

第八章 系統行程管理



學習目標

- 了解行程
- 系統行程介紹
- 觀察系統狀態
- 管理背景工作

章節大綱

8-1 行程狀態指令ps

8-3 行程的管理

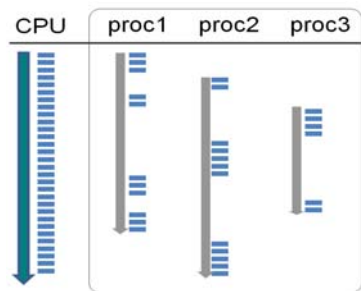
8-2 背景行程

8-4 進階行程管理

備註：可依進度點選小節

系統行程

- 將一顆CPU的可用時間切割為很小的單位
- 多件工作可以分配到CPU 的時間單位
- 每一個行程皆被分配一個唯一的編號，稱為行程編號PID



3

8-1 行程狀態指令ps

- 執行了一個指令或程式時，系統會產生一個新的行程來處理這個指令的工作
- 給予一個PID
- 使用ps指令(Process status)來觀察行程的資訊

```
[root@proj ~]# ps
PID TTY          TIME CMD
10687 pts/1        00:00:00 bash
10711 pts/1        00:00:00 ps
```

4

8-1 行程狀態指令ps

- 加上-f選項來顯示更詳細的資訊。

```
[root@proj ~]# ps -f
UID    PID    PPID  C  STIME TTY   TIME   CMD
root   10687  10685  0  14:18 pts/1  00:00:00 -bash
root   10721  10687   0  14:20 pts/1  00:00:00 ps -f
```

5

8-1 行程狀態指令ps

- 活用行程列表ps

- 系統管理工作中使用ps指令觀察在系統背景中執行的服務
- 使用ps指令加上選項aux

```
[root@fc ~]# ps aux |more
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.1  2008   772 ?        Ss   Feb18    0:02 /sbin/init
root         2  0.0  0.0      0     0 ?        S<   Feb18    0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?        S<   Feb18    0:00 [migration/0]
root         4  0.0  0.0      0     0 ?        S<   Feb18    0:02 [ksoftirqd/0]
root         5  0.0  0.0      0     0 ?        S<   Feb18    0:00 [watchdog/0]
root         6  0.0  0.0      0     0 ?        S<   Feb18    0:09 [events/0]
root         7  0.0  0.0      0     0 ?        S<   Feb18    0:00 [khelper]
root        80  0.0  0.0      0     0 ?        S<   Feb18    0:00 [kintegrityd/0]
```

6

8-1 行程狀態指令ps

● 活用行程列表ps

● 每個欄位代表的意義

欄位名稱	說明
USER	該行程的擁有者
PID	該行程的PID
%CPU	CPU使用率
%MEM	記憶體使用率
VSZ	虛擬記憶體的使用量，以KB為單位
RSS	固定占用的記憶體，以KB為單位
TTY	該行程是由那一個終端機編號所產生的，因為上圖都是系統服務，因此TTY顯示為問號"?"
STAT	行程目前狀態，S代表正處於睡眠中，R代表在執行當中。
START	記錄行程被啟動時的日期
TIME	實際使用CPU的時間
COMMAND	該行程的指令

7

8-1 行程狀態指令ps

● 活用行程列表ps

● 配合使用篩選指令grep，只列出符合服務名稱的該行

```
[root@fc ~]# ps aux |grep crond
root      1878  0.0  0.2  6192 1208 ?        Ss   Feb18   0:05 crond
root      14043 0.0  0.1  5404   692 pts/1    S+   11:02   0:00 grep crond
[root@fc ~]#
```

8

8-1 行程狀態指令ps

● 互動工具top

- top指令是個互動式的行程觀察工具
- 提供功能鍵以切換顯示的資訊

```
top - 11:09:49 up 4 days, 7:23, 2 users, load average: 0.00, 0.00, 0.00
Tasks: 83 total, 1 running, 82 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.0%us, 0.1%sy, 0.0%ni, 99.9%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 514024k total, 363656k used, 150368k free, 48880k buffers
Swap: 610460k total, 0k used, 610460k free, 223184k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
14044	root	20	0	2424	924	736	R	1.8	0.2	0:00.11	top
1	root	20	0	2008	772	564	S	0.0	0.2	0:02.39	init
2	root	15	-5	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	RT	-5	0	0	0	S	0.0	0.0	0:00.00	migration/0
4	root	15	-5	0	0	0	S	0.0	0.0	0:02.41	ksoftirqd/0
5	root	RT	-5	0	0	0	S	0.0	0.0	0:00.00	watchdog/0
6	root	15	-5	0	0	0	S	0.0	0.0	0:10.01	events/0
7	root	15	-5	0	0	0	S	0.0	0.0	0:00.00	khelper

9

8-1 行程狀態指令ps

● 行程之前間的相關性 pstree

- pstree指令能以樹狀結構列出系統行程的清單
- 顯示系統行程間的關係

