



OPERATING SYSTEM

LAB 04 | ĐỊNH THỜI TIẾN TRÌNH

Thực hiện mô phỏng các giải thuật định thời tiến trình, qua đó người học sẽ hình dung rõ hơn về cách Hệ điều hành tổ chức khối PCB cũng như cách thực hiện các tiêu chí định thời khác nhau dưới góc độ lập trình.



MỤC TIÊU

Sau LAB này, sinh viên sẽ có thể:

1. Mô phỏng các giải thuật định thời FCFS, SJF, SRTF, và Round Robin
2. Tính toán các chỉ số tiêu chuẩn định thời dưới góc độ người dùng



NỘI DUNG

THỰC HÀNH

1. Hiện thực giải thuật FCFS
2. Hiện thực các giải thuật còn lại



HIỆN THỰC GIẢI THUẬT FCFS

1.1. Phân tích bài toán

01.



1.1. Phân tích bài toán

Cho tập tiến trình sau:

Process	Arrival	Burst
P1	0	6
P2	2	7
P3	5	8
P4	9	3
P5	12	6

Tính các thông số:

- Thời gian chờ đợi trung bình
- Thời gian đáp ứng trung bình
- Thời gian hoàn thành trung bình



1.1. Phân tích bài toán

Process	Arrival	Burst
P1	0	6
P2	2	7
P3	5	8
P4	9	3
P5	12	6

?

$$RT(P_i) = Start(P_i) - Arrival(P_i)$$

Thời gian đáp ứng trung bình:

$$ART = \frac{1}{5} \sum_{i=1}^5 RT(P_i) = \frac{RT(P_1) + RT(P_2) + RT(P_3) + RT(P_4) + RT(P_5)}{5}$$



1.1. Phân tích bài toán

Process	Arrival	Burst
P1	0	6
P2	2	7
P3	5	8
P4	9	3
P5	12	6

?

$$TaT(P_i) = Finish(P_i) - Arrival(P_i)$$

Thời gian hoàn thành trung bình:

$$ATaT = \frac{1}{5} \sum_{i=1}^5 TaT(P_i) = \frac{TaT(P_1) + TaT(P_2) + TaT(P_3) + TaT(P_4) + TaT(P_5)}{5}$$



1.1. Phân tích bài toán

Process	Arrival	Burst
P1	0	6
P2	2	7
P3	5	8
P4	9	3
P5	12	6

$$WT(P_i) = TaT(P_i) - Burst(P_i)$$



Thời gian chờ đợi trung bình:

$$AWT = \frac{1}{5} \sum_{i=1}^{5} WT(P_i) = \frac{WT(P_1) + WT(P_2) + WT(P_3) + WT(P_4) + WT(P_5)}{5}$$



1.1. Phân tích bài toán

Với mỗi tiến trình:

INPUT	OUTPUT
PID	Start time
Arrival Time	Finish time
Burst Time	Waiting time
	Response time
	Turnaround time

OUTPUT cho toàn bài:

Giản đồ Gantt	Average Waiting time
	Average Turnaround time



HIỆN THỰC GIẢI THUẬT FCFS

1.2. Xây dựng ý tưởng

01.

