นายธนพล วงศ์อาษา 62010356

นายสุทธิราช ภูโท 62010966

**การทดลองที่ 9 การศึกษาและปรับแก้อินพุทและเอาท์พุทต่าง ๆ**

**ระบบเสียงดิจิทัล**

1. ผู้อ่านสามารถทดสอบโดยพิมพ์คำสั่งต่อไปนี้

$ cat /proc/asound/ALSA

บันทึกผลลัพธ์ในพื้นที่ว่างต่อไปนี้

**ตอบ** \*ขอแก้ไขคำสั่งเพื่อให้แสดงข้อมูลจริงออกมา (มิเช่นนั้นจะแจ้งว่า cat: /proc/asound/ALSA: Is a directory)

pi@raspberrypi:/ $ cat proc/asound/ALSA/pcm0p/info

card: 0

device: 0

subdevice: 0

stream: PLAYBACK

id: bcm2835 ALSA

name: bcm2835 ALSA

subname: subdevice #0

class: 0

subclass: 0

subdevices\_count: 7

subdevices\_avail: 7

2. ผู้อ่านสามารถทดสอบโดยพิมพ์คำสั่งต่อไปนี้

$ cat /proc/asound/card0

บันทึกผลลัพธ์ในพื้นที่ว่างต่อไปนี้และเปรียบเทียบกับผลลัพธ์ก่อนหน้าว่าแตกต่างกันหรือไม่

**ตอบ** \*ขอแก้ไขคำสั่งเพื่อให้แสดงข้อมูลจริงออกมา (มิเช่นนั้นจะแจ้งว่า cat: /proc/asound/card0: Is a directory)

pi@raspberrypi:/ $ cat proc/asound/ALSA/pcm0p/info

card: 0

device: 0

subdevice: 0

stream: PLAYBACK

id: bcm2835 ALSA

name: bcm2835 ALSA

subname: subdevice #0

class: 0

subclass: 0

subdevices\_count: 7

subdevices\_avail: 7

ข้อมูลกับข้อก่อนหน้าเหมือนกัน

3. ค้นคว้าเพิ่มเติมเพื่อหาความหมายของ Symbolic Link และจดบันทึก

**ตอบ** Symbolic Link (Symlink) คือลิงก์ที่ทำการ “ชี้” ไปยังไฟล์หรือโฟลเดอร์อื่น ๆ ในตัวเครื่อง มีความแตกต่างกับ Shortcut ในระบบปฏิบัติการณ์ Windows ที่ Symlink ทำงานที่ระดับระบบไฟล์และไม่เป็นไฟล์จริง ๆ แต่ Shortcut จะนับเป็นไฟล์

4. พิมพ์คำสั่งนี้ในโปรแกรม Terminal

$ cat /proc/asound/cards

โดยคำสั่ง cat ซึ่งได้อธิบายแล้วในการทดลองที่4 ภาคผนวก D สามารถอ่านไฟล์และแสดงข้อมูลภายใน

ไฟล์ผ่านทางหน้าจอแสดงผล บันทึกในที่ว่างต่อไปนี้

**ตอบ** pi@raspberrypi:~ $ cat /proc/asound/cards

0 [ALSA ]: bcm2835\_alsa - bcm2835 ALSA

bcm2835 ALSA

อภิปรายผลที่ได้ ดังนี้ ผลลัพธ์ได้จากบอร์ด Pi3 ใช้ชิพ BCM283\_ แต่ยังใช้ไดรเวอร์เสียงเดียวกันกับ

BCM283\_ โดย หมายเลข 0 คือ หมายเลขของระบบเสียงที่ติดตั้งใช้งานเพียงระบบเดียว และตรงกับ

อุปกรณ์ชื่อ \_\_\_\_0

**ตอบ** BCM2711 (ใช้ Pi4) ตรงกับไดรเวอร์ของ BCM2835 โดยตรงกับอุปกรณ์ชื่อ card0

**รายชื่ออุปกรณ์กับพอร์ท USB**

1. ในการทดลองนี้ ขอผู้อ่านให้ดึงหัวเชื่อมต่อ USB ของเมาส์ที่ใช้อยู่ออก แล้วพิมพ์คำสั่งนี้ใน

โปรแกรมTerminal

$ lsusb

2. บันทึกผลลัพธ์ของผู้อ่าน

**ตอบ**

Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub

Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub

Bus 001 Device 004: ID 04d9:0006 Holtek Semiconductor, Inc.

