

## Programing

### สำหรับ Devops

#### 1. จงอธิบายการทำงานของภาพข้างด้านล่างนี้อย่างละเอียด

```
lerdpisith_botnoi@redis-sme-develop:~$ ip a
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: ens4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1460 qdisc mq state UP group default qlen 1000
    link/ether 42:01:0a:00:01:01 brd ff:ff:ff:ff:ff:ff
    inet 10.0.1.7/32 metric 100 scope global dynamic ens4
        valid_lft 3048sec preferred_lft 3048sec
    inet6 fe80::4001:aff:fe00:107/64 scope link
        valid_lft forever preferred_lft forever
lerdpisith_botnoi@redis-sme-develop:~$
```

กรอบสีเขียว คำสั่ง ip เป็นคำสั่งสำหรับการจัดการเครือข่าย a หมายถึง address ใช้เพื่อแสดงข้อมูลเกี่ยวกับการตั้งค่าเครือข่ายทั้งหมดบนเครื่อง

กรอบสีเหลือง

- ens4 เป็น Network Interface Card สำหรับการเชื่อมต่อกับเครือข่ายภายนอก ภายใต้ "<>" บ่งบอกถึงสถานะของ Interface เช่น
  - Interface สามารถใช้งาน Broadcast และ Multicast ได้
  - UP หมายถึง Interface พร้อมใช้งาน
  - LOWER\_UP หมายถึง ระดับ Physical layer ของ Interface พร้อมใช้งาน
- mtu คือ Interface นี้สามารถส่งข้อมูลได้สูงสุด 1460 bytes ต่อ Frame ถ้าเกินก็จำเป็นต้องแยกข้อมูลออกเป็นอีก Frame
- qdisc mq คือ Queueing discipline เป็นกลไกที่ใช้ในการจัดการคิวของแพ็กเก็ตข้อมูลที่รอส่ง โดยใช้ qdisc mq
- state UP บ่งบอกว่า Interface นี้พร้อมใช้งาน
- group default เอาไว้จัดกลุ่ม Interface เพื่อให้ง่ายต่อการจัดการ Interface ซึ่ง Interface นี้จะอยู่ในกลุ่มเริ่มต้น
- qlen 1000 บ่งบอกจำนวนสูงสุดของแพ็กเก็ตข้อมูลที่สามารถอยู่ในคิวรอส่งได้สูงสุด 1000 packages

