

# CS222

# Operating System

## Lab 1: Linux Practice

1<sup>st</sup> Semester 2024

Kasidit Chanchio

Department of Computer Science

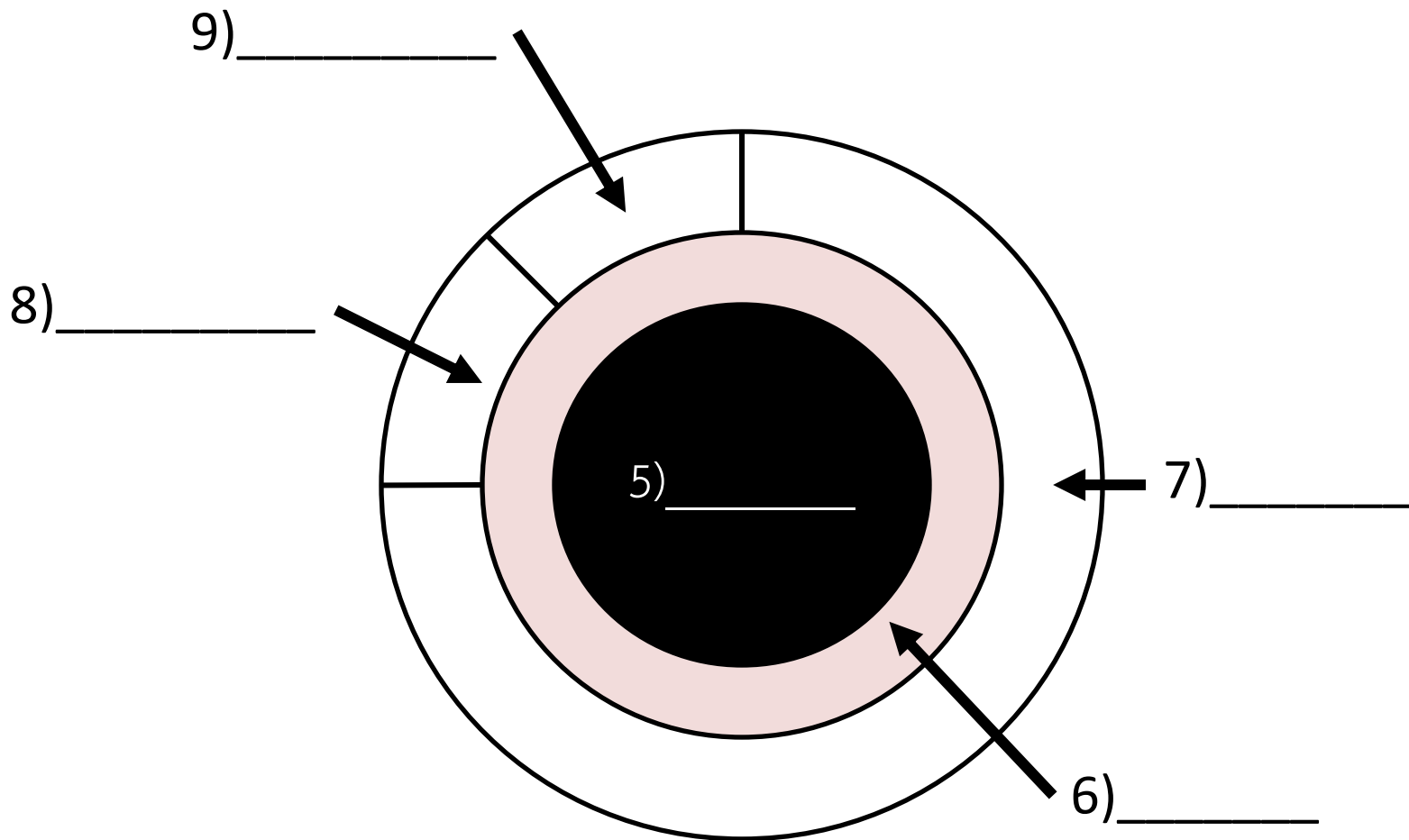
Faculty of Science and Technology

Thammasat University

# ตอบคำถาม UNIX/Linux แบบสั้น

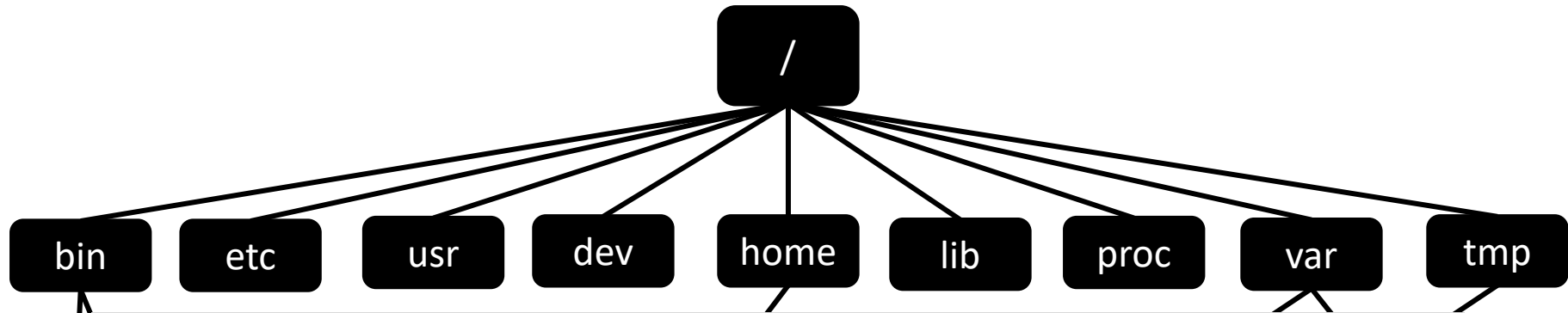
1. ระบบ UNIX เป็น OS แบบใด
2. ใครคือผู้พัฒนาระบบ UNIX
3. ระบบ Linux คืออะไร
4. ใครคือผู้พัฒนาระบบ Linux

จงเติมคำต่อไปนี้



# ทดลองทำ

- ให้ login เข้าสู่เครื่อง Ubuntu linux
- Username: Room107
- Password: cstu107
- ผู้ใช้จะมี account ของตนและจะเริ่มทำงานใน home directory
  - สมมติว่า home ของ นศ คือ Room107
  - ให้สมมติว่าไดเรกทอรี user1 ใน Slide คือ Room107 ของเครื่องของ นศ



bas

```
$ mkdir mydir
```

```
$ cd mydir
```

```
$ touch myfile1
```

```
$ touch ../myfile2
```

```
$ cp /etc/hosts myfile3
```

```
$ mv ../myfile2 /tmp/myfile2
```

```
$ ls -l → ทำอะไร 10) _____
```