Bus 001 Device 003: ID 05e3:0610 Genesys Logic, Inc. 4-port hub

Bus 001 Device 002: ID 2109:3431 VIA Labs, Inc. Hub

Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub

3. ผู้อ่านเสียบเมาส์กลับเข้าไปที่พอร์ท USB ใหม่อีกครั้ง แล้วแสดงรายชื่ออุปกรณ์USB ด้วยคำสั่ง

$ lsusb

เช่นเดิม บันทึกผลลงในพื้นที่จัดเตรียมไว้ให้โปรดสังเกตการเปลี่ยนแปลง

**ตอบ**

pi@raspberrypi:~ $ lsusb

Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub

Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub

Bus 001 Device 004: ID 04d9:0006 Holtek Semiconductor, Inc.

Bus 001 Device 003: ID 05e3:0610 Genesys Logic, Inc. 4-port hub

**Bus 001 Device 009: ID 093a:2510 Pixart Imaging, Inc. Optical Mouse**

Bus 001 Device 002: ID 2109:3431 VIA Labs, Inc. Hub

Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub

4. รายการที่เพิ่มขึ้น คือ

**ตอบ** Bus 001 Device 009: ID 093a:2510 Pixart Imaging, Inc. Optical Mouse

**รายละเอียดการเชื่อมต่ออุปกรณ์กับพอร์ท USB**

1. ผู้อ่านสามารถล้างบัฟเฟอร์โดยใช้คำสั่ง ต่อไปนี้

$ sudo dmesg -C

โดย -C คือ Clear เป็นคำสั่งเพิ่มเติมให้dmesg ล้างข้อความในบัฟเฟอร์ออก โปรดสังเกต ตัว C ใหญ่

หลังจากนั้น ผู้อ่านทดสอบโดยการถอดเมาส์ออก แล้วเสียบกลับเข้าไปใหม่

2. ผู้อ่านจะต้องถอดและเสียบเมาส์กลับเข้าไปใหม่อีกรอบ

3. ผู้อ่านสามารถแสดงข้อความที่เพิ่มเข้ามาในบัฟเฟอร์ได้อีก โดยเรียกคำสั่ง

$ sudo dmesg

4. จดบันทึก

**ตอบ**

pi@raspberrypi:~ $ sudo dmesg -C

pi@raspberrypi:~ $ sudo dmesg

[164661.201867] usb 1-1.3.4: USB disconnect, device number 12

[164665.338327] usb 1-1.3.4: new low-speed USB device number 13 using xhci\_hcd

[164665.475792] usb 1-1.3.4: New USB device found, idVendor=093a, idProduct=2510, bcdDevice= 1.00

[164665.475807] usb 1-1.3.4: New USB device strings: Mfr=1, Product=2, SerialNumber=0

[164665.475820] usb 1-1.3.4: Product: USB Optical Mouse

[164665.475832] usb 1-1.3.4: Manufacturer: PixArt

[164665.485377] input: PixArt USB Optical Mouse as /devices/platform/scb/fd500000.pcie/pci0000:00/0000:00:00.0/0000:01:00.0/usb1/1-1/1-1.3/1-1.3.4/1-1.3.4:1.0/0003:093A:2510.000A/input/input9

[164665.485935] hid-generic 0003:093A:2510.000A: input,hidraw2: USB HID v1.11 Mouse [PixArt USB Optical Mouse] on usb-0000:01:00.0-1.3.4/input0

5. อภิปรายผลลัพธ์ที่บันทึกได้ในพื้นที่ว่างต่อไปนี้

**ตอบ** เมื่อทำการเช็คครั้งแรก ระบบจะรันหาอุปกรณ์ที่ต่อกับเครื่องบนบัสและจะพบว่ามีการเชื่อมต่อต่าง ๆ ได้แก่ Linux Foundation root hub 2.0, Linux Foundation root hub 3.0, Holtek semiconductor, Genesys Logic 4-port hub, VIA Lab hub, และ Linux Foundation root hub 2.0

แต่เมื่อทำการต่อเมาส์เข้าไปเพิ่มแล้วเช็คใหม่ จะเห็นอุปกรณ์เพิ่มอีกอย่างคือ Pixart Optical Mouse เพิ่มมาจากตอนแรกอีก

**รายชื่ออุปกรณ์เครือข่าย**

1. ผู้อ่านสามารถตรวจสอบรายชื่ออุปกรณ์สำหรับเชื่อมต่อเครือข่ายได้จากคำสั่ง **ifconfig** ทางโปรแกรม

Terminal ตัวอย่างผลลัพธ์เป็นดังนี้

$ ifconfig

2. เติมข้อมูลหรือตัวเลขในช่องว่าง \_ ที่เตรียมไว้ให้จากผลลัพธ์ที่ได้ต่อไปนี้ ซึ่งลำดับรายการอาจแตกต่างกัน

