



01-Network-Interface-Configuration

Network Tool and Package in LINUX

ใน Linux มีเครื่องมือและแพ็คเกจเสริมที่ช่วยในการจัดการ Network Interface เพิ่มเติม เช่น `ifconfig`, `ip`, `netplan`, `NetworkManager`, และอื่น ๆ อีกมากมายที่ช่วยให้การจัดการเครือข่ายในระบบ Linux นั้นง่ายขึ้นและมีประสิทธิภาพมากยิ่งขึ้น

Net-tools Package

แพ็คเกจนี้ประกอบด้วยเครื่องมือที่สำคัญสำหรับการควบคุม subsystem ของเครือข่ายของเคอร์เนล Linux

- ประกอบด้วยการตั้งค่าพื้นฐาน `arp`, `ifconfig`, `netstat`, `rarp`, `nameif` และ `route`
- นอกจากนี้ แพ็คเกจนี้ยังประกอบด้วยยูทิลิตี้ที่เกี่ยวข้องกับฮาร์ดแวร์เครือข่ายบางประเภท `plipconfig`, `slattach`, `mii-tool`
- แก้ไขคุณสมบัติขั้นสูงของการกำหนดค่า IP ได้จาก `iptunnel`, `ipmaddr`

```
$ sudo apt install net-tools // สำหรับการติดตั้ง package นี้
```

ประกอบด้วยคำสั่งดังนี้

Command	Description
<code>arp</code>	ใช้กับโมดูลเคอร์เนล Linux ARP จัดการแคช ARP ของระบบ
<code>rarp</code>	ใช้เพื่อจัดการตาราง RARP ของเคอร์เนล
<code>ifconfig</code>	กำหนดค่าอินเทอร์เฟซเครือข่าย
<code>ipmaddr</code>	เพิ่ม ลบ และแสดง multicast addresses ของอินเทอร์เฟซ
<code>iptunnel</code>	เพิ่ม เปลี่ยนแปลง ลบ และแสดง tunnels ของอินเทอร์เฟซ
<code>mii-tool</code>	ตรวจสอบหรือตั้งค่าสถานะของหน่วย Media Independent Interface (MII) ของอินเทอร์เฟซเครือข่าย
<code>nameif</code>	ตั้งชื่ออินเทอร์เฟซเครือข่ายตามที่อยู่ MAC

netstat	ใช้เพื่อรายงานการเชื่อมต่อเครือข่าย ตารางเส้นทาง และสถิติอินเทอร์เฟซ
route	ใช้เพื่อจัดการตารางเส้นทางของ IP (Routing Table)
slattach	แนบอินเทอร์เฟซเครือข่ายเข้ากับสายอนุกรม ซึ่งจะช่วยให้คุณสามารถใช้รอกัดเทอร์มินัลปกติสำหรับลิงก์แบบ point-to-point ไปยังคอมพิวเตอร์เครื่องอื่นได้
plipconfig	ใช้เพื่อปรับแต่งพารามิเตอร์อุปกรณ์ PLIP อย่างละเอียดเพื่อปรับปรุงประสิทธิภาพ

ข้อดีและข้อเสียของการใช้ net-tools

ข้อดี	ข้อเสีย
ทำความเข้าใจได้ง่ายต่อการใช้งาน	ไม่สนับสนุนในระบบเวอร์ชันล่าสุด
สนับสนุนเวอร์ชันเก่าของ Linux	ถูกแทนที่ด้วยคำสั่ง ip ที่มีประสิทธิภาพมากกว่า

การใช้ net-tools ยังมีข้อดีในเชิงความคุ้นเคยและการเข้าถึงง่าย แต่ควรระวังว่าอาจไม่มีคุณสมบัติหรือการรองรับที่เทียบเท่ากับเครื่องมือที่ใช้สำหรับการจัดการเครือข่ายบนระบบ Linux ที่ใหม่กว่านี้

Netplan

ระบบปฏิบัติการ Debian/Ubuntu มักใช้ **netplan** หรือ **ifupdown** เพื่อกำหนดค่าเครือข่าย โดยใช้ไฟล์ YAML หรือไฟล์ข้อความธรรมดาตามลำดับ ซึ่งจะเก็บอยู่ในไดเรกทอรี **/etc/netplan** โดยปกติจะมาพร้อมกับระบบของ ubuntu อยู่แล้ว แต่ถ้าจะใช้ในระบบปฏิบัติการอื่นๆ ควรติดตั้ง package ดังนี้

```
$ sudo apt update
$ sudo apt install netplan.io
```

Netplan อ่านการกำหนดค่าเครือข่ายจาก **/etc/netplan/*.yaml** ซึ่งเขียนโดยผู้ดูแลระบบ ระหว่างการบูตครั้งแรก Netplan จะสร้างไฟล์การกำหนดค่าเฉพาะแบ็กเอนด์ใน **/run** เพื่อส่งต่อการควบคุมอุปกรณ์ให้กับไดมอนด์เครือข่ายเฉพาะ ปัจจุบัน Netplan ทำงานร่วมกับตัวเรนเดอร์ที่รองรับเหล่านี้

- NetworkManager
- Systemd-networkd

ประกอบด้วยคำสั่งดังนี้

Command	Description
netplan generate	ใช้ /etc/netplan เพื่อสร้างการตั้งค่าที่จำเป็นสำหรับตัวเรนเดอร์ให้อยู่ในรูปแบบ .YAML file
netplan apply	ใช้การกำหนดค่าทั้งหมดสำหรับตัวเรนเดอร์ โดยรีสตาร์ทตามความจำเป็น
netplan try	ใช้การกำหนดค่าและรอการยืนยันจากผู้ใช้ จะย้อนกลับหากเครือข่ายเสียหายหรือไม่ได้รับการยืนยัน
netplan get	แสดงข้อมูลค่าที่อยู่ในไฟล์ /etc/netplan/*.yaml
ls nano cat /etc/netplan/*.yaml	list หรือแสดงข้อมูลค่าที่อยู่ในไฟล์ และสามารถแก้ไขได้ด้วย editor ที่ไฟล์ /etc/netplan/*.yaml