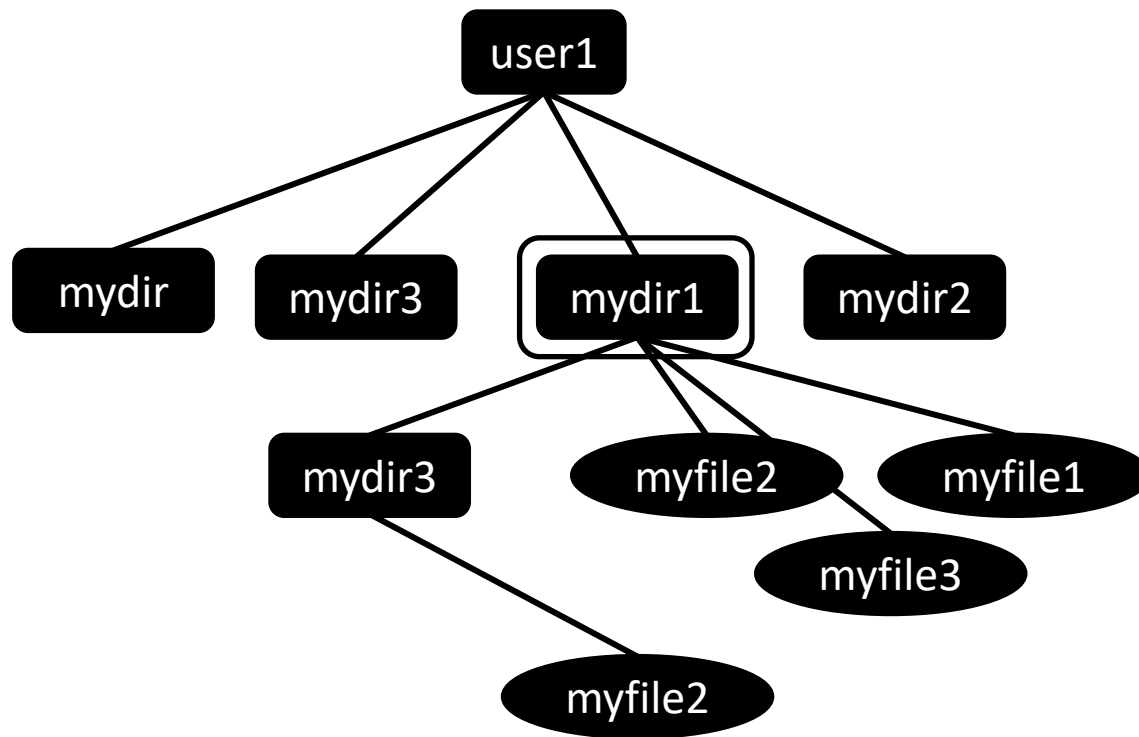
```
$ cat myfile3 → ทำอะไร 11) _____
```

```
$ ls .. → ทำอะไร 12) _____
```

```
$ ถ้าจะย้าย myfile3 ไปไว้ที่ .. จะต้องออกคำสั่งอะไร 13) _____
```

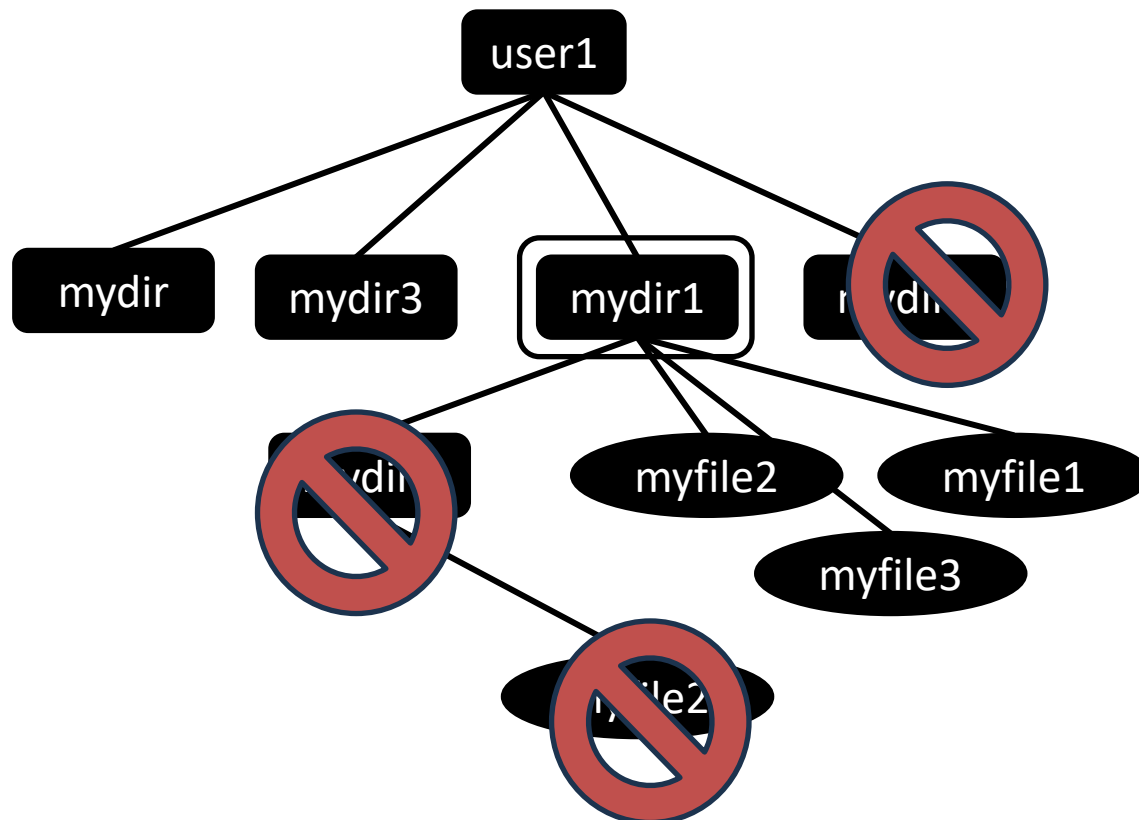
# ทดสอบ

14) สมมติให้ User1 คือ login user Room107 ในเครื่องที่ นศ ใช้  
หรือเครื่องของ นศ จงสร้างโครงสร้าง directory ในภาพ



# ทดสอบ

15) จงแสดงคำสั่งเพื่อลบ ไดเรกทอรีและไฟล์ในภาพ



# ทดสอบ

16. จงใช้คำสั่ง cat เพื่อสร้างไฟล์ชื่อ mytextfile
17. จงใช้คำสั่ง echo เพื่อพิมพ์ “hello” ตามด้วย วันที่และเวลาปัจจุบัน แล้ว.  
ตามด้วย “cs222”
18. จงสร้าง directory ~/sub1/sub2 และใช้คำสั่ง nano สร้างไฟล์ file2

```
GNU nano 5.4 file2
fruit unit_price amount staff
orange 50 20 somchai
banana 100 7 natty
apple 200 30 somnuk
mango 60 20 alice
apple 130 100 bob
orange 10 3000 john
```



คำตอบ

```
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line
```



## ทดสอบ

19. จงตอบว่า pipe คืออะไร มีประโยชน์อย่างไร
20. จงทำตาม slide ต่อไปนี้ และนำคำตอบสุดท้าย ผลลัพธ์ของ  
cat abcd2.txt ส่ง

```
[user1@vm1:~/sub2/sub1$  
[user1@vm1:~/sub2/sub1$ ls  
file1  file2  
[user1@vm1:~/sub2/sub1$ cat file2  
fruit  unit_price  amount  staff  
orange 50          20      somchai  
banana 100         7        natty  
apple 200         30       somnuk  
mango 60         20       alice  
apple 130        100      bob  
orange 10        3000     john  
[user1@vm1:~/sub2/sub1$ cat file2 | sed "1d"  
orange 50          20      somchai  
banana 100         7        natty  
apple 200         30       somnuk  
mango 60         20       alice  
apple 130        100      bob  
orange 10        3000     john  
[user1@vm1:~/sub2/sub1$ cat file2 | sed "1d" | wc -l  
6  
[user1@vm1:~/sub2/sub1$ cat file2 | sed "1d" | grep apple  
apple 200         30       somnuk  
apple 130        100      bob  
[user1@vm1:~/sub2/sub1$ cat file2 | sed "1d" | cut -d' ' -f1  
orange  
banana  
apple  
mango  
apple  
orange  
user1@vm1:~/sub2/sub1$
```

```
[user1@vm1:~/sub2/sub1$
```

```
[user1@vm1:~/sub2/sub1$ ls
```

```
file1  file2
```

```
[user1@vm1:~/sub2/sub1$ cat file2
```

fruit	unit_price	amount	staff
orange	50	20	somchai
banana	100	7	natty
apple	200	30	somnuk
mango	60	20	alice
apple	130	100	bob
orange	10	3000	john

```
[user1@vm1:~/sub2/sub1$
```

```
[user1@vm1:~/sub2/sub1$ ls
```

```
file1  file2
```

