

BASH shell script ชุดหาากิน

1. ชื่อแฟ้มนามสกุลอะไรก็ได้เช่น .sh หรือไม่มีก็ได้
2. บรรทัดแรกของแฟ้ม shell script ต้องเป็น
#!/bin/bash เสมอ
3. เครื่องหมาย # นำเป็น comment ยกเว้นบรรทัดแรก
4. เครื่องหมาย ; หมายถึงขึ้นบรรทัดใหม่
5. exit # ออกจากโปรแกรม
6. sh myscript.sh # เรียกใช้คำสั่งปรกติทั่วไป
7. sh -x mycript.sh # debug คำสั่ง
8. ./myscript.sh # วิธีสั่งงานอีกแบบ แต่ต้องตั้งค่า
chmod +x myscript.sh

ตัวแปร อักษรเล็ก-ใหญ่ ผลต่างกัน

9. VAR="My Value" # ตั้งค่าตัวแปร
10. echo "\${VAR}" # ผล My Value
11. echo "\${VAR:-0}" # ถ้า VAR ไม่มีข้อมูล ให้แทนด้วย 0
12. echo '\${VAR}' # ผล \${VAR}
13. VAR= # ลบตัวแปร
14. VAR=\$(date) # รับค่าตัวแปรจากคำสั่ง date
15. VAR=`date` # รับค่าตัวแปรจากคำสั่งอีกแบบ
16. sh myscript.sh AA BB # สั่งงานแบบส่งตัวแปร
echo \$1 \$2 # ผล AA BB

standrad file 0=in , 1=out, 2=error

17. ls /tmp > MyFile # เอาผลลัพธ์เข้าแฟ้ม
18. cat /dev/null > MyFile # ลบข้อมูลในแฟ้ม
19. ls /temp 2> MyFile # เอา error เข้าแฟ้ม
20. ls /temp 2> /dev/null # ไม่สนใจ error

21. ls /tmp | wc # ส่งผลลัพธ์ไปให้ wc ทำงานต่อ
if ตรวจสอบ ใช้ได้กับ **test**
22. if ["\${VAR1}" -lt "\${VAR2}"]; then
 ชุดคำสั่ง
fi
ทดสอบตัวเลข -lt น้อยกว่า -gt มากกว่า
-eq เท่ากัน -ne ไม่เท่ากัน -ge มากกว่าหรือเท่ากับ
-le น้อยกว่าหรือเท่ากับ
ทดสอบข้อความ = ข้อความเหมือนกัน != ต่างกัน

23. if [-s "\${VAR}"]; then
 ชุดคำสั่ง
fi
ทดสอบแฟ้ม -s มีแฟ้มและมีข้อมูลด้วย
ทดสอบข้อความ -n มีข้อความ -z ไม่มีข้อความ
ทดสอบไต่เรคเตอรี -d เป็นไต่เรคเตอรี

24. if [...]; then
 ชุดคำสั่ง
else
 ชุดคำสั่ง
fi

while วนรอบตามสถานะ ใช้ได้กับ **test**

25. while [...] ; do
 ชุดคำสั่ง
 continue # วนต่อไป
 break # เลิกวน
done

26. while read LINE ; do
 ชุดคำสั่ง
done < MyFile # อ่านจากแฟ้มเข้าตัวแปร \${LINE}
27. while true ; do
 # วนไม่รู้จบ
done

for วนรอบตามสถานะ

28. VAR="1 2 3"
for ID in \${VAR} ; do
 ชุดคำสั่ง
 continue # วนต่อไป
 break # เลิกวน
done

case ตรวจสอบ

29. case "\${VAR}" in
 AA) ชุดคำสั่ง ;;
 *) ชุดคำสั่ง ;;
esac

read อ่านจากแป้นพิมพ์ ต้องมี Enter ด้วย

30. read MyKey # อ่านเข้าไว้ในตัวแปร \${MyKey}

exit ออกจากโปรแกรม

31. exit 0 # ออกมาด้วยค่า 0

คำสั่งจากโปรแกรม ชุดหาकिन**awk** คัดแยกข้อความ

32. พิมพ์ข้อความเฉพาะช่องที่เลือก

```
ls -l /etc | awk '{print $8}' # พิมพ์ช่องที่ 8
```

