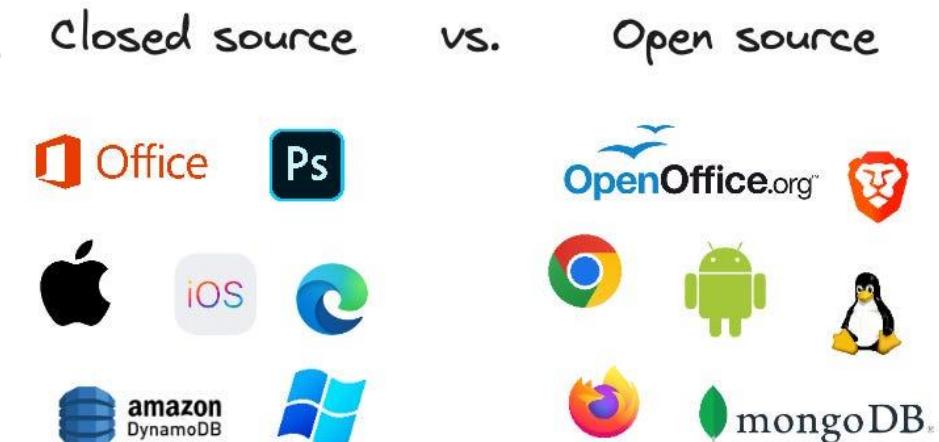
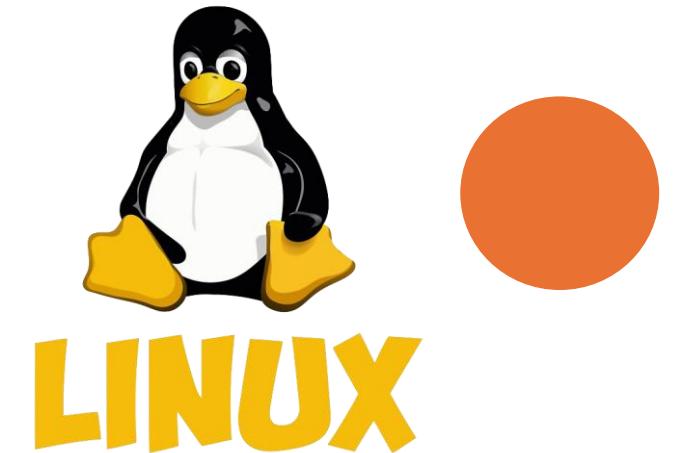


Another Linux Distribution

By Sommanut Ketpong

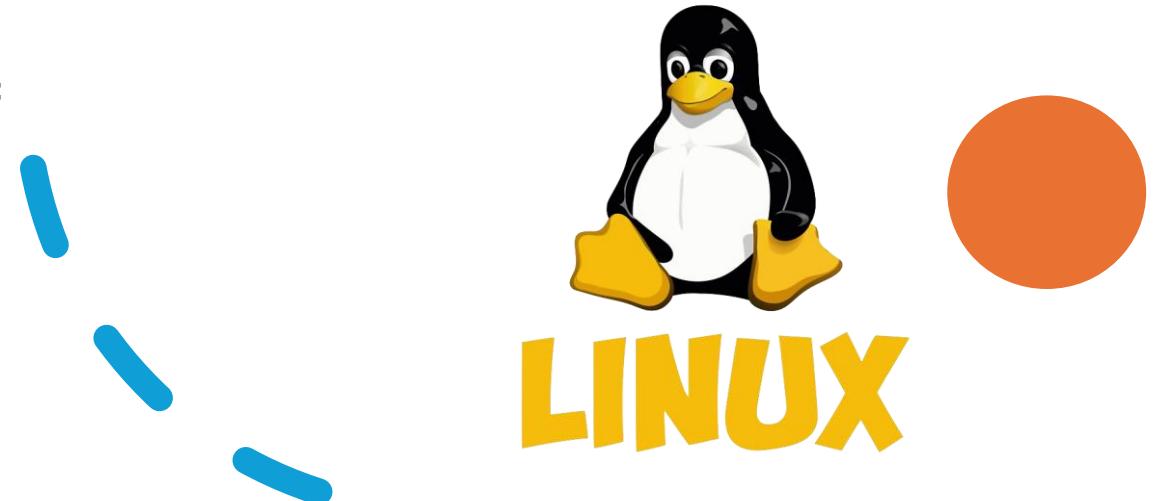
Linux Distributions

- Linux Distribution: เป็นระบบปฏิบัติการ Linux ที่ปรับแต่งให้เหมาะสมกับการใช้งานเฉพาะด้าน มีการรวมซอฟต์แวร์ เครื่องมือ และการตั้งค่าต่าง ๆ เข้าไว้ด้วยกัน ทำให้ใช้งานได้สะดวกโดยไม่ต้องตั้งค่าใหม่
- ตัวอย่างดิสโกรยอดนิยมที่เหมาะสมกับการเรียนรู้, เชิร์ฟเวอร์, หรือการพัฒนาระบบ เช่น:
 - Debian: ดิสโกรที่ขึ้นชื่อเรื่องความเสถียรและกوارดูแลรักษาความปลอดภัย ใช้แพ็คเกจระบบ .deb และเป็นพื้นฐานของหลายดิสโกร เช่น Ubuntu
 - Arch Linux: เน้นความยืดหยุ่นและการปรับแต่งเองอย่างอิสระ ให้ผู้ใช้ติดตั้งซอฟต์แวร์ตามความต้องการและใช้ระบบจัดการแพ็คเกจ pacman
 - SUSE Linux Enterprise: ระบบที่ออกแบบมาสำหรับองค์กร เน้นความเสถียรและการสนับสนุนการทำงานค้า ใช้ในอุตสาหกรรมและงานวิจัยต่าง ๆ



Debian: จุดเด่นและการใช้งาน

- Debian: เป็นที่รู้จักในด้านเสถียรภาพ ความน่าเชื่อถือ และ การสนับสนุนระยะยาว มีการจัดการซอฟต์แวร์ที่มั่นคงและ ปลอดภัย
- เมฆะสำหรับ: เซิร์ฟเวอร์ที่ต้องการความเสถียรสูง เช่น เว็บเซิร์ฟเวอร์ ฐานข้อมูล และระบบที่ต้องการ uptime สูง
- ระบบแพ็คเกจ: ใช้ apt และ .deb ซึ่งง่ายต่อการติดตั้ง และอัปเดตแพ็คเกจ

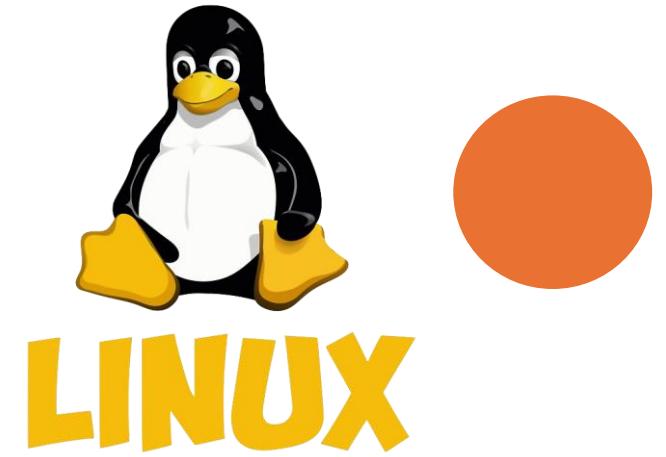


closed source vs. Open source



Arch Linux: จุดเด่นและการใช้งาน

- Arch Linux: มีแนวคิดแบบ KISS (Keep It Simple, Stupid) ให้ผู้ใช้ควบคุมทุกขั้นตอนในการติดตั้งและปรับแต่ง เมื่อมาถึงผู้ใช้ที่ต้องการเรียนรู้โครงสร้างภายในของ Linux
- เหมาะสำหรับ: นักพัฒนาและผู้ใช้ที่ต้องการระบบที่เบาและปรับแต่งได้ตามใจชอบ
- ระบบแพ็คเกจ: ใช้ pacman สำหรับจัดการแพ็คเกจและมี AUR (Arch User Repository) เป็นคลังซอฟต์แวร์เพิ่มเติม

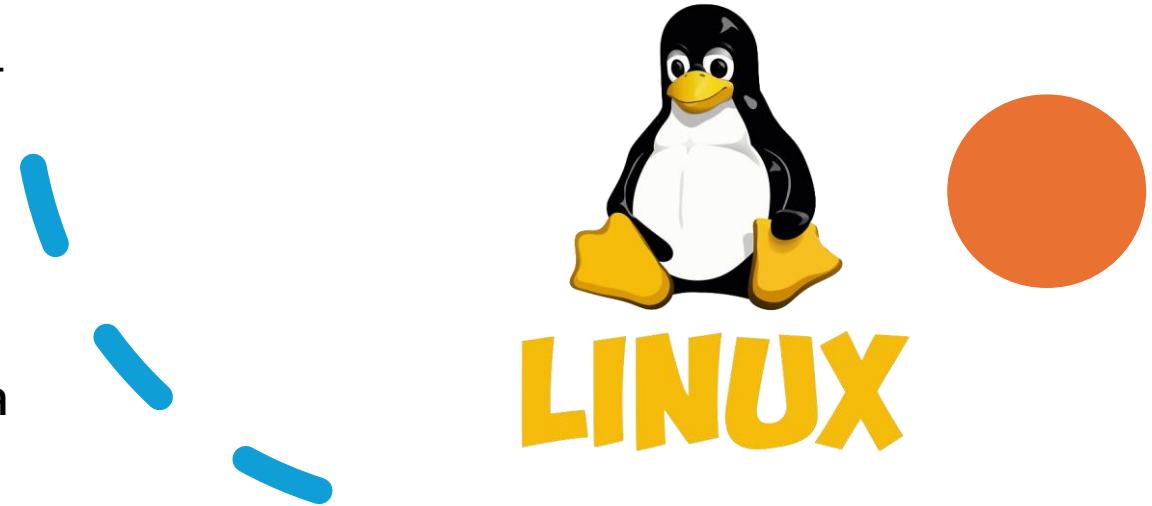


closed source vs. Open source



SUSE Linux Enterprise: จุดเด่นและการใช้งาน

- SUSE Linux Enterprise: ได้รับการสนับสนุนโดยองค์กร
กางการค้า มีเครื่องมือ YaST สำหรับการจัดการระบบที่ใช้งานง่ายและช่วยดูแลโครงสร้างพื้นฐานขององค์กรได้ดี
- แนะนำสำหรับ: การใช้งานในองค์กรขนาดใหญ่ที่ต้องการ
การสนับสนุนเชิงพาณิชย์ในด้านความเสถียรและการ
จัดการระบบ
- การปรับแต่ง: รองรับการตั้งค่าเซิร์ฟเวอร์ การประมวลผล
ประสิทธิภาพสูง และการปรับแต่งด้านความปลอดภัยที่สูง



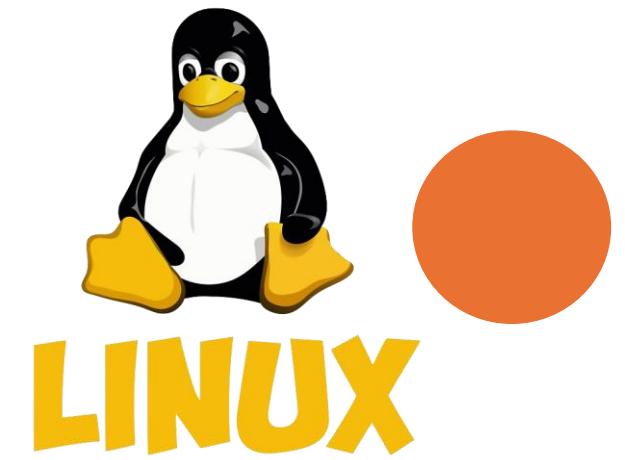
closed source vs. Open source



ແຜນຝັງເປົ້າຍບເກີຍບຄຸນສມບັດຂອງ Linux Distributions

ຕາරາງເປົ້າຍບເກີຍບດ້ານ ຈຸດເດັ່ນ, ແພັກເກຈກີ່ໃຊ້, ກລຸ່ມຜູ້ໃຊ້
ເປົ້າໝາຍ, ຮະດັບກາຮສນັບສຸນ, ແລະ ກາຮໃຊ້ງານທີ່ເໝາະສົມ

Distribution	ຈຸດເດັ່ນ	ຮະບນແພັກເກຈ	ເໜາະສຳຫັບ	ກາຮສນັບສຸນ
Debian	ເສົ່າຍរກາພ, ຄວາມປລອດກັບຍ	.deb	ເຊື່ອົ້ວເວຼວ, ອອກຕົກຂານາດ ໃໝ່	ຊຸມຊົນ, ຮະຍະ ຍາວ
Arch Linux	ຢືດຫຸ່ນ, ປ່ຽນແຕ່ງເວັງ	pacman	ນັກພົມນາ, ຜູ້ໃຊ້ຂັ້ນສູງ	ຊຸມຊົນ
SUSE Linux	ຄວາມເສົ່າຍຮ, ສນັບສຸນ ອອກຕົກ	.rpm	ອອກຕົກຂານາດ ໃໝ່, HPC	ເຊິ່ງພານີ້ຍົງ



