



# คำสั่งเชลล์และเชลล์สคริป

สำหรับนักศึกษาชั้นปีที่ 2 สาขาวิชาวิศวกรรมคอมพิวเตอร์

กรุงฤทธิ์ กิติศรีวงศ์พันธุ์

Email : songrit@npu.ac.th

สาขาวิชาวิศวกรรมคอมพิวเตอร์  
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนครพนม

# ວັນນີ້

---

- คำสั่งລົບຖະບັນພື້ນຈານ
- ກາຣເຢຍນເຊລເລົກສຄຣີປ

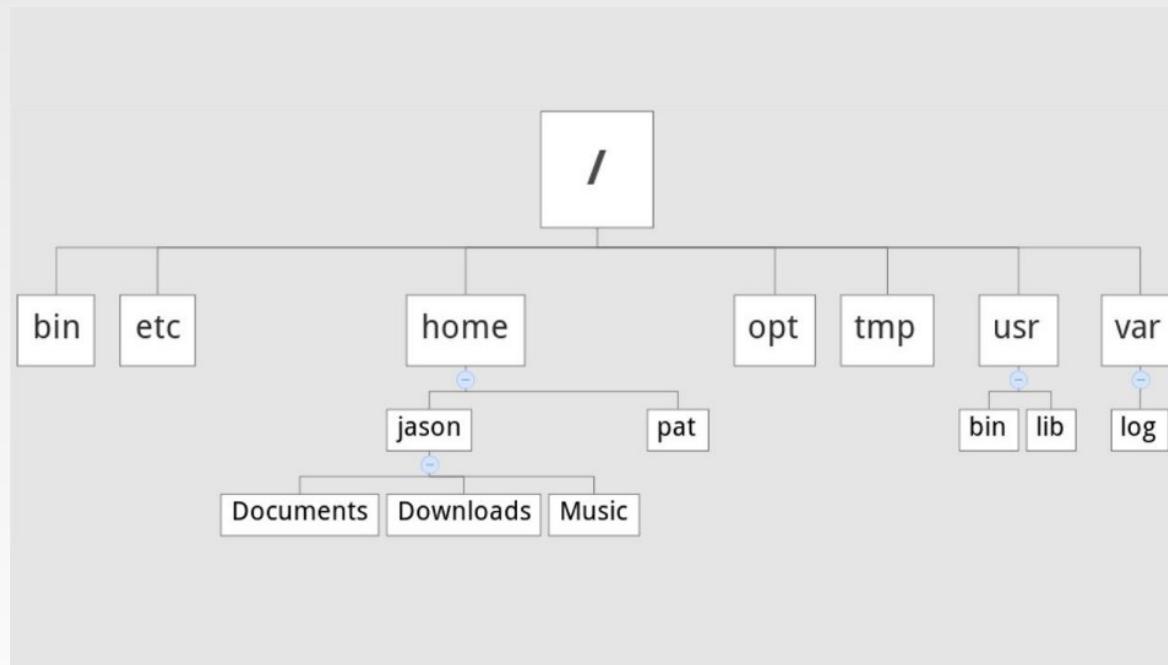
# ไฟล์และคำสั่ง

---

- โครงสร้างไฟล์
- ชนิดของไฟล์
- คำสั่งเกี่ยวกับไฟล์

# โครงสร้างระบบจัดเก็บไฟล์

- จัดสรรไฟล์เรียงตามลำดับชั้น (Tree)
- ใช้คำสั่ง ls (มาจาก list) แสดงรายชื่อไฟล์
- มี / อยู่ชั้นบนสุด เรียกว่า ราก (root)