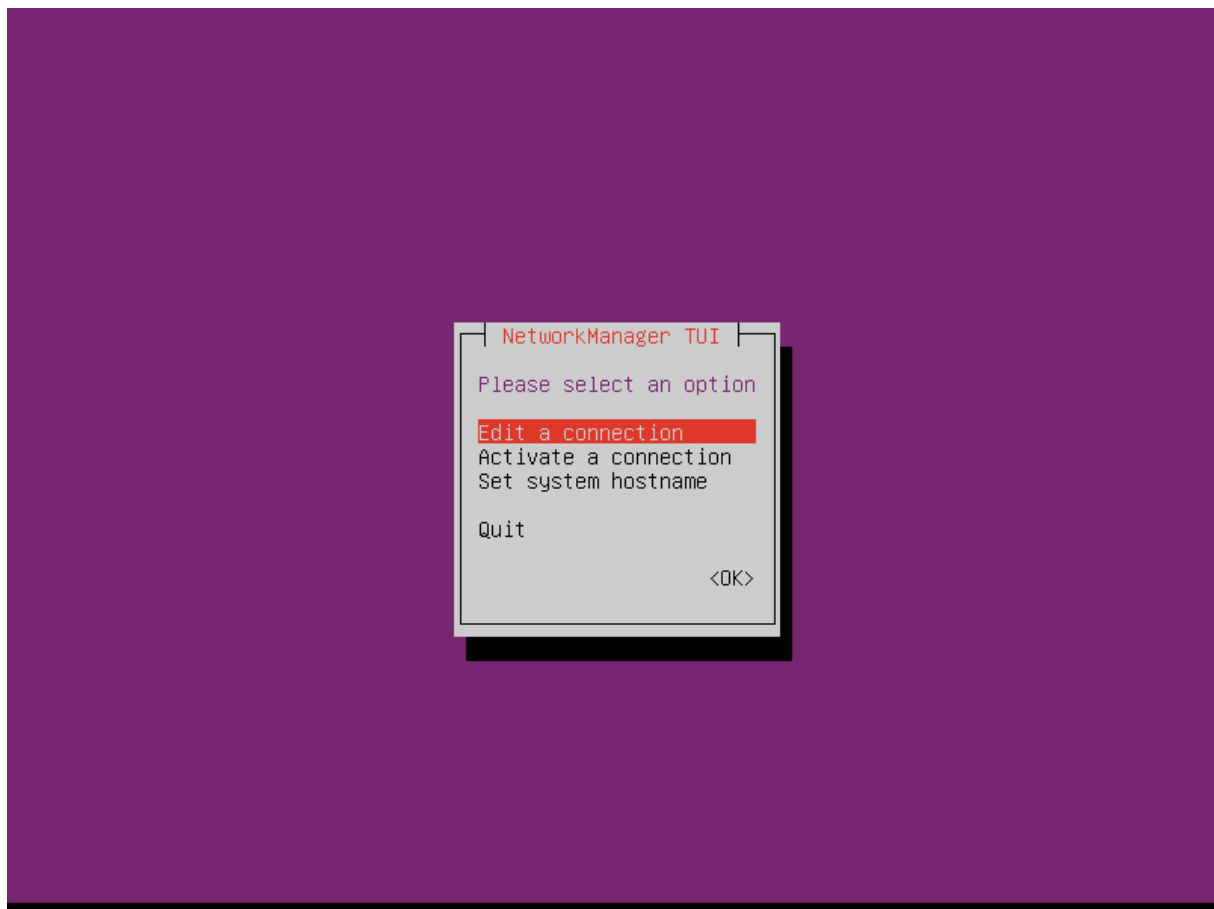
ข้อดีและข้อเสียของการใช้ netplan

ข้อดี	ข้อเสีย
ทำความเข้าใจได้ง่ายต่อการใช้งาน	ไม่สนับสนุนในระบบเวอร์ชันเก่า
รองรับระบบปฏิบัติการหลายรุ่น	ต้องมีความรู้ในการใช้ YAML
การจัดการแบบ Declarative	ข้อผิดพลาดในไฟล์ YAML ที่เกิดขึ้นได้
ยืดหยุ่นในการกำหนดค่า	ความซับซ้อนของการกำหนดค่า
การอัปเดตคอนฟิกง่าย	

Text User Interface for controlling Network Manager (nmtui)

nmtui เป็นแอปพลิเคชัน TUI ที่ใช้สำหรับการโต้ตอบกับ NetworkManager เมื่อเริ่มต้น nmtui ผู้ใช้จะได้รับพร้อมท์ให้เลือกกิจกรรมที่จะดำเนินการ เว้นแต่จะถูกระบุเป็นอาร์กิวเมนต์แรก ติดตั้งได้โดย package

```
// ติดตั้ง package
$ sudo apt update
$ sudo apt install network-manager
// เริ่มต้นใช้งาน
$ nmtui
```



[ภาพที่ 1] ใช้คำสั่ง nmtui จะเข้าสู่หน้า TUI ในการตั้งค่า

ข้อดีและข้อเสียของการใช้ NMTUI

ข้อดี	ข้อเสีย
ใช้งานได้เลยผ่านทาง Terminal	ไม่สามารถกำหนดค่าเครือข่ายที่ซับซ้อนได้
เป็นเครื่องมือที่เหมาะสมสำหรับผู้ที่ไม่ค่อยมีความชำนาญในการกำหนดค่าเครือข่ายผ่านทาง CLI	ข้อจำกัดในการแก้ไขปัญหา อาจทำให้เกิดความสับสน