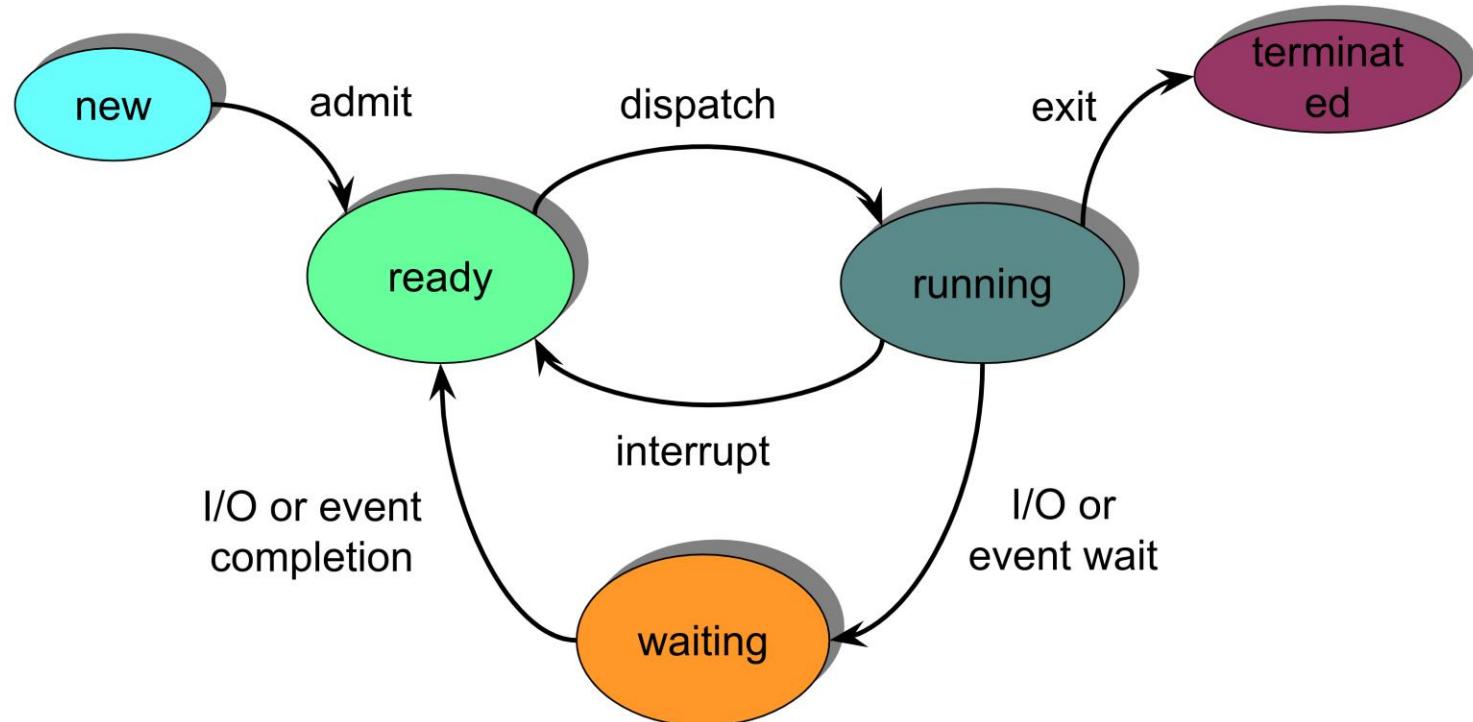


1.2. Xây dựng ý tưởng

- Hệ điều hành quản lý tiến trình thông qua các PCB
→ Mô phỏng **PCB**

```
typedef struct {  
    int iPID;  
    int iArrival, iBurst;  
    int iStart, iFinish,  
        iWaiting, iResponse,  
        iTaT;  
} PCB;
```

1.2. Xây dựng ý tưởng





1.2. Xây dựng ý tưởng

Input[5]	ReadyQueue[5]	TerminatedArray[5]
Pi (Arrival, Burst)		
P1 (0, 6)		
P2 (2, 7)		
P3 (5, 8)		
P4 (9, 3)		
P5 (12, 6)		
iRemain = 5	iReady = 0	iTerminated = 0



1.2. Xây dựng ý tưởng

Input[5] Pi (Arrival, Burst)	ReadyQueue[5]	TerminatedArray[5]
P2 (2, 7)	P1 (0, 6)	
P3 (5, 8)		
P4 (9, 3)		
P5 (12, 6)		
iRemain = 4	iReady = 1	iTerminated = 0

P1

0

6



1.2. Xây dựng ý tưởng

Input[5]	ReadyQueue[5]	TerminatedArray[5]
Pi (Arrival, Burst)		
P2 (2, 7)	P1 (0, 6)	
P3 (5, 8)		
P4 (9, 3)		
P5 (12, 6)		
iRemain = 4	iReady = 1	iTerminated = 0

P1

6

0



1.2. Xây dựng ý tưởng

Input[5] Pi (Arrival, Burst)	ReadyQueue[5]	TerminatedArray[5]
P4 (9, 3)	P1 (0, 6)	
P5 (12, 6)	P2 (2, 7)	
	P3 (5, 8)	
iRemain = 2	iReady = 3	iTerminated = 0

P1

0

6



1.2. Xây dựng ý tưởng

Input[5] Pi (Arrival, Burst)	ReadyQueue[5]	TerminatedArray[5]
P4 (9, 3)	P2 (2, 7)	P1 (0, 6)
P5 (12, 6)	P3 (5, 8)	
iRemain = 2	iReady = 2	iTerminated = 1

P1

0

6



1.2. Xây dựng ý tưởng

Input[5] Pi (Arrival, Burst)	ReadyQueue[5]	TerminatedArray[5]
P4 (9, 3)	P2 (2, 7)	P1 (0, 6)
P5 (12, 6)	P3 (5, 8)	
iRemain = 2	iReady = 2	iTerminated = 1



0

6

13



1.2. Xây dựng ý tưởng

Input[5]	ReadyQueue[5]	TerminatedArray[5]
Pi (Arrival, Burst)		
	P2 (2, 7)	P1 (0, 6)
	P3 (5, 8)	
	P4 (9, 3)	
	P5 (12, 6)	
iRemain = 0	iReady = 4	iTerminated = 1



1.2. Xây dựng ý tưởng

Input[5]	ReadyQueue[5]	TerminatedArray[5]
Pi (Arrival, Burst)		
	P3 (5, 8)	P1 (0, 6)
	P4 (9, 3)	P2(2, 7)
	P5 (12, 6)	
iRemain = 0	iReady = 3	iTerminated = 2





1.2. Xây dựng ý tưởng

Input[5]	ReadyQueue[5]	TerminatedArray[5]
Pi (Arrival, Burst)		
	P3 (5, 8)	P1 (0, 6)
	P4 (9, 3)	P2(2, 7)
	P5 (12, 6)	
iRemain = 0	iReady = 3	iTerminated = 2