**ตอบ**

pi@raspberrypi:~ $ ifconfig

eth0: flags=4099<UP,BROADCAST,MULTICAST> mtu **1500**

ether dc:a6:32:6f:11:57 txqueuelen 1000 (Ethernet)

RX packets 0 bytes 0 (0.0 B)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 0 bytes 0 (0.0 B)

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**

inet6 ::1 prefixlen 128 scopeid 0x10<host>

loop txqueuelen 1000 (Local Loopback)

RX packets 15 bytes 1156 (1.1 KiB)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 15 bytes 1156 (1.1 KiB)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu **1500**

inet **192.168.88.13** netmask **255.255.255.0** broadcast **192.168.88.255**

inet6 fe80::998b:b8ef:5bde:9ade prefixlen 64 scopeid 0x20<link>

ether **dc:a6:32:6f:11:58** txqueuelen 1000 (Ethernet)

RX packets 458546 bytes 91562703 (87.3 MiB)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 471728 bytes 128402188 (122.4 MiB)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

3. โปรดสังเกตคำเริ่มต้นในแต่ละรายการ ค้นคว้า และกรอกรายละเอียดเพิ่มเติม ดังนี้

• eth0 หมายถึง  **ตอบ** Ethernet Interface

• lo0 หมายถึง  **ตอบ** Loopback Interface/Localhost

• wlan0 หมายถึง  **ตอบ** Wireless Local Area Network Interface

**การเปิด/ปิดอุปกรณ์เครือข่าย**

1. ผู้อ่านสามารถเปิดอุปกรณ์eth0 ได้ตามต้องการแล้วทำการตรวจสอบ ดังนี้

$ sudo ifconfig eth0 down

$ ifconfig

จดว่าข้อความใดที่บ่งบอกว่า eth0 ไม่ทำงานแล้ว

**ตอบ**

pi@raspberrypi:~ $ sudo ifconfig eth0 down

pi@raspberrypi:~ $ ifconfig

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536

inet 127.0.0.1 netmask 255.0.0.0

inet6 ::1 prefixlen 128 scopeid 0x10<host>

loop txqueuelen 1000 (Local Loopback)

RX packets 15 bytes 1156 (1.1 KiB)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 15 bytes 1156 (1.1 KiB)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet 192.168.88.13 netmask 255.255.255.0 broadcast 192.168.88.255

inet6 fe80::998b:b8ef:5bde:9ade prefixlen 64 scopeid 0x20<link>

ether dc:a6:32:6f:11:58 txqueuelen 1000 (Ethernet)

RX packets 458724 bytes 91582859 (87.3 MiB)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 471943 bytes 128449516 (122.4 MiB)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

2. ผู้อ่านสามารถเปิดอุปกรณ์eth0 ได้ตามต้องการแล้วทำการตรวจสอบ ดังนี้

$ sudo ifconfig eth0 up

$ ifconfig

จดว่าข้อความใดที่บ่งบอกว่า eth0 ทำงานแล้ว

**ตอบ**

pi@raspberrypi:~ $ sudo ifconfig eth0 up

pi@raspberrypi:~ $ ifconfig

eth0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500

ether dc:a6:32:6f:11:57 txqueuelen 1000 (Ethernet)

RX packets 0 bytes 0 (0.0 B)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 0 bytes 0 (0.0 B)

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

inet6 ::1 prefixlen 128 scopeid 0x10<host>

loop txqueuelen 1000 (Local Loopback)

RX packets 15 bytes 1156 (1.1 KiB)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 15 bytes 1156 (1.1 KiB)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet 192.168.88.13 netmask 255.255.255.0 broadcast 192.168.88.255

inet6 fe80::998b:b8ef:5bde:9ade prefixlen 64 scopeid 0x20<link>

ether dc:a6:32:6f:11:58 txqueuelen 1000 (Ethernet)

RX packets 458751 bytes 91585829 (87.3 MiB)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 471981 bytes 128456332 (122.5 MiB)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

3. ผู้อ่านสามารถใช้คำสั่ง ifconfig สำหรับปิด อุปกรณ์wlan0 ดังนี้

$ sudo ifconfig wlan0 down

$ ifconfig

จดว่าข้อความใดที่บ่งบอกว่า wlan0 ไม่ทำงานแล้ว

**ตอบ**

pi@raspberrypi:~ $ sudo ifconfig wlan0 down

pi@raspberrypi:~ $ ifconfig

eth0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500

ether dc:a6:32:6f:11:57 txqueuelen 1000 (Ethernet)