Network Manager

Network Manager คือชุดเครื่องมือสำหรับกำหนดค่าอุปกรณ์เครือข่ายของเครื่อง Linux สิ่งสำคัญที่สุดคือมีการกำหนดค่าอัตโนมัติของอุปกรณ์เครือข่ายในเครื่อง Linux ผ่านบริการ Network Manager แน่นอนว่าเราสามารถ

กำหนดค่าด้วยตนเองได้หลายวิธี นอกจากนี้แพ็คเกจยังมีส่วนต่อประสานกับผู้ใช้แบบกราฟิกและยังมี API สำหรับการเข้าถึงของบุคคลที่สาม

1. **Installation** : ไม่ว่าในกรณีใด เราสามารถติดตั้งแพ็คเกจ Network Manager ผ่านทางตัวจัดการแพ็คเกจ เช่น apt หรือ yum

```
$ sudo apt-get install network-manager // for debian
# yum install NetworkManager // for Fedora
```

2. **Service Execution** : ที่สำคัญเราต้องแน่ใจว่าบริการนั้นเปิดใช้งานอยู่ ดังนั้นเราอาจตรวจสอบด้วย `systemctl status`

```
$ systemctl status NetworkManager
```

```
putter@authority:~$ systemctl status NetworkManager
● NetworkManager.service - Network Manager
   Loaded: loaded (/lib/systemd/system/NetworkManager.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2024-02-11 03:29:21 UTC; 51min ago
     Docs: man:NetworkManager(8)
  Main PID: 839 (NetworkManager)
    Tasks: 3 (limit: 2220)
   Memory: 10.6M
      CPU: 115ms
   CGroup: /system.slice/NetworkManager.service
           └─839 /usr/sbin/NetworkManager --no-daemon

Feb 11 03:29:21 authority NetworkManager[839]: <info> [1707622161.8009] ifupdown: management mode: >
Feb 11 03:29:21 authority NetworkManager[839]: <info> [1707622161.8038] ifupdown: interfaces file >
Feb 11 03:29:21 authority NetworkManager[839]: <info> [1707622161.8115] dhcp-init: Using DHCP clie>
Feb 11 03:29:21 authority NetworkManager[839]: <info> [1707622161.8119] device (lo): carrier: link>
Feb 11 03:29:21 authority NetworkManager[839]: <info> [1707622161.8123] manager: (lo): new Generic>
Feb 11 03:29:21 authority NetworkManager[839]: <info> [1707622161.8142] device (enp0s3): carrier: >
Feb 11 03:29:21 authority NetworkManager[839]: <info> [1707622161.8213] manager: (enp0s3): new Eth>
Feb 11 03:29:21 authority NetworkManager[839]: <info> [1707622161.8244] failed to open /run/networ>
Feb 11 03:29:21 authority NetworkManager[839]: <info> [1707622161.8384] manager: startup complete
Feb 11 03:29:21 authority NetworkManager[839]: <info> [1707622161.8489] modem-manager: ModemManag>
lines 1-21/21 (END)
```

[ภาพที่ 2] ตรวจสอบสถานะของ Network Manager

ถ้ายังไม่ start หรือจะ enable อัตโนมัติตอนที่เครื่องรีบูตขึ้นมาให้ใช้คำสั่ง

```
# systemctl start NetworkManager
# systemctl enable NetworkManager
```

3. **Configuration Files** : ไฟล์การกำหนดค่าของ Network Manager อยู่ใน `/etc/NetworkManager` นอกจากนี้ การตั้งค่าการเชื่อมต่อจะถูกจัดเก็บไว้ในไฟล์เดอรัยย่อยการเชื่อมต่อระบบ โดยส่วนใหญ่ เราจะใช้คำสั่ง `nmcli` เพื่อกำหนดค่าอุปกรณ์เครือข่ายของเรา เนื่องจากจะเกิดการเปลี่ยนแปลงในไฟล์การกำหนดค่าไว้ เราจึงไม่จำเป็นต้องแก้ไขด้วยตนเอง

เครื่องมือที่สำคัญเพิ่มเติม

Application or Tool	Description
nmcli	เครื่องมือบรรทัดคำสั่งที่ช่วยให้ผู้ใช้และสคริปต์โต้ตอบกับ NetworkManager โปรดทราบว่า nmcli สามารถใช้บนระบบที่ไม่มี GUI
nmtui	ส่วนต่อประสานผู้ใช้แบบ curses-based อย่างง่าย (TUI) สำหรับ NetworkManager
nm-connection-editor	เครื่องมืออินเทอร์เฟซผู้ใช้แบบกราฟิกสำหรับงานบางอย่างที่ยูทิลิตี้ศูนย์ควบคุมยังไม่ได้จัดการ เช่น การกำหนดค่าการเชื่อมต่อและการเชื่อมต่อแบบทีม

control-center	เครื่องมือส่วนต่อประสานกราฟิกกับผู้ใช้ที่ GNOME Shell มอบให้สำหรับผู้ใช้เดสก์ท็อป
network connection icon	เครื่องมือส่วนต่อประสานกราฟิกกับผู้ใช้ที่ GNOME Shell นำเสนอสถานะการเชื่อมต่อเครือข่ายตามที่รายงานโดย NetworkManager ไอคอนมีหลายสถานะที่ทำหน้าที่เป็นตัวบ่งชี้ภาพสำหรับประเภทการเชื่อมต่อที่คุณกำลังใช้อยู่

ข้อดีและข้อเสียของการใช้ NetworkManager

ข้อดี	ข้อเสีย
จัดการเครือข่ายอัตโนมัติ	ใช้ทรัพยากรระบบจำนวนมาก
จัดการเครือข่ายได้หลายรูปแบบ	การเชื่อมต่อเครือข่ายอาจไม่เสถียรเท่าที่ควร
รองรับการเชื่อมต่อ VPN	การตั้งค่าที่ซับซ้อน
มีส่วนขยาย (Plug-ins) และการปรับแต่ง	ความเสถียรภาพของเวอร์ชัน
ใช้งานได้บนหลายแพลตฟอร์ม	