33. awk '{print \$1;"\$2}' # พิมพ์คั่นด้วย ;

cut ตัดอักษร

34. แยกตามฟิลด์ ; เลือกฟิลด์ที่ 1

```
echo "${VAR}" | cut -d';' -f 1 | cut -c 2-6
```

35. ตัดอักษรที่ 2 ถึง 6

```
echo "${VAR}" | cut -c 2-6
```

bc คำนวณคณิตศาสตร์

36. COUNT=\$(echo \${COUNT} + 1|bc) # เพิ่มค่าตัวเองทีละ 1

date วันที่เวลา

37. วันที่ในรูปแบบ YYYYmmddHHMM

```
date "+%Y%m%d%H%M"
```

38. วันที่เมื่อ 10 วันก่อนที่ผ่านมา

```
date -d "- 10 day" "+%Y%m%d%H%M"
```

echo สั่งพิมพ์ข้อความ

39. echo "My = \${AA}" # ผล My = ค่าของ AA

40. echo -n "ABC" # พิมพ์ข้อความไม่ขึ้นบรรทัดใหม่

exec ออกไปและเรียกคำสั่งอื่นทำงานต่อ

41. exec sh myscript.sh

grep ตรวจสอบข้อความ

42. คัดเลือกเฉพาะที่มีข้อความ

```
ls -l /etc | grep "network" # มีคำว่า network
```

43. grep -v "network" # ไม่มีคำว่า network

44. grep -w "net" # มีคำว่า net เท่านั้น

45. grep -i "network" # อักษรใหญ่เล็กก็ได้

```
46. grep -c "." # นับบรรทัดของแฟ้ม
```

47. grep -w "^net\$" # ทั้งบรรทัดมีแต่ net

48. grep -m 1 "net" # แสดงผลแค่อันเดียว

49. grep "[0-9]" # ขึ้นต้นด้วยตัวเลข

50. grep "[[:digit:]]\$" # ทั้งบรรทัดมีแต่ตัวเลข

51. grep "^..105" # ขึ้นต้นอักษรอะไรก็ได้ 2 ตัวแล้วตามด้วย 105

head บรรทัดจากบรรทัดแรก

52. ls -l /var | head -2 # เลือก 2 บรรทัดแรก

53. head -n -3 # เลือกตั้งแต่บรรทัดแรก ยกเว้น 3 บรรทัดสุดท้าย

sed เปลี่ยนแปลงข้อความ

54. ls -l / | sed "s/var/ABC" # เปลี่ยน var เป็น ABC

55. sed "s;var;ABC" # ใช้เครื่องหมาย ; ก็ได้

56. sed "s/\[.\D\]/g" # แปลง Esc[D เป็น Return

57. sed "s/\r/\n/g" # แปลง Return เป็น Newline

58. sed "s/^/ABC;/g" # แทรกเพิ่ม ABC หน้าบรรทัด

sleep หยุดรอ

59. sleep 10 # หยุดรอ 10 วินาที

sort เรียงข้อมูล

60. sort \${MYFILE} # จัดเรียงข้อมูลในแฟ้ม

61. sort -t'.' -n -k1,1 -k2,2 -k3,3 -k4,4 # เรียงเลข ip

tail บรรทัดนับจากท้าย

62. ls -l /var | tail -2 # เลือก 2 บรรทัดสุดท้าย

63. tail -n +3 # เลือกตั้งแต่บรรทัดที่ 3 ลงไป

tr แปลงรูปแบบ

64. echo "a"|tr [:lower:] [:upper:] # อักษรตัวเล็กเป็นตัวใหญ่

uniq ข้อมูลซ้ำ

65. sort \${MYFILE} | uniq # จัดเรียงเลือกเฉพาะข้อมูลที่ uniq

66. uniq -d # เลือกเฉพาะข้อมูลที่ซ้ำ

wget ดาวน์โหลดแฟ้ม

67. wget "url" -O /path/MyFile

68. wget "url" -O - | .. # ดาวน์โหลดออกที่ stdout