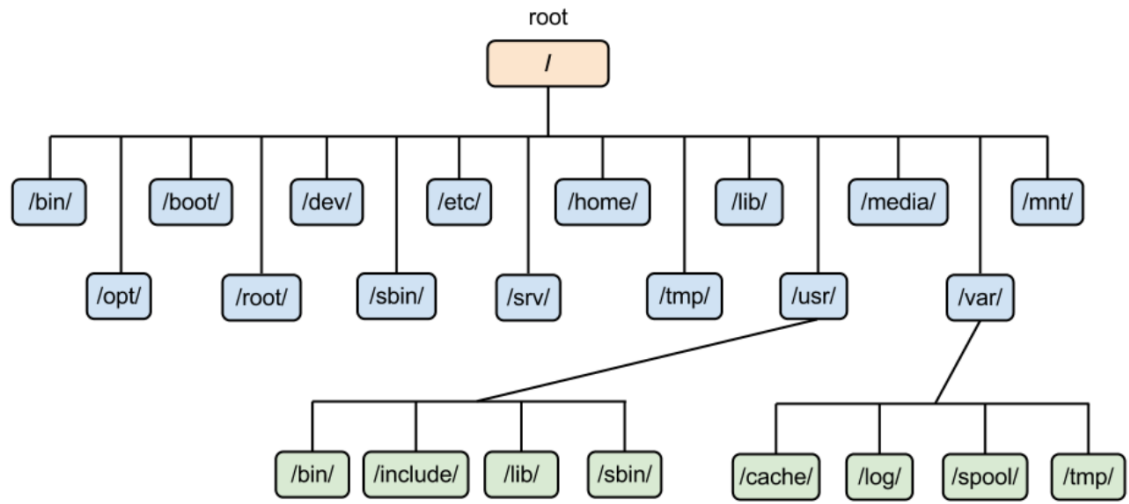
กรอบสีม่วง

- inet 10.0.1.7/32 คือ ipv4 ที่มี host ip เป็น 10.0.1.7 มี subnet mask เป็น 255.255.255.255 หมายความว่า IP นี้เป็น Private ip และมีเพียงเครื่องเดียวในเครือข่าย
- metric 100 เป็นค่าความสำคัญของเส้นทางที่ใช้ส่งข้อมูล
- scope global คือ ip นี้สามารถใช้งานได้ทุกเครือข่าย
- dynamic คือ ip นี้เป็น Dynamic ip ซึ่งถูกตั้งให้อัตโนมัติ
- ens4 คือ Network interface Card

ip 10.0.1.7/32 ค่าความสำคัญของเส้นทาง คือ 100 สามารถใช้งานได้ทั้งเครือข่ายซึ่ง ip นี้เป็นแบบ dynamic ip ที่ถูกกำหนดค่าให้โดยอัตโนมัติ

กรอบแดง เป็น ipv6 สามารถใช้งานได้แค่ในเครือข่ายเดียวกันเท่านั้น (Subnet mask เดียวกัน)