ใช้คำสั่ง tree และรายการระบบไฟล์ได้

# /home

---

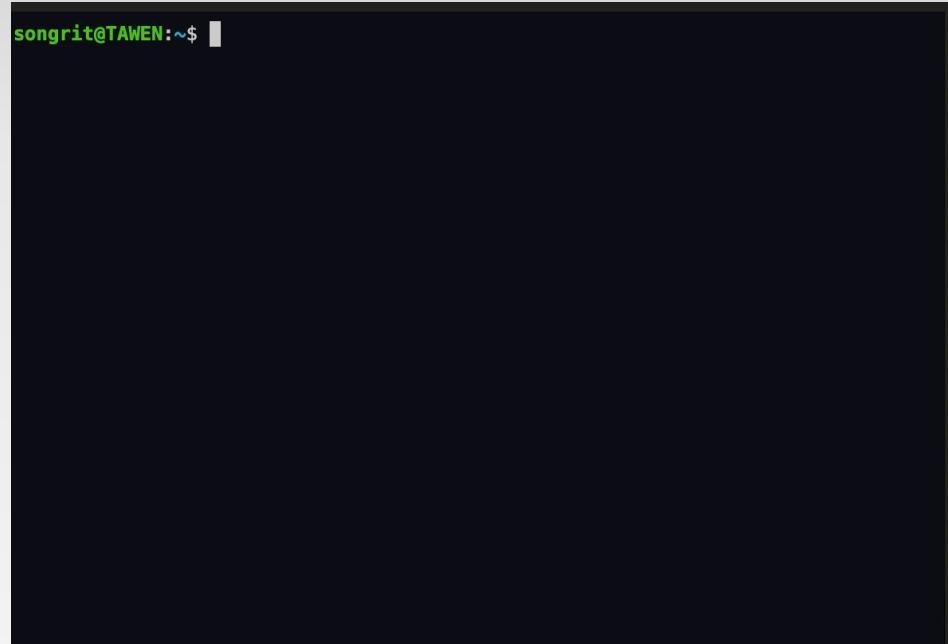
- ໄດ້ເຮັກໂທຮ່ວມຈໍາແວຄເຄາກ໌
- ເກີບແວຄເຄາກ໌ ຜູ້ໃຊ້ກັ້ງໝາດຖື່ມໍໃໝ່ root
- ໃຊ້ຄຳສັ່ງ cd ຈະກຳໃຫ້ກລັບໄປຕຳແໜ່ນໄດ້ເຮັກໂທຮ່ວມຈໍາແວຄເຄາກ໌



# /root

---

- ໄດ້ເຮັດວຽກທອරີສໍາຫຼັບຜູ້ມີສຶກຮີສູງສຸດໃນຮະບບ
- ເຮັດວຽກວ່າ “ມີສຶກຮີສູງສຸດ”
  - ມີສຶກຮີເກົ່າກັບ root
  - ຜ່ອ user : root
- ID ຜ່ອ root ເກົ່ານັ້ນສາມາດເຂົ້າໃຊ້ໄດ້ເຮັດວຽກທອරີນີ້ໄດ້
- ຄວດໃຊ້ສຶກຮີ root ເພາະຍາມຈຳເປັນ
- ການເພີ່ມສຶກຮີຈາກ user ກ້ວ່າໄປເປັນ root ກໍາໄດ້ດ້ວຍຄໍາສັ່ງ
  - sudo



songrit@TAWEN:~\$ █

# ไฟล์และคำสั่ง

---

- โครงสร้างไฟล์
- ชนิดของไฟล์
- คำสั่งเกี่ยวกับไฟล์

# ชนิดไฟล์ใน Windows

---

- ระบบ Windows มีไฟล์หลัก 3 ชนิด
  - Directory (Folder)
  - Regular file
- แต่ละ regular file มีคุณสมบัติ (attribute)
  - ผู้ที่มีสิทธิ์ใช้ไฟล์ (permission)
  - เป็นไฟล์ซ่อน (hidden file)
  - เป็นไฟล์ execute
  - เป็น shortcut file
  - เป็นไฟล์ word, excel

# ชนิดไฟล์ใน Linux

---

- ลิบุกช์มของว่า สิ่งที่ OS ติดต่อได้เป็นไฟล์กั้งหมด
- แต่ลักษณะการติดต่อนั้นขึ้นอยู่กับคุณสมบัติของไฟล์
  - ไฟล์ชนิดไฟล์กั่วไป
  - ไฟล์ชนิดไดเร็กทอรี
  - ไฟล์อื่นเช่น ไฟล์อุปกรณ์ ไฟล์หน่วยความจำ
- คำสั่งตรวจสอบไฟล์ใช้คำสั่ง ls -l ใช้ได้กับทั้ง
  - ไฟล์
  - ไดเร็กทอรี
  - อุปกรณ์ฮาร์ดแวร์ , หน่วยความจำ
  - Driver
- มีการกำหนดสิทธิ์เข้าใช้ไฟล์
  - เป็น 3 กลุ่ม
  - เจ้าของไฟล์ กลุ่มของผู้ใช้ และ คนอื่น

# คำสั่ง ls -l

---

```
songrit@TAWEN:~$
```

# ลักษณะของไฟล์

- ไฟล์กั่วไป (Regular file) (-)
- ไดเร็คทอรี (Directory) (d)
- ลิงค์ไปยังไฟล์อื่น (link) (l)

permission

```
cpe@server:~$ ls -l
total 4
-rw-rw-r-- 1 cpe cpe    0 Aug 27 15:51 abc.txt
lrwxrwxrwx 1 cpe cpe 23 Aug 27 15:51 dhclient.conf -> /etc/dhcp/dhclient.conf
drwxrwxr-x 2 cpe cpe 4096 Aug 27 15:52 thedirectory
cpe@server:~$
```