การประยุกต์ใช้คำสั่งกับการใช้งานต่างๆ

Check Ethernet hardware

ใช้คำสั่ง `lspci -vv` เพื่อดูสิ่งที่ Linux ตรวจพบการใช้งานได้ของ Hardware ทั้งหมดหรือใช้คำสั่ง `lspci -vv | grep Ethernet` เราจะเฉพาะ hardware ที่เป็น Ethernet แต่ไม่แสดงรายละเอียด

| lspci เป็นยูทิลิตี้สำหรับแสดงข้อมูลเกี่ยวกับบัส PCI ในระบบและอุปกรณ์ที่เชื่อมต่ออยู่

```
00:03.0 Ethernet controller: Intel Corporation 82540EM Gigabit Ethernet Controller (rev 02)
  Subsystem: Intel Corporation PRO/1000 MT Desktop Adapter
  Control: I/O+ Mem+ BusMaster+ SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B-
  DisINTx-
  Status: Cap+ 66MHz+ UDF- FastB2B- ParErr- DEVSEL=medium >TAbort- <TAbort- <MAbort- >SERR- <P
  ERR- INTx-
  Latency: 64 (63750ns min)
  Interrupt: pin A routed to IRQ 19
  Region 0: Memory at f0200000 (32-bit, non-prefetchable) [size=128K]
  Region 2: I/O ports at d020 [size=8]
  Capabilities: <access denied>
  Kernel driver in use: e1000
  Kernel modules: e1000
```

[ภาพที่ 3] ใช้คำสั่ง `lspci -vv | more` แล้วค้นหา ethernet controller

```
$ lspci -vv | grep Ethernet
00:03.0 Ethernet controller: Intel Corporation 82540EM Gigabit Ethernet Contr
```

คำสั่ง `lshw` สามารถระบุฮาร์ดแวร์เครือข่ายทั้งหมดที่พร้อมใช้งานสำหรับระบบของคุณ โดยให้ข้อมูลโดยละเอียดเกี่ยวกับความสามารถด้านฮาร์ดแวร์ของอะแดปเตอร์เฉพาะ รวมถึงข้อมูลบัส รายละเอียดไดรเวอร์ และความสามารถที่สนับสนุนทั้งหมด

```

root@authority:/# lshw -class network
*-network
   description: Ethernet interface
   product: 82540EM Gigabit Ethernet Controller
   vendor: Intel Corporation
   physical id: 3
   bus info: pci@0000:00:03.0
   logical name: enp0s3
   version: 02
   serial: 08:00:27:b1:d5:82
   size: 1Gbit/s
   capacity: 1Gbit/s
   width: 32 bits
   clock: 66MHz
   capabilities: pm pcix bus_master cap_list ethernet physical tp 10bt 10bt-fd 100bt 100bt-fd 1000bt-fd autonegotiation
   configuration: autonegotiation=on broadcast=yes driver=e1000 driverversion=5.15.0-92-generic duplex=full ip=192.168.1.191 latency=64 link=yes mingnt=255 multicast=yes port=twisted pair speed=1Gbit/s
   resources: irq:19 memory:f0200000-f021ffff ioport:d020(size=8)
root@authority:/#

```

[ภาพที่ 4] การแสดงผลจากคำสั่ง `lshw -class network`

File and Directory

ในระบบปฏิบัติการ Linux เบื้องต้น เมื่อ NIC ถูกติดตั้งและกำหนดค่าแล้ว ข้อมูลเกี่ยวกับ NIC จะถูกเก็บไว้ในไฟล์ที่อยู่ในไดเรกทอรี `/etc/network/` ซึ่งมีไฟล์หลายไฟล์ที่สร้างขึ้นโดยอัตโนมัติหรือปรับแต่งเพื่อกำหนดค่าต่าง ๆ ของเครือข่ายตัวอย่างเช่น :