## 2. จงอธิบายการทำงานของภาพข้างด้านล่างนี้อย่างละเอียด

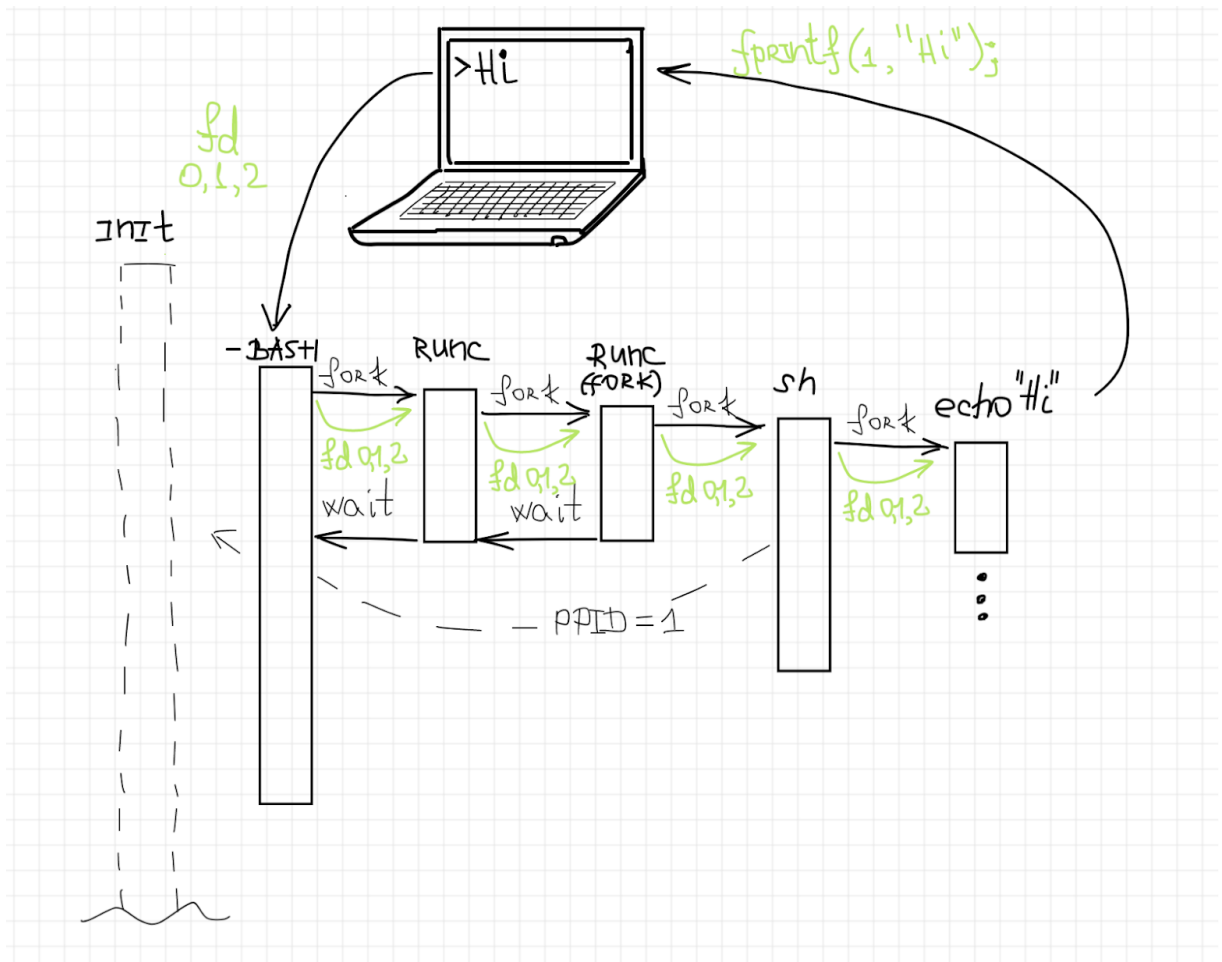


เป็นภาพโครงสร้างไดเรกทอรีแบบ Unix ซึ่ง Unix จะมองทุกอย่างเป็นไฟล์

- / (root) เป็นจุดเริ่มต้นของระบบไฟล์ เป็นไดเรกทอรีหลักที่รวมทุกไดเรกทอรีเข้าด้วยกัน
- /bin (binary) เป็นที่เก็บโปรแกรมที่จำเป็นสำหรับการทำงานของระบบปฏิบัติการ เช่น ls, cd, cp, rm, cat, mkdir และอื่นๆ ส่วนใหญ่อยู่ในรูปแบบของ binary files
- /boot เป็นที่เก็บไฟล์ที่จำเป็นสำหรับการบูตระบบ เช่น เคอร์เนล (kernel) ของระบบปฏิบัติการ, โหลดเดอร์ (loader) และ configuration file ที่เกี่ยวข้องกับการบูต
- /dev (device) เป็นที่เก็บไฟล์ devices ต่างๆ ในระบบ /dev จะมีไฟล์เป็นตัวแทนของอุปกรณ์ฮาร์ดแวร์ที่ระบบปฏิบัติการสามารถเข้าถึงได้ และยังสามารถควบคุมอุปกรณ์ฮาร์ดแวร์ได้โดยตรงผ่านไฟล์ได้
- /etc (etc.) เป็นที่เก็บไฟล์ configuration ต่างๆ ของระบบ เช่น ไฟล์กำหนดค่าเครือข่าย, ไฟล์ service configuration ต่างๆ (เช่น Apache, MySQL, NGINX), และไฟล์ user configuration
- /home เป็นที่เก็บไดเรกทอรีส่วนตัวของผู้ใช้แต่ละคน โดยชื่อไดเรกทอรีจะตรงกับชื่อผู้ใช้ เช่น /home/user1, /home/user2 เป็นต้น ในไดเรกทอรีส่วนนี้ ผู้ใช้สามารถสร้างไฟล์และไดเรกทอรีของตนเองได้
- /lib (library) เป็นที่เก็บไลบรารีต่างๆ ที่โปรแกรมต่างๆ ต้องใช้ในการทำงาน ไลบรารีเหล่านี้เป็นชุดของฟังก์ชันที่ถูกนำมาใช้ซ้ำในโปรแกรมต่างๆ เพื่อลดขนาดของโปรแกรมและเพิ่มประสิทธิภาพ