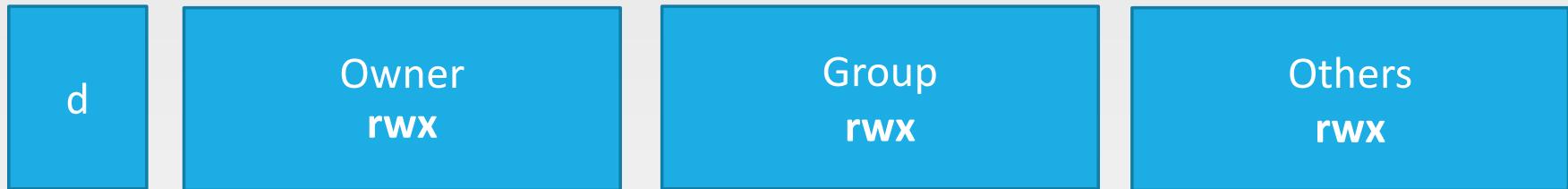
# ສຶກຮີໃນກາຣໃໝ່ໄຟລ໌

---

- ແບ່ງຜູ້ໃໝ່ເປັນ 3 ກລຸ່ມ
  - ເຈົ້າຂອງໄຟລ໌ (Owner)
  - ກລຸ່ມຜູ້ໃໝ່ (Groups)
  - ດັວກລຸ່ມອື່ນ (Others)
- Owner ເປັນຜູ້ສ້າງໄຟລ໌ → ເຈົ້າຂອງໄຟລ໌
- Groups ຄື່ອເຫດຂອງ user ທີ່ມີສຶກຮີຕ່ວໄຟລ໌
- Others ຄື່ອກລຸ່ມອື່ນທີ່ໄມ່ໃໝ່ Owner ແລະ Groups

# គុណសមបាតិទាំងអេ

- ចូល (Read) → r
- ផ្ទើរ (Write) → w
- ការិត (Execute) → x



```
cpe@server:~$ ls -l
total 4
-rw-rw-r-- 1 cpe cpe
lrwxrwxrwx 1 cpe cpe
drwxrwxr-x 2 cpe cpe
cpe@server:~$
```

# ข้อมูลของไฟล์

File	Type	Owner	Group	Size	Date	Time	File
-rw-r-----	1	root	shadow	934	Jul 21	08:18	shadow
-rw-r-----	1	root	shadow	804	Jul 21	08:18	shadow-
-rw-r--r--	1	root	root	103	Feb 14	2019	shells
drwxr-xr-x	2	root	root	4096	Feb 14	2019	skel
-rw-r--r--	1	root	root	100	Jun 25	2018	sos.conf

- ใช้คำสั่ง ls -l
  - Permission
  - Owner
  - Group
  - Size
  - Date
  - File name

# ไฟล์และคำสั่ง

---

- โครงสร้างไฟล์
- ชนิดของไฟล์
- คำสั่งเกี่ยวกับไฟล์

# คำสั่งเกี่ยวกับข้องกับไฟล์

---

- คำสั่งท่องระบบไฟล์
- คำสั่งจัดการไฟล์
- คำสั่งกำหนดสิทธิ์แก่ไฟล์
- คำสั่งเลือกแสดงผล

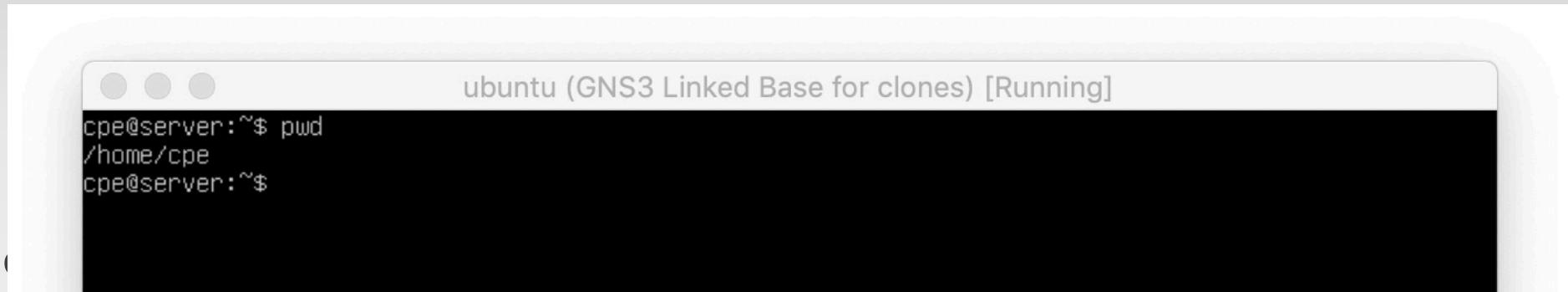
# คำสั่งท่องระบบไฟล์

---

- **pwd** (Present Working Directory)
  - → ดูตำแหน่งปัจจุบัน
- **ls** (List) → ดูรายชื่อไฟล์ในตำแหน่งปัจจุบัน
- **cd** (Change directory) → เปลี่ยนตำแหน่งที่อยู่