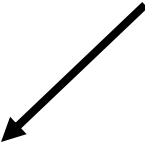
```
[user1@vm1:~/sub2/sub1$ cat file2
```

fruit	unit_price	amount	staff
orange	50	20	somchai
banana	100	7	natty
apple	200	30	somnuk
mango	60	20	alice
apple	130	100	bob
orange	10	3000	john

```
[user1@vm1:~/sub2/sub1$ cat file2 | sed "1d"
```

orange	50	20	somchai
banana	100	7	natty
apple	200	30	somnuk
mango	60	20	alice
apple	130	100	bob
orange	10	3000	john

ตัด 1 บรรทัดออก



```
[user1@vm1:~/sub2/sub1$
```

```
[user1@vm1:~/sub2/sub1$ ls
```

```
file1  file2
```

```
[user1@vm1:~/sub2/sub1$ cat file2
```

```
fruit  unit_price  amount  staff
orange  50           20     somchai
banana  100           7       natty
apple  200           30     somnuk
mango  60            20     alice
apple  130          100     bob
orange  10          3000    john
```


```
[user1@vm1:~/sub2/sub1$ cat file2 | sed "1d"
```

```
orange  50           20     somchai
banana  100           7       natty
apple  200           30     somnuk
mango  60            20     alice
apple  130          100     bob
orange  10          3000    john
```

```
[user1@vm1:~/sub2/sub1$ cat file2 | sed "1d" | wc -l
```

```
6
```

นับจำนวน บรรทัด  
ของ output



```
[user1@vm1:~/sub2/sub1$
```

```
[user1@vm1:~/sub2/sub1$ ls
```

```
file1  file2
```

```
[user1@vm1:~/sub2/sub1$ cat file2
```

fruit	unit_price	amount	staff
orange	50	20	somchai
banana	100	7	natty
apple	200	30	somnuk
mango	60	20	alice
apple	130	100	bob
orange	10	3000	john

```
[user1@vm1:~/sub2/sub1$ cat file2 | sed "1d"
```

orange	50	20	somchai
banana	100	7	natty
apple	200	30	somnuk
mango	60	20	alice
apple	130	100	bob
orange	10	3000	john

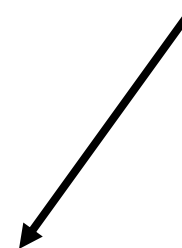
```
[user1@vm1:~/sub2/sub1$ cat file2 | sed "1d" | wc -l
```

```
6
```

```
[user1@vm1:~/sub2/sub1$ cat file2 | sed "1d" | grep apple
```

<b>apple</b>	200	30	somnuk
<b>apple</b>	130	100	bob

พิมพ์เฉพาะบรรทัดที่  
มี string apple

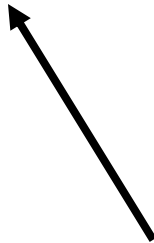


```

[user1@vm1:~/sub2/sub1$ cat file2 | sed "1d"
orange  50          20      somchai
banana  100         7       natty
apple   200         30      somnuk
mango   60          20      alice
apple   130         100     bob
orange  10          3000    john
[user1@vm1:~/sub2/sub1$ cat file2 | sed "1d" | wc -l
6
[user1@vm1:~/sub2/sub1$ cat file2 | sed "1d" | grep apple
apple 200         30      somnuk
apple 130         100     bob
[user1@vm1:~/sub2/sub1$ cat file2 | sed "1d" | cut -d' ' -f1
orange
banana
apple
mango
apple
orange
user1@vm1:~/sub2/sub1$ █

```

ใช้ ' ' ช่องว่าง เป็นตัวแยก  
 Field แลพพิมพ์เฉพาะ field ที่ 1  
 ของแต่ละบรรทัด



```
user1@vm1:~/sub2/sub1$ cat file2 | sed "1d" | cut -d' ' -f1 | sort
```

```
apple  
apple  
banana  
mango  
orange  
orange
```

เรียงลำดับจากน้อยไปมาก





```
user1@vm1:~/sub2/sub1$ cat file2 | sed "1d" | cut -d' ' -f1 | sort
```

apple

apple

banana

mango

orange

orange

```
user1@vm1:~/sub2/sub1$ cat file2 | sed "1d" | cut -d' ' -f1 | sort | uniq
```

apple

banana

mango

orange

ตัดคำซ้ำในบรรทัด  
ที่ติดกันออก



```
user1@vm1:~/sub2/sub1$ cat file2 | sed "1d" | cut -d' ' -f1 | sort
```

apple

apple

banana

mango

orange

orange

```
user1@vm1:~/sub2/sub1$ cat file2 | sed "1d" | cut -d' ' -f1 | sort | uniq
```

apple

banana

mango

orange

```
user1@vm1:~/sub2/sub1$ cat file2 | sed "1d" | cut -d' ' -f1 | sort | uniq | wc -l
```

4

```
user1@vm1:~/sub2/sub1$ cat file2 | sed "1d" | cut -d' ' -f1 | sort
```

```
apple
```

```
apple
```

```
banana
```

```
mango
```

```
orange
```

```
orange
```

```
user1@vm1:~/sub2/sub1$ cat file2 | sed "1d" | cut -d' ' -f1 | sort | uniq
```

```
apple
```

```
banana
```

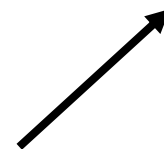
```
mango
```

```
orange
```

```
user1@vm1:~/sub2/sub1$ cat file2 | sed "1d" | cut -d' ' -f1 | sort | uniq | wc -l
```

```
4
```

```
user1@vm1:~/sub2/sub1$ cat file2 | sed "1d" | cut -d' ' -f1 | sort | uniq > abcd.txt
```



Redirect output

ไปไว้ในไฟล์ abcd.txt

```
user1@vm1:~/sub2/sub1$ cat file2 | sed "1d" | cut -d' ' -f1 | sort
```

```
apple  
apple  
banana  
mango  
orange  
orange
```

```
user1@vm1:~/sub2/sub1$ cat file2 | sed "1d" | cut -d' ' -f1 | sort | uniq
```

```
apple  
banana  
mango  
orange
```

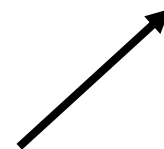
```
user1@vm1:~/sub2/sub1$ cat file2 | sed "1d" | cut -d' ' -f1 | sort | uniq | wc -l
```

```
4
```

```
user1@vm1:~/sub2/sub1$ cat file2 | sed "1d" | cut -d' ' -f1 | sort | uniq > abcd.txt
```

```
user1@vm1:~/sub2/sub1$ sort -r < abcd.txt
```


```
orange  
mango  
banana  
apple
```



Redirect output

ไปไว้ในไฟล์ abcd.txt

```
user1@vm1:~/sub2/sub1$ cat file2 | sed "1d" | cut -d' ' -f1 | sort
apple
apple
banana
mango
orange
orange
user1@vm1:~/sub2/sub1$ cat file2 | sed "1d" | cut -d' ' -f1 | sort | uniq
apple
banana
mango
orange
user1@vm1:~/sub2/sub1$ cat file2 | sed "1d" | cut -d' ' -f1 | sort | uniq | wc -l
4
user1@vm1:~/sub2/sub1$ cat file2 | sed "1d" | cut -d' ' -f1 | sort | uniq > abcd.txt
user1@vm1:~/sub2/sub1$ sort -r < abcd.txt
orange
mango
banana
apple
user1@vm1:~/sub2/sub1$ cat abcd.txt | sort -r | tee abcd2.txt | wc -l
4
```