- /media เป็นที่เก็บจุดเชื่อมต่อของอุปกรณ์เก็บข้อมูลภายนอก เช่น แฟลชไดรฟ์, ฮาร์ดดิสก์ภายนอก เมื่อเราเสียบอุปกรณ์เหล่านี้เข้ากับคอมพิวเตอร์ ระบบปฏิบัติการจะสร้างไคเร็กทอรีย่อยขึ้นมาใน /media เพื่อให้เราสามารถเข้าถึงข้อมูลในอุปกรณ์เหล่านั้นได้
- /mnt (mount) เป็นที่เก็บจุดเชื่อมต่อของระบบไฟล์อื่นๆ ที่ถูก mount เข้ามา เช่น ระบบไฟล์เครือข่าย หรือพาร์ติชันอื่นๆ ของฮาร์ดดิสก์
- /opt (optional) เป็นที่เก็บโปรแกรมที่ติดตั้งเพิ่มเติม เช่น โปรแกรมที่ไม่ได้ติดตั้งมาพร้อมกับระบบปฏิบัติการ
- /root ไคเร็กทอรีส่วนตัวของ root user (ผู้ดูแลระบบ)
- /sbin (system binary) เป็นที่เก็บโปรแกรมสำหรับผู้ดูแลระบบ เช่น fdisk, mkfs, shutdown เป็นต้น
- /srv (server, service) เป็นที่เก็บข้อมูลที่เกี่ยวข้องกับบริการต่างๆ เช่น เว็บเซิร์ฟเวอร์, เซิร์ฟเวอร์ FTP
- /tmp (temp) เป็นที่เก็บไฟล์ชั่วคราวที่โปรแกรมต่างๆ สร้างขึ้นขณะทำงาน ไฟล์เหล่านี้จะถูกลบออกโดยอัตโนมัติเมื่อระบบรีบูตหรือโปรแกรมที่สร้างไฟล์เหล่านั้นสิ้นสุดการทำงาน
- /usr (unix system resources) เป็นที่เก็บไฟล์ต่างๆ ที่เกี่ยวข้องกับผู้ใช้ เช่น โปรแกรม, ไลบรารี, และเอกสาร
  - /usr/bin: บรรจุโปรแกรมสำหรับผู้ใช้ทั่วไป
  - /usr/include: บรรจุไฟล์ส่วนหัว (header files) สำหรับการพัฒนาโปรแกรม
  - /usr/lib: บรรจุไลบรารีสำหรับโปรแกรม
  - /usr/sbin: บรรจุโปรแกรมสำหรับผู้ดูแลระบบ
- /var (variable) เป็นที่เก็บไฟล์ที่มีการเปลี่ยนแปลงบ่อย เช่น log ไฟล์, spool ไฟล์, และไฟล์ฐานข้อมูล
  - /var/cache เป็นที่เก็บไฟล์แคชที่โปรแกรมต่างๆ ใช้เพื่อเพิ่มความเร็วในการทำงาน
  - /var/log: เป็นที่เก็บ log ไฟล์ที่บันทึกกิจกรรมต่างๆ ของระบบ
  - /var/spool: เป็นที่เก็บไฟล์ที่รอการประมวลผล เช่น ไฟล์พิมพ์
  - /var/tmp: เป็นที่เก็บไฟล์ชั่วคราวที่โปรแกรมต่างๆ สร้างขึ้น

### 3. จงอธิบายการทำงานของภาพข้างด้านล่างนี้อย่างละเอียดและเป็นการทำงานของอะไร



ผมคิดว่ารูปภาพนี้เป็นการใช้ bash สร้าง Environment เพื่อแสดงข้อความ "Hi" โดยใช้ sh เรียกคำสั่ง echo "Hi" เขียนไปที่ไฟล์ index ที่ 1 ของ fd ซึ่งเป็นไฟล์ของ Standard output

Bash เรียกใช้คำสั่ง runc เพื่อจัดการกับโปรเซสที่ต้องการรันภายในคอนเทนเนอร์ โดย runc จะทำการสร้างคอนเทนเนอร์ใหม่และแยกทรัพยากรออกจากระบบปฏิบัติการหลัก fork() จะถูกเรียกเพื่อสร้าง child process ในคอนเทนเนอร์ ภายในขั้นตอน runc(fork) จะสร้าง sh process ซึ่ง sh จะรันคำสั่ง echo "Hi" เป็น child process

โปรเซสลูกเหล่านี้จะใช้ file descriptors จากโปรเซสพ่อแม่เพื่อจัดการ I/O โดย output ของคำสั่งจะถูกส่งไปที่ stdout (fd 1) ซึ่งจะแสดงผลที่หน้าจอ

ในรูปภาพ bash จะรอ runc และ runc(fork) processes ถ้า process เหล่านี้หยุดลงก่อน sh process จะหยุดลง sh process จะไม่มี parents process ดังนั้น sh process จะส่ง PPID = 1 ไปที่ init เพื่อให้ init ดูแลโปรเซสนั้นต่อ เพื่อไม่ให้เกิด Zombie process