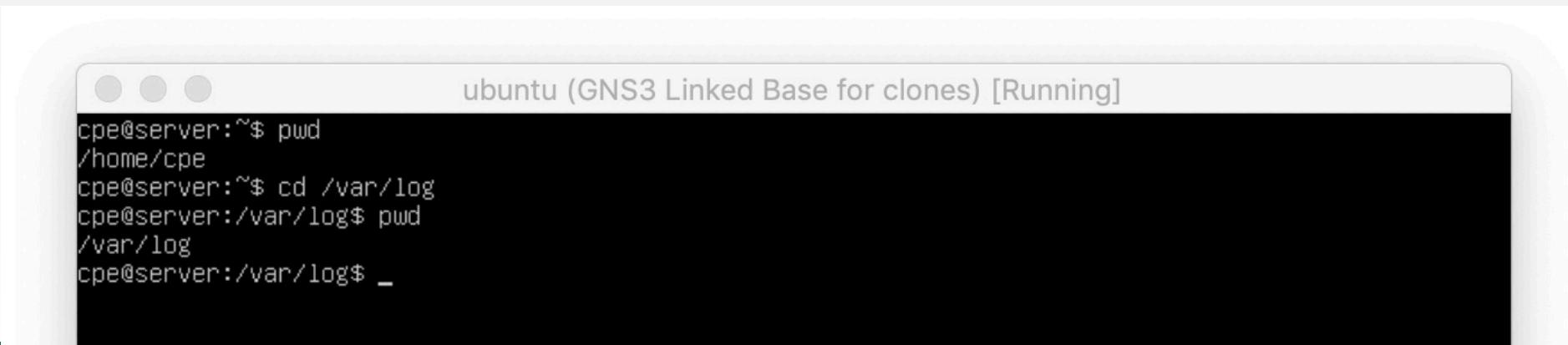
# คำสั่ง pwd

- pwd



```
cpe@server:~$ pwd
/home/cpe
cpe@server:~$
```

- pwd



```
cpe@server:~$ pwd
/home/cpe
cpe@server:~$ cd /var/log
cpe@server:/var/log$ pwd
/var/log
cpe@server:/var/log$ _
```

# คำสั่ง cd

---

- ใช้เปลี่ยนตำแหน่งที่อยู่ปัจจุบัน
- cd <ไดเรกทอรีที่ต้องการเปลี่ยน>
- cd อย่างเดียวแปลว่ากลับบ้าน เช่น user : cpe
  - /home/cpe

# คำสั่ง ls

- คำสั่ง list ไฟล์
- ls -a แสดงรายการไฟล์ทั้งหมด รวมไฟล์
  - ซ่อน (ไฟล์ที่ขึ้นต้นด้วย . เป็นไฟล์ซ่อนอัตโนมัติ เช่น .bashrc)
  - ไฟล์ซ่อน . อย่างเดียวหมายถึง ไดเรกทอรีปัจจุบัน
  - ไฟล์ซ่อน .. หมายถึงไดเรกทอรีบน หนึ่งขั้น
- ls -l แสดงรายการแบบรายละเอียดครบ (long)

```
drwxr-xr-x 4 root root      4096 Feb 14  2019 X11
drwxr-xr-x 4 root root      4096 Feb 14  2019 xdg
-rw-r--r-- 1 root root      477 Mar 16  2018 zsh_command_not_found
cpe@server:/etc$ _
```

# **คำสั่งเกี่ยวกับข้องกับไฟล์**

---

- คำสั่งท่องระบบไฟล์
- คำสั่งจัดการไฟล์
- คำสั่งกำหนดสิทธิ์แก่ไฟล์
- คำสั่งเลือกแสดงผล

# การจัดการไฟล์

---

- สร้างไฟล์
- ดูเนื้อหาในไฟล์
- ลบไฟล์
- คัดลอกไฟล์
- สร้างลิงค์
- กำหนดผู้มีสิทธิ์ใช้ไฟล์