```
[root@fc ~]# pstree
init-- /usr/bin/sealer
    --NetworkManager
    --acpid
    --anacron
    --atd
    --auditd-- audispd-- {audispd}
        --{auditd}
    --avahi-daemon-- avahi-daemon
    --bluetoothd
    --bonobo-activati-- {bonobo-activati}
```

10

8-2 背景行程

- 正在前端執行的行程稱為「前景行程 (foreground process)」得等到這個子行程結束後，並出現提示字元，才能再執行下一個工作指令
- 使用bash的背景行程控制方式，把子行程放到背景去執行
- 在指令的最後，加上一個符號&

```
[root@proj ~]# updatedb &
[1] 10780
[root@proj ~]#
```

11

8-2 背景行程

- 背景行程練習

```
[root@proj ~]# sleep 60
[1]+  Stopped                  sleep 60
[root@proj ~]# bg
[1]+ sleep 60 &
[root@proj ~]# ps -f
```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	18409	18407	0	22:03	pts/1	00:00:00	-bash
root	18437	18409	0	22:04	pts/1	00:00:00	sleep 60
root	18438	18409	0	22:04	pts/1	00:00:00	ps -f

```
[root@proj ~]#
```

按下 Ctrl+z

觀察背景行程

12

8-2 背景行程

● 觀察背景與前景行程－jobs

● 用 jobs 指令來觀察背景行程資訊

```
[root@proj ~]# sleep 60 &
[1] 18483
[root@proj ~]# sleep 100
[2]+ Stopped                  sleep 100 按下Ctrl+Z
[root@proj ~]# jobs
[1]- Running                  sleep 60 &
[2]+ Stopped                  sleep 100 此行程為停止狀態
[root@proj ~]#
```

13

8-2 背景行程

● 將背景行程取回前景－fg

● 將背景行程sleep 60取回前景

```
[root@proj ~]# jobs
[1] 18507
[root@proj ~]# sleep 60 &
[1]+  Running                  sleep 60 &
[root@proj ~]# fg              將背景行程取回
sleep 60
-
[root@proj ~]#                按下Ctrl+C 中斷行程
```

14

8-3 行程的管理

● 優先權

- 行程有優先權的設計
- 優先順序以編號-20到19來表示，這個數字稱為 nice值(nice value)
- 越低的數字代表行程擁有最高的優先權
- nice值預設通常為0
- Fedora 13的預設nice值為0

```
[root@proj ~]# ps o comm,pid,ni
bash      18409   0
ps        18900   0
[root@proj ~]#
```

nice 值預設為 0

15

8-3 行程的管理

● 指定優先權值

- nice指令可以在執行指令時，用來指定其nice值
- nice指令的-n選項，給定一個正數或負數

```
[root@proj ~]# nice -n -3 sleep 60 &
[1] 18932
[root@proj ~]# ps o comm,pid,ni
COMMAND    PID    NI
bash       18409   4
sleep      18932  -3
ps         18933   4
```

nice 值由預設值 0 減 3 等於 -3

16

8-3 行程的管理

● 刪除行程kill

- kill指令能將目前運作的行程刪除
- kill指令送出訊號至某行程
- 能否結束行程最後還是要看行程本身

```
[root@fc ~]# sleep 30 &          (產生 sleep 30 秒的背景行程)
[1] 2799
[root@fc ~]# ps                  (在系統中執行，PID 為 2799)
  PID TTY          TIME CMD
 2748 pts/0    00:00:00 bash
 2799 pts/0    00:00:00 sleep
 2801 pts/0    00:00:00 ps
[root@fc ~]# kill 2799          (送出結束訊號至 2799)
[1]+  Terminated                  sleep 30      (該行程結束了)
[root@fc ~]# ps                  (行程已不在了)
  PID TTY          TIME CMD
 2748 pts/0    00:00:00 bash
 2802 pts/0    00:00:00 ps
```

17

8-4 進階行程的管理

● 行程優先權序值

- Linux內部為每一個行程定義了「優先權序值PRI (priority)」
- PRI值越小，則該行程越能優先被系統執行
- PRI值會加上NICE值，成為最終的PRI優先權序值

