

EXPECT script ชุดหากิน

1. ชื่อแฟ้มนามสกุลอะไรก็ได้เช่น .exp หรือไม่มีก็ได้
2. บรรทัดแรกของแฟ้ม expect script ต้องเป็น
#!/usr/bin/expect
3. เครื่องหมาย # นำเป็น comment ยกเว้นบรรทัดแรก
4. เครื่องหมาย ; หมายถึงขึ้นบรรทัดใหม่
5. exit # ออกจากโปรแกรม
6. expect myscript.exp # เรียกใช้คำสั่งปรกติทั่วไป
7. ./myscript.exp # วิธีสั่งงานอีกแบบ แต่ต้องตั้งค่า
chmod +x myscript.exp
8. สามารถใช้ TCL (Tool Command Language) ได้

ตัวแปร อักษรเล็ก-ใหญ่ ผลต่างกัน

9. set VAR "My Value" # ตั้งค่าตัวแปร
10. set VAR [timestamp] # ตั้งค่าตัวแปร ผลจากคำสั่ง
11. \$expect_out(buffer) # ค่าที่ได้รับจากการคอยรับข้อมูล
12. expect myscript.exp AA BB CC # สั่งงานแบบส่งตัวแปร
set SERVER [lindex \$argv 0] # ผล AA
set USER [lindex \$argv 1] # ผล BB
set PASSWORD [lindex \$argv 2] # ผล CC

expect รอข้อความ

13. expect # รออะไรก็ได้
14. expect "OK" # รอคำว่า OK
15. expect \$VAR # รอข้อความตัวแปร
16. expect "OK" { ชุดคำสั่ง } # ถ้าพบ OK ให้ทำคำสั่ง

17. ตรวจสอบข้อความ แล้วทำงานเงื่อนไข

```
expect {
  "OK" { ชุดคำสั่ง }
  "NO" { ชุดคำสั่ง }
}
```

for วนรอบตามสถานะ

```
18. for {set VAR 1} {$VAR < 5} {incr VAR 1} {
  ชุดคำสั่ง
  exp_continue # วนต่อไป
  break # เลิกวน
}
```

if ตรวจสอบ

```
19. if { $VAR == 0 } {
  ชุดคำสั่ง
} else {
  ชุดคำสั่ง
}
```

while วนรอบตามสถานะ

```
20. while { $VAR > 0 } {
  ชุดคำสั่ง
  exp_continue # วนต่อไป
  break # เลิกวน
}
```

```
21. while 1 {
  # วนไม่รู้จบ
}
```

log_user การแสดงผล

```
22. log_user 0 # ไม่ต้องแสดงผล
match_max 100000 # ขนาดข้อความที่บันทึก
23. match_max 100000
puts พิมพ์ข้อความ
24. puts "yes"
```

send ส่งข้อความ

```
25. send "hello $VAR "
26. send "hello\r"
27. send \003 # ส่ง Control_C
```

sleep หยุดรอ

```
28. sleep 60
```

spawn แยก process ออกไปทำงาน

```
29. spawn telnet HOST
```

system เรียกให้ระบบทำงานแล้วกลับมา

```
30. system date
```

จัดการแฟ้ม

```
31. set MYFILE [open /tmp/test.txt w] # เปิดแฟ้มเขียน
32. puts $MYFILE "hello" # เขียนแฟ้ม
33. puts $MYFILE $expect_out(buffer) # เขียนแฟ้ม
34. close $MYFILE # ปิดแฟ้ม
```