# สร้างไฟล์

---

- สร้างไฟล์เปล่า
  - **touch abc.txt** → สร้างไฟล์เปล่าชื่อ abc.txt
- สร้างไฟล์ด้วยคำสั่ง echo
  - echo "Hello" > abc.txt (ไฟล์ abc.txt มีเนื้อหา Hello)
  - echo "Hello2" >> abc.txt (เพิ่ม Hello2 ต่อท้าย ในไฟล์ abc.txt)
- สร้างไฟล์คำสั่ง dd โดยกำหนดขนาดเอง
  - สร้างไฟล์ 1GB
  - **dd if=/dev/zero of=file.txt count=1024 bs=1048576**
    - if → input file, of → output file
    - count จำนวนรอบ
    - bs → block size หน่วย ไบต์

# สร้างไฟล์ (ชัดไดเร็คทอรี่)

---

- คำสั่งสร้างไดเร็คทอรี
  - `mkdir abc` → สร้างชื่อ abc
  - `-p` ใช้สร้าง subdirectory ที่ยังไม่เคยมี
  - `mkdir -p ~/abc/1/2/3`
  - ไม่เคยมีไดเร็คทอรี 1/2/3
  - ช่วยให้สร้างไฟล์ในคำสั่งเดียว

# คำสั่งดูไฟล์

---

- cat
- more
- head
- tail

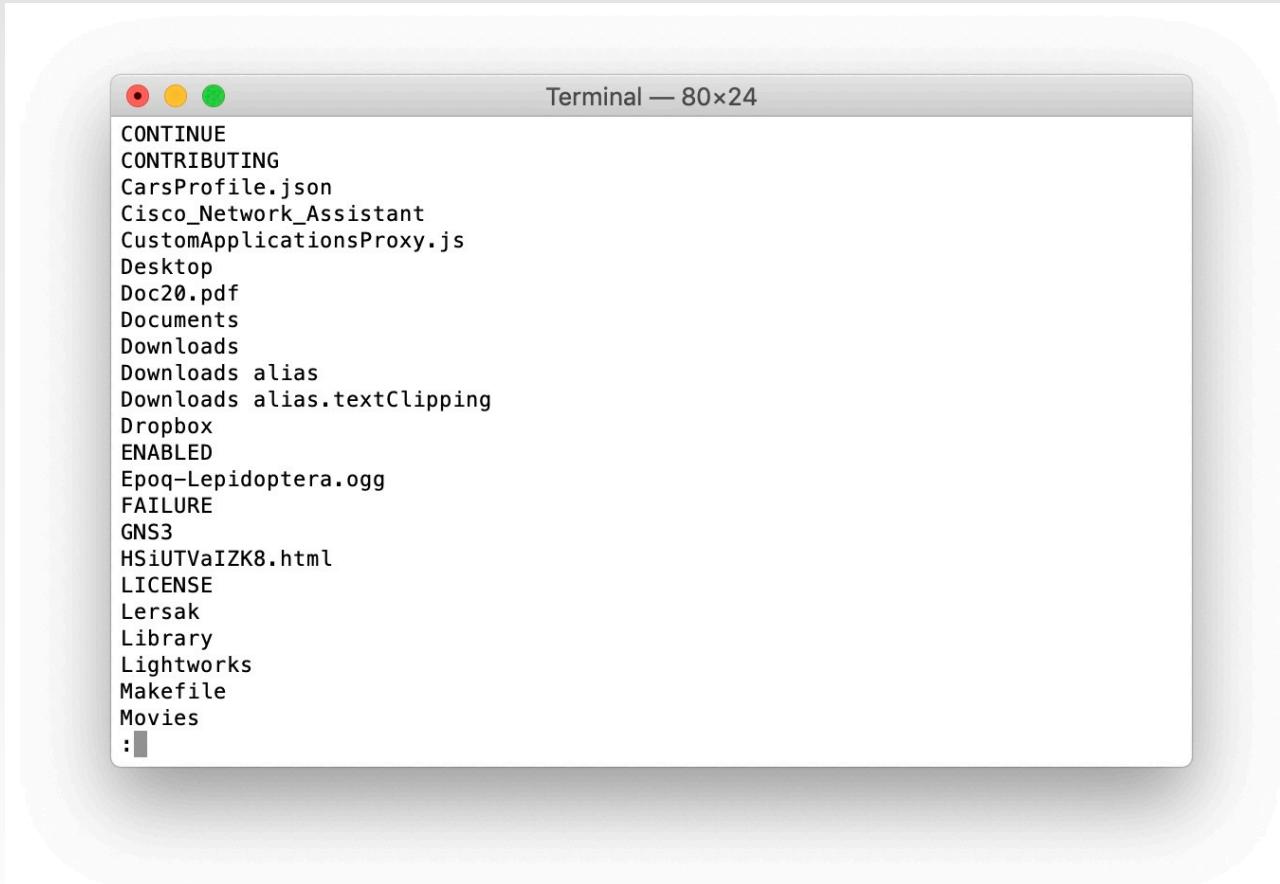
# คำสั่ง cat

---

- อ่าน text file
- cat abc.txt → อ่านข้อความในไฟล์ abc.txt
- ใช้ร่วมกับเครื่องหมาย > เพื่อเปลี่ยนทิศทางเช่น
  - cat abc.txt > abc-2.txt ← อ่านไฟล์ abc.txt และเขียนไฟล์ใหม่เป็น abc-2.txt
  - cat c.txt >> abc.txt ← อ่านไฟล์ c.txt และเขียนต่อท้าย abc.txt

# คำสั่ง more

- ดูไฟล์หนึ่งหน้าจอ
- เคลื่อนหน้าจอ



# คำสั่ง head

---

- อ่านไฟล์นับจากส่วนหัว
- อ่านไฟล์ auth.log 10 บรรทัดนับจากต้นไฟล์
  - head /var/log/auth.log
  - head -10 /var/log/auth.log
  - head -5 /var/log/auth.log → อ่านไฟล์ 5 บรรทัด

# คำสั่ง tail

- อ่านไฟล์เมื่อัน head แต่นับจากท้ายไฟล์
  - tail -5 /var/log/auth.log อ่าน 5 บรรทัด นับจากท้าย
- อ่านไฟล์และคอยดูการอัพเดตไฟล์
  - tail -f /var/log/syslog
  - กด ctl+c ออกจากการโปรแกรม

```
parmor_parser"
Aug 27 10:24:39 server kernel: [ 75.584493] audit: type=1400 audit(1566901479.912:28): apparmor="S
TATUS" operation="profile_replace" profile="unconfined" name="snap.core.hook.configure" pid=1208 com
m="apparmor_parser"
Aug 27 11:32:32 server kernel: [ 4148.518972] e1000: enp0s3 NIC Link is Down
Aug 27 11:32:36 server kernel: [ 4152.551222] e1000: enp0s3 NIC Link is Up 1000 Mbps Full Duplex, Fl
ow Control: RX