1. `/etc/network/interfaces` : ไฟล์นี้เป็นไฟล์หลักที่ใช้กำหนดค่าเครือข่ายของระบบ Debian และ Ubuntu ในไฟล์นี้คุณสามารถกำหนดค่าต่าง ๆ เช่น IP address, netmask, gateway, DNS servers, และอื่น ๆ สำหรับแต่ละ NIC ได้
2. `/etc/netplan` : ในปัจจุบันมีเครื่องมือที่ชื่อว่า **"Netplan"** เป็นเครื่องมือกำหนดค่าเครือข่าย Ubuntu เวอร์ชันล่าสุดทั้งหมด Netplan ขึ้นอยู่กับระบบการกำหนดค่าที่ใช้ YAML ซึ่งทำให้กระบวนการกำหนดค่าง่ายขึ้น แทนที่ไฟล์การกำหนดค่า `/etc/network/interfaces` แบบเก่าใน Ubuntu
3. `/etc/udev/rules.d/70-snap.snapd.rules` : ไฟล์นี้เก็บข้อมูลเกี่ยวกับการกำหนดชื่ออินเทอร์เฟซของ NIC แต่ละตัว โดยแต่ละ NIC จะมีหมายเลข MAC address และชื่ออินเทอร์เฟซที่ถูกกำหนดให้ ไฟล์นี้สร้างขึ้นโดยอัตโนมัติเมื่อระบบทำการสร้างรายการของ NIC

```

GNU nano 6.2                                70-snap.snapd.rules
# This file is automatically generated.

# Concatenation of all ModemManager udev rules
# do not edit this file, it will be overwritten on update

ACTION!="add|change|move|bind", GOTO="mm_cinterion_port_types_end"
SUBSYSTEMS=="usb", ATTRS{idVendor}=="1e2d", GOTO="mm_cinterion_port_types"
GOTO="mm_cinterion_port_types_end"

LABEL="mm_cinterion_port_types"
SUBSYSTEMS=="usb", ATTRS{bInterfaceNumber}=="?*", ENV{MM_USBIFNUM}="$attr{bInterfaceNumber}"

# PHS8
ATTRS{idVendor}=="1e2d", ATTRS{idProduct}=="0053", ENV{MM_USBIFNUM}=="01", SUBSYSTEM=="tty", ENV{ID_MM_CINTERION_PORT_IGNORE}="1"

# PLS8 port types
# ttyACM0 (if #0): AT port
# ttyACM1 (if #2): AT port
# ttyACM2 (if #4): GPS data port
# ttyACM3 (if #6): unknown
# ttyACM4 (if #8): unknown
ATTRS{idVendor}=="1e2d", ATTRS{idProduct}=="0061", ENV{MM_USBIFNUM}=="00", SUBSYSTEM=="tty", ENV{ID_MM_CINTERION_PORT_IGNORE}="1"
ATTRS{idVendor}=="1e2d", ATTRS{idProduct}=="0061", ENV{MM_USBIFNUM}=="02", SUBSYSTEM=="tty", ENV{ID_MM_CINTERION_PORT_IGNORE}="1"
ATTRS{idVendor}=="1e2d", ATTRS{idProduct}=="0061", ENV{MM_USBIFNUM}=="04", SUBSYSTEM=="tty", ENV{ID_MM_CINTERION_PORT_IGNORE}="1"
ATTRS{idVendor}=="1e2d", ATTRS{idProduct}=="0061", ENV{MM_USBIFNUM}=="06", ENV{ID_MM_PORT_IGNORE}="1"
ATTRS{idVendor}=="1e2d", ATTRS{idProduct}=="0061", ENV{MM_USBIFNUM}=="08", ENV{ID_MM_PORT_IGNORE}="1"

# PLS62 family non-mbim enumeration uses alternate settings for 2G band management
ATTRS{idVendor}=="1e2d", ATTRS{idProduct}=="005b", ENV{ID_MM_CINTERION_MODEM_FAMILY}="imt"

# PLS62 family non-mbim enumeration
# ttyACM0 (if #0): AT port
# ttyACM1 (if #2): AT port
# ttyACM2 (if #4): can be AT or GNSS in some models

[ Read 1138 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location  M-U Undo
^X Exit      ^R Read File ^N Replace   ^U Paste     ^J Justify   ^_ Go To Line M-E Redo

```

[ภาพที่ 5] เข้าไปยัง /etc/udev/rules.d/70-snap.snapd.rules

3. `/sys/class/net/`: นี่เป็นไดเรกทอรีที่เก็บข้อมูลเกี่ยวกับอินเทอร์เฟซเครือข่ายทั้งหมดในระบบ เช่นชื่อของอินเทอร์เฟซสถานะของอินเทอร์เฟซ เป็นต้น

```

putter@authority:~$ cd /sys/class/net/
putter@authority:/sys/class/net$ ls -la
.  ..  enp0s3  lo
putter@authority:/sys/class/net$ cd enp0s3
putter@authority:/sys/class/net/enp0s3$ ls -la
.  ..  carrier_down_count  gro_flush_timeout  netdev_group  speed
..  carrier_up_count   ifalias            operstate          statistics
addr_assign_type      device             ifindex            phys_port_id
address                dev_id             iflink              phys_port_name
addr_len               dev_port           link_mode           phys_switch_id
broadcast              dormant            mtu                  power
carrier                duplex              name_assign_type    proto_down
carrier_changes         flags              napi_defer_hard_irqs  queues
type
uevent
putter@authority:/sys/class/net/enp0s3$ cat address
08:00:27:b1:d5:82
putter@authority:/sys/class/net/enp0s3$

```

[ภาพที่ 6] เข้าดูข้อมูลของ network interface enp0s3

enp0s3 → en = ethernet, p = PCI, 0s3 = ตัวเลขของ PCI | lo → Loopback Address

การตั้งค่าเครือข่ายในระบบ Debian ส่วนใหญ่จะใช้ไฟล์ `/etc/network/interfaces` เพื่อกำหนดค่า NIC แต่ละตัว และข้อมูลเพิ่มเติมเกี่ยวกับ NIC อาจถูกเก็บไว้ในไฟล์อื่น ๆ ที่เกี่ยวข้องในระบบเช่นเดียวกับไฟล์ที่กล่าวถึงข้างต้น

Listing Network Interface Name

ดังนั้น, เราสามารถใช้คำสั่ง `ls` และระบบไฟล์ `sys` เพื่อแสดงรายการของอินเทอร์เฟซเครือข่ายที่มีอยู่อย่างรวดเร็ว แต่ละรายการในไดเรกทอรี `/sys/class/net` จะแทนอินเทอร์เฟซเครือข่ายทั้ง physical และ virtual :

```
$ ls /sys/class/net
enp0s3 lo
```

หากต้องการดูรายละเอียดเพิ่มเติมเกี่ยวกับอินเทอร์เฟซเครือข่าย เราสามารถใช้คำสั่ง `ip link` หรือ `ip addr` :

```
root@authority:~# ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000
    link/ether 08:00:27:b1:d5:82 brd ff:ff:ff:ff:ff:ff
root@authority:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:b1:d5:82 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.191/24 metric 100 brd 192.168.1.255 scope global dynamic enp0s3
        valid_lft 81342sec preferred_lft 81342sec
    inet 192.168.1.120/25 brd 192.168.1.127 scope global enp0s3:0
        valid_lft forever preferred_lft forever
    inet6 2001:fb1:a9:adac:a00:27ff:feb1:d582/64 scope global dynamic mngtmpaddr noprefixroute
        valid_lft 259127sec preferred_lft 172727sec
    inet6 fe80::a00:27ff:feb1:d582/64 scope link
        valid_lft forever preferred_lft forever
root@authority:~# _
```