RX packets 0 bytes 0 (0.0 B)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 0 bytes 0 (0.0 B)

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

inet6 ::1 prefixlen 128 scopeid 0x10<host>

loop txqueuelen 1000 (Local Loopback)

RX packets 4 bytes 240 (240.0 B)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 4 bytes 240 (240.0 B)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

4. ผู้อ่านสามารถใช้คำสั่ง ifconfig สำหรับเปิด อุปกรณ์wlan0 ดังนี้

$ sudo ifconfig wlan0 up

$ ifconfig

จดว่าข้อความใดที่บ่งบอกว่า wlan0 ทำงานแล้ว

**ตอบ**

pi@raspberrypi:~ $ sudo ifconfig wlan0 up

pi@raspberrypi:~ $ ifconfig

eth0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500

ether dc:a6:32:6f:11:57 txqueuelen 1000 (Ethernet)

RX packets 0 bytes 0 (0.0 B)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 0 bytes 0 (0.0 B)

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

inet6 ::1 prefixlen 128 scopeid 0x10<host>

loop txqueuelen 1000 (Local Loopback)

RX packets 4 bytes 240 (240.0 B)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 4 bytes 240 (240.0 B)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500

ether dc:a6:32:6f:11:58 txqueuelen 1000 (Ethernet)

RX packets 67 bytes 7103 (6.9 KiB)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 76 bytes 13282 (12.9 KiB)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

5. นอกเหนือจากการเปิดปิดอุปกรณ์เครือข่าย ผู้อ่านสามารถตรวจสอบรายชื่อเครือข่าย WiFi ที่บอร์ดเคย

เชื่อมต่อสำเร็จได้จากไฟล์ wpa\_supplicant.conf ซึ่งจะบันทึกรายละเอียดต่างๆ ของการเชื่อมต่อนั้นๆ

รวมถึงพาสเวิร์ด (password) โดยพิมพ์คำสั่งต่อไปนี้ในโปรแกรม Terminal

$ cat /etc/wpa\_supplicant/wpa\_supplicant.conf

บันทึกผลที่ได้โดยกรอกในช่อง \_ เท่านั้น

**ตอบ**

network={

ssid="**exit**"

psk="\*\*\*\*\*\*\*\*\*\*\*\*"

key\_mgmt=**WPA-PSK**

disabled=1

}

• ssid หมายถึง  **ตอบ** ชื่อเครือข่าย

• ssid ย่อมาจาก  **ตอบ** Service Set Identifier

• psk ย่อมาจาก **ตอบ** Pre-Shared Key

• key\_mgmt คือ **ตอบ** รายการของ Authenticated Key Management Protocols

**การตรวจสอบการเชื่อมต่อกับเครือข่ายเบื้องต้น**

เมื่อผู้อ่านเปิดและทำการเชื่อมต่อสำเร็จ แล้วจึงสามารถตรวจสอบการเชื่อมต่อในระดับชั้นเครือข่าย โดยใช้คำ

สั่ง ping ใน Terminal ดังนี้

$ ping <ip address or host name>

การตรวจสอบการเชื่อมต่อเบื้องต้น คือ การ ping ไปหาเราเตอร์ฝั่งต้นทางที่บอร์ดเชื่อมต่อ ผู้อ่านสามารถ

สืบค้นหมายเลขไอพีของเราเตอร์ที่ต้นทาง โดยสังเกตที่inet ของ eth0 หรือ wlan0 ว่าเริ่มต้นด้วยหมายเลข

192.168.x.y ซึ่งเราเตอร์ต้นทางมักจะมีหมายเลข 192.168.x.1 หรือ 192.168.x.254

นี่เป็นตัวอย่างผลลัพธ์ที่ได้ของคำสั่ง ping 192.168.1.1 ที่ผู้อ่านจะต้องเติมหมายเลขลงใน \_\_ที่เตรียมไว้ให้

**ตอบ**

pi@raspberrypi:~ $ ping 192.168.88.1

PING 192.168.88.1 (192.168.88.1) 56(84) bytes of data.