Aug 27 11:32:49 server kernel: [ 4164.646896] e1000: enp0s3 NIC Link is Down
Aug 27 11:32:51 server kernel: [ 4166.663534] e1000: enp0s3 NIC Link is Up 1000 Mbps Full Duplex, Fl
ow Control: RX
Aug 27 12:24:12 server kernel: [ 5575.884176] e1000: enp0s3 NIC Link is Down
Aug 27 12:24:14 server kernel: [ 5577.900621] e1000: enp0s3 NIC Link is Up 1000 Mbps Full Duplex, Fl
ow Control: RX
^C
cpe@server:/etc$ tail -f /var/log/kern.log
Aug 27 10:24:39 server kernel: [ 75.499136] audit: type=1400 audit(1566901479.824:25): apparmor="S
TATUS" operation="profile_replace" profile="unconfined" name="/snap/core/7396/usr/lib/snapd/snap-con
fine" pid=1205 comm="apparmor_parser"
Aug 27 10:24:39 server kernel: [ 75.499913] audit: type=1400 audit(1566901479.824:26): apparmor="S
TATUS" operation="profile_replace" profile="unconfined" name="/snap/core/7396/usr/lib/snapd/snap-con
fine//mount-namespace-capture-helper" pid=1205 comm="apparmor_parser"
Aug 27 10:24:39 server kernel: [ 75.522257] audit: type=1400 audit(1566901479.848:27): apparmor="S
TATUS" operation="profile_replace" profile="unconfined" name="snap-update-ns.core" pid=1207 comm="ap
parmor_parser"
Aug 27 10:24:39 server kernel: [ 75.584493] audit: type=1400 audit(1566901479.912:28): apparmor="S
TATUS" operation="profile_replace" profile="unconfined" name="snap.core.hook.configure" pid=1208 com
m="apparmor_parser"
Aug 27 11:32:32 server kernel: [ 4148.518972] e1000: enp0s3 NIC Link is Down
Aug 27 11:32:36 server kernel: [ 4152.551222] e1000: enp0s3 NIC Link is Up 1000 Mbps Full Duplex, Fl
ow Control: RX
Aug 27 11:32:49 server kernel: [ 4164.646896] e1000: enp0s3 NIC Link is Down
Aug 27 11:32:51 server kernel: [ 4166.663534] e1000: enp0s3 NIC Link is Up 1000 Mbps Full Duplex, Fl
ow Control: RX
Aug 27 12:24:12 server kernel: [ 5575.884176] e1000: enp0s3 NIC Link is Down
Aug 27 12:24:14 server kernel: [ 5577.900621] e1000: enp0s3 NIC Link is Up 1000 Mbps Full Duplex, Fl
ow Control: RX
```