[ภาพที่ 7] การแสดงผลจากคำสั่งของ `ip addr` และ `ip link`

นอกจากนี้ยังมีคำสั่งอื่นๆอีก ยกตัวอย่างเช่น

Command	Description
<code>lspci</code>	แสดงรายการอุปกรณ์ PCI ทั้งหมด
<code>lshw</code>	ระบุอินเทอร์เฟซอีเธอร์เน็ตและฮาร์ดแวร์ NIC
<code>dmidecode</code>	แสดงรายการข้อมูลฮาร์ดแวร์ทั้งหมดจาก BIOS
<code>ifconfig</code>	ยูทิลิตี้การกำหนดค่าเครือข่ายที่ล้าสมัย
<code>hwinfo</code>	Probe Linux สำหรับการ์ดเครือข่าย
<code>ethtool</code>	ดูไดเวอร์ NIC/การ์ดและการตั้งค่าบน Linux
<code>cat /proc/net/dev</code>	dev pseudo-file มีข้อมูลสถานะอุปกรณ์เครือข่าย

ตัวอย่างการเรียกใช้งาน

```
# lspci | egrep -i --color 'network|ethernet|wireless|wi-fi'
# lshw -class network -short
# ip -br -c link showcat /proc/net/dev
$ cat /proc/net/dev
```



```

putter@authority:~$ lspci | egrep -i --color ethernet
00:03.0 Ethernet controller: Intel Corporation 82540EM Gigabit Ethernet Controller (rev 02)
putter@authority:~$ sudo lshw -class network -short
[sudo] password for putter:
H/W path          Device          Class          Description
=====
/0/100/3           enp0s3           network        82540EM Gigabit Ethernet Controller
putter@authority:~$ ip -br -c link show
lo                UNKNOWN        00:00:00:00:00:00 <LOOPBACK,UP,LOWER_UP>
enp0s3            UP             08:00:27:b1:d5:82 <BROADCAST,MULTICAST,UP,LOWER_UP>
putter@authority:~$ cat /proc/net/dev
Inter-|   Receive                                           | Transmit
face |bytes  packets errs drop fifo frame compressed multicast|bytes  packets errs drop fifo co
lls carrier compressed
lo:   8360    98      0    0    0      0          0          0    8360    98      0    0    0
0:    0       0      0    0    0      0          0          0     0       0      0    0    0
enp0s3: 2226258 4981    0    0    0      0          0    4548  19736  136    0    0    0
0:    0       0      0    0    0      0          0          0     0       0      0    0    0

```

[ภาพที่ 8] การแสดงผลจากการใช้คำสั่งตัวอย่างด้านบน

Enabling and disabling Network Interfaces

สามารถเช็คสถานะการทำงานของ interface นั้นๆ ได้ด้วยคำสั่งดังนี้

1. ใช้คำสั่ง `ip link show dev [interface-name]`
2. ใช้คำสั่ง `ip a sh dev [interface-name]`

```

putter@authority:~$ ip link show dev enp0s3
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group def
ault qlen 1000
    link/ether 08:00:27:b1:d5:82 brd ff:ff:ff:ff:ff:ff
putter@authority:~$ ip link show dev lo
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen
1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
putter@authority:~$

```

[ภาพที่ 9] การแสดงผลจากคำสั่งของ `ip link show dev enp0s3`

“UP” ใน `<BROADCAST,MULTICAST,UP,LOWER_UP>` คือสิ่งที่บ่งชี้ว่าอินเทอร์เฟซนี้ยังเปิดใช้งานอยู่ ถ้าไม่ได้เปิดใช้งานสถานะจะเป็น “DOWN” หรือไม่มี “UP” เกิดขึ้น

การเปิดและปิดสถานะของ interfaces ทำได้หลากหลายวิธีด้วยกัน ยกตัวอย่างเช่น

1. ใช้คำสั่ง `ip link set [interface-name] up/down`
2. ใช้คำสั่ง `netscript ifup|ifdown|ifqos|ifreload [interface-name]|all`
3. ใช้คำสั่ง `ifconfig [interface-name] up/down`
4. ใช้คำสั่ง `ifup [interface-name]` เพื่อเปิดและ `ifdown [interface-name]` เพื่อปิด

```

root@authority:/# ip a sh dev enp0s3
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:b1:d5:82 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.191/24 metric 100 brd 192.168.1.255 scope global dynamic enp0s3
        valid_lft 85048sec preferred_lft 85048sec
    inet6 2001:fb1:ab:fb3b:391b:2bb1:d3de:25da/64 scope global temporary dynamic
        valid_lft 258953sec preferred_lft 84561sec
    inet6 2001:fb1:ab:fb3b:a00:27ff:feb1:d582/64 scope global dynamic mngtmpaddr noprefixroute
        valid_lft 258953sec preferred_lft 172553sec
    inet6 fe80::a00:27ff:feb1:d582/64 scope link
        valid_lft forever preferred_lft forever
root@authority:/# ifconfig enp0s3 down
root@authority:/# ip a sh dev enp0s3
2: enp0s3: <BROADCAST,MULTICAST> mtu 1500 qdisc fq_codel state DOWN group default qlen 1000
    link/ether 08:00:27:b1:d5:82 brd ff:ff:ff:ff:ff:ff
root@authority:/# _

```

[ภาพที่ 10] การแสดงผลของ interface ทั้งก่อนปิดและหลังปิด enp0s3

การใช้แต่ละคำสั่งจะขึ้นอยู่กับการทำงานของระบบ Linux ที่เอามาใช้ไม่ว่าจะเป็น version, package ที่นำมาติดตั้ง, ระบบปฏิบัติการ (Debian หรือ Fedora) หรือปัจจัยความเข้ากันในรูปแบบอื่นๆ