เรียงจากมากไปน้อย

พิมพ์ไฟล์ออก stdout  
และเก็บใน abcd2.txt

```
user1@vm1:~/sub2/sub1$ cat file2 | sed "1d" | cut -d' ' -f1 | sort
apple
apple
banana
mango
orange
orange
user1@vm1:~/sub2/sub1$ cat file2 | sed "1d" | cut -d' ' -f1 | sort | uniq
apple
banana
mango
orange
user1@vm1:~/sub2/sub1$ cat file2 | sed "1d" | cut -d' ' -f1 | sort | uniq | wc -l
4
user1@vm1:~/sub2/sub1$ cat file2 | sed "1d" | cut -d' ' -f1 | sort | uniq > abcd.txt
user1@vm1:~/sub2/sub1$ sort -r < abcd.txt
orange
mango
banana
apple
user1@vm1:~/sub2/sub1$ cat abcd.txt | sort -r | tee abcd2.txt | wc -l
4
user1@vm1:~/sub2/sub1$ cat abcd2.txt
orange
mango
banana
apple
user1@vm1:~/sub2/sub1$ █
```



คำตอบ

# ทดสอบ

21. จงตอบว่า permission flag ของไฟล์ คืออะไร

1. 755 หมายถึงอะไร
2. 700 หมายถึงอะไร
3. 000 หมายถึงอะไร

22. จงทำตาม slide ต่อไปนี้ และ copy ผลลัพธ์ของคำสั่ง

```
$ ls -l abcd.txt
```

ใส่ลงในคำตอบ

```
kasidit@enterprise:~/Desktop/CS222$ cp /etc/hosts abcd.txt
kasidit@enterprise:~/Desktop/CS222$
kasidit@enterprise:~/Desktop/CS222$ ls -l
total 4
-rw-r--r-- 1 kasidit kasidit 225 ม.ค. 31 14:31 abcd.txt
kasidit@enterprise:~/Desktop/CS222$
kasidit@enterprise:~/Desktop/CS222$ chmod 444 abcd.txt
kasidit@enterprise:~/Desktop/CS222$ ls -l
total 4
-r--r--r-- 1 kasidit kasidit 225 ม.ค. 31 14:31 abcd.txt
kasidit@enterprise:~/Desktop/CS222$
kasidit@enterprise:~/Desktop/CS222$ rm abcd.txt
rm: remove write-protected regular file 'abcd.txt'? n
kasidit@enterprise:~/Desktop/CS222$ chmod 000 abcd.txt
kasidit@enterprise:~/Desktop/CS222$ ls -l
total 4
----- 1 kasidit kasidit 225 ม.ค. 31 14:31 abcd.txt
kasidit@enterprise:~/Desktop/CS222$
kasidit@enterprise:~/Desktop/CS222$ rm abcd.txt
rm: remove write-protected regular file 'abcd.txt'? y
kasidit@enterprise:~/Desktop/CS222$ ls -l
total 0
kasidit@enterprise:~/Desktop/CS222$
```



คำตอบ



## ทดสอบ

23. จงทำตาม slide เกี่ยวกับ redirection ต่อไปนี้  
และ copy เนื้อหาของ content2.txt ใส่ลงในคำตอบ

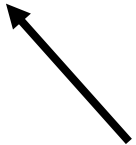
```
[user1@vm1:~$ ls
mydir  mydir2  sub2
[user1@vm1:~$
[user1@vm1:~$ ls mydir
link3  link4  myfile1  myfile3
[user1@vm1:~$ cat mydir/myfile3 > content1.txt
[user1@vm1:~$ ls -l content1.txt
-rw-rw-r-- 1 user1 user1 218 Aug 25 14:47 content1.txt
```



Redirect stdout

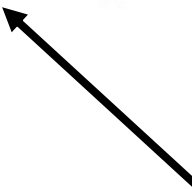
ไปเก็บใน content1.txt

```
[user1@vm1:~$ ls
mydir  mydir2  sub2
[user1@vm1:~$
[user1@vm1:~$ ls mydir
link3  link4  myfile1  myfile3
[user1@vm1:~$ cat mydir/myfile3 > content1.txt
[user1@vm1:~$ ls -l content1.txt
-rw-rw-r-- 1 user1 user1 218 Aug 25 14:47 content1.txt
[user1@vm1:~$
[user1@vm1:~$ cat mydir > content2.txt
cat: mydir: Is a directory
```



ผมพยายามที่จะ cat ไดเรกทอรี  
ซึ่งจะเกิด error และจะนำ  
สิ่งที่ปรากฏบนหน้าจอไปไว้ในไฟล์ content2.txt

```
[user1@vm1:~$ ls
mydir  mydir2  sub2
[user1@vm1:~$
[user1@vm1:~$ ls mydir
link3  link4  myfile1  myfile3
[user1@vm1:~$ cat mydir/myfile3 > content1.txt
[user1@vm1:~$ ls -l content1.txt
-rw-rw-r-- 1 user1 user1 218 Aug 25 14:47 content1.txt
[user1@vm1:~$
[user1@vm1:~$ cat mydir > content2.txt
cat: mydir: Is a directory
[user1@vm1:~$ ls -l content2.txt
-rw-rw-r-- 1 user1 user1 0 Aug 25 14:48 content2.txt
[user1@vm1:~$
```



content2.txt มีเนื้อหาเป็น 0 คือไม่มีเนื้อหา  
เพราะไม่มีผลทาง stdout มีแต่ error message  
ทาง stderr

```
[user1@vm1:~$ ls
mydir  mydir2  sub2
[user1@vm1:~$
[user1@vm1:~$ ls mydir
link3  link4  myfile1  myfile3
[user1@vm1:~$ cat mydir/myfile3 > content1.txt
[user1@vm1:~$ ls -l content1.txt
-rw-rw-r-- 1 user1 user1 218 Aug 25 14:47 content1.txt
[user1@vm1:~$
[user1@vm1:~$ cat mydir > content2.txt
cat: mydir: Is a directory
[user1@vm1:~$ ls -l content2.txt
-rw-rw-r-- 1 user1 user1 0 Aug 25 14:48 content2.txt
[user1@vm1:~$
[user1@vm1:~$ cat mydir 2> err.txt
[user1@vm1:~$ ls -l err.txt
-rw-rw-r-- 1 user1 user1 27 Aug 25 14:49 err.txt
[user1@vm1:~$ cat err.txt
cat: mydir: Is a directory
[user1@vm1:~$
```

ต้องเปลี่ยนไปใช้ 2> content2.txt แทน

```
[user1@vm1:~$ cat mydir > content2.txt 2>&1
[user1@vm1:~$ cat content2.txt
cat: mydir: Is a directory
[user1@vm1:~$ 
[user1@vm1:~$ find /etc -name netplan > content2.txt 2>&1
[user1@vm1:~$ cat content2.txt
find: '/etc/ssl/private': Permission denied
find: '/etc/polkit-1/localauthority': Permission denied
find: '/etc/multipath': Permission denied
/etc/netplan
user1@vm1:~$ █
```