# คำสั่งเกี่ยวกับข้องกับไฟล์

---

- คำสั่งท่องระบบไฟล์
- คำสั่งจัดการไฟล์
- คำสั่งกำหนดสิทธิ์แก่ไฟล์
- คำสั่งเลือกแสดงผล

# คำสั่งเกี่ยวกับข้องกับไฟล์

---

- คำสั่งท่องระบบไฟล์
- คำสั่งจัดการไฟล์
- **คำสั่งกำหนดสิทธิ์แก้ไฟล์**
- คำสั่งเลือกการแสดงผล

# คำสั่งกำหนดสิทธิ์แก่ไฟล์

---

- ใช้กำหนดคุณสมบัติ ข้อมูลควบคุมไฟล์
- **chmod** (change mode)
- **chown** (change owner)
- **chattr** (change attribute)

# คำสั่ง chmod

---

- ใช้กำหนด / เปลี่ยนแปลงลักษณะไฟล์ให้สามารถ
  - Read / Write / Execution
- **chmod +r abc.txt**
  - กำหนดให้ไฟล์ abc.txt อ่านได้ทุกคน
- **chmod -r abc.txt**
  - เอาสิทธิ์การอ่านออกจากการไฟล์ ไม่ให้ใครอ่านได้ (นอกจางเจ้าของ)
- **chmod +w abc.txt → ให้สิทธิ์เขียนไฟล์กับทุกคน**
- **chmod +w-x → ให้สิทธิ์เขียนไฟล์ แต่ไม่ให้สิทธิ์รันกับทุกคน**

# คำสั่ง chmod

---

- คำสั่ง chmod กำหนดสิทธิ์ตามผู้ใช้
- Permission ด้วยไบนารี่
- ให้สิทธิ์ แทนด้วย 1 , ไม่ให้สิทธิ์เป็น 0

<b>d</b>	<b>rwx</b>	<b>rwx</b>	<b>rwx</b>
—	---	---	---
	<b>Owner</b>	<b>Group</b>	<b>World</b>
---	---	---	---

# คำสั่ง chmod

---

- ให้เจ้าของไฟล์มีสิทธิ์ทุกอย่างแต่ผู้เดียว

0	1	1	1	0	0	0	0	0	0	0	0
—	—	—	—	—	—	—	—	—	—	—	—
Owner	Group	World									

- $111_2 \ 000_2 \ 000_2 = 7 \ 0 \ 0$
- chmod 700 abc.txt**

# คำสั่งเกี่ยวกับข้องกับไฟล์

---

- คำสั่งท่องระบบไฟล์
- คำสั่งจัดการไฟล์
- คำสั่งกำหนดสิทธิ์แก่ไฟล์
- คำสั่งเลือกแสดงผล

# คำสั่งเลือกแสดงผล

---

- grep
- sed
- cut
- awk

# คำสั่ง grep

---

- เลือกแสดงผล
- grep "test" abc.txt
  - เลือกแสดงผลเฉพาะบรรทัดที่มีข้อความ test
- grep -v "test" abc.txt
  - เลือกแสดงผลเฉพาะบรรทัดที่ไม่มีข้อความ test

# คำสั่ง sed

---

- คำสั่งแทนที่ข้อความ
- 'ส่วนหน้า/ข้อความคันหา/ข้อความทดแทน/ส่วนหลัง'
- **sed 's/ข้อความคันหา/ข้อความทดแทน/g'**
  - **s** ใช้แทนการคันหาที่รวม ช่องว่าง (space)
  - ข้อความคันหาแบบตรงทุกคำ ลำดับถูก
  - ข้อความที่จะนำมาแทนที่
  - **g** ให้คันจนจบไฟล์ (ไม่หยุดที่การพบข้อความแรก)

# คำสั่ง sed

---

- คำสั่งแทนที่ข้อความ
- `sed "s/test number one>Hello world/g" abc.txt`
  - แทนที่ข้อความ Hello world ด้วย test number one
  - แล้วแสดงออกหน้าจอ
  - `sed "s/test number one>Hello world/g" abc.txt > abc2.txt`
    - เขียนลงไฟล์ abc2.txt