0

6

13

21



1.2. Xây dựng ý tưởng

Input[5]	ReadyQueue[5]	TerminatedArray[5]
Pi (Arrival, Burst)		
	P4 (9, 3)	P1 (0, 6)
	P5 (12, 6)	P2 (2, 7)
		P3 (5, 8)
iRemain = 0	iReady = 2	iTerminated = 3





1.2. Xây dựng ý tưởng

Input[5]	ReadyQueue[5]	TerminatedArray[5]
Pi (Arrival, Burst)		
	P4 (9, 3)	P1 (0, 6)
	P5 (12, 6)	P2 (2, 7)
		P3 (5, 8)
iRemain = 0	iReady = 2	iTerminated = 3
<p>P1 P2 P3 P4</p> <p>0 6 13 21 24</p>		

1.2. Xây dựng ý tưởng

Input[5]	ReadyQueue[5]	TerminatedArray[5]
Pi (Arrival, Burst)		
	P5 (12, 6)	P1 (0, 6)
		P2 (2, 7)
		P3 (5, 8)
		P4 (9, 3)
iRemain = 0	iReady = 1	iTerminated = 4



1.2. Xây dựng ý tưởng

Input[5]	ReadyQueue[5]	TerminatedArray[5]			
Pi (Arrival, Burst)					
	P5 (12, 6)	P1 (0, 6)			
		P2 (2, 7)			
		P3 (5, 8)			
		P4 (9, 3)			
iRemain = 0	iReady = 1	iTerminated = 4			
P1	P2	P3			
0	6	13	21	24	30

1.2. Xây dựng ý tưởng

Input[5]	ReadyQueue[5]	TerminatedArray[5]
Pi (Arrival, Burst)		P1 (0, 6)
		P2 (2, 7)
		P3 (5, 8)
		P4 (9, 3)
		P5 (12, 6)
iRemain = 0	iReady = 0	iTerminated = 5



0 6 13 21 24 30



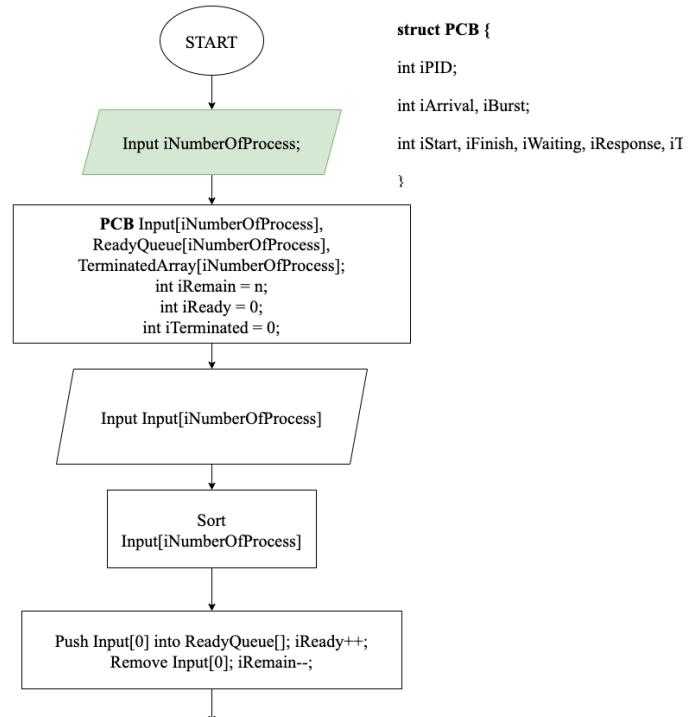
HIỆN THỰC GIẢI THUẬT FCFS

1.3. Lưu đồ giải thuật

01.



1.3. Lưu đồ giải thuật



Gợi ý lưu đồ giải thuật
định thời FCFS



1.3. Lưu đồ giải thuật

Bài tập tại lớp: 01

Hoàn thiện lưu đồ giải thuật FCFS

Bài tập tại lớp: 02

Hiện thực mô phỏng giải thuật FCFS dựa trên lưu đồ đã vẽ trên



BÀI TẬP

02.



2. Bài tập

Cho các yêu cầu sau:

1. Vẽ lưu đồ giải thuật
2. Chạy thủ công một test case (theo quy trình đã trình bày) dựa trên lưu đồ
3. Thực hiện code cho giải thuật, trong đó, Arrival Time của mỗi tiến trình được tạo ngẫu nhiên trong đoạn [0, 20], Burst Time của mỗi tiến trình được tạo ngẫu nhiên trong đoạn [2, 12]
4. Kiểm tra chương trình trên ít nhất 03 test case, mỗi test case 05 tiến trình, so sánh kết quả chạy thủ công và chạy code cho mỗi test case



2. Bài tập

Bài tập: 01

Thực hiện các yêu cầu trên với giải thuật SJF

Bài tập: 02

Chọn một trong hai giải thuật SRTF và Round Robin và thực hiện các yêu cầu trên



2. Bài tập

Bonus: 01

Thực hiện các yêu cầu trên với giải thuật còn lại trong Bài tập 2