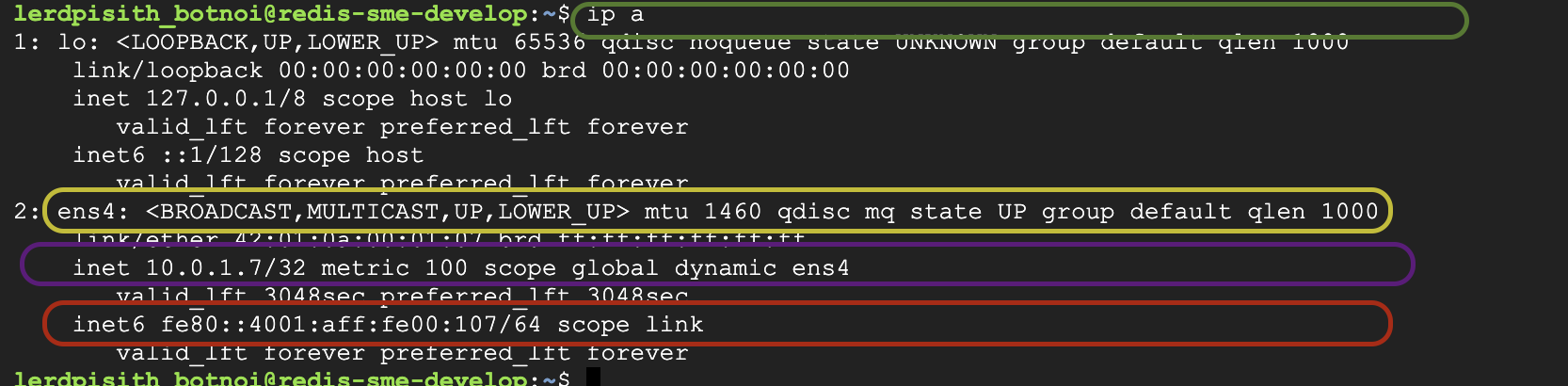
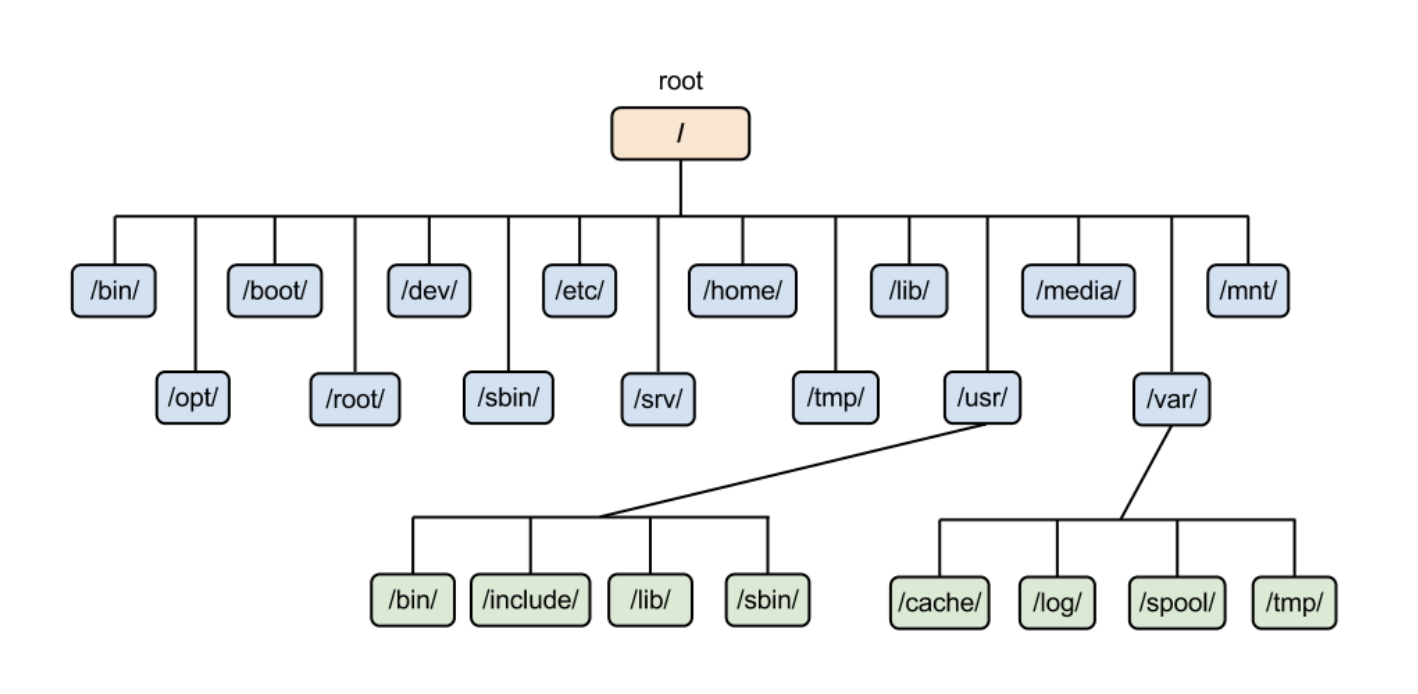
1. **จงอธิบายการทำงานของภาพข้างด้านล่างนี้อย่างละเอียด**

