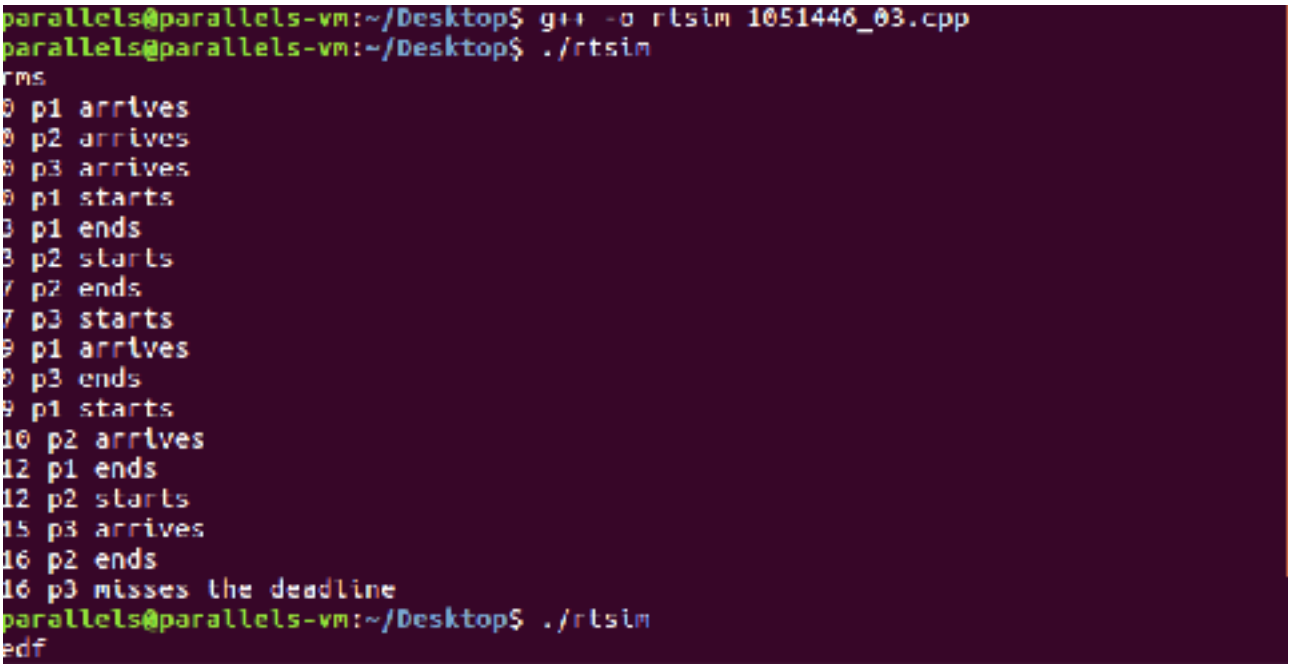
作業系統概論 Prog. #3 Real-time Process Scheduling