# Regular expression

---

- การใช้ Regular expression กับคำสั่ง sed
- **sed 's/[0-9]/Number/g' abc.txt**
  - s เลือก match ส่วนที่เป็นช่องว่างด้วย ย่อจาก space
  - g อ่านไฟล์ทั้งหมด ย่อจาก global
  - คันหาตัวเลข 0 ถึง 9 และแทนที่ด้วยข้อความ Number
- **sed 's/^mkdir/Found/g' abc.txt**
  - หาข้อความที่มี m,k,d,i,r (สลับกันได้) และแทนที่ด้วย Found

# คำสั่ง cut

---

- เป็นคำสั่งตัด string
- echo "Test 1 2 3" |cut -d"1" -f 1
  - -d ใช้อักษร 1 เป็นตัวแบ่งข้อความเป็น 2 ส่วน
  - -f 1 เลือกข้อมูลส่วนแรก
  - -f 2 เลือกข้อมูลส่วนหลัง

# คำสั่ง awk

- awk /a:k/ อ่าน อ็อก
- ใช้เลือกแสดงผลเฉพาะส่วนที่สนใจ
- จัดตาม column

```
cpe@server: ~$ ping www.google.com > ping.txt
^C
cpe@server: ~$ cat ping.txt
PING www.google.com (216.58.203.68) 56(84) bytes of data.
64 bytes from ku109s03-in-f4.1e100.net (216.58.203.68): icmp_seq=1 ttl=63 time=65.1 ms
64 bytes from ku109s03-in-f4.1e100.net (216.58.203.68): icmp_seq=2 ttl=63 time=99.8 ms
64 bytes from ku109s03-in-f4.1e100.net (216.58.203.68): icmp_seq=3 ttl=63 time=101 ms
64 bytes from ku109s03-in-f4.1e100.net (216.58.203.68): icmp_seq=4 ttl=63 time=94.2 ms
64 bytes from ku109s03-in-f4.1e100.net (216.58.203.68): icmp_seq=5 ttl=63 time=98.6 ms
64 bytes from ku109s03-in-f4.1e100.net (216.58.203.68): icmp_seq=6 ttl=63 time=82.4 ms
64 bytes from ku109s03-in-f4.1e100.net (216.58.203.68): icmp_seq=7 ttl=63 time=76.5 ms
64 bytes from ku109s03-in-f4.1e100.net (216.58.203.68): icmp_seq=8 ttl=63 time=73.8 ms

--- www.google.com ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7013ms
rtt min/avg/max/mdev = 65.180/86.559/101.665/12.973 ms
cpe@server: ~$ cat ping.txt
```

# คำสั่ง awk

---

- ต้องการเฉพาะเลข icmp\_seq และ time
- **cat rtt.txt | grep icmp\_seq**
- |awk '{print }'
- |awk '{print \$6,\$8}'
- |sed 's/icmp\_seq=///g'
- |sed 's/time=///g'
  
- **cat ping.txt|grep icmp\_seq|awk '{print \$6,\$8}'|sed 's/icmp\_seq=///g'|sed 's/time=///g'**

# Shellscript

---

- เชลล์สคริป เป็น การนำคำสั่งเชลล์มาทำงานร่วมกัน
- คำสั่งเชลล์ทำงานตามลำดับทีละบรรทัด
- คำสั่งเชลล์ทำหน้าที่ได้แบบภาษาโปรแกรมมิ่ง
  - มีการกำหนดตัวแปร
  - มีคำสั่งเงื่อนไข
  - มีคำสั่งวนลูป

# การกำหนดตัวแปร

---

- ตัวแปรในหน่วยความจำ
  - age=19
  - name='Paul Pheonix'
  - name='นาย เอเบซ'
- การพิมพ์ค่าของตัวแปร
  - echo \$age
  - echo "\$name is \$age years old"

# ตัวดำเนินการเชิงเลข

```
#!/bin/bash  
x=3  
y=5  
z=`echo $x + $y | bc`  
echo $z
```

เป็นภาษาอื่นได้ เช่น  
*python, ruby, nodejs*

```
#!/bin/bash  
x=3  
y=5  
z=$((x+y))  
echo $z
```

# การกำหนดตัวแปร

- รับค่าตัวแปรจากเชลล์
- ภายในโปรแกรมจะรับค่าเรียงตามลำดับ @1 @2 @3 ...

```
#!/bin/bash
# cal-1.sh

a=@1
b=@2

echo "$a + $b"
```

```
chmod 755 cal-1.sh
./cal-1.sh
```

# คำสั่งกำหนดเงื่อนไข

---

- If/Else

# คำสั่งวนรอบ

---

# សរុប

---

- ផ្តល់ជានវបនបបបភិបតិ Multi-user , Multi—task
- សិក្សា (Linux) គ្រប់គ្រង Unix
- គម្រោងផ្តល់ជាន