64 bytes from 192.168.88.1: icmp\_seq=**1** ttl=**64** time=1.26 ms

64 bytes from 192.168.88.1: icmp\_seq=**2** ttl=**64** time=2.47 ms

64 bytes from 192.168.88.1: icmp\_seq=**3** ttl=**64** time=4.46 ms

64 bytes from 192.168.88.1: icmp\_seq=**4** ttl=**64** time=1.41 ms

64 bytes from 192.168.88.1: icmp\_seq=**5** ttl=**64** time=1.71 ms

64 bytes from 192.168.88.1: icmp\_seq=**6** ttl=**64** time=1.25 ms

64 bytes from 192.168.88.1: icmp\_seq=**7** ttl=**64** time=1.18 ms

64 bytes from 192.168.88.1: icmp\_seq=**8** ttl=**64** time=3.82 ms

64 bytes from 192.168.88.1: icmp\_seq=**9** ttl=**64** time=6.42 ms

**กิจกรรมท้ายการทดลอง**

3. ใช้คำสั่ง ifconfig ปิดอุปกรณ์lo0 แล้วใช้คำสั่ง ping 127.0.0.1 ว่ามีการตอบสนองกลับมาหรือไม่เปิด

อุปกรณ์lo0 แล้ว ping อีกรอบ จงอธิบายว่า 127.0.0.1 คือ อะไร

**ตอบ**

pi@raspberrypi:~ $ sudo ifconfig lo down

pi@raspberrypi:~ $ ping 127.0.0.1

PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.

--- 127.0.0.1 ping statistics ---

18 packets transmitted, 0 received, 100% packet loss, time 677ms

ไม่สามารถเชื่อมต่อไปยัง 127.0.0.1 ได้เมื่อทำการสั่ง sudo ifconfig lo down

pi@raspberrypi:~ $ sudo ifconfig lo up

pi@raspberrypi:~ $ ping 127.0.0.1

PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.

64 bytes from 127.0.0.1: icmp\_seq=1 ttl=64 time=0.099 ms

64 bytes from 127.0.0.1: icmp\_seq=2 ttl=64 time=0.088 ms

64 bytes from 127.0.0.1: icmp\_seq=3 ttl=64 time=0.106 ms

64 bytes from 127.0.0.1: icmp\_seq=4 ttl=64 time=0.102 ms

64 bytes from 127.0.0.1: icmp\_seq=5 ttl=64 time=0.093 ms

64 bytes from 127.0.0.1: icmp\_seq=6 ttl=64 time=0.093 ms

--- 127.0.0.1 ping statistics ---

6 packets transmitted, 6 received, 0% packet loss, time 173ms

rtt min/avg/max/mdev = 0.088/0.096/0.106/0.014 ms

เมื่อทำการสั่ง sudo ifconfig lo up จะสามารถเชื่อมไปยัง 127.0.0.1 ได้ซึ่ง 127.0.0.1 คือ IP ของ Loopback address (Localhost)

5. ใช้คำสั่ง ping เพื่อตรวจสอบการเชื่อมต่อไปยัง [www.google.com](http://www.google.com)

**ตอบ**

pi@raspberrypi:~ $ ping www.google.com

PING www.google.com (172.217.166.132) 56(84) bytes of data.

64 bytes from kul09s13-in-f4.1e100.net (172.217.166.132): icmp\_seq=1 ttl=112 time=45.6 ms

64 bytes from kul09s13-in-f4.1e100.net (172.217.166.132): icmp\_seq=2 ttl=112 time=24.3 ms

64 bytes from kul09s13-in-f4.1e100.net (172.217.166.132): icmp\_seq=3 ttl=112 time=78.0 ms

64 bytes from kul09s13-in-f4.1e100.net (172.217.166.132): icmp\_seq=4 ttl=112 time=27.0 ms

64 bytes from kul09s13-in-f4.1e100.net (172.217.166.132): icmp\_seq=5 ttl=112 time=38.6 ms

64 bytes from kul09s13-in-f4.1e100.net (172.217.166.132): icmp\_seq=6 ttl=112 time=26.6 ms

64 bytes from kul09s13-in-f4.1e100.net (172.217.166.132): icmp\_seq=7 ttl=112 time=32.0 ms

64 bytes from kul09s13-in-f4.1e100.net (172.217.166.132): icmp\_seq=8 ttl=112 time=28.5 ms

64 bytes from kul09s13-in-f4.1e100.net (172.217.166.132): icmp\_seq=9 ttl=112 time=24.4 ms

64 bytes from kul09s13-in-f4.1e100.net (172.217.166.132): icmp\_seq=10 ttl=112 time=117 ms

64 bytes from kul09s13-in-f4.1e100.net (172.217.166.132): icmp\_seq=11 ttl=112 time=105 ms

64 bytes from kul09s13-in-f4.1e100.net (172.217.166.132): icmp\_seq=12 ttl=112 time=84.3 ms

--- www.google.com ping statistics ---

12 packets transmitted, 12 received, 0% packet loss, time 27ms

rtt min/avg/max/mdev = 24.261/52.601/116.983/32.529 ms