Ethernet Interface settings

ethtool เป็นโปรแกรมที่แสดงและเปลี่ยนการตั้งค่าการ์ด Ethernet เช่น การต่อรองอัตโนมัติ, ความเร็วพอร์ต, โหมดดูเพล็กซ์, และ Wake-on-LAN

```

root@authority:/# ethtool enp0s3
Settings for enp0s3:
    Supported ports: [ TP ]
    Supported link modes:   10baseT/Half 10baseT/Full
                           100baseT/Half 100baseT/Full
                           1000baseT/Full
    Supported pause frame use: No
    Supports auto-negotiation: Yes
    Supported FEC modes: Not reported
    Advertised link modes:  10baseT/Half 10baseT/Full
                           100baseT/Half 100baseT/Full
                           1000baseT/Full
    Advertised pause frame use: No
    Advertised auto-negotiation: Yes
    Advertised FEC modes: Not reported
    Speed: 1000Mb/s
    Duplex: Full
    Auto-negotiation: on
    Port: Twisted Pair
    PHYAD: 0
    Transceiver: internal
    MDI-X: off (auto)
    Supports Wake-on: umbg
    Wake-on: d
    Current message level: 0x00000007 (7)
                           drv probe link
    Link detected: yes
root@authority:/# _

```

[ภาพที่ 11] การแสดงผลจากคำสั่งของ ethtool enp0s3

Network configuration using Netplan

การแก้ไขค่าเริ่มต้นของ interface ที่อยู่ใน netplan

1. ค้นหาชื่อของอินเทอร์เฟซเครือข่ายที่ใช้งานอยู่ที่คุณต้องการกำหนดค่า
2. ไฟล์การกำหนดค่าเริ่มต้นของ Netplan อยู่ภายใต้ไดเรกทอรี /etc/netplan

3. เปิดไฟล์การกำหนดค่าในตัวแก้ไขใด ๆ เพื่อแก้ไขไฟล์การกำหนดค่า

```
$ ls /etc/netplan/
00-installer-config.yaml
$ sudo nano /etc/netplan/*.yaml
```



```
GNU nano 6.2 /etc/netplan/00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: true
      version: 2
```

[ภาพที่ 12] nano editor ที่ไฟล์ของ /etc/netplan/00-installer-config.yaml

โดยจะมีส่วนประกอบเรียงกันเป็นลำดับชั้นตามตัวอย่างด้านล่างนี้

```
network:
  ethernets:
    enp0s3: // ส่วนตรงนี้เป็นชื่อ interface
      Dhcp4: true|false //จะให้ IP เป็น dynamic หรือ static
      Addresses: [192.168.1.102/24] // เป็นเลข IP address และ netmask
      Gateway: 192.168.1.1 // gateway address
      Nameservers:
        Addresses: [8.8.8.8] // dns server นั้นๆ
      version: 2
```

สามารถเพิ่มและกำหนดแก้ไขในส่วนต่างๆได้ แต่ต้องระวังรูปแบบการเขียนและการเว้นช่องไฟ เพราะเป็น case-sensitive

Network configuration using NetworkManager

การดูรายการอุปกรณ์เครือข่าย การเชื่อมต่อ และสถานะเครือข่ายทั่วไป ตัวอย่างเช่น อุปกรณ์เครือข่ายประกอบด้วยการ์ดเครือข่ายเป็นหลัก ในขณะที่การเชื่อมต่อคือการกำหนดค่าที่กำหนดให้กับอุปกรณ์เครือข่าย

- ดูรายชื่ออุปกรณ์เครือข่ายทั้งหมดในเครื่อง Linux ของด้วย `nmcli device` และสามารถแสดงคุณสมบัติที่ละเอียดมากขึ้นได้ด้วย เช่น ที่อยู่ IP ของอุปกรณ์โดยใช้คำสั่ง `nmcli device show [interface-name]`

```

putter@authority:~$ nmcli device
DEVICE  TYPE      STATE      CONNECTION
enp0s3  ethernet  unmanaged  --
lo       loopback  unmanaged  --
putter@authority:~$ nmcli device show enp0s3
GENERAL.DEVICE:           enp0s3
GENERAL.TYPE:             ethernet
GENERAL.HWADDR:           08:00:27:B1:D5:82
GENERAL.MTU:              1500
GENERAL.STATE:            10 (unmanaged)
GENERAL.CONNECTION:       --
GENERAL.CON-PATH:         --
WIRED-PROPERTIES.CARRIER: on
IP4.ADDRESS[1]:           192.168.1.191/24
IP4.GATEWAY:              192.168.1.1
IP4.ROUTE[1]:             dst = 0.0.0.0/0, nh = 192.168.1.1, mt = 100
IP4.ROUTE[2]:             dst = 192.168.1.0/24, nh = 0.0.0.0, mt = 100
IP4.ROUTE[3]:             dst = 192.168.1.1/32, nh = 0.0.0.0, mt = 100
IP6.ADDRESS[1]:           2001:fb1:ab:fb3b:a00:27ff:feb1:d582/64
IP6.ADDRESS[2]:           fe80::a00:27ff:feb1:d582/64
IP6.GATEWAY:              fe80::1
IP6.ROUTE[1]:             dst = 2001:fb1:ab:fb3b::/64, nh = ::, mt = 100
IP6.ROUTE[2]:             dst = fe80::/64, nh = ::, mt = 256
IP6.ROUTE[3]:             dst = ::/0, nh = fe80::1, mt = 100
putter@authority:~$ _