คำตอบ

การใช้ 2>&1 ทำให้ error message ถูก  
redirect จาก stderr ไปยัง stdout

3 บรรทัดแรกเป็น error message  
บรรทัดสุดท้ายเป็น output

เป็นการ Redirect จาก stdout ไปยัง  
content2.txt

## ทดสอบ

24. จงทำตาม slide เกี่ยวกับ tar และ gzip ต่อไปนี้

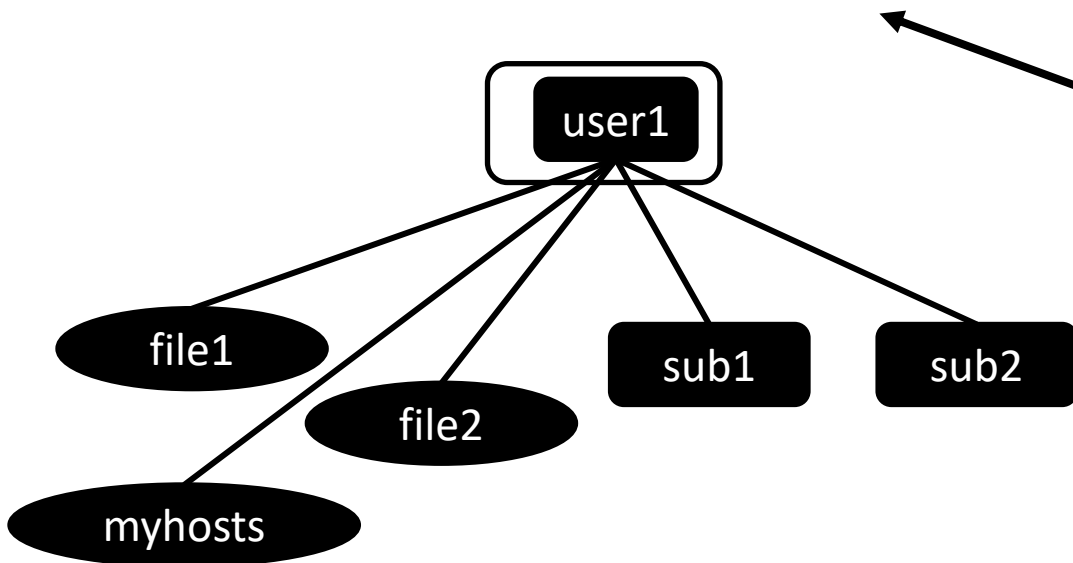
และ copy ผลลัพธ์ของคำสั่ง

```
$ tar xvf mytar1.tar
```

ใส่ลงในคำตอบ

# การสร้าง archive ไฟล์ด้วยคำสั่ง tar

```
[user1@vm1:~$ ls  
file1  file2  sub1  sub2  
[user1@vm1:~$  
[user1@vm1:~$ cp /etc/hosts myhosts
```



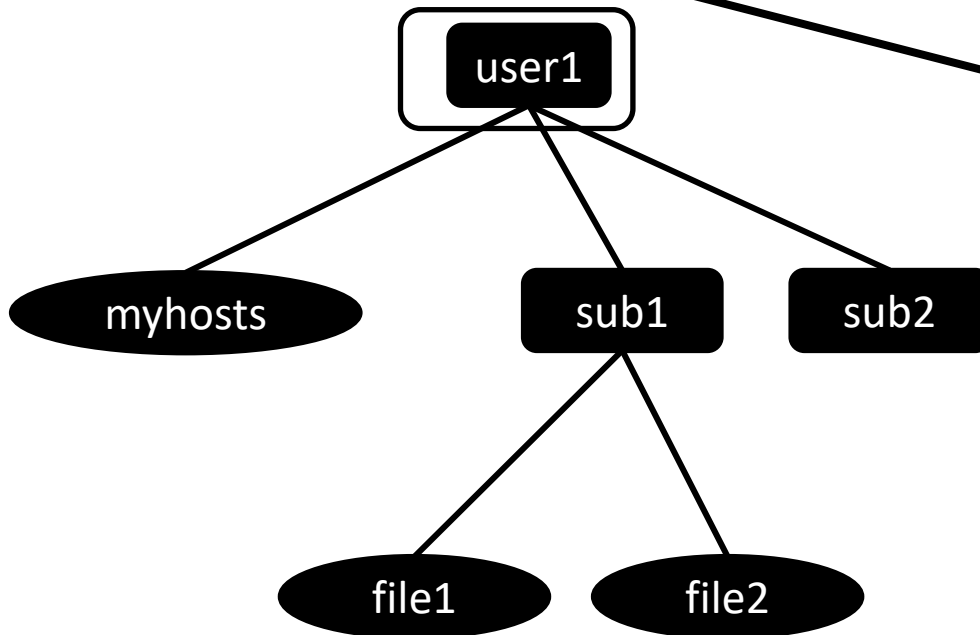
ถ้าใช้คำสั่ง ls จะมีผลลัพธ์  
เป็นอย่างไร



# การสร้าง archive ไฟล์ด้วยคำสั่ง tar

```
[user1@vm1:~$ ls  
file1 file2 sub1 sub2  
[user1@vm1:~$  
[user1@vm1:~$ cp /etc/hosts myhosts  
[user1@vm1:~$ mv f* sub1
```

ย้ายไฟล์ทุกไฟล์ที่มีชื่อนำหน้าด้วย  
f ไปไว้ใน Directory sub1



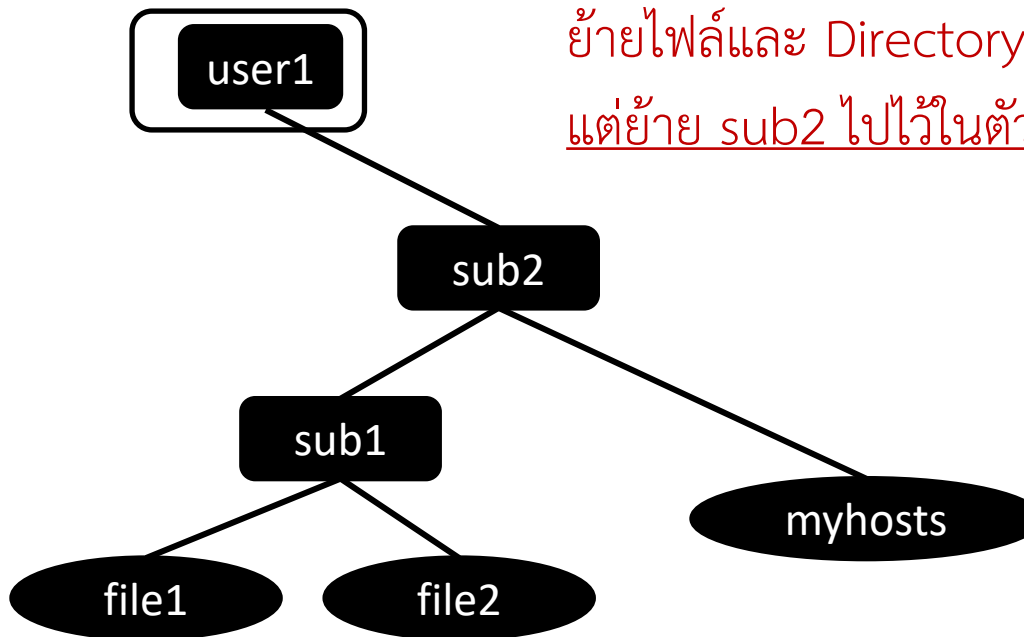
ถ้าใช้คำสั่ง ls จะมีผลลัพธ์  
เป็นอย่างไร

# การสร้าง archive ไฟล์ด้วยคำสั่ง tar

```
[user1@vm1:~$ ls
file1  file2  sub1  sub2
[user1@vm1:~$
[user1@vm1:~$ cp /etc/hosts myhosts
[user1@vm1:~$ mv f* sub1
[user1@vm1:~$ mv * sub2
mv: cannot move 'sub2' to a subdirectory of itself, 'sub2/sub2'
[user1@vm1:~$
```

พยายามย้ายทุกไฟล์ทุกไฟล์  
และ directory ไปไว้ใน


Directory sub2



ย้ายไฟล์และ Directory อื่นไปไว้ใน sub2 ได้หมด  
แต่ย้าย sub2 ไปไว้ในตัวเองไม่ได้

# การสร้าง archive ไฟล์ด้วยคำสั่ง tar

```
[user1@vm1:~$ ls
file1  file2  sub1  sub2
[user1@vm1:~$
[user1@vm1:~$ cp /etc/hosts myhosts
[user1@vm1:~$ mv f* sub1
[user1@vm1:~$ mv * sub2
mv: cannot move 'sub2' to a subdirectory of itself, 'sub2/sub2'
[user1@vm1:~$
[user1@vm1:~$ ls */*
sub2/myhosts
```



```
sub2/sub1:
file1  file2
```