```
[root@fc ~]# sleep 200 &
[1] 14979
[root@fc ~]# ps -l
F S  UID  PID  PPID  C  PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S   0 14925 14920 0   80   0 - 1548 wait pts/1    00:00:00 bash
0 S   0 14979 14925 0   80   0 - 1281 hrtime pts/1    00:00:00 sleep
4 R   0 14980 14925 0   80   0 - 1451 - pts/1    00:00:00 ps
```

PRI 值預設為 80, NICE 值為 0

18

8-4 進階行程的管理

● 調整行程優先權值－renice

- 如果是已在執行中的行程，可以使用renice指令
 - renice NICE值 PID
- 想讓目前的bash指令得到比較高的執行優先權，可以變更目前系統背景的bash行程的NICE值為-15

bash行程PID為14925

```
[root@fc ~]# ps -l
```

F	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	TTY	TIME	CMD
4	S	0	14925	14920	0	80	0	-	1548	wait	pts/1	00:00:00	bash
4	R	0	14987	14925	0	80	0	-	1450	-	pts/1	00:00:00	ps

```
[root@fc ~]# renice -15 14925
```

14925: old priority 0, new priority -15

將bash行程NICE值改為-15

```
[root@fc ~]# ps -l
```

F	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	TTY	TIME	CMD
4	R	0	14925	14920	0	65	-15	-	1548	-	pts/1	00:00:00	bash
4	R	0	14989	14925	0	65	-15	-	1451	-	pts/1	00:00:00	ps

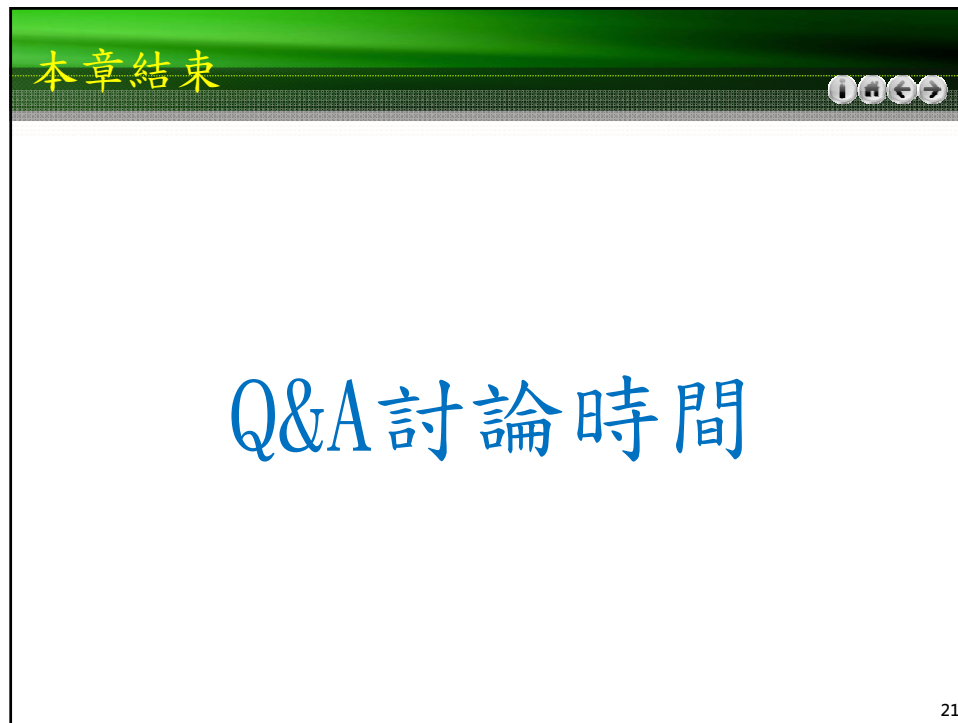
19

8-4 進階行程的管理

● 行程資訊目錄/proc

- 行程資訊目錄/proc記錄系統運作時期時內部核心的狀態
- 目錄內，可看到每一個正在執行的行程的資訊
- /proc目錄中的虛擬檔案並不是一般檔案，使用者不能編輯虛擬檔案
- 系統為了將內部資訊提供給使用者的一個方法
- 系統能夠「展示」系統內部的現況

20



本章結束

Q&A討論時間

21