```

[ภาพที่ 13] แสดงผลของคำสั่ง `nmcli device` และ `nmcli device show enp0s3`

- สามารถรับการเชื่อมต่อเครือข่ายที่มืออยู่ด้วยการใช้คำสั่ง `nmcli connection`
- การแสดงสถานะของอแด็ปเตอร์ไร้สายด้วยการใช้คำสั่ง `nmcli radio`
- การเปิดใช้งานและปิดใช้งานการเชื่อมต่อด้วยการใช้คำสั่ง `nmcli connection up | down [interface]`

Temporary Renaming Network Interface

คำสั่ง `ip` สามารถใช้เปลี่ยนชื่ออินเทอร์เฟซเครือข่ายชั่วคราว เช่น การเปลี่ยนชื่อ 'enp0s3' เป็น 'main_enp' ซึ่งเป็นสิ่งที่มีประโยชน์ในการทดสอบหรือแก้ไขปัญห อินเทอร์เฟซเครือข่ายที่จะถูกเปลี่ยนชื่อจำเป็นต้องถูกปิดใช้งานก่อน

```
$ ip link set dev enp0s3 down
```

จากนั้นเราก็จะสามารถเปลี่ยนชื่อ 'enp0s3' เป็น 'main_enp' ได้ด้วยคำสั่ง

```
$ ip link set dev enp0s3 name main_enp
```

สุดท้ายก็สามารถใช้งาน interface ที่ถูกเปลี่ยนได้โดยเปิดการใช้งาน

```
$ ip link set dev enp0s3 up
```

หลังจากดำเนินการตามคำสั่งด้านบน, ชื่ออินเทอร์เฟซเครือข่ายจะถูกเปลี่ยนเป็น main_enp จนกว่าระบบจะบูตใหม่

Permanent Renaming Network Interface

สำหรับระบบ Ubuntu และเป็น Debian-based system, วิธีที่แนะนำให้ใช้ในการทำให้การเปลี่ยนชื่ออินเทอร์เฟซแบบถาวรแม้หลังจากรีบูตคือการใช้ udev rule

เราจะเปิดไฟล์ `udev rules` ที่เกี่ยวข้องกับอินเทอร์เฟซเครือข่ายโดยใช้ตัวแก้ไขข้อความพร้อมสิทธิ์ของผู้ดูแลระบบ ไฟล์นี้ชื่อว่า `70-persistent-net.rules` และอยู่ในไดเรกทอรี `/etc/udev/rules.d/` ถ้าไม่มีไฟล์ตามที่กล่าวมา เราสามารถสร้างมันเองได้

```
$ sudo nano /etc/udev/rules.d/70-persistent-net.rules
```

หลังจากเปิดไฟล์ 70-persistent-net.rules แล้ว ให้ค้นหบรรทัดที่สอดคล้องกับอินเทอร์เฟซเครือข่ายที่เราต้องการเปลี่ยนชื่อ และแก้ไขพารามิเตอร์ "NAME" ด้วยชื่อใหม่ที่ต้องการ

```
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="*", ATTR{address}=="02:42:ac:11:00:00", ATTR{dev_id}=="0x0", ATTR{type}=="1", KERNEL=="eth*", NAME="neweth0"
```

หลังจากที่เปลี่ยนชื่อ interface เรียบร้อยแล้ว เราจำเป็นต้องรู้ระบบ Linux เพื่อใช้การเปลี่ยนชื่ออินเทอร์เฟซเครือข่ายเป็นไปอย่างแบบถาวร เมื่อรีบูต เราสามารถยืนยันได้ว่าเปลี่ยนชื่ออินเทอร์เฟซเครือข่ายสำเร็จแล้วโดยใช้คำสั่ง ip หรือ ifconfig

Reference

1. archlinux. Network configuration. https://wiki.archlinux.org/title/Network_configuration.
2. Ubuntu. Configuring networks. <https://ubuntu.com/server/docs/network-configuration>.
3. Karim Buzdar. (2023). How to Configure Networking on Ubuntu with Netplan. <https://vitux.com/how-to-configure-networking-with-netplan-on-ubuntu/>
4. debianwiki. NetworkConfiguration. https://wiki.debian.org/NetworkConfiguration#Setting_up_an_Ethernet_Interface
5. Aashish Khadka. (2023). How to Rename a Network Interface in Linux? <https://www.baeldung.com/linux/rename-network-interface>
6. Canonical Ltd. (2024). Ubuntu and Canonical are registered trademarks of Canonical. <https://netplan.io/>
7. Netplan Documentation. <https://netplan.readthedocs.io/en/stable/netplan-tutorial/>
8. Net-Tools. <https://www.kali.org/tools/net-tools/>
9. Miglen Evlogiev. (2020). Linux networking tools. <https://gist.github.com/miglen/70765e663c48ae0544da08c07006791f>
10. **Text User Interface for controlling NetworkManager.** <https://linuxcommandlibrary.com/man/nmtui>
11. Vivek Gite. (2023). How To: Linux Show List Of Network Cards. <https://www.cyberciti.biz/faq/linux-list-network-cards-command/>
12. baeldung. (2022). Configure Network Settings Using Network Manager in Linux. <https://www.baeldung.com/linux/network-manager>.
13. Red Hat Customer Portal. Chapter 2. Getting Started with NetworkManager. https://access.redhat.com/documentation/th-th/red_hat_enterprise_linux/7/html/networking_guide/sec-checking_the_status_of_networkmanager
14. ChatGPT. (2024). Netplan ติดตั้งใน Ubuntu. <https://chat.openai.com/share/aa99791b-1c1e-41e3-a2cf-919681180ff8>
15. Fedora Documentation. <https://dsilas.fedorapeople.org/deployment-guide/html/s1-networkscripts-interfaces.html>

เขียนและเรียบเรียงเนื้อหาโดย นายภควัฒน์ พันธุภักต์วงษ์ 65070165 สื่อที่เป็นรูปภาพทั้งหมดในหัวข้อนี้เป็นของผู้
เขียนเอง อนุญาตให้นำไปใช้ได้