1. List ชื่อไฟล์ใน directory ปัจจุบันและชื่อไฟล์และชื่อใดแรกทอริใน subdirectory ของ directory ปัจจุบัน
2. list ชื่อไฟล์และ subdirectory ของทุก subdirectory

ถ้าออกคำสั่ง \$ ls \* ผลลัพธ์จะต่างจาก \$ ls \*/\* อย่างไร

# การสร้าง archive ไฟล์ด้วยคำสั่ง tar

```
[user1@vm1:~$ ls
file1  file2  sub1  sub2
[user1@vm1:~$
[user1@vm1:~$ cp /etc/hosts myhosts
[user1@vm1:~$ mv f* sub1
[user1@vm1:~$ mv * sub2
mv: cannot move 'sub2' to a subdirectory of itself, 'sub2/sub2'
[user1@vm1:~$
[user1@vm1:~$ ls */*
sub2/myhosts
```

```
sub2/sub1:
file1  file2
[user1@vm1:~$ tar cvf mytar1.tar sub2
sub2/
sub2/myhosts
sub2/sub1/
sub2/sub1/file1
sub2/sub1/file2
[user1@vm1:~$ ls -l
total 16
-rw-rw-r-- 1 user1 user1 10240 Aug 25 03:16 mytar1.tar
drwxrwxr-x 3 user1 user1  4096 Aug 25 03:14 sub2
[user1@vm1:~$
```

คำสั่งเหล่านี้ทำให้โครงสร้าง

← ไดรกทอรีเปลี่ยนอย่างไร (กราฟ)

คำสั่ง tar cvf

c = create

v = verbose

f = ใช้ชื่อถัดไปเป็นชื่อ archive file

# การ compress ไฟล์ด้วย gzip

คำสั่งบีบอัดไฟล์

```
[user1@vm1:~$ ls -l
total 16
-rw-rw-r-- 1 user1 user1 10240 Aug 25 03:16 mytar1.tar
drwxrwxr-x 3 user1 user1 4096 Aug 25 03:14 sub2
[user1@vm1:~$
[user1@vm1:~$ gzip mytar1.tar
[user1@vm1:~$
[user1@vm1:~$ ls -l
total 8
-rw-rw-r-- 1 user1 user1 479 Aug 25 03:16 mytar1.tar.gz
drwxrwxr-x 3 user1 user1 4096 Aug 25 03:14 sub2
[user1@vm1:~$
[user1@vm1:~$ mv mytar1.tar.gz /tmp
[user1@vm1:~$ cd /tmp
```

ย้ายทาร์ไฟล์

# การ uncompress/untar ไฟล์

```
[user1@vm1:/tmp$ ls
mytar1.tar.gz
snap.lxd
systemd-private-c34050aa8824428aae2ce7cfdb2ee666-systemd-logind.service-6p9Km5
systemd-private-c34050aa8824428aae2ce7cfdb2ee666-systemd-resolved.service-j2NotJ
systemd-private-c34050aa8824428aae2ce7cfdb2ee666-systemd-timesyncd.service-GsW9ym
[user1@vm1:/tmp$
[user1@vm1:/tmp$ gzip -d mytar1.tar.gz
[user1@vm1:/tmp$ ls
mytar1.tar
snap.lxd
systemd-private-c34050aa8824428aae2ce7cfdb2ee666-systemd-logind.service-6p9Km5
systemd-private-c34050aa8824428aae2ce7cfdb2ee666-systemd-resolved.service-j2NotJ
systemd-private-c34050aa8824428aae2ce7cfdb2ee666-systemd-timesyncd.service-GsW9ym
[user1@vm1:/tmp$ tar xvf mytar1.tar
sub2/
sub2/myhosts
sub2/sub1/
sub2/sub1/file1
sub2/sub1/file2
[user1@vm1:/tmp$ ls
mytar1.tar  systemd-private-c34050aa8824428aae2ce7cfdb2ee666-systemd-logind.service-6p9Km5
snap.lxd   systemd-private-c34050aa8824428aae2ce7cfdb2ee666-systemd-resolved.service-j2NotJ
sub2       systemd-private-c34050aa8824428aae2ce7cfdb2ee666-systemd-timesyncd.service-GsW9ym
[user1@vm1:/tmp$
```