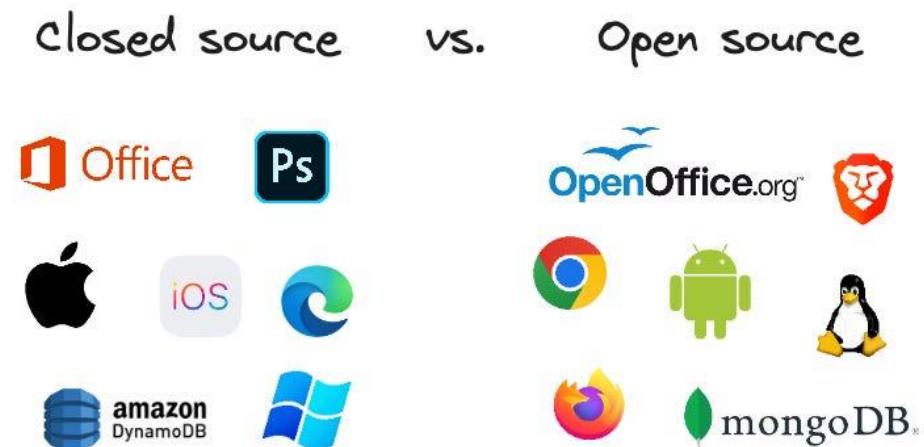
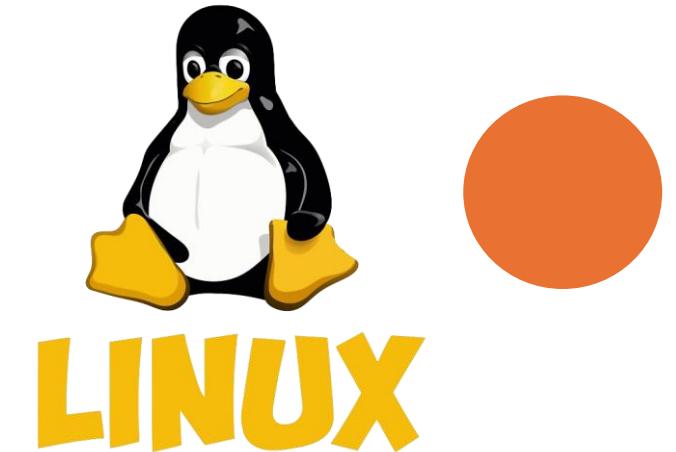
source vs. Open source



ความแตกต่างระหว่าง UNIX และ Linux

ความแตกต่างหลักระหว่าง UNIX และ Linux

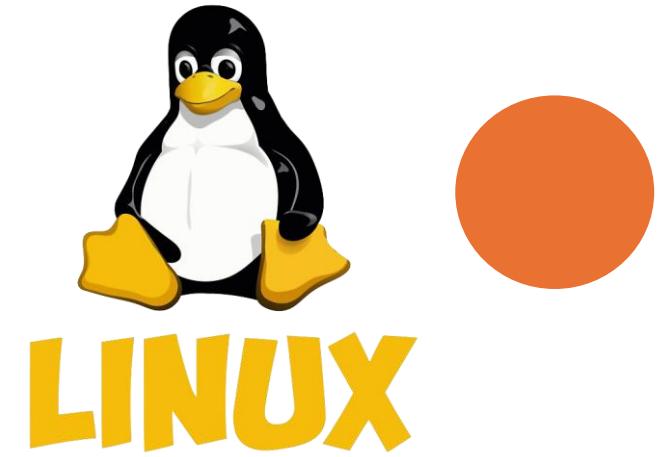
- การออกแบบและสถาปัตยกรรม:
 - UNIX: เป็นระบบปฏิบัติการที่พัฒนามาตั้งแต่ช่วงปี 1970 และมักจะเป็นระบบปิด (Proprietary) ที่ควบคุมการเข้าถึงซอฟต์แวร์
 - Linux: เป็นระบบที่เปิดเผยแพร่ซอฟต์แวร์โดยใช้แนวคิดแบบโอเพนซอร์ส ทำให้ผู้ใช้สามารถแก้ไขและแจกจ่ายได้ อิสระ
- การใช้งานและเป้าหมาย:
 - UNIX: ใช้ในเซิร์ฟเวอร์องค์กรขนาดใหญ่, ธุรกิจที่ต้องการความเสถียร เช่น การเงินและธนาคาร
 - Linux: ใช้งานได้หลากหลายทั้งบนเดสก์ท็อป, เซิร์ฟเวอร์, อุปกรณ์ IoT และในระบบคลาวด์
- การสนับสนุนฮาร์ดแวร์:
 - UNIX: มักจะออกแบบให้รองรับฮาร์ดแวร์เฉพาะของผู้ผลิต เช่น IBM AIX สำหรับ IBM Hardware
 - Linux: รองรับฮาร์ดแวร์ที่หลากหลายตั้งแต่คอมพิวเตอร์เดสก์ท็อปไปจนถึงชูเปอร์คอมพิวเตอร์และอุปกรณ์ฝังตัว (Embedded Devices)



ปรียบเทียบคุณสมบัติของ UNIX และ Linux

- ตารางสรุปปรียบเทียบคุณสมบัติสำคัญ

คุณสมบัติ	UNIX	Linux
ประเภท	ระบบปิด (Proprietary)	ระบบเปิด (Open Source)
การใช้งาน	เซิร์ฟเวอร์องค์กร	เซิร์ฟเวอร์, เดสก์ท็อป, IoT
การสนับสนุนฮาร์ดแวร์	ฮาร์ดแวร์เฉพาะ	หลากหลาย
การพัฒนาและปรับแต่ง	จำกัด	เปิดกว้าง
ลิขสิทธิ์	ควบคุมโดยบริษัทใหญ่	ชุมชนและนักพัฒนาทั่วโลก



LINUX

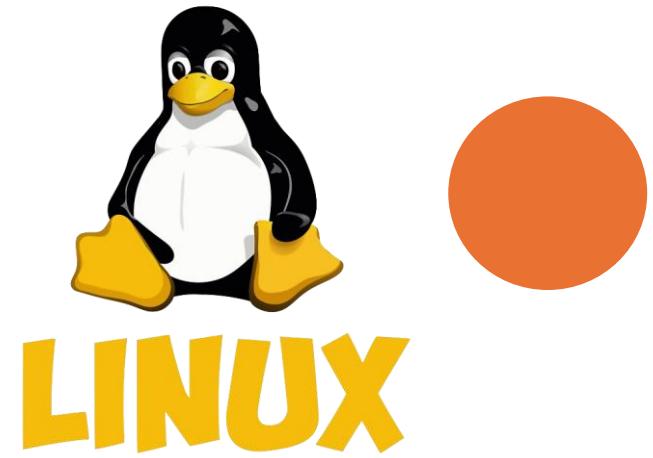
sed source vs. Open source



การจัดการ Root และ User Prompt

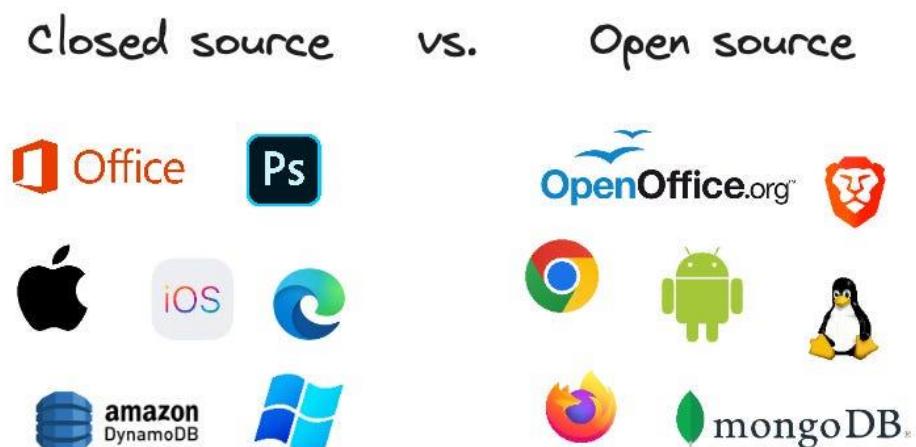
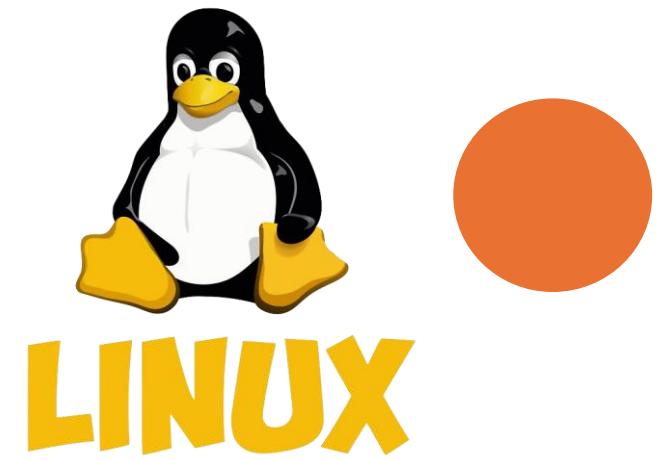
บทบาทและความสำคัญของ Root (Superuser)

- Root (Superuser): บัญชีผู้ใช้ที่มีสิทธิ์สูงสุดในระบบ Linux สามารถทำทุกอย่างได้ในระบบ เช่น การติดตั้งและถอนการติดตั้งโปรแกรม การจัดการระบบไฟล์กั้งหมด และการกำหนดค่าความปลอดภัย
- ข้อควรระวัง: การใช้ root គุ่ระมัดระวังเพราะอาจทำให้ระบบเสียหายได้หากใช้คำสั่งผิด เนื่องจากคำสั่งจะถูกประมวลผลโดยไม่จำกัดสิทธิ์
- ตัวอย่าง: เมื่อเข้าสู่ระบบด้วยบัญชี root จะเห็น prompt เป็น # ซึ่งแสดงว่าสามารถเข้าถึงสิทธิ์สูงสุดในระบบได้



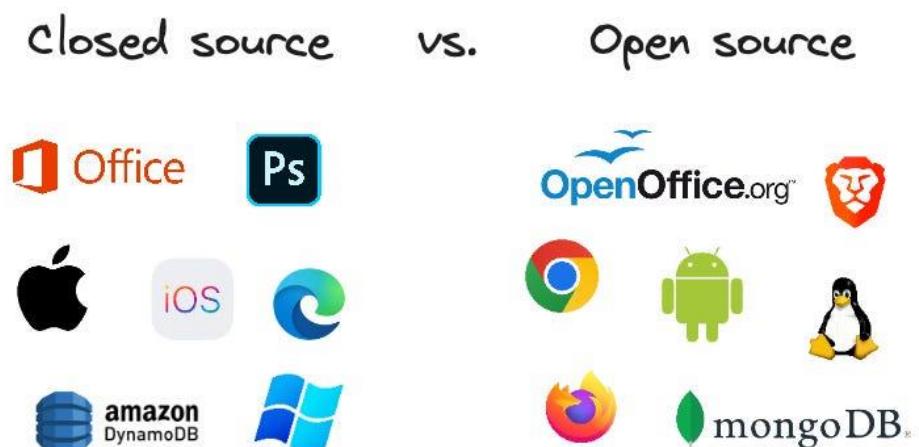
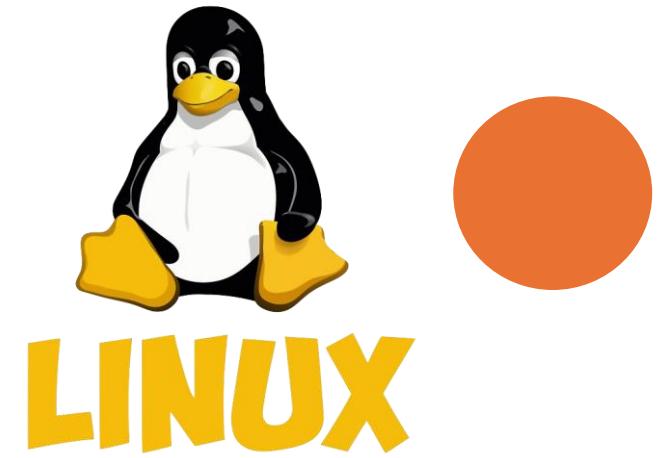
คำสั่ง sudo และข้อดีในการใช้งาน

- Sudo: คำสั่งที่อนุญาตให้ผู้ใช้หัวไปเรียกใช้คำสั่งที่ต้องการสิทธิ์สูงกว่าได้โดยไม่ต้องเข้าสู่ระบบเป็น root
- ประโยชน์ของ Sudo:
- ลดความเสี่ยง: ช่วยป้องกันการกำจัดผลลัพธ์โดยไม่ได้ตั้งใจ
- สามารถจำกัดสิทธิ์: ผู้ดูแลสามารถกำหนดว่าใครมีสิทธิ์ใช้ sudo และใช้ได้กับคำสั่งใดบ้าง
- รูปแบบ: การใช้ sudo จะต้องพิมพ์ sudo นำหน้าคำสั่ง เช่น sudo apt update
- ข้อควรจำ: จะต้องใส่รหัสผ่านผู้ใช้เมื่อใช้ sudo



ความแตกต่างระหว่าง Root Prompt และ User Prompt

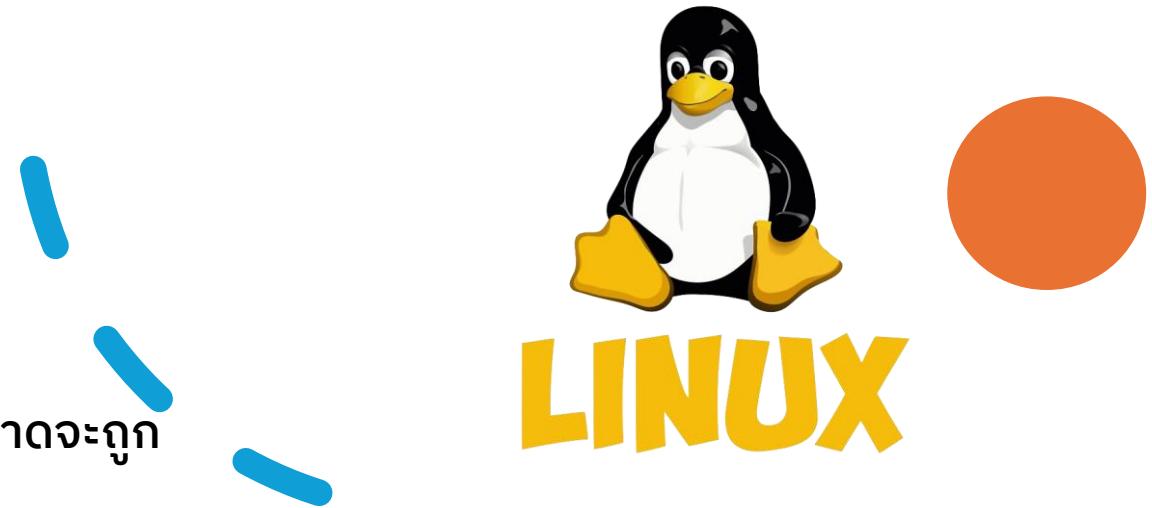
- User Prompt (\$): ใช้สำหรับผู้ใช้ทั่วไป (regular user) ที่มีสิทธิ์จำกัด ทำได้เฉพาะการจัดการไฟล์และการใช้งานทั่วไปที่ไม่ส่งผลกระทบต่อระบบโดยรวม
- Root Prompt (#): ใช้สำหรับ root หรือผู้ใช้ที่เรียกใช้งานด้วย sudo และถึงสิทธิ์สามารถเข้าถึงและแก้ไขการตั้งค่าระบบได้
- ข้อสังเกต: การทำงานบน root prompt ต้องระมัดระวัง เพราะคำสั่งผิดพลาดอาจกระทบต่อระบบกังหัน ขณะที่ user prompt จำกัดสิทธิ์เพื่อความปลอดภัย