****

จากภาพ วงสีเขียวเป็นการใช้คำสั่ง ip a ที่เป็นคำสั่งในการเรียกดู ip address ที่จะแสดง network interface ทั้งหมดโดยจากภาพจะมีอยู่ 2 interface วงสีเหลืองเป็น interface ที่ 2 ที่มีชื่อคือ ens4 มาจาก en : ethernet และ s4 : PCI Express device slot 4 <BROADCAST, MULTICAST, UP, LOWER\_UP> เป็นการบอกสถานะต่างๆ โดย BROADCAST คือ interface นี้สามารถกระจายข้อมูลให้ device อื่นใน network นี้ได้ MULTICASTING คือ สามารถส่งข้อมูลให้เป็นกลุ่มๆ ใน network เดียวกันได้ UP คือ interface นี้ใช้ได้อยู่สามารถเชื่อมต่อกับ network ควบคุมโดยผู้ดูแลระบบ และเป็น software level LOWER\_UP คือ physical link เชื่อมต่ออยู่ เป็น hardware level ถัดมา mtu 1460 คือ maximum transmission unit หมายถึงขนาด packet ที่ใหญ่ที่สุดที่ส่งได้จะเท่ากับ 1460 bytes qdisc mq หรือ queuing discipline หรือวิธีการเข้าคิวเป็น multi queue จะเป็นการเข้าคิวที่สามารถให้คิว packets มาพร้อมกันได้ state UP คือ state ของ interface นี้ใช้งานได้อยู่สามารถเชื่อมกับ network ได้ group default คือ การจัดหมวดหมู่ของ interface ในที่นี้คือ default สุดท้าย qlen 1000 คือ จำนวนความยาวของคิวสำหรับ interface นี้ ซึ่งคือ 1000 packets วงสีม่วงบ่งบอกถึง IPv4 ที่มี address 10.0.1.7 และมี /32 subnet mask หมายถึงมี 10.0.1.7 เป็น ip เฉพาะ ip เดียวใน network นี้ metric 100 คือการบอกถึงความสำคัญให้กับ routing protocols ว่า interface นี้มีความสำคัญมากน้อยเท่าไหร่ ต้องให้ความสำคัญกับอะไรก่อน โดยยิ่งค่าน้อยยิ่งมีความสำคัญมาก scope global เป็นการบอกถึง scope ของ ip นี้ว่าสามารถใช้ได้แบบ global คือใช้นอก local network ด้วยก็ได้ dynamic บอกว่า ip นี้กำหนดแบบ dynamic คือสามารถเปลี่ยนแปลงได้ สุดท้าย ens4 เป็นชื่อของ interface โดย en คือ ethernet s4 คือ PCI Express device slot 4 วงสีแดงบอกถึง IPv6 ที่มี address fe80::4001:aff:fe00:107 โดย fe80 เป็น prefix บอกว่าเป็น link local address สามารถใช้ได้ภายใน local :: เป็นการเขียนย่อย โดย ip เต็มๆคือ fe80:0000:0000:0000 และ 4001:0aff:fe00:107 เป็นชื่อ interface และ /64 คือ prefix length โดยบอกว่า 64 bits แรกใช้สำหรับระบุ network portion สุดท้าย scope link เป็นการบอกถึง scope ของ ip นี้ว่าสามารถใช้ได้กับ local network เท่านั้น

1. **จงอธิบายการทำงานของภาพข้างด้านล่างนี้อย่างละเอียด**

****

จากภาพนี้แสดงให้เห็นถึงระบบการจัดเก็บ file ของ linux ที่เป็นแบบ hierarchical tree structure โดยที่จะมี node แรกที่เป็น root directory จะมี path คือ / ที่จะแยกย่อยออกมาได้อีกหลาย directory หรือที่เรียกว่า subdirectory คือ /bin /boot /dev /etc /home /lib /media /mnt /opt /root /sbin /srv /tmp /usr /var โดยแต่ละ directory จะอยู่ภายใต้ parent node ซึ่งคือ root ก็จะมี / นำหน้า โดยแต่ละ directory จะมีหน้าที่แตกต่างกันดังนี้

/bin เก็บ binary ที่ executable หรือ ไฟล์ binary ที่เอาไว้รัน program ต่างๆ

/boot เก็บไฟล์ที่ใช้สำหรับ boot ระบบ

/dev เก็บ device ไฟล์เช่น driver ของ keyboard หรือ hardware ต่างๆ

/etc เก็บการตั้งค่าของระบบ config ไฟล์ต่างๆ

/home เก็บ directory สำหรับ users ต่างๆ เช่น /home/john

/lib เก็บ library ต่างๆสำหรับ binary ที่ต้องการ

/media เก็บไฟล์ของพวก usb drives, cd

/mnt เก็บไฟล์ที่ mount จากแหล่งอื่นมาเก็บไว้ที่ directory นี้ได้

/opt เก็บ optional software packages และ third-party applications

/root เก็บ directory สำหรับ root (superuser)

/sbin เก็บ binary ที่ executable สำหรับ system หรือ ไฟล์ที่สำคัญสำหรับระบบ

/srv เก็บ services ไฟล์ที่ได้มาจากระบบ

/tmp เก็บ temporary ไฟล์ หรือ ที่เก็บไฟล์ชั่วคราว

/usr เก็บไฟล์ที่ได้จากการ install software

/var เก็บไฟล์ที่ใช้เป็นตัวแปร หรือ variable ที่มีการเปลี่ยนแปลงบ่อยๆ เช่น logs, email inboxes ฯลฯ

และจะสังเกตุได้ว่า usr จะมี subdirectory เป็น /bin /include /lib /sbin โดยถ้าต้องการจะเข้าถึง directory นี้จะต้องเอา path ของ parent node มา ซึ่งคือ usr จะทำให้ได้ directory ต่างๆที่จะเข้าถึงคือ /usr/bin /usr/include /usr/lib /usr/sbin และ var มี subdirectory เป็น /cache /log /spool /tmp ก็เช่นเดียวกัน จะได้ /var/cache /var/log /var/spool /var/tmp โดยแต่ละ directory ที่ยังไม่ได้อธิบาย คือจะมี

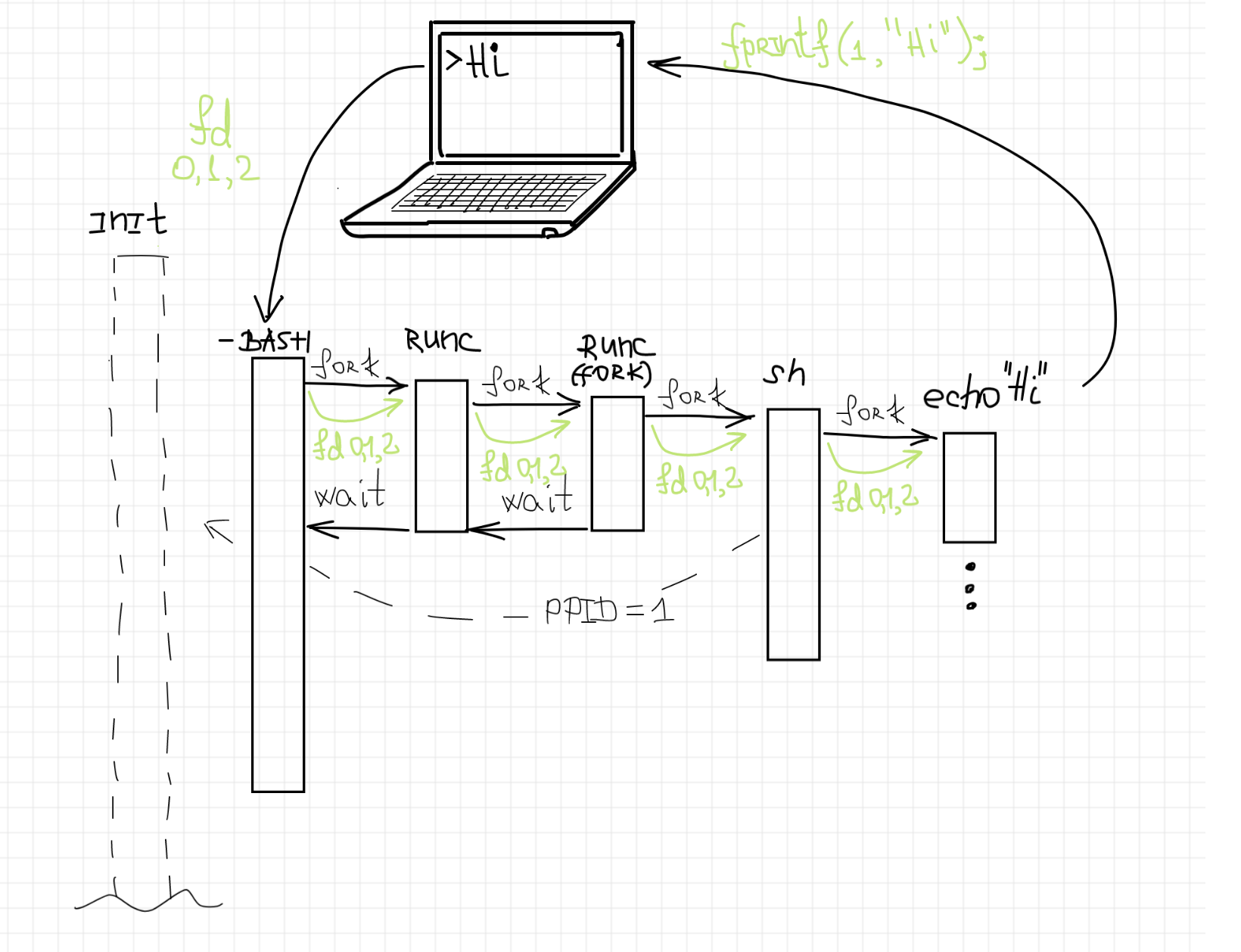
/include เก็บ header ไฟล์ของ C compiler

/cache เก็บ cache ที่ได้จาก application

/log เก็บ log files ว่าทำอะไรไปบ้าง

/spool เก็บไฟล์ที่รอ process หรือไฟล์ที่อยู่ใน queue

1. **จงอธิบายการทำงานของภาพข้างด้านล่างนี้อย่างละเอียดและเป็นการทำงานของอะไร**

****

จากภาพเป็นแผนภาพการแสดงถึงขั้นตอนวิธีการ execute command โดย execute ใน container ด้วย โดยมีขั้นตอนดังนี้ ขั้นแรกเป็น terminal ของ laptop และ user มีการส่ง Hi ลงไปใน shell ในที่นี้ของ user คือ bash โดยมีการกำหนด fd หรือ file descriptors 0, 1, 2 ซึ่งแต่ละค่าหมายถึง 0 คือ Standard input 1 คือ Standard output 2 คือ Standard error ทั้งหมดใช้เพื่อให้ process รับมือกับ input output error ได้ ว่าจะต้องได้อะไรบ้าง หลังจากนั้น bash ทำการ fork คือการสร้าง process ลูกในการรัน runc รวมไปถึงส่ง file descriptors ต่อไปด้วย และจะเห็นว่ามี wait ด้วย หมายความว่า process bash ที่เรียกจะต้องรอให้ process ลูกรันจนจบก่อนจึงจะไปคำสั่งถัดไป ถัดมา runc fork runc(fork) หมายถึง runc เรียก process ลูกอีก process แล้วก็ส่งต่อ file descriptors กับ wait ที่ต้องให้ process runc(fork) เสร็จก่อนถึงจะจบ หรือ ทำอย่างอื่นต่อได้ ถัดมาใน containers (ที่ได้จากการใช้คำสั่ง runc) ก็มีการเรียก process ลูกอีกที ในการเปิด sh เพื่อรับและรันคำสั่งต่างๆ และก็ยังคงส่ง file descriptors มาต่อ แต่ runc จะส่งคำสั่งที่ต้อง execute ใน sh มาด้วยคือคำสั่งในการแสดง Hi เมื่อ runc ทำหน้าที่ของมันเสร็จคือสร้าง process ลูกในการรันคำสั่งตัว runc จะ terminate ตัวเอง แล้วจะค่อยๆ terminate parent process ที่เรียกมันเพื่อบริหารทรัพยากร ตอนนี้จะทำให้ sh ไม่มี PPID เนื่องจากตัวที่เรียก terminate ไปแล้ว ระบบจะเซ็ต PPID ให้เป็น 1 ซึ่งคือถูกเรียกโดยตัวเริ่มต้นของระบบ ทำให้ไม่กลายเป็น zombie process เพราะจะมีตัว INIT ที่คอยรอ exit status ถัดมา sh ก็จะเรียก process ลูกในการรันคำสั่ง echo “Hi” และคำสั่งอื่นๆหากมีเพิ่ม สุดท้ายก็จะส่งผลลัพธ์ที่ได้ออกไป จากในแผนภาพก็จะส่ง fprintf(1,”Hi”) ที่บอก file descriptors เป็น 1 คือแสดงแค่ Standard Output เป็น “Hi” ก็จะสิ้นสุดในการ execute คำสั่งนี้ หน้า terminal ก็จะรอคำสั่งถัดไป