1051446游采蓉

- 輸出結果：

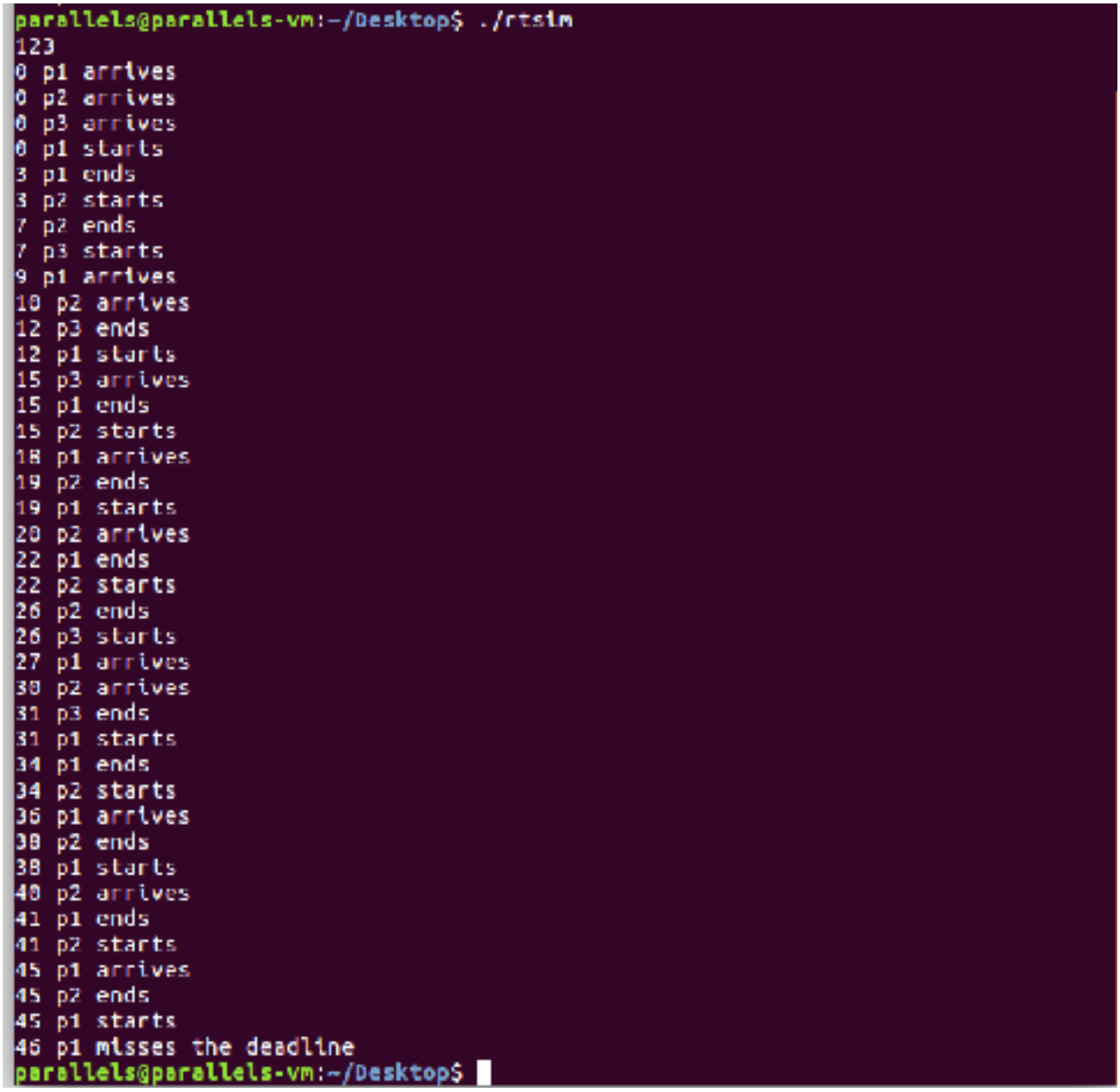
輸入rms



輸入edf

```
parallels@parallels-vm:~/Desktop$ ./rtsim
edf
0 p1 arrives
0 p2 arrives
0 p3 arrives
0 p1 starts
3 p1 ends
3 p2 starts
7 p2 ends
7 p3 starts
9 p1 arrives
10 p2 arrives
12 p3 ends
12 p1 starts
15 p3 arrives
15 p1 ends
15 p2 starts
18 p1 arrives
19 p2 ends
19 p1 starts
20 p2 arrives
22 p1 ends
22 p2 starts
26 p2 ends
26 p3 starts
27 p1 arrives
30 p2 arrives
31 p3 ends
31 p1 starts
34 p1 ends
34 p2 starts
36 p1 arrives
38 p2 ends
38 p1 starts
40 p2 arrives
41 p1 ends
41 p2 starts
45 p1 arrives
45 p2 ends
45 p1 starts
46 p1 misses the deadline
parallels@parallels-vm:~/Desktop$ ./rtsim
123
```

隨機輸出rms 或edf



■程式碼解說：

a.能順利以RMS排程 2 ≤ n ≤ 5 個 processes，可輸出從0~300 ms的模擬結果

在 rms_process 的 structure裡，存放

```
int pid;
int arrival_time;
int CPU_burst;
int deadline;
int period;
bool arrive;
```

分別存放pid，arrival_time，CPU_burst，deadline，period

bool arrive 是存放他是否已經到了 ，到了則改狀態為true

一開始先sort依照period為排放順序，current_process為現在目前在執行的process
miss是用在紀錄是否已經有process miss掉deadline如果是就設為true，
process_end是在記錄上一個process執行完後的時間

用一個for loop去模擬跑每個時間的排程狀況，跑到有miss deadline或是時間到了

首先 跑for loop 去看這個時間有哪些 process已經抵達 ，然後再跑去檢查看有沒有process已經抵達，且他抵達時間小於目前時間，若成立之後再來判斷，他時間剛好是抵達，要輸出
Starts， 不過在那之前如果有其他的process在執行，就要先讓他結束，才能開始，優先權高的process先執行。

如果已經有process在執行，且執行完了，輸出process ends，並記錄他目前執行完的時間到process_end， 並將這個執行完的process的arrival_time加上period，deadline 也要改成arrival_time + period，最後要把arrive 設成 false。並且找到下一個要開始的process並讓他開始執行。

最後檢查是不是有process miss deadline 如果有就將bool形態的miss設為true然後跳出迴圈結束模擬。

b. 能順利以EDF排程 2 ≤ n ≤ 5 個 processes，可輸出從0~300 ms的模擬結果

edf則是跟rms有類似概念，不過要在每做完一個process 就執行一次sort檢查優先權，而edf優先權是看誰最靠近就先做。

c. 可隨意顯示0~300 ms 其中任何一段時間兩個演算法的排程模擬結果

若隨意輸入非edf或rmf的值，則隨機產生其中一個排程