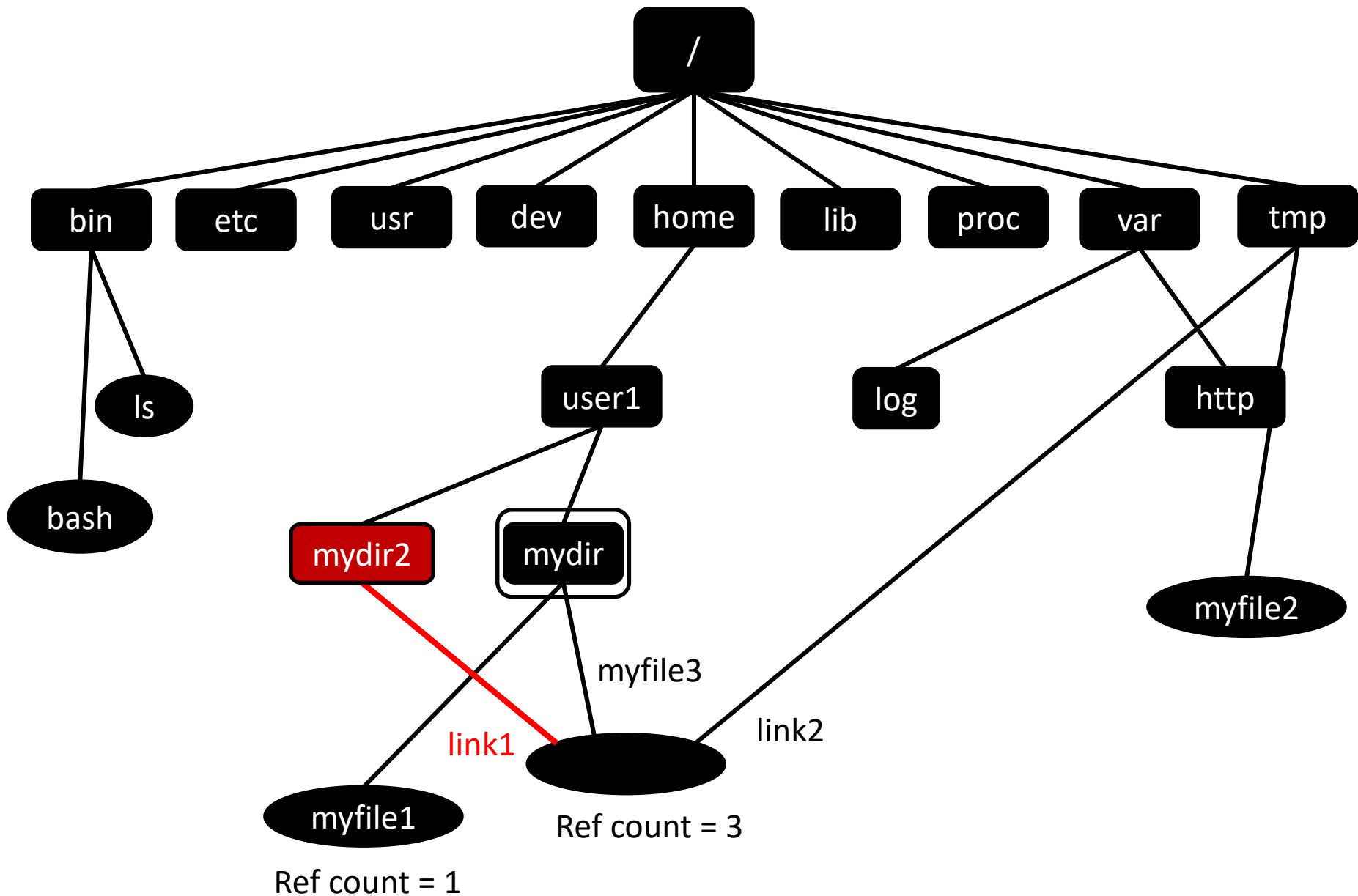
คำสั่งขยายไฟล์

แตกทาร์ไฟล์

คำตอบ

## ทดสอบ

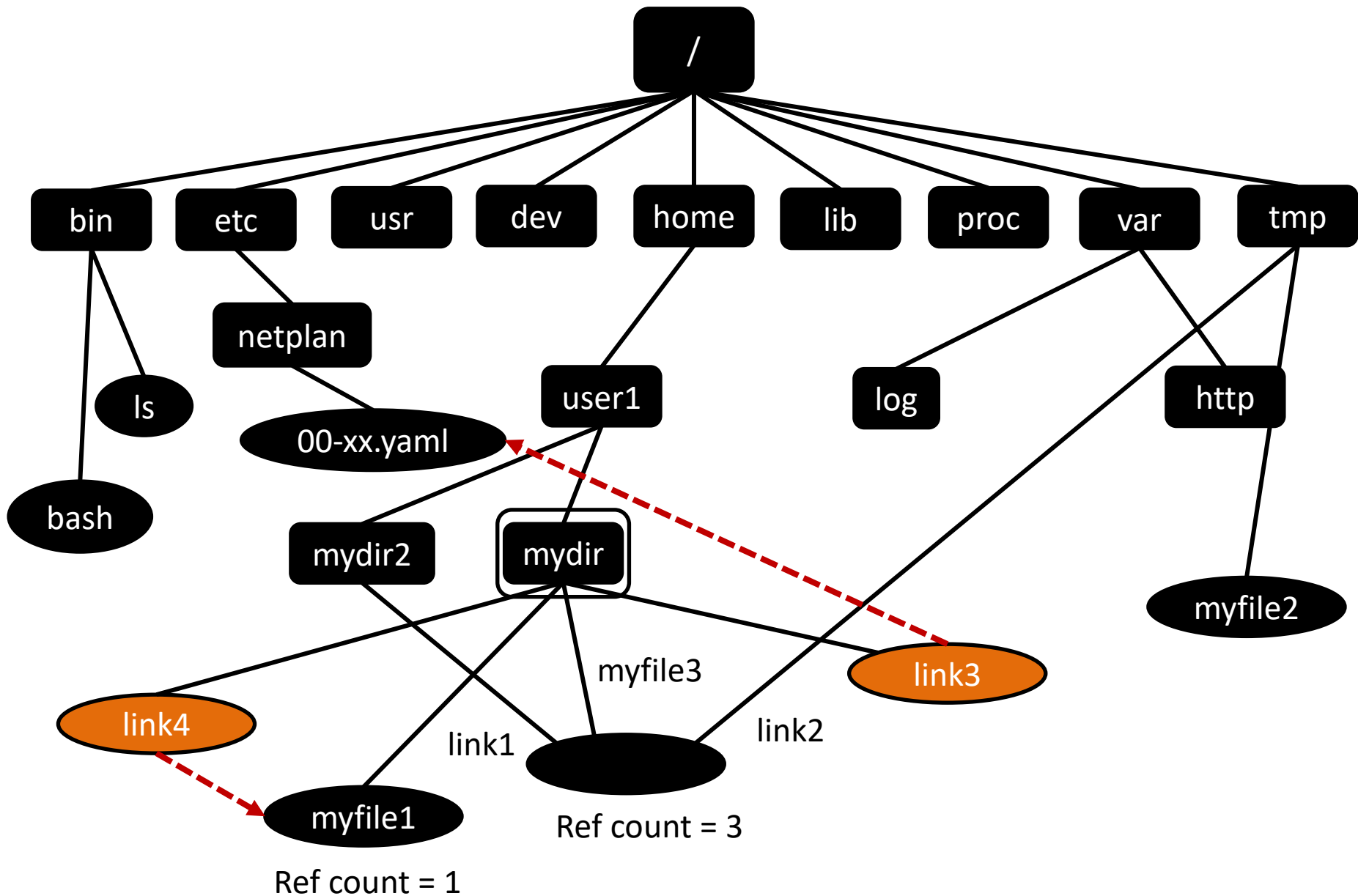
25. จงแสดงคำสั่งสร้าง hard links ดังภาพ สมมุติว่า user1 คือ Room107





## ทดสอบ

26. จงแสดงคำสั่งสร้าง symbolic links ดังภาพ สมมุติว่า user1 คือ Room107



## ทดสอบ

- 27. จงนำผลของคำสั่ง `$echo $VAR1` ใน slide สุดท้ายมาใส่เป็นคำตอบ
- 28. จงแสดงคำสั่งสร้าง local variable มาหนึ่งตัวอย่าง
- 29. จงแสดงคำสั่งสร้าง environment variable มาหนึ่งตัวอย่าง

```
[user1@vm1:~$ set | more
BASH=/bin/bash
BASHOPTS=checkwinsize:cmdhist:complete_fullquote:expand_
basciiranges:histappend:interactive_comments:login_shell
BASH_ALIASES=()
BASH_ARGC=([0]="0")
BASH_ARGV=()
BASH_CMDS=()
BASH_COMPLETION_VERSIONINFO=([0]="2" [1]="11")
BASH_LINENO=()
BASH_SOURCE=()
BASH_VERSIONINFO=([0]="5" [1]="1" [2]="4" [3]="1" [4]="rele
BASH_VERSION='5.1.4(1)-release'
COLUMNS=98
DIRSTACK=()
EUID=1001
GROUPS=()
HISTCONTROL=ignoreboth
HISTFILE=/home/user1/.bash_history
HISTFILESIZE=2000
HISTSIZ=1000
HOME=/home/user1
HOSTNAME=vm1
HOSTTYPE=aarch64
IFS=$' \t\n'
LANG=C.UTF-8
LC_CTYPE=C.UTF-8
LESSCLOSE='/usr/bin/lesspipe %s %s'
LESSOPEN='| /usr/bin/lesspipe %s'
LINES=30
--More--
```

```
[user1@vm1:~$  
[user1@vm1:~$ echo $USER  
user1  
[user1@vm1:~$ echo $HOME  
/home/user1  
[user1@vm1:~$ echo $SHELL  
/bin/bash  
[user1@vm1:~$ echo $PATH  
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:  
/bin:/usr/games:/usr/local/games:/snap/bin  
user1@vm1:~$ echo $PWD  
/home/user1  
user1@vm1:~$ echo $HOSTNAME  
vm1  
user1@vm1:~$ echo $HOSTTYPE  
aarch64  
[user1@vm1:~$  
[user1@vm1:~$ which ls  
/usr/bin/ls  
[user1@vm1:~$ which env  
/usr/bin/env  
[user1@vm1:~$ which date  
/usr/bin/date  
[user1@vm1:~$ which ifconfig  
[user1@vm1:~$ which netplan  
/usr/sbin/netplan  
[user1@vm1:~$
```

ลิสต์ของที่เก็บ executable

แสดงตำแหน่งที่เก็บ  
ถ้าไม่พบก็จะไม่แสดงอะไร

```
user1@vm1:~$  
user1@vm1:~$ VAR1="test v1"  
user1@vm1:~$ echo $VAR1  
test v1
```

```
user1@vm1:~$ bash  
user1@vm1:~$  
user1@vm1:~$ echo $VAR1  
  
user1@vm1:~$ exit
```

```
exit  
user1@vm1:~$ echo $VAR1  
test v1
```

```
user1@vm1:~$ export VAR1
```

```
user1@vm1:~$ bash  
user1@vm1:~$  
user1@vm1:~$ echo $VAR1  
test v1  
user1@vm1:~$  
user1@vm1:~$ exit
```

```
exit  
user1@vm1:~$ unset VAR1  
user1@vm1:~$ echo $VAR1
```

```
user1@vm1:~$
```