การใช้งาน Symbol Re-direction และ Pipe

พื้นฐานการใช้งาน Symbol Re-direction

- > : เขียนผลลัพธ์ไปยังไฟล์ใหม่ (เขียนกับไฟล์เดิม)
 - ตัวอย่าง: ls > output.txt
- >> : เพิ่มผลลัพธ์ต่อท้ายไฟล์ที่มีอยู่แล้ว
 - ตัวอย่าง: echo "เพิ่มเติมข้อมูล" >> output.txt
- 2> : เขียนข้อผิดพลาดไปยังไฟล์ (เช่น stderr)
 - ตัวอย่าง: ls /invalid_path 2> error.txt (ข้อผิดพลาดจะถูกบันทึกใน error.txt)

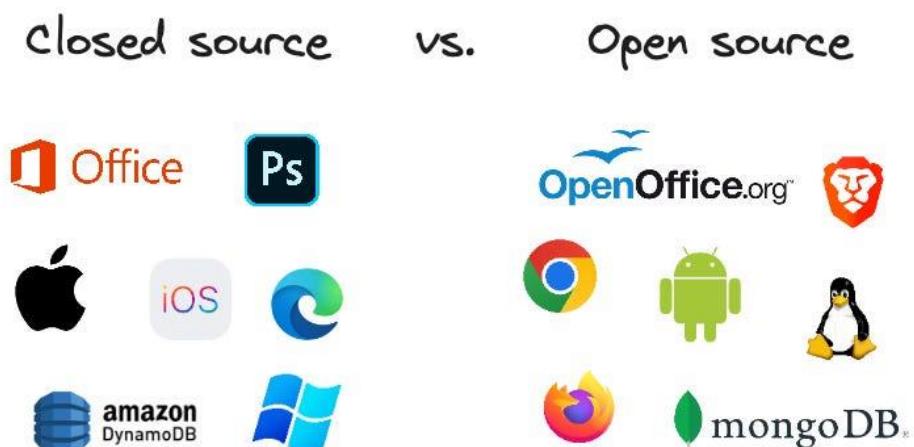
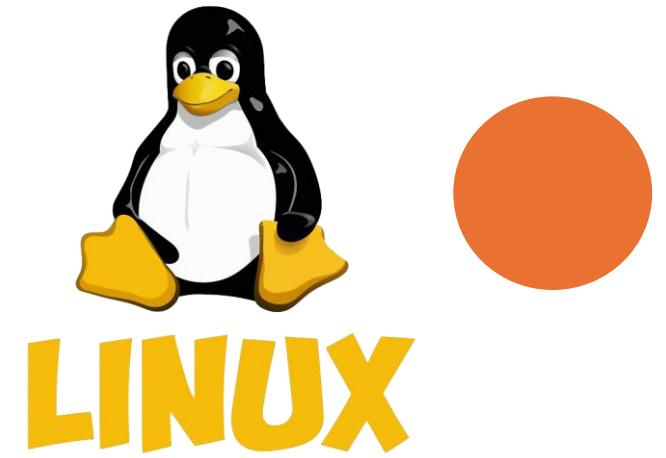


closed source vs. Open source



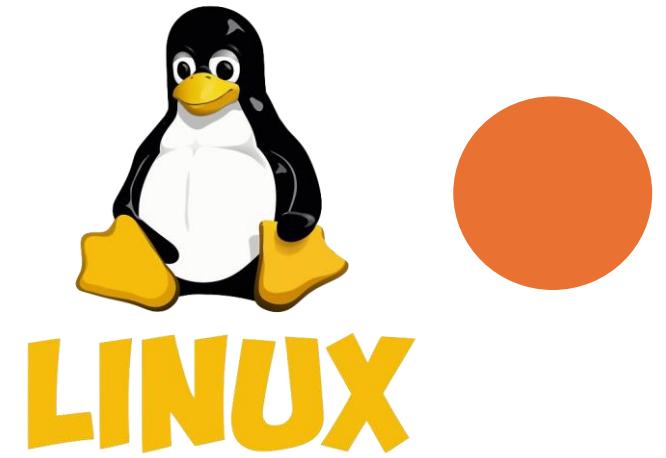
การใช้งาน Pipe (|) เพื่อเชื่อมต่อคำสั่ง

- | (Pipe): ใช้ส่งผลลัพธ์ของคำสั่งหนึ่งไปเป็นอินพุตของอีกคำสั่ง
 - ตัวอย่าง: ls | grep "config" (แสดงเฉพาะไฟล์หรือโฟลเดอร์ที่มีคำว่า "config")
- Pipe ช่วยให้คำสั่งต่าง ๆ ทำงานร่วมกันได้และทำให้การจัดการข้อมูลสะดวกขึ้น
- การประยุกต์ใช้: เช่น ใช้เพื่อกรองข้อมูล การค้นหา หรือการจัดรูปแบบผลลัพธ์



การใช้งาน Pipe (|) เพื่อเชื่อมต่อคำสั่ง

- | (Pipe): ใช้ส่งผลลัพธ์ของคำสั่งหนึ่งไปเป็นอินพุตของอีกคำสั่ง
 - ตัวอย่าง: ls | grep "config" (แสดงเฉพาะไฟล์หรือโฟลเดอร์ที่มีคำว่า "config")
- Pipe ช่วยให้คำสั่งต่าง ๆ ทำงานร่วมกันได้และทำให้การจัดการข้อมูลสะดวกขึ้น
- การประยุกต์ใช้: เช่น ใช้เพื่อกรองข้อมูล การค้นหา หรือการจัดรูปแบบผลลัพธ์



closed source vs. Open source



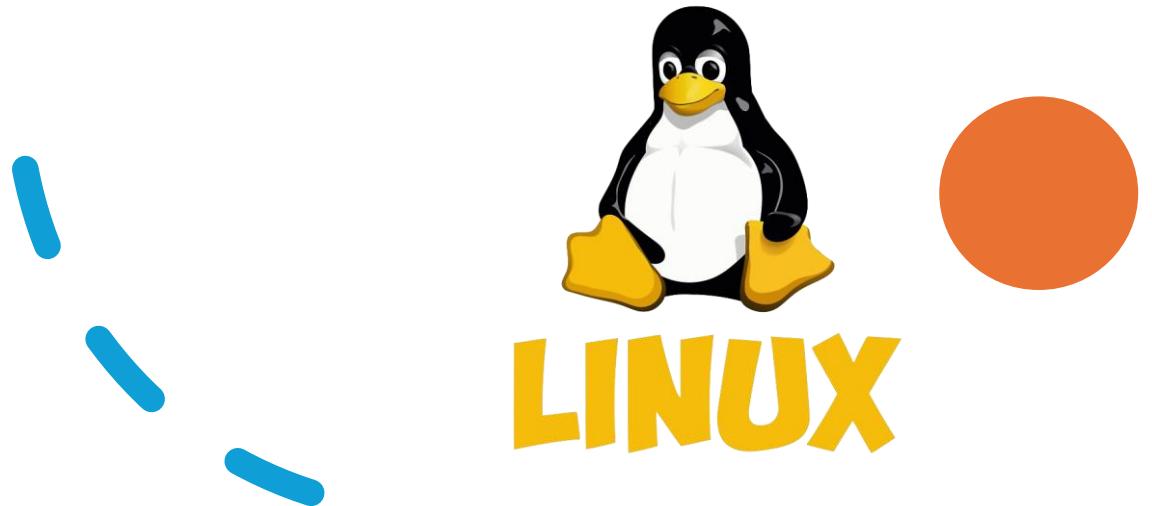
พื้นฐานของ Shell Script

โครงสร้างของ Shell Script

- การใช้ตัวแปร:
 - ประกาศตัวแปร: name="John"
 - เรียกใช้งานตัวแปร: echo \$name
- เงื่อนไข:
 - ใช้คำสั่ง if เพื่อกำการตรวจสอบเงื่อนไข
 - ตัวอย่าง:

Bash

```
if [ "$name" == "John" ]; then
    echo "Welcome back, John!"
fi
```



closed source vs. Open source

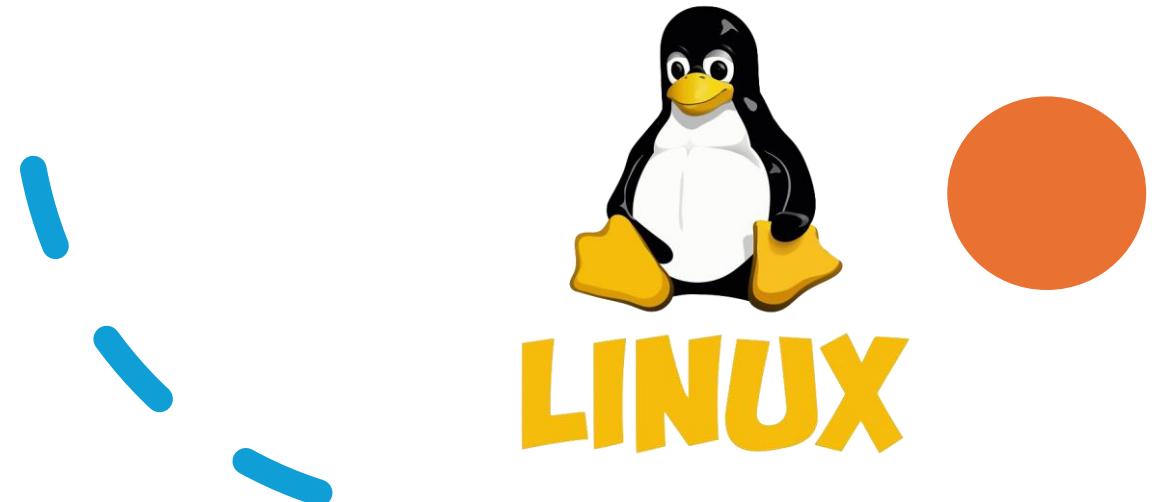


โครงสร้างของ Shell Script

- การวนลูป:
 - ใช้คำสั่ง for และ while เพื่อกำการวนลูป
 - ตัวอย่าง for:

Bash

```
for i in {1..5}; do  
    echo "Number $i"  
done
```

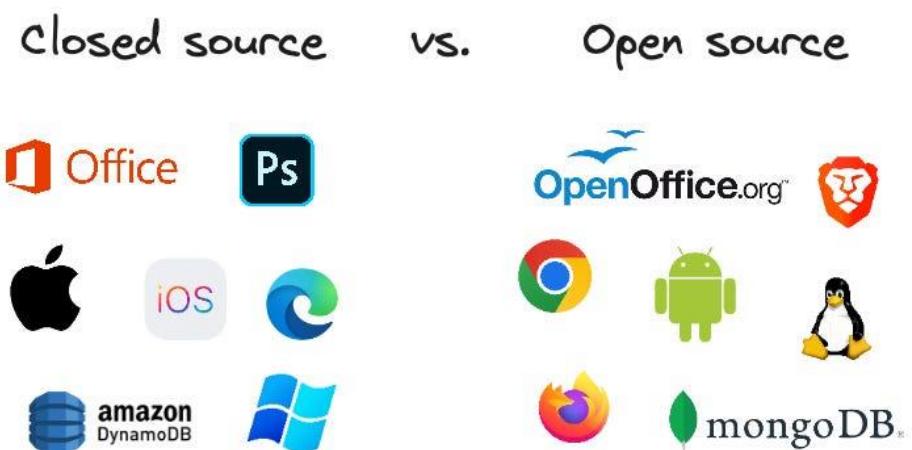
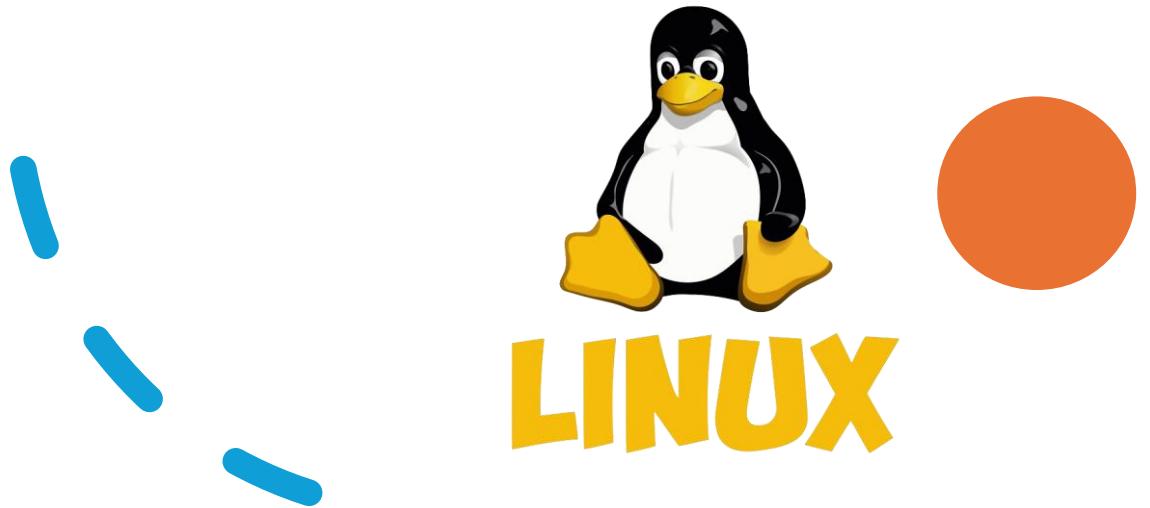


closed source vs. Open source



คำสั่งพื้นฐานใน Shell Script

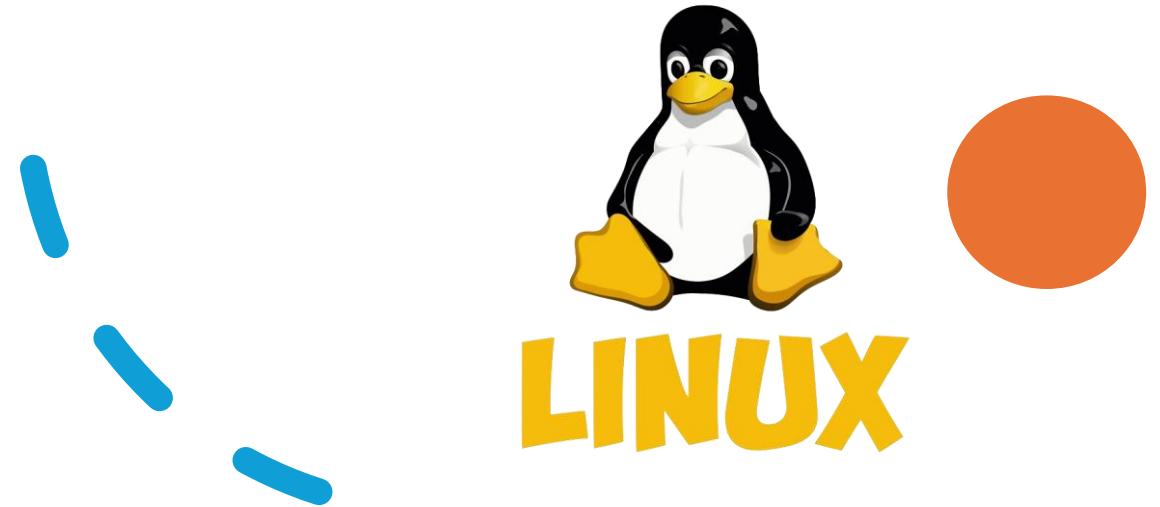
- echo: แสดงข้อความ
 - ตัวอย่าง: echo "Hello, World!"
- read: รับข้อมูลจากผู้ใช้
 - ตัวอย่าง: read name
- if: ตรวจสอบเงื่อนไข
- for: วนลูปตามจำนวนที่กำหนด
- while: วนลูปจนกว่าเงื่อนไขจะเป็นเท็จ



การตั้งงานอัตโนมัติด้วย Cron (Scheduled Tasks with cron)

การใช้งาน Cron

- Cron เป็นโปรแกรมใน Linux ที่ใช้สำหรับตั้งงานอัตโนมัติ (Scheduled Tasks) ที่จะทำงานตามเวลาที่กำหนด
- หมายความว่าการกำหนดเวลาที่ต้องการให้ทำซ้ำ เช่น การสำรองข้อมูล, การส่งอีเมล, หรือการอัปเดตระบบ
- Cron ทำงานในพื้นหลัง และไม่ต้องการการแทรกแซงจากผู้ใช้เมื่อทำงาน



closed source vs. Open source

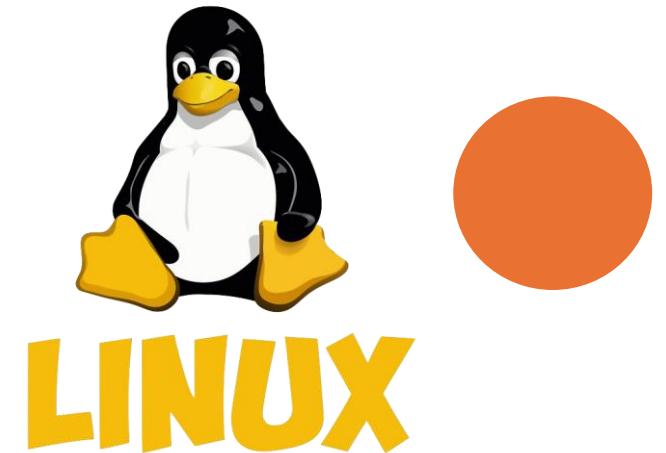


ไฟล์ Crontab

- ไฟล์ Crontab เป็นไฟล์ที่เก็บข้อมูลการตั้งค่างานอัตโนมัติของผู้ใช้
- โครงสร้างของ Crontab ประกอบด้วย:

```
* * * * * /path/to/command
```

T	T	T	T	T
วันในสัปดาห์ (0 - 7) (0 และ 7 หมายถึงวันอาทิตย์)				
เดือน (1 - 12)				
วันของเดือน (1 - 31)				
ชั่วโมง (0 - 23)				
นาที (0 - 59)				



- ตัวอย่าง:
 - 0 3 * * * /home/user/backup.sh: รันสคริปต์ backup.sh ทุกวันเวลา 03:00 น.

closed source vs. Open source



การตั้งค่า Cron Job

- การเพิ่มงานอัตโนมัติใน Cron ใช้คำสั่ง crontab -e เพื่อแก้ไขไฟล์ Crontab
- ตัวอย่างการใช้งาน:
 - สำรองข้อมูลทุกวันเวลา 3 โมงเช้า:

Bash

```
0 3 * * * /home/user/backup.sh
```

- ส่งอีเมลรายสัปดาห์ทุกวันจันทร์เวลา 8 โมงเช้า:

Bash

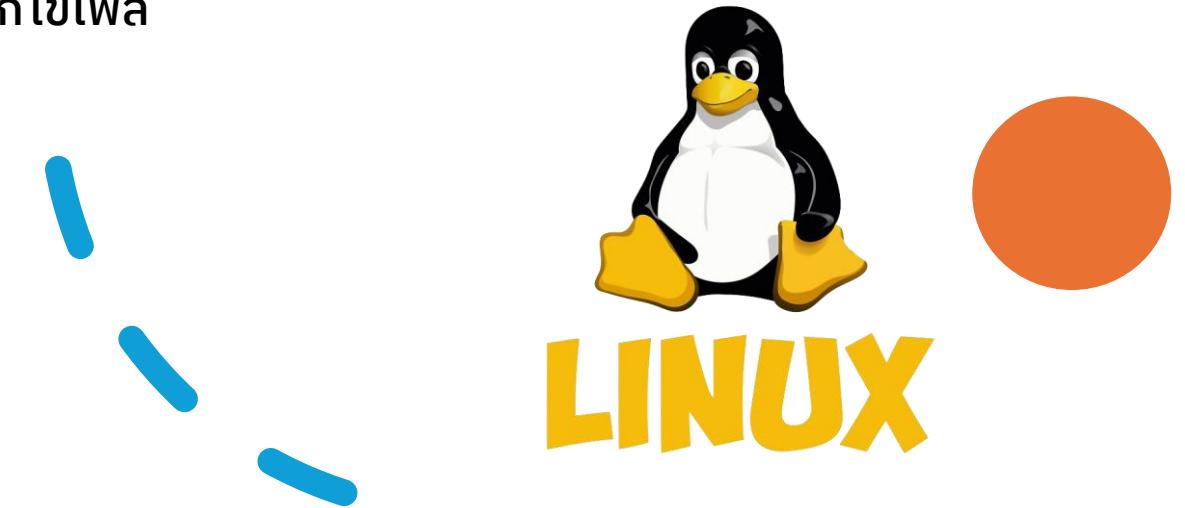
```
0 8 * * 1
```

```
/home/user/send_report.sh
```

- อัปเดตแพ็กเกจทุกวันอังคารเวลา 2 โมงเช้า:

Bash

```
0 2 * * 2 /usr/bin/apt update && /usr/bin/apt
upgrade -y
```



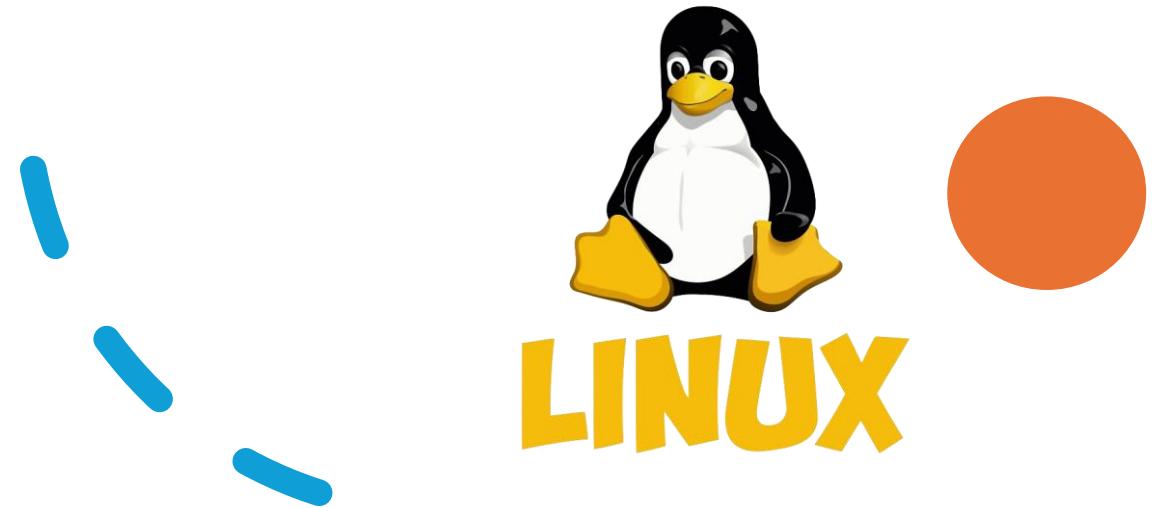
closed source vs. Open source



การจัดการพาร์ติชันดิสก์ (Disk Partition Management)

ความสำคัญของการจัดการพาร์ติชันดิสก์

- การจัดการพาร์ติชันดิสก์ช่วยเพิ่มความยืดหยุ่นในการใช้พื้นที่เก็บข้อมูล
- พาร์ติชันแยก เช่น /home, /var ทำให้ง่ายต่อการบริหารจัดการและสำรองข้อมูล
- การขยายหรือลดพาร์ติชันช่วยปรับให้เหมาะสมกับการใช้งานของระบบ

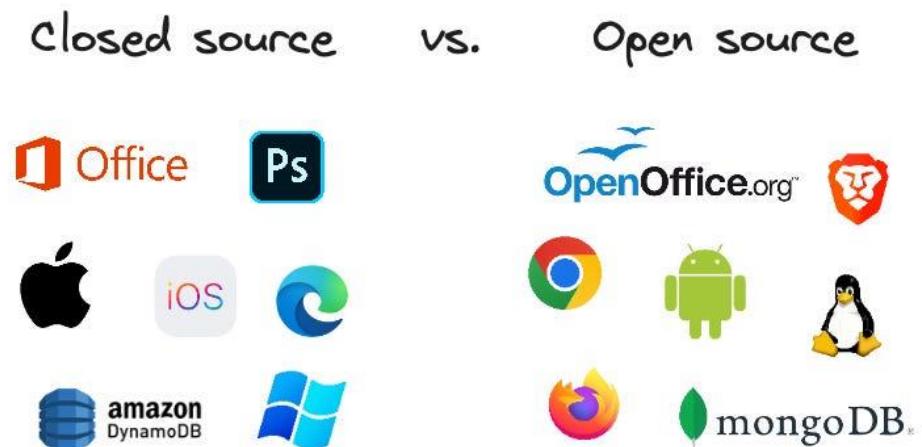
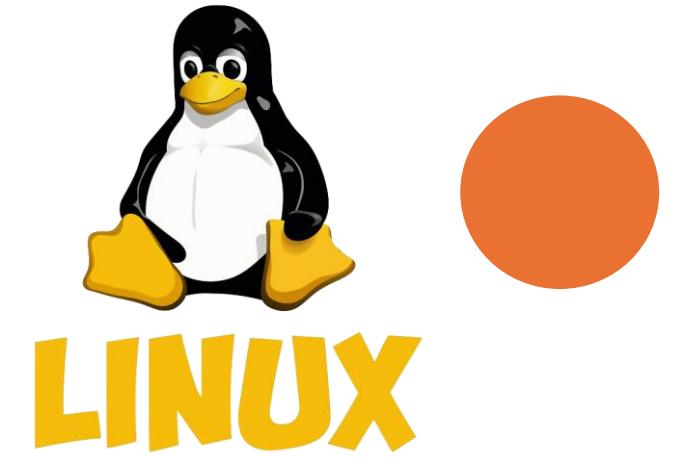


closed source vs. Open source



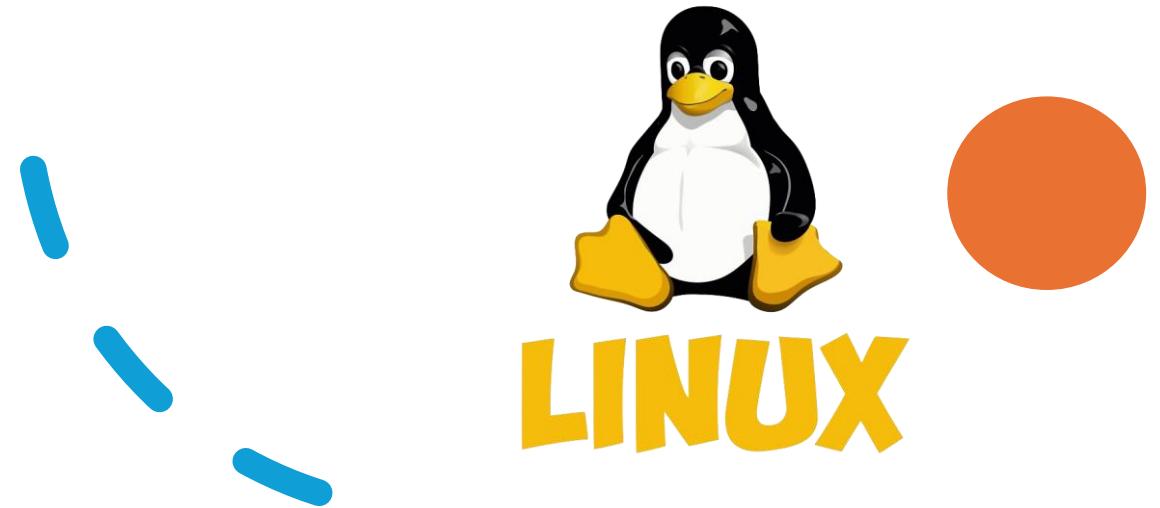
เครื่องมือการจัดการพาร์ติชัน: fdisk และ parted

- fdisk:
 - ใช้สร้าง, ลบ, และจัดการพาร์ติชันบนดิสก์ (เมกะสำหรับ MBR)
 - คำสั่งหลัก:
 - fdisk /dev/sdX เพื่อเลือกดิสก์
 - ใช้คำสั่ง n เพื่อสร้างพาร์ติชันใหม่, d เพื่อลบพาร์ติชัน
- parted:
 - ใช้จัดการพาร์ติชันที่มีขนาดใหญ่กว่า 2TB และรองรับ GPT
 - คำสั่งหลัก:
 - parted /dev/sdX เพื่อเข้าสู่ parted shell
 - คำสั่ง mkpart เพื่อสร้างพาร์ติชันใหม่



ขั้นตอนการสร้างและจัดการพาร์ติชัน

- ขั้นตอนการใช้ fdisk:
 - ใช้คำสั่ง fdisk /dev/sdX เพื่อเลือกดิสก์
 - กด n เพื่อสร้างพาร์ติชันใหม่
 - เลือกชนิดของพาร์ติชัน (Primary/Extended)
 - ระบุขนาดพาร์ติชัน
 - กด w เพื่อบันทึกการเปลี่ยนแปลง
- ขั้นตอนการใช้ parted:
 - ใช้คำสั่ง parted /dev/sdX
 - ใช้คำสั่ง mkpart เพื่อสร้างพาร์ติชัน
 - ระบุระบบไฟล์ เช่น ext4, swap
 - ใช้คำสั่ง print เพื่อดูรายละเอียดพาร์ติชัน



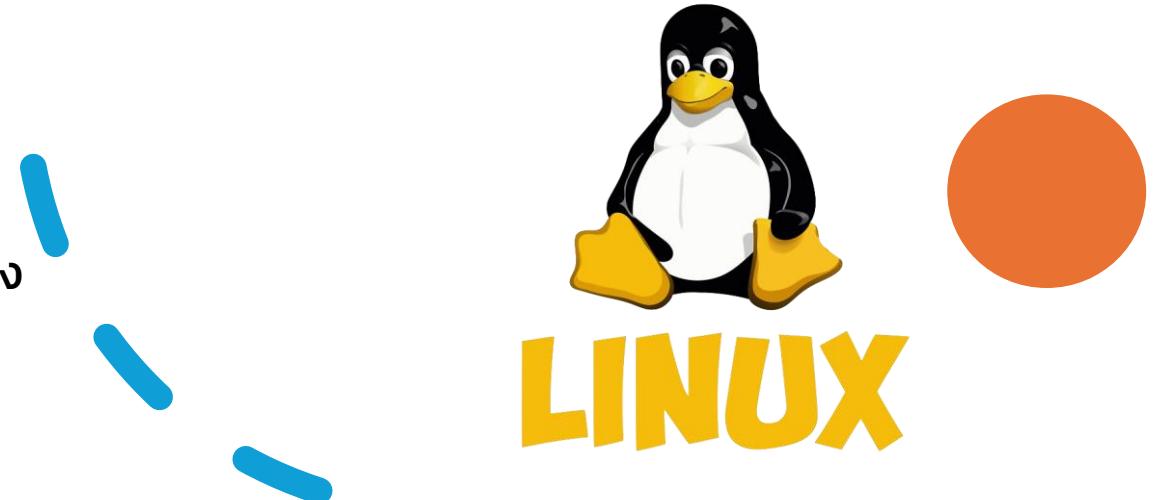
closed source vs. Open source



เครื่องมือในการตรวจสอบและควบคุมระบบ (Monitor and Control System Tools)

การตรวจสอบและจัดการกระบวนการในระบบ Linux

- top: ใช้ตรวจสอบการทำงานของทรัพยากรของระบบแบบเรียลไทม์ เช่น CPU, RAM, และกระบวนการที่ใช้งานอยู่
- ps: แสดงรายชื่อกระบวนการที่ทำงานอยู่ในขณะนี้
- kill: ใช้เพื่อหยุดกระบวนการที่ไม่ต้องการ โดยระบุ PID ของกระบวนการนั้น

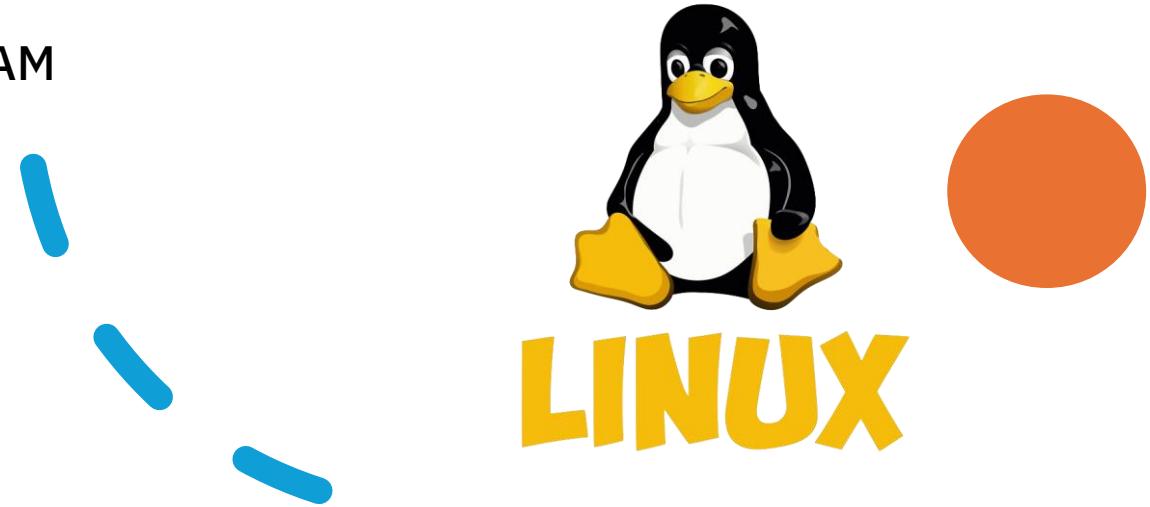


closed source vs. Open source



การใช้คำสั่ง top เพื่อการตรวจสอบระบบ

- คำสั่ง top แสดงข้อมูลสำคัญ เช่น การใช้งาน CPU และ RAM แบบเรียลไทม์
- การเลื่อนดูโปรแกรมต่าง ๆ และจัดลำดับตามการใช้งาน ทรัพยากร
- ตัวเลือก:
 - กด q เพื่อออกจาก top
 - กด k เพื่อยุดโปรแกรมที่ไม่ต้องการ (ต้องใส่ PID)

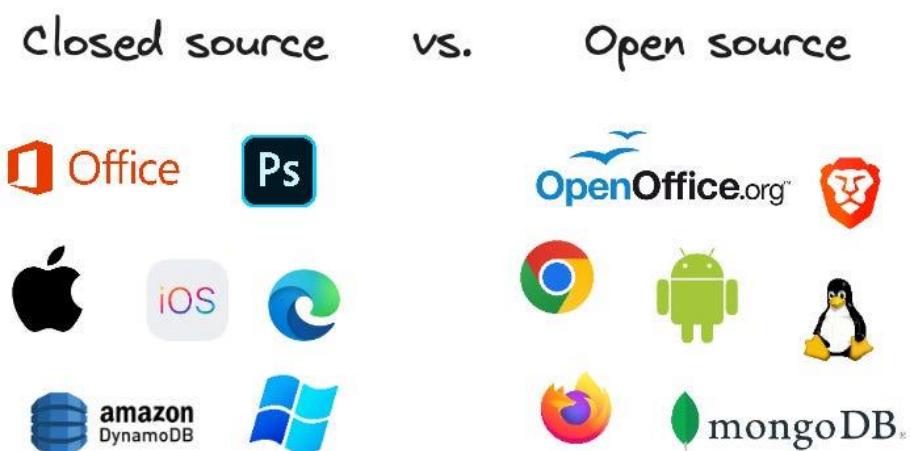
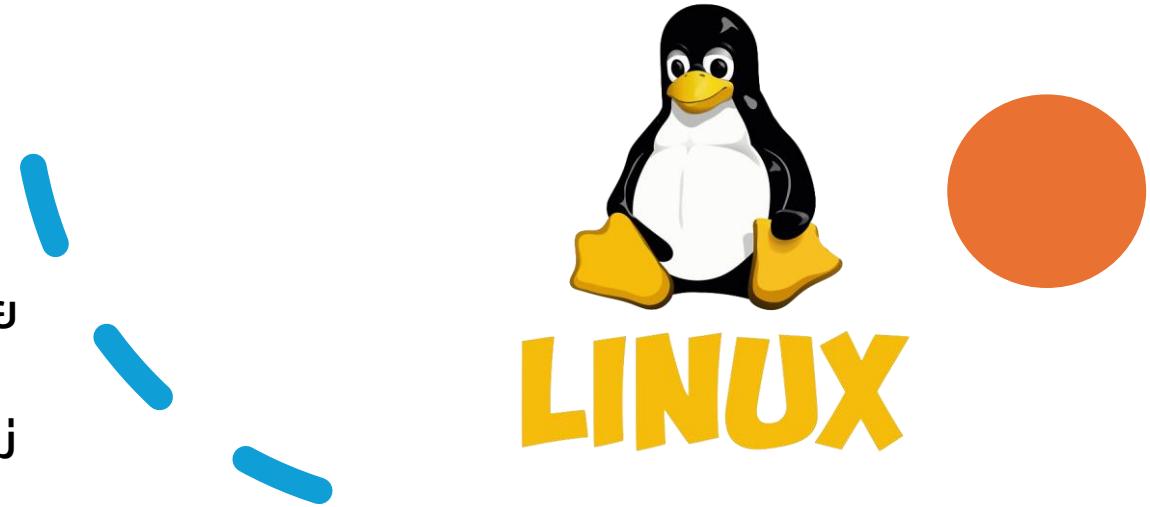


closed source vs. Open source



การใช้คำสั่ง ps และ kill เพื่อจัดการโปรเซส

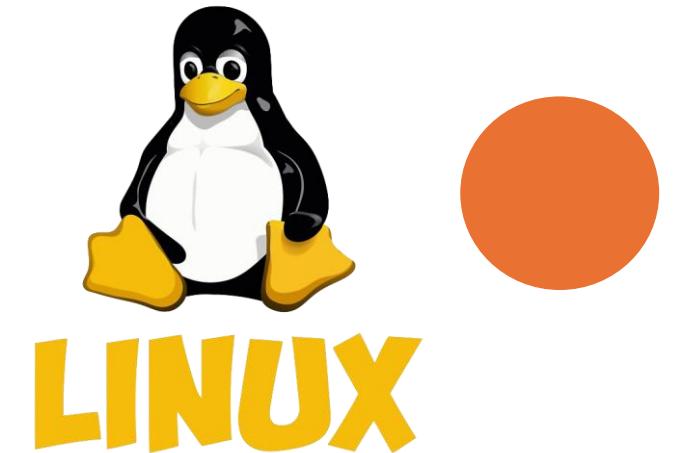
- ps: ใช้แสดงรายชื่อกระบวนการ ตัวเลือก ps aux จะแสดงกระบวนการทั้งหมดที่รันอยู่
- kill: ใช้หยุดโปรเซสที่ไม่ต้องการ โดยระบุ PID ของกระบวนการที่ต้องการหยุด
 - ตัวอย่างคำสั่ง kill -9 <PID> ใช้หยุดกระบวนการด้วยการบังคับ (force)
- การระบุ PID โดยใช้ ps และการใช้ kill เพื่อยุติโปรเซสที่ไม่ตอบสนอง



การตรวจสอบไฟล์ Log ของระบบและบริการ (System and Service Log Files)

ประเภทและโครงสร้างของ Log Files ใน Linux

- ไฟล์ Log ใน Linux มีบทบาทสำคัญในการบันทึกเหตุการณ์ต่างๆ ในระบบ ช่วยให้ผู้ดูและระบบสามารถตรวจสอบและแก้ไขปัญหาได้
- ตัวอย่าง Log Files ที่สำคัญ:
 - /var/log/syslog: บันทึกข้อมูลกั่งไปเกี่ยวกับระบบ
 - /var/log/auth.log: บันทึกข้อมูลการเข้าสู่ระบบและการใช้งานสิทธิ์ของผู้ใช้ (เฉพาะใน Debian-based)
 - /var/log/messages: บันทึกการทำงานของระบบ (สำหรับ Red Hat-based)

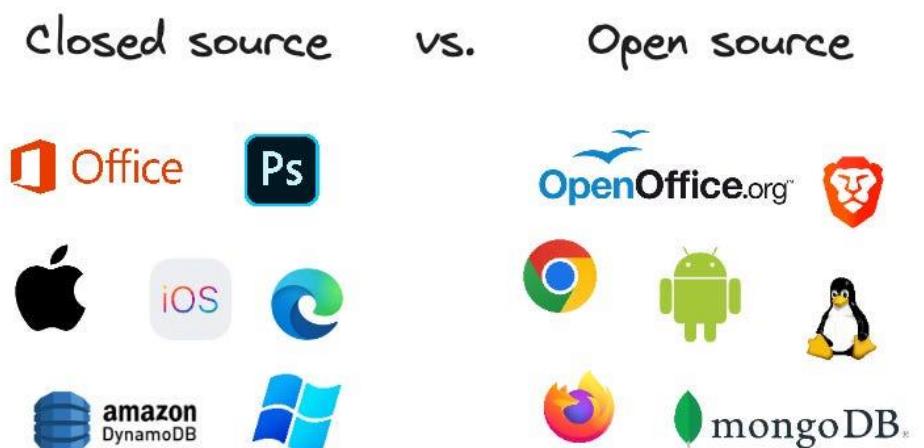
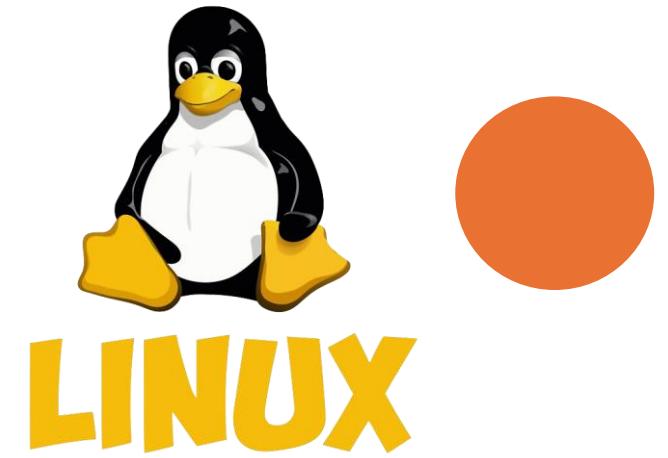


closed source vs. Open source



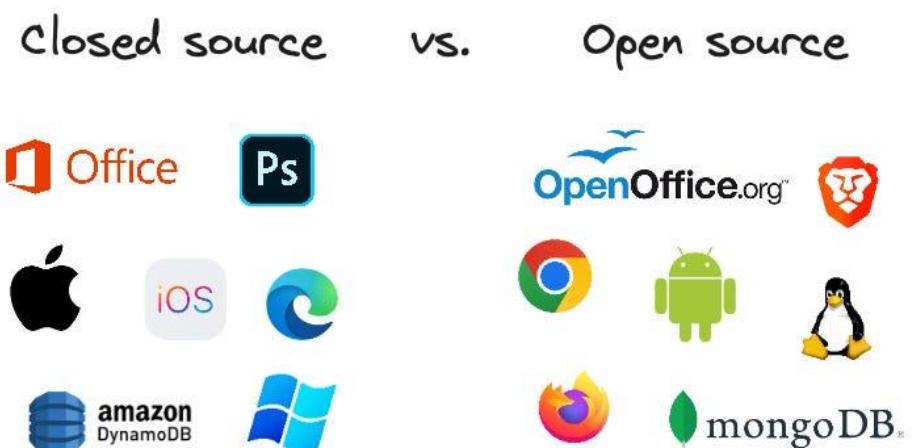
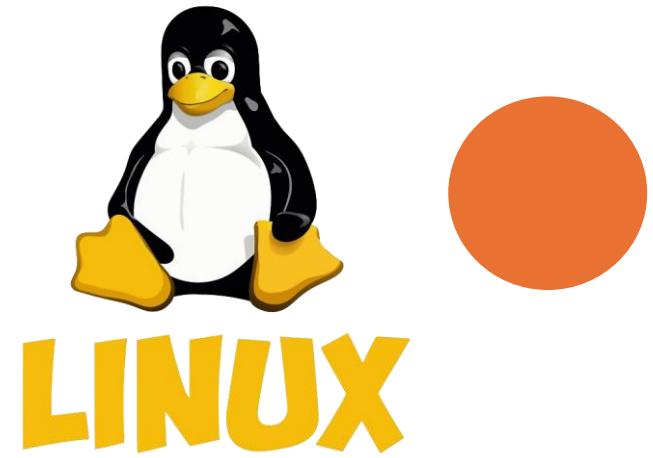
คำสั่งพื้นฐานในการตรวจสอบ Log Files

- คำสั่ง tail ใช้เพื่อดูบรรทัดท้าย ๆ ของไฟล์ Log แบบเรียลไทม์ (tail -f <filename>)
- คำสั่ง grep ใช้กรองข้อมูลใน Log Files เพื่อค้นหาข้อความที่ต้องการ เช่น ข้อความที่มีคำว่า "error"
- ตัวอย่าง:
 - tail -f /var/log/syslog: ดูข้อมูลแบบเรียลไทม์
 - grep "error" /var/log/syslog: กรองข้อความที่มีคำว่า "error" จากไฟล์ Log



การอ่านและวิเคราะห์ข้อมูลใน Log Files

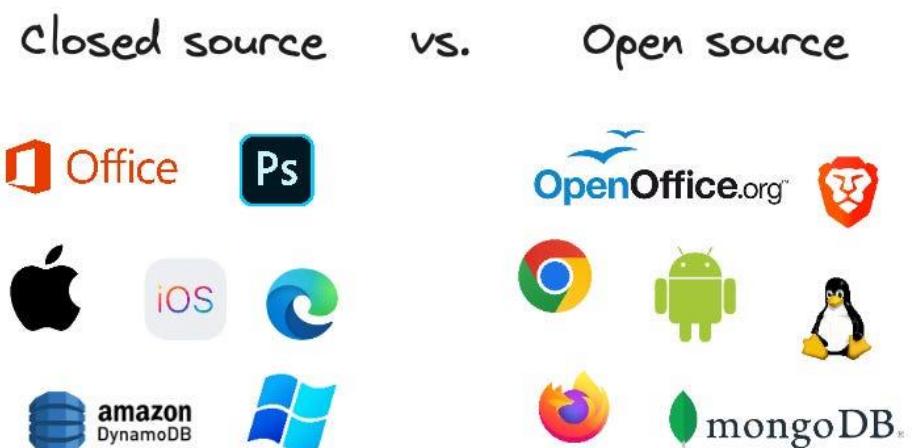
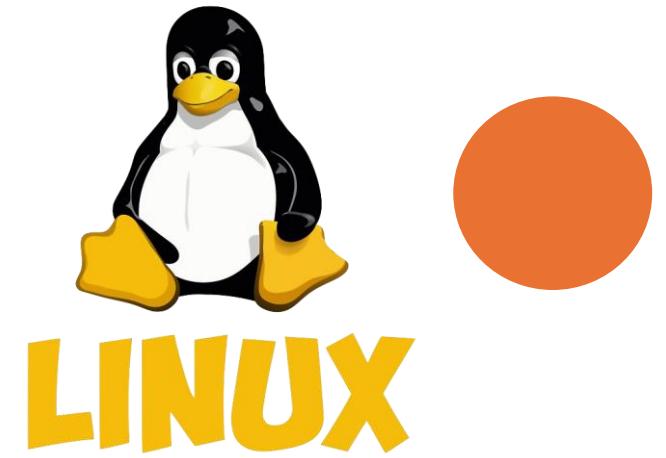
- การตรวจสอบ Log Files ช่วยให้เห็นภาพรวมของเหตุการณ์ เช่น การเชื่อมต่อ, การเข้าสู่ระบบ, หรือปัญหาต่าง ๆ ที่เกิดขึ้น
- ตัวอย่างเหตุการณ์:
- การแจ้งเตือนเมื่อมีการใช้งาน root หรือการพยายามเข้าสู่ระบบที่ผิดพลาด
- การตรวจสอบปัญหาเครือข่ายหรือการทำงานของบริการที่หยุดชะงัก



การติดตั้ง Fedora

การติดตั้ง Fedora

1. ขั้นตอนการติดตั้ง Fedora บน Virtual Machine (VM) หรือเครื่องจริง
2. การตั้งค่าพื้นฐานในระหว่างการติดตั้ง
3. การตั้งค่าหลังติดตั้ง Fedora



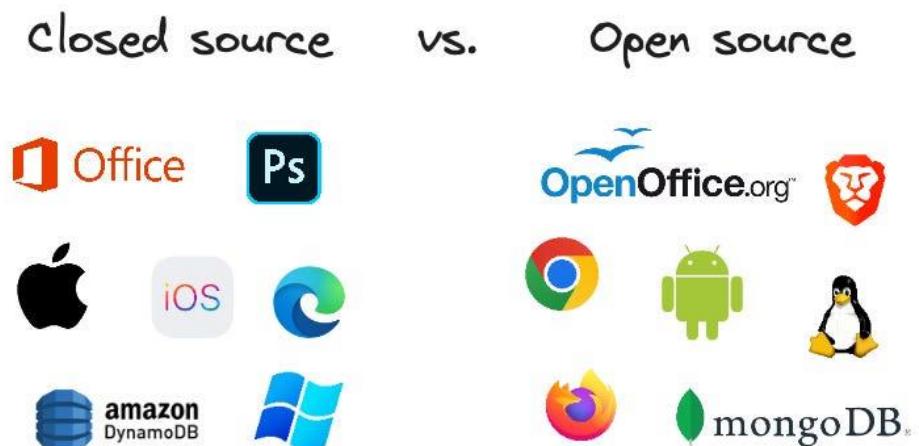
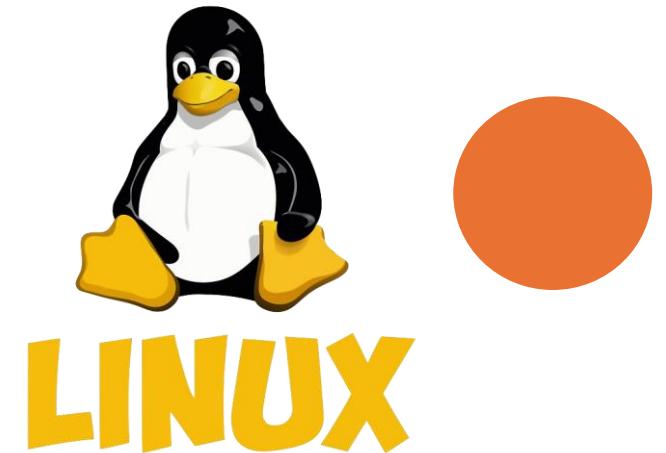
สรุปและ Q&A

Day 3: การจัดการระบบขั้นสูงและบริการ ใน Linux

ความเข้าใจพื้นฐานเกี่ยวกับ RAID และ Linux RAID Subsystems

พื้นฐานเกี่ยวกับ RAID (Redundant Array of Independent Disks)

- RAID คือการรวมดิสก์หลายตัวเข้าด้วยกันเพื่อเพิ่มความเร็ว (Performance) หรือเพิ่มความปลอดภัยของข้อมูล (Redundancy)
- ประเภทของ RAID:
 - RAID 0: รวมดิสก์หลายตัวเข้าด้วยกันเพื่อเพิ่มความเร็ว (ไม่มีการสำรองข้อมูล)
 - RAID 1: สำรองข้อมูลด้วยการทำ Mirror ระหว่างดิสก์สองตัว
 - RAID 5: ใช้ดิสก์หลายตัวร่วมกัน พร้อมความสามารถในการสำรองข้อมูลหากดิสก์ตัวใดตัวหนึ่งเสีย (ต้องใช้ดิสก์อย่างน้อย 3 ตัว)
 - RAID 10: ผสมระหว่าง RAID 1 และ RAID 0 เพื่อให้ได้ทั้งความเร็วและความทนทาน (ต้องใช้ดิสก์อย่างน้อย 4 ตัว)



การใช้งาน Software RAID บน Linux และเครื่องมือ mdadm

- Software RAID: จัดการ RAID โดยไม่ต้องใช้ฮาร์ดแวร์พิเศษ เพียงแค่ติดตั้ง Software RAID ผ่านระบบปฏิบัติการ
- เครื่องมือ mdadm:
 - ใช้ในการสร้าง, จัดการ, และตรวจสอบ Software RAID บน Linux
 - คำสั่งพื้นฐาน:

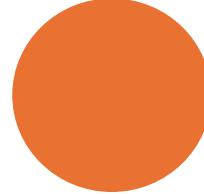
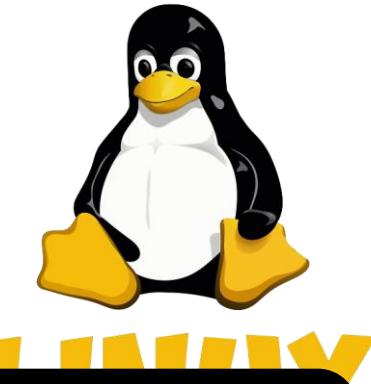
Bash

```
mdadm --create --verbose /dev/md0 --level=<level> --raid-devices=<num>
<devices>
```

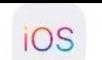
- ตัวอย่างสร้าง RAID 1:

Bash

```
mdadm --create --verbose /dev/md0 --level=1 --raid-devices=2 /dev/sda1
/dev/sdb1
```



Open source



กิจกรรมปฏิบัติการ - การสร้างและจัดการ RAID ด้วย mdadm

- การตั้งค่า RAID ด้วยคำสั่ง mdadm:
- ตัวอย่างขั้นตอน:
 - ตรวจสอบอุปกรณ์ที่ใช้สร้าง RAID ด้วยคำสั่ง lsblk
 - สร้าง RAID 1 :

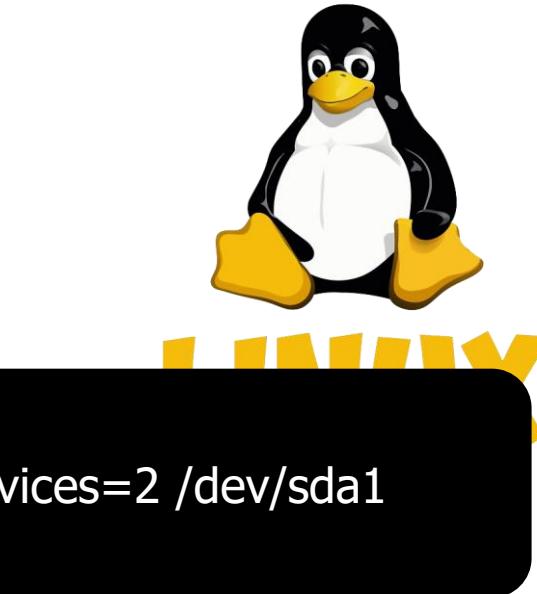
Bash

```
mdadm --create --verbose /dev/md0 --level=1 --raid-devices=2 /dev/sda1  
/dev/sdb1
```

- ตรวจสอบสถานะ RAID :

Bash

```
cat /proc/mdstat
```



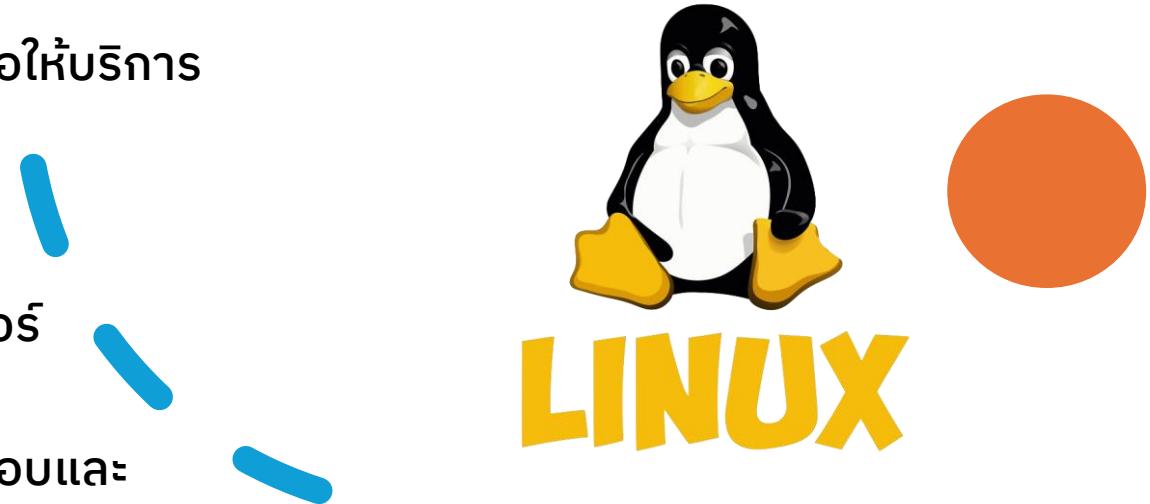
closed source vs. Open source



บทบาทและงานของ System Administrator

บทบาทของ System Administrator

- System Administrator: ดูแลและจัดการระบบ Linux เพื่อให้บริการทำงานอย่างราบรื่น
- บทบาทสำคัญ:
 - ดูแลรักษาความเสถียรของระบบ
 - ตรวจสอบการทำงานและความปลอดภัยของเซิร์ฟเวอร์
 - จัดการผู้ใช้และกำหนดสิทธิ์การเข้าถึง
- ความสำคัญของการทำงานในเชิงป้องกัน เช่น การตรวจสอบและปรับปรุงระบบให้กันสมัยอยู่เสมอ

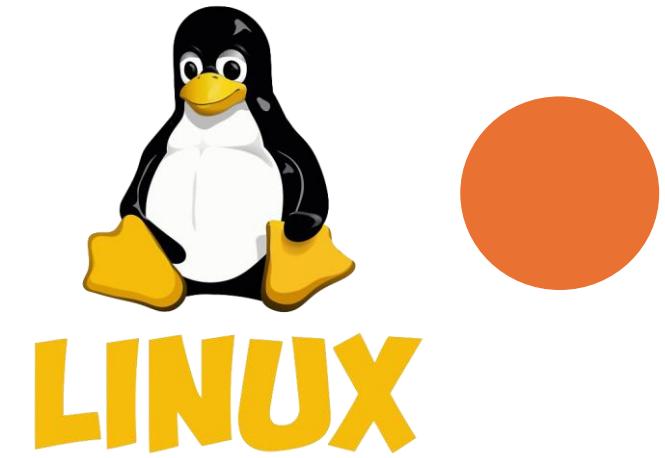


closed source vs. Open source



งานประจำวันของ System Administrator

- งานที่ทำเป็นประจำ:
 - ตรวจสอบสถานะระบบ: ใช้เครื่องมือต่างๆ เช่น top, df, ps เพื่อตรวจสอบการใช้งาน CPU, หน่วยความจำ และพื้นที่ดิสก์
 - การอัปเดตซอฟต์แวร์: ติดตั้งและอัปเดตแพ็กเกจต่างๆ เพื่อรักษาความปลอดภัย
 - การสำรองข้อมูล: วางแผนและจัดการสำรองข้อมูลอย่างสม่ำเสมอ
 - ดูแลความปลอดภัย: ตั้งค่ากำหนดไฟร์วอลล์และตรวจสอบ log เพื่อป้องกันภัยคุกคามทางไซเบอร์

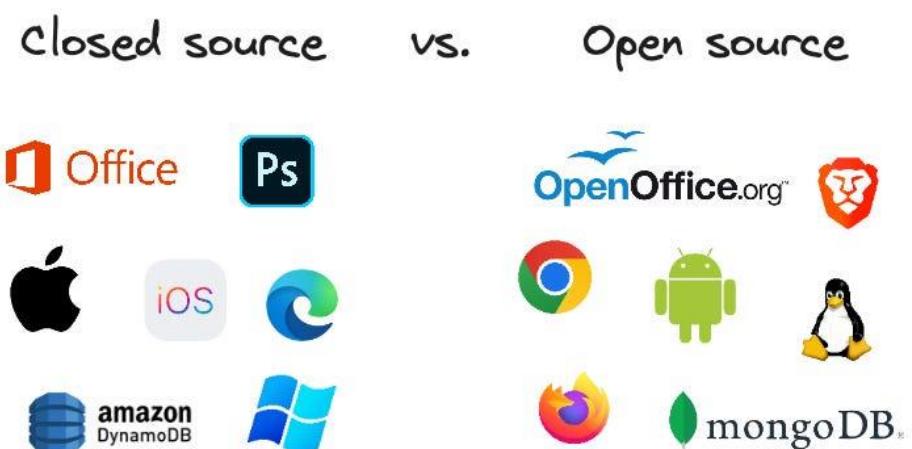
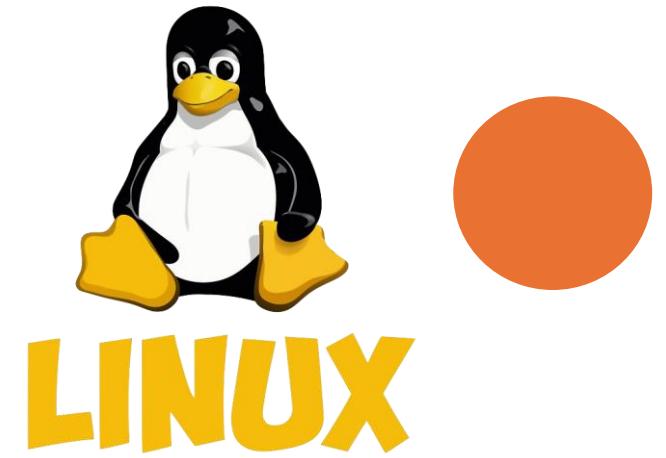


closed source vs. Open source



การแก้ไขปัญหาและการกู้คืนข้อมูล

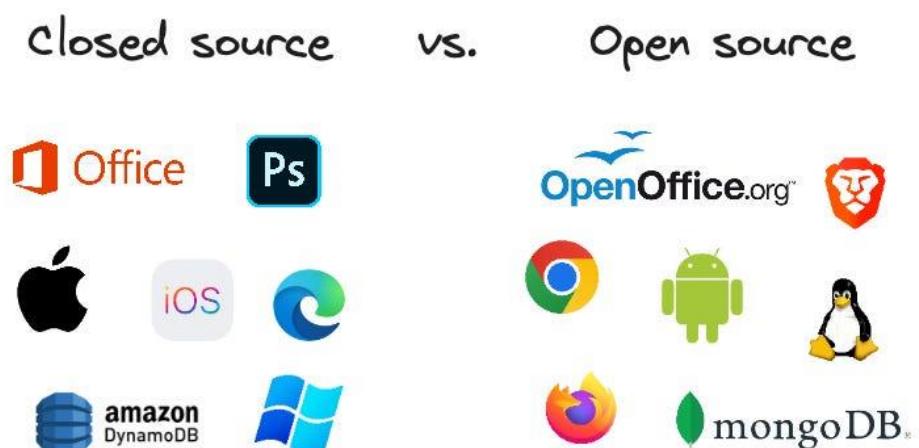
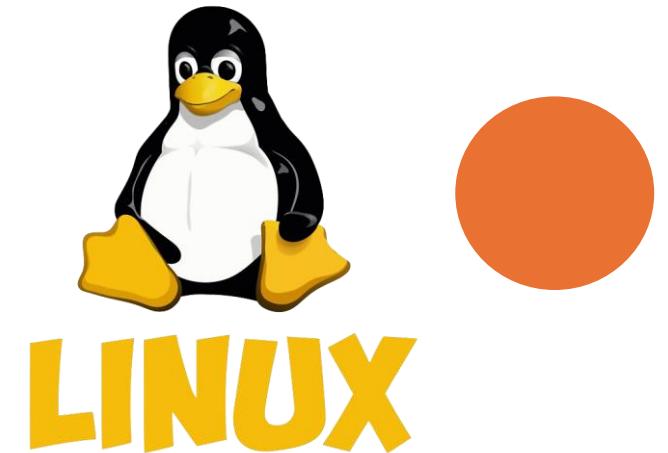
- การแก้ไขปัญหา:
 - แก้ไขปัญหาที่เกิดขึ้นในระบบ เช่น การเชื่อมต่อเครือข่ายขัดข้องหรือบริการที่หยุดทำงาน
 - ใช้คำสั่ง grep, tail, journalctl ในการตรวจสอบ log และวินิจฉัยปัญหา
- การสำรองและกู้คืนข้อมูล:
 - จัดการและทดสอบการสำรองข้อมูล (Backup) อย่างสม่ำเสมอ
 - มีแผนการกู้คืนข้อมูล (Disaster Recovery) เมื่อเกิดความเสียหาย



การจัดการเครือข่ายเบื้องต้นด้วยคำสั่ง TCP/IP (Basic TCP/IP Networking Commands)

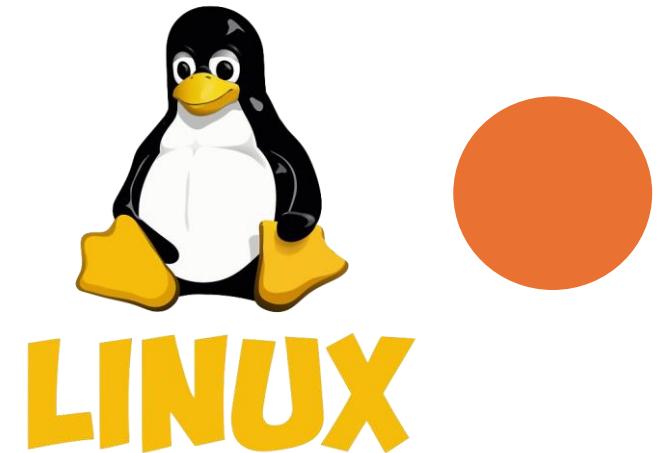
การใช้งานคำสั่งเครือข่ายพื้นฐานใน Linux

- คำสั่งเครือข่ายพื้นฐาน:
 - ifconfig - แสดงข้อมูลของ network interfaces เช่น IP address และสถานะการเชื่อมต่อ
 - netstat -tuln - แสดงสถานะของพอร์ตและบริการที่กำลังเปิดใช้งาน
 - route -n - ดู routing table เพื่อแสดงเส้นทางที่ข้อมูลเครือข่ายใช้
- การตั้งค่า IP Address และตรวจสอบสถานะเครือข่าย:
 - ตั้งค่า IP Address ชั่วคราว: ifconfig eth0 192.168.1.10
 - netmask 255.255.255.0



การตรวจสอบการเชื่อมต่อเครือข่ายด้วยคำสั่ง TCP/IP

- ตรวจสอบการเชื่อมต่อ:
 - ping [hostname/IP] - ทดสอบการเชื่อมต่อไปยังโฮสต์เป้าหมาย
 - traceroute [hostname/IP] - ตรวจสอบเส้นทางที่ข้อมูลใช้ในการเดินทางไปยังโฮสต์เป้าหมาย
- ตรวจสอบการเชื่อมต่อเครือข่าย:
 - netcat หรือ nc - ใช้ตรวจสอบการเชื่อมต่อ TCP/UDP ports



Network Configuration Files

By Sommanut Ketpong

Network Configuration Files

- ไฟล์สำคัญ:

- /etc/hosts:** ใช้สำหรับการแมป IP address กับ hostname
 - ตัวอย่าง:

```
Bash
```

```
127.0.0.1 localhost  
192.168.1.100 myserver
```

- /etc/hostname:** ใช้กำหนดชื่อเครื่อง (hostname) ของระบบ
 - ตัวอย่าง:

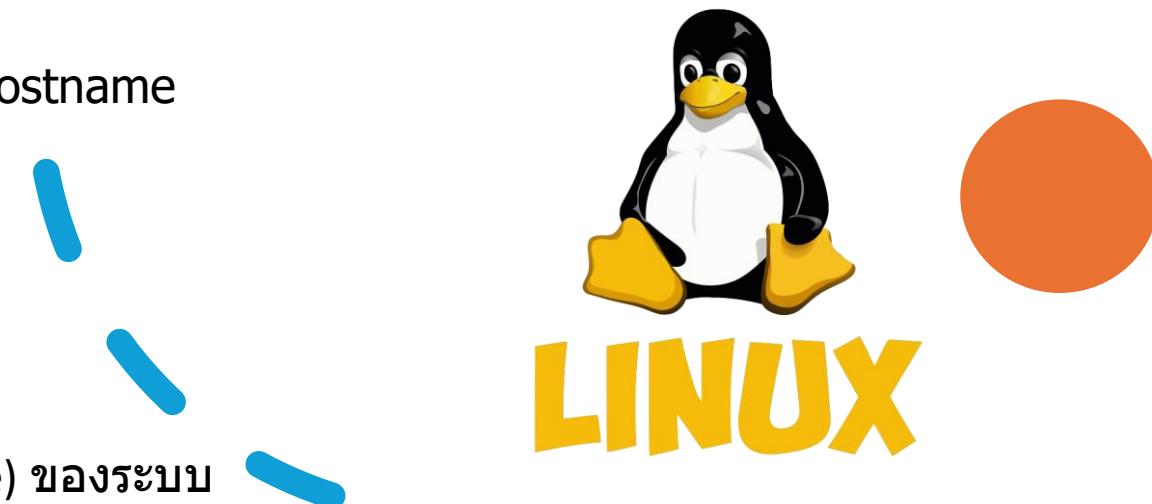
```
Bash
```

```
myserver
```

- /etc/nsswitch.conf:** กำหนดวิธีการค้นหาชื่อ (name resolution) เช่น การค้นหาจาก hosts, DNS
 - ตัวอย่าง:

```
Bash
```

```
hosts: files dns
```



closed source vs. Open source



Network Configuration Files

- การแก้ไขไฟล์:
 - ใช้โปรแกรมแก้ไขไฟล์เช่น nano, vi, หรือ vim
 - คำสั่งสำหรับเปิดไฟล์:
 - แก้ไข /etc/hosts:

Bash

```
sudo nano /etc/hosts
```

- แก้ไข /etc/hostname :

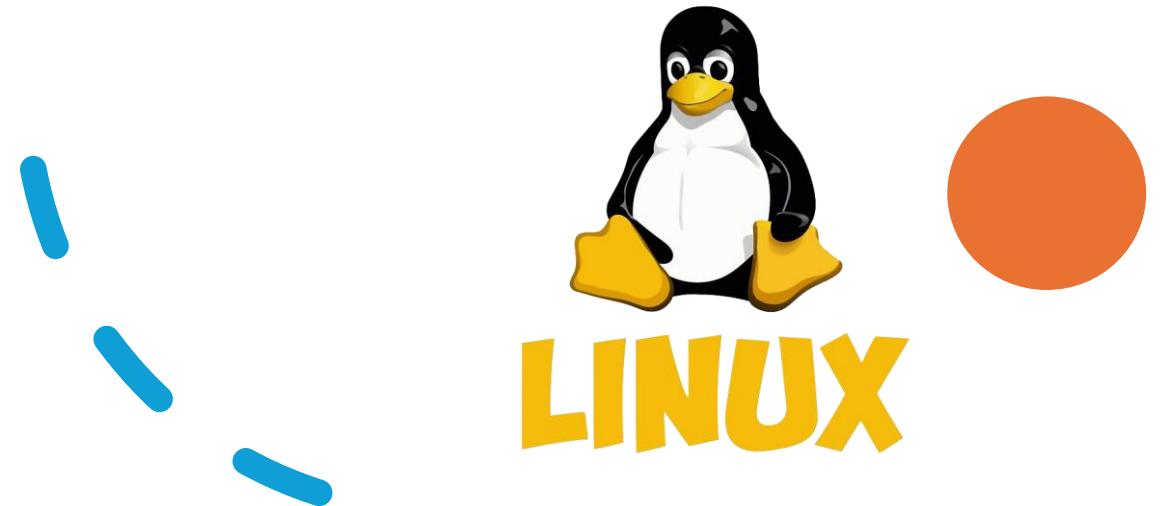
Bash

```
sudo nano /etc/hostname
```

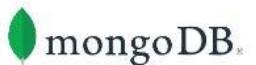
- แก้ไข /etc/nsswitch.conf :

Bash

```
sudo nano /etc/nsswitch.conf
```



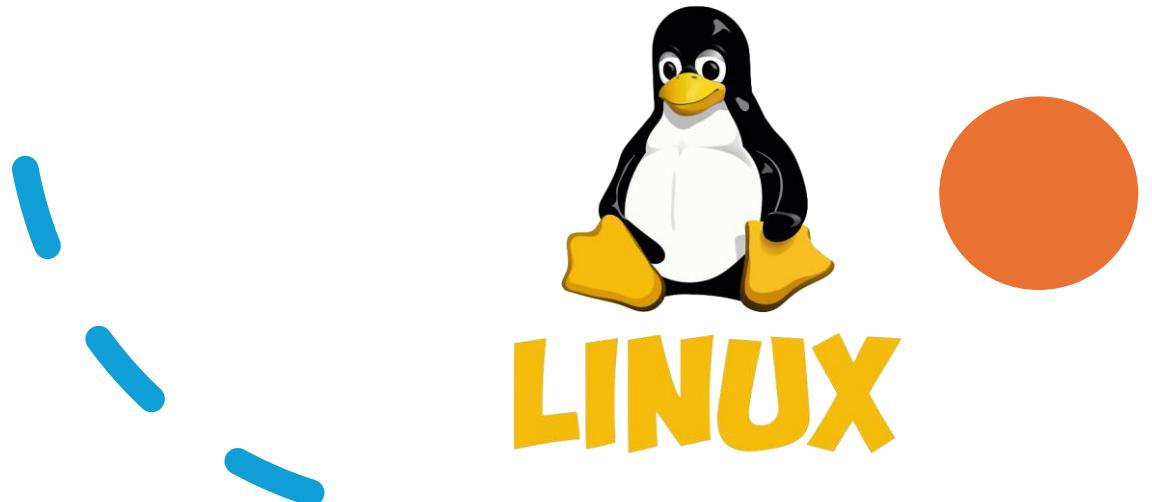
closed source vs. Open source



การอัปเดตซอฟต์แวร์และการจัดการแพ็คเกจ (Software Updates and Package Management)

การจัดการแพ็คเกจใน Linux

- เครื่องมือจัดการแพ็คเกจ:
- YUM (Yellowdog Updater, Modified): ใช้สำหรับ Red Hat, CentOS
- APT (Advanced Package Tool): ใช้สำหรับ Ubuntu, Debian



closed source vs. Open source



การอัปเดตซอฟต์แวร์และแพ็กเกจ

- เพื่อรักษาความปลอดภัยและเสถียรภาพของระบบ
 - คำสั่ง:
 - สำหรับ YUM:

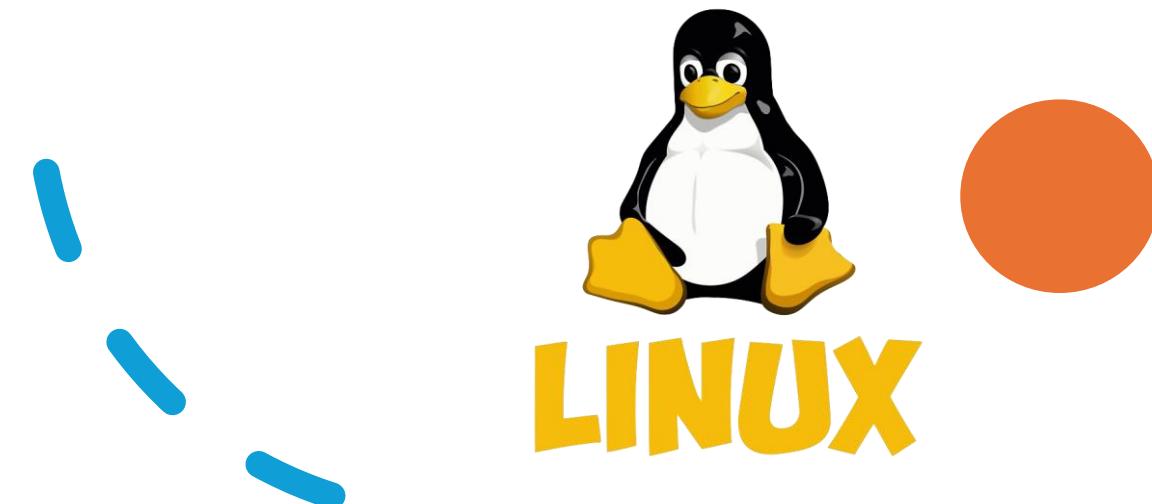
Bash

```
sudo yum update
```

- สำหรับ APT:

Bash

```
sudo apt update  
sudo apt upgrade
```



closed source

vs.

Open source



การอัปเดตซอฟต์แวร์และแพ็กเกจ

- การค้นหา ติดตั้ง อัปเดต และลบแพ็กเกจ

- คำสั่ง:

- สำหรับ YUM:

```
Bash
```

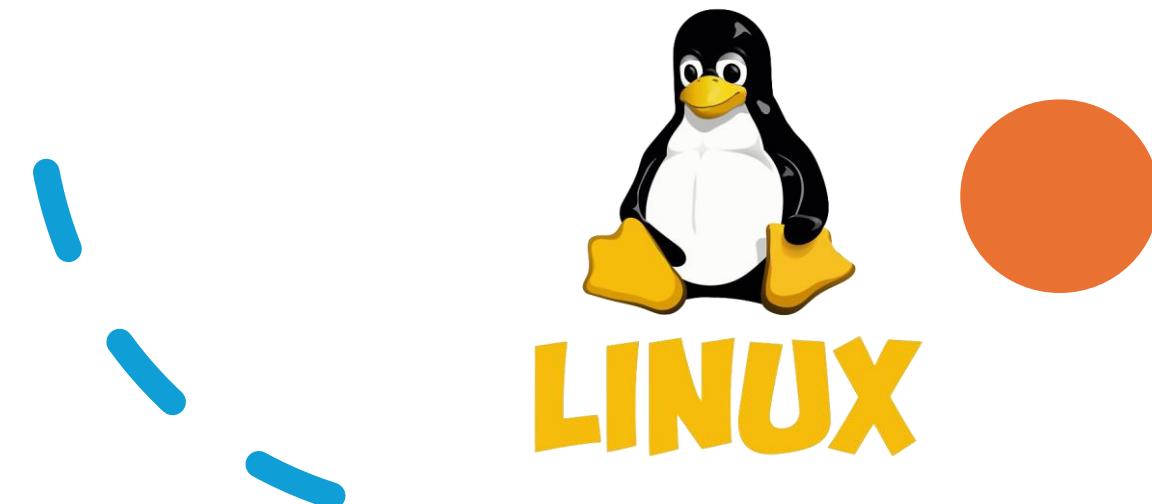
```
sudo yum install <package_name>
```

- สำหรับ APT:

```
Bash
```

```
sudo apt-get install <package_name>
```

```
sudo apt-get remove <package_name>
```



closed source

vs.

Open source

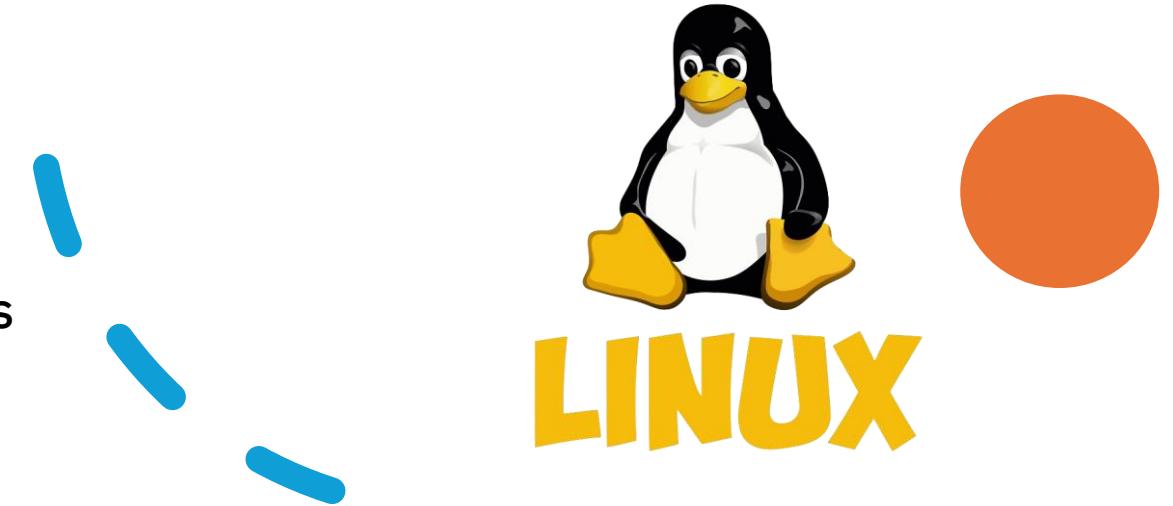


Remote Access to Linux

By Sommanut Ketpong

การจัดการแพ็กเกจใน Linux

- การใช้งาน Telnet และ SSH:
 - Telnet: การเชื่อมต่อที่ไม่ปลอดภัย ใช้โปรโตคอลที่ไม่เข้ารหัส
 - ไม่แนะนำให้ใช้งานในระบบที่มีความปลอดภัยสูง
 - SSH (Secure Shell): การเชื่อมต่อที่ปลอดภัย ใช้การเข้ารหัส
 - รองรับการรับส่งข้อมูลที่ปลอดภัย



closed source vs. Open source



การตั้งค่า SSH

- การตั้งค่า SSH:
 - เปิดใช้งาน SSH โดยการติดตั้งแพ็กเกจ OpenSSH Server:

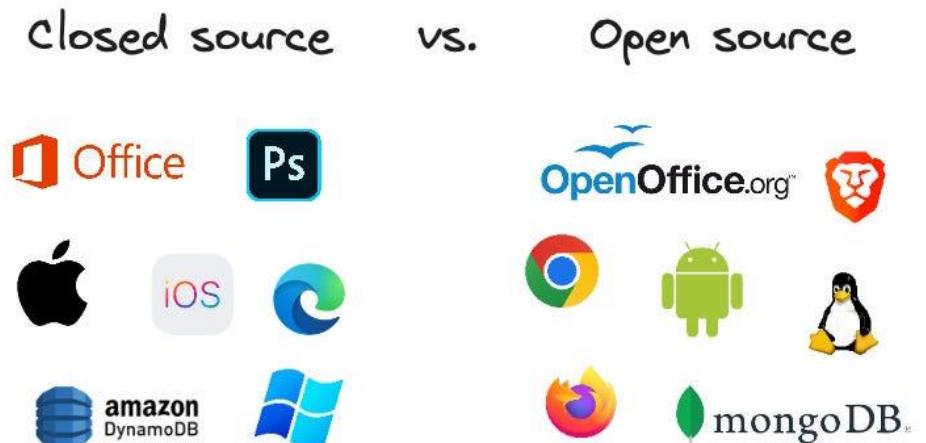
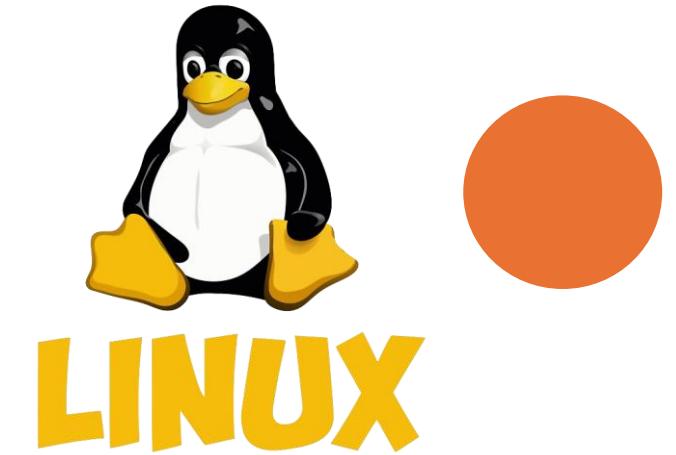
Bash

```
sudo yum install openssh-server # สำหรับ Red Hat/CentOS  
sudo apt-get install openssh-server # สำหรับ Ubuntu/Debian
```

- การกำหนดค่า SSH:
 - ไฟล์กำหนดค่า: /etc/ssh/sshd_config
 - ตั้งค่าพอร์ต, การอนุญาตผู้ใช้, และการปิดการเข้าถึง root
 - ตัวอย่างคำสั่ง:

Bash

```
sudo nano /etc/ssh/sshd_config
```



สรุปและ Q&A

Thank you