เมื่อ VAR1 เป็นตัวแปรโลคอล  
เมื่อเรียกโปรแกรมเชลใหม่ (ซ้อน)  
เชลใหม่จะไม่รู้จัก VAR1

ทำให้ VAR1 เป็น  
Environment variable

คำตอบ

ยกเลิก VAR1 และเคลียร์ค่า

## ทดสอบ

30. จงทำตาม slide ต่อไปนี้ และแสดงผลลัพธ์ของ \$ which a.sh

# การเพิ่ม directory ใน \$PATH

```
$ cd $HOME
$ mkdir bin
$ echo "echo test scripts" > $HOME/bin/a.sh
$ chmod 755 $HOME/bin/a.sh; mkdir mydir/scripts
$ cp $HOME/bin/a.sh mydir/scripts
$ chmod 755 mydir/scripts/*
$ echo "echo test scripts 2" >> mydir/scripts/a.sh
$ mkdir $HOME/bin
$ PATH=$PATH:$HOME/bin
$ echo $PATH
$ which a.sh
$ a.sh
$ PATH=$HOME/mydir/scripts:$PATH
$ which a.sh
```

ถ้าต้องการลบ \$HOME/bin และ  
\$HOME/mydir/bin ออกจาก PATH  
ต้องทำอย่างไร



# ทดสอบ

31. จงทำตาม slide ต่อไปนี้ และแสดงผลลัพธ์ของ  
การรัน script `hello1.sh`

# การสร้าง Shell script Part 1

- ใช้ “nano” เพื่อสร้าง a script file
- บรรทัดแรกต้องใส่  
#!/bin/bash
- เราสามารถใส่คำสั่ง command lines ใน script

```
$ nano hello1.sh
#!/bin/bash
echo "Hello world!"
pwd
ls
```

# การกำหนดค่า Permission Flag ให้ Shell Script

```
$ cd ; cd mydir
```

```
$ ./hello.sh
```

What do you see..

```
$ ls -l hello.sh
```

What do you see..

```
$ chmod 755 hello.sh
```

```
$ ls -l hello.sh
```

What do you see..

```
$ ./hello.sh
```

What happen..

```
$ chmod 000 hello.sh
```

```
$ ls -l hello*
```

What does it mean..

```
$ chmod 100 hello.sh
```

```
$ ls -l hello*
```

```
$ chmod 777 hello.sh
```

## ทดสอบ

32. จงทำตาม slide ต่อไปนี้ และแสดงผลลัพธ์ของ  
การรัน script `hello2.sh`

# การสร้าง Shell script

- ผ่านค่า Parameters จาก CLI
- Working directory ในขณะที่สคริปต์ประมวลผลและหลังประมวลผลเสร็จ
- คำสั่ง sleep

```
$ nano hello2.sh
```

```
#!/bin/bash
```

```
echo "Hello world" ${1}
```

```
echo ${0}
```

```
cp /etc/hosts wk${1}.txt
```

```
sleep 20
```

```
pwd
```

```
echo done
```

## ทดสอบ

33. จงทำตาม slide ต่อไปนี้ และแสดงผลลัพธ์ของ

```
$ ./hello2.sh 3 &
```

```
$ ./hello2.sh 4 &
```

```
$ ps -u
```

## การประมวลผล background process และแสดงสถานะ

```
$ cd $HOME/mydir
```

```
$ ./hello2.sh 1
```

```
$ ./hello2.sh 2
```

What happen? อธิบายว่าสคริปต์ทำอะไร

```
$ ./hello2.sh 3 &
```

```
$ ./hello2.sh 4 &
```

```
$ ps -u
```

What do you see..



คำตอบ

```
$ ps -elf | grep hello
```

## ทดสอบ

34. จงสร้าง hello3.sh และทำตาม slide ต่อไปนี้ และแสดงผลลัพธ์ของ

```
$ ps -elf | grep hello3
```

และแสดงคำสั่ง kill processes



# การสร้าง Shell script

- ผ่านค่า Parameters จาก CLI
- Working directory ในขณะที่สคริปต์ประมวลผลและหลังประมวลผลเสร็จ
- คำสั่ง sleep รับค่าพารามิเตอร์

```
$ nano hello3.sh
```

```
#!/bin/bash
```

```
echo "Hello world" ${1}
```

```
echo ${0}
```

```
cp /etc/hosts wk${1}.txt
```

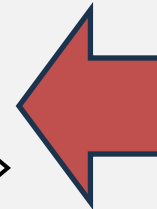
```
sleep ${2}
```

```
pwd
```

```
echo done
```

## การดูสถานะและ kill process

```
$ ./hello3.sh 3 5  
$ ./hello3.sh 6 500 &  
$ ./hello3.sh 7 1000 &  
$ ps -elf  
$ ps -elf | grep hello  
$ kill -9 <process id of hello 6 500>  
$ kill -9 <process id of hello 7 1000>
```



คำตอบ

## ทดสอบ

35. จงสร้าง printname.sh และทำตาม slide ต่อไปนี้ และแสดงผลลัพธ์ของ

```
$ ./printname.sh
```

# สร้าง shell script แบบให้ป้อนค่า input variables

- ตัวแปร first และ last เป็นตัวแปร shell variables
- คำสั่ง read เป็นคำสั่งอ่านค่าสตริงที่ผู้ใช้ป้อนเข้าสู่ตัวแปร -p เป็นการแสดงข้อความก่อนที่จะอ่านค่าตัวแปรจาก CLI
- \$first และ \$last เป็นการดึงค่าที่เก็บในตัวแปรมาใช้

```
$ nano printname.sh
#!/bin/sh
read -p "enter your name: " first last
echo "First name: $first"
echo "Last name: $last"
```

## ทดสอบ

36. จงสร้าง C programs และทำตาม slide ต่อไปนี้ และแสดงผลลัพธ์ของ

```
$ ./hello
```

ติดตั้ง C compiler และ compile C program

```
$ sudo apt install gcc make  
(or apt install build-essentials)  
$ nano hello.c  
#include <stdio.h>  
main(){  
    printf("hello world :)\n");  
}  
$ gcc hello.c  
$ ./a.out  
$ gcc -o hello hello.c  
$ ./hello
```

## ทดสอบ

34. จงสร้าง C programs และทำตาม slide ต่อไปนี้ และแสดงผลลัพธ์ของ

```
$ ./hello
```

# การคอมไพล์ object code และ link โปรแกรม (1)

```
$ mkdir myprograms
$ cd myprograms
$ nano hellosub1.c
#include <stdio.h>
#include "hello1.h"
void fun1(void){
    printf("hello1 %d\n", CONSTANT1);
}
$ gcc -c hellosub1.c
```



# การคอมไพล์ object code และ link โปรแกรม (1)

```
$ nano hello1.h  
$ cat hello1.h  
#define CONSTANT1 100  
$
```

## การคอมไพล์ object code และ link โปรแกรม (2)

```
$ nano hellosub2.c
#include <stdio.h>


void fun2(void){
    printf("hello2\n");
}
$ gcc -c hellosub2.c
$ ls -l
```

## การคอมไพล์ object code และ link โปรแกรม (3)

```
$ nano hellomain.c
#include <stdio.h>
extern void fun1(void);
extern void fun2(void);

int main(){
    fun1();
    fun2();
}

$ gcc -c hellomain.c
$ ls -l
```



function call  
prototypes

## การคอมไพล์ object code และ link โปรแกรม (4)

```
$ gcc -o hello hellomain.o \  
> hellosub1.o hellosub2.o  
$ ls -l  
$ file *